ISCTE IUL
Instituto Universitário de Lisboa

Department of Information Science and Technology

Automated CV Screening

Mário Rivotti Hauptfleisch

Dissertation submitted as partial fulfillment of the requirements for the degree of

Master in Telecommunications and Computer Engineering

Advisor:
Doctor Luís Miguel Martins Nunes
ISCTE

Co-Advisor:
Doctor Sérgio Miguel Carneiro Moro
ISCTE

December 5, 2018

Acknowledgments

I would first like to thank ISCTE and Siemens for this partnership and accepting me to make this dissertation.

At Siemens, i would like to thank Miguel Batista for the help and ideas provided through the whole making of this dissertation. I would also like to thank in the HR team, Isabel Marta, that solved all the doubts and questions that i had related to the database and the job application process. Still at Siemens, i would to thank all the hiring managers, for their time grading the CVs that made the data used in the final experiments.

At ISCTE, i would first like to thank my teachers and advisors, Luís Nunes and Sérgio Moro for the amazing support, motivation and guidance given through this study, without your teachings this would have been harder. I would also like to thank prof. Fernando Batista and Ana Almeida, for your comments reviewing this document.

To my dear friends, a big thank you for all the moments shared, for all the lessons that you taught me, all the smiles and laughs that have kept me motivated through this journey, it has truly been a wonderful experience.

At last but not least, i would to give a warm thank you to my mother for, well, everything.

To all of you, thank you.

Abstract

In this dissertation was predicted the outcome of job applications using machine learning. The focus of the outcome is on the first stage of the recruitment process, where there is a very large number of applications. The data used to train classifiers and predict targets was provided by a company's human resources tool, where they receive structured online applications with their corresponding CVs. The applications are for different job positions in the same company.

The first part the experiments is the preprocessing of the data. Here the reader can find a description of the variables and how they are processed with preprocessing techniques, to be used by the machine learning algorithms. In the second part you will find the results for the predictions made by the different algorithms. The ones that were used in this thesis are: Decision Tree, Random Forest, Gradient Boosted Tree, SVM and Artificial Neural Networks, all form the python's sklearn package.

The outcomes that are predicted in this work are if the candidate passed the first stage of the screening process, if the candidate failed the overall process, if the candidate was hired, and the grades given to the applications labeled by the hiring managers.

Results showed that by using the variable 'Job ID', that describes the job that each candidate is applying, improved the predictions significantly. Without using the 'Job ID' the best accuracies achieved were around the 75%. Using the 'Job ID', the best accuracies were around 90%. Overall the Random Forest and Gradient Boosted Tree had the best results. The attributes that contributed the most to predict the different targets were the specification and area of study, the highest education achieved, the number of languages spoken, and the distance from home to work.

Resumo

Nesta dissertaçao é feita uma previsão do resultado de candidaturas de emprego usando algoritmos de machine learning. O foco das previsões é o resultado das candidaturas na primeira fase do processo de aprovação, onde há um número elevado de candidaturas. Os dados usados para treinar os modelos são provenientes de uma ferramenta de recursos humanos, onde as candidaturas estão organiadas numa forma estruturada com os respectivos CVs associados. Estas candidaturas são para diferentes posições dentro da mesma empresa. A primeira parte das experiências corresponde ao pré-processamento dos dados. Aqui o leitor pode encontrar uma descrição das variáveis e a forma como são processadas para serem usadas pelos algoritmos de machine learning. Os algoritmos usados nesta dissertação são os seguintes: árvore de decisão, random forest, gradient boosted tree, SVM e redes neuronais. As implementações dos algoritmos, são todas provenientes da biblioteca de python sklearn. As previsões que são feitas neste trabalho, correspondem ao desfecho da primeira fase, ao desfecho final, se o candidato falha em qualquer uma das fases e também à nota da candidatura que foi dada pelos recrutadores da empresa. Os resultados das experiências mostram que a variável "Job ID", que corresponde a uma dada posição da empresa, melhoram significativamente as previsões feitas pelos algoritmos. Sem usar o "Job ID", a percentagem de precisão ronda os 75%, ao usar a variável a percentagem ronda os 90%. Os algoritmos que obtiveram os melhores resultado ao longo da dissertação, foram o random forest e a gradient boosted tree. Os atributos que tiveram o maior impacto na previsões, foram a especificação e área de estudo, o maior grau de ensino obtido pelo candidato, o número de línguas faladas e as distância da casa ao trabalho.

Contents

List of Figures

List of Tables

List of Acronyms

- CART - Classification and Regression Trees
- CHAID - Chi-squared Automatic Interaction Detection
- FN - False negatives
- FP - False positives
- FPR - False positive rate
- HM - Hiring Manager
- HR - Human Resources
- kNN - K-Nearest Neighbors
- LDA - Latent Dirichlet Allocation
- LR - Linear Regression
- MLP - Multi Layer Processing
- NN - Neural Networks
- RegTree - Regression Tree
- ROC - Receiver operating characteristic
- SVM - Singular Vector Machine
- TN - True negatives
- TP - True positives
- TPR - True positive rate

1 Introduction

1.1 Framework and motivation of the theme

Big companies receive huge amounts of job applications, for a wide range of positions, and therefore they have to analyze them in a short time. The main goal of this work is to predict the outcome of new job applications, finding that it is possible to automatically screen candidates would reduce the work load of Human Resources and consequently, reduce the costs of the company.

Any company is the result of what people working there achieve to produce. Therefore, attracting the best human resources (HR) is imperative for organizations to thrive and excel in the markets where they compete [14]. Furthermore, globalization led by innovative transportation and information systems' technologies made the world a smaller place where people move freely worldwide in search for the best job, i.e., the one that best fulfills each one's needs and motivations [15].

"Research shows that when high performance HR practices, including recruiting and selection systems, are designed to be aligned with corporate goals, and employees who can help achieve these goals are hired, the company's financial performance can be positively influenced" [13].

Most of the hiring managers of this company are team leaders, and it's often that they don't have the time to analyze all the curricula for their available positions. It is in the first stage of the screening process when help is needed, where the hiring managers have a lot of options, and the task to compare the candidates and select the best is more time consuming. The automated curriculum screening concept has emerged to help in this task by relieving HR staff of the burden of selecting the most interesting curricula [13].

The job applications were submitted on a HR online tool for this company. To apply for a job, the candidates must fill an online form, where they submit their applications to the HR tool. The applications are structured in a single table, where each row is a job application submitted by a candidate and each column is an attribute of the job application or the candidate. The candidates can also upload their CVs that will be attached to the job application. All the experiments in this work are made using this data.

When a new application arrives at the system, the hiring manager will evaluate it, only by reading the candidate's filled data and the corresponding CV, and decide if the application goes through to next phase, this step of the process is called "HM Review". If the candidate passes this first phase, it will be interviewed in person, or by a phone/video call. After this phase, the hiring managers decide who to to hire.

Job classification and further screening will be made using several machine learning techniques. The techniques will be evaluated to test if it's possible to imitate the recruitment process using the company's data. The algorithms that will be tested in this work are Decision Tree, Random Forest, Gradient Boosted Tree, SVM, Artificial Neural Networks, all from the python's scikit-learn package [29]. Before the algorithms are tested a preprocessing of the data will be made, where the reader can also find a description of the candidates' attributes.

1.2 Research Issues and Goals

The main goal is to build a model that replicates the decisions and recruitment process made by the HR department. Not only to classify the final outcome of the recruitment process, but also to determine outcomes of several recruitment phases.

Another goal is to determine if there is any correlation not only for the winning candidates, but also for the losing candidates.

Proving that there are fields more relevant than others, would determine that the requested information may need to be revised.

Here are some of the research issues that this work

- Is it possible to replicate HR decisions using data from the recruitment process?
- Is there any correlation on the forms filled by winning job applicants?
- Are there fields on the provided data more relevant than others? If so, are there fields that are absolutely irrelevant, and others that are crucial to the application decision?

1.3 Research Methods

1: Objectives definition – The first step is to define a clear goal of the project,

2: Project development – After the goals are defined, the project development starts by collecting and understanding of all the data available, plan what algorithms to be used that have proven the best results in the past with similar problems.

3: Project implementation – Start coding the preprocessing and the model that will be used to predict different results.

4: Test the project – After implementing a model or system, it will be tested with candidates that have ended their application, so that we can compare the results obtained with what happened in real life.

5: Repeat 3 and 4 Study different attributes and features and how they impact the screening of a job a applicant.

6: Evaluation – Test the model in real life to see if applicants that were hired actually had good performances in the company.

1.4 Dissertation's structure and organization

Section 2 refers to the literature review, it's a summary of similar work made previously.

Section 3 is where a brief description of the machine learning algorithms used in this work.

In section 4, is where the preprocessing of the data is described. In this section the reader will find a description of the candidates' attributes, decisions taken to replace, change and create new features.

In section 5, the reader will find the experiments where the algorithms are tested to predict 4 different targets.

Section 6 has the main conclusions of this dissertation, contributions for the scientific and corporate communities, and the study limitations.

2 Literature Review

In this chapter we present a few ideas and methods that were used in similar problems, such as decisions that were made to make the preprocessing of the dataset, algorithms that are used to classify candidates and methods that can be applied to try to simulate the recruitment process.

In [10] the authors used datasets made from information gathered for two years from an HR department and has a total of 3578 records, in which 5% were hired and 95% of them were rejected. The goal was to try to predict if a candidate would be hired or not. If all the records in the dataset were used to make a prediction model, all algorithms would be very good at classifying the candidates. This happens because if it classifies all of them as rejected, the model would have 95% of accuracy and might be disguised as a good classifier. To avoid this problem, a new dataset was generated from the original with an equal number of rejected and not rejected candidates.

Two clustering algorithms, Fuzzy C-means Clustering and K-means Clustering, and also three trees, ID3, C4.5 and CART were used to classify the candidates. The results were measured using the accuracy of each algorithm, number of correct predictions divided by the total number of predictions. The dataset was divided in two, one for each year of the collected data and two experiments were made, in each case, one of the datasets was used as the training data, and the other as the test data. In table 1 are the results for the Fuzzy C-means and K -means, and in table 2 the results for the decision tree algorithms.

Algorithm used	**% of accuracy 1**	**% of accuracy 2**
Fuzzy C-means	52.10%	63.13%
K - means	53.54%	69.62%

Table 1: Results obtained in [10] in using clustering techniques

The authors have stated that clustering techniques have poor accuracy and aren't suitable for this problem domain due to the nature of the data."

Algorithm used	**% of accuracy 1**	**% of accuracy 2**
Id3	45.12%	50.27%
C4.5	77.29%	79.12%
C4.5 Unpruned	76.73%	78.73%
Cart	72.12%	77.29%
Cart Unpruned	72.75%	76.57%

Table 2: Results obtain on [10] in two datasets using decision trees

The authors have also used the C4.5 unpruned tree to show a few inferred rules extracted from the tree.

Rules Inferred
If MarksInProgramming $>$ 5 AND PercentageInHigher-SecSchool $>$ 90 AND MarksInTechnical $>$ 35 AND PercentageInBE $>$ 70 then Selected
If College = X OR College = Y AND PercentageInBE $>$ 70 AND PercentageInBE $<$ 80 AND PercentageIn-HigherSecSchool $>$ 90 then Selected

Table 3: Some inferred rules extracted from the C4.5 unpruned tree in [10]

To conclude, all trees achieved similar results, but the tree that had the best results was the C4.5 pruned. It can be observed that the unpruned trees had worst results than the pruned trees of the same type. The best result was only 79.12% or accuracy. We feel that this result could be improved by using ensemble algorithms (forests) or boosted trees.

In [11] a "framework for human resource data mining to explore the relationships between personnel profiles and work behaviors" has been made. Here the goal is not to classify the candidates as opposed to the case in study and the research made in [10], the goal is to try to relate job performances with employees attributes. Job performances are ranked in three levels: "'outstanding' (top 10%), 'successful' (85%) and 'bottom' (5%)". These were given as an annual evaluation from the company. Even though the goal is not exactly the same, it's quite similar because the candidates that are applying for a job also have a set of attributes, and the performance that will be predicted could be related to the success of a candidate's screening process.

The employees attributes are listed below:

Age, Gender, Marital Status,

Work Experience: "yes" if it has one year of work experience.

Degree: 4 categories: high school level and below, junior college degree, bachelor degree, and master and above

Major subjects: "There were 52 different major subjects, however, only eleven majors have more than 50 samples and the other majors without sufficient samples are grouped into one category denoted as "others"".

School/School tiers: "Originally, there were 114 different universities in the data. However, this variable was transformed into four categories including three tiers of Taiwanese universities and fourth denoting the others."

Recruitment channel: The recruitment channels include the internal channel and the external channel.

The last five attributes can also be found in the dataset that will be studied, and similar preprocessing decisions described above could be tested. These were the ones used as inputs to make the decision trees analysis since the first three were removed "due to the concern of discrimination".

The authors "used CHAID as the data mining tool to explore the latent relation-

ships among the input employee profiles and target variables of work behaviors such as job performance, retention, and turnover reasons." Every possible tree structure was tested to find relationships between the attributes and job performances.

Confidence and lift were used to evaluate the model. "Confidence denotes the prediction accuracy that a subset can be categorized into a specific class. Lift is a ratio that is commonly used to assess the performance of the classification method."

Some of the rules derived from the tree are listed in table 4

Rules Inferred	**Lift**
IF recruit channel = external THEN he/she will perform with a level of improvement needed. (n = 95; confidence = 84%)	1.06
If College = X OR College = Y AND PercentageInBE > 70 AND PercentageInBE < 80 AND PercentageInHigherSecSchool > 90 then Selected	1.05

Table 4: Some rules inferred form the CHAID model used in [11]

Employees recruited from external channel, have proven a worst performance from internal channels because internals usually stay longer that externals. An internal with a masters degree is preferred than an external with a PhD.

As it was expected, candidates with a higher degree and graded from high tier colleges are better classified. Employees with more experience are also preferred.

The authors stated that neural networks should be studied in future research, "to compare various approaches and may thus integrate them for better exploration of complex interrelationships among the input personnel variables and target work behaviors."

In [4] the authors present an intelligent screening process that matches information present in *resumes* with some properties listed for a specific job position in order to rank candidates. As opposed to the study referenced above, here the goal is not to determine if a candidate will be hired or not, it's to rank candidates for each job title. Here the resumes are submitted in free form, and the first step that is presented is to define nine categories that represent the main features for each applicant:

Domain of jobs, Job title, Position, Knowledge, Experience, Location, Salary, Qualification level, Job choices.

Each feature does not have the same importance, so each one is weighed as a score according to a domain expert. The weighs are listed in table 5. After defining the weighs, the candidates go through a filtering process before being ranked, only the ones that that go through to the ranking process, are the ones that meet a minimum threshold [12]. This threshold is defined by number of strings matched in a resume with a set of strings defined for each job title. For example, a person that applies for an IT Programmer job, will be more likely to go through to ranking

process if it has the string "IT" in its resume.

Parameters	**Weight**
Job Title	21%
Location	21%
Qualification Level	16%
Job Type	16%
Salary	15%
Years of Experience	11%

Table 5: An example of some standard parameters and their respective weight Parameters Weightage in Percentage Level of Important [4]

Although the authors of this paper do not explore the exact same problem, there are some similarities. Since some of these categories can be found in our dataset, some of the techniques used in this paper can be explored.

This filtering process could also be used this study to try to remove outliers from the algorithms. Also new features can be created based on matching strings from some fields of the dataset, with some predefined qualities defined for each job title. For example, 'Worked at Google as SysAdmin', 'Studied Engineering at MIT' could be defined as very good qualities for a certain job, and if a candidate has a certain number of very good qualities, a new feature would be added to that candidate. These new features could also be weighed as stated above.

In [6] is presented a system to evaluate job applicants using machine learning algorithms to rank candidates. It extracts user information from online forms, and for some users it has the information of their LinkedIn profiles and blogs. The candidates features are represented in Booleans, where it describes if it has a certain skill or not, or numerical, for example, the years of experience. Personality mining is also used to improve the data, previous work has shown that it is possible to extract an authors' personality, mood and emotions out of blog posts [7, 8, 9]. The features and personality data are fed to machine learning techniques (LR, RegTree (M5, REP), SVR) to rank the candidates. The algorithms that presented the best results were the trees and SVR as opposed to the linear regression.

A study has been made to compare classification algorithms that predict students final exam grade [5]. The grades go from 1 (lowest) to 5 (highest) and the final classification of the algorithms in this experiment is in two different versions, the first grades the students in 2 classes, and the second in 3 classes, for example: 2 classes: 5, 4, 3 = 'Good' ; 2, 1 = 'Poor'; 3 classes: 5, 4 = 'Very Good'; 3='Standard'; 2, 1 = 'Poor'. The attributes to make the classification were collected by a learning management system (Moodle), 6 out of 10 are related to scores and the rest are related to number of logins and time spent in different parts of Moodle. 5 algorithms were used, LogReg, NN, RF, SVM, kNN. The performance of the prediction models depends on the attribute and class selections, but overall good results were achieved on these 5 algorithms. This study can be compared to problem in hand, if each attribute is translated into a score, for example, the university and previous job of the

applicants could be classified as a score. The final outcome of the job classification could be a ranking or a grade comparing to the study in [5]. The difficulty on the current problem relies on deciding what classification to attribute to each variable.

In [2] the authors apply data mining techniques on two datasets, the first one with Facebook profiles and the second one with CVs. The CV screening dataset was has 294 CVs and they were evaluated by 980 participants. The participants had to provide a report with an attribute specific and overall evaluation of the CVs they examined. Then data mining algorithms were used to find which factors were more important and the most impact on the ratings given, and construct the best predictive model that describes the data. A multiple linear regression model was used to determine the importance of each attribute.

Latent Dirichlet Allocation (LDA) was used to extract key features from the resumes. Topics such as creative, responsible, student and engineering were the results of using LDA on a set of CVs. This algorithm could be used in the fields of the dataset that are free text to extract new features for each candidate.

Results in this paper have indicated that the information on the resumes is not enough to screen candidates, people consider more other attributes like attitude and personality.

3 Machine Learning Algorithms

In this section, a brief description of the algorithms used in this work are presented below. The ones that are used are Decision Tree, Random Forest, SVM, Artificial Neural Network and Gradient Boosted Tree.

The Random Forest and the Gradient Boosted Tree were used in this work because these are the algorithms that have obtained better results in similar classification problems. Although the SVM, Artificial Neural Networks and Decision tree are likely to perform worse they will be tested, to prove if indeed they worse than the ones described before.

3.1 Decision Trees

A simple way to understand what a decision tree is, is to think of it as a structure of internal nodes, branches and leaves.

The nodes of the trees represent questions asked about the given data, e.g. "is the salary expected bigger than 1500?", or "does the candidate have a Ph.D?". Each node can only have one parent node, and typically each one has two or more branches. These branches represent the answers to the questions made from the nodes connected above, e.g. "yes" or "no". At the end of branch, there can be another node asking a new "question", or a leaf connected to that branch. The leaves are ate the end of the tree, they hold the information of the predicted result, or the classification of a given problem, e.g. "hired" or "not hired". An example of a simple decision tree is represented in fig 1. Questions are asked in a top-down fashion, they start at the root until they a reach the end, a leaf.

Trees are mainly used in classification problems and in general, "decision tree classifiers have good accuracy, but successful use of it depends on the data at hand" [10].

There are many types of decision trees, some examples are ID3, C4.5, CART, CHAID and MARS.

ID3 and the C4.5 were both idealized by Ross Quinlan [32], [33]. Id3 (Iterative Dichotomizer) was the first implementation. It makes the splits by calculating each attribute's entropy or information gain and the one that has the lowest entropy or the highest information gain is selected to split the data, then it generates the next level by repeating the same process with the remaining attributes. This is called a greedy approach because it selects the best attribute for the given moment and never tests it again. When the split made generates a node where the data is all from the same class, it is labeled as a leaf, with the respective classification. A disadvantage of this approach is that it can lead sometimes to overfitting.

The C4.5 is an improvement of the ID3, it handles both continuous and discrete data, the ID3 handles only discrete data. It also handles missing values. C4.5 has another significant improvement, it does pruning, that is, it removes parts of some of the branches and reduces the susceptibility to overfitting.

CART stands for classification and regression tree, and the main difference is that it can make regression trees, this means that it can predict a real value instead of a class [33].

Some advantages of using a decision tree is that the models generated can be

visualized, and therefore, they can be simple to understand and interpret [25]. The decisions created by the models are all represented in the generated tree. In a neural network, for example, the model created is more difficult to understand because there is no visual model that's human readable. Another advantage is that feature relevance can be easily calculated, to understand which variables had the most impact in predicting the results.

However, decision trees have a big disadvantage. Over-complex trees can be created and sometimes they do not generalize data well. This problem is called overfitting, this happens when a model gets to good predicting a certain test data based on the training data used to predict a certain outcome. When the model tries to predict unseen data, it will probably perform poorly. [25]

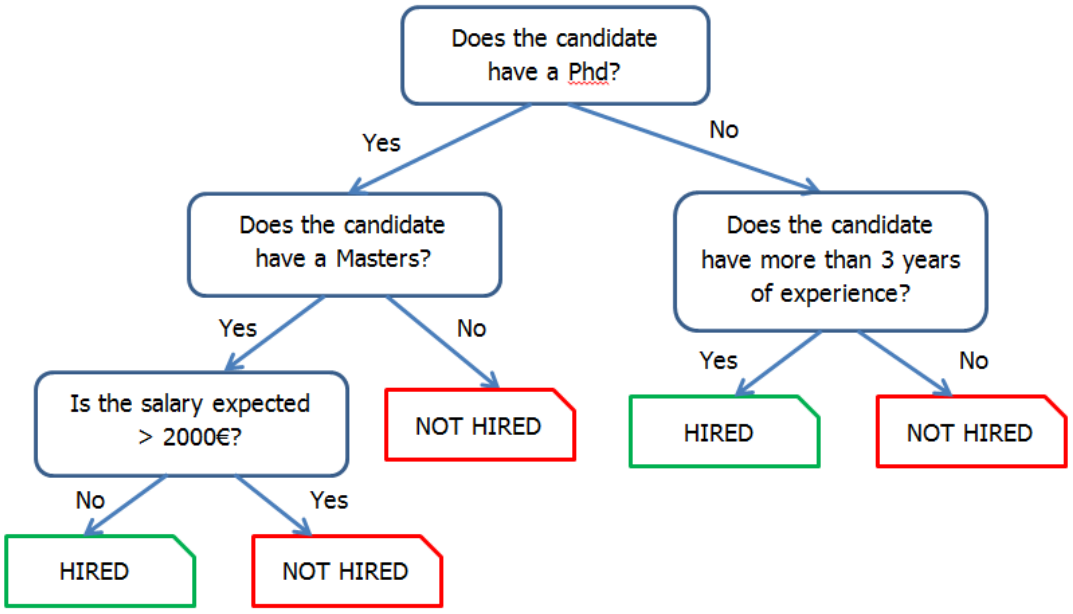

Figure 1: Example of a decision tree

3.2 Random Forest

"Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest." [16] In other words, random forests generate a vast number of random independent learning methods (e.g. decision trees), and them it calculates the mode or the mean of them if we are talking about classification or regression respectively. This idea is called bagging, "where to grow each tree a random selection (without replacement) is made from the examples in the training set". [16]

This leads to one big advantage of using random forests instead of decision trees, because it usually prevents overfitting.

3.3 Gradient Boosted Trees

Boosting is an ensemble technique, in which the predictors are not made independently like the random forests, but sequentially.

Here each predictor depends on the results of the previous models. It is like a learning method, where the next models learn from the mistakes made before. These predictors can be of a variety of learning algorithms, in our case they will be decision trees like in the random forest model. Here the "observations are not chosen based on the bootstrap process, but based on the error".[27]

Unlike the random forest, in the gradient boosted trees, the overfitting is more likely to occur if the criteria used to stop the learning process is not chosen carefully, because the error rate could converge to a very low value, meaning that the predictions made suit very well the test data, but when used on unseen data it will probably have a bad performance.

3.4 Support Vector Machines

Support Vector Machines represent the data as p-dimensional vectors, and then it splits the data into classes, using the best hyperplane of (p-1) dimensions as a border. The best hyperplane is the one where the points of each class, are the farthest away from it, an example is represented in fig 2. In fig 2 are three predictors created to split both classes. H1 has the worst performance, where the line does not separate both classes. H2 performs better than H1 because it can split them, but not with a great margin. H3 performs the best, because it also separates the classes but with a bigger margin than H2. To achieve this, the distance pf each point to the hyperplane must be calculated, and the plane that has the biggest margin to each class is the best classifier. This is called a linear separation of the data.

Usually in real scenarios, the data can't be easily split using linear hyperplanes, when this happens, a non linear approach must be made. The data points are mapped to higher dimension space, to try to find a hyperplane that can split the classes with higher margins [18]. In fig 3, shows an example of this procedure, here the data cannot be separated using linear functions. To solve this problem, the 2D points are mapped to a 3D space, it can bee seen that after the mapping is made, the separation is easier.

3.5 Artificial Neural Network

An artificial neural network (ANN) is a structure composed by inputs, outputs and nodes making the connections between them. Sections of these nodes are represented as layers. An example of a neural network's structure is represented in fig 4. There are different types of ANN, they can be feed-forwards, where the information flows in only in one direction, from the inputs, through the nodes and to the outputs. As the data flows through the different layers, mathematical functions are applied in each node of the several layers until it an output is calculated. There are also recurrent networks, where the nodes can have an information feedback of the subsequent nodes.

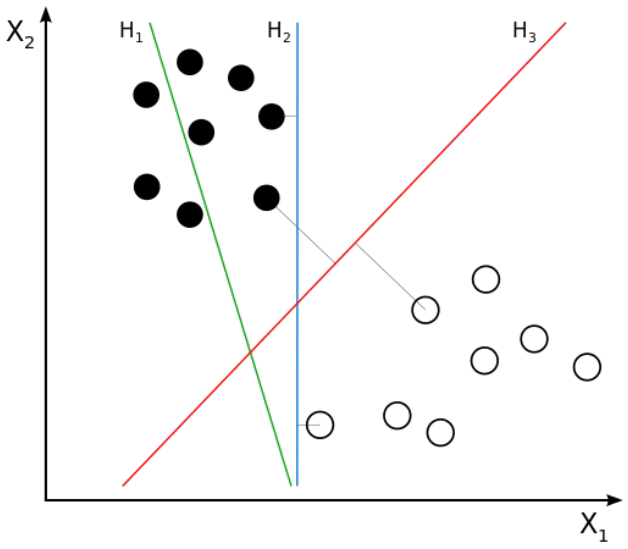

Figure 2: Example of a 3 SVM functions separating two classes. H1 performed the worst, it couldn't separate them. H2 can separate both classes but with a low margin. H3 separates them the best [26]

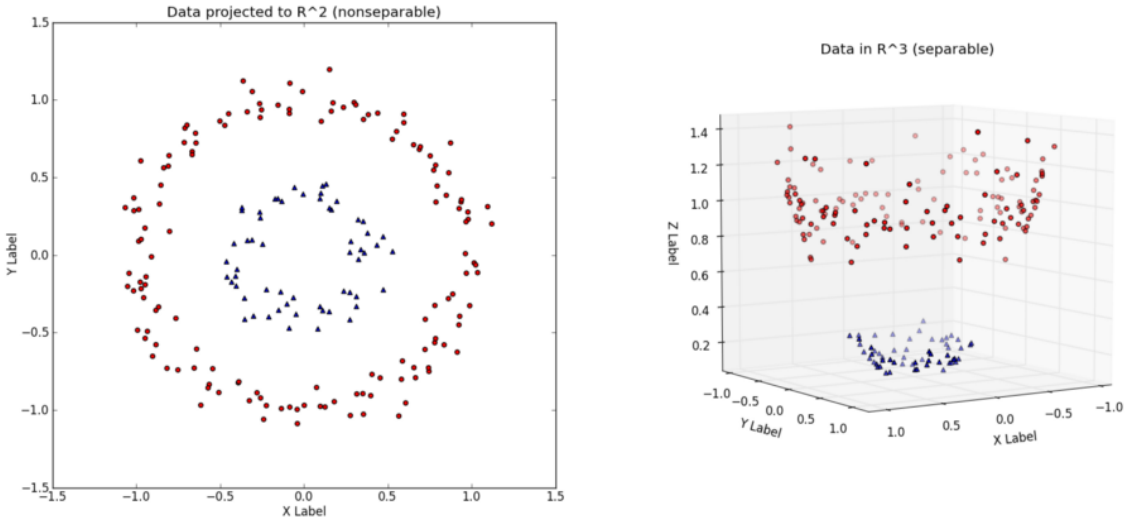

Figure 3: Example of applying a kernel and mapping the 2D points to 3D points for an easier class separation using SVM [27]

The learning method of the ANN starts with a random vector of values, or weighs that will be represented by the nodes of the network. Then the network is tested to check if the output is the expected or not. If the output is not expected, then changes are applied to previous weighs and the process start again according to a certain learning rate. The process will stop when the error rate converges to a defined limit or when the number of iterations chosen reaches the limit. The learning rate must be chosen carefully, because if it's too big, the algorithm can produce high oscillations in the error rate, and the rate could not converge to a limit. The information that was learned by the previous epoch (iteration) could also be lost. If the learning is too low, the network could converge to a limit really fast, and could not reach the

full potential of the network.

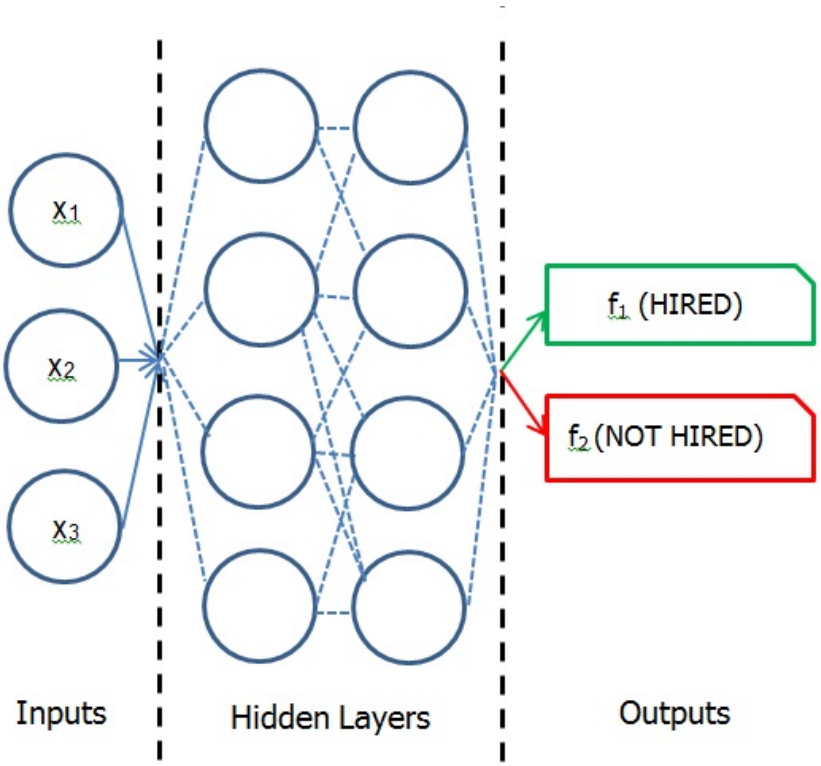

Figure 4: Example of an artificial neural network

4 Preprocessing

To be able to mimic the recruitment process, the first step is to analyze the data and start the preprocessing of the dataset. The dataset is from a recruitment software that an Engineering company uses to accept job applications for a wide range of positions. To make the preprocessing, decisions need to be made in order to change attributes, create or remove features from each candidate.

After the first preprocessing has been made, several algorithms and models will be tested to try to make simple predictions. If these simple predictions achieve good results, more complex strategies can be applied in order to improve results.

To process the data, python will be used because there are a lot of simple, efficient and open-source tools that were made by a vast community of researchers and developers. The code will be written using Jupyter Notebooks because it provides and easy way to interact with data by running small parts of the code, without having to run the full program. Also outputs can be easily saved for further examination.

The ultimate goal is to provide a system that replicates the hiring decisions made by the HR department through the several stages of the recruitment process.

The dataset has 5043 rows with 79 columns. Each row represents a candidate to a given position at a given time, or else, a job application, and each column is an application's attribute. There are columns that represent the candidate's attributes, that they have submitted themselves on the online HR tool, and there are some that are system attributes that will not be used.

There are fields that are empty in all rows, these need to be removed since they don't have any information. 16 columns where found and they are listed in table 6. After deleting all the empty columns, some columns could have the same value in all rows and these will also be removed since they don't add any specific information to distinguish each candidate. The columns found are listed in table 7.

Null Columns
RM Team
Applicant Type
Offer Accepted Date
Effective Start Date
Short List
Assessment
Background Check
Hired
Failed - Test
Failed - Assessment
Offer Accepted
Offer Retracted
Auto Disqualified
Profile Deleted by Applicant
Profile Deleted by Admin
Candidate Declined DPCS

Table 6: Null columns in the dataset

Same Value Columns
RM
Level 1
Level 2
Level 3
Job Requisition Country
Job Requisition City
Job Family

Table 7: Columns with the same value for all candidates in the dataset

The fields "Application ID" and "New Employee GID" will also be removed since they represent IDs from the database.

There are fields that represent the candidates' personal data. These fields will be removed since they don't add valuable information to create a model. A candidate will not be hired by having a "better" name or phone number. The fields that will be removed are listed in table 8.

Personal Columns
First Name
Last Name
Email Address
Phone 1
Phone 2

Table 8: Personal data to be removed

The following sections of this chapter will describe the candidates' attributes, and how they were processed in order to be used by the machine learning algorithms.

Job ID

The field "Job ID" represents the job that the candidates are applying in the company. There are 90 different positions available.

Distance to Work

There is more personal information about the candidates that can be used to create new features, for instance, the country, state and city of residence can be used to determine if the candidates live near or far away from the company. This is usually a decision factor, since a candidate that lives near the company is usually preferred from one that lives far away. Some hiring managers prefer to do interviews in person, so sometimes someone that lives far away might not even the opportunity to be interviewed. There is another problem associated with people that live outside Europe, sometimes they don't have the visa necessary to work in Portugal.

The "geopy" python library was used to calculate this distance. By using the city and country of residence as an input, the city coordinates can be extracted, and then, the distance between the location of the company, and the city of residence can be calculated. After having the distance calculated, it must be categorized in some way, or else, there would be to many categories, one for each city. Each candidate's distance calculated, will be categorized according to table 9.

Distance (km)	**Category**
d < 15	0
15 <= d < 30	1
30 <= d < 100	2
100 <= d < 350	3
350 <= d < 1000	4
d > 1000	5

Table 9: Categories for the distance from home to company

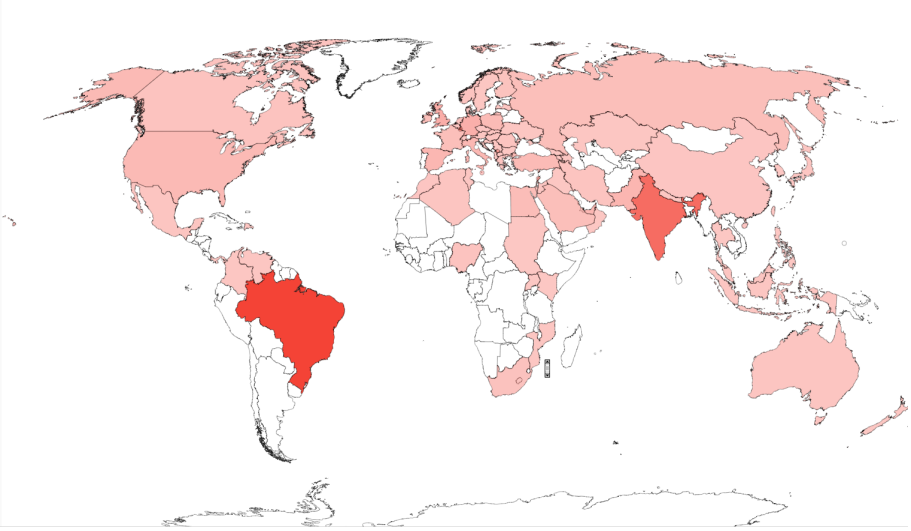

Figure 5: Distribution of the country of the candidates (Portugal was removed so that the tones of red would be more visible. Darker red represents more candidates that light red)

Type of Role

This field describes the type of role that the candidates are applying, it can be "Trainee" or "Professional". The characteristics of these two types of employees are different, for example, for a "Trainee", it's experience is not relevant at all, as opposed to the studies they made and which college they attended. For a "Professional" employee, it's more important their work experience and area of expertise.

The "Professional" applications represent 95% of all the applications.

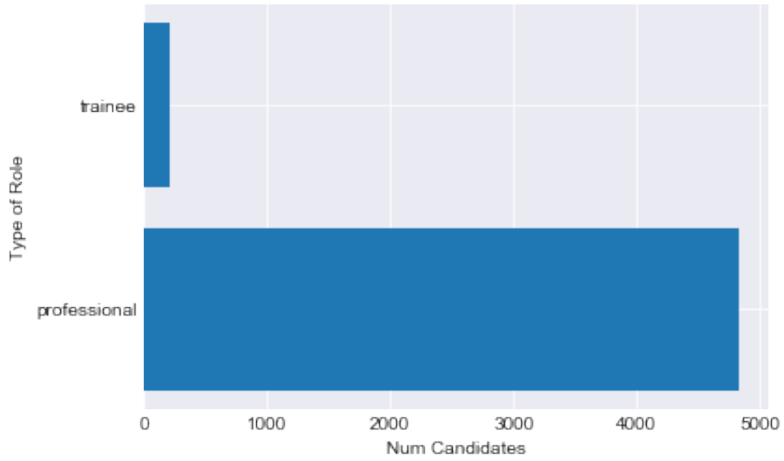

Figure 6: Type of roles

Career Level

This attribute represents the level of experience of each candidate. There are four different categories: recent graduate, mid-level professional, experienced professional, early professional. Their distribution is shown in fig 7. Since there only four categories, these will be transformed into dummies. This means that for each category, there will be a new binary attribute created that will state if the candidate has that career level or not. For example, if a candidate is recent graduate, it will the column recent graduate as 1, and the other three columns as 0.

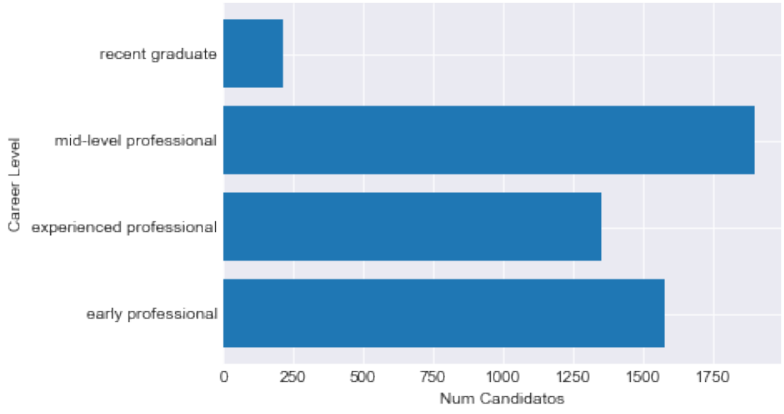

Figure 7: Career level categories

Language Skills

This field indicates the languages spoken by each candidate. Each field has all the languages spoken by each candidate and the level of proficiency associated to each language. There are 4 levels, and they go from 1 (beginner) to 4 (mother tongue). Levels 2 and 3 represent intermediate and fluent respectively.

In fig. 8 is an example of how this field is formatted for each candidate. All the languages' information is condensed into a single field and needs to be organized in a

```
Portuguese |  SpeakingProf: 04 Mother Tongue\nEnglish |
SpeakingProf: 02 Intermediate\nFrench |  SpeakingProf: 01
Beginner'
```

Figure 8: Language skills example

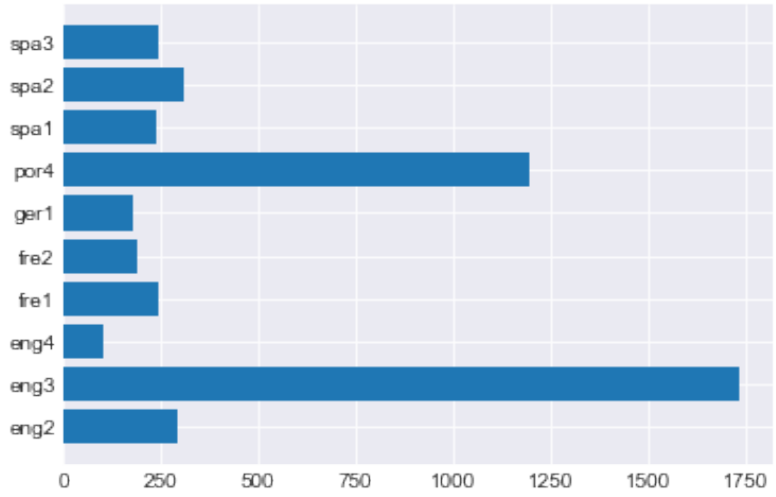

Figure 9: Languages most spoken by the candidates

way that represents relevant information and that can be used to create new features. We propose to create an object that has the language spoken by the candidates and it's corresponding proficiency. Each candidate can have multiple objects since they can introduce as many languages as they want.

In fig 9 can be seen the distribution of the languages most spoken by the candidates. The language that the candidates speak the most is English with level 3 of proficiency, even though the jobs that the candidates are applying are in Portugal, it is curious to see that despite that, the language that most people speak is English.

Nowadays it's crucial to speak English in almost every job, so a new feature will be created stating that the candidate speaks English, in this example, if it has a level of proficiency of 3 or 4. Since the jobs that the candidates are applying are for positions in Portugal, and the company's headquarters are in Germany, speaking Portuguese and German is usually a decision factor depending on the position they are applying.

Speaking several languages is also considered a good quality, so two new features will be added, one with the number of languages the candidates speak, with a level of proficiency of 2 or higher, and another indicating if the candidate is polyglot. A candidate is considered polyglot if it speaks 4 or more languages with a proficiency level of 2 or higher.

To conclude, 5 new features will be added: speaks english; speaks german; speaks portuguese; is polyglot; number of languages.

University of Highest Education Achieved

This field represents the university of highest education achieved by each can-

didate. The name of the university is chosen from a set of names, so there are no problems of different candidates inputting different names for the same university.

This attribute is very important and it can be a crucial decision factor. A candidate that studied in a prestigious university, such as Harvard or MIT is more likely to get a job than a candidate that studied in a not very well known college.

To determine if a university is prestigious, the names were compared to a list of the top 500 universities ranked by 'The Times Higher Education' in 2016 [31]. If a match occurs, that candidate will have a new feature indicating that he studied in a prestigious university. 83 of these universities were matched with 164 candidates having one of these chosen as their "university of highest education achieved".

Since the jobs are for positions in Lisbon, someone who studied in Lisbon could have an upper hand comparing to someone that didn't. So a new feature was created indicating that the candidate studied in Lisbon. As we are doing this for someone that studied in Lisbon, the same will be made by candidates that have studied in Portugal. 2696 candidates studied in Portugal, and 1937 of these studied in Lisbon.

Hiring Managers prefer certain universities, so a new feature will be created indicating that the candidate studied in one of those. A survey has been made to the hiring managers and the list of preferred universities in listed in table 10. 1528 candidates have this new feature.

Instituto Superior Técnico
Universidade Nova de Lisboa
Universidade de Lisboa
ISCTE
Universidade de Coimbra
ISEG
Universidade do Porto

Table 10: Preferred universities by the hiring managers

To conclude, four new features were added to each candidate: studied in Portugal; studied in Lisbon; studied in a prestigious university; studied in a preferred university.

Highest Education Achieved

This field represents the highest degree of education achieved by each candidate. In this field there are a total of 27 categories, which at first glance seams a lot for a level of educational degree. They are listed in table 40 and their distribution in fig 11.

It can be observed that all of the degrees start with a number, and degrees that start with the same number have a lot similarities or represent the exact same thing. For instance, all the categories in group number 4 in fig 11 represent masters students. Since they represent the same, these should be grouped in one single category called 'masters'. So strings that start with the same number will be grouped in the same category. After the grouping was made, instead of the initial 27 categories, now we only have 8. Their final distribution is represented in fig 12.

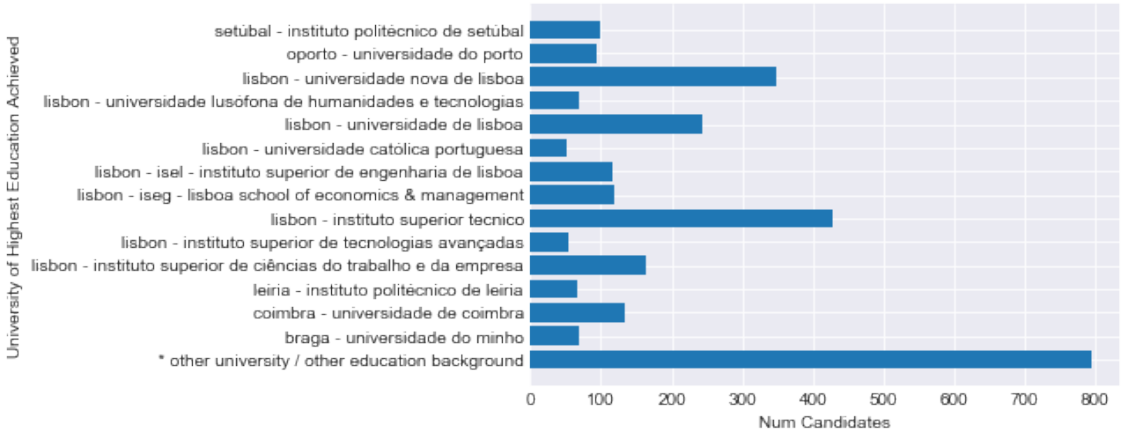

Figure 10: Top 15 universities filled by candidates

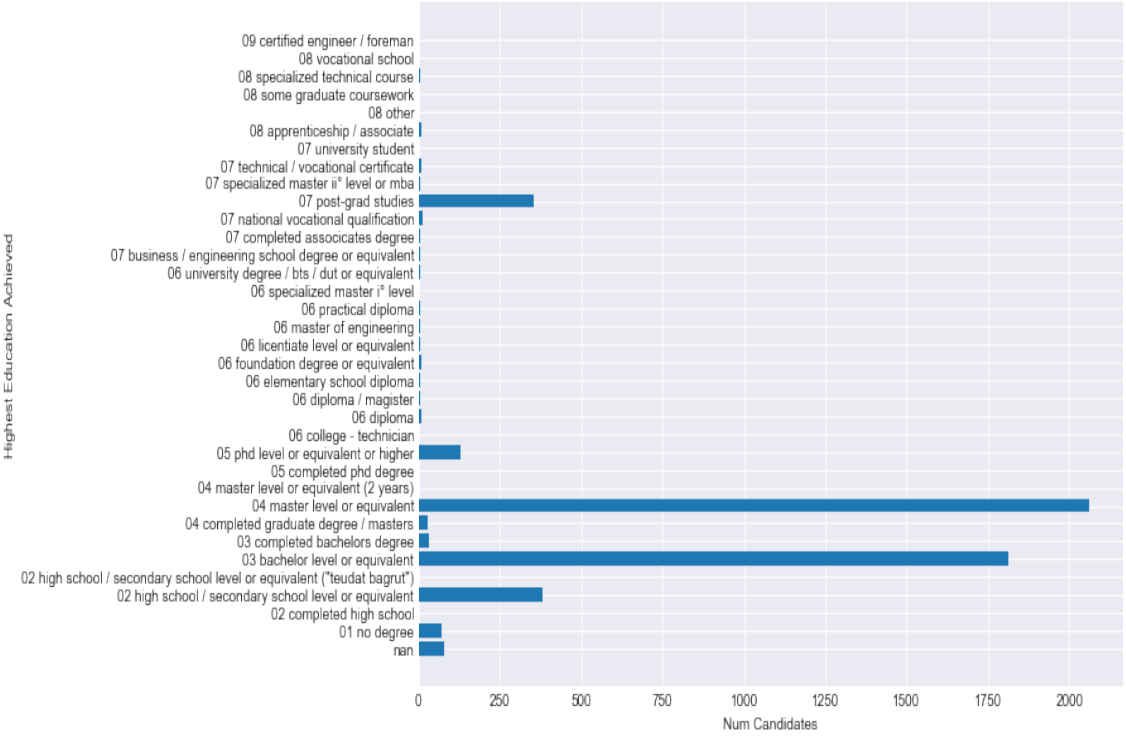

Figure 11: Distribution of highest education levels

Area of Study

After analyzing the options for 'Area of Study', it can be seen that there are 6 main areas and 46 sub-areas. These are considered main areas because they represent 93% of the dataset. They can also be easily distinguished by the name of the area, since they are represented with a number before the description. The other categories are represented by a very small percentage of the candidates, and therefore they will be grouped to the main areas according to table 41.

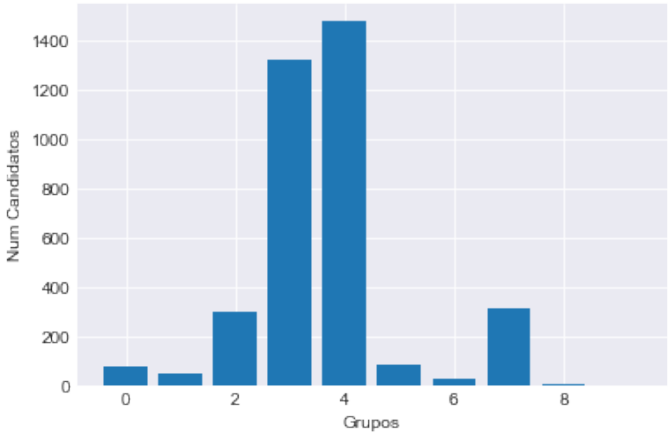

Figure 12: Final distribution of "Highest Education Achieved" after grouping similar areas

Is Internal

This attribute states if the application comes from a person that is already employed at the company and wants to apply for a different position. It could be interesting to find if applications from inside the company are favored by the managers. This attribute is a boolean, and therefor, dummies will be created from this attribute. Only 108 candidates have an internal application.

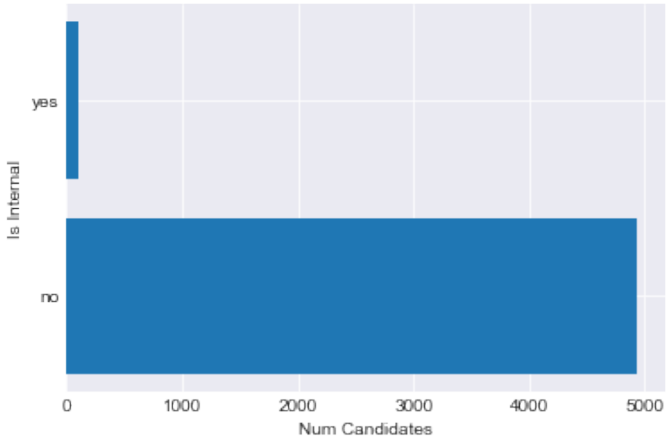

Figure 13: Distribution of the classes for the field Is Internal

Willing to Relocate

The fields represents how willing the candidate is to relocate if it is necessary. It has three levels of willingness, level 1 states that the candidate doesn't want to relocate at all. Level 2 indicates that the candidate is willing to relocate for a short period of time. Level three states that the candidate doesn't mind to relocate at all. The distribution of this field is represented in fig 14. The field will only be transformed into dummies.

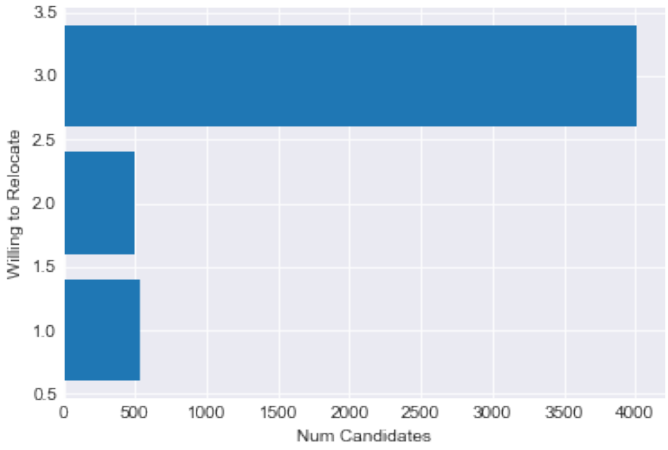

Figure 14: Distribution of the classes for the field Willing to Relocate

Certification Licenses

This field holds the information about the certifications that the candidate has. This is a free text field, and the candidate can input anything he wants. 550 candidates have filled this part of the form, and the rest left it blank. As a free text field, and with an existence of a vast different certifications each candidate can have, almost every entry is different from all of the others. It is hard to extract valuable information with so a field that is very sparse. A simple approach that can be made, is to create a new boolean feature stating that the candidate has a certification. This will be made by evaluating if the candidate has written anything in this field, if it has a more than a character it will be considered that the corresponding candidate has a certification license. In the end, the same 550 candidates ended having this variable.

Comments and Tags

These fields are for the hiring managers to input comments and tags about the candidate and the interview process. Some information could be extracted from here, but there are only 28 candidates that have comments associated with their application and only 7 with tags associated. There is another problem with the comments and tags section, this is filled by the hiring manager after the application was reviewed, so it won't be possible to create a feature for an application that just arrived to the system. Taking this in consideration, we have decided to remove these fields from the database.

Agency/Job Board Name

This section is to indicate if the candidate application came from an agency. There are 9 different agencies/job board names and their distribution is listed in fig 15.

6 of the classes (Employee_Referral, fb_global, Kariyer.net, LinkedInWrap, LinkedIn(Wrap), StepStone), only represent 0.4% , so a new group will be created named "others", with all these 6 classes grouped together. After the grouping was made, dummies will be created for each class and added as new features.

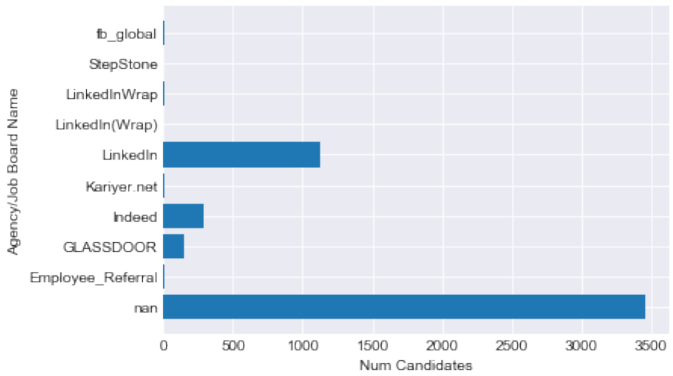

Figure 15: Distribution of the classes for the field Agency/Job Board Name

5 Experiments

The first experiments have an objective of predicting a certain result of a candidate's job application, or else, to predict a certain variable (target) of the dataset.

The models that will be tested are the ones that have proven better results in similar problems, the following will be used along the subsequent experiments: Decision Tree, Random Forest, Gradient Boosted Tree. We will also be testing with SVM and the Artificial Neural Network, described in the results tables as MLP (Multi Layer Processing), to see how they perform. All the algorithms used are form the python's sklearn package.

To create the models, the training set is always 80% of the dataset, and the test set is 20%, where the predictions will be made. To test each model, each algorithm was run 20 times, and for each model we will calculate the corresponding accuracies (min, max, average), ROC Scores (min, max, average), and also the confusion matrix for the model that had the best accuracy and the best ROC Score. In the results tables, if there is only one confusion matrix in an algorithm result, it means the model that had the best accuracy is the same as the model with the best ROC Score. If there are two, the one on the left always is for the model with the best accuracy, and the one on the right is for the model with the best ROC Score. Both confusion matrices must be analyzed because the model with the best accuracy could not be the same as the model with the best ROC Score. All the tables with these results will have these three metrics. The features that will be used are ones that were described in detail in the preprocessing chapter, these are listed in table 11.

Attributes Used
Distance to Work
Type of Role
Career Level
Speaks English
Speaks German
Speaks Portuguese
Is Polyglot
Number of Languages Spoken
Studied in Portugal
Studied in Lisbon
Studied in a prestigious university
Studied in a preferred university
Highest Education Achieved
Area of Study
Is Internal
Willing to Relocate
Certification Licenses
Comments and Tags
Agency/Job Board Name

Table 11: Attributes used in the experiments

Accuracy:

The accuracy will be calculated dividing the number of correct predictions by the total number of predictions made. The result will be the accuracy for that model.

Confusion Matrix:

The confusion matrix is a way of representing the final predictions of the model. This matrix is for binary classifiers, where there are only two possible classifications for each prediction. The is matrix is two by two, where it has the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). "If the instance is positive and it is classified as positive, it is counted as a true positive; if it is classified as negative, it is counted as a false negative. If the instance is negative and it is classified as negative, it is counted as a true negative; if it is classified as positive, it is counted as a false positive" [19]. Fig 12 shows how these values are displayed in the confusion matrix. The more true positives and true negatives the model can predict the better. Overfitting is "the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably" [21].

Predicted Class	1	0
1	True Positives	False Negatives
0	False Positives	True Negatives

Table 12: Confusion Matrix

Receiver operating characteristic (ROC) curve:

The ROC curve is a metric to evaluate the performance of a classifier. This curve is represented on an two dimensions graph, relating the true positive rate on the y-axis and the false positive rate on the x-axis (fig 16 17). To generate the curve, the algorithm that is making the predictions, must be able to define a probability of a certain instance of being a positive. For example, in our case, a candidate would be an instance, and each candidate has a certain probability of being hired (positive) or not hired (negative) based on their attributes. This probability acts as classification threshold. For example, if we define a threshold of 0.5, this means that all candidates that have a probability of being hired higher than 0.5, will be classified as hired, and the other will be classified as not hired. In figs 16 and 17, we can see different examples that will help to explain how the ROC curve is generated. The x-axis represents the probability of a candidate being in a certain class (blue for negatives and red for positives), and the y-axis represents the number of candidates that have that probability.

Let's assume that the total number of candidates in each class is 250. For a probability of 0.7, there are 50 positive candidates, and 0 negative candidates, for a probability of 0.5, there are 10 that are positives, and 10 that are negatives, and for a probability of 0.1, there are 10 negative candidates.

For example, in fig 16, for a threshold of 0.8, there will be 50 candidates classified as positive (area to the right of the threshold), and all of the other as negative. Since

there are 250 of each class, the true positive rate (TPR) will be 50/250, that results in 0.2. The false positive rate (FPR) will be 0, because there are no candidates that will be classified negative that are actually positive. So a point in the ROC curve will be created at (0, 0.2). If we keep moving the threshold to 0.6, the FPR will remain 0, while the TPR will increase to almost 1. This will generate the left part of the curve. Now between 0.6 and 0.4 is where the slope begins, because now, not all candidates that will be classified as positive, are actually positive, so the FPR will begin to increase (x-axis). At a threshold of 0.4, all the candidates that are positive will be classified as positive, so the TPR stabilizes at 1 as the FPR begins to increase to 1, until the threshold reaches 0, this will generate the right upper side of the curve. This type of curve is representative of a good prediction model, since both classes are well separated.

Now if we look at the graph in fig 17, this represents a worse model because both classes overlap more, make it more difficult to predict each candidate. Now, at a threshold of 0.8 is where the FPR begins to increase, where before, that only happened at 0.6. In this example the curve will be closer to the diagonal. If we imagine another example where the both classes overlap even more, the curve will be even closer to the diagonal. [19] [20]

To conclude, the bigger the area under the ROC curve, the better the model. This metric will be calculated for each of the models used in the following experiments.

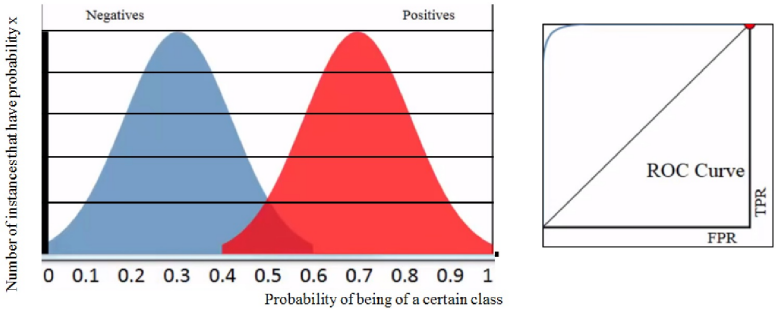

Figure 16: An example of a good class distribution and its respective ROC curve [20]

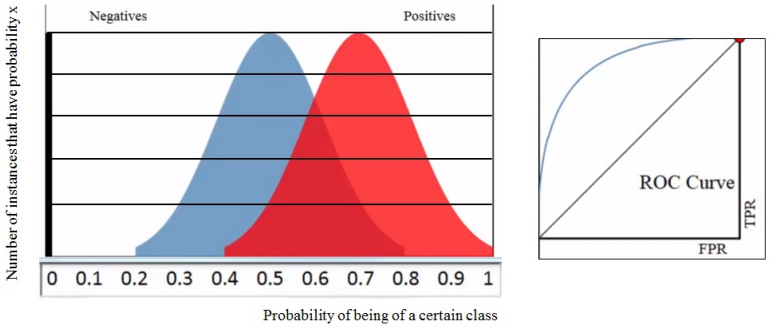

Figure 17: An example of a class distribution and its respective ROC curve, that had lower results, when comparing with fig 16 [20]

5.1 Predicting target: 'HM Review'

This field represents whether the candidate passed the first phase of the application or not, these are the ones that were chosen to be reviewed by the hiring managers, and were not immediately rejected after reading the corresponding CVs. The field is represented with the date of the review on the database, so if the candidate doesn't have a date, it means that it failed the first phase. To make this a field a possible target, we will change it to a boolean type, 1 (true) means that the candidate has a date on database and has passed the first phase, 0 (false) means that the candidate doesn't have a date and consequently, didn't pass the first phase. 3523 (69.8%) candidates were reviewed positively and passed the first phase and 1519 (30.2%) candidates didn't.

In table 13 we can see the first results for this section. The Decision Tree had the worst maximum accuracy, achieving 67.6%. SVM and the ANN had similar accuracies, around 72%. The Random Forest and GB Tree scored the best with 74.4% and 75.4% respectively. Despite the decision tree having a lower performance in terms of accuracy, it performed the best in the ROC Score metric, with a score of 62.9%. SVM performed the worst with only 55.8% for the max ROC Score. The average accuracies of these three algorithms are really close to the percentage of candidates that were reviewed (69.7%), and after analyzing the confusion matrix, it can be concluded that the algorithm predicted very well the true positives and very poorly the true negatives. This leads to conclusion that the models are biased to predict almost every candidate as a reviewed candidate. We conclude that the overall performance was bad due to the very low ROC scores, and that's a good indicative that the models created to predict this target aren't good.

In tables 15, 16, 17 are the feature importances for the tree models. The Specification of Study and the Area of Study play the biggest role on predicting the target on these models, with the exception of the RF, that had the mid-level professional variable as the most significant. It is curious to see that some of the new features created, as the number of languages (num_langs) and the distance from home to work (distance), are in the top 10 most important features in these three models. This means that creating those features had a positive impact in the predictions.

To try to improve the scores of the models, we will add the variable 'Job ID' to the dataset. At first we tried to use only variables that are characteristics of a candidate, but the Job ID represents the job that they are applying to, and the algorithms could find relationships between candidates that passed the first phase for the same job. The Job ID will be added to the attributes as a variable to see if it has any impact on the final models.

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.622	0.675	0.647	0.577	0.615	0.593	[169 182] [228 682]
Random Forest	0.699	0.744	0.723	0.561	0.591	0.577	[83 268] [95 274] [55 855] [68 824]
SVM	0.673	0.720	0.696	0.497	0.503	0.500	[0 353] [13 379] [0 908] [23 846]
MLP	0.679	0.723	0.700	0.500	0.605	0.531	[118 249] [100 794]
GB Tree	0.687	0.754	0.721	0.555	0.608	0.576	[99 242] [68 852]

Table 13: Results for target 'HM Review'

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.851	0.894	0.875	0.825	0.870	0.8537	[297 68] [66 830]
Random Forest	0.876	0.912	0.895	0.814	0.879	0.850	[296 74] [37 854]
SVM	0.683	0.723	0.701	0.500	0.504	0.501	[0 349] [13 360] [0 912] [23 865]
MLP	0.674	0.780	0.754	0.600	0.802	0.748	[328 54] [224 655]
GB Tree	0.902	0.928	0.915	0.866	0.897	0.881	[307 68] [21 865]

Table 14: Results predicting target 'HM Review' with 'Job ID' as an attribute

Feature	Importance %
Specification of Study	16.93
Area of Study	11.75
Highest Education Achieved	10.56
distance	9.51
mid-level professional	8.33
num_langs	7.62
uni_lisboa_bool	4.37
uni_pref_bool	3.98
speaks_eng	3.40
job board	3.28

Table 15: Feature importances for decision tree model for results in table 13

By adding the information of the job that the candidates were applying, the accuracy of the algorithms improved substantially. The Decision Tree, Random Forest, GB Tree, and SVM improved the most, achieving an average of accuracy of 86.6%, 90%, 91.5% and 87% respectively. The MLP didn't improve much, staying only at 72.5%. The confusion matrices also improved a lot, getting much closer to ideal matrix, and consequently the ROC Score improved. This leads to the conclusion that the 'Job ID' has a big impact to predict this target.

An analysis of the distribution of the current target will be made to check if the target is biased is someway. Out of the 89 jobs available, 17 of them classified all candidates as not reviewed and 19 of them as reviewed. 61 jobs have 90% of their

Feature	Importance %
mid-level professional	13.01
Specification of Study	11.82
Area of Study	11.54
experienced professional	9.43
Highest Education Achieved	7.70
num_langs	6.67
distance	5.34
early professional	4.21
corporate site	3.10
job board	3.01

Table 16: Feature importances for Random Forest model for results in table 13

Feature	Importance %
Specification of Study	18.19
Area of Study	15.68
mid-level professional	8.08
early professional	6.92
num_langs	6.84
experienced professional	5.53
Highest Education Achieved	5.36
distance	5.17
job board	3.90
glassdoor	3.13

Table 17: Feature importances for GB Tree model for results in table 13

Feature	Importance %
Job ID	56.67
early professional	15.89
Specification of Study	4.57
Highest Education Achieved	3.67
Area of Study	3.46
distance	2.50
num_langs	2.04
uni_pt_bool	1.12
linkedin	0.93
job board	0.87

Table 18: Feature importances for decision tree model for results in table 14

Feature	Importance %
Job ID	50.85
mid-level professional	6.01
Specification of Study	5.48
Area of Study	5.05
experienced professional	4.58
early professional	4.22
Highest Education Achieved	3.62
num_langs	2.86
distance	2.57
job board	1.67

Table 19: Feature importances for Random Forest model for results in table 14

candidates in one class. In tables 18, 19, 20, that show the feature importances for the tree models, it can be seen the high importance of the Job ID in predicting the

Feature	Importance %
Job ID	61.65
early professional	5.87
Area of Study	4.66
speaks_por	4.49
Specification of Study	4.14
Highest Education Achieved	3.19
num_langs	2.78
distance	2.58
mid-level professional	1.81
speaks_eng	1.24

Table 20: Feature importances for GB Tree model for results in table 14

target, because it always has more than 50% of importance in the different models.

Taking this in consideration, it explains the high correlation between the 'Job ID' and the 'HM Review'. This leads to the conclusion that most of the candidates that were reviewed are not influenced by the data that they filled, but by the job that they are applying to.

To try to eliminate the biased candidates, the algorithms will be run once more only with jobs that have classified less than 95% of the candidates in one class. This leaves 2215 candidates, 1644 (74%) that were reviewed, and 571 (26%) that weren't reviewed. The 95% margin was chosen because if it was lower than 90%, only 30% of the original dataset would be used in the new dataset. By using the 95% margin, 77% of the original dataset will be used. The algorithms will be run again as they were in the previous example and the results are showed in table 21 without the Job ID and in table 22 with Job ID.

After removing the biased candidates, for the predictions made without using the Job ID, the average accuracy of each algorithm improved around 3%.

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.635	0.718	0.683	0.569	0.642	0.6078	[63 79] [70 69] [77 335] [91 324]
Random Forest	0.735	0.787	0.755	0.580	0.652	0.608	[53 85] [33 383]
SVM	0.711	0.783	0.744	0.505	0.585	0.523	[28 97] [23 406]
MLP	0.709	0.778	0.745	0.600	0.703	0.649	[50 86] [83 54] [37 381][83 334]
GB Tree	0.722	0.785	0.755	0.577	0.633	0.608	[45 87] [32 390]

Table 21: Results for candidates that applied for jobs that had less than 95% of their candidates classified for one class (without 'Job ID')

5.2 Predicting target: 'All that failed'

In this section we will try to predict a new target. Here all of the dataset will be used again. In the database there are three possible ways to fail, 'Failed - HM Review', 'Failed - Screening' and 'Failed - Interview'. 'Failed - HM Review' is the opposite of the target predicted in section 5.1, or else, the candidate didn't pass the

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.659	0.717	0.693	0.575	0.645	0.618	[68 84] [73 329]
Random Forest	0.718	0.771	0.747	0.560	0.624	0.594	[43 92] [50 91] [35 384] [44 369]
SVM	0.699	0.780	0.740	0.506	0.681	0.540	[10 114] [77 48] [8 422] [109 320]
MLP	0.709	0.792	0.746	0.570	0.694	0.657	[53 74] [74 52] [41 386] [85 343]
GB Tree	0.715	0.782	0.751	0.589	0.655	0.611	[54 75] [46 379]

Table 22: Results for candidates that applied for jobs that had less than 95% of their candidates classified for one class (with 'Job ID')

first phase of the application. If a candidate has 'Failed - Screening' or 'Failed - Interview', means that it failed in the screening phase or in the interview. These steps of the application occur after the 'HM Review'. It is important to try to predict if a candidate will fail, not mattering the phase they are on. We will group all of these candidates and create a new target for them, stating that they failed. The next models will have this new attribute as the prediction target and will be called from now on 'all failed'. There are 49 candidates that have 'Failed - Screening' flag, 54 with 'Failed - Interview' flag and 1651 with 'Failed - HM Review' flag. This makes a total of 1754 candidates with the new target created.

The results for predicting this target are in tables 23 and 24. Once again the high discrepancy of the scores between using the 'Job ID' or not. For the tree models, it can be seen that the accuracies improved almost 20% for the decision tree, 12% for the RF. The BG Tree improved the most, achieving 90% for the maximum accuracy. Once again, the results show the high dependency of having the Job ID present in the attributes to make good predictions.

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.604	0.661	0.632	0.561	0.607	0.582	[662 179] [613 212] [248 172] [231 205]
Random Forest	0.669	0.708	0.685	0.593	0.631	0.609	[725 125] [243 168]
SVM	0.638	0.668	0.654	0.509	0.608	0.540	[829 2] [643 194] [417 13] [234 190]
MLP	0.633	0.684	0.655	0.500	0.588	0.504	[863 0] [684 125] [398 0] [303 149]
GB Tree	0.667	0.715	0.687	0.588	0.638	0.612	[736 112][719 124] [259 154] [250 168]

Table 23: Results for target 'all failed' (without 'Job ID')

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.844	0.861	0.853	0.823	0.849	0.837	[743 84] [746 101] [91 343] [76 338]
Random Forest	0.795	0.849	0.826	0.749	0.817	0.789	[754 57] [133 317]
SVM	0.638	0.670	0.654	0.508	0.605	0.552	[824 3] [634 197] [413 21] [238 192]
MLP	0.666	0.735	0.684	0.543	0.647	0.565	[650 167] [197 247]
GB Tree	0.868	0.893	0.882	0.847	0.889	0.869	[719 76] [59 407]

Table 24: Results for target 'all failed' (with 'Job ID')

5.3 New target - 'Hired'

In this section we will try to predict if the candidate was hired or not. The database does not have this information, despite the field 'Hired' being present in the database, all the rows are empty.

To try to find out the candidates that were hired, the company provided a list with the current employees. The list has 382 names in it. This list doesn't have an ID that can we can use to relate the database, where we have the applications of the candidates, and the hired candidates, that are in the list. The only way to make a match, is to compare the names of the candidates that are in the database ('First Name' and 'Last Name') with the full names on the employee list. This can result in inaccurate matches. For example, a hired person that has a common first name and last name, could have two or more applications associated to it, when in reality it was only one. It is even possible to have an error if a single exact match occurs, because it not guaranteed that despite having the same name, they are actually the same person. After trying to match all candidates, there are only 86 candidates that had the same name, but only 42 candidates had an exact match, the rest matched one employee on the employee list, with two or more applications on the dataset. Having this taken in count, the next results will have a big uncertainty associated with it.

The first experiment will be with the 86 hired candidates and all of the dataset, including the the 'Job ID'.

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.934	0.957	0.944	0.597	0.695	0.638	[920 14] [10 3]
Random Forest	0.952	0.969	0.962	0.523	0.632	0.567	[932 0] [924 1] [15 0] [21 1]
SVM	0.948	0.970	0.961	0.498	0.500	0.500	[919 0] [28 0]
MLP	0.909	0.939	0.923	0.499	0.531	0.511	[932 0] [15 0]
GB Tree	0.951	0.971	0.961	0.520	0.611	0.568	[920 0] [26 1]

Table 25: Results for target: 'Hired'

The results in table 25 show very high accuracy levels, with all models achieving more than 93% but very low ROC Scores, indicating once again poor quality models. This is explained because the target represents 1% of the dataset, which makes this a very imbalanced target, implicating that the results of the algorithm, were biased to classify the entire test set to not hired. The dataset will need to be balanced, there are two techniques used in machine learning that will be used to balance both classes in the dataset: undersampling and oversampling.

Undersampling:

Under-sampling is a method used to "balance class distribution through the random elimination of majority class examples. The rationale behind it is to try to balance out the dataset in an attempt to overcome the idiosyncrasies of the machine learning algorithm. The major drawback of random undersampling is that this method can discard potentially useful data that could be important for the induction process." [22]

To make the undersampling, the dataset will be reduced three times, maintaining 50%, 20% and 10% of the candidates that weren't hired at random, and maintaining always all of the hired candidates. Once again, the algorithms will be run in the same way as in the previous sections.

In table 26 are the results for the undersampling of 50%. The scores improved a lot, now all the ROC scores are above 82%, with the decision tree reaching 96.9% and the GB Tree and Random Forest with ROC Scores around 88%. This could mean that the models are getting much better at predicting this target, but we start to see a big difference between the minimum and maximum ROC Scores for each model. For example, in the GB Tree, the minimum was 68.7% and the maximum was 87.9%, this is a difference of almost 20%. This means that the results for best models, were to overfitted for the training samples. Even if the overfitting was not present in the experiments, tables 27, 28 and 29, show that once again the Job ID has always more than 43% of importance.

The results in table 30 and 31, that correspond to the undersampling of 20% and 10% respectively, show the same patterns that were described for the results of undersampling at 50%. Even with a more balanced class, show ROC scores very low, and once again poor predicting models were created.

We will try to predict the same target, but now without the Job ID. In tables 32, 33 and 34 are the results of the predictions without using the Job ID. For an undersample of 50% (table 32), the results still show a biased predictions towards the class with the most representatives, with ROC Scores around the 50%. By undersampling more to 20% and 10% (tables 33 and 34) of the candidates, the algorithms improved a little their ROC scores, but still they are very low. This again shows the high importance of the Job ID in predicting the outcome.

It can be concluded that the models created performed poorly, meaning that the target created probably has too many errors and inconsistencies, or that in fact is unpredictable. The correlation between the job that the candidates are applying the outcome of hired or not, should not be so high.

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.944	0.971	0.962	0.830	0.969	0.904	[618 7] [10 26]
Random Forest	0.949	0.974	0.964	0.742	0.887	0.799	[618 7] [10 26]
SVM	0.932	0.956	0.943	0.608	0.826	0.692	[622 3] [598 27] [26 10] [11 25]
MLP	0.933	0.956	0.945	0.703	0.936	0.834	[608 13] [597 28] [16 24] [3 33]
GB Tree	0.900	0.965	0.932	0.687	0.879	0.770	[617 12] [608 15] [9 23] [7 31]

Table 26: Undersampling 50% - Target: 'Hired'

Feature	Importance %
Job ID	43.09
Specification of Study	13.57
Area of Study	5.91
Highest Education Achieved	5.54
distance	5.30

Table 27: Feature importances for decision tree model for results in table 26

Feature	Importance %
Job ID	47.36
Specification of Study	9.02
Highest Education Achieved	5.82
Area of Study	5.00
distance	4.99

Table 28: Feature importances for Random Forest model for results in table 26

Feature	Importance %
Job ID	43.41
Specification of Study	15.49
num_langs	8.63
distance	6.17
Highest Education Achieved	4.62

Table 29: Feature importances for GB Tree model for results in table 26

Oversampling:

Here we will do an oversampling to balance the class we have to predict. "Random over-sampling is a non-heuristic method that aims to balance class distribution through the random replication of minority class examples. Several authors [23], [24] agree that random over-sampling can increase the likelihood of occurring overfitting, since it makes exact copies of the minority class examples. In this way, a symbolic classifier, for instance, might construct rules that are apparently accurate, but actually cover one replicated example." [22]

To to the oversampling we will duplicate the hired candidates 2, 4 and 8 times. It can be seen that the Job ID always has a big impact on predicting the target, here the algorithms will be tested without using the Job ID. In table 35 are the results for oversampling 2 times the hired candidates, the decision tree had substantially better

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.894	0.949	0.929	0.679	0.887	0.747	[248 2] [241 11] [12 12] [4 18]
Random Forest	0.905	0.956	0.933	0.567	0.774	0.672	[253 1] [247 6] [11 9] [9 12]
SVM	0.925	0.958	0.940	0.734	0.877	0.815	[252 1] [239 12] [10 11] [4 19]
MLP	0.909	0.956	0.935	0.627	0.881	0.796	[249 8] [245 13] [4 13] [3 13]
GB Tree	0.909	0.956	0.934	0.608	0.867	0.772	[250 6] [241 14] [6 12] [4 15]

Table 30: Undersampling 20% - Target: 'Hired'

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.842	0.925	0.883	0.708	0.855	0.772	[122 7] [4 13]
Random Forest	0.842	0.918	0.885	0.627	0.834	0.720	[121 7] [5 13]
SVM	0.849	0.938	0.903	0.687	0.876	0.809	[123 4] [5 14]
MLP	0.842	0.925	0.888	0.698	0.936	0.815	[120 8] [115 10] [3 15] [1 20]
GB Tree	0.856	0.925	0.890	0.684	0.872	0.774	[121 4] [113 10] [7 14] [4 19]

Table 31: Undersampling 10% - Target: 'Hired'

results than the other algorithms, reaching a maximum of 81.5% of ROC Score and a minimum ROC Score of 64%, which is a big difference from the maximum obtained. The RF and GB Tree had similar results between them with a max ROC Score of 53% and 54% respectively. SVM and MLP had also similar results between them, and also the lowest between all of them. Once again the decision tree is proving less resistance to overfitting. By oversampling to 4 times (table 36), all algorithms improved a little, but the same problems happened as before, the decision tree got to overfitted, RF and GB Tree scored low ROC Scores and the SVM and MLP scored again very low ROC Scores. With an oversampling of 8 times (table 36]), the scores improved even more, with RF achieving a ROC Score of 78.9%. Here the RF model is better than GB Tree, which only had 67.7%.

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.917	0.958	0.938	0.477	0.562	0.509	[610 15] [599 12] [12 2] [24 4]
Random Forest	0.956	0.983	0.969	0.497	0.500	0.500	[628 0] [623 0] [11 0] [16 0]
SVM	0.953	0.977	0.964	0.499	0.500	0.500	[624 0] [616 0] [15 0] [23 0]
MLP	0.947	0.977	0.966	0.500	0.500	0.500	[624 0] [619 0] [15 0] [20 0]
GB Tree	0.958	0.977	0.968	0.497	0.533	0.502	[623 1] [14 1]

Table 32: Undersampling 50% - Target: 'Hired' (without job id)

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.827	0.898	0.868	0.504	0.614	0.561	[235 12] [15 4]
Random Forest	0.895	0.959	0.920	0.492	0.538	0.507	[255 0] [240 2] [11] [22 2]
SVM	0.868	0.936	0.915	0.500	0.500	0.500	[249 0] [237 0] [17 0] [29 0]
MLP	0.902	0.951	0.929	0.500	0.500	0.500	[253 0] [246 0] [13 0] [20 0]
GB Tree	0.883	0.940	0.912	0.484	0.557	0.516	[248 0] [248 1] [16 2] [15 2]

Table 33: Undersampling 20% - Target: 'Hired' (without 'Job ID')

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.669	0.797	0.755	0.448	0.640	0.537	[100 16] [96 19] [11 6] [10 8]
Random Forest	0.767	0.880	0.825	0.475	0.562	0.519	[116 0] [110 3] [16 1] [17 3]
SVM	0.805	0.880	0.838	0.500	0.500	0.500	[117 0] [16 0]
MLP	0.767	0.880	0.847	0.500	0.500	0.500	[117 0] [16 0]
GB Tree	0.759	0.880	0.827	0.497	0.609	0.549	[114 5] [110 1] [11 3] [17 5]

Table 34: Undersampling 10% - Target: 'Hired' (without 'Job ID')

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.954	0.970	0.963	0.640	0.815	0.723	[1228 21] [1216 34] [17 16] [11 21]
Random Forest	0.960	0.973	0.967	0.500	0.537	0.513	[1247 0] [1241 0] [34 1] [38 3]
SVM	0.948	0.970	0.961	0.498	0.500	0.500	[1252 0] [30 0]
MLP	0.952	0.976	0.963	0.499	0.500	0.500	[1249 0] [33 0]
GB Tree	0.949	0.970	0.960	0.499	0.541	0.517	[1250 1] [1239 3] [31 0] [36 4]

Table 35: Oversampling 2x - Target: 'Hired' (without 'Job ID')

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.951	0.969	0.959	0.876	0.933	0.907	[1221 28] [1193 52] [14 82] [9 90]
Random Forest	0.921	0.947	0.935	0.549	0.638	0.592	[1254 2] [1247 5] [69 20] [67 26]
SVM	0.906	0.937	0.922	0.500	0.511	0.503	[1258 0] [85 2]
MLP	0.914	0.928	0.921	0.500	0.500	0.500	[1248 0] [1240 0] [97 0] [105 0]
GB Tree	0.914	0.939	0.930	0.543	0.606	0.573	[1245 1] [1230 3] [81 18] [88 24]

Table 36: Oversampling 4x - Target: 'Hired' (without 'Job ID')

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.956	0.968	0.963	0.955	0.974	0.966	[119 44] [4 245]
Random Forest	0.885	0.926	0.908	0.692	0.789	0.737	[1239 8] [102 143]
SVM	0.819	0.845	0.834	0.500	0.514	0.505	[1256 2] [229 5]
MLP	0.917	0.942	0.928	0.500	0.877	0.661	[1259 0] [233 0]
GB Tree	0.855	0.889	0.871	0.623	0.677	0.653	[1247 23] [1222 33] [142 80] [147 90]

Table 37: Oversampling 8x - Target: 'Hired' (without 'Job ID')

5.4 Predicting targets: Grades of Applications

The results of the previous experiments concluded that the dataset was highly biased to make good predictions if the Job ID was present in the dataset. This may have happened because the application were the people of HR updated the information, was not updated in a consisting way, since it was not mandatory by the company.

To have reliable classifications of the job applications, HMs were asked to classify CVs that correspond to the job applications in the dataset that we have been using along the previous sections. In total 250 CVs were classified and tagged by the HMs. The classification had three possible grades: 0 (very bad), 1 (others) 2 (very good). The classifications were made this way, because one of the main goals, is to try to predict if a candidate is very good, to be immediately interviewed, or if it's very bad, so that it would be interviewed after the good candidates. The HMs were asked to classify the applications like they would in a real scenario, and the CVs were randomly selected from each job title to avoid biases and inconsistencies. The distribution of the grades that the HMs gave to the CVs are represented in fig 18.

The first goal is to determine if the Job ID still has a big impact on predicting the grades, this time with the grades given by HMs.

Since now we have 3 possible grades, and the goal is to determine if a candidate is very good, or very bad, we will make binary predictions, one for each grade: grade 0 vs others; grade 1 vs others; grade 2 vs others.

The results to predict each grade are in tables 42, 43, 44. The summary of these results are presented in table 39 To predict grade 0, GB Tree had the best accuracy and ROC Score out of all the algorithms, scoring 69.8% and 70% respectively. RF preformed second with 68.3% and 67.4% on the same metrics. Predicting grade 1 shows better results in terms of accuracy, but this happens because grade 1 is less representative vs the other grades than grade 0. Looking at the ROC scores, RF and the Decision tree performed best this time, scoring 66.8% and 66.3% respectively. Predicting grade 2 proved to be harder, probably because this time, the grade is even less representative vs the other grades. It's curious to see that this time, the GB Tree performed better than the RF and the Decision tree performed the best this time with a ROC Score of 65.8%. In tables 51, 52, 53, 54, 55, 56, 57, 58, 59, are the feature importances for the models created with the Decision Tree, RF and GB Tree predicting all the grades. It can be seen that the Specification of Study, job ID, Area of Study and num_langs have almost always the biggest impact. This time with a more consistent and reliable target, the Job ID doesn't have the high importance that had in the previous sections, where the targets were inconsistently created. This shows the importance of consistent labeling of the target to be predicted.

The results without using the Job ID are in tables 45, 46, 47. The summary of these resutls are in table 39 The scores were similar as the ones obtained in the previous results.

Since now we are dealing with CVs, some new features will be created from them. The first attribute is related to the number of pages in the CV. A very long CV is usually considered a bad attribute because people should be able to resume the information in a concise way. This variable will be called "num_pages". The

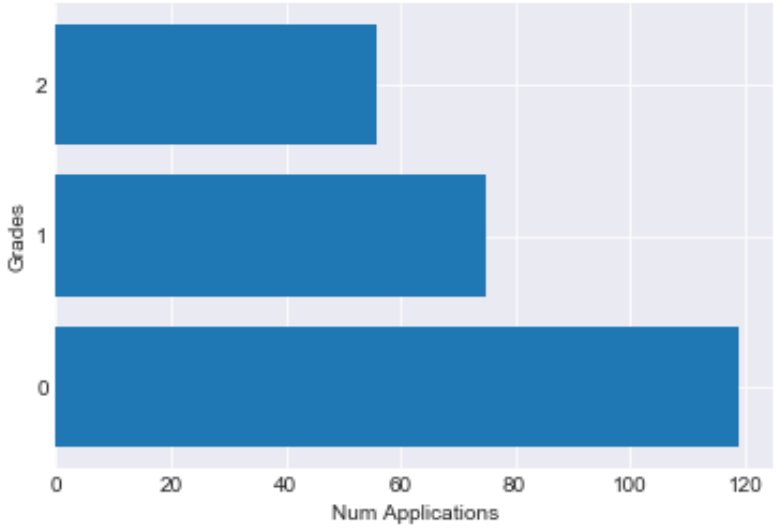

Figure 18: Distribution of the grades given by all HMs

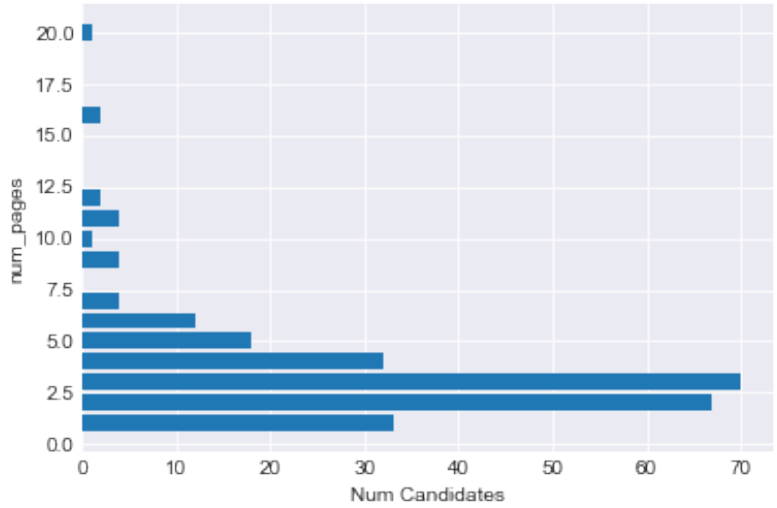

Figure 19: Distribution of the variable num_pages

distribution of the number of pages in the CV is in fig 19. Two more attributes will be created. To create these attributes, keywords were created for each job description. Then we will match the words present in each CV with the corresponding keywords for the job that the candidates are applying. The new attributes that will be created represent the number of total matches between words in the CVs and the keywords, and the other represents the number of different words matched. The name of these attributes will be "total_count" an"diff_count". The distribution for these attributes are in fig 21 and 20. These three attributes will be added to the dataset to see if there are any improvements. The results for the new predictions are showed in tables 48, 49 and 50. The new features should have made the scores better, but in fact, in almost all cases it scored worse.

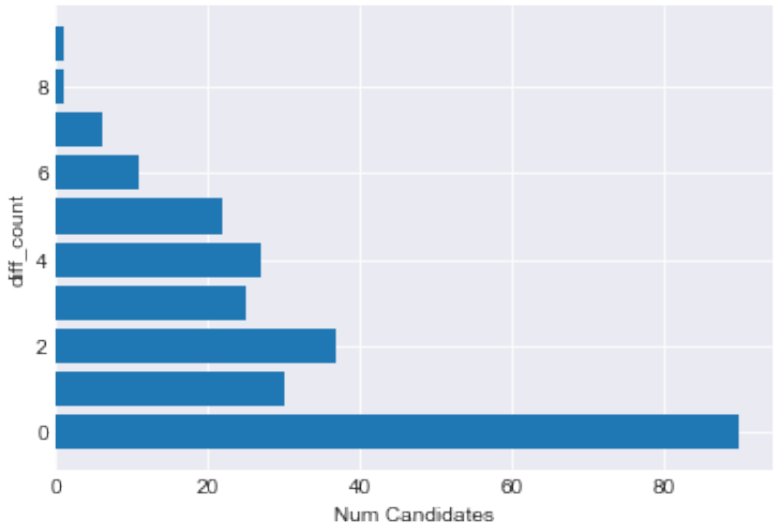

Figure 20: Distribution for the variable diff_count

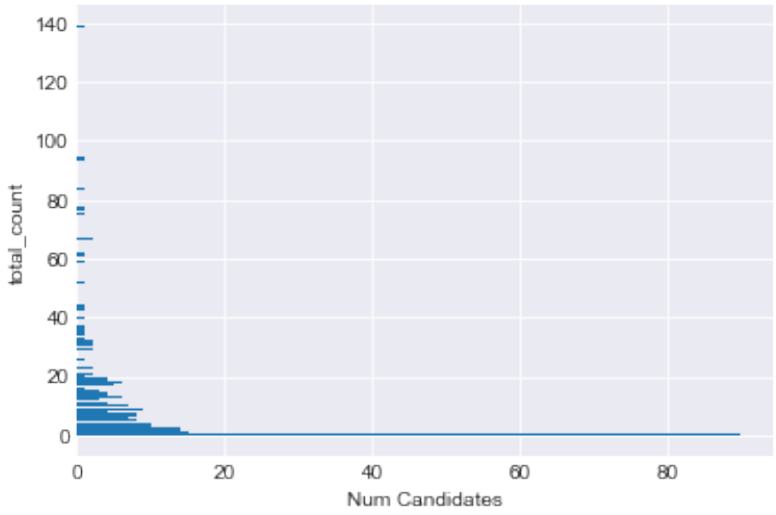

Figure 21: Distribution for the variable total_count

Algorithm	Max Accuracies			Max ROC Scores		
	Grade 0	**Grade 1**	**Grade 2**	**Grade 0**	**Grade 1**	**Grade 2**
Decision Tree	0.635	0.714	0.746	0.638	0.658	0.663
Random Forest	0.683	0.841	0.810	0.674	0.668	0.566
SVM	0.587	0.689	0.689	0.638	0.546	0.510
MLP	0.556	0.762	0.857	0.501	0.500	0.500
GB Tree	0.698	0.746	0.810	0.700	0.624	0.635

Table 38: Results of predicting the labeled grades with the Job ID

Algorithm	Max Accuracies			Max ROC Scores		
	Grade 0	**Grade 1**	**Grade 2**	**Grade 0**	**Grade 1**	**Grade 2**
Decision Tree	0.603	0.730	0.746	0.621	0.644	0.657
Random Forest	0.619	0.730	0.825	0.655	0.600	0.576
SVM	0.587	0.778	0.873	0.593	0.547	0.500
MLP	0.667	0.730	0.905	0.667	0.670	0.504
GB Tree	0.651	0.730	0.762	0.636	0.615	0.565

Table 39: Results of predicting the labeled grades without the Job ID

6 Conclusions and Recommendations

6.1 Main conclusions

In the first section of the results (section 13), where the target to predict is whether the candidate will go through the first phase of the screening process, there is a big difference between using the Job ID as an attribute or not. Without using the Job ID, the best scores were achieved by the Random Forest and Gradient Boosted Tree with accuracies of 74.4% and 75.5%. With the Job ID as an attribute, both algorithms scored best again, but with 90% and 91.5%. The feature importances of the tree models, show that it had more than 50% in all of them, this means that there is a high correlation between the Job ID and the predicted target. Without looking at the feature importances, one might think that accuracies around 90% would replicate the screening process very well, but results showed that the impact produced by Job ID was too high, this means that the results of the applications depended more on the job that the candidates' are applying to instead of the their features. Without using the Job ID, the best accuracies were around the 75% margin, this means that the decisions made by the recruiters were not replicated as well, but the algorithms could be used to help hiring managers to save time, and analyze the candidates that were predicted positively first, in hope of finding the right candidate sooner than it would be by reviewing all the candidates "manually".

In the second section of of the results (section 5.2), a new target was created with all the candidates that failed along the several stages of the process. The results were similar to the previous target, again the Job ID shows a big correlation with the predicted target. The conclusions regarding whether or not the decisions can be replicated, are similar to the ones stated above.

The jobs that have reviewed more than 95% of the candidates in one class were removed to try to eliminate the Job ID dependency. This resulted in reducing the dataset to 2215 candidates from the original 5043. Results show that the algorithms improved 3% without using the Job ID.

After reducing the dataset, more than 50% of candidates were removed. This happens because there a lot of jobs that have either all of the candidates as reviewed positive or all of them negative. There was an inconsistent tagging of the job application results, because the tool used is not mandatory by the company, even some managers had slight different usages for each tag.

To try to eliminate the high correlation between the results of the applications and the Job ID, in the final results, each hiring manager was asked to review and grade a set of CVs, providing a total of 250 CVs. There were 3 possible grades 0 (very bad), 1 (others) and 2 (very good). Here it can be concluded that this time with a more consistent labeling of the targets, the Job ID wasn't as important as the previous ones, but it was still one of the attributes that had the highest feature importances. The best ROC Scores are around 65% for the three grades, this means that this time the algorithms didn't perform as well as the ones stated above. The lower scores obtained in these results, could mean that with a more consistent labeling, the predictions are harder to make, and that it is in fact, hard to replicate the hiring managers' decisions using the features described in this work. A possible explanation could be the fact that the features used are all objective, but the candidates are not hired only by these features, soft skills are also important, like verbal communication, cooperation and teamwork, sympathy, learning ability,

and these should also be taken in consideration.

The features that had the most impact in predicting these targets, were consistent in all of the experiments. The attributes that contributed the most to predict the different targets were the specification and area of study, the highest education achieved, the number of languages spoken, and the distance from home to work. Here it is proven the importance of adjusting or creating new features, and not just blindly using the features that were already in the dataset. For example, the distance to work and the number of languages spoken were not explicitly present in the dataset, they were created because they represented important characteristics of the problem at hand. This lead to the conclusion that the preprocessing of the dataset, is very important when using machine learning algorithms and should not be left out.

Some curious interpretations that can be concluded after analyzing the importance of the features, is that the universities where the candidates have studied are not as important as the academic degree that they achieved. This is proven because the importance of the variable "highest education achieved", that represents the academic degree, was always superior to the ones created to represent specific or groups of universities, like the Portuguese universities, the preferred universities by the hiring managers or the top rated ones. Another curious conclusion, is that the feature that states if the candidate is internal or not, could create a misconception that an internal candidate is better than an external because it is already working at the company, but in fact, the low importance of the feature across all experiments, proves that in fact it is not as decisive as one might think.

To conclude, the results of the experiments showed that the replication of the screening process was not as accurate as expected, but the algorithms can still be used to assist hiring managers by indicating which candidates are more likely to perform best in the screening process, and thus saving them time by not having to analyze all the applications for a given position.

6.2 Contributions for the scientific and corporate communities

For further investigations and studies, the code produced making this work can easily be reused in the future to test the models again with more data, or in other machine learning fields. It provides a simple way of testing multiple algorithms, where the results of the metrics used in this study (Accuracy, AUC ROC Score, Confusion matrix) can be easily examined. This can be done for both academic and corporate communities.

6.2.1 Contributions at an academic level

At an academic level, the notebooks made to write this paper, can be reused or studied as an example of the process of making predictions with machine learning, from the preprocessing part to the production and analysis of the results. The code in the notebooks has different usages of the pandas and sklearn python packages that are common in other areas, other than job applications.

6.2.2 Contributions at a corporate level

At a corporate level, although the experiments did not result in creating good models that replicated the screening process, it showed that from now on, all the steps of the application process need to be registered in a more consistent way among all hiring managers. The platform should also mandate the filling of the several steps of the job application process to keep all the information of each candidate in each application step. If these changed are followed, the models created will be a lot more accurate and will replicate better the several steps of the application.

6.3 Study limitations

There were limitations in this study. The first one was that the predicted target of "HM Review" (variable that states if the candidate passed the first screening phase) was not labeled with consistency among the hiring managers. This happens because this platform does not require the managers to update the information about the job applications, some only use it to view the information of the job applications, and updates the results of the applications if they feel it helps them organize them. For instance, the target "Hired", that seemed like a promising predicting target to be explored, but it was empty in all rows. This happened because once a candidate is hired, in most cases the application is closed without updating which candidate was hired in the platform. To try to label the hired target, the company provided a list of the current employees so that a possible match between the applications and the hired employees. Here there was no id to make a match between the tables, only the names could be compared to find these matches. The platform only required the candidates to fill their first and last name, and by doing this, it resulted in duplicated names and multiple matches between employees and job applications.

In the final experiments, the limitations where on the number of candidates that were labeled in the dataset, only 250 candidates were used.

6.4 Proposals for future investigations

A limitation of the final experiments was the number of job applications that was labeled. A proposal for future investigations, would be a follow up of this study, but with more data, to verify if the results improve, or that in fact, that it is hard to make very accurate predictions with these attributes.

Soft skills were not tested in this work, a proposal for a future investigation is to add features related to soft skills and verify how they impact in replicating the hiring manager's decisions.

Another proposal is to make these type of predictions, but with features created from social media (LinkedIn, Facebook), replicating what hiring manager would look for when searching a person online, to check if a good public profile is relevant to be hired or not.

7 References

[1] Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. 2014 Science and Information Conference, 372–378. https://doi.org/10.1109/SAI.2014.6918213

[2] Bachrach, Y. (2015). Human judgments in hiring decisions based on online social network profiles. Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015.
https://doi.org/10.1109/DSAA.2015.7344842

[3] Blei, D. M., Edu, B. B., Ng, A. Y., Edu, A. S., Jordan, M. I., & Edu, J. B. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993–1022. https://doi.org/10.1162/jmlr.2003.3.4-5.993

[4] Koh, M. F., & Chew, Y. C. (2015). Intelligent Job Matching with Self-learning Recommendation Engine. Procedia Manufacturing, 3(Ahfe), 1959–1965. https://doi.org/10.1016/j.promfg.2015.07.241

[5] Gamulin, J., Gamulin, O., & Kermek, D. (2014). Comparing classification models in the final exam performance prediction. 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2014 - Proceedings, (May), 663–668.
https://doi.org/10.1109/MIPRO.2014.6859650

[7] J.A. Gill, S. Nowson and J. Oberlander, "What are they blogging about? Personality, topic, and motivation in blogs", Proc. of AAAI ICWSM. 2009

[8] G. Mishne, "Experiments with mood classification in blog posts", Proc. of 1st Workshop on Stylistic Analysis Of Text For Information Access Style 2005. 2005

[9] J.W. Pennebaker and L. King, "Linguistic Styles: Language Use as an Individual Difference," Journal of Personality and Social Psychology, vol. 77, 1999, pp. 1296–1312.

[10] Sivaram, N., & Ramar, K. (2010). Applicability of Clustering and Classification Algorithms for Recruitment Data Mining. International Journal of Computer Applications, 4(5), 23–28. https://doi.org/10.5120/823-1165

[11] Chien, C. F., & Chen, L. F. (2008). Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry. Expert Systems with Applications, 34(1), 280–290. https://doi.org/10.1016/j.eswa.2006.09.003

[12] V. Sarda, P. Sakaria, S. Nair, Relevance Ranking Algorithm for Job Portals, International Journal of Current Engineering and Technology, 2014.

[13] Buckley, P., Minette, K., Joy, D., & Michaels, J. (2004). The use of an automated employment recruiting and screening system for temporary profes-

sional employees: A case study. Human Resource Management, 43(2–3), 233–241. https://doi.org/10.1002/hrm.20017

[14] Holbeche, L. (2009). Aligning human resources and business strategy. Routledge.

[15] Jackson, S. E., Schuler, R. S., & Werner, S. (2009). Managing human resources (p. 358). Mason, OH: South-Western Cengage Learning.

[16] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

[17] Friedl, M. a. M. A., & Brodley, C. E. C. E. (1997). Decision tree classification of land cover from remotely sensed data. Remote Sensing of Environment, 61(3), 399–409. https://doi.org/10.1016/S0034-4257(97)00049-7

[18] Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 20(3), 273–297. https://doi.org/10.1023/A:1022627411411

[19] Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

[20] Data School (2014, November 19). ROC Curves and Area Under the Curve (AUC) Explained [Video File]. Retrieved from
https://www.youtube.com/watch?v=OAl6eAyP-yo

[21] Overffiting - (2018). In oxforddictionaries.com

[22] Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets : A review. Science, 30(1), 25–36. https://doi.org/10.1007/978-0-387-09823-4_45

[23]] N. V. Chawla, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Oversampling TEchnique. Journal of Artificial Intelligence Research, 16:321- 357, 2002

[24] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One sided selection. In Proceedings of the Fourteenth International Conference on Machine Learning, pages 179-186, Nashville, Tennesse, 1997. Morgan Kaufmann.

[25] Scikit-learn.org. (2018). 1.10. Decision Trees — scikit-learn 0.20.0 documentation. [online] Available at: http://scikit-learn.org/stable/modules/tree.html [Accessed 5 Oct. 2018].

[26] En.wikipedia.org. (2018). Support vector machine. [online] Available at: https://en.wikipedia.org/wiki/Support_vector_machine [Accessed 10 Oct. 2018].

[27] Medium. (2018). Gradient Boosting from scratch – ML Review – Medium. [online] Available at: https://medium.com/mlreview/gradient-boosting-from-scratch-

1e317ae4587d [Accessed 10 Oct. 2018].

[28] Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets : A review. Science, 30(1), 25–36. https://doi.org/10.1007/978-0-387-09823-4_45

[29] Scikit-learn.org. (2018). scikit-learn: machine learning in Python — scikit-learn 0.20.0 documentation. [online] Available at: http://scikit-learn.org/stable/ [Accessed 10 Oct. 2018].

[30] Perceptron, M. (2010). Neural Networks, 257–259. https://doi.org/10.1016/0893-6080(94)90051-5

[31] World University Rankings. (2016) [online] Available at: https://www.timeshighereducation.com/world-university-rankings/2016/world-ranking

[32] J.R. QUINLAN, Induction of Decision Trees, 1986, Machine Learning 1:81-106

[33] Quinlan, J. R. (1993). C4. 5: programs for machine learning (Vol. 1). Morgan kaufmann.

[34] Sharma, S., Agrawal, J., & Sharma, S. (2013). Classification Through Machine Learning Technique: C4.5 Algorithm based on Various Entropies. International Journal of Computer Applications, 82(16), 975–8887. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.402.2386&rep=rep1&type=pdf

8 Attachment

8.1 Preprocessing tables

'01 no degree'
'02 completed high school'
'02 high school / secondary school level or equivalent,
'02 high school / secondary school level or equivalent ("teudat bagrut")'
'03 bachelor level or equivalent'
'03 completed bachelors degree'
'04 completed graduate degree / masters'
'04 master level or equivalent'
'04 master level or equivalent (2 years)'
'05 completed phd degree'
'05 phd level or equivalent or higher'
'06 college - technician'
'06 diploma'
'06 diploma / magister'
'06 elementary school diploma'
'06 foundation degree or equivalent'
'06 licentiate level or equivalent'
'06 master of engineering'
'06 practical diploma'
'06 specialized master i° level'
'06 university degree / bts / dut or equivalent'
'07 business / engineering school degree or equivalent'
'07 completed associcates degree'
'07 national vocational qualification'
'07 post-grad studies'
'07 specialized master ii° level or mba'
'07 technical / vocational certificate'
'07 university student'
'08 apprenticeship / associate'
'08 other'
'08 some graduate coursework'
'08 specialized technical course'
'08 vocational school'
'09 certified engineer / foreman'

Table 40: Categories for 'Highest Education Achieved' attribute

Sub Area	Main Area
political science foreign language human resources psychology linguistics visual arts music literature history design	01 arts and humanities
business finance economics management accounting marketing law quality managament	02 business and law
computer science information systems business informatics	03 computer sciences
electrical engineering engineering general electronics biomedical engineering engineering management physics mechanical engineering civil engineering chemical engineering mechatronics engineering	04 engineering
biology biological science science health and safety management	05 natural sciences, medicine and pharmacy
other geography drafting organization development communication public relations statistics mathematics other technology	06 other
	nan

Table 41: Grouping of the areas of study

8.2 Results tables

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.460	0.635	0.536	0.457	0.638	0.536	[23 14] [9 17]
Random Forest	0.444	0.683	0.548	0.450	0.674	0.553	[28 5] [15 15]
SVM	0.381	0.587	0.500	0.400	0.638	0.505	[18 22] [4 19]
MLP	0.381	0.556	0.481	0.466	0.501	0.498	[35 0] [28 0]
GB Tree	0.476	0.698	0.568	0.482	0.700	0.570	[24 11] [8 20]

Table 42: Results of predicting grade 0 with Job ID

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.492	0.714	0.602	0.433	0.663	0.540	[37 12] [31 17] [6 8] [5 10]
Random Forest	0.571	0.841	0.674	0.421	0.668	0.532	[48 1] [9 5]
SVM	0.556	0.810	0.689	0.479	0.546	0.510	[51 0] [46 2] [12 0] [13 2]
MLP	0.571	0.762	0.694	0.492	0.500	0.500	[46 2] [13 2]
GB Tree	0.540	0.746	0.643	0.446	0.624	0.541	[48 0] [35 14] [15 0] [7 7]

Table 43: Results of predicting grade 1 with Job ID

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.587	0.746	0.640	0.422	0.658	0.522	[40 9] [41 10] [7 7] [7 5]
Random Forest	0.651	0.810	0.733	0.459	0.566	0.514	[51 1] [50 4] [11 0] [6 3]
SVM	0.556	0.810	0.689	0.479	0.546	0.510	[54 0] [9 0]
MLP	0.667	0.857	0.771	0.500	0.500	0.500	[52 1] [10 0]
GB Tree	0.651	0.810	0.723	0.436	0.635	0.532	[47 3] [43 8] [9 4] [7 5]

Table 44: Results of predicting grade 2 with Job ID

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.429	0.603	0.533	0.442	0.621	0.534	[22 6] [19 16]
Random Forest	0.381	0.619	0.521	0.391	0.655	0.528	[22 19] [5 17]
SVM	0.413	0.587	0.521	0.413	0.593	0.521	[25 8] [14 23] [18 12] [5 21]
MLP	0.460	0.667	0.551	0.446	0.679	0.550	[22 6] [15 20]
GB Tree	0.460	0.651	0.544	0.457	0.636	0.547	[27 8] [14 14]

Table 45: Results of predicting grade 0 without Job ID

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.556	0.730	0.626	0.457	0.644	0.537	[40 7] [33 15] [10 6] [6 9]
Random Forest	0.571	0.730	0.663	0.422	0.600	0.514	[44 1] [39 6] [16 2] [12 6]
SVM	0.603	0.778	0.689	0.460	0.547	0.503	[48 2] [47 3] [12 1] [11 2]
MLP	0.540	0.730	0.642	0.422	0.670	0.518	[46 0] [34 9] [17 0] [9 11]
GB Tree	0.492	0.730	0.636	0.428	0.615	0.510	[42 4] [37 3] [13 4] [16 7]

Table 46: Results of predicting grade 1 without Job ID

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.571	0.746	0.648	0.360	0.657	0.517	[42 8] [40 12] [8 5] [5 6]
Random Forest	0.651	0.825	0.741	0.435	0.576	0.510	[50 1] [46 5] [10 2] [9 3]
SVM	0.683	0.873	0.778	0.422	0.500	0.495	[55 1] [7 0]
MLP	0.635	0.905	0.770	0.439	0.504	0.495	[57 0] [42 6] [6 0] [13 2]
GB Tree	0.635	0.762	0.714	0.429	0.565	0.513	[47 4] [45 5] [11 1] [10 3]

Table 47: Results of predicting grade 2 without Job ID

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.444	0.635	0.543	0.443	0.634	0.543	[22 12] [11 18]
Random Forest	0.397	0.635	0.550	0.398	0.635	0.553	[21 12] [11 19]
SVM	0.460	0.667	0.529	0.459	0.668	0.540	[22 12] [9 20]
MLP	0.381	0.571	0.504	0.500	0.500	0.500	[36 0] [27 0]
GB Tree	0.444	0.619	0.542	0.450	0.622	0.543	[25 6] [18 14]

Table 48: Results of predicting grade 0 with new features

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.540	0.667	0.599	0.428	0.624	0.534	[38 6] [30 13] [15 4] [9 11]
Random Forest	0.587	0.730	0.660	0.452	0.594	0.523	[43 5] [36 2] [12 3] [19 6]
SVM	0.603	0.778	0.689	0.460	0.547	0.503	[48 2] [39 6] [12 1] [12 6]
MLP	0.635	0.810	0.697	0.500	0.500	0.500	[51 0] [44 0] [12 0] [19 0]
GB Tree	0.524	0.714	0.619	0.424	0.600	0.513	[40 7] [38 3] [11 5] [16 6]

Table 49: Results of predicting grade 1 with new features

Algorithm	Accuracy			ROC Score			Conf Matrix
	Min	Max	Average	Min	Max	Average	
Decision Tree	0.571	0.746	0.651	0.445	0.678	0.529	[41 5] [37 13] [11 6] [5 8]
Random Forest	0.635	0.810	0.721	0.438	0.583	0.498	[50 3] [45 0] [9 1] [15 3]
SVM	0.651	0.857	0.757	0.500	0.500	0.500	[54 0] [41 0] [9 0] [22 0]
MLP	0.667	0.841	0.775	0.500	0.500	0.500	[53 0] [47 0] [10 0] [16 0]
GB Tree	0.603	0.778	0.710	0.408	0.650	0.522	[44 8] [6 5]

Table 50: Results of predicting grade 2 with new features

8.3 Feature Importances

Feature	Importance %
Area of Study	17.87
Job ID	12.26
Specification of Study	11.66
num langs	11.16
Highest Education Achieved	9.04
linkedin	7.21
distance	5.77
uni pref bool	4.38
early professional	3.59
uni rank bool	3.30

Table 51: Feature importances for decision tree model in results 42 (grade 0 with Job ID)

Feature	Importance %
Job ID	16.02
Specification of Study	14.43
num langs	9.84
Area of Study	9.12
Highest Education Achieved	7.93
distance	6.17
uni pt bool	4.17
uni lisboa bool	3.03
uni pref bool	2.91
corporate site	2.70

Table 52: Feature importances for Random Forest model in results 42 (grade 0 with Job ID)

Feature	Importance %
Job ID	20.34
Specification of Study	18.72
Area of Study	17.56
Highest Education Achieved	8.76
num langs	4.60
uni pt bool	4.40
early professional	3.58
linkedin	2.66
uni pref bool	2.40
distance	2.26

Table 53: Feature importances for GB Tree model in results 42 (grade 0 with Job ID)

Feature	Importance %
Job ID	17.47
Specification of Study	16.62
num langs	10.94
Area of Study	10.41
Highest Education Achieved	9.94
linkedin	6.29
experienced professional	5.47
corporate site	4.44
distance	3.64
early professional	3.24

Table 54: Feature importances for decision tree model in results 43 (grade 1) with Job ID

Feature	Importance %
Job ID	14.20
Specification of Study	12.45
Area of Study	12.33
num langs	10.94
Highest Education Achieved	9.84
distance	5.72
uni pt bool	4.05
job board	3.47
corporate site	2.69
uni lisboa bool	2.66

Table 55: Feature importances for Random Forest model in results 43 (grade 1 with Job ID)

Feature	Importance %
Specification of Study	18.68
num langs	13.69
Highest Education Achieved	12.25
Job ID	11.26
Area of Study	9.98
uni pt bool	4.36
Certification Licences	3.55
corporate site	3.46
uni lisboa bool	3.41
linkedin	2.56

Table 56: Feature importances for GB Tree model in results 43 (grade 1 with Job ID)

Feature	Importance %
Specification of Study	15.98
num langs	14.57
Highest Education Achieved	11.71
Area of Study	9.67
Job ID	7.75
job board	7.20
distance	5.08
corporate site	4.65
uni lisboa bool	4.64
Certification Licences	2.98

Table 57: Feature importances for decision tree model in results 44 (grade 2 with Job ID)

Feature	Importance %
Specification of Study	15.00
Job ID	14.31
Highest Education Achieved	11.24
num langs	9.22
Area of Study	9.08
Certification Licences	5.20
experienced professional	4.01
distance	3.84
uni pref bool	3.72
uni pt bool	3.71

Table 58: Feature importances for Random Forest model in results 44 (grade 2 with Job ID)

Feature	Importance %
Specification of Study	19.75
Area of Study	18.60
Highest Education Achieved	15.67
Job ID	12.83
Certification Licences	5.81
distance	4.85
uni rank bool	3.84
num langs	3.34
linkedin	2.90
job board	2.00

Table 59: Feature importances for GB Tree model in results 44 (grade 2 with Job ID)

Feature	Importance %
Specification of Study	23.77
Highest Education Achieved	14.42
Area of Study	13.11
num langs	12.98
uni pt bool	4.30
distance	4.12
mid-level professional	3.43
glassdoor	3.28
uni pref bool	2.51
Certification Licences	2.49

Table 60: Feature importances for decision tree model in results 45 (grade 0 without Job ID)

Feature	Importance %
Specification of Study	14.19
Area of Study	13.71
num langs	11.36
Highest Education Achieved	10.45
distance	7.85
uni pt bool	4.78
early professional	4.22
linkedin	3.67
experienced professional	3.36
professional	2.73

Table 61: Feature importances for Random Forest model in results 45 (grade 0 without Job ID)

Feature	Importance %
Area of Study	24.36
Specification of Study	17.71
num langs	12.10
Highest Education Achieved	8.27
distance	6.33
early professional	3.99
glassdoor	3.60
uni lisboa bool	3.39
uni pref bool	2.91
uni pt bool	2.78

Table 62: Feature importances for GB Tree model in results 45 (grade 0 without Job ID)

Feature	Importance %
Specification of Study	24.96
Area of Study	11.49
num langs	10.02
Highest Education Achieved	8.26
distance	6.36
uni pt bool	5.61
uni pref bool	3.89
corporate site	3.26
early professional	3.20
professional	3.13

Table 63: Feature importances for decision tree model in results 46 (grade 1 without Job ID)

Feature	Importance %
Area of Study	14.22
Specification of Study	14.01
num langs	12.75
Highest Education Achieved	12.41
distance	7.01
uni lisboa bool	4.19
uni pt bool	3.68
early professional	3.48
corporate site	2.77
job board	2.76

Table 64: Feature importances for Random Forest model in results 46 (grade 1 without Job ID)

Feature	Importance %
Specification of Study	21.51
Area of Study	15.68
Highest Education Achieved	12.72
num langs	9.97
trainee	4.02
linkedin	3.51
uni pref bool	3.44
uni lisboa bool	3.37
corporate site	3.30
recent graduate	3.27

Table 65: Feature importances for GB Tree model in results 46 (grade 1 without Job ID)

Feature	Importance %
Specification of Study	14.15
num langs	13.14
Highest Education Achieved	10.12
distance	9.84
Certification Licences	7.98
Area of Study	7.51
uni pt bool	6.15
uni pref bool	5.42
linkedin	4.63
experienced professional	3.69

Table 66: Feature importances for decision tree model in results 47 (grade 2 without Job ID)

Feature	Importance %
Specification of Study	18.78
Highest Education Achieved	13.98
Area of Study	11.63
num langs	10.17
distance	6.58
linkedin	4.17
early professional	3.47
uni pref bool	3.14
experienced professional	3.08
uni lisboa bool	2.88

Table 67: Feature importances for Random Forest model in results 47 (grade 2 without Job ID)

Feature	Importance %
Specification of Study	21.92
Highest Education Achieved	15.70
num langs	11.69
Area of Study	11.63
Certification Licences	7.58
distance	4.78
recent graduate	3.10
uni pref bool	3.02
is polig	2.81
uni rank bool	2.68

Table 68: Feature importances for GB Tree model in results 47 (grade 2 without Job ID)