INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Análise inteligente de importação de bases de dados de contactos

DUARTE NUNO MAGALHÃES DANTAS COELHO
Julho de 2018

POLITÉCNICO
DO PORTO

INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MASTER'S THESIS

Intelligent Analysis of Contact Databases' Importation

Author:
DUARTE NUNO COELHO

Advisor:
PHD ANA MADUREIRA

Supervisor:
PHD IVO PEREIRA

A thesis submitted in fulfillment of the requirements for the Masters Degree in Informatics Engineering

July 3, 2018

"Data! Data! Data! I can't make bricks without clay!"

Sir Arthur Conan Doyle

Resumo

INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
Departamento de Engenharia Informática

Mestrado em Engenharia Informática

Intelligent Analysis of Contact Databases' Importation

by DUARTE NUNO COELHO

Plataformas de marketing cujo modelo de negócio assenta em soluções de marketing digital para outras organizações tendem a lidar com diversos canais de comunicação, sendo que o e-mail é, por norma, um dos principais. É natural que, para conseguirem entregar e-mails pelos seus utilizadores, estas plataformas utilizem endereços e domínios próprios. No entanto, isto pode levar a grandes quedas em entregabilidade se fornecedores de serviços de internet colocarem esses endereços/domínios em *black lists*. A razão pela qual esta possibilidade pode ocorrer está relacionada com o facto da plataforma poder vir a ser utilizada inapropriadamente por *spammers* para criar campanhas de *spam*. Assim sendo, existe a necessidade destas empresas tentarem ao máximo que *spam* não ocorra através dos seus sistemas de modo a não prejudicar o seu modelo de negócio. Esta tese propõe a criação de um sistema de prevenção de *spam* baseado na análise preventiva de bases de dados de contactos importadas utilizando uma abordagem de *machine-learning*. De modo a implementar o sistema proposto diversos passos foram desenvolvidos. Primeiramente o problema foi definido e a pesquisa do estado de arte foi realizada. Posteriormente, foi feita uma análise relativa ao valor que a solução traria para a empresa. Tendo o valor criado pela solução sido estabelecido, foi definido o design da solução proposta tendo-se realizado levantamento de requisitos, exposto alternativas de design e definido casos de uso esperados no sistema final. Passou-se, depois, para a fase de desenvolvimento, onde foi realizado o levantamento de *features* importantes para a classificação de *spam*, as quais foram posteriormente utilizadas para treinar diferentes protótipos de classificadores. Sendo que as capacidades destes diversos classificadores foram, depois dos mesmos serem afinados perante as especificações do problema, disponibilizadas através de uma API simples e bem documentada. Finalmente, foi realizado um estudo computacional de modo a analisar o desempenho dos vários protótipos criados para a solução de classificação desenvolvida.

Abstract

INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
Informatics Engineering Department

Masters Degree in Informatics Engineering

Intelligent Analysis of Contact Databases' Importation

by DUARTE NUNO COELHO

Marketing platforms, whose main business model is to provide digital marketing solutions to marketers, usually deal with multiple communication channels, e-mail being one of the main ones. It is only natural, that they need certain domain and e-mail addresses in order to accomplish their task of delivering their users e-mails. However, a problem arises if internet service providers black list the organization's domains or addresses, as the solution's deliverability rate greatly decrease. The reason why this happens is due to the improper use of the platform's capabilities by spammers. It, therefore, becomes a necessity for these organizations to try and block spam from occurring on their system, so as to prevent damaging their business model. This thesis proposes the creation of a spam prevention system based on judgment of user's contact databases while using a machine learning approach. The present work is divided in various steps, which include: a state of the art research, a value analysis, the design of the solution, that solution's development and performed tests/evaluation. Through this work an understandable solution to the presented problem was created based on machine-learning technologies. This solution was thoroughly tested and then made available to external usage through a simple and well documented Application Program Interface (API).

Acknowledgements

I would like to thank both my advisor professor Ana Madureira and my supervisor Ivo Pereira for all the support given up to this point, and their availability to answer my questions whenever needed.

I would like to thank professor Ana Moura for clearing my doubts when confronted with statistical analysis.

I would like to thank E-goi for providing an adequate environment and equipment for the completion of this Master's thesis.

I would like to thank my family for all they have done for me throughout my life, especially my mother, because god knows she deserves it.

Finally, I would like to thank my friends Rui Almeida, André Rodrigues and José Ricardo for always keeping my spirits up.

Contents

List of Figures

List of Tables

Acronyms

ABC	AdaBoost Classifier. ix, x, 59, 62, 64, 72, 74, 75, 76, 79, 82
ANN	Artificial Neural Networks. 20, 22, 23, 24
API	Application Program Interface. iii, 36, 37, 47, 48, 52, 54, 56, 64, 65, 66, 81, 82
AUC	Area Under the Curve. 69, 76, 78, 79, 80
BAG-DT	Bagged Decision Tree. 22, 23, 24, 25, 59, 81
BST-STMP	Boosted Stumps. 22, 24
BST-DT	Boosted Decision Tree. 22, 23, 24
CDB	Contact Database. 3, 4, 21, 22, 29, 30, 31, 36, 41, 48, 49, 50, 51, 52, 53, 54, 56, 58, 59, 64, 65, 66, 68, 70, 75, 78, 81, 82
DB	Database. 50, 53, 54
DKIM	DomainKeys Identified Mail. 14, 17
DMARC	Domain Message Authentication Reporting & Conformance. 14, 17
DT	Decision Tree. 7, 20, 22, 24, 25
EDA	Exploratory Data Analysis. 4, 17, 18, 25, 49
ESP	E-mail Service Provider. 3
FFE	Fuzzy Front End. 27, 28
IP	Internet Protocol. 14, 15, 16, 37
ISP	Internet Service Provider. 3, 6, 15, 16, 17, 36, 61
KNN	K Nearest Neighbours. 20, 22, 23, 24
LLE	Locally Linear Embedding. 11
NB	Naïve Bayes. 7, 20, 24, 25, 59, 81
NBC	Naïve Bayes Classifier. ix, x, 59, 64, 72, 74, 76, 79, 82
NCD	New Concept Development. 27, 28
NCV	Nested Cross-validation. 62
PCA	Principal Component Analysis. 11

REST	Representational State Transfer. 64
RF	Random Forest. 22
RFC	Random Forest Classifier. ix, x, 59, 62, 64, 72, 74, 75, 76, 79, 82
ROC	Receiver Operating Characteristic. ix, xi, 69, 75, 76, 78, 79, 80
RVM	Relevance Vector Machine. 20
SE	Standard Error. xi, 78
SPF	Sender Policy Framework. 14, 17
SVC	Support Vector Classifier. ix, 59, 64, 72, 73, 76, 79, 82
SVM	Support Vector Machine. 7, 20, 22, 23, 24, 25, 59, 81
UC	Use Case. viii, 54, 56

Dedicated to my family and friends.

Chapter 1

Introduction

Regular marketing as been declining in deference of digital marketing in the last years. However, this activity also has threats inherent to the environment where it occurs. Digital marketing platforms create solutions which allow marketers to create marketing campaigns through communication channels such as e-mail. However, many times they end up being used as a way to propagate spam. This Masters work aims to partially fight this occurrences through the use of machine learning algorithms and will be developed through an internship at E-goi.

1.1 Context

In recent years, the act of traditional marketing, a job that involves encouraging people to buy a product or service[1] has been gradually losing relevance to the concept of e-marketing, the process of advertising and selling products and services on the internet[2]. This is a direct consequence of the rapid growth of the World Wide Web in the 1990s and the subsequent burst of the dot-com bubble, which led to a mainstreaming of the internet and related technologies.

E-marketing is a concept deeply rooted with internet, however seems to misunderstand what the internet really amounts to. Most people think about the Web[3] when they hear internet, however this idea is extremely restrictive. Many technologies use internet without directly interfering with the web, such as mobile apps, software and hardware used in customer relationships [1].

Additionally, something often forgotten is that, at its core, e-marketing remains the same as traditional marketing, organizations must always meet their customers needs. More than that, it could be said that the internet was the final nail in the power shift from organizations to users. By giving full control to the user over what he wanted to do, marketers could not keep persons captive while watching advertisements.

The e-marketing environment offers opportunities for new products, strategies, tactics, markets, media and channels, which means it greatly contributes for business as a whole. Individual users now have more power, not only due to the amount of control they have over what they hear and/or watch at any time, but because they have easy access to much more information, that was available, some decades ago. This also means that competition between organizations is both more fair, as little businesses can freely promote their products. But harsher, as a product that is not good will quickly disappear in face of its competitors. Not only that, but good products tend to do better, as the possible audience of a satisfied user is much greater than in the past, which represents an extreme version of word to mouth, where users themselves market the product [1].

[1] According to the Cambridge Dictionary: https://bit.ly/2MtbYis
[2] According to the Cambridge Dictionary: https://bit.ly/2N5zzGZ
[3] Portion of the internet that supports graphical user interface.

Taking into account these facts, the use of e-marketing became an integral part of any modern organization, and in order to ease the process of creating, managing and distributing campaigns over different channels, organizations like E-goi were created.

1.2 Organization

E-goi (Figure 1), created by Miguel Gonçalves, is an organization based on Matosinhos, Portugal and is currently defined as a Software as a Service multichannel marketing automation platform. As the definition implies, it employs an array of communication channels in order to bring marketing campaigns to fruition, such as: e-mail, sms, voice, smart sms, push notifications, and, in the near future, Facebook Messenger. The organization's mission is to create efficient digital marketing solutions which can be intuitively used by its clients. Its vision is to be a brand that is recognized by their customers [2].

E-goi directs its business model, i.e. marketing automation solutions to all types of organizations, from micro enterprises to large multinationals, while also not forgetting small and medium businesses, which can greatly benefit from some of the organization's proposed payment plans [2]. E-goi's payment options include:

1. Free plan [3] - allows for up to five thousand contacts and fifteen thousand non cumulative e-mails per month.
2. Normal plans [4] - whose price adapts to the customer needs, meaning the price depends on the needed database and e-mail message limits.
3. Corporate plans - which is dependent on discussion between the corporation and E-goi.

This offer extends over more than fifty countries the main ones being: Portugal, Brazil, Colombia, Spain, and most of Latin America.

E-goi is, presently used by various renowned national and international organizations, such as Porto Editora,The Phone House, Fnac, IKEA, Sephora, Altice, Jerónimo Martins Group, Sonae and many more.

The platforms main capabilities include, but are not limited to [5]:

- Creation and management of contact lists. This includes the ability to use those lists to create campaigns based on default templates through an easy to use editor;
- Access to useful statistics which contribute to the optimization of the marketing process of any business, such as: results triggered by different kinds of newsletters, online behaviour of customers, and success/deliverability rates of created campaigns.
- Multiple implementations with a variety of widely used applications like WordPress, Salesforce, Magento, Prestashop and more.

FIGURE 1: E-goi's Logo

1.3 Problem

Currently, E-goi has nearly three hundred thousand users (checked on February 2018) users from all over the world. Over 250 new account sign-ups occur each day (in average) and hundreds of thousands of people log into its platform every hour. These users then use E-goi to send more than 20 million e-mails per day. All of this so that they can more easily and reliably transmit to their clients the news and information they really need to know about their business.

Each of the new accounts created daily import their Contact Databases (CDBs), which can contain over a million subscribers each, in order to create their campaigns. This all happens while existing accounts continue to, frequently, import even more contacts.

Currently, the organization uses a rule based system in their CDB importation system. This system takes into account various aspects of the user's information (such as account type, location, etc) and the contents of the database to be imported to reach a conclusion about whether the database should or not be imported, in order to avoid various types of malicious behaviours (namely, spamming the e-mails that were loaded).

Internet Service Providers (ISPs), term which will be used in an interchangeable fashion with E-mail Service Provider (ESP) throughout this document, are, in a general manner, strongly opposed this type of activity (spamming). It is not uncommon for e-mail addresses or even domains associated with this type of practices to get marked as a target whose e-mails should be blocked or even fully deleted. This means that the organization should be very careful when approaching this point, as one wrongly classified/detected spammer could be responsible for a block to the e-mails sent by many other users.

Due to that, the presence of a CDB importation analysis system is crucial. However, the type of rule's based system, which is currently in use, is extensive and not easily editable in case a change needs to be made to the way the importations are judged. It may even miss key characteristics that would make for good indicators of the user's intent to realize malicious actions.

As such the main problem presented through this Masters work is the creation of a system that can be more easily editable than the current one, while also being more efficient and accurate, and also taking into account more, possibly relevant, criteria.

1.4 Objectives

In order to prevent potentially malicious behaviours and improve aspects related to the platform's deliverability rate and overall reputation, E-goi wants to research and develop machine learning mechanisms in order to perform an intelligent analysis of CDBs' importation. This will help solve some of the problems presented by the current system such as the low adaptability degree and the complex edition process.

Based on the many registry occurrences created from the organization's platform, several behaviours can be recommended at the moment the importation occurs, and if changing the system's behaviour becomes a necessity, the algorithm responsible for the CDB evaluation can simply be re-trained, which greatly reduces the amount of effort necessary in relation to the current system.

The following specific objectives and results are worth mentioning:

- Analysis of the current CDB importation system;
- Perform state of the art research for the topics encompassed by the problem;
- Research regarding machine learning algorithms;

- Research of available tools for machine learning algorithm development;
- Development and implementation of a new CDB importation intelligent analysis system;
- Analysis and evaluation of the success rate of developed prototype algorithms.

1.5 Approach

A concise view of this problem's approach would be composed by the following phases:

1. Bibliographic research regarding the state of the art including:
   - Machine learning definition;
   - Classification of machine learning approaches and techniques;
   - Tools for developing machine learning algorithms.
2. Realization of a value analysis relative to the proposed solution;
3. Gathering and manipulation of available registry data into a format appropriate for analysis and other operations;
4. Analysis and description of the main decisions necessary to develop and implement the desired solution;
5. Model the solution while taking into account the researched technologies as well as good practices of software engineering;
6. Creation of CDB intelligent importation system prototypes (based on different classification approaches);
7. Specification of the simulation plan and evaluation metrics to be used in the algorithm's computational study;
8. Actual computational study and reaching conclusions.

1.6 Document Structure

This document is composed by five chapters in order to present the developed solution to the reader in a complete, but easy to understand fashion.

The first chapter shown is the Introduction were a brief contextualization and description of the problem to which this thesis pertains is presented. Additionally, the Maters work's main objectives, as well, as approach taken to accomplish them are also introduced.

Afterwards, a chapter regarding Literature Review is put forward in order to better understand the various underlying topics that are important to the problem at hand. These topics are Machine Learning (ML) and Problem / Deliverability respectively, and are followed by a current view of the state of the art regarding, both technologies used in Exploratory Data Analysis (EDA) and ML, as well as previous works that act upon spam detection using ML techniques.

After this, it is explained how this Masters work introduces innovation and generates value for both the client and the organization in chapter Value Analysis. This is done through various methods and the use of standard models such as the value proposition and business model canvas.

Next, comes the Design chapter, were possible alternatives for the problem's solution are presented, as well as a general view to the data available to be used in both exploratory analysis as well as the machine learning approaches.

What follows is a section relative to Development of the solutions, where the various metrics used to compare different prototypes and statistic tests utilized to verify the validity of those comparisons are presented. Additionally, sensibility tests are performed in order to assert what are the most relevant features, both in the current system, and in the developed prototype's case.

Chapter 2

Literature Review

In this chapter relevant information relating to the topics involved in the current problem, namely, machine learning and deliverability, will be presented. In the case of deliberability, emphasis will be put on its correlation with spam, and how ISPs fight it. For machine learning, an introduction to the different existing types will be made while also putting forward example techniques used in each of them. Afterwards, some articles regarding the use of machine learning for spam detection and binary classification will be shown.

2.1 Machine Learning

The term machine learning was first used in the 1959's book *Some Studies in Machine Learning Using the Game of Checkers* by Arthur Lee Samuel [6], an American computer scientist who would later receive the computer pioneer award by IEEE in 1987.

In his book, Samuel described how he had trained a computer to effectively play chess by programming it to take conclusions from playing against itself. As pointed out *"at the end of each self-play game a determination is made of the relative playing ability of Alpha, as compared with Beta, by a neutral portion of the program"* [6]. Up until this point in history, the general assumption held in computer sciences was that a computer was not capable of performing a task without having been explicitly programmed to do so. Samuel's program, however, proved the opposite. It disprove this belief, thus the term "machine learning" was coined as "*a field of study that gives computer the ability to learn without being explicitly programmed*", based on a definition given in the book.

In 1997 a more formal definition of the term was proposed by Tom Mitchell[7] which wrote that a program could be defined as being based on machine learning if he could obtain better results at a given task as long as he obtains more data (experience) and there is still room for growth, which is a very similar definition to that of learning when applied to living beings.

As for core concepts of the area, one cannot effectively talk about machine learning without first knowing what a feature and a label are. To put it in simple terms:

- A feature is an input, it is a characteristic or attribute of the points or objects present in a data-set (e.g. in the case we want to predict a person's gender based on its height and weight, then those characteristics are features).
- A label, on the other hand is a conclusion, an output, it is an attribute used to classify a point of a data-set based on its features (e.g. in the case we want to predict a person's gender based on its height and weight, then the gender to be concluded is the label, which can vary from case to case).

Machine learning can be said to have three main research focuses. However, all of these focuses tend to interact with each other. Meaning research into one of them, usually, also leads to discoveries in the others. These focuses are [8]:

1. Task-Oriented Studies - the development and analysis of learning systems to improve performance in a predetermined set of tasks (also known as the "engineering approach");

2. Cognitive Simulation - the investigation and computer simulation of human learning processes;

3. Theoretical Analysis - the theoretical exploration of the space of possible learning methods and algorithms independent of application domain.

From these three research points, it can be said that machine learning approaches usually target either task-oriented studies, or cognitive simulation. Two recent examples of these areas, respectively, can be observed in the AlphaGo algorithm [9] (developed by Google's DeepMind) and Cleverbot [10], an algorithm to have conversations with humans (created by British AI scientist Rollo Carpenter).

In general, the branch can be divided into three major types of approach, and each approach has various methods that can be applied to it. The major types of machine learning (and some of the most popular methods for each) will be discussed in the following sections. These methods are only a few out of the many that exist for each approach and are described in order to give the reader some general knowledge regarding the type of algorithms used in each approach. A general overview of the machine learning algorithms in existence can be observed in Figure 2.

2.1.1 Supervised Learning

In Supervised Learning the labels for a set of values are appropriately defined and known beforehand, therefore, the computer learns from a previous classification performed by a third party. One could say it follows a teacher or role-model [12]. It includes the sub-areas of classification and regression.

Classification is, normally, adopted when the label used to classify input data is a finite set of discrete values [12] (e.g. the rank attributed to a diamond's cut). Possible techniques and classification methods include:

- Decision Tree (DT) - are structures that classify data in a case by case fashion by sorting instances based on their features' values. The nodes in the tree represent a feature of a data instance/point to be classified and the branches spreading from it represent a value that the node can acquire. By flowing through the tree since the starting node it is possible for an instance to be classified based on its features values [13]. An example of a decision tree can be seen in Figure 3.

- Naïve Bayes (NB) Network - an NB network is a model for probability relationships between a set of features. Its structure is represented as a directed acyclic graph, where features lead to one another, while being independent from elements that are not their parents. Each root feature has an occurrence probability and each child element has various probability's of occurring or not based on its parents probability of occurrence or non-occurrence [13]. An illustrative example can be observed in Figure 4.

- Support Vector Machine (SVM) - is a technique that tries to divide the classification space pertaining to a set of data instances, in order to classify them, by using a dividing plane known as hyper-plane (since it may affect a large number of dimensions). To find this hyper-plane, the algorithm first tries to find the optimal separating planes based on the various points present in the classification space. After

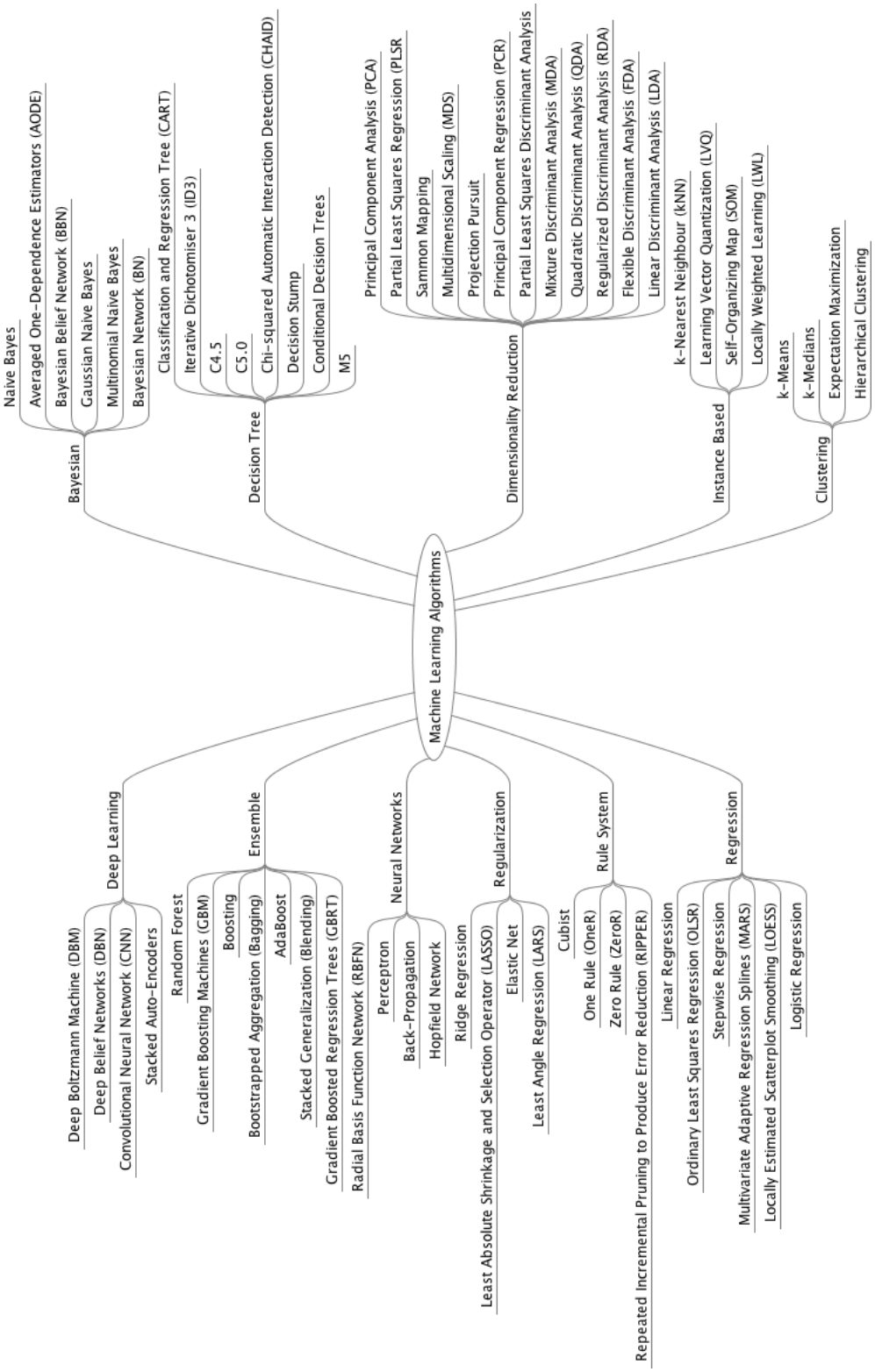


FIGURE 2: Overview of Various Machine Learning Algorithms [11]

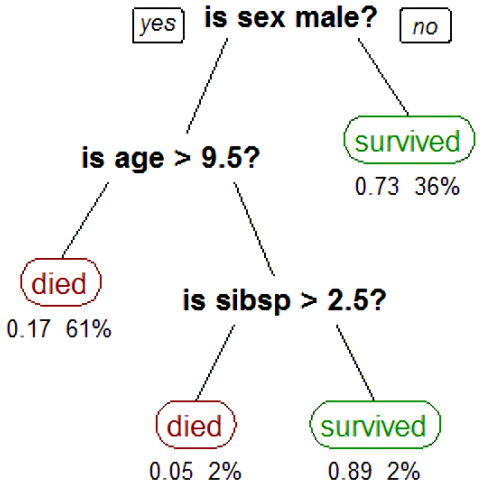


FIGURE 3: Example of a Decision Tree showing survival of passengers on the Titanic ("sibsp" is the number of spouses or siblings aboard). The figures under the leaves show the probability of survival and the percentage of observations in the leaf. [14]

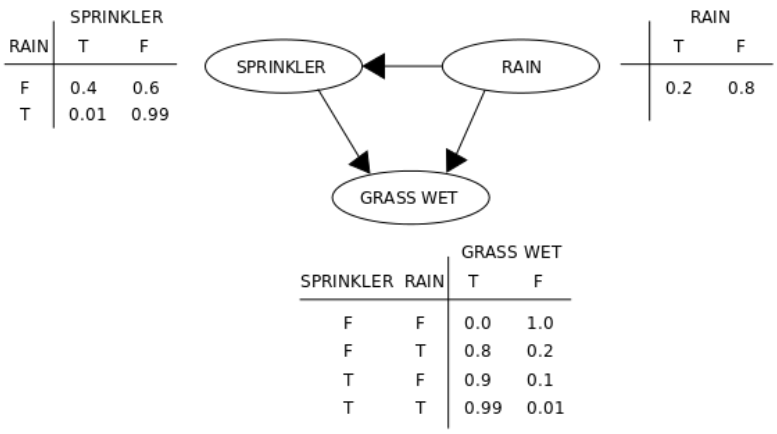


FIGURE 4: Example of a Bayesian Network where the probability of the grass being wet is calculated based on the probabilities of raining and/or the sprinklers being turned on. The probability of the sprinklers being turned on is also affected by wether or not rain occurs. [15]

having obtained these, it then adjusts the planes basing its judgments on the points that lie on their margin, it takes into account factors like the distance to which the points are from the plane and the concentration of margin points in each side of it. These points that affect the dividing hyper-plane are known as support vectors, and eventually the first planes generated are adapted into a final well adjusted hyper-plane which effectively divides the data [13]. Refer to Figure 5, for an image pertaining to this process. Most problems, however, involve non-separable data that has no well defined hyper-plane, in those cases the use of kernel functions is employed. These functions map data to higher-dimensional spaces where hyper-planes can be successfully created, those spaces are referred to as *feature spaces* [13].

Regression is, commonly, used when the output that classifies the input data consists of one or more continuous variables [12], e.g. the final price of a diamond based on his attributes. Possible techniques and methods used for regression include:

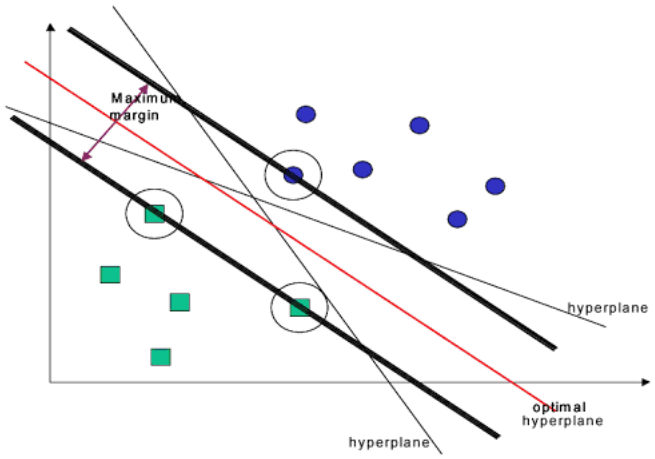


FIGURE 5: Example the Creation of an Hyper-plane [13]

- Linear Regression - this type of model as existed for a long time in of statistical analysis and it is a very important tool in that field. The model is composed by trying to predict the linear equation that generates an outcome Y given a vector of inputs $X^T = (X_1, X_2, ..., X_p)$, in this case the model would be:

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

  In machine learning the term $\hat{\beta}_0$ is known as the bias, in regular statistics it is also known as the intercept. A frequently used method to fit a linear model is the least squares method [16].

- Ridge Regression - this method is a shrinkage method, meaning it shrinks the regression coefficients by imposing a penalty on their size. Shrinkage methods retain a subset of the data as predictors and discard the rest, subset selection produces a model that is interpretable and has possibly lower prediction error than the full model [16].

2.1.2 Unsupervised Learning

In Unsupervised Learning no classification is provided for a set of values used as an algorithm's input [12], therefore this approach is normally used for clustering and dimensional reduction. The purpose of clustering is to find similar groups based on the features present on the input data [12], that is to establish possible relations between data points based on the similarity of their features. Possible clustering techniques and methods include:

- K-means - a method for finding clusters and cluster centers based on unlabeled data. To operate it, one chooses the number of wanted clusters, and K-means iteratively moves cluster centers in order to reduce variance in that cluster. It does this by repeating two steps until convergence after randomly selecting the starting centers from the data to be clustered. The steps include: identification of the points that are closer to a specific cluster center than any other and compute the means those points' features which turns into the new center for that cluster [16]. An example displaying the K-means iterative process can be seen in Figure 6.

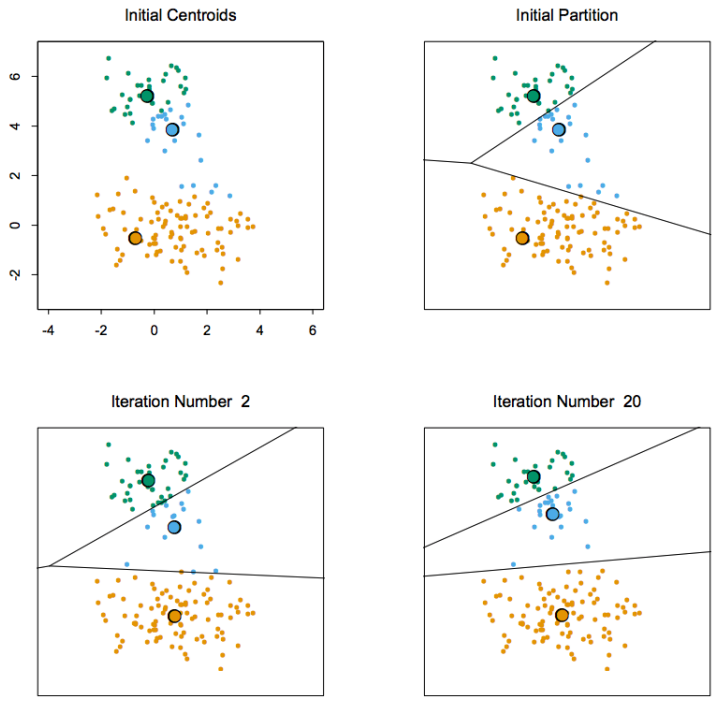


FIGURE 6: Example the Iterative Process Followed by K-means [16]

- Hierarchical Clustering - a method that requires the user to specify a measure of dissimilarity between groups of data instances. By repeatedly joining clusters that have the least differentiation between them (*agglomerative*, bottom-up), or dividing (*divisive*, top-down) clusters into their most dissimilar components, an hierarchy is formed were the top element contains all points of inputted data and the bottom elements are composed of only one point each [16]. A result of hierarchical clustering can be seen in Figure 7.

Dimensional reduction is applied when there is a necessity to project the data from a high-dimensional space down to two or three dimensions. It is usually used for the purpose of visualization, or when there are an extreme number of features (e.g. in face recognition problems). Methods and techniques that can be used for dimensional reduction include:

- Principal Component Analysis (PCA) - is a multivariate technique that evaluates data in which instances are described by various inter correlated variables. Essentially PCA aims to extract the most important information from the available data and reduce its size by only keeping the most important components. It does this by computing new variables which are called principal components and are obtained by doing linear combinations of the original variables. The first principle component is always the one with highest data variation, the second one follows the same principle, however, it has to be orthogonal to the first one, same happens to the third one, however that would have to be orthogonal to both the first and second components, and so on [17]. An example of the application of PCA to a reduce three dimensional data to two dimensional data can be seen in Figure 8.

- Locally Linear Embedding (LLE) - a method that "*computes low-dimensional neighborhood preserving embeddings of high-dimensional inputs*" [19]. LLE transfers its input data to a single coordinate system of lower dimensions. It does this by exploiting the local symmetries of linear reconstructions. It is normally used for complex structures that

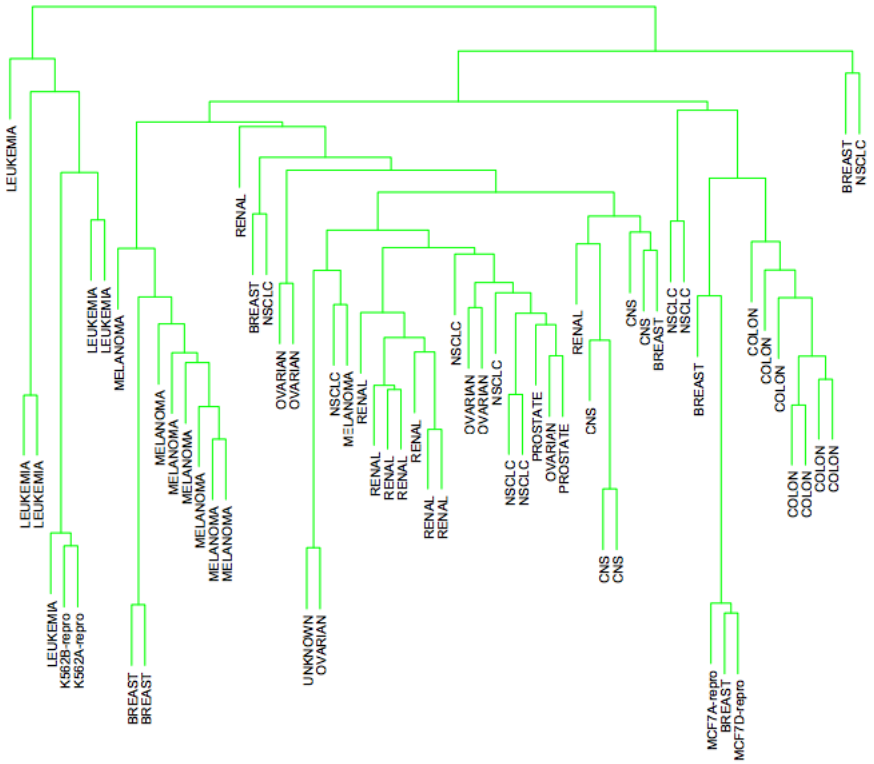


FIGURE 7: Example of Agglomerative Hierarchical Clustering (Based on Human Tumor Data) [16]

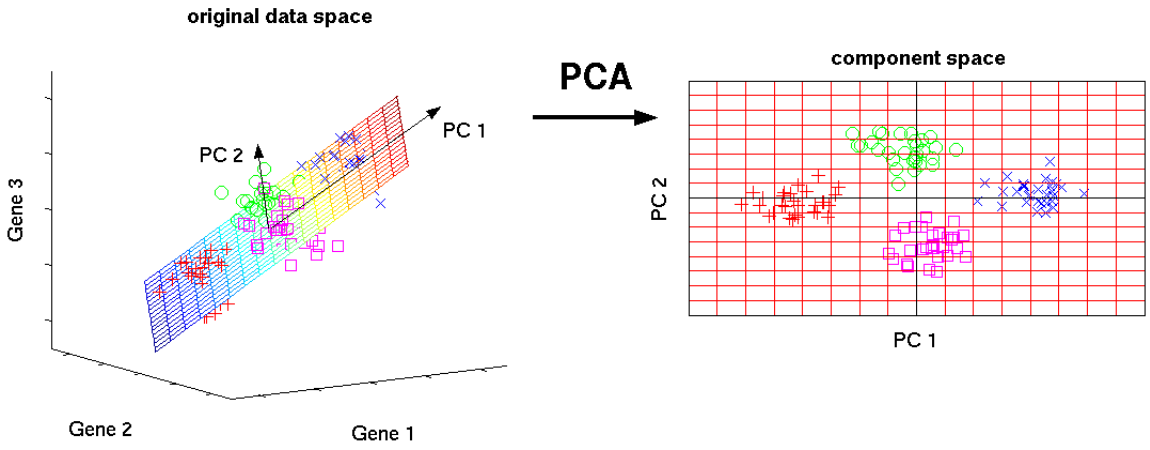


FIGURE 8: Example of PCA Reduction of Data in a 3-D Space to a 2-D One [18]

present folds in their data distributions [19] (e.g. human faces). An example of the process taken by applying an LLE algorithm can be observed in Figure 9.

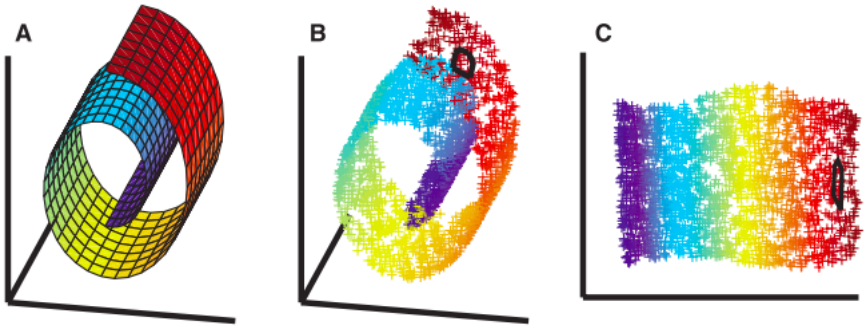


FIGURE 9: Example Steps of the LLE Dimensional Reduction Process [19]

2.1.3 Reinforcement Learning

Reinforcement learning is an approach which tries to make algorithms find the actions to take in a given situation in order to maximize the accomplishment rate of a certain objective. It does this, by either giving a "reward" to the algorithm when he achieves certain objectives, or by delivering a "penalty" when he fails to do so. It is a similar process to what one's mind may take while playing a game like chess, where there are always one or two optimal moves to make [12]. Two algorithms encompassed by this approach are:

- Q-learning - these methods are a type of model-free reinforcement learning where agents learn through their interaction with their environment. To put it in a straightforward fashion, a computational agent does an action and is either rewarded or penalized, while also ending up in a different state then the one he was in. By performing all types of actions, in all states, repeatedly, over a period of time an agent effectively learns which ones are the best (the most rewarding, or less penalizing) [20]. While this algorithms may still be affected by a policy[1], they gradually grow closer to optimal conditions independently from the present one [22].
- Monte Carlo Methods - these serve as a way of solving regular reinforcement learning problems based on averaging the results obtained by running diverse tests. The methods require a model, although it may be purely representative of the possible transactions. By running several tests (episodes that may require the completion of various actions) and updating the policy between each of them, via the average results obtained up to that point, a close to optimal solution can be obtained [22].

Having introduced the core concepts and approaches related to the machine learning aspect of this work, it is now time to present the ones pertinent to the problem's theme.

2.2 Problem / Deliverability

An e-mail flux can be judged while taking into account various types of relevant metrics. Nonetheless, in almost every case, the most important one is deliverability.

[1] A description of an agent's behaviour. It is essentially a function which describes the correct action to take given a state. Policies can either be stationary, which means only the state matters for an action decision, or non-stationary, meaning time is also an important factor in the decision [21].

Deliverability is, as one would expect from the name, the ability to make a given e-mail flux reach completion. That is, the ability to successfully deliver an e-mail to a subscriber's inbox. It is often used as a metric that some marketers as well as digital marketing platforms use to gauge the likelihood of their e-mail campaigns reaching their subscribers [23]. Different factors can affect the deliverability of an e-mail flux. The main ones are:

- The reputation of the involved parties, specifically:
  - Domains
  - Internet Protocol (IP) addresses
- The presence and validity of authentication. Through mechanisms such as Domain Message Authentication Reporting & Conformance (DMARC)[2], Sender Policy Framework (SPF)[3], and DomainKeys Identified Mail (DKIM)[4]
- The e-mail message's content

These factors are, additionally, the main tools used to classify a given e-mail as spam or regular e-mail. Which is only to be expected since spam is the main reason why e-mail deliverability is a study topic to begin with. That is to say the deliverability rate of a given domain tends to improves as its suspicion level of being a spammer drop. However, the relation between the two is not easily described. To understand the full picture one has to first understand the concept of spam.

2.2.1 Spam Contextualization

Spam is normally defined as irrelevant or unsolicited messages sent over the Internet, typically to a large number of users[5]. That definition is faulty, or not entirely correct, as was stated in [24], just because an e-mail is unsolicited it woill not warrant its classification as spam, for instance, commercial bulk e-mailing may give the user information about the product he really needs at a specific moment in time. Additionally, although many spam messages are indeed irrelevant that also does not justify their classification as spam, as an example the reception of an e-mail containing a comedy skit may be irrelevant information, but if the user is amused the message will very likely not be classified as spam.

Due to spam not fitting the mold of a "normal" concept due to potential contradictions in its definition, it is easier to think of the concept as being an abstract combination of diverse characteristics possessed by an e-mail message. These characteristics which contribute to a message's classification as spam are [24]:

- E-mail Content - Certain topics are more likely to be accounted as spam than others, e.g. a simple message informing a person of an event happening at their organization is less likely to be spam than an e-mail informing the reader of five easy methods on how to lose weight;
- Intent of the Sender - The attitude displayed by anyone towards a virus ridden message sent willingly by some unknown party is different than the one which would be displayed towards the same image sent by a friend which was unaware of the fact. Which means that the intent of the sender also partially affects a message's classification as spam;

[2]Refer to: https://dmarc.org/
[3]Refer to: http://www.openspf.org/
[4]Refer to: http://www.dkim.org/
[5]According to the Oxford Dictionary: https://en.oxforddictionaries.com/definition/spam

- Consequences to the Receiver - The consequences of an e-mail message's content towards the receiver affect its classification of the message as being spam. Given two users that receive the same message relating to weight loss, one may welcome the message, while the other may be offended by its contents. Nonetheless the more detrimental the content of a message is to a user the more likely it is to be considered spam;
- Consent of the Receiver - An user is more likely to classify messages that he has not solicited as spam in relation to messages towards whose reception he has given consent;
- Relationship Between Sender and Receiver - The relationship, or lack of thereof, between the sender and the receiver affects a message's classification. A person should always be more lenient towards messages sent by people they know personally;
- Accountability of the Sender and the Degree of Deception - The use of a valid and real return e-mail address, as well as the presence of a sender's e-mail address which remains consistent towards multiple messages make it less likely that an e-mail message is considered spam. This as to do with the same thought process applied when we say that a masked person is more likely to perform a robbery than an unmasked person, that is, an unmasked person is more accountable for its actions, and therefore less likely to commit malicious acts;
- Number of Identical E-mails Sent - The quantity of similar e-mails received by a person also affects spam classification. Evidently, a person is more likely to classify a message as spam the more times that message is sent to him;
- Illegality - Not every type of spam violates the law, but the messages that do are, in almost every case, spam;
- Message Size - A messages size contributes for its classification. Specially in the case of system administrators, bigger messages are more likely to be classified as dangerous spam than smaller messages.

The combination of these factors to various degrees are what make a user consider an e-mail message as spam. Nonetheless, that classification is subordinate to an user's interpretation, which is not a reliable metric if taken into account in a case by case fashion. Due to that, ISPs normally classify an IP address and/or domain as being a spammer only when multiple users flag him as being one, when it send e-mails to a spam trap (a concept which will be explained in the next topic), or when it keeps sending messages that are not delivered correctly, either because errors always occur or because the receiver's e-mail addresses do not exist.

Apart from the definition that is attributed to the core concept, spam is undoubtedly the biggest problem present in any e-mail system, and this problem as been growing over the years (as one can see in [25]). That being the case one should known what mechanisms can and are employed by ISPs in order to try and mitigate this threat.

2.2.2 Spam Classification by ISPs

For e-mail marketers, as well as digital marketing platforms, not allowing spam to pass through filters is a very important task, as is equates to a much better delivery rate, which in turn translates to a larger profit overall. This, however, is only plausible if one is aware

of the means ISPs use in order to block regular spam. In this way, it is possible to create differentiation points between marketer's messages and regular spam, meaning it is harder for ISPs to block it.

Normally, ISPs employ an array of techniques in order to prevent and possibly classify an e-mail and/or domain as spam/spammer, these include [26]:

- Image and URL Blocking - this is done since, many spammers use a unique URL embedded image or simple URL in order to identify if an e-mail is valid, whereupon they proceed to send more spam to that e-mail;
- Complaint Rates - addresses and domains whose e-mail receive to many complains from its users are, normally, classified as spammers. However, since ISPs tend to recommend their users to follow a policy of "fire first, ask questions later" in regards to spam (they advocate that you denounce a message as spam at any amount of suspicion, as some smarter spammers tend to use the usually useful unsubscribe link as a means of validation) this means that a lot of valid e-mail ends up classified as spam;
- Bounce Rates - bounce refers to the act of an e-mail message not being correctly delivered due to either a failure having occurred somewhere along the way to the receiver (soft bounce, these can normally be fixed by simply re-sending the message), or because the e-mail no longer exists, or did not exist to begin with (hard bounce). ISPs tend to consider domains that have high bounce rates as spammers, their argument is that if bounces are ignored then that domain's owner cannot be a genuine marketer as ignoring them is something that would also negatively affect him;
- White-lists - these are quite literally lists kept by ISPs where specific IP addresses and domains known to be legitimate are held. Usually, in order to get white-listed a marketer needs:
  - A secure and correctly configured sending server;
  - A reverse DNS set-up on the servers;
  - A low complaint rate (preferably below 1 per cent);
  - Bounce rates under 10 per cent.
- Content Filters - these typically work by searching through an e-mail for specific words, that are common in spam, and then giving it a score based on the words detected. If the e-mail passes a certain threshold it is considered spam;
- Testing Tools - these are tools that analyze e-mail and score it based on a variety of its characteristics, whereupon it may be treated as spam if it passes a certain amount of points. These are, normally, based on the "Spam Assassin" filter, which is widely used by ISPs. However, they tend to specify more criteria to the filter, which means that even if a message passes a default "Spam Assassin" test, it is not guaranteed to pass that filter in an ISP context;
- Black-lists and Spam Traps - black-lists are the opposite of the white-lists presented above. These are lists were IP addresses and domains that have a reputation of sending spam are stored, and the presence of a domain in one of them affects his delivery rates very negatively. An address/domain may be placed in a black-list in two different situations:
  1. If people consistently complain about e-mail sent from a specific address;

2. If a given address sends e-mail to a spam trap. Spam traps are specific e-mail addresses that are not publicly disclosed, but instead are made invisible in websites pages (for instance, in places like the contacts page of a website and other such places). Because they are not disclosed to the public, and remain generally unobtainable unless one takes action in order to see them, only spammer have regular access to these e-mails as they use e-mail harvesting software in order to form big lists of potential targets.

- Accreditation Schemes - these method consists in e-mail analysis performed by a third-party that later presents its results to the ISP. However, good results on these tests are not a guarantee of a message's delivery;
- E-mail Authentication - that is, guaranteeing that a person is not faking the identity of the sender. This can be accomplished through different methods, although no set standard exists, some examples of tools to be used are DMARC, SPF and DKIM.

Taking into account the techniques employed by ISPs to block/classify spam it is easier for e-mail marketers to not only improve their deliverability, but also, create their own filtering systems that also contribute for the reduction of overall spam.

2.3 State of the Art

2.3.1 Technologies

The technologies used in a project tend to affect the final result one ends up getting. It is only common sense that if one wants to paint something he will use a brush and not an hammer. In the same way it makes sense that, before starting a project, one should take a brief look at the technologies commonly used in the type of work it will require.

2.3.1.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a middlestep between data collection and true statistical analysis. In this phase, it is natural to use various forms of visualization techniques and simple metrics, in order to understand what the collected data can be used for/in. It effectively serves as a pointer that may inform you if the path that you were thinking of walking in terms of analysis was indeed the correct one [27]. EDA is constituted by various areas, which include:

- Identification of data types;
- Data visualization;
- Potential outlier (points/instances that are clearly different from the norm) detection;
- Identification of data distributions;
- Revealing possible trends.

Nowadays, data analysis is relatively simpler than it was some years ago thanks to the great quantity of tools that exist and allow to perform it, such as Google's FusionTable, OpenRefine and Orange (open source data visualization platform), even relatively simple tools like microsoft excel allow to perform actions related to it. However, there are two specific tools that seem to be much more used than all others, one relatively new and another one that as been active for quite some time. These tools are, respectively:

- Pandas (Python data analysis library) - is an open source library for Python that provides high-performance, easy-to-use and rich data structures and functions, aimed at making the process of working with structured data fast and easy. Its developments started back in 2008, initially the author only wanted to accomplish a set of requirements that he felt were not readily answer by any other solution. Over the years, Pandas became an extensive library capable of solving most of statistical problems. Making only a brief summary of the essentials that the package brings to the table, we get the following capabilities [28]:
  - Introduction of three data structures to Python. Namely series, data-frames, and index objects;
  - Re-indexing of points/instances;
  - Ability to drop entries from axes;
  - Data indexing, selection, and filtering;
  - Arithmetic and data alignment;
  - Function application and mapping;
  - Sorting and ranking of imputed data based on a given criterion.
- R - is an integrated software suite that allows easy data manipulation, calculation and visualization, and exists since 1993. It can be regarded as an implementation of the S language, developed by Rick Becker, John Chambers and Allan Wilks [29], as well as the base for the S-PLUS systems. R is, additionally, widely used as a statistics system, although its functionalities surpass those of a typical statistics system. As referred in [30], the authors prefer to think of it as an environment within which many classical and modern statistical techniques are made available for use. This is made possible through the utilization of packages. The main capabilities of R include [30]:
  - Effective data handling and storage;
  - Numerous operators for calculations on arrays and matrices;
  - A large collection of tools for data analysis;
  - Graphical options that allow ease of use for both data analysis and simple display either on computer or hand copy;
  - A well developed, simple and effective programming language, which allows for the creation of all algorithmic structures anyone familiarized with programming expects (conditionals, loops, user defined functions, etc.).

2.3.1.2 Machine Learning Algorithm Development

Similarly to the case of EDA, nowadays, there are a number of tools available that make the process of developing a machine learning algorithm much easier than what it once was. From these numerous tools, two of them stood out in relation to the others, namely Weka [31] and Scikit-learn [32].

Weka [31] is a machine learning workbench that aims to ease the process of applying a machine learning technique to a real world problem. In opposition to other such projects, Weka tries to emphasize a working environment for the domain specialist rather than a machine learning expert, which makes it more accessible than other tools. The technology as existed since 1994, throughout the years it has remained in constant development and now it is a staple in the area. Its main functions include, but are not limited to:

- Provide a great deal of interactive tools for data manipulation;
- Allow numerous ways to perform result visualization;
- Perform database linkage;
- Cross-validation capabilities;
- Comparison to rule sets.

On the other hand, Scikit-learn [32] is a Python package that offers implementations for a great variety of machine learning algorithms, while maintaining an easily used interface integrated with its core language. By doing this, it answers a growing need for statistical data analysis capabilities by non-specialists in the software and web industries. Under the hood Scikit-learn uses: the Numpy package, which provides a base data structure used for data and model parameters; the Scipy package, which provides efficient algorithms for linear algebra, sparse matrix representation, special functions and basic statistical functions; and the Cython package, which allows for the usage of a language for combining C and Python, making it easy to reach the performance of compiled languages with Python-like syntax and high-level operations. Additionally, in terms of performance Scikit-learn is very well of against its direct competitors (other Python machine learning oriented packages), only finding competition in external solution in other languages and/or platforms. A comparison between Scikit-learn and other packages can be observed in Table 1. In it, the computation time for some algorithms implemented in the major machine learning toolkits, accessible in Python, are compared. For comparison purposes the Madelon data set composed od 4400 instances and 500 attributes is used. For further imformation refer to [32].

TABLE 1: Classification Time in Seconds for the Madelon Data-set in Different Python Machine Learning Packages (adapted from [32])

	scikit-learn	mlpy	pybrain	pymvpa	mdp	shogun
Support Vector Classification	**5.2**	9.47	17.5	11.52	40.48	5.63
Lasso (LARS)	**1.17**	105.3	-	37.35	-	-
Elastic Net	**0.52**	73.7	-	1.44	-	-
K-Nearest Neighbors	0.57	1.41	-	**0.56**	0.58	1.36
PCA (9 components)	**0.18**	-	-	8.93	0.47	0.33
K-Means (9 clusters)	1.34	0.79	*	-	35.75	**0.68**

- : Not implemented
* : No convergence within an hour

2.3.2 Previous Works

The idea of using machine learning for either filtering/blocking spam is not new. Through the last decade, specially with spam becoming more prominent, works in this area have become relatively common.

2.3.2.1 Spam Detection

Most of the works in this area focus on the performance of different approaches and/or use of different algorithms in the act of spam detection. That fact can be clearly observed through the two following works.

In the paper, *A comparative study for content-based dynamic spam classification using four machine learning algorithms* [33], the authors propose the execution of an empirical evaluation of four different machine learning algorithms, relatively to their spam classification capabilities. The encompassed algorithms include: one based in NB; an Artificial Neural Networks (ANN); a SVM; and a Relevance Vector Machine (RVM)[6].

The various approaches were evaluated based on different data-sets and feature sizes in terms of: accuracy, the percentage of e-mail correctly classified by the algorithm; spam precision, the ratio of spam e-mail correctly classified from all e-mail classified as spam; and spam recall, the proportion between the e-mails which the algorithm managed to classify as spam and the true number of spam e-mails present in the testing set [33]. Precision and recall have direct correlation in binary classification. If the positive class' recall grows the precision of the negative class will drop, and if the precision of the positive class grows the recall of the negative class will lower. In that way, these two metrics can be said to be complementary and interdependent.

The results are depicted in Table 2 and Figure 10 (details regarding the experiment that originated these results can be read at [33]). It is possible to conclude that:

- The NN classifier is unsuited to be used by itself in this task, as it is the one that consistently gets the lowest results;
- Both the SVM and RVM classifiers seem to slightly outperform the NB classifier;
- Comparatively to SVM, RVM provides similar results with less relevance vectors and faster testing time, although the learning procedure is slower overall.

TABLE 2: Accuracy for the SpamAssassin/Babletext Corpus Using the Different Methods [33]

Size (training: testing)	Method (SpamAssassin / Babletext)			
	NB	ANN	SVM	RVM
20:80	92.7%/93.8%	85.3%/87.7%	95.2%/96.0%	96.1%/93.9%
30:70	91.3%/90.4%	86.6%/89.8%	94.8%/95.4%	95.1%/94.6%
40:60	90.7%/92.0%	86.1%/86.3%	96.3%/96.4%	94.8%/94.6%
50:50	94.0%/94.5%	92.4%/85.9%	95.8%/94.6%	94.2%/95.8%
60:40	92.2%/92.2%	83.5%/90.2%	97.0%/95.9%	95.6%/96.3%
70:30	91.8%/93.1%	84.0%/84.2%	96.0%/96.1%	96.0%/96.5%

Similarly to the previous case, in the Paper, *A Relative Study of Spamming Detection Using Machine and Non Machine Learning Classifier* [35], the authors perform a comparative evaluation of different machine learning approaches targeting the problem of spam detection. In this case, the algorithms chosen as candidates to solve the problem were NB, Random Forest[7], K Nearest Neighbours (KNN)[8], and SVM.

[6]Classification technique in which a linear model based on a NB approach which is functionally identical to a SVM is used [34]

[7]Classification algorithm achieved through the combination of various DT's, in which every particular tree's decision counts as a vote for a specific label to be attributed to an instance, and the label with most votes is the one used [35]

[8]Classification technique in which a data instance is classified with the same label as the most common one present in its K nearest neighbours [35]

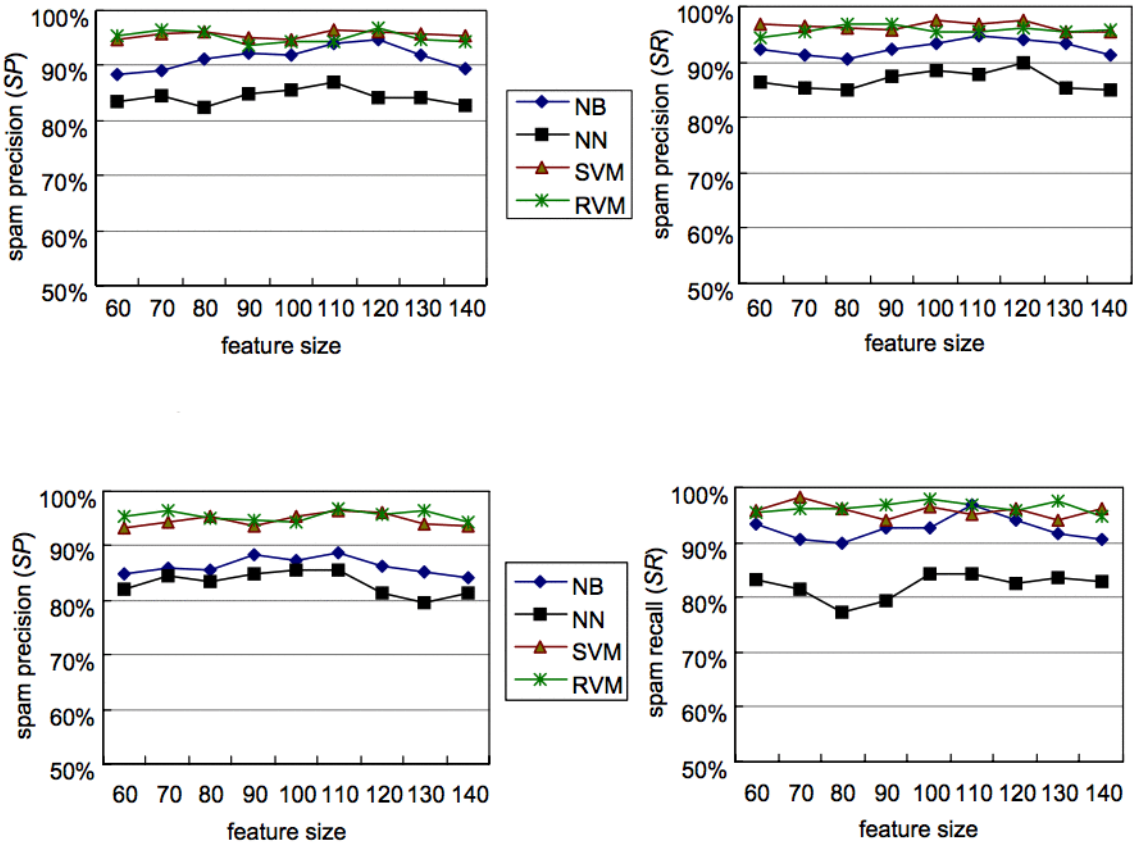


FIGURE 10: Differences in Spam Precision and Recall for Different Methods and Different Corpus (Top Two Graphs for SpamAssassin Corpus and Bottom Two for Babletext Corpus) [33]

In paper [35], specific data related to the performance of each algorithm is not provided, however, a set of the characteristics to take into account when choosing any of them is presented, which can be seen in Table 3.

TABLE 3: Specific Characteristics of Each Approach (adapted from [35])

Approach	Overall Conclusion
NB	Resilient in regards to noise in input data. Can be implemented quickly in relation to other classifiers.
RF	Deals well with imbalanced input data.
SVM	The use of a Kernel (which corresponds to a dot product in a usually high-dimensional feature space) allows this approach to perform separations in complex boundaries, which would be hard in the other's case.
KNN	The algorithm is quite time consuming, as such it is recommended for it to be paired to with a feature selection algorithm in an attempt to decrease processing times.

2.3.2.2 Binary Classification

In this work, the focus does not, immediately, fall in the classification of the e-mails present in an imported CDB, but in the statistics generated by the current system before a decision

is made (statistics like the number of suspicious e-mail, or e-mail that is/seems to be a spam trap, etc). These should not be inherently wrong, as their creation is based on rules made through the experience of the organization's workers in an area in which they could be considered experts, as such, a good approach should be to try and use them to create a simpler classifier than one which would use all e-mails present in a CDB as a direct input.

There are various works pertaining to the performance of classifiers in binary classification problems, that is, in problems that require the division of elements into two groups, which are relevant, since the final decision a classifier for the present problem should reach is weather or not to import a given CDB.

One such work presenting a comparison between the performance of supervised learning algorithms in relation to binary problem scenarios is *An Empirical Comparison of Supervised Learning Algorithms Using Different Performance Metrics* [36]. In this work the authors used various data-sets and performance metrics in order to compare seven supervised learning algorithms, these include:

1. KNN - This algorithm is used to perform classification. Its decision rule works by simply providing a class label to the input pattern based on the class labels represented by the K-closest neighbors to that pattern [37] (some examples of this method's usage in spam and text categorization cases can be read in [38] and [39]);

2. ANN - An artificial neural network is a mathematical or computational model that tries to replicate the structure and/or function of biological ones. The network is constituted by smaller units (artificial neurons), which combine in order to form a more powerful computation device. By using these networks one can do things like model complex relationships between variables and spot patterns in large data quantities [40] (some examples of work accomplished with this type of algorithm in spam detection can be read about in [41] and [42]);

3. DT - Structures that classifies data in a case by case fashion by sorting instances based on their features' values, refer to section 2.1 (examples of their usage in spam detection and filtering can be read about in [43] and [44]);

4. Bagged Decision Tree (BAG-DT) - Bagging is a method to enhance the accuracy of unstable classification methods like decision trees. In bagging, a number of bootstrap data-sets, are generated, each consisting of various cases drawn at random from the main one. A decision tree is built for each of the samples. The predicted class corresponding to a new input is obtained by a plurality vote among the those classifiers [45] (examples of this technique applied to spam prevention and filtering can be read at [46] and [47]). Random Forest (RF) is a sub-type of bagged decision trees where feature and data re-sampling randomization occur;

5. Boosted Decision Tree (BST-DT) - A boosting algorithm is one in which various algorithms with a poor error rate (weak learners) are combined to form an ensemble that has a much better performance (strong learner) than all its composing parts [48] (examples of BST-DT's usage in spam detection and filtering can be found in [49] and [48]). In the case of DT's the most commonly mentioned boosting algorithm seems to be AdaBoost;

6. Boosted Stumps (BST-STMP) - The theory behind these is the same as the aforementioned point, however stumps are no more than single leveled decision trees, which may pose worse results overall [36] (no relevant exemplifying papers about BST-STMP's in spam detection and filtering were found);

7. SVM - A technique that tries to divide the classification space of a set of data instances, in order to classify them, by using a dividing plane known as hyper-plane, refer to section 2.1 (relevant literature relating to the usage of SVM's in spam detection and filtering includes [50] and [51]).

These algorithms were then evaluated according to three different metric groups (each containing three metrics), namely: threshold metrics, ordering/rank metrics and probability metrics. Additionally, this comparison between algorithms occurred using seven different binary classification problems (each referring to a different data-set whose sources can be checked at [36]) which had the characteristics presented in Table 4.

TABLE 4: Binary Problems Characteristics (adapted from [36])

Problem	**Number of Attributes**	**Training Size**	**Testing Size**
ADULT	14/104	4000	35222
COV_TYPE	54	4000	25000
LETTER.p1	16	4000	14000
LETTER.p2	16	4000	14000
MEDIS	63	4000	8199
SLAC	59	4000	25000
HS	200	4000	4366

The results of the performed comparisons were then normalized and averaged over the nine testing metrics (which included accuracy, area under the respective ROC curve, average precision, and others), in order to create a table that would allow to understand the performance of each algorithm, in each problem, and in relation to each other. The result can be observed in Table 5. In order to better understand the variance of the various algorithms across metrics and problems, the observation of Table 2 at [36] is strongly advised.

TABLE 5: Normalized Score of Each Algorithm by Problem
(Using the Nine Performance Metrics) (adapted from [36])

	COV_TYPE	ADULT	LETTER.p1	LETTER.p2	MEDIS	SLAC	HS	**MEAN**
ANN	0.8294	0.9425	0.9555	0.9327	**0.9885**	0.9431	**0.9625**	**0.9363**
SVM	0.8433	0.9235	**0.9643**	**0.9846**	0.8838	0.9274	0.9552	**0.9260**
BAG-DT	**0.9314**	**0.9836**	0.8969	0.8642	0.8486	**0.9515**	0.8541	**0.9043**
KNN	0.8844	0.8504	0.9284	0.9478	0.7472	0.8717	0.8380	**0.8668**
BST-DT	0.9069	0.8557	0.8903	0.8757	0.5752	0.8943	0.8652	**0.8376**
DT	0.7429	0.9342	0.7853	0.7719	0.6287	0.8594	0.7007	**0.7747**
BST-STMP	0.7022	0.8729	0.3782	0.6239	0.7421	0.8078	0.6907	**0.6883**

From the results obtained from the various performed tests the authors reached diverse conclusions. The main ones were that [36]:

- Overall, based on the mean score presented in Table 5, ANN's, SVM's and BAG-DT were the most efficient algorithms;
- SVM's success is probably due to the low dimensionality of the problems used, which is known be a favorable environment for the algorithm;
- Although when taking into account probability metrics (that is, "*metrics that are uniquely minimized, in expectation, when the predicted value for each case coincides with the true underlying probability of that case being positive*") BST-DT perform poorly overall, if we take into account only threshold and rank metrics, this algorithm is the best performing one out of all;

- Although KNN performs well if attributes are scaled, by their gain ratio, for instance. However it is not as competitive as other choices;
- Single DT's did not perform well comparatively to other methods regardless of type;
- Lastly, BST-STMP's does not work nearly as well as BST-DT's. However these do outperform single DT's in five of the six performance and rank metrics.

In terms of computational costs, the conclusions reached by the authors were that ANN's were the most costly algorithm, since they had a lot of parameters that needed to be adjusted. Next in the list were SVM's that in equal fashion also had many parameters that needed to be adjusted, although still less ANN's. KNN always proved less computationally costly than SVM's, which may be due to the training sets being composed of only four thousand point. In the case of DT's simple ones are the cheapest overall, however those present poor performance. BST-DT's present good performance, however their boosting process is not easily parallelized, which makes them relatively expensive. The same cannot be said for BAG-DT which present both good performance and low computational cost since the bagging process is easily executed in parallel [36].

Another work worth mentioning in the area of classification techniques comparison is *Supervised Machine Learning: A Review of Classification Techniques* [52]. In this paper the authors introduce a series of classification techniques while also presenting a short review about them. The algorithms approached in the paper were mostly introduced before hand except for one and include: DT; ANN; NB; KNN; SVM; Rule-learners[9].

After introducing all these algorithms and providing a review of pros and cons of each of them individually. The author took a series of conclusions based on his previous work. The conclusions taken were the following [52]:

- SVMs and ANNs tend to work much better when dealing with data containing high dimensionality and continuous features, while logic-based systems (such as DTs and rule learners) perform better when dealing with discrete features;
- It is generally accepted that KNN is very sensitive to pointless features. To a lesser degree, the presence of these features can make the training of ANNs much slower and more impractical;
- DTs tend to not perform well when diagonal partitioning is a necessity, due to the division of the data space always occurring orthogonally to the axis of one variable and in parallel to all other axes. In these cases, ANNs and SVMs are more adequate;
- NB approaches tend to be very quickly trained. Similarly DTs can also be trained quite fast, at least when compared with ANNs and SVMs;
- In terms of computational costs KNN has the highest memory impact both during training as well as classification. In terms of parameters that need to be tuned by the user the most costly ones are ANNs and SVMs which have more parameters than the remaining techniques;
- Logic-based algorithms (like DTs and rule-learners) are much easier to interpret than the cases of ANNs, SVMs and KNN;

[9] Algorithms which create a set of rules that represent each possible class (label) by disjunctive normal form. Their goal is to build as small a rule-set as possible while also being consistent with the training data. They differ from a DT in that they only evaluate the quality of the set of instances covered by the candidate rule, while a DT evaluates the average quality of a number of disjointed sets (one for each value of the feature being tested) [52].

- DTs have an opposite operational profile in relation to the one present in NB (if one is accurate in a situation the other is probably not). Logic-based algorithms have similar profiles, as do ANNs and SVMs;
- There is no single algorithm that clearly outperforms all others.

A summary of these and more conclusions were provided in paper [52] and can be observed in Table 6.

2.4 Summary

Throughout this section various technologies and works pertaining to this Masters work's operation areas were presented in order to contextualize their current state. It is now in order that some conclusions are taken from the different points presented.

Relatively to the technologies used for both EDA and machine learning oriented development, there are no clear winners. All technologies presented have more than enough functionalities to be able to solve any problem that may appear during the Masters work's course. More than that, all of them have solid recognition in their communities. Nonetheless, a decision was made in regards to technology choice.

For EDA the chosen technology was R, due to its lower learning curve, as well as knowledge of its great visualization capabilities. Additionally, R has more support for statistical analysis in general, due to the context of its creation, which should be a favorable differentiation point. For algorithm development the chosen solution was Scikit-learn, this decision was based on it being a Python package was the main differentiation point, since it allows to a closer development process to "regular" programming.

Tackling the judgment of which machine learning algorithms to choose, the decision is not so abstract. In the various works presented, there are algorithms which constantly present good results. The clear best performer are SVMs, both in regular spam classification and in binary classification problems, they consistently present good results. Additionally, they have good attributes in regards to classification speed and accuracy, which should be important factors when taking into account the problem's context. The second best algorithm is the group of various approaches related to DTs, especially BAG-DTs. DTs generally present interesting characteristics regarding their tolerance to almost any type of data thrown at them. Plus, by bagging them it is possible to easily obtain accuracy levels which would require much more work in the case of a single tree (in which case, repeated tuning would be necessary). These particularities make them an attractive choice. One other algorithm was chosen to serve as a kind of control point in regards to the previous two, that was NB. Since the algorithm is simple, relatively fast to develop and train, easily understood, and has been widely used since the formation of this study area, it should prove as a good comparison point to the other two approaches.

It is worth noting that the technological decisions presented above are by no means final. They were taken as a starting point to the work's development while taking into into account the state of the art research presented beforehand. If a chosen technology proves to be insufficient, or if one of the alternatives shows a characteristic of special relevance towards the work's development, then that alternatives will potentially also be used.

TABLE 6: Algorithm Comparison (Adapted From [52])
(**** represents the best and * the worst performance)

Algorithm	DT	ANN	NB	KNN	SVM	Rule-learners
Accuracy	**	***	*	**	****	**
Learning speed relative to the number of attributes and instances	***	*	****	****	*	**
Classification speed	****	****	****	*	****	****
Missing value tolerance	***	*	****	*	**	**
Irrelevant attributes tolerance	***	*	**	**	****	**
Redundant attributes tolerance	**	**	*	**	***	**
Highly interdependent attributes tolerance	**	***	*	*	***	**
Dealing with discrete / binary / continuous attributes	****	*** (not discrete)	*** (not continuous)	*** (not directly discrete)	** (not discrete)	*** (not directly continuous)
Tolerance to noise	**	**	***	*	**	*
Dealing with danger of overfitting	**	*	***	***	**	**
Attempts for incremental learning	**	***	****	****	**	*
Transparency of classification	****	*	****	**	*	****
Model parameter handling	***	*	****	***	*	***

Chapter 3

Value Analysis

Creation of new value is not something trivial. The main way in which companies acquire/produce value for themselves is by first having new product/feature ideas and then exploring them. This process however, is extremely risky, there are numerous cases where an idea seen as brilliant does not find support from the market, there are also cases where an idea starts being developed only to find out there is some specific constraint that does not allow for its implementation. Each idea a organization has chosen to try out requires funds and resources, and if they fail those are effectively wasted.

In this chapter, the use of various tools and techniques is employed in order to understand if the developed idea really creates value for the contextual business.

3.1 Innovation / New Concept Development

Innovation is, normally, a concept associated with change. In this context change usually results from creation, it is something inherent to the concept of innovating, as anyone would realize by simply checking its dictionary definition which reads "*the introduction of something new*"[1].

In business terms, however, the word's definition has added meaning. It is not enough to introduce an idea or concept, it is also necessary to analyze and implement it in order to generate value that a customer will feel he can afford to pay for[2]. This is the case because in business innovation comes forth in response to a demand, the idea is not so much an "eureka" moment as it is a simple answer to something the customers needs or wants, and from the transformation of that answer into a reality that generates benefits to both the client and the organization innovation is created.

Before talking about the new concept development model, it is necessary to provide some insight so as to how it came into being.

The typical innovation process is divided into three main parts, as is possible to observe in Figure 11. From these three parts arguably the one that is most essential to the process is the Fuzzy Front End (FFE), which is the area were the new idea or concept serves as base for the entire process is generated or suggested. However, standard ways of judging this step of the overall innovation process are few and far in-between, this is due to the difficulty generated from common terms and definitions that allowed for the comparison of FFE elements not existing across companies.

In order to solve this problem, Peter A.Koen and his co-authors suggested, in a book published in 2002 titled *The PDMA ToolBook for New Product Development* [53], a theoretical construct that was meant to provide insight and a common terminology for the FFE. To

[1] According to the Merriam-Webster Dictionary: https://www.merriam-webster.com/dictionary/innovation

[2] According to the Business Dictionary: http://www.businessdictionary.com/definition/innovation.html

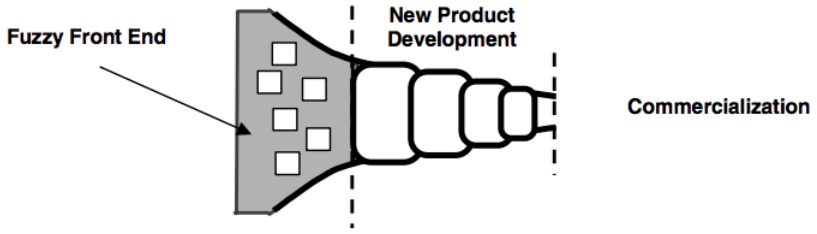


FIGURE 11: The Innovation Process [53]

do this, this step of the process would be divided in various similar points derived from many FFE experiences in different companies. Throughout their discussion of both the similarities and differences present in the data regarding experiences using FFE, they had obtained from different people, they realized that a sequential process was not appropriate for this step of the overall process. From the identification of similarities, discussion and solving of differences, and the realization that a sequential structure was not adequate the New Concept Development (NCD) model was created.

The developed NCD model consists of three key parts [53] (Figure 12):

1. The engine or bull's-eye, which is composed by organization elements that influence in a direct or indirect fashion the activities present in the inner spoke area;

2. The inner spoke area, which defines five controllable activity elements (opportunity identification, opportunity analysis, idea generation and enrichment, idea selection, and concept definition) of the FFE;

3. The influencing factors, consisting of organizational capabilities, the outside world, and the enabling sciences, basically, any indirect factors that my influence an idea's development and are hard to predict or control by the organization itself.

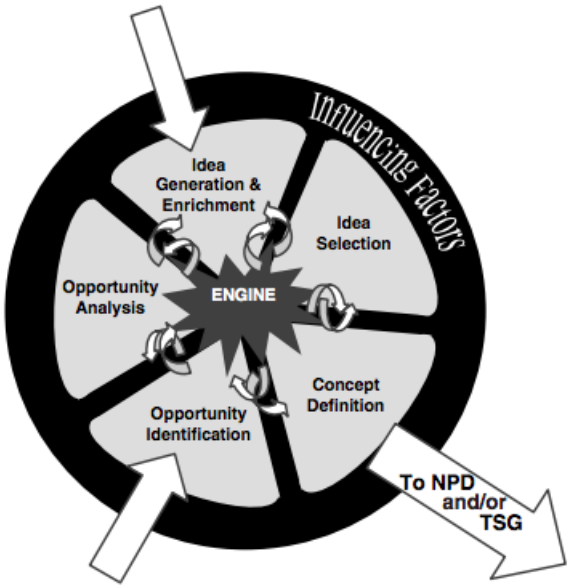


FIGURE 12: The New Concept Development Model [53]

By applying these different parts of the NCD model to the current context we can find and analyze factors that may be previously unknown to us, as well as discover how different characteristics of the problem are affected by both the organization's and environmental components. So as to do that, the second point of the model (the five controllable activity elements) will be applied to the problem at hand, as it is the one that mainly deals with the origination of the core concept/idea. The obtained result is presented in the following sections.

3.1.1 Opportunity Identification

The opportunity presented in this context is created by a variety of factors.

First and foremost, there is the obvious fact that when a organization's up-time on its services decreases so does its productivity, which unquestionably results in a negative impact in both its sales and reputation. Applying this logic to the problem at hand, we can understand that a organization whose main business is e-mail marketing should try its best to avoid getting its e-mails and domains blocked, because that would cause loss of profit.

Secondly, an opportunity is bound to appear where ever a solution needs improvement, for instance, if a simple piece if fabric cannot block water well enough, then we apply a water repellent substance to it. In the same vein, if a importation judging solution already in place cannot follow its expectations at a reasonable level, it is only natural that a new one should be used.

Thirdly, although related to the second reason, a product may be able to be changed to adapt to a certain weakness. However, one should account for the time and effort it takes for it to be altered, if it takes a long time and a lot of effort to adapt a solution to new problems in an environment in which there is a tendency for them to appear, then an opportunity for a new solution may appear.

It is the combination of these three points in combination with the great potential for evolution present in the involved technologies that first generated the idea of creating a CDB judging solution based on machine learning techniques.

3.1.2 Opportunity Analysis

In order to ground the opportunity created from the combination of factors described in the previous point, it is necessary to somehow measure the advantages and sacrifices one would sustain by pursuing that opportunity. To do that, one can use various methods, such as comparison tables relating to an opportunity's benefits, as well as analyzing analytic data which affects it direct or indirectly.

In this case, a SWOT analysis (a method normally used to assist the formulation of a strategy [54]) was used to try and expose a more general vision of the advantages and disadvantages associated with taking the opportunity presented. This SWOT analysis can be seen in Figure 13, and its main points are the following:

- Strengths - By taking hold of the presented opportunity a solution that is more efficient and accurate, easier to edit and to adapt can be created;
- Weaknesses - This, however, warrants that a certain price will have to be paid for the development of that solution, and since it will, in most cases, be different than the current one, an adaptation period will be necessary. Additionally, depending on the technologies use the solution may not be as easily understood as the current one, which could be a negative factor;

- Opportunities - By pursuing the opportunity, it may be possible to further increase its potential by making the implemented solution use newer technologies that are rich in potential, the same can be said if its development process takes into account the current state of spam, since it would be more prepared to the cases at hand;
- Threats - Potential threats to the pursuit of this opportunity are scarce and have a low probability of occurring, factors like the organization's bankruptcy or the technologies involved suddenly growing deprecated could be used as examples.

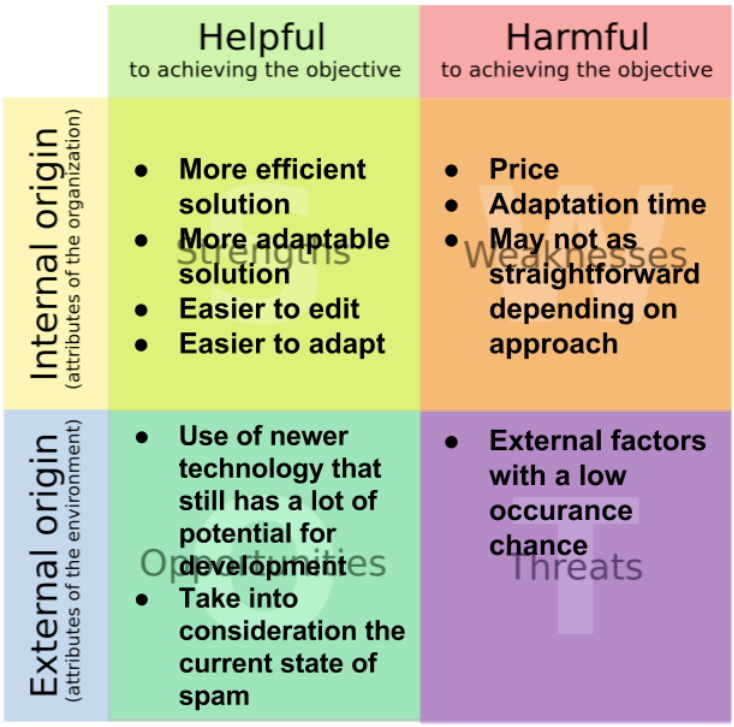


FIGURE 13: Opportunity SWOT Analysis (adapted from [55])

By comparing the various benefits and risks, it is possible to conclude that chasing the opportunity created by the components presented in the first element is a worthwhile objective, that may very well lead to profit for the organization.

This fact is made even more relevant if one takes into account the increase of spam e-mail over the years, which as even surpassed a total of 99% of all e-mail traffic in the last year (seen in Figure 14). This makes the development of a spam preventing system an advantageous venture.

3.1.3 Idea Generation & Enrichment

Having concluded that the opportunity was worth pursuing, it is now necessary to generate possible ideas and/or other solutions that may answer to its requirements.

Taking into account the factors that generated the opportunity, as well as the strengths and opportunities related with its implementation, a search was conducted for techniques an technologies that could be used in this scenario.

After analyzing the search's results, various organization members discussed potential solutions. In the end, the solutions that were presented as possible alternatives to the solution were:

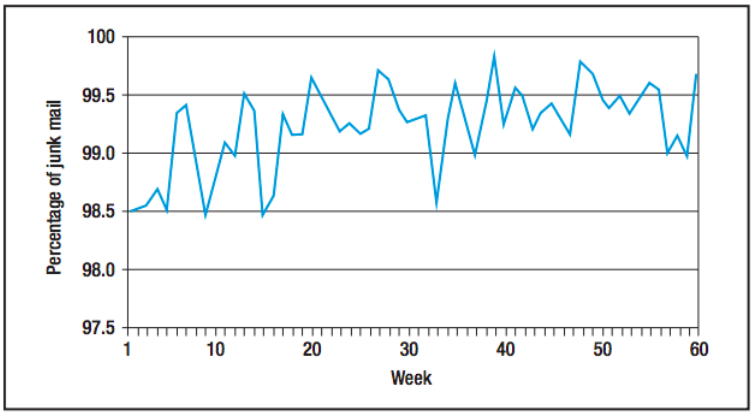


FIGURE 14: Growth of Spam Percentage over 60 Weeks up to June 2017 [25]

- Restructuring and changing the judging methodology of the current CDB judging system;
- Implementing a new system based on a machine learning approach.

3.1.4 Idea Selection

Since the first option presented in the previous point would require a great time investment and would not guarantee a better overall performance, it was decided to opt for the second option. That is, to take a machine learning approach when developing the new solution. Because the technology is relatively recent, as a lot of room for growth, and allows for many easy ways to change its functioning (judging) method, this solution was considered to be the right choice between the two options.

3.1.5 Concept Definition

The opportunity presented by the factors pointed in the first step of this list can be answered by a CDB judging system based in machine learning techniques and technologies, which allows for a more efficient and accurate detection of possibly malicious databases, therefore bringing indirect profit to the organization by extending the possible up-time of its services, while also making the solution more easily editable/adaptable in case a change need to be made to its functioning method.

3.2 Solution's Value

3.2.1 Value

The term value can be used in a variety of ways. Usually value as different meanings depending on whether it refers to the producer or the customer. The producer may derive value from loyal customers, as those are beneficial to business in a number of ways. Clients' value in the other hand tends to stem from conditions like the quality of a product/service they bought, the effort they placed in acquiring it, the social interactions experienced with the producer, etc. Even if the same item and user are taken into account, its value may differ as it is dependent on context [56].

In regards to the value a certain item offers to a user, Miles [57], in his 1961 work, presents four different types:

1. Use Value - which relates to the properties that allow a product or service to accomplish the task it was designed to do;
2. Esteem Value - accomplished by the properties/characteristics that make a person want the product;
3. Cost Value - the sum of all the costs involved with the production of an item or service;
4. Exchange Value - which is comprehended as the characteristics that allow for the easy exchanging of an item for something else that is wanted (products such as collection books, in a good state, are a good example, as their value may even increase over time).

The above information presented in Miles' work [57] allow us to understand that there is no universal definition for value. Nonetheless, four tries at a definition, created by Zeithaml [58], based on is 1988 study were:

1. Value is low price - "*Some respondents equated value with low price, indicating that what they had to give up was most salient in their perceptions of value*";
2. Value is whatever I want in a product - "*Other respondents emphasized the benefits they received from the product as the most important components of value*";
3. Value is the quality I get for the price I pay - "*Other respondents conceptualized value as a trade-off between one 'give' component, price, and one 'get' component, quality*";
4. Value is what I get for what I give - "*Finally, some respondents considered all relevant 'get' components as well as all relevant 'give' components when describing value*".

This only serves as further confirmation that value depends not only of context cut also of the one judging it (something that becomes fairly obvious thought daily occurrences).

Regarding the case at hand, its value is not as easy to define as a regular product, as it is a service being developed for the organization's own profit. Nonetheless, if we try to apply Zeithaml's definitions of value to the present case, we would obtain a result akin to what is present in Table 7.

3.2.2 User Value

Activity is understood, usually, as the actions we take to accomplish a goal, *"Human activity exists as action or a chain of actions*" [59]. Experience, however, is different, it involves a dimension of as it requires conscious awareness, and thus, though about a given action's process [59].

Through experience we think about a product, we rationalize it, and ultimately we address it meaning. Users interact with products within the context of their goals, needs, cultural expectations, physical context, and emotions. And products, with their tangible and intangible qualities, influence the way users interact with them. User value is created as a result of the interaction between what the product provides and what the users bring in terms of their goals, needs, limitations, etc [60].

In simpler terms, it would not be wrong to say that user or customer value is born from the way in which an individual experiences a product. Through this conclusion, two divisions could be traced:

1. There are products which can be directly (physically) experienced (e.g. a car)

TABLE 7: Solution's Value through Zeithaml's Definitions

Definition	Value
Value is low price	The solution is being developed in the context of an internship, additionally it doesn't make use of paid software, that being the case its overall price is low for the organization, as its only costs are transport money and some of the employees time.
Value is whatever I want in a product	As in this case the product is being tailored to fit the exact necessities of the organization there is no doubt it will answer to most, if not all, of what the client wants.
Value is the quality I get for the price I pay	The solution will be accomplished according to engineering good practices, and will use modern and innovative technology. Taking into account the overall low price referred in the first definition, the value generated should be very good.
Value is what I get for what I give	What was said in the last definition applies here as well. Additionally, as the solution will make the process for which it is being developed easier than the one current one in place, it generates even more value.

2. There are products that can only be experienced by the effect they have on our daily activities (e.g. electricity)

This last type could be said to be valued through "change". That is, we only truly consider its value when a transformation occurs that affects our daily activities (normally negative alterations, e.g. an electric outage).

The current Masters work's solution could be said to be part such cases. The organization already has an importation decision system in place that is used many times a day in its business processes and does its job without disturbing work flow most of the time.The problem arise when a change in either the decision model is necessary, or a client importation is wrongly blocked or allowed to pass, as those three situations can cause changes in the organization's regular work flow.

The new solution would prove offer more value over the current one in regards to the facts referred above, due to how it would require less attention as it would be more easily change and, possibly, better at deciding which importations should or not be blocked.

3.2.3 Perceived Value

"*Customer perceived value (CPV) is the difference between the prospective customer's evaluation of all the benefits and all the costs of an offering and the perceived alternatives*" [61]. Total customer value is perceived as the monetary value of the bundle of all benefits (economic, functional, etc.) customers expect from a given idea. Be it because of the products, services, personnel, or even the image involved. Total customer cost is understood as the bundle of costs that customers expect to take in the effort to identify and acquire a specific deal [61]. Summarizing:

$$CPV = Total_{CustomerValue} - Total_{CustomerCosts}$$

Applying this information to the work at hand, we get that the organization's (that in this case can be considered the customer) perceived value is given by the difference between the benefits they will reap from the solution, such as: easier evaluation model edition; potentially better decision taking; incorporation of up and coming technology that will have a great impact in future projects; interaction with outside corporations; etc. And the costs they intake by trying to accomplish this solution: spend resources to find an

intern, spend money and time to allow the intern to understand the problem, spend time of said intern to research and test approaches to the imagined solution, between others.

3.2.4 Longitudinal Perspective of Value

In is 2003 work, Woodall [62] introduced a longitudinal perspective in regards to customer value. This perspective was divided in four phases/positions, that were both temporal and, consequently, cumulative, in the matter of customer value. Meaning each phase contributed for a part of the overall value that a customer attributed to a product and what succeeded in a given phase could affect the result of others.

These four phases through which customer value is perceived were:

1. Ex Ante (Pre-purchase)
2. Transaction (Moment of trade)
3. Ex Post (Post-purchase)
4. Disposition (After use)

If applied to the case at hand the benefits and sacrifices to the customer according to the four phases suggested by Woodall would look similar to what can be observed in Table 8.

TABLE 8: Solution's Value through a Longitudinal Perspective

	Benefits	**Sacrifices**
Pre-purchase	Expectation of a good solution capable of doing a better judging than the one used before it	Prerequisite of searching and understanding the very basics of a new technology in order to fully understand the advantages
Moment of trade	Acquisition of a good solution that can be easily upgraded and changed throughout the years thanks to being based on machine learning.	Costs related to learning and understanding the new solution fully (partially fixed through documentation)
Post-purchase	Full understanding of the solution as well as the way it can be changed and edited to adapt to new contexts	Work related to the preparation of data-sets necessary to retrain a machine learning algorithm in case of changes.
After use	Adaptation capacity through the use of a new algorithm lets the solution keep up with its job until a better mechanism eventually presents itself	

3.3 Value Proposition

A value propositions serves an integral role in the creation and eventual sale of any new product and/or service. It essentially boils down to a referendum as to why a client should buy a product being offered by a organization, what benefits brings, what pains it relieves

and what can the client expect to obtain with the full package. As the Cambridge dictionary puts it "*a reason given by a seller for buying their particular product or service, based on the value it offers customers*" [63].

In order to organize and understand the underlying ideas of a value proposition, a step anyone can take before writing it out is to fill a value proposition canvas. As it is a simple way to understand your customers needs, and design products and services they want. In this case the filled value proposition canvas can be seen in Figure 15.

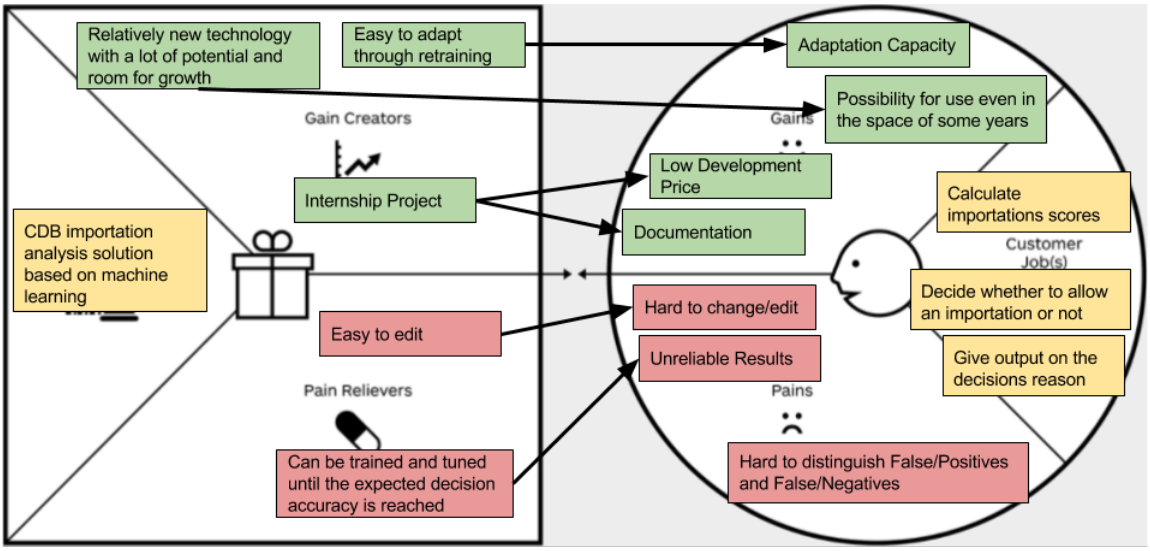


FIGURE 15: Value Proposition Canvas (adapted from [64])

By writing out the relations shown in the value proposition canvas (Figure 15) a fully fledged value proposition can be obtained. In this case, its is proposed the creation of a CDB importation analysis system based on machine learning, that would take on some of its customer's jobs by first calculating the overall score of an importation, then deciding whether or not it should pass and finally providing the reason to that decision. This solution would, simultaneously, provide benefits for the customer by being easily adaptable, being based on new and relevant technology, cheap, and having plenty of documentation, as well as relieving some of its pains due to allowing for easy edition, and attributing reliable results.

3.4 Business Model Canvas

The business model canvas was proposed by Alexander Osterwalder in 2010 in is book *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers* [65], in it the canvas model is presented as being a standardized format of describing well-known business model patterns so that they may be immediately useful for anyone that may need to use them for business model design.

A business model describes the rationale of how an organization creates, delivers and captures value and the concept of one must be simple, relevant, and intuitively understandable, while not oversimplifying the complexities of its functions.

A possible business model that could be applied to the present context is presented in Figure 16.

As one can observe, the business model canvas is composed of nine different areas. All are essential to the judging an organization or project in its entirety. The definition and explanation of those parts are as follows [65]:

- Key Partners - This section includes the various parties, outside the organization, that contribute to bring a project or product to fruition. In this case, as this Masters work is something done by the organization in order to guarantee is continuous operation there are no key partners involved;
- Key Activities - Describes what are the activities without which a project/product won't be able to be developed or succeed. For this case, such activities include the gathering, parsing and exploratory analysis of data related to importations made by E-goi's users, through which a valid classifier can be trained. Additionally it also includes the creation of an API for the developed classifiers once the base is created and tested;
- Key Resources - The gist of this section is that key resources allow a organization to create and offer a value proposition, reach markets, maintain relationships with Customer Segments and earn revenues. In the business model canvas (16), we can see that the most important resources for the current work are: the data related to importations made by E-goi's users (as it is the very base of the entire problem), the software and hardware necessary for development and possible hosting of the solution, and the platform through which users interact with the system;
- Value Proposition - The value proposition is constituted by the various characteristics that make a buyer opt for a solution in detriment of other alternatives. In this case, the qualities through which the solution can be differentiated are the fact that it efficiently and accurately judges user's CDB importations, while at the same time being easy to edit and adapt to business changes;
- Customer Relationship - It essentially exposes the way in which the organization deals with customers with differing characteristics. In the current Masters work, the main client of the developed solution is the organization, as such it is expected that the established relation should be a long term one, as the solution would answer to a core business necessity;
- Channels - The section describes the methods through which a product/service is provided to the end user. In this context this are a would include the API through which, in the future, the classifier will be made available for organization use;
- Customer Segments - This area designates any type of customer that given organization has the intention of reaching with its product/service. In the current scenario, the most specific segment targeted by the solution is the organization itself, however, it should also be noted that the users of E-goi's platforms and services would also benefit from it both directly (as there would be a better chance that a benign user can do his importation without worries) and indirectly (as it would allow for E-goi to avoid problems with ISPs regarding spam, which in turn means a greater up-time for its services);
- Cost Structure - Pertains to all expenses necessary to create, publicize, maintain and distribute a given product/service. As in this case the developed solution is internal to the organization the cost structure would be composed of expenses related to its development and maintenance, which means that the costs that compose this area of the canvas are the software and hardware necessary to solution development and maintenance, as well as the salary provided to the person working on those tasks;
- Revenue Streams - Present the profit related to actions regarding the product/service in question. The current scenario does not allow for the generation of direct profits, it

does, however, enable the organization to have a greater chance to avoid having any sort of trouble with ISPs regarding spam, which means there is a lower probability of their e-mails and domains getting blocked, allowing for a better service up-time.

3.5 Porter's Value Chain

In his 1995 work *Competitive Advantage* Porter [67] states "*Competitive advantage cannot be understood by looking at a firm as a whole. It stems from the many discrete activities a firm performs in designing, producing, marketing, delivering, and supporting its product.*". Later in the book, Porter introduced the value chain as a systematic way of examining all the activities an organization performs and how they interact. It does this by separating a organization into its core activities in order to understand both inherent costs as well as potential sources of differentiation [67].

Usually the model used to represent a organization's value chain is that which Porter used in his own first publication (Figure 17).

In this model value activities are divided into two types, primary activities and support activities. Primary activities are related to the physical production of a product and its sale to the buyer, as well as assistance. Support activities back the primary activities and each other by obtaining and providing resources, as well as other necessary functions[67].

In terms of primary activities Porter's proposed value chain is divided into five different parts [67]:

1. Inbound Logistics - Refer to activities related with receiving, storing, and propagating inputs to the product. When applied to this Master works context, one could insert activities like the collection and storage of the various data imputed by a user when using the various areas of the E-goi platform (contact lists, forms, etc.) in this area;

2. Operations - Frames activities related with converting inputs into final products (outputs). In this case, actions like the maintenance of a viable multi-channel marketing automation platform, as well as the creation of user statistics should be considered;

3. Outbound Logistics - Activities related with the collection, storage, and physical distribution of the product to consumers. Because, in the present context, the product is intangible the closest relation that could be drawn to Porter's original idea of outbound logistics would be the maintenance and upkeep of the E-goi platform, as well as its public API;

4. Marketing and Sales - Activities associated with preparing ways for customers to buy the product and also urging them to do so. Actions such as promotional videos announcing new features, tutorials on how to use them, extensive support, ease of use and heavy reliance on web marketing, as well as journalistic marketing (too a lesser extent) can be taken into account in this zone;

5. Service - Activities that translate to ways of maintaining and improving the product. In here, fit activities like providing a help-desk, managing the various IT systems used in the organization's activity and managing the infrastructures used to develop and maintain the organization's business.

In regards to secondary activities Porter's proposed value chain is divided into four different areas [67]:

The Business Model Canvas

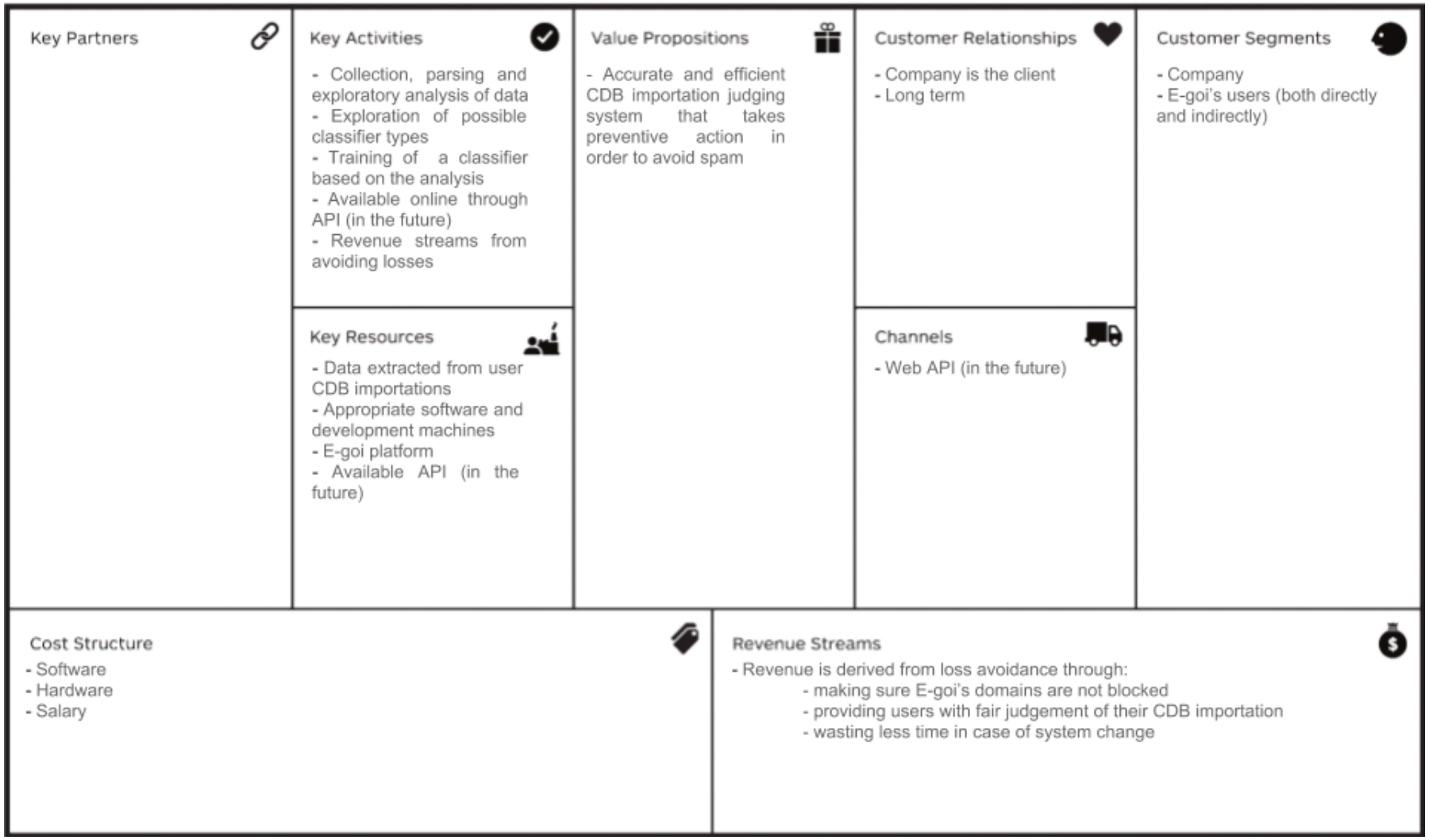


FIGURE 16: Business Model Canvas (adapted from [66])

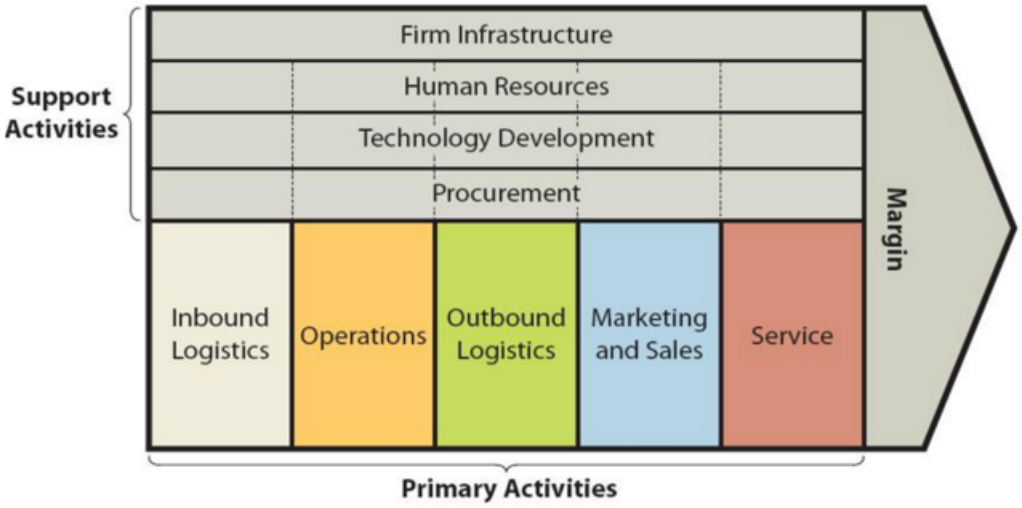


FIGURE 17: Porter's Value Chain Diagram [68]

1. Procurement - Refers to the act of acquiring inputs which contribute to the organization's value chain. Actions as the acquisition of adequate software, buying IP addresses, and acquiring servers should be taken into account from the present case;

2. Technology Development - Consists of the various activities that strive to make both the product and the process better than their current form. The experimentation of recent, efficient, technologies, as well as new methodologies, and consequent incorporation of those that bring good results to the business, are some actions the current organization performs which contribute to technology development;

3. Human Resource Management - Activities related to the recruitment, hiring, training, development and compensation of the different personnel present in a organization. Actions such as the realization of internships, the use of training programs and the chance for future reoccurring promotions are actions that can be included in this zone of the value chain;

4. Firm Infrastructures - Consists of a number of activities that both directly and indirectly contribute to the organization's infrastructure. Such activities include: general management; planning; finance; accounting; legal operations; government affairs; and quality management. In its structure, the organization contains, both an accounting as well as quality management departments. Additionally, there are also various people who are responsible for planning and dealing with legal affairs, respectively (although different types of affairs are dealt with by different individuals who specialize in them).

As it is possible to see from the many points above E-goi seems to have a value chain that, while being adapted due to the immaterial nature of the product, answers to the various areas of Porter's proposed value chain. The fact that the organization's structure fits with its proposed value chain improves its ability to both create and sustain competitive advantages, which is always a good indicator of a given organization's level of competitiveness [67].

3.6 Analytic Hierarchy Process (AHP)

The AHP method was first presented by Tomas L. Saaty [69] in 1980 in his article *The analytic hierarchy process: planning, priority setting, resources allocation*. It is one of the most widely known and used decision support methods and helps in the resolution of problems that are evaluated using multiple criteria.

The method stands on the Newtonian and Cartesian way of thinking. This means it gradually tries to decompose and segregate the problem into smaller ones such that these pieces can then be associated and synthesized [70].

It can be divided into three main steps: hierarchic division, priority definition and logic consistency.

3.6.1 Hierarchic Division

The problem is divided hierarchically in different levels (normally in objectives, evaluation criteria, and alternatives). This division of the various aspects that compose the overall problem allows an easier understanding and visualization of it [70].

When applied to the problem at hand this hierarchization would look something like what can be perceived in Figure 18. As one can see the main objective to be accomplished in the current scenario is to develop a better contact database importation solution than the one currently available.

This new solution should be evaluated according to: how easily it is maintained when compared to the current one, how well it judges a database while having into account its characteristics, which characteristics it uses to judge a database (the more relevant a characteristic is the better).

According to the aforementioned main objective and the three evaluation criteria, three alternatives exist that can be considered a possible solution, those are to use: an unsupervised machine learning approach; a supervised approach where exploratory data analysis is performed beforehand in order to find out relevant DB characteristics; or an approach where the main characteristics are found through an exploratory data analysis, and are then provided to the algorithm while also preserving some of the ones that do not appear so relevant to human eyes, but may be able to impact a decision if analyzed by a machine, combining in essence a supervised and an unsupervised approach.

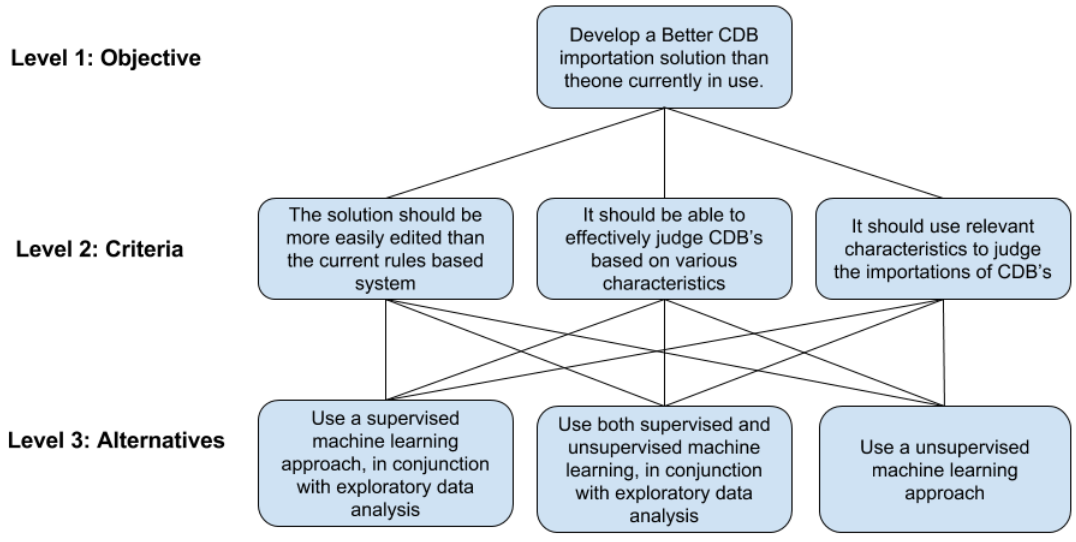


FIGURE 18: AHP Diagram

3.6.2 Priority definition

This phase seeks to establish priorities in both criteria and alternatives presented in the previous page. In order to do this one must first decide a priority between their different criterion and then establish a rank for each alternative in relation to each criterion. This allows one to realize which solution provides a better answer to a given criterion [70]. In order to compare each standard, one should first rank each of them against each other from one to nine (Figure 19). These results were obtained by computing the most common results of an inquiry with the same structure as the one provided in the image.

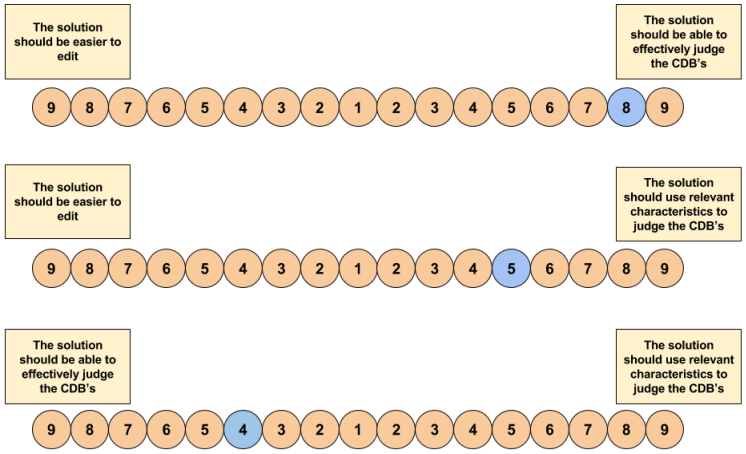


FIGURE 19: Criterion Comparison

After having done this, a comparison matrix should be built and used to discover the weight of each criterion (Table 10 and Table 11). It should be noted that in order to safely obtain the weight of a given criterion in relation to others one must first normalize the columns of the comparison matrix and then compute the average of the rows of the obtained result [71]. This process can be observed at Table 9. The result will be a value that is safe to directly compare and adequately represents the weight of an instance, this value can also be referred to as Eigen Value. In general, the bigger an instance's value is in relation to others the more importance that one carries.

TABLE 9: Weight Computation Process

	A	B	C	Weight (Eigen Value)
A	$aa/sumA$	$ab/sumB$	$ac/sumC$	$(aa/sumA + ab/sumB + ac/sumC)/NumberofColumns$
B	$ba/sumA$	$bb/sumB$	$bc/sumC$	$(ba/sumA + bb/sumB + bc/sumC)/NumberofColumns$
C	$ca/sumA$	$cb/sumB$	$cc/sumC$	$(ca/sumA + cb/sumB + cc/sumC)/NumberofColumns$
Column Sum	$sumA = aa + ba + ca$	$sumB = ab + bb + cb$	$sumC = ca + cb + cc$	

In the following tables each criterion is represented by a capital letter:

- A = The solution should be easier to edit
- B = The solution should be able to effectively judge the CDBs
- C = The solution should use relevant characteristics to judge the CDBs

TABLE 10: Criterion Comparison Matrix

	A	B	C
A	1	1/8	1/5
B	8	1	4
C	5	1/4	1

TABLE 11: Criterion Weight

Criterion	Weight(Eigenvalue)	Importance
A	0.06	Lowest
B	0.70	Greatest
C	0.24	Medium

Having obtained the relative weight of each criterion in relation to one another, it is now necessary to do the same procedure for each criterion regarding the alternatives available to answer it. That being the case, first, a priority as to be established between each alternative for every criterion (as seen in Figure 20, once again, these results were obtained by computing the most common results of an inquiry with the same structure as the one provided in the image).

Afterwards, a matrix is established containing each of the relative priorities for the various alternatives in each criterion and, from those matrices, the weight of each alternative is calculated for a given criterion (this process is presented in Table 12). It should be noted, that these weights are calculated in the same manner which was explained when computing the criteria weight above. The process can be observed at Table 9.

In the following tables each alternative is represented by a capital letter:

- X = Supervised machine learning and exploratory data analysis.
- Y = Unsupervised machine learning.
- Z = Both supervised and unsupervised machine learning and exploratory data analysis.

TABLE 12: Alternative Weight per Criterion

A				
	X	Y	Z	Weights (Eigenvalue)
X	1	1	1	0.33
Y	1	1	1	0.33
Z	1	1	1	0.33
B				
	X	Y	Z	Weights (Eigenvalue)
X	1	5	1/3	0.29
Y	1/5	1	1/6	0.08
Z	3	6	1	0.63
C				
	X	Y	Z	Weights (Eigenvalue)
X	1	4	1/3	0.28
Y	1/4	1	1/5	0.09
Z	3	5	1	0.63

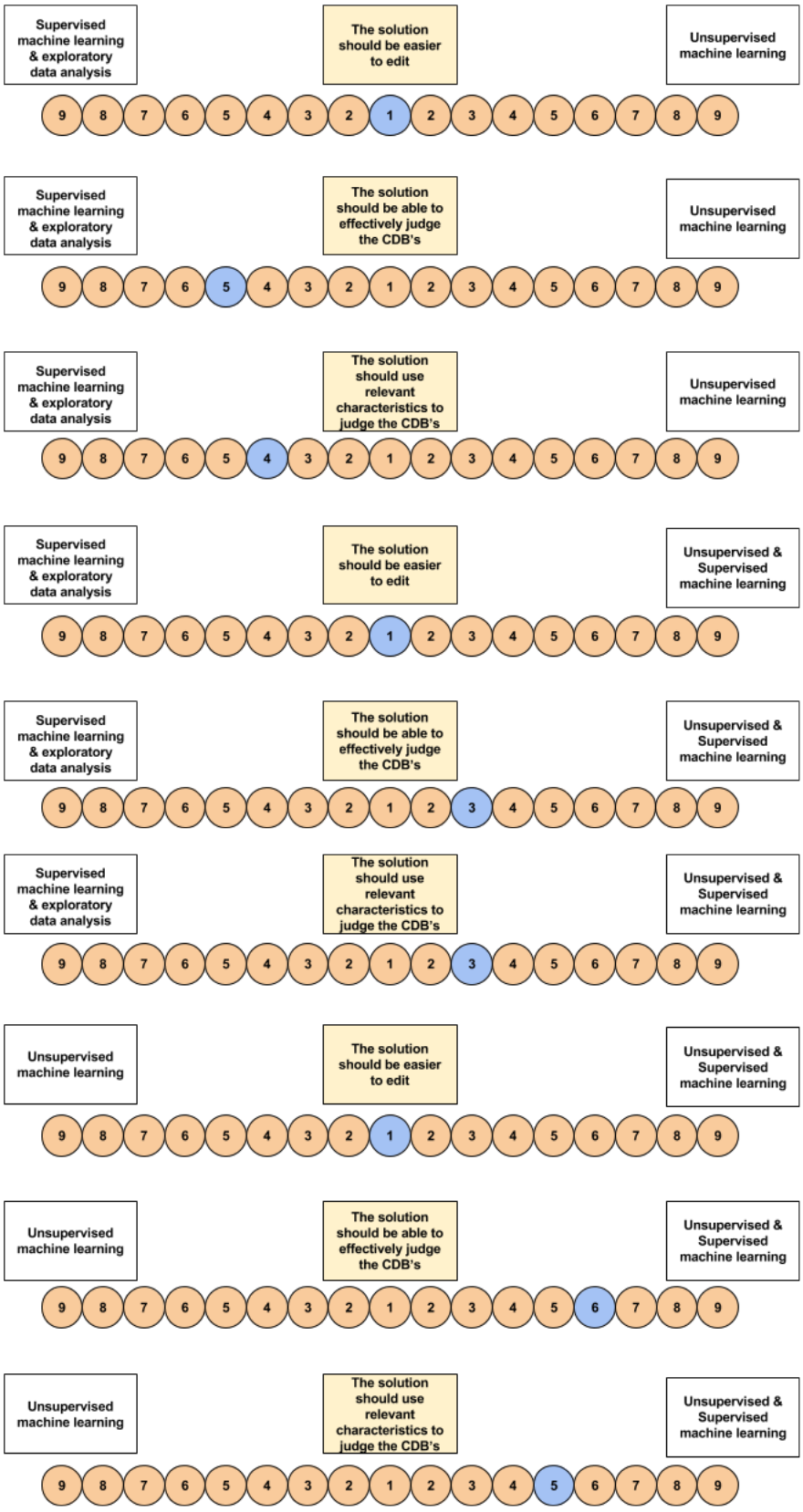


FIGURE 20: Alternative Comparison per Criterion

Since the priorities between alternatives for each criterion have been found. It becomes necessary to find the global priority between each alternative. To do this, it is necessary to compute the sum of the products of each alternatives' criterion weight (also known as local priorities) and that criterion's weight in relation to the other ones. In other word, if a given alternative U as a weight of 0.3, 0.5 and 0.1, for three criteria that have a relative weight of 0.1, 0.6 and 0.3, respectively, then that alternatives global weight (GW) can be found by:

$$GW_U = 0.3 \times 0.1 + 0.5 \times 0.6 + 0.1 \times 0.3 = 0.36$$

If this process is followed for each alternative, then Table 13 is obtained, allowing the perception of the global weight of each alternative in relation to the others.

TABLE 13: Total Weight of the Alternatives

	A		B		C		Global Weight
X	0.06×0.33	+	0.70×0.29	+	0.24×0.28	=	0.29
Y	0.06×0.33	+	0.70×0.08	+	0.24×0.09	=	0.10
Z	0.06×0.33	+	0.70×0.63	+	0.24×0.63	=	0.61

3.6.3 Logic Consistency

The concept of consistency derives from the fact that the numeric values being worked are derived from subjective values given based on an individual's preference, which makes the possibility of obtaining a perfect final matrix akin to zero.

As referred in [71], "*If you prefer an apple twice as much than a pear and a pear twice as much than an orange; how much would you prefer an apple with respect to an orange? The mathematically consistent answer is 4. Similarly, in a criteria comparison matrix, if we provide a value of 2 to the first criterion over the second and assign a value of 3 to the second criterion with respect to the third one, the value of preference of the first criterion with respect to the third one should be* $2 \times 3 = 6$. *However, if the decision-maker has assigned a value such as 4, 5, or 7, there would be a certain level of inconsistency in the matrix of judgments.*".

So as to take into account this inconsistency that results from the inclusion of human opinion in a mathematical method, it becomes necessary to find both the consistency index(CI) and the consistency ratio (CR) for the comparison matrices (judgment matrices) presented beforehand in both Figure 19 and 20. In order to apply any of these operations, one must first know how to compute the values of CI and CR. CI is calculated as

$$CI = (\lambda_{max} - n)/(n-1)$$

where n is the number of elements being compared (three in this case), and λ_{max} is a rating parameter calculated through the average of the ratio between the weighted sums of a given variable's values and its respective weight. When CI is divided by the consistency index of a random matrix denominated RI (whose tabled values can be checked at Table 14) the ratio obtained from that operation is CR [71], reiterating:

$$CR = CI/RI$$

TABLE 14: Random Consistency Index Values

Matrix Dimension	1	2	3	4	5	6	7	8	9	10
Random Consistency Index	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Usually a matrix is considered consistent if its CR is less than 0.1 [71].

In relation to λ_{max} its calculation requires one to know a comparison matrix's weights. After obtaining this prerequisite it is possible to compute its value by iterating through the following steps:

1. Suppose we have the following Table 15, and its relative weights.

TABLE 15: λ_{max} Calculation Process: Step 0

	D	E	F	Weight (Eigen Value)
D	dd	de	df	WD
E	ed	ee	ef	WE
F	fd	fe	ff	WF

2. The first step to take is multiply each column member by the weight of that variable, as seen in Table 16.

TABLE 16: λ_{max} Calculation Process: Step 1

	D	E	F
D	$wdd = dd \times WD$	$wde = de \times WE$	$wdf = df \times WF$
E	$wed = ed \times WD$	$wee = ee \times WE$	$wef = ef \times WF$
F	$wfd = fd \times WD$	$wfe = fe \times WE$	$wff = ff \times WF$

3. Next the values of each row should be summed in order to obtain a weighted sum, as seen in Table 17.

TABLE 17: λ_{max} Calculation Process: Step 2

	D	E	F	Weighted Sum
D	wdd	wde	wdf	$WSD = wdd + wde + wdf$
E	wed	wee	wef	$WSE = wed + wee + wef$
F	wfd	wfe	wff	$WSF = wfd + wfe + wff$

4. The third step is accomplished by dividing the weighted sum of the matrix rows by their respective weight, as seen in Table 18.

TABLE 18: λ_{max} Calculation Process: Step 3

Weighted Sum		Weight	Result
WSD	/	WD	$= RD$
WSE	/	WE	$= RE$
WSF	/	WF	$= RF$

5. The final step to take in order to obtain λ_{max} is to simply compute the average of the values obtained in the last step. Reiterating:

$$\lambda_{max} = \frac{RD + RE + RF}{Numberof Results}$$

Having applied each of these operations to the aforementioned judgment matrices, Table 19 was obtained. The fact that all consistency ratio values for every judgment matrix

are less than 0.1 serves as proof of those matrices consistency and, due to that, validates the various operations that took place with them.

TABLE 19: Consistency Index and Ratio per Criterion

	CI	CR
Criterion Judgment Matrix	0.047	0.081
A's Judgment Matrix	0	0
B's Judgment Matrix	0.047	0.081
C's Judgment Matrix	0.043	0.074

3.6.4 AHP Conclusion

After applying the three main steps of the AHP method and obtaining the global weights of each of the alternatives defined in its first step. It is now possible to establish a mathematically proven degree of relevance between them, which should be taken into account. In this case it was proven that the most relevant alternative would be Z followed by Y and finally X, since their global weight were 0.61, 0.29 and 0.10 respectively.

3.7 Summary

Through this chapter, the concept of innovation was introduced. It was then explained how innovation could be achieved, by answering a problem by means not yet implemented, which in turn correlated to the present work's context. After this, the process of idea generation was explained, having resulted in a possible solution to the work's problem. Various alternatives for that solution were then evaluated based on a set of criteria and a mathematical proven degree of relevance was established between them.

Chapter 4

Design

This chapter will present the various requirements inherent to this Masters work's realization. Additionally, it will present a possible structure for the solution to be developed, as well as explain choices made in each case. It will, then, present some information about the data which is under analysis, so that the reader may obtain a deeper insight of the metrics available. Finally, it will explain the use cases which would be expected to be made available from the API responsible for providing the developed system's capabilities to external users.

4.1 Requirement Gathering

An important aspect of any software development effort is understanding and managing requirements, which can then be used as a standing point for further design and implementation.

According to Loucopoulos & Karakostas [72], the term requirements engineering can be defined as "*the systematic process of developing requirements through an iterative co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats and checking the accuracy of the understanding gained*". This major process (Figure 21) can then be divided into three minor ones [72]:

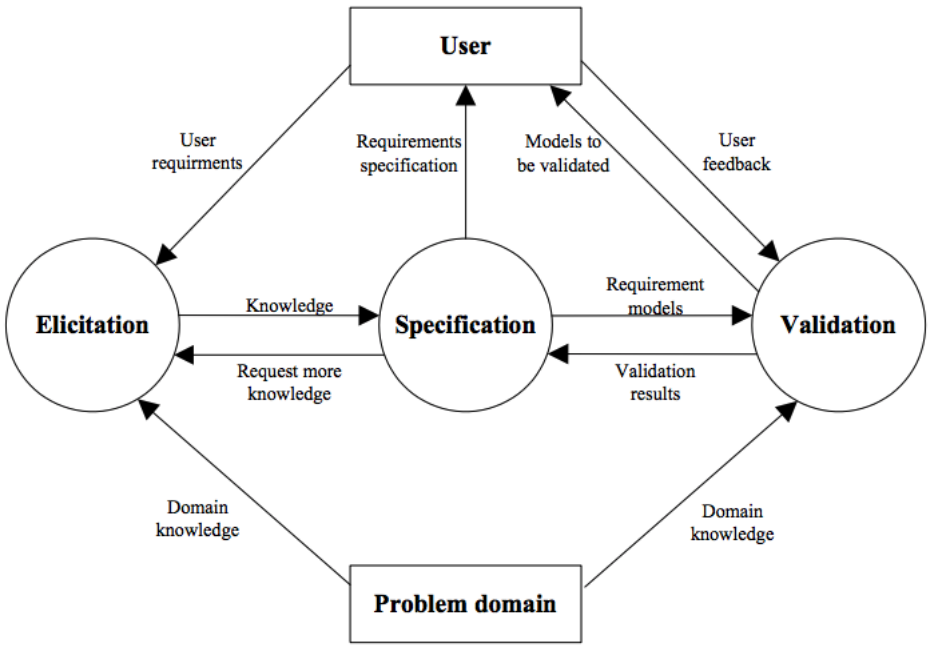


FIGURE 21: Processes which Constitute Requirement Engineering [72]

1. Elicitation - This phase is constituted by the attainment of knowledge regarding the various aspects of the problem, through different sources, with the intent to the produce a formal specification of the prerequisites needed to solve the problem. When solving a new problem the first step should always be to understand it;

2. Requirement Specification - In this phase formal specifications of the project's requirements are defined in order to allow these to be used in future development stages. These specifications are used as a bridge between users and software developer and normally do not describe through which methods a functionality should be achieved;

3. Requirement Validation - This phase can be perceived as an ongoing process that is triggered when new information is obtained and incorporated, it essentially works as a way to guarantee that no wrong information ends up making its way to the project's requirements. It is normally accomplished by having regular meetings both with the development team and with the project's client.

Having introduced the relevant requirement engineering concepts necessary for further comprehension of this document, now a deeper look at each of the separate minor processes will be presented.

4.1.1 Elicitation

This process was mostly accomplished through the everyday realizations that one seems to attain through working in a project, from both talking to other people present, that possess potentially relevant information, and from interacting with the problem's components and technologies.

Additionally, the copious amount of research needed to achieve a state of the art also ended up contributing for this process in various ways.

4.1.2 Requirement Specification

Requirements can generally be divided into two different categories:

- **Functional** - Pertaining to behaviours or functions a system and/or component must be able to perform. The following functional requirements were identified:
  - The system should be able to read an importation log file, or otherwise have access to its information in order to analyze it;
  - The system should be able to analyze an imported CDB basing this process on its judging machine learning algorithm;
  - The system should provide an option to train the machine learning algorithm which is used in its CDB analysis;
  - The system should be able to make its various functionalities available for outside use (likely through an API).

- **Non-functional** - Pertaining to the way in which a given system/components will perform a function or behaviour. Aspects like performance, or reliability, which are not as easily tested or evaluated. The following nonfunctional requirements were identified:
  - Analyze the current CDB importation solution's judging method;

- Perform state of the art research regarding the relevant topics encompassed by this problem;
- Perform research regarding the several machine learning algorithms that may be able to be applied to the current problem, in order to choose the ones most likely to be used as prototypes;
- Perform research regarding the tools available for both performing EDA and developing machine learning algorithms, and select appropriate options for each case;
- Perform an analysis in order to detect the various characteristics, specific to both the user's account and importation which should be taken into account in order to effectively judge the CDB;
- Create a test plan in order to validate the prototypes accuracy and effectiveness.

4.1.3 Validation

Regarding the validation process for the aforementioned requirements this was accomplished through a great number of meetings with client (a procedure which is made relatively simple since, in this case, the client is the organization itself).

4.2 Domain Modeling

In general terms a domain model equates to an appropriate representation of the various concepts inherent to a given problem, which include not only their characteristics/attributes, but also the relation they establish between each other. A domain model essentially boils down to "*a rigorously organized and selective abstraction of the knowledge in a domain expert's head*" [73].

In the present problem's case, due to its nature, the domain can be summarized to two essential entities:

1. Clients - Which are the several platform users that can perform CDB importations. These possess the following attributes (this is only a list of the relevant ones, as a complete one would be very extensive):

   - Client ID
   - Country
   - E-mail Forwarding Block Reason
   - Confidence
   - Subscriber Total
   - New Subscriber Total
   - Importation Number
   - Total of Imported Subscribers from Importations
   - Total of Updated Subscribers from Importations
   - Total of Ignored Subscribers from Importations
   - Total of Invalid Subscribers from Importations
   - Total of Removed Subscribers from Importations
   - Recency (account age in days)
   - Frequency (total amount of payments in the last year)
   - Monetary (Total paid amount)

2. Importations - Each of which is performed by a client and can end up being either successful or unsuccessful. These have the following attributes (only the relevant ones):

- Importation ID
- Importation Hash
- Server
- Date
- Hour
- Client ID
- Suspect E-mails Percentage
- Soft Spam E-mail Percentage
- Hard Spam E-mail Percentage
- Similar E-mail Percentage
- Nonsense E-mail Percentage
- Suspect E-mail Points
- Soft Spam E-mail Points
- Hard Spam E-mail Points
- Similar E-mail Points
- Nonsense E-mail Points
- Malicious E-mail Points Sum
- Pacgoi Points
- Risk Factor
- Database (DB) Size
- Minimum E-mail Username Length
- Maximum E-mail Username Length
- Mean E-mail Username Length
- Final Score
- Imported Subscribers
- Imported Subscribers Percentage
- Importation Outcome

It should be explained why these are the only two objects present in the domain model. This happens since they are the only complex objects that directly affect the current problem's context. While the presence of other domain concepts such as subscribers can be inferred from the two objects' attributes. These do not directly have any impact on the decision to import a database. In the subscriber's case the only reason why they are relevant is their quantity, as that could be used as an indirect way to judge a given CDB's level of danger.

4.3 Process Modeling

In an organization a business process can be defined as a group of activities that is executed in a specific order to accomplish a goal. As an organization grows so does the amount of processes which it may execute, and, due to that, a need to document those various processes comes forth. In order to document those processes, normally, a business process model is used. It consists of a group of activity's which present constraints to the execution of one another, i.e. their execution follows a specific order and depends of the outcomes of the various parts that comprise the process [74].

The process encompassed by this maters thesis is a simple one. It starts when a user places a CDB importation request. When this happens, that CDB is analyzed by E-goi's system in order to obtain its important metrics (e.g. percentage of suspicious e-mail, percentage of spam-trap e-mail, etc), which are then sent to the developed judging algorithm. Based on the various features (metrics) which were sent to it, the judging algorithm makes a decision to either accept or deny the importation. This decision is then transmitted back to E-goi's main system, which takes the appropriate action based on that decision, and then, informs the user. The business process model representing the aforementioned process can be observed in Figure 22.

4.4 Component Structure

The system being developed would, ideally, be constituted of four different components.

Firstly, a component responsible for the parsing of CDB importation log files would be necessary. This component (CDB importation log parser) would, evidently, interpret

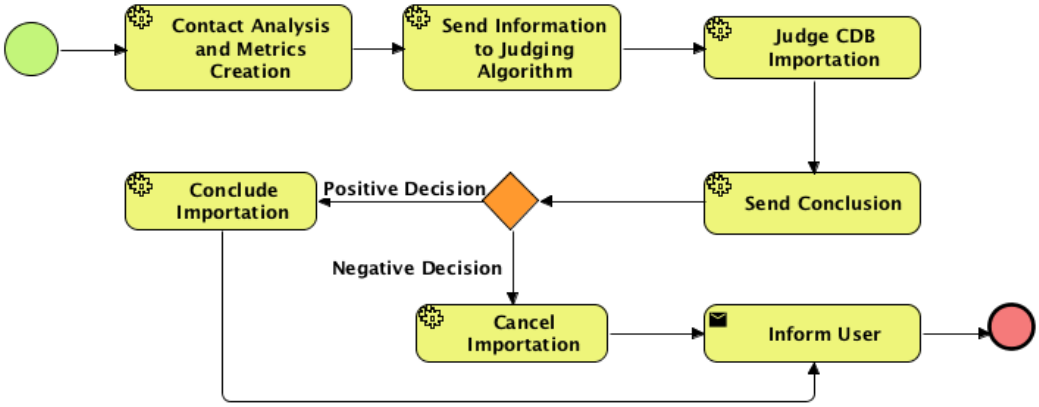


FIGURE 22: Proposed Process Diagram

the log file made available to him in order to obtain the relevant information present in it, which would then be stored internally so that another component could use it.

The second component would be a data-set creator. This component would make use of the information parsed by the previous component while supplementing it with additional user data from daily report files (which are generated by E-goi's system and are stored in csv format), in order to form complete data-sets with appropriate information and format to be used in the training of the developed judging algorithms/classifiers.

The third component, data-set pre-processor, would be employed in order to actively load and transform a created data-set so it could be used by the classifiers. For example, it would be useful in situations that require a categorical variable to be encoded or even standardized.

Lastly, the judging algorithm would be compromised of the machine-learning algorithm to be used in the decision regarding a database importation. When a request to judge a given CDB is made, this component would analyze its metrics in order to reach a conclusion of whether or not the importation is safe. A possible components diagram for the solution can be seen in Figure 23.

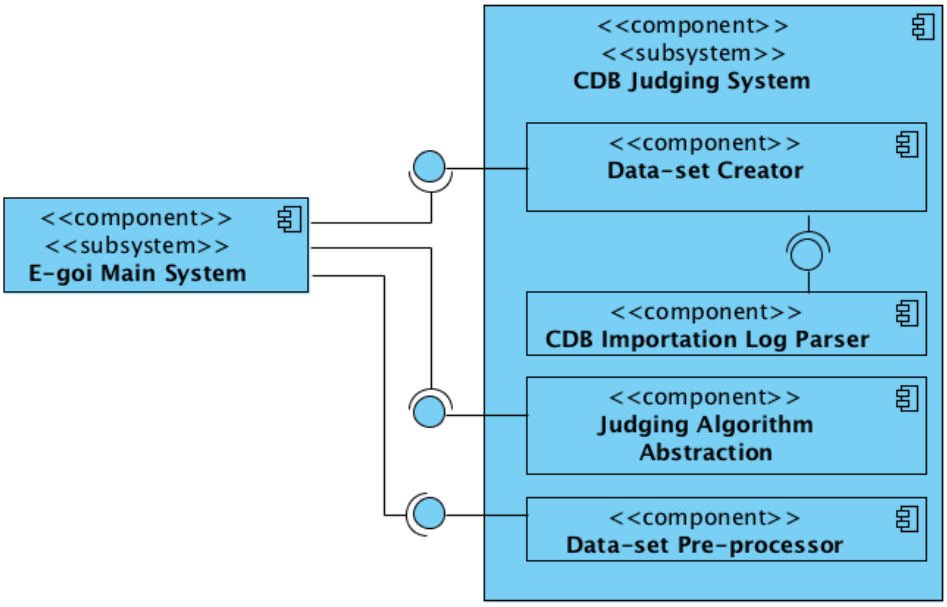


FIGURE 23: Proposed Component Diagram 1

This structure suffers, however, from some problems. By making each component responsible for making their capabilities available to outside systems, they are inadvertently taking on more responsibilities than they should. Another problem is that by making each component take on that responsibility, three different endpoints are created, when to an outsider's eyes the functionalities pertain to a single system, which does not make sense.

That being the case, a possible alternative to the structure presented beforehand could be accomplished by using one component responsible for making the systems capabilities available to the outside and orchestrating the multiple steps necessary to allow their proper operation. This component (judging system API) and the proposed alternative can be further observed in Figure 24.

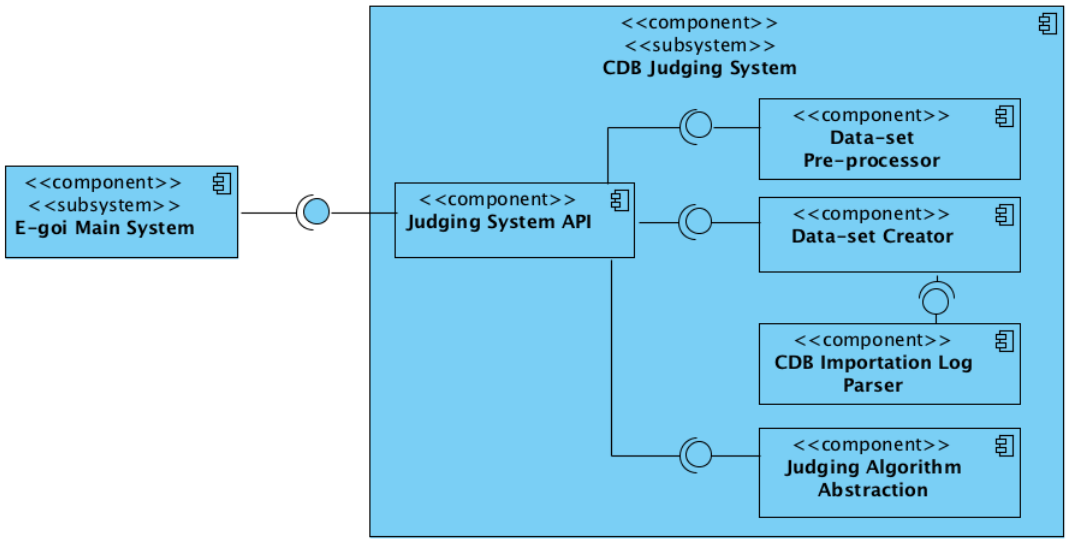


FIGURE 24: Proposed Component Diagram 2

The problem's solution will follow a component structure similar to the one observed in Figure 24. By opting for this structure instead of the first one presented (Figure 23) some benefits like the centralization of interaction logic in a component and the display of single endpoint to the outside world are obtained, which are good characteristics for the final solution to have.

4.5 Data Modeling

A data model can be thought as a type of abstraction that provides a conceptual representation of the data used, or otherwise involved, on a program's operation. It helps users understand the types of data involved in their program's storage by representing the various objects and relationships that compose it. By doing this, it transmits a very simple view of how types of data relate to one another, without delving into concepts relating to implementation [75].

In this case, data can be provided from two main sources:

1. Importation Logs - logs present in each of the production servers, which contain a registry of every CDB importation which occurs in that server;

2. Diary Reports - report files which contain information pertaining to the users which where active during that date.

Each of these sources provides data relative to a different domain concept. The log files contain data relative to importations while the diary reports have data regarding the

users which perform them. The relationship established between the two concepts, and the various attributes that constitute them, can be perceived in Figure 25. It should be noted that the presented attributes are only the main ones, as there are too many to clearly represent them in a diagram of sensible size.

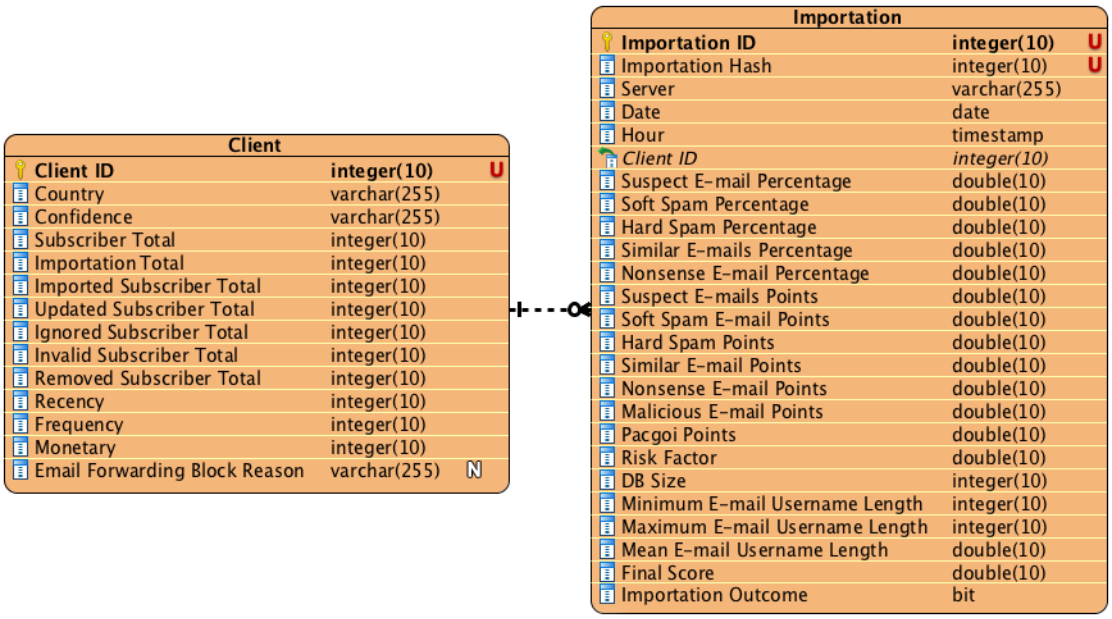


FIGURE 25: Entity Relationship Diagram

As is easily observable through the entity relationship diagram (Figure 25), a relationship of one-to-many exists between a user and his importations (i.e. a user can perform various importations). Additionally each user and importation possess a great array of attributes which characterize them. Some of them may not be easily understood by someone who is not fully aware of concepts pertaining to both the problem and organization's domain. As such an explanation will be provided for those specific cases:

- Client:
  - Confidence - represents the level of thrust that can be put into the user. It always start at level 0 and then the value changes according to the actions performed by the user;
  - Recency - the age of a given user account in days;
  - Frequency - is a set of levels which translate to the total number of payments made by a user during the last year;
  - Monetary - the total amount of money a user as paid.
- Importation:
  - **X** E-mail Percentage - the e-mail percentage of a given type (X). Obtained by computing the ratio of e-mails of that type, which were present in the CDB, by the DB size;
  - **X** Points - all points relating to e-mails of a type (X) are a score given by the current CDB judging system based on the amount of e-mail of a given type detected in the present importation. Pacgoi[1] points are a score attributed to an importation, based on a weighted sum used by the current judging system;

[1]Pacgoi is an internal system which does daily scans targeting the results obtained by user's campaigns. It then adjusts a user's Pacgoi score based on his campaigns results, either positively or negatively.

- Risk Factor - index computed as a result from the values of the recency from the user performing the importation, the DB size, and Pacgoi points;
- Final Score - final score computed by the current CDB decision system based on the sum of all e-mail related points of a given importation, and its risk factor;
- Y Username Length - the various statistical measures (Y) relating to the e-mail user-names present in the imported database (in the case of a DB containing two contacts abc@example.com and defg@sexample.com, the minimum will be 3, the maximum 4, and the mean 3.5).

Something that should be clarified is that, internally speaking, the concept of soft-spam and hard-spam do not directly co-relate to that of soft and hard bounces. Soft-spam e-mails are e-mails that are most commonly targeted by spam, the best example of soft spam e-mails are role e-mail addresses [2]. Hard-spam e-mail are addresses to where, no matter the case, messages should not be sent, these include e-mails that break their domain's rules (for example, aaa@gmail.com, as gmail requires a user-name of at least 6 characters), or known spam traps.

4.6 API Design

The developed system's API will be a core component to make its functionalities available to the external environment. There are three main system functions which the API should be able to provide. These translate to the following use cases:

1. The user should be able to create training data-sets easily through the API, which should use information present in both importation logs and report file;
2. The user should be able to train any of the available algorithms using the most recently created training data-set;
3. The user should be able to use the API in order to predict the outcomes of CDB importations given they select one of the available classifiers.

Bellow, the proposed sequence diagrams employed to exemplify a successful run of these three UCs (figures 26, 27 and 28, respectively) are presented.

As is possible to observe, for UC1, the user starts by sending a POST to an API resource (will be defined in a later section) containing information relevant to the creation of a new data-set, in this case, starting date and ending date. After the API verifies both the request's body structure as well as its content, it accepts the request and immediately returns a 202 - Accepted status code to the User. This happens since the creation process is slow, which means an immediate answer cannot be provided, translating into an asynchronous flow. After an accepted status is sent, the API creates an instance of data-set creator (DC), which instantiates a CDB importation log parser (LP). The API then uses DC to initiate the new data-set creation process. Firstly, LP is used to parse the various CDB importation logs in order to retrieve an object containing all relevant information present in the logs from start to end date (LO). Then, it sorts through the available report files and retrieves an object containing the data pertaining to users which were active in that date range (RFD). Once those two data objects are created it cross-references the information present in the log files with the one in the report files, complements it and organizes it in a matrix style format (DS). Finally, it saves that formed matrix as the new data-set, transmits

[2] E-mail addresses that are not associated with a particular person, but rather with a organization, department, position or group of recipients (at http://kb.cakemail.com/en/admin_deliverability/what.is.a.role.account/)

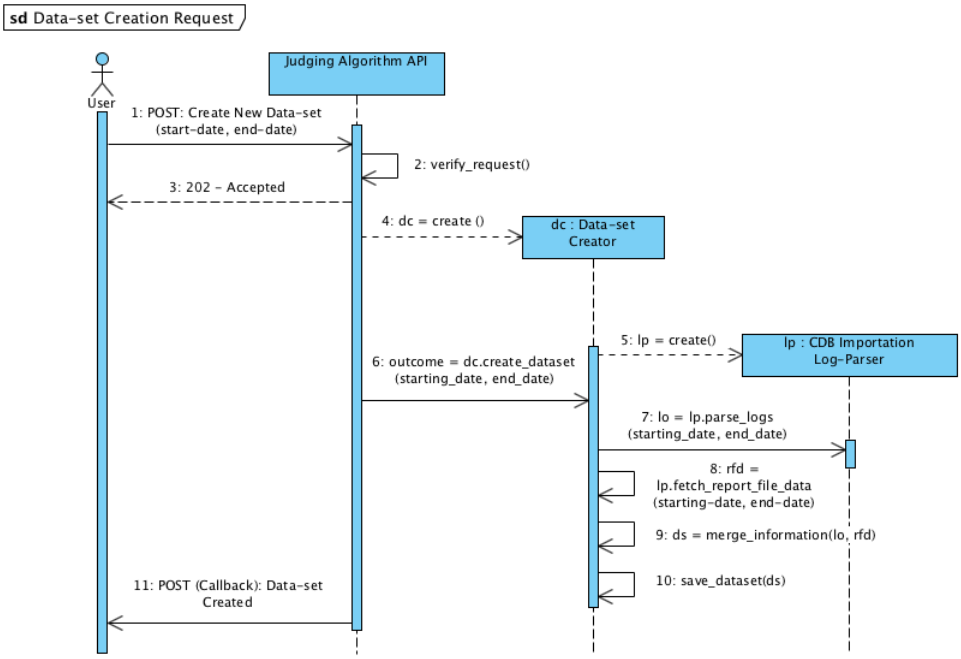


FIGURE 26: Proposed Sequence Diagram for UC1

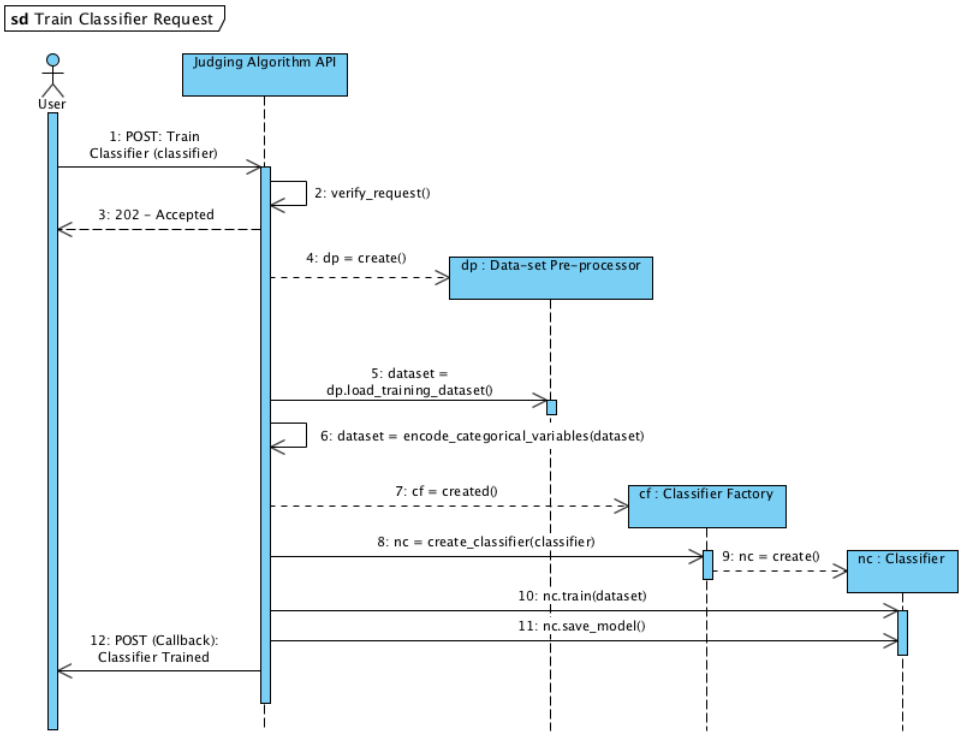


FIGURE 27: Proposed Sequence Diagram for UC2

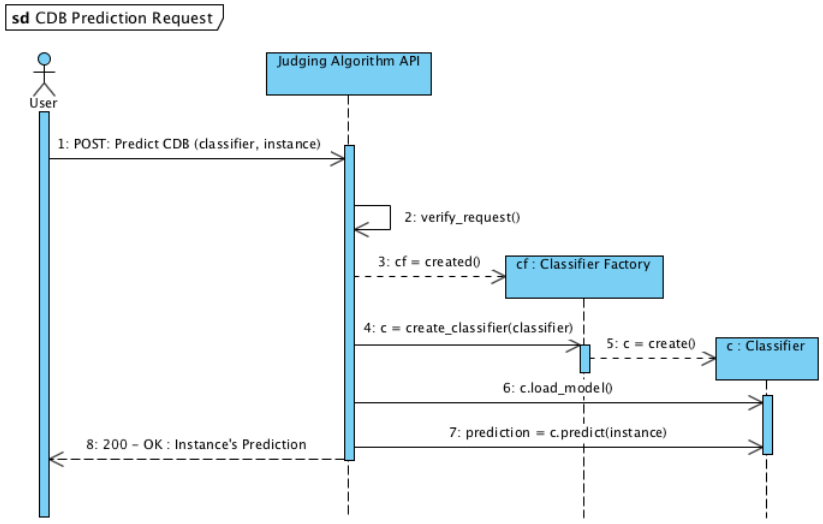


FIGURE 28: Proposed Sequence Diagram for UC3

the process outcome to the API which transmits it to the User by performing a POST to a callback passed in the original data-set creation request.

UC2 starts when a specific API resource receives POST request from th User targeted at specific classifier. The API verifies the request to check if it is appropriate and, due to the same reasons specified in UC1 immediately returns a 202-Accepted status code (if the verification is not concluded successfully an appropriate status code is sent instead). It then creates an instance of data-set pre-processor (DP) which it uses to load the current training data-set. Some categorical features of this data-set are then encoded so they may be used by the classifier. Afterwards, the API initializes a classifier factory (CF) and uses it to create and return a proper newly created classifier object (NC) basing the decision on the value passed by the User upon the initial request. The formed classifier is then trained using the loaded data-set and its model is saved so it may be used in the future. Upon the conclusion of this operation the API transmits its result to the User by performing a POST to a callback passed in the original data-set training request.

Finally, UC3 starts, similarly to the other cases, with the User performing a POST to a specific API resource. This request is verified both in terms of structure and content, however, since the prediction process is relatively fast, a 202-Accepted status code is not immediately returned to the user, i.e. this flow is synchronous. After the verification, the API instantiates a classifier factory (CF), which it uses to create an appropriate classifier object (C) by passing it the classifier present in the original User request. Once a proper classifier is formed, the most recently saved model for that algorithm type is loaded. Then the CDB importation instance passed by the User is fed to the classifier, which predicts its most likely outcome. That prediction is received by the API, which transmits it and a 200-OK status code back to the User.

4.7 Summary

In this chapter, requirement gathering was performed in order to obtain both functional and non-functional requirements necessary to the solution of this work's problem. After this step, the core concepts to the problem's domain were presented and briefly described. While taking the two previous points into consideration, some design hypotheses were given for both the model followed by the business process which will use the final solution, and the structure which will be followed by it in regards to its components. Afterwards,

the data which is used by the solution, as well as its origin, was exposed, having also explained some specific attributes which might have been harder to understand. Finally, the use cases expected from the API responsible for providing the system's capabilities to external users were planned and the rationale behind their sequence order was explained.

Chapter 5

Development

In this chapter, the development process for the proposed solution will be presented and described. In order to do that, the various phases necessary to that process will be discussed in detail, while also being presented in a orderly and logical fashion so as to help the reader understand it globally.

5.1 Feature Acquisition & Selection

To train the classifiers which are responsible for deciding if an CDB importation is valid or not, it is important to take into account certain features that are more relevant to that decision. The first step taken to obtain those features was to collect all the features to which access was permissible. This proved to be quite a lengthy process as a direct database access was not available, which meant the ways in which information could be obtained were roundabout. Two information fountains were used:

1. Importation log files (to obtain information pertinent to a importation)
2. Administrative report files (to obtain info about the users)

While the report files were generated in *csv* format, the same cannot be said about the log files. Hence, the creation of a log parser became a necessity. After the parser was completed the number of different features available for use was around two hundred and thirty.

Many of those features could be considered to be "noise", i.e. while they were clearly related to the business, they did not correlate (at the very least not directly) to the outcome of an user's importation. However, making a decision on that assumption alone would be a hasty and quite tick-headed. As such, after reuniting with the deliverability specialists at E-goi and discussing the issue, the two-hundred thirty fields were reduced to about twenty features that either affected the old judgment system or whose statistics were dependent on a importation's result.

These features were then divided based on their correlation coefficient computed by performing sensitivity analysis targeting them (for a more detailed explanation refer to the Computational Study chapter, subsections Sensitivity Analysis Tests and Sensitivity). Since a correlation coefficient of $[0.2 - 0.3]$ is normally considered the threshold between a negligible and weak relationship (as seen in Table 1 of [76]), the features with a correlation coefficient greater than 0.2 were used to train the prototype classifiers, while the ones with a lower correlation coefficient were discarded.

5.2 Data-set Pre-processing

After finding appropriate features for the solution, the next step taken was to create a class that could be used in order to make the necessary changes to a data-set containing the

relevant features in order to use it to train a classifier. This class ended up supporting various capabilities relevant to the pre-processing of the training data-set.

First, it supports extraction of relevant features from a passed data-set which contained more features that, while important, should not be used in the training process (such as, an importation's ID).

Secondly, it supports data-set balancing either through under or over-sampling techniques (i.e. techniques that try to balance an unbalanced model by either reducing the amount of instances in the majority class, or by increasing the amount of instances in the minority class). Something extremely useful, since the training data-sets for this problem tend to be unbalanced. This occurs due to a low occurrence rate of blocked importations, and can lead to negative effects in the classifiers' training if not properly taken care of. Implemented techniques include random under-sampling, near-miss under-sampling and synthetic minority over-sampling.

Thirdly, it has the capability to encode any categorical feature in order to transform it in a numerical one, it does this by creating a sub-feature for each value that the categorical feature can take and then marking them with a one on the sub-feature corresponding to an instance's value while leaving the other ones at zero.

Fourthly, it allows for the standardization of features, i.e. it allows for the normalization of the scale in numerical features. It does this by indicating that the biggest value detected for a feature is one, while the lowest detected value is zero, and all the other values follow the proportion becoming a value between zero and one.

While there exists a last capability associated with this class it is best presented at a later point, therefore it will be omitted, for now.

5.3 Classifier Implementation

Classification is one of the main focuses of this work. Before hand it had been mentioned that an implementation for three different algorithm types would be created (BAG-DT, SVM, NB). However, four classifiers were implemented:

1. A Naïve Bayes Classifier (NBC) (following a Gaussian NB approach)
2. A Support Vector Classifier (SVC)
3. A Random Forest Classifier (RFC) (which constitutes a specific case of BAG-DTs)
4. An AdaBoost Classifier (ABC)

All of these algorithms are employed for the same purpose of solving the binary classification problem at hand. That is, to predict if a CDB importation should, or not, occur. That being the case, it is only natural that they may be interchangeable between each other so as to allow for a greater system usability as well as an easier expansion, if it becomes a necessity. In essence, the point to make is that these implementations should be modular, because that will allow an easy edition of the prediction system as well as an easier time in case a new approach needs to be implemented.

So as to accomplish that modularization, a classifier abstraction was created which was inherited by all classifiers implemented. This abstraction defined a set of not implemented methods which must be defined in any class that inherits it. The main ones include:

- Train Classifier
- Predict Instance

- Predict Multiple Instances
- Estimate Accuracy through Cross-validation
- Compute Precision, Recall, F-score and Support
- Persist Model
- Load Model

All of these methods are self-explanatory, however, a short-explanation regarding how the data-set unbalance problem was dealt with in the training process should be provided.

As it has been aforementioned, the pre-processing class implements methods capable of under-sampling majority classes, and over-sampling the minority. These, however, each have their downfall.

In the case of under-sampling techniques, while they are an effective counter to the label unbalance present in the training data-set, the truth is their use effectively cuts a major part of all data present in the majority class, which is real. To put it into numbers let us say we have 25000 instances in a data-set, a not far off from reality prediction for the amount of blocked importations in those instances would be about 500. In order for the training unbalance problem to be solved through under-sampling, both the number of instances of both the majority and minority class should be equal, however that would mean that 24000 instances that could provide information about real cases would be discarded.

For over-sampling techniques the problem is somehow reversed. While no information is discarded, a huge amount of information regarding the minority class as to be inferred from a low number of occurrence. Taking the example given above, 24000 minority class instances would have to be extrapolated from 500 actual ones.

The solution adopted, after some research, in order to solve this conundrum was to use a capability provided by scikit-learn itself called class weights. This solution gives a fine balance between the advantages of both under and over-sampling since it does not generate new instances, but it also does not throw out existing information.

The way it works is the following: if, when training a classifier, the attribute class weight is set to "balanced", then the weight to set for each of the different classes is computed by the formula [77, 78]:

$$Ratio = \frac{TotalNumberofSamples}{NumberofClasses \times NumberofClassSamples}$$

This means that while the minority class will get a boost for each of its instances, the majority class' weight multiplier will decrease. This happens in such a way that a balance is struck by applying both of the calculated ratios during the training process.

Take, for example, the 25000 instances that were referred before, where only 500 were part of the minority class, that means the majority class was composed of 24500 instances. By computing the weight ratios of each class using the formula presented before-hand we get:

$$MinorityRatio = \frac{25000}{2 \times 500} = 25$$

$$MajorityRatio = \frac{25000}{2 \times 24500} \approx 0.5102$$

If we now compute the number of apparent instances for each class while taking into account their weight ratios, we see:

$$MinorityInstances = 500 \times 25 = 12500$$

$$MajorityInstances = 24500 \times 0.5102 \approx 12500$$

However, even this solution is not perfect as it is prone to noise in certain cases, e.g. when there are several outliers in the minority class since their weight also gets boosted.

There is no perfect solution for this dilemma. Nevertheless, as more data is acquired, it is expected that, whatever noise may be generated in the balancing, will be practically null when faced with the raw quantity of other correct class instances.

5.4 Classifier Tuning

The next big step of the solution's development was the fine tuning of the various algorithms and high-level comparison between them. Yet, before we can advance, some important concepts should be introduced.

First and foremost, the concept of confusion matrix. A confusion matrix is a way to summarize the classification performance of a given classifier when fed a specific set of data. Normally, it is a two dimensional matrix were one of the axis presents the true class of an instance/sample and the other shows the class assigned by the classifier to those instances/samples [79]. Due to the problem at hand being one related to binary classification, any confusion matrix originated from it will follow a two by two structure similar to the one seen at Table 20.

TABLE 20: Confusion Matrix Basic Structure

		Predicted Class	
		Positive Class	**Negative Class**
True Class	**Positive Class**	True Positive (TP)	False Negative (FN)
	Negative Class	False Positive (FP)	True Negative (TN)

Now that the concept of confusion matrix is clearly shown through Table 20, it is much easier to clarify other evaluation metrics, such as:

- Precision $= \dfrac{TP}{TP + FP}$
- Recall $= \dfrac{TP}{TP + FN}$

As was previously stated during the Literature Review, precision and recall are two evaluation metrics which are normally used as a pair. Precision can be defined as the ratio of truly positive cases in the pool of all cases which were predicted to be positive. Recall, on the other hand, can be defined has the ratio of truly positive cases in the pool of all actually positive cases. While precision expresses the proportion of actually relevant data points in the group the data points which the model says were relevant, recall expresses the ability to find all relevant instances in a data-set.

Therefore, the next step to take is to identify if both of these metrics have the same relevance in the current problem's context, and, if that is not the case, which of these metrics is more important.

From all metrics being measured, it is fairly obvious that the recall of blocked importations should be considered the most important, up to a reasonable threshold. This has to do with the amount of risk involved in both types of misclassification. **For clarity's sake let us say, from now on, that in the case of confusion matrices applied to this problem, the positive class is equivalent to the blocked/malicious importations while the negative class represents permissible ones**. If a false positive occur, i.e. if a seemingly correct

not malicious importation is predicted to be part of the positive class, then while damage does occur in at a client/user level, that damage is easily fixed and does not spread to other users. On the other hand if false negatives occur there is a chance ISPs may take action which creates a not easily fixable and very widespread problem (system level).

While taking this into account the implemented classifiers were a target of hyper-parameter[1] tuning, in order to maximize the efficiency for the positive class' recall. To do this, the use of Nested Cross-validation (NCV) was employed. A "regular" cross-validation approach, in this case, would use the same data to tune the hyper-parameters and evaluate its performance, which could could cause over-fitting [2]. In order to avoid this problem, NCV uses a series of train/validation/test set splits, similarly to what can be seen in Figure 29. For more information refer to [80].

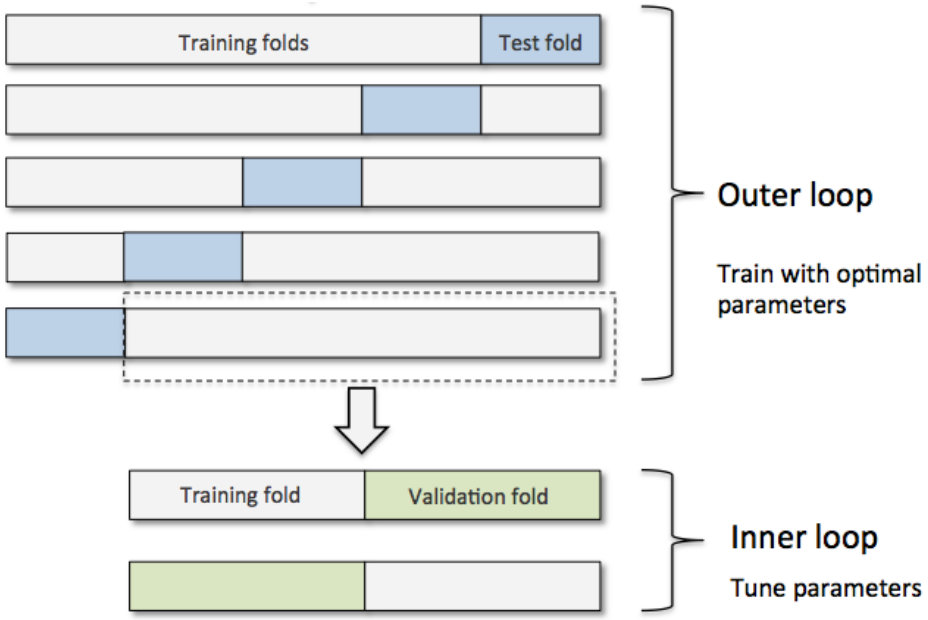


FIGURE 29: Nested Cross-validation Approach [81]

By evaluating different hyper-parameter combinations for each of the four classifiers, these were turned in the way which most benefited the recall of the positive class.

Once the tuning process was concluded, a high-level comparison between the four algorithms performance took place. To do this a regular fifteen fold cross-validation strategy was used in order to compute the sum of the various confusion matrices generated throughout the process, which is then utilized to compute evaluation metrics. The various summed confusion matrices as well as classification reports can be seen bellow (Tables 21, 22, 23, 24, 25, 26, 27, 28).

TABLE 21: Naïve Bayes Classifier's Confusion Matrix

		True Class	
		Positive Class	**Negative Class**
Predicted Class	**Positive Class**	264	345
	Negative Class	749	23395

[1] A parameter whose value is set before the training process begins and whose value affects the outcome of the trained algorithm.

[2] Refers to the case where a model is so adapted to its training data that it negatively impacts the performance of the model on any new data.

TABLE 22: Naïve Bayes Classifier's Classification Report

	Precision	**Recall**	**Support (Total Instances)**
Positive Class	0.261	0.433	609
Negative Class	0.985	0.969	24144
Avg / Total	0.968	0.956	24753

TABLE 23: Support Vector Classifier's Confusion Matrix

		True Class	
		Positive Class	**Negative Class**
Predicted Class	**Positive Class**	442	167
	Negative Class	161	23983

TABLE 24: Support Vector Classifier's Classification Report

	Precision	**Recall**	**Support (Total Instances)**
Positive Class	0.733	0.726	609
Negative Class	0.993	0.993	24144
Avg / Total	0.987	0.987	24753

TABLE 25: Random Forest Classifier's Confusion Matrix

		True Class	
		Positive Class	**Negative Class**
Predicted Class	**Positive Class**	499	110
	Negative Class	60	24084

TABLE 26: Random Forest Classifier's Classification Report

	Precision	**Recall**	**Support (Total Instances)**
Positive Class	0.893	0.819	609
Negative Class	0.995	0.998	24144
Avg / Total	0.993	0.993	24753

TABLE 27: AdaBoost Classifier's Confusion Matrix

		True Class	
		Positive Class	**Negative Class**
Predicted Class	**Positive Class**	514	95
	Negative Class	79	24065

TABLE 28: AdaBoost Classifier's Classification Report

	Precision	**Recall**	**Support (Total Instances)**
Positive Class	0.867	0.844	609
Negative Class	0.996	0.997	24144
Avg / Total	0.993	0.993	24753

As it is clear to analyze through the previous tables, there is, in most cases, a significant difference in performance for both precision and recall. The only exception to this case is the different present between the RFC ans ABC where a bit of the positive class' precision (2.6% reduction) is traded for more recall (2.5% increase). Due to the previously

discussed reasons, as well as the several confusion matrices and classification reports, we can roughly attribute the following efficiency order to the four classifiers:

$$ABC \geq RFC > SVC > NBC$$

However, this rough ordering does not take training time into account, which, as we will see in Computational Study chapter, is a major evaluation factor for a certain classifier.

5.5 API Implementation

Once the algorithms were tuned and ready to be used, a simple API that could make the project's functions to the outside environment was developed. This API followed the Representational State Transfer (REST) architectural style[3] and was developed using the Falcon web API framework [4].

The developed API is distributed through four resources:

1. **Classifier/Judging Algorithm Resource -** This is the simplest resource available. Both in terms of implementation and comprehension. It only provides a GET method which returns the list of classifiers made available through the API. In this case: Naïve Bayes, Support Vector, Random Forest and AdaBoost classifier;

2. **Training Resource -** This resource is pretty self-explanatory. It is used in any case where the API's user wants to train or re-train one of the available algorithms used to make predictions about CDB importations. In this case two different methods are made available a GET and a POST. The POST is the main method used to trigger a training process for a given classifier and allows for the user to send a callback link to where, upon conclusion, an appropriate message is sent. If a callback is not sent then the conclusion message for the training process can be returned via the GET, meaning this method could be used to employ a polling approach. In order to allow the API to continue listening for new requests even during the training of a classifier, this operation occurs in a new process separate from the main one;

3. **Predict Resource -** This resource allows for the user to make prediction about a passed CDB instance, while using a given classifier. It does this trough a simple POST method in which only the CDB instance's data should be passed, in an array format. It returns both the simple conclusion of whether or not the CDB importation is valid and the certainty percentage for each class the classifier can judge;

4. **Data-set Resource -** This resource allows for the expansion or creation of the data-set used to train the classifiers. Similarly to the training resource, the POST allows for a callback to be sent in order to receive a conclusion message, and, if it is not sent, that message can be retrieved through the GET method. The expansion and creation operations are also taken care of in a separate process, allowing for the continuous work of the API. The distinction between expanding the current data-set or creating a new one is indicated through the presence of certain parameters passed in the body when a POST occurs. In any case, a "end-date" needs to be passed, since that is the date up to which the CDB instance's, present in the data-set used for classifier training, will reach. The process is distinguished by the presence of a "start-date", since if this one is not present, the API will assume that the user just wants to expand the current data-set up to an "end-date". This means that if both fields are passed a

[3]Refer to: https://bit.ly/2lEHcYm
[4]Refer to: https://bit.ly/2KuZnxL

new data-set composed of CDB instances which occurred in that date range will be created.

For additional information regarding the developed API, the documentation generated trough its specification can be checked at Appendix A.

5.6 Pruning

Due to the huge amount of CDB importations that that take place at the E-goi platform (about 150 instances per day, in 2018), if the data-set is expanded without taking any other action (in order to better train the classifiers) it may result in such a big data-set that it becomes unsustainable both in terms of file size, as well as training time.

Therefore, it becomes necessary to somehow address this problem. This is done by the using a under-sampling approach called NearMiss, specifically, the 3rd variant of this approach denominated NearMiss3, which takes out a given number of the closest majority class samples for each minority class sample (further information can be consulted at [82]). The NearMiss3 implementation used in this case is a two step algorithm: first, for each minority class instance, their m nearest-neighbors will be kept; then, the majority instances whose average distance to the k nearest neighbors is the largest are selected. This way, a good distribution of points close and further away to the decision boundary is kept, which should help keep the model trained from the under-sampled data-set stable in terms of results. An example of this approach being used can be seen in Figure 30, where the purple dots represent the minority class, and the group composed of both the green and yellow dots is the majority class.

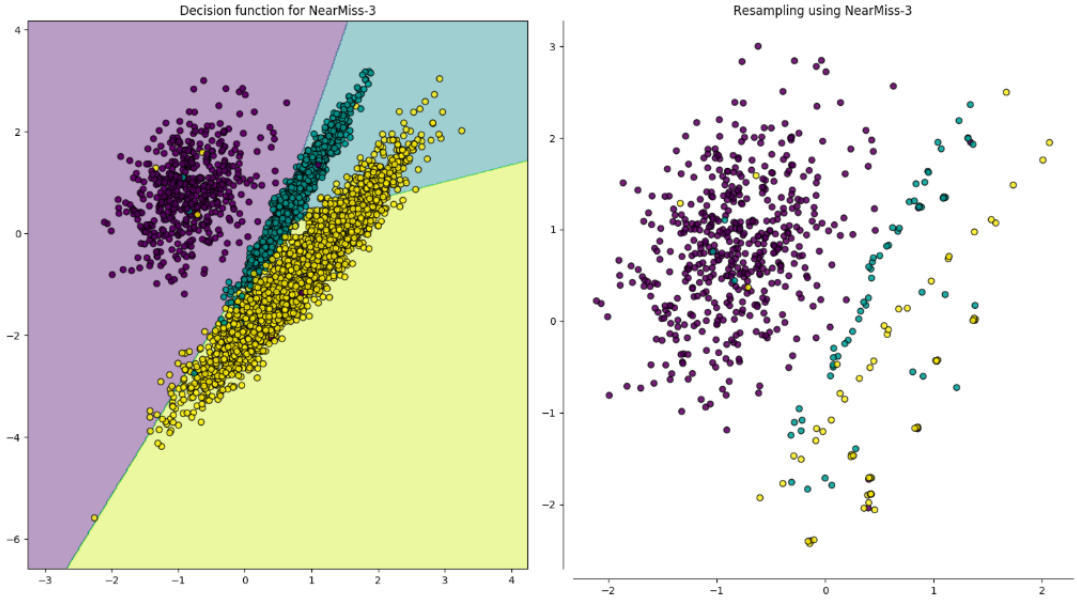


FIGURE 30: Example of NearMiss3 Application (Adapted from [83])

5.7 Unit Testing

Throughout the entire development process various unit tests were create so as to confirm that no bugs existed in the developed solution. In this section the various tests created for

each of the developed classes will be listed, as well as the outcome type (success or failure) obtained for each of them in a recent run.

- Data-set Pre-processing:
  - Load data-set ..... Success
  - Encode categorical feature ..... Success
  - Decode categorical feature ..... Success
  - Standardize features ..... Success
  - Random Under-sample ..... Success
  - NearMiss Under-sample ..... Success
  - SMOTE Over-sample ..... Success
  - Create train/test split ..... Success
- Naïve Bayes Classifier:
  - Train classifier ..... Success
  - Predict instance ..... Success
  - Cross-validate classifier ..... Success
  - Persist and load model ..... Success
- Support Vector Classifier:
  - Train classifier ..... Success
  - Predict instance ..... Success
  - Cross-validate classifier ..... Success
  - Persist and load model ..... Success
- Random Forest Classifier:
  - Train classifier ..... Success
  - Predict instance ..... Success
  - Cross-validate classifier ..... Success
  - Persist and load model ..... Success
- AdaBoost Classifier:
  - Train classifier ..... Success
  - Predict instance ..... Success
  - Cross-validate classifier ..... Success
  - Persist and load model ..... Success
- API:
  - Get classifier list ..... Success
  - Train classifiers (via callback) ..... Success
  - Train classifiers (via polling) ..... Success
  - Predict CDB instance (all classifiers) ..... Success
  - Expand data-set (via callback) ..... Success
  - Expand data-set (via polling) ..... Success
  - Create data-set (via callback) ..... Success
  - Create data-set (via polling) ..... Success

5.8 Summary

Throughout this chapter the various phases which composed the development of the proposed solution for the problem at hand were presented. First, the steps taken in order to obtain useful features to train the prototype classifiers were explained, and the process through which relevant features were chosen was exposed. Afterwards, different classes relating to data-set processing and the various classifier prototypes were explained, along with their capabilities and purpose. Once, the base structure was in place, the process through which tuning was performed in the various developed classifiers was clarified, which led to the conclusion of the prototypes' development. Next, the composition of the proposed system's API was described, as well as its functionalities. Finally, a listing of the several unit tests performed across the development process was provided.

Chapter 6

Computational Study

The present Masters work has, as its main goal, to create a viable solution to a currently rules based system for CDBs importation by using machine learning capabilities. In order to accomplish this type of system various tests are necessary, both in order to know how the current system is affected by the different features that compose its model, and also be able to compare the different alternatives of the final machine learning based system.

This chapter will describe the hypotheses that should be confirmed in order to achieve the aforementioned goals, as well as the process and tests used in order to validate those hypotheses. After this description, the hypotheses will be verified through the tests presented beforehand.

6.1 Hypotheses

In order to develop and validate the proposed system, several different conjectures should be validated.

One such hypotheses is that, for the current system, different variables (features) affect the outputs to different degrees. This should be tested and confirmed, as it could affect the features which should be used in the developed classifier's training.

Additionally, a reasonable hypothesis would be that diverse algorithms have different performances both in training speed, precision and recall, as such those two metrics should be compared between them in order to identify the one which performs better in each case.

Finally, it is fairly clear that, depending on the prototype algorithm used, the number of false/positives and false/negatives will tend to vary, as such, their performance in this regard should be compared, in order to find out if there truly is a difference between them.

6.2 Evaluation Methodology

Normally, an evaluation methodology is defined as a plan to help others to better understand the various phases needed to obtain a valid evaluation [84]. Because hypotheses and metrics have already been defined and the performed tests will be further talked about in the following section, focus will be given to the processes through which values for the metrics to be tested are obtained. For instance, if one wishes to measure its students capabilities in a given subject, the most commonly used evaluation methodology are exams.

In the present case, there is no need for many complex processes, since the metrics being measured and used in the different proposed tests are mostly directly computed by the algorithms training and execution (e.g. the training time). The metrics for which the use of a process is needed are precision/recall and false/positive false/negative rates, since both variables are not a direct result of execution and need to be computed through extra

operations. For precision and recall, a possible way to obtain good measures is to employ a cross-validation scheme for their computation through a series of train/test splits of an original data-set. In the case of false/positives and false/negatives, data collection can be performed simply by performing predictions for known cases and storing both the predicted and expected results as well as the classifier's confidence level in that decision.

6.3 Metrics & Tests

In order to test the different cases presented in the Hypotheses section three different kinds of test are used.

In the case of testing the influence of features on the final decision for the current system, Sensitivity Analysis Tests would be adequate. To compare both training speed, as well as precision and recall between developed prototypes, tests that allow the performance of comparisons in a case by case fashion, such as t tests, the Mann Whitney "U" test (a.k.a. Wilcoxon rank-sum test), Wilcoxon Signed-Rank Test, or even Sign Test should be used. However the decision of which of these is more appropriate heavily depends on both the collected data and their specific characteristics, as such the used test will be referred when describing how the hypothesis was evaluated. Finally, in order to measure and compare false/positive false/negative rates and overall performance of the prototypes ROC Tests are the most suitable option.

Each test was chosen due to being appropriate for the specific hypotheses which was being evaluated. This means that it was chosen taking into account prerequisites inherent to the available data, as well as the type of hypotheses which was being tested. Sensitivity analysis specifically targets the way in which different feature inputs change a model's output. Hypothesis test's which compare two populations and allow for a rank to be establish between them are appropriate for taking conclusions regarding the classifiers' precision, recall and training time. Finally, ROC curves are a staple in the measurement of classifier performance as well as analysis of their false/positives and false/negatives.

6.3.1 ROC Tests

ROC are useful tools in classifier evaluation. A ROC plot displays the performance of a binary classification method with either continuous or discrete output. It shows not only the sensitivity, proportion of correctly classified positive cases (a.k.a. recall, or true positive rate), but also the specificity (although this metric is plotted as 1-specificity and can also be referred to as true negative rate), the proportion of correctly classified negative cases as the output is moved over the range of possible values [85].

In ROC plots the Area Under the Curve (AUC) measures the performance of a given classifier and is normally used as a comparison point between classifiers, in which the one with a higher AUC is considered better [85]. The normally used approach to prove whether a real different exists between AUCs is to compute a critical ratio (z). The necessary variables are both the area and standard error associated with each curve (A_1, A_2 and SE_1, SE_2, respectively) and the estimated correlation between the two areas (r, which can vary between a range of tabled values). The critical ratio can then be computed by applying the formula [86]:

$$z = \frac{A_1 - A_2}{\sqrt{SE_1^2 + SE_2^2 - 2rSE_1SE_2}}$$

The computed critical ratio can then be easily converted into a regular $p - value$, allowing for an easy conclusion regarding the hypothesis of whether the difference in the areas under the two ROC curves derived from the same set of data is random or real.

6.3.2 Sensitivity Analysis Tests

A possible definition for sensitivity analysis is as follows: "*The study of how uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input*" [87]. However this definition is not of much use without knowing what a model means in this context. Simply put a model is a representation of a given system or problem made through a mathematical approach (e.g. a simple body mass index prediction model could be accomplished by simply computing the formula $BMI = Weight/Height^2$, in which weight is measured in kilograms and height in meters). Knowing this we can then say that sensibility analysis can be summed up as an activity that aims to investigate the degree to which a variable used in a model (e.g. the height in the BMI example) can affect that models results (i.e. the BMI itself).

There is an array of different approaches one can take when performing sensitivity analysis. In this context two methods will be presented: one from which conclusions are taken through plot observation, and another that tries to establish a mathematical relation relatively to the sensitivity caused by a variable. These are:

1. **Scatterplots:** Scatterplots are useful because they easily allow the comparison of different variables relatively to their underlying pattern. The presence of a possible pattern in a scatterplot is usually a good sign of the sensitivity between a variable and a model's output. The more clear the pattern is, the more sensible the output is to that variable. For instance, in Figure 31, Y is crescently more sensitive to Z_1, Z_2, Z_3 and Z_4, that is the case since it is clearly possible to see that a pattern gradually becomes more clear in each case's plot [87].

2. **Correlation Coefficients:** One more pragmatic way to acquire a sensitivity measure of an input variable relatively to the produced output is to compute the correlation coefficient existent between the two variables. In statistical terms, correlation represents a method of assessing a possible association between two variables. Correlation is measured by a correlation coefficient, which represents the strength of the association between the variables in question. It is a dimensionless quantity that takes a value in the range $[-1, 1]$. A correlation coefficient of zero indicates that no relationship exists between two variables, and a correlation coefficient of -1 or +1 indicates a perfect relationship. If there is a strong direct correlation the coefficient will tend to approach 1. On the other end if a strong inverse relationship will tend to generate values closer to -1 [76]. Various methods exist to compute this metric, some include: Pearson product-moment correlation coefficient, Spearman's rank correlation coefficient, Kendall's rank correlation coefficient and Point-biserial correlation coefficient [88].

6.4 Testing

Having presented the various hypotheses which should be validated as well as the different tests to be used the only thing left to do is to actually perform them. In order to simplify things it should be noted that any normality requirement from a statistics test, unless stated otherwise, is being respected. This is normally assured through the use of the central limit theorem since a significant number of instances is used (e.g. $n > 30$) [89].

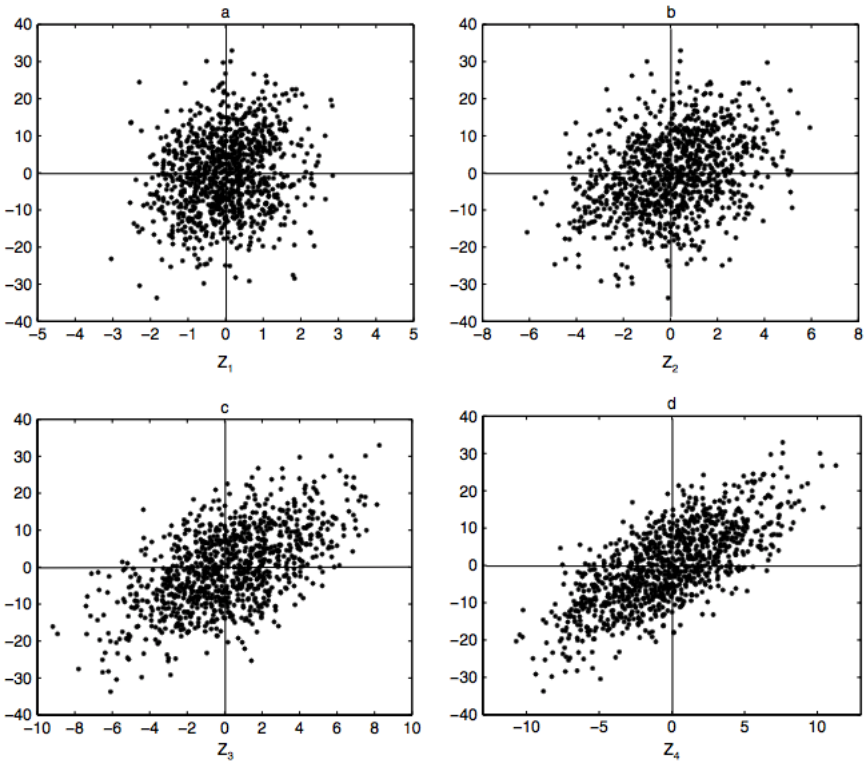


FIGURE 31: Scatterplot of Output Variable Y versus Input Variables Z_1,Z_2,Z_3 and Z_4 [87]

6.4.1 Sensitivity

It was suggested that the different features used by the current CDB judging system affect outputs in varying degrees. In order to test this conjecture, a balanced data-set of approximately one-thousand observations was created as a starting point. After this, for each feature which could affect the output of an importation, a plot was created in order to understand the relation between the two.

However, since the output variable at hand is binary there are multiple cases where a plot does not clearly show if a relationship exists, or even if it is stronger or weaker than one seen before. To accurately measure the strength of the relation between the different variables and the system's output the correlation coefficient between the two was computed. For continuous variables (e.g. hard-spam and soft-spam percentages), a Point-biserial correlation coefficient[1] was computed. For categorical variables (such as an user's country or motive for being blocked), Pearson's chi square test was employed followed by the use of Cramér's V[2]. Since all computed *p-values* were inferior to the *alpha* for a confidence level of 95%, we can say that the performed tests are valid.

Due to the performed tests being valid, we can easily observe and compare the relationship established between each feature and the current system's output. The various plots and correlation coefficient's can be observed in Sensibility Analysis Results.

6.4.2 Training Time

One of the hypotheses suggested above stated that the training speed for the various algorithms should be different. In order to verify if this was indeed the case, ten different

[1]Refer to: https://bit.ly/2ybr61D
[2]Refer to: https://bit.ly/2AKG9wH

runs for training sets with varying numbers of instances were executed for each classifier. A timer was started when each training session began and stopped when it finished. In this way, a set of points related to the training time of each classifier was created. This data can be checked in Table 29:

TABLE 29: Time (s) Necessary to Train a Classifier
Based on a Variable Number of Instances

Number of Instances	NBC	SVC	RFC	ABC
2624	0.1221	8.1793	0.4216	3.8770
5248	0.2903	70.9106	0.8345	11.0461
7872	0.3594	151.6717	1.2347	22.1670
10496	0.6371	226.6897	1.7288	29.7363
13120	0.6438	694.9852	2.1989	40.3456
15744	1.0776	1173.5485	2.7002	52.8532
18368	1.2128	1735.5273	3.6840	66.4181
20992	1.3016	2432.1006	3.6970	63.6106
23616	1.4847	2711.2702	4.3315	92.9697
26240	1.6045	3748.4617	4.8756	120.9695

Trough this data and by applying a generalized linear model to each group of observations, the following plots were created (Figure 32):

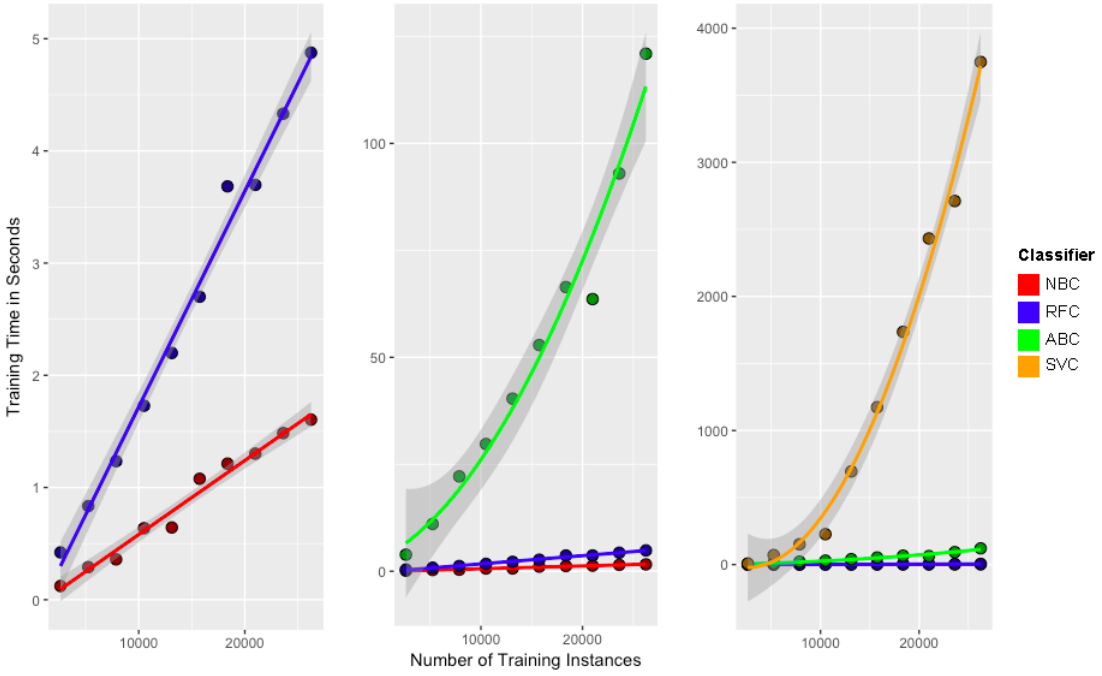


FIGURE 32: Classifier's Training Time (s) per Number of Training Instances

Here the plot was divided into tree different views containing two, three and then four classifiers, so as to easily transmit the discrepancy in training times between each case.

If this plot was used to trace an ordered list ascendingly in regards to which algorithm was less impacted in terms of training time by the number of instances, then the following order would be established:

$$NBC < RFC < ABC < SVC$$

However, this plot by itself does not constitute solid statistical evidence, hence an appropriate test is necessary.

In this case, the use of paired t-tests[3] would be adequate, however, the different populations cannot be guaranteed to follow a normal distribution (verified trough the use of Shapiro–Wilk's normality test[4]) and their differences are skewed. Hence a non-parametric equivalent, that does not rely on these two factors is necessary. The chosen type of test for this task was a two-sample sign test[5].

By applying this test to each pair of classifiers, the following Table 30 was defined:

TABLE 30: Computed $p - values$ for Two-sample Sign Test Between each Classifier's Populations

	NBC-RFC	NBC-ABC	NBC-SVC	RFC-ABC	RFC-SVC	ABC-SVC
$p - value$	0.002	0.002	0.002	0.002	0.002	0.002

The main point of interest for this test was if each population was truly different from the other. Hence, a two-sided test was employed. Due to that, the null and alternative hypotheses could be defined as:

$$H_0 : \widetilde{P1} - \widetilde{P2} = 0$$
$$H_1 : \widetilde{P1} - \widetilde{P2} \neq 0$$

in which P1 stands for population one and P2 for population two.

Since for each combination of classifier's populations, the resulting *p-value* was less than the α for a confidence level of 95%, the null hypothesis is rejected. This means that the true median difference between populations is not equal to zero, which indicates that they are distinct. Which corroborates that the previously established order between classifiers, regarding which was less affected by the number of training instances, is valid. It also means that the plots presented in Figure 32 can be taken as an accurate representation of the difference between the classifier's training time.

6.4.3 Precision & Recall

One other hypothesis which was suggested, stated that there should be distinguishable differences between the performance of the various algorithms, both in terms of precision as well ass recall.

In order to test this conjecture the first step taken was to collect data regarding the classification provided by each algorithm. To do that, the various algorithms were repeatedly exposed to a fifteen-fold cross-validation scheme where a StratifiedKFold[6] approach was used in order to obtain various balanced train/test splits (which were formed simultaneously). The resulting confusion matrices of each cross-validation cycle were then summed and stored. In this way, the process was repeated until enough data could be stored so as to allow for a quasi-normal distribution to occur according to the central limit theorem. The number of observations collected for each classifier was forty two, in which twenty one targeted the positive class and the rest the negative class.

It should be mentioned that the SVC, specifically, was not included in this test. This decision derived from the fact that its training time was to great to allow for the repetitive use of the employed data generating strategy in a pragmatic fashion.

[3]Refer to: https://bit.ly/2INBvBt
[4]Refer to: https://bit.ly/1gc8eyQ
[5]Refer to: https://bit.ly/2J7DR1j
[6]Refer to: https://bit.ly/2L3LIdR

An adequate test to compare the various classifier's precisions and recalls would be to use a t-test[7]. However, a t-test's assumptions include homogeneity of variance, which does not exist in this context, as was verified by performing Levene's Test[8]. Hence, a new equivalent test which does not rely on that factor was selected. That test was the Mann–Whitney U test [9].

Since a rough comparison between classifiers had been made (in the Classifier Tuning section), there exists a general idea of how they should stack up to one another. Due to this, in order to compare NBC to both RFC and ABC, a one sided test to the left is adequate. The results of such a test, targeting both overall precision and recall can be observed in Table 31.

TABLE 31: $p - values$ of Score Distribution Comparison Between NBC and Other Classifiers Using a Left-tailed Mann–Whitney Test

	NBC - RFC	NBC - ABC
Precision $p - value$	3.118×10^{-5}	3.116×10^{-5}
Recall $p - value$	3.103×10^{-5}	3.091×10^{-5}

If we assume that DS_{NBC} represents the distribution of scores for the NBC results and DS_{XC} represents the distribution of scores for results of the other classifier. Then, the null and alternative hypothesis for this test could be defined as:

$$H_0 : DS_{NBC} = DS_{XC}$$
$$H_1 : DS_{NBC} < DS_{XC}$$

Considering the presented *p-values*, it can be said that, for a confidence level of 95%, both RFC and ABC possess better overall precision and recall values than NBC, since in both cases the null hypothesis was rejected. Which means that both RFC and ABC are more efficient than NBC in regards to classification.

The next step would be to compare the RFC with the ABC. Since both classifiers proved to have close scores before-hand, the first test should be two-sided so as to assert if a real different exists in the overall precision and recall distributions of both classifiers. By applying the test the results observed in Table 32 were obtained.

TABLE 32: $p - values$ of Score Distribution Comparison Between RFC and ABC Using a Two-sided Mann–Whitney Test

	RFC - ABC
Precision $p - value$	0.9039
Recall $p - value$	0.9679

For this test, assuming that DS_{RFC} represents the distribution of scores for results of the RFC and DS_{ABC} represents the distribution of scores for results of the ABC, the null and alternative hypothesis can be defined as:

$$H_0 : DS_{RFC} = DS_{ABC}$$
$$H_1 : DS_{RFC} \neq DS_{ABC}$$

[7]Refer to: https://bit.ly/2Lyi6pI
[8]Refer to: https://bit.ly/2sbQbnK
[9]Refer to: https://bit.ly/1RqN1D2 & https://bit.ly/2L1aQl7

Hence, for a confidence level of 95% we cannot deny the null hypothesis. Meaning there are no significant differences between both classifiers regarding the overall performance of both precision and recall.

However, that does not guarantee that no substantial difference exists for the precision and recall of both classifiers if both positive (blocked CDB importations) and negative (valid CDB importations) classes are judged separately. In order to verify if a difference was present in the classifiers score distributions for each class type, the test was repeated using the data for both positive and negative classes independently. The obtained results are shown in Table 33.

TABLE 33: $p - values$ of Score Distribution Comparison Between RFC and ABC per Class Type Using a Two-sided Mann–Whitney Test

RFC - ABC	Negative Class	Positive Class
Precision $p - value$	$3.105 \times 10^{-}8$	$2.182 \times 10^{-}7$
Difference between mean score ranks	$-5.660 \times 10^{-}4$	$1.083 \times 10^{-}2$
Recall $p - value$	$5.926 \times 10^{-}8$	$2.733 \times 10^{-}8$
Difference between mean score ranks	$3.709 \times 10^{-}4$	$-2.405 \times 10^{-}2$

Taking into account that the null and alternative hypotheses for these tests are the same as the ones defined in the last case, various conclusions can be discerned if both the *p-values* and the difference between mean score ranks for each case are taken into consideration (keep in mind the order established in the comparison):

1. For a confidence level of 95% it is safe to reject the null hypothesis of equality between distribution of scores for any combination of factors used in the test, meaning the alternative hypothesis is true at this confidence level;

2. Due to the previous point, and the positive mean score rank difference between the compared classifiers. It is possible to conclude that the distribution of scores for RFC is greater than the one for ABC both in the case of precision of the positive class, as well as recall of the negative class;

3. Inversely, it is possible to conclude that the distribution of scores for ABC is greater than the one for RFC both in the case of precision of the negative class, as well as recall of the positive class.

These are also supported by the visual observation of the plotted data points used in each comparison, as can be analyzed in Figure 33.

If, as was stated beforehand in the Classifier Tuning section of this Maters' Thesis, more emphasis should be given to the recall of the positive class, then the best performing classifier is AdaBoost. However, for the sake of clarity, the various rankings (greater rank equals better performance) for each pair of metric/class are presented bellow:

- Negative Class' Precision: $NBC < RFC < ABC$
- Negative Class' Recall: $NBC < ABC < RFC$
- Positive Class' Precision: $NBC < ABC < RFC$
- Positive Class' Recall: $NBC < RFC < ABC$

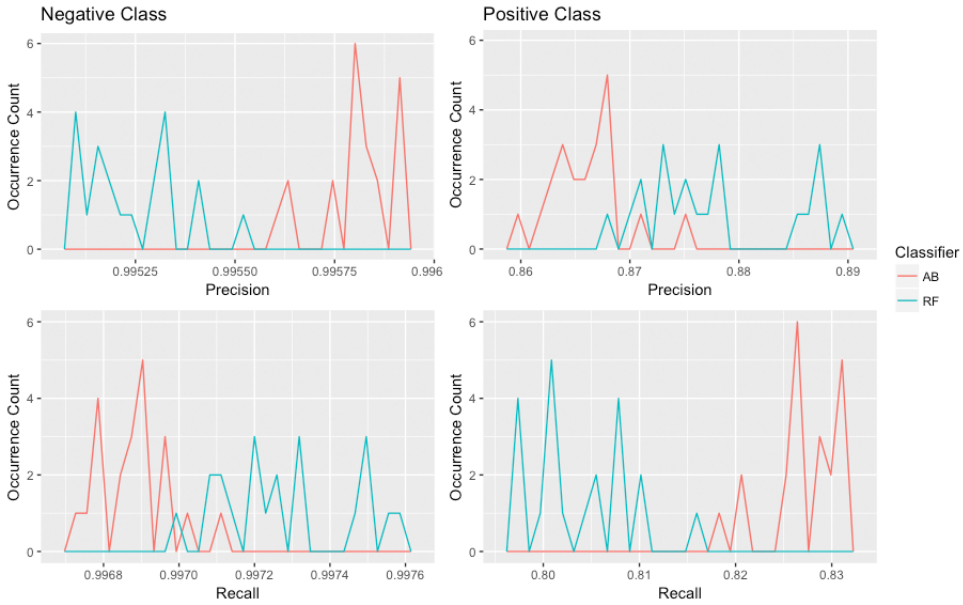


FIGURE 33: Occurrence Count of Precision and Recall Scores for Each Classifier and per Class Type

6.4.4 Type I and II Errors

Beforehand, it was mentioned that the number of false/positives and false/negatives would be different depending on the algorithm used in the classification task. It is self-evident that the amount of false/positives and false/negatives correlate directly and inversely to the amount of true/positives and true/negatives. This means, that by measuring a given classifier's true/positive and true/negative rates we can safely validate if there is a different between the number of false/positives and false/negatives produced by the two. Here, ROCs excel since they provide an easy way to both observe and measure the performance of an algorithm regarding its true/positive and true/negative rates.

Taking into account the previous discussion, we can express the hypotheses being evaluated as:

$$H_0 : AUC1 - AUC2 = 0$$
$$H_1 : AUC1 - AUC2 \neq 0$$

This is possible since the AUC of a ROC increases proportionally to an algorithm's true/positive rate and inversely to its true/negative rate (since false/positive rate = 1 - specificity, also known as true/positive rate), which means that a bigger area will represent a better performing algorithm in terms of false occurrences.

In order to apply this logic to the developed classifiers, the first step taken was to extract data targeting the classification probabilities generated by the same train/test split in each of their cases.This train/test split was based on a data-set with 26240 entries, and its division followed a ratio of 70% training data to 30% testing data, meaning that the classification probabilities were based in 7872 instances.

Once that was concluded the next step taken was to use that data to plot the ROC for each individual classifier and calculate their AUC. The obtained ROCs were depicted on Figures 34, 35, 36, 37:

If we make a direct comparison without further analysis then the order established between algorithms, in regards to AUC, would be: Naïve Bayes < Support Vector Machine < Random Forest < AdaBoost. This ordering, however, may not be correct, as the differences

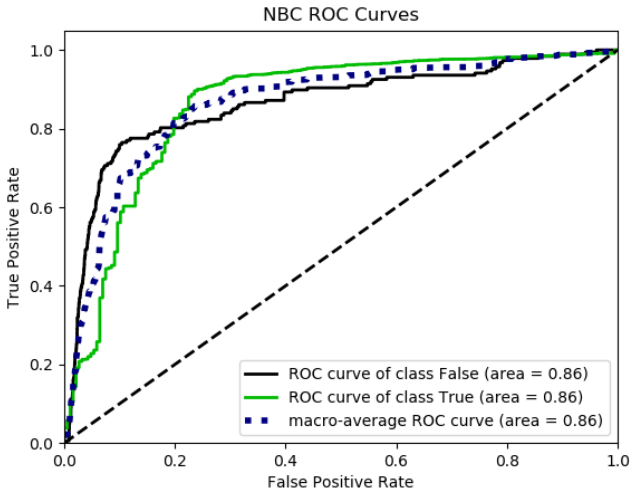


FIGURE 34: ROC generated for NBC

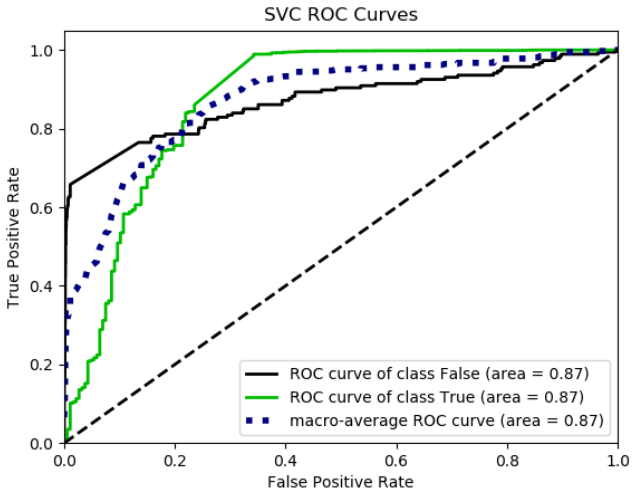


FIGURE 35: ROC generated for SVC

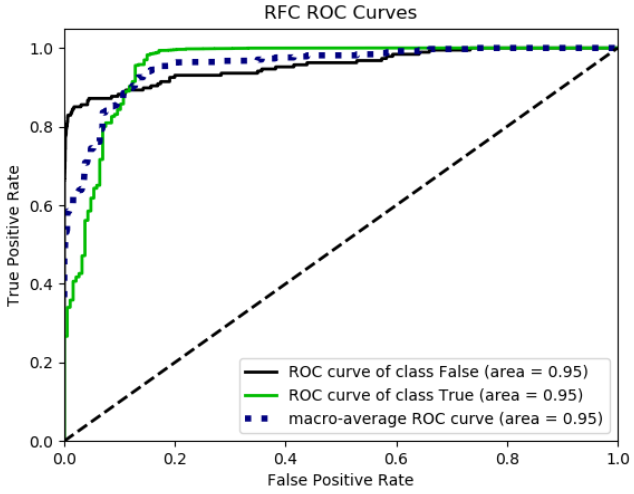


FIGURE 36: ROC generated for RFC

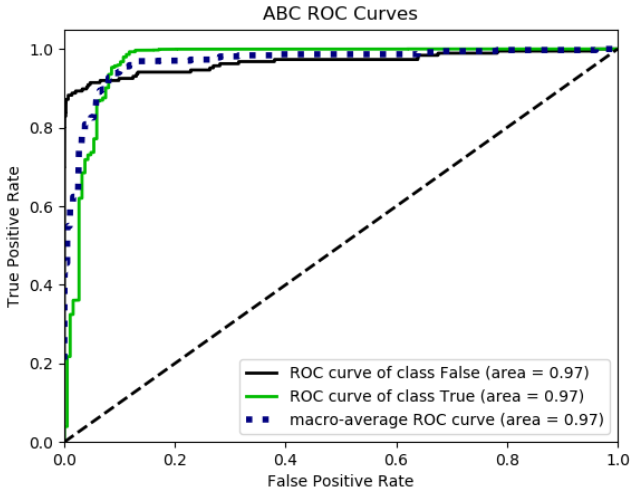


FIGURE 37: ROC generated for ABC

between each of the ROCs could be caused by a random set of factors. Hence, it is necessary to compute the critical ratio for the various AUCs being compared, so as to ascertain that the observed differences between areas are, indeed, real. This can be done trough the formula discussed in 6.3.1, however, the computation of the standard errors related to each area is required, as well as the correlation coefficient between areas.

The standard error of each of the ROCs was obtained by computing the square root of the variance of each ROC (as was suggested by the pROC's package author [90]). The computed values are presented in Table 34.

TABLE 34: SE For Each Classifier's ROC

	NBC	SVC	RFC	ABC
SE	0.01890369	0.01788823	0.01482855	0.01199246

In order to find the correlation coefficient between the various ROCs' AUCs a more complex operation was necessary (which is discussed in dept at [86]). First and foremost, two intermediate correlation coefficients, r_T and r_F were computed for each pair of ROCs. The first, r_T, is the correlation coefficient for the ratings given to CDBs, which were imported successfully by the two classifiers. The second, r_F, is the correlation coefficient of the ratings given to CDBs, whose importations were blocked, by the two classifiers. This was done trough the simple use of Kendall's τ[10] for each case and was proceeded by calculating the averages between r_T and r_F in each case. Next, the average area for each pair of ROCs being compared was found. With these two averages the correlation coefficient between areas (r) was determined through the use of Table I of [86]. This information is summarized through Table 35.

Once both the ROCs' standard errors and the correlation coefficients between each pair were found, the various critical values (z) were computed, through the expression

$$z = \frac{A_1 - A_2}{\sqrt{SE_1^2 + SE_2^2 - 2rSE_1SE_2}}$$

[10]Refer to: https://bit.ly/2Jhrtf8

TABLE 35: Computation of r
*A stands for area and Av stands for average

	NBC - SVC	NBC - RFC	NBC - ABC	SVC - RFC	SVC - ABC	RFC - ABC
r_T	0.01914511	0.03770009	0.04730588	0.556145	0.5099295	0.6875441
r_F	0.09218412	0.2981595	0.2981595	0.5874717	0.6389271	0.879199
Av	0.05566462	0.1679298	0.1727327	0.5718083	0.5744283	0.7833716
A1	0.86	0.86	0.86	0.87	0.87	0.95
A2	0.87	0.95	0.97	0.95	0.97	0.97
Av	0.865	0.905	0.915	0.91	0.92	0.96
r	0.04	0.11	0.11	0.49	0.47	0.68

Since a rough comparison between classifiers had previously been done, as well as an estimation of their ROCs' AUCs, and only improvements are of interest, then a one-tailed test is an appropriate approach, which implies the following null and alternative hypothesis:

$$H_0 : AUC1 - AUC2 = 0$$
$$H_1 : AUC1 - AUC2 > 0$$

For this type of test, the *p-value* for each critical ratio z is easily obtainable through the use of tabled values for the Gaussian distribution, which can be seen in Table 36.

TABLE 36: Critical Ratio z and Respective $p - value$ for each ROC Pair

	NBC-SVC	NBC-RFC	NBC-ABC	SVC-RFC	SVC-ABC	RFC-ABC
z	0.392146	3.963695	5.178009	4.78157	6.176189	1.811791
$p - value$	0.3483	< 0.0000	< 0.0000	< 0.0000	< 0.0000	0.0351

From these values, and considering that both a confidence level of 95% is used and that the comparisons occurred in the reverse order of that is presented in Table 36 (e.g. ABC was compared with RFC and not the reverse, which meant a positive numerator when applying the z ratio formula). It is possible to take the following conclusions:

- The AUC difference between the NBC's ROC and the SVC's ROC is not statistically significant, since the computed *p-value* is superior to a 5% significance level.
- The AUC difference between every combination of either NBC or SVC's ROCs and the ROCs of RFC or ABC is statistically significant, since the computed *p-value* is inferior to α for a 95% confidence level, which allows for the rejection of the null hypothesis, meaning there is a reason to believe that the diferrences between the AUCs are significant.
- The AUC difference between the RFC's ROC and the ABC's ROC is statistically significant, since the computed *p-value* is superior to α for a 95% confidence level. However it should be noted that the same would not hold true in case of a 99% confidence level.

Restating the aforementioned order established by the AUCs of the various classifiers' ROCs. It is now possible to express it, with a confidence level of 95%, as:

$$NBC = SVC < RFC < ABC$$

This effectively means that in terms of false classification cases, their number tend to decrease has the classifier's rank in this order increases.

6.5 Summary

Throughout this chapter, the various objectives which should be achieved by the developed solution were described. These were then translated to hypotheses, which rely on the valid testing of specific metrics in order to either be proven or denied. The process through which these metrics would as well as the various tests to be used for each case were presented and then explained. After this, the aforementioned tests were performed in order to reach a conclusion regarding the established hypotheses. In each case, the various steps necessary for the test performance were explained, and a conclusion regarding the tested hypothesis was reached.

The conclusions obtained form each of the tested hypotheses were the following:

- First, in terms of training time, the effective performance order between classifiers is $SVC < ABC < RFC < NBC$ (a greater rank means a lower training time). It should be mentioned that the SVC has a much greater training time growth rate than the other classifiers;
- Secondly, regarding the classifiers performance for the recall of the positive class, the performance order would be $NBC < RFC < ABC$ (in which a greater rank equals a better recall score);
- Finally, concerning the AUC of each classifier's ROC curve (meaning their type I and II error rates), the established performance order was $NBC = SVC < RFC < ABC$ (in which a greater rank means a lower rate of occurrence for these errors).

Chapter 7

Conclusion

Throughout this document the process necessary to the creation of a new CDB importation solution for a digital marketing platform was presented.

First, the organizational background for the problem, as well as its root cause, was explained and a possible solution was proposed. Taking into account the possible solution, as well as the problem's context, state of the art research was performed. It also included previous work involving similar topics. Additionally, a brief view of the technologies which could be used in the development process was provided. From this process, it was concluded that the new CDB importation judgment prototypes to be developed should use a SVM, a BAG-DT and a NB approaches. However, an additional Adaptative Boosting based approach was also developed, and it was decided that the tools to be used would be R and Python (namely scikit-learn).

Succeeding this first phase, an analysis regarding the value which could be extracted from the solution was executed. It was done since the organization should not implement a solution if it did not bring them any additional value to their current condition. This analysis effort was accomplished using various proven standards such as Porter's value chain and the business model canvas. Plus, the analytic hierarchy process was used in order to establish a mathematical relationship between the types of solution which could be developed, while taking into account a set of criteria which should be present in the final result. It was concluded that the approach that should be taken in terms of machine-learning was a supervised approach, which, at the same time, benefited from unsupervised discovery of missed features and tuning of developed classifier prototypes.

Having grounded the fact that the proposed system was indeed worth pursuing, the necessary functional and non-functional requirements for the solution were listed. Taking those into account, the ideal business process in which it would be used was drawn. Two possibilities for the solution's structure were presented, the second being the most appropriate one (Figure 24), and the inherent data model was explained. Also, the API through which the system should be made available had its main use cases planned and properly explained.

To clarify the way in which the planned architecture was accomplished, its development was disclosed. Firstly, the selection process for the procurement of the several features which should be used to train the classifier prototypes was explained. After which, the components responsible for data-set pre-processing and abstraction of prototype classifiers were presented. The process through which the various classifiers were tuned/optimized was, then, elucidated. The routes present in the solution's API, as well as their capabilities were exposed and explained. Lastly, the various unit test which accompanied the entire process were listed.

In order to help and evaluate the solution's development, various hypotheses were proposed to be validated. These hypotheses were: that the features used by the current judging system affected its output by varying degrees; that the training speed and precision/recall of the prototypes developed for the new system were different between each

other to a significant level; and that the false/positives and false/negatives for each classifier prototype was similarly different by a significant degree. For each of them, a proper statistical test adequate to the case was chosen and then performed in order to confirm or deny those hypotheses. The conclusions reached through each test were explained.

In conclusion, **this thesis described a valid approach to the creation of a CDB importation analysis system based on machine-learning techniques for spam prevention**. This was done in order to counter the misuse of E-goi's marketing platform by possible spammers, who would try to use its capabilities to produce spam campaigns, **while at the same time keeping the system more flexible to change/evolution than its past iteration**. By basing the current system in machine-learning technologies, good results were obtained while simultaneously allowing for easy system editability via the data used to train it.

From the different developed prototypes **the testing performed in order to compare each of them to one another** (in the Computational Study chapter) **allowed for the establishment of the following order in terms of overall quality: NBC < SVC < RFC < ABC**. **The capabilities present in each were then made available to external users through a documented API in order to guarantee the system's ease of use**. Currently, the system is being test in a controlled environment with real data and it is expected that good results shall be obtained.

Future work for this solution include further optimization of the features used in the classifier's training, as well as the creation of more friendly data storage infra-structures. Further progress may be accomplished through implementation of other classifier approaches, for which the system is prepared due to that concept's abstraction.

Bibliography

[1] J. Strauss *et al.*, *E-marketing*. Routledge, 2016.

[2] J. e. a. Santos, "E-goi (description, diagnosis, development plan and swot analysis)." Report accomplished for the subject of CORGA in the context of ISEP's bachelors in Informatics Engineering, 2017.

[3] "E-goi plan 5001," 2018. Available at https://www.e-goi.com/egoi-plan-5001/.

[4] "E-goi regular plan," 2018. Available at https://www.e-goi.com/pricing.

[5] "E-goi funcionalidades," 2018. Available at https://www.e-goi.pt/funcionalidades/.

[6] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.

[7] T. M. Mitchell, *Machine Learning*. McGraw-Hill, Inc., 1 ed., 1997.

[8] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, "An overview of machine learning," in *Machine Learning*, Symbolic Computation, pp. 3–23, Springer, Berlin, Heidelberg, 1983.

[9] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[10] R. Carpenter, "Cleverbot [computer program]," 2006. Available at http://www.cleverbot.com/.

[11] "Data science overview," 2018. Available at http://www.newhorizonanalytics.com/data-science-overview/.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

[13] I. G. Maglogiannis, *Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies*. IOS Press, 2007.

[14] "Decision tree learning," 2018. Available at https://en.wikipedia.org/wiki/Decision_tree_learning.

[15] "Bayesian network," 2018. Available at https://en.wikipedia.org/wiki/Bayesian_network.

[16] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1. Springer series in statistics New York, 2001.

[17] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[18] "Pca - principal component analysis," 2012. Available at http://www.nlpca.org/pca_principal_component_analysis.html.

[19] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[20] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[21] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[22] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.

[23] "What is email deliverability? campaign monitor," 2017. Available at https://www.campaignmonitor.com/resources/glossary/email-deliverability/.

[24] K. E. Himma and H. T. Tavani, *The handbook of information and computer ethics*. John Wiley & Sons, 2008.

[25] L. Hatton and A. John, "Delivering genuine emails in an ocean of spam," *IEEE Software*, vol. 34, no. 4, pp. 11–15, 2017.

[26] S. Duffy, "A guide to email deliverability for b2c email marketers," *Journal of Direct, Data and Digital Marketing Practice*, vol. 9, no. 2, pp. 156–167, 2007.

[27] V. Cox, *Translating Statistics to Make Decisions: A Guide for the Non-Statistician*. Apress, 2017.

[28] W. McKinney, *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, Inc., 2012.

[29] R. A. Becker, J. M. Chambers, and A. R. Wilks, "The new s language," *Pacific Grove, Ca.: Wadsworth & Brooks*, 1988.

[30] D. M. S. W. N. Venables and the R Core Team, *An introduction to R Notes on R: A Programming Environment for Data Analysis and Graphics*. Network Theory, 3.4.3 ed., 2017.

[31] G. Holmes, A. Donkin, and I. H. Witten, "Weka: A machine learning workbench," in *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*, pp. 357–361, IEEE, 1994.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[33] B. Yu and Z.-b. Xu, "A comparative study for content-based dynamic spam classification using four machine learning algorithms," *Knowledge-Based Systems*, vol. 21, no. 4, pp. 355–362, 2008.

[34] M. E. Tipping, "The relevance vector machine," in *NIPS*, 1999.

[35] S. Das, "A relative study of spamming detection using machine and non machine learning classifier," *International Journal Of Computer Engineering & Applications*, vol. 11, 2017.

[36] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd international conference on Machine learning*, pp. 161–168, ACM, 2006.

[37] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE transactions on systems, man, and cybernetics*, pp. 580–585, 1985.

[38] P. Cunningham and S. J. Delany, "k-nearest neighbour classifiers," *Multiple Classifier Systems*, vol. 34, pp. 1–17, 2007.

[39] S. Jiang, G. Pang, M. Wu, and L. Kuang, "An improved k-nearest-neighbor algorithm for text categorization," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1503–1509, 2012.

[40] N. Gupta, "Artificial neural network," *Network and Complex Systems*, vol. 3, no. 1, pp. 24–28, 2013.

[41] A. T. Sabri, A. H. Mohammads, B. Al-Shargabi, and M. A. Hamdeh, "Developing new continuous learning approach for spam detection using artificial neural network," *European Journal of Scientific Research*, vol. 42, no. 3, pp. 525–535, 2010.

[42] A. Nosseir, K. Nagati, and I. Taj-Eddin, "Intelligent word-based spam filter detection using multi-neural networks," *IJCSI International Journal of Computer Science Issues*, vol. 10, no. 2, pp. 1694–0814, 2013.

[43] S. Chakraborty and B. Mondal, "Spam mail filtering technique using different decision tree classifiers through data mining approach-a comparative performance analysis," *International Journal of Computer Applications*, vol. 47, no. 16, 2012.

[44] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary pso with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Systems*, vol. 64, pp. 22–31, 2014.

[45] J. S. Rao and W. J. Potts, "Visualizing bagged decision trees.," in *KDD*, pp. 243–246, 1997.

[46] A. K. Sharma and S. Sahni, "A comparative study of classification algorithms for spam email data analysis," *International Journal on Computer Science and Engineering*, vol. 3, no. 5, pp. 1890–1895, 2011.

[47] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri, "Know your neighbors: Web spam detection using the web topology," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 423–430, ACM, 2007.

[48] H. Drucker and C. Cortes, "Boosting decision trees," in *Advances in neural information processing systems*, pp. 479–485, 1996.

[49] X. Carreras and L. Marquez, "Boosting trees for anti-spam email filtering," *arXiv preprint cs/0109015*, 2001.

[50] G. Caruana, M. Li, and Y. Liu, "An ontology enhanced parallel svm for scalable spam filter training," *Neurocomputing*, vol. 108, pp. 45–57, 2013.

[51] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural networks*, vol. 10, no. 5, pp. 1048–1054, 1999.

[52] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.

[53] P. Belliveau, A. Griffin, and S. Somermeyer, *The PDMA toolbook 1 for new product development*. John Wiley & Sons, 2004.

[54] R. G. Dyson, "Strategic development and swot analysis at the university of warwick," *European journal of operational research*, vol. 152, no. 3, pp. 631–640, 2004.

[55] "The swot analysis and what it can do for you," 2017. Available at http://www.aleanjourney.com/2017/03/guest-postthe-swot-analysis-and-what-it.html.

[56] A. Lindgreen and F. Wynstra, "Value in business markets: What do we know? where are we going?," *Industrial Marketing Management*, vol. 34, no. 7, pp. 732–748, 2005.

[57] E. R. Miles, "Techniques of value analysis and engineering," *Ekistics*, vol. 56, no. 336, pp. 119–121, 1989.

[58] V. A. Zeithaml, "Consumer perceptions of price, quality, and value: a means-end model and synthesis of evidence," *The Journal of marketing*, vol. 52, pp. 2–22, 1988.

[59] A. N. Leont'ev, *Activity, Consciousness, and Personality*. Prentice-Hall Englewood Cliffs, Nj, 1978.

[60] S. Boztepe, "User value: Competing theories and models," *International Journal of Design*, vol. 1, pp. 55–63, 2007.

[61] P. Kotler and K. L. Keller, *Creating Customer Value, Satisfaction, and Loyalty*. Pearson Education, Inc., 2009.

[62] T. Woodall, "Conceptualising 'value for the customer': An attributional, structural and dispositional analysis," *Academy of Marketing Science Review*, vol. 12, 2003.

[63] "Value proposition meaning in the cambridge english dictionary," 2017. Available at https://dictionary.cambridge.org/dictionary/english/value-proposition.

[64] "The value proposition canvas," 2017. Available at https://strategyzer.com/canvas/value-proposition-canvas.

[65] A. Osterwalder and Y. Pigneur, *Business model generation: a handbook for visionaries, game changers, and challengers*. John Wiley & Sons, 2010.

[66] "The business model canvas," 2017. Available at https://strategyzer.com/canvas/business-model-canvas.

[67] M. E. Porter, *Competitive advantage: creating and sustaining superior performance*. New York: Free Press, 1985.

[68] "Apple value chain analysis," 2017. Available at https://research-methodology.net/apple-value-chain-analysis/.

[69] T. L. Saaty, "The analytic hierarchy process: planning, priority setting, resources allocation," *New York: McGraw*, vol. 281, 1980.

[70] C. S. Marins, D. d. O. Souza, and M. d. S. BARROS, "O uso do método de análise hierárquica (ahp) na tomada de decisões gerenciais–um estudo de caso," *XLI SBPO*, vol. 1, 2009.

[71] E. Mu and M. Pereyra-Rojas, "Understanding the analytic hierarchy process," in *Practical Decision Making using Super Decisions v3*, pp. 7–22, Springer, 2018.

[72] P. Loucopoulos and V. Karakostas, *System requirements engineering*. McGraw-Hill, Inc., 1995.

[73] E. Evans, *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley Professional, 2004.

[74] M. Weske, *Business Process Management*. Springer, 2012.

[75] R. Elmasri and S. Navathe, *Fundamentals of database systems*. Addison-Wesley Publishing Company, 2010.

[76] M. M. Mukaka, "A guide to appropriate use of correlation coefficient in medical research," *Malawi Medical Journal*, vol. 24, no. 3, pp. 69–71, 2012.

[77] "Compute class weight," 2018. Available at http://scikit-learn.org/dev/modules/generated/sklearn.utils.class_weight.compute_class_weight.html.

[78] G. King and L. Zeng, "Logistic regression in rare events data," *Political analysis*, vol. 9, no. 2, pp. 137–163, 2001.

[79] K. M. Ting, "Confusion matrix," in *Encyclopedia of Machine Learning and Data Mining*, pp. 260–260, Springer, 2017.

[80] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *Journal of Machine Learning Research*, vol. 11, no. Jul, pp. 2079–2107, 2010.

[81] "How do i evaluate a model?," 2018. Available at https://sebastianraschka.com/faq/docs/evaluate-a-model.html.

[82] I. Mani and I. Zhang, "knn approach to unbalanced data distributions: a case study involving information extraction," in *Proceedings of workshop on learning from imbalanced datasets*, vol. 126, 2003.

[83] "Comparison of the different under-sampling algorithms," 2016. Available at http://contrib.scikit-learn.org/imbalanced-learn/stable/auto_examples/under-sampling/plot_comparison_under_sampling.html.

[84] M. Baehr, "Evaluation methodology," *Pacific Crest Faculty Development Series*, 2004.

[85] X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J.-C. Sanchez, and M. Müller, "proc: an open-source package for r and s+ to analyze and compare roc curves," *BMC bioinformatics*, vol. 12, no. 1, p. 77, 2011.

[86] J. A. Hanley and B. J. McNeil, "A method of comparing the areas under receiver operating characteristic curves derived from the same cases.," *Radiology*, vol. 148, no. 3, pp. 839–843, 1983.

[87] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

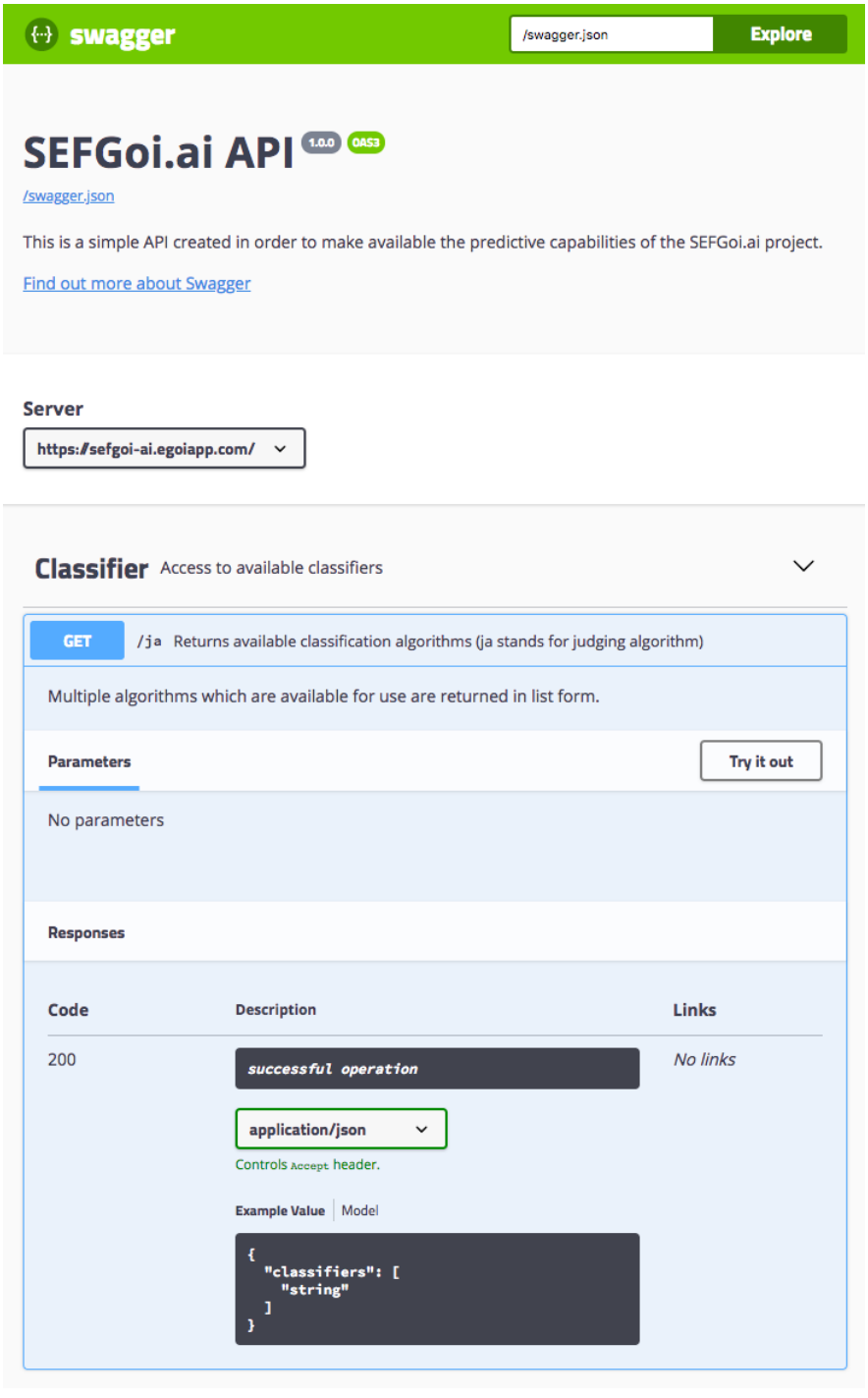
[88] "Correlation (pearson, kendall, spearman)," 2018. Available at http://www.statisticssolutions.com/correlation-pearson-kendall-spearman/.

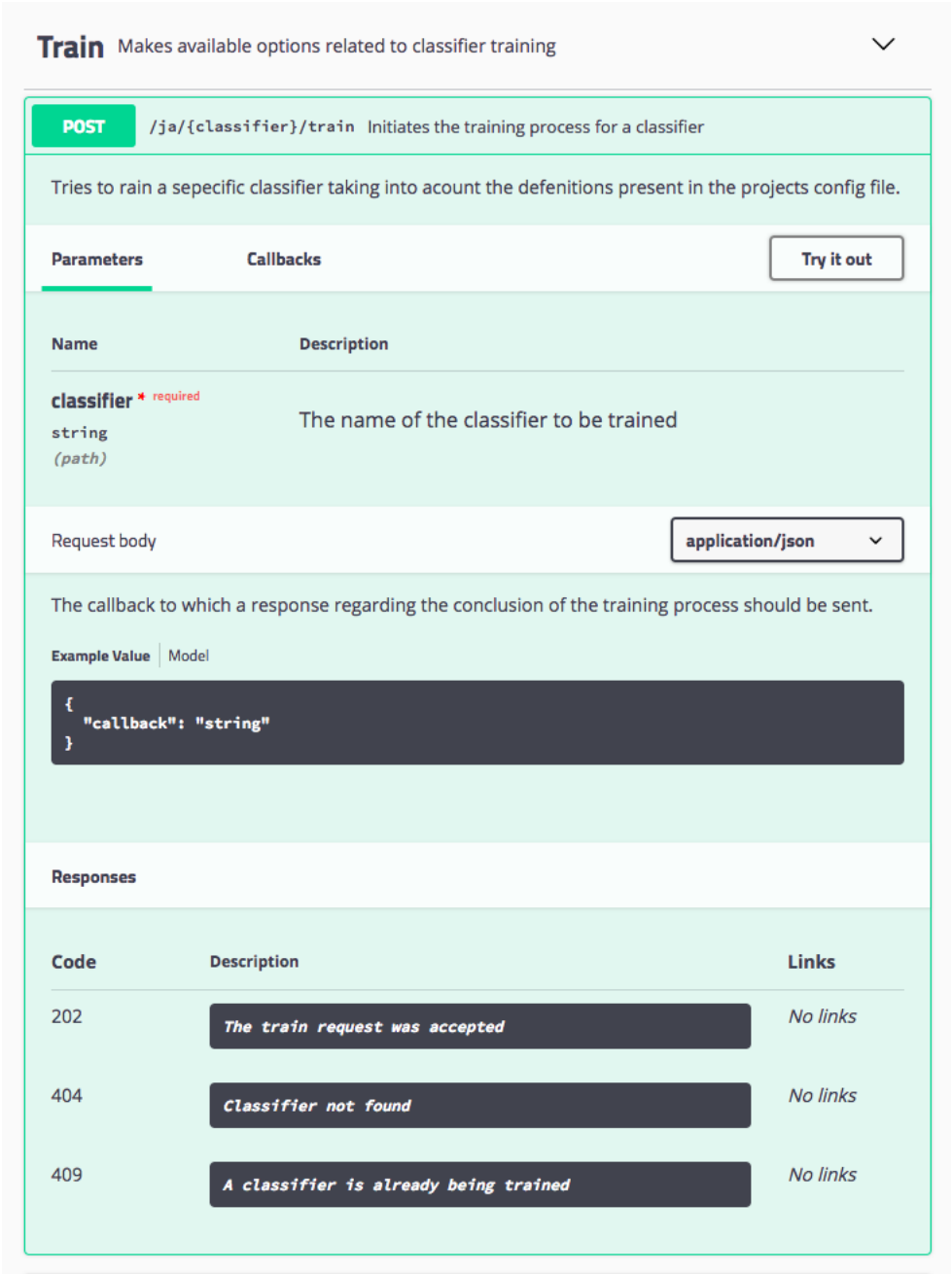
[89] "Central limit theorem," 2016. Available at http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_Probability/BS704_Probability12.html.

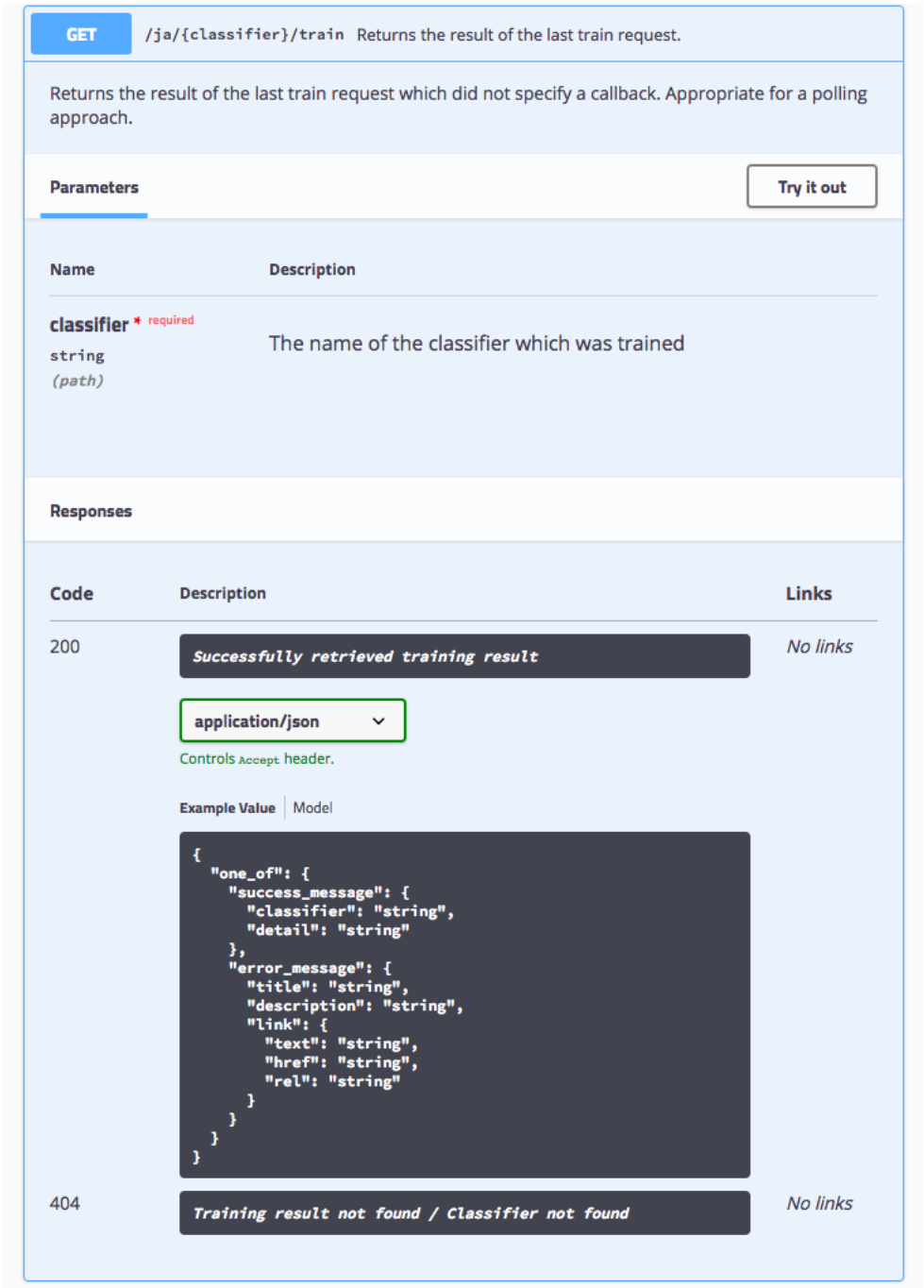
[90] "Get the standard error of an auc," 2014. Available at https://stackoverflow.com/questions/21456536/how-can-i-get-the-standard-error-of-an-auc-in-rocr.

Appendix A

API's Swagger UI




Train
Makes available options related to classifier training
POST
/ja/{classifier}/train
Initiates the training process for a classifier
Tries to rain a sepecific classifier taking into acount the defenitions present in the projects config file.
Parameters
Callbacks
Try it out
Name
Description
classifier * required
string
(path)
The name of the classifier to be trained
Request body
application/json
The callback to which a response regarding the conclusion of the training process should be sent.
Example Value
Model
{
"callback": "string"
}
Responses
Code
Description
Links
202
The train request was accepted
No links
404
Classifier not found
No links
409
A classifier is already being trained
No links



GET /ja/{classifier}/train Returns the result of the last train request.

Returns the result of the last train request which did not specify a callback. Appropriate for a polling approach.

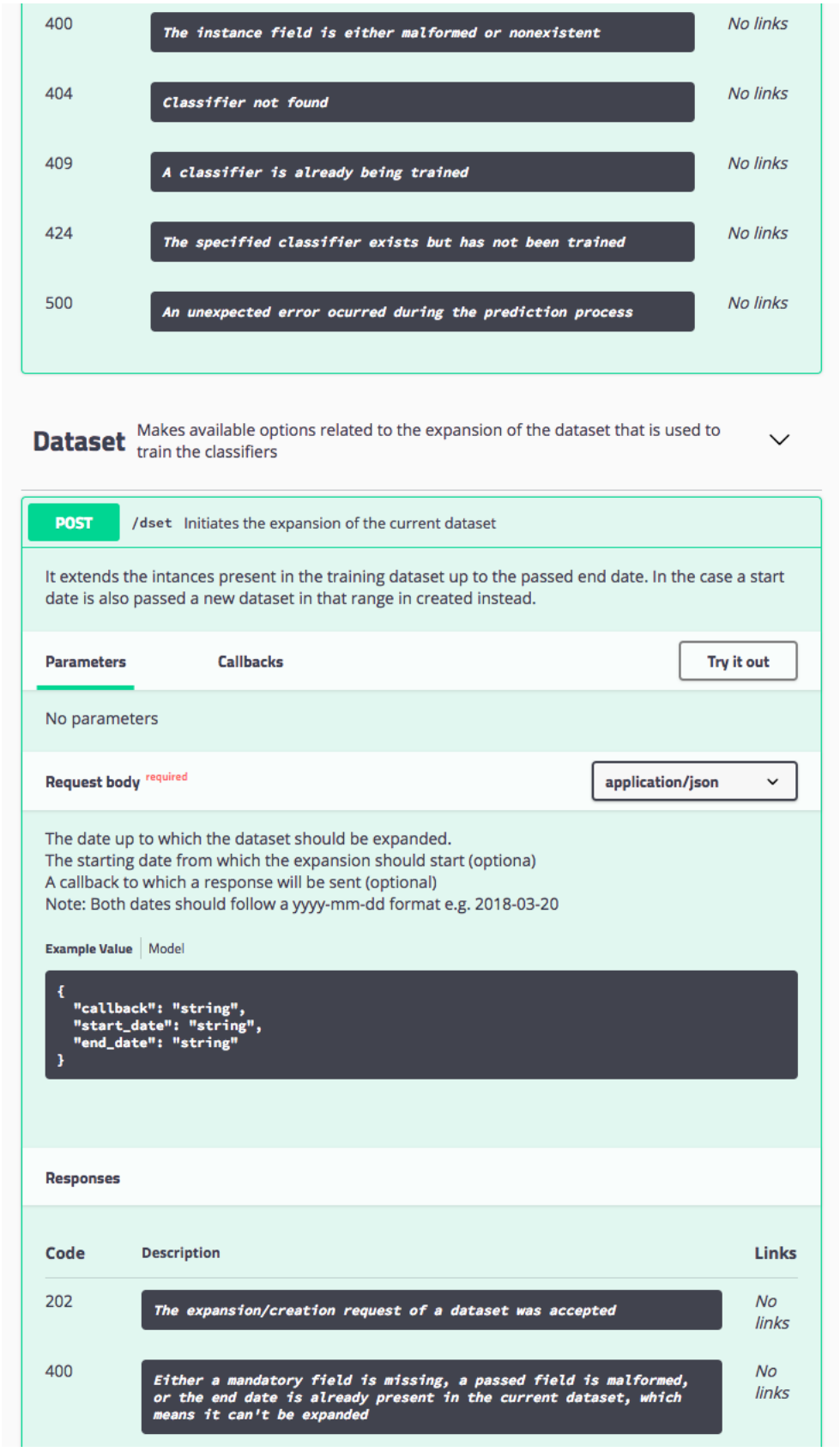
Parameters Try it out

Name	Description
classifier * required string *(path)*	The name of the classifier which was trained

Responses

Code	Description	Links
200	*Successfully retrieved training result* application/json Controls Accept header. Example Value | Model { "one_of": { "success_message": { "classifier": "string", "detail": "string" }, "error_message": { "title": "string", "description": "string", "link": { "text": "string", "href": "string", "rel": "string" } } } }	*No links*
404	*Training result not found / Classifier not found*	*No links*


Predict
Makes available options related to instance prediction by a classifier
POST
/ja/{classifier}/predict
Triggers a prediction by a specific classifier
The specified classifier tries to predict the outcome for the passed instance.
Parameters
Try it out
Name
Description
classifier * required
string
(path)
The name of the classifier used to get a prediction
Request body required
application/json
The instance to be predicted by the classifier.
Note: the instance array should be composed of strings and/or numbers (floats)
Example Value
Model
{
"instance": [
"xxxxx",
0.15163,
0,
0,
1624,
-27.349479779,
3,
1319,
4,
28,
16,
"espanha",
"",
1
]
}
Responses
Code
Description
Links
200
Successfully predicted the passed instance's outcome
No links
application/json
Controls Accept header.
Example Value
Model
{
"outcome": true,
"probability": {
"true": 0,
"false": 0
}
}



400	*The instance field is either malformed or nonexistent*	*No links*
404	*Classifier not found*	*No links*
409	*A classifier is already being trained*	*No links*
424	*The specified classifier exists but has not been trained*	*No links*
500	*An unexpected error ocurred during the prediction process*	*No links*

Dataset Makes available options related to the expansion of the dataset that is used to train the classifiers

POST `/dset` Initiates the expansion of the current dataset

It extends the intances present in the training dataset up to the passed end date. In the case a start date is also passed a new dataset in that range in created instead.

Parameters **Callbacks** **Try it out**

No parameters

Request body required **application/json**

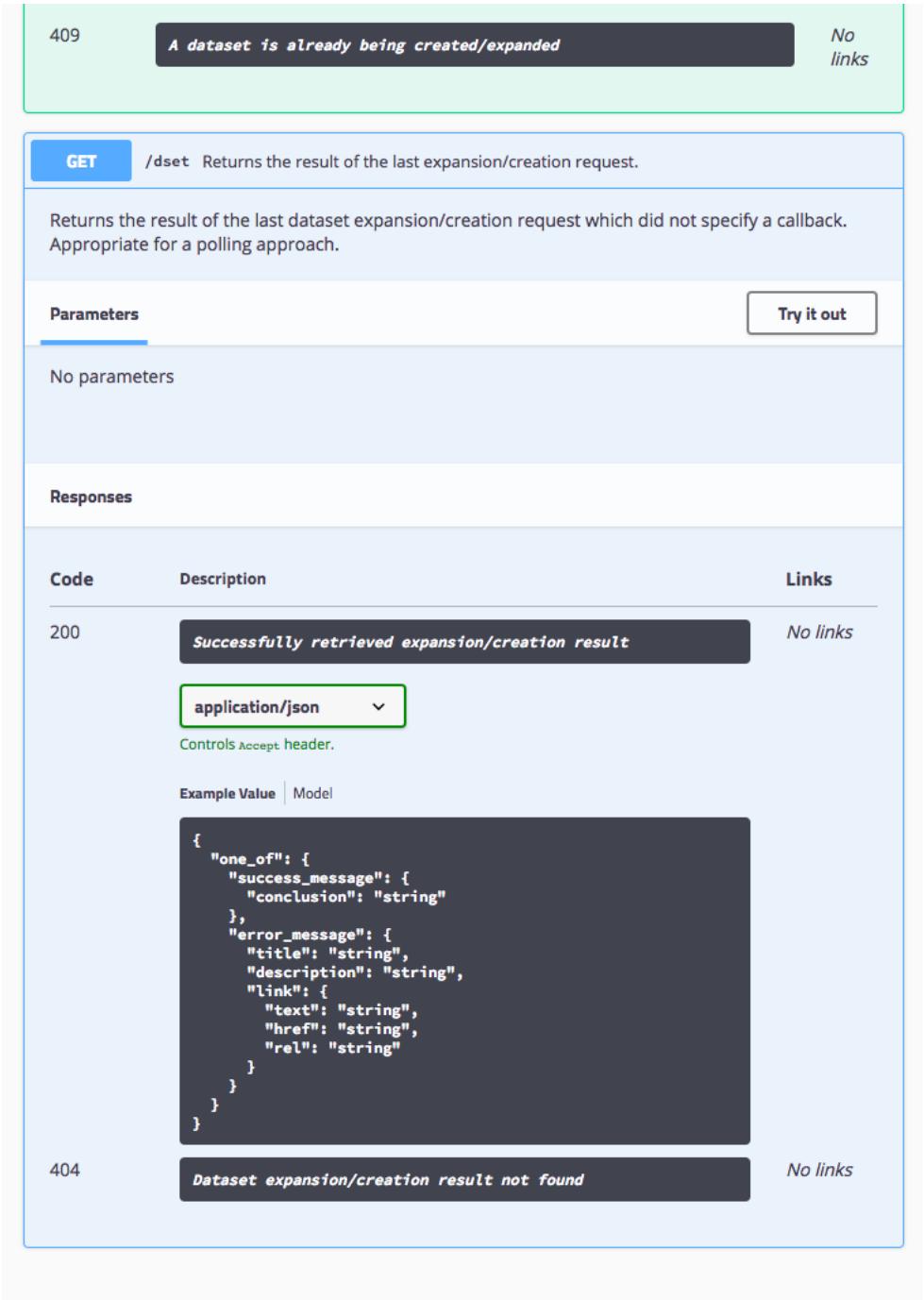
The date up to which the dataset should be expanded.
The starting date from which the expansion should start (optiona)
A callback to which a response will be sent (optional)
Note: Both dates should follow a yyyy-mm-dd format e.g. 2018-03-20

Example Value | Model

```
{
  "callback": "string",
  "start_date": "string",
  "end_date": "string"
}
```

Responses

Code	Description	Links
202	*The expansion/creation request of a dataset was accepted*	*No links*
400	*Either a mandatory field is missing, a passed field is malformed, or the end date is already present in the current dataset, which means it can't be expanded*	*No links*



409 — *A dataset is already being created/expanded* — *No links*

GET `/dset` Returns the result of the last expansion/creation request.

Returns the result of the last dataset expansion/creation request which did not specify a callback. Appropriate for a polling approach.

Parameters — **Try it out**

No parameters

Responses

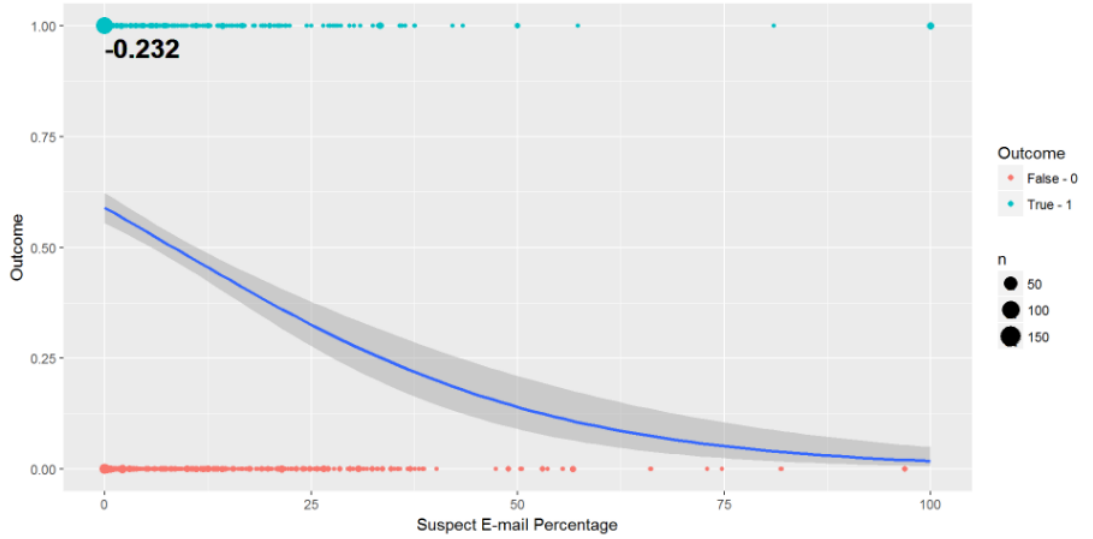
Code	Description	Links
200	*Successfully retrieved expansion/creation result* application/json Controls Accept header. Example Value | Model	*No links*
404	*Dataset expansion/creation result not found*	*No links*

```
{
  "one_of": {
    "success_message": {
      "conclusion": "string"
    },
    "error_message": {
      "title": "string",
      "description": "string",
      "link": {
        "text": "string",
        "href": "string",
        "rel": "string"
      }
    }
  }
}
```

Appendix B

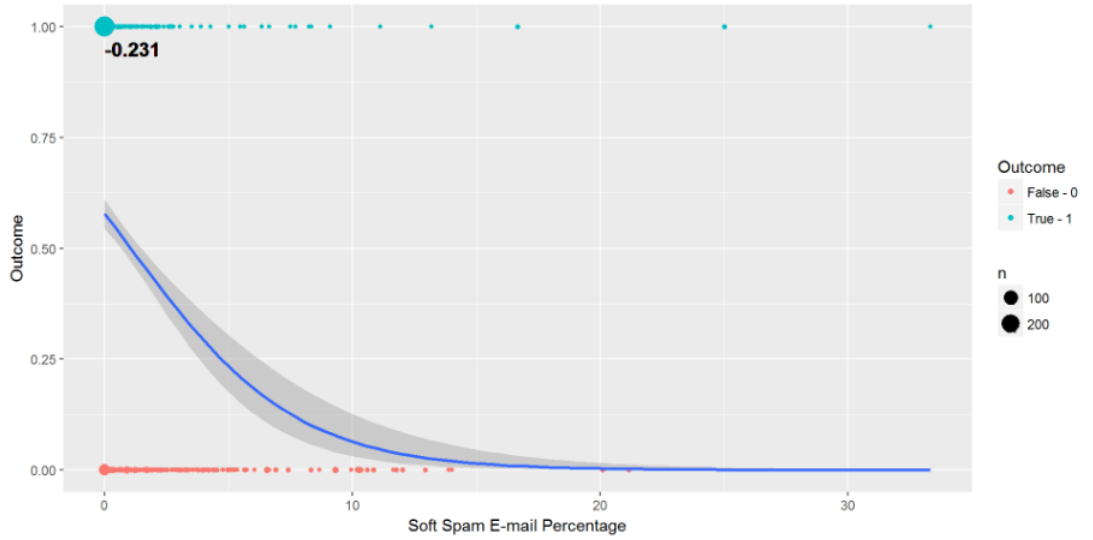
Sensibility Analysis Results

Suspect E-mail



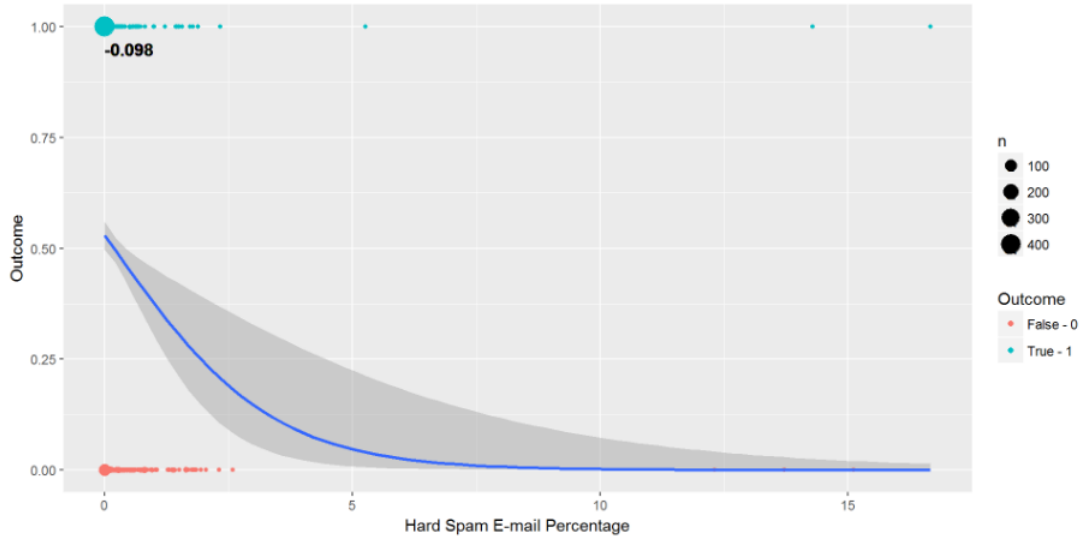

```
## Point-Biserial Correlation Estimate = -0.2321122, p-value = 1.053177e-16
```

Soft Spam E-mail



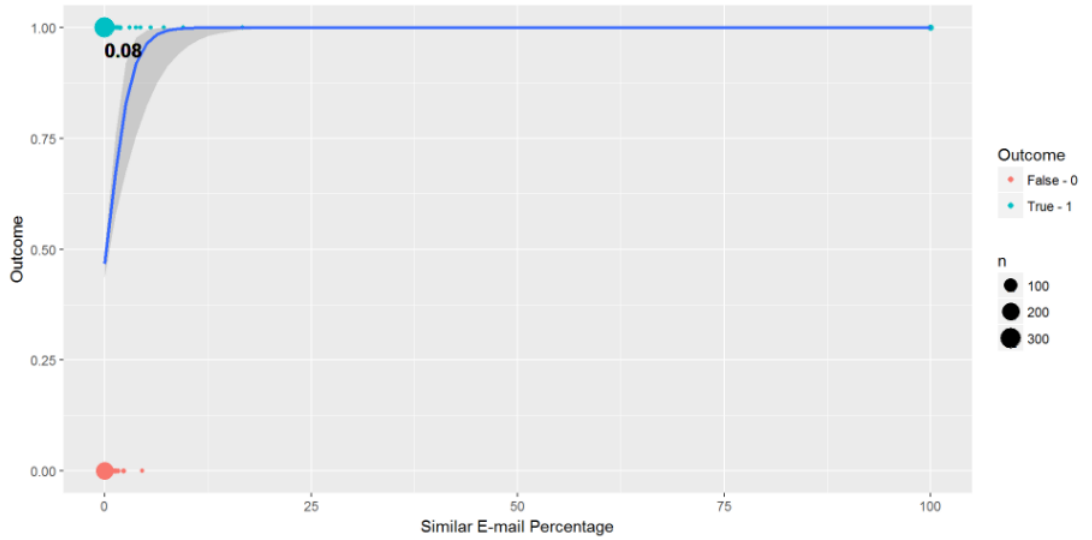

```
## Point-Biserial Correlation Estimate = -0.2311621, p-value = 1.412137e-16
```

Hard Spam E-mail



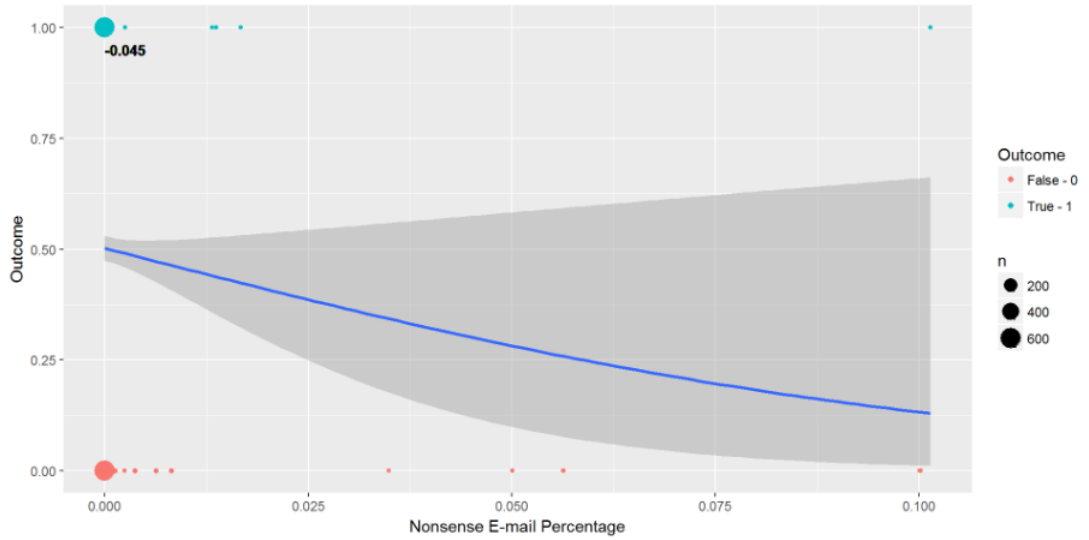

```
## Point-Biserial Correlation Estimate = -0.5164045, p-value = 7.168503e-86
```

Similar E-mail



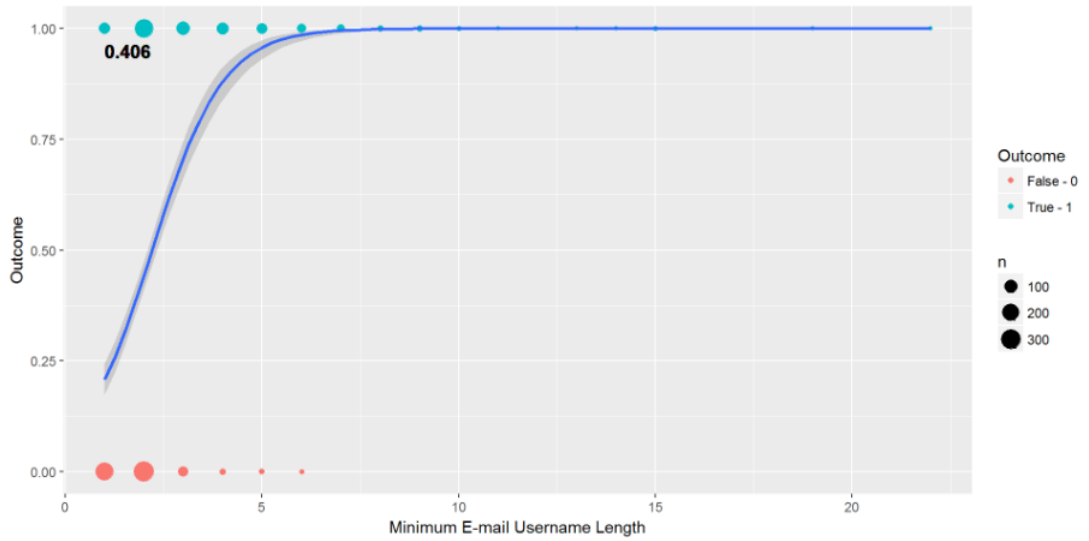

```
## Point-Biserial Correlation Estimate = 0.07967005, p-value = 0.004894409
```

Nonsense E-mail



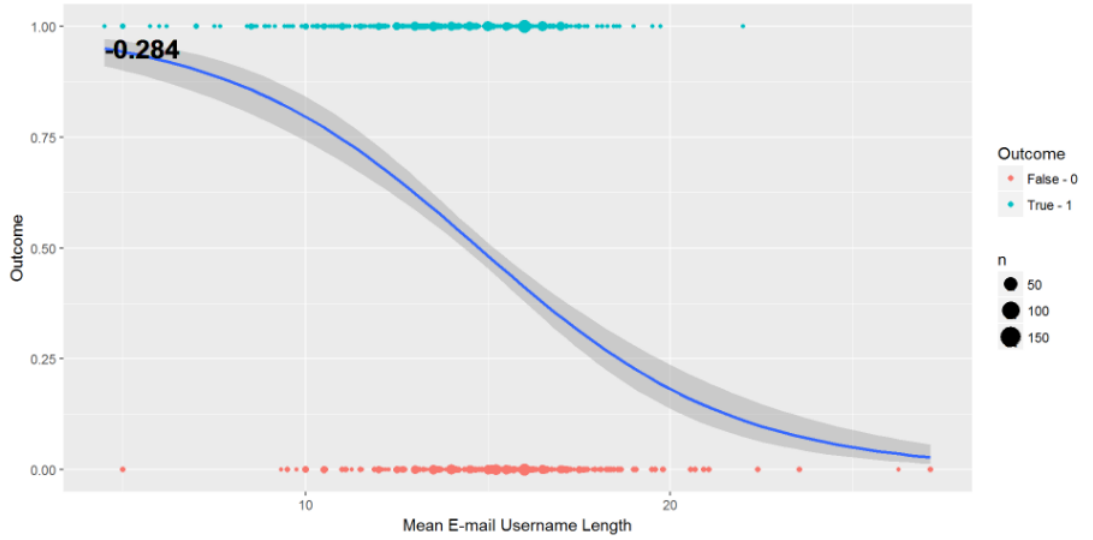

```
## Point-Biserial Correlation Estimate = -0.07295711, p-value = 0.009990963
```

E-mail Username Min Length



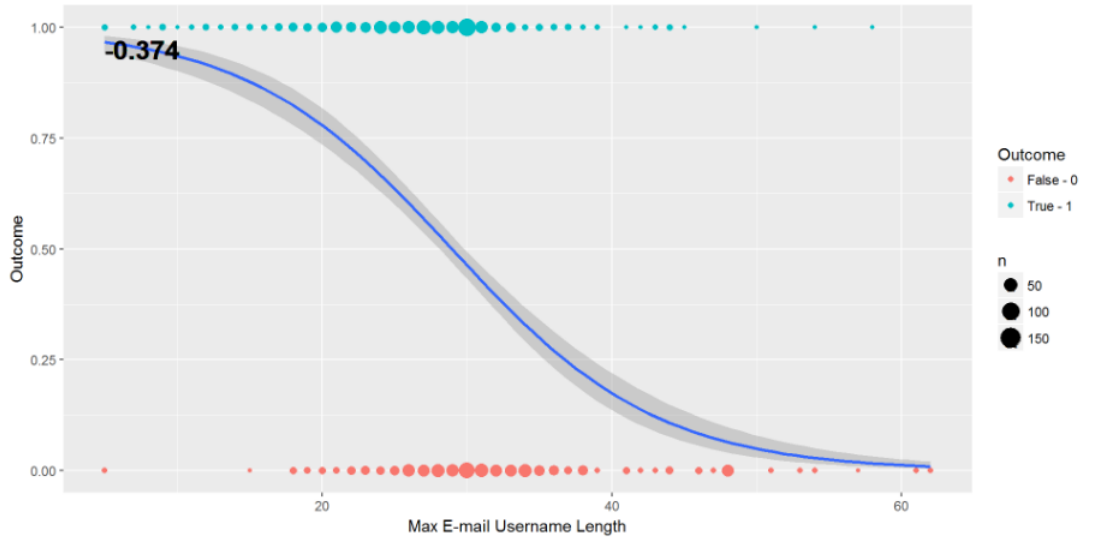

```
## Point-Biserial Correlation Estimate = 0.4055358, p-value = 1.616311e-50
```

E-mail Username Mean Length



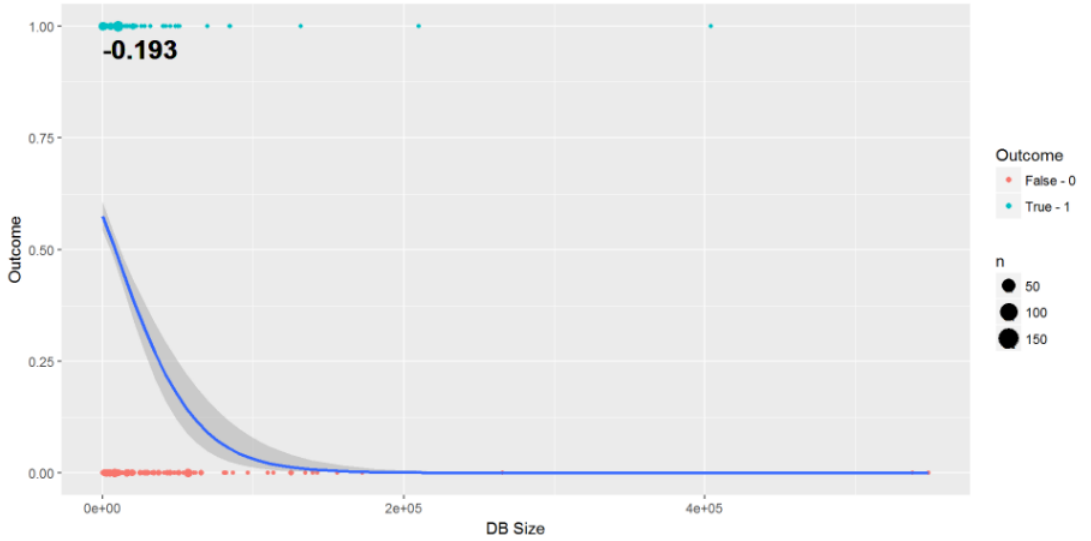

```
## Point-Biserial Correlation Estimate = -0.283509, p-value = 1.833105e-24
```

E-mail Username Max Length



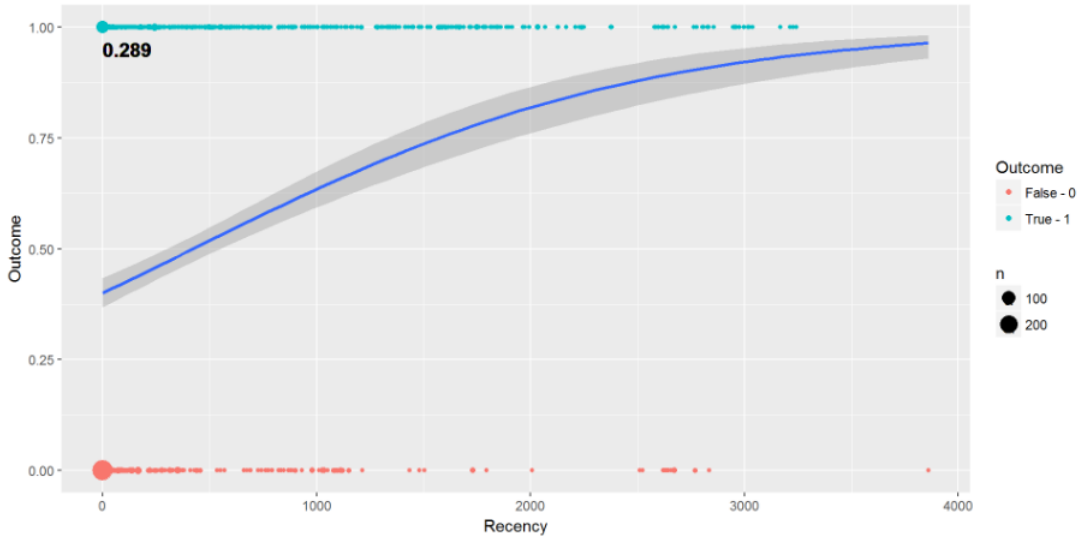

```
## Point-Biserial Correlation Estimate = -0.3744368, p-value = 9.402568e-43
```

DB Size



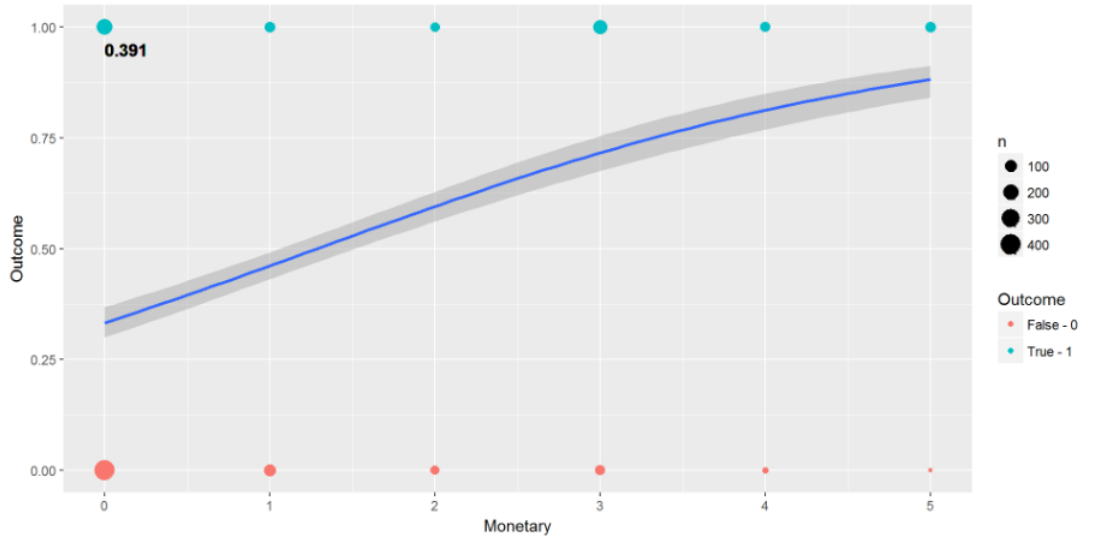

```
## Point-Biserial Correlation Estimate = -0.4975835, p-value = 6.479072e-79
```

Recency



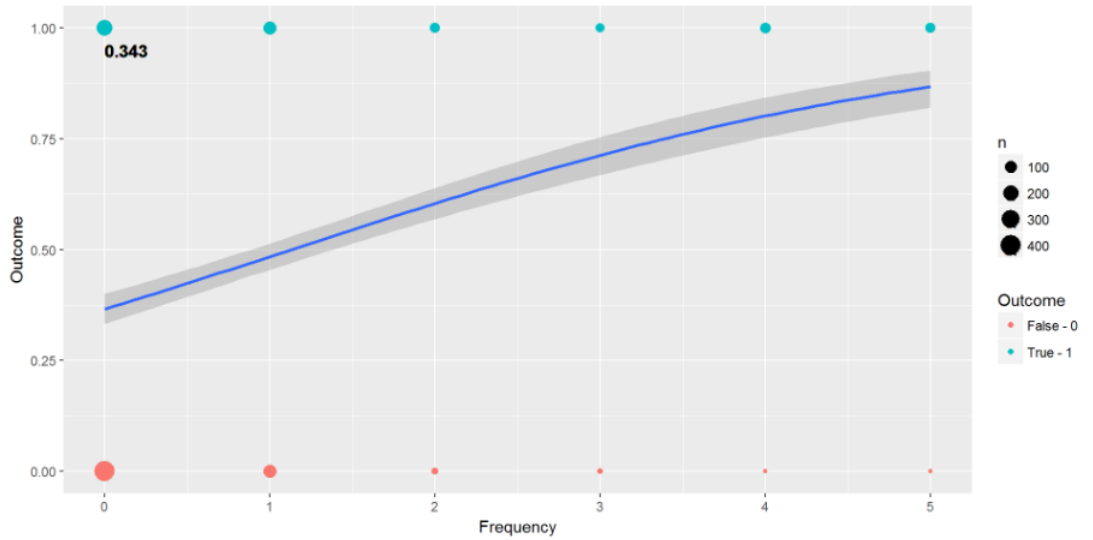

```
## Point-Biserial Correlation Estimate = 0.2892801, p-value = 1.913621e-25
```

Monetary



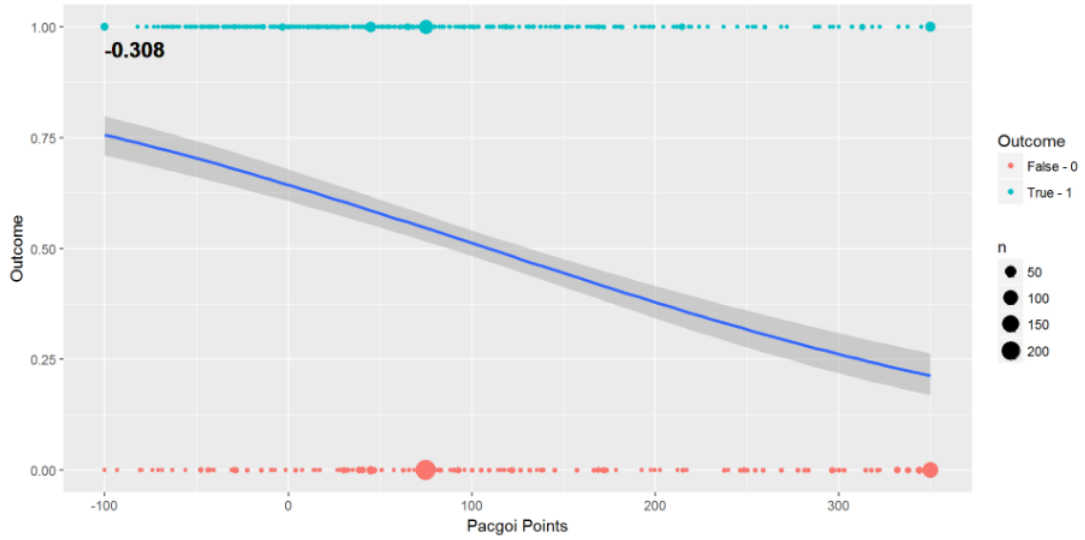

```
## Point-Biserial Correlation Estimate = 0.3911549, p-value = 7.997776e-47
```

Frequency



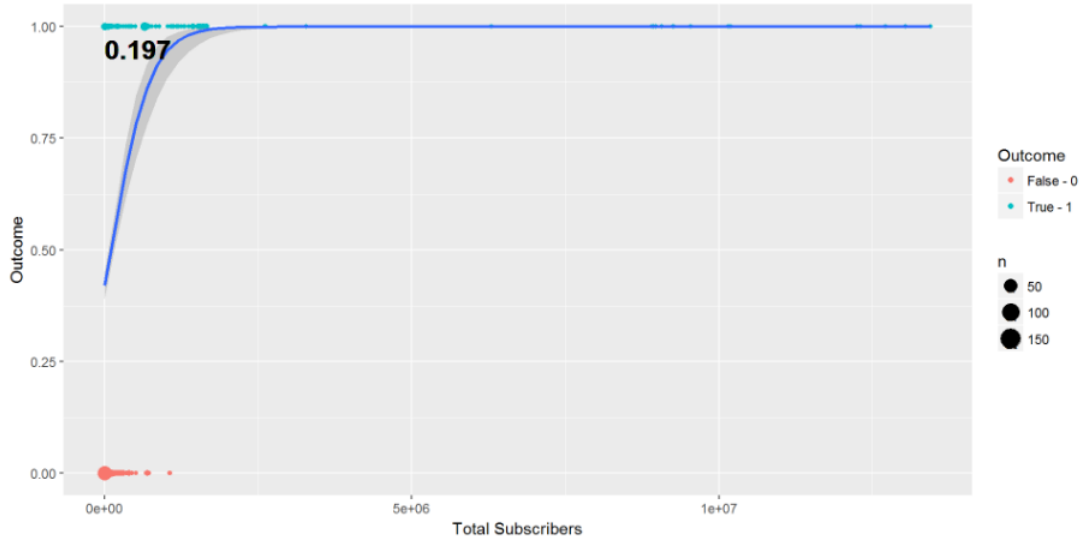

```
## Point-Biserial Correlation Estimate = 0.3428554, p-value = 1.086552e-35
```

Pacgoi Points



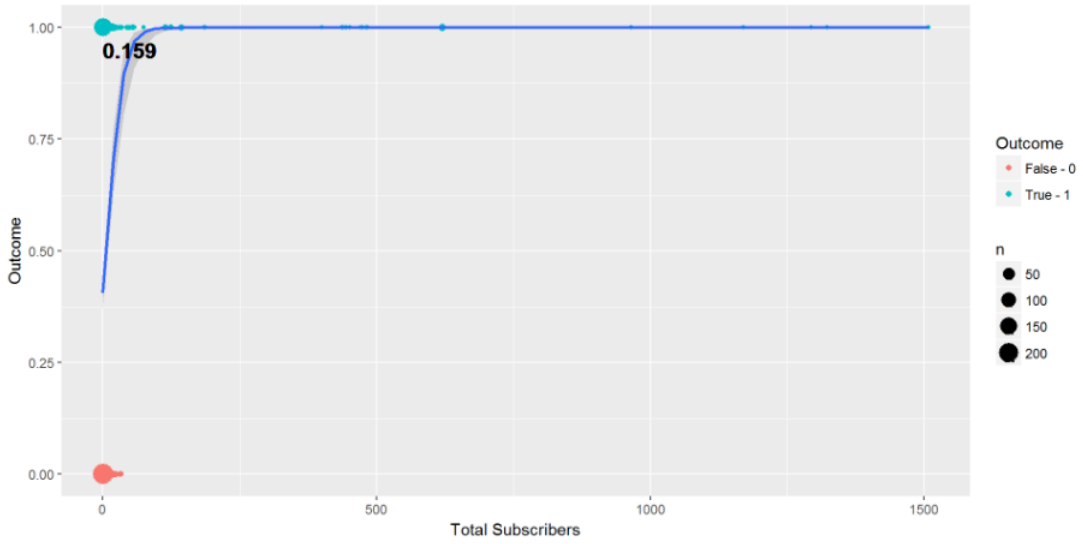

```
## Point-Biserial Correlation Estimate = -0.3078303, p-value = 9.333727e-29
```

Total Subscribers



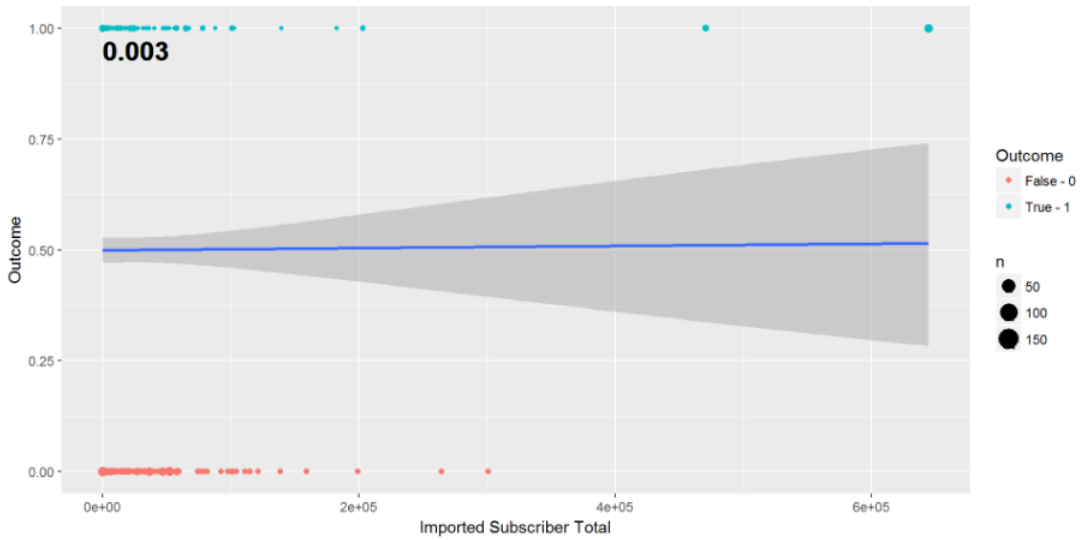

```
## Point-Biserial Correlation Estimate = 0.1973143, p-value = 2.108006e-12
```

Importation Total



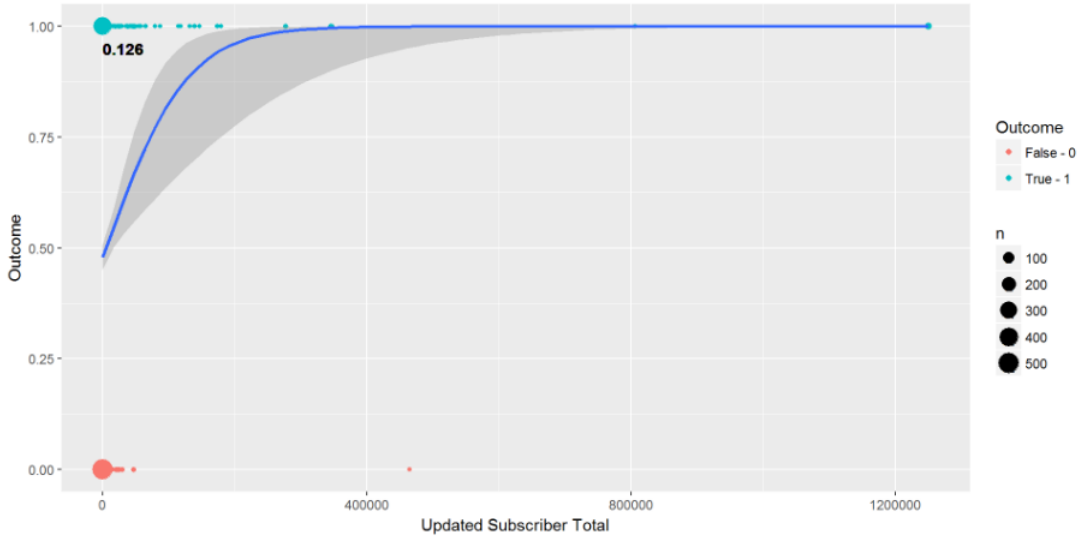

```
## Point-Biserial Correlation Estimate = 0.1594884, p-value = 1.510591e-08
```

Imported Subscribers Total



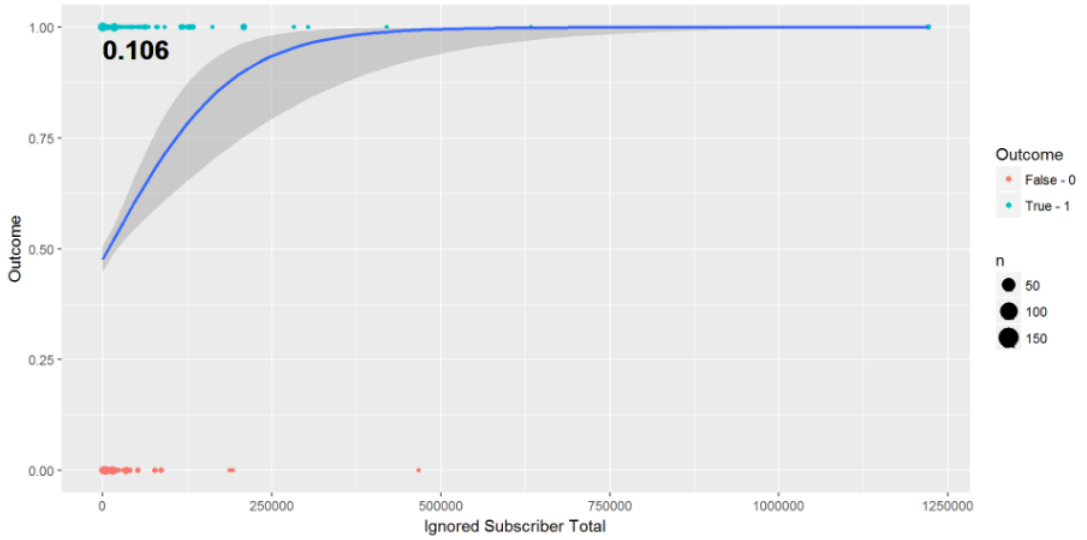

```
## Point-Biserial Correlation Estimate = 0.003451743, p-value = 0.9031206
```

Updated Subscribers Total



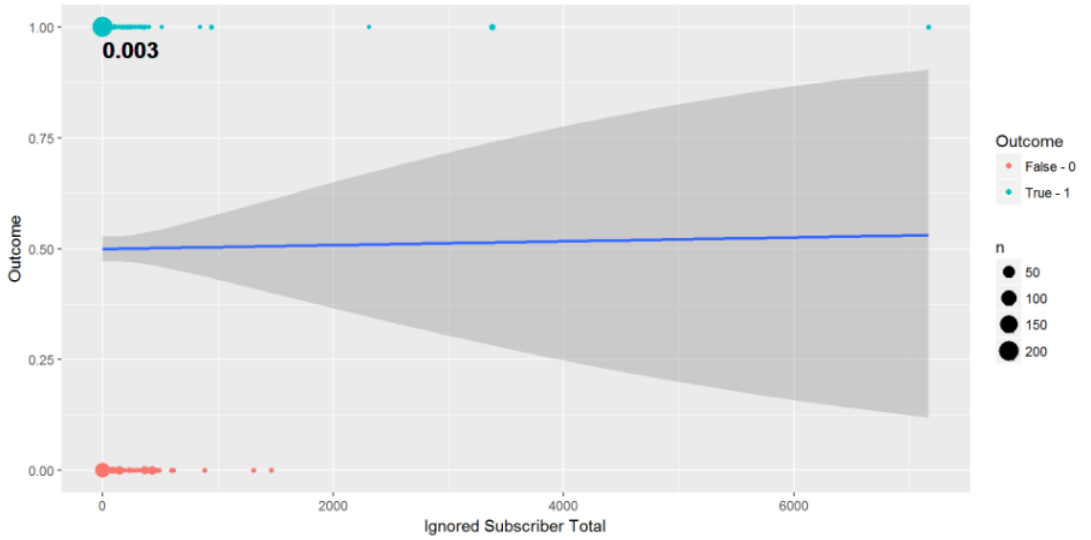

```
## Point-Biserial Correlation Estimate = 0.1258657, p-value = 8.34363e-06
```

Ignored Subscriber Total




```
## Point-Biserial Correlation Estimate = 0.1062147, p-value = 0.0001725754
```

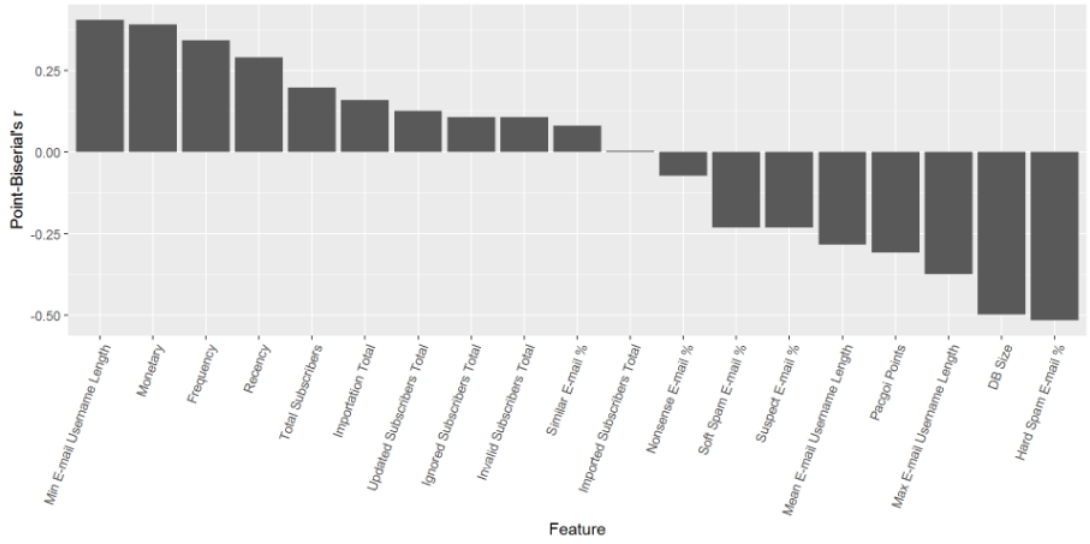
Invalid Subscriber Total



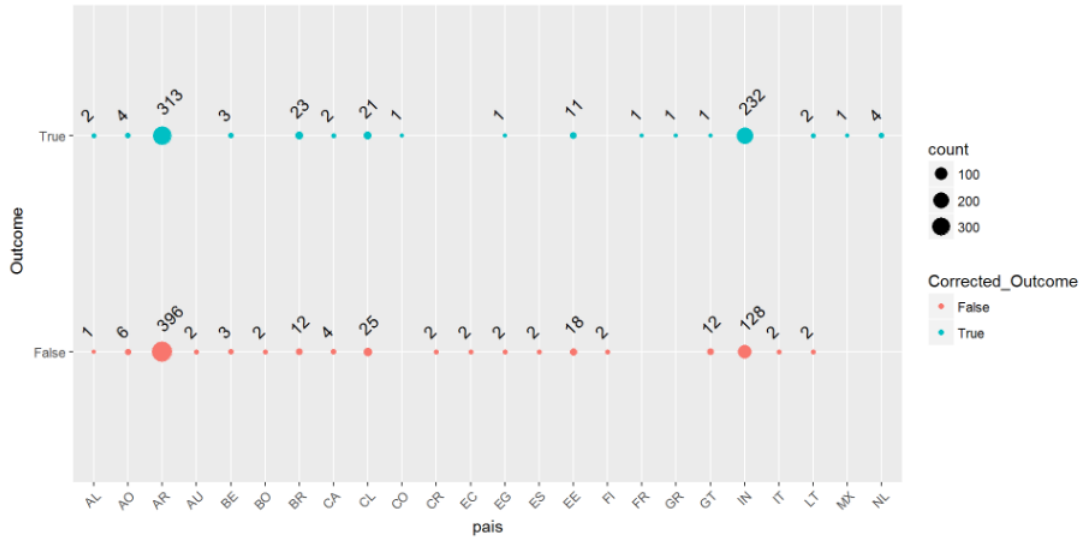

```
## Point-Biserial Correlation Estimate = 0.1062147, p-value = 0.0001725754
```

Feature	Point-Biserial's r	Point-Biserial's p-value
Suspect E-mail %	-0.2321122	0.0000000
Soft Spam E-mail %	-0.2311621	0.0000000
Hard Spam E-mail %	-0.5164045	0.0000000
Similar E-mail %	0.0796700	0.0048944
Nonsense E-mail %	-0.0729571	0.0099910
Min E-mail Username Length	0.4055358	0.0000000
Mean E-mail Username Length	-0.2835090	0.0000000
Max E-mail Username Length	-0.3744368	0.0000000
DB Size	-0.4975835	0.0000000
Recency	0.2892801	0.0000000
Monetary	0.3911549	0.0000000
Frequency	0.3428554	0.0000000
Pacgoi Points	-0.3078303	0.0000000
Total Subscribers	0.1973143	0.0000000
Importation Total	0.1594884	0.0000000
Imported Subscribers Total	0.0034517	0.0094961
Updated Subscribers Total	0.1258657	0.0000083
Ignored Subscribers Total	0.1062147	0.0001726
Invalid Subscribers Total	0.1062147	0.0001726

Correlation Summary



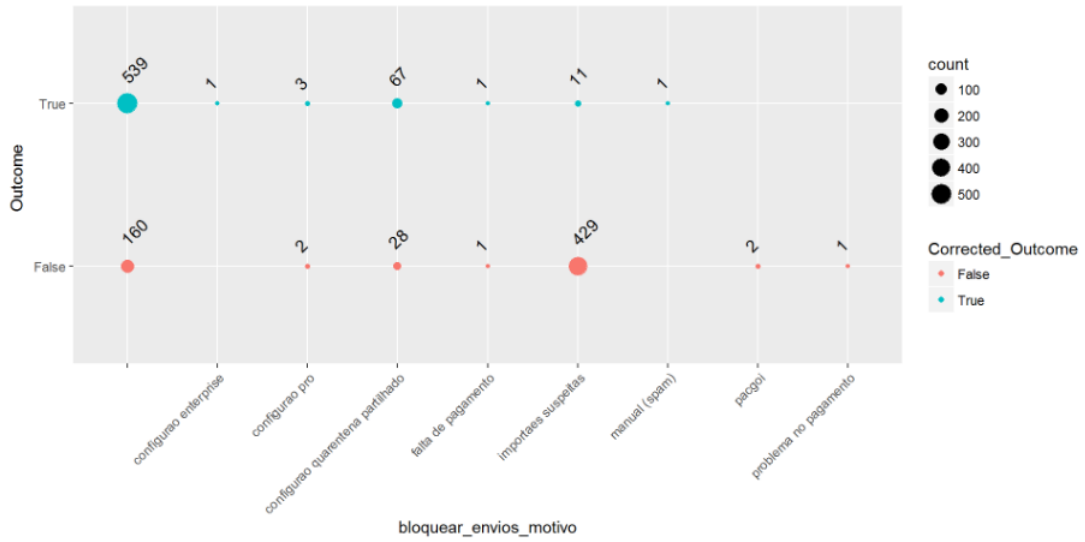
Country




```
##
##  Pearson's Chi-squared test with simulated p-value (based on 2000
##  replicates)
##
## data:  bds$pais and bds$Corrected_Outcome
## X-squared = 78.297, df = NA, p-value = 0.0004998
```

```
## CramÃ©r V / Phi: 0.2506758
```

Send Block




```
##
##  Pearson's Chi-squared test with simulated p-value (based on 2000
##  replicates)
##
## data:  bds$bloquear_envios_motivo and bds$Corrected_Outcome
## X-squared = 623.81, df = NA, p-value = 0.0004998
```

```
## CramÃ©r V / Phi: 0.7075638
```