OULU UNIVERSITY OF
APPLIED SCIENCES

Ravindu Rajatheva

Application and Business Use Cases of Machine Learning

Application and Business Use Cases of Machine Learning

Ravindu Rajatheva
Thesis
Spring 2019
Business Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Business Information Technology

Author: Ravindu Rajatheva
Title of Bachelor´s thesis: Application and Business Use Cases of Machine Learning
Supervisor: Teppo Räisänen
Term and year of completion: Spring and Year 3 Number of pages: 40

Regarding my background as a student of OAMK Business Information Technology I have learnt several programming languages and their uses. However, I have not been taught machine learning nor have I been familiar with what it can achieve. Being both a technology and business student I think it is important to teach OAMK students machine learning since it will teach them about modern technologies as well help them in the business side by aiding decision making. This thesis will present how simple it is to implement machine learning as well to demonstrate its effectiveness by using real life examples. It is to motivate the reader that they can use machine learning as well with little to no experience and to highlight the benefits gained from the experience. Examples of machine learning models will be coded and run in python language since it has many resources as well as support for machine learning. The models covered in the thesis are linear regression, K Nearest Neighbors, and decision tree classifier. The list of concepts will include overfitting and underfitting, supervised and unsupervised machine learning, and classification and regression. The main focus of this thesis will be supervised learning. After these models have been coded, examples of them used in real life will be discussed.

The resulting scores from the models created have been relatively high therefore it can be understood that machine learning is easy to implement and beneficial to research upon. The models use can also be extended to uses case found in real life such as selecting the best tenant or estimating the price of a used car. However, it must be reiterated that the models as well as the process of machine learning in this thesis is simple and will not reflect real life situations. But a lot can be learnt for those who are unfamiliar to machine learning and possibly motivate readers to use machine learning in their whether for personal use or in work life.

Keywords:

Machine Learning

CONTENTS

1 INTRODUCTION TO MACHINE LEARNING

Machine learning is a field in computer science which incorporates computer programs to analyze large amounts of data using mathematical methods such as mean squared distance, or decision trees to classify and to generate predictions based on data. This technology has already been implemented in large companies such as Netflix (Libby Plummer 2017, cited 05.05.2019) to build their recommendation system. However, with increasing processing power and available resources makes machine learning easier to use for a wider audience than before, even for one person.

Though it has gotten easier there are still some hurdles before one can gain the advantages that come with machine learning. Therefore, thesis will show case three simple methods on implementing when it comes to utilizing machine learning. Readers can learn the process and understand how more businesses besides technological companies could gain potential benefits which include optimizing decision-making processes, reducing losses, and overall supporting the growth of a business. How can this be beneficial? This will be explained further on in the thesis with examples and analogies.

1.1 Concepts and Models

The term machine learning has been thrown around the past few years as a buzzword and has been misinterpreted often times. People are led to believe that machine learning is synonymous with automation and artificial intelligence. They are similar but to put it shortly, machine learning comprises of methods that discovers relationships between data by identifying them and classifying them into set categories. For example, siri uses machine learning techniques to accurately hear what an apple user is saying (Audio Software Engineering and Siri Speech Team 2018, cited 05.05.2019) but siri is considered as AI. Moving on, one might ask how does the machine learn? The focus is only on supervised ML where learning refers to induction (Ryszard S. Michalski, Jaime G. Carbonell, Tom M. Mitchell, cited 26.03.2019) which is when a supervisor gives the outcome variable for data used, for example, whether the outcome is positive or negative for diabetes diagnoses.

1.2 Why Machine Learning

First of all, there are many uses and functions possible with a computer; developing games, creating web pages, data analysis, research, calculations and more. With programming comes numerous opportunities, most important, in my opinion in terms of capabilities is machine learning. Since with enough data and an accurate machine learning model one can potentially predict the future, making it is possible for a person to create an algorithm which can recognize numbers in images, or make diagnoses. The most interesting aspect is of machine learning are the possible and already real applications of it which will be discussed further in the thesis.

Secondly, implementing machine learning algorithms have become easier and more convenient. From recognizing numbers to diagnosing diabetes, the feasibility of creating such algorithms have increased. There is no need to have a strong mathematical or programming background to build these programs. In chapters five to seven readers can see how simple the process of using machine learning with python is.

1.3 Why Businesses should pay attention to it

Machine learning has many applications that businesses could utilize, examples of applications are energy output, inventory management, and recommendation systems. Energy output is an issue that gets drastic as energy demands of a business, building, or place increase. A lot of money can be saved if these categories if energy is handled efficiently. According to an article by the verge, google has employed Deepmind to operate Google's servers in order to reduce energy waste. Deepmind specializes in machine learning and artificial intelligence, in this case they used a machine learning algorithm to optimize the power usage in Google's data center. An estimated 15% decrease in energy usage has been calculated after Deepmind's implementation of a ML algorithm (James Vincent 2016, cited 29.03.2019)

Another instance in which Google used Deepmind is in the use of efficient wind farming. Wind farming is a complex task since in order to yield a viable amount of energy one must make too sure operate wind turbines at most opportune moments. Since there can be many times when the energy created is less than the energy used by the wind turbines. Deepmind utilized ML once again but this time to monitor the weather in order to check the best times to turn on wind turbines

so that they can yield maximum amount of energy and to turn them off is there are no high wind currents (Nick Statt, cited 29.03.2019)

1.4 How it can improve a business

At the moment businesses not directly involved with analytics may not see a reason to use machine learning. This could be that there is no previous data gathered to be used by this industry. However, it must be noted that companies such as Amazon and Google are investing heavily into machine learning as can be seen with Deepmind regardless of their losses (Sam Shead, cited 06.05.2019). Though it is possible to make profits with machine learning in large companies, smaller companies should look into to the trends and benefits that comes with ML. While it is unlikely that businesses not involved in analytics would employ ML due to lack of data, business should pay attention to the benefits ML can bring.

If a bank uses characteristics of persons to determine if they will be able to pay a loan back (Jon Walker 2019, cited 06.05.2019), the same principle could be extended in similar settings such as a school. Instead of approving loans based on characteristics an alternative use could be to calculate which students will have problems with exams or keeping up with other students. This could give schools a system to help those who left behind and help them catch with their peers.

2 THEORY

Training experience, choosing the target function, choosing a representation of the target function, and finally, selecting a function approximation algorithm. These are the four principles of machine learning (Tom M. Mitchel 1997, 27.03.2019) however, in this thesis we will focus mainly in principles of training experience and selecting an appropriate machine learning algorithm. The reason for doing so is since there are many algorithms readily available for anyone to use and learn from which are quite reliable. Therefore, the thesis will demonstrate three uses of ML algorithms and for which purposes they could be used for. Meanwhile, the four principles are four more serious situations for example in the situation of creating siri's ability to hear people from a distance (Audio Software Engineering and Siri Speech Team).

Training experience refers to the data and parameters that will be used to train the machine learning model. When creating an algorithm that can predict which patient is most likely to have "X" condition, data in this scenario could include age, gender, weight, blood type, and similar types of attributes. Determining the price of a car? The data can range from the year of the car, years possessed, previous owners, mileage, model, options, crashes and so forth. Certain steps must be considered when choosing which data to process, which attributes should be left out, and if needed, does it need to be transformed. Training experience can also refer to when AI plays against itself and learns how to play a game. This method will be talked about more later on in the thesis. Although, this is still recording data and knowing which data is important. The second step depends on the data and the goal of the algorithm, is it classification or predicting values such as temperature. If the data is linear, where a newer car costs more, then one would employ linear regression to determine the relationship between the attributes and their price outcome. Thus, making it possible for one to accurately predict the price of a car based on its attributes. Or if the data is diagnosing if patient has a condition such as diabetes or not then classification algorithms are used (KNN or decision tree). When the data is gathered and understood (linear, categorical values, clustering), the proper algorithm is selected, leaves only implementation.

2.1 Supervised and Unsupervised Learning

In this thesis supervised machine learning algorithms will be the focus. The main application of unsupervised machine learning is in data analysis and not in generating predictions outside the dataset. In contrast, supervised learning applications consist of image and speech recognition, recommendations systems, weather forecasting, and more. These applications are the motivation and incentive to research supervised machine learning.

In supervised learning there is a further division between algorithms, they are classification and regression algorithms. Before discussing the difference between the two it must be understood that in supervised learning, the data being used contains the outcome variable. For example, let us say that there is a dataset containing emails with the outcome of the email being either "genuine" or "spam", essentially, the answer is in the data. The learning algorithm will be trained based on the features (input data) and then generate predictions (output data). The difference between regression and classification is that predictions in classification algorithms are discrete values such as those mentioned above while in regression, the predicted values are continuous. Continuous values can be temperature observations, or the energy output of machines measured in joules (Mathworks, 2016, section 1, cited 30.03.2019).

There are many training models available and this may seem intimidating initially, since finding the best algorithm is key. But there are some factors to look for when considering a machine learning model. These factors are the type of data, size of the data, and distribution of data. In my opinion distribution of the data is most import since size of data and the type of data is easily visible. Data visualization shows if the data is linear or non-linear which helps to reduce the available training models one would consider employing in their program or project (Mathworks).

2.2 Classification and Regression

Classification and regression are the two categories present in supervised learning. As stated previously, classifications have discrete predictions, "is an email spam or not", whereas regression provides continuous values as predictions such as a temperature reading. In this thesis there will be one example of regression and two of classification (Mathworks).

2.3 Overfitting vs Underfitting

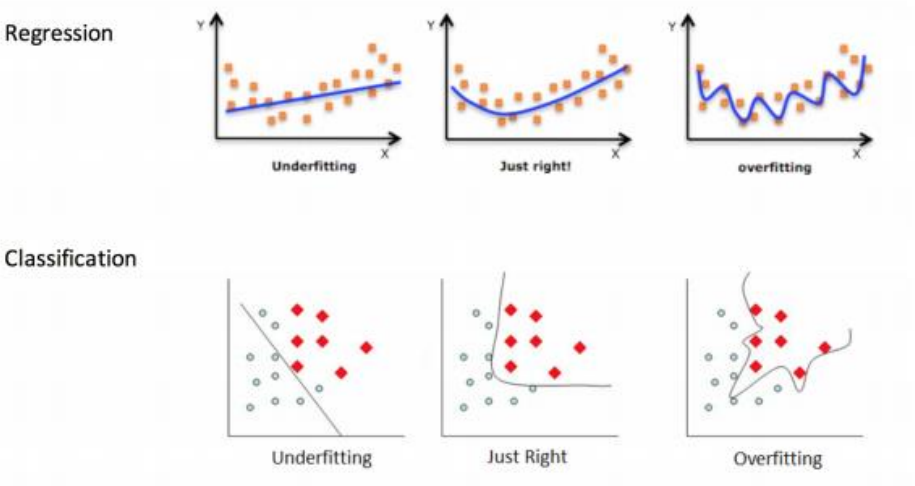

Figure 1 Overfitting and Underfitting

The image above (see 1 in appendix) displays overfitting, it is the term used when a machine learning models has gained bias to a particular training set, and when the model has loss generalizability. When it comes to machine learning there is only so much data, the goal is to find the trend in the data. For example, in figure 1 the "just right" classification grid shows a model which is able to categorize the data in a broad and general manner. In overfitting the split between data is too specific thus when the machine learning model is used to categorize new data it might do so incorrectly. Meanwhile, this problem won't occur during underfitting. Overfitting can occur when there is no testing set for which the model can be scored on. If the model is given a dataset to learn and not any information to test upon, the model will most likely gain a bias in relation to the dataset and therefore be flawed. To solve this problem a simple step can be taken which is dividing the dataset between training and testing. This method can be seen in further chapters and is easily implemented using python.

Underfitting is when the model is not able to produce relationships or gather useful patterns from the data. This can happen for a number of reasons, the machine learning model is poor for the data, the data is either too small in size or is missing a lot of values, or both. Solutions for underfitting include gathering more data, changing the model, or training the data several more times (Udacity 2016, cited 19.11.2018).

3 HISTORY OF MACHINE LEARNING

The first program that learn as it ran was created in 1952, the checkers program by Arthur Samuel. It was a program which looked at the status of boards, player positions and opponent of multiple checkers games and was shown a result of the board's state after a play was made. With the help of supervision, the computer was told which moves were good to play and how to assess a board state. The other method the checker's program used was a lookahead search, where hypothetical positions of board pieces would be mapped from their current positions. Then the paths that lead to the highest score, higher score meaning higher chance of winning, were selected from the hypothetical positions (Samuel's Checkers Player, 2004, cited 02.04.2019). The program was able to perform so well that it beat the Connecticut state champion. This showed great promise in machine learning and what can be achieved (Gio Wiederhold, John McCarthy, Ed Feigenbaum, cited 02.04.2019).

NetTalk is a program that can learn how to pronounce words similar to humans. NetTalk uses English words taken from Webster's pocket dictionary as input and the output are phonemes. The program is able to "talk" is based on the surrounding letters, the context of a word. For example when the input are 5 letter words, the program "sees" the first letter, then the preceding two and then the final two letters. Based on this context the program is able to accurately guess how to pronoune words and thus read English text (Terrence J. Sejnowski. Charles R. Rosenberg, 1986, cited 14.05.2019).

3.1 Modern Machine Learning

With the advancement of technology and rising processing power there are new artificial intelligence programs, some of which are deep blue, alphago, and OpenAI. Deep blue gained its status in popularity after defeating the chess grandmaster in the year 1997, popularizing the man vs machine scene. Alphago created by Deepmind, was able to beat the greatest go player considered by many in 2016. OpenAI is a company and not the program itself, however its name is used when competing against human players in the game called Dota 2. OpenAI was able to beat the top human players in a 1 vs 1 situation in 2017. OpenAI used 5 neural networks to train 5 virtual players to perform actions which are then rewarded if they result in a positive outcome

such as damage to opponents. The neural networks are penalized if there is a negative outcome which would be getting damaged by opponents (Siraj Raval, 2018). Below are more detailed descriptions of the AI technology and their background.

Go is a complex game in which players move stones, either black or white, where each player has to surround the other or capture empty spaces for territory. Though these rules are straightforward the number of combinations possible in go are greater than a googol which is 10 to the power of 100 (The story of AlphaGo so far 2017, cited 20.04.2019).

Unlike Alphago, the Dota2 AI created by OpenAI did not use tree search, which is a method of looking at possibilities of the next move and how to progress onwards. Rather, OpenAI made the program learn Dota2 from scratch and made the program play games with itself. The program garnered more attention as a new version called 5, being the total amount of neural networks used to compete against 5 human players. The program competed against some of the best human players of Dota2 and won (OpenAI Five Benchmark: Results 2018 cited 25.04.2019). The program used reinforcement learning where actions that resulted in positive outcomes, i.e. damage to opponents, were rewarded while negative actions were discouraged (Siraj Raval). Though that is just one side of the whole program, there is also learning the moves of the character's, which moves to use and when would be the optimum time to use them. Overall it was quite the achievement and pushed the limits of what AI can achieve.

3.2 Python

Python 1.0 was a programming language released in January 1994 created by Guido Van Rossum. The first python code was developed in February 1991 it had core functionalities such as list, strings and few other data types. Additionally, it was an object-oriented langue and it had a module system. In 2000, python 2.0 was released which supported Unicode and other notable functions. The current version of python is version number 3, the latest being 3.7 (History of Python 2018, cited 06.11.2018).

This thesis will demonstrate how simple and how effective machine learning programs can be created in python without the need of prior knowledge. Python is already known to contain many

dependencies or libraries which are very helpful for completing an array of tasks including machine learning, data analysis, and data visualization. There are even web development features in python with UI design support as well, additionally for each of these functions there are plenty of resources created by python users and developers alike. To those new in programming a library is a resource with precompiled routines which programs are able to use. Routines and functions are stored in a library whereby a user can call or reference them (Techopedia, cited 13.05.2019).

```
import pandas

df = pandas.read_csv('Credit_Approval.txt')
```

Figure 2 Calling a function

The image above (figure 2) is an example on how to call upon a library's functions. First, a user will import the specified library, then to use the library functions simply type the library's name with a dot to refer to the functions. In the above case the function being called is "read_csv" which does as implied, the function reads input from a csv file (comma separated variables). In the continuing chapters of the thesis there will be images displaying csv files.

Python Libraries

The first library to be discussed is Pandas. Pandas is an open source and complex data analysis tool with the additional ability to create data structures as well (Pandas, cited 15.11.2018). As can be seen from figure 1 pandas is used to read data from csv files. Numpy is a commonly used package useful for linear algebra, creating arrays, and other capabilities (Scipy Org, cited 15.11.2018). Using pandas makes it easier for one to pull values from a dataset and then using numpy to transform the input into an array. This method can be seen further along the thesis since the data which models train on, are located in csv files. Matplotlib is a plotting library that produces graphs, histograms, and other figures representing data (Matplotlib, cited 15.11.2018). Matplotlib makes it easy and convenient to show data points along with trend lines created by machine learning models. Sklearn or scikit learn is the library which will be used to import machine learning models such as linear regression and K Nearest Neighbors. Sklearn is built on numpy, scipy and matplotlib. It is also open source and mainly a simple and effective tool for data mining and analysis. (Sklearn, cited 16.11.2018)

4 DATA PREPROCESSING

Data can either be numerical or word values. They can represent almost anything, but data needs to be recorded properly, consistently, and there must be enough or this will lead to insufficient machine learning algorithms. Data in numerical form can be persons' health factors (height, weight, gender, body fat percentage) with the target variable being if the person is diabetic or not. In the chapter describing the KNN algorithm, there will be an image displaying relevant health factors connected with diabetes. Further examples of data include financial data, energy output of machines, CO2, and O2 (sensor data). These are some real-world examples of data which have been used in machine learning; there also exists some practice datasets of sensor data, financial data, and more.

Before using machine learning algorithms, it is important to realize that in real world settings that data is not always smooth, clean, and or free from errors. Instead, data can be noisy, have missing values, and erroneous data. A common example is missing data values.
A list of data preprocessing methods includes cleaning, integration, transformation, and reduction. To clean the data one can, ignore the empty data columns altogether, fill in the empty data with a constant, or a mean value (David Chappell, cited 05.11.2018).

It is necessary to focus on which data is important to use for machine learning. Focusing on the data and its attributes is called feature selection (MathWorks, cited 30.03.2019) Integration of multiple databases may lead up to redundant data attributes such as customer number and customer id. Furthermore, attributes which can be derived from other attributes are also redundant. To search for these closely related attributes correlation analysis is used to minimize redundant data. This is a quite technical but gets more relevant as one delves deeper into data science field (David Chappell).

4.1 Acquiring data

Information is all around us but in some cases the information needed isn't available. When it comes to machine learning gathering data is the first step and these are some websites which are useful for finding data, they are: Kaggle, Reddit, Amazon Web Services, Google, UCI, and KDNuggets (Kunal Jain, cited 30.03.2019). The datasets mainly used in this thesis are from UCI's datasets (UCI Machine Learning, cited 31.03.2019).

Name	Description
Kaggle	A platform where users can find datasets, upload datasets, interact and work with other users so they can build models together. There is also support for answering questions and challenges presented to users.
Quandl	Contains datasets and live data gathered from banks, government and private organizations that are accessible to users who have a premium subscription. Those that don't still have access to numerous datasets which include house prices, material prices, currency prices, and others. Mostly regression datasets.
UCI	University of California Irvine's archive of datasets existing from 1987 that include classification and regression datasets.

Table 1 Websites for Datasets

4.2 Cleaning the Data

An important transformation is min-max normalization which takes the lower end of the input and the higher end and transforms them to a range from 0-1 respectively (Wang Taehyung, cited 14.11.2018, slide 34).

Then, there is data reduction, which is similar to removing data integration, where it is reducing some parts of the data while maintaining the integrity. For example, when analyzing the growth of coffee sales attributes such as places the “coffee has been bought” can be ignored since other attributes as the “time of year”, and “amount of coffee bought” will maintain useful data for the analysis. (Wang, Taehyung)

Data transformation is useful, but it must be carried out thoroughly. During my initial machine learning program, I had made the erroneous decision to convert dates to integers through my unique method of multiplying the dates into seconds, essentially creating large floating point numbers. The machine learning model score using this new data was high, approximately 90% but predictions were false since predictions would generate values ten times bigger than appearing in real life scenarios. The goal of my program was to measure temperature based on attributes related to weather but changing time to floating point numbers was the wrong approach. As time goes on the converted date values grew larger which created a false upward trend and thus the temperature recordings increased as well. The solution to this problem is transforming time so it is not variables that keep increasing but to treat it as a cyclical variable. To change time variables into a cyclical variable one could use sine and cosine transformations, this is useful to know when creating time series (Ian, London, cited 13.05.2019). In classification algorithms, encoding is converting categorical values such as “good”, “bad”, and “neutral” to binary numbers consisting of zeroes and ones. For example, red, blue, and green would be converted to 001, 010, and 100.

5 LINEAR REGRESSION

Linear regression is one of the most common and a fundamental machine learning algorithm. The principles behind linear regression are establishing a linear relationship of the data points using a line of best fit. To determine how the line is created, the sum of squares is used to calculate the squared sum of all data points and their deviation from the mean (Will Kenton, cited 22.04.2019). Linear regression models are used when one needs an algorithm that is easy to understand and fast to implement. They are also great to be used as a base to compare performance with other machine learning algorithms. (Mathworks 2016, cited 22.04.2019) An advantage to linear regression is that it is less likely to overfit than other algorithms, unlike algorithm which use a polynomial fit (Andriy Burkov, cited 22.04.2019).

5.1 Case Linear Regression

The program to be made will to predict the strength of concrete based of features such as: cement mixture, blast furnace slag, fly ash, water amount, plastic compound, aggregates, and age of the mixture (see appendix 2). This data includes the outcome of different overall mixtures and shows each mixtures strength. Having the outcome means it will be supervised machine learning since the result is explicitly stated.

1	Cement	Blast Furnace Slag	Fly Ash	Water	Superplasticizer)	Coarse Aggregate	Fine Aggregate	Age (day)	Concrete compressive strength
2	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	79.99
3	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	61.89
4	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	40.27
5	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	41.05
6	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	44.30
7	266.0	114.0	0.0	228.0	0.0	932.0	670.0	90	47.03
8	380.0	95.0	0.0	228.0	0.0	932.0	594.0	365	43.70
9	380.0	95.0	0.0	228.0	0.0	932.0	594.0	28	36.45
10	266.0	114.0	0.0	228.0	0.0	932.0	670.0	28	45.85
11	475.0	0.0	0.0	228.0	0.0	932.0	594.0	28	39.29
12	198.6	132.4	0.0	192.0	0.0	978.4	825.5	90	38.07
13	198.6	132.4	0.0	192.0	0.0	978.4	825.5	28	28.02
14	427.5	47.5	0.0	228.0	0.0	932.0	594.0	270	43.01
15	190.0	190.0	0.0	228.0	0.0	932.0	670.0	90	42.33
16	304.0	76.0	0.0	228.0	0.0	932.0	670.0	28	47.81
17	380.0	0.0	0.0	228.0	0.0	932.0	670.0	90	52.91
18	139.6	209.4	0.0	192.0	0.0	1047.0	806.9	90	39.36
19	342.0	38.0	0.0	228.0	0.0	932.0	670.0	365	56.14

Figure 3 Concrete Attribute Data

The data used, shown in figure 3, is that of concrete and the goal of the program to be made will be determining the concrete compressive strength (the output variable, prediction variable) based on the listed values (input variables). The data was taken from UCI (University of California Irvine) and it is a relatively simple dataset since all the attributes are numerical and the number of instances is approximately 1000. It must also be noted that there are no missing values in the dataset, however if they were it is enough to replace the empty value with the mean value of the specific category.

```
1  import pandas as pd
2  from sklearn.linear_model import LinearRegression
3  from sklearn.model_selection import train_test_split
4  import xlrd
5  import numpy as np
6
7  df = pd.read_excel('Concrete.xls')
8
9  X = np.array(df.drop(['Concrete compressive strength'],1))
10 y = np.array(df['Concrete compressive strength'])
11
12 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
13
14 model = LinearRegression().fit(X, y)
15 print(model.score(X, y))
16
17 print(model.predict([[540,0,0,150,2.5,980,400,50]]))
```

Figure 4 Linear Regression Program

From lines 1 to 4 in figure 4, the necessary dependencies are imported which included randomizing the data and the linear regression model. On line 7 a data frame is created to read and store the values contained in file “Concrete.xls” which are the values in figure 3. In line 9 on figure 4, the input variables are stored into “X” and the outcome variable is stored in “Y” variable. Randomizing data takes place in figure 4 line 12. The process of “train_test_split” is taking the now randomized data so and splitting into two groups, one to train the model, and the other will be used to compare with the model’s predictions to calculate the accuracy. Line 14 in figure 4 shows the linear regression model is fitted with the values and in line 15 the accuracy of the model can be displayed when the program is executed. The last line in figure uses the prediction function of the model with values that someone may want to see the predicted outcome variable of, in this case, the concrete compressive strength.

```
========= RESTART:
0.638280529831525
>>>
```

Figure 5 Regression Performance

The number displayed above in figure 5 refers to the performance of the program made, how accurate the model used with the parameters given in predicting the outcome of concrete strength. The number is approximately 0.64, or 64% which is not a good degree of accuracy.

```
======== RESTART:
[54.58633387]
>>>
```

Figure 6 Regression Prediction

Figure 6 displays the predicted concrete strength according to the values in line 17 of figure 4. However, since the accuracy of the trained model is quite low the predicted outcome can be dismissed.

5.2 Linear Regression Application

Linear Regression is a broad field in machine learning and is often associated with price forecasting of stocks since it is a common statistical method. As with linear relationships the example above used attributes such as amount of fly ash and age to determine the strength of concrete. This principle need not be applied in complicated matters such as the stock market, more businesses should employ machine learning so that they are able to make better and more informed decisions. Extending the application of the above program, it is possible to use linear regression to calculate the price of a car. This could be an effective method of pricing cars in a used car dealership or even when an individual seller wishes to sell their own car. Based on pre-existing data it is possible to make an accurate price evaluation of one's car. There are several steps involved if one wishes to create a model which can estimate the price of a car. They include gathering the data, cleaning or transforming the data, using an appropriate machine learning model, testing performance of the model, and generating predictions.

A less popular, real-life application of linear regression similar to car valuation is real-estate appraisal. DW Slater Company Real-Estate Appraisal services employ machine learning services in their use to appraise properties. Similar to the example above, they analyze attributes including lot sizes, location, and number of rooms. They then use regression methods to produce an accurate appraisal of a property (Shannon Slater 2016, cited 26.04.2019). The processing of

implementing machine learning is probably related to the car valuation steps listed above. First of all, DW Slater would gather data, in the article it was mentioned that they would check the prices of properties in nearby locations. Then from this data necessary adjustments can be made then using the appropriate algorithm which is linear regression in this case. Finally, it is a matter of optimizing the model using methods where disregarding some data and focusing on selected attributes.

Application Case	Suitability
Estimating prices	Ideal
Financial forecasting	Ideal
Image recognition	Non-ideal
Email Spam	Non-ideal

Table 2 Linear Regression Applicability

6 K NEAREST NEIGHBORS

Besides linear regression there are multiple other popular machine learning algorithms. Unlike linear regression, K nearest neighbor is a classification algorithm. K nearest neighbor algorithm is an instance-based learning algorithm. Meaning when encountering a new instance, observation, or data-point the algorithm will refer to earlier instances to classify the new ones. Unlike other learning algorithms, instance-based learning algorithms have the ability to produce a different approximation of a target function for each instance. This means there is not one approximation of a target function for all data points but an individual one for each data point. This is advantageous when the target function increases in complexity (Tom M. Mitchell, cited 27.03.2019). There are many real-life applications of this algorithm. For example it can be used to predict the likelihood of diabetes.

K nearest neighbors is best used when the dependent variable or the class attribute can take a certain range of numbers. For instance, binary choice as to whether a person has diabetes which is displayed as “1” or doesn’t have diabetes “0”. The class attribute should be discrete values and for this thesis the example code will use datasets where the class attribute will be either binary or contain a small range of numbers.

6.1 Case KNN

The first example used for K Nearest neighbors will be the program determining diabetes in a patient. The first step of this program would be to clean the dataset, however the one acquired has all numeric data, no missing data or missing data has been filled in. Additionally, the dataset is of adequate size with approximately 2000 observations. The dataset itself is taken from a hospital in Frankfurt, Germany that was uploaded to kaggle (see appendix 3).

```
Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin,BMI,DiabetesPedigreeFunction,Age,Outcome
2,138,62,35,0,33.6,0.127,47,1
0,84,82,31,125,38.2,0.233,23,0
0,145,0,0,0,44.2,0.63,31,1
0,135,68,42,250,42.3,0.365,24,1
1,139,62,41,480,40.7,0.536,21,0
0,173,78,32,265,46.5,1.159,58,0
4,99,72,17,0,25.6,0.294,28,0
8,194,80,0,0,26.1,0.551,67,0
2,83,65,28,66,36.8,0.629,24,0
2,89,90,30,0,33.5,0.292,42,0
4,99,68,38,0,32.8,0.145,33,0
4,125,70,18,122,28.9,1.144,45,1
3,80,0,0,0,0,0.174,22,0
6,166,74,0,0,26.6,0.304,66,0
5,110,68,0,0,26,0.292,30,0
2,81,72,15,76,30.1,0.547,25,0
7,195,70,33,145,25.1,0.163,55,1
6,154,74,32,193,29,3,0,839,39,0
```

Figure 7 Diabetes Attribute Data

The image above (figure 7) displays the dataset for creating a simple diabetes detecting program. Although there is only a portion of the dataset, there were no missing numbers or erroneous data within the dataset. It must be reminded that this is a convenient dataset which doesn't accurately represent real life situations. The top row in figure 7 shows the names of the variables and the values are displayed in the bottom. As one can see the values are separated using a comma, these files are called comma-separated variable files. Datasets will be presented in this format throughout the thesis. The class attribute is the "Outcome" and it is represented as "y" and is the dependent variable. The other attributes are what constitutes X and are the independent variables.

```
1  import pandas as pd
2  import numpy as np
3  from sklearn import preprocessing, neighbors
4  from sklearn.model_selection import train_test_split
5
6  df = pd.read_csv('diabetes.csv')
7
8  X = np.array(df.drop(['Outcome'],1))
9  y = np.array(df['Outcome'])
10
11 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
12
13 model = neighbors.KNeighborsClassifier()
14 model.fit(X_train, y_train)
15 score = model.score(X_test, y_test)
16 print(score)
17
18 print(model.predict([[1,98,75,25,50,25,0.145,21]]))
```

Figure 8 KNN Program

Figure 8 displays a simple program which trains a model to predict the likelihood a woman has diabetes based on certain criteria. Lines in figure 8 from 1 to 4 are importing libraries, especially the library sklearn which contains the machine learning model, K Nearest Neighbors. From lines 6 until 9 the independent variables and dependent variable is separated into data frames X and y. The letter "y" is used to represent the variable we are trying to predict the outcome of whether a given woman has or doesn't have diabetes. X are all the independent variables consisting of blood pressure, skin thickness, insulin levels, and others shown in the top row of figure 8. In line 11 80% of the data from figure 8 is used to train the K nearest neighbor model and the remaining 20% is used as new data to make predictions on. The chosen 80% are random data samples from the dataset in order to clear any bias. In lines 13 to 14 in figure 8, the model is referenced from the library on line 3 then the X and y input is fitted into the machine learning model. The model is therefore trained and on line 15 the model score is calculated based on "new" data which is the test data, finally line 16 prints the score. The process of the program starts with data being read from a file as the input for a machine learning model. Based on the data, the model will create a relationship between the independent variables (X) and dependent variable. Then based on the relationship predictions can be generated. A score of accuracy can also be produced to see how well the model can successfully predict if a woman has or doesn't have diabetes.

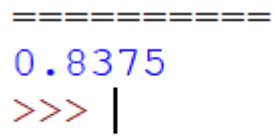

Figure 9 KNN Performance

When line 16 in figure 8 is executed the score is printed as seen on figure 9 which is 0.8375, or 83.75% accuracy. This is a good degree of accuracy for an algorithm when practicing machine learning for a beginner.

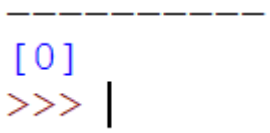

Figure 10 KNN Prediction

The number 0 seen in figure 10 refers to the outcome, the diabetes diagnosis, based on the values listed on line 18 in figure 8. The 0 refers to negative diagnosis of diabetes.

6.2 KNN Application

A study published by Elsevier on the "Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm" (M. Akhil jabbar, B.L. Deekshatulu, Priti Chandra, cited 25.04.2019) is good demonstration of KNN's capabilities and effectiveness. Although the model in the study to classify heart diseases used genetic algorithms paired with KNN, but this is common in real world settings since just one type of algorithm is not often enough. Additionally, using a combination of algorithms is better since it can increase accuracy but might come at a cost since combined models take more time to train and generate predictions.

Results showed that the combined model, with the use of cross-validation, was able to predict with an accuracy of 67.5%. It was stated earlier that 60% is not accurate enough to use to real world settings, however, this depends on the situation and in the medical side there are a lot of factors which lower the possibility to build a highly accurate model. Especially when considered, heart disease contains a lot of risk factors and though there is voluminous data it is still quite a challenge to make accurate predictions based on one's conditions. KNN is used often in the medical field due to the nature of diagnosis where splitting symptoms for example into different categories can help a doctor know which condition is affecting the patient and which isn't.
Below is a table presenting which situations would be appropriate to use KNN and when it would not be (Parikshit, Joshi, cited 13.05.2019).

Application Case	Suitability
Image recognition	Ideal
Diagnosing patients	Ideal
Financial forecasting	Non-ideal
Weather forecasting	Non-ideal

Table 3 KNN Applicability

7 DECISION TREE CLASSIFIER

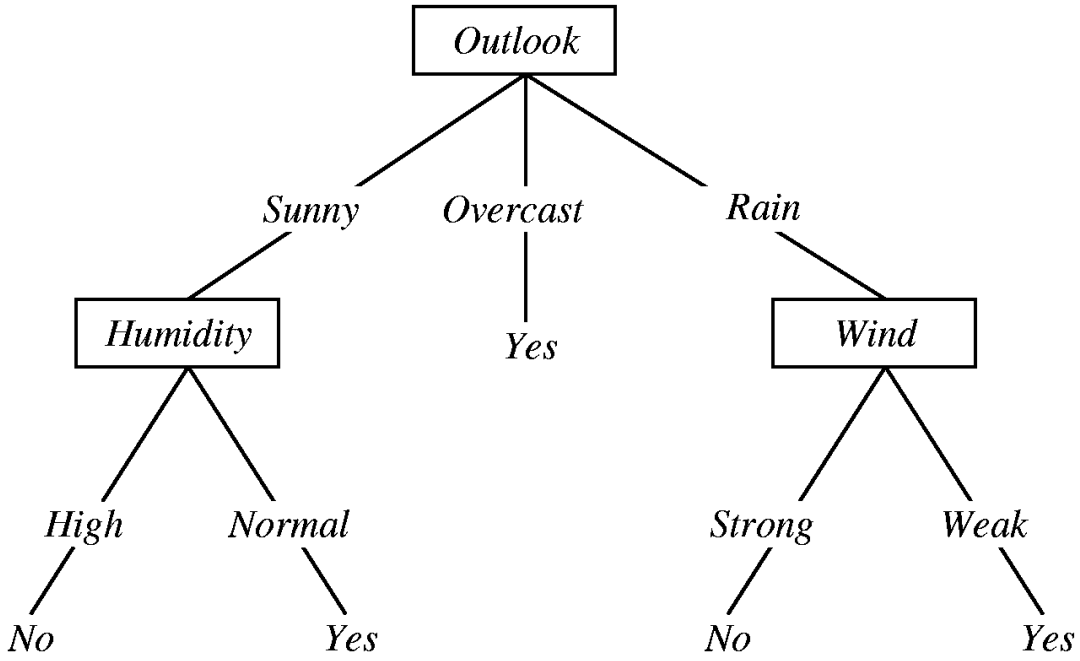

Figure 11 Representation of Decision Tree

The image above (figure 11) shows what decision trees were originally known for, graphical representations of decision making (Benjamin Cohen, cited 25.04.2019). This is still the same principle when it comes to the programming aspect of decision trees but instead probabilities are used and calculated with questions similar to humidity or wind in the image above. These are questions formulated by the decision tree model based on which data, variable, or feature gives the most information (Ben Keen, cited 25.04.2019).

7.1 Case Decision Tree Classifier

The program to be made involves using data gathered from the UCI (see appendix 2) archive to determine if the fertility of a man is normal or altered. The variables present in the data file include season, age, past diseases, accident or trauma, past surgery, high fevers, frequency of alcohol consumption, smoking, hours spent sitting, and the diagnosis outcome.

```
-0.33,0.69,0,1,1,0,0.8,0,0.88,N
-0.33,0.94,1,0,1,0,0.8,1,0.31,O
-0.33,0.5,1,0,0,0,1,-1,0.5,N
-0.33,0.75,0,1,1,0,1,-1,0.38,N
-0.33,0.67,1,1,0,0,0.8,-1,0.5,O
-0.33,0.67,1,0,1,0,0.8,0,0.5,N
-0.33,0.67,0,0,0,-1,0.8,-1,0.44,N
-0.33,1,1,1,1,0,0.6,-1,0.38,N
1,0.64,0,0,1,0,0.8,-1,0.25,N
1,0.61,1,0,0,0,1,-1,0.25,N
1,0.67,1,1,0,-1,0.8,0,0.31,N
1,0.78,1,1,1,0,0.6,0,0.13,N
1,0.75,1,1,1,0,0.8,1,0.25,N
1,0.81,1,0,0,0,1,-1,0.38,N
```

Figure 12 Attributes concerning fertility

Season in which the analysis was performed. 1) winter, 2) spring, 3) Summer, 4) fall. (-1, -0.33, 0.33, 1)

Age at the time of analysis. 18-36 (0, 1)

Childish diseases (ie , chicken pox, measles, mumps, polio) 1) yes, 2) no. (0, 1)

Accident or serious trauma 1) yes, 2) no. (0, 1)

Surgical intervention 1) yes, 2) no. (0, 1)

High fevers in the last year 1) less than three months ago, 2) more than three months ago, 3) no. (-1, 0, 1)

Frequency of alcohol consumption 1) several times a day, 2) every day, 3) several times a week, 4) once a week, 5) hardly ever or never (0, 1)

Smoking habit 1) never, 2) occasional 3) daily. (-1, 0, 1)

Number of hours spent sitting per day ene-16 (0, 1)

Output: Diagnosis normal (N), altered (O)

Figure 13 Fertility Attributes' description

From figure 12 the values of variables can be seen and to understand what these values mean we learn from the definitions provided in figure 13. The first variable are seasons that are mapped to numbers 1, 0.33, -1, and -0.33. This is a simple data transformation method where the numbers are transformed or mapped between ranges 0 to 1 or 1 to -1. This transformation is done on the other variables as well. Some variables such as smoking are mapped onto three values, 1, 0, and -1.

```
1  import pandas as pd
2  from sklearn.model_selection import train_test_split
3  from sklearn import tree
4  import numpy as np
5
6  df = pd.read_csv('Fertility.csv')
7
8  X = np.array(df.drop(['Diagnosis'],1))
9  y = np.array(df['Diagnosis'])
10
11 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
12
13 model = tree.DecisionTreeClassifier()
14 model = model.fit(X_train, y_train)
15
16 print(model.score(X_test, y_test))
17 print(model.predict([[1,0.61,1,0,0,1,0.6,0,0.23]]))
18
19
```

Figure 14 Decision Tree Program

In figure 14 lines 3, the decision tree model is imported and on line 6 the data from figure 12 is imported. Lines 8 and 9 in the figure above show the data being split into X and y (input variables, outcome variable). Cross validation, calling and training the model occurs in lines 11 to 14. In lines 16 the score is printed and line 17 is used to check the prediction for the given values.

```
=========
0.75
>>>
```

Figure 15 Decision Tree Performance

After line 16 in figure 14 is executed the score printed on the screen is 0.75 as seen from figure 15. This is pretty good score for a simple model and considering that there was only around 100 instances available.

```
---------
['N']
>>>
```

Figure 16 Decision Tree Prediction

Based on input variables listed on line 17 (figure 14), the prediction for sperm diagnosis is “N”, meaning normal diagnosis. Overall the model is able to perform well considering the low number

of instances. However, the model might still not be able to generalize outside the dataset due to the low number of instances.

7.2 Decision Tree Application

Auto-insurance companies have to process large amounts of claims, forms, and documents. A paper commissioned by the Ghana national auto insurance used a regression and decision tree algorithm to analyze which age groups have a higher probability of requesting a claim (Frempong et al. 2017, cited 25.04.2019). The study found out that private owners of cars from age 18 - 48 are more likely to request a claim than 48-year-old people and above. The data used on the paper contained approximately 1,500 instances consisting of seven variables with the target variable being claim status. The combination of the regression and classification (decision tree) algorithm yielded low error rates and thus concluded that the algorithm is able to perform in practice as well.

Essentially the algorithm split the characteristics or attributes of these drivers and based on probability classified which distinct values (age, sex, private or corporate car) would be likely to file an insurance claim. This could easily be applied to other scenarios as well. Suppose you are a landlord who possesses a property to rent with a number of potential applicants. The goal here would be to find a reliable tenant who is most likely to pay their rent on time. Just like the auto-insurance study the data used to create a model to find potential renters will include attributes such as age, sex, occupation and more. However, to build such a model will require pre-existing data since the number of renters can be too low to produce any meaningful results. After data is procured, the next step would be to clean or transform the data, so it is in proper format. Then to select an appropriate model which will be the decision tree classifier and afterwards, test the accuracy of the model created. If all steps are followed successfully, you as the landlord should be able to select the candidate with the highest possibility of paying rent on time.
The table below display which use cases would be ideal for decision tree classifiers (Parikshit, Joshi, cited 13.05.2019)

Application Case	Suitability
Email Spam	Ideal
Loan Approval	Ideal
Estimating house prices	Non-ideal
Estimating risk	Non-ideal

Table 4 Decision Tree Applicability

8 DESIGN PROCESS

When designing an algorithm to solve a problem there are many factors to consider before deciding the steps to take to the solution. Is there data? If there is no data then no learning can occur fortunately there are many available datasets online. They include stock prices, house prices, images and more. These data can be used to develop ML algorithms that can be used in real life practice. For example, one could build an accurate model that predicts digits written on paper by using the MNIST dataset (Yann LeCun et al., cited 29.04.2019) and then apply this model in practice. However, when there is no existing data or relevant data what steps could a person take? The first step would be to gather enough data which would take time depending on what is meant to be learnt. For example, weather data record multiple decades worth of data, while developing an image recognition model might need 10,000 images. The next step is to make necessary adjustments to data if needed, mapping values, encoding values or decreasing the range of values through normalization. Finding the most suited algorithm would be the next step or one could create their own algorithm altogether. Understanding the data and its outcome can help sift through which algorithms to consider implement. Is the outcome to identify a condition, or estimate price, will help with knowing which direction to go. It must be noted that these are the steps to consider in supervised machine learning.

In chapters 5, 6, and 7 there are three different algorithms with each being best used for individual cases. Linear regression is for analyzing linear trends such as determining the price of a car given its mileage, age, number of replacements, model, and number previous owners. K Nearest Neighbors is best used to categorize data and then classify the new data based on the categories learnt, that is why it is often used in image recognition. Similarly, decision tree classifier is also suited for classification. With the example displayed in chapter 7, the algorithm was effective in its ability to diagnose which patient is normal and which is altered. This is binary classification since either the patient belongs to a population of normal diagnosis or to the altered diagnosis.

8.1 Approach

Given the background of the design process, the author was asked by their thesis advisor on how one would proceed to create the blueprint to develop a machine learning algorithm that can play football.

According to Tom M. Mitchell, in order to create the checker's algorithm, one might think the parameters are where will the opponent checker pieces move and then try to build a model which will predict their next movement. However, this is very taxing to program since there is a lot of search space meaning in this context there is a lot of room of where the opponents checker piece will move. This program will end up being too slow and inefficient. However, the algorithm was built instead by showing board states where moves have already been made with the outcome showing if the board state is positive or not. Now the focus is not on the next position the piece will occupy, rather how the opponent will play. If we can extend this principle one might be able to make a machine learning algorithm that is able to play football. When creating this model, one might think like the checkers example that the parameters to process should be the XY coordinates of the ball in the football field. But this is similar to finding the next tile of the opponent piece in checkers, the search space is too large, and the algorithm will be taxed heavily trying to process all this information. Just like having board states, the idea should be to record data of the ball possession of players before a goal is scored.

8.2 Novel Problem

When given a novel problem to conduct machine learning on, there should be three steps to consider. They are: training experience, selecting an appropriate algorithm, and finally implementing the algorithm. In training experience one needs to consider which data is important, are there any redundant variables that won't have relevant information. For example, when estimating house prices, data that can be left out can be color of the house. Whereas size and age of the house might matter more. What is the goal of the ML model to be made? As mentioned before, is it estimating price or could it be to identify digits based on images of handwritten text. Basically, is the goal of the model regression or classification? Final step is to select the best of either category. Decision tree is good when it comes to binary classification (is patient X healthy or not), while KNN is good for classifying images or categorizing numbers and

letters, and linear regression is best for data with linear relationships. Understanding one's own data and goal will make these three steps much easier to complete.

9 DISCUSSION AND CONCLUSION

Analyzing the possibility of a positive diagnosis to estimating the strength of concrete, machine learning provides many benefits from price forecasting to medical diagnoses. Although this is not always the case, benefits of machine learning have already been demonstrated as can be seen from the DW Slater Real-Estate case, auto-insurance model, and heart disease detection model. The models shown in chapters 5-7 are simple but they are effective. It can be very easy to implement machine learning once a person has learnt the basics of it and the rewards for learning are definitely an incentive. People often think machine learning is a term reserved only for the complex situations, such as facial recognition or one needs to be an expert in machine learning to gain the benefits of it. This is not the case, even with simple techniques there can be valuable information gathered from machine learning that can be used in real life situations. With machine learning being easier to implement and more data available online gives ideal conditions to implement machine learning in a wider array of applications.

Some issues to be addressed are the similarities in chapters 5, 6, and 7. The lines presented in the code for linear regression and other models are almost identical except for a few lines. It must also be noted that guidance for the code has been learnt from the video creator “Sentdex” whose videos have taught the author much about machine learning. Regarding the similarity in code, the reason for this is that no changes are required since the machine learning models are imported. The code is meant to show how simple the whole process can be, that there are already available resources for one to gather and conduct their own machine learning experiments. Although these programs are do not tackle every area of machine learning, a significant part can still be learnt just with the code shown in chapters 5 – 7. All that is required if one wants to utilize machine learning is to prepare the data, find the right algorithm, and train the model. Another issue in this thesis is the process of machine learning displayed is basic. Since data procurement did not need much effort due to online sources and models used were not too complex. This doesn’t reflect real life challenges that people go through which would be when gathering data. Even when one manages to collect data it might not be might be noisy which will require the user to take additional steps before using it as input. Finally, in the thesis the author has mentioned the score of the model which has been used interchangeably with the accuracy of the model and the performance of the model. It must be clarified that the author means the score of the model because it is possible for the model to generate false predictions. A model is wrong when the

input is handled incorrectly, for example the data is not transformed correctly which leads to the model creating false relationships. The score of this incorrect model can still be high but it is predicting based on data that hasn't been transformed accordingly thus, a high score doesn't necessarily mean an accurate model.

Hopefully this thesis encourages readers to practice machine learning so that they are able to gain a deeper understanding and appreciation of concepts such as recommendation systems or financial forecasting. Possibly even find novel use cases in scenarios such as small businesses or one's own startup that can profit from the use of machine learning, or help to advance the field of medicine by creating a unique machine learning algorithm. In conclusion, the author hopes the reader has learnt how machines learn as well how to apply it.

10 REFERENCES

Audio Software Engineering and Siri Speech Team. 2018. Optimizing Siri on HomePod in Far-Field Settings. Hakupäivä 05.05.2019. https://machinelearning.apple.com/2018/12/03/optimizing-siri-on-homepod-in-far-field-settings.html.

Andriy Burkov. 2019. The Hundred-Page Machine Learning Book. Hakupäivä 22.04.2019.

David Chappell. 2015. Introducing Azure Machine Learning. Hakupäivä 05.11.2018. https://download.microsoft.com/download/3/B/9/3B9FBA69-8AAD-4707-830F-6C70A545C389/Introducing_Azure_Machine_Learning.pdf

Benjamin Cohen. 2016. Introduction to Machine Learning & NLP with Python and Weka. Hakupäivä 25.04.2019. https://www.codementor.io/benjamincohen/intro-to-machine-learning-nlp-with-python-andweka-argnk39jr.

David, Cournapeau. Matthieu, Brucher. Scikit Learn. Hakupäivä 15.11.2018. https://scikit-learn.org/stable/index.html.

IBM. Deep Blue. Hakupäivä 20.05.2019. https://www.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/

M. Akhil jabbar, B.L. Deekshatulu, Priti Chandra. Classification of Heart Disease Using K-Nearest Neighbor and Genetic Algorithm. 2013. Procedia Technology. Volume 10. ISSN 2212-0173. Pages 85-94. http://www.sciencedirect.com/science/article/pii/S2212017313004945.

Kunal, Jain. 2016. 25+ websites to find datasets for data science projects. Hakupäivä 30.03.2019. https://www.analyticsvidhya.com/blog/2016/11/25-websites-to-find-datasets-for-data-science-projects/

Parikshit, Joshi. 2017. When to Use Linear Regression, Clustering, or Decision Trees. Hakupäivä 13.05.2019. https://dzone.com/articles/decision-trees-vs-clustering-algorithms-vs-linear

Ben, Keen. 2017. Decision Tree Classifier in Python using Scikit-Learn. Hakupäivä 25.04.2019. http://benalexkeen.com/decision-tree-classifier-in-python-using-scikit-learn/.

Will, Kenton. 2019. Sum of Squares. Hakupäivä 22.04.2019. https://www.investopedia.com/terms/s/sum-of-squares.asp

2018. KNeighborsClassifier. Hakupäivä 07.03.2019. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html.

Yann LeCun, Corinna Cortes, Christopher J.C. Burges. THE MNIST DATABASE of handwritten digits. Hakupäivä 29.04.2019. http://yann.lecun.com/exdb/mnist/.
Ian, London. 2016. Encoding cyclical continuous features - 24-hour time. Hakupäivä 13.05.2019. https://ianlondon.github.io/blog/encoding-cyclical-features-24hour-time/
MathWorks. 2016. Introducing Machine Learning. Hakupäivä 30.03.2019.
Matplotlib Development Team. Matplotlib. Hakupäivä 15.11. 2018. https://matplotlib.org/.
Ryszard S. Michalski, Jaime G. Carbonell, Tom M. Mitchell. 1983. Machine Learning An Artificial Intelligence Approach. Palo Alto, California. Morgan Kaufman. Hakupäivä 26.03.2019
Tom M. Mitchell. 1997. Machine Learning. First Edition. McGraw-Hill. Singapore. Hakupäivä 27.03.2019.
Frempong, Nana & Nicholas, Nimo & Boateng, Maxwell. 2017. Decision Tree as a Predictive Modeling Tool for Auto Insurance Claims. International Journal of Statistics and Applications. 2017. 117-120. 10.5923/j.statistics.20170702.07
OpenAI. 2018. OpenAI Five Benchmark: Results. Hakupäivä 20.04.2019. https://openai.com/blog/openai-five-benchmark-results/
Libby Plummer. 2017. This is how Netflix's top-secret recommendation system works. Hakupäivä 05.05.2019. https://www.wired.co.uk/article/how-do-netflixs-algorithms-work-machine-learning-helps-to-predict-what-viewers-will-like.
Python-course.eu. History of Python. Hakupäivä 05.11.2018. https://www.python-course.eu/python3_history_and_philosophy.php.
Python Data Analysis Library. Hakupäivä 15.11.2018. https://pandas.pydata.org/.
35. Siraj, Raval. 2018. OpenAI Five vs Dota 2 Explained. Youtube.
2005. Samuel's Checkers Player. Hakupäivä 02.04.2019. http://www.incompleteideas.net/book/ebook/node109.html
Scipy Org. Numpy. Hakupäivä 15.11.2018. http://www.numpy.org/.
Terrence J. Sejnowski. Charles R. Rosenberg. 1986. The Spacing Effect On NetTalk, A Massively-Parallel Network. Hakupäivä 14.05.2019.
Sentdex. 2016. K Nearest Neighbors Application – Practical Machine Learning Tutorial with Python p.14. Youtube.
Sam Shead. 2018. DeepMind Losses Grew To $368 Million In 2017. Hakupäivä 06.05.2019. https://www.forbes.com/sites/samshead/2018/10/05/deepmind-losses-grew-to-302-million-in-2017/.

Shannon Slater. 2016. What is Regression Analysis and How Do Appraisers Use it? Hakupäivä 26.04. 2019. https://www.dwslaterco.com/single-post/2016/04/27/What-is-Regression-Analysis-and-How-Do-Appraisers-Use-it.
Nick Statt. 2019. Google and DeepMind are using AI to predict the energy output of wind farms. Hakupäivä 29.03.2019. https://www.theverge.com/2019/2/26/18241632/google-deepmind-wind-farm-ai-machine-learning-green-energy-efficiency
2017. The story of AlphaGo so far. Hakupäivä 20.04.2019. https://deepmind.com/research/alphago/
Wang, Taehyung. Data Preprocessing. Hakupäivä 14.11.2018. https://www.csun.edu/~twang/595DM/Slides/Week2.pdf.
Techopedia. Software Library. Hakupäivä 13.05.2019. https://www.techopedia.com/definition/3828/software-library
UCI Machine Learning Repository. Hakupäivä 31.03.2019. http://archive.ics.uci.edu/ml/index.php
Udacity. 2016. Overfitting. Youtube.
James Vincent. 2016. Google uses DeepMind AI to cut data center energy bills. Hakupäivä 29.03.2019. https://www.theverge.com/2016/7/21/12246258/google-deepmind-ai-data-center-cooling
Jon Walker. 2019. Artificial Intelligence Applications for Lending and Loan Management. Hakupäivä 06.06.2019. https://emerj.com/ai-sector-overviews/artificial-intelligence-applications-lending-loan-management/.
Gio Wiederhold, John McCarthy, Ed Feigenbaum. Professor Arthur Samuel. Hakupäivä 02.04.2019. https://cs.stanford.edu/memoriam/professor-arthur-samuel

11 APPENDIX

1. http://www.hg.schaathun.net/FPIA/Slides/09OF.pdf - Overfitting vs Underfitting
2. http://archive.ics.uci.edu/ml/index.php - Datasets used in both linear regression and decision tree
3. https://www.kaggle.com/johndasilva/diabetes#diabetes.csv – Diabetes Dataset for KNN