Predikcija depresije temeljem objava na društvenim mrežama

Tenšić, Monika

Master's thesis / Diplomski rad

2019

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:211:252678

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

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

UNIVERSITY OF ZAGREB
FACULTY OF ORGANIZATION AND INFORMATICS
VARAŽDIN

Monika Tenšić

PREDICTION OF DEPRESSION USING SOCIAL MEDIA POSTS

MASTERS THESIS

Varaždin, 2019.

UNIVERSITY OF ZAGREB

FACULTY OF ORGANIZATION AND INFORMATICS

V A R A Ž D I N

Monika Tenšić

JMBAG: 1191231563

Study programme: Information and Software Engineering

PREDICTION OF DEPRESSION USING SOCIAL MEDIA POSTS

MASTERS THESIS

Mentor:

Doc. dr. sc. Dijana Oreški

Varaždin, july 2019.

Monika Tenšić

Statement of Authenticity

I declare that my masters thesis is the original result of my work and that I did not use any sources other than those mentioned in it. Ethically appropriate and acceptable work methods and techniques were used to create the thesis.

The author confirmed by accepting the provisions in the FOI-radovi

---

Summary

In this thesis, I use machine learning research for identifying signs of depression in social media posts. Depression and mental health problems, according to numerous studies, are evident in posts on social networks, which means we can use them to predict depression in individuals. In order to choose which type of algorithm is the best one for this application, comparison of different types of model training algorithms has been made. Data models used for training purposes were collected on online databases, created by other individuals. For algorithm development I used Python, and NTLK platform for training, testing and saving the model. Naive Bayes is used to create a classifier, which can detect signs of depression in the text by word analysis.

Key words: depression; machine learning; neural networks; artificial neural network; deep learning; supervised learning; unsupervised learning; semi-supervised learning; reinforcement learning; feature learning; sparse dictionary learning; anomaly detection; association rule learning; decision tree learning; support vector machines; bayesian networks; genetic algorithms; social media; mental health; prediction; java; python; lisp; frameworks; libraries; javascript

Table of contents

1. Introduction 1

2. Depression and social media 2
2.1. Mental health 2
2.1.1. Depression 2
2.1.2. Suicide 4
2.2. Social media 5
2.2.1. Positive social media influence 6
2.2.2. Negative social media influence 6
2.3. Social media and depression 7

3. Machine learning 8
3.1. What is machine learning? 8
3.2. Machine learning methods 8
3.2.1. Supervised learning 9
3.2.2. Unsupervised learning 11
3.2.3. Semi-supervised learning 13
3.2.4. Reinforcement learning 15
3.2.5. Feature learning 16
3.2.6. Sparse dictionary learning 17
3.2.7. Anomaly detection 18
3.2.8. Association rule learning 19
3.3. Machine learning models 20
3.3.1. Artificial neural network 20
3.3.2. Decision tree learning 21
3.3.3. Support vector machines 22
3.3.4. Bayesian networks 23
3.3.5. Genetic algorithms 24
3.4. Usage and technologies 25
3.4.1. Usage 25
3.4.2. Technologies 26

4. Prediction of depression using machine learning algorithms 30
4.1. The idea 30
4.2. The data 31
4.3. Problems identified 34

5. Implementation 35
5.1. Simple example 35
5.2. My implementation 38
5.2.1. Backend side 38
5.2.2. Front end side 41

6. Conclusion 45

Bibliography 49

List of images 51

1. Introduction

Major health institutions and recognized doctors have talked about connection between social media and depression. They mostly agree that scrolling through social media makes you feel like your life is not good enough, that you don't look as good as most people or don't have as much fun as others. According to many researchers depression caused by social media is the reason for very high suicide rate in younger population. Suicide rate in 2018 was highest after the year 2000. Depression is a big problem in today's world, not only in the younger population but in complete population active or inactive on social media. Not all depressed people are having problems caused by social media, but can show signs of depression in their posts which can be used to detect mental health problems and help people.

Machine learning, also known as ML, is a method of data analysis that is used for automated analytical model building. ML is very valuable in industries that work with large amounts of data. Using ML in order to predict outcomes can be used in order to work more efficiently or to gain an advantage over competitors. Since ML can work with large amounts of data and social media sites are basically big collections of user data, ML can be used in order to analyse the user data and make different actions upon them. Those actions can be various suggestions, customizing the displayed content, and many more.

Reading about possibilities with ML, I have decided to see into the topic and investigate if it is possible to recognize mental health problems using social media and modern technologies.

2. Depression and social media

What is depression and why are so many people affected by it? Is there a connection between depression and social media? Can anything be done about it?

2.1. Mental health

2.1.1. Depression

Depression (major depressive disorder) is a medical condition, a serious illness that affects human emotions, actions and the way they think. It causes loss of will to do activities one has once enjoyed, causes feelings of sadness and lack of ability and will for work and day to day obligations.
List of symptoms:

- feeling sad or having a depressed mood,
- changes in appetite - changes in weight,
- loss of interest in activities once enjoyed,
- difficulty concentrating,
- difficulty thinking and making decisions,
- feelings of worthlessness or guilt,
- lack of energy,
- sleeping problems - lack of sleep (insomnia) or increased need for sleep (hypersomnia),
- increased need for procrastination and
- thoughts of suicide or death.

If these symptoms are visible and affect ones life for more than two weeks going to the doctor is recommended. There can be other reasons for these symptoms as some diseases can mimic the depression. This is why they should not be neglected.

According to the WHO (World Health Organization) depression is the most common illness worldwide and affects more than 22% of population. Interesting fact is that 7.1% of the American adults are people suffering from this illness in a given year [1]. Depression is the leading cause of disease-related disability in women and the shocking news is that the lifetime prevalence of a major depressive disorder in women (21.3%) is almost twice that in men (12.7%). Data used in documenting this ratio has been collected in different parts of the world, different countries and ethnic groups [2]. It has been documented that men and children show similar incidence rates when women of a certain age show increase in mental health

problems. This is often being related to the child birth as it sometimes causes postpartum depression, but the it has been documented that the differences in prevalence appear around the age of 10 years which is much earlier than we were expecting. This differences persist until midlife an then they disappear [3].

Figure 1 shows a diagram of 12-month prevalence of major depressive episode 18+, by gender, race and age. The y-axis shows percentage of people in a certain group affected by this illness. The difference about depression affecting males and females can be also seen on this diagram and it show that around 9% of females suffer from this illness and this data also shows the depression rate for males is almost 5%. Concerning is the fact that people of age between 18 and 25 have the rate of almost 11% and those of age between 26 and 50 have the depression rate of around 7.5%. Depression rate for population older than 50 is around 5% which shows that depression is more prominent between younger population. Interesting is also the fact that AI/AN (American Indian/Alaska Native) population is the most depressed according to these information but are followed by White population. The least depressed population are the Asians.

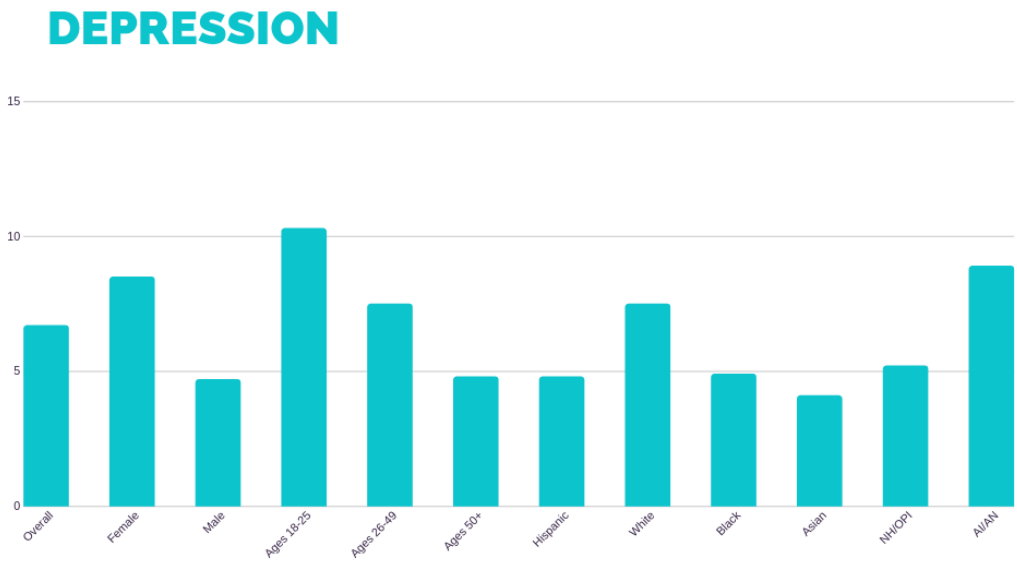


Figure 1: 12-month prevalence of major deppresive episode among adults (Created according to: Rabah Kamal, 2017)

On figure 2 we can see share of the population with depression. The darkest blue shows most depressed nationalities and light green, almost yellow shows those that are least depressed. We can see that Alaska, North America, Maroco, Scandinavian countries (Sweden and Finland), Iran and Australia are the most depressed nations. Croatia is colored with light green color and which can be read as rate between 2.5% and 3%. Most countries are colored with blue color which means that the depression rate is around 4%.

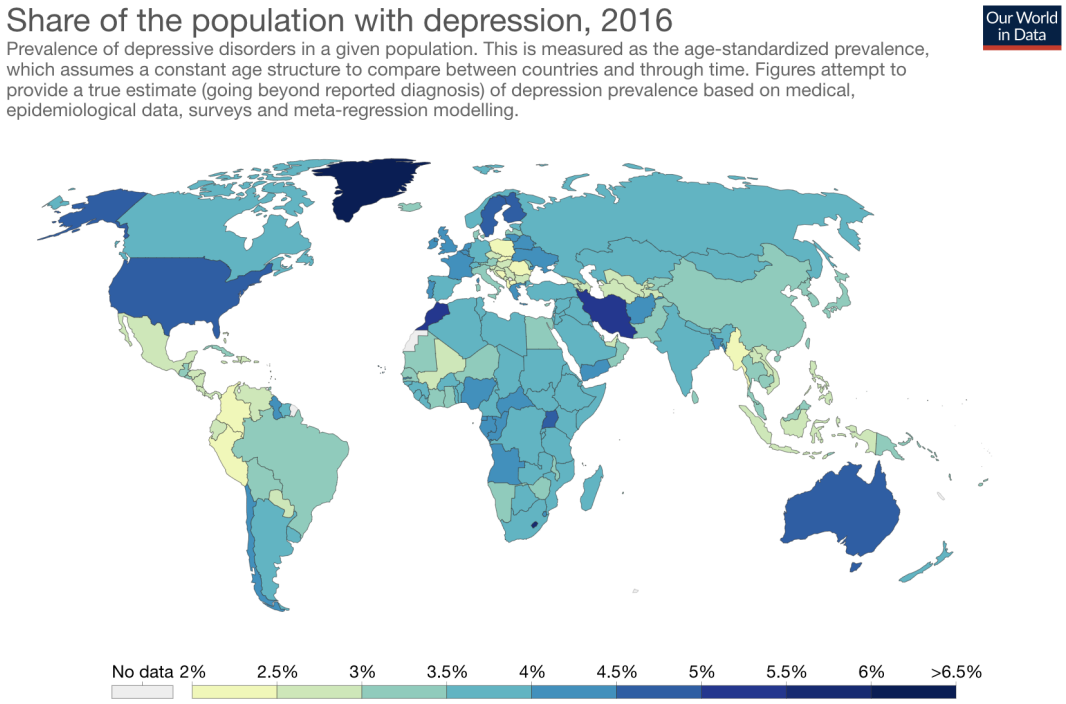


Figure 2: Share of the population with depression (Source: Hannah Ritchie, 2018)

2.1.2. Suicide

Huge problem in today's population are suicide rates which are higher then ever. According to WHO [6] there are about 800 000 deaths due to suicide every year. There are 525 600 minutes in one year, which would mean that every 40 seconds one person takes its own life. Data says that for each suicide that happened there were around 20 other attempted suicides.

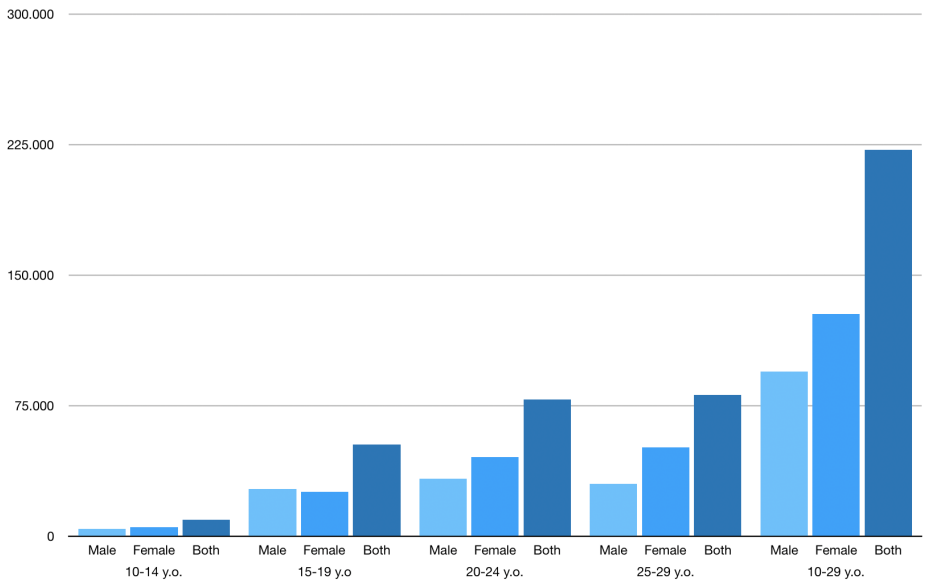


Figure 3: Suicide rates (Created according to: WHO, 2017)

Almost 225 000 out of 800 000 suicides are of those that are 29 years old or younger. This makes almost 30% of all suicides. Figure 3 shows data on suicides by age groups and sex. In almost every age group female suicide rates are higher than male, but after age of 20 the difference becomes significant. What strikes me the most is the number of the number of suicides in the two youngest groups of people (10 to 19 years old) because combined they add up to a number just below 75 000. [6]

2.2. Social media

Social media refers to interactive computer-mediated technologies that are imagined to facilitate the creation and sharing of information, ideas, photos, events, etc efficiently, quickly and in real-time. It can be accessed trough mobile phones, laptops, tablets and computers. Most social media sites have the associated apps. Nowadays people use social media for many different purposes. For example, it can be used for marketing, but also for finding jobs or even finding employees.



Figure 4: Social media (Illustrated by me)

Many statistics show that most popular social media sites are Facebook, Instagram, Snapchat, Youtube and Twitter. There are some statistics that say that 74% people use Facebook on daily basis, 63% use Snapchat on daily basis and 60% use Instagram. Each of them has its associated app for mobile phones which makes them easy accessible. Boredom is the main reason for opening these apps on phones, but there are also different reasons as job opportunities, media marketing, but also just scrolling and having fun.

As social media got more popular, there are certain people that became "famous". For example, we have a lot of popular youtubers, Instagram influencers and so on. One might ask them self what are influencers. Urban dictionery [7] states that incluencers are: "Someone with lack of intelligence and a lot of free time, followed by tons of idiots on some social network, usually instagram. "Influencers" are called so because their opinion worths something only to primates with IQ lower than room temperature, therefore those primates tend to follow influencer's actions without thinking about it."

This might be the worst definition ever, as it is clearly using harsh language and does not speak well about others. But, it can be used to explain what is happening. Influencers are, therefore, people followed and admired by many, that are using their social media to become famous, to show others how awesome their life is, to do campaigns for many factories and thereby making money.

2.2.1. Positive social media influence

Is there such thing as positive social media influence? Of course it is. There are many ways to use social media in a positive way. I would divide positive sides of social networks into several groups:

- impact on learning,
- career impact,
- globalization and
- influence on social relations.

Social media can be used to create study groups and allows students to exchange learning materials in matter of seconds and enables them to teach each other certain subjects. Students that were absent from school can get information on homework or tests and don't need to have any consequences. Related to this is also use of social media in order to learn new skills or improve in skills learned in school. It can be even used for changing career paths and there are many Youtube videos of people that claim to have used social media networks to learn new skills so they could change jobs and be happier. Such example is Anjana Vakil, an English teacher turned software developer[8]. Using social media sites such as very famous LinkedIn, you can now apply to different jobs in seconds, or just show off your skill by posting different projects you have worked on.

It makes technology somewhat more appealing to the older generations and can be used to motivate them so they would learn more about technology and how to use it. What also makes social media appealing is the way it makes us connected to people in different parts of world not giving us the feeling of distance between. Older generations can talk to their relatives that live in different states when younger use it to connect with people around the world and make friends. Social media in most countries gives us the right to open communication, makes it easier to discuss certain subjects but also makes shy people a bit more open and gives the opportunity to open up before actually talking to someone in real life. Nowadays social media is used as a tool for conducting business and political campaigns. It can be utilized to spread social awareness and kindness, share love and positivism.

2.2.2. Negative social media influence

Although using social media can have a very positive affect it can also lead to serious problems, especially for younger population. There are many stories about young people being bullied and because of it, they decided to take their own lives. Negative sides of using social media:

- depression and anxiety,
- cyberbullying,

- fear of missing out,
- unrealistic expectations,
- negative body image,
- addiction and many more.

These consequences of using social media are interdependent, which means that usually if you are affected by one of these negative sides, you are or will be experiencing rest too. Most common problem with social media is cyberbullying which intents to cause negative body image and problems with depression and anxiety. Cyberbullying was at its highest peak during mid 2000s, but nowadays kids are more aware of their actions against each other. By any means does this mean that cyberbullying is not a problem anymore, since it is still one of the most common problems in younger populations.

With invention of Facebook, Snapchat and Instagram people started to show their life on social media and in most cases it was not a real representation of life they were living. This became a trend and almost everyone started to enhance their pictures and looks so people following them would see how wonderful their life is. Not only did this cause negative body images in youth, but also created unrealistic expectations. Not being able to achieve those unrealistic expectations, having a fear of missing out on that great life someone has shown on their social media causes more frequent use of social media which leads to addiction. All of these problems lead to even worse one and that is depression.

2.3. Social media and depression

Due to the rapid increase in the number of users of social networks, numerous researches have been made dealing with the impact of social networks on the mental health of individuals. It is very interesting that the results of the researches do not agree on their findings, that is, some of the studies are even contradictory. I have found two studies which state completely different research results. Coyne, Rogers, Zurcher, *et al.* research, which was taken during eight years, states that time spent on social media does not impact mental health and is not associated with increased mental health issues. On the other hand, research done by O'Keeffe and Clarke-Pearson states that cyberbullying, and exposure to inappropriate content can lead to mental health problems.

For one thing, all the articles I have been able to find and read are in agreement, and it is that signs of depression are visible in individuals' social media posts. Although it is uncertain whether the use of social networks is strictly related to depression, it is a certain fact that we can use the posts on social media in order to predict ones mental health.

3. Machine learning

3.1. What is machine learning?

By the definition, machine learning (ML) is an application of artificial intelligence (AI) that provides systems the ability to learn and improve their knowledge from its experience.

It is the scientific study of algorithms and statistical models that computer systems can use in order to perform a specific task. What is different while using ML instead of ordinary algorithms is that task can be performed without using explicit instructions, instead by relying on patterns and interfaces. ML is focused on the development of different computer programs that can use different data in order to learn it for themselves. Data used for learning is called "training data" and it enables the algorithm to learn and make future predictions or decisions. It is closely related to computational statistics, which is used for making predictions using computers. There is also a big influence of mathematics, which is used for algorithm optimization. While talking about ML there is often mentioned term "data mining". Data mining is a field of study within ML. It focuses on exploratory data analysis and unsupervised learning. There are different approaches to machine learning which include different learning methods, models and training models.

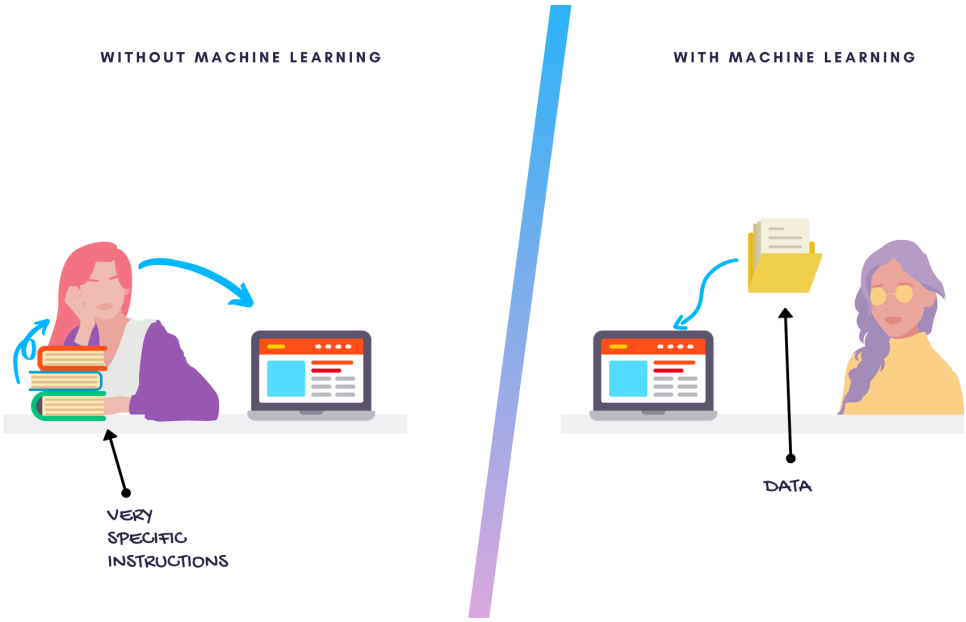


Figure 5: Machine learning (Illustration made by me)

3.2. Machine learning methods

There are many different learning methods. For each problem the engineer should determine what would be the best learning method and which learning algorithm is going to be implemented.

3.2.1. Supervised learning

It is a ML task of learning a function that maps input to an output based on example input-output pairs. It uses training data to infer a function which is later used for mapping new examples. Name "supervised learning" is used because the process of an algorithm learning from the training data sets can be compared to a teacher supervising the learning process. Best case scenario would allow for the algorithm to determine the class labels for unseen instances. There are some steps defined that need to be followed in order to solve a given problem of supervised learning:

1. Determine what kind of data should be used as a training set. In shape analysis this could be a matter of a certain shape, or even many different shapes next to each other.
2. Training set that is a representative of a real-world use of the function needs to be gathered. Here, set of inputs is gathered as well as set of outputs, most likely from human experts.
3. Determine the input feature representation of the learned function.
4. Determine the structure of the learned function and its corresponding learning algorithm.
5. Run the algorithm on the gathered training set. This may require the user to determine some control parameters.
6. Evaluation of the results. Accuracy of the learned function needs to be checked.

The most important step is choosing which algorithm to use. There is a wide range of supervised learning algorithms, yet each has its own strengths and weaknesses. There is not a single algorithm that could be called the best for all supervised learning problems. In fact, while working on each problem there should be used algorithm that works best for that specific problem. There are four main issues that should be taken into consideration while choosing supervised learning algorithm:

1. **Bias-variance tradeoff** - this issue is related to training data sets. The problem is that you can have several equally good data sets. By choosing a certain data set a learning algorithm becomes biased for a particular input. This can be used with malice aforethought. A learning algorithm has variance for a certain input if it predicts different outputs when trained with different training data sets. This means there is a prediction error for a learned classifier and it is related to the sum of the bias and the variance of the learning algorithm. There is a tradeoff between bias and variance and it is very important that algorithms with low bias are not too flexible because that would mean it could fit each data set differently.
2. **Function complexity and amount of training data** - Amount of training data needed algorithm to learn depends on a complexity of a true function. This means that for a simple problem there will be needed small amount of data while for complex problems there will be needed larger amounts.

3. **Dimensionality of the input space** - Next thing we need to worry about is dimensionality of the input space. Even if the true function depends on small number of features, if the input feature vectors have high dimension the learning problem will be difficult. There can be done dimensionality reduction where the engineer manually removes irrelevant features.

4. **Noise in the output values** - Even if the desired output values are sometimes incorrect the learning algorithm should not attempt to match a function exactly to the training examples. It would lead to overly fitted algorithm. If the function I am trying to learn is too complex for my learning model, there can also be over fitting which would lead to corrupting the training data. This is called deterministic noise.

There are some other factors to consider: data should also be checked for redundancy, presence of interactions and non-linearities and heterogeneity. This is why the engineer should compare multiple learning algorithms and decide which one works best for the current problem. These problems can be grouped into two categories of problems: **classification** - the output variable is a category, such as “pink” or "blue" or “cat” and "book" and **regression** - the output variable is a real value, such as "temperature", "pressure" or “dollars” or “weight”.

There are many supervised learning algorithms but these are the most popular ones: support vector machines, linear regression, logistic regression, naive Bayes, linear discriminant analysis, decision trees, k-nearest neighbor algorithm, neural networks and similarity learning.

3.2.2. Unsupervised learning

Unlike supervised learning, unsupervised learning algorithms take data sets that contain only inputs, find structure in the data by grouping it or clustering. The data used by learning algorithms is not labeled or classified which means that these kind of algorithms need to identify certain similarities and commonalities in the data and based on their presence or absence create groups or clusters. This kind of learning is also known as self-organization. There are two main methods used in unsupervised learning: **principal component** and **cluster analysis**.

Principal component analysis is a statistical procedure also known as **PCA**. To explain this, we first need to remind ourselves of coordinates translation from one coordinate system to another. Coordinates translation is displaying the coordinates of one system in another system and it is possible for all coordinate systems. This means we can change the basis of a space to which ever coordinates we wish to. This can also be applied in much higher-dimensional spaces. If we were to have 200 000 dimensions in some space, we could choose a basis and only 400 most significant vectors. These chosen vectors are what we call principal components. This newly created space should be much smaller in dimension but maintain the complexity of the data. It allows us to get an understanding of the dataset's organization.

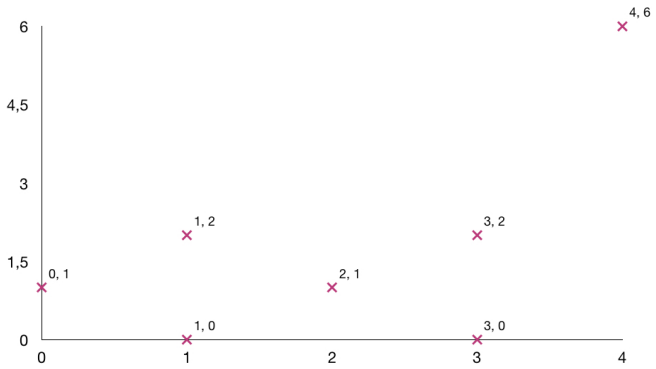


Figure 6: Coordinate system (Illustration made by me)

Figure 6 shows a basic coordinate system with center in (0,0) and some coordinates. Those same coordinates can be shown in a different coordinate system, which is shown by figure 7. New coordinate system is rotated for 30 degrees.

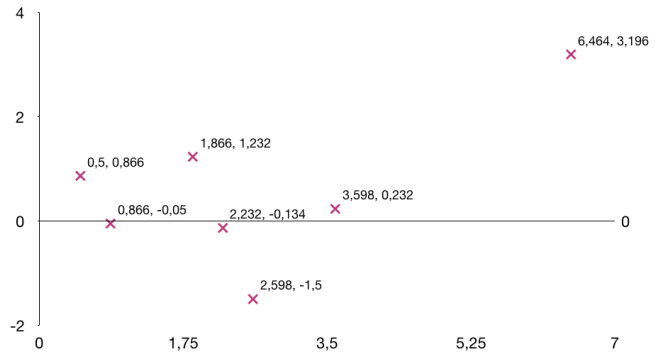


Figure 7: Coordinate system after translation (Illustration made by me)

Cluster analysis is being used in unsupervised learning when we want to group or create segments of datasets that have common attributes in order to extrapolate algorithmic relationships. It identifies common attributes in the data and based on absence or presence of those attributes groups the data. Cluster analysis is not one specific algorithm, but can be looked at like a general task that needs to be solved. The goal can be achieved by a combination of various algorithms that differ in their understanding of what makes a certain cluster. Cluster analysis is not an automatic task and therefore it consists of iterative process of learning and optimization that involves trial and failure. In some cases, more often than not, it is necessary to modify data preprocessing and model parameters until we get the result that meets our requirements and the desired properties.

Figure 8 shows simplified picture of how data is grouped into clusters. Each color represents one cluster.

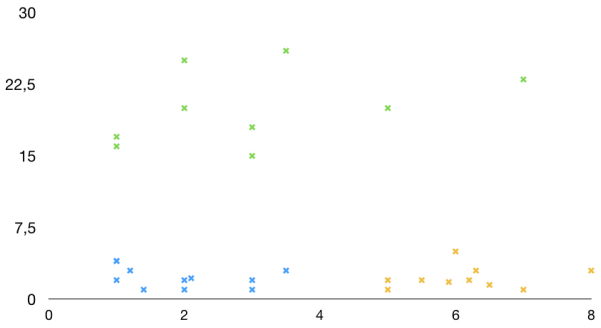


Figure 8: Clusters (Illustration made by me)

3.2.3. Semi-supervised learning

It is typically a combination of a small amount of labeled data and a large amount of unlabeled data. Researches have shown that having a small amount of labeled data in combination with a big amount of unlabeled data can make a big improvement in learning accuracy. Based on the labeled data, unlabeled data is being separated into clusters. The only downside is the fact that we need a highly skilled human agent or an experiment in order to acquire labeled data. Although acquiring labeled data can be quite expensive, acquiring unlabeled data is really inexpensive.

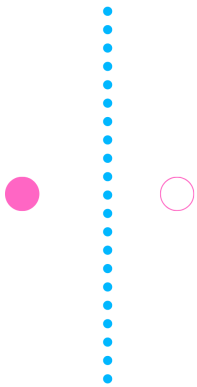


Figure 9: Labeled data (Illustration made by me)

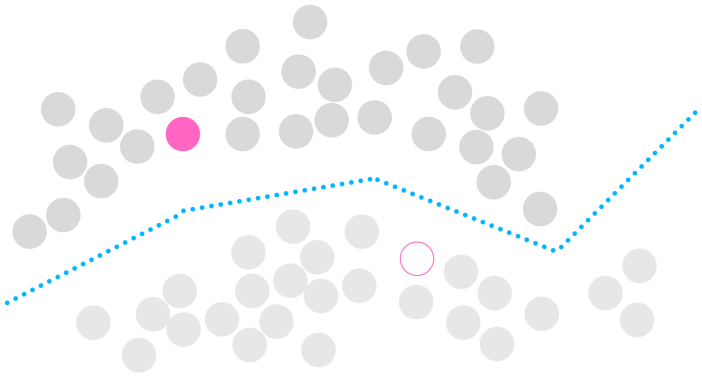


Figure 10: Unlabeled and labeled data (Illustration made by me)

Figures 9 and 10 show an example of an effect unlabeled data has in semi-supervised learning. Figure 9 shows how boundary is created after the encounter with only one positive (pink filled circle) and one negative (pink outlined circle) example. If we add unlabeled data, the boundary we adopt is shown on a figure 10. This happens while performing clustering and labeling them with labeled data and pushing the boundary far away from regions with high density.

We can have huge amounts of unlabeled data, but in order to get any use of it we need to assume some structure to the fundamental distribution of data. There are three types of assumptions used by semi-supervised learning algorithms:

- **Continuity assumption** - this assumption is based on geometrical position. It means that points that are close to each other are very likely sharing a label.
- **Manifold assumption** - this assumption is used when learning is using distances and densities defined on the manifold. This is based on a fact that the input data can lie on a manifold that has much lower dimension.
- **Cluster assumption** - this assumption is based on the fact that data can create discrete clusters and the points that are in those clusters share a label.

Semi-supervised learning has very important role in machine learning privacy. There are many different approaches in which the supervised data is presumed private, and a student model is trained using only unlabeled presumed public data. This gives us the guarantee of efficient learning that does not rely on user data.[11]

3.2.4. Reinforcement learning

The next area of ML we are going to talk about is reinforcement learning. It is about taking the right measures to maximize some notion of cumulative reward in a particular situation and is employed by various software and machines to optimize the behavior or path taken in a particular situation. It is one of three basic machine learning paradigms: supervised learning, unsupervised learning and reinforcement learning. The input should be an initial data state from which the model will start, where outputs differ depending on a solution to a particular problem. The training is based on input where the model will return a state. Based on the returned state the user will decide to punish or reward the model. To find the best solution we need to find the best reward.

The greatest difference between supervised learning and reinforcement learning lays in the fact that in reinforcement learning labelled input/output pairs do not need to be present. In the absence of the training dataset, reinforcement learning is bound to learn from its own experience. Also, a big difference is in giving labels to sequences of dependent decisions, while in supervised learning labels are given to each decision. Example of reinforcement learning would be a chess game.

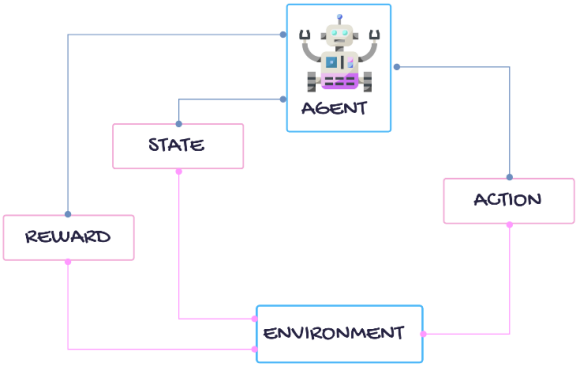


Figure 11: Reinforcement learning (Illustration made by me)

Reinforcement learning is studied in many other disciplines such as game theory, information theory, multi-agent systems, statistics and many other, which can be attributed to its broad application and generality.

Highly known algorithm for control learning is **brute force** which is also known as generate and test. It is a very general problem solving technique which entails two steps. First step is returning samples while following each possible policy and the second step is choosing the policy with largest expected return. The biggest problem while using this kind of algorithm is large, or even infinite, number of policies.

3.2.5. Feature learning

Feature learning is also known as representation learning and is a set of techniques that give a certain system the ability to automatically discover the representations needed for feature detection or classification from raw input data. This allows the machine to learn the features but also to perform specific tasks. It can be both unsupervised and supervised.

Supervised feature learning is, as expected, learning from labeled data. The data label is a help in this case because it enables the system to calculate an error term. This error term gives us the information about the degree to which the system fails to produce the label and can be used in order to correct the learning process.

Unsupervised feature learning is, contrary to supervised learning, learning from unlabeled data. Its goal is to discover low-dimensional features that seizes some structure based on the input data. It enables a form of semi-supervised learning where features that are learned from unlabeled data set can be used in order to improve performance in a supervised setting.

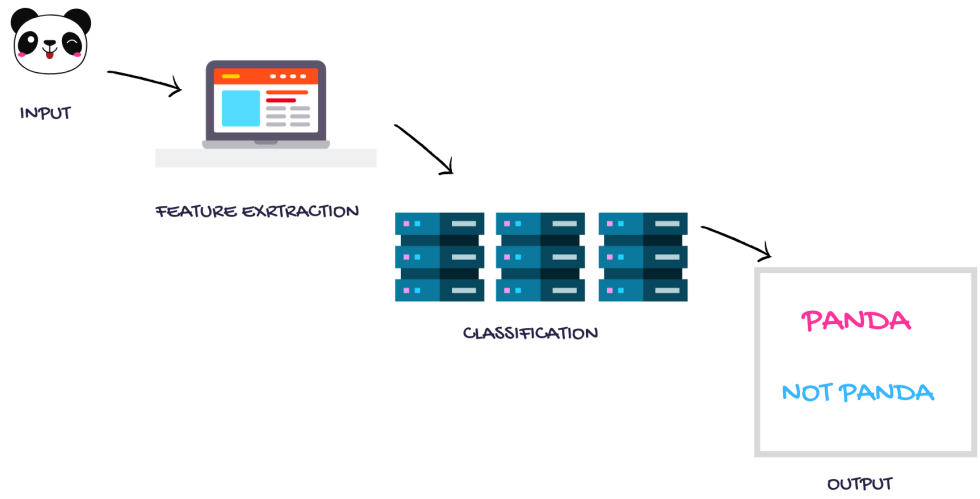


Figure 12: Feature learning (Illustration made by me)

The biggest advantage of using feature learning is lack of need for manual feature engineering, which is otherwise necessary, and this is why a machine can learn both a specific task and learn the features themselves.

3.2.6. Sparse dictionary learning

Dictionary learning is a part of signal processing and machine learning. Its main goal is finding a dictionary in which some training data has a sparse representation. The dictionary is considered to be better if the representation is sparser. It is in the form of a linear combination of basic elements and basic elements themselves. Small elements that compose a dictionary are called atoms. Structure of those dictionaries and the problem setup allow the dimension of the signals being represented to be much higher than those being observed. This type of algorithms iteratively adapt an initial dictionary to a certain signal class. Observations from this signal class can be sparsely coded in the initial dictionary with low error.

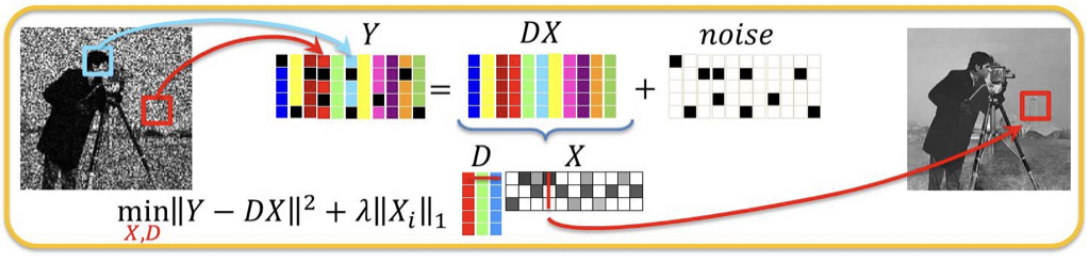


Figure 13: Dictionary learning (Data61, 2015)

Dictionary learning can be used on any type of sensor modality, any type of data (statistical, dynamic, discrete, etc.) and on optimization, factorization or estimation problems [13]. This type of machine learning is often used for speech enhancement or image/video corrections. Latest iPhones have the image fusion ability which is actually using dictionary learning on images. Figure 13 shows in the simplest way possible how this algorithms work. In the start we have our image with a lot of noise and after using the algorithm on the picture it removed noise from the picture and created much clearer version. In order to replace pixels that were previously noise, this algorithm would change such pixel with one that is similar to pixels around it. The most important part of the noise removal is training the dictionary to fit the input data in order to significantly improving the sparsity.

There have been developed numbers of algorithms to solve this problem such as method of optimal directions (MOD), K-SVD, stochastic gradient descent, Lagrange dual method, LASSO, parametric training methods and online dictionary learning (LASSO approach).

3.2.7. Anomaly detection

Anomaly detection is the process of identifying unexpected elements in data sets. Unexpected elements are all the items or events that that are different from the norm, or better said, instances that seem to fit least to the remainder of the data set. It is often applied on unlabeled data and therefor is called unsupervised anomaly detection and this is the most common usage. Supervised anomaly detection requires data sets that have been labeled as "normal" and "abnormal".[14] Supervised anomaly detection requires training a classifier. And, naturally, there are also a semi-supervised anomaly detection techniques. These techniques construct a model that represents normal behavior from a given normal training data set.

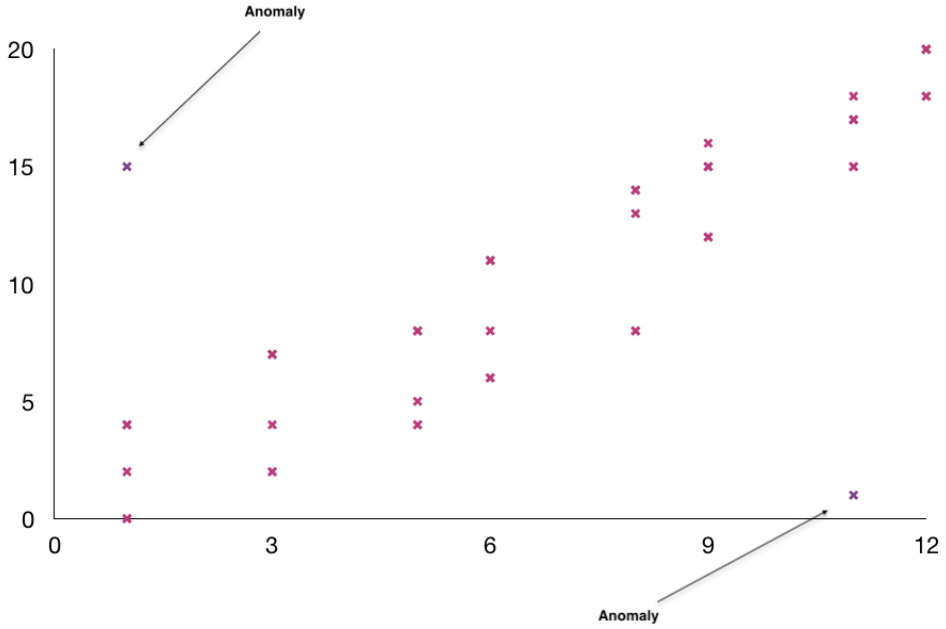


Figure 14: Anomaly detection (Illustration created by me)

Figure 14 is a visual representation of data scatter where most elements are centered linearly around the $y = x$. There are two anomalies, one above the $y = x$ and one below. It is visible that those elements are far from any other, therefor it is easy to identify them.

Anomaly detection can be used in many different domains: fraud detection, fault detection, intrusion detection, event detection, health monitoring and monitoring ecosystem disturbances. Often this would be used in preprocessing in order to remove any anomalies from data sets. Most popular anomaly detection techniques are: density-based techniques, one-class support vector machines, replicator neural networks, autoencoders, long short-term memory neural networks, Bayesian networks, hidden Markov models (HMMs) and many others.

3.2.8. Association rule learning

Association rule learning is somewhat different than other machine learning methods. It is rule based and used to discover relationships hidden in large data sets. These rules are used to find relationships between elements present in different entries and can be used in many different ways. One of the most popular examples would be Netflix which recommends new TV shows based on the list of movies and/or TV series you already watched or liked and compare it to lists from other users with similar picks.

ID	Watchlists
1	{Hulk, Captain America, Iron Man, Batman, Pretty little liars, Riverdale}
2	{Hulk, Captain America, Iron Man, Superman, Breaking Bad}
3	{Hulk, Captain America, Breaking Bad, Harry Potter movies}
4	{Hulk, Captain America, Iron Man, Breaking Bad}
5	{Hulk, Captain America, Breaking Bad, Superman}
6	{Hulk, Captain America, Iron Man, Batman, Riverdale}

Figure 15: Association rule learning (Illustration created by me)

If you look at the data set shown on figure 15, you may be able to notice certain patterns. Here are some noticeable patterns:

1. Most people who watch Hulk and Captain America also watch Iron Man.
2. Most people who watch Hulk and Captain America also watch Breaking Bad.
3. Most people who watch Hulk, Iron Man and Captain America also watch Batman.

Rules have two parts: antecedent and consequent and their relation is $\{antecedent\} \rightarrow \{consequent\}$. Our Netflix example would create following rules:

- $\{Hulk, CaptainAmerica\} \rightarrow \{IronMan\}$
- $\{Hulk, CaptainAmerica\} \rightarrow \{BreakingBad\}$
- $\{Hulk, CaptainAmerica, IronMan\} \rightarrow \{Batman\}$

In order to ignore patterns in data that occur by chance, we need to calculate support and confidence. By calculating support we get the information about the significance of the rule whereas by calculating confidence learn how likely for the rule $\{A\} \rightarrow \{B\}$ A and B will occur together.

In addition to the above Netflix example association rules are used in many application areas such as Web usage mining, intrusion detection, continuous production and bioinformatics.

3.3. Machine learning models

Mathematical model is a representation of a system using mathematical concepts and language, whereas statistical model is a parameterized set of probability distributions. Models in general have multiple purposes, such as descriptive, prescriptive or predictive analytics. Same as in any other field, the goal of developing models in machine learning is to get insights from data which then allows us to make better business decisions. We could say that a model is a distilled representations of a greater picture.

There are various types of models that have been researched and used for machine learning. Such models are: artificial neural networks, deep learning, decision tree learning, support vector machines, Bayesian network and genetic algorithm. While training a machine learning model it is needed to collect a big, representative sample of data from a certain training data set. This data can be in many forms and collected from each individual user of a service. This brings us to the biggest problem in training a machine learning called overfitting. Overfitting can best be explained on an example of image recognition. Giving an algorithm cat and dog images, should result in having two clusters, one consisting of all of the cat pictures and another one with dog images. It is important to analyze output data for certain inputs in order to catch sight of overfitting. If we would to find a dog in a cluster where only cats should be, it would mean that our algorithm has problems with overfitting and needs some more training.

3.3.1. Artificial neural network

Artificial neural networks are most commonly used tool in machine learning. These are brain-like systems which are meant to replicate the process of humans learning. It is made of interconnected group of nodes that meant to represent network of neurons in a brain.[15] These networks have input layers, hidden layers and output layers. Hidden layer consists of small units that take the input and transform it to such data that the layer output can use. These kinds of networks are useful while finding patterns that are too complex for a human to extract.

I will offer an explanation of deep learning neural network on a very simple example. For example, if we want to build an algorithm that would recognize different objects for example a ball from a brick, our algorithm should take the input and create layers that would contain different features. First layer could be brightness of pixels. Second could recognize edges in the pictures which is based on difference in pixels. Next layer could be recognition of different textures, fifth layer could recognize shapes. At this point, the researcher who has trained the network needs to look at the outputs and label them. This step is needed in first few iterations, while afterwards algorithm can carry out its own classification tasks. In the beginning the goal was to use artificial neural networks in order to solve problems in the same way a human brain would, but over time, in order to perform specific tasks, it moved away from biology. Mostly it is used for task such as speech recognition, face recognition, machine translation, special network filtering and playing video games.

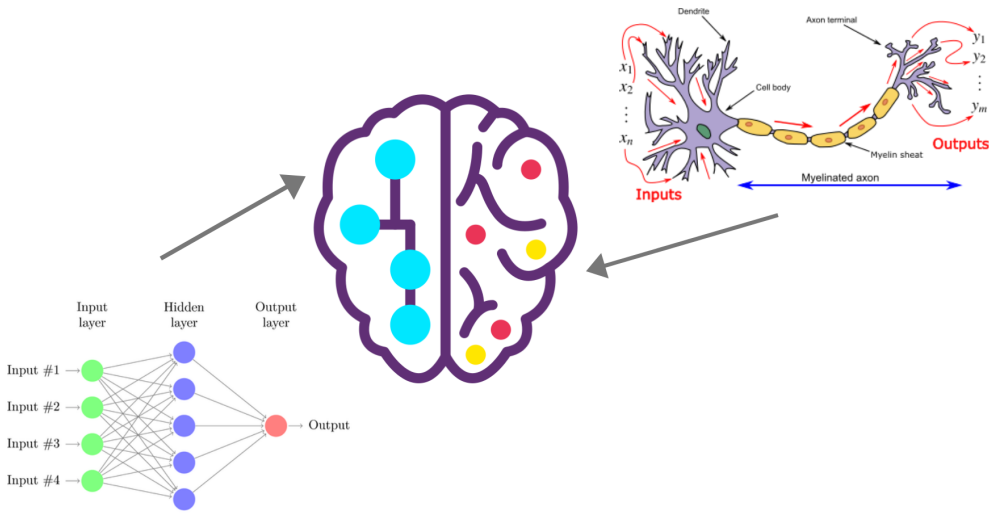


Figure 16: Artificial neural network (Illustrated using Eland and Vu-Quoc)

3.3.2. Decision tree learning

Haijian Shi has said that the decision trees are powerful predictors and that these models are a good representation of structure of a data set[18]. The name comes from its tree-like structure. Each decision tree consists of root node, decision nodes and/or chance nodes and branches. For this to be applied successfully, attribute that would make a root node needs to be selected. Figure 17 shows a simple decision tree that deals with the problem of deciding what is the weather like.

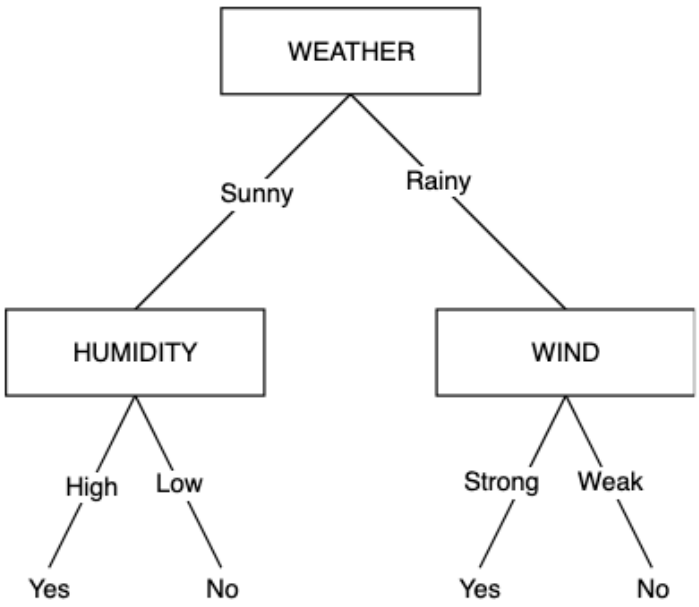


Figure 17: Decision tree (Illustrated by me)

Decision tree learning uses a decision tree for prediction purposes to transform from

observations about the certain subject to conclusions about its target value. In decision tree learning the tree acts as a predictive model. It is used in many fields such as statistics, data mining and machine learning. The process of making a tree is also called tree growing and consists of deciding which features to choose and which conditions can be used for splitting. It also needs to be aware of when to stop. In order to limit the splitting there should be either set a minimum number of training inputs or maximum depth of the model. For example, we can set a minimum of 5 training inputs, which would mean that there needs to be at least 5 occasions in which a certain decision is reached. Otherwise the leaf would be ignored. The second case is when we define maximum depth which indicates how long can the longest patch from a root to the leaf be. Pruning can be used to increase the performance of the tree. This process removes branches that are connected to low-importance features and lowers the complexity of the tree.

3.3.3. Support vector machines

In a book called "Support Vector Machines and other kernel-based learning methods" written by Mello Cristianini and John Shawe-Taylor [19] they say support vector machines (SVMs), often called support vector networks, use a hypothesis space of linear functions in feature spaces of high dimensions. They are trained with a learning algorithm that implements a learning bias based on a statistical learning theory [20], [21].

What this actually means is that SVM training algorithm would build a model which predicts in which category it should place a certain input. This requires a set of training examples that are marked to belong to either one of the two possible categories.

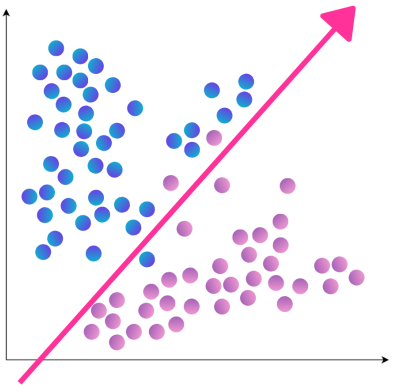


Figure 18: Low regularization value (Illustrated by me)

Figure 18 and figure 19 show examples of changing regularization values. This parameter, also known as C parameter from python's scikit-learn library, informs the SVM optimization of importance of avoiding misclassifying each training example. This means that larger value leads to choosing a smaller-margin hyperplane if it gets all the training points classified in correct classes.

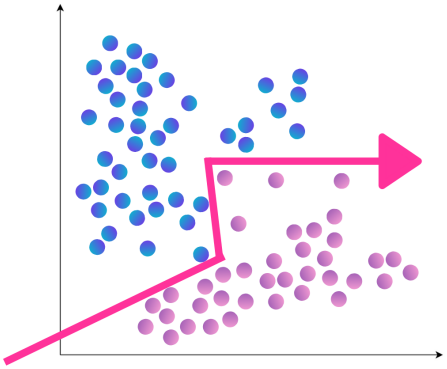


Figure 19: High regularization value (Illustrated by me)

3.3.4. Bayesian networks

Bayesian networks have many names, such as Bayes networks, belief networks, decision networks, probabilistic directed acyclic graphical model or Bayes model. A Bayes network is a probabilistic graphical model that represents a set of variables that are part of Bayesian inference for probability computations.[22] These networks aim to model conditional dependencies using directed acyclic graph.

Most common example shows the probabilistic relationships between diseases and symptoms. The network could be used to find the probabilities of presence of certain disease, if it were to be given symptoms. It can also be used in order to find probabilities of presence of oils in the ground.

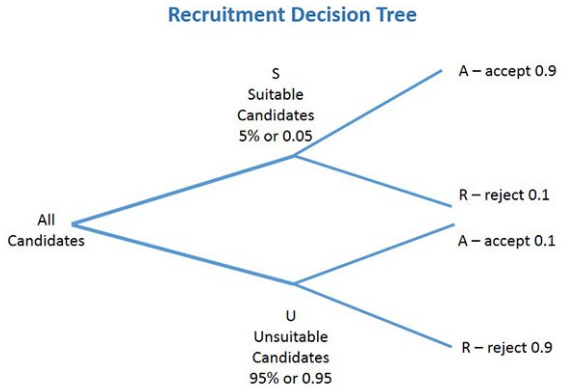


Figure 20: Bayes and decision tree (Dragons8mycat)

Bayes networks can often be shown using decision tree structures. Each branch has its probability of happening, as well as the probability of not happening. Depending on many factors, such as type of node. There are two types of nodes: decision nodes and chance nodes.

3.3.5. Genetic algorithms

A genetic algorithm is a metaheuristic that mimics the process of natural selection. This process is inspired by process of natural selection, explained by Charles Darwin, where the fittest individuals are selected for reproduction in order to produce individuals of the next generation.[24] Genetic algorithm, which uses bio-inspired operators such as crossover, mutation and selection, was introduced by John Holland but was later extended by his own student, David E. Goldberg. Machine learning was mostly used in order to improve the performance of these algorithms.

The whole process of natural selection has five defined steps. First two would be generating the initial population and computing fitness using fitness function. Generating the initial population happens only once by the run of algorithm. Then, we have selection, crossover, mutation and computing fitness. These four steps are in the loop until population converged. Algorithm stops once it reaches the population convergence.

```
START
Generate the initial population
Compute fitness
REPEAT
    Selection
    Crossover
    Mutation
    Compute fitness
UNTIL population has converged
STOP
```

Generating the initial population is a process that starts with a set of individuals called a population. Each individual, a solution to the problem we want to solve, is characterized by genes. Genes are sets of parameters that joined together form a chromosome. Each chromosome represents a solution. For each individual in a population you need to calculate how fit it is using a fitness function. This gives us information about the competitiveness that each individual has over the other individuals. In IT world it defines how good is a certain solution. This process occurs before entering the loop, and is repeated at the end of each loop. Selection is a process in which the fittest individuals are chosen to pass their genes to the next generation. Crossover is a process during which mating of a pair of parents happens. Mutation happens when some of the bits in the bit string flip, but has low random probability. The end of a process is reached when certain conditions are fulfilled and a solution matches minimum criteria. These algorithms are mainly used in order to solve optimization problems and are useful in many domains.

3.4. Usage and technologies

After getting to know basics about certain types of algorithms and their models, the key thing is to decide which technology to use. I have already mentioned many areas where it can be applied, but to draw some conclusions, machine learning can be used to solve almost any type of problem.

3.4.1. Usage

Nowadays, most common usage of machine learning is use of virtual personal assistants, such as Siri, Google now or Alexa. In a movie Iron Man, the main character has his own version of personal assistant, called Jarvis. Well known inventor of Facebook has his own version of Jarvis implemented in his home. This kind of personal assistants are shown to be able to establish a personal connection with his owners, which would make them even more useful. [25] Despite the absence of a system like Jarvis, many users are to use Siri on their iOS devices, Alex on Amazon products and Google now on Android devices on daily basis. These assistants can book appointments, call contacts, tell weather, tell jokes, find restaurants near us and many more. Machine learning algorithms are often used for sales and marketing. If you ever visited Amazon web site or similar sites, chances of your search data being collected in order to perform better at prediction which products an individual buyer would like to buy. Stores also use beacons to scan devices of their customers in order to show them different ads and promote their products. Similar algorithms are used by social media sites in order to recommend us users we might know. There are also many different ways to use machine learning in financial domain. For example, there are many pointers that could predict drop of stocks price or drop of virtual currency value.

An interesting way to use machine learning is making traffic predictions. Google maps use this kind of algorithm in order to predict traffic. Many people use GPS navigation services which collect data and share it with a central server. Despite the fact that not every user uses GPS, the data about location and velocities is being collected, saved and used by machine learning algorithms in order to estimate the traffic. Machine learning is also used in order to estimate the price of a cab ride. In this case all of the data of previous cab rides is being used to estimate the price of a new ride. Personally, my favourite example of using machine learning in everyday life is making smart cars. These cars can tell you all the needed data about needing to fill up the gas tank, changing the tires, changing car parts and it is all predicted using machine learning and data collected by many different sensors. There are also autonomous vehicles, also known as self-driving cars, which use algorithms in order to drive from point A to point B. These algorithms are trained to avoid obstacles and crashes, to follow traffic rules and mimic human driving. These algorithms are interpreting traffic-control devices without a need of a driver to actively operate any of the vehicle's control systems.

These are just some of the most popular cases of using machine learning. Some other fields in which machine learning is being used are healthcare, security, finance, marketing, e-commerce, robotics, BI, sales and CRM.

3.4.2. Technologies

Before starting to develop an algorithm, it is very important to see into advantages and disadvantages of using different programming languages. One should ask themselves few simple questions in order to decide on which language to use for the presented problem:

1. What do I need to do?
2. What programming languages am I familiar with?
3. Do I have enough time to use a new programming language?
4. Which programming language would best suit my needs?

Artificial intelligence and therefore, machine learning, is the fastest growing field of computer science. As previously mentioned, machine learning is being used in everyday basis which means that more and more computer systems are adopting this technology. Is there the best programming language for machine learning? No. There many programming languages that can be used for machine learning, but it is recommended to choose a programming language that comes with pre-built libraries. These programming languages have advanced support of data science and data models. In the beginning of 2019 Github has announced its programming language popularity list based on machine learning contributions. Based on their analysis on top of the list is Python, which is followed by C++, JavaScript, Java, C#, Julia, Shell, R, Typescript and Scala.[26] There are many more articles commenting about this subject, but most of them agree about these programming languages for machine learning development:

1. Python
2. R
3. JavaScript
4. Java
5. Lisp

Fabisiak has made a great analysis of these five programming languages and their usage in machine learning. His analysis of a programming language involves taking into consideration different aspects like: difficulty to learn, salary, available documentation, libraries, how is it used and which pros and cons does it have.

Python is a high-level, general-purpose, interpreted programming language created by Guido van Rossum. It was first released in 1991, but upgraded and therefore improved through the years. Its syntax is very simple and easy to learn, and there are numerous online courses which allow you to learn the basics in just few hours of learning. The language has its own core philosophy summarized in *PEP 20 – The Zen of Python*.[28]. It is one of the most popular programming languages thanks to the aforementioned traits. Python is used in web development,

but its usefulness in data science and AI is what sets it apart from others. It has extensive number of frameworks and libraries developed specifically for usage in machine learning sphere. Amazon and Google are known to use python for their projects such as Google's Gmail and Amazon's web store. Google has developed a TensorFlow framework in order to help them categorize spam mail, but has since broaden its functionalities.

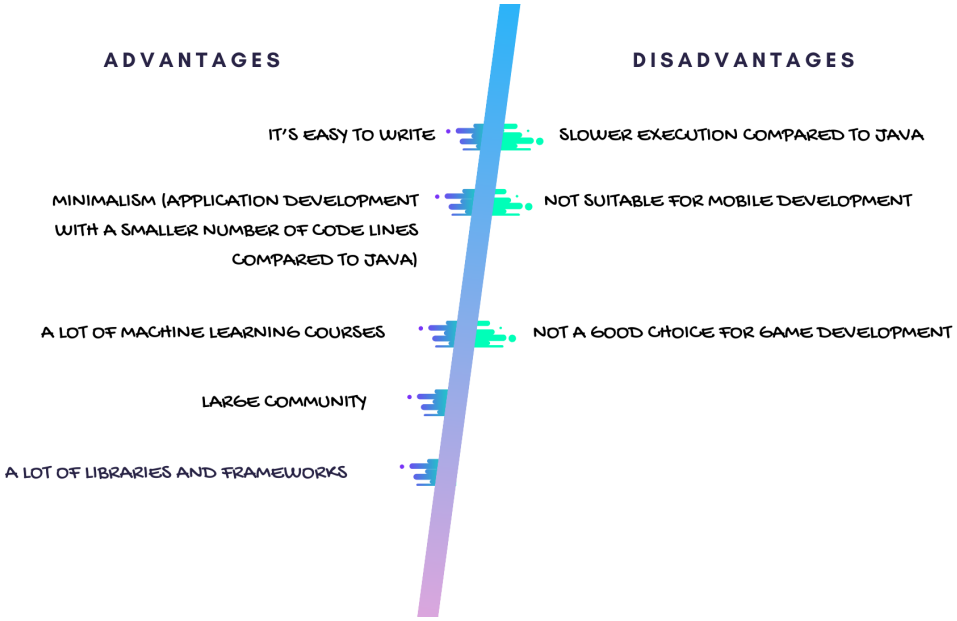


Figure 21: Python pros and cons (Illustrated by me)

R is a programming language as well as a free software environment. It is used for statistical computing, predictive analysis and graphics. R is often used among data miners in order to make data analysis. It was created by Ross Ihaka and Robert Gentleman at the University of Auckland and is currently under management of the R Foundation and the R Development Core Team. R is interpreted and dynamically typed language that has, much like Python, lots of packages, libraries, and materials that can help you in the learning process. Google, Uber, the New York Times, Bank of America and ANZ Bank are known to use R, as well as Facebook and Twitter.



Figure 22: R pros and cons (Illustrated by me)

JavaScript, abbreviated as JS, is a just-in-time compiled, high-level, object-oriented programming language. It was developed by Brendan Eich and first appeared in 1995. It is easy to learn the JavaScript basics, but it requires a lot of attention. It is much easier to learn if the person has already programmed in other programming languages, but it still belongs to the group of languages that are indeed easy to learn. Many popular sites use JS, such as Wikipedia, WordPress, Facebook, LinkedIn and eBay. TenserFlow.js library is one of the most popular machine learning development and training libraries and brain.js library if used in the field of neural networks.

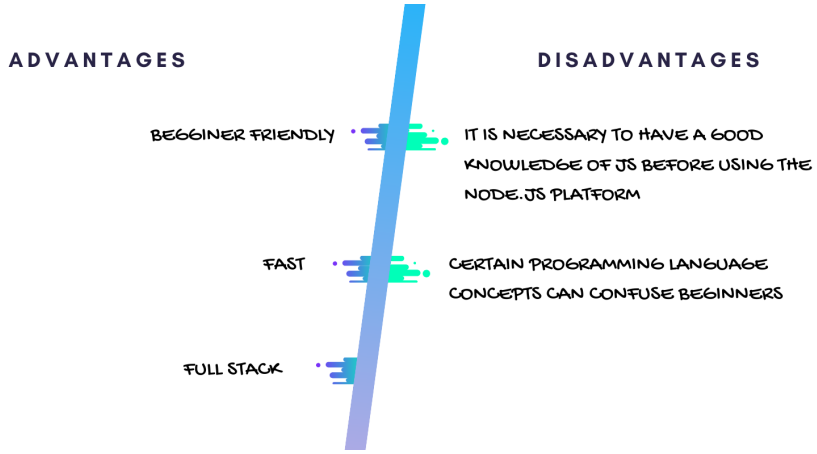


Figure 23: JavaScript pros and cons (Illustrated by me)

Java is a set of computer software and specifications. This is compiled and strongly typed programming language, developed by James Gosling at Sun Microsystems. It has been popular for many years and its popularity is not dropping. Java is used in many different computing platforms. Its usage reaches from embedded devices and mobile phones to enterprise servers and supercomputers. Compared to Python, Java is much faster and has better performance execution of the program. Nevertheless, Java requires a lot more time, effort and takes much longer to learn than Python. It is well known that companies like YouTube, Amazon, eBay and LinkedIn use Java for server-side.

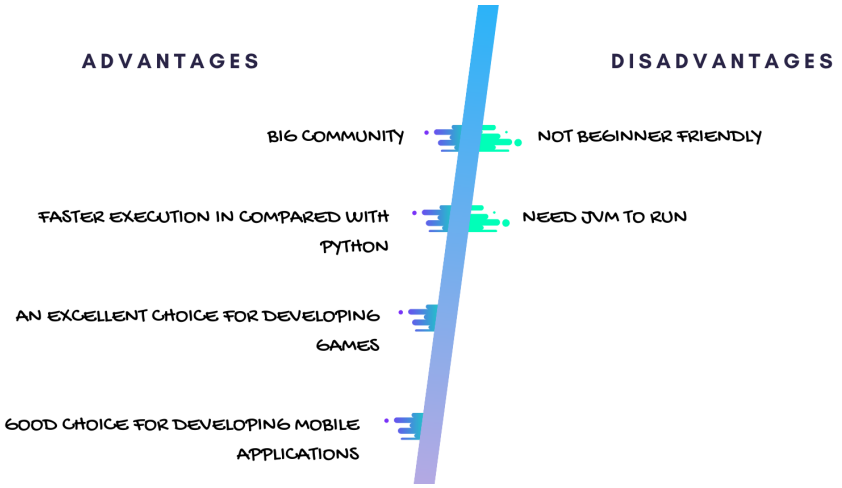


Figure 24: Java pros and cons (Illustrated by me)

Lisp is not one programming language, but a family of languages with a distinctive, fully parenthesized prefix notation. It was specified in 1958 by Steve Russell, Timothy P. Hart, and Mike Levin, and therefore is the second-oldest high-level programming language in use. Lisps most popular dialects are Clojure and Common Lisp. Most popular well known app that uses Lisp is Grammarly, which uses Common Lisp. The biggest problem with this programming language family is that it requires a lot of time and effort to learn. It is much easier to learn if the person previously used different programming languages, while it is extremely hard to learn for complete beginners. The next problem is a small community, which resulted with a small amount of framework and libraries.

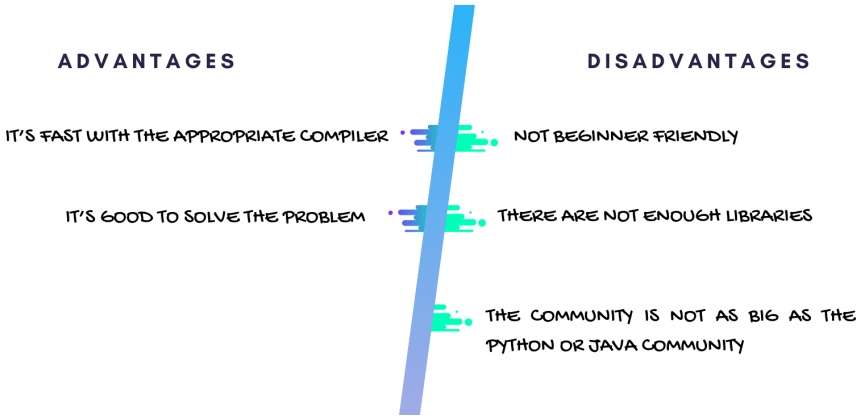


Figure 25: Lisp pros and cons (Illustrated by me)

Taking all things into consideration, we can not really say that one language is much better than others. Each has its own pros and cons, and what Fabisiak concluded was that Python and JavaScript deserve to be winners of this comparison. I would agree, but this is after taking into consideration the difficulty to learn each language and availability of frameworks and libraries.

4. Prediction of depression using machine learning algorithms

The whole topic of the master thesis is based on the assumption that machine learning provides the ability to identify mental illnesses and thus can lead to a reduction in suicide rates. Nowadays, almost every person has an account on at least one of the social media sites. We are all aware that our use of such services collects information about the things we view, the people we hang out with, the places we have been to, and many more. So, if we can use social networks to gather information about the popularity of particular sites or topics, why not do something good and discover the issues that users are having by analyzing the data? Research shows that depression is a big problem among young people, but also among middle-aged and older people, and most of these people do not know where to seek help or how to help themselves.

4.1. The idea

People are using social media more and more each day, which has prompted some of the leading mobile and computer manufacturers to implement algorithms that track how much time a user actually spends on such sites. By spending time on these pages, users, in addition to keeping track of what their friends and acquaintances are posting, need to share details from their personal lives. These details include information on individual's thinking, emotional state, activities, communication, and socialization. This data provides us with a indicators of certain feelings, such as helplessness, guilt, self-hatred or worthlessness. These four feelings are the characteristics of major depression.[29] Often mentioned as an indicator of depression is also difference in activity on social media, but it could be a result of other events in life. Therefore, we should not come to any conclusions based on this kind of information. What we should pay attention on is emotion, linguistic style and language. The goal would be to teach the algorithm to identify signs of depression using the data sets that have labeled data. Data used for the teaching of the algorithm could consist of pure textual posts, but also of image descriptions or links to songs. Ideally, the algorithm can recognize the tone of the post by analyzing the text, and it should be able to receive large amounts of data at the input and only find portions of the input that are significant for the further process.

Johannes Eichstaedt has, joined with other researchers, used an algorithm to analyze data of collected on social media profiles of consented users. They have used the algorithm to distinguish linguistic cues that could be used for predicting depression. The team has researched different posts and realized that they can extract emotional language markers. Eichstaedt and researchers state that one of the most significant markers is increased use of first-person pronouns, such as "I", "me" or "mine" in combination with depressed emotional and cognitive cues[30]. The idea is to use emotional language markers to train the algorithm, which should make it able to recognize different emotional states. Figure 26 shows how different words are classified and labeled as a marker for a certain emotion. We have examples of

depressed mood and loneliness, but this kind of classification can be made for almost every word in the dictionary.

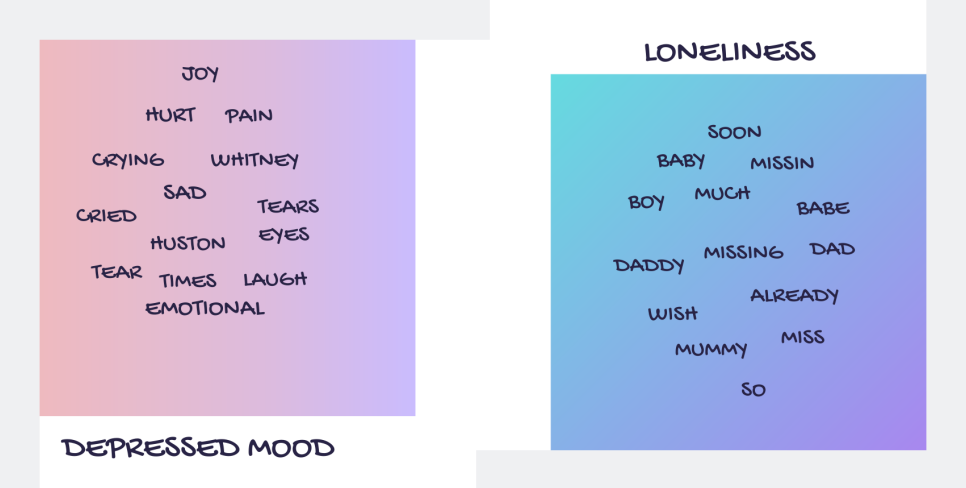


Figure 26: Emotion markers examples (Illustrated by me)

4.2. The data

Collecting the data and creating data sets can take a lot of effort and time if done manually. It expects a lot of research and doing manual checking of data to be sure that we are using the right data to teach the algorithm. This is the main reason why researchers mostly use data sets that are already available online and free to use. Even after finding data sets that match our requirements, it is very important to check where and when was this data collected and what was it used for.

Second, very important step, is knowing the difference between training, validation and test data sets. If you were to use the same data for training and testing, test results would be perfect meaning that it would appear that the model has no flaws and does not have over-fitting problems. Training is usually iterative in order to have the model ready to do a, more or less, solid job. This does not mean there is no correction needed, but simply means that the model can use a certain number of variations of input examples, which can be used to train it and make it even better[31]. This process is divided in three steps:

1. **the model examines the data** - this step corresponds to the training data set. Training data set includes the set of inputs to train the model on, which is done by adjusting the parameters.

2. **the model learns from its mistakes** - this is the evaluation process. The validation data set is used in order to train the model and to keep it as precise as possible. This will give us model with tuned parameters based on the evaluation results.

3. **conclude on how well the model performs** - this step can be looked at as the last evaluation after the training phase has finished, and is done to check the generalization of the model.

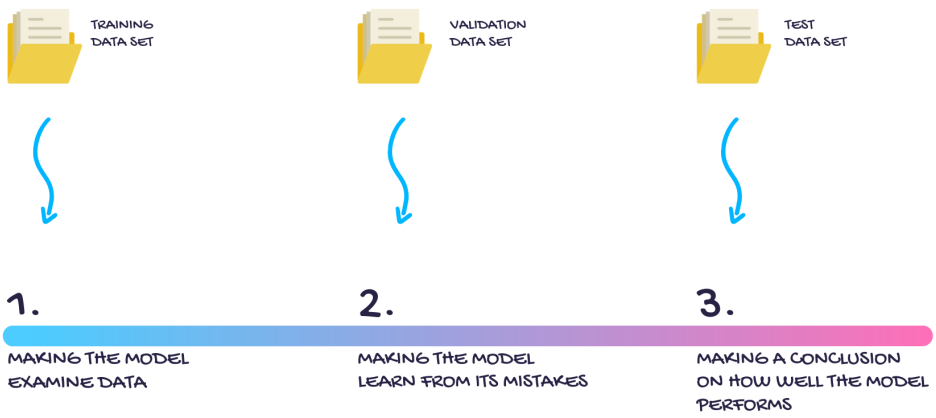


Figure 27: Model training steps (Illustrated by me)

The crucial decision is related to the source of the data. Should we collect data ourselves or should we used previously collected data that is available online? I would say it very much depends on the research you are doing and on the credibility of the author of the data. I have been doing my research, reading many articles, and have found some data sets that would be useful for my research. The data set that stood out was Go, Bhayani, and Huang data set. This data set is a collection of 1 600 000 labeled twitter posts formatted into a .csv record consisting of 6 comma-separated fields. Going from left to right the fields are:

- **target** - represents polarity of the tweet (0 = negative, 2 = neutral, 4 = positive),
- **id** - tweet id (e.g. 1002),
- **date** - the date when the tweet was published,
- **flag** - the query flag,
- **user** - the user that tweeted the tweet and
- **text** - the tweet.

The figure 28 shows a small piece of data, in order to give you a better picture of their structure. If you look at the target column, you can notice only values "0" and "4", and this is because neutral data will not be used for training of the algorithm. Next column shows the id of a tweet. Date column shows full date stamp, and the flag shows "NO_QUERY" for every entry, since no queries were used while collecting this data. User column shows user handles of people posting the tweet. Lastly, sixth column contains tweet text that is used to determine whether or not a tweet is depressing. Shown below is a small portion of the complete data set. Positive portion of this data set is used in creation of Viritas data set, in combination with depressed tweets collected directly from Twitter.

DATA SET

TARGET	ID	DATE	FLAG	USER	TEXT
0	1467812784	Mon Apr 06 22:20:20 PDT 2009	NO_QUERY	bayofwolves	@smarrison i would've been the first, but i didn't have a gun. not really though, zac snyder's just a doucheclown.
0	1467812799	Mon Apr 06 22:20:20 PDT 2009	NO_QUERY	HairByJess	@iamjazzyfizzle I wish I got to watch it with you!! I miss you and @iamlilnicki how was the premiere?!
0	1467812964	Mon Apr 06 22:20:22 PDT 2009	NO_QUERY	lovesongwriter	Hollis' death scene will hurt me severely to watch on film wry is directors cut not out now?
0	1467813137	Mon Apr 06 22:20:25 PDT 2009	NO_QUERY	armotley	about to file taxes
0	1467813579	Mon Apr 06 22:20:31 PDT 2009	NO_QUERY	starkissed	@LettyA ahh ive always wanted to see rent love the soundtrack!!
0	1467813782	Mon Apr 06 22:20:34 PDT 2009	NO_QUERY	gi_gi_bee	@FakerPattyPattz Oh dear. Were you drinking out of the forgotten table drinks?
0	1467813985	Mon Apr 06 22:20:37 PDT 2009	NO_QUERY	quanvu	@alydesigns i was out most of the day so didn't get much done
0	1467813992	Mon Apr 06 22:20:38 PDT 2009	NO_QUERY	swinspeedx	one of my friend called me, and asked to meet with her at Mid Valley today...but i've no time *sigh*
0	1467814119	Mon Apr 06 22:20:40 PDT 2009	NO_QUERY	cooliodoc	@angry_barista I baked you a cake but I ated it
0	1467814180	Mon Apr 06 22:20:40 PDT 2009	NO_QUERY	viJILLante	this week is not going as i had hoped
0	1467814192	Mon Apr 06 22:20:41 PDT 2009	NO_QUERY	Ljelli3166	blagh class at 8 tomorrow
0	1467814438	Mon Apr 06 22:20:44 PDT 2009	NO_QUERY	ChicagoCubbie	I hate when I have to call and wake people up
0	1467814783	Mon Apr 06 22:20:50 PDT 2009	NO_QUERY	KatieAngell	Just going to cry myself to sleep after watching Marley and Me.
0	1467814883	Mon Apr 06 22:20:52 PDT 2009	NO_QUERY	gagoo	im sad now Miss.Lilly
4	1990538859	Mon Jun 01 05:07:58 PDT 2009	NO_QUERY	shukisin	@iamjoellee lols exercise is good for you.
4	1990538892	Mon Jun 01 05:07:58 PDT 2009	NO_QUERY	Marianna26	@omfgitsella idk if were gonna hang out hahaha .... but we need too
4	1990538919	Mon Jun 01 05:07:58 PDT 2009	NO_QUERY	reichan	after all, it's monday's nite karaoke time for my landlady
4	1990538928	Mon Jun 01 05:07:58 PDT 2009	NO_QUERY	serkantoto	Checking out Bing. It's better than Cuil at least.
4	1990538940	Mon Jun 01 05:07:58 PDT 2009	NO_QUERY	matthewbuckland	the new cricinfo.com looks great ... but kind of miss the old design
4	1990538954	Mon Jun 01 05:07:58 PDT 2009	NO_QUERY	GMTriffid	Joy o joy!!! Another late night Enjoy it Skip !
4	1990539018	Mon Jun 01 05:07:59 PDT 2009	NO_QUERY	georgiawonder	@giagia Just had a third nephew myself on Saturday
4	1990539105	Mon Jun 01 05:08:00 PDT 2009	NO_QUERY	fm014	Just had some chocolate muesli... waiting for the happiness hormones to kick in ... then back to work
4	1990539109	Mon Jun 01 05:08:00 PDT 2009	NO_QUERY	jpoh	@Rob_Banks hah yes, that coffee table on a bath tub is cool
4	1990539124	Mon Jun 01 05:08:00 PDT 2009	NO_QUERY	Nijnana	Sunbathing (#)! Hi hurry up, I want my new cellphone
4	1990539145	Mon Jun 01 05:08:00 PDT 2009	NO_QUERY	nananadelonge	thanks God for todaaaaaay!!!!!!! <3

Figure 28: A part of the 1 600 000 data set (Illustrated by me)

I divided the data set provided by Virita into training and test data sets. Using BigML, the data set was divided into training part which is 85% of whole data set and the rest is test data set. Figure 29 shows the structure, first column being the tweet text and the second column

being the label.

trainData

just had a real good moment. i missssssssss him so much,	positive
is reading manga http://plurk.com/p/mzp1e	positive
@comeagainjen http://twitpic.com/2y2lx - http://www.youtube.com/watch?v=zoGfqvh2ME8	positive
@lapcat Need to send 'em to my accountant tomorrow. Oddly, I wasn't even referring to my taxes. Those are supporting evidence, though.	positive
ADD ME ON MYSPACE!!! myspace.com/LookThunder	positive
so sleepy. good times tonight though	positive
@SilkCharm re: #nbn as someone already said, does fiber to the home mean we will all at least be regular now	positive
23 or 24.√Ø¬ø¬ΩC possible today. Nice	positive
Good morning everybody!	positive
Finally! I just created my WordPress Blog. There's already a blog up on the Seattle Coffee Community ... http://tinyurl.com/c5uufd	positive
kisha they cnt get over u til they get out frm under u just remember ur on top	positive
@nicolerichie Yes i remember that band, It was Awesome, Will you please reply	positive
I really love reflections and shadows	positive
@blueaero ooo it's fantasy? i like fantasy novels will check it out	positive
@rokchic28 no probs, I sell nothing other than my blog http://snedwan.com I'll have to get a listen to your band, on iTunes?	positive
@shipovalov "NOKLA connecting people" ?? ??????? ???????	positive
Once again stayed up to late and have to start too early It is a good thing I like my job	positive

Figure 29: Final data set (Illustrated by me)

4.3. Problems identified

As with any other algorithm development, I ran into some problems that stopped me from moving forward. First, and the biggest problem, was finding a data set that has labeled data in order to use it for training the algorithm. There are thousands of data sets available online, but almost most were useless for my case. Even after finding some data sets that are relevant to the subject and could be used in order to train my algorithm, authors of the data sets often do not give the information about meanings of different fields in the data set files. This was, for example, a big issue with Eichstaedt's data sets. Although they contain very useful data, his files contain many fields that are not well specified and create a big problem when using them because they may be misinterpreted by an individual user.

Second problem is finding a data set that is big enough, and therefore useful for the implementation. Data sets that are too small can be used to train the algorithm, but the question is how accurate this algorithm will be and can the final product be used to classify posts if its accuracy is not great. With problems like this, accuracy level should be around 90%.

5. Implementation

To implement and train the model, I used Python 3.6 [34], NLTK platform [35] for development of a model, Flask [36] for backend service and BigML [37] for dealing with data. NLTK is a platform for building Python programs to work with human language data. I will be using Bayesian model to train the algorithm, since I have the data that can be used to calculate probabilistic values and therefore can be used for prediction of depression in social media posts. Naive Bayes, in my examples imported from NLTK, is one of the most commonly used methods for sentiment classification, but has some weaknesses. It can often assume independencies between dependant words, but despite that has a very good performance .My idea of implementation and training of the model is to use data sets in order to train the model to recognize depressive statements and to turn this algorithm into API for a Angular application. This application has the input field and calls the method for calculating the sentiment, and after the calculation finishes it returns "Depressed" or "Positive" result.

5.1. Simple example

To better explain the learning process and the process of classification, I created a simple implementation. Listing 5.1. shows the import, first step, we need to do in order to be able to use the NTLK tools.

```
1 import nltk as n
```

Algorithms 5.1: NLTK import

I have declared two small arrays, one with tweets that have positive sentiment named **positive_tweets**, and one named **negative_tweets**, that contains tweets with negative sentiment.

```
1 positive_tweets = [('happy #charitytuesday @theNSPCC @SparksCharity @SpeakingUpH4H',
      'positive'),
2                    ('thanks God for todaaaaaay!!!!!!!  &lt;3', 'positive'),
3                    ('@Rob_Banks hah yes, that coffee table on a bath tub is cool ','
    positive'),
4                    ('Just had some chocolate muesli...waiting for the happiness
    hormones to kick in..then back to work','positive'),
5                    ('@giagia Just had a third nephew myself on Saturday ','positive'
    ),
6                    ('I feel great this morning', 'positive'),
7                    ('I am so excited about the concert', 'positive'),
8                    ('Happy 38th Birthday to my boo of alll time!!! Tupac Amaru
    Shakur','positive')]
9
10 negative_tweets= [('is upset that he cant update his Facebook by texting it... and
    might cry as a result  School today also. Blah', 'negative'),
11                   ('my whole body feels itchy and like its on fire', 'negative'),
12                   ('Need a hug', 'negative'),
13                   ('Sad, sad, sad. I dont know why but I hate this feeling  I wanna
    sleep and I still cant', 'negative'),
```

```
14                  ('Falling asleep. Just heard about that Tracy girls body being
    found. How sad  My heart breaks for that family.', 'negative'),
15                  ('I do not like this car', 'negative'),
16                  ('is strangely sad about LiLo and SamRo breaking up.', 'negative')
    ,
17                  ('He is my enemy', 'negative')]
```

Algorithms 5.2: Positive and negative tweets

Third array is simply called **tweets** and contains words from positive_tweets and negative_tweets that are longer than three letters and do not contain any other symbols except letters.

```
1
2 tweets = []
3
4 for (words, sentiment) in positive_tweets + negative_tweets:
5     words_separated = [e.lower() for e in words.split() if len(e) >= 3]
6     words_filtered = [w.lower() for w in words_separated if w.isalpha()]
7     tweets.append((words_filtered, sentiment))
8
9 print(tweets)
```

Algorithms 5.3: Filtered words

Array **tweets** now consists of array of words and of the sentiment value for this specific array. Line 9, print(tweets), gives the following result:

```
1  [(['happy'], 'positive'), (['thanks', 'god', 'for'], 'positive'), (['hah', 'that',
     'coffee', 'table', 'bath', 'tub', 'cool'], 'positive'), (['just', 'had', 'some',
      'chocolate', 'for', 'the', 'happiness', 'hormones', 'kick', 'back', 'work'], '
     positive'), (['just', 'had', 'third', 'nephew', 'myself', 'saturday'], 'positive
     '), (['feel', 'great', 'this', 'morning'], 'positive'), (['excited', 'about', '
     the', 'concert'], 'positive'), (['happy', 'birthday', 'boo', 'alll', 'tupac', '
     amaru', 'shakur'], 'positive'), (['upset', 'that', 'cant', 'update', 'his', '
     facebook', 'texting', 'and', 'might', 'cry', 'result', 'school', 'today', 'blah'
     ], 'negative'), (['whole', 'body', 'feels', 'itchy', 'and', 'like', 'its', 'fire
     '], 'negative'), (['need', 'hug'], 'negative'), (['dont', 'know', 'why', 'but',
     'hate', 'this', 'feeling', 'wanna', 'sleep', 'and', 'still', 'cant'], 'negative'
     ), (['falling', 'just', 'heard', 'about', 'that', 'tracy', 'girls', 'body', '
     being', 'how', 'sad', 'heart', 'breaks', 'for', 'that'], 'negative'), (['not', '
     like', 'this', 'car'], 'negative'), (['strangely', 'sad', 'about', 'lilo', 'and'
     , 'samro', 'breaking'], 'negative'), (['enemy'], 'negative')]
```

Algorithms 5.4: Filtered words printed

We need to extract the list of word features from the tweets. This is a list of every distinct word ordered by how frequency it appeared. Functions get_all_words_in_tweets(), get_word_features() and extract_features are used to get the list. Feature extractor is used to determine which words will be used while creating a classifier.

```
1
2 def get_all_words_in_tweets(tweets):
3     all_words = []
```

```
    for (words, sentiment) in tweets:
      all_words.extend(words)
    return all_words

def get_word_features(wordlist):
    wordlist = n.FreqDist(wordlist)
    word_features = wordlist.keys()
    return word_features

word_features = get_word_features(get_all_words_in_tweets(tweets))

def extract_features(document):
    document_words = set(document)
    features = {}
    for word in word_features:
        features['contains(%s)' % word] = (word in document_words)
    return features
```

Algorithms 5.5: Word features

Next step is making a training set. we can use apply_features method from ntlk platform in combination with extract_features method. This method will take tweets and create a training set which can be further used in making a classifier.

```

training_set = n.classify.apply_features(extract_features, tweets)

print(training_set)
```

Algorithms 5.6: Training set

The training set we generated in listing 5.6. is now used on NaiveBayesClassifier train method. The Naive Bayes classifier uses the prior probability of each label which is the frequency of each label in the training set, and the contribution from each feature. If some word appears in 1 of 4 of the positive tweets and none of the negative tweets, this would mean that likelihood of the **'positive'** label is to be multiplied by 0.2.

```

classifier = n.NaiveBayesClassifier.train(training_set)

print(classifier.show_most_informative_features(32))
```

Algorithms 5.7: Classifying

Now we can test our algorithm using classify function. We need to give the function an argument, which is in my case a simple 'I am very happy' statement. After running the algorithm, we get the 'Positive' result, which means this is not a negative statement.

```
tweet = 'I am very happy'

print(classifier.classify(extract_features(tweet.split())))
```

Algorithms 5.8: Testing

5.2. My implementation

In section 5.1. I have shown how, based on small data set (two arrays), model can be train in order to recognize sentiment in simple statements. But, what would happen if we were to use much larger data set? The model ability of recognizing sentiment and its predictions would be more accurate. For training a better model I will use a data set that contains almost nine thousand tweets, positive and depressive.

The algorithm used for training the model is fairly similar to the one previously explained, but has the ability to read from csv files, ability to save the model and to use test data set to test its accuracy. Since I want to use the trained model over and over again, I will not be running this script every time, since the training of the algorithm takes a long time. In order to have fast responses, I saved the model and can use it without having to train it again.

5.2.1. Backend side

```
import nltk as n
import csv
from nltk.stem import PorterStemmer
import pickle
import string

all_tweets = []

tweets = []

with open('trainingSet.csv') as csv_file:
    csv_reader = csv.reader(csv_file, delimiter=';')
    line_count = 0
    for row in csv_reader:
        all_tweets.append((row[0], row[1]))


for (words, sentiment) in all_tweets:
    words_separated = [e.lower() for e in words.split() if len(e) >= 3]
    table = str.maketrans(dict.fromkeys(string.punctuation))
    new_s = [ws.translate(table) for ws in words_separated]
    hash = [h.replace('#', '') for h in new_s]
    at = [a.replace('@', '') for a in hash]
    words_filtered = [w.lower() for w in at if w.isalpha()]
    stemmer = PorterStemmer()
    words = [stemmer.stem(word) for word in words_filtered]
    tweets.append((words, sentiment))


def get_all_words_in_tweets(tweets):
    all_words = []
    for (words, sentiment) in tweets:
      all_words.extend(words)
    return all_words
```

```
35
36
37 def get_word_features(wordlist):
38     wordlist = n.FreqDist(wordlist)
39     word_features = wordlist.keys()
40     return word_features
41
42
43 word_features = get_word_features(get_all_words_in_tweets(tweets))
44
45
46 def extract_features(document):
47     document_words = set(document)
48     features = {}
49     for word in word_features:
50         features['contains(%s)' % word] = (word in document_words)
51     return features
52
53
54 training_set = n.classify.apply_features(extract_features, tweets)
55
56 print(training_set)
57
58 classifier = n.NaiveBayesClassifier.train(training_set)
59
60
61 print(classifier.show_most_informative_features(5))
62
63 f = open('depression_classifier.pickle', 'wb')
64 pickle.dump(classifier, f)
65 f.close()
66
67 test_tweets = []
68 test_tweets_filtered = []
69
70 with open('test.csv') as csv_file:
71     csv_reader = csv.reader(csv_file, delimiter=',')
72     line_count = 0
73     for row in csv_reader:
74         test_tweets.append((row[0]))
75
76 for words in test_tweets:
77     words_separated = [e.lower() for e in words.split() if len(e) >= 3]
78     words_filtered = [w.lower() for w in words_separated if w.isalpha()]
79     words_joined = " ".join(str(x) for x in words_filtered)
80     test_tweets_filtered.append(words_joined)
81
82
83 for tweet in test_tweets_filtered:
84     print(tweet)
85     print(classifier.classify(extract_features(tweet.split())))
```

Algorithms 5.9: Training and testing the model

Since I wanted to use the model with an Angular app, I needed to create a server side of the application. I created a rest API which has a function called **sentiment** that takes text as an argument. This text is then being classified using trained model, which is loaded using pickle. There are two more methods, but those methods are strictly related to communication between server and client side through browser.

```
from flask_cors import CORS;
from flask import Flask, request, jsonify, make_response;
import pickle
import simple

app = Flask(__name__)


CORS(app)


f = open('depression_classifier.pickle', 'rb')
classifier = pickle.load(f)
f.close()


def sentiment(text):
    result = classifier.classify(simple.extract_features(text.split()))
    return result


def preflight():
    response = make_response()
    response.headers.add("Access-Control-Allow-Origin", "*")
    response.headers.add('Access-Control-Allow-Headers', "*")
    response.headers.add('Access-Control-Allow-Methods', "*")
    return response


def _corsify_actual_response(response):
    response.headers.add("Access-Control-Allow-Origin", "*")
    return response


@app.route("/", methods=['GET'])
def index():
    return "Welcome to my Masters Thesis"


@app.route("/api/sentiment/<string:text>", methods=['POST', 'OPTIONS'])
def api_get_sentiment(text):
    if request.method == 'OPTIONS':
        return preflight()
    elif request.method == 'POST':
        new = text.replace('_', ' ')
        print(new)
        snt = sentiment(new)
```

```
48          return _corsify_actual_response(jsonify(snt))
49      else:
50          raise RuntimeError("something wrong")
51
52
53  if __name__ == '__main__':
54      app.run(debug=True)
```

Algorithms 5.10: Server side

5.2.2. Front end side

Since I wanted to be able to better show how the algorithm works, I have developed an Angular web application which uses backend created with Flask. This backend accepts the input text from web, sends it to the model which uses classify function in order to calculate if the text is depressed or positive. As visible on figure 30., this web application is very simple. It has one input field which is used for typing in the text we want to check, and by clicking on 'check sentiment' button we get the result printed beneath the input field (visible on figure 31). In the code shown below, we can see that I created a service in an Angular application that communicates with the backend in Python. Later, this service is used in the ModelView part of the application where by calling the service method we get a response that we can display on a View.

```
1  import { Injectable } from '@angular/core';
2  import {Observable, of} from 'rxjs';
3  import {HttpClient, HttpHeaders} from '@angular/common/http';
4  import {tap} from 'rxjs/operators';
5
6  export interface Sentiment {
7    text: string;
8    result: string;
9  }
10
11 export interface ISentimentService {
12   calculateSentiment(input: string): Observable<Sentiment>;
13 }
14
15
16 @Injectable({
17   providedIn: 'root'
18 })
19 export class SentimentTestingService implements ISentimentService {
20   calculateSentiment(input: string): Observable<Sentiment> {
21     return of({
22       text: 'I am really depressed',
23       result: 'Depressed'
24         }
25     );
26   }
27 }
28
29 @Injectable({
```

```
  providedIn: 'root'
})
export class SentimentService implements ISentimentService {

  private apiUrl = 'http://127.0.0.1:5000';
  private sentiment: Sentiment;
  headers: {
    'Origin': '*',
    'Access-Control-Allow-Headers': '*',
    'Access-Control-Allow-Origin': '*',
  };

  constructor(
      private http: HttpClient
  ) { }

  calculateSentiment(input: string): Observable<Sentiment> {
    console.log(input);
    const re = new RegExp(' ', 'g');
    const str = input.replace(re, '_');
    console.log(str);

    return this.http.post<Sentiment>(this.apiUrl + '/api/sentiment/' + str, {
    input
  }, {
      headers: new HttpHeaders(this.headers)
    }).pipe(
        tap(result => {
        this.sentiment = result;
      })
    );
  }
}
```

Algorithms 5.11: Client side (Angular service - Model)

```
import { Component, OnInit } from '@angular/core';
import {FormControl, FormGroup, Validators} from '@angular/forms';
import {Sentiment, SentimentService} from '../services/sentiment.service';

@Component({
  selector: 'app-sections',
  templateUrl: './sections.component.html',
  styleUrls: ['./sections.component.css']
})
export class SectionsComponent implements OnInit {
  focus;
  focus1;
  input: string;
  hidden: boolean;
  sentimentForm = new FormGroup({
    text: new FormControl('', Validators.required)
  });
  private sentiment: Sentiment;
```

```
  constructor(
      private  service: SentimentService
  ) { }

  ngOnInit() {
    this.hidden = true;
  }

  getSentiment() {
    this.hidden = false;
    this.input = this.sentimentForm.controls.text.value;
    console.log(this.input);
    this.service.calculateSentiment(this.input).subscribe(result => {
      console.log(result);
      this.sentiment = result;
    });
  }

}
```

Algorithms 5.12: Client side (Angular .ts - ViewModel)

```
<section class="section pb-0 section-components">
  <div class="container mb-5">
    <div class="mb-3">
      <small class="text-uppercase font-weight-bold">Input text you want to test</small>
    </div>
    <div class="row" [formGroup]="sentimentForm">
      <div class="col-lg-4 col-sm-6">
        <div class="form-group">
          <input type="text" formControlName="text" placeholder="" class="form-control" />
        </div>
      </div>
    </div>
    <button class="btn btn-1 btn-outline-primary" type="button" (click)="getSentiment()">Check sentiment</button>
  </div>
</section>
<section class="section section-components">
  <div class="col-lg-6 mt-5 mt-lg-0" [hidden]="hidden" *ngIf="sentiment">
    <ngb-tabset [justify]="'center'" class="custom-tab-content flex-column flex-md-row" type="pills">
      <ngb-tab title="{{sentiment}}">
        <ng-template ngbTabContent>
          <p class="description">{{input}}</p>
        </ng-template>
      </ngb-tab>
    </ngb-tabset>
  </div>
</section>
```

Algorithms 5.13: Client side (Angular .html - View)

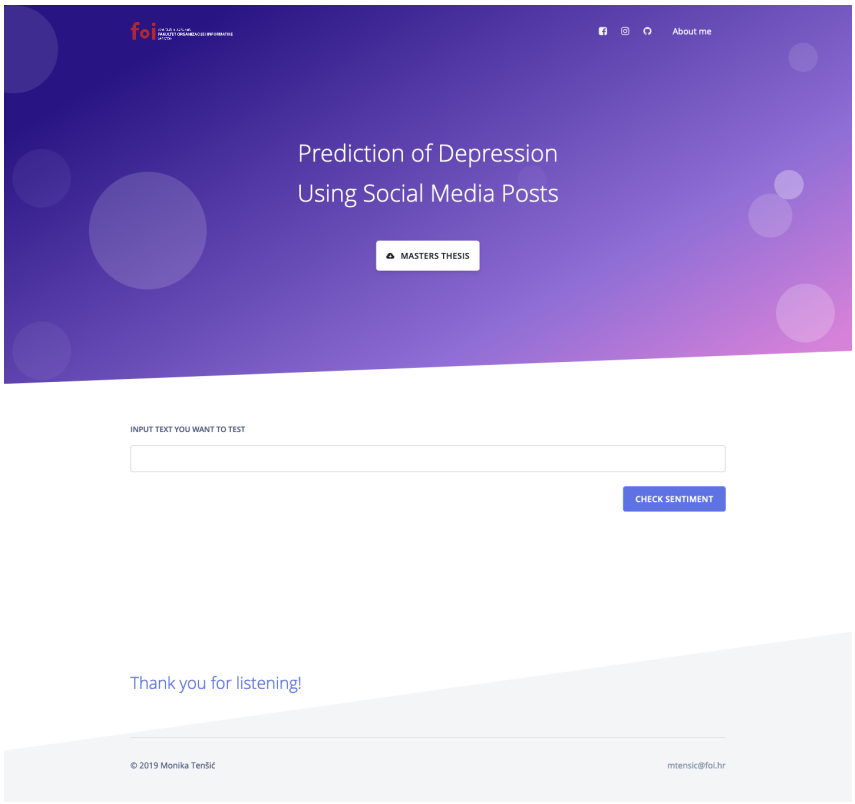


Figure 30: Web page (Illustrated by me)

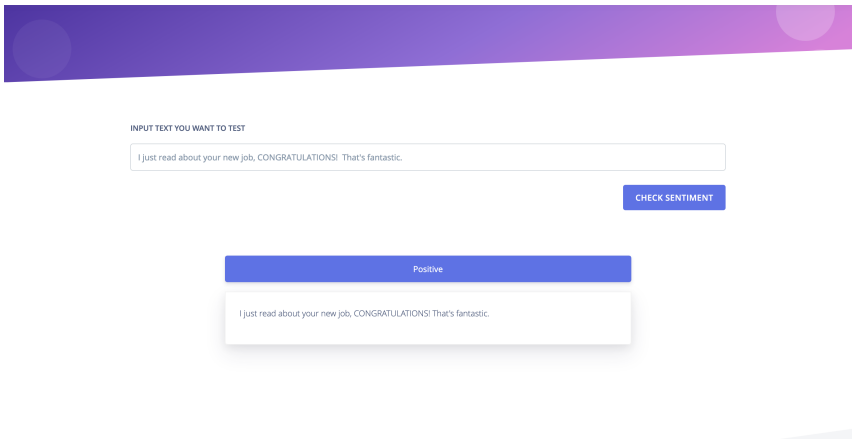


Figure 31: Web page - result (Illustrated by me)

Testing how the algorithm works and if it can classify the tweets correctly, I have notices it has problems while identifying certain statements. Looking into the problem I have realised it is a problem with the size of a data set. If the data set would be bigger, the algorithm would be more accurate and the mistakes of classifying would be smaller in numbers.

6. Conclusion

Doing lots of research and reading many different articles I have learned that machine learning is very appreciated in health communities and have seen numerous other applications of machine learning that were not in my mind up to that point. As I began to develop a model training algorithm, I realized that there were a number of different aspects to look at. From word length, to the meaning of particular words in different situations. For example, if the word "sad" always appears in a positive context in your data set, such as "I haven't been sad for years! I'm so happy!", your model will categorize that word into positive words and while calculating the sentiment of some sad text it will contribute a positive segment to the overall sentiment of the sentence.

While this is a pretty big problem, I find that it can be solved if much larger data sets are used. Since I took only a few positive tweets and few negative ones for the algorithm from section 5.1. Simple example, this model was relatively inaccurate in its estimations. It could recognize simple sentences and calculate the sentiment, but it wasn't until I used a much larger set of data and trained the new model that I received a relatively accurate tool to determine if the sentence was depressing or positive.

Given that owners of large social networks have huge databases, I find that they should use statuses and tweets to create data sets. Since no personal user information is required to train the algorithm, this would be a great way to create a very accurate model. Each status area or tweet should use such a model before posting to interpret whether the text is depressing, aggressive, offensive, or similar, and in these cases provide the user with some other actions. The purpose of this would be to reduce depression in social media users, both by suggesting different ways to help with depression and by reducing bullying and bullying.

To sum up, given all that can be done using machine learning algorithms, I find that too little is being done to improve quality of human lives.

Bibliography

[1] Depression and B. S. Alliance, *Depression statistics*. [Online]. Available: https://www.dbsalliance.org/education/depression/statistics/.

[2] R. E. Noble, "Depression in women", *Metabolism - Clinical and Experimental*, vol. 54, no. 5, pp. 49–52, May 2005, ISSN: 0026-0495. DOI: 10.1016/j.metabol.2005.01.014. [Online]. Available: https://doi.org/10.1016/j.metabol.2005.01.014.

[3] R. C. Kessler, P. Berglund, O. Demler, R. Jin, K. R. Merikangas, and E. E. Walters, "Lifetime Prevalence and Age-of-Onset Distributions of", *Arch Gen Psychiatry*, vol. 62, no. June, pp. 593–602, 2005, ISSN: 17238617. DOI: 10.1001/archpsyc.62.6.593. arXiv: arXiv:1011.1669v3. [Online]. Available: http://archpsyc.jamanetwork.com/article.aspx?doi=10.1001/archpsyc.62.6.593.

[4] K. F. F. Rabah Kamal, "12-month prevalence of any mental illness among adults ages", Tech. Rep., 2017. [Online]. Available: https://www.healthsystemtracker.org/chart-collection/current-costs-outcomes-related-mental-health-substance-abuse-disorders/%7B%5C#%7Ditem-phobias-major-depression-common-mental-health-disorders-among-adults-u-s.

[5] M. R. Hannah Ritchie, *The share of population with depression*, 2018. [Online]. Available: https://ourworldindata.org/mental-health.

[6] WHO, *No Title*, 2017. [Online]. Available: https://www.who.int/mental%7B%5C_%7Dhealth/prevention/suicide/suicideprevent/en/.

[7] U. Dictionery, *Urban dictionery*, 2019. [Online]. Available: https://www.urbandictionary.com/.

[8] A. Vakil, *Functional programming*, 2015. [Online]. Available: https://www.youtube.com/watch?v=e-5obm1G_FY.

[9] S. M. Coyne, A. A. Rogers, J. D. Zurcher, L. Stockdale, and M. Booth, "Does time spent using social media impact mental health?: An eight year longitudinal study", *Computers in Human Behavior*, vol. 104, p. 106 160, Mar. 2020, ISSN: 07475632. DOI: 10.1016/j.chb.2019.106160. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0747563219303723.

[10] G. S. O'Keeffe and K. Clarke-Pearson, "The Impact of Social Media on Children, Adolescents, and Families", *PEDIATRICS*, vol. 127, no. 4, pp. 800–804, Apr. 2011, ISSN: 0031-4005. DOI: 10.1542/peds.2011-0054. [Online]. Available: http://pediatrics.aappublications.org/cgi/doi/10.1542/peds.2011-0054.

[11] V. Vanhoucke, *The Quiet Semi-Supervised Revolution*, 2019. [Online]. Available: https://towardsdatascience.com/the-quiet-semi-supervised-revolution-edec1e9ad8c.

[12] Data61, *Dictionary Learning for Vision - Data61 projects & tools*, 2015. [Online]. Available: https://research.csiro.au/data61/dictionary-learning-for-vision/ (visited on 12/10/2019).

[13] T. J. S. Kenneth Kreutz-Delgado, "Dictionary Learning Algorithms for Sparse Representation", vol. 15, no. 2, pp. 349–396, DOI: 10.1162/089976603762552951.

[14] S. Li, *Anomaly Detection for Dummies*, 2019. [Online]. Available: https://towardsdatascience.com/anomaly-detection-for-dummies-15f148e559c1.

[15] J. Cannady, "Artificial Neural Networks for Misuse Detection", PhD thesis, Nova Southeastern University, 2005. [Online]. Available: http://pld.cs.luc.edu/courses/intrusion/fall05/cannady.artificial%7B%5C_%7Dneural%7B%5C_%7Dnetworks%7B%5C_%7Dfor%7B%5C_%7Dmisuse%7B%5C_%7Ddetection.pdf.

[16] M. Eland, *Neural network*. [Online]. Available: https://res.cloudinary.com/practicaldev/image/fetch/s--kakYFNCR--/c%7B%5C_%7Dlimit%7B%5C%%7D2Cf%7B%5C_%7Dauto%7B%5C%%7D2Cfl%7B%5C_%7Dprogressive%7B%5C%%7D2Cq%7B%5C_%7Dauto%7B%5C%%7D2Cw%7B%5C_%7D880/https://thepracticaldev.s3.amazonaws.com/i/j4igfbcbeafcuwmthyoy.png (visited on 12/09/2019).

[17] L. Vu-Quoc, *Neuron*. [Online]. Available: https://commons.wikimedia.org/wiki/File:Neuron3.png (visited on 12/09/2019).

[18] H. Shi, "Best-first Decision Tree Learning", 2007. [Online]. Available: https://researchcommons.waikato.ac.nz/handle/10289/2317.

[19] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Third. Cambridge: Cambridge University Press, 2000, ISBN: 9780511801389. DOI: 10.1017/CBO9780511801389. [Online]. Available: http://ebooks.cambridge.org/ref/id/CBO9780511801389.

[20] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines", *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, Jul. 1998, ISSN: 1094-7167. DOI: 10.1109/5254.708428. [Online]. Available: http://ieeexplore.ieee.org/document/708428/.

[21] S. Patel, *Chapter 2 : SVM (Support Vector Machine) — Theory - Machine Learning 101 - Medium*, 2017. [Online]. Available: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72 (visited on 11/15/2019).

[22] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian Network Classifiers", *Machine Learning*, vol. 29, no. 2/3, pp. 131–163, 1997, ISSN: 08856125. DOI: 10.1023/A:1007465528199. [Online]. Available: http://link.springer.com/10.1023/A:1007465528199.

[23] Dragons8mycat, *Bayes predictive analysis and why what it can do for your business | The Spatial Blog*. [Online]. Available: https://dragons8mycat.com/2014/04/24/bayes-predictive-analysis-and-why-what-it-can-do-for-your-business/ (visited on 12/10/2019).

[24] Vijini Mallawaarachchi, *Introduction to Genetic Algorithms — Including Example Code*, 2017. [Online]. Available: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3 (visited on 11/19/2019).

[25] P. Imrie, "Virtual Personal Assistants—A different approach to supporting the end user", in *STPIS'17*, 2017. [Online]. Available: https://www.researchgate.net/publication/318462667%7B%5C_%7DVirtual%7B%5C_%7DPersonal%7B%5C_%7DAssistants-A%7B%5C_%7Ddifferent%7B%5C_%7Dapproach%7B%5C_%7Dto%7B%5C_%7Dsupporting%7B%5C_%7Dthe%7B%5C_%7Dend%7B%5C_%7Duser.

[26] M. Chand, *Best Programming Language for Machine Learning*, 2019. [Online]. Available: https://www.c-sharpcorner.com/article/best-programming-language-for-machine-learning/ (visited on 11/25/2019).

[27] R. Fabisiak, *The best programming language for Artificial Intelligence and Machine Learning*, 2019. [Online]. Available: https://medium.com/duomly-blockchain-online-courses/the-best-programming-language-for-artificial-intelligence-and-machine-learning-538486b462c (visited on 11/25/2019).

[28] T. Peters, *PEP 20 – The Zen of Python | Python.org*, 2004. [Online]. Available: https://www.python.org/dev/peps/pep-0020/ (visited on 11/26/2019).

[29] M. De Choudhury, M. Gamon, S. Counts, and E. Horvitz, "Predicting Depression via Social Media", Tech. Rep., 2013. [Online]. Available: www.aaai.org.

[30] J. C. Eichstaedt, R. J. Smith, R. M. Merchant, L. H. Ungar, P. Crutchley, D. Preoţiuc-Pietro, D. A. Asch, and H. A. Schwartz, "Facebook language predicts depression in medical records.", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 115, no. 44, pp. 11 203–11 208, Oct. 2018, ISSN: 1091-6490. DOI: 10.1073/pnas.1802331115. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/30322910%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6217418.

[31] A. Al-Masri, *What Are Training, Validation and Test Data Sets in Machine Learning?*, 2018. [Online]. Available: https://medium.com/datadriveninvestor/what-are-training-validation-and-test-data-sets-in-machine-learning-d1dd1ab09bae (visited on 12/04/2019).

[32] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision", 2009, [Online]. Available: https://www.kaggle.com/kazanova/sentiment140.

[33] R. Virita, *Detecting-Depression-in-Tweets: Detecting Depression in Tweets using Baye's Theorem*. [Online]. Available: https://github.com/viritaromero/Detecting-Depression-in-Tweets (visited on 12/12/2019).

[34] Python Software Foundation, *Welcome to Python.org*. [Online]. Available: https://www.python.org/.

[35] NLTK Project, *Natural Language Toolkit — NLTK 3.4.5 documentation*, 2019. [Online]. Available: https://www.nltk.org/index.html.

[36] A. Ronacher, *Flask - Full Stack Python*. [Online]. Available: https://www.fullstackpython.com/flask.html (visited on 12/12/2019).

[37] *BigML*, 2019. [Online]. Available: https://bigml.com/.

List of Figures

1. 12-month prevalence of major deppresive episode among adults (Created according to: Rabah Kamal, 2017) . . . 3
2. Share of the population with depression (Source: Hannah Ritchie, 2018) . . . 4
3. Suicide rates (Created according to: WHO, 2017) . . . 4
4. Social media (Illustrated by me) . . . 5
5. Machine learning (Illustration made by me) . . . 8
6. Coordinate system (Illustration made by me) . . . 11
7. Coordinate system after translation (Illustration made by me) . . . 12
8. Clusters (Illustration made by me) . . . 12
9. Labeled data (Illustration made by me) . . . 13
10. Unlabeled and labeled data (Illustration made by me) . . . 13
11. Reinforcement learning (Illustration made by me) . . . 15
12. Feature learning (Illustration made by me) . . . 16
13. Dictionary learning (Data61, 2015) . . . 17
14. Anomaly detection (Illustration created by me) . . . 18
15. Association rule learning (Illustration created by me) . . . 19
16. Artificial neural network (Illustrated using Eland and Vu-Quoc) . . . 21
17. Decision tree (Illustrated by me) . . . 21
18. Low regularization value (Illustrated by me) . . . 22
19. High regularization value (Illustrated by me) . . . 23
20. Bayes and decision tree (Dragons8mycat) . . . 23
21. Python pros and cons (Illustrated by me) . . . 27
22. R pros and cons (Illustrated by me) . . . 27
23. JavaScript pros and cons (Illustrated by me) . . . 28

24. Java pros and cons (Illustrated by me) 28
25. Lisp pros and cons (Illustrated by me) 29

26. Emotion markers examples (Illustrated by me) 31
27. Model training steps (Illustrated by me) 32
28. A part of the 1 600 000 data set (Illustrated by me) 33
29. Final data set (Illustrated by me) 34

30. Web page (Illustrated by me) 44
31. Web page - result (Illustrated by me) 44

Table of contents

5.1. NLTK import 35
5.2. Positive and negative tweets 35
5.3. Filtered words 36
5.4. Filtered words printed 36
5.5. Word features 36
5.6. Training set 37
5.7. Classifying 37
5.8. Testing 37
5.9. Training and testing the model 38
5.10. Server side 40
5.11. Client side (Angular service - Model) 41
5.12. Client side (Angular .ts - ViewModel) 42
5.13. Client side (Angular .html - View) 43