UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

EIT DIGITAL

SOFTWARE AND SERVICE ARCHITECTURES

MASTER THESIS

Machine learning applied to behavioral finance in a startup

Martin Haver

2018

MASTER THESIS

Título:	Machine learning applied to behavioral finance in a startup
Autor:	Martin Haver
Tutor:	Carlos Rodríguez Raposo
Supervisor:	Dr. Carlos A. Iglesias
Departamento:	Departamento de Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente:	—
Vocal:	—
Secretario:	—
Suplente:	—

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos
Grupo de Sistemas Inteligentes

MASTER THESIS

Machine learning applied to behavioral finance in a startup

July 2018

Abstract

Machine learning (ML) is quickly evolving and spreading into many different areas of business and life in general. For this reason, growing number of companies of all size look how to utilize the potential of ML to enhance their products and services and get a competitive advantage.

One of such companies is young Spanish start-up based in Galicia, called OpSeeker. It develops tools for financial mentoring of potential small investors by using knowledge based on behavioral finance. The initial challenge when starting an ML project in a small company is to select a suitable tool from the multitude currently available. To select a proper tool at the beginning can save precious resources as it will make all the subsequent efforts more effective. However, small companies may be lacking time or in-house expertise to perform this selection of a right tool-set.

Therefore, one of two goals of this thesis is to develop an easy-to-use methodology that would allow such companies quickly get a recommendation of a technology to use, based just on the properties of the project the company is working on and resource limitations.

This methodology will be then used as a basis for practical part of this thesis - to select proper tools for one of the Opseeker´s projects - using ML for classifying users of their tool, based on the user's behavior. The motivation behind this is to better understand the different groups of users in order to personalize the service for them and therefore provide a higher value.

The practical part of this thesis consists of executing a research among potential users of OpSeeker's users to identify three different groups of users based on their stance towards investing. The data collected in this research will be used to train a classification algorithm that will serve as a basis for personalization of OpSeeker's services in order to provide higher value for the its customers and users.

Keywords: machine learning, classification, clustering, tool selection methodology, start-up, behavioral finance, chat-bot, ML tool selection

Acknowledgement

I would like to express my gratitude towards both of my tutors, Carlos Rodríguez Raposo and Carlos A. Iglésias for their practical feedback and support. I also thank the rest of the team at OpSeeker, namely Gonzalo Camiña Ceballos and Lucka Hefnerová, who helped tremendously to make this thesis happen.

Contents

CHAPTER 1

Introduction

This chapter is going to introduce the context of the project, including a brief overview of all the different parts that will be discussed in the project. It will also break down a series of objectives to be carried out during the realization of the project. Moreover, it will introduce the structure of the document with an overview of each chapter.

1.1 Context

The theory of exponential speed of technological progress is well known and accepted. This theory, in short, states that the advancements the humanity have made so far are being combined to form other advancements that would be unreachable by other means. This speeds up the process of creating new scientific advancements and their ever increasing number fuels the further innovation. Moreover, some of these innovations, like improved automated assembly lines contribute to decreased price of components, which is essential to spread inventions among wide public and thus to turn inventions into innovations. Innovations are in turn essential to motivate, by economic means, further scientific progress and thus the cycle continues with ever increasing speed.

The core topic of this thesis is Machine Learning (ML) a highly important subset of a research field called Artificial Intelligence (AI). According to many leading high-tech industry personalities, the AI will be one of the main driving forces of further technological progress for the years to come. As stressed by Sundar Pichai [14], the CEO of Google, it will do more for humanity than discovery of electricity.

Some of reasons for this belief are:

- The ML and AI will free the resources in form of man-hours from tasks that can be easily (or relatively easily – i.e. that the process of a task´s automation is less costly than continuing to do it manually) automated, and these resources can be transferred to areas requiring a creative type of work – which, at least for now, is not deemed as an area in which machines and algorithms perform better than humans.

- With enormous volumes of data that organizations have collected over the years, we have a lot of potentially useful information waiting to be discovered. Employing machine learning technology, these data-sets can be analyzed very fast and found insights applied while they are relevant.

- Making AI accessible for wide public, as we start to see it already today, for example in form of smart assistants like Siri or Cortana, also an average person without specialized education in computer science is able to access a form of super-intelligence, enabling him to make better informed decisions and generally to achieve more in his life.

AI became a formal scientific discipline in 1956 and over the time produced several waves of excitement as well as sub sequential loss of interest following the failure to deliver on public expectations [22]. However the reason many believe AI will deliver significant progress this time, is the phenomenon of exponential progress. We already possess fundamental under-

lying technologies, like cheap, small and powerful microprocessors alongside with extensive body of theoretical research that will serve as the enablers of AI-powered benefits.

A subset of Artificial Intelligence, Machine Learning is as a method of developing an algorithm, such that the more it performs a task, the better results it is achieving, without being explicitly programmed for the given task [20]. This feature is especially useful when dealing with a large data-sets where the logic of the reality this data-set represents, is unknown and would be very costly to discover it by standard means. Some of the particular applications of ML include:

- Email filtering
- Computer vision
- Optical character recognition
- Unsupervised data mining applications

The purpose of this master thesis is to solve a real problem proposed by a small Spanish start-up constituted in Galicia, called OpSeeker. It is a fin-tech start-up, consisting of 4 co-founders and one employee, providing online financial coaching with the focus in behavioural finance and in making long-term investing more relatable.

The mission is to nudge individuals towards long-term investing using a set of tools that are "gamifying" the process, and guiding the user through it so that the cognitive and emotional biases are minimized and the user becomes prepared to start investing long-term. The business model is B2B - the tools are provided to financial institutions (banks, online banks, pension companies, robo-advisors, etc) in a form of SaaS, serving as a channel to attract individuals and encourage them to convert into long-term investors.

One of the above mentioned tools is the Investment Accelerator - using this tool, an experience of long-term investing for n years is simulated (n is defined depending on the financial institution's needs, typically 30 years). As the user clicks the "Accelerate" button, a new return of a future-year is generated and visualized in a chart. Together with the visualization a Decision Bot (DCB) tool starts talking to the user and explaining the return evolution – the user interacts with the DCB tool choosing form a predefined set of answers or types a custom one. The logic behind DCB is a decision tree: predefined branches connected using relational database tables.

After the user finishes the Investment Accelerator simulation process, and action button pointing the user to financial institution´s landing page focused on long-term investing is unlocked. Once the user completes the Investment Accelerator process, the user will be presented with the possibility of buying a real investment product. One of the indicators

for the success of the application is the number of people buying the investment product suggested through the application.

The practical part of this thesis is focused on making the logic behind DCB less rigid as OpSeeker is interested in looking for the usage of AI to make the DCB more tailor-made to the particular user's characteristics and behavioural patterns. For example, one of the points that OpSeeker wants to tackle is what is in behavioural finance known as loss aversion. People tend to accept lower returns in exchange of minimizing the possibilities of losing. Many years of experience show that well diversified portfolios, with important exposition to variable income are significantly more profitable than portfolios with high percentages of fixed income. However, many people opt for the latter rather than the first.

The important question here is why are investors scared to choose the portfolios which a priori would give them more profitability in the long run. Investors would argue different reasons, such as:

- Lack of information about how variable income markets work
- Association of variable income markets with randomness
- Lack of trust towards financial institutions, etc.

Therefore, this machine learning project aims to detect what are the fears of a to-be investor. The input of the system is the conversation with the DCB, and the output is the fears that the user has. During the training of the system it will incorporate control questions asked in such way, as to determine whether the classification by the algorithm was correct or incorrect and this insight will be use by the machine learning algorithm to fine-tune its parameters in order to improve its accuracy. After the training, the algorithm should be able to categorize users even without these control questions.

One example of a different, future application of ML to OpSeeker would be to analyze user´s behaviour with respect to outcome of a simulation – a change in the market, based on user´s characteristics. Some investors may be more stable, others may start to panic when they will see their stocks have plummeted. The information regarding what kind of strategies is the client applying and more importantly, what is his behavioural pattern can be very useful, not only for the sake of personalizing the user experience (i.e. personalized coaching in order to avoid investment decisions motivated by other than rational though, for example, by a panic attack) but also for the banks who are the clients of OpSeeker, who may, with the consent of the user, suggest various different products to better suit his personality and improve his overall experience and satisfaction with the banking services.

1.2 Project goals

The objective is to answer following questions:

- How to select a proper tool-set for a small company starting with its first ML project?
- How to identify groups of users based on their stance towards investing?

1.3 Project tasks

The above-mentioned questions will be answered through following tasks:

- Developing a methodology for selection of a proper tool-set
- Executing a case-study to gain insight into OpSeeker´s potential user-base´s behaviour
- Developing an ML model to cluster users into groups based on what they fear the most when considering getting into investing

1.4 Structure

This project is divided in 2 main parts. First part is theoretical – the design of a methodology - is independent of the second, practical part – developing an ML algorithm, which is built using this very methodology. The detailed structure of the thesis is as follows:

- ***Chapter 2:*** List of important technology and theoretical foundations that are used in this thesis
- ***Chapter 3:*** The methodology for selecting a proper ML tool-set
- ***Chapter 4:*** The architecture of the ML project
- ***Chapter 5:*** Design and execution of a case study to get an insight into user-base of OpSeeker and to collect data for the ML model
- ***Chapter 6:*** Description and pre-processing of the collected data-set
- ***Chapter 7:*** Building the ML model
- ***Chapter 8:*** Conclusions

CHAPTER 2

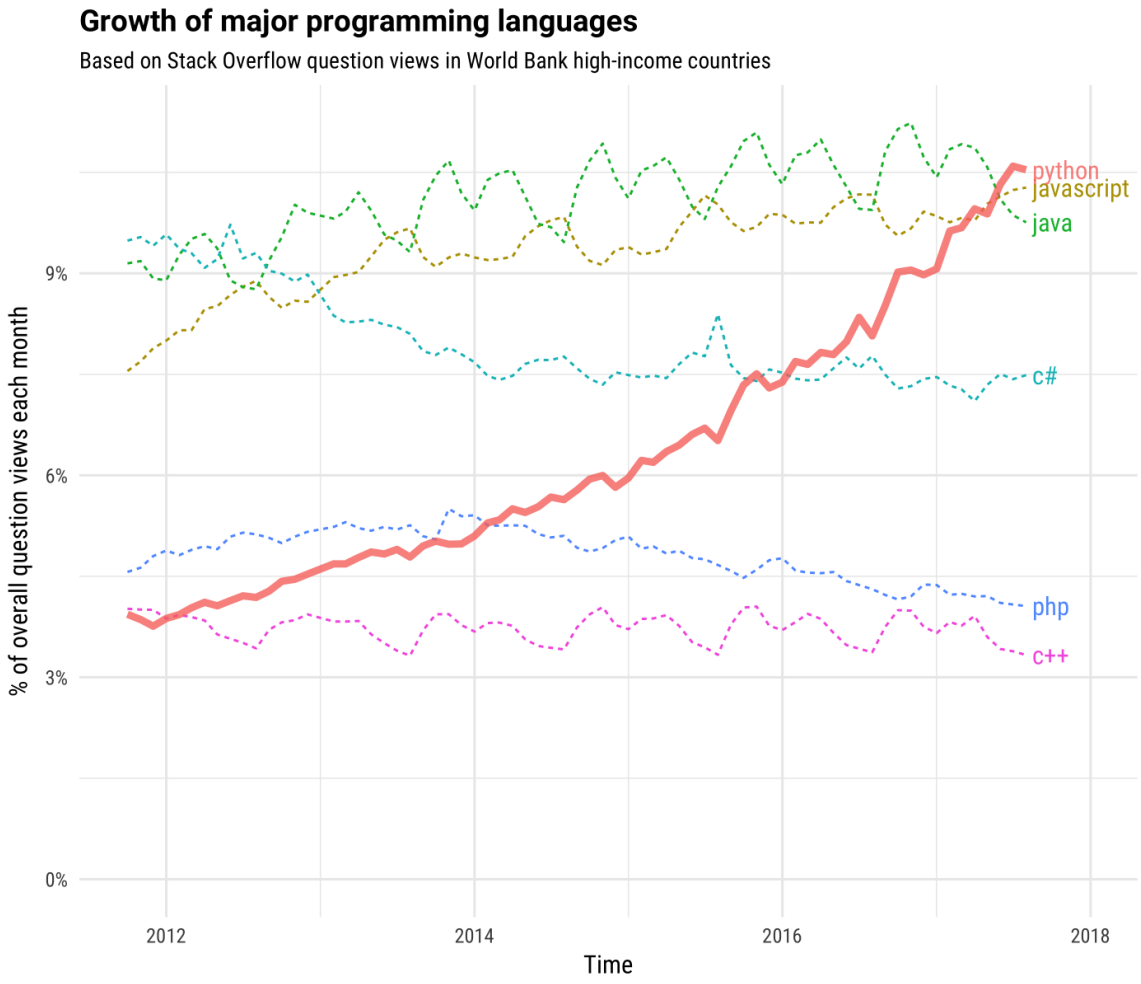
Enabling Technologies

This chapter offers a brief review of the main technologies that have made possible this project, as well as some of the related published works.

2.1 Python

Python is a universal, high-level programming language, used in wide range of domains such as web development or scientific programs. Its versatility is demonstrated in the fact that it is currently among the fastest growing programming languages. (Figure 2.1).

Figure 2.1: Growth of selected programming languages [19]



Although it is not as efficient in terms of speed of code execution as lower-level programming languages like C, its popularity is growing. The advantage of using Python is that the syntax it provides is much less error prone, due to its superior human readability, it is also easier to learn that Java or C++. Another reason this thesis will use Python for every included source code is its wide range of libraries related to ML and active supporting community of users.

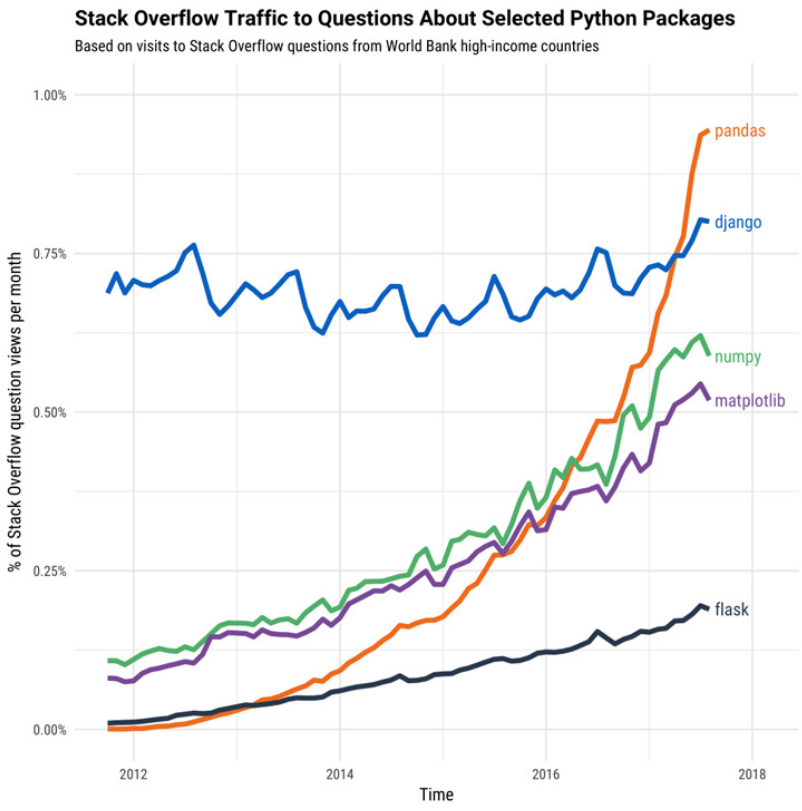
2.2 Scikit-learn

Scikit-learn [8] is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. This package focuses on bringing machine learning to non-specialists using a general-purpose high-level language. Emphasis is put on ease of use, performance, documentation, and API consistency. It has minimal dependencies and is distributed under the simplified BSD license, encouraging its use in both academic and commercial settings.

2.3 Pandas

Pandas [17] is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. It is typically used for data-handling procedures, like reading from CSV, Json or other typical data sources, executing one-hot coding, etc. As seen on the figure, it is used more than web-development libraries such as Django or Flask, which shows the increasing tendency to use Python with ML projects and for data science purposes in general. (Figure 2.2).

Figure 2.2: Python libraries usage [15]



2.4 Chat-bot

A computer program which conducts a conversation via auditory or textual methods [25]. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, thereby passing the Turing test. Chat-bots are typically used in dialog systems for various practical purposes including customer service or information acquisition. Some use sophisticated natural language processing systems, but many simpler systems scan for keywords within the input, then pull a reply with the most matching keywords, or the most similar wording pattern, from a database.

The chat-bot used in this thesis is developed by OpSeeker as one of the tools that form core of its online services. It was loaded with a new conversation which was designed to collect data about user behaviour. Also several other adjustments were made, for example adding a count-down timer to put pressure on user´s in order to get as spontaneous answers as possible, as over-thinking while responding could skew the data collected.

2.5 Behavioral finance

Due to strong influence of the discipline of behavioural finance on the main product of OpSeeker and therefore on this thesis, I consider important it to introduce potentially unacquainted reader with this discipline.

Behavioural economics [24] and its subcategory, behavioural finance is a relatively new discipline pioneered, among others, by Israeli researchers Daniel Kahneman and Amos Tversky in the late 80´s. Their "Prospect theory", they published in 1979 was using cognitive psychology to explain various divergences of economic decision making from neo-classical theory. In the following years more research was applied in the field and one of the most recent Nobel Prize for economics laureates, Richard Thaler, also received his prize for contributions to behavioral economics and his pioneering work in establishing that people are predictably irrational in ways that defy economic theory. This suggests that the field of behavioral economics is a valuable research field, able to drive the progress of human societies further.

One example of a how biases influences our daily lives is a confirmation bias [10] - our tendency to pay more attention to only that evidence, that confirms with our prior beliefs. Having adopted an interpretation of a reality, we force everything to conform to it, even if our interpretation was faulty. Learning how to recognize that we are under influence of this or other biases is not easy, but with suitable training definitely possible.

Despite its potential usefulness, there are not many practical applications of behavioral economics to be seen in the real world, perhaps because it is still a relatively new research

field, which means the wide public did not have enough time to acquaint itself with the topic. Also it is very rare to find a university that teaches behavioral economics or finance. Hence the proposition of OpSeeker is quite unique as it is one of the few companies trying to put the body of behavioral finance research into practice. My thesis functions as one of the stepping stones or enablers for this goal, as AI empowers implementation of behavioral finance at scale required by OpSeeker´s clients, such as banks or insurance companies.

CHAPTER 3

Methodology for ML tools selection

This chapter describes the methodology developed in order to aid small companies with their first steps in ML projects.

3.1 The need for a methodology

To start developing the ML algorithm one of the first steps is to select proper tools from the multitude currently available. This is quite a challenge in itself, since the goal is to select the best tool-set possible in terms of performance and suitability while keeping in mind the limitations, i.e. time, money and skill constraints.

The idea to design a methodology came from inside OpSeeker, this start-up was interested in augmenting its services using ML, but were unsure where to start and did not have a lot of spare time to dedicate to this cause. Therefore, because no such methodology exists so far, I will propose my own, in order to help smaller entities, like OpSeeker with starting their journey into ML-driven projects. This methodology, will be then used in practice on the above-mentioned task of classification of OpSeeker´s users – small to-be investors.

My priority when thinking about the new methodology is usability. I believe this can be achieved by sticking to simplicity (no complicated, lengthy documentation) and usable even when the current state of the art technology moves forward, which means to make it as detached from specific branded technologies as possible and rather keep it based on underlying concepts that stand a higher chance to remain in use for a longer period of time.

A start-up [18] is a company dedicated to fast growth. As such, it has to focus on only the most important activities to achieve the goal. Therefore, especially with respect to one of the prevalent start-up ideologies like the lean start-up, which dictates to eliminate the non-essential tasks and iterate rapidly to meet the needs of early customers, the resources a company is able to allocate elsewhere are very limited. However, many companies, even those that do not work on AI as on their main product, are looking to reinforce their main value proposal by employing some of AI related technologies, whether it is a matter of current fashion that serves to attract more investors and customers or whether it is really potentially beneficial.

This methodology aims to serve to exactly these companies, eager to find a fast way to potentially most suitable tools able to solve their problems, it aims to clarify the complicated world of AI at least a bit, to save time and effort, both of which translate into money.

3.2 Dimensions of the methodology

My proposal is built on an idea that there exists a most suitable approach to solve a problem of given properties, considering limitations a company is constrained by and I aim to devise such a set of questions, answering which will get the user as close to this ideal approach as possible.

Therefore, I propose three main dimensions which the methodology will be using:

- Limitations of a company
  - Number of people
  - Skills missing that are needed to work with particular
  - Cash to acquire specific hardware or software
- Properties of a problem the company wants to solve
  - Technical type of the problem (supervised, unsupervised, reinforced)
  - Type of data (textual, media, labelled or unlabelled, etc.)
- Recommended approach and specific tools
  - The overall approach (in-house, cloud)
  - Recommended programming language and associated ML libraries
  - Recommended ready-to-use paid/free services and tools

First two dimensions are inputs, while the third one is an output. In some cases the company might not be aware of basic properties of the problem they are working on, from ML point of view, however to determine these are beyond the scope of this work. To get to the output, the company states its limitations and describe the problem it is working on. This will eliminate the approaches relying on more resources than are available and will lead to a selection of the most suitable approach and tools to solve the problem, with respect to the constraints.

I acknowledge that the two input dimensions can be complex in the amount of detail that can potentially play a role when selecting the proper tool. However to cover all of these would go against my priority to make the methodology as usable as possible, without the need to painstakingly state every aspect that can influence the process. Therefore, I have devised a set of six questions that cover all the most important factors for both, limitations of the company and properties of the problem the company is working on.

3.3 Using the methodology

First step is to answer the following set of questions. In question 2, several options can be chosen. All the other questions are of a single-choice type.

1. **How many people will be dedicated full time to work on the ML project? (A person working part time equals to $\frac{1}{2}$ of full-time)**

   a) 0,5 - 1

   b) 2 - 3

c) 4 - 5

d) 6 and more

2. **What are their IT skills? Does somebody of them know any of the following on at least intermediate level?**

   a) Python

   b) Java

   c) C++/C#

   d) Scala

   e) R

3. **What is your total budget (excluding salaries) for SW/HW tools/server run-time, etc.**

   a) less than 100 euro

   b) 100 – 1000 euro

   c) 1000 - 5000 euro

   d) 5000+ euro

4. **What do you want to achieve with ML**

   a) Supervised learning

   b) Unsupervised learning

   c) Reinforced learning

5. **What is the size of the data-set you will be working with?**

   a) less than 1000 samples

   b) 1000 – 10000 samples

   c) 10000 – 50000 samples

   d) 50000 – 200000 samples

   e) 200 000+ samples

6. **Does the data-set contain confidential data?**

   a) Yes

   b) No

After the questions have been answered, the next step is to use the mapping table introduced in chapter 3.4 and count the point values for respective answer. The resulting recommendation is composed from two technological solutions with the highest sum of point values. The decision between cloud or in-house deployment is calculated similarly.

This is an example of a result:

1. Recommended solution - Scala + MLlib (28 points)

2. Best alternative - R (24 points)

   + Deployment in the cloud (15 points)

Interpretation of this result is that for the given problem, considering the limitations of the company, a most suitable approach is to develop the ML program in Scala using MLlib framework. Alternatively if for whatever reason the first recommended approach does not seem right, the company is also encouraged to try the best alternative solution which is developing in R language.

The recommendation to deploy in the cloud means that it is either financially or practically more efficient to go for a cloud solution, such as AWS from Amazon or Google´s Cloud engine. These providers offer flexible payment options for rental of their cloud-based computing resources. The prices in mid-2018 start at approximately 0,3 euro/hour for the low-specs configurations.

3.4 Mapping table

When thinking about how to represent the third, output dimension, I tried to reflect different types of usages, i.e. standard ML problems, regardless of specific technological solutions. To give a meaningful recommendation, it is necessary to link these usages with a specific solution. This makes sense because even if some programming languages are considered multi-purpose, they still have strengths and weaknesses due to influence of specific compatible libraries that are developed with a specific goal in mind, differences in difficulty when learning the language, etc.

To achieve this linkage, each answer option in every question is assigned a point value for every recommendable solution. The values range from 1 to 10 and they represent the

relative suitability of the solution with respect to the option selected and to other solutions. The full resulting mapping table can be found in Appendix A.

One specific feature of this scale is that only this question´s number four (what does a company want to achieve with ML?) options are expected to reach point values of 7-10 (for some of the answers) to create a larger point value distance between individual recommended approaches. The reason is that this question is designed to be a defining one, the aspect of a good technological fit of a solution to a problem is considered one of the priorities.

3.5 ML Tools

There is a substantial and quickly growing number of ML related technological solutions. Many of these are branded solutions. Some of them are open source, others require a paid license. The goal of this methodology, however, is to keep it as detached from these specific branded solutions as possible. The reason is that because of the fast pace of current progress in ML field, it is likely that any comparison or analysis of these tools will not be relevant in mere few months. I made an exception however, because I want to include one ready to use branded solution.

The majority of the solutions I propose here are in fact underlying technologies that other branded solutions use. Programming languages such as Python, Java or Scala alongside with frameworks such as Apache Spark and Hadoop are all relevant in the ML sphere and all are more likely to persist for a longer period of time than specific branded solutions built on top of them.

These solutions are typically a programming language with a specific library that is currently the most used one for ML purposes, measured by inquires at StackOverflow. The reason I am including a specific library, like Scikit-learn for Python or MLlib for Scala is that the number of ML-related libraries available is huge and would be impossible to compare them all.

The complete list of the solutions this methodology is able to potentially recommend, is as follows:

3.5.1 Scala + MLlib

Scala is a general-purpose programming language providing support for functional programming and a strong static type system. Designed to be concise, many of Scala's design decisions aimed to address criticisms of Java.

Scala source code is intended to be compiled to Java byte-code, so that the resulting executable code runs on a Java virtual machine. Scala provides language inter-operability

with Java, so that libraries written in both languages may be referenced directly in Scala or Java code. Like Java, Scala is object-oriented, and uses a curly-brace syntax reminiscent of the C programming language. Unlike Java, Scala has many features of functional programming languages like Scheme, Standard ML and Haskell, including currying, type inference, immutability, lazy evaluation, and pattern matching. It also has an advanced type system supporting algebraic data types, higher-order types, and anonymous types. The name Scala is a portmanteau of scalable and language, signifying that it is designed to grow with the demands of its users.

With respect to ML, Scala is becoming the key language in the development of functional products that work with big data, as the latter need stability, flexibility, high speed, scalability, etc. Often, in a research phase, analysis and models are done in Python and then implemented in Scala during production [4].

Most projects focused on ML using Scala are using a library called MLlib. This library is now a standard component of Apache Spark project [23], which was itself written in Scala. Spark is a unified analytics engine for big data processing, with built-in modules for streaming, SQL, machine learning and graph processing. One of its strong features, and therefore a feature of MLlib is the ability to easily create distributed programs. If an algorithm is written in Scala in a functional way, it will allow for faster execution in a cluster as the computing task will be distributed automatically, and very efficiently among the individual machines, thanks to properties of functional programming. Using Spark as a parallel processing framework is then the next logical step towards creating an efficient, distributed application.

3.5.2 Python + Scikit-learn

Python together with its widely used ML library were introduced in chapter 2 about enabling technologies because they are the main technologies used to develop an ML model for this thesis.

3.5.3 Java + Weka

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte-code that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2018, Java is one of the most popular programming languages. The language derives much of its syntax from

C and C++, but it has fewer low-level facilities than either of them.

Weka is the most popular pick as a machine learning library for Java for data mining tasks, where algorithms can either be applied directly to a data-set or called from your own Java code. It contains tools for functions such as classification, regression, clustering, association rules, and visualization. This free, portable and easy-to-use library supports clustering, time series prediction, feature selection, anomaly detection and more. The name is short for Waikato Environment for Knowledge Analysis, it can be defined as a collection of tools and algorithms for data analysis and predictive modelling along with graphical user interfaces.

3.5.4 R

R is a programming language and free software environment for statistical computing and graphics that is supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. Polls, surveys of data miners, and studies of scholarly literature databases show that R's popularity has increased substantially in recent years. As of June 2018, R ranks 10th in the TIOBE index, a measure of popularity of programming languages.

The source code for the R software environment is written primarily in C, FORTRAN, and R itself. It is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems. While R has a command line interface, there are several graphical front-ends, most notably RStudio and RStudio Server, which are the only GUIs developed by the R Foundation, responsible for maintaining the official version of the language. Integrated development environments are available from third parties.

R, with its focus on statistic and mathematical computing, that is closely related to ML, does not have a predominant ML library, instead there are plenty packages for specific algorithms. For this reason I list here R on its own, without a specific library.

3.5.5 BigML

BigML, Inc. is a Machine learning company that provides software as a service (SAAS) for manipulating and analyzing data. The service can be used in production mode or development mode. Development mode is free but limited in the size of tasks that can be completed. Production mode is a paid mode and credits can be purchased ad-hoc in blocks or on a subscription basis. This is a familiar pattern from other cloud based services like storage or compute servers.

BigML provides three main modes to use the service:

- Web Interface: A web user interface that is fast and responsive. The web interface guides the analyst through the process of uploading data and making a descriptive or predictive model and evaluating the model or making predictions as needed.

- Command Line Interface: A command line tool called bigmler built upon the mature Python API for the service that allows more flexibility than the web interface such as the choice of making predictions against a constructed model locally or remotely, and performing tasks such as cross-validation to approximate model accuracy.

- API: A RESTful API is provided that can be used directly via curl commands or via a wrapper in most common programming languages.

The reason I include a proprietary tool that requires a paid licence, when used in production, is that I believe it is beneficial to have at least one in the methodology as a representative of much larger group of similar on-line on-demand solutions. I have chosen particularly BigML, for its user friendly interface and sufficient limits for size of data-sets which can make it an ideal tool for people without coding skills who find themselves in need to analyze data-sets using diverse ML tools.

3.6 Rationale for the mapping table

The central piece of the methodology, the sorting table was developed starting with several hypotheses. Because the values in this table are central for making reasonable recommendations, these hypotheses were then justified using data available and by consulting several star-ups and practitioners in the field. Here is explained the rationale of point values for all of the six questions.

3.6.1 Number of people

Bigger teams can better handle complicated frameworks – and harness their power better to get good results. In theory the more people the better in general, but this question aims to recommend technology that is less complicated to learn to smaller teams and the more complex one to larger teams.

3.6.2 Experience with technology

Experience saves a lot of time studying new technology so if some team-members are skilled with a particular framework, it is a good reason to consider using it further.

3.6.3 Budget

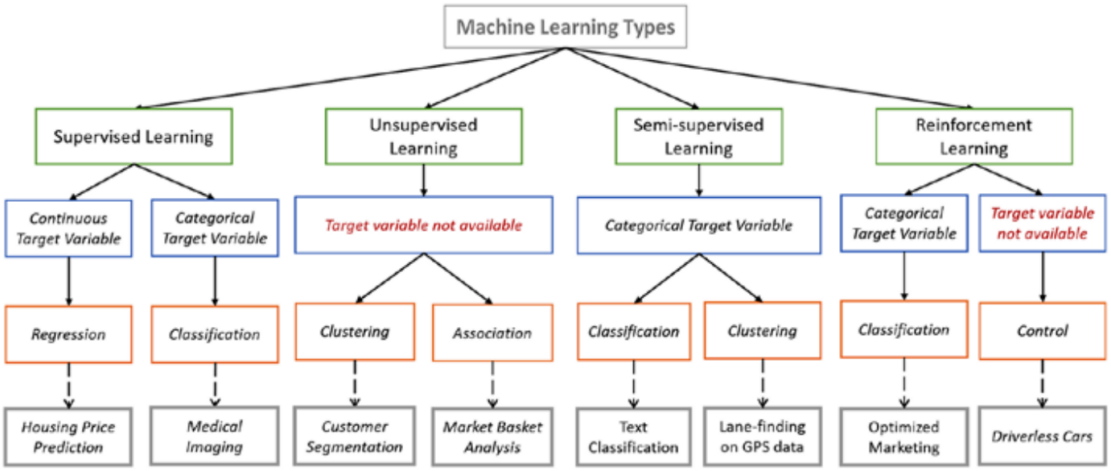
With a small budget the deployment options are reduced to using existing hardware or to try a solution like BigML. With higher budgets the options include credits that can be exchanged for computer power from providers like Amazon, Google or similar.

Other option is to purchase a dedicated server, that can but not necessarily has to, include GPU chips, which are well suited for numerical operations. Several companies, including Intel and Nvidia are offering a dedicated servers for ML, counting multiple GPUs, which are many times faster than general purpose PC. However the downside is a very high price that can in most cases be around 7.000 euros for one high performing ML server.

3.6.4 Goal of the project

The criterion here was to score technological solutions based on how many different algorithms do they offer for different usages of ML. There are multiple ways how to differentiate between types of ML algorithms, but I decided to go for one that is simple and easy to understand, i.e. by dividing ML algorithms into supervised, unsupervised and reinforced learning. Sometimes, as seen for example in figure 3.1, some practitioners list also fourth category - Semi-supervised learning, however as within this category the technological solutions offer very few or none algorithms, it will not be included in the methodology.

Figure 3.1: Basic division of ML types [16]



- Supervised learning

  Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from

labelled training data consisting of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances.

Common algorithms: Nearest Neighbor, Naive Bayes, Decision Trees, Linear Regression, Support Vector Machines (SVM), Neural Networks

- Unsupervised learning

  Unsupervised machine learning is the machine learning task of inferring a function that describes the structure of "unlabelled" data (i.e. data that has not been classified or categorized). Since the examples given to the learning algorithm are unlabelled, there is no straightforward way to evaluate the accuracy of the structure that is produced by the algorithm—one feature that distinguishes unsupervised learning from supervised learning and reinforcement learning.

  A central application of unsupervised learning is in the field of density estimation in statistics, though unsupervised learning encompasses many other problems (and solutions) involving summarizing and explaining various key features of data.

  Common Algorithms: K-means clustering, Association Rules

- Reinforced learning

  Reinforced learning (RL) aims at using observations gathered from the interaction with the environment to take actions that would maximize the reward or minimize the risk. Reinforcement learning algorithm (called the agent) continuously learns from the environment in an iterative fashion. In the process, the agent learns from its experiences of the environment until it explores the full range of possible states.

  RL is a type of Machine Learning, and thereby also a branch of Artificial Intelligence. It allows machines and software agents to automatically determine the ideal behaviour within a specific context, in order to maximize its performance. Simple reward feedback is required for the agent to learn its behaviour; this is known as the reinforcement signal.

  Common Algorithms: Q-Learning, Temporal Difference, Deep Adversarial Networks

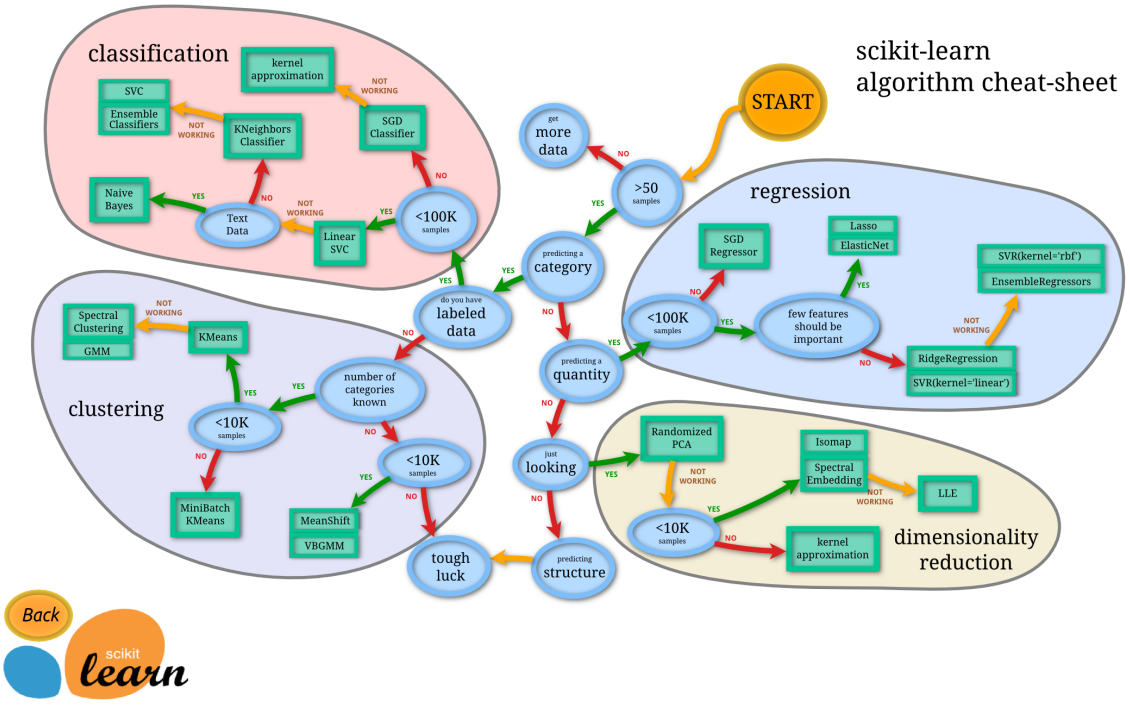
R [5] is a powerful language for ML. This stems from very large number of algorithms already implemented. However all of these are all provided by third parties, which makes their usage very inconsistent. This slows the user down, a lot, because he has to learn

how to model data and how to make predicts with each algorithm in each package, again and again. Therefore the point value in the mapping table has to reflect the wide range of algorithms being offset by their inconsistency with each other.

MLlib for Scala has evolved a lot in newer versions of Spark, with many of the most used ML algorithms added, several of them available also in a version suitable for parallel computing. The number of algorithms however is considerably smaller that in Weka, Scikit-learn or R. Those with ambition to develop reinforced learning models in Scala should look instead to other library called Deeplearning4j, a framework usable both with Scala and Java.

Scikit-learn library for Python is well established and this is reflected by large number of both supervised and unsupervised ML algorithms. These are mostly implemented using low-level programming languages, such as C, therefore are very good performance-wise. According to Scikit´s documentation, reinforced learning is out of scope of the library and therefore algorithms for this type of ML are not included.

Figure 3.2: A visual tool for selecting a suitable algorithm from Scikit-learn library. [21]



Weka for Java has good selection of all types of ML algorithms. Being written in Java, they share an advantage with Python and Scala algorithms, that is being relatively easy to implement in the production, a feature that is more difficult to achieve in R. **BigML**, an online tool offers a selection of seven supervised and four unsupervised algorithms that are all

easy to configure without any coding at all. This is an advantage for less demanding tasks, but those wishing for greater control over their algorithms, may want to look elsewhere. Also its competitors on the market provide a similar offering within their not-paid versions.

3.6.5 Size of data-sets

Measured in number of samples, size of data-sets is one of the key factors determining the time needed to train an ML model. With increasing performance of standard desktops or even laptops, these machines are able to process smaller data-sets in reasonable time. Medium sized data-sets, however, would be a good candidate for distributed processing in the cloud as flexible, pay-as-you-go offers of cloud providers can significantly reduce the computing time. Very large data-sets however, can become very costly to process in the cloud, so unless it is a one-time operation, acquiring dedicated servers is recommended, like ones possessing GPUs that have been shown to significantly reduce the training time, especially in tasks involving deep learning [7].

With respect to specific technologies that are used to process data-sets, higher point values were awarded to those with better implementation of distributed data processing frameworks, namely Apache Spark. Another important note is that R language operates in RAM, therefore its scalability is limited to RAM size. There is an option to use an HDFS connector, but this adds another layer of complexity to the development.

3.6.6 Data Confidentiality

Although the main big cloud providers like Amazon AWS claim to be fully GDPR compliant and secure against crypto attacks, some clients might feel more assured to use a service if they are guaranteed that their data will not leave the physical premises of the company that is using them.

3.7 Disclaimers for use

The methodology, naturally, does not claim to provide the correct answer for all the cases, simply because of the sheer variety of problems that can be solved by applying some part of the AI universe. Its aim is rather to provide a "good enough" starting point for most common situations, with some rationale, for a company to understand which path is likely to be the correct one and why.

This in my opinion can save some precious time, by narrowing down the focus of the company to potentially most relevant approaches. One of the concerns when speaking about such selective methodology is its potentially limiting effect, i.e. a company may be misled

on a wrong path. In the case when the provided rationale of the mapping table for selecting a particular approach does not make sense to a company, it is of course free to disregard the recommendation and is also welcome to share a feedback with the author of this thesis, in order to improve the methodology further.

As a side note I would like to add, in case nobody in the company/team has any knowledge with programming language mentioned in the question number two, it is recommended to start with learning Python, which despite being widely recognized as a beginner´s language, it remains a powerful tool for ML and allows fast prototyping, therefore even if it is not the best fit for the particular project, it will be discovered early and time losses will be minimized.

Another note is that for some particular cases, like developing a deep learning algorithms or working with media (images, videos,...) data-sets, there is very likely a better tool to use, for example Tensorflow or similar, that have not been mentioned yet. The reason is that the number of theses specific usages would inflate the volume of this methodology too much, far beyond the scope of this master thesis.

CHAPTER 4

Architecture

This chapter describes the overall architecture of the project, with the connections between the different components involved on the development of the project.

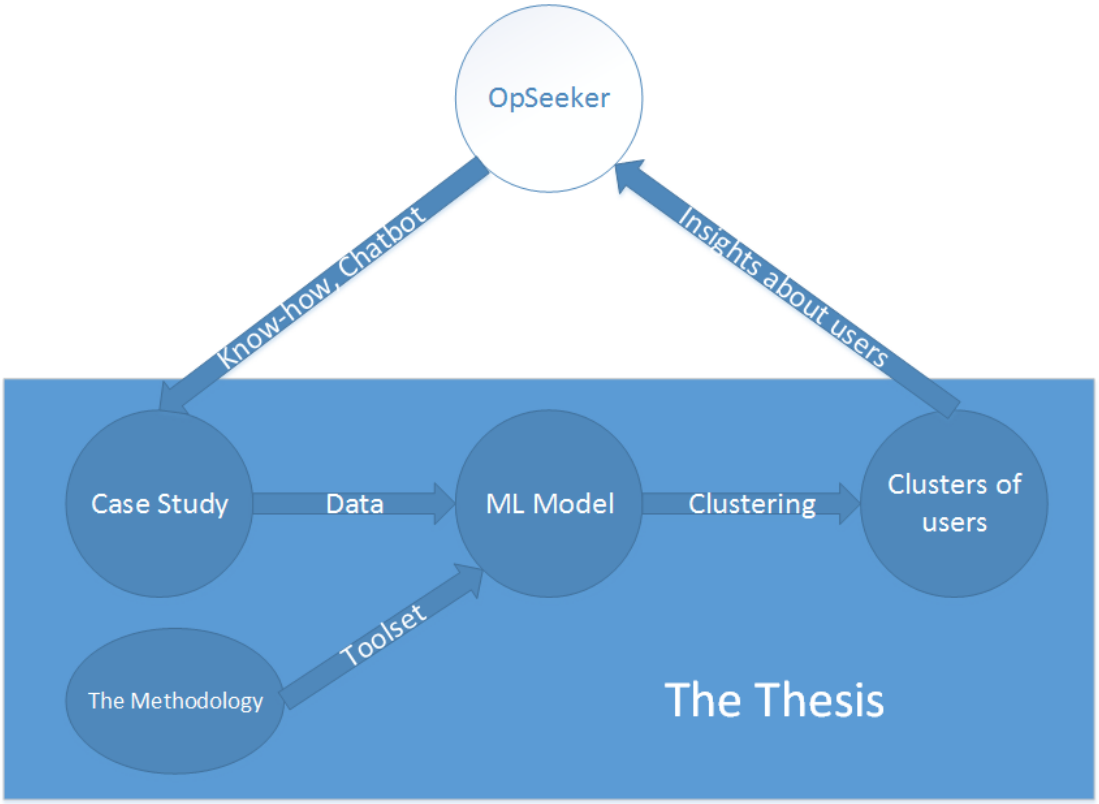
4.1 Architecture

To collect necessary data for the training of the ML algorithm, we have, in cooperation with OpSeeker, designed and executed a case study.

This study was designed based on OpSeeker´s knowledge in behavioural finance and using its proprietary chat-bot technology as well as a list of people interested in the OpSeeker's project, who became the first respondents in the case study.

The methodology which was introduced in chapter 3, was then used to select a proper tool-set for the subsequent development of the model, which was then trained using the data collected from the case-study. The outcome of the model are clusters of users, grouped together based on their stance towards investing. Interpreted results of of the clustering will be used to augment OpSeeker´s tools and to personalize the service it provides – which is the chief goal of the thesis.

Figure 4.1: Architecture of the project



CHAPTER 5

Case study

In this chapter I will describe a case study executed to collect data on OpSeeker's users' behaviour. It will be outlined the whole process of design and execution of the case study.

5.1 Context

This chapter is describing the whole process of design and execution of a case study which has 2 main objectives:

- Learn about behaviour of potential future users of OpSeeker´s services
- Collect data that will be used to train the ML model

OpSeeker´s objective is to nudge people to invest more and better. Under this perspective it is in OpSeeker's interest to provide to each user the most suitable content. In order to do this, OpSeeker financial team has considered that the best way is to characterize users based on their fears while investing. OpSeeker would like to know what the fears of the users are only by analyzing how they interact with the application without making explicit questions. The hypothesis therefore is that users of OpSeeker´s application can be put into pre-determined categories based on their behaviour when using the application. This task is a pattern recognition problem and AI was proven in past to be very useful in solving such problems with reliability and speed, which are both necessary when recognizing patterns at scale.

To start, two things are necessary:

- To determine the categories we will be putting users into – This task is an important challenge in itself and is composed from 2 key aspects:

  - Business logic from a point of view of OpSeeker – i.e. what number of categories is a good balance between sufficient customization and workload invoked by personalizing the service for every group of users.

  - Psychological research – What are the truths about the behaviour of people when it comes to making investment decisions.

- To find out what people, falling into a category, have in common – the classification problem for a machine learning algorithm.

The first thing was provided by one of the founders of OpSeeker, Gonzalo Camiña Ceballos, based on his experience. Some researchers argue not to differentiate people based on their investment strategies but rather on the fears or reasons in general that keep them from making good (in terms of what was empirically proven to be good in past) investment decisions. One of the main lessons learned from behavioural economy/finance is that individuals are frequently driven by their not-so-rational subconscious, intuitions and even if they try to be rational, there are many biases that come to play that in the end have big influence on their actions. The resulting set consists of six fears or reasons that we assume play the biggest role when an individual is thinking about investing.

5.2 Categories of fears

As mentioned above, the personalization of service will be done based on categories of reasons that keep people from investing and/or making optimal investment decisions. These six categories, outlined by OpSeeker´s CEO are as following:

- *"Markets are a casino":*

  The fear that movements on the financial markets are more than by anything else ruled by chance. This fear is quite easy to understand, as from a perspective of someone without access to relevant information sources and/or knowledge about how financial markets function, the movements of prices may appear as random and unpredictable.

  Unpredictability is, of course real to some degree, but with sufficient information, that professionals handling financial portfolios possess, it is possible to develop such strategy that will mitigate the dangers connected with volatility on the markets, especially if investing in the long term.

- *"I don't have enough information to feel prepared to start investing":*

  Similar as above, however these people know the source of their insecurity lies in the insufficiency of information they have. This fear can be tackled by educating users about how the markets work, with importance being put on presenting the information in as clear way as possible, not to deter users without extensive background in mathematics/economics, who may struggle with some more complex formulas, charts and concepts.

- *"Financial institutions are trying to take advantage of me":*

  Humans are prone to be skeptical against trusting institutions wielding big power. Banks and other organizations handling money on behalf of their clients are not an exception from this rule. This may be rooted in ancient times when loaning money for profit was deemed immoral. This image is only strengthened by a multitude of conspiracy theories and recent scandals, such as Caja Madrid scandal [9]. Despite all of this, banks and other financial institutions have a big share on the speed the economy is growing and societies evolving. The law applies and can be used to protect the clients. With proper information, both parties can be highly profitable, the banks and their clients.

- *"I don't have enough disposable income to make investment worth it":*

  In Spain, the average household net-adjusted disposable income per capita is USD 1927 a month and the OECD average is USD 2547 a month [13]. This, in comparison

with investment gains that are advertised usually in lower percentages, like 5%, can make people think, understandably, that with low amount of money it is impossible to save enough money. This stems from two reasons, first, it is difficult to imagine yourself in a span of 30 or 50 years, i.e. a time range when investing really makes sense even with small amount of money.

Second, humans don't cope well with understanding of how exponential functions, and among them saving or investing, work. It seems it is just not natural to our brains, as opposed to linear functions. However, it is a fact that thanks to exponential function of saving in a long term, an individual can save money providing him with financial security even if he is able to put aside only a very small amount on a regular basis.

For example, saving aside an equivalent of price of one coffee (worth $2) every day, with interest rate being 8%, in 50 years a person can save a significant amount of $428,145. Indeed, we have to take into account also inflation and taxes but despite these, the amount saved is, in most countries, enough to retire without relying on other sources.

- *"Capital gains tax is too high":*

  In 2017, residents of Spain pay tax on savings income progressively at 19% (0-6,000 euro), 21% (6,000-50,000 euro) and 23% (over 50,000 euro) [3].

- *"I do not care about my financial future":*

  Many adults say that saving for retirement is like saving for a stranger. We may plan well in near-term, want to take care of our parents, children and even pets but it is difficult to understand the needs of our own future self [12]. This may be one reason, beyond pure carelessness (you-only-live-once spirit), why people do not take saving for future seriously enough.

  However, most people will get old one day and therefore the need for financial security at advanced age is real. This issue can be tackled by at least two ways, by either positive motivation (showing the benefits of saving for retirement) or negative motivation (showing the grave consequences not saving can have). Different groups of people would perhaps respond differently to these two approaches, the question which to choose, however is out of the scope of this thesis.

  These categories are what OpSeeker believes are the most important in determining a person´s behavior at the market. However, it is very likely a person does not suffer only from one fear, more likely everybody is motivated by a different combination of various fears. With this in mind, the test was designed in such a way as to reflect these

different combinations. Elements from several fears are included in single question to determine their relative strength, the goal being a determination of a profile of a user with dominant and a secondary fear that can be later addressed in a chat-bot conversation.

5.3 The test

In order to both, collect data for an ML algorithm and to learn about behavior of OpSeeker´s potential users, we decided to use an adjusted version of OpSeeker´s Decision Chat-bot (DCB). The respondents were asked to participate in our research that tackles a problem with sub-optimal investment decisions and using the DCB they responded to questions.

This section covers the whole process of designing, executing and evaluating this test. There were several challenges to be solved, including GDPR compliance, design of a chat-bot scrip that would be complete from our data-requirement point of view, but not exhausting for the user. These challenges had to be tackled with the project time-line and our limited resources in mind.

5.3.1 Development of a chat-bot conversation

Several team-members separately wrote their first version of the test and after providing feedback, the best parts from each part were joined into one single conversation, which was refined further. The emphasis in the design phase was put on following aspects:

- The developed dialogues will be later used in production, therefore they have to have a high added value for the user. The conversation´s objective is to nudge users to invest better.

- To keep the respondent engaged – The dialogue relies on short questions or answers rather than long monologues. The respondent is frequently expected to interact with the application, by either answering to a DCB´s question or by simply confirming that he understood the conversation so far.

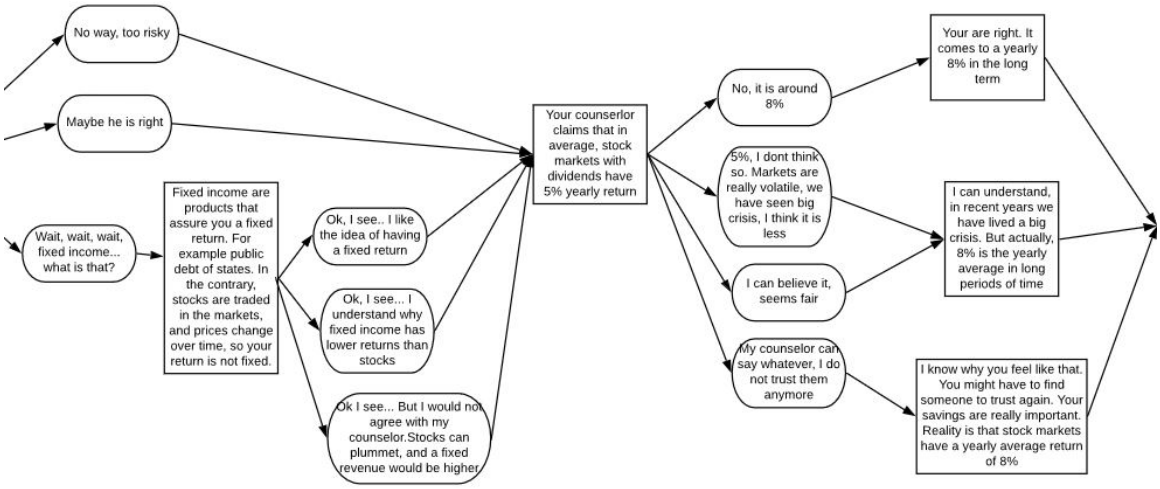
- According to research, if people are interacting with a machine that pretends to be human as well, it may make them feel anxious and uncomfortable. This phenomenon is known as “Uncanny valley” [11]. We made sure people know that they are interacting with a machine, to make it more relatable, we named the chat-bot “Lucie”.

- To provide some value for the respondent – Using a simple but not easy question – and a tendency of humans to seek only evidence supporting their beliefs, the respondents

got to see one of the biases in the practice and hopefully they will think about this lesson in future.

In this design phase, we have adopted principles from a psychometric test known as Facet5, which we considered an objective way to test unbiased behaviour. Moreover we have employed a Likert scale to measure the degree of inclination of a user towards a particular fear. The tool used to develop the conversations is called Lucidchart, an online tool that allows to create various diagrams. Figure 5.1 shows a small sample of how the conversation looked like in the design phase:

Figure 5.1: Design of the conversation



The full conversation has several distinct parts:

1. Introduction and necessary obtainment of explicit consent for data handling.

2. A set of questions measuring how much is a particular fear relevant for a person. For example, as seen in Figure 5.2, each question has two different statements and user has to press buttons labeled 1, 2 or 3 to state with which statement he agrees more. If he agrees with the first statement he answers 1, if with the second he answers 3. If he is somewhere in the middle he uses 2. The users are told they will have in average 10 seconds per statement couple, in order to prevent over-thinking and make them respond in as spontaneous manner as possible.

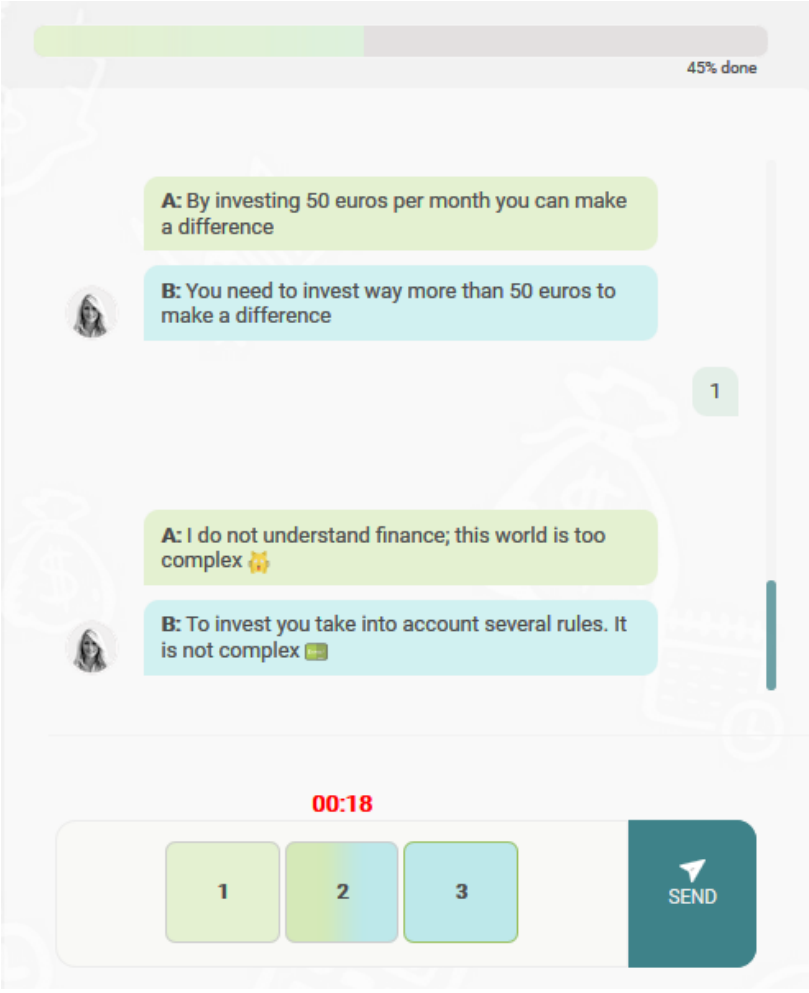
3. Questions inquiring about the user´s stance towards finance in general.

4. A case study that tests user´s familiarity with effects of annual average return and compound interest. The user is presented with a task to estimate investment gains

of someone who would be investing 3 euros a day for 50 years. The point of this section is to identify people who are not aware of potential high returns resulting from long-term investing, even with small disposable income.

The users are told they will have in average 10 seconds per statement couple. The conversation covers more than fifty different scenarios that differ based on user´s responses, especially if the user demands better explanation of some concept, but is very rigid at the moment. After finishing this project, the chat-bot will adapt to user´s behavior in much smarter way, thanks to the ML algorithm, which is the ultimate goal of the practical part of this thesis. All of the user´s answers in this test are logged and will form the data-set for the ML algorithm. The test was available both in English and Spanish to cover broader user-base.

Figure 5.2: Front-end of the chat-bot used in the test



5.3.2 Data protection

General Data Protection Regulation (GDPR) [6] is a regulation in EU law on data protection and privacy for all individuals within the European Union (EU) and the European Economic Area (EEA). It also addresses the export of personal data outside the EU and EEA areas. The GDPR aims primarily to give control to citizens and residents over their personal data and to simplify the regulatory environment for international business by unifying the regulation within the EU.

Ley Orgánica de Protección de Datos de Carácter Personal (LOPD) [26] is a Spanish law that guarantees and protects the processing of personal data, public liberties and fundamental human rights, and especially of their personal and family honor and privacy. Its main objective is to regulate the treatment of data and files, of a personal nature, regardless of the support in which they are treated, the rights of citizens over them and the obligations of those who create or treat them. This law affects all data that refers to registered humans on any support, computer or otherwise. Excluded from this regulation are those data collected for domestic use, classified materials of the state and those files that collect data on Terrorism and other forms of organized crime.

Our test involves private and sensitive data of the respondents, mainly from Spain but also from other parts of the EU. Therefore we needed to make sure that we are LOPD and GDPR compliant before executing our research. This fact was stressed in an email used to deliver the test and also at the beginning of the test, to tackle concerns some respondents may have, especially with relation to recent internet privacy scandals, such as the one involving Facebook and Cambridge Analytica. Also, we consulted a legal firm which provided us with a report on the actions to take to ensure GDPR compliance:

In GDPR philosophy exists the idea that the user should be in control of their data. In the context of our test, actual steps taken include:

1. Explaining the user that the clicks were going to be collected anonymously for a test, and giving the opportunity to stop immediately the test.

2. At the end of the conversation the email is asked. We explicitly explain for what purpose this email is collected – in this case it is for sharing the results of our research. However the user is also free to finish the test without entering a valid email address.

By undertaking these steps, we have not only fulfilled the requirements of the law in Spain and EU to avoid serious penalties, but moreover achieved higher credibility of the test.

5.3.3 Number of samples

One of the tasks, when planning to collect data is to estimate how much data is actually needed and therefore to know when to stop the collection process. The general answer is that it always depends on the properties of a particular project. The size of data necessary depends on several properties of the project:

- What is the minimum accuracy we want to achieve
- How many categories are we going to have

Minimum sample size estimation is a discipline in itself which is outside the scope of this thesis and I have not been able to find estimates for projects similar to this thesis. After consultation with the tutor, It was established that collecting a hundred samples per category and employing methods to make the most of the data-set, it would provide results with satisfactory accuracy. However we set an internal goal to reach at least one hundred samples per category.

5.3.4 Test execution

We first tried to reach as many possible users with our mailing campaigns. However due to the low clicking rates we decided that we needed to enter other channels in order to be able to reach a desired number of actions.

Our rates in the mailing campaigns were as follows:

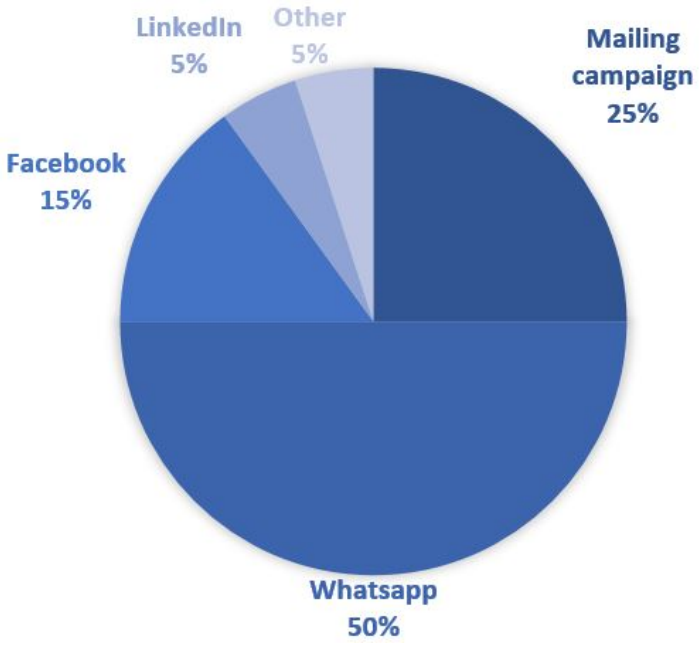
- Open Rate – Average of 40%
- Clicking Rate – Average of 3%

With this number the sample needed was too big for our resources. So we decided to target through other channels such as Whatsapp, Facebook and LinkedIn social networks.

Overall, the test was delivered to approximately 4000 people over various channels. The participants in our sample form a significant majority between the age of 18 and 35. This age was also the one targeted through our email campaigns. The composition of users who started the conversation with the chat-bot, based on the channel over which they were contacted is shown in figure 5.3

86% of participants did the test from Spain, 5% from Mexico and 9% from other countries. 80% percent of the users did the test from their mobile devices. Thanks to the fact that the team at OpSeeker developed the chat-bot with a responsive front-end, the users had guaranteed an optimal user-experience while using the chat-bot from any device. The

Figure 5.3: Sources of participants in the case study



total count of the people who started the conversation with the chat-bot after one month of collecting data reached approximately six hundred. This would satisfy our initial goal set to collect one hundred per category. However, we had to take into account than many conversations logged will be incomplete or otherwise corrupted and therefore not usable for training a ML algorithm.

Thus, still during the data collecting period, a decision was taken to reduce the desired number of categories from six to three. The chat-bot conversation was simplified slightly and generally adjusted as to reflect this new approach and to make the conversation shorter and easier for the users to complete.

The resulting reduced list of fears is as follows:

- *"I do not have enough information to feel prepared to start investing"*
- *"I don not have enough disposable income to make investment worth it"*
- *"I do not care about my financial future"*

CHAPTER 6

Data preparation

This chapter describes the data-set, its form, structure and size. I also discuss the steps that were necessary to prepare the data-set for subsequent training of the ML model.

6.1 Available data-set

The raw data-set contains in total 590 logged conversations. It comes in a form of JSON objects that are stacked upon each other in a single file. The schema of one object – a set of responses of one single user is following:

- $oid: unique identifier of a particular conversation
- _class: specifies a type of conversation
- userId: uniquely identifies a user of the test
- boxes: a container element holding "boxes", each box contains details about one question in the conversation, therefore there are as many boxes as questions in the particular test – a number depending on whether the user completed the whole test and which were his responses during the test. A complete conversation has between 20 and 30 boxes. A single box contains these fields:
  - blockCode: a section of a conversation
  - boxCode: specific question within a section
  - dateTime: time when the question was shown to the user
  - answer: details about a user´s answer to a question
    - dateTime: time when the user answered
    - type: whether the response had a form of a button or a text field
    - payload: a value ranging from 0 to 3, corresponding to which button the user pressed in response to a question. It can also contain a text input from the user in particular questions.

Example of a structure of one "box" element:

"boxes": [{ "blockCode": "Intro1", "boxCode": "1", "dateTime": { "$date": "2018-06-24T12:04:56.380Z" }, "answer": { "dateTime": { "$date": "2018-06-24T12:05:10.279Z" }, "type": "button", "payload": "0" } },.....

6.2 Pre-processing

A quality ML model can be trained only using quality data-set. Initial analysis of the data-set revealed that the raw data received from the chat-bot contained many unfinished or otherwise corrupted conversations that could not be used to train the model and therefore data cleaning was necessary.

Secondly, because the raw data contain a lot of information not useful for ML training purposes, another task was to extract only the useful information - the ID of the user identifying the conversation and payloads from all the questions.

However, it turned out that not all the conversations contained all the questions. This is an issue if we want to avoid skewing the results by comparing incomparable answers. I performed another analysis of the data-set in order to find out which questions were shared for all the complete conversations. It turned out there were 19 such questions, this means all the extra questions had to be removed.

Another challenge was caused by a front-end bug of the chat-bot. Under specific circumstances, the user could get stuck at any point in the conversation and had to refresh the web-page to be able to continue. Although this bug was discovered rapidly by the team at OpSeeker, several logged conversations contained duplicate boxes as a result of double logging, when only one contained a payload, which could potentially confuse any algorithm I will use later.

To tackle all of the above-mentioned tasks, I have developed a parser in Python that would take the JSON file containing the raw data as an input and produce an output CSV file, filtering out following items in the process:

- Incomplete conversations, i.e. those containing less than 19 boxes with payloads
- Boxes that do not appear in all the complete conversations
- Duplicate boxes

After the pre-processing, the usable data had shrunk to 264 conversations which is approximately 45% of the original data-set. Each row in the resulting CSV file contains an ID of a user and payloads for questions he answered. Every row in the resulting CSV file represents a set of answers from one respondent and has the following form:

userID	**Q1ID**	**Q2ID**	...	**Q19ID**
id	payload	payload	...	payload

Question identifiers Q1ID, Q2ID... link the respondents answer with a particular question.

CHAPTER 7

Model building

This chapter will describe the achieved goals done by the master thesis following some the key points developed in the project.

7.1 Toolkit

Before creating the actual model, the first step was to test my own methodology in practice and answer its question to find out which framework is the most suitable one for this particular project. Here are my answers, highlighted in green:

1. **How many people will be dedicated full time to work on the ML project? (A person working part time equals to $\frac{1}{2}$ of full-time)**

   a) 0,5 - 1

   b) 2 - 3

   c) 4 - 5

   d) 6 and more

2. **What are their IT skills? Does somebody of them know any of the following on at least intermediate level?**

   a) Python

   b) Java

   c) C++/C#

   d) Scala

   e) R

3. **What is your total budget (excluding salaries) for SW/HW tools/server run-time, etc.**

   a) less than 100 euro

   b) 100 – 1000 euro

   c) 1000 - 5000 euro

   d) 5000+ euro

4. **What do you want to achieve with ML**

   a) Supervised learning

   b) Unsupervised learning

c) Reinforced learning

5. What is the size of the data-set you will be working with?

a) less than 1000 samples

b) 1000 – 10000 samples

c) 10000 – 50000 samples

d) 50000 – 200000 samples

e) 200 000+ samples

6. Does the data-set contain confidential data?

a) Yes

b) No

The resulting point values for each technological solution are:

Technology	Sum of point values
Java + Weka	20
Scala + MLlib	22
Python + Scikit-learn	29
BigML	25
R	21
Cloud	6
On premise	18

With the policy of the methodology of providing two recommended approaches, one main and second as an alternative, the sorted recommendation looks as follows:

1. **Best fit:** Python + Scikit-learn (29 points)
2. **Best alternative:** BigML (25 points)

+ Deployment on existing HW on premise instead of in the Cloud

7.2 K-means algorithm

After having everything ready, the proper toolkit – Python with its Scikit-learn library and a clean data-set, the next step is to develop a ML model. After consulting the documentation of Scikit-learn, I decided to go first with an algorithm known as K-means [1], because it is basic and also the most widely used one for unsupervised clustering problems, such as categorizing users of an application to a fixed number of categories with previously unknown features.

The k-means problem within Scikit library is solved using Lloyd's algorithm. The average complexity is given by O(k n T), were n is the number of samples and T is the number of iteration. The worst case complexity is given by O($n^{(k+2/p)}$) with n = n_samples, p = n_features. In practice, the k-means algorithm is very fast (one of the fastest clustering algorithms available), but it falls in local minima. That's why it can be useful to restart it several times.

K-means algorithm works as follows [2]:

1. **Assignment step:** Assign each observation (sample) to the cluster whose mean has the least squared Euclidean distance, this is intuitively the "nearest" mean. (Mathematically, this means partitioning the observations according to the Voronoi diagram generated by the means).

2. **Update step:** Calculate the new means to be the centroids of the observations in the new clusters.

These steps are then repeated as long as there are changes in assignments of samples into clusters.

The complete model building process included following steps:

1. Load the data-set

2. Transform "payload" values, which are numerical, into alphabetical as a preparation for one-hot coding.

3. Perform one-hot coding

4. Train the model using K-means with K=3

5. Calculate centroids of the clusters

6. Export the centroids

The cleaned data-set contained "payload" values from range 0-3 but these are not ordinal, meaning there is no intrinsic order between various values. One option would be performing so called label encoding, however the problem with label encoding is that it assumes higher the categorical value, better the category. This is not true for our data. Therefore, it was necessary to perform one-hot coding. It means introducing dummy, binary variables to the data-set. This means creating as many new columns as there are different variables in an existing column and doing this for all of the columns, except the first - userID column.

The number of desired clusters - 3, was given by a decision described in chapter 5.3.4. The goal of the clustering was to discover the centroids of these three clusters.

7.3 Results

The resulting data-set, containing three 56-dimensional clusters has the following form:

Centroid	**Q1Id_1**	**Q1Id_2**	...	**Q19Id_x**
1	0.72	0.28	...	
2	0.76	0.24	...	
3	0.94	0.06	...	

There are 56 dimensions as a result of one-hot coding described above. The actual values are within a range 0-1 and represent a percentage of users that chose an option with respective payload for respective answer. The table above is interpreted as 72% of users chose a button with payload 1 and 28% with payload 2. This also means that sum of percentages for all the options within one single question is always 100%.

Division of samples into clusters is following (out of 264 samples):

1. **Cluster 1:** 68 samples
2. **Cluster 2:** 88 samples
3. **Cluster 3:** 108 samples

I am not allowed to share full results of the clustering, because they may contain potentially valuable insights, exploitable by different entities and I am bound to OpSeeker by a non-disclosure agreement. However I can provide here an interpretation of the results

and how OpSeeker can benefit from them. Following are selected possible observations describing users belonging to each of the respective data-sets. B

- **Cluster 1:**
  - Like finance
  - **Dominant fear:** For this cluster was not possible to identify a dominant fear with sufficient support in the data.
- **Cluster 2:**
  - Impatient
  - Overestimate their financial knowledge
  - **Dominant fear:** *I don't have enough disposable income to start investing*
- **Cluster 3:**
  - Admit their financial knowledge is not very good
  - **Dominant fear:** *I don't have enough information to feel prepared to start investing*

The characteristics listed here all have at least 70% support in the data. These are only some of the possible interpretations of the data-set and I expect that if the analysis had been done by somebody more skilled in psychometric or financial-related analysis, the results would have been better. However, even now we can see that clusters possess some distinctive features that differentiate them from each other, and even though the interpretation is far from being straightforward, we have avoided a feared result of clusters being so similar or ambiguous as to prevent any attempt for extracting useful insights.

In future, when a new user starts using OpSeeker's services, he will answer to a subset of questions that were part of the test used in our case study. Based on his answers, it will be predicted (again, using k-means algorithm) to which of the three clusters he likely belongs and therefore it will be possible to personalize the conversation, such as it will put specific emphasis on issues users from given cluster consider more important. His data will then enter the ML database and whole training process will be restarted. The power of this system is in its self-reinforcement, the more users will use the services, the more accurate will be the predictions.

CHAPTER 8

Conclusions

This chapter will describe the achieved goals done by the master thesis following some the key points developed in the project.

8.1 Conclusions and achieved goals

This thesis has accomplished the objectives defined initially. The list of key goals achieved is the following:

- A usable methodology was designed and used in practice on a real world problem
- A case study was designed and executed to collect data for ML model and to learn about biases that influence potential users of OpSeeker's services
- Three distinct groups of users were discovered and from their centroids, usable insights were extracted to personalize the OpSeeker's service in future

Moreover. the methodology that I introduced was proven to be easy to use and able to recommend a good starting point when starting to develop an ML algorithm. It may appear as simplistic, but simplicity was one of its author´s priorities.

However, to develop a similar methodology that would cover more aspects of ML projects and the whole ML environment in general, backed up by a feedback of the ML community, would make up for an interesting research topic.

Regarding the ML model, after performing the clustering and extracting insights about the users, the next step should be to deploy this system in production and to prepare personalized conversations. When this will be done, new users entering the application will be classified based on their first interactions with the system and may enjoy even higher level of financial coaching than currently.

This thesis was an ambitious attempt to use now mature ML tools and apply them on a perspective field of behavioral finance, all in a dynamic environment of a start-up. According to the representatives of OpSeeker, the results the ML model provides are very promising and I believe this thesis will pave a way to enhancing OpSeeker's services and therefore provide bigger value added for both its customers and users.

My education received from EIT Masterschool, both at University of Trento and Politécnica de Madrid was key for developing this master thesis. Entrepreneurial education was important for self management and driving the individual parts of the thesis forward. Subjects about machine learning and software in the cloud gave me necessary theoretical foundations and overview in this field. Moreover, the relative freedom that I have enjoyed when choosing subjects turned out to be equally important, as the subjects I have chosen raised my interest and introduced me to ML using Python and Scala frameworks and skills I have gained were very useful when developing this master thesis.

The most important aspect of this thesis is that it does not solve a theoretical problem, but a real, practical one, proposed by OpSeeker, a start-up. Therefore, even if this thesis

does not introduce any groundbreaking discoveries, it provides solid value for a part of European start-up ecosystem, which I understand is one of the goals of EIT Masterschool.

8.2 Future works

There are several ways how this thesis can be improved and expanded. One way is to run the data collecting process for longer and train the algorithm with bigger data-set. This would enable us to be more confident where inferring something about the users.

K-means is not the only clustering algorithm, there are many more and each has some advantages and disadvantages. Therefore it would be interesting to build the model using some other algorithm and compare the results with K-means.

Even within K-means we can get interesting results when trying to run the algorithm with different value of K. Perhaps this will be the case in future when OpSeeker will have more user data to train the model with, which will enable K-means with value of K higher than current three.

Regarding the proposed methodology, it would be beneficial, albeit time-consuming to send out a questionnaire addressed to wider audience composed of ML professionals, asking for a feedback on the methodology, which I think would significantly raise its value and credibility.

Last but not least, as the methodology always recommends also a "best alternative" approach, it would be interesting to try to build the same solution with this different approach and then compare the experiences during the development.

APPENDIX A

Methodology for selecting ML tools (ready-for-use)

This appendix contains a ready-for-use, condensed version of the methodology introduced in this thesis.

This is a methodology for selecting a suitable machine learning tool, when starting your first ML project as a small or medium company. It takes into account limitations of the company and characteristics of the project. It is described in greater detail in chapter 3 of this thesis.

First step is to answer the following set of questions. In question 2, several options can be chosen. All the other questions are of a single-choice type.

1. **How many people will be dedicated full time to work on the ML project? (A person working part time equals to $\frac{1}{2}$ of full-time)**

   a) 0,5 - 1

   b) 2 - 3

   c) 4 - 5

   d) 6 and more

2. **What are their IT skills? Does somebody of them know any of the following on at least intermediate level?**

   a) Python

   b) Java

   c) C++/C#

   d) Scala

   e) R

3. **What is your total budget (excluding salaries) for SW/HW tools/server run-time, etc.**

   a) less than 100 euro

   b) 100 – 1000 euro

   c) 1000 - 5000 euro

   d) 5000+ euro

4. **What do you want to achieve with ML**

   a) Supervised learning

   b) Unsupervised learning

c) Reinforced learning

5. What is the size of the data-set you will be working with?

a) less than 1000 samples

b) 1000 – 10000 samples

c) 10000 – 50000 samples

d) 50000 – 200000 samples

e) 200 000+ samples

6. Does the data-set contain confidential data?

a) Yes

b) No

Second step is to go through following six tables and sum up point values of the options you have chosen in step one.

Question 1: Number of people				
Technology	a) 0,5-1	b) 2-3	c) 4-5	d) 6+
Java + Weka	3	4	6	7
Scala + MLlib	4	5	6	7
R	5	6	7	5
Python + Scikit	7	7	6	6
BigML	7	3	1	1

Question 2: Previous experience					
Technology	a) Python	b) Java	c) C#	d) Scala	e) R
Java + Weka	1	6	1	1	1
Scala + MLlib	1	1	1	6	1
R	1	1	1	1	6
Python + Scikit	6	1	1	1	1
BigML	3	3	3	3	5

Question 3: Budget (in eur)				
Technology	a) 0-100	b) 100-1k	c) 1k-10k	d) 5k+
Java + Weka	2	3	3	3
Scala + MLlib	2	3	3	3
R	2	3	3	3
Python + Scikit	2	3	3	3
BigML	4	5	1	1
Cloud	2	7	4	2
On premise	6	2	2	6

Question 4: The type of ML			
Technology	a) Supervised	b) Unsupervised	c) Reinforced
Java + Weka	9	10	6
Scala + MLlib	8	6	5
R	9	9	6
Python + Scikit	10	9	0
BigML	5	6	0

Question 5: Size of data-set (in n of samples)					
Technology	a) 1-1000	b) 1k-10k	c) 10k-50k	d) 50k-200k	e) 200k+
Java + Weka	2	3	4	4	4
Scala + MLlib	2	3	4	4	4
R	2	3	4	4	4
Python + Scikit	4	4	3	2	0
BigML	6	3	1	0	0
Cloud	1	3	6	4	1
On premise	6	3	2	3	6

Question 6: Data confidentiality		
Technology	a) Yes	b) No
Cloud	3	5
On premise	6	5

The recommended technology to use is the one with the highest sum of point values and has two parts:

- Programming language / tool (Java/Scala/R/Python/BigML) with related ML library
- Choice of deployment (Cloud or on premise)

APPENDIX B

K-means clustering centroids

This appendix contains centroids resulting from K-means clustering.

Figure B.1: Centroids of the three clusters of users

Cluster n	0	1	2	3	4	5	6	7	8	9
1	0.72059	0.27941	1	1.04E-17	0.83824	0.16176	0.85294	0.14706	0.69118	0.07353
2	0.26136	0.73864	0.96591	0.03409	0.45455	0.54545	0.56818	0.43182	0.36364	0.20455
3	0.93519	0.06481	0.99074	0.00926	0.92593	0.07407	0.93519	0.06481	0.73148	0.13889
Cluster n	10	11	12	13	14	15	16	17	18	19
1	0.23529	0.11765	0.25	0.63235	0.69118	0.07353	0.23529	0.13235	-3.47E-17	0.86765
2	0.43182	0.44318	0.30682	0.25	0.39773	0.21591	0.38636	0.31818	0.07955	0.60227
3	0.12963	0.55556	0.24074	0.2037	0.72222	0.09259	0.18519	0.15741	0.01852	0.82407
Cluster n	20	21	22	23	24	25	26	27	28	29
1	0.89706	0.07353	0.02941	0.30882	0.51471	0.17647	0.91176	0.08824	1.04E-17	0.88235
2	0.57955	0.25	0.17045	0.45455	0.34091	0.20455	0.875	0.09091	0.03409	0.39773
3	0.50926	0.15741	0.33333	0.32407	0.5	0.17593	0.83333	0.15741	0.00926	0.90741
Cluster n	36	37	38	39	40	41	42	43	44	45
1	0.02941	0.5	0.47059	0.16176	0.83824	1.39E-17	0.07353	0.01471	0.80882	0.10294
2	0.13636	0.48864	0.32955	0.15909	0.82955	0.01136	0.20455	0.06818	0.52273	0.20455
3	0.10185	0.55556	0.30556	0.17593	0.77778	0.0463	0.08333	0.02778	0.81481	0.07407
Cluster n	46	47	48	49	50	51	52	53	54	55
1	0.05882	-7.77E-16	0.94118	0.04412	0.13235	0.19118	0.63235	0.07353	0.89706	0.02941
2	0.32955	0.40909	0.26136	0.42045	0.43182	0.10227	0.04545	0.44318	0.51136	0.04545
3	0.24074	0.73148	0.02778	0.53704	0.44444	0.01852	-1.11E-16	0.56481	0.40741	0.02778

Note: Table B.1 contains centroids of the three clusters produced by K-means algorithm, described in chapter 7. Each cluster has 55 centroids, each belonging to a particular payload in a particular answer.

Bibliography

[1] D. Arthur and S. Vassilvitskii. How slow is the k-means method?, 2006.

[2] CH. M. Bishop. Pattern recognition and machine learning, 2006.

[3] Blevinsfranks. Capital gains tax on selling property and shares in spain, 2017.

[4] I. Bobriakov. Top machine learning libraries for ml in 2018, 2018.

[5] J. Brownlee. How to get started with ml in r, 2014.

[6] European Comission. Data protection in the eu, 2018.

[7] F. T. Council. For ml it is all about gpus, 2017.

[8] Pedregosa F. et al. Scikit-learn: Machine learning in python, 2011.

[9] The Guardian. Former caja madrid directors accused of misusing company credit card, 2014.

[10] D. Kahneman. Thinking fast and slow, 2011.

[11] D. MacDorman, K. F. Chattopadhyay. Categorization-based stranger avoidance does not explain the uncanny valley effect, 2017.

[12] A. Malito. This is why most people dont save money for retirement, 2016.

[13] OECD. Oecd better life index, 2017.

[14] S. Pichai. Google ceo: A.i. is more important than fire or electricity, 2018.

[15] U. Pisuwala. Factors that will drive python growth in 2018, 2018.

[16] proft.me. Types of machine learning algorithms, 2015.

[17] Pydata. Pandas, 2018.

[18] E. Ries. The lean startup, 2011.

[19] D Robinson. The incredible growth of python, 2017.

[20] A. L. Samuel. Computer games i., 1988.

[21] Scikit-learn.org. Choosing the right estimator, 2017.

[22] H. A. Simon. The shape of automation for men and management, 1965.

[23] Wikipedia. Apache spark.

[24] Wikipedia. Behavioral economics.

[25] Wikipedia. Chatbot.

[26] Wikipedia. Lopd.