Primjena metode stabla odlučivanja za donošenje odluka u bankarskom sektoru

Ercegović, Marin

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:646969

Rights / Prava: Attribution-NonCommercial-NoDerivs 3.0 Unported/Imenovanje-Nekomercijalno-Bez prerada 3.0

Download date / Datum preuzimanja: **2024-03-28**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Marin Ercegović

PRIMJENA METODE STABLA ODLUČIVANJA ZA DONOŠENJE ODLUKA U BANKARSKOM SEKTORU

DIPLOMSKI RAD

Varaždin, 2020.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Marin Ercegović

Matični broj: 46380/17-R

Studij: Organizacija poslovnih sustava

PRIMJENA METODE STABLA ODLUČIVANJA ZA DONOŠENJE ODLUKA U BANKARSKOM SEKTORU

DIPLOMSKI RAD

Mentor/Mentorica:

Izv. prof. dr. sc. Nina Begičević Ređep

Varaždin, rujan 2020.

Marin Ercegović

Izjava o izvornosti

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

---

Sažetak

Zbog turbulentne prirode i uvjeta poslovanja modernih poduzeća naglašava se važnost procesa odlučivanja te se sve više resursa usmjerava u poboljšavanje istog. Banke i druge financijske organizacije moraju na adekvatan način odgovoriti na sve zahtjeve brzo-promjenjive i turbulentne suvremene poslovne okoline. Iz tog razloga daje se veliki značaj načinu vođenja, odnosno odlučivanja u organizaciji te više no ikad do izražaja dolaze sposobnosti i različiti stilovi vođenja i odlučivanja menadžera. Zbog naglaska na efikasnost i brzinu u procesu odlučivanja te postojanje rizika i nesigurnosti u poslovnoj okolini, menadžeri sve više koriste moderne metode za pomoć u odlučivanja poput: analize osjetljivosti, matrice rizika te stablo odlučivanja. Stablo odlučivanja kao metoda za odlučivanje u uvjetima rizika i nesigurnosti predstavlja praktični grafičko-vizualizacijski alat za donošenje jednokratnih sekvencijalnih odluka. Osim toga, bankarske i druge financijske institucije u današnje vrijeme barataju velikim količinama podataka te im je više nego ikada potreban adekvatan odgovor na probleme modernog poslovanja. Tradicionalni sustavi potpore odlučivanja temeljeni na pravilima postaju nedostatni te iziskuju veliku količinu intervencija od strane analitičara. Strojno učenje i povezani prediktivni modeli, poput stabla odlučivanja, nasumične šume i pojačavanja, predstavljaju adekvatnu podlogu za izradu sustava za donošenje kvalitetnih i pravovremenih odluka. Kao glavni slučajevi korištenja strojnog učenja u bankarskom sektoru ističu se procjena kreditnog rizika i detekcija prijevare. U ovom radu se objašnjava teoretska i matematička pozadina metode stabla odlučivanja i povezanih modela koji se na njoj temelje. Na primjerima iz prakse objašnjavaju se koristi od upotrebe stabla odlučivanja u svrhu procjene kreditnog rizika i detekcije prijevare. Naposljetku predstavlja se vlastita implementacija sustava detekcije prijevare na temelju sintetičkoga skupa podataka pomoću prediktivnih modela stabla odlučivanja te je izrađena komparativna analiza performansi modela stabla odlučivanja, nasumične šume i pojačavanja u svrhu detekcije financijske prijevare.

Ključne riječi: odlučivanje; stilovi odlučivanja; nesigurnost i rizik; strojno učenje; bankarski sektor; stablo odlučivanja; detekcija prijvare; nasumična šuma

Sadržaj

1. Uvod 1

2. Metode i tehnike rada 3

3. Odlučivanje 4

3.1. Vrste odluka 5

3.2. Stilovi odlučivanja 7

3.2.1. Autokratski i demokratski stil odlučivanja 7

3.2.2. Vroom-Yetton-Jagov model odlučivanja 8

3.2.3. Ostali stilovi odlučivanja 10

3.3. Načini odlučivanja 12

3.3.1. Programirano i neprogramirano odlučivanje 12

3.3.2. Individualno i skupno odlučivanje 13

3.3.3. Intuitivno i racionalno odlučivanje 16

3.3.4. Odlučivanje s obzirom na okolnosti odlučivanje 17

4. Odlučivanje u uvjetima nesigurnosti i rizika 19

4.1. Nesigurnost i rizik općenito 19

4.1.1. Rizik 20

4.1.2. Nesigurnost 20

4.1.3. Procjena rizika i nesigurnosti 21

4.1.4. Donošenje odluka u jednoj fazi 22

4.2. Kriteriji odlučivanja u uvjetima nesigurnosti i rizika 22

4.2.1. Maksimin kriterij (Waldov kriterij) 24

4.2.2. Maksimaks kriterij 25

4.2.3. Hurwitzov kriterij 25

4.2.4. Kriterij minimalnog žaljenja (Savageov kriterij) 26

4.2.5. Praktičan pristup 27

4.2.6. Laplaceov kriterij 28

4.3. Praktične metode za odlučivanje u uvjetima nesigurnosti i rizika 28

4.3.1. Stabo odlučivanja 29

4.3.2. Matrica rizika 32

4.3.3. Analiza osjetljivosti 32

5. Stablo odlučivanja 34

5.1. Uvod u stablo odlučivanja 34

5.2. Elementi stabla odlučivanja . . . 35
5.3. Koraci metode stabla odlučivanja . . . 36
5.4. Programska podrška za metodu stabla odlučivanja . . . 37
5.4.1. TreePlan . . . 37
5.4.2. PrecisionTree . . . 38
5.4.3. SilverDecisions . . . 39
5.5. Primjer primjene metode stabla odlučivanja . . . 40
5.5.1. Definiranje problema odlučivanja: . . . 40
5.5.2. Izgradnja logičkog modela stabla odlučivanja . . . 42
5.5.3. Računanje očekivanih vrijednosti odluka uz pomoć postupka računanja unatrag . . . 44
5.5.4. Pronalaženje optimalnog puta postupkom računanja unaprijed . . . 44
5.5.5. Pohranjivanje modela stabla odlučivanja uz pomoć SilverDecisions alata . 46

6. Uvod u strojno učenje . . . 47
6.1. Pregled strojnog učenja . . . 48
6.1.1. Nadzirano učenje . . . 49
6.1.2. Evaluacija performansi nadziranog učenja . . . 51
6.1.3. Nenadzirano učenje . . . 53
6.2. Metodologija strojnog učenja . . . 56
6.2.1. Metodologija nadziranog učenja . . . 56
6.2.2. Metodologija nenadziranog učenja . . . 57
6.3. Strojno učenje u bankarskom sektoru . . . 58
6.3.1. Detekcija anomalija . . . 58
6.3.2. Ostali slučajevi korištenja strojnog učenja u bankarskom sektoru . . . 61

7. Metodologija stabla odlučivanja . . . 65
7.1. Indukcija stabla odlučivanja . . . 66
7.1.1. ID3 . . . 68
7.1.2. C4.5 . . . 68
7.1.3. CART . . . 69
7.2. Tipovi stabla odlučivanja . . . 69
7.2.1. Regresijska stabla . . . 69
7.2.2. Klasifikacijska stabla . . . 71
7.3. Kriterij zaustavljanja . . . 73
7.4. Podrezivanje stabla . . . 74
7.5. Napredne tehnike konstrukcije stabla odlučivanja . . . 76
7.5.1. Pakiranje . . . 76
7.5.2. Nasumična šuma . . . 77
7.5.3. Pojačavanje . . . 78

8. Implementacija sustava detekcija prijevare temeljem sintetičkog skupa podataka 80
8.1. Dobavljanje i analiza podataka . . . 81

8.1.1. Dobavljanje podataka . . . 81
8.1.2. Analiza podataka . . . 82
8.2. Čišćenje podataka . . . 86
8.2.1. Inženjering atributa . . . 86
8.2.2. Zamijena kategoričkih atributa . . . 87
8.2.3. Podjela skupa podataka . . . 88
8.2.4. Standardizacija podataka . . . 88
8.2.5. Balansiranje podataka . . . 88
8.3. Građenje i obuka modela . . . 89
8.3.1. Sklearn - DecisionTreeClassifier . . . 89
8.3.2. Sklearn - RandomForestClassifier . . . 89
8.3.3. xgboost.sklearn - XGBClassifier . . . 90
8.4. Testiranje modela . . . 90
8.4.1. Generiranje performansi - sklearn - DecisionTreeClassifier . . . 91
8.4.2. Generiranje performansi - sklearn - RandomForestClassifier . . . 91
8.4.3. Generiranje performansi - xgboost.sklearn - XGBClassifier . . . 93
8.4.4. Usporedba performansi modela . . . 93

9. Primjeri upotrebe stabla odlučivanja u bankarskom sektoru . . . 95
9.1. Primjeri iz strojnog učenja . . . 95
9.1.1. Procjena kreditnog rizika: primjena metode stabla odlučivanja u ruralnoj banci PT BPR X . . . 95
9.1.2. Komparativna analiza algoritama stabla odlučivanja za detekciju prijevara putem kreditnih kartica . . . 98
9.2. Primjeri iz teorije odlučivanja . . . 101
9.2.1. Odabir banke prilikom ugovaranja poslovnog kredita . . . 101
9.2.2. Odabir poduzeća za investiciju od strane fonda rizičnog kapitala . . . 106

10. Zaključak . . . 109

Popis literature . . . 115

Popis slika . . . 118

1. Uvod

Odlučivanje je vrlo širok pojam koji označuje najjednostavniji izbor poput odabira odjeće za svaki dan, ali i donošenje važnijih odluka kao što su npr. odabir životnog partnera. Pojam poslovno odlučivanje sve više zaokuplja pozornost ne samo menadžera kao osnovnih nositelja i subjekata poslovnog odlučivanja već i znanstvene i stručne javnosti [1].

Stilovi odlučivanja u najužoj su vezi sa stilovima vodstva, odnosno stilovi odlučivanja menadžmenta su u najvećoj mjeri podudarni s stilovima vodstva jer se njihov posao, u najvećoj mjeri, svodi na donošenje odluka, odnosno odlučivanje. U teoriji i praksi vodstva stilovi odlučivanja se u pravili kreću između dvije krajnosti, odnosno između autokratskog i demokratkskog stila odlučivanja [1]. Nadalje, u radu se definiraju vrste odluka s različitih aspekata situacija odlučivanja i samih osoba koje donose odluke.

Osim toga, u radu se obrađuje proces odlučivanja u situacijama nesigurnosti i rizika. Odlučivanje u uvjetima rizika podrazumijeva odlučivanje u okolnostima u kojima rezultati nisu sigurni, ali su poznate vjerojatnosti postizanja različitih rezultata [1]. Nesigurnost je okolnost s kojom se menadžeri u suvremenim organizacijama susreću na dnevnoj bazi, a uzrokovana je dinamikom, nestabilnošću i rapidnim promjenama koje se dešavaju u modernim organizacijama [1]. Obrađene metode za odlučivanje u uvjetima nesigurnosti i rizika su:

- Stablo odlučivanja,
- Matrica rizika,
- Analiza osjetljivosti.

Metoda stabla odlučivanja je objašnjena u radu iz dva aspekta:

1. teorije odlučivanja,
2. strojnog učenja.

Kada o stablu odlučivanja govorimo s aspekta teorije odlučivanja, stablo odlučivanja (engl. *Decision tree*) je grafički model za vizualizaciju procesa odlučivanja kad se rješavanje problema odlučivanja svodi na donošenje više sukcesivnih odluka te se primjenjuje u donošenju kompleksnih odluka. Uz stablo odlučivanja vežemo postupak računa očekivane vrijednosti [1]. Osim teorijskog aspekta, prikazan je i postupak korištenja metode kroz pokazni primjer, kao i povezana programska podrška za korištenje metode.

Strojno učenje (engl. *Machine learning*) važan je aspekt modernog poslovanja i istraživanja. Strojno učenje pomaže računalnim sistemima da progresivno poboljšavaju svoje performanse korištenjem različitih algoritama i modela. Algoritmi strojnog učenja automatski grade matematičke modele koji se kasnije koriste za donošenje odluka bez da su eksplicitno programirani za tu namjenu [2]. Strojno učenje dijeli se u dva glavna podtipa: nadzirano i nenadzirano učenje. Nadzirano učenje je najraširenij oblik strojnog učenja, a koristi se za riješavanje problema klasifikacije i regresije [3].

Institucije bankarskog sektora danas više nego ikada iziskuju brzinu i točnost u procesu odlučivanja i potpore odlučivanja. Iz tog razloga napušta se zastarjela paradigma temeljena na sustavima pravila te se koriste napredni sustavi za potporu u odlučivanju u obliku prediktivnih modela baziranih na strojnom učenju. Detekcija prijevara (engl. *Fraud detection*) najjasniji je slučaj detekcije anomalija (engl. *Anomaly detection*) zasnovanih na strojnom učenju ili umjetnoj inteligenciji u bankarskom sektoru. Nadalje, tradicionalni sustavi bazirani na pravilima postaju nedostatni te često rezultiraju u stopama lažno pozitivnih vrijednosti (engl. *False Positive, FP*) koje prelaze 90%. To stvara ogroman broj lažno pozitivnih upozorenja koja zatim moraju biti očišćena putem ljudskih intervencija [4]. Detekcija prijevare predstavlja klasifikacijski problem nadziranog strojnog učenja, a za potrebe rješavanja istog u ovom radu fokus je na metodama:

- stablo odlučivanja (engl. *Decision tree*),
- nasumična šuma (engl. *Random forest*),
- pojačavanje (engl. *Boosting*),

te je objašnjena njihova teoretska i matematička osnova. Osim detekcije prijevare obrađuje se i niz drugih slučajeva korištenja strojnog učenja u bankarstvu poput procjene kreditnog rizika. Na temelju povezanih akademskih članaka predstavljena je implementacija takvih sustava na konkretnim poslovnim primjerima.

U radu je objašnjena izrada vlastitog programskog rješenja za detekciju financijske prijevare koje se temelji na prethodno spomenutim metodama i tehnikama strojnog učenja. Za potrebe implementacije sustava detekcije prijevare, odnosno stabla odlučivanja iz područja strojnog učenja koristi se programski jezik Python3 i sintetički skup podataka o mobilnim financijskim transakciijama Paysim.

Zadnji dio rada sadrži realne primjere upotrebe metode stabla odlučivanja u bankarstvu, odnosno predstavljena su dva primjera iz strojnog učenja te dva primjera uporabe metode stabla odlučivanja iz područja teorije odlučivanja. Primjeri iz strojnog učenja odnose se na procjenu kreditnog rizika i detekciju privjera različitim algoritmima stabla odlučivanja. Nadalje, primjeri iz teorije odlučivanja odnose se na korištenje metode prilikom podizanja kredita i odabira kompanije za investiranje od strane fonda rizičnog kapitala. U samom zaključku rada predstavljena je usporedba i komparativna analiza stabla odlučivanja iz područja teorije odlučivanja i područja strojnog učenja.

2. Metode i tehnike rada

Za obradu teoretskog dijela i razrade teme u radu se koriste primarni i sekundarni izvori literature. Literatura koja je korištena u radu je uglavnom u obliku stručnih knjiga na temu odlučivanja teorije odlučivanja, strojnog učenja i stabla odlučivanja te povezanih internetskih članaka. Primjeri iz prakse objašnjeni su temeljem povezanih akademskih članaka na temu primjene stabla odlučivanja u bankarskom sektoru. Za potrebe prikaza praktičnog provođenja metode stabla odlučivanja iz sfere teorije odlučivanja, kao i za izradu povezanih primjera iz bankarskog sektora koristi se programski alat *SilverDecisions*. Za potrebe izrade vlastitog primjera implementacije sustava detekcije prijevare, odnosno primjenu stabla odlučivanja u sferi strojnog učenja koristi se programski jezik *Python3* s povezanim programskim paketima, odnosno bibliotekama:

- *Pandas* - brzi, snažan, jednostavan alat za analizu i manipulaciju podataka razvijen za *Python* programski jezik,
- *Matplotlib* - Sveobuhvatna programska biblioteka za stvaranje statičkih, animiranih i interaktivnih vizualizacija u *Pythonu*,
- *Numpy* - fundamentalni paket za znanstveno računanje s *Pythonom*,
- *Seaborn* - *Python* alat za vizualizaciju podataka baziran na *matplotlib* paketu koji pruža mogućnosti informativnih statističkih grafova,
- *Sklearn* - biblioteka koja pruža jednostavne i efikasne alate za prediktivnu podatkovnu analizu bazirana na *NumPy*, *SciPy* i *Matplotlib* paketima,
- *Xgboost* - skalabilna, mobilna biblioteka za implementaciju metode pojačavanje gradijenta (engl. *Gradient boosting*),
- *Imblearn* - paket za balansiranje neuravnoteženih skupova podataka.

Kao podloga za izradu prediktivnih modela korišten je sintetički skup podataka o mobilnim financijskim transakcija *Paysim* koji je javno dostupan i preuzet s internetske platforme za objavu javno dostupnih skupova podataka: *www.kaggle.com*.

3. Odlučivanje

Prema Sikavici, Hunjaku i Begičević Ređep i dr. [1] pojam **odlučivanje** je u najužoj vezi s pojmovima kao što su upravljanje, rukovođenje i menadžment. Prema Mesconu, Alberu i Khedouri (kao što se citira u [1]) **odlučivanje** je vrlo širok pojam koji označuje najjednostavniji izbor poput odabira odjeće za svaki dan, ali i donošenje važnijih odluka kao što su npr. odabir životnog partnera. Pojam **poslovno odlučivanje** sve više zaokuplja pozornost ne samo menadžera, kao osnovnih nositelja i subjekata poslovnog odlučivanja, već i znanstvene i stručne javnosti. Nadalje, za bavljenje poslovnim odlučivanjem, odnosno općenito odlučivanjem pojam odlučivanje je potrebno smjestiti unutar integralnog procesa upravljanja, ali je isto tako potrebno utvrditi i odnose na relaciji:

- upravljanje i odlučivanje,
- rukovođenje i odlučivanje,
- menadžment i odlučivanje.

Odlučivanje kao pojam je u vrlo širokoj upotrebi pa se tako može govoriti o odlučivanju na različitim područjima rada i života, kao npr. [1]:

- odlučivanje u osobnom životu,
- odlučivanje u poduzećima ,
- odlučivanje u organizacijama,
- odlučivanje u široj zajednici (država i njezine institucije).

"Za razliku od odlučivanja u osobnom životu, u svim drugim situacijama može se govoriti o poslovnom odlučivanju, bez obzira je li riječ o odlučivanju u poduzeću, banci, školi, vladi, ministarstvima ili primjerice Hrvatskom saboru" [1, str. 3]. U poslovnom odlučivanju ključnu ulogu igraju menadžeri te je menadžersko odlučivanje najvažniji dio poslovnog odlučivanja. Iako u poslovnom svijetu najvažnije odluke donose upravljači, a neke operativne odluke donose izvršitelji, najveći postotak najvažnijih odluka donose menadžeri. Kada se govori o poslovnom odlučivanju, odnosno odnosu upravljanja, rukovođenja, menadžmenta i odlučivanja, nužno je ukazati na razlike između navedenih funkcija u sferi poslovanja, posebno zbog odnosa odlučivanja prema tim funkcijama [1]. Svaka organizacija, bez obzira na svoju veličinu, djelatnost ili orijentiranost, razlikuje tri vrste funkcija:

- poslovne funkcije - istraživanje i razvoj, nabava, upravljanje ljudskim resursima, proizvodnja, prodaja i financije,
- funkcije koje proizlaze iz položaja u procesu rada - upravljanje, rukovođenje/menadžment i izvršenje,

- odlučivanje - ova funkcija označuje na koji način se ostvaruju ostale funkcije - poslovne, upravljačke i menadžerske odnosno izvršne.

To znači da se donošenjem odluka, odnosno odlučivanjem ostvaruju sve funkcije u organizaciji te iz tog razloga poslovno odlučivanje dobiva istaknuto mjesto u svakoj organizaciji [1].

3.1. Vrste odluka

Odlučivanje je proces koji traje određeno vrijeme, a završava donošenjem **odluke**. Trajanje procesa odlučivanja zavisi o vrsti odluke, a kreće se u rasponu od djelića sekunde pa do dugotrajnijeg procesa koji se može mjeriti mjesecima i godinama. **Odluka** se definira kao izbor između alternativa, odnosno izbor jedne ili više alternativnih mogućnosti djelovanja pri rješavanju problema. Odluka je izbor između više pravaca djelovanja orijentiranih na ostvarenje cilja, odnosno donošenje optimalnog izbora. Odluka se najjednostavnije može definirati kao izbor između više inačica [1]. Prema Simon (kao što se citira u [1]) odluke se dijele na **programirane** i **neprogramirane** odluke:

- programirane odluke - rabe se za rješavanje rutinskih problema u situacijama koje se ponavljaju. Takve odluke temelje se na ustaljenim kriterijima odlučivanja. Kod programiranih odluka koraci u odlučivanju poznati su s obzirom na ranija iskustva u donošenju istovrsnih odluka,
- neprogramirane odluke - koriste se u situacijama koje se ne ponavljaju, odnosno u izvanrednim situacijama. Neprogramirane odluke se dakle donose u situacijama u kojima se netko nađe po prvi puta. U takvim odlukama ne postoji poznati postupak i model za donošenje odluka, već je svaka takva situacija jedinstvena i zahtjeva individualni pristup rješavanju.

Hellriegel i Slocum (kao što se citira u [1]) razlikuju **rutinske, adaptivne i inovativne odluke**. Što je problem u većoj mjeri poznat, odluka će biti više rutinska i obrnuto, dok nizak stupanj poznavanja problema podrazumijeva donošenje inovativnih odluka. Detaljnije:

- rutinske odluke - rutinske ponavljajuće, svakodnevne odluke nekog radnog mjesta i izvršitelja na tom radnom mjestu. Broj rutinskih odluka najčešće je obrnuto proporcionalan razini na kojoj se pojedino radno mjesto nalazi. Radna mjesta viših razina podrazumijevaju manji broj rutinskih odluka i obrnuto,
- inovativne odluke - odluke koje se ne ponavljaju, a iziskuju kreativno mišljenje i napor pojedinaca ili skupine.

Prema Dahlu i Lindbloomu (kao što se citira u [1]) odluke možemo podijeliti s aspekta prirode problema o kojem se odlučuje:

- strukturirane odluke - donose se u okviru određene strukture kojom su i uvjetovane. Najčešće se radi o programiranim odlukama na nižim razinama menadžmenta,

- nestrukturirane odluke - odluke koje se donose u situacijama bez stalnih uloga i linija komunikacije. To su najčešće neprogramirane odluke koje se donose na razini vrhovnog menadžmenta.

Prema Chandu [5] s aspekta menadžerskih odluka, odluke je moguće klasificirati u tri glavne kategorije:

- strateške odluke - **strateške odluke** predstavljaju velike odabire akcija i utjecaja jednog dijela ili cijelog poduzeća. One direktno doprinose ostvarivanju zajedničkih ciljeva poduzeća ili organizacije. Imaju dugoročne implikacije na poslovno poduzeće. Strateške odluke su najčešće nestrukturirane te zato zahtijevaju primjenu poslovne procjene, evaluacije i intuicije u definiranju problema. Takve odluke baziraju se na parcijalnom znanju i faktorima okruženja koji su nesigurni i dinamični, a donose se na višim razinama menadžmenta,
- taktičke odluke - ove odluke povezuje se s implikacijima strateških odluka. One su usmjerene prema razvoju divizijskih planova, strukturiranju tijekova rada, uspostavljanju distribucijskih kanala te alokaciji resursa kao što su: ljudski resursi, materijali, novac. Ove odluke donose se na srednjoj razini menadžmenta,
- operativne odluke - **operativne odluke** povezuju se s svakodnevnim poslovanjem poduzeća. One imaju kratkoročni horizont te se donose ponavljajuće. Takve odluke baziraju se na činjenicama povezanim s događajima koji ne zahtijevaju visoku razinu poslovne procjene. Operativne odluke donose se na najnižim razinama menadžmenta.



Slika 1: Vrste odluka s obzirom na važnost u organizaciji (Izvor: vlastita izrada prema [6])

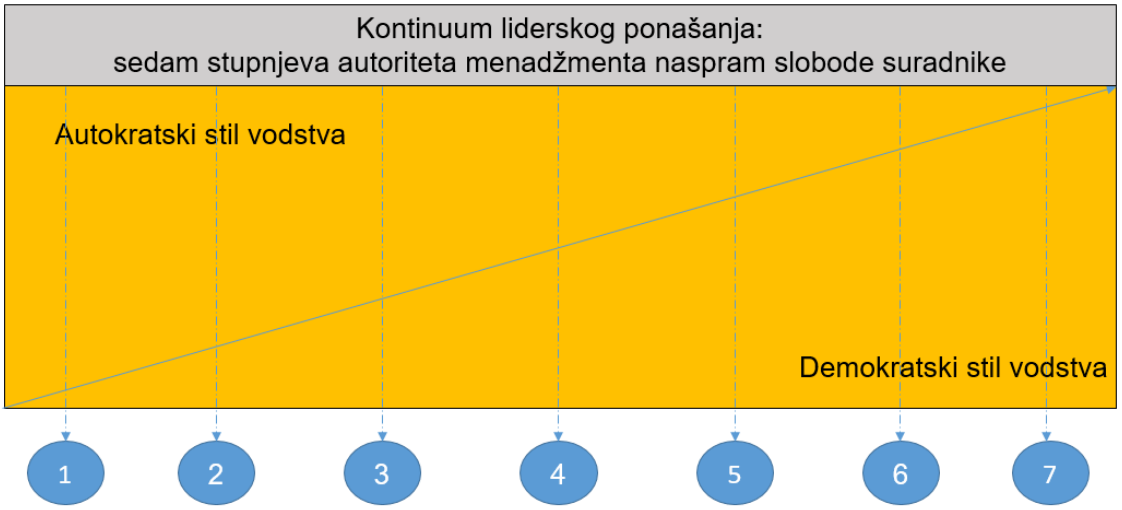
Slika 1 prikazuje prethodno spomenute odluke s aspekta menadžmenta i važnosti za organizaciju u strukturiranom obliku.

3.2. Stilovi odlučivanja

Sikavica, Hunjak, Begičević Ređep i dr. [1] navode da su **stilovi odlučivanja** u najužoj vezi sa stilovima vodstva, odnosno stilovi odlučivanja menadžmenta su najviše podudarni s stilovima vodstva jer se njihov posao, u najvećoj mjeri, svodi na donošenje odluka, odnosno odlučivanje. U teoriji i praksi vodstva stilovi vodstva se u pravilo kreću između dvije krajnosti, odnosno između autokratskog i demokratkskog stila dlučivanja.

3.2.1. Autokratski i demokratski stil odlučivanja

Clayton [7] spominje model vodstva lidera pod nazivom kontinuum liderskog ponašanja (engl. *Leadership Behaviour Continuum*) predstavljen 1959. od strane autora Tannenbaum i Schmidt prikazan na slici 2.



Slika 2: Kontinnum liderskog ponašanja (Izvor: vlastita izrada prema [7])

Prema kontinuumu liderskog ponašanja postoje sedam različitih stilova vodstva menadžera [7]:

1. menadžer donosi odluku i objavljuje ju - ovo je isključivo autoritarni stil vodstva koji ne uzima u obzir druga stajališta,
2. menadžer "prodaje svoju odluku" - menadžer zauzima poziciju donositelja odluka, ali zagovara odluku apelirajući na korist koju će grupa imati od nje. Koristi se kada je potrebna podrška grupe,
3. menadžer predstavlja odluku i poziva na pitanja - menadžer je i dalje u kontroli, ali dozvoljava grupi da istraži ideje kako bi bolje razumjeli odluku (ali ne mora uzeti u obzir njihovo mišljenje),

4. menadžer predstavlja okvirnu odluku, podložnu promjenama - u ovoj situaciji uzima se u obzir mišljenje grupe. Menadžer identificira i rješava problem, ali konzultira tim prije donošenja vlastite odluke,

5. menadžer predstavlja problem, prima prijedloge i zatim donosi odluku - u ovoj situaciji menadžer i dalje ima finalni autoritet za donošenje odluka, ali dijeli odgovornost za nalaženje riješena sa grupom, koja direktno utječe na krajnje riješenje,

6. menadžer definira ograničenja unutar kojih grupa donosi odluku - ovaj slučaj podrazumijeva da odgovornost za donošenje odluke dobiva grupa. Menadžer definira problem i postavlja ograničenja sa kojima grupa može raditi te potencijalno ograničava finalnu odluku,

7. menadžer omogućuje grupi da donosi odluku, ovisno o organizacijskim ograničenjima - grupa dobiva toliko slobode koliko sam menadžer može dozvoliti. Menadžer pomaže grupi i obvezuje se poštivati donesenu odluku.

Prema Sikavici, Hunjaku, Begičević Ređep i dr. [1] autori modela sa slike 2 Tannenbaum i Schmidt nazvali su ove stilove vodstva s lijeva na desno: **autokratski, patrijarhalni, savjetodavni, konzultantski, participativni, delegirajući i demokratski**. Kao što se u svim teorijama vodstva, stilovi vodstva kreću od autokratskog do demokratskog, tako se i svi stilovi odlučivanja kreću između ove dvije krajnosti. **Autokratski stil odlučivanja** predstavlja način odlučivanja u kojem jedna osoba ima neograničenu vlast i moć u donošenju odluka. Svi ostali zaposlenici u poduzeću isključivo su izvršitelji odluka glavnog menažmenta, odnosno direktora ili predsjednika uprave koji odlučuju na taj način. **Demokratski stil odlučivanja** ili **participativno-demokratski stil odlučivanja** podrazumijeva: postojanje više razina odlučivanja u samoj organizaciji, stvarnu moć u odlučivanju nižih razina menadžmenta, poštovanje hijerarhije hijerarhije odlučivanja, jasno razgraničavanje odluka po vrstama na niže i više razine menadžmenta. Nadalje, u demoktratskom stilu odlučivanja donositelj odluke konzultira suradnike prije nego što donese odluku [1].

3.2.2. Vroom-Yetton-Jagov model odlučivanja

Gibson, Ivancevich i Konopaske [8] predstavljaju **Vroom-Yettonov model odlučivanja**, odnosno vodstva. U razvoju modela, Vroom i Yetton iznjeli su sljedeće pretpostavke:

- model treba biti od vrijednosti za vođe ili menadžere u odabiru stila vodstva u danoj situaciji,
- niti jedan jedinstveni stil nije aplikativan u svim situacijama,
- glavni fokus mora biti sam problem koji se pokušava riješiti i situacija u kojoj problem nastaje,
- stil vodstva korišten u jednoj situaciji ne bi trebao ograničavati stilovi korištene u drugim situacijama,

- nekoliko društvenih procesa utječe na količinu sudjelovanja podređenih u rješavanju problema.

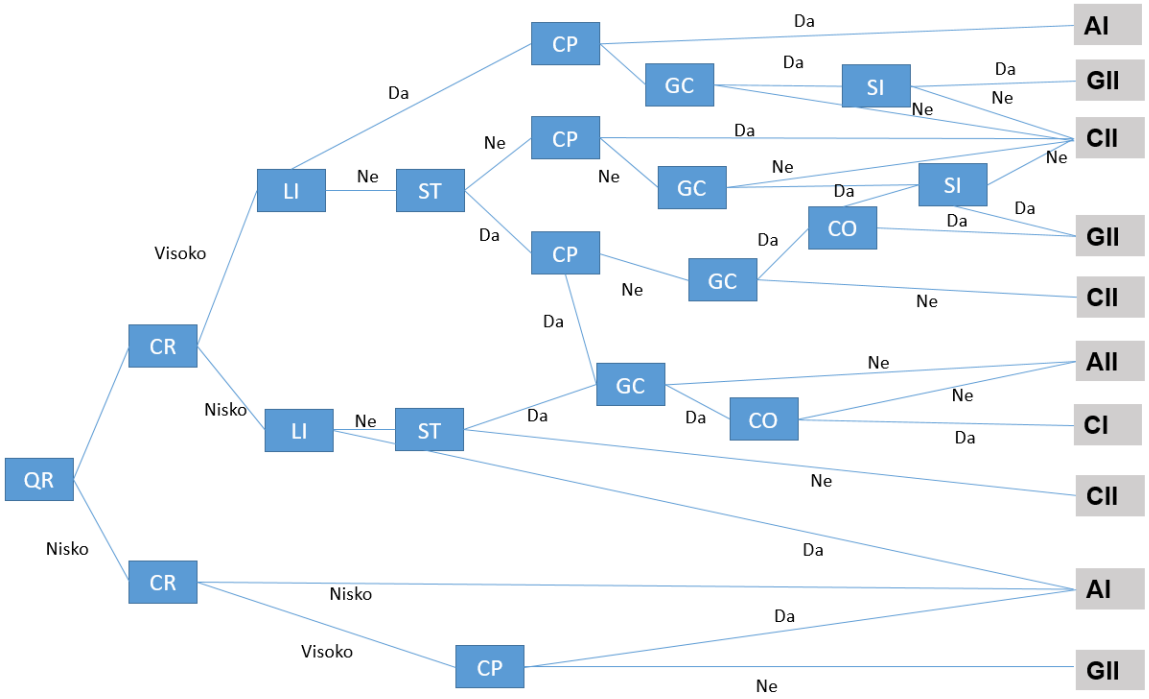
Također, stilovi vodstva/odlučivanja u Vroom - Yettonov modelu su sljedeći [8]:

- autokratski I (AI) - vođa rješava problem ili donosi odluku,
- autokratski II (AII) - vođa dobiva informaciju od suradnika te odlučuje samostalno o riješenju problema,
- konzultativni I (CI) - vođa individualno konzultira suradnike, dobiva ideje i sugestije, a naposljetku sam donosi odluku,
- konzultativni II (CII) - vođa dijeli problem s suradnicima u sklopu sastanka te dobiva njihove ideje i sugestije. Nakon toga donosi odluku koju može, a i ne mora, biti rezultat utjecaja mišljenja i stavova suradnika,
- grupni ili kolaborativni II (GII) - vođa analizira problem zajedno sa suradnicima u sklopu sastanka te zajedno generira i evaluira alternative te se postiže konsenzus. Uloga vođe u ovom stilu jest da potiče grupnu raspravu te usmjerava fokus na problem.

Kako bi poboljšali točnost i predvidljivost inicijalnog modela, Vroom i Jago razvili su modificirani model. Novi model pod nazivom **Voom-Yetton-Jagov model** koristi ključne stavke prethodnog modela. Koristi iste termine za opisivanje procesa odlučivanja - AI, AII, CI, CII i GII, sa dodatkom novih termina GI (Grupni I) i DI (Demoktratski I) za donošenje individualnih odluka. Kao i prethodni model, novi model evaluira efikasnost participacije suradnika na kvalitetu odluke, prihvaćanje odluke, razvoj suradnika i vrijeme.

Slika 3 prikazuje Vroom-Yeton-Jagov model vodstva/odlučivanja u obliku stabla odlučivanja. Slijede objašnjenja pojmova koji se nalaze u čvorovima odluka [8]:

- QR - zahtjev za kvalitetom (engl. *Quality Requirement*) - koliko je bitna tehnička kvaliteta odluke,
- CR - zahtjev za predanošću (engl. *Commitment Requirement*) - koliko je bitna predanost suradnika u donošenju odluke,
- LI - informacije Vođe (engl. *Leader's Information*) - ima li vođa dovoljno informacija da donese visoko-kvalitetnu odluku,
- ST - struktura problema (engl. *Problem Structure*) - da li je problem dobro strukturiran,
- CP - vjerojatnost predanosti (engl. *Commitment Probability*) - kolika je vjerojatnost da suradnici budu predani odluci ako ju vođa donosi samostalno,
- GC - podudarnost cilja (engl. *Goal Congruence*) - dijele li suradnici dijele organizacijske ciljeve koji se postižu rješavanjem problema,
- CO - konflikt suradnika (engl. *Subordinate Conflict*) - da li postoji vjerojatnost izbivanja konflikta među suradnicima u odabiru odluke,



Slika 3: Vroom-Yeton-Jagov model vodstva/odlučivanja (Izvor: vlastita izrada prema [8], str. 348)

- SI - informacije suradnika (engl. *Subordinate Information*) - imaju li suradnici dovoljno informacija da donesu visoko-kvalitetnu odluku.

Odabir stila vodstva/odlučivanja temeljem Vroom-Yetton-Jagovog modela prikazan je stablom odlučivanja na slici 3 te ovisi o odgovorima na prethodno iznesena pitanja. Iako se ovaj pristup ističe svojom nenadmašenosti u smislu znanstvene validnosti i praktične koristi, ima nekoliko ograničenja i mana. Model prisiljava osobu da donosi isključivo određene odgovore (*da* ili *ne*). Nadalje, poslovne situacije nisu toliko jednostavne za kategorizaciju; u mnogim situacijama niti *da* niti *ne* nisu točni. Model je također kritiziran zbog njegove kompleksnosti [8].

3.2.3. Ostali stilovi odlučivanja

Sikavica, Hunjak, Begičević Ređep i dr. [1] navode kako različiti autori predstavljaju različite klasifikacije stilova odlučivanja, no svima im je zajedničko, bez obzira na nazive stilova, da se kreću u već poznatom rasponu menadžerskog ponašanja. Tako npr. Driver, Brousseau i Hunsaker (kao što se citira u [1]) navode sljedeće stilove odlučivanja:

- **odlučan stil** - obilježava ga usmjerenost na djelovanje te upotreba minimuma informacija za rješavanje problema. U ovom stilu donositelj odluke traži minimum informacija kako bi donio zadovoljavajuću odluku, što znači da donositelj odluke ne traga za najboljim rješenjem, već za zadovoljavajućim,
- **fleksibilan stil** - stil odlučivanja u kojem donositelj odluka barata s većom količinom informacija, što mu omogućava da razvije veći broj alternativa rješenja problema kako

bi lakše mogao izabrati najbolju alternativu, odnosno donijeti najbolju odluku,

- **hijerarhijski stil** - stil odlučivanja koji obilježava upotreba velike količine informacija i analiza, na temelju kojih donositelj odluke iz malog broja inačica, odnosno alternativa kreira jednu određenu mogućnost rješavanja problema,
- **integrativni stil** - obilježava ga upotreba velike količine informacija, što omogućuje generiranje velikog broja inačica rješavanja problema prije nego se donositelj odluke odluči za pojedino rješenje,
- **sustavni stil** - najsloženiji od prethodno navedenih stilova. Karakterizira ga upotreba velike količine informacija kako bi se lakše razumjela različita gledišta odlučivanja i inačice rješavanja problema.

Razlike prethodno navedenih stilova odlučivanja očituju se u količini informacija koju koriste pri generiranju inačica rješavanja problema te broju inačica koje se generiraju prije donošenja finalne odluke. Odabir stila uvelike ovisi o situaciji o kojoj se donositelj odluke nalazi, odnosno da li potrebno donijeti brzu ili kvalitetnu odluku [1].

Prema Robinsu i Judgeu [9] postoje tri različita stila odlučivanja: racionalni model odlučivanja, model odlučivanja ograničene racionalnosti (engl. *Bounded rationality*) i intuitivni model odlučivanja [9]:

- **racionalni model odlučivanja** - koraci racionalnoga modela odlučivanja su:
  1. definiranje problema,
  2. identificiranje kriterija odlučivanja,
  3. alociranje težina kriterijima,
  4. razvoj alternativa,
  5. evaluacija alternativa,
  6. selekcija najbolje alternative.

  Česta pomisao je da je najbolji donositelj odluke racionalan, konzistentan i da maksimizira vrijednost u specificiranim okvirima. Racionalni model donošenja odlučivanja oslanja se na brojne pretpostavke, uključujući da donositelj odluke ima cjelovite informacije i može identificirati sve relevantne alternative na nepristran način te odabrati alternativu s najvećom korisnošću,
- **model ograničene racionalnosti** - ograničena sposobnost obrade informacija kod ljudi onemogućava asimilaciju i razumijevanje svih informacija potrebnih za optimizaciju odlučivanja. Ljudski um ne može u potpunosti formulirati i riješiti složene probleme na racionalan način te iz tog razloga djeluje u granicama ograničene racionalnosti. Ljudi konstruiraju pojednostavljene modele koji izvlače bitna svojstva problema bez hvatanja cjelokupne kompleksnosti. Model ograničene racionalnosti omogućava generiranje dovoljno dobrih alternativa te donošenje odluke na brz i efikasan način,

- **intuitivni model odlučivanja** - intuitivni model predstavlja najmanje racionalan način donošenja odluka temeljenih na nesvjesnom procesu kreiranom iz dosadašnjih iskustva. Javlja se van svjesne misli; oslanja se na holističke asocijacije ili veze između dijelova informacija; brz je i djeluje afektivno, što znači da obično uključuje emocije. Iako intuicija nije racionalna, ona nije nužno pogrešna. Intuicija se ne suprotstavlja racionalnoj analizi, već djeluje kao njezina logička dopuna.

Zaključno, bilo bi pogrešno tvrditi da je neki od prethodno navedenih stilova odlučivanja bolji od drugoga, već je na samom donositelju odluka da procjeni koji stil odlučivanja odgovara pojedinoj situaciji i problemu koji se rješava.

3.3. Načini odlučivanja

Prema Sikvaici, Hunjaku, Begičević Ređep i dr. [1] izbor načina odlučivanja ponajviše ovisi o prirodi problema o kojemu se odlučuje. Različiti načini odlučivanja međusobno se ne isključuju, već se mogu primijeniti istovremeno. Kada se govori o načinima odlučivanja ponajprije se misli na različite aspekte odlučivanja, odnosno različiti načini odlučivanja daju različite odgovore na pitanja: o kojim problemima se odlučuje, tko odlučuje, kako se odlučuje, pod kakvim se okolnostima odlučuje.

3.3.1. Programirano i neprogramirano odlučivanje

Programirano i neprogramirano odlučivanje u najvećoj je vezi s programiranim i neprogramiranim odlukama. "**Programirano odlučivanje** jedan je od načina odlučivanja koji se primjenjuje za rješavanje rutinskih problema koji se ponavljaju, tj. koji imaju repetitivno obilježje. Dakle, u svim onim situacijama odlučivanja kad se donositelj odluke suočava s poznatim, svakodnevnim, rutinskim problemima, koristi će se programiranim odlučivanjem." [4, str. 214]. Programirano odlučivanje karakteriziraju poznati koraci u odlučivanju te se takav način odlučivanja koristi za rješavanje problema koji se ponavljaju. U programiranom odlučivanju donositelju odluke poznate su sve potrebne informacije, problem je dobro definiran te su poznate alternative rješavanja problema [1]. Prema Daftu (Kao što se citira u [1]) programirano odlučivanje karakteriziraju:

- dobro strukturirani problemi,
- jasni kriteriji procedure odlučivanja,
- primjerena dostupnost informacija potrebnih za odlučivanje,
- lako određivanje alternativa za rješavanje problema,
- relativna sigurnost u uspješnost odabrane inačice.

Neprogramirano odlučivanje se pak primjenjuje u situacijama koje nisu redovne i koje se ne ponavljaju. Dakle, neprogramirano odlučivanje primjenjuje se u novim, neizvjesnim

i neistraženim situacijama. Nadalje, neprogramirano odlučivanje koristi se za rješavanje jedinstvenih, složenih i nestrukturiranih problema te iz tog razloga zahtijeva određenu razinu kreativnosti i intuicije u donošenju odluka [1]. Prema Daftu (Kao što se citira u [1]) neprogramirano odlučivanje karakterizira:

- slabo definirani problem o kojem se odlučuje,
- primjena u izvanrednim situacijama,
- primjena u novim situacijama koje se ne ponavljaju,
- razvoj malog broja alternativa za rješavanje problema, a najčešće samo jedne.

Programirano i neprogramirano odlučivanje pojavljuju se na suprotnim stranama kontinuuma odlučivanja, odnosno kao dvije krajnosti u procesu odlučivanja [1].

3.3.2. Individualno i skupno odlučivanje

Kada se o odlučivanju govori s aspekta subjekta odlučivanja, odnosno onoga tko odlučuje, najčešće se govori o **pojedinačnom** i **skupnom** odlučivanju. Osim toga, način odlučivanja ovisi i o vrsti odluke, pa će se razlikovati odlučivanje o operativnim, taktičkim i strateškim odlukama. Operativne ili rutinske odluke najčešće su pojedinačnog karaktera, dok se o taktičkim i strateškim pitanjima donose skupne odluke, bez obzira tko ih u organizaciji donosi [1].

Prema Sikavici, Hunjaku, Begičević Ređep i dr. [1] **pojedinačno odlučivanje** situacija je u kojoj odluku donosi jedna osoba ili pojedinac. Glavna razlika u odnosnu na skupno odlučivanje očituje se u broju sudionika odlučivanja, što povlači i razlike u načinu i trajanju procesa odlučivanja. Pojedinačno odlučivanje najčešće je brže i jednostavnije od skupnog, što ne mora nužno značiti da je lakše, jer odgovornost za pogreške snosi samo jedna osoba. Istraživanja na temu individualnoga odlučivanja identificirala su četiri različita pristupa individualnom odlučivanju [10]:

- analitički pristup - analitičke donositelje odluka najbolje karakterizira pažljivost u donošenju odluka s sposobnošću nošenja s novim i neočekivanim situacijama,
- konceptualni pristup - donositelji odluka koji se koriste ovim stilom najčešće koriste informacije iz više izvora i razmatraju veći broj alternativa. Njihov fokus su dugoročne odluke i jako su dobri u pronalaženju kreativnih rješenja,
- direktivni pristup - korisnici direktivnog stila odlučivanja imaju malu toleranciju na dvosmislenost i fokus im je racionalnost u odlučivanju. Oni su efektivni i logički, ali njihov fokus na efikasnost često rezultira u odlukama donesenim na temelju minimalne količine informacija i malog broja razmotrenih alternativa,
- bihevioristički pristup - menadžeri koje karakterizira korištenje biheviorističkog stila imaju veliku brigu za ljude u organizaciji i njihov razvoj. Brinu se za dobrobit svojih suradnika i obično razmatraju prijedloge od drugih.

Zbog toga, glavna prednost pojedinačnog odlučivanja je brzina procesa odlučivanja s obzirom da je pojedinac brži od skupine. Osim toga, kod donošenja važnih odluka preferira se mišljenje informiranog stručnjaka te se izbjegavaju problemi skupnog odlučivanja, kao što je skupno mišljenje. Kao najveća slabost individualnog odlučivanja ističe se činjenica da pojedinac najčešće može razviti manji broj alternativa za rješavanje problema od skupine [1].

S druge strane, **skupno odlučivanje** je prema Sikavici, Hunjaku, Begičević Ređep i dr. [1] način poslovnog odlučivanja u kojem odluke donose skupine ljudi strukturirane prema različitim pravilima. Neke od osnova za formiranje skupine su: vlasništvo, menadžerske funkcije, zajednički rad i slično. Prema Robbinsu i Coulteru [11] rijetke su organizacije koje ne koriste odbore, radne skupine, komisije, timove i slične grupe kako bi donosile grupne odluke. Prema Moorheadu i Griffinu (kao što se citira u [1]) glavni čimbenici koji utječu na uspješnost skupnog odlučivanja su sastav skupine, veličina skupine, kohezivnost (unutrašnja povezanost) i norme ponašanja grupe te svaki od spomenutih čimbenika više ili manje utječu na uspješnost i kvalitetu skupnog odlučivanja.

Osim toga, prema Robbinsu i Coulteru [11] skupine ljudi koje sudjeluju u skupnom odlučivanju mogu se razvrstati prema sastavu sudionika:

- timovi za rješavanje problema - timovi ljudi najčešće iz istog odjela ili funkcijskog područja uključeni u nastojanja poboljšanja poslovnih aktivnosti ili rješavanja specifičnih problema,
- samoupravni radni timovi - formalna grupa suradnika koji funkcioniraju bez menadžera i odgovorni su za kompletan segment poslovanja. Takav oblik skupine zadužen je za obavljanje posla i samoupravljanje, što najčešće znači planiranje i zakazivanje vlastitog posla, dodjela zadataka pojedincima te donošenje operativnih odluka i rješavanje problema,
- Komplementarni timovi - skupina suradnika sastavljena od pojedinaca iz različitih funkcijskih područja u poduzeću,
- virtualni timovi - virtualni tim je pojam koji se koristi za skupinu ljudi koji koriste tehnologiju kako bi prevazišli fizičku dislociranost sudionika, u korist postizanju zajedničkog cilja. Takvi timovi su najčešće više orijentirani zadatku zbog činjenice da se članovi najčešće nikad nisu susreli uživo.

Također, skupine je moguće podijeliti prema tome jesu li privremene i trajne te se temeljem toga mogu formirati različiti tipovi timova: prirodni radni timovi, mali timovi za projekte, timovi za posebne svrhe i međufunkcijski timovi. Osim po trajanju, skupine se može još i razvrstati prema heterogenosti ili homogenosti skupine te prema veličini skupine, što na kraju uvelike utječe na sam proces donošenja grupnih odluka [1].

Sikavica, Hunjak, Begičević Ređep i dr. [1] navode kako su glavni **problemi skupnog odlučivanja**:

1. **polarizacija mišljenja članova skupine** - u procesu grupnog odlučivanja pojedini članovi skupine mogu zastupati različita, pa čak i ekstremna stajališta. Do polarizacije između

članova skupine može doći iz različitih interesa ili mišljenja članova, različitog položaja u organizaciji te zbog različitih znanja ili sposobnosti članova,

2. **skupno mišljenje** - do problema skupnog mišljenja dolazi kada se javi jedinstveno mišljenje svih članova skupine, koje se najčešće javlja kao želja svih članova da se postigne sporazum pod svaku cijenu. Do skupnog mišljenja može doći iz dva razloga: način mišljenja u skupini i međusobna povezanost članova skupine. Kako bi se skupine oslobodila skupnog mišljenja, kao potencijalnoga problema potrebno je poticati pojedinačno mišljenje i izražavanje te oformiti više paralelnih skupina koje odlučuju o istom problemu,
3. **participacija zaposlenika u procesu odlučivanja** - participacija zaposlenika u odlučivanju najčešće produljuje sam proces, a u situacijama kada je potrebno donijeti odluku u kratkom roku, to može biti vrlo loše. Stupanj participacije zaposlenika u procesu odlučivanja najviše ovisi o stilu odlučivanja menadžmenta organizacije i o strukturi same organizacije. Ukoliko menadžer prakticira demokratski stil te ako je organizacijska struktura decentralizirana, povećava se vjerojatnost da će zaposlenici participirati u procesu odlučivanja.

Osim toga, zbog većeg broja zaposlenika koji sudjeluju u procesu skupnog odlučivanja te potrebe za organizacijom i vođenjem samog procesa koriste se neke od tehnika skupnog odlučivanja [1]:

- oluja mozgova (engl. *brainstorming*) - tehnika odlučivanja koja se temelji na intenzivnoj diskusiji između članova skupina. Temelj ove tehnike je razvijanje ideja, odnosno alternativa rješavanja problema od strane svakog sudionika te generiranje velikog broja inačica i naposljetku odabir najbolje inačice,
- tehnika nominalne skupine - u ovoj tehnici ideje razvijaju članovi skupine, a potiče se kreativnost svakog člana. Skupina pomaže u objašnjavanju prezentiranih ideja te ih zatim ocjenjuje i na kraju bira najbolju ideju kao riješenje problema o kojem se odlučuje,
- *Delphi* metoda - tehnika skupnog odlučivanja koja se temelji na postizanju konsenzusa između stručnjaka koji donose odluke na način da se upotrebljava niz upitnika. Postupak odlučivanja sličan je tehnici nominalne skupine, a glavna razlika je da se članovi skupine ne susreću licem u lice te je idealno kada ne znaju tko je sve uključen u proces odlučivanja. Ova metoda je prvi puta primijenjena u organizacija *Rand Corporation* u SAD-u za potrebe predviđanja utjecaja nuklearnog napada na SAD.

Kao i individualno odlučivanje, skupno odlučivanje ima svoje prednosti i nedostatke. Prema Robbinsu i Coulteru [11] glavne prednosti skupnog odlučivanja su:

- grupe generiraju potpune informacije i znanje,
- grupe donose raznolikost iskustva i perspektiva procesu donošenja odluka,
- grupe generiraju različitije alternative zbog veće količine i raznolikosti informacija,

- grupe povećavaju prihvatljivost i legitimnost riješena,
- pojedinci ne potkopavaju odluke kojima su sami pomogli u razvoju.

S druge strane nedostatci skupnog odlučivanja su [11]:

- grupama često treba više vremena da bi se došlo do rješenja nego pojedincu,
- dominantna i glasna manjina često može utjecati na odluke grupe,
- skupno mišljenje može potkopati kritičko mišljenje u grupi te na taj način utjecati na kvalitetu odluke,
- pojedinci dijele odgovornost, ali je odgovornost svakog pojedinca unutar grupe dvosmislena .

Odabir načina odlučivanja u smislu subjekta odlučivanja uvelike ovisi o prirodi problema o kojem se odlučuje i strukturi organizacije te ne postoji zlatno pravila kada koristiti neki od prethodno spomenutih pristupa.

3.3.3. Intuitivno i racionalno odlučivanje

Prema Mesconu, Albertu i Khedouri (kao što se citira u [1]) svaka odluka je spoj triju čimbenika: intuicije, prosudbe i racionalnosti, pa se temeljem toga može govoriti o:

- intuitivnom odlučivanju,
- odlučivanju na temelju prosuđivanja,
- racionalnom odlučivanju.

Prema Sikavici i Bahtijarević-Šiberu [12] **intuitivno odlučivanje** je odlučivanje na temelju intuicije, odnosno osjećaja donositelja odluke. U slučaju intuitivnog odlučivanja donositelj odluke nije u mogućnosti objasniti razloge zašto je u odlučivanju postupio kako je postupio. Racionalno je taj proces nemoguće objasniti jer kod intuitivnog odlučivanja donositelj odluke bira između mogućih alternativa. Ukoliko je broj alternativa manji, postoji veća vjerojatnost da će se intuitivnim odlučivanjem donijeti prava odluka, odnosno izabrati prava alternativa. Intuitivno odlučivanje u privatnom i poslovnom životu daje male šanse izbora najpovoljnije odluke te ga je iz tog razloga potrebno svesti na najmanju moguću mjeru. Intuitivno odlučivanje moguće je primijeniti u svim fazama procesa odlučivanja, a posebno u fazi identifikacije problema i fazi odlučivanja o načinu rješavanja problema [1].

Odlučivanje na temelju prosuđivanja ima veću vrijednost od intuitivnog odlučivanja, a manju od racionalnoga. Općenito, ovaj način odlučivanja bliži je intuitivnom nego racionalnom odlučivanju. Ovaj način odlučivanja koristi se u situacijama koje se ponavljaju, odnosno kod donošenja programiranih odluka. Jednostavnije, radi se o odlučivanju na temelju prijašnjih iskustava i znanja za iste ili slične situacije. Primjena ovog načina odlučivanja uvelike ovisi o

tome da li se ponovljena odluka donosi u istim ili sličnim uvjetima kao neka prijašnja. Radi se o relativno jeftinom i brzom načinu odlučivanja, ali ograničenog dometa jer se odnosi uglavnom na ponavljajuće situacije [12].

Racionalno odlučivanje je najbolji način odlučivanja te se radi o načinu odlučivanja temeljenom na analitičkom pristupu koji se sastoji od određenih faza te se koristi u situacijama koje se ne ponavljaju. Jedan od najutjecajnijih zastupnika ideje racionalnosti u odlučivanju bio je M.Weber te je rezultat njegove teorije idealne birokracije oblik racionalnosti koji je postao standard u većini teorija odlučivanja. Racionalno odlučivanje je najpogodniji način odlučivanja u uvjetima u kojima se tokom vremena ne mijenjaju parametri, tj. čimbenici utjecaja na odlučivanje. Iako je ovaj način odlučivanja najpouzdaniji, ima i određena ograničenja u smislu dometa, što nije pogodno za nestabilnu i turbulentnu okolinu koja se brzo mijenja, u kakvoj se danas donose najvažnije poslovne odluke [1].

3.3.4. Odlučivanje s obzirom na okolnosti odlučivanje

S obzirom na okolnosti u kojime se odlučuje, prema Sikavici, Hunjaku, Begičević Ređep i dr. [1], može se govoriti o odlučivanju u uvjetima: sigurnosti, rizika i nesigurnosti. Prema Dawsonu [13] količina informacija i znanja koju posjeduju donositelji odluka u procesu odlučivanja razmotriti s tri stajališta:

1. poznavanje i definicija problema,
2. poznavanje mogućih alternativa akcija,
3. poznavanje ishoda, odnosno posljedica svake od alternativa.

Uzimajući u obzir varijacije ova tri faktora moguće je govoriti o odlučivanju u uvjetima sigurnosti, rizika i nesigurnosti. Prva okolnost, prema Dawson [13] odnosno **sigurnost** karakterizirana je dobro definiranim i razumljivom problemu, kao i jasnom identifikacijom mogućih alternativa rješavanja problema kojima su poznate specifične posljedice.

Druga okolnost, odnosno okolnost **rizika** karakterizirana je približnom definicijom problema i približnim razumijevanjem potencijalnih alternativa rješavanja problema čijim posljedicama mogu biti dodijeljene vjerojatnosti [13]. Prema Sikavici, Hunjaku, Begičević Ređep i dr. [1] rizik je jedna od okolnosti odlučivanja, smještena između sigurnosti i nesigurnosti. "Odlučivanje u uvjetima rizika jest odlučivanja u okolnostima u kojima rezultati nisu sigurni, ali su poznate vjerojatnosti za postizanje različitih rezultata." [4, str. 270]. Stupanj vjerojatnosti pojave određenog događaja rangira se od 0 do 1, tako da je zbroj vjerojatnosti za sve mogućnosti jednak 1. Dakle kada govorimo o odlučivanju u uvjetima rizika, poželjno je da donositelj odluke svim alternativama dodijeli vjerojatnost. Nadalje, za utvrđivanje vjerojatnosti svake pojedine inačice donositelj odluke može se koristiti matematičkim modelima ili procjenom temeljem iskustva. Kada se vjerojatnosti provedbe određene inačice utvrđuju matematičkim putem, npr. na temelju povijesnih podataka govorimo o **objektivnoj vjerojatnosti**. Nasuprot tome, kada se do vjerojatnosti dolazi procjenom temeljem prijašnjih iskustava govori se o

subjektivnoj vjerojatnosti. U većini situacija nemoguće je eliminirati rizik, ali se postavlja pitanje da li ga je moguće smanjiti [1].

Najmanje poželjna, ali vrlo česta situacija je odlučivanje u uvjetima nesigurnosti. Nesigurnost je okolnost s kojom se menadžeri u suvremenim organizacijama susreću na dnevnoj bazi, a uzrokovana je dinamikom, nestabilnošću i rapidnim promjenama koje se dešavaju u modernim organizacijama [1]. Okolnost **nesigurnosti** karakterizirana je lošom definicijom problema te slabim razumijevanjem mogućih alternativa s manjkom znanja o posljedicama nekih ili svih alternativa [13]. Nadalje, prema Sikavici, Hunjaku, Begičević Ređep i dr. [1] kada govorimo o odlučivanju u uvjetima nesigurnosti, najčešće govorimo o dva slučaja:

- donositelj odluke ne zna sve inačice riješena problema,
- donositelj odluke zna moguće inačice riješena problema, ali ne zna vjerojatnost uspješnosti svake mogućnosti.

Odluke se donose u okolnostima nesigurnosti kada nije moguće odrediti vjerojatnost ostvarenja svake mogućnosti, odnosno kada je nemoguće odrediti vjerojatnost postizanja mogućih rezultata [1].

Osim toga, prema Brunssonu (kao što se citira u [1]) nesigurnost se može podijeliti na:

- kognitivna ili spoznajna nesigurnost - nesigurnost pri kojoj pojedinac nije siguran odgovara li njegova spoznaja cjeline stvarnoj situaciji,
- nesigurnost u prosuđivanju - pojedinac može predvidjeti rezultate svojih akcija, ali nije potpuno siguran u rezultate svog odabira,
- nesigurnost u procjenjivanju - nesigurnost pojedinca u odlučivanju pri donošenju procjene na temelju postojećih činjenica.

4. Odlučivanje u uvjetima nesigurnosti i rizika

U današnje vrijeme ljudi svakodnevno donesu mnogo odluka, poput kupnje određenog predmeta ili ulaganja u isti. Neke od tih odluka su jednostavne, ali kada u isto vrijeme postoji mnogo mogućnosti za donošenje odluka te se pojavljuju uvjeti (okolnosti) **rizika** i **nesigurnosti** koji uvelike utječu na donošenje konačne odluke od strane odlučitelja [14]. Nadalje, prema iskustvu je poznato da su ljudi koji donose odluke nakon dobro promišljenih izračuna, bez obzira da li se radi o poslovnoj ili privatnoj odluci u manjini. Isto tako je poznato da kada se odluke donose u uvjetima rizika, ljudi najčešće zanemaruju normativna pravila odlučivanja te se najčešće oslanjaju na intuiciju i sljutne koje se u tom trenutku čine ispravnima. Prema Kurhadeu i Wankhadeu [14] deskriptivna teorija daje objašnjenja zašto ljudi donose odluke na način da zanemaruju sugestije normativnih pravila donošenja odluka u uvjetima nesigurnosti i rizika. Također, u svakoj organizaciji, pri donošenju odluka potrebno je intenzivno analizirati njezinu strukturu i organizacijsku kulturu pošto oboje uvelike utječu na proces donošenja odluka. Rizici i nesigurnosti projektnog razvoja proizlaze iz različitih izvora grešaka poput: grešaka u podacima, grešaka u modelima i grešaka u predviđanju. Istraživanja pokazuju kako je najbitniji faktor koji utječe na rizik i nesigurnost rezultat grešaka u predviđanju. Općenita pretpostavka je da se odluke ekonomskog tipa donose kada postoji potpuna informacija, odnosno kada i kupci i proizvođači ovise o točnom znanju o poduzeću, kupcu i uvjetima tržišta. Kada ne postoji potpuna informacija o npr. lansiranju novog proizvoda (makroekonomski uvjeti, ponašanja potrošača), nesigurnost koja se povezuje s donošenjem i ishodom takvih odluka uvelike se povećava ovisno duljini horizonta odlučivanja [14].

4.1. Nesigurnost i rizik općenito

Okolnost **rizika** karakterizirana je približnom definicijom problema i približnim razumijevanjem potencijalnih alternativa rješavanja problema čijim posljedicama mogu biti dodijeljene vjerojatnosti [13]. Prema Sikavici, Hunjaku, Begičević Ređep i dr. [1] rizik je jedna od okolnosti odlučivanja, smještena između sigurnosti i nesigurnosti. Odlučivanje u uvjetima rizika podrazumijeva odlučivanje u okolnostima u kojima rezultati nisu sigurni, ali su poznate vjerojatnosti postizanja različitih rezultata [1]. Kada govorimo o odlučivanju u uvjetima rizika, poželjno je da donositelj odluke svim alternativama dodijeli vjerojatnost pojave (dok zbroj svih vjerojatnosti mora biti jednak 1). Za utvrđivanje vjerojatnosti svake pojedine inačice donositelj odluke može koristiti matematičkim model ili procjenu temeljem vlastitog iskustva. Kada se vjerojatnosti provedbe određene inačice utvrđuju matematičkim putem, npr. na temelju povijesnih podataka govorimo o **objektivnoj vjerojatnosti**. Nasuprot tome, kada se do vjerojatnosti dolazi procjenom temeljem prijašnjih iskustava govori se o **subjektivnoj vjerojatnosti**. U većini situacija nemoguće je eliminirati rizik te se postavlja pitanja da li ga je i u kojoj mjeri ga je moguće smanjiti [1]. S druge strane, vrlo česta situacija koja se događa prilikom odlučivanja je odlučivanje u uvjetima nesigurnosti. **Nesigurnost** je okolnost s kojom se menadžeri u suvremenim organizacijama susreću na dnevnoj bazi, a uzrokovana je dinamikom, nestabilnošću i rapidnim promjenama koje se dešavaju u modernim

organizacijama [1]. Okolnost **nesigurnosti** karakterizirana je lošom definicijom problema te slabim razumijevanjem mogućih alternativa s manjkom znanja o posljedicama nekih ili svih alternativa [13].

4.1.1. Rizik

Jednostavna definicija rizika prema Kurhadeu i Wankhadeu [14] je vjerojatnost donošenja odluke o metodi ili prilici za postizanje boljeg ishoda. Cilj procjene rizika je smanjiti potencijalne negativne ishode neke specifične odluke. U velikom broju izvora literature "rizik" se definira kao koncept koji se sastoji od dva glavna elementa:

1. vjerojatnost pojave negativnog događaja tijekom životnog vijeka rada objekta,
2. rezultirajuća posljedica prilikom pojave negativnog događaja.

Prvi pojam uključuje **procjenu rizika**, a drugi pojam se odnosi na **menadžment rizika**. Procjena rizika je znanstveni zadatak, dok je menadžment rizika područje istraživanja regulatronih mjera baziranih na procjeni rizika uzimajući u obzir: legalnu, političku, socijalnu, ekonomsku, okolišnu i inženjersku prirodu problema koji proizlazi iz rizika [14]. Općenito, situacija u kojoj je ishod nepoznat donositelju odluke, odnosno u kojoj donositelj odluke nije siguran u ishod koji će se dogoditi naziva se riskantna situacija.

4.1.2. Nesigurnost

Prema Kurhadeu i Wankhadeu [14] **nesigurnost** je stanje manjka znanja. Razna područja istraživanja čije je zajedničko područje istraživanja nesigurnost, nemaju sporazum o definiciji terminologiji ili klasifikaciji nesigurnosti. Također se može reći da ne postoji dobar način za opisivanje događaja koji se pojavljuju u uvjetima nesigurnosti. Sama riječ **nesigurnost** naglašava da odluka donesena u procesu donošenja odluke mora biti donesena na bazi nepotpunog znanja o događajima koji još fizički ne postoje. Glavni izvori iz kojih proizlazi nesigurnost su [14]:

- greške u podacima (engl. *Data errors*) - nesigurnosti u vezi događaja iz prošlosti,
- greške u predviđanjima (engl. *Forecasting errors*) - nesigurnosti u vezi budućih događaja,
- greške u modelima (engl. *Model errors*) - preostale pogreške, razlike između promatranog i onoga što je prikazano modelom.

Osim toga, prema Sikavici, Hunjaku, Begičević Redep i dr. [1], kada govorimo o odlučivanju u uvjetima nesigurnosti, najčešće govorimo o dva slučaja:

- donositelj odluke ne zna sve inačice riješena problema,
- donositelj odluke zna moguće inačice riješena problema, ali ne zna vjerojatnost uspješnosti svake mogućnosti.

Odluke se donose u okolnostima nesigurnosti kada nije moguće odrediti vjerojatnost ostvarenja svake mogućnosti, odnosno kada je nemoguće odrediti vjerojatnost postizanja mogućih rezultata [1]. Osim toga, prema Brunssonu (kao što se citira u [1]) nesigurnost se može podijeliti na:

- kognitivna ili spoznajna nesigurnost - nesigurnost pri kojoj pojedinac nije siguran odgovara li njegova spoznaja cjeline stvarnoj situaciji,
- nesigurnost u prosuđivanju - pojedinac može predvidjeti rezultate svojih akcija, ali nije potpuno siguran u rezultate svog odabira,
- nesigurnost u procjenjivanju - nesigurnost pojedinca u odlučivanju pri donošenju procjene na temelju postojećih činjenica.

4.1.3. Procjena rizika i nesigurnosti

Istraživanja su utvrdila da mnoge europske zemlje koriste scenarije za istraživanje rizičnog učinka i neizvjesnosti projekata ulaganja koja uključuju istraživanje scenarija, osjetljivosti i procjena na temelju vjerojatnosti. Slijedi objašnjenje navedenih tehnika [14]:

1. **analiza scenarija** - analiza scenarija povezana je s istraživanjem različitih alternativnih opcija koje dobro funkcioniraju u okolnostima minimalnog rizika. Trenutno, analiza scenarija je jednostavan alat korišten za procjenu rizika i nesigurnosti u budućnosti. S obzirom da je budućnost nesigurna i uključuje rizik, jedan način procjene rizika u takvim situacijama je konstrukcija potencijalnih scenarija te pronalaženje najbolje opcije koje dobro funkcioniraju u okolnostima minimalnog rizika. Analiza scenarija započinje s definiranjem različitih scenarija, njihovih kriterija, utjecaja i rizika. Utjecaji i rizici mogu uključivati kreiranje tablice rezultata za svaki scenarij. Proces donošenja odluke može biti napravljen na temelju scenarija koji posjeduju najveću korist, odnosno najbolji odnos cijene i ishoda, minimalnog rizika i utjecaja. Analiza scenarija ne predviđa što će se dogoditi u budućnosti te ne daje vjerojatnosti događaja, već daju naznaku kakvi su mogući rezultati s obzirom na dane alternative,
2. **analiza osjetljivosti** - analize osjetljivosti nisu ništa drugo nego identifikatori glavnih izvora nesigurnosti. Provode se s ciljem identifikacije varijabli koje uvelike doprinose razini nesigurnosti predviđanja. Ulazne varijable koje imaju visoku osjetljivost u odnosu na buduća predviđanja se mjere na početku analize te se prilikom predviđanja u obzir uzimaju faktori koji su najosjetljiviji u donošenju odluka. Glavni cilj analize osjetljivosti je identifikacija glavnih efekata i interakcije efekata ulaznih varijabli. Nesigurnosti se najčešće pojavljuju kao rezultati grešaka u podacima i grešaka predviđanja,
3. **procjena na temelju vjerojatnosti** - procjena na temelju vjerojatnosti nije ništa drugo nego identifikacija različitih nesigurnosti te je isključivo statistička metoda. Prvo se uspostavljaju matematičke funkcije povezanih faktora koji utječu na proces donošenja odluke te se nakon toga kvantificiraju i modeliraju nesigurnosti ulaznih varijabli tih funkcija

uz pomoć vjerojatnosti distribucije i njezinih statističkih parametara, poput: srednje vrijednosti i koeficijenta varijacije. Rezultat analize je vjerojatnosna distribucija izlaznih parametara. Alternative koje su riskantnije od ostalih rezultiraju većim gubitkom s većim vjerojatnostima. Također, važno je napomenuti da linije rizika same po sebi nisu dovoljne kako bi donositelj odluke donio ispravnu odluku, već su mu potrebni i moguće koristi svake od alternativa, kao i njihove vjerojatnosti. Osim toga, bitan je i stav prema riziku donositelja odluke.

4.1.4. Donošenje odluka u jednoj fazi

Prema Kurhadeu i Wankhadeu [14] kao što i samo ime govori, donošenje odluka u jednoj fazi (engl. *One stage decision making*) sastoji se od:

- identifikacija tijeka akcija dostupnih donositelju odluke u različitim mogućim događajima,
- razvoj matrice plaćanja,
- odabir tijeka akcija u suglasnosti s određenim kriterijem.

Za probleme koji se rješavaju donošenjem odluka u jednoj fazi razvijaju se tablica plaćanja i tablica žaljenja. Plaćanje je uvjetna vrijednost ili uvjetni profit ili gubitak. Plaćanje je uvjetnog karaktera jer je povezano s određenim tijekom akcija i potencijalnim događajima. Tablica plaćanja predstavljena je matricom uvjetnih vrijednosti povezanih sa svim mogućim kombinacijama akcija i događaja. Rezultirajući ishodi različitih kombinacija akcija i događaja (stanja okoline) mogu biti alternativno izraženi terminom gubitka prilike (engl. *Opportunity loss*) ili žaljenja. Žaljenje je definirano kao iznos plaćanja koji je izgubljen kao rezultat pogrešne odluke, odnosno sub-optimalnog tijeka akcija. Matricu plaćanja moguće je transformirati u matricu žaljenja na način da se oduzme gubitak od maksimalnog plaćanja za svaki redak. Nakon postavljanja tablice plaćanja slijedi proces donošenja odluke. Postoje nekoliko kriterija ili pravila donošenja odluke koji ovise o stavu prema riziku samog donositelja odluke, a to su **kriteriji za donošenje odluka u uvjetima rizika** [14]:

1. načelo maksimalne vjerojatnosti - kriterij u kojem donositelj odluke prvo razmatra događaj koji je najvjerojatniji. Nakon toga odlučuje o tijeku akcija koji ima maksimalno uvjetno plaćanje u odnosu na ovaj događaj,
2. princip očekivanja - nakon što se postave vjerojatnosti pojave određenog događaja (objektivno ili subjektivno), donositelj odluke računa očekivano plaćanje na način da množi potencijalno plaćanje s vjerojatnosti pojave događaja te se na kraju odabire onaj tijek aktivnosti koji ima najveće potencijalno plaćanje.

4.2. Kriteriji odlučivanja u uvjetima nesigurnosti i rizika

Sikavica, Hunjak, Begičević Ređep i dr. [1] navode da se pod odlučivanjem u uvjetima nesigurnosti smatra svaka situacija kada donositelj odluka ne raspolaže svim informacijama

koje bi mu omogućile donošenje procjene o tome koja vjerojatnost se može pridružiti pojedinoj posljedici. Informacije kojima donositelj odluke raspolaže u takvoj situaciji su:

- aktivnosti (odluke) koje može poduzeti,
- posljedice svake od mogućih odluka s obzirom na situacije u kojima će se te odluke realizirati.

Informacije za opis problema u uvjetima neizvjesnosti prikazuju se u obliku tablice (matrice) plaćanja prikazane tablicom 13.

	Stanja okoline			
Alternative	$S1$	$S2$	...	S_n
A_1	v_{11}	v_{12}	...	v_{1n}
A_2	v_{21}	v_{22}	...	v_{2n}
...	...	...	...	...
A_m	v_{m1}	v_{m2}	...	v_{mn}

Tablica 1: Matrica plaćanja (Izvor: vlastita izrada prema [4, str. 463])

Slijede objašnjenja oznaka matrice:

- A_i - i-ta aktivnost (akcija, odluka),
- S_j - j-to stanje okoline,
- v_{ij} - plaćanje (posljedica) poduzimanja akcija A_i ako je okolina u stanju S_j.

Donositelj odluke u situaciji odlučivanja prikazanoj na slici 2 pri donošenju odluke može primijeniti različite kriterije odlučivanja [1]:

1. kriterij maksimin (Waldow kriterij),
2. kriterij maksimaks,
3. Hurwitzov kriterij,
4. kriterij minimalnog žaljenja (Savageov kriterij),
5. praktičan pristup,
6. Laplaceov kirterij.

Kako bi se lakše konceptualizirali kriteriji, koristiti će se numerički primjer predstavljen od strane autora Sikavica, Hunjak, Begičević Ređep i dr. [1], koji se odnosi na prodavača novina. **Primjer**: *Prodavač novina nabavlja novine po 5 kn, a prodaje ih po 7kn. Po svakom prodanom primjerku ostvaruje zaradu od 2kn. Od neprodanih zaliha prodavač ne ostvaruju nikakvu dobit.*

U trenutku odluke o količini novina koju naručuje, prodavač ne zna koliko će se novina sutra zapravo prodati te se izlaže situaciji neizvjesnosti u smislu naručivanja pre malog ili pre velikog broja novina. Dakle radi se o problemu nesigurnosti, a iz tablice 2 mogu se vidjeti plaćanja svake pojedine kombinacije aktivnosti i stanja okoline. Iz tablice se vidi da ako prodavač naruči 13 primjeraka, a proda 11 novina, zaraditi će samo 12 kn, dok će u slučaju 13 prodanih novina zaraditi 26 kn. Ključno pitanje je kakvu odluku donijeti te je za njegovu odluku ključan njegov stav prema riziku.

	Potražnja					
Naručeno	10	11	12	13	14	15
10	20	20	20	20	20	20
11	15	22	22	22	22	22
12	10	17	24	24	24	24
13	5	12	19	26	26	26
14	0	7	14	21	28	28
15	-5	2	9	16	23	30

Tablica 2: Matrica plaćanja - zarada od prodaje novina ovisno o potražnji (Izvor: vlastita izrada prema [4, str. 465])

4.2.1. Maksimin kriterij (Waldov kriterij)

Maksimin kriterij odlučivanja u uvjetima nesigurnosti karakterizira odbojnost prema riziku. Za svaku akciju A_i identificira se najslabiji ishod v_{i*} i bira se akcija A_k za koje je ta vrijednost najveća tj. tako da vrijedi [1]:

$$v_{k^*} = \max_i v_{i^*} = \max_i(\min_i v_{ij}). \tag{4.1}$$

Nadalje, primjenom formule 4.1 na prethodno predstavljeni problem odlučivanja s prodavačem novina dobivamo tablicu 3. Za svaku pojedinu alternativu od A_1 do A_6 bira se najgori scenarij, odnosno najmanje plaćanje v_{ij} te se naposljetku odabire akcija A_1 jer ona u najgorem scenariju daje najveći povrat.

	Potražnja						v_{i*}
Naručeno	10	11	12	13	14	15	
10	20	20	20	20	20	20	**20**
11	15	22	22	22	22	22	15
12	10	17	24	24	24	24	10
13	5	12	19	26	26	26	5
14	0	7	14	21	28	28	0
15	-5	2	9	16	23	30	-5

Tablica 3: Odluka - maksimin (Izvor: vlastita izrada)

4.2.2. Maksimaks kriterij

Za razliku od maksimin kriterij, maksimaks kriterij karakterizira sklonost riziku. Za svaku aktivnost A_i identificira se najbolji ishod v_{i*} te se bira A_k koja maksimizira vrijednost, odnosno takva da [1]:

$$v_k^* = \max_i v_i^* = \max_i(\max_i v_{ij}). \tag{4.2}$$

	Potražnja						v_i*
Naručeno	10	11	12	13	14	15	
10	20	20	20	20	20	20	20
11	15	22	22	22	22	22	22
12	10	17	24	24	24	24	24
13	5	12	19	26	26	26	26
14	0	7	14	21	28	28	28
15	−5	2	9	16	23	30	**30**

Tablica 4: Odluka - maksimaks (Izvor: vlastita izrada)

Primjenom maksimaks kriterija za svaku pojedinu alternativu odabire se najbolji scenarij te se na kraju odabire akcija A_i koja daje maksimalni povrat. Dakle, primjenom formule 4.2 na predstavljeni problem, ispravna odluka je A_6, odnosno naručiti 15 novina te bi ta odluka donijela najveće moguće potencijalno plaćanje.

4.2.3. Hurwitzov kriterij

Hurwitzov kriterij uvodi koncept modeliranja optimizma. Prethodna dva spomenuta kriterija svrstavaju ljude prema sklonosti riziku u one koji su skloni riziku i one koji izbjegavaju rizik. Pomoću Hurwitzovog kriterija i parametra optimizma α t.d. $0 < \alpha \leq 1$ moguće je modeliranje optimizma, odnosno za svaku akciju A_i računa se Hurwitzova vrijednost [1]:

$$H(i) = \alpha v_i^* + (1-\alpha)v_{i^*} \tag{4.3}$$

	Potražnja						v_{ij*}	$v_{ij}*$	$H(i) = \alpha \cdot v_{ij*} + (1-\alpha)v_{ij}*$		
Nar.	10	11	12	13	14	15			0.3	0.5	0.8
10	20	20	20	20	20	20	20	20	**20**	**20**	20
11	15	22	22	22	22	22	15	22	17.1	18.5	20.6
12	10	17	24	24	24	24	10	24	14.2	17	21.2
13	5	12	19	26	26	26	5	16	11.3	15.5	21.8
14	0	7	14	21	28	28	0	28	8.4	14	**22.4**
15	−5	2	9	16	23	30	−5	30	5.5	12.5	0.5

Tablica 5: Odluka - Hurwitz (Izvor: vlastita izrada)

Na temelju vrijednosti parametra optimizma α bira se akcija A_i koja maksimizira $H(k)$, a sam α karakterizira stav donositelja odluke prema riziku. Nadalje, primjenom Hurwitzovog kriterija na prethodno predstavljeni problem prodaje novina, za svaku kombinaciju akcije i stanja v_{ij} i vrijednosti parametra optimizma računa se H_i za vrijednost tog parametra. Tako je npr. za vrijednost parametra optimizma $0,3$ ispravna odluka naručiti 10 primjeraka novina, dok je za vrijednost $\alpha = 0,8$ ispravna odluka naručiti 14 primjeraka novina. Nadalje, indeks optimizma može se lako podesiti temeljem nečije sklonosti prema riziku te je to njegova glavna prednost.

4.2.4. Kriterij minimalnog žaljenja (Savageov kriterij)

Žaljenje se definira kao osjećaj za propuštenom prilikom, odnosno ukoliko u zadanom problemu prodavač naruči samo 12 primjeraka novine, a drugi dan se pojavi npr. 15 kupaca, prodavač novina će najvjerojatnije osjećati žaljenje za propuštenom prilikom [1]. Nadalje prema Sikavici, Hunjaku, Begičević Ređep i dr. [1], ako želimo donijeti odluku s ciljem minimalizacije žaljenja zbog propuštene zarade, postupamo na sljedeći način:

1. za svaku akciju računa se žaljenje u odnosu na svako stanje okoline, tj. izrađuje se tablica žaljenja pomoću: $r_{ij} = \max_k v_{kj} - v_{ij}$ (razlika između najboljeg mogućeg ishoda zbog nastupanja stanja j i poslijedice ako je izabrana akcija A_i),
2. za svaku akciju A_i računa se maksimalno žaljenje: $\rho_i = \max_j r_{ij}$ (najgori ishod za akciju A_i),
3. bira se akcija A_k koja ima minimalnu vrijednost maksimalnog žaljenja, odnosno za koju vrijedi:
   $\rho_k = \min_i(\max_j r_{ij})$.

Savageov kriterij baziran je na činjenici da odlučitelj ne može utjecati na stanje okoline, ali posljedice proizlaze iz njegovog izbora akcije. Stoga se mogu uspoređivati posljedice različitih akcija koje bi on mogao poduzeti u odnosu na pojedino stanje.

	10	11	12	13	14	15
10	20	20	20	20	20	20
11	15	22	22	22	22	22
12	10	17	24	24	24	24
13	5	12	19	26	26	26
14	0	7	14	21	28	28
15	−5	2	9	16	23	30
$\max v_{kj}$	20	22	24	26	28	30

Tablica 6: Računanje maksimalnog ishoda za svaku kombinaciju akcije i stanja okoline (Izvor: vlastita izrada)

Tablica 6 prikazuje izračun maksimalnih plaćanja za svaku pojedinu kombinaciju akcije a_k i stanja s_j. Nakon što se izračuna $maxv_{kj}$, računa se dalje: r_{ij} i ρ_k.

	10	11	12	13	14	15	ρ_i
10	20	20	20	20	20	20	10
11	15	22	22	22	22	22	**8**
12	10	17	24	24	24	24	10
13	5	12	19	26	26	26	15
14	0	7	14	21	28	28	20
15	-5	2	9	16	23	30	25

Tablica 7: Tablica žaljenja i izračun ρ_i (Izvor: vlastita izrada)

Tablica 6 prikazuje izračun vrijednosti $\max v_{ij}$, a navedene vrijednosti prikazane su u ćelijama na presjeku vrijednosti akcija i stanja. Dalje se računa ρ_i prema tablici 7, odnosno maksimalno žaljenje za svaku akciju A_i. Nakon toga primjenom formule iz trećeg koraka lako se može iščitati da je optimalna odluka primjenom Savageovog kriterija akcija A_2.

4.2.5. Praktičan pristup

Praktičan pristup pretpostavlja poznavanje vjerojatnosti nastupanja određenog događaja, odnosno stanja okoline. Situacija u kojoj su poznate vjerojatnosti nastupanja stanja okoline u kojima će se realizirati posljedice odluke naziva se odlučivanju u uvjetima rizika. Ključni kriterij koji se primjenjuje u uvjetima rizika jest **očekivana vrijednost** ili kriterij očekivane vrijednosti [1]. Ako od ostvarenoga događaja D koji ima vjerojatnost p očekuje dobit visine D, očekivana vrijednost definira se kao produkt:

$$E = D \cdot p. \tag{4.4}$$

U takvom slučaju odlučivanja, koji podrazumijeva poznavanje vjerojatnosti pojave određenog stanja okoline, za svaku pojedinu akciju računa se njezina očekivana vrijednost, tj.:

$$EV(A_i) = \sum_j v_{ij} P(S_j) \tag{4.5}$$

S_j	10	11	12	13	14	15	
P_{Sj}	0.05	0.15	0.25	0.25	0.2	0.1	
A_i							$EV(A_{ij})$
10	20	20	20	20	20	20	20
11	15	22	22	22	22	22	21.65
12	10	17	24	24	24	24	**22.25**
13	5	12	19	26	26	26	21.1
14	0	7	14	21	28	28	18.2
15	-5	2	9	16	23	30	13.9

Tablica 8: Primjena kriterija očekivane vrijednosti na problem prodaje novina (Izvor: vlastita izrada)

U navedenom slučaju kao najbolja odluka uzima se ona akcija koja daje najveću očekivanu vrijednost [1]. Ako stanjima okoline iz prethodnog primjera pridružimo vjerojatnosti pojave istih, možemo primijeniti kriterij očekivane vrijednosti na prethodno predstavljeni problem. Tablica 8 prikazuje primjenu praktičnog pristupa, odnosno kriterija očekivane vrijednosti na prethodno predstavljeni problem prodaje novina. Stanjima okoline S_j pridružene su vjerojatnosti nastupanja stanja $P(S_{ij})$ te se prema formuli 4.5 lako izračuna očekivana vrijednost za svaku akciju, odnosno računa se suma umnožaka pojedinačnih akcija i svih stanja okoline pomnoženih s vjerojatnostima nastupanja istih. Primjenom formule 4.5 na predstavljeni problem lako se zaključi da je najbolja odluka prema navedenom kriteriju naručiti 12 primjeraka novina, odnosno odlučiti se za akciju A_3.

4.2.6. Laplaceov kriterij

Laplaceov kriterij također se temelji na pojmu očekivane vrijednosti i pretpostavci da donositelj odluke ne poznaje vjerojatnosti nastupanja mogućih stanja okoline te slobodno može pretpostaviti da su vjerojatnosti nastupanja svakog pojedinog stanja jednake [1].

A_i	10	11	12	13	14	15
$P_{(Sj)}$	0.166667	0.166667	0.166667	0.166667	0.166667	0.166667
$EV_{(Aii)}$	20	20.83	20.5	19	16.33	12.5

Tablica 9: Primjena Laplaceovog kriterija na problem prodaje novina (Izvor: vlastita izrada)

Tablica 9 prikazuje primjenu Laplaceovog kriterija na problem prodaje novina. Svakom stanju okoline S_j pridružena je ista vjerojatnost pojave koja iznosi približno 1,66667. Primjenom formule 4.5 na zadani problem lako se zaključi da je prema Laplaceovom kriteriju najbolja odluka A_2, odnosno narudžba od 11 novina.

4.3. Praktične metode za odlučivanje u uvjetima nesigurnosti i rizika

Praktične metode koje se koriste za odlučivanje u uvjetima nesigurnosti i rizika koje će biti detaljnije razrađene u sklopu ovog poglavlja su:

- stablo odlučivanja,
- matrica rizika,
- analiza osjetljivosti.

4.3.1. Stabo odlučivanja

Prema Sikavici, Hunjaku, Begičević Ređep i dr. [1] **stablo odlučivanja** (engl. *Decision tree*) je grafički model za vizualizaciju procesa odlučivanja kad se rješavanje problema odlučivanja svodi na donošenje više sukcesivnih odluka te se primjenjuje u odlučivanju, odnosno u donošenju kompleksnih odluka. Uz stablo odlučivanja veže se postupak računa **očekivane vrijednosti** inačica odluka u uvjetima rizika. Kriterij očekivane vrijednosti za donošenje odluka primjenjuje donositelj odluke koji ima neutralno stajalište prema riziku, a on glasi:

$$EV(A_i) = \sum_j v_{ij} P(S_j) \tag{4.6}$$

Primjenom kriterija očekivane vrijednosti bira se ona akcija koja ima najveću očekivanu vrijednost (primjer primjene kriterija očekivane vrijednosti na prethodno predstavljenom problemu prodaje novina može se vidjeti u tablici 8.) [1].

Nadalje, kako bi se jednostavnije prikazala primjena metode stabla odlučivanja u procesu odlučivanja, koristi se primjer predstavljen od strane autora Sikavice, Hunjaka, Begičević Ređep i dr. [1].

Primjer:

Poduzetnik razmatra lansiranje nove linije proizvoda te je do sadašnjeg trenutka uložio znatna sredstva u njezin razvoj, ali za to mu je potrebno ulaganje u visini 200000 kn bez sigurnosti u budući poslovni uspjeh novog proizvoda. Poduzetnik procjenjuje da postoji 50% vjerojatnost da potražnja bude dovoljno velika za ostvarivanje financijske dobiti od 500000 kn, 30% da potražnja bude mala, odnosno da se ostvari financijska dobit od 100000 kn te 20% vjerojatnost da novi proizvod neće ostvariti nikakvu potražnju, odnosno da ostvari poslovni gubitak od 200000 kn. Situacija odlučivanja predstavljena primjerom vidljiva je u tablici 10.

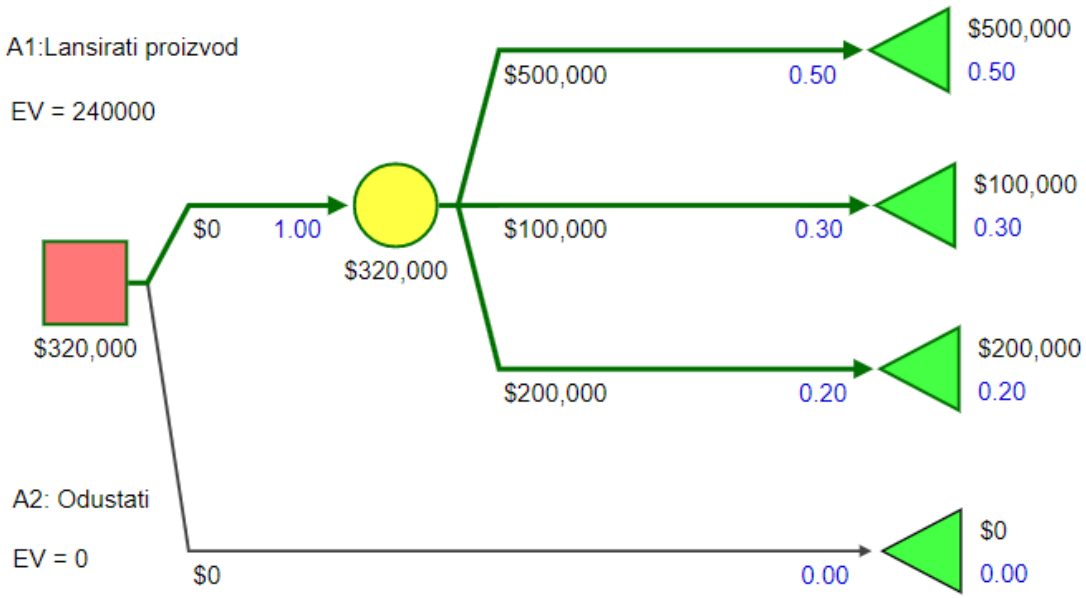
	$VP(0.5\%)$	$MP(0.3\%$	$BP(0.2\%)$	EV
Lansirati proizvod	500000	100000	-200000	240000
Odustati	0	0	0	0

Tablica 10: Tablica odlučivanja za problem lansiranja novog proizvoda (Izvor: vlastita izrada)

Ako primijenimo metodu stabla odlučivanja na zadani problem, prva odluka - lansiranje proizvoda dobiva tri različite grane kao što je prikazano na slici 4.

Simboli koji se koriste u ovom grafičkom modelu su:

- kvadrat - čvor odluke,
- krug - čvor slučaja,
- grane - povezuju čvorove u stablu,
- trokut - krajnji čvor .

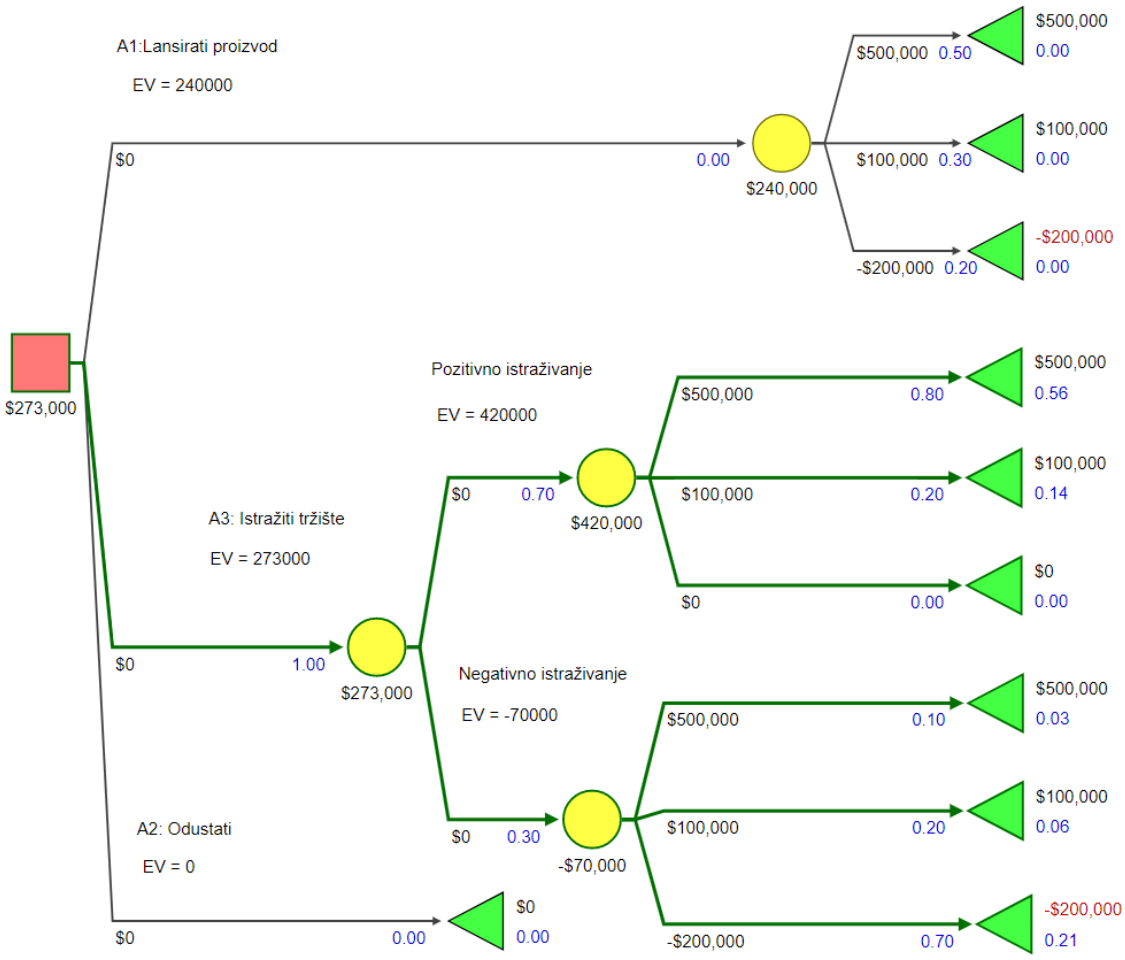


Slika 4: Stablo odlučivanje za problem lansiranja proizvoda (Izvor: vlastita izrada)

Čvor odluke je čvor koji je pod kontrolom donositelja odluke, dok čvor slučaja nije pod kontrolom odlučitelja, odnosno on sadrži sve mogućnosti koje se mogu dogoditi te je izvor potencijalnih rizika. Ishodi slučaja navode se na kraju svake grane te su ovisno o alatu prikazani trokutom ili je naznačena numerička vrijednost na kraju grane. Nadalje, na temelju ishoda i vjerojatnosti slučajeva računa se očekivana vrijednost čvora slučaja. U skladu s kriterijem očekivane vrijednosti, odluka se donosi na način da se u čvoru odluke bira ona grana koja vodi prema čvoru slučaja s najvećom očekivanom vrijednošću [1]. Prema modelu prikazanom na slici 4, za problem lansiranja proizvoda ispravna odluka bi bila A1 - lansirati proizvod. Konačna odluka također ovisi i o stajalištu prema riziku samog donositelja odluke, npr. ukoliko je za poduzetnika vjerojatnost od 20% da projekt propadne prevelika on će najvjerojatnije donijeti odluka o odustajanju. Nadalje, kada govorimo o metodi stabla odlučivanja, nije moguće zaobići situaciju povezanih odluka. U nekim djelatnostima nije moguće donositi jednokratne odluke zbog prirode poslovnih procesa. Takve povezane poslovne odluke donose se i modeliraju uz pomoć metode stabla odlučivanja. Kako bi se objasnio postupak povezanih odluka proširuje se prethodni primjer inovacije proizvoda [1].

Primjer:

Nova pretpostavka je da se prije izlaska na tržište izvrši istraživanje tržišta. Marketinška agencija daje procjenu da postoji 70 % vjerojatnosti za pozitivan rezultat istraživanja te da su vjerojatnosti stanja tržišta u tom slučaju: p(VP) = 80%, p(MP) = 20%, p(BP) = 0. Nadalje, u slučaju negativne analize tržišta za koju procjenjuju da je vjerojatnost 30%, procjenjuju: p(VP) = 10%, p(MP) = 20%, p(BP) = 70%.



Slika 5: Stablo odlučivanje za problem lansiranja proizvoda s istraživanjem tržišta (Izvor: vlastita izrada)

Slika 5 prikazuje model stabla kao rješenje proširenog problema lansiranog proizvoda. Primjenom kriterija očekivane vrijednosti lako se vidi temeljem modela da je ispravna odluka, odluka A3 koja maksimizira očekivanu vrijednost. Nadalje, za donositelja odluke iz prethodnog primjera važno je pitanje kolika je isplativost istraživanja tržišta, odnosno koja mu je cijena. Ako se pretpostavi da će informacija dobivena istraživanjem tržišta biti u potpunosti točna, govorimo o **cijeni potpune informacije** (engl. *Expected Value Of Perfect Information - EVPI*). EVPI računamo pomoću formule [1]:

$$EVPI = EV(VP) + EV(MP) = 0,5 \cdot 500000 + 0,3 \cdot 100000 = 280000 \tag{4.7}$$

Prethodno se utvrdilo da je EV izlaska na tržište za zadani problem bez istraživanja tržišta = 240000 kn, a očekivana vrijednost odluke s potpunom informacijom EVPI = 280000 kn, lako se može zaključiti da se za dodatnu informaciju o stanju tržišta ne isplati platiti više od razlike tih iznosa, odnosno 40000 kn.

4.3.2. Matrica rizika

Prema Sikavici, Hunjaku, Begičević Ređep i dr. [1] **matrica rizika** je jednostavan grafički alat za vizualizaciju rizika s obzirom na dvije dimenzije:

- vjerojatnost rizičnog događaja,
- razina posljedice (najčešće u monetarnom obliku).

Pri tom se rizik definira kao produkt vjerojatnosti i posljedice. Matrica rizika primjenjuje se da bi se odredili prioriteti između rizičnih događaja u procesu upravljanja rizicima. Tretman rizičnog događaja ovisi o poziciji koju zauzima u samoj matrici. Što je događaj u većoj mjeri "dolje - lijevo" veća je i spremnost donositelja odluke da ga se prihvati bez posebne brige. Nadalje, s pomicanjem u smjeru "gore - desno" po matrici rizika raste i vjerojatnost nastupanja rizičnog događaja i raste važnost njegove posljedice te se riziku posvećuje veća pozornost. Najčešće se u matrici rizika rizici označavaju bojama, pa što je događaj rizičniji, to je označen tamnijom nijansom crvene boje. Osim toga, ljestvica pomoću koje se procjenjuje vjerojatnost ovisi o prirodi događaja, pa se uz numeričke vrijednosti i procjene učestalosti nastupanja u odnosu na neko razdoblje, često primjenjuju i kvalitativne procjene [1]. Primjer matrice rizika može se vidjeti na slici 6.



Slika 6: Primjer matrice rizika (Izvor: vlastita izrada)

4.3.3. Analiza osjetljivosti

Analiza osjetljivosti (engl. *Sensitivity analysis*) jedan je od jednostavnijih postupaka pomoću kojih se prevladava nesigurnost i rizik u poslovnom odlučivanju, a provodi se uz pomoć determinističkog kvanitativnog modela. Postupak se sastoji od analize u kojoj mjeri moguće promjene ulaznih varijabli modela utječu na izlazne vrijednosti. Osim toga temeljem analize

osjetljivosti mogu se procijeniti posljedice realizacije neke odluke u različitim scenarijima [1]. Nadalje, definicija analize osjetljivosti prema Kurhadeu i Wankhadeu [14] govori da, analize osjetljivosti nisu ništa drugo nego identifikatori glavnih izvora nesigurnosti. Provode se s ciljem identifikacije varijabli koje uvelike doprinose razini sigurnosti predviđanja. Ulazne varijable koje imaju visoku osjetljivost u odnosu na buduća predviđanja mjere se na početku analize te se prilikom predviđanja u obzir uzimaju faktori koji su najosjetljiviji u donošenju odluka. Glavni cilj analize osjetljivosti je identifikacija glavnih efekata i interakcije efekata ulaznih varijabli. Nesigurnosti se najčešće pojavljuju kao rezultati grešaka u podacima i grešaka predviđanja. Nadalje, postupak analize osjetljivosti je prema Sikavici, Hunjaku, Begičević Ređep i dr. [1] određen matematičkim modelom na kojem se temelji odluka i iz svojstva tog modela proizlaze mogućnosti da se odgovori na određeni broj pitanja, čime se postiže veća sigurnost u predviđanju, odnosno donosi bolja odluka. Postupak analize osjetljivosti može se lako ilustrirati uz pomoć jednostavnog modela točke pokrića predstavljen od autora Dvorski i Kovšca (kao što se citira u [1]).

Matematički model točke pokrića:

- ulazni podaci: P_c (prodajna cijena), v_t (varijabilni troškovi prosjek), FT (fiksni troškovi), q (količina proizvoda),
- relacije: $UP = q \cdot p_c$ (ukupan prihod), $VT = q \cdot v_t$ (varijabilni trošak uk.), $UT = FT + VT$ (ukupni troškovi), $FR = UP - UT$ (financijski rezultat),
- izlazni podaci: $q_p = FT/(p_c - v_t)$ (broj proizvoda čijom prodajom se pokrivaju ukupni troškovi).

Kako bi se analizirala isplativost npr. proizvodnje nekog proizvoda pomoću prethodno predstavljenoga modela utvrđuje se uzročno posljedična veza između ulaznih i izlaznih varijabli, odnosno u kojoj mjeri prodajna cijena, varijabilni, fiksni troškovi i količina proizvoda utječu na pokrivanje troškova proizvodnje (odnosno ostvarivanje dodatnog profita). Poznavanje točke pokrića (odnosno točke u kojoj su troškovi nastali od ulaza jednaki prihodima ostvarenima prodajom proizvoda) vrlo je važno za planiranje proizvodnje, a i za kasnije predviđanje prodaje.

5. Stablo odlučivanja

Olivas [15] navodi da **stabla odlučivanja** imaju široki raspon različitih primjena. Koriste se u mnogim različitim disciplinama, uključujući medicinsku dijagnozu, kognitivne znanosti, umjetnu inteligenciju, teoriju igara, inženjering i rudarenje podacima. Unatoč ovom trendu postoji vrlo malen broj kvalitetnih i jasnih uvoda u koncepte stabla odlučivanja. Stabla odlučivanja nude velik broj prednosti u odnosu na druge metode analize alternativa, a to su [15]:

- **grafički prikaz** - stabla odlučivanja omogućuju grafičku prezentaciju alternativa, mogućih ishoda i njihovih vjerojatnosti na jednostavan i shematski način. Vizualni pristup je posebno koristan u razumijevanju sekvencijalnih (povezanih) odluka te povezanih ishoda,
- **učinkovitost** - pomoću stabla odlučivanja moguće je brzo izraziti kompleksne alternative na jasan način. Moguće je vrlo jednostavno modificirati stablo odlučivanja pomoću novih informacija kada one postanu dostupne. Standardna notacija za stabla odlučivanja je vrlo jednostavna za usvojiti i primijeniti,
- **otkrivanje znanja** - stabla odlučivanja omogućuju korisniku usporedbu alternativa koje se razmatraju - čak i bez potpune informacije u smislu rizika i nesigurnosti. Pojam očekivane vrijednosti EV kombinira relativne troškove investicije, predviđena plaćanja i vjerojatnosti u jedinstvenu numeričku vrijednost. Očekivana vrijednost otkriva ukupne vrijednosti svih razmatranih alternativa,
- **dopuna postojećim alatima** - stablo odlučivanja može se koristiti u kombinaciji s drugim alatima projektnog menadžmenta. Na primjer pomoću metode stabla odlučivanja moguće je evaluirati projektni plan/raspored.

5.1. Uvod u stablo odlučivanja

Prema Sikavici, Hunjaku, Begičević Ređep i dr. [1] **Stablo odlučivanja** (engl. *Decision tree*) je grafički model za vizualizaciju procesa odlučivanja kad se rješavanje problema odlučivanja svodi na donošenje više sukcesivnih odluka te se primjenjuje u odlučivanju, odnosno u donošenju kompleksnih odluka.

Druga definicija stabla odlučivanja prema Olivasu [15] govori da je stablo odlučivanja metoda koja korisniku omogućuje donošenje ispravnih odluka u situacijama velikih uloga i rizika. Stablo odlučivanja koristi grafički pristup kako bi se usporedile moguće alternative i dodijelile im se vrijednosti kombinacijom vjerojatnosti, troškova i plaćanja u specifične numeričke vrijednosti. Stablo odlučivanja nije ništa drugo nego grafički prikaz mogućih alternativa i njihovih posljedica.

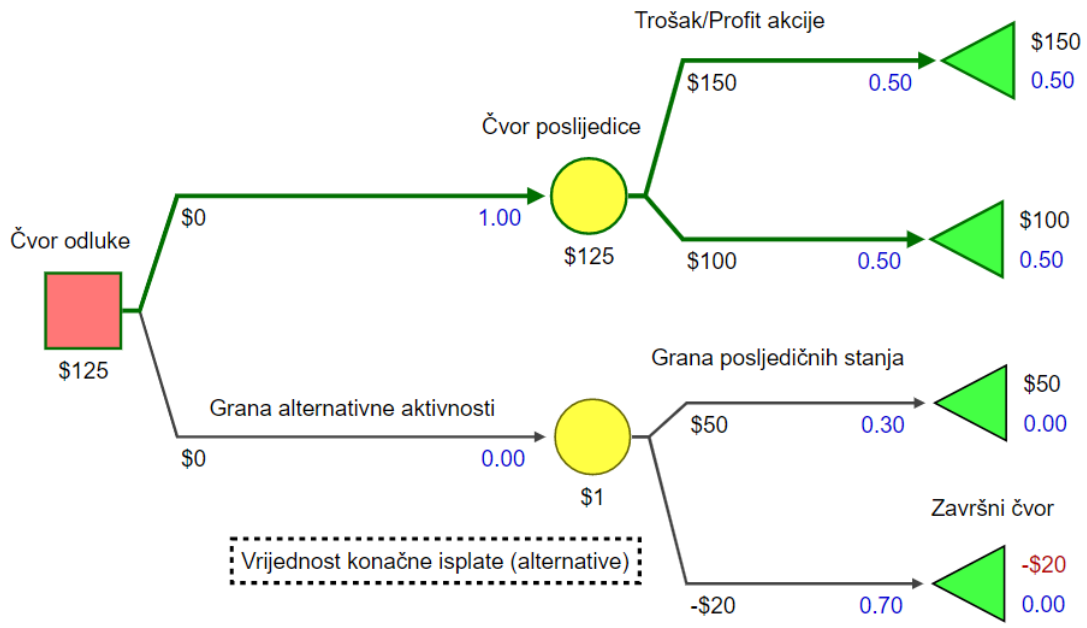
Nadalje, prema Begičević Ređep [16] odluka se najčešće ne može promatrati kao izolirani, pojedinačni događaj, već kao prva u nizu nekoliko međusobno povezanih odluka tijekom nekog budućeg perioda. Donositelj odluke prije donošenja odluke razmatra čitave

serije odluka istovremeno. Stablo odlučivanja se pokazalo kao jedan od najpraktičniji alata za rješavanje ovakvih problema odlučivanja.

5.2. Elementi stabla odlučivanja

Slika 7 prikazuje jednostavno stablo odlučivanje pomoću kojeg će se lakše objasniti elementi i terminologija samog stabla. Prema Begičević Ređep [16] osnovni elementi stabla odlučivanja su:

1. **čvorovi odluke** - u modelu se obično prikazuju pravokutnikom. To je čvor kojem donositelj odluke treba odlučiti za jedan alternativni smjer aktivnosti od ograničenog broja alternativa,
2. **grane alternativnih aktivnosti** - prikaz alternativnih smjerova aktivnosti koji izlaze s desne strane čvora odluke. Sadrže informacije o alternativnim aktivnostima od kojih jednu biramo kao odluku. Bira se ona alternativa za koju se očekuje da će donijeti najviše koristi za odlučitelja,
3. **trošak/profit akcije** - pridružuje se grani svake od alternativa. Radi se o cijeni koštanja ili dobitka alternative. Ovaj trošak/prihod koristi se kod računanja konačne vrijednosti alternativa na listovima stabla odlučivanja. Ovaj trošak/prihod se odnosi i na alternativne odluke i na moguća posljedična stanja kada se zadaje i vjerojatnost nastanka istih,
4. **čvorovi mogućih posljedica** - na modelu se prikazuju kružićem te daju informaciju o



Slika 7: Jednostavno stablo odlučivanja s jednim čvorom odluke. (Izvor: vlastita izrada)

tome koji se događaji mogu očekivati u određenoj točki procesa odlučivanja,

5. **grane mogućih posljedičnih stanja** - moguća posljedična stanja prikazuju se granama desno od čvorova mogućih posljedica. Svakoj grani posljedičnih stanja, pridružuje se procijenjena vjerojatnost njene pojave i upisuje se iznad ili ispod grane,
6. **završni čvorovi** - nazivaju se još i čvorovi posljedica ili listovi,
7. **vrijednost konačne isplate financijske koristi** - očekivana vrijednost odabrane alternative.

5.3. Koraci metode stabla odlučivanja

Prema Begičević Ređep [16] prije konstrukcije modela stabla odlučivanja opisno se zadaje problem odlučivanja koji služi za konstrukciju stabla odlučivanja. Takav model sadrži sve informacije o alternativama koje će se u stablu koristiti, kao i njima pridruženi izvori podataka. Koraci izgradnje stabla odlučivanja i donošenje odluka sastoje se od [16]:

1. **izgradnja logičkog modela stabla odlučivanja** - svim čvorovima odluka i posljedica te granama alternativnih akcija i posljedičnih stanja pridružuju se u kronološkom redoslijedu:
   - vjerojatnosti pojave pojedine posljedice,
   - parcijalni tok novca (potreban priljev ili odljev sredstava kako bi se stablo nastavilo granati),
2. **računanje očekivanih vrijednosti odluka uz pomoć postupka računanja unatrag** (engl. *Rollback algorithm)* - postupak računanja unatrag započinje na krajnjim čvorovima stabla i kreće unatrag, sve do početnog čvora odluke. Svakom čvoru se pridružuje ekvivalentna očekivana vrijednost, odnosno:
   - na završnom čvoru izračunata je konačna vrijednost alternative (ulazni podatak),
   - čvoru posljedica pridružuje se očekivane vrijednosti izračunate kao:

$$EV_{i-1} = \sum_j p_j EV_i, i \in \{1, 2, ..., n\}, j \in \{1, 2, ..., m\} \tag{5.1}$$

     - EV_{i-1} - očekivana vrijednost u i-1 čvoru,
     - EV_i - očekivana vrijednost u čvoru i,
     - p_j - vjerojatnost grane j koja izlazi iz čvora posljedica i-1,
   - čvoru odluka pridružena je najveća od prethodno izračunatih očekivanih vrijednosti neposrednih sljedećih čvorova u stablu odlučivanja,
3. **pronalaženje optimalnog puta postupkom računanja unaprijed** - nakon što su izračunate sve očekivane vrijednosti za svaki čvor, lako se može prepoznati optimalan put u stablu odlučivanja, računanjem prema naprijed od početnog čvora odluke. Očekivana vrijednost početnog čvora jednaka je očekivanoj vrijednosti grane koja se nalazi na optimalnom putu. Analogno se promatra sljedeći čvor sve do završnog čvora.

5.4. Programska podrška za metodu stabla odlučivanja

Autori Sikavica, Hunjak, Begičević Ređep i dr. [1] navode da su najpoznatiji komercijalni alati, odnosno programska podrška za korištenje metode stabla odlučivanja TreePlan i PrecisionTree. Nadalje, oba spomenuta alata su kompatibilna s najpoznatijim tabličnim kalkulatorom Microsoft Excel.

5.4.1. TreePlan

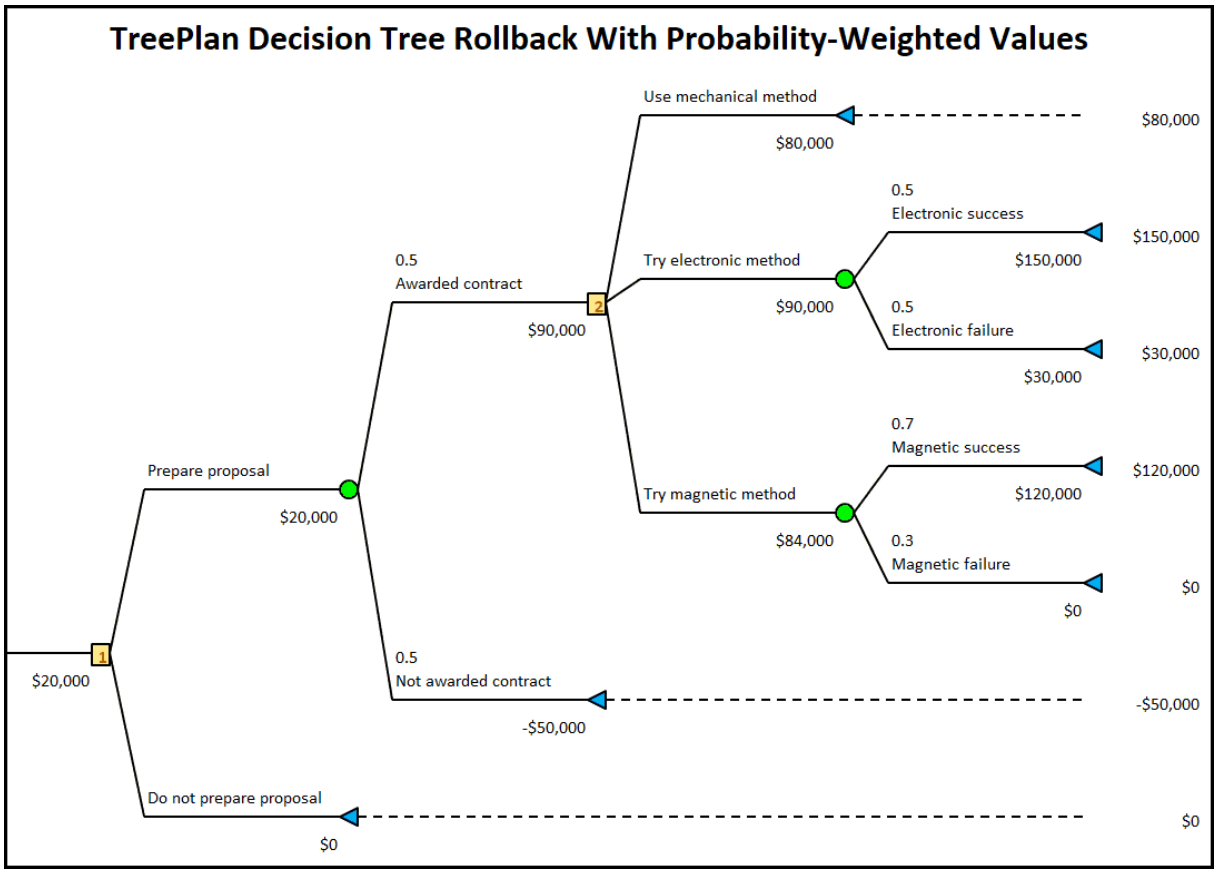
TreePlan pomaže korisnicima u kreiranju dijagrama stabla odlučivanja u sklopu Excel sučelja korištenjem dijaloških okvira [17]. TreePlan omogućuje kreiranje formula za sumiranje tokova novca kako bi se dobile izlazne vrijednosti pomoću *rollback* algoritma za kreiranje optimalne strategije odlučivanja.
Prednosti TreePlan programskog alata [17]:

- smanjivanje vremena potrebnog za kreiranje modela stabla odlučivanja,
- dva načina korištenja: brzo pokretanje bez instalacije i permanentna instalacija,
- formatiranje stabla odlučivanja,
- korištenje TreePlan softvera za selekciju grupa ćelija ili oblika i formatiranje pomoću Excelovih komandi,
- korištenje TreePlan dijagrama za objašnjenje analize kolegama,
- provjera hipoteza uz pomoć "što-ako" analiza korištenjem Excel komandi,
- integriranje modela stabla odlučivanja s ostalim modelima koje nudi Microsoft Excel.

Proces kreiranja dijagrama stabla odlučivanja uz pomoć TreePlan alata obuhvaća [17]:

1. otvaranje Excel proračunske tablice,
2. odabir alata | *"Decision Tree or Add-Ins"* | *"Decision Tree (Excel 2010 & 2013 & 2016)"*,
3. klik na *"New Tree"* gumb i pokretanje TreePlan-a za kreiranje inicijalnog stabla s jednim čvorom i dvije grane,
4. selektiranje čvora i pokretanje TreePlan-a za primjenu strukture stabla,
5. unos imena grana, tokova novca i vjerojatnosti,
6. odabir optimalne strategije uz pomoć modela za odabrani problem.

Slika 8 prikazuje demonstrativan primjer generiranog modela stabla odlučivanja preuzeti s službene internetske stranice TreePlan softvera te se iz njega lako može iščitati da TreePlan metoda generiranja stabla uključuje sve prethodno obuhvaćene elemente modela stabla odlučivanja, odnosno: čvorove odluke, grane alternativnih aktivnosti, troškovi/profiti akcija, čvorove posljedica, grane posljedičnih stanja, završne čvorove te vrijednosti konačne isplate financijske koristi.



Slika 8: Stablo odlučivanja izrađeno pomoću TreePlan alata [17]

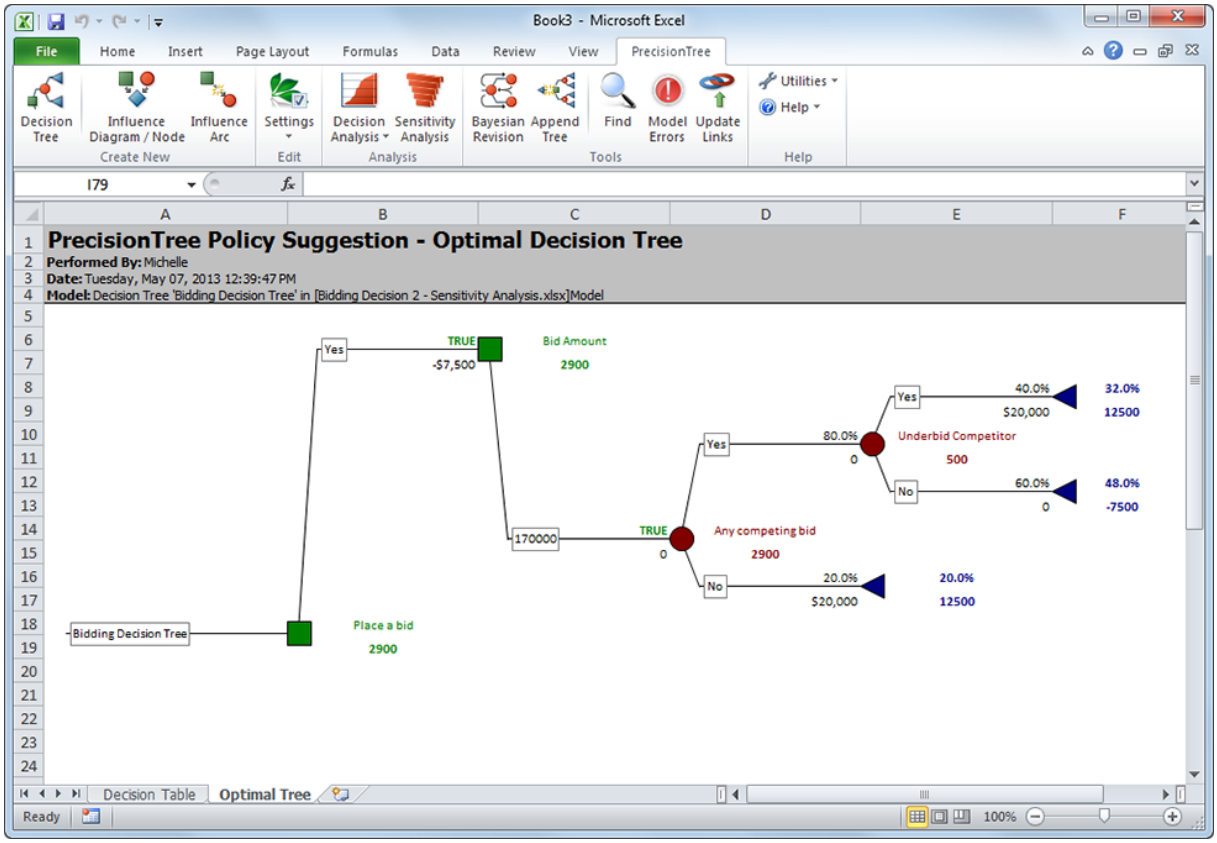
5.4.2. PrecisionTree

PrecisionTree pomaže korisnicima adresirati probleme kompleksnih sekvencijalnih odluka uz pomoć vizualnoga generiranja, organizacije i analize modela stabla odlučivanja unutar Microsoft Excela [18]. Uz pomoć PrecisionTree programskog alata moguće je vidjeti plaćanja i vjerojatnosti svake pojedine grane alternativne odluke u stablu odlučivanja. Funkcije alata PrecisionTree moguće je dodati u bilo koju ćeliju proračunske tablice što omogućuje veliku fleksibilnost u kreiranju modela odlučivanja. PrecisionTree određuje najbolju odluku u svakom čvoru odluke i označava grane te odluke s vrijednošću *"TRUE"*. Kada je model stabla odlučivanja kreiran, PrecisionTreejev algoritam analize odluke kreira statistički izvještaj najbolje odluke te uspoređuje tu odluku s drugim alternativama [18]. Osim toga, neke od prednosti PrecisionTree programskog alata su:

- radi s modelima Microsoft Excela,
- vizualni prikaz odluka,
- mogući pregled svih mogućih opcija i vjerojatnosti događaja,
- omogućuje odabir najbolje odluke sa sigurnošću,
- omogućuje predstavljanje kreiranih modela drugima.

Osim toga, PrecisionTree alat pomaže donositi odluke za sve vrste problemskih područja, uključujući [18]:

1. menadžment resursa,
2. selekciju portfelja,
3. analizu realnih opcija,
4. istraživanje i produkciju.



Slika 9: Stablo odlučivanja kreirano pomoću PrecisionTree alata [18]

Slika 9 prikazuje korištenje PrecisionTree alata unutar Microsoft Excel programskog alata te prikazuje generirani jednostavni model stabla odlučivanja s svim prethodno pobrojanim elementima.

5.4.3. SilverDecisions

SilverDecisions je za razliku od prethodna dva komercijalna alata društveno pokrenut *open source* projekt koji korisnicima na jednostavan način omogućava kreiranje kompleksnih modela stabla odlučivanja [19]. SilverPrecisions programski alat korišten je u ovom radu za potrebe pokaznih primjera kreiranja modela stabla odlučivanja zbog njegove *open source* prirode i jednostavnosti korištenja te su svi prethodni primjeri kreirani uz pomoć ovog alata. Koraci za izgradnju modela stabla odlučivanja uz pomoć SilverDecisions alata su sljedeći [19]:

1. pokretanje aplikacije s praznim platnom,
2. pritiskanje desnog gumba na mišu na bilo kojem dijelu platna, nakon čega se otvara kontekstni meni. Selektiranje *Add Decision Node* opcije. Kontekstni meni se također može otvoriti dvoklikom na bilo koji dio platna,
3. nakon dodavanja čvora odluke pojavljuje se oznaka "!!" na gornjoj desnoj strani stabla odluke što u prijevodu znači da stablo nije valjano (ne smije imati čvor odluke bez čvorova djece),
4. selekcija čvora odluke vrši se lijevim klikom, nakon čega se otvara *Decision Node* dijalog u koji upisujemo ime odluke,
5. uz pomoć *Details* opcije na lijevom panelu korisničkog sučelja moguće je dodati ime i objašnjenje samog modela stabla odlučivanja,
6. u slijedećem koraku moguće je dodavanje čvorova djece nakon čvora odluke. Čvorovi djeca mogu se dodati: lijevim klikom na čvor odluke i pritiskanjem kombinacije gumbova ctrl,alt,c dvaput ili desnim klikom na čvor odluke i selekcijom opcije *Add Chance Node* iz konteksnog menija,
7. nakon što korisnik nauči kako manipulirati s čvorovima unutar stabla slijedi manipulacija granama. Uz pomoć izbornika na lijevoj strani sučelja moguće je svakoj grani dodati naziv i plaćanje,
8. nakon što se kreiraju svi željeni čvorovi odluke i čvorovi mogućih posljedica slijedi dodavanje završnih čvorova i postavljanje njihovih vjerojatnosti. Zadana vrijednost za vjerojatnost svakog završnog čvora je "#" te ju je pomoću sučelja potrebno ručno ažurirati. Zbroj vjerojatnosti na završnim čvorovima mora biti jednaka 1 ili stablo odlučivanja nije valjano. Ukoliko ručno ne zadamo vjerojatnosti pojave pojedinog događaja, softver ih automatski računa, odnosno svakom završnom događaju daje istu vjerojatnost pojave.

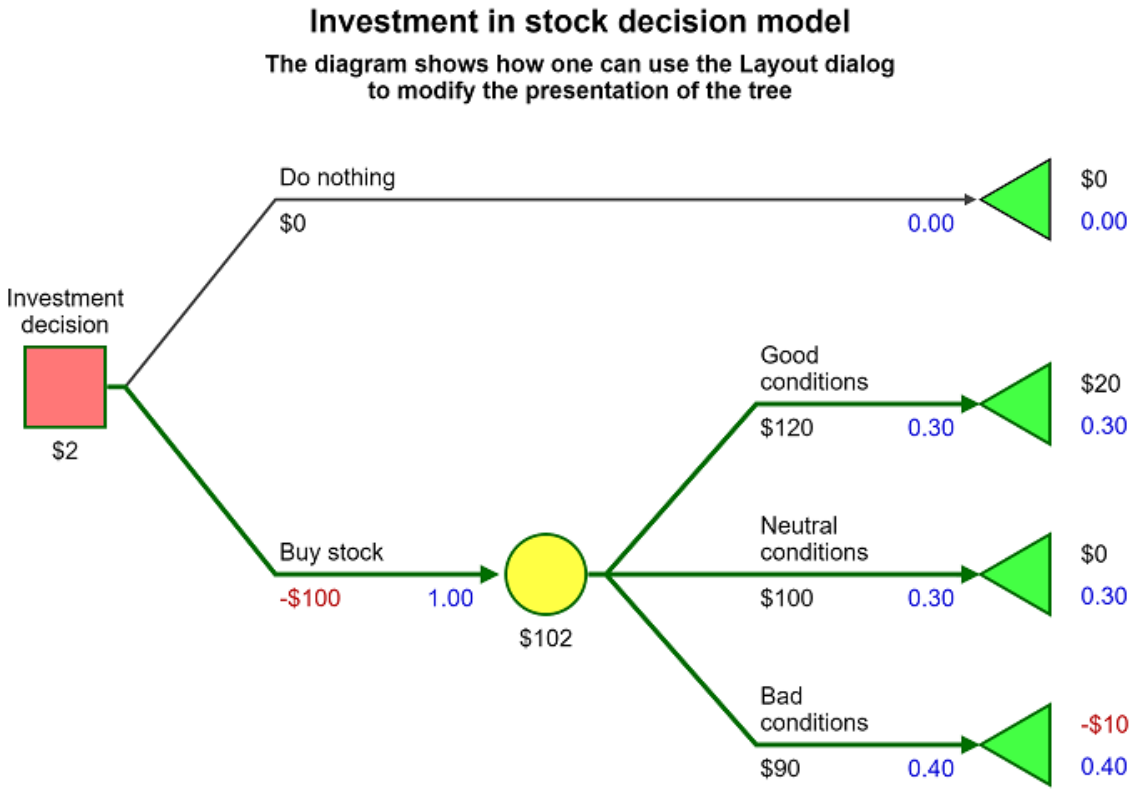
Slika 10 prikazuje jednostavni model stabla odlučivanja kreiran uz pomoć SilverDecisions alata preuzeti s njihovog službenog github repozitorija.

5.5. Primjer primjene metode stabla odlučivanja

U sklopu ovog poglavlja prikazati će se praktični primjer kreiranja modela stabla odlučivanja uz pomoć SilverDecisions alata za potrebe rješavanja fiktivnog problema donošenja odluke o nabavi određene količine proizvoda X.

5.5.1. Definiranje problema odlučivanja:

Prmjer: *Prodajno poduzeće XYZ uvodi novi proizvod "proizvod X" na tržište te nije sigurno o inicijalnoj reakciji kupaca na njihov proizvod. Menadžment tvrtke donosi odluku*



Slika 10: Stablo odlučivanja kreirano pomoću SilverDecisions alata [20]

o izlasku na tržište s različitim količinama proizvoda ili pak odustajanju od plasiranja novog proizvoda. Cijene za različite scenarije su sljedeće:

- *cijena nabave velike količine proizvoda, odnosno 100.000 jedinica proizvoda iznosi 500000 kn,*
- *cijena nabave srednje količine proizvoda, odnosno 60.000 jedinica proizvoda iznosi 320000 kn,*
- *cijena nabave male količine proizvoda, odnosno 20.000 proizvoda iznosi 150000 kn.*

Ovnisno o situaciji na tržištu, mogući su sljedeći scenariji:

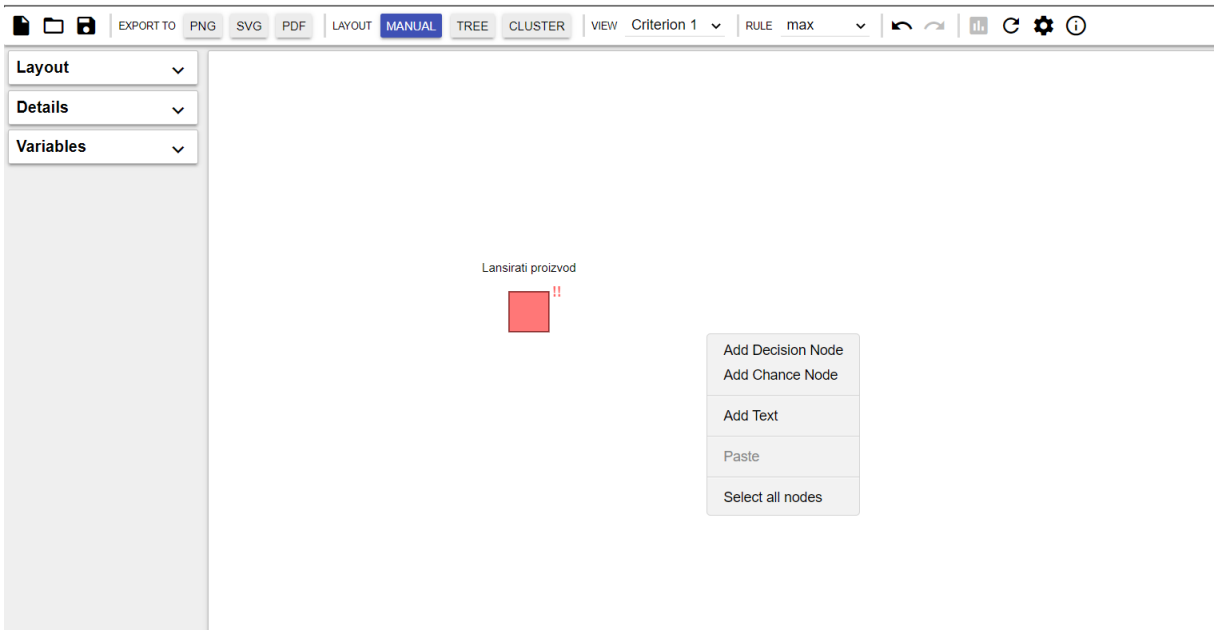
- *velika potražnja za proizvodom - 30 kn dobiti po nabavljenoj jedinici proizvoda,*
- *srednja potražnja za proizvodom - 15 kn dobiti po nabavljenoj jedinici proizvoda,*
- *mala potražnja za proizvodom - -10 kn gubitka po jedinici proizvoda*

Prihod u slučaju odustajanja od plasiranja proizvoda iznosi 0 kn, ali se gubi inicijalna investicija u razvoj proizvoda od 75000kn. Prema analizi tržišta provedenoj od strane odijela za analize poduzeća XYZ vjerojatnost za veliku potražnju proizvoda X iznosi 0.25, za srednju potražnju 0.35, dok vjerojatnost za malu potražnju proizvoda iznosi 0.4. Menadžment poduzeća prema

danoj troškovnoj analizi treba donijeti odluku o tome da li će izaći na tržište s novim proizvodom, ili pak odustati od lansiranja proizvoda i pretrpjeti novčanu štetu od 75000 kn.

5.5.2. Izgradnja logičkog modela stabla odlučivanja

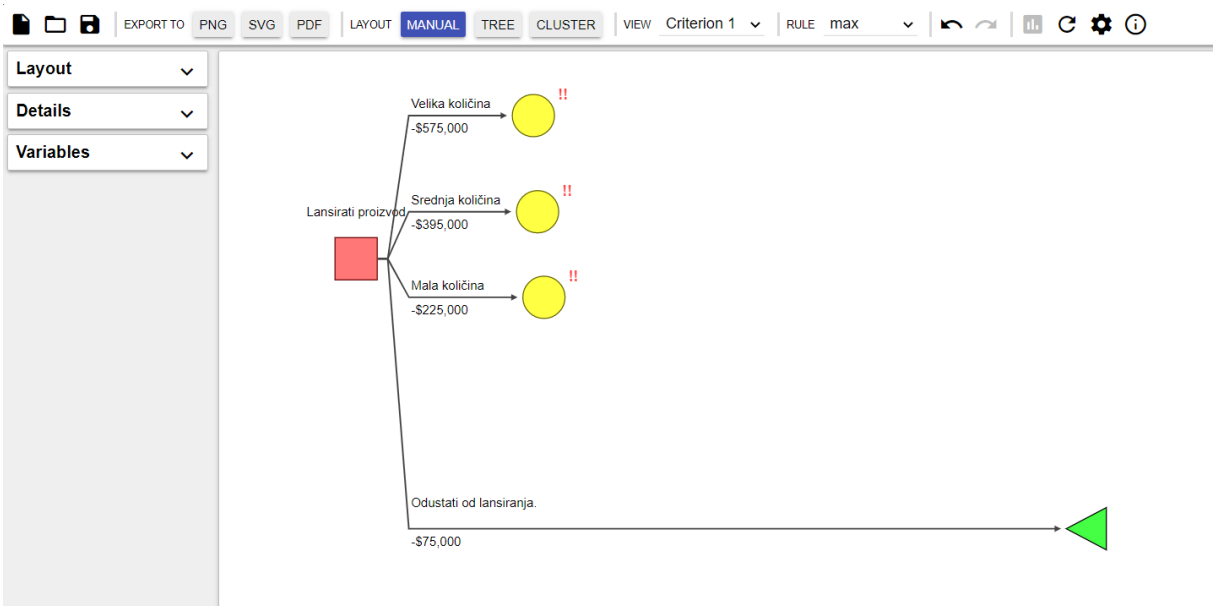
Prvi korak u kreiranju logičkog modela stabla odlučivanja je pokretanje SilverDecisions programskog alata u web pregledniku. Nakon otvaranja alata prikazuje se početno sučelje s praznim platnom na kojem je moguće dodavanje elemenata stabla. Slika 11 prikazuje početno sučelje alata kao i kreirani početni čvor odluke koji se kreira uz pomoć desnog klika na platnu i *Add Decision Node* opcije nakon čega mu je moguće dodati i ime (u ovom slučaju čvor odluke zove se lansirati proizvod). Drugi korak u izgradnji logičkog modela jest dodavanje čvorova



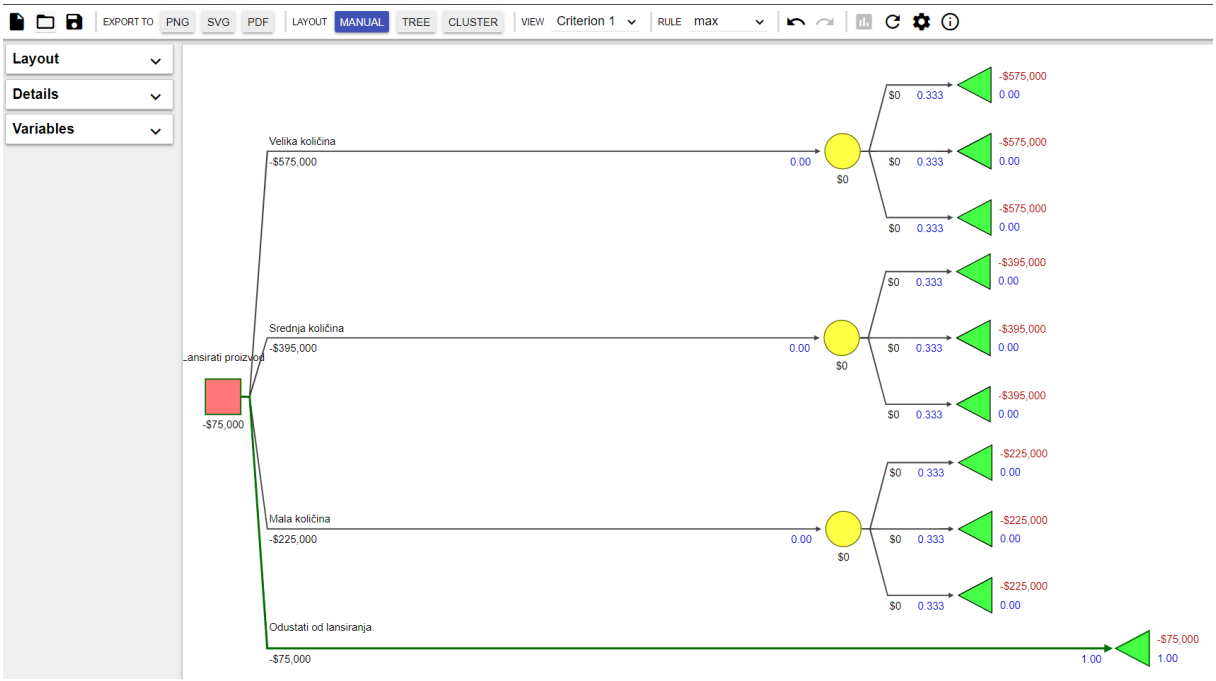
Slika 11: Početno sučelje SilverDecisions alata i definiranje inicijalnog čovra odluke. (Izvor: vlastita izrada)

posljedica za svaku pojedinu moguću odluku, kao i terminalni čvor za slučaj odustajanja od lansiranja samog proizvoda. Čvorovi posljedice dodaju se na način da se pritisne desni klik na čvor odluke i odabere opcija *Add Chance Node* (ili u slučaju dodavanja krajnjeg čvora odabiremo opciju *Add Terminal Node*). Nadalje, svakoj grani alternativnih aktivnosti pridružuje se odgovarajući naziv i trošak, kao što je prikazano na slici 12. Tako na primjer prva grana alternativnih aktivnosti, odnosno ona koja se odnosi na donošenje odluke o narudžbi velike količine proizvoda ima pridružen trošak od 575000 kn (500000 kn za nabavu proizvoda i 750000 kn utrošenih u razvoj proizvoda).

Završni čvorovi za svaki čvor posljedičnog stanja dodaju se na ranije spomenut način. Odnosno, desnim klikom na čvor posljedica i odabirom opcije *Add Terminal Node* dodaje se završni čvor selektiranom čvoru posljedice. Kako za svaku granu alternativnih aktivnosti imamo po tri različita posljedična stanja (velika, srednja i mala potražnja) dodajemo tri završna čvora za svaki posljedični čvor kao što je prikazano na slici 13.

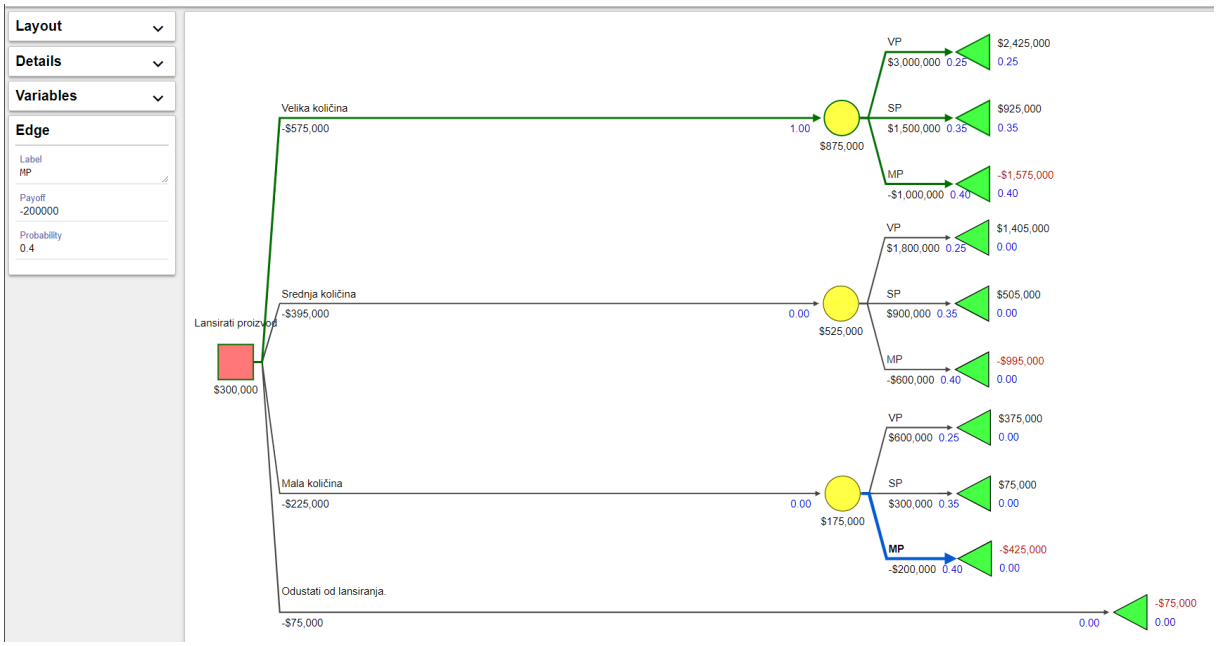


Slika 12: Dodavanje novih čvorova odluke i krajnjeg čvora (Izvor: vlastita izrada)



Slika 13: Dodavanje završnih čvorova za svako od posljedičnih stanja posljedičnih čvorova (Izvor: vlastita izrada)

Sljedeći korak u izgradnji logičkog modela uz pomoć SilverDecisions alata jest određivanje vjerojatnosti događanja svake pojedine grane posljedičnih stanja, određivanje profita svake pojedine grane, kao i imenovanje svake grane. Kako bi grani dodijelili prethodno spomenute vrijednosti potrebno je lijevim klikom označiti samu granu i u *Edge* dijelu glavnog izbornika ručno unijeti vrijednosti kao što je prikazano na slici 14.



Slika 14: Određivanje vrijednosti za vjerojatnosti, plaćanja i nazive posljedičnih grana (Izvor: vlastita izrada)

5.5.3. Računanje očekivanih vrijednosti odluka uz pomoć postupka računanja unatrag

Iako SilverDecisions programski alat računanje očekivanje vrijednosti i postupak algoritma odrađuje samostalno, za pokazne potrebe primjera prikazati će se ručni postupak računanja očekivane vrijednosti uz pomoć postupka računanja unatrag. Dakle, računamo očekivanu vrijednost čvorova posljedica uz pomoć formule za računanje očekivane vrijednosti:

$$EV_{i-1} = \sum_j p_j EV_i, i \in \{1, 2, ..., n\}, j \in \{1, 2, ..., m\} \tag{5.2}$$

Odnosno:

$EV_{vk} = 3000000 \cdot 0.25 + 1500000 \cdot 0.35 + (-1000000 \cdot 0.40) = 875000$

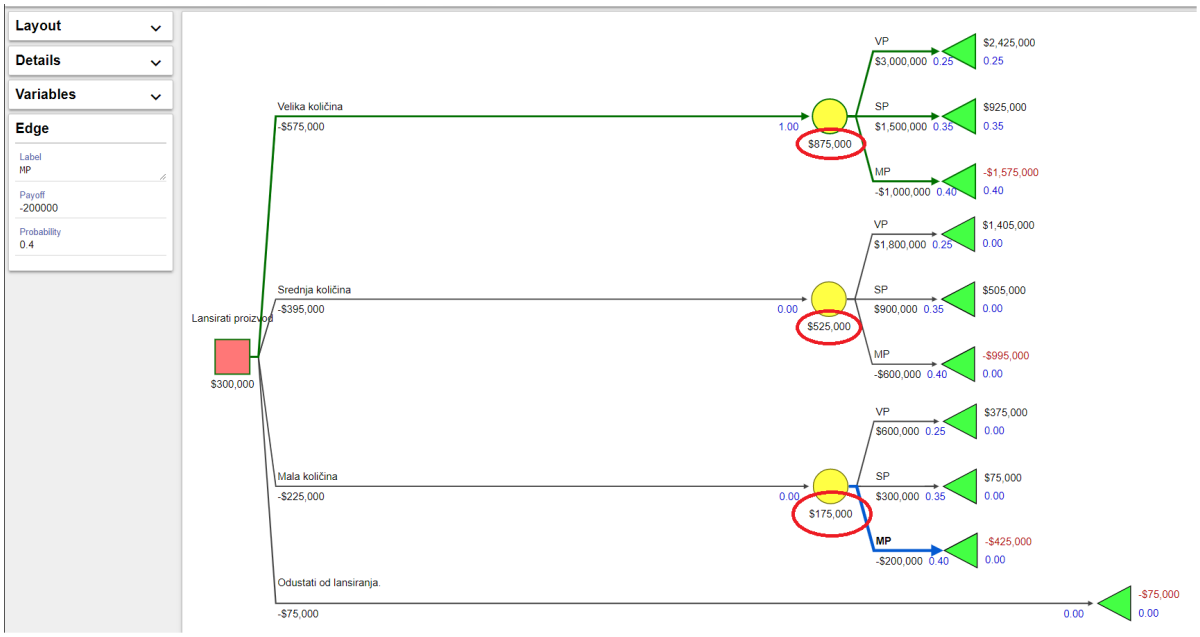
$EV_{sp} = 1800000 \cdot 0.25 + 900000 \cdot 0.35 + (-600000 \cdot 0.40) = 525000$

$EV_{mp} = 600000 \cdot 0.25 + 300000 \cdot 0.35 + (-200000 \cdot 0.40) = 175000$

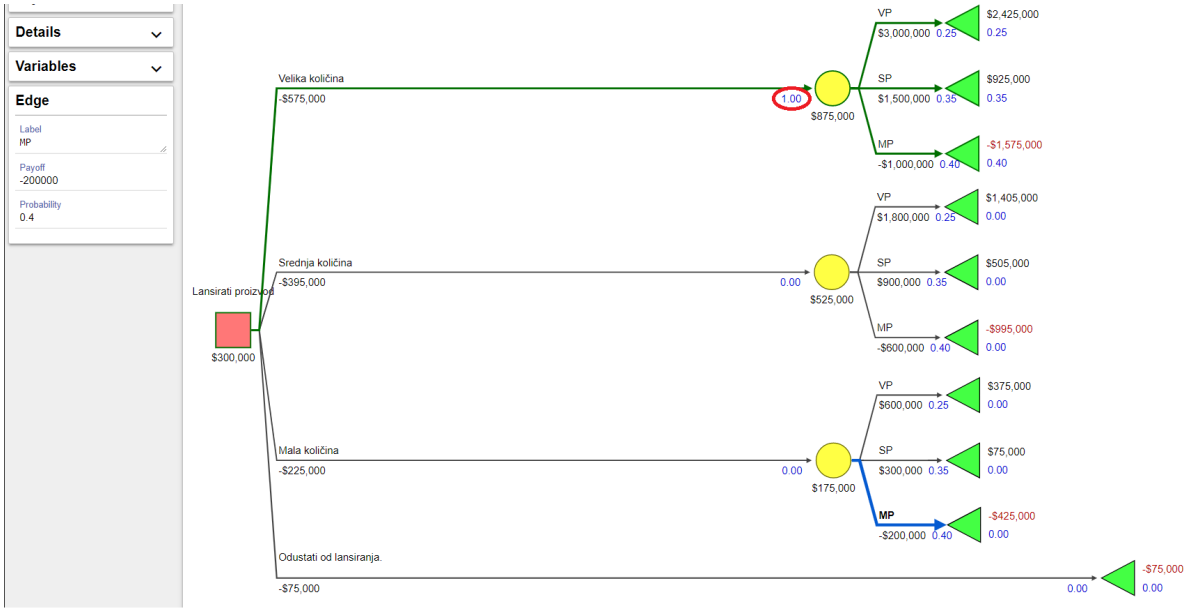
Nadalje, u alatu SilverDecisions su vidljive očekivane vrijednosti ispod čvorova posljedica te ih alat automatski računa pomoću gore navedene formule za računanje očekivane vrijednosti kao što je prikazano na slici 15.

5.5.4. Pronalaženje optimalnog puta postupkom računanja unaprijed

Nakon što se izračunaju očekivane vrijednosti za svaki pojedini čvor posljedica, računa se optimalna vrijednost početnog čvora. Optimalna vrijednost početnog čvora računa se



Slika 15: Računanje očekivane vrijednosti uz pomoć SilverDecisions alata (Izvor: vlastita izrada)



Slika 16: Pronalaženje optimalnog puta uz pomoć SilverDecisions alata (Izvor: vlastita izrada)

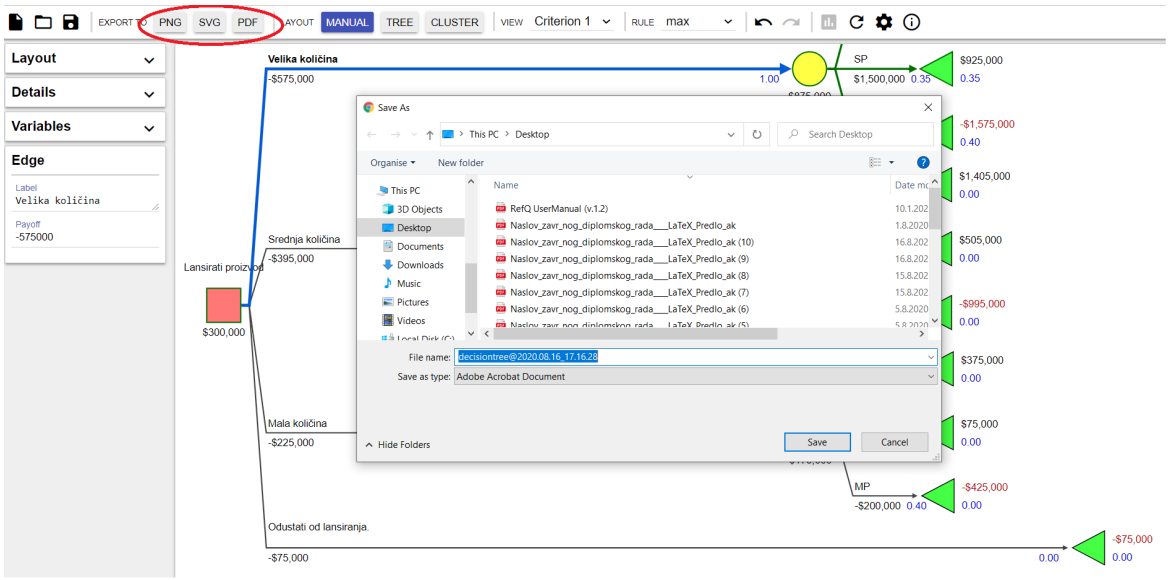
postupkom računanja unaprijed, odnosno dodjeljuje mu se očekivana vrijednost grane na optimalnom putu. Ako se pretpostavi da je cilj problema odlučivanja o lansiranju proizvoda bio maksimizacija novčane dobiti, lako se može zaključiti koja grana aktivnosti predstavlja optimalan put. Nadalje, za zadani problem odlučivanja optimalna odluka jest naručiti veliku količinu proizvoda jer ona daje najveću novčanu dobit, a dok se uračunaju troškovi ona iznosi 405000 kn. SilverDecisions programski alat i ovaj korak računa automatski, odnosno označava optimalnu posljedičnu granu zelenom bojom i dodjeljuje joj vrijednost 1.00, dok svim ostalim granama alternativnih aktivnosti pridružuje vrijednost 0.00 kao što je prikazano na slici 16.

5.5.5. Pohranjivanje modela stabla odlučivanja uz pomoć SilverDecisions alata

SilverDecisions programski alata nudi fleksibilnost u vidu pohrane samih modela stabla odlučivanja te je iste moguće pohraniti za buduće dorade ili pak prezentaciju projektnom timu. Pohrana modela stabla odlučivanja moguća je u tri formata:

- .png,
- .svg,
- .pdf,
- .json.

Kako bi se postojeći model stabla odlučivanja pohranio potrebno je kliknuti na jednu od tri ponuđene opcije za pohranu dijagrama kao što je prikazano na slici 17 te nakon toga izabrati željeni direktorij i naziv datoteke te kliknuti na gumb *ok*.



Slika 17: Pohrana modela stabla odlučivanja uz pomoć SilverDecisions alata (Izvor: vlastita izrada)

6. Uvod u strojno učenje

Strojno učenje (engl. *Machine learning*) važan je aspekt modernog poslovanja i istraživanja. Strojno učenje pomaže računalnim sistemima da progresivno poboljšavaju svoje performanse korištenjem različitih algoritama i modela. Algoritmi strojnog učenja automatski grade matematičke modele koji se kasnije koriste za donošenje odluka bez da su eksplicitno programirani za tu namjenu [2].

Postoje više definicija strojnog učenja, a prema Bishopu [21] **strojno učenje** znanstveno je proučavanje algoritama i statističkih modela koje računalni sustavi koriste za obavljanje određenog zadatka bez korištenja izričite upute, oslanjajući se na obrasce i zaključke. Strojno učenje je podskup **umjetne inteligencije**. Algoritmi strojnog učenja grade matematički model koji je baziran na uzorku podataka poznatom kao "podaci za obuku", kako bi se donijela predviđanja ili odluke bez izričitog programiranja za izvršavanje nekog zadatka.

Objema definicijama zajedničko je da strojno učenje definiraju kao proces koji računalnim sustavima omogućuje obavljanje određenog zadatka bez da su eksplicitno programirani za tu namjenu.

Također, Murphy [3] navodi da živimo u eri velikih podataka (engl. *Big data*). Ova vrsta podataka zahtijeva automatizirane metode analize podataka, što upravo strojno učenje pruža. Konkretno **strojno učenje** definira kao skup metoda koje automatski otkrivaju obrasce u podacima, a zatim koriste nepokrivene obrasce za predviđanje budućih podataka, ili izvršavaju druge vrste odlučivanja u uvjetima nesigurnosti.

Prema Glasbergenu [22] četrdesetih godina prošlog stoljeća izumljen je prvi računalni sustav s ručnim upravljanjem, ENIAC. U to se vrijeme riječ "računalo" koristila kao ime za čovjeka s intenzivnim numeričkim mogućnostima računanja, pa je ENIAC nazvan numeričkim računarskim strojem. Od početka je ideja bila izgraditi stroj koji će moći oponašati ljudsko razmišljanje i učenje. Pedesetih godina prošlog stoljeća vidimo prvi program računalnih igara koji je tvrdio da može pobijediti svjetskog prvaka dame. Ovaj je program uvelike pomogao igračima dame u poboljšanju njihovih vještina. Otprilike u isto vrijeme, Frank Rosenblatt izumio je Perceptron koji je bio vrlo jednostavan klasifikator, ali kada se umrežilo više jedinica istog, postali su snažno čudovište, što je s obzirom na to vrijeme bio pravi proboj. Zahvaljujući statistici strojno učenje je postalo vrlo poznato u 1990-ima. Presjek informatike i statistike urodio je probabilističkim pristupima umjetnoj inteligencija (engl. *Artificial intelligence*). To je pomaknulo polje dalje prema pristupima upravljanim podacima, imajući raspoložive velike količine podataka, znanstvenici su počeli graditi inteligentne sustave koji su bili u stanju analizirati i učiti iz velikih količina podataka. Kao vrhunac, IBM-ov Deep Blue sustav pobijedio je svjetskog prvaka u šahu, velikog majstora Garryja Kasparova.

Postoje mnogo primjena strojnog učenja, uključujući [23]:

- poljoprivreda
- bankarstvo
- osiguranje
- anatomija
- ekonomija
- marketing

- medicina
- telekomunikacije
- optimizacija
- adaptivne web stranice
- bioinformatika
- računalne mreže
- detekcija prijevara putem bankovnih kartica
- kvaliteta podataka
- klasifikacija DNA sekvenci
- analiza financijskih tržišta
- prepoznavanje rukopisa
- strojno prevođenje
- obrada prirodnog jezika
- internet oglašavanje
- sustavi preporuka
- programsko inženjerstvo
- prepoznavanje govora
- dokazivanje teorema

6.1. Pregled strojnog učenja

Strojno učenje se obično dijeli u dva glavna tipa. Cilj **prediktivnog** ili **učenja pod nadzorom** (engl. *Supervised learning*) je naučiti preslikavanje (mapiranje) od ulaza x do izlaza y. Ako uzmemo u obzir da oznaka $\mathcal{D}$ predstavlja skup ulazno-izlaznih parova predstavljena formulom:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$$

U najjednostavnijem okruženju svaki ulaz (podatak za obuku) x_i je $\mathcal{D}$-dimenzionalni vektor brojeva koji predstavlja, recimo, visinu i težinu neke osobe. Takve ulaze, nazivamo **značajke**, **atributi** ili **kovarijable**. Međutim, općenito, x_i bi mogao biti složen strukturirani objekt, poput slike, rečenice, e-poruke, vremenskog niza, molekularnog oblika, grafa itd. Slično, oblik izlaza ili **varijabla odgovora** (engl. *Response variable*) može biti bilo što, ali većina metoda pretpostavlja da je y_i **kategorijska** ili **nazivna** varijabla iz nekog konačnog skupa, $y_i \in \{1, ..., (C)\}$ (kao npr. spol - muško ili žensko), ili da y_i predstavlja skalar (npr. razina primanja). Kad je y_i kategoričan, problem je poznat kao **klasifikacija** ili **prepoznavanje uzoraka**, a dok je y_i skalar, problem je poznat pod pojmom **regresija**. Postoji i treća varijanta, poznata pod pojmom **poredana regresija** (engl. *Ordinal regression*) koja se događa samo kada prostor oznaka $\mathcal{Y}$ ima neki prirodni redoslijed, kao što su ocjene A-F [3].

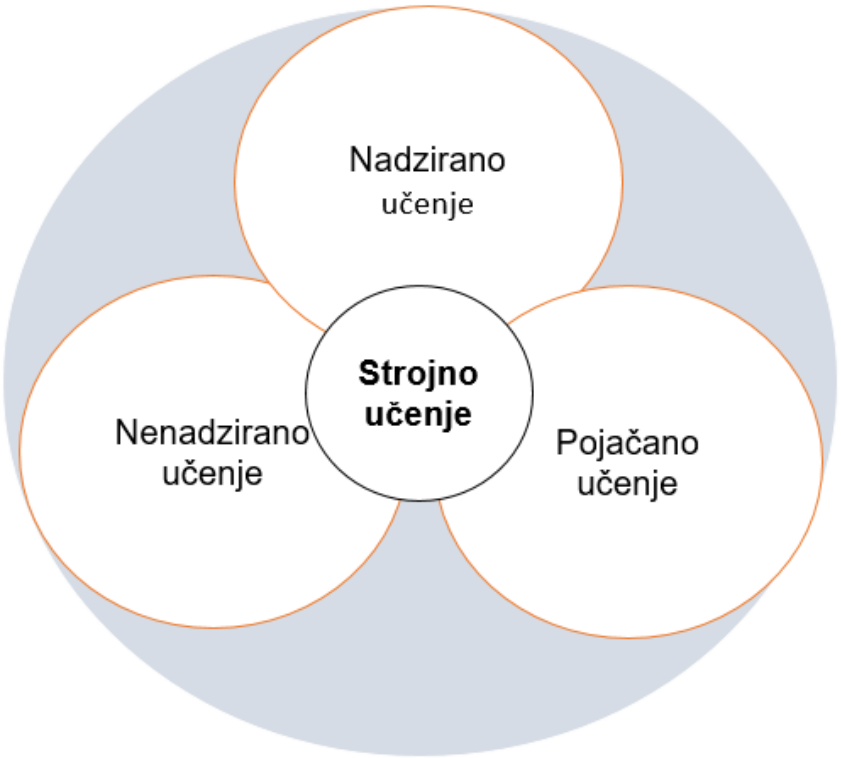
Drugi glavni tip strojnog učenja je **deskriptivno** ili **nenadzirano učenje** (engl. *unsupervised learning*). Kod nenadziranog učenja postoje samo inputi i može se prikazati formulom:

$$\mathcal{D} = \{(x_i)\}_{i=1}^{N}$$

Cilj nenadziranog učenja je pronalaženje neobičnih uzoraka u podacima. Drugi naziv za ovaj tip učenja je **otkrivanje znanja u podacima**. Ovo je puno manje definiran problem jer se ne definira koji obrasci se traže te ne postoji metrika koja na očiti način mjeri pogreške (za razliku od nadziranog učenja, gdje možemo usporediti svoje predviđanje y za dani x do promatrane vrijednosti) [3].

Nadalje, prema Alpaydinu [24] cilj **nadziranog učenja** je naučiti mapiranje od inputa do outputa čije su točne vrijednosti dane od nekog nadzornika. U **nenadziranom učenju** nema nadzornika te postoje samo ulazni podaci. Cilj takvog učenja je pronalaženje pravilnosti u ulazima, odnosno u takvim prostorima podataka postoje određeni uzorci koji se pojavljuje češće nego drugi i želi se naučiti što se u pravilu događa, a što ne.

U nekim aplikacijama izlaz sustava je redoslijed radnji. U takvom slučaju nije bitna niti jedna zasebna radnja; ono što je važno je politika koja je slijed ispravnih radnji za postizanje cilja. Ne postoji najbolja akcija u bilo kojem posrednom stanju; akcija je dobra ako je dio dobre politike. U takvom slučaju, program strojnog učenja trebao bi biti u stanju procijeniti dobrobit politika i učiti iz dobrih dobrih postupaka akcije kako bi mogao generirati politiku. Takve metode učenja nazivaju se pojačanim algoritmima učenja ili **pojačanim učenjem** (engl. *Reinforcement learning*) [24].



Slika 18: Vennov dijagram strojnog učenja (Izvor: vlastita izrada prema [25])

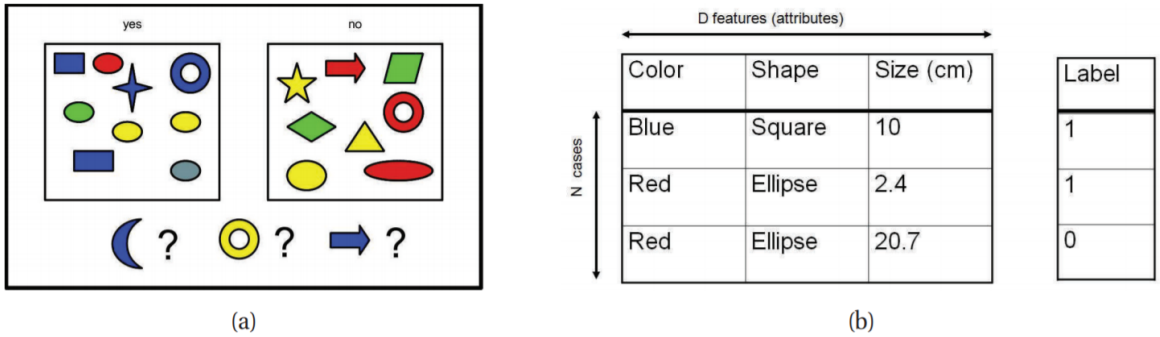
6.1.1. Nadzirano učenje

Nadzirano učenje je u praksi najrašireniji oblik strojnog učenja. Također, kao što se navodi ranije u tekstu, dva glavna problema koja se rješavaju nadgledanim učenjem su: regresija i klasifikacija.

Prema Murphyu [3] **klasifikacija** je oblik nadziranog učenja u kojem je cilj naučiti

mapiranje od ulaza x do izlaza y, gdje $y_i \in \{1, ..., (C)\}$, a (C) predstavlja ukupni broj klasa. Ako je $(C) = 2$ radi se o **binarnoj klasifikaciji** (u tom slučaju se često pretpostavlja $y \in \{0, 1\}$). Ako je pak $(C) > 2$, onda se radi o **višeklasnoj klasifikaciji**. Ako se oznake klasa međusobno ne isključuju (npr., netko može biti klasificiran kao visok i jak), onda se to naziva klasifikacija s više oznaka, ali to se smatra predviđanjem više srodnih oznaka binarne klase (tzv. višestruki izlazni model). Kada se koristi izraz „klasifikacija“, on znači klasifikaciju više klasa s jednim izlazom, osim ako nije naznačeno drugačije.

Formalizirano, problem klasifikacije predstavlja **aproksimaciju funkcije**. Ako se pretpostavi da je $y = f(x)$ za neku nepoznatu funkciju f, a cilj učenja je procijeniti funkciju f s obzirom na označeni skup podataka za obuku, predviđanje gradimo temeljem $\hat{y} = \hat{f}(x)$. (Za označavanje koristimo simbol šešira ' ˆ ' kako bi označili da se radi o procjeni). Glavni cilj **klasifikacije** je predviđanje temeljem novih podataka, što znači da predviđamo temeljem podataka koje naš model još nije vidio i taj postupak zove se **generalizacija** (predviđanje temeljem postojećih podataka je jednostavno, jer odgovor možemo samo potražiti)[3].



Slika 19: Ilustracija primjera nadziranog učenja [3, str. 3]

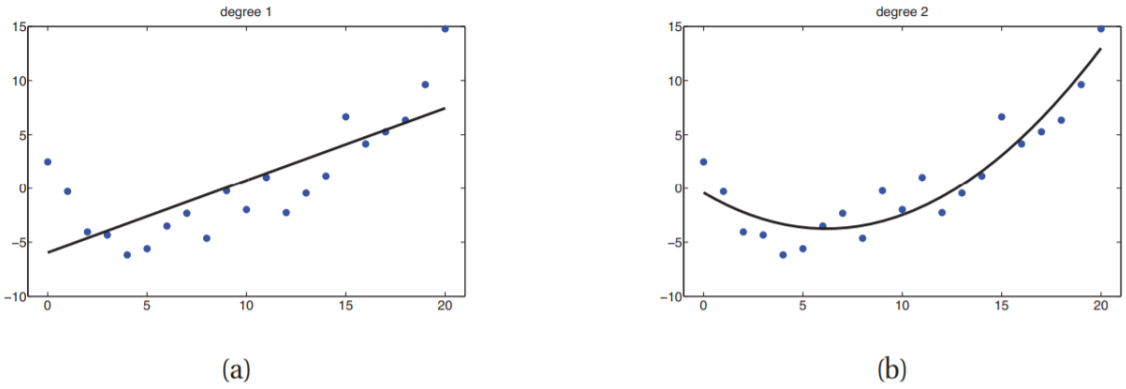
Primjer Klasifikacije:
Slika 19 sadrži dvije ilustracije; ilustracija (a) predstavlja skup označenih podataka za učenje u obliku obojanih oblika, zajedno s tri neoznačena testna slučaja. Ilustracija (b) reprezentira podatke za učenje u obliku $\mathcal{N}$ x $\mathcal{D}$ matrice. Stupac i predstavlja atribute vektora x_i. Zadnji stupac predstavlja oznaku, $y_i \in \{0, 1\}$ (*yes/no*). Dakle, prema ilustraciji (b) postoje dvije klase objekta koje su označene oznakama 0 i 1. Ulazi su predstavljeni kao obojani oblici. Svaki ulaz opisan je s setom atributa ili svojstava $\mathcal{D}$, koji su spremljeni u matricu $\mathcal{X}$ dizajna $\mathcal{N}$ x $\mathcal{D}$ kao što je naznačeno na slici 2. Ulazni atributi x mogu biti diskretni, kontinuirani ili kombinacija obje vrste. Zajendo s atributima postoji i vektor y koji sadrži oznake za obuku. Nadalje na slici imamo i tri testna ulaza: plavi mjesec, žuti krug i plava strelica. Niti jedan od ta tri slučaja nije se pojavio u ulazima za obuku te je iz tog razloga potrebna **generalizacija** izvan okvira podataka za obuku. Razumna pretpostavka bila bi da plavi mjesec treba poprimiti vrijednost $y = 1$ jer svi plavi oblici imaju oznaku 1 u podacima za obuku. Žuti krug je teži za klasificirati jer su neki žuti oblici označeni s $y = 1$ dok drugi poprimaju $y = 0$ te su neki krugovi označeni s y=1, a drugi $y = 0$. Stoga nije jasno koja je prava oznaka za slučaj žutog kruga. Slično tome, točna oznaka za plavu strelicu nije jasna.

Murphy [3] navodi da je klasifikacija najčešće korišteni oblik strojnog učenja te se koristi

za rješavanje mnogih zanimljivih i često zahtjevnih problema u stvarnom svijetu. U nastavku navedeno je nekoliko takvih primjera:

- klasifikacija dokumenata i filtriranje "spama" u e-pošti,
- klasifikacija slika i prepoznavanje rukopisa,
- detektiranje i prepoznavanje lica.

Prema Murphy [3] drugi oblik nadziranog učenja je **regresija**, koja radi na isti način kao i klasifikacija osim činjenice da varijabla odgovora (izlaz) više nije diskretna, već kontinuirana.



Slika 20: Prikaz (a) linearne i (b) polinomijalne regresije [3, str. 9]

Primjer Regresije:
Slika 20 predstavlja jednostavni primjer: postoje jedinstveni skalarni ulazi $x_i \in \mathbb{R}$ i jedinstveni skalarni izlaz $y_i \in \mathbb{R}$. Ilustracija (a) predstavlja primjer **linearne regresije** u kojem je regresija dana jedinstvenom linijom koja predstavlja najbolje uklapanje ulaznih podataka za dani matematički kriterij, tj. za dani ulaz x_i daje aproksimaciju prosječne vrijednosti izlaza y_i, odnosno utvrđuje korelaciju ulaza x_i i izlaza y_i.

Nadalje, Murphy [3] navodi primjere problema regresije u stvarnom svijetu:

- predviđanje sutrašnje cijene dionica s obzirom na trenutne tržišne uvjete,
- predviđanje starosti gledatelja određenog videa na Youtube-u,
- predviđanje temperature na bilo kojem mjestu unutar zgrade koristeći vremenske podatke i različitih senzora.

6.1.2. Evaluacija performansi nadziranog učenja

Portilla [26] navodi da se nakon izvršenja procesa strojnog učenja koriste određene mjere performansi kako bi se evaluirale performanse samih modela. Tipično u svakom zadatku klasifikacije model može postići dva rezultata: točna predikcija klase ili netočna predikcija klase. Kao primjer toga može se referirati slika 2 koja predstavlja binarnu klasifikaciju geometrijskih oblika koji mogu poprimiti oznaku 0 ili 1.

	Predviđena vrijednost	
Stvarna vrijednost	Pozitivna (P')	Negativna (N')
Pozitivna (P)	Istitnito pozitivna engl. *True Positive* (TP)	Lažno negativna engl. *False Negative* (FN)
Pozitivna (N)	Lažno pozitivna engl. *False Positive* (FP)	Istinito negativna engl. *True Negative* (TN)

Tablica 11: Matrica zabune (Izvor: vlastita izrada prema [27])

Tablica 11 prikazuje matricu zabune (engl. *confusion matrix*). Matrica zabune je alat koji se koristi kako bi se evaluirale **performanse klasifikacijskih modela**. Sama matrica sastoji se od četiri numeričke vrijednosti [28]:

- TP - istinito pozitivna vrijednost (engl. *True positive*) - predstavlja broj istinito predviđenih pozitivnih klasa,
- TN - istinito negativna vrijednost (engl. *True negative*) - predstavlja broj istinito predviđenih negativnih klasa,
- FP - lažno pozitivna vrijednost (engl. *False positive*) - predstavlja broj lažno predviđenih pozitivnih klasa,
- FN - lažno negativna vrijednost (engl. *False negative*) - predstavlja broj lažno predviđenih negativnih klasa.

Nadalje, **ključne mjere klasifikacija** su [26]:

- točnost (engl. *Accuracy*) - u problemima klasifikacije predstavlja broj točnih predviđanja modela podijeljen s ukupnim brojem predviđanja:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN},$$

- opoziv (engl. *Recall*) - sposobnost modela da pronađe sve relevantne (pozitivne) slučajeve u skupu podataka, izražena kao:

$$Recall = \frac{TP}{TP+FN},$$

- preciznost (engl. *precision*) - sposobnost modela da identificira samo relevantne slučajeve u skupu podataka, izražena kao:

$$Precision = \frac{TP}{TP+FP},$$

- F1-ocjena (engl. *F1-score*) - koristi se u slučajevima kada želimo pronaći optimalni spoj preciznosti i opoziva (harmonijska sredina), uzima u obzir preciznost i opoziv te je izražena kao:

$$F1 = 2 * \frac{precision * recall}{precision + recall}.$$

Glavna točka za pamćenje u vezi matrice zabune i predstavljenih metrika jest da one predstavljaju alate pomoću kojih se uspoređuju predikcije modela s realnim stanjem. Koja od ovih metrika je dobar pokazatelj performansi, uvelike ovisi o specifičnoj situaciji i problemu.

Nadalje, Portilla [26] navodi da je regresija specifični zadatak koji pokušava predvidjeti kontinuirane vrijednosti (za razliku od klasifikacije u kojoj predviđamo kategorije). Glavne metrike za evaluaciju **performansi regresijskih zadataka** su:

- srednja apsolutna greška (engl. *Mean absolute error*) - predstavlja srednju vrijednost apsolutnih grešaka, a izražena je kao:

$$\frac{1}{n}\sum_{i}^{n} = 1^{|y_i - \hat{y}_i|},$$

- srednja kvadratna greška (engl. *Mean squared error*) - predstavlja srednju vrijednost kvadrata grešaka (koristi se češće jer bolje adresira veće greške), a izražena je kao:

$$\frac{1}{n}\sum_{i}^{n} = 1^{(y_i - \hat{y}_i)^2},$$

- korijen srednje kvadratne greške (engl. *Root mean squared error*) - predstavlja drugi korijen srednje vrijednost kvatrata grešaka (ujedno i najpopularnija metrika):

$$\sqrt{\frac{1}{n}\sum_{i}^{n} = 1^{(y_i - \hat{y}_i)^2}}.$$

6.1.3. Nenadzirano učenje

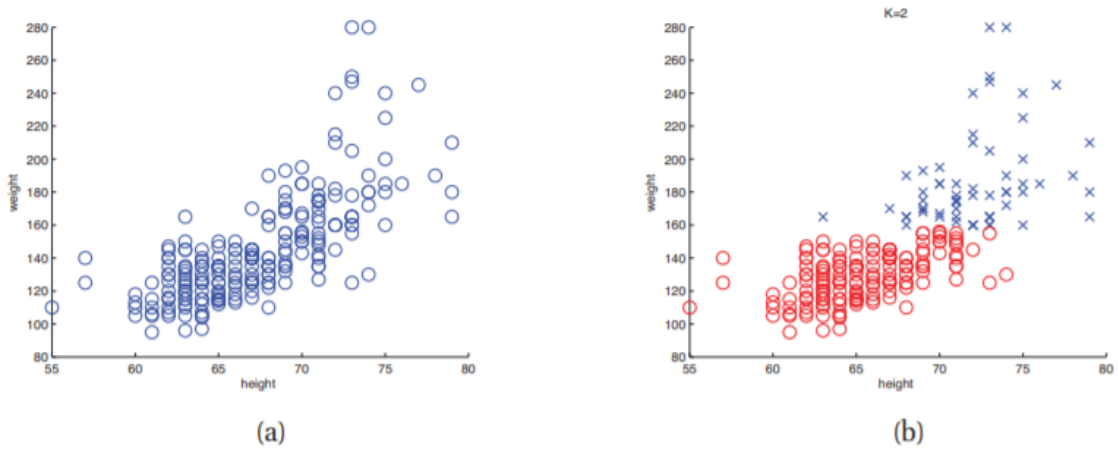
Za razliku od nenadziranog učenja nadzirano učenje je dobro istraženo i razumljivo područje. Npr., ako se traži predviđanje binarnog ishoda iz određenog skupa podataka, postoji dobro razvijen skup alata, kao i jasno razumijevanje kako procijeniti kvalitetu dobivenih rezultata. Suprotno tome, **nenadzirano učenje** (engl. *Unsupervised learning*) češće je mnogo zahtjevnije. Sam proces nenadziranog učenje je puno subjektivniji te ne postoji jednostavni cilj za analizu, kao npr. predviđanje odgovora. Nenadzirano učenje se često izvodi kao dio istraživačke analize podataka. Nadalje, može biti istraživački teško procijeniti rezultate dobivene nenadgledanim metodama učenja, budući da ne postoji opće prihvaćeni mehanizam za izvođenje unakrsne provjere ili potvrđivanja rezultata na neovisnom skupu podataka. Razlog tome je jednostavan. Ako se prediktivni model postavi koristeći tehniku nadziranog učenja, tada je moguće napraviti provjeru rezultata tako da se mjeri točnost varijable odgovora y_i za ulaz x_i koji nismo koristili u postavljanju samog modela. Međutim, kod nenadziranog učenja, ne postoji način takve provjere, jer ne postoje varijable odgovora - problem je nenadgledan [29].

Prema Murphyu [3], **nenadzirano učenje** je tip strojnog učenja u kojem su dani samo izlazni podaci, bez ikakvih ulaza. Cilj je otkriti zanimljivu strukturu u podacima; to se ponekad naziva **otkrivanje znanja u podacima**. Za razliku od nadziranog učenja, nije poznat željeni rezultat za svaki ulaz. Umjesto toga, zadatak nenadziranog učenja se formalizira u obliku **procjene gustoće** (engl. *Density estimation*), odnosno želi se izgraditi model oblika:

$$p(x_i|\theta).$$

Postoje dvije glavne razlike u odnosnu na slučaj nadziranog učenja. Prvo, umjesto $p(y_i|x_i, \theta)$, problem se izražava s $p(x_i|\theta)$; to jest, nadzirano učenje je uvjetovana procjena gustoće, dok je nenadzirano učenje neuvjetovana procjena gustoće. Drugo, x_i je vektor atributa te se koriste multivarijantni modeli vjerojatnosti, dok kod, nadziranog modela y_i predstavlja samo jednu varijablu koju pokušavamo predvidjeti. To znači da za većinu problema nadziranog učenja možemo koristiti jednovarijantne modele vjerojatnosti, što značajno pojednostavljuje problem.

Murphy [3] navodi da je **otkrivanje klastera** (engl. *Discovering clusters*) jedan od oblika nenadziranog učenja. Otkrivanje klastera rješava problem klasteriranja podataka u odvojene skupine (klastere). Na primjer, slika 21 (a) prikazuje neke 2d podatke, koji predstavljaju visine i težine grupe od 210 ljudi. Na prvi pogled se zaključuje da postoje različiti klasteri ili podgrupe, ali nije moguće odrediti točan broj istih. K predstavlja broj klastera. Prvi cilj otkrivanja klastera je procijeniti distribuciju na broj klastera, $p(K|\mathcal{D})$; ovaj podatak određuje postojanje podgrupa među podacima. Jednostavnosti radi, distribucija $p(K|\mathcal{D})$ često je predstavljena svojim modom, $K^* = arg\,max_{Kp(K|\mathcal{D})}$. U slučaju nadziranog učenja unaprijed je poznato koliko klastera (klasa) postoji, dok je u slučaju nenadziranog učenja određivanje broja klastera proizvoljno, tj. nije unaprijed definirano. Odabir modela prave (prikladne) kompleksnosti zove se **selekcija modela**. Drugi cilj metode je procijeniti kojem klasteru svaki podatak pripada. Neka je $z_i \in \{1, ..., K\}$ element kojem pripada podatak i. (z_i je primjer skrivene ili latentne varijable, jer nikad ne postoji u skupu podataka za obuku.) Može se zaključiti kojem klasteru pripada svaki podatak na način da se izračuna $z_i^* = argmax_k p(z_i = k|x_i, \mathcal{D})$, kao što je prikazano na Slici 21 (b).



Slika 21: Prikaz (a) visine i težine nekih ljudi i (b) moguće klasteriranje s K = 2 klastera [3, str. 10]

Neke primjene takvog klasteriranja primjenom metode otkrivanja klastera iz realnog svijeta su [3]:

- u astronomiji za otkrivanje novih tipova zvijezda,
- u e-poslovanju česta pojava je klasteriranje korisnika prema njihovim ponašanjima

prilikom kupnje ili surfanja weba (kako bi ih se kasnije moglo ciljati različitim sadržajima oglašavanja),

- u biologiji se klasteriranje koristi kako bi se otkrile npr. podskupine stanica.

Prema Murphyu [3], drugi česti oblik nenadziranog učenja je **otkrivanje latentnih faktora**. U situacijama u kojima se barata s podacima visoke dimenzionalnosti, često je korisno smanjiti promatranu dimenzionalnost projiciranjem podataka na pod-prostor niže dimenzionalnosti koji može uhvatiti "esenciju" podataka. Takav proces naziva se **redukcija dimenzionalnosti**. Motivacija koja leži u pozadini ove tehnike je činjenica da iako podaci djeluju visoko dimenzionalni, u pozadini najčešće stoji niska razina varijabilnosti, koja odgovara **latentnim faktorima**. Na primjer, kada modeliramo izgled slika lica, najčešće su u pozadini vrlo mali broj latentnih faktora koji opisuju većinu varijabilnosti, kao što su osvjetljenje, poza, identitet, itd., kao što je prikazano na slici 22. Kada se koriste kao ulazi za druge statističke metode, takve nisko dimenzionalne reprezentacije često daje bolju prediktivnu točnost, jer se fokusiraju na esenciju objekta, filtrirajući nebitna svojstva. Nadalje, najčešći pristup smanjenju dimenzionalnosti je **analiza osnovnih komponenti** (engl. *Principal Components Analysis, PCA*). Analiza osnovnih komponenti može se smatrati nenadziranom verzijom (više-izlazne) linearne regresije, gdje promatramo visoko-dimenzionalni odgovor y, ali ne i nisko-dimenzionalni uzrok z. iz tog razloga model poprima formu $z \rightarrow y$. Redukcija dimenzionalnosti, a posebno PCA primjenjuju se u mnogim različitim područjima, a neka područja uključuju:

- u biologiji se često koristi PCA kako bi se interpretirali podaci gena,
- u obradi prirodnog jezika se često koristi varijanta PCA pod nazivom latentna semantička analiza za vraćanje dokumenata,
- u procesiranju signala (akustičkih ili neuralnih) se često koristi ICA (varijanta PCA) kako bi se signali podijelili prema različitim izvorima,
- u računalnoj grafici se podaci o snimanju pokreta često projiciraju na prostore manje, dimenzionalnosti i koriste kako bi se kreirale animacije.



Slika 22: Prikaz (a) 25 nasumično odabranih 64x64 pikselskih slika (b) Prosječni vektor i prva tri vektora komponentnih osnova (Eigenfaces) [3, str. 12]

6.2. Metodologija strojnog učenja

Kao što se navodi u ranijem dijelu teksta nadzirano učenje predstavlja proces pomoću kojeg predviđamo diskretne ili kontinuirane vrijednosti iz nekog skupa podataka temeljem danog ulaza (u kojima postoji nadzornik, odnosno u kojima se može provjeriti točnost predviđanja samog modela raznim metrikama). Nasuprot tome, kod nenadziranog učenja ne postoji jednostavni cilj za analizu (npr. predviđanje klase), već pomoću njega pronalazimo neobične strukture i ponašanja u podacima (otkrivanje znanja u podacima). Nadalje, pošto postoji takva fundamentalna distinkcija između ova dva oblika strojnog učenja, svaki od njih pretpostavlja vlastitu metodologiju.

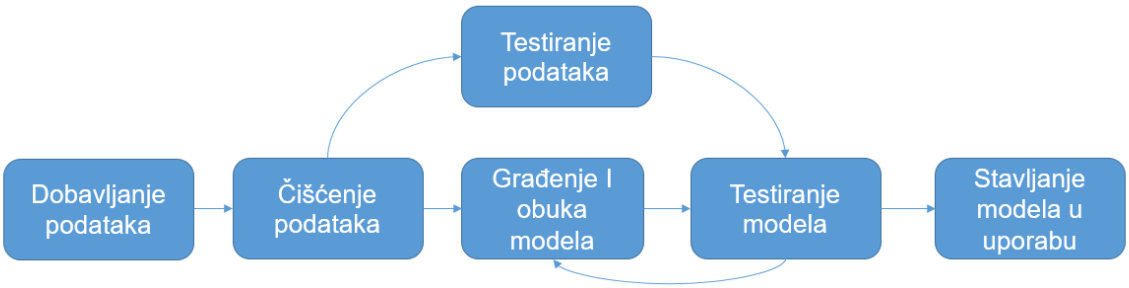
6.2.1. Metodologija nadziranog učenja

Prema Portilli [26], metodologija strojnog učenja dijeli se na sljedeće korake(metode):

- **dobavljanje podataka** (engl. *Data acquisition*) - dobavljanje podataka (od klijenata, senzora, itd.),
- **čišćenje podataka** (engl. *Data cleaning*) - čišćenje i formatiranje podataka u oblik pogodan za daljnju analizu,
- **dijeljenje podataka** (engl. *Data splitting*) - dijeljenje skupa podataka na podskup za testiranje i podskup za obuku,
- **građenje i obuka modela** (engl. *Model training and building*) - Postavljanje modela temeljem podataka za obuku i prilagođivanje (iterativni proces, uključuje građenje modela temeljem rezultata testiranja),
- **testiranje modela** (engl. *Model testing*) - iterativni proces testiranja modela pomoću

podskupa podataka za testiranje (traje dokle god nije postignuta zadovoljavajuća razina performansi modela) ,

- **stavljanje modela u uporabu** (engl. *Model deployment*) - prebacivanje modela na produkcijsku okolinu (svakodnevna uporaba u poslovanju).

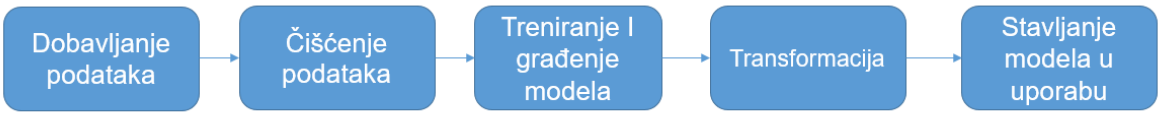


Slika 23: Metodologija nadziranog učenja (Izvor: vlastita izrada prema [26])

6.2.2. Metodologija nenadziranog učenja

Prema Portilli [26] metodologija nenadziranog učenja dijeli se na:

- **dobavljanje podataka** (engl. *Data acquisition*) - dobavljanje podataka(od klijenata, senzora, itd.),
- **čišćenje podataka** (engl. *Data cleaning*) - čišćenje i formatiranje podataka u oblik pogodan za daljnju analizu,
- **treniranje i građenje modela** (engl. *Model training and building*) - postavljanje parametara modela ovisno o specifičnostima zadatka i skupa podataka,
- **transformacija** (engl. *Transformation*) - procesiranje podataka i dobivanje zaključaka,
- **stavljanje modela u uporabu** (engl. *Model deployment*) - prebacivanje modela na produkcijsku okolinu (svakodnevna uporaba u poslovanju).



Slika 24: Metodologija nenadziranog učenja (Izvor: vlastita izrada prema [26])

6.3. Strojno učenje u bankarskom sektoru

Graham, Rajaona i Moolenaar [4] navode da je **detekcija prijevara** (engl. *Fraud detection*) najjasniji slučaj detekcije anomalija (engl. *Anomaly detection*) zasnovanih na strojnom učenju ili umjetnoj inteligenciji u bankarstvu. Nadalje, tradicionalni sustavi bazirani na pravilima postaju nedostatni te često rezultiraju u stopama lažno pozitivnih vrijednosti (engl. *False Positive, FP*) koje prelaze 90%. To stvara ogroman broj lažno pozitivnih upozorenja koja zatim moraju biti očišćena putem ljudskih intervencija. Ponavljajuće djelovanje analitičara dovodi do povećanja neosjetljivosti na lažno pozitivne rezultate, povećavajući operativni i regulatorni rizik u procesu. Nadalje, tradicionalni sustavi su po definiciji reaktivni jer mogu generirati upozorenja samo temeljem prethodnih pravila. Alternativno, sustavi temeljeni na strojnom učenju i umjetnoj inteligenciji djeluju proaktivno i poboljšavaju djelovanje regulatornog sustava upozorenja kao i tijek rada analitičara smanjujući buku bez odbacivanja upozorenja. Tablica 12 predstavlja razlike pristupa baziranog na pravilima i pristupa baziranog na strojnom učenju u vidu detekcije prijevara.

Pristup baziran pravilima	**Pristup baziran na strojnom učenju**
Hvatanje samo očiglednih scenarija prevare	Pronalaženje skrivenih scenarija i implicitnih korelacija u podacima
Potrebno puno više ručnog posla kako bi se popisali sva moguća pravila detekcije	Automatska detekcija mogućih scenarija prijevare i generiranje pravila
Višestruki koraci verifikacije koji umanjuju korisničko iskustvo	Smanjeni broj koraka verifikacije
Dugoročno procesiranje	Procesiranje u realnom vremenu

Tablica 12: Razlike pristupa detekciji prijevara (Izvor: vlastita izrada prema [30])

Nadalje, AltexSoft [30] navodi da strojno učenje omogućava kreiranje algoritama koji procesuiraju velike setove podataka s velikim brojem varijabli te pomažu pronaći skrivene korelacije između ponašanja korisnika i vjerojatnosti prijevarnih akcija. Sustavi bazirani na strojnom učenju u odnosu na sustave temeljene na pravilima omogućuju brže procesiranje podataka te zahtijevaju manje ljudskog napora i koraka verifikacije kod detekcije prijevarnih transakcija. Osim toga, sustavi temeljeni na strojnom učenju omogućuju automatizaciju procesa detekcije potencijalnih prijevarnih scenarija i procesiranje podataka u realnom vremenu.

6.3.1. Detekcija anomalija

Prema Grahamu, Rajaoni i Moolenaaru [4] **detekcija anomalija** odnosi se na pronalaženje zanimljivih obrazaca (odljevi, iznimke, osobitosti itd.) koji odstupaju od očekivanog ponašanja unutar skupa podataka. Kao aplikacijska domena unutar područja detekcija anomalija, detekcija prijevara dominira u bankarskoj industriji. **Detekcija prijevara** koristi otkrivanje anomalija radi otkrivanja ponašanja namijenjenog zavaravanju ili lažnom predstavljanju agenta. Uobičajeni primjeri uključuju prijevare s kreditnim karticama ili čekovima,

ali detekcija prijevara seže i u druge financijske sfere, uključujući osiguranje. Osim toga, detekcija anomalija zahtijeva sustav koji je agilan i koji stalno uči zbog tri razloga. To su:

- sama priroda slučajeva uporabe za otkrivanje anomalija pretpostavlja da agenti namjerno pokušavaju proizvesti ulazne podatke koji liče na validne,
- anomalije su po svojoj prirodi neočekivane, tako da je važno da metode koje se koriste za adresiranje istih budu prilagodljive podacima i njihovim promjenama,
- slučajevi korištenja u financijama izuzetno su vremenski osjetljivi; tvrtke i kupci si ne mogu priuštiti bilo kakvo čekanje u vidu brzine transakcija ili trgovine dionicama (takva čekanja mogu imati velike posljedice).

Osim toga, važno je napomenuti da, bez obzira na najčešće slučajeve uporabe (otkrivanje prijevare ili kvara sustava), anomalije nisu uvijek loše - odnosno ne moraju uvijek značiti da nešto nije u redu. Detekcija anomalija također može biti korištena, na primjer, za otkrivanje ili predviđanje malih promjena u ponašanju kupaca ili korisnika. Takve promjene mogu rezultirati promjenom prodaje, razvojem proizvoda ili novom marketinškom strategijom, omogućavajući preciznije predviđanje tržišta i držanje koraka s tržišnim trendovima [4].

Prema Grahamu, Rajaoni i Moolenaaru [4] osnovni tipovi anomalija koji se mogu identificirati:

- **anomalije točke** (engl. *Point anomaly*): anomalije točke su pojedinačni, anomalni slučajevi unutar jednog većeg skupa podataka. Na primjer, transakcija koja predstavlja jedan bilijun dolara bila bi takva anomalija, jer bi to bilo više novca nego što i najbogatiji konglomerati naprave za godinu dana. Sustavi za detekciju anomalija često počinju s identifikacijom anomalija točke, koje se dalje mogu koristiti za otkrivanje suptilnijih kontekstualnih ili kolektivnih anomalija,
- **kontekstualne (ili uvjetne) anomalije** (engl. *Conextual/ conditional anomalies*): to su podatkovne točke za koje se u određenom kontekstu smatra da su neispravne. Dobar primjer je sljedeća transakcija; dok se smatra da je 10.000 dolara unutar raspona mogućih iznosa transakcije, ukoliko je iznos izvan kreditnog limita, tada je transakcija očigledno anomalna. Kada promatramo prostorne podatke, zemljopisna širina i dužina su konteksti, dok je kod vremenskog niza podataka kontekst vrijeme,
- **kolektivne anomalije** (engl. *Collective anomalies*): kada više povezanih skupova podataka ili dijelova istog skupa podataka zajedno djeluju anomalno s obzirom na čitav skup podataka (čak i kad određeni pojedinačni skupovi podataka ne sadrže anomalije). Na primjer, recimo da postoje podaci o kreditnoj kartici koja se koristi za kupnju u SAD-u, ali i skup podataka koji prikazuje transakciju podizanja gotovine s kartice u Francuskoj u isto vrijeme. Do kolektivne anomalije može doći ako se u jednom skupu podataka naizgled ne dogodi anomalija, već se problem signalizira mjerenjem raznih komponenata više skupova podataka zajedno.

Nadalje, detekcija anomalija pristup je koji može biti koristan u različitim slučajevima uporabe. Ali temeljni, objedinjujući faktor je sposobnost otkrivanja malih promjena ili razlika u sustavu koje bi u suprotnom mogle proći neopaženo. Otkrivanje anomalija pomoću strojnog učenja omogućuje ljudima (ili drugim sustavima) da poduzmu adekvatne mjere temeljene na takvim iznimkama.

Prema Grahamu, Rajaoni i Moolenaru [4] **proces implementacije sustava detekcije anomalija** sastoji se od:

1. **odabir i razumijevanje slučaja korištenja**: prvi korak uspješne implementacije sustava detekcija anomalija je razumijevanje potreba poslovanja, odnosno koji je okvir zahtjeva i ciljeva za dani sustav. Čak i u sferi financija i bankarstva različiti projekti (sustavi) imaju različite definicije anomalija. Iz tog razloga zadaće projektnog tima su:
    - definirati i neprekidno usavršavati definiciju anomalije (moguće je da će se neprestano mijenjati te je periodično potrebna ponovna evaluacija),
    - definirati ciljeve i parametre za projekt u cjelini. Primjerice, krajnji cilj vjerojatno nije sama detekcija anomalija, već nešto veće što utječe na poslovanje, poput blokiranja lažnih transakcija,
    - utvrditi, nakon što se otkrije anomalija, što će sustav učiniti sljedeće (npr. slanje drugom timu eksperata na analizu),
    - razvoj plana za praćenje i evaluaciju uspjeha sustava,
    - identificirati frekvenciju detekcije anomalija za dani slučaj korištenja.
2. **dohvaćanje podataka**: imati što više podataka za analizu omogućit će izradu preciznijih modela jer se nikad ne zna koji atributi mogu ukazivati na anomaliju. Korištenje više vrsta i izvora podataka omogućuje kretanje banaka izvan okvira anomalija točke u prepoznavanje sofisticiranijih kontekstualnih ili kolektivnih anomalija. Raznolikost samih izvora, kao i podataka je ključna za uspjeh projekta. Primjerice, moguće je da je prevarant ostao unutar normalnog raspona stvarnih navika korisnika, što na kraju rezultira validnim transakcijskim podacima. No, podaci o upotrebi bankomata ili web dnevnika računa mogu otkriti nepravilnosti,
3. **analiza, čišćenje i transformacija podataka**: prilikom detekcije anomalija ova faza je još važnija nego obično jer podaci često sadrže buku (najčešće ljudske greške) koja je potencijalno slična stvarnim anomalijama. Stoga je važno razlikovati između njih dvoje i uklonite sve problematične podatke koji bi mogli stvoriti lažno pozitivne rezultate. Preferirani način izrade sustava za detekciju anomalija je početak s unaprijed klasificiranim setom podataka (odnosno nizom podataka za koje unaprijed znamo jesu li validni ili prijevarni). To je najčešće najjednostavniji put naprijed jer omogućuje klasifikaciju uz pomoć nadziranih metoda strojnog učenja,
4. **prediktivna analitika**: postoje dvije primarne arhitekture za izgradnju sustava otkrivanja anomalija:

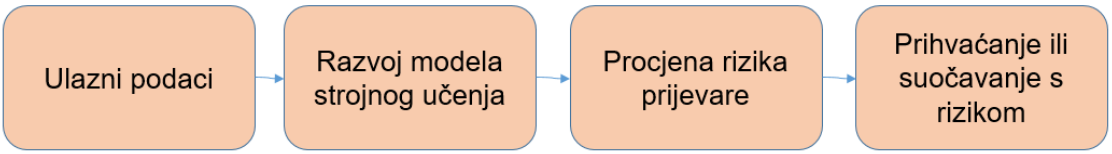
(a) Nadzirano otkrivanje anomalija koje se može koristiti ako imate označeni (klasificirani) skup podataka u kojem je poznato da li je podatak validan ili ne,

(b) Nenadzirano otkrivanje anomalija, u kojem se koristi neoznačeni skup podataka (tj. skup podataka bez informacije o validnosti podataka),

5. **vizualizacija podataka**: vizualizacije su posebno korisne u procesu izgradnje i testiranja modela otkrivanja anomalija jer ponekad su one najjasniji način da se vide iznimke, posebno u vrlo velikim skupovima podataka,

6. **stavljanje u uporabu i iteriranje**: otkrivanje prijevara i anomalija u bankama općenito je izuzetno vremenski osjetljivo. Iz tog razloga je produkcijska primjena sustava detekcije anomalija i predviđanje na "živim" podacima važnije no ikad. Nadalje, stavljanje sustava u produkciju nije kraj. Za sustave otkrivanja prijevare i bilo koje druge anomalije ključno je kontinuirano praćenje i ponavljanje kako bi model nastavio učiti i biti efikasan usprkos stalnim promjenama okoline i ponašanja.

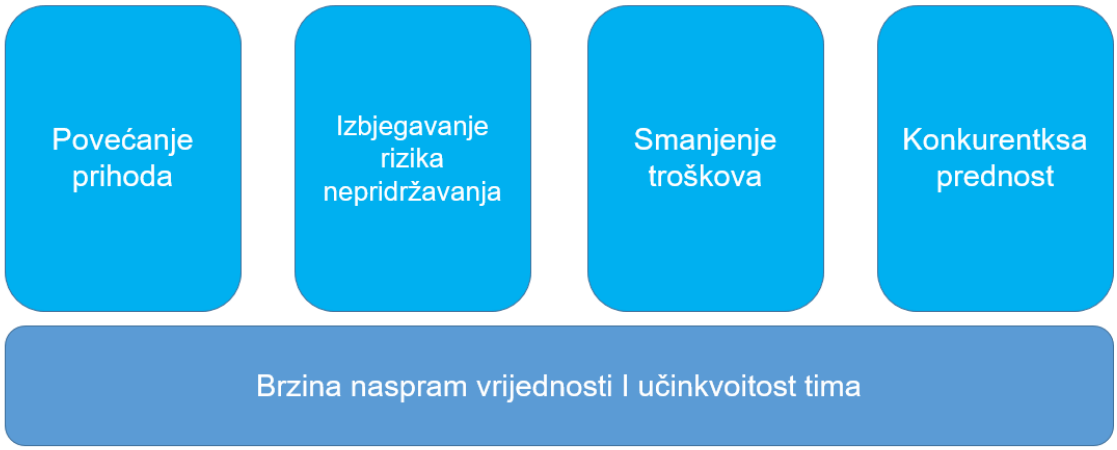
Osim toga, **detekcija prijevara** skup je aktivnosti koje se poduzimaju kako bi se spriječilo da se novac ili imovina steknu lažnim izgovorom. Otkrivanje prijevara primjenjuje se u mnogim industrijama poput npr. bankarstva i osiguranja. U bankarstvu prijevara može uključivati krivotvorenje čekova ili korištenje ukradenih kreditnih kartica [31]. Tipični **proces detekcije prijevara** predstavljen je slikom 25. Prijevara u financijskom sektoru predstavlja zamršen problem koji postoji kod svih pružatelja usluga, bez obzira na njihovu veličinu i broj korisnika. Radi se o dobro raspravljenom problemu s poznatim komplikacijama, a strojno učenje je njegovo potencijalno rješenje. Strojno učenje u financijama i bankarstvu može procijeniti ogromne skupove transakcija u realnom vremenu. Štoviše, mogućnost učenja iz rezultata i ažuriranja modela umanjuje ljudski angažman u procesu. Koristeći tehnike strojnog učenja dobavljači mogu označiti povijesne podatke kao prijevare ili ne te na taj način sustav uči kako prepoznati aktivnosti koje izgledaju sumnjivo [32].



Slika 25: Proces detekcije prijevara (Izvor: vlastita izrada prema [32])

6.3.2. Ostali slučajevi korištenja strojnog učenja u bankarskom sektoru

Osim detekcije anomalija i u užem smislu detekcije prijevara u bankarskom sektoru postoji niz drugih slučajeva primjena strojnog učenja i umjetne inteligencije te svi oni pripadaju u jednu od pet kategorije koje su prikazane na slici 26.



Slika 26: Kategorije slučaja korištenja strojnog učenja i umjetne inteligencije u bankarstvu (Izvor: vlastita izrada prema [33, str. 14]

Prema Moolenaaru, Rughaniu i Ménardu [33] glavni izazov financijskih i bankarskih organizacija nije ne pronalaženje mogućih slučaja primjena, već odabir onog slučaja primjene koji će se pokazati kao uspješna podatkovno pokrenuta inicijativa bazirana na potencijalnom pozitivnom učinku na poslovanje. Zbog toga su kategorije slučaja korištenja strojnog učenja i umjetne inteligencije u bankarstvu i financijama veoma široke. Glavne kategorije slučaja korištenja strojnog učenja i umjetne inteligencije u bankarskom i financijskom sektoru su:

1. **povećanje prihoda** (engl. *Increased revenue*)
    - dodjela prihoda (engl. *Revenue attribution*) - u želji da se približe svojim kupcima, institucije komercijalnog bankarstva pokreću koncept knjige poslovanja (*tngl. Book of business*), koji podrazumijeva da IT timovi stvaraju pravila koja govore kako svakog klijenta rasporediti u nadležnost jednog ili više odjela. U teoriji, to omogućuje rukovoditeljima i osoblju da se lakše usredotoče na pružanje usluga klijentima koji su u njihovoj nadležnosti. Međutim, skup pravila koja su stvorena za alokaciju klijenata mogu značajno utjecati na strukturu zaposlenika poslovnice te na taj način remetiti prirodni tok poslovanja. Iz tog razloga se za stvaranje jasnih, interoperabilnih modela knjiga poslovanja koristi strojno učenje,
    - osiguranje - pored otkrivanja prijevare banke koje nude usluge osiguranja također imaju koristi od sustava temeljenih na strojnom učenju kako bi poboljšali optimalnu cijenu, konverziju, trošak polaganja prava i predviđanje. Strojno učenje za optimizaciju cijena omogućuje djelatnostima osiguranja dinamičko praćenja stanja na tržištu i prilagođavanje cijena na temelju obrazaca koji se otkrivaju na tržištu. Nemogućnost točne izolacije zahtjeva koji jamče brzo namirenje može koštati milijune i značiti da neki zahtjevi budu značajno preplaćeni. Razvijanje sustava temeljnog na strojnom učenju koji je sofisticiraniji i nijansiraniji od sustava temeljnog na pravilima može u ovom pogledu znatno poboljšati performanse i smanjiti gubitke. Konačno, takvi sustavi mogu osigurateljima precizno i u mnogim slučajevima

automatski predvidjeti broj i veličinu zahtjeva te dati rukovoditeljima osiguranja točnije predviđanje gubitaka,

2. **izbjegavanje rizika nepridržavanja** (*engl. Avoiding non-compliance risk*)

   - sigurnost na internetu (engl. *Cyber security*) - prema Forbesu (kao što se citira u [33]), tipična Američka tvrtka za financijske usluge napadnuta je zapanjujuće 1 milijardu puta godišnje što je više od 30 napada u sekundi. Kako bi tvrtke ostale u toku sa brojem i stupnjem sofisticiranosti takvih napada potrebni su najsuvremeniji sustavi, a takvi sustavi temelje se na strojnom učenju i umjetnoj inteligenciji. Razvoj sustava za otkrivanje anomalija temeljen na strojnom učenju omogućuje upotrebu širokih i različitih izvora podataka koji su ključni za pronalaženje igle u plastu sijena anomalija. Razni agenti i zlonamjerni softveri namjerno pokušavaju proizvesti ulazne podatke koji izgledaju autentično. Prilagođavanje i učenje iz ove stvarnosti je kritično i to je nešto što se može postići samo sa primjenom strojnog učenja i umjetne inteligencije,
   - prognoza kreditnog rizika i gubitaka: izvješće McKinsey za 2018. godinu navodi (kao što se citira u [33]) da je povećanje snage kreditnog prediktivnog modela od jedan posto u dosegu većine banaka. Način na koji je to izvedivo je upotrebom strojnog učenja pomoću kojeg je moguće na brz i efikasan način analizirati podatke iz više izvora te tako donositi bolje kreditne odluke nego kreditni analitičari. Nadalje, poželjno je koristiti decidiranu platformu kako bi se postigla adekvatna razina transparentnosti potrebna za interoperabilnost modela kreditnih rizika i gubitaka,
   - u ovu grupu slučaja korištenja pripada i detekcija prijevara koja je pobliže objašnjena u ranijem dijelu teksta,

3. **smanjenje troškova** (engl. *Decreased costs*)

   - zamjena sustava temeljenih na pravilima - korištenje sustava temeljenih na strojnom učenju umjesto sustava temeljenih na pravilima u cijelom poduzeću, bez obzira na liniju poslovanja ima smisla s obzirom na to da sustavi temeljeni na pravilima generiraju veliki broj lažnih pozitiva. Također, sustavi strojnog učenja mogu iskoristiti i ocijeniti upozorenja iz sustava temeljenih na pravilima te na taj način pomoći analitičarima prepoznati rizična upozorenja i točno prosljeđivati informacije odgovarajućim timovima za istrage,
   - Automatizacija regulatornog izvještavanja - bankarske i financijske djelatnosti podliježu mnogim regulativama te zbog toga imaju povećane zahtjeve za izvještavanjem. Iz tog razloga veliki dio takvih poslova odrađuje se ručno u konsolidiranim proračunskim tablicama. Rezultat toga je loša učinkovitost i povećani angažman radnika. Također ručno kreiranje izvještaja povećava rizik pogrešaka i ovisnost o nekolicini ljudi koji ih kreiraju. Zajedno s procesom upravljanja podacima koji adresira dio ovog problema, koristi se strojno učenje za adekvatnije otkrivanje anomalija i napredniju automatizaciju,

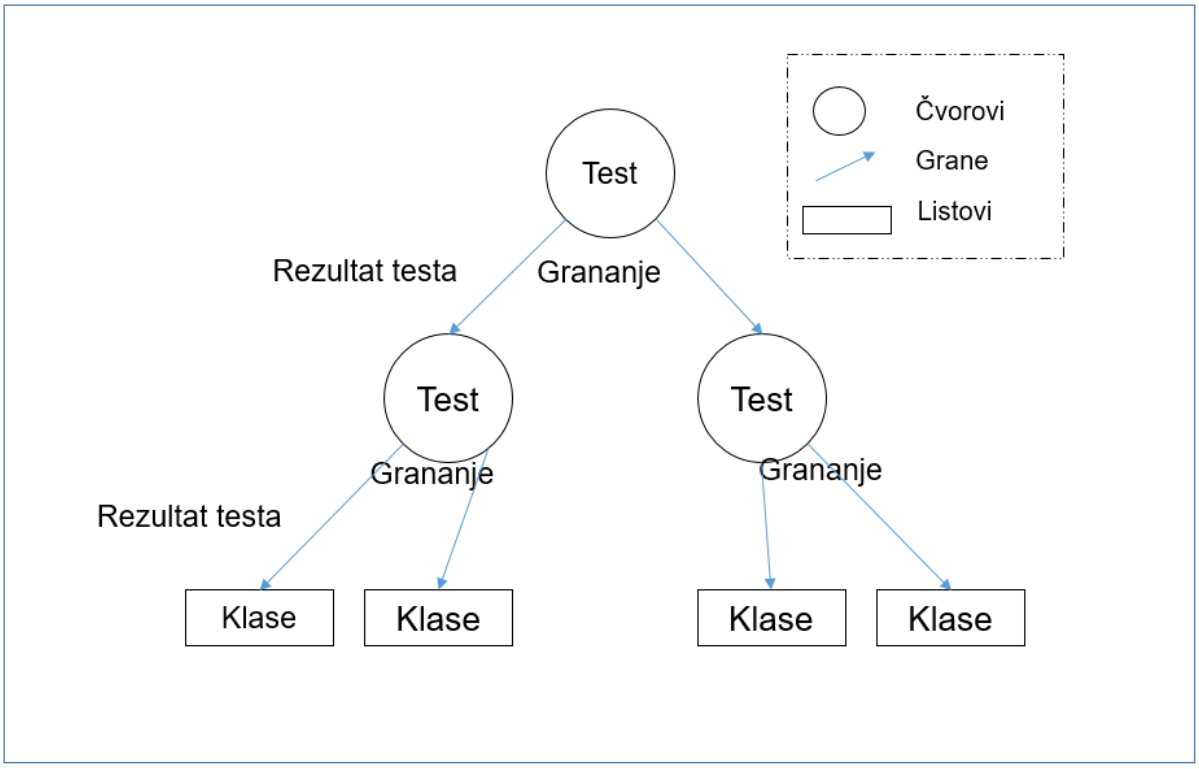
4. **konkurentska prednost** (engl. *Competetive edge*) - analiziranje novčanih tokova, konkurentska analiza i analiza ciljnih pokazatelja; prediktivno modeliranje tržišta koje se temelji na velikim količinama podataka iz velikog broja izvora utječe na veću informiranost i sposobnost ulagača da udovolje zahtjevima klijenata,

5. **brzina naspram vrijednosti i učinkovitost tima** (engl. *Speed-to-value & team efficiency*)

    - poboljšanje i automatizacija procesa - Poboljšanje ili automatizacija procesa za *ETL* (engl *Extract, Transform, Load*) i pripremu podataka donose organizacijsku vrijednost putem brzine uvida i timske učinkovitosti,
    - pojednostavljivanje upravljanja marketinškim resursima (engl. *Streamlining marketing resource management*)- korištenje specijaliziranih platformi strojnog učenja, umjetne inteligencije i znanosti podataka omogućuje uvođenje dosljednosti i konzistentnosti u proces upravljanja marketinškim resursima i prelazaka različitih modela u fazu uporabe. Upravljanje marketinškim resursima zahtijeva konzistentan proces kojim modeli prelaze iz razvoja u uporabu kako bi se ostvarila maksimalna koordinacija između timova.

7. Metodologija stabla odlučivanja

Stablo odlučivanja (engl. *Decision tree*) je prediktivni model koji se može primijeniti na klasifikacijski i regresijski problem. Stablo odlučivanja predstavlja hijerarhijski model odluka i njihovih posljedica. Donositelj odluke koristi stablo odlučivanja kako bi identificirao strategiju kojom će postići svoj cilj. Kada se stablo odlučivanja koristi za rješavanje klasifikacijskog zadatka zove se klasifikacijsko stablo, a kada se koristi za regresijski problem zove se regresijsko stablo [34]. Osim toga, prema Kozaku [35] stabla odlučivanja na jednostavan način reprezentiraju kompleksne koncepte koji su definirani skupovima svojstava. Drugim riječima, stabla odlučivanja reprezentiraju funkcije koje preslikavanju vrijednosti atributa u skup klasa odluke koje predstavljaju dopustive hipoteze. Stabla odlučivanja koriste se u operativnim istraživanjima u analizi odluka za identificiranje optimalne strategije postizanja cilja. **Stablo odlučivanja** je aciklički usmjereni graf čija se hvatišta zovu čvorovi (engl. *Nodes*), a rubovi grane (engl. *Branches*), čvorovi bez potomaka - listovi (engl. *Leaves*), a korijenski čvor je jedino hvatište bez roditelja (engl. *Root node*).



Slika 27: Konceptualni model stabla odlučivanja (Izvor: vlastita izrada prema [35, str. 4]

Nadalje, svi čvorovi sadrže testove atributa koji se generiraju temeljem kriterija grananja i oni predstavljaju načine prema kojima se određuju grananja podataka prema vrijednostima njihovih atributa. Svi testni rezultati predstavljeni su granama (primjer strukture stabla odlučivanja prikazan je na slici 27). Specijalni tip stabla odlučivanja je binarno stablo odlučivanja u kojem svaki čvor osim terminalnih čvorova ima točno dva potomka. Na taj način se opservacije uvijek dijele u točno dva podskupa. Stabla odlučivanja ovog tipa grade se npr. pomoću **CART** algoritma. Prema Rokachu i Maimonu, [34] **stablo odlučivanja** jest klasifikator

izražen kao rekurzivno particioniranje prostora predviđanja. Stablo odlučivanja sastoji se od čvorova koji formiraju ukorijenjeno stablo (engl. *Rooted tree*), odnosno usmjereno stablo koje ima jedan čvor bez ulaznih rubova koji nazivamo korijenski čvor. Svi ostali čvorovi imaju točno jedan ulazni rub. Čvor s izlaznim rubovima naziva se unutarnji ili testni čvor. Svi ostali čvorovi nazivaju se listovi ili terminalni čvorovi.

7.1. Indukcija stabla odlučivanja

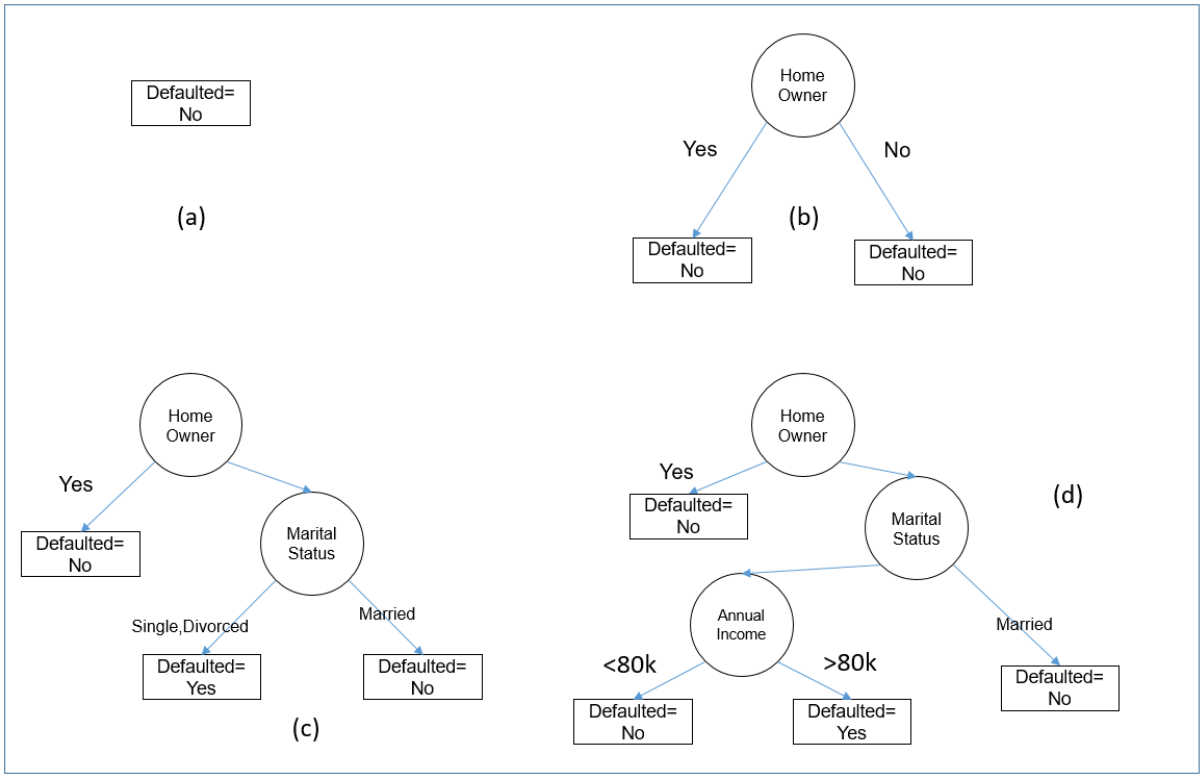
Prema Tanu, Steinbachu i Kumaru [36] postoje eksponencijalno velik broj stabla odlučivanja koja se mogu konstruirati iz danog skupa podataka. Neka od tih stabla mogu biti točnija od drugih, a pronalaženje optimalnih stabla je u računalnom smislu neisplativo zbog eksponencijalne veličine prostora traženja. Štoviše, razvijeni su efikasni algoritmi za indukciju razumno točnog, ali suboptimalnog stabla odlučivanja u razumnom vremenu. Navedeni algoritmi koriste pohlepnu strategiju koja gradi stablo odlučivanja kreiranjem serije lokalnih optimuma odluka o tome koji atribut koristiti za particioniranje podataka. Jedan takav algoritam je **Huntov algoritam**, koji je baza mnoštva algoritama za indukciju stabla odlučivanja poput:

- ID3,
- C4.5,
- CART.

U Huntovom algoritmu, stablo odlučivanja se gradi kao rekurzivno particioniranje skupa podataka za učenje u sukcesivno čišće podskupe. Ako D_t predstavlja skup podataka za učenje povezanih s čvorom t i $y = (y_1, y_2, ..., y_c)$ predstavljaju oznake klase, definicija Huntovog algoritma za problem klasifikacije je:
Korak 1: Ako svi zapisi u D_t pripadaju istoj klasi y_i, onda je t krajnji čvor s oznakom y_t
Korak 2: Ako D_t sadrži zapise koji pripadaju više klasa, vrši se selekcija putem testiranja uvjeta atributa kako bi se skup particionirao u manje podskupe. Kreira se novi čvor za svaki ishod testiranja uvjeta i zapisi iz D_t se distribuiraju u nove čvorove. Algoritam se dalje rekurzivno primjenjuje na svaki novi čvor [36].
Primjer indukcije Huntovim algoritmom: Kako bismo ilustrirali funkcioniranje algoritma, zamislimo problem predviđanja sposobnosti primaoca kredita za podmirenje obveza. Set podataka za obuku može se konstruirati s podacima o bivšim korisnicima kredita (tablica 13). Svaki zapis predstavlja osobne informacije o primaocu kredita kao i oznaku klase koja govori da li je kredit bio vraćen ili ne. Inicijalno stablo prikazano na slici 28 (a) ima samo jedan čvor s oznakom klase *"Defaulted = No"* što znači da je većina korisnika vratila dugovanje. Kako bismo poboljšali stablo, zapise je potrebno dijeliti u manje podskupe kao što je prikazano na slici 28 (b). Odnosno, vrši se particioniranje u dva podskupa temeljem testiranja uvjeta na atributu *"Home Owner"*.

Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125k	No
2	No	Married	100k	No
3	No	Single	70k	No
4	Yes	Married	120k	No
5	No	Divorced	95k	Yes
6	No	Married	60k	No
7	Yes	Divorced	220k	No
8	No	Single	85k	Yes
9	No	Married	75k	No
10	No	Single	90k	Yes

Tablica 13: Skup podataka o klasifikaciji kredita (Izvor: vlastita izrada prema [36, str. 153])



Slika 28: Primjena Huntovog algoritma na skupu podataka sa tablice 13 (Izvor: vlastita izrada prema [36, str. 154])

Nadalje, može se primijetiti kako su svi korisnici koji posjeduju dom (oni koji vrijednost atributa *"Home Owner"=Yes*) uspjeli vratiti posuđena sredstva. Iz tog razloga lijevi čvor poprima oznaku *"Defaulted = No"* (Slika 28 (b)), dok za desni čvor nastavljamo rekurzivno primjenjivati Huntov algoritam sve dok svi zapisi ne budu pripadali istoj klasi. Stabla koja rezultiraju nakon svakog rekurzivnog koraka prikazana su na slici 28 (c) i 28 (d). Huntov algoritam će raditi ukoliko u skupu podataka postoje sve kombinacije vrijednosti atributa i svaka kombinacija ima jedinstvenu oznaku klase. Nadalje, svaki algoritam učenja za indukciju stabla odlučivanja mora adresirati sljedeće probleme:

1. **kako granati podatke za učenje?** - svaki korak procesa građenja stabla mora odabrati atribut po kojem testira uvjet kako bi podatke podijelio u manje podskupe. Za implementaciju ovog koraka, algoritam mora pružati metodu za specifikaciju uvjeta testiranja za različite tipove atributa kao i objektivnu mjeru za evaluaciju kvalitete svakog uvjeta,
2. **kako zaustaviti proceduru grananja?** - potreban je uvjet zaustavljanja kako bi se završio proces građenja stabla.

7.1.1. ID3

Quinlan (kao što se citira u [34]) navodi da je ID3 vrlo jednostavan algoritam konstrukcije stabla odlučivanja. Kao kriterij grananja koristi informacijski dobitak (engl. *Information gain*) te završava kada sve opservacije pripadaju jedinstvenoj ciljnoj vrijednosti atributa ili kada je najbolji informacijski dobitak veći od nule. ID3 ne koristi proceduru podrezivanja te ne može baratati s nedostatnim vrijednostima pojedinih atributa. Glavna prednost ID3 algoritma jest njegova jednostavnost te se često koristi za potrebe učenja. Ali, ovaj algoritam ima i nekoliko nedostataka:

- ID3 ne garantira optimalno riješenje, već često zapinje u lokalnim optimumima zbog svoje pohlepne strategije grananja,
- ID3 često postaje preprilagođen podacima za obuku,
- ID3 je dizajniran isključivo za nominalne atribute te je podatke s kontinuiranim vrijednostima potrebno konvertirati u nominalne skupine.

7.1.2. C4.5

C4.5 predstavlja poboljšanu verziju ID3 algoritma te je prezentiran od strane istog autora - Rossa Quinlana (kao što se citira u [34]). Ovaj algoritam za potrebe odabira najboljih grananja koristi omjer dobitka (engl. *Gain ratio*), a grananje prestaje kada broj instanci koje granamo pređe određeni prag. C4.5 može baratati i s numeričkim vrijednostima te se može inducirati iz seta podataka koje sadrži nedostatne vrijednosti. C4.5 pruža nekolicinu poboljšanja u odnosu na ID3 algoritam:

- C4.5 koristi strategiju podrezivanja stabla pomoću koje se odbacuju grane koje ne doprinose točnosti modela i zamjenjuju se listovima,
- C4.5 omogućuje baratanje s setovima podataka koji imaju nedostatne vrijednosti (označene s "?"),
- C4.5 barata s kontinuiranim vrijednostima atributa.

C5.0 predstavlja ažuriranu, komercijalnu verziju C4.5 algoritma te pruža nekolicinu poboljšanja. C5.0 je puno efikasniji u odnosu na C4.5 u smislu memorije i vremena računanja. U određenim

slučajevima pruža značajna ubrzanja u odnosu na C4.5 te omogućuje proceduru jačanja (engl. *Boosting*) koja poboljšava prediktivne performanse samog modela [34].

7.1.3. CART

CART (engl. *Classification And Regression Trees*) metodologija predstavljena je 1984. godine od Breimana, Friedmana, Olshena i dr. [37] u njihovoj istoimenoj monografiji na temu stabla odlučivanja, odnosno klasifikacijska i regresijska stabla. Glavna karakteristika CART je konstrukcija binarnih stabla na način da svaki interni čvor ima točno dva izlazna ruba. Grananja se selektiraju pomoću gini indeksa (engl. *Twoing kriterija, Gini index*), a dobiveno stablo podrezuje se pomoću troškovno-kompleksnog podrezivanja (engl. *Cost-complexity prunning*). Bitna karakteristika CART pristupa je sposobnost generiranja regresijskih stabala u kojima pomoću listova predviđamo realne brojeve, a ne klase [34]. Za potrebe konstrukcije stabla koristi se rekurzivno particioniranje seta podataka za učenje pomoću "podijeli pa vladaj" strategije (engl. *Divide and conquer strategy*). Odabir najboljih grananja atributa bazira se na heurističkom kriteriju [35].

7.2. Tipovi stabla odlučivanja

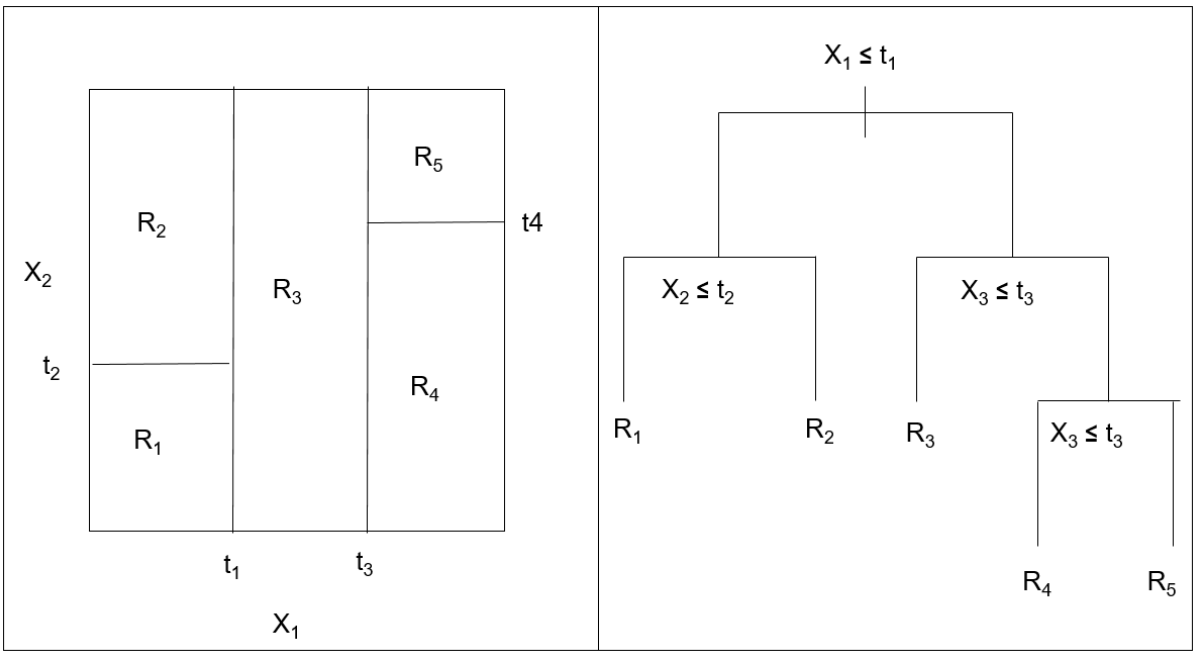
Stabla odlučivanja dijele su u dva glavna podskupa: **klasifikacijska stabla** i **regresijska stabla**. Kada se stablo odlučivanja koristi za rješavanje klasifikacijskog zadatka zove se klasifikacijsko stablo, a kada se koristi za regresijski problem zove se regresijsko stablo [34].

7.2.1. Regresijska stabla

Proces građenja regresijskog prema Jamesu, Wittenu, Hastieu i dr. [38] ugrubo pretpostavlja dva koraka:

1. dijeljenje prostora predviđanja - skup svih mogućih vrijednosti za $X_1, X_2, ..., X_p$ u J - jedinstvenih ne preklapajućih regija $R_1, R_2, ...R_j$,
2. za svaku opservaciju koja spada u regiju R_j, daje se ista predkcija, koja je predstavljena kao srednja vrijednost varijable odgovora svih opservacije za učenje u R_j.

Pretpostavimo da su u koraku 1 dobivene dvije regije, R_1 i R_2 i da je srednja vrijednost varijable odgovora podataka za učenje u prvoj regiji jednaka 10, a u drugoj 20. Tada za danu opservaciju $X = x$ predviđa se vrijednost 10 ako je $x \in R_1$, a ukoliko je $x \in R_2$ predviđa se 20. Nadalje, postavlja se pitanje kako prema koraku 1 konstruirati regije $R_1, R_2, ...R_j$. Regije mogu biti bilo kojeg oblika, ali najčešće dijelimo prostor predviđanja na visoko-dimenzionalne pravokutnike ili kutije radi jednostavnosti interpretacije prediktivnog modela. Cilj je pronaći pravokutnike $R_1, R_2, ...R_j$ koji minimiziraju **preostali zbroj kvadrata (engl. *Residual Sum of***



Slika 29: Lijevo:Primjer dijeljenja 2-d prostora pomoću binarnog grananja s 5 regija, Desno: prikaz pomoću stabla odlučivanja (Izvor: vlastita izrada prema [38, str.308])

***Squares, RSS*)** predstavljen formulom:

$$\sum_{j=1}^{J}\sum_{i\in R_j}(y_i - \hat{y}_{R_j})^2 \tag{7.1}$$

$\hat{y}_{R_j}$ predstavlja srednju vrijednost odgovora opservacija u $j-tom$ pravokutniku. Pošto je računalno neisplativo uzeti u obzir sve moguće particije prostora atributa u J pravokutnika koristimo pohlepni *"top-down"* (odozgo prema dolje) pristup poznat kao **rekurzivno binarno grananje** (engl. *Recursive binary splitting*). Algoritam počinje odozgo i završava na dnu stabla jer počinje od vrha stabla gdje sve opservacije pripadaju istoj regiji te se sukcesivno grana u prostor predviđanja; svako grananje naznačeno je s dva nova čvora u stablu. Pohlepan je jer je svaki sljedeći korak u procesu građenja stabla najbolji u tom trenutku (ne uzima u obzir korake koji bi pridonijeli građenju boljeg stabla u budućim koracima). Kako bi se provelo rekurzivno binarno grananje prvo je potrebno odabrati ulaz (neki od atributa po kojem će se odrediti grananje) X_j i točku dijeljenje s takvu da dijeljenje prostora predviđanja na regije $\{X|X_j < s\}$ i $\{X|X_j \geq s\}$ vodi do najvećeg mogućeg smanjenja vrijednosti RSS. Odnosno uzimaju se u obzir svi ulazi $X1, ..., X_p$ i sve moguće vrijednosti točke dijeljenja s za svaki od ulaza i odabire se par takav da rezultirajuće stablo ima najmanji mogući RSS. Točnije, za svaki j i s definira se par polu-prostora. Ali, ovoga puta umjesto da se dijeli cijeli prostor predviđanja, dijeli se samo jedna od prethodno identificiranih regija. Sada postoje tri regije i treba ih dalje podijeliti kako bi minimizirali vrijednost RSS. Proces se ponavlja sve dok se ne dođe do kriterija zaustavljanja (npr. mogu se dijeliti regije sve dok svaka ne sadrži više od pet opservacija). Finalno kada su kreirane regije $R_1, ..., Rj$ moguće je predvidjeti varijablu odgovora za danu testnu opservaciju

koristeći srednju vrijednost opservacija za učenje u toj regiji:

$$R_1(j,s) = \{X|X_j < s\} \; i \; R_2 = \{X|X_j \geq s\} \tag{7.2}$$

te se traži vrijednosti j i s koje minimiziraju funkciju

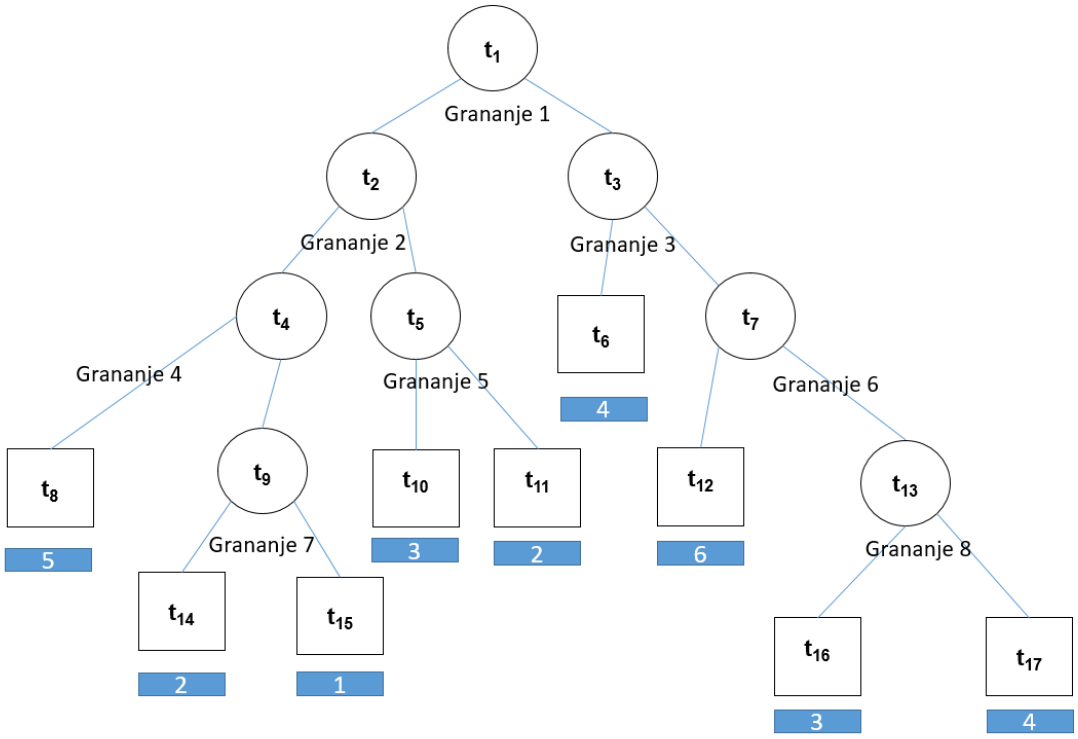
$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2 \tag{7.3}$$

u kojoj $\hat{y}_{R_1}$ predstavlja srednji vrijednost odgovora za podatke za učenje u $R_1(j,s)$, a $\hat{y}_{R_2}$ predstavlja srednji vrijednost odgovora za podatke za učenje u $R_2(j,s)$. Nadalje, ponavlja se proces traženja najboljih parova j i s kako bi se minimalizirali RSS unutar svake od rezultirajućih regija.

Slika 29 predstavlja objašnjeni pristup na primjeru s 5 regija.

7.2.2. Klasifikacijska stabla

Prema Breimanu, Friedmanu i Olshenu i dr. [37] **binarni stablasto strukturirani klasifikatori** (engl. *Binary tree structured classifiers*) konstruirani su ponavljajućim grananjem skupa X u dva podskupa.



Slika 30: Proces grananja stabla (Izvor: vlastita izrada prema [37, str. 28]

Slika 30 prikazuje proces grananja za hipotetsko stablo u kojem su t_2 i t_3 disjunktni čvorovi, za koje vrijedi $t = t_2 \cup t_3$. Slično, t_4 i t_5 su disjunktni i vrijedi $t_2 = t_4 \cup X_5$ (također isto vrijedi i za $t_3 = t_6 \cup t_7$). Nadalje, čvorovi koji se ne granaju, u ovom slučaju $t_6, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_{16}$ i t_{17}

zovu se **krajnji čvorovi** (engl. *Terminal nodes*). **Krajnji čvorovi** označavaju se pravokutnikom, dok se unutarnji čvorovi i korijenski čvor t_1 označavaju kružnicom. Krajnji čvorovi formiraju particiju od t_1 te svaki krajnji čvor dobiva oznaku klase. Moguća je situacija u kojoj dva ili više krajnja čvora imaju istu oznaku klase. Particija koja odgovara klasifikatoru dobivena je na način da se napravi unija svih krajnjih čvorova koji odgovaraju istoj klasi. Prema tome: $A_1 = t_{15}$, $A_3 = t_{10} \cup t_{16}$, $A_5 = t_8$, $A_2 = t_{11} \cup t_{14}$, $A_4 = t_6 \cup t_{17}$ i $A_6 = t_{12}$. Grananja su formirana uvjetima na koordinatama od $x = (x_1, x_2, ...)$. Na primjer, grananje 1 u t_2 i t_3 moglo bi biti oblika:

$$t_2 = \{x : x_4 \leq 7\}, t_3 = \{x : x_4 > 7\} \tag{7.4}$$

Nadalje, grananje 3 od t_3 u t_6 i t_7:

$$t_6 = \{x \in X_3 : x_3 + x_5 \leq -7\}, t_7 = \{x \in X_3 : x_3 + x_5 > -2\} \tag{7.5}$$

Dakle, stablasti klasifikator predviđa klasu za svaku vrijednost vektora x na sljedeći način: Definicija prvog grananja određuje da li x ide u čvor t_2 ili t_3 prema jednadžbi (7.4). Ako x ide u t_3, tada se po definiciji grananja 3 određuje da li x ide u t_6 ili t_7. Kada x dođe do krajnjeg čvora dobiva predviđenu oznaku klase koje odgovara oznaci klase tog krajnjeg čvora.

Prema Breimanu, Friedmanu, Olshenu i dr. [37] **konstrukcija klasifikacijskog stabla** bazira se oko tri koraka:

1. odabir grananja,
2. odluka o kriteriju zaustavljanja,
3. dodjela svakog krajnjeg čvora odgovarajućoj klasi.

Zbog toga, srž problema klasifikacije se svodi na koji način iskoristiti skup podataka L kako bi se odredila grananja, krajnji čvorovi i njihove klase (cijela priča je u pronalaženju dobrih grananja i određivanje trenutka kada prekinuti s grananjem).

Prema James, Witten, Hastie i dr. [38] **klsaifikacijska stabla** koriste se za predviđanje kvalitativnog odgovara, odnosno predviđa se da svaka opservacija pripada najčešćoj klasi iz skupa podataka za učenje u regiji kojoj pripada. Kod interpretacije rezultata klasifikacijskih stabala važne su **predikcije** klase koje korespondiraju pojedinoj regiji krajnjeg čvora i proporcije klase unutar opservacija seta podataka za učenje koje pripadaju toj regiji. Glavna razlika u odnosu na konstrukciju **regresijskih stabala** je da se za rekurzivno binarno grananje ne koristi RSS, nego neki drugi kriterij. Kod konstrukcije ili rasta klasifikacijskog stabla kao kriterij za **binarno grananje** koristi se **stopa pogreške u klasifikaciji** (engl. Classification error rate). Svakoj opservaciji u danoj regiji želi se dodijeliti najčešća oznaka klase dobivena temeljem podataka za učenje u toj regiji. Iz tog razloga koristimo **stopu pogreške u klasifikaciji** (engl. Misclassification rate) kako bi predstavili frakciju podataka koji ne pripadaju najčešćoj klasi te regije (odnosno mjera predstavlja točnost klasifikacije za dani čvor):

$$E = 1 - \max_k(\hat{p}_{mk}) \tag{7.6}$$

U formuli (7.6) $\hat{p}_{mk}$ predstavlja proporciju opservacija u $m-tom$ čvoru (koji predstavlja regiju R_m s N_m opservacija) koji su $k-te$ klase.

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k) \tag{7.7}$$

Nadalje, stopa pogreške u klasifikaciji nije dovoljno osjetljiva mjera za građenje stabla, pa se u praksi preferira koristiti druge mjere. **Gini indeks** (engl. Gini index) definiran je kao mjera ukupne varijance između K klasa:

$$G = \sum_{k=1}^{K} (\hat{p}_{mk})(1 - \hat{p}_{mk}) \tag{7.8}$$

Gini indeks poprima malu vrijednost ako su sve vrijednosti $\hat{p}_{mk}$ blizu nula ili jedan. Iz tog razloga gini indeks se referira kao mjera čistoće čvora - mala vrijednost označava da čvor sadrži uglavnom opservacije jedne klase. Alternativa gini indeksu je **entropija**:

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} log \hat{p}_{mk} \tag{7.9}$$

S obzirom da vrijedi $0 \leq \hat{p}_{mk} \leq 1$ i $0 \leq -\hat{p}_{mk} log \hat{p}_{mk}$ može se lako pokazati da entropija poprima vrijednost blizu nula kada su sve vrijednosti $\hat{p}_{mk}$ blizu nula ili jedan. Stoga, entropija ima malu vrijednost ako je $m-ti$ čvor čist (odnosno opservacije koje završavaju u njemu poprimaju uglavnom vrijednosti jedne klase). Zapravo su gini indeks i entropija u numeričkom smislu vrlo slične mjere. Kod konstrukcije klasifikacijskog stabla tipično se koriste entropija ili gini indeks kako bi se evaluirala kvaliteta određenog grananja pošto su ova dva pristupa osjetljiviji na čistoću čvorova. Bilo koji od ova tri pristupa se može koristiti kod podrezivanja stabla, ali stopa pogreške u klasifikacije se preferira ako je cilj točnost konačnog podrezanog podstabla.

7.3. Kriterij zaustavljanja

Kao što se navodi u prethodnim poglavljima svaki algoritam za građenje stabla odlučivanja mora sadržavati uvjet završavanja binarnog rekurzivnog grananja podataka, odnosno kriterij zaustavljanja koji govori kada se prestaje s grananjima određenog čvora. Prema Breimanu, Friedmanu, Olshenu i dr. [37] sljedeća pravila mogu se koristiti kao kriterij zaustavljanja.

- ako čvor postane čist; tj., svi slučajevi imaju identične vrijednosti varijable odgovora (čvor se dalje ne dijeli),
- ako sve vrijednosti u nekom čvoru imaju identičnu vrijednost odgovora za svaki ulaz,
- ako trenutna dubina stabla dosegne maksimalnu dubinu specificiranu od strane korisnika,
- ako je veličina čvora manja od minimalne veličine specificirane od strane korisnika čvor se ne dijeli,

- ako grananje čvora rezultira čvorovima djecom čija veličina je manja od minimalne veličine čvora djeteta specificirana od strane korisnika čvor se ne dijeli,
- ako je za najbolje grananje $S*$ čvora t, poboljšanje $\delta(s*, t) = p(t)\delta(s*, t)$ manje od minimalnog poboljšanja specificiranog od strane korisnika.

Nadalje, bolji pristup je da se umjesto korištenja kriterija zaustavljanja izgradi maksimalno stablo na kojem se primjenjuje tehnika podrezivanje kako bi se dobilo optimalno podstablo.

7.4. Podrezivanje stabla

Kako bi se izbjegao fenomen preprilagođavanja modela (engl. *Model overfitting*) koji označava da je dani model previše prilagođen podacima za obuku, što može značiti da postigne loše performanse na testnom skupu podatka, koristi se koncept **podrezivanja stabla** (engl. *Tree prunning*) [38]. Prema Jamesu, Wittenu, Hastieu i dr. [38] ova strategija pretpostavlja konstrukciju vrlo velikog stabla T_0 i podrezivanje istoga kako bi se dobilo podstablo (engl. *Sub-tree*). Intuitivno, cilj ovog pristupa je dobiti podstablo koje minimizira grešku predviđanja na skupu podataka za testiranje. Kako bi izbjegli generiranje velikog broja mogućih podstabala koristimo metodu **podrezivanja najslabije karike** (engl. *Weakest ling prunning, Cost-complexity prunning*). Umjesto da u obzir uzimamo sva moguća podstabla, u obzir uzimamo samo onu sekvencu stabala indeksiranu s nenegativnim parametrom za podešavanje α. Jednostavnije Jiawei, Micheline i Jian [39] navode da algoritam podrezivanja najslabije karike koji se koristi u CART metodi pretpostavlja troškovnu kompleksnost stabla kao funkciju broja listova i stopu pogreške stabla. Proces počinje od dna stabla te za svaki krajnji čvor N računa troškovnu kompleksnost podstabla koje počinje od N te troškovnu kompleksnost stabla ako bi podstablo od N bilo uklonjeno i zamijenjeno listom. Ukoliko uklanjanje podstabla čvora N rezultira manjom troškovnom kompleksnosti, podstablo se uklanja, a u protivnom se zadržava. Sljedeći algoritam predstavlja konstrukciju stabla odlučivanja pomoću metode podrezivanja:

1. izrada velikog stabla T_0 pomoću rekurzivnog binarnog grananja temeljem skupa podataka za obuku zaustavljajući se tek kada svaki krajnji čvor ima manje od zadanog minimalnog broja opservacija,
2. primjena metode podrezivanja najslabije karike kako bi se dobila sekvenca najboljih podstabala u obliku funkcije od α (za svaku vrijednost α postoji najbolje podstablo dobiveno podrezivanjem nekog od internih čvorova početkom t_0),
3. podjela podataka za učenje K puta. Za svaki k=1,...,K:
   (a) ponovi korake 1 i 2 na svim podacima osim $k - tog$ dijela,
   (b) iskoristi neku od mjera evaluacije performansi, npr. srednja kvadratna greška ako se radi o problemu regresije (referirati se na poglavje 3.1.2.) na $k - tom$ dijelu, kao funkciju od α,

   napravi prosjek rezultata s obzirom na svaku vrijednost od α i odaberi α kako bi se minimizirala srednja kvadratna greška,

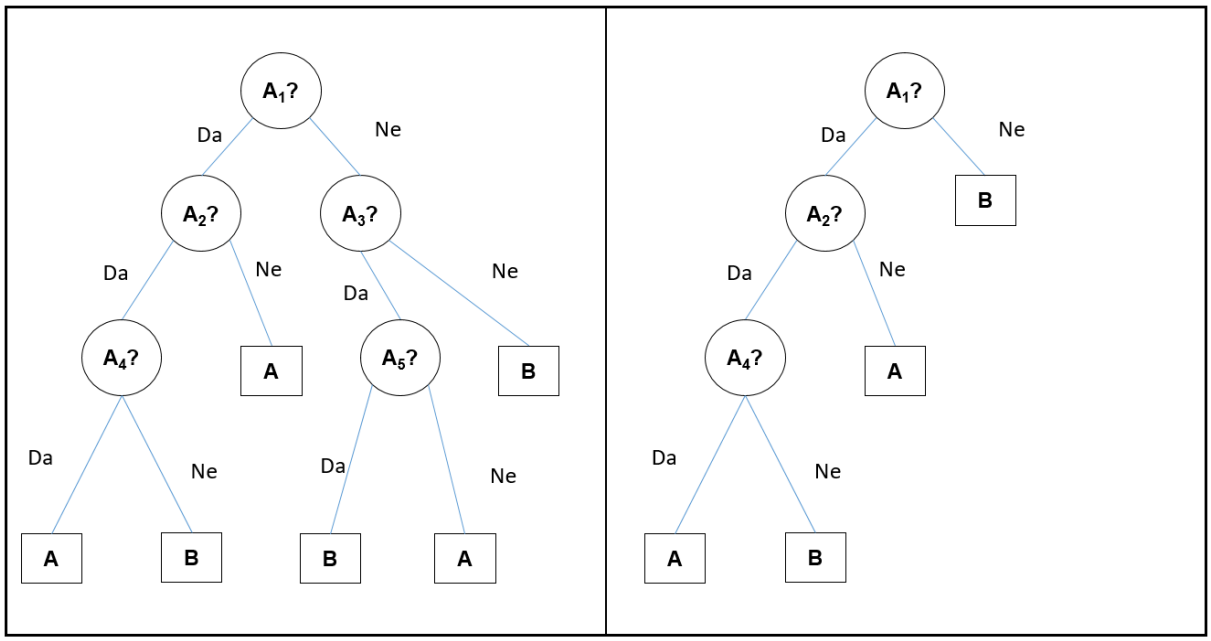
4. vrati podstablo iz koraka 2. koje odgovara odabranoj vrijednosti α.

Za svaku vrijednost α postoji podstablo $T \subset T_0$ takvo da:

$$\sum_{m=1}^{|T|} \sum_{x_i \in R_j} (y_i - \hat{y}_{R_m})^2 + \alpha |T| \tag{7.10}$$

bude najmanji mogući. U ovoj jednadžbi $|T|$ označava broj krajnjih čvorova stabla T, R_m predstavlja podskup prostora predviđanja $m - tog$ krajnjeg čvora, a $\hat{y}_{R_m}$ predstavlja predviđeni odgovor asociran s R_m, odnosno srednju vrijednost odgovora za opservacije u R_m. Parametar podešavanja α kontrolira kompromis između kompleksnosti podstabla i prilagođenosti podacima za učenje. Ako $\alpha = 0$, tada podstablo $T = T_0$. S povećanjem vrijednosti parametra α raste i cijena većeg broja krajnjih čvorova, dok količina prikazano formulom 7.10 nastoji biti minimizirana za manja stabla. S povećanjem parametra α dolazi do podrezivanja unutarnjih čvorova na ugniježđen i predvidljiv način, pa je lako dobiti sekvencu podstabala kao funkciju od α [38].

Nadalje, prema Jiwaeiu, Michelineu, Jianu [39] **podrezivanje stabla** pretpostavlja uklanjanje podstabala od konstruiranog maksimalnog stabla. Podstablo se podrezuje na određenom čvoru na način da se uklanjaju njegove grane i zamjenjuju se listom. List poprima oznaku najčešće klase uklonjenog podstabla.



Slika 31: Primjer stabla odlučivanja prije i poslije procesa podrezivanja (Izvor: vlastita izrada prema [39, str. 345])

Na primjer, ako uzmemo u obzir stablo prikazano slikom 31 i njegovo podstablo koje počinje u čvoru $A_3?$ u verziji stabla prije podrezivanja te pretpostavimo da je najčešća oznaka klase ovog podstabla "B". Spomenuto podstablo se podrezuje i zamjenjuje s listom koji također poprima oznaku klase "B".

7.5. Napredne tehnike konstrukcije stabla odlučivanja

Performanse stabla odlučivanja ovisno o podacima mogu biti vrlo loše. Iz tog razloga koriste se napredne tehnike agregiranja stabala odlučivanja kako bi se na jednostavan način poboljšanje performanse i izgradili kvalitetniji modeli [40]. **Pakiranje** (engl. *Bagging*), **pojačavanje** (engl. *Boosting*) i **nasumična šuma** (engl. *Random forest*) tehnike koriste pojedinačna stabla odlučivanja kao građevni blok pomoću kojeg se konstruiraju snažni prediktivni modeli [38].

7.5.1. Pakiranje

James, Witten, Hastie i dr. [38] navode da je **pakiranje** (engl. *Bagging*) procedura opće namijene pomoću koje se smanjuje varijanca metode za statističko učenje te se često spominje i koristi u kontekstu stabla odlučivanja. Za dani skup od n neovisnih opservacija $Z_1 ... Z_n$, svaka s varijancom σ^2 vrijedi da se srednja vrijednost $\bar{Z}$ dobiva kao omjer σ^2/n. Drugim riječima, usrednjavanje skupa opservacija smanjuje varijancu. Iz tog razloga prirodni način smanjivanja varijance i povećavanja prediktivne točnosti modela je uzimanje većeg broja skupova podataka za obuku iz populacije te kreiranje odvojenih prediktivnih modela temeljem svakog pojedinog skupa podataka. Nakon izrade odvojenih prediktivnih modela izvršava se usrednjavanje rezultirajućih predikcija. Drugim riječima, računa se $\hat{f}^1(x), \hat{f}^2(x), ..., \hat{f}^B(x)$ korištenjem B zasebnih skupova podataka za obuku te se izvršava usrednjavanje kako bi se dobio jedinstveni model niske varijance predstavljen s:

$$\hat{f}_{avg}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(x) \tag{7.11}$$

Naravno, ovaj proces nije praktičan jer često ne postoji pristup velikom broju skupova podataka za obuku. Osim toga, pakiranje se može izvršiti na način da se jedinstveni skup podataka za obuku podijeli u niz uzoraka ukupnog skupa. Kod ovakvog pristupa prvo se skup podataka za obuku dijeli na B različitih uzoraka, a potom se radi obuka modela na $b - tom$ uzorku kako bi se došlo do $\hat{f}^{*b}(x)$ te se na kraju usrednjavaju sve predikcije, kako bi se dobilo:

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x) \tag{7.12}$$

Ovakav pristup izgradnji prediktivnog modela naziva se pakiranje [38]. Iako pakiranje znatno poboljšava predikcije velikog broja regresijskih modela posebno je korisno kod stabla odlučivanja. Kako bi primijenili pakiranje na regrisijska stabla potrebno je konstruirati B regresijskih stabla korištenjem B pakiranih skupova podataka za obuku te na kraju izračunati srednju vrijednost rezultirajuće predikcije. Nadalje, kada se pakiranje koristi kod konstrukcije klasifikacijskih stabla umjesto srednje vrijednosti, radi se ukupna predikcija najčešće klase unutar B predikcija [38]. Također, ispostavlja se kako postoji vrlo jednostavan način procjene testne greške pakiranog modela bez da se koristi unakrsna validacija ili skup podataka za

validaciju. Lako se može pokazati da u prosjeku svako pakirano stablo koristi oko 2/3 opservacija, dok se preostala 1/3 opservacija ne koristi za konstrukciju modela te se naziva **opservacija iz vreće** (engl. *Out Of Bag Observation - OOB)*. Koristeći OOB opservacije možemo predvidjeti odgovor za itu opservaciju korištenjem stabla u kojoj je ta opservacija bila OOB (na taj način dobiva se B/3 predikcija za itu opservaciju). Kako bi dobili jedinstvenu predikciju za itu opservaciju, uzima se prosjek predviđenih odgovora za regresiju ili glas većine (engl. *Majority vote*) za klasifikaciju. Na taj način dobiva se OOB predikcija za svaku od n opservacija iz kojeg se može dobiti cjelokupni OOB MSE za problem regresije ili klasifikacijska greška za problem klasifikacije. Rezultirajuća OOB greška predstavlja validnu procjenu testne greške pakiranog modela, zato jer se odgovor za svaku opservaciju predviđa samo uz pomoć stabala kod kojih se ta opservacija nije koristila pri učenju. Osim toga, glavna mana korištenja pakiranja u odnosu na jedinstveno stablo odlučivanja jest smanjenje interpretabilnosti modela [38].

7.5.2. Nasumična šuma

James, Witten, Hastie i dr. [38] navode da **nasumične šume** (engl. *Random forests*) pružaju poboljšanje u usporedbi s pakiranim stablima koristeći tehniku koja dekorelira pojedinačna stabla. Isto kao kod pakiranja, kod nasumičnih šuma konstruira se određeni broj stabla odlučivanja pomoću pakiranih uzoraka podataka za obuku. Glavna razlika između ove dvije tehnike je da kod konstrukcije nasumičnih šuma svaki puta kada se u konstrukciji pojedinačnog stabla razmatra grananje odabire se nasumični uzorak od m predskazivača (atributa) od p ukupnog broja predskazivača koji se nalaze u setu podataka. Grananje smije koristiti isključivo jedan od m predskazivača. Prilikom svakog grananja uzima se novi skup m predskazivača te se tipično uzima $m \approx \sqrt{p}$ kao broj atributa koji se uzimaju u obzir prilikom izvršavanja svakog grananja (odnosno m je drugi korijen od p ukupnog skupa predskazivača). Drugim riječima, prilikom konstrukcije nasumične šume, kod svakog grananja u stablu algoritam ne smije uzeti u obzir većinu predskazivača koji su na raspolaganju. Ova tehnika koristi se kako bi se izbjegla korelacija glavnih predskazivača između pojedinačnih stabla (npr. ukoliko u skupu podataka postoji jaki predskazivač, većina stabala će kod tehnike pakiranja koristiti taj jaki predskazivač za početno grananje što će na kraju rezultirati činjenicom da će sva pakirana stabla međusobno izgledati vrlo slično). Posljedično, predikcija pakiranih stabala biti će korelirana što znači da pakiranje ne rezultira velikim smanjenjem varijance u odnosu na jedinstveno stablo. Osim toga, nasumične šume svladavaju prethodno navedeni problem na način da forsiraju svako pojedinačno grananje da u obzir uzimaju samo podskup skupa predskazivača. Zbog toga, u prosjeku $(p - m)/p$ grananja neće uzeti u obzir jake predskazivače te će na taj način drugi predskazivači imati veću šansu da budu odabrani kao kriterij grananja. Drugim riječima, na taj način postiže se dekorelacija stabala, što prosjek rezultirajućih stabala čini manje varijabilnim, a time i pouzdanijim. Nadalje glavna razlika tehnike nasumične šume i pakiranje je odabir podskupa predskazivača m. Ako $m = p$, tada tehnika nasumične šume odgovara tehnici pakiranja. Korištenjem malog broja kao vrijednost od m u konstrukciji nasumične šume poželjno je kada postoji veći broj koreliranih predskazivača te najčešće rezultira u smanjenju testne greške i OOB greške [38]. Prema Friedmanu, Hastieu,

Tibshirani [41] sljedeći algoritam služi za rješavanje regresijskih ili klasifikacijskih problema pomoću metode nasumične šume:

1. za $b = 1$ do B:
   (a) napravi uzorak $Z*$ veličine N od podataka za obuku,
   (b) konstruiraj stablo nasumične šume T_b pomoću uzorka podataka rekurzivnim ponavljanjem koraka za svaki krajnji čvor stabla sve dok se ne postigne minimalna veličina čvora n_{min},
      i. odaberi m nasumičnih atributa od p atributa,
      ii. odaberi najbolje atributa za grananje od m,
      iii. izvrši binarno grananje čvora po odabranom atributu,
   (c) vrati agregaciju stabala $\{T_b\}_1^B$.

Za dobivanje nove predikcije u točki x:
Regresija: $\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$
Klasifikacija: Neka je $\hat{C}_b(x)$ klasa predikcije $btog$ drva nasumične šume.
Onda $\hat{C}_{rf}^B(x) = majority_vote\{\hat{C}_b(x)\}_1^B$.

7.5.3. Pojačavanje

Prema Jamesu, Wittenu, Hastieu i dr. [38] **pojačavanje** (engl. *boosting*) je općeniti pristup koji može biti primijenjen na veliki broj statističkih metoda učenja za probleme klasifikacije i regresije. Pojačavanje radi na sličan način kao i pakiranje, ali glavna razlika se očituje u činjenici da se stabla konstruiraju sekvencijalno. Svako stablo koristi informacije prethodno konstruiranih stabala. Pojačavanje ne pretpostavlja dijeljenje seta podataka za učenje na uzorke, već koristiti modificiranu verziju originalnog seta podataka. Pojačavanje kao i prethodne metode koriste velik broj stabla odlučivanja $\hat{f}^1, ..., \hat{f}^B$. Za razliku od konstruiranja jedinstvenog velikog modela stabla odlučivanja koji je sklon preprilagođavanju, pristup pojačavanjem uči polako. Prema Jiaweiu, Michelineu i Jianu [39] pojačavanje pretpostavlja određivanje težina svakoj opservaciji za obuku. Iterativno se kreira k klasifikatora. Nakon što se konstruira klasifikator M_i ažuriraju se težine na opservacijama kako bi sljedeći klasifikator M_{i+1} bolje adresirao opservacije za učenje koje su prethodno krivo klasificirane. Finalni "pojačani" klasifikator, M* kombinira glasove svakog pojedinog klasifikatora u kojem je težina glasa svakog klasifikatora predstavljena kao funkcija njegove točnosti. James, Witten, Hastie i dr. [38] navode da pojačavanje koristi tri parametra za podešavanje:

- broj stabla (klasifikatora) B. Za razliku od slučajne šume i pakiranja, pojačavanje može postići preprilagođenost podacima ako je B pre velik. Za odabir vrijednosti B koristi se unakrsna validacija,
- parametar smanjivanja λ koji najčešće poprima vrijednost malog pozitivnog broja. Ovaj parametar kontrolira stopu kojom model uči. Tipične vrijednosti λ su između 0.01 i 0.001,

a pravi odabir ovisi o problemu. Vrlo mala vrijednost λ zahtijeva korištenje vrlo velike vrijednosti B kako bi se postigle dobre performanse,

- broj d grananja svakog pojedinog stabla koji kontrolira kompleksnost pojačane agregacije stabala. Najčešće $d = 1$ pruža zadovoljavajuće rezultate te je u tom slučaju svako drvo "panj" koji se sastoji od jednog grananja. Općenito d predstavlja interakcijsku dubinu (engl. *Interaction depth*) i kontrolira redoslijed interakcije pojačanog modela, jer d grananja mogu uključivati najviše d varijabli.

Sljedeći algoritam predstavlja metodu pojačavanja na primjeru regresijskih stabala:

1. postavi $\hat{f}(x) = 0$ i $r_i = y_i$ za svaki i u skupu podataka za obuku,
2. za $b = 1, 2, ..., B$, ponovi:
   (a) prilagodi stablo $hatf^b$ sa d grananja ($d+1$ krajnjih čvorova) skupu podataka za obuku (X, r),
   (b) ažuriraj $\hat{f}$ dodavanjem smanjenje verzije novog stabla,
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x) \tag{7.13}$$
   (c) ažuriraj ostatke,
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i) \tag{7.14}$$
3. Vrati pojačani model,
   $$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x). \tag{7.15}$$

8. Implementacija sustava detekcija prijevare temeljem sintetičkog skupa podataka

Opće poznata činjenica je da postoji manjak javno dostupnih skupova podataka o financijskim uslugama te posebno u novonastaloj domeni mobilnih transakcija. Financijski skupovi podataka posebno su važni onima koji provode istraživanja na području otkrivanja prijevara. Dio problema proizlazi iz privatne prirode takvih financijskih transakcija, što dovodi do manjka javno dostupnih podataka iz navedene sfere. Za potrebu istraživanja detekcije prijevara u financijskim sustavima razvijen je sintetički skup podataka generiran korištenjem Paysim simulatora. Paysim koristi agregirane podatke iz privatnog skupa podataka za generiranje sintetičkog skupa podataka koji nalikuje uobičajenom radu transakcija te uključuje zlonamjerno ponašanje kako bi se kasnije procijenila uspješnost metoda detekcije prijevara. Iz tog razloga se za potrebe praktičnog prikaza detekcija prijevare korištenjem metode stabla odlučivanja i povezanih metoda u ovom radu koristi prethodno navedeni skup podataka. Paysim simulira mobilne novčane transakcije na temelju uzorka stvarnih transakcija servisa mobilnog plaćanja koji se koristi u jednoj afričkoj zemlji. Izvorne zapise osigurala je multinacionalna tvrtka koja je pružatelj mobilne financijske usluge koja se trenutno koristi u više od četrnaest zemalja širom svijeta **Paysim**.

Prema Lopez-Rojasu, Elmiru, Axelssonu **Paysim** sintetički skup podataka skaliran je na jednu četvrtinu izvornog skupa i preuzet je s: *www.Kaggle.com* [42].

Alat koji se koristi u svrhu dohvaćanja, čitanja, grafičkog prikaza i primjenu modela nad skupom podataka je programski jezik Python3 [43]. Moduli koji su korišteni u implementaciji riješenja:

- pandas - brzi, snažan, jednostavan alat za analizu i manipulaciju podataka razvijen za Python programski jezik [44], [45],
- matplotlib - sveobuhvatna programska biblioteka za stvaranje statičkih, animiranih i interaktivnih vizualizacija u Pythonu [46],
- numpy - fundamentalni paket za znanstveno računanje s Pythonom [47], [48],
- seaborn - Python alat za vizualizaciju podataka baziran na matplotlib paketu koji pruža mogućnosti informativnih statističkih grafova [49],
- sklearn - biblioteka koja pruža jednostavne i efikasne alate za prediktivnu podatkovnu analizu bazirana na NumPy, SciPy i matplotlib paketima [50],
- xgboost - skalabilna, mobilna biblioteka za implementaciju metode pojačavanje gradijenta (engl. gradient boosting) [51],
- imblearn - paket za balansiranje neuravnoteženih skupova podataka [52].

8.1. Dobavljanje i analiza podataka

Prva faza u implementaciji prediktivnog modela strojnog učenja odnosi se na dohvaćanje podataka i analizu vrijednosti atributa samih zapisa u skupu podatka. Ovo je ključna faza u procesu implementacije jer se u njoj stvara slika o podacima te se analiziraju moguće prepreke koje proizlaze iz prirode samih podataka.

8.1.1. Dobavljanje podataka

Skup podataka spremljen je u .csv formatu te se za čitanje istog koristi *"read_csv()"* funkcija koja je dio pandas modula. Korištenjem *.head()* funkcije prikazuje se prvih deset zapisa iz same datoteke kao što je prikazano na slici 32.

```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt

Paysim = pd.read_csv('PS_20174392719_1491204439457_log.csv')

Paysim.head(10)
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.00	160296.36	M1979787155	0.0	0.00	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.00	19384.72	M2044282225	0.0	0.00	0	0
2	1	TRANSFER	181.00	C1305486145	181.00	0.00	C553264065	0.0	0.00	1	0
3	1	CASH_OUT	181.00	C840083671	181.00	0.00	C38997010	21182.0	0.00	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.00	29885.86	M1230701703	0.0	0.00	0	0
5	1	PAYMENT	7817.71	C90045638	53860.00	46042.29	M573487274	0.0	0.00	0	0
6	1	PAYMENT	7107.77	C154988899	183195.00	176087.23	M408069119	0.0	0.00	0	0
7	1	PAYMENT	7861.64	C1912850431	176087.23	168225.59	M633326333	0.0	0.00	0	0
8	1	PAYMENT	4024.36	C1265012928	2671.00	0.00	M1176932104	0.0	0.00	0	0
9	1	DEBIT	5337.77	C712410124	41720.00	36382.23	C195600860	41898.0	40348.79	0	0

Slika 32: Prvih deset zapisa iz Paysim skupa (Izvor: vlastita izrada)

Sljede objašnjenja atributa koje posjeduje svaki zapis u datoteci **Paysim**:

1. **step** - Preslikava jedinicu vremena (korak) u stvarnom svijetu. U ovom slučaju 1 korak je 1 sat. Ukupno koraka - 744 (30 dana simulacije),
2. **type** - tip transakcije: CASH-IN, CASH-OUT, DEBIT, PAYMENT, TRANSFER,
3. **amount** - iznos transakcije u orignalnoj valuti,
4. **nameOrig** - ključ klijenta (platitelja) koji je pokrenuo transakciju,
5. **oldbalanceOrig** - inicijalno stanje računa platitelja prije transakcije,
6. **newbalanceOrig** - novo stanje računa platitelja nakon transakcije,

7. **nameDest** - ključ klijenta koji je primatelj u transakciji,
8. **oldbalanceDest** - inicijalno stanje računa primatelja transakcije,
9. **newbalanceDest** - novo stanje računa primatelja nakon transakcije,
10. **isFraud** - ovaj atribut označava transakcije koje su u simulaciji napravili lažni agenti, dakle transakcije u kojima je došlo do prijevare (pražnejnje sredstava računa i prijenos na neki drugi račun),
11. **isFlaggedFraud** - pokušaj transfera više od 200.000 sredstava u jednoj transakcije.

8.1.2. Analiza podataka

Pomoću *.info()* funkcije moguće je prikazati informacije o tipovima atributa koji se koriste u zapisima. Kao što je vidljivo na sljedećem dijelu programskog koda većina atributa je numeričkog tipa *"int64"* i *"float64"* te je nekolicina atributa tipa *"object"* (radi se o tekstualnim atributima).

```
Paysim.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
step              int64
type              object
amount            float64
nameOrig          object
oldbalanceOrg     float64
newbalanceOrig    float64
nameDest          object
oldbalanceDest    float64
newbalanceDest    float64
isFraud           int64
isFlaggedFraud    int64
dtypes: float64(5), int64(3), object(3)
memory usage: 534.0+ MB
```

Pomoću *.describe()* funkcije dohvaćaju se: zbroj, prosječna vrijednost, standardna devijacija, percentili te minimalna i maksimalna vrijednost pojedinih atributa u skupu podataka koji su vidljivi na slici 33.

```
pd.options.display.float_format = "{:.2f}".format
Paysim.describe().transpose()
```

Nadalje, uz pomoć *.isnull().sum().max()* funkcije se provjera postojanje nepostojećih *"NaN"* vrijednosti te se iz izlaza može zaključiti da su sve vrijednosti atributa popunjene.

```
Paysim.isnull().sum().max()
0
```

	count	mean	std	min	25%	50%	75%	max
step	6362620.00	243.40	142.33	1.00	156.00	239.00	335.00	743.00
amount	6362620.00	179861.90	603858.23	0.00	13389.57	74871.94	208721.48	92445516.64
oldbalanceOrg	6362620.00	833883.10	2888242.67	0.00	0.00	14208.00	107315.18	59585040.37
newbalanceOrig	6362620.00	855113.67	2924048.50	0.00	0.00	0.00	144258.41	49585040.37
oldbalanceDest	6362620.00	1100701.67	3399180.11	0.00	0.00	132705.66	943036.71	356015889.35
newbalanceDest	6362620.00	1224996.40	3674128.94	0.00	0.00	214661.44	1111909.25	356179278.92
isFraud	6362620.00	0.00	0.04	0.00	0.00	0.00	0.00	1.00
isFlaggedFraud	6362620.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

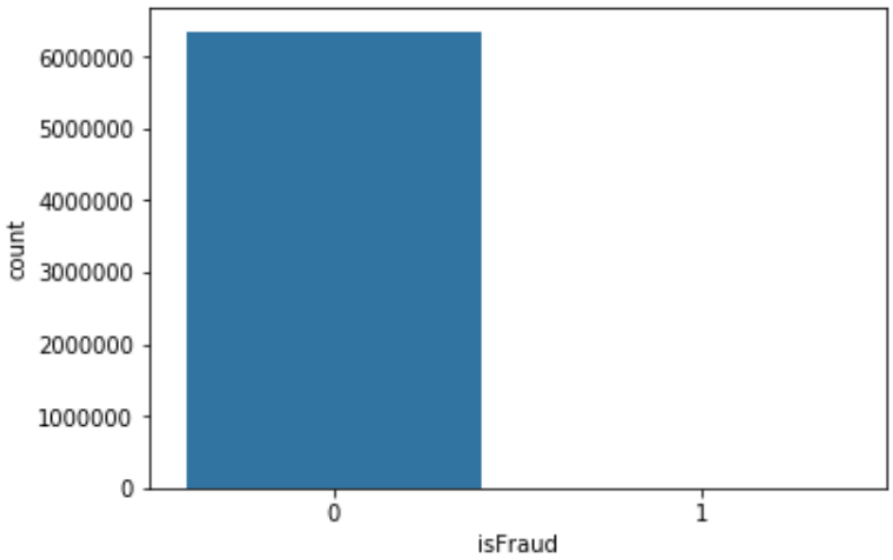
Slika 33: Prikaz agregiranih numeričkih podataka iz skupa (Izvor: vlastita izrada)

Sljedeće dvije funkcije koriste se kako bi se pobrojale prijevarne i validne transakcije u skupu podataka. Temeljem izlaza koji vraćaju funkcije u skupu podataka nalazi se 8213 prijevarnih transakcija i 6354407 validnih transakcija što govori o **neuravnoteženosti** skupa podataka u smislu tipova transakcija.

```
Paysim[Paysim['isFraud'] != 0]['isFraud'].count()
8213
Paysim[Paysim['isFraud'] == 0]['isFraud'].count()
6354407
```

Pomoću sljedeće funkcije izrađuje se graf prikazan na slici 34 koji prikazuje zbroj validnih (0) u odnosu na prijevarne (1) transakcije.

```
sns.countplot(x='isFraud', data = Paysim)
```



Slika 34: Graf prikaza zbroja validnih u odnosu na prijevarne transakcije (Izvor: vlastita izrada)

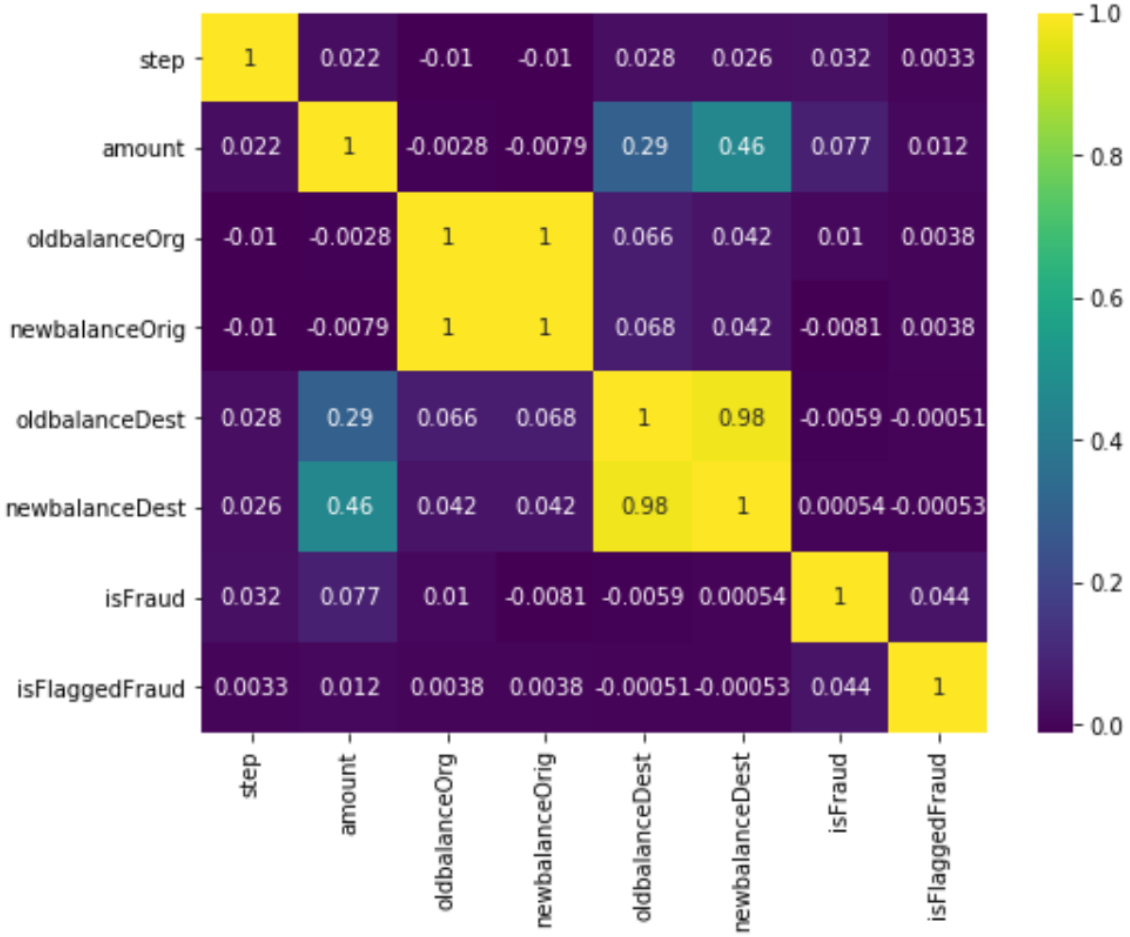
```
print('NP:', round(Paysim['Prijevara'].value_counts()[0]/len(Paysim) * 100,2),
```

```
'\% skupa podataka')
print('P:', round(Paysim['Prijevara'].value_counts()[1]/len(Paysim) * 100,2),
'\% skupa podataka')
```

```
Nema prijevare: 99.87 % skupa podataka
Prijevara: 0.13 % skupa podataka
```

Nadalje, uz pomoć prethodne dvije funkcije vidljivo je da postotak transakcija u kojima nema prijevare iznosi 99.87%, dok 0.13% od ukupnog broja podataka otpada na prijevarne transakcije, što još više govori o neuravnoteženosti samog skupa podataka.

Prema Statistic Solutions [53] **korelacija** prikazuje odnos između dvije ili više varijabli. Točnije, **korelacija** je stupanj mjere što znači da može biti pozitivna, negativna ili savršena. Drugim riječima, pozitivna korelacija znači da ako postoji povećanje (smanjenje) u jednoj varijabli, tada postoji istovremeno povećanje (smanjenje) u drugoj varijabli. Negativna korelacija je vrsta korelacije kod koje, ako postoji smanjenje (povećanje) jedne varijable, tada postoji istovremeno povećanje (smanjenje) druge varijable. Savršena korelacija je tip korelacije kod koje promjena u jednoj varijabli uzrokuje ekvivalentnu promjenu druge varijable [53].



Slika 35: Matrica korelacije atributa (Izvor: vlastita izrada)

Kako bi se prikazali stupnjevi korelacije između atributa koristi se funkcija:

```
plt.figure(figsize=(8,6))
```

```
sns.heatmap(Paysim.corr(), annot=True, cmap='viridis')
```

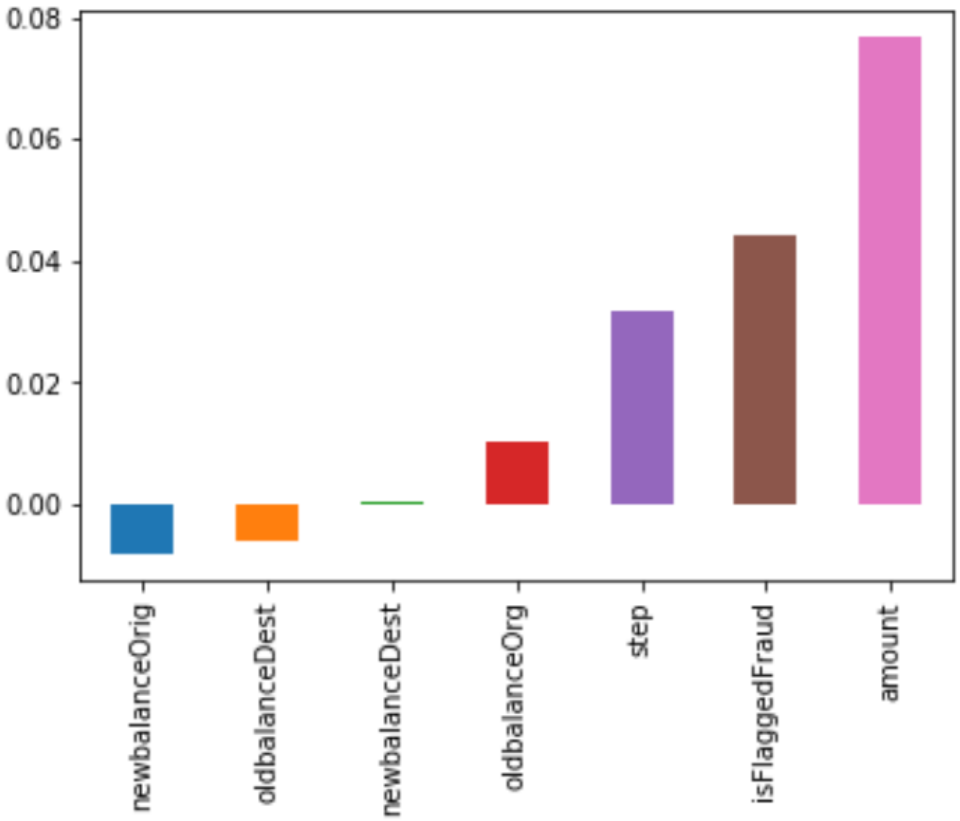
koja generira matricu korelacije atributa prikazanu na slici 35.

Sa matrice korelacije prikazane na slici 35 može se isčitati da su atributi *oldbalanceOrg* i *newbalanceOrig* te *oldbalanceDes*t i *newbalanceDest* savršeno korelirani što potvrđuje njihove opise predstavljene u prethodnom dijelu teksta.

Pomoću funkcije:

```
Paysim.corr()['isFraud'].sort_values()[:-1].plot(kind='bar')
```

generira se stupčasti graf koji prikazuje korelacije atributa u odnosu na ciljnu vrijednost "isFraud" i prikazan je na slici 36. Temeljem grafa prikazanog na slici 36 može se zaključiti

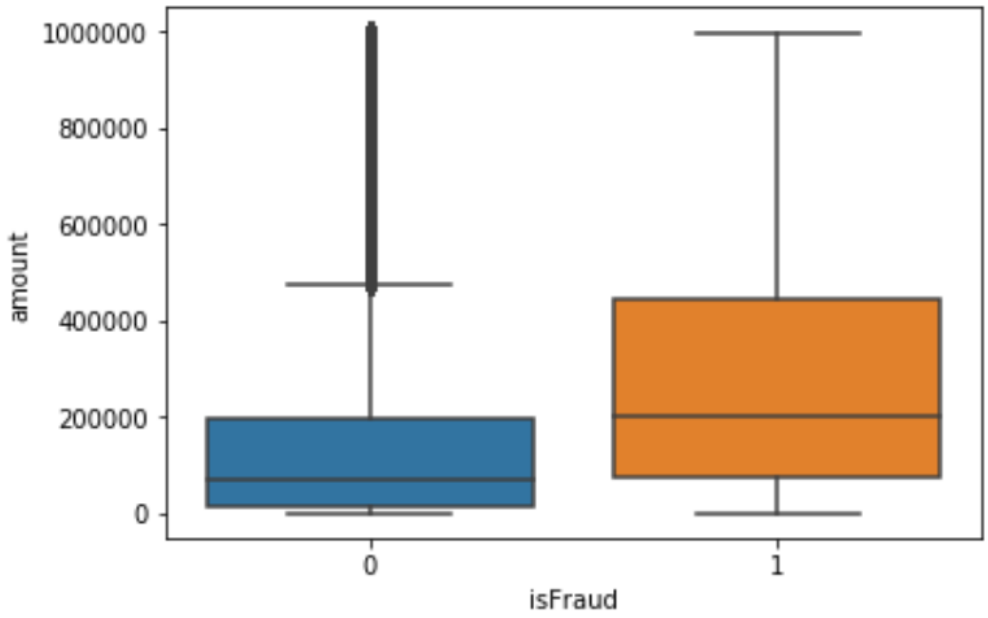


Slika 36: Stupčasti graf korelacija atributa u odnosu na "isFraud" atribut (Izvor: vlastita izrada)

da imamo male stupnjeve korelacije između ciljne vrijednosti i ostalih atributa što znači da niti jedan od atributa nije savršeni predskazivač ciljne vrijednosti. Pošto atribut *"amount"* ima najveći pozitivni stupanj korelacije s ciljnom vrijednošću, pretpostavka je da je on i najbolji predskazivač ciljne vrijednosti. Pomoću sljedeće funkcije može se lako analizirati odnos između *"isFraud"* i *"amount"* atributa:

```
sns.boxplot(x='isFraud', y='amount', data=Paysim)
```

Navedena funkcija generira graf prikazan na slici 37. Sa grafa se može lako iščitati da je prosječna vrijednost "amount" kolone jednaka 200000 jedinica za slučaj prijevarne transakcije, dok je za slučaj valdine transakcije puno niža i iznosi oko 100000 jedinica.



Slika 37: Graf odnosa "isFraud" i "amount" atributa (Izvor: vlastita izrada)

Nadalje uz pomoć *.describe()* funkcije:

```
Paysim.groupby('isFraud')['amount'].describe()
```

moguće je pobliže analizirati odnos "isFraud" i "amount" atributa, kao što je vidljivo na slici 38.

	count	mean	std	min	25%	50%	75%	max
isFraud								
0	6226487.00	129934.22	156737.00	0.01	12974.47	70188.61	198292.87	999988.34
1	5507.00	289757.24	265619.90	0.00	73110.82	201281.36	445481.59	998110.47

Slika 38: Tablični prikaz odnosa "isFraud" i "amount" atributa (Izvor: vlastita izrada)

8.2. Čišćenje podataka

Nakon analize podataka slijedi faza čišćenja i prilagođavanja podataka.

8.2.1. Inženjering atributa

U prethodnom dijelu teksta navedeno je da atributi *"nameOrig"* i *"nameDest"* predstavljaju identifikatore ili ključeve u izvornom sistemu. Pošto takav atribut nije pogodan za obuku samog modela iz njih je moguće generirati novi kategorički atribut pomoću sljedećeg niza naredbi.

```
Paysim["tip_transakcije"] = np.nan
```

```
Paysim.loc[Paysim.nameOrig.str.contains('C') &
Paysim.nameDest.str.contains('C'),"tip_transakcije"] = "CC"
Paysim.loc[Paysim.nameOrig.str.contains('C') &
Paysim.nameDest.str.contains('M'),"tip_transakcije"] = "CM"
Paysim.loc[Paysim.nameOrig.str.contains('M') &
Paysim.nameDest.str.contains('C'),"tip_transakcije"] = "MC"
Paysim.loc[Paysim.nameOrig.str.contains('M') &
Paysim.nameDest.str.contains('M'),"tip_transakcije"] = "MM"
```

Nakon izvršavanja navedenih naredbi kreiran je novi atribut "tip_transakcije" koji se temelji na "nameOrig" i "nameDest" izvorišnim atributima.

```
Paysim2['tip_transakcije'][Paysim2['isFraud']==1].unique()
array(['CC'], dtype=object)
Paysim2['tip_transakcije'][Paysim2['isFraud']==0].unique()
array(['CM', 'CC'], dtype=object)
```

Također pomoću prethodne dvije naredbe vidljivo je da prijevarne transakcije poprimaju vrijednost atributa "tip_transakcije" - "CC", dok validne transakcije mogu poprimiti vrijednosti "CC" i "CM". Nakon generiranja novog atributa, stari atributi *"nameDest"* i *"nameOrig"* koji više nisu potrebni se uklanjaju iz skupa podataka uz pomoć naredbe:

```
Paysim.drop(['nameOrig','nameDest'],axis=1,inplace=True)
Paysim.drop(['nameOrig','nameOrig'],axis=1,inplace=True)
```

8.2.2. Zamijena kategoričkih atributa

Prediktivni modeli **scikit-learn** programske biblioteke mogu isključivo baratati numeričkim vrijednostima [54]. Iz tog razloga je potrebno pretvoriti kategoričke atribute u oblik pogodan za daljnju analizu. Nadalje, ***"dummy"*** varijabla je numerička varijabla koja se koristi za reprezentiranje kategoričkih podataka poput spola, rase, političke opredijeljenosti, itd [55].

```
tip_transakcije = pd.get_dummies(Paysim['tip_transakcije'], drop_first=True)
Paysim2=pd.concat([Paysim.drop('tip_transakcije', axis=1),tip_transakcije],axis=1)
dummies = pd.get_dummies(Paysim['type'], drop_first=True)
Paysim2=pd.concat([Paysim.drop('type',axis=1),dummies],axis=1)
```

Prethodne četiri naredbe koji koriste *.get_dummies()* funkciju **pandas** paketa služe za zamjenu kategoričkih atributa *"type"* i "tip_transakcije" odgovarajućim numeričkim atributima. Nakon zamijene kategoričkih atributa zapisi poprimaju oblik prikazan na slici 39.

	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud	CM	CASH_OUT	DEBIT	PAYMENT	TRANSFER
0	1	9839.64	170136.0	160296.36	0.0	0.0	0	0	1	0	0	1	0
1	1	1864.28	21249.0	19384.72	0.0	0.0	0	0	1	0	0	1	0
2	1	181.00	181.0	0.00	0.0	0.0	1	0	0	0	0	0	1
3	1	181.00	181.0	0.00	21182.0	0.0	1	0	0	1	0	0	0
4	1	11668.14	41554.0	29885.86	0.0	0.0	0	0	1	0	0	1	0

Slika 39: Prvih pet zapisa skupa podataka nakon zamijene kategoričkih varijabli (Izvor: vlastita izrada)

8.2.3. Podjela skupa podataka

Prije nego što se krene u fazu konstrukcije i obuke modela, originalan skup podataka potrebno je podijeliti na **skup podataka za obuku** i **skup podataka za testiranje**.

```
from sklearn.model_selection import train_test_split
X = Paysim.drop("isFraud",axis=1).values
y = Paysim['isFraud'].values
X_train, X_test, y_train, y_test = train_test_split(X, y test_size=0.3)
```

Kako bi se podaci podijelili na skup podataka za obuku i testiranje koristi se funkcija ***train_test_split()*** koja pripada **sklearn** paketu. Nadalje, pomoću prethodno navedenih naredbi originalni skup podataka dijeli se na skup podataka za obuku *"X_train"* i *"y_train"* i *"X_test"* i *"y_test"* u omjeru 7:3.

8.2.4. Standardizacija podataka

Standardizacija skupa podataka je česti zahtjev za veliki broj modela strojnog učenja koji su implementirani u **sklearn** paketu. Razlog tome je činjenica da se prediktivni modeli ponašaju čudno ukoliko podaci ne slijede neki oblik standardne distribucije [56]. Kako bi postigli standardizaciju podatka koristimo ***"MinMaxScaler"*** koji je dio sklearn paketa, a zadaća mu je da svaki atribut transformira u broj između 0 i 1 po formulama:
$X_{std} = (X - X.min(axis = 0))/(X.max(axis = 0) - X.min(axis = 0))$
$X_{scaled} = X_{std} * (max - min) + min$ [57].
Nadalje, uz pomoć sljedećih naredbi odrađuje se standardizacija, odnosno skaliranje podataka.

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

8.2.5. Balansiranje podataka

Kao što je zaključeno u fazi analize podataka, Paysim skup podataka je vrlo neuravnotežen te je omjer validnih u odnosu na prijevarne transakcije iznosi 99.87 : 0.13. Kako bi se navedeni problem riješio koristi se tehnika **preuzorkovanja** (engl. *Oversampling*). Tehnika preuzorkovanja koristi se kada je jedna ciljna klasa skupa podataka u manjini, a zadaća joj je dupliciranje manjinske klase kako bi se postigla ravnoteža klasa [58]. Za potrebe preuzorkovanja u ovom primjeru koristi se *"RandomOverSampler"* funkcija **imblearn** paketa pomoću sljedećih naredbi:

```
from imblearn.over_sampling import RandomOverSampler
from collections import Counter
rus = RandomOverSampler(random_state=101)
X_resampled, y_resampled = rus.fit_resample(X_train, y_train)
print(sorted(Counter(y_resampled).items()))
------------------------------------------------------------
```

```
[(0, 4448038), (1, 4448038)]
```

Nakon izvršavanja prethodnih naredbi jedna i druga ciljna klasa u skupu podataka za obuku imaju isti broj zapisa te on iznosi 4448038. Na taj način se postiže ravnoteža između validnih i prijevarnih transakcija u skupu podataka za obuku

8.3. Građenje i obuka modela

Slijedi faza građenja i obuka modela. Za potrebe primjera implementacije sustava detekcije prijevare u ovom radu se koriste tri različita prediktivna modela:

- **sklearn - DecisionTreeClassifier** - jednostavni klasifikator stabla odlučivanja implementiran pomoću CART metodologije,
- **sklearn.ensamble - RandomForestClassifier** - klasifikator nasumične šume koji odgovara nizu klasifikatora stabla odlučivanja na različitim pod-uzorcima skupa podataka te koristi prosječne vrijednosti za poboljšanje prediktivne točnosti,
- **xgboost.sklearn - XGBClassifier** - klasifikator koji koristi pojačavanje, odnosno tzv. *"gradient boosting"* - pristup u kojem se novi modeli stabla odlučivanja treniraju kako bi se predviđali ostatci, odnosno greške prethodnih modela.

8.3.1. Sklearn - DecisionTreeClassifier

Građenje i obuka modela stabla odlučivanja za potrebe klasifikacije uz pomoću sklearn paketa izvršava se pomoću sljedećih naredbi:

```
# Train model:
from sklearn.tree import DecisionTreeClassifier
dtree=DecisionTreeClassifier()
dtree.fit(X_resampled,y_resampled)

# Predict on test data:
predictionsDT = dtree.predict(X_test)
```

Pomoću klase "*DecisionTreeClassifier()*" radi se inicijalizacija objekta stabla odlučivanja s osnovnim postavkama. Pomoću funkcije "*dtree.predict(X_test)*" izvršava se predviđanje vrijednosti ciljne klase temeljem testnog skupa podataka.

8.3.2. Sklearn - RandomForestClassifier

Građenje i obuka modela nasumične šume pomoću sklearn paketa izvršava se pomoću sljedećih naredbi:

```
# Train model:
from sklearn.ensemble import RandomForestClassifier
parametersRF = {'n_estimators':500,'oob_score':True,
```

```
'class_weight': "balanced",'n_jobs':-1}
RF = RandomForestClassifier(**parametersRF)
fitted_vals = RF.fit(X_resampled, y_resampled)

# Predict on test data
predictionsRF = RF.predict(X_test)
```

Pomoću klase "*RandomForestClassifier(**parametersRF)*" inicijalizira se model s parametrima koji su navedeni "*parametersRF*" objektu. Ključni parametar koji se postavlja prilikom inicijalizacije modela je "*'n_estimators':500*" koji označava da se u nasumičnoj šumi konstruira 500 različitih stabla odlučivanja. Za potrebe odabira grananja koristi se osnovna vrijednost - "*'criterion':"gini"*" što znači da se za odabir najboljih grananja koristi gini indeks. Pomoću funkcije "*RF.predict(X_test)*" radi se predviđanje ciljne klase temeljem testnog skupa podataka uz pomoću modela.

8.3.3. xgboost.sklearn - XGBClassifier

Građenje i obuka modela pojačavanja stabla pomoću xgboost.sklearn paketa izvršava se pomoću sljedećih naredbi:

```
# Train model
from xgboost.sklearn import XGBClassifier
parametersXGB = {'max_depth':5,'n_jobs':-1,
                 'learning_rate':0.2, 'n_estimators':1000}
XGB = XGBClassifier(**parametersXGB)

fitted_vals = XGB.fit(X_resampled, y_resampled)

# Predict on test data
predictionsXGB = XGB.predict(X_test)
```

Uz pomoć klase "*XGBClassifier(**parametersXGB)*" odvija se inicijalizacija modela s parametrima koji su navedeni u "*parametersXGB*" objektu. Ključni parametri koji se postavljaju prilikom inicijalizacije modela su "*'max_depth':5*" koji označava maksimalni broj grananja pojedinog stabla, "*'learning_rate':0.2*" koji označava parametar smanjivanja i "*'n_estimators':1000*" označava broj stabala odlučivanja koji će se kreirati prilikom obuke samog modela. Nadalje uz pomoć funkcije "*.predict(X_test)*" koja se poziva na klasifikatoru radi se predviđanje vrijednosti ciljne klase temeljem testnog skupa podataka.

8.4. Testiranje modela

Slijedi faza testiranja i evaluacije modela. Za potrebe evaluacije performansi prediktivnih modela u ovom radu koriste se mjere: matrica zabune, odaziv, preciznost, F1-ocjena, točnost i krivulja operativne karakteristike prijamnika - (engl. *Receiver operating characteristic*).

Pomoću sljedećih naredbi izvršava se import potrebnih klasa za generiranje metrika performansi:

```
from sklearn.metrics import classification_report,confusion_matrix,
accuracy_score, roc_curve, auc, precision_score
```

8.4.1. Generiranje performansi - sklearn - DecisionTreeClassifier

Izvršavanjem sljedećih naredbi generiraju se metrike performansi modela stabla odlučivanja kao što su prikazane na slici 40.

```
# Evaluating model
CM_DT = confusion_matrix(y_test,predictionsDT)
CR_DT = classification_report(y_test,predictionsDT)
fprDT, recallDT, thresholds_DT = roc_curve(y_test, predictionsDT)
AUC_DT = auc(fprDT, recallDT)
resultsDT = {"Confusion␣Matrix":CM_DT,
"Classification␣Report":CR_DT,"Area␣Under␣Curve":AUC_DT}
#Output
for measure in resultsDT:
    print(measure,":␣\n",resultsDT[measure],"\n")
```

```
Area Under Curve :
 0.9231759044205929

Classification Report :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   1906369
           1       0.88      0.85      0.86      2417

    accuracy                           1.00   1908786
   macro avg       0.94      0.92      0.93   1908786
weighted avg       1.00      1.00      1.00   1908786


Confusion Matrix :
 [[1906079     290]
 [    371    2046]]
```

Slika 40: Prikaz mjera performansi - sklearn - DecisionTreeClassifier (Izvor: vlastita izrada)

8.4.2. Generiranje performansi - sklearn - RandomForestClassifier

Izvršavanjem sljedećih naredbi generiraju se metrike performansi modela nasumične šume kao što su prikazane na slici 41.

```
# Evaluating model
CM_RF = confusion_matrix(y_test,predictionsRF)
CR_RF = classification_report(y_test,predictionsRF)
```

```
fprRF, recallRF, thresholdsRF = roc_curve(y_test, predictionsRF)
AUC_RF = auc(fprRF, recallRF)
resultsRF = {"Confusion Matrix":CM_RF,"Classification Report":CR_RF,"Area Under Curve":AUC_RF}
# Output
for measure in resultsRF:
    print(measure,": \n",resultsRF[measure])
```

```
Area Under Curve :
 0.902961326788647
Classification Report :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   1906369
           1       0.97      0.81      0.88      2417

    accuracy                           1.00   1908786
   macro avg       0.98      0.90      0.94   1908786
weighted avg       1.00      1.00      1.00   1908786

Confusion Matrix :
 [[1906302      67]
 [    469    1948]]
```

Slika 41: Prikaz mjera performansi - sklearn - RandomForestClassifier (Izvor: vlastita izrada)

```
Area Under Curve :
 0.978333604455584

Classification Report :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   1906369
           1       0.80      0.96      0.87      2417

    accuracy                           1.00   1908786
   macro avg       0.90      0.98      0.94   1908786
weighted avg       1.00      1.00      1.00   1908786


Confusion Matrix :
 [[1905789     580]
 [    104    2313]]
```

Slika 42: Prikaz mjera performansi - xgboost.sklearn - XGBClassifier (Izvor: vlastita izrada)

8.4.3. Generiranje performansi - xgboost.sklearn - XGBClassifier

Izvršavanjem sljedećih naredbi generiraju se metrike performansi modela pojačavanja stabla kao što su prikazane na slici 42.

```
# Evaluating model
CM_XGB = confusion_matrix(y_test,predictionsXGB)
CR_XGB = classification_report(y_test,predictionsXGB)
fprXGB, recallXGB, thresholds_XGB = roc_curve(y_test, predictionsXGB)
AUC_XGB = auc(fprXGB, recallXGB)
resultsXGB = {"Confusion␣Matrix":CM_XGB,
"Classification␣Report":CR_XGB,"Area␣Under␣Curve":AUC_XGB}
# Output
for measure in resultsXGB:
    print(measure,":␣\n",resultsXGB[measure],"\n")
```

8.4.4. Usporedba performansi modela

Slijedi usporedba performansi modela prema generiranim mjerama.

Usporedba metrika: preciznost, opoziv i f-1 ocjena:

DecisionTreeClassifier	RandomForestClassifier	XGBClassifier
Klasa 0: 1.00	1.00	1.00
Klasa 1: 0.88	0.97	0.80
Macro avg: 0.94	0.98	0.90

Tablica 14: Usporedba metrike - preciznost (engl. Precision)

Model s najvećom vrijednošću metrike preciznost, odnosno sposobnosti da točno klasificira samo prepoznate instance je RandomForestClassifier.

DecisionTreeClassifier	RandomForestClassifier	XGBClassifier
Klasa 0: 1.00	1.00	1.00
Klasa 1: 0.85	0.81	0.96
Macro avg: 0.94	0.90	0.98

Tablica 15: Usporedba metrike - opoziv (engl. Recall)

Model s najvećom vrijednošću metrike opoziv, odnosno sposobnosti da točno klasificira sve relevantne instance u skupu podataka je XGBClassifier.

DecisionTreeClassifier	RandomForestClassifier	XGBClassifier
Klasa 0: 1.00	1.00	1.00
Klasa 1: 0.86	0.88	0.87
Macro avg: 0.93	0.94	0.94

Tablica 16: Usporedba metrike - f-1 ocjena (engl. F-1 score)

Modeli s najvećom vrijednošću metrike f-1 ocjene, odnosno harmonijskom sredinom preciznosti i opoziva su RandomForestClassifier i XGBClassifier koji su postigli iste performanse.

Usporedba metrike: matrica zabune

DecisionTreeClassifier	RandomForestClassifier	XGBClassifier
TP: 19606079	1906302	1905789
FP: 371	469	104
FN: 290	67	580
TN: 2046	1948	2313

Tablica 17: Usporedba metrike - matrica zabune (confusion matrix)

Ako uzmemo u obzir da skup podataka za testiranje sadrži puno veći broj instanci klase 0, odnosno validnih transakcija XGBClassifier ima najbolje performanse s obzirom na mjere TN (istinitno negativna vrijednost) - 2313, što znači da je model od ukupno 2417 slučaja prijevarnih transakcija istinito klasificirao 2313 transakcije (također vrijednost FP, odnosno lažno pozitivnih transakcija je najmanja za XGBClassifier). Što se tiče mjera TP i FN najbolje rezultate pak daje RandomForestClassifier. Ako se uzme u obzir pretpostavka da je trošak većeg broja lažno pozitivnih transakcija veći od troška većeg broja lažno negativnih transakcija najbolje performanse daje model XGBClassifier.

Usporedba metrike: AUC - ROC krivulja

Prema Narkhede [59] AUC - ROC krivulja je mjera performansi koja se koristi kod problema klasifikacije. ROC ili krivulja operativne karakteristike prijamnika predstavlja krivulju vjerojatnosti, dok AUC - prostor ispod krivulja (engl. *Area Under Curve, AUC*) predstavlja stopu separabilnosti klasa. Što je veća vrijednost AUC, model je bolji u predviđanju TP i TN vrijednosti. Analogno, što je veći AUC, to je model bolji u razlikovanju između različitih klasa koje predviđa.

DecisionTreeClassifier	RandomForestClassifier	XGBClassifier
AUC vrijednost: 0.9231	0.9029	0.9783

Tablica 18: Usporedba metrike - AUC (engl. area under curve)

Kao što se može iščitati iz tablice 18, najbolje performanse mjere AUC ponovno prikazuje XGBClassifier. Zaključno se XGBClassifier pokazao kao najbolji prediktivni model za detekciju prijevarnih transakcija na ovom skupu podataka.

9. Primjeri upotrebe stabla odlučivanja u bankarskom sektoru

Ovo poglavlje fokusirati će se na praktične primjere upotrebe modela stabla odlučivanja iz područja strojnog učenja za specifične probleme bankarskog i financijskog sektora. Prvi primjer odnosi se na implementaciju modela za procjenu kreditnog rizika u jednoj Indonezijskoj ruralnoj banci. Drugi primjer se odnosi na implementaciju sustava detekcije prijevare pomoću stabla odlučivanja i komparativnu analizu performansi različitih algoritama za implementaciju stabla odlučivanja. Treći i četvrti primjer prikazati će upotrebu stabla odlučivanja iz područja teorije odlučivanja za rješavanje problema podizanja kredita kredita i investiranja od strane fonda rizičnog kapitala.

9.1. Primjeri iz strojnog učenja

9.1.1. Procjena kreditnog rizika: primjena metode stabla odlučivanja u ruralnoj banci PT BPR X

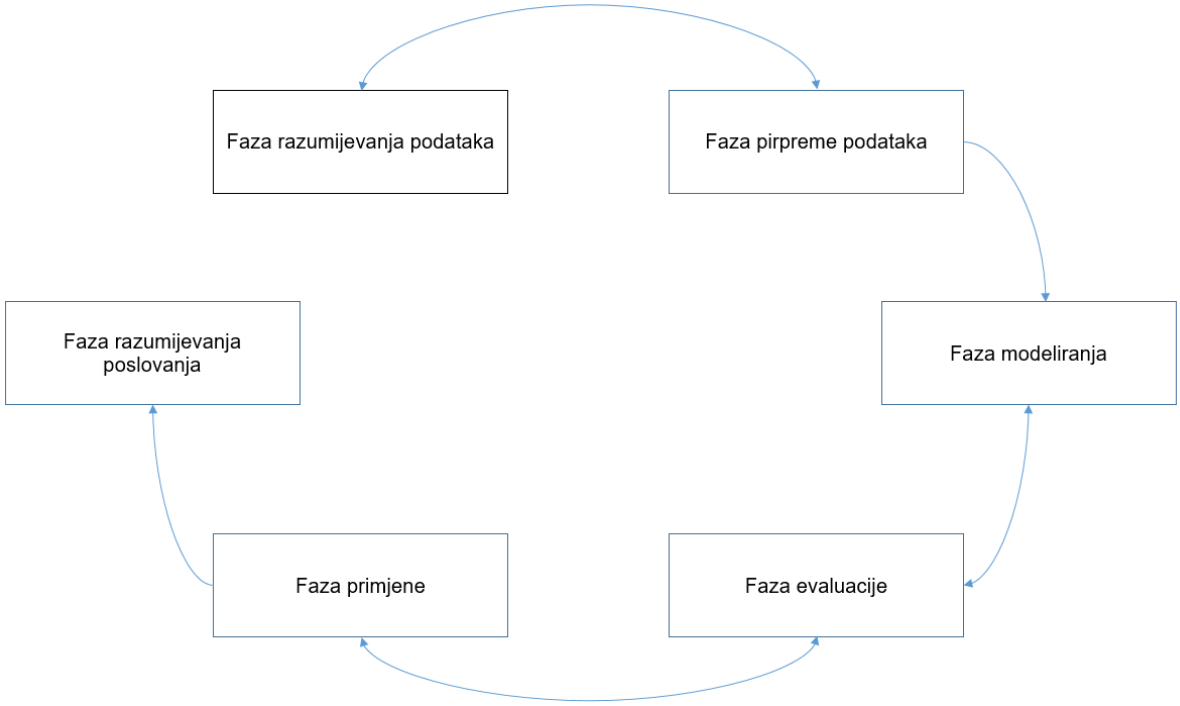
Ruralna banka (engl. *Rural bank, Bank Perkreditan Rakyat, BPR, People's Credit Bank*) je banka koja ima specifičnu poslovnu djelatnost u posluživanju uzajamnih zajmova i štednja za mala ruralna područja, mikro, mala i srednja poduzeća (engl. *Cicro Small Medium Enterprises,MSME*). U istraživanju se koristi pojam BPR kada se spominje ruralna banka. Glavna razlika između poslovnih i ruralnih banaka je da za razliku od poslovnih, ruralne banke ne pružaju usluge platnih transakcija. To znači da se ruralne banke uglavnom fokusiraju na uzajamne zajmove ili usluge štednje. Nadalje, Bank Indonesia (cenralna banka Republike Indonezije) na svojoj web stranici, obavještava da u travnju 2012. postoji 1667 konvencionalnih BPR-a koji ukpuno imaju više od 12 milijuna računa od kojih četvrtina otpada na račune uzajamnih zajmova, dok ostatak otpada na štedne račune [60].

Kreditna procjena je istraživanje zahtjeva za kredit. Prema Rivaiu (kao što se citira u [60]) kreditna procjena se izvodi za procjenu da li potencijalni zajmodavac ima poslovne aktivnosti koje su izvedive, profitabilne i da li će kredit biti vraćen na vrijeme. Najčešće se procjena kredita izvodi od strane bankarskih službenika koji su dio povjerenstva za ocjenjivanje. Ovakve procjene se odvijaju kako bi se analizirali svi faktori uključeni u zahtjev za kreditom kao što su: financijske performanse i kreditni rejting zajmodavca [60].

Prema Mandali, Nawangpalupiju, Praktiktou [60] istraživanje se izvršilo s ciljem identificiranja faktora koji su potrebni ruralnoj banci da ocjene kreditne zahtjeve. Cilj istraživanja je smanjiti broj nekvalitetnih zajmova. Ocjenjivali su se trenutni kriteriji odlučivanja o ocjeni kreditnog rizika te se nakon toga implementirao model **stabla odlučivanja** pomoću metodologije rudarenja podacima. **Model procjene kreditnog rizika** primjenjuje se na PT BPR X banku lociranu na Baliju koja ima 1082 zajmodavca od kojih je 11,99% identificirano kao slučajevi loših zajmova, što je PT BPR X kategoriziralo kao banku s lošim rezultatima. Prema Bank Indonesia (kao se citira u [60]) dobro poslujuća BPR ima stopu nenaplativih zajmova

(engl. *Non-performing loan's - NPL*) manju od 5%. Cilj samog istraživanja je bio smanjiti stopu NPL i poboljšati performanse kretidne procjene u PT BPR X izradom kvalitetnijeg modela kreditne procjene.

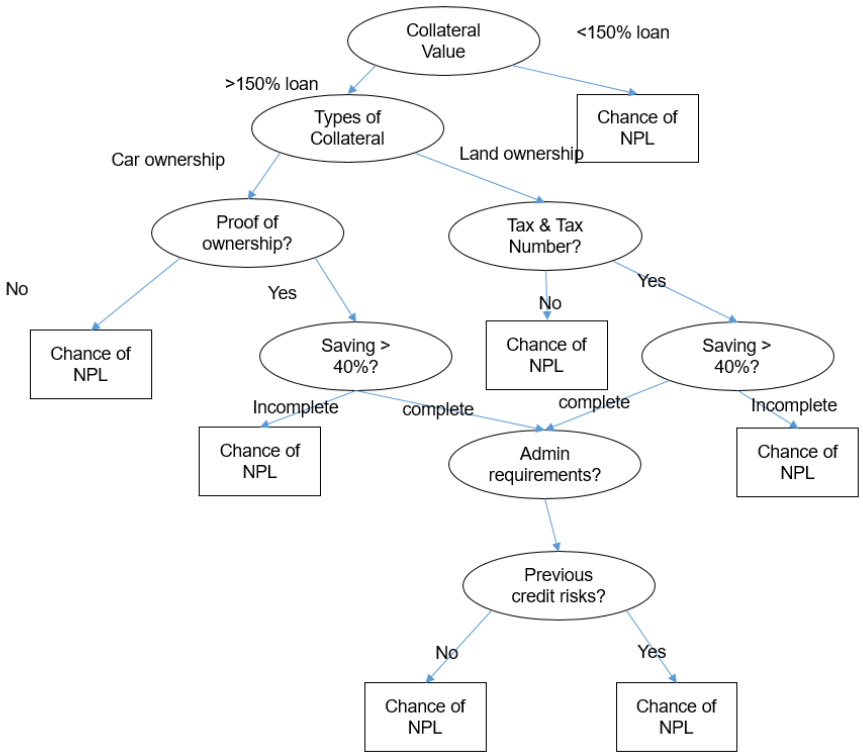
Metodologija istraživanja i implementacije novog modela kreditne procjene sastoji se od 6 koraka i prikazana je na slici 43, a uključuje: fazu razumijevanja poslovanja, fazu razumijevanja podataka, fazu pripreme podataka, fazu modeliranja, fazu evaluacije i fazu primjene. Za potrebe izrade modela procjene kredita koristio se **C5.0** algoritam za konstrukciju stabla odlučivanja.



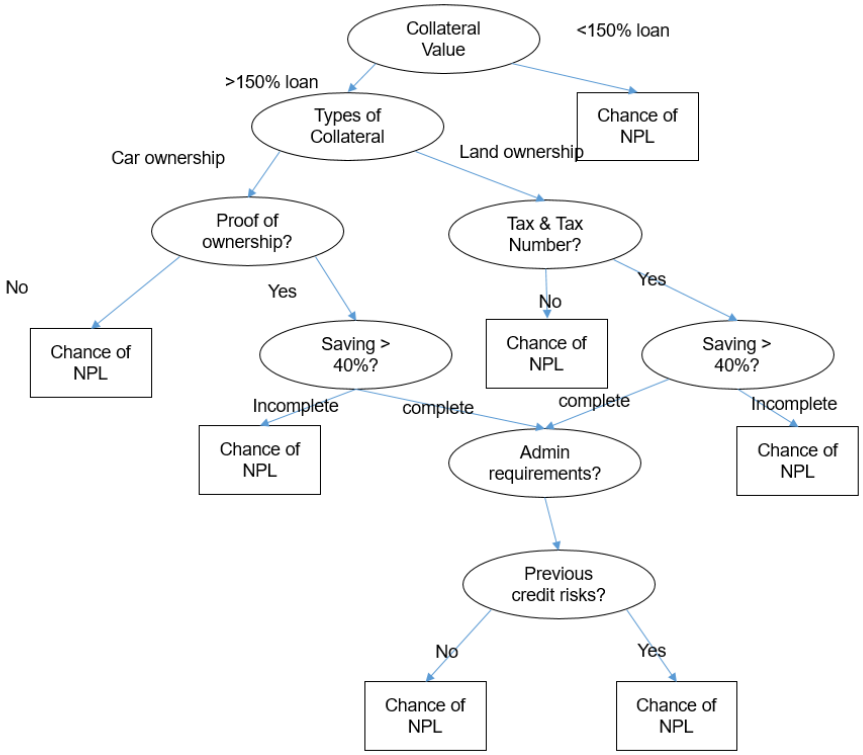
Slika 43: Metodologija implementacije modela stabla odlučivanja za PT BPR X (Izvor: vlastita izrada prema [61, str. 4])

Proces procjene kreditnog rizika u PT BPR X pratio je proceduru slanja zahtjeva, verifikacije podataka, proces procjene kredita i odluke o odobravanju ili odbijanju zahtjeva. Podaci koje PT BPR koristi za procjenu su: spol, dob, iznos kredita, mjesečni prihodi, mjesečni troškovi tekuća mjesečna plaćanja, uštеđevina, vrste kolaterala, vrijednosti kolaterala, period zajma, tip poslovne aktivnosti, izvor financiranja i ocjena prethodnih kredita.Prethodno korišteni model kreditne procjene koristi vrijednost kolaterale kao korijenski čvor, a ocjene prethodnih kredita je terminalni čvor koji služi kao kriterij zaustavljanja. Također prethodni model nije koristio sve atribute koji su prikupljeni u procesu procjene, a prikazan je na slici 44. Nova verzija modela stabla odlučivanja primjenom C5.0 algoritma generirana u fazi modeliranja u kojem su iskorišteni svi atributi prikupljeni u fazi prikupljanja podataka prikazana je na slici 45.

Nakon zamjene starog modela novim modelom procjene kreditnog rizika, procjenjuje se pad stope nenaplativih zajmova - NPL na 3% što dokazuje da novi model procjenjuje kreditni rizik



Slika 44: Prethodni model procjene kreditnog rizika PT BPR X (Izvor: vlastita izrada prema [61, str. 5])



Slika 45: Novi model procjene kreditnog rizika PT BPR X (Izvor: vlastita izrada prema [61, str. 6)

	Trnutni model	%	Predloženi model	%	Test	%
Performing loan	36.996.600	88,01	6.531.700	94,14	10.519.900	96,75
NPL	5.038.800	11,19	2.275.600	5,86	353.700	3,25
Total	42.035.400	100,00	38.807.300	100,00	10.873.600	100,00

Tablica 19: Novi model procjene kreditnog rizika PT BPR X - numeričke vrijednosti izražene u tisućama rupija (Izvor: vlastita izrada prema [61, str. 7])

nekoliko puta bolje od prethodnog modela. Modelom se potvrđuje da je vrijednost kolaterala i dalje najbitniji kriterij procjene te da većina atributa prikupljenih u fazi pripreme nisu relevantni za procjenu rizika kredita. Usporedba novog modela sa starim vidljiva je u tablici 19.

Zaključno, primjenom modela stabla odlučivanja baziranog na C5.0 algoritmu smanjena je stopa nenaplativih zajmova s 11,99% na 3,25% (odnosi se na testni set podataka) te je dokazana korisnost primjene modela u procesu procjene kreditnog rizika u navedenoj ruralnoj banci [60].

9.1.2. Komparativna analiza algoritama stabla odlučivanja za detekciju prijevara putem kreditnih kartica

Anis, Ali i Yadav [61] navode da su slučajevii prijevare putem kartičnih plaćanja dramatično porasli u zadnjih dvadesetak godina zbog pojave e-poslovanja i općenito povećanja korištenja kartičnih plaćanja. Neilson (kao što se citira u [61]) navodi da je broj kreditnih kartica koje cirkuliraju globalno iznosio 2,039.3 milijuna i da su bile korištene u 64.24 bilijuna transakcija. Zbog porasta količine kreditnih kartica i kartičnih plaćanja preko e-plaćanja prevaranti koriste sve sofisticiranije tehnike prijevara putem kartičnih plaćanja. Takve prijevarne aktivnosti predstavljaju jedinstvene i globalne izazove bankarskim i drugim financijskim institucijama koje izdaju kreditne kartice. Roberds (kao što citiraju Anis, Ali i Yadav, [61]) navodi da studija izvršena od strane Američkog Bankarskog Udruženja iz 1996 procjenjuje da bruto gubitak od prijevarnih kartičnih transakcija u SAD-u iznosi 790 milijuna dolara.

Nadalje, prema Anisu, Aliju, Yadavu [61] **detekcija prijevare** kartičnih plaćanja je računalno kompleksan zadatak. Veliki problem u sferi detekcije prijevare putem kartičnih plaćanja predstavljaju neuravnoteženi skupovi podataka (što je specifično za ovaj problem jer najčešće postoji puno veći broj validnih transakcija od prijevarnih). Autori navode da ne postoji jedinstveni klasifikator koji savršeno radi detekciju prijevare jer oni služe za predviđanje potencijalnih prijevarnih transakcija. Iz tog razloga definiraju se osnovne karakteristike klasifikatora za potrebe sistema detekcije prijevare:

- klasifikator mora točno identificirati prijevare (mora imati visoku stopu istinito pozitivnih vrijednosti),
- klasifikator mora detektirati prijevare brzo,
- klasifikator ne smije označavati validne transakcije kao prijevarne (mora imati nisku stopu

lažno pozitivnih vrijednosti).

Za ovo komparativno istraživanje detekcije prijevare autori su iskoristili dva skupa podataka o kreditnim karticama koji su fizički dostupni na repozitoriju Sveučilišta Massachusetts Amherst.

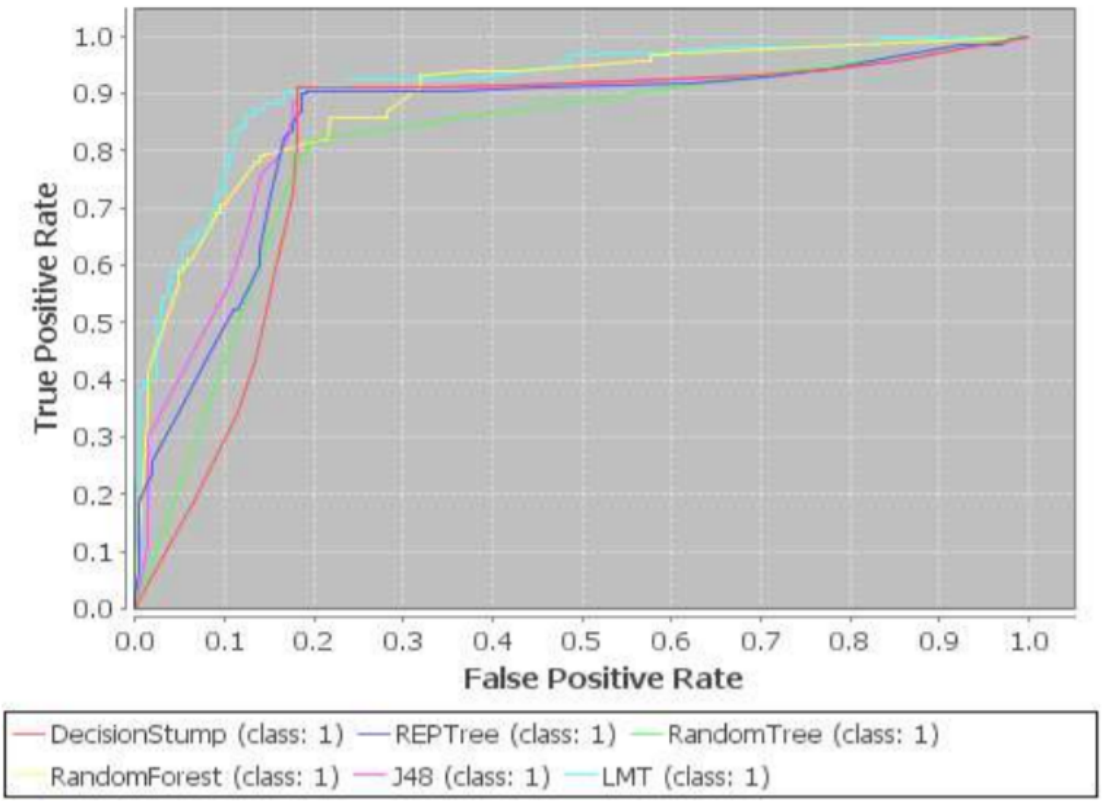
Dataset	Features	Datasize	TrainingSize	TestingSize	Imb ratio
Aus	14	547	366	181	70/30
Gem	21	700	688	332	70/30

Tablica 20: Tablica karakteristika skupova podataka (Izvor: vlastita izrada prema [62, str. 8])

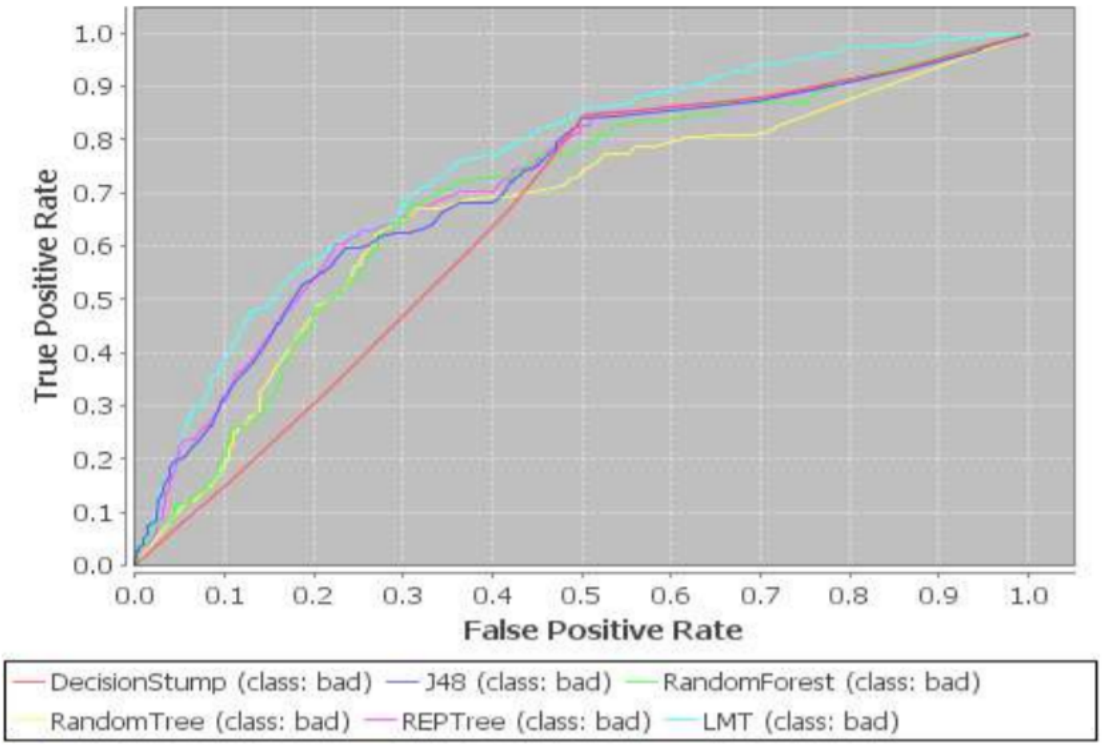
Nadalje, glavni cilj istraživanja bio je pronalazak najboljeg klasifikatora za rad s neuravnoteženim skupovima podataka, nasumičnim odabirom veličine uzoraka i nasumičnim odabirom atributa grananja. Tablica 20 prikazuje značajke svakog pojedinog skupa podataka. Prvi skup podataka predstavlja transakcijske podatke za Australsku banku, dok se drugi odnosi na Njemačku banku.

Za potrebe implementacije modela stabla odlučivanja autori koriste WEKA (engl. *The Waikato Environment for Knowledge Analysis*) alat koji služi za podatkovnu analizu različitih algoritama strojnog učenja, a uključuje i module za pred-procesiranje podataka. Nadalje, autori koriste i uspoređuju performanse šest različitih algoritama stabla odlučivanja: decision stump, random forest, J48, random tree, REP Tree i LMT (engl. *Logistic model tree*). Podatke su iterativno particionirali u različite neuravnotežene omjere: 99:1, 97.5:2.5, 95:5, 90:10, 80:20, 70:30, 60:40 i 50:50. Također skupove podatka su podijelili u **skup podataka za obuku** i **testni skup podataka**. Za potrebe evaluacije performansi klasifikatora koristili su **operativnu krivulju prijemnika** (engl. *Receiver Operating Characteristic Curve, ROC*) i **F1-ocjenu** (referirati se na poglavje 6.1.2.). ROC predstavlja 2D graf koji na x-osi prikazuje istinito pozitivne vrijednosti, dok na y-osi prikazuje lažno pozitivne vrijednosti. Nadalje, ROC predstavlja grafički alat za distinkciju optimalnih od ne optimalnih klasifikatora. Nakon što su skupove podataka podijelili na podatke za obuku i testiranje, konstruirali su modele stabla odlučivanja pomoću skupova podataka za obuku i deseterostruke unakrsne validacije. Slika 46 i 47 predstavljaju ROC graf za svaki od dvaju setova podataka.

Zaključno, komparativnom analizom rezultirajućih modela autori zaključuju da je random forest najbolji klasifikator koji su koristili u istraživanju. Rezultati su pokazali da pri povećanju omjera neuravnoteženosti u podacima random forest i LMT klasifikatori funkcioniraju vrlo dobro što je vidljivo i iz prethodna dva grafa. Osim toga, samo istraživanje je pokazalo kako su varijante modela stabla odlučivanja vrlo efikasan alat u implementaciji sustava detekcije prijevare putem kartičnih plaćanja [61].



Slika 46: ROC graf za Australski set podataka [62, str. 15]



Slika 47: ROC graf za Njemački set podataka [62, str. 15]

9.2. Primjeri iz teorije odlučivanja

9.2.1. Odabir banke prilikom ugovaranja poslovnog kredita

Za potrebe prikaza primjera odabira banke prilikom ugovaranja poslovnog kredita koristi se proizvoljni primjer predstavljen u narednom dijelu teksta.

Prmjer:

IT poduzeće XYZ pokreće novu uslugu za svoje korisnike za koju joj je potreban kredit u iznosu od 1000000 kn. Uprava tvrtke mora donijeti odluku o podizanju kredita od jedne od tri banke koje su u razmatranju. Uvjeti različitih poslovnih banaka su slijedeći.

- *Alternativa 1 - Banka A: 1000000 kn zajma uz fiksnu kamatnu stopu od 5% i vrijeme otplate kredita 10 godina. Naknada za obradu kredita iznosi 50000 kn,*
- *Alternativa 2 - Banka B: 1000000 kn zajma uz progresivnu kamatnu stopu od 4.7% za prvih 5 godina i 5.6 % za narednih 5 godina (vrijeme otplate 10 godina, bez naknade obrade kredita),*
- *Alternativa 3 - Banka C: 100000 kn zajma uz varijabilnu kamatnu stopu s fiksnim dijelom od 3% plus varijabilnim dijelom koji ovisi o situaciji na tržištu (vrijeme otplate kredita 10 godina uz trošak obrade kredita u iznosu od 10000kn),*

Ovnisno o situaciji na tržištu, ukoliko se uprava poduzeća odluči za alternativu 3, odnosno banku C mogući su sljedeći scenariji:

- *povoljno stanje - varijabilni dio kamatne stope 1.5% uz vjerojatnost pojave događaja 0.4,*
- *neutralno stanje - varijabilni dio kamatne stope 3% uz vjerojatnost pojave događaja 0.3,*
- *nepovoljno stanje - varijabilni dio kamatne stope 6% uz vjerojatnost pojave događaja 0.3.*

Ukoliko menadžment tvrtke odustane od podizanja kredita, snosi trošak resursa uloženih u razvoj projektnog plana od 500000 kn, dok je cilj donošenja odluke, odnosno odabira alternativa minimizacija konačnih troškova kamata kredita.

Za potrebe pojednostavljivanja problema odlučivanja iz primjera pretpostaviti ćemo da se radi o otplati kredita na temelju godišnjih kamatnih stopa te da se kredit otplaćuje jednakim otplatnim kvotama.

Otplata kredita jednakim otplatnim kvotama:

Prema Divjak i Erjavecu [62] kod otplate kredita jednakim otplatnim kvotama vrijedi pravilo konstantnih otplatnih kvota. Temeljem toga vrijede sljedeće pretpostavke:

- R_k= konst. = R, dakle vrijedi sljedeća formula (9.1):

$$R = \frac{K}{n} \tag{9.1}$$

- kamate se računaju pomoću ostatka duga prethodnog razdoblja pa imamo:

$$I_k = O_{k-1} \cdot \left(\frac{p}{100}\right) \tag{9.2}$$

  dakle kamata je jednaka otplatnoj kvoti prethodnog razdoblja pomnoženom kamatnom stopom,

- što se tiče anuiteta vrijedi:

$$a_k = R + I_k \tag{9.3}$$

- za ostatak duga vrijedi:

$$O_k = O_{k-1} - R \tag{9.4}$$

- iz navedenih formula sljede formule pomoću kojih je moguće izravno izračunati ostatak duga i anuitet k-tog razdoblja:

$$O_k = K \cdot \left(1 - \frac{k}{n}\right) \tag{9.5}$$

$$a_k = \frac{K}{n}\left((n-k+1) \cdot \frac{p}{100} + 1\right) \tag{9.6}$$

Nadalje, prije konstrukcije logičkog modela stabla odlučivanja potrebno je izračunati troškove kamata za sve moguće alternative, odnosno scenarije

Alternativa 1 - Banka A *Rješenje:*

$$\begin{array}{l} K = 1.000.000,00 \\ n = 10 \\ p = 5 \quad \Rightarrow \quad r = 1,005 \\ \hline a = ? \end{array}$$

Za početak uvrštavamo poćetne podatke u formulu (9.1), odnosno

$$R = \frac{K}{n} = \frac{1000000}{10} = 100000$$

Iznos otplatne kvote je konstantan i on iznosi 1000 kuna. Nadalje, otplatnu tablicu popunjavamo na način da prvo upišemo konstantne otplatne kvote, a zatim računamo a_k, I_k, O_k prema prethodno definiranim formulama. Ako uvrstimo početne vrijednosti K i R u formulu (9.4) dobivamo:

$$O_k = O_{k-1} - R = 1000000 - 100000 = 900000$$

O_1 prema tome iznosi 7000 kn Zatim prema formuli (9.2) računamo kamatu za prvo razdoblje:

$$I_1 = 1000000 \cdot \frac{5}{100} = 50000$$

Vrijednost anuiteta a_1 dobivo na način da konstantnoj otplatnoj kvoti dodamo kamatu za to razdoblje: a_1 = 100000 + 50000 Navedene korake ponavljamo sve dok ne ispunimo otplatnu

tablicu, odnosno sve dok kredit nije otplaćen.

razdbolje k	anuitet a_k	kamate I_k	otplatna kvota R_k	ostatak duga O_k
0.	-	-	-	1000000,00
1.	15000,00	50000,00	100000,00	900000,00
2.	145000,00	45000,00	100000,00	800000,00
3.	140000,00	40000,00	100000,00	700000,00
4.	135000,00	35000,00	100000,00	600000,00
5.	130000,00	30000,00	100000,00	500000,00
6.	125000,00	25000,00	100000,00	400000,00
7.	120000,00	20000,00	100000,00	300000,00
8.	115000,00	15000,00	100000,00	200000,00
9.	110000,00	10000,00	100000,00	100000,00
10.	105000,00	5000,00	100000,00	0,00
Ukupno Kamate:	275000			

Za ostale alternative, odnosno scenarije koristiti će se skraćeni postupak, odnosno samo prikaz računa putem otplatne tablice.

Alternativa 2 - Banka B *Rješenje:*

$$\begin{array}{l} K = 1.000.000,00 \\ n = 10 \\ p = 4.7(5godina) \\ \underline{p = 5.6(5godina)} \\ a = ? \end{array}$$

razdbolje k	anuitet a_k	kamate I_k	otplatna kvota R_k	ostatak duga O_k
0.	-	-	-	1000000,00
1.	14700,00	47000,00	100000,00	900000,00
2.	142300,00	42300,00	100000,00	800000,00
3.	137600,00	37600,00	100000,00	700000,00
4.	132900,00	32900,00	100000,00	600000,00
5.	128200,00	28200,00	100000,00	500000,00
6.	128000,00	28000,00	100000,00	400000,00
9.	122400,00	22400,00	100000,00	300000,00
8.	116800,00	16800,00	100000,00	200000,00
9.	111200,00	11200,00	100000,00	100000,00
10.	105600,00	5600,00	100000,00	0,00
Ukupno Kamate:	272000			

Alternativa 3 - Banka C - Stanje - povoljno *Rješenje:*

$$\begin{array}{l} K = 1.000.000,00 \\ n = 15 \\ p = 3 + 1.5 = 4.5 \\ \hline a = ? \end{array}$$

razdbolje k	anuitet a_k	kamate I_k	otplatna kvota R_k	ostatak duga O_k
0.	-	-	-	1000000,00
1.	14500,00	45000,00	100000,00	900000,00
2.	14050,00	40500,00	100000,00	800000,00
3.	136000,00	36000,00	100000,00	700000,00
4.	131500,00	31500,00	100000,00	600000,00
5.	127000,00	27000,00	100000,00	500000,00
6.	122500,00	22500,00	100000,00	400000,00
7.	118000,00	18000,00	100000,00	300000,00
8.	113500,00	13500,00	100000,00	200000,00
9.	109000,00	9000,00	100000,00	100000,00
10.	104500,00	4500,00	100000,00	0,00
Ukupno Kamate:	247500			

Alternativa 3 - Banka C - Stanje - neutralno

$$\begin{array}{l} K = 1.000.000,00 \\ n = 15 \\ p = 3 + 3 = 6 \\ \hline a = ? \end{array}$$

razdbolje k	anuitet a_k	kamate I_k	otplatna kvota R_k	ostatak duga O_k
0.	-	-	-	1000000,00
1.	160000,00	60000,00	100000,00	900000,00
2.	154000,00	54000,00	100000,00	800000,00
3.	148000,00	48000,00	100000,00	700000,00
4.	142000,00	42000,00	100000,00	600000,00
5.	136000,00	36000,00	100000,00	500000,00
6.	130000,00	30000,00	100000,00	400000,00
7.	124000,00	24000,00	100000,00	300000,00
8.	118000,00	18000,00	100000,00	200000,00
9.	112000,00	12000,00	100000,00	100000,00
10.	106000,00	6000,00	100000,00	0,00
Ukupno Kamate:	330000			

Alternativa 3 - Banka C - Stanje - nepovoljno

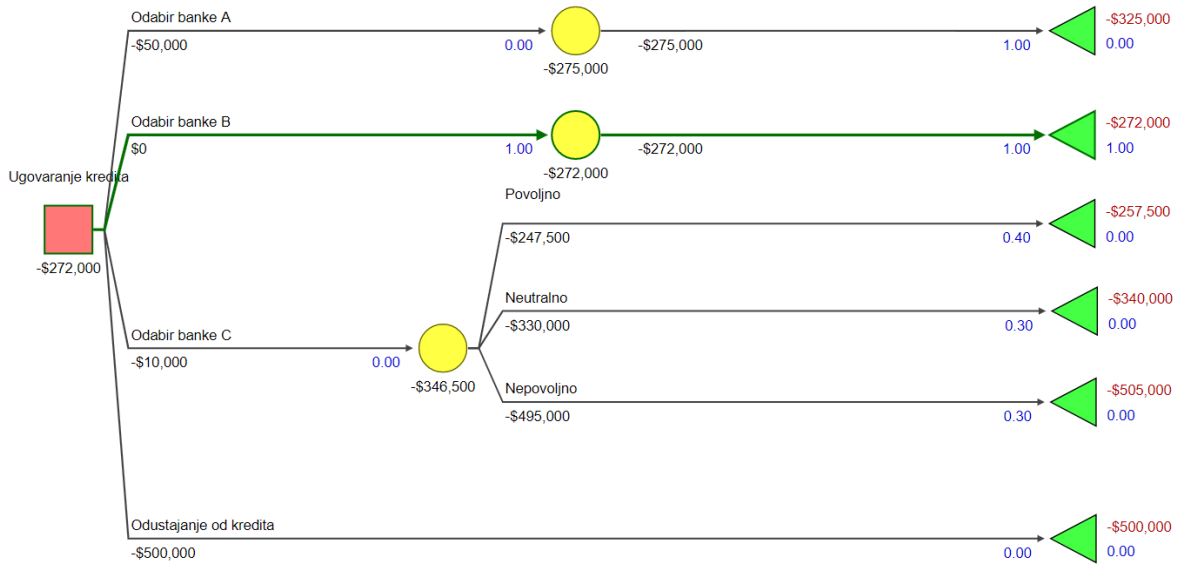
$$
\begin{aligned}
K &= 1.000.000,00 \\
n &= 15 \\
p &= 3 + 6 = 9 \\
\hline
a &= ?
\end{aligned}
$$

razdbolje k	anuitet a_k	kamate I_k	otplatna kvota R_k	ostatak duga O_k
0.	-	-	-	1000000,00
1.	190000,00	90000,00	100000,00	900000,00
2.	181000,00	81000,00	100000,00	800000,00
3.	172000,00	72000,00	100000,00	700000,00
4.	163000,00	63000,00	100000,00	600000,00
5.	154000,00	54000,00	100000,00	500000,00
6.	145000,00	45000,00	100000,00	400000,00
7.	136000,00	36000,00	100000,00	300000,00
8.	127000,00	27000,00	100000,00	200000,00
9.	118000,00	18000,00	100000,00	100000,00
10.	109000,00	9000,00	100000,00	0,00
Ukupno Kamate:	495000			

Izgradnja logičkog modela stabla odlučivanja i donošenje odluke putem SilverDecisions alata:

Nakon što su se izračunali troškovi svake kombinacije odluke i stanja okoline, slijedi konstrukcija logičkog modela stabla odlučivanja uz pomoć SilverDecisions programskog alata. Za početak se dodaje čvor odluke koji sadrži četiri alternativne grane aktivnosti, odnosno: Odabir banke A, odabir banke B, odabir banke C i odustajanje od kredita. Nakon toga čvor posljedica na trećoj grani, odnosno grani alternative C dobiva tri različite grane posljedičnih stanja kao što je navedeno u primjeru te se naposljetku svim granama posljedičnih stanja pridružuju odgovarajući završni čvorovi. Osim toga dodaju se nazivi čvorova i grana, kao i troškovi pojedinih akcija i posljedičnih stanja te njihovih vjerojatnosti. Finalni logički model stabla odlučivanja za problem ugovaranja kredita prikazan je na slici 48.

Iz samog modela prikazanog na slici 48 vidljivo je da su na svim čvorovima posljedica izračunate odgovarajuće očekivane vrijednosti. Početni čvor odluke "Ugovaranje kredita" poprima očekivanu vrijednost "EV" alternative koja predstavlja najmanje kamatne troškove u odnosu na ostale alternative. Prema modelu stabla odlučivanja optimalna odluka je "Odabir banke B" prikazano zelenom bojom na slici 48 te označeno vjerojatnošću "1.00" na grani alternativne aktivnosti. Odluka "Odabir banke B" predstavlja optimalnu odluku u rješavanju ovog problema te ona minimizira troškove kamata koji u ovom slučaju iznose 272000 kn.



Slika 48: Logički model stabla odlučivanja za problem ugovaranja kredita poduzeća XYZ (Izvor: vlastita izrada)

9.2.2. Odabir poduzeća za investiciju od strane fonda rizičnog kapitala

Prmjer: *Fond rizičnog kapitala ABC razmatra ulaganje u jednu od tri IT startup kompanije. Menadžment fonda analizira poslovne planove i dosadašnje poslovanje svake od triju kompaniju. Iznosi ulaganja za svaku pojedinu kompaniju variraju te ovise o grani kojom se kompanija bavi:*

- *kompanija A - digitalna marketing agencija - potrebna investicija od 500000kn,*
- *kompanija B - softverska kompanija za razvoj računalnih igara - potrebna investicija 950000kn,*
- *kompanija C - startup čije se poslovanje temelji na razvoju nove inovativne aplikacije X koja još ne postoji na tržištu - potrebna investicija 1200000kn.*

Ovisno o tržišnoj situaciji i potražnji za uslugama/proizvodima navedenih kompanija moguća su tri različita scenarija u kojima svaka od navedenih firmi ostvaruje različiti prinos, odnosno:

- *situaciija A - vjerojatnost nastupanja 0.35: Marketinška agencija ostvaruje značajan tržišni udio te ostvaruje prinos od 170% od inicijalne investicije za fond rizičnog kapitala; Softverska kompanija nije uspjela razviti niti jedan proizvod s kojim ostvaruju značajne prihode; Aplikacija X kompanije C doživljava podvojeno mišljenje na tržištu te donosi prinos od 120% inicijalne investicije,*
- *situacija B - vjerojatnost nastupanja 0.25: Marketinška agencija posluje zadovoljavajuće i ostvaruje prihod od 120% inicijalne investicije. Kompanija B razvija novu inovnativnu*

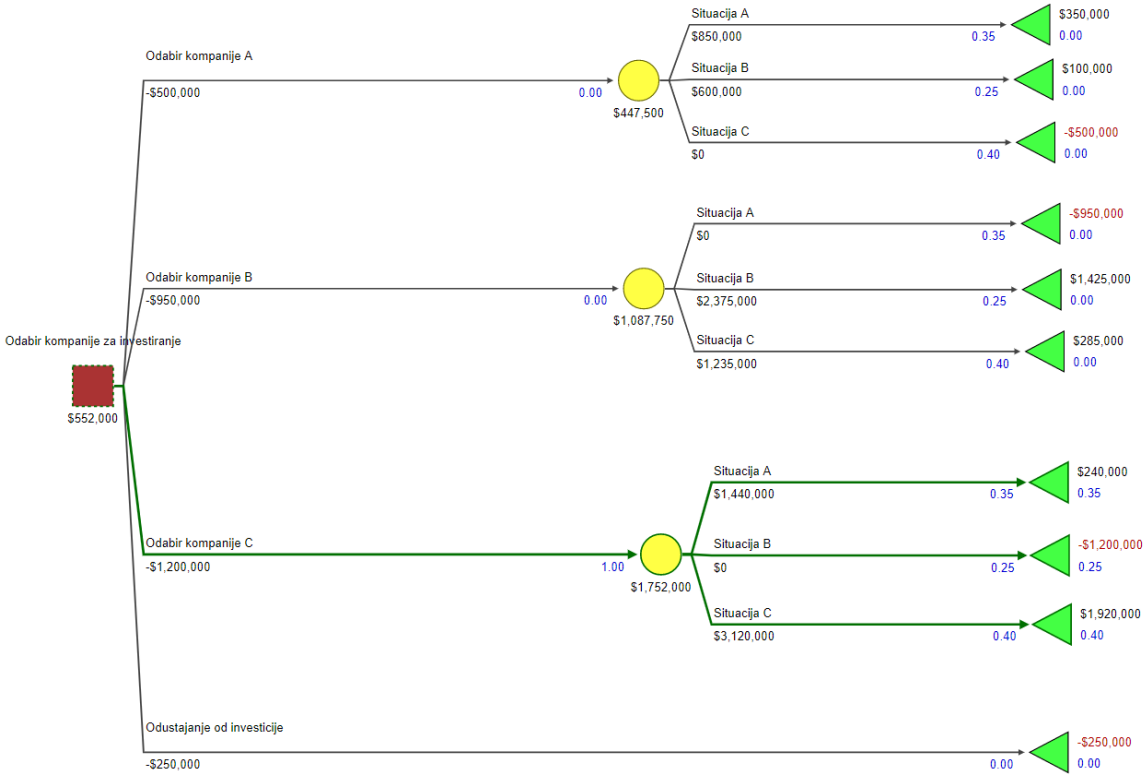
računalnu igru te ostvaruje povrat inicijalne investicije od 250%. Kompanija C ne uspjeva plasirati proizvod na tržište te ne ostvaruje nikakav povrat inicijalnog ulaganja,

- *situacija C - vjerojatnost nastupanja 0.4: Marketinška agencija ne uspjeva zauzeti tržišni udio te ne ostvaruje nikakav prinos. Sofverska kompanija se uspjeva pozicionirati na tržištu te ostvaruje prinos od 130% inicijalne investicije. Aplikacija kompanije C ostvaruje veliki uspjeh te kompanija ostvaruje prinos od 260% inicijalne investicije.*

U slučaju odustajanja od investiranja u bilo koju od tri kompanije, fond rizičnog kapitala gubi inicijalnu investiciju od 250000kn utrošenu u analizu tržišta. Cilj donošenja odluke o investiciji je maksimiziacija povrata uloženih sredstava u odabranu kompaniju.

Izgradnja logičkog modela stabla odlučivanja:

Logički model stabla odlučivaja i provedba metode odrađuje se uz pomoć programskog alata



Slika 49: Logički model stabla odlučivanja za problem investicije fonda rizičnog kapitala ABC (Izvor: vlastita izrada)

SilverDecisions. Kao prvi korak provedbe metode stabla odlučivanja uz pomoć SilverDecisions alata otvara se novo platno za izradu modela. Nakon toga dodaje se prvi čvor odluke "Ugovaranje kredita". Nadalje, dodaju se četiri grane alternativnih aktivnosti, odnosno: "Odabir kompanije A", "Odabir kompanije B", "Odabir kompanije C", "Odustajanje od investicije" te im se pridružuju povezani troškovi. Svakom čvoru posljedica se nakon toga dodaju odgovarajuće grane posljedičnih stanja zajedno s povezanim vjerojatnostima nastupanja događaja, odnosno posljedičnih stanja. Na kraju se svakoj grani posljedičnog stanja dodaje odgovarajući završni čvor kojemu se pridružuje konačna vrijednost novčane isplate ili gubitka. Konačni logički model

stabla odlučivanja za problem investiranja fonda rizičnog kapitala vidljiv je na slici 49.

Izračun očekivanih vrijednosti uz pomoć postupka računanja unatrag:
Iako SilverDecisions programski alat računanje očekivanje vrijednosti i postupak algoritma odrađuje samostalno, za provjeru može se izračunati uz pomoć formule. Dakle računamo očekivanu vrijednost čvorova posljedica uz pomoć formule za računanje očekivane vrijednosti:

$$EV_{i-1} = \sum_j p_j EV_i, i \in \{1, 2, ..., n\}, j \in \{1, 2, ..., m\} \tag{9.7}$$

Odnosno:
$EV_A = 850000 \cdot 0.35 + 600000 \cdot 0.25 + 0 \cdot 0.40 = 447500$
$EV_B = 0 \cdot 0.35 + 2375000 \cdot 0.25 + 1235000 \cdot 0.40 = 1087750$
$EV_C = 1440000 \cdot 0.35 + 0 \cdot 0.25 + 3120000 \cdot 0.40 = 1752000$

Nadalje, u samom SilverDecisions alatu su vidljive očekivane vrijednosti ispod čvorova posljedica te ih alat automatski računa pomoću gore navedene formule za računanje očekivane vrijednosti kao što je prikazano na slici 49.

Pronalaženje optimalnog puta postupkom računanja unaprijed:
Nakon što se izračunaju očekivane vrijednosti za svaki pojedini čvor posljedica, računa se optimalna vrijednost početnog čvora. Optimalna vrijednost početnog čvora računa se postupkom računanja unaprijed, odnosno dodjeljuje mu se očekivana vrijednost grane na optimalnom putu. Ako se pretpostavi da je cilj problema odlučivanja o investiranju u kompaniju bio maksimizacija povratne investicije, lako se može zaključiti koja grana aktivnosti predstavlja optimalan put. Nadalje, za zadani problem odlučivanja optimalna odluka jest investirati u tvrtku C jer ona daje najveću povrat uloženih sredstava. Nakon što se uračunaju troškovi ona iznosi 552000 kn. SilverDecisions programski alat i ovaj korak računa automatski, odnosno označava optimalnu posljedičnu granu zelenom bojom i dodjeljuje joj vrijednost 1.00, dok svim ostalim granama alternativnih aktivnosti pridružuje vrijednost 0 kao što je prikazano na slici 49.

10. Zaključak

Odlučivanje, odnosno poslovno odlučivanje važan je aspekt modernog poslovanja, pa tako i banaka i financijskih institucija. Suvremeno poslovanje i turbulentna poslovna okolina te promijenjivi tržišni uvjeti iziskuju od bankarskih i financijskih institucija, kao i njihovih menadžera konstantno usavršavanje u vidu stilova vođenja i odlučivanja. Suvremeni menadžer mora biti u stanju u najkraćem mogućem roku procijeniti problemsko područje te donijeti optimalnu i pravovremenu odluku. Sam proces donošenja odluke u složenim korporativnim okruženjima koje karakterizira nesigurnost i rizik vrlo je složen te iziskuje veliki angažman od strane menadžmenta.

Kao odgovor na zahtjeve takvih složenih i dinamičnih okruženja razvijene su metode za pomoć kod donošenja odluka u uvjetima nesigurnosti i rizika, a u ovom radu stavlja se naglasak na metodu stabla odlučivanja. Stablo odlučivanja u sferi teorije odlučivanja predstavlja moćan grafičko-vizualizacijski alat pomoću kojeg uprava i menadžment mogu na jednostavan način iskoristiti kod donošenja kompleksnih sekvencijalnih jednokratnih odluka. Sama metoda oslanja se na postupak računanja očekivanih vrijednosti uz pomoć tzv. *rollback* algoritma. Metoda je vrlo fleksibilna te se u bankarskom i financijskom smislu može koristiti za razne svrhe; npr. kod procjene kreditne sposobnosti klijenta, prilikom ugovaranja kreditnoga osiguranja, prilikom procjene isplativosti investicije te kod niza drugih situacija.

Nadalje, implementacija sustava temeljenih na strojnom učenju pokazala se kao važan aspekt suvremenog poslovanja većine poslovnih sustava, pa tako i institucija bankarskog sektora. Strojno učenje omogućuje računalnim sustavima obavljanje kompleksnih zadataka bez da su eksplicitno programirani za tu namjenu. Glavni oblici sustava strojnog učenja su nadzirano i nenadzirano učenja, a odabir pristupa ovisi o specifičnom problemu koji se želi riješiti njihovom implementacijom.

Postoji niz slučajeva korištenja strojnog učenja u bankarskom sektoru, a kao najznačajniji slučajevi pokazali su se detekcija prijevara i procjena kreditnog rizika. Metodologija strojnog učenja zajedno s njezinim matematičkim prediktivnim modelima poput stabla odlučivanja, nasumične šume i pojačavanja predstavljaju paket alata koji poslovnim bankama i drugim institucijama omogućuju izradu kompleknsih i efikasnih sustava za potporu odlučivanju.

Stablo odlučivanja i povezane tehnike u sferi strojnog učenja pokazale su se kao ključni alat u izradi prediktivnih modela i u širem smislu sustava za potporu odlučivanja u bankarstvu. Stablo odlučivanja je prediktivni model strojnog učenja koji se može primijeniti na klasifikacijske i regresijske probleme, a karakterizira ga hijerarhijski pristup i pristup temeljen na rekurzivnom binarnom grananju. Ukoliko su performanse stabla odlučivanja nedostatne, za rješavanje određenog problema, koriste se napredne tehnike agregiranja stabla odlučivanja, odnosno tehnike nasumične šume i pojačavanje. Oba pristupa u svojoj srži sadrže koncept stabla odlučivanja, a glavna razlika je što za razliku od pojedinačnog stabla koriste agregacije stabala u kreiranju prediktivnih modela.

Konkretnim praktičnim primjerom izrade sustava za detekciju prijevare temeljenog na

sintetičkom skupu podataka pokazala se korist od izrade prediktivnih modela kao pomoć u detektiranju prijevarnih transakcija u nekom financijskom sustavu. Za potrebe demonstracije performansi koristili su se različiti modeli: DecisionTreeClassifier, RandomForestClassifier i XGBClassifier. Na temelju korištenih metrika: preciznost, opoziv, f-1 ocjena, matrica zabune i AUC pokazalo se kako XGBClassifier koji se bazira na konceptu pojačavanja stabala daje najbolje performanse kod problema detekcije prijevarnih transakcija u nekom financijskom sustavu.

Zadnji dio rada prikazuje niz primjera upotrebe metode stabla odlučivanja u praksi, odnosno realnom sektoru. Prva dva obrađena primjera obuhvaćaju primjenu stabla odlučivanja iz sfera strojnog učenja u svrhu procjene kreditnog rizika i detekcije prijevare. Druga dva primjera odnose se na primjenu metode stabla odlučivanja iz teorije odlučivanja te se riješava problem odlučivanja podizanja kredita u poslovnoj banci i problem odabira tvrtke za ulaganje od strane fonda rizičnog kapitala. Razlike metode stabla odlučivanja iz područja strojnog učenja i teorije odlučivanja su sljedeće:

- u teoriji odlučivanja stablo odlučivanja se koristi kao grafički vizualizacijski alat, dok se u strojnom učenju koristi kao prediktivni model,
- u teoriji odlučivanja stablo odlučivanja koristi se kod riješavanja problema odlučivanja kod donošenja kompleksnih jednokratnih sekvencijalnih odluka, dok se u strojnom učenja pomoću njega riješavaju klasifikacijski i regresijski problemi nad velikim količinama podataka,
- u teoriji odlučivanja se prilikom konstrukcije logičkog modela stabla koristi dodatni čvor koji se naziva čvor odluke te ovisi o odluci koju donosi sam odlučitelj,
- u strojnom učenju nema smisla vizualno prikazivati modele stabla zbog velikih količina podataka i veličine stabla, dok je u teoriji odlučivanja to primarna funkcija metode,
- algoritmi koji se koriste za potrebe grananja u modelima stabla odlučivanja iz strojnog učenja najčešće koriste samo binarna grananja zbog računalne efikasnosti, dok kod stabla odlučivanja iz područje teorije odlučivanja često postoje n-arna grananja, ovisno o broju alternativa i potencijalnih stanja okoline.

Sličnosti ove dvije metode naziru se u činjenici da imaju iste korijene te da koriste sličnu terminologiju i grafičke prikaze, dok im se uvelike razlikuje primjene.

Zaključno, kako bi bankarski sektor i financijske institucije mogli držati korak s suvremenim tehnologijama i metodologijama rada, potrebna im je prilagodba te potpora u odlučivanju koja se temelji na suvremenim prediktivnim modelima strojnog učenja, kao i grafičko-vizualnim modelima stabla odlučivanja iz sfere teorije odlučivanja.

Popis literature

[1] P. Sikavica, T. Hunjak, N. Begičević Ređep i T. Hernaus, *Poslovno odlučivanje*. Školska knjiga d.d., Zagreb, 2014.

[2] D. Foote Keith, *A Brief History of Machine Learning*, en-US, ožujak 2019. adresa: https://www.dataversity.net/a-brief-history-of-machine-learning/ (pogledano 8. 2. 2020).

[3] P. Murphy Kevin, *Machine learning: a probabilistic perspective*, serija Adaptive computation and machine learning series. Cambridge, MA: MIT Press, 2012, ISBN: 978-0-262-01802-9.

[4] K. Graham, H. Rajaona i J. Moolenaar, *Guidebook; Fraud and anomaly detection in banking*, en. 2019. adresa: https://pages.dataiku.com/fraud-detection-banking (pogledano 23. 2. 2020).

[5] S. Chand, *Decisions Making: Strategic, Tactical and Operational Decisions | Business Management*, en-US, Library Catalog: www.yourarticlelibrary.com, rujan 2013. adresa: https://www.yourarticlelibrary.com/information-technology/decisions-making-strategic-tactical-and-operational-decisions-business-management/10271 (pogledano 28. 6. 2020).

[6] *Understanding business - Higher Business management Revision*, en-GB, Library Catalog: www.bbc.co.uk. adresa: https://www.bbc.co.uk/bitesize/topics/z9mngk7 (pogledano 28. 6. 2020).

[7] M. Clayton, *Robert Tannenbaum & Warren Schmidt: Leadership Continuum*, en-GB, Library Catalog: www.pocketbook.co.uk Section: Decision Making, svibanj 2017. adresa: https://www.pocketbook.co.uk/blog/2017/05/02/robert-tannenbaum-warren-schmidt-leadership-continuum/ (pogledano 28. 6. 2020).

[8] J. Gibson, J. Ivancevich i R. Konopaske, *Organizations: Behavior, Structure, Processes*, 14. izdanje. McGraw-Hill/Irwin, 2011, ISBN: 0078112664,9780078112669.

[9] S. P. Robbins i T. A. Judge, *Essentials of Organizational Behavior*, 12. izdanje. Pearson, 2013, ISBN: 0132968509,9780132968508.

[10] iEduNote.com, *Individual Decision Making Styles*, en-US, Library Catalog: www.iedunote.com, ožujak 2019. adresa: https://www.iedunote.com/individual-decision-making-styles (pogledano 1. 7. 2020).

[11] S. P. Robbins i M. Coulter, *Management, 11th Edition*, 11th. Prentice Hall, 2011, ISBN: 0132163845,9780132163842.

[12] P. Sikavica i F. Bahtijarević-Šiber, *Menadžment: Teorija menadžmenta i veliko empirijsko istraživanje u Hrvatskoj*. Masmedia d.o.o., 2004, 2004, ISBN: 9531574553, 9789531574556.

[13] S. Dawson, *Analysing Organisations*. Macmillan Education UK, 1996, ISBN: 978-0-333-66095-9,978-1-349-24846-9.

[14] M. Kurhade i R. Wankhade, „An Overview on Decision Making Under Risk and Uncertainty", en, sv. 5, br. 4, str. 7, 2013.

[15] R. Olivas, *Decision trees - A Primer for Decision-making Proffesionals*, en. 2007, str. 32.

[16] N. Begičević Ređep, *Uvod u stabla odlučivanja*, 2017.

[17] T. Software, *TreePlan Decision Tree Excel Add-in • TreePlan Software*, en-US. adresa: https://treeplan.com/ (pogledano 15. 8. 2020).

[18] Palisade, *PrecisionTree – Decision trees for Microsoft Excel*, en, 2020. adresa: http://www.palisade.com/precisiontree/default.asp (pogledano 15. 8. 2020).

[19] SilverDecisions, *SilverDecisions*, 2020. adresa: http://silverdecisions.pl/ (pogledano 15. 8. 2020).

[20] *SilverDecisions/SilverDecisions*, en. adresa: https://github.com/SilverDecisions/SilverDecisions (pogledano 15. 8. 2020).

[21] M. Bishop Christopher, *Pattern recognition and machine learning*, serija Information science and statistics. New York: Springer, 2006, ISBN: 978-0-387-31073-2.

[22] R. Glasbergen, *A Brief History of Machine Learning*, en-US, lipanj 2017. adresa: https://provalisresearch.com/blog/brief-history-machine-learning/ (pogledano 8. 2. 2020).

[23] Wikipedia, *Machine learning*, en, Page Version ID: 939563848, veljača 2020. adresa: https://en.wikipedia.org/w/index.php?title=Machine_learning&oldid=939563848 (pogledano 8. 2. 2020).

[24] E. Alpaydin, *Introduction to machine learning*, serija Adaptive computation and machine learning. Cambridge, Mass: MIT Press, 2004, OCLC: ocm56830710, ISBN: 978-0-262-01211-9.

[25] D. Silver, *Machine Learning branches venn diagram by David Silver in 2020 | Machine learning, Data science, Diagram*, en, 2020. adresa: https://www.pinterest.com/pin/219761656794012482/ (pogledano 9. 2. 2020).

[26] J. Portilla, *Machine learnning overwiew*, en, 2016. adresa: https://www.udemy.com/course/python-for-data-science-and-machine-learning-bootcamp (pogledano 16. 2. 2020).

[27] M. Velic, *Uvod_ML*, 2015. adresa: https://www.slideshare.net/MarkoVelic1/uvodml (pogledano 23. 2. 2020).

[28] T. Fawcett, „An introduction to ROC analysis”, en, *Pattern Recognition Letters*, sv. 27, br. 8, str. 861–874, lipanj 2006, ISSN: 01678655. DOI: 10.1016/j.patrec.2005.10.010. adresa: https://linkinghub.elsevier.com/retrieve/pii/S016786550500303X (pogledano 23. 2. 2020).

[29] G. James, D. Witten, T. Hastie i R. Tibshirani, *An introduction to statistical learning: with applications in R*, eng, Corrected at 8th printing, serija Springer texts in statistics. New York Heidelberg Dordrecht London: Springer, 2017, OCLC: 1005701164, ISBN: 978-1-4614-7137-0 978-1-4614-7138-7.

[30] AltexSoft, *Fraud Detection: How Machine Learning Systems Help Reveal Scams in Fintech, Healthcare, and eCommerce*, en-US, Library Catalog: www.altexsoft.com. adresa: https://www.altexsoft.com/whitepapers/fraud-detection-how-machine-learning-systems-help-reveal-scams-in-fintech-healthcare-and-ecommerce/ (pogledano 31. 3. 2020).

[31] M. Rouse, *What is fraud detection? A definition from WhatIs.com*, en, 2019. adresa: https://searchsecurity.techtarget.com/definition/fraud-detection (pogledano 2. 3. 2020).

[32] A. Oleksiuk, *5 Uses of Machine Learning in Finance and FinTech | Intellias Blog*, en-US, Library Catalog: www.intellias.com Section: Blog post, siječanj 2019. adresa: https://www.intellias.com/5-use-cases-of-machine-learning-in-fintech-and-banking/ (pogledano 31. 3. 2020).

[33] J. Moolenaar, H. Rughani i P. Ménard, *White Paper: AI in Banking*, de-de, Library Catalog: pages.dataiku.com, 2019. adresa: https://pages.dataiku.com/ai-in-banking (pogledano 2. 4. 2020).

[34] L. Rokach i O. Maimon, *Data mining with decision trees: theory and applications*, en, Second edition. Hackensack, New Jersey: World Scientific, 2015, ISBN: 978-981-4590-07-5.

[35] J. Kozak, *Decision Tree and Ensemble Learning Based on Ant Colony Optimization*, en, serija Studies in Computational Intelligence. Cham: Springer International Publishing, 2019, sv. 781, ISBN: 978-3-319-93751-9 978-3-319-93752-6. DOI: 10.1007/978-3-319-93752-6. adresa: http://link.springer.com/10.1007/978-3-319-93752-6 (pogledano 6. 5. 2020).

[36] P.-N. Tan, M. Steinbach i V. Kumar, *Introduction To Data Mining*. Addisson-Wesley New York, 2005.

[37] L. Breiman, J. H. Friedman, R. . Olshen i C. . Stone, *Classification And Regression Trees*, English. CHAPMAN & HALL/CRC, 1984, ISBN: 0-412-04841-8.

[38] G. James, D. Witten, T. Hastie i R. Tibshirani, *An Introduction to Statistical Learning*, en, serija Springer Texts in Statistics. New York, NY: Springer New York, 2013, sv. 103, ISBN: 978-1-4614-7137-0 978-1-4614-7138-7. DOI: 10.1007/978-1-4614-7138-7. adresa: http://link.springer.com/10.1007/978-1-4614-7138-7 (pogledano 17. 4. 2020).

[39] H. Jiawei, K. Micheline i P. Jian, *Data mining. Concepts and Techniques, 3rd Edition*, en, Third edition. Morgan Kaufmann, 2012, ISBN: 978-012-3814-79-1.

[40] H. Ishwaran i J. S. Rao, „Decision Trees, Advanced Techniques in Constructing", en, *Encyclopedia of Medical Decision Making*, 2455 Teller Road, Thousand Oaks California 91320 United States: SAGE Publications, Inc., 2009, ISBN: 978-1-4129-5372-6 978-1-4129-7198-0. DOI: 10.4135/9781412971980.n98. adresa: http://sk.sagepub.com/reference/medical/n98.xml (pogledano 1. 6. 2020).

[41] J. H. Friedman, T. Hastie i R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, English. Springer New York, 2009.

[42] *Synthetic Financial Datasets For Fraud Detection*, en, Library Catalog: www.kaggle.com. adresa: https://kaggle.com/ntnu-testimon/paysim1 (pogledano 3. 6. 2020).

[43] G. Van Rossum i F. L. Drake Jr, *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[44] W. McKinney, „Data Structures for Statistical Computing in Python", *Proceedings of the 9th Python in Science Conference*, S. van der Walt i J. Millman, ur., 2010, str. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[45] T. pandas development team, *pandas-dev/pandas: Pandas*, verzija 0.23.4, veljača 2020. DOI: 10.5281/zenodo.3715232. adresa: https://doi.org/10.5281/zenodo.3509134.

[46] J. D. Hunter, „Matplotlib: A 2D graphics environment", *Computing in Science & Engineering*, sv. 9, br. 3, str. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

[47] T. E. Oliphant, *A guide to NumPy*. Trelgol Publishing USA, 2006, sv. 1.

[48] S. Van Der Walt, S. C. Colbert i G. Varoquaux, „The NumPy array: a structure for efficient numerical computation", *Computing in Science & Engineering*, sv. 13, br. 2, str. 22, 2011.

[49] M. Waskom, O. Botvinnik, P. Hobson, J. B. Cole, Y. Halchenko, S. Hoyer, A. Miles, T. Augspurger, T. Yarkoni, T. Megies, L. P. Coelho, D. Wehner, cynddl, E. Ziegler, diego0020, Y. V. Zaytsev, T. Hoppe, S. Seabold, P. Cloud, M. Koskinen, K. Meyer, A. Qalieh i D. Allan, *seaborn: v0.5.0 (November 2014)*, verzija v0.5.0, studeni 2014. DOI: 10.5281/zenodo.12710. adresa: https://doi.org/10.5281/zenodo.12710.

[50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot i E. Duchesnay, „Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, sv. 12, str. 2825–2830, 2011.

[51] T. Chen i C. Guestrin, „XGBoost: A Scalable Tree Boosting System", *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, serija KDD '16, San Francisco, California, USA: ACM, 2016, str. 785–794, ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. adresa: http://doi.acm.org/10.1145/2939672.2939785.

[52] G. Lemaître, F. Nogueira i C. K. Aridas, „Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning”, *Journal of Machine Learning Research*, sv. 18, br. 17, str. 1–5, 2017. adresa: http://jmlr.org/papers/v18/16-365.html.

[53] S. Solutions, *Correlation*, Library Catalog: www.statisticssolutions.com Section: correlation, svibanj 2009. adresa: https://www.statisticssolutions.com/statistics-correlation/ (pogledano 4. 6. 2020).

[54] C. Dingwall Nick Potts, *Are categorical variables getting lost in your random forests?*, en, 2016. adresa: https://roamanalytics.com/2016/10/28/are-categorical-variables-getting-lost-in-your-random-forests/ (pogledano 14. 6. 2020).

[55] *Dummy Variable: Definition*. adresa: https://stattrek.com/statistics/dictionary.aspx?definition=dummy-variable (pogledano 5. 6. 2020).

[56] *6.3. Preprocessing data — scikit-learn 0.23.1 documentation*. adresa: https://scikit-learn.org/stable/modules/preprocessing.html (pogledano 5. 6. 2020).

[57] *sklearn.preprocessing.MinMaxScaler — scikit-learn 0.23.1 documentation*. adresa: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html (pogledano 5. 6. 2020).

[58] *What is over sampling and under sampling? - Definition from WhatIs.com*, en, Library Catalog: whatis.techtarget.com. adresa: https://whatis.techtarget.com/definition/over-sampling-and-under-sampling (pogledano 5. 6. 2020).

[59] S. Narkhede, *Understanding AUC - ROC Curve*, en, Library Catalog: towardsdatascience.com, svibanj 2019. adresa: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 (pogledano 14. 6. 2020).

[60] I. G. N. N. Mandala, C. B. Nawangpalupi i F. R. Praktikto, „Assessing Credit Risk: An Application of Data Mining in a Rural Bank”, en, *Procedia Economics and Finance*, sv. 4, str. 406–412, 2012, ISSN: 22125671. DOI: 10.1016/S2212-5671(12)00355-3. adresa: https://linkinghub.elsevier.com/retrieve/pii/S2212567112003553 (pogledano 2. 6. 2020).

[61] M. Anis, M. Ali i A. Yadav, „A COMPARATIVE STUDY OF DECISION TREE ALGORITHMS FOR CLASS IMBALANCED LEARNING IN CREDIT CARD FRAUD DETECTION”, en, str. 18, 2015.

[62] B. Divjak i Z. Erjavec, *Financijska matematika*. TIVA - Tiskara Varaždin : Fakultet organizacije i informatike, 2007., 2007.

Popis slika

1. Vrste odluka s obzirom na važnost u organizaciji (Izvor: vlastita izrada prema [6]) 6
2. Kontinnum liderskog ponašanja (Izvor: vlastita izrada prema [7]) 7
3. Vroom-Yeton-Jagov model vodstva/odlučivanja (Izvor: vlastita izrada prema [8], str. 348) 10
4. Stablo odlučivanje za problem lansiranja proizvoda (Izvor: vlastita izrada) 30
5. Stablo odlučivanje za problem lansiranja proizvoda s istraživanjem tržišta (Izvor: vlastita izrada) 31
6. Primjer matrice rizika (Izvor: vlastita izrada) 32
7. Jednostavno stablo odlučivanja s jednim čvorom odluke. (Izvor: vlastita izrada) . 35
8. Stablo odlučivanja izrađeno pomoću TreePlan alata [17] 38
9. Stablo odlučivanja kreirano pomoću PrecisionTree alata [18] 39
10. Stablo odlučivanja kreirano pomoću SilverDecisions alata [20] 41
11. Početno sučelje SilverDecisions alata i definiranje inicijalnog čovra odluke. (Izvor: vlastita izrada) 42
12. Dodavanje novih čvorova odluke i krajnjeg čvora (Izvor: vlastita izrada) 43
13. Dodavanje završnih čvorova za svako od posljedičnih stanja posljedičnih čvorova (Izvor: vlastita izrada) 43
14. Određivanje vrijednosti za vjerojatnosti, plaćanja i nazive posljedičnih grana (Izvor: vlastita izrada) 44
15. Računanje očekivane vrijednosti uz pomoć SilverDecisions alata (Izvor: vlastita izrada) 45
16. Pronalaženje optimalnog puta uz pomoć SilverDecisions alata (Izvor: vlastita izrada) 45
17. Pohrana modela stabla odlučivanja uz pomoć SilverDecisions alata (Izvor: vlastita izrada) 46

18. Vennov dijagram strojnog učenja (Izvor: vlastita izrada prema [25]) 49
19. Ilustracija primjera nadziranog učenja [3, str. 3] 50
20. Prikaz (a) linearne i (b) polinomijalne regresije [3, str. 9] 51
21. Prikaz (a) visine i težine nekih ljudi i (b) moguće klasteriranje s K = 2 klastera [3, str. 10] 54
22. Prikaz (a) 25 nasumično odabranih 64x64 pikselskih slika (b) Prosječni vektor i prva tri vektora komponentnih osnova (Eigenfaces) [3, str. 12] 56
23. Metodologija nadziranog učenja (Izvor: vlastita izrada prema [26]) 57
24. Metodologija nenadziranog učenja (Izvor: vlastita izrada prema [26]) 57
25. Proces detekcije prijevara (Izvor: vlastita izrada prema [32]) 61
26. Kategorije slučaja korištenja strojnog učenja i umjetne inteligencije u bankarstvu (Izvor: vlastita izrada prema [33, str. 14] 62
27. Konceptualni model stabla odlučivanja (Izvor: vlastita izrada prema [35, str. 4] . . 65
28. Primjena Huntovog algoritma na skupu podataka sa tablice 13 (Izvor: vlastita izrada prema [36, str. 154]) 67
29. Lijevo:Primjer dijeljenja 2-d prostora pomoću binarnog grananja s 5 regija, Desno: prikaz pomoću stabla odlučivanja (Izvor: vlastita izrada prema [38, str.308]) 70
30. Proces grananja stabla (Izvor: vlastita izrada prema [37, str. 28] 71
31. Primjer stabla odlučivanja prije i poslije procesa podrezivanja (Izvor: vlastita izrada prema [39, str. 345]) 75
32. Prvih deset zapisa iz Paysim skupa (Izvor: vlastita izrada) 81
33. Prikaz agregiranih numeričkih podataka iz skupa (Izvor: vlastita izrada) 83
34. Graf prikaza zbroja validnih u odnosu na prijevarne transakcije (Izvor: vlastita izrada) 83
35. Matrica korelacije atributa (Izvor: vlastita izrada) 84
36. Stupčasti graf korelacija atributa u odnosu na "isFraud" atribut (Izvor: vlastita izrada) 85
37. Graf odnosa "isFraud" i "amount" atributa (Izvor: vlastita izrada) 86
38. Tablični prikaz odnosa "isFraud" i "amount" atributa (Izvor: vlastita izrada) 86
39. Prvih pet zapisa skupa podataka nakon zamijene kategoričkih varijabli (Izvor: vlastita izrada) 87
40. Prikaz mjera performansi - sklearn - DecisionTreeClassifier (Izvor: vlastita izrada) 91
41. Prikaz mjera performansi - sklearn - RandomForestClassifier (Izvor: vlastita izrada) 92

42. Prikaz mjera performansi - xgboost.sklearn - XGBClassifier (Izvor: vlastita izrada) 92

43. Metodologija implementacije modela stabla odlučivanja za PT BPR X (Izvor: vlastita izrada prema [61, str. 4]) 96

44. Prethodni model procjene kreditnog rizika PT BPR X (Izvor: vlastita izrada prema [61, str. 5]) 97

45. Novi model procjene kreditnog rizika PT BPR X (Izvor: vlastita izrada prema [61, str. 6) 97

46. ROC graf za Australski set podataka [62, str. 15] 100

47. ROC graf za Njemački set podataka [62, str. 15] 100

48. Logički model stabla odlučivanja za problem ugovaranja kredita poduzeća XYZ (Izvor: vlastita izrada) 106

49. Logički model stabla odlučivanja za problem investicije fonda rizičnog kapitala ABC (Izvor: vlastita izrada) 107

Popis tablica

1. Matrica plaćanja (Izvor: vlastita izrada prema [4, str. 463]) 23
2. Matrica plaćanja - zarada od prodaje novina ovisno o potražnji (Izvor: vlastita izrada prema [4, str. 465]) 24
3. Odluka - maksimin (Izvor: vlastita izrada) 24
4. Odluka - maksimaks (Izvor: vlastita izrada) 25
5. Odluka - Hurwitz (Izvor: vlastita izrada) 25
6. Računanje maksimalnog ishoda za svaku kombinaciju akcije i stanja okoline (Izvor: vlastita izrada) 26
7. Tablica žaljenja i izračun ρ_i (Izvor: vlastita izrada) 27
8. Primjena kriterija očekivane vrijednosti na problem prodaje novina (Izvor: vlastita izrada) 27
9. Primjena Laplaceovog kriterija na problem prodaje novina (Izvor: vlastita izrada) 28
10. Tablica odlučivanja za problem lansiranja novog proizvoda (Izvor: vlastita izrada) 29
11. Matrica zabune (Izvor: vlastita izrada prema [27]) 52
12. Razlike pristupa detekciji prijevara (Izvor: vlastita izrada prema [30]) 58
13. Skup podataka o klasifikaciji kredita (Izvor: vlastita izrada prema [36, str. 153]) . 67
14. Usporedba metrike - preciznost (engl. Precision) 93
15. Usporedba metrike - opoziv (engl. Recall) 93
16. Usporedba metrike - f-1 ocjena (engl. F-1 score) 93
17. Usporedba metrike - matrica zabune (confusion matrix) 94
18. Usporedba metrike - AUC (engl. area under curve) 94
19. Novi model procjene kreditnog rizika PT BPR X - numeričke vrijednosti izražene u tisućama rupija (Izvor: vlastita izrada prema [61, str. 7]) 98
20. Tablica karakteristika skupova podataka (Izvor: vlastita izrada prema [62, str. 8]) 99