Bachelor's Thesis

PREDICTION OF SUCCESSFUL ENTREPRENEURS USING MACHINE LEARNING

Leonidas Vokos

UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

April 2020

Declaration of Authorship

I, LEONIDAS VOKOS , hereby declare that I am the sole author of this Bachelor Thesis titled, 'Prediction of successful entrepreneurs using Machine Learning' and that I have not used any sources other than those listed in the bibliography and identified as references. I further declare that I have not submitted this thesis at any other institution in order to obtain a degree. This Bachelor Thesis was submitted for partial fulfillment of the requirements for obtaining the degree of Computer Science of the Department of Computer Science of the University of Cyprus, under the supervision of Assistant Professor, Mr.George Pallis.

Signed: ______________________________

Date: ______________________________

April 2020

Acknowledgments

I would like to express my heartfelt appreciation to my supervisor Mr.George Pallis, Assistant Professor in the Computer Science Department of University of Cyprus, for giving me an opportunity to work on a such trend topic thesis. He gave me all the support and encouragement throughout the elaboration of my thesis project.

Moreover, I would like to thank Mr.Dimosthenis Stefanidis, researcher in the Computer Science Department of University of Cyprus, for his excellent guidance and the support he gave me in getting familiar with the thesis topic and helped me learn an incredible amount of new technologies. His instructions and suggestions were very important in the completion of this research and being far more experienced than me, he willingly availed me out with his help and daily support.

Finally, I would like to acknowledge and thank my family, specifically my parents Vasilis and Sofia for all the support they provided to me during my undergraduate studies.

Abstract

Undoubtedly nowadays everybody owning a business of any kind considers themselves an entrepreneur and, in most cases, an entrepreneur with a bright future on the funding receiving aspect. Even though, most of them are new entrepreneurs, they believe that their startups will survive for a long time based on the idea that they will manage to ensure all the needed future funding for their startups until they end up being ‘a big waste of money and time’ as many of the unexperienced ones would say, or ‘a good lesson learned’ as other more experienced would say. Who is right and why is a small part of this study’s results.

So how can new entrepreneurs ensure future funding? Despite the bad news we so often hear about the number of small businesses failing, the news really isn't all that bad: Thousands of small businesses startup every year and a good percentage of those startup’s entrepreneurs have learned what it really takes to survive the early startup years and how to ensure the funding that will be needed in the future.

We try to predict whether entrepreneurs will receive a funding or not based on specific information about them. After working with a Crunchbase dataset of entrepreneurs, which also included information from other studies, we discovered that the ones who successfully received funding share some common traits and so do the ones who did not.

Despite the many tries and approaches taken by a lot, due to the problem’s complexity of what is considered a successful funding receival, when and who can receive that funding and the missing and complex data, there does not exist a deterministic result in which we can refer as correct and accurate prediction. We, in order to keep things simple, define a successful funding receival as more than 0 existing funding rounds or in other words at least an existing funding to have been received.

Then, we select all the entrepreneurs from the dataset, as non-entrepreneurs are included, and remove several characteristics that have an immediate relation with this funding, in order to help with the creation of a good prediction model based on these data. Following up, we convert all the alphabetic format data to numerical format and try different

approaches to predict the missing values or to delete them in order to find the best outcome.

Furthermore, we scale our final data and then begin the dynamic parameter hyper-tuning for 9 different machine learning algorithms, including Neural Network. With the use of Over-Sampling and Under-Sampling strategies we ensure that the training phase on the next step will be made upon a well-balanced dataset.

Finally, we receive some input requested by the user and based on each machine learning algorithm we return our predictions, followed by a prediction probability, whether he or she will successfully receive a funding or not. We use F1 score and accuracy as metric scores, but F1 score is our primary metric.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Motivation

Starting a business can be terrifying. Many startup myths threaten to hold back even the best-intentioned entrepreneurs. Based on (Bowman,2020)[1] the statistics do not do much for confidence: 20 percent of new entrepreneurs' startups fail in their first year and only 50 percent survive through their fifth year. Despite of those discouraging numbers, today there are close to 400 million new entrepreneurs worldwide. Many people looking to start a business hesitate because they don't know what it will take to get started. Based on "Small Business Trends" magazine, and more exactly on (Mansfield, 2019) [12] of all small businesses started in 2014: 80% made it to the second year (2015), 70% made it to the third year (2016), 62% made it to the fourth year (2017) and 56% made it to the fifth year (2018) as shown in figure 1.1. 29% of them ran out of financial resources as shown in figure 1.2 and based on figure 1.3 almost 77% of them had their activity based upon personal funds.

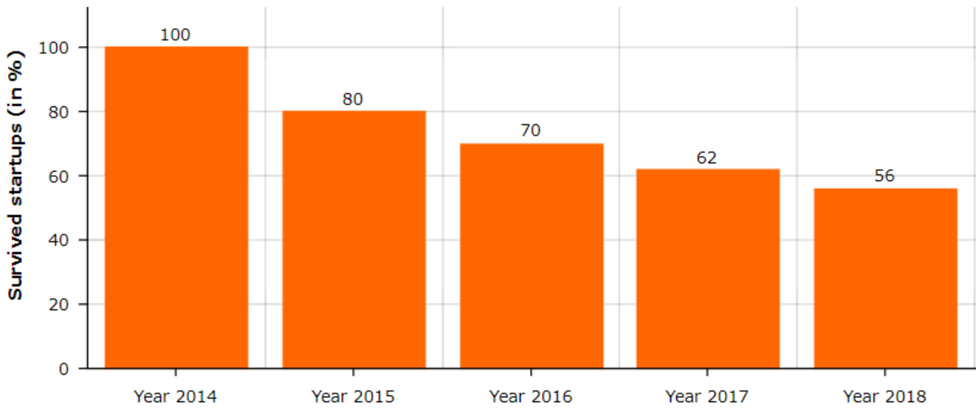

Figure 1.1: Survival rate of new startups over years

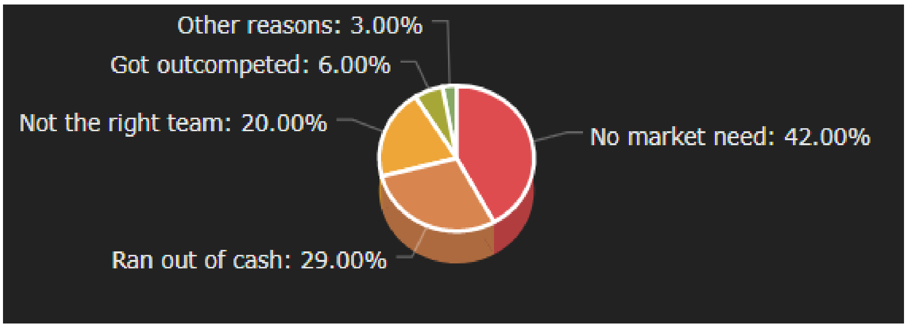

Figure 1.2: Failure reasons of new startups

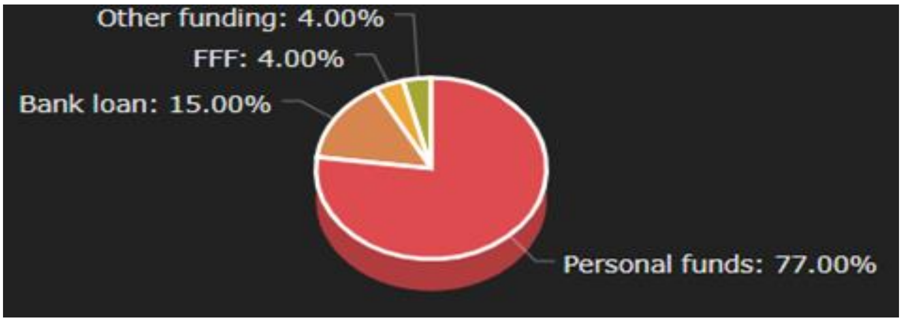

Figure 1.3: Fund resources of new startups

Founders of a previously successful business have a 30% chance of funding receival success with their next venture, founders who have failed at a prior business have a 20% chance of succeeding versus an 18% chance of funding receival success for first time entrepreneurs. But, while the failure rates for new startups are high, business failure rates are actually in a pattern of long-term decline. The rate of entrepreneurs in the US failing has fallen by 30 percent since 1977. (Shane, 2016) [18]

(Chattopadhyay and Ghosh, 2002) [15] identify the potential of entrepreneurship as a tool to create a dynamic economy that has been increasingly recognized in most developing countries. In India, after independence the entrepreneurship power was recognized. It has been almost 50 years that some small-scale industrial programs arose in order to create economic development, but the funding receival success of these efforts has not been satisfactory.

So, what could be the main reason of these unsuccessful funding receival despite the entrepreneurship efforts? Maybe the surrounding environment, maybe the social factor or even the entrepreneur by himself? This is a question to be answered through this study.

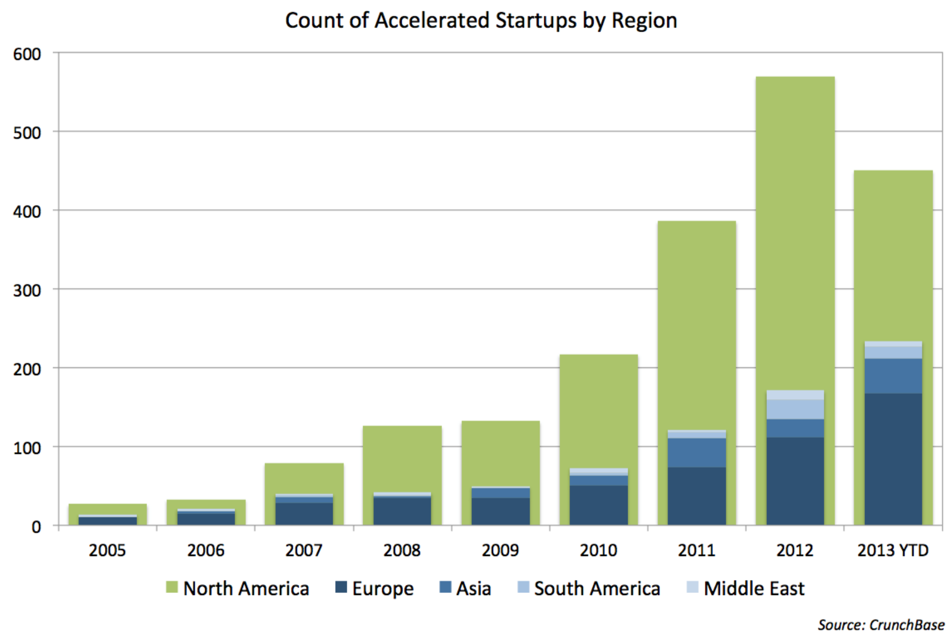

Figure 1.4: Count of Accelerated Startups by Region

So, given this development of entrepreneurship worldwide, such as the yearly accelerated startups by region shown in figure 1.4 above, we can use it to study how entrepreneurs behave and what characteristics they have that make them earn the trust of investors and receive a funding. In that way we can see the similarities of all entrepreneurs', who successfully received a funding, characteristics from the beginning of their careers and find if any of those characteristics had impact on that successful funding receival. This would help new entrepreneurs with their startups and based on their current characteristics we could inform them on the probabilities of receiving a funding or not based on other successful entrepreneurs who managed to succeed in the funding receival aspect.

1.2 Challenges

By definition, entrepreneurial funding receival success is a challenging issue from every perspective. First of all, who can successfully receive an important funding and what an important funding is considered? Many papers and articles show different information based on experience, characteristics and social media presence, of which entrepreneurs have more chances of successfully receiving a funding. But there does not exist a correct generally accepted answer to this question.

(Angel, Jenkins and Stephens, 2018) [14] mention that entrepreneurship research has focused on different conceptions of what entrepreneurial success of funding receival means and the factors that help predict it, but yet failed to find out which entrepreneurs have the potential of receiving a funding in a general way. When entrepreneurial funding receival success has been studied at the individual level, it was tried to identify common funding receival success criteria and examine the importance of these criteria to the entrepreneur, but it was very possible that entrepreneurs may have had different conceptions of these criteria and this could influence how entrepreneurs developed their own startups, whether they were successful in receiving a funding or not.

When analysts have tried to understand what funding receival success meant to entrepreneurs, their main goal was to identify the most common criteria that entrepreneurs usually used to define this funding receival success, such as personal satisfaction and wealth gaining, and then to understand the importance entrepreneurs give on these criteria (Orser & Dyke, 2009; Wach et al., 2016; Gorgievski et al., 2011; Fisher et al. 2014)[3,5,11,16]. While these studies were concentrated on mainly the funding receival success of the individual entrepreneurs and not on the firm's success of funding receival, as most studies have previously done, they still do not manage to fully explain which entrepreneurs have the potential of receiving a funding.

As (Advisors to the Ultra-Affluent - Groco, 2019) [7] identifies “everybody wants to become a ‘successful entrepreneur’ but what makes them successful is a mystery of their mind.” Until this day there has not been an explanation of which entrepreneurial characteristics are most important when it comes to successfully receiving a funding.

In addition, the collection of data of entrepreneurs, from CrunchBase, is another challenging part because in the characteristics included many missing fields exist which do not help at the prediction phase and a wrong prediction of these missing data would result in a wrong final prediction outcome. For example, a wrong prediction of a lot of missing characteristics could result in a one-sided funding receival success result, thing that would impact the training phase of machine learning algorithms and the prediction at general.

Along with the previous challenges, we had to deal with some technical restrictions. As we will mention later, some algorithms like Neural Network required libraries which had many conflicts with libraries already used for other machine learning algorithms, so we had to test these algorithms in different script executions.

1.3 Contributions

The ultimate goal of our research is to examine how specific data will help us predict the success of an entrepreneur receiving a funding or not. Therefore, we analyze a lot of data from CrunchBase database, including information from other studies too (Nicolaou N. et al.; Shane S. et al.) [13,19], and the impact of each feature on the final prediction result. We use a lot of methods to preprocess the data, which suffers from missing fields and with the use of many machine learning algorithms and Neural Network we analyze the best prediction of them. Furthermore, we implement some very effective automated parameter hyper-tuning for the algorithms’ parameters. In addition, we examine the impact of each missing field’s prediction or deletion method on missing data and combined by other data preprocessing strategies we compare our outcomes.

Moreover, we have the honor to contribute in this trend issue of entrepreneurial funding receival success prediction.

To sum up, our contributions are as follows:

- Examine and preprocess the big data of the CrunchBase database and its' features.
- Based on that information try to make a prediction on entrepreneurial success on receiving a funding or not.
- Extract specific characteristics of entrepreneurs who successfully were predicted to receive a funding.

Our contribution to entrepreneurial funding receival success prediction using machine learning must be effective and help the investigations on this issue.

1.4 Outline Contents

Chapter 1. Introduction
In the introduction chapter, we briefly present how entrepreneurship expanded nowadays and how many new entrepreneurs successfully receive a funding or not. Furthermore, we mention the challenges of this topic, mainly the definition of entrepreneurial funding receival success and big data challenges. Lastly, we explain the contribution of our research to that area and what we want to export as output.

Chapter 2. Literature & Related work
In the second chapter, we mainly focus on analyzing literature work on entrepreneurial funding receival success prediction area and their results. Moreover, we also study and analyze other work done on similar topics such as characteristics of entrepreneurs that have the potential of receiving a funding and we explain how their work is similar to ours and how we will adjust it to our needs.

Chapter 3. Methodology
The methodology chapter defines our methodology and explains each step we made very precisely. We described how we find our dataset and the role of it in our study. Moreover, we analyze the information of the dataset and describe the whole preprocessing phase. Then we present how we used the processed data in combination with the machine learning algorithms in order to make a correct training phase, as well

as the parameter hyper-tuning of each machine learning algorithm and Neural Network. Finally, we try to make the best possible prediction of user's input with the most precise accuracy.

Chapter 4. Evaluation

In chapter 4 we discuss the way we evaluate our results. We compare the machine learning algorithms results, in each step, together in order to choose the best results and confirm them by comparing with already existing data's results. Furthermore, we extract the characteristics of the entrepreneurs who were predicted to receive a funding.

Chapter 5. Conclusion

In this final chapter we briefly describe how our results can be used into further analysis on entrepreneurial funding receival success prediction topic and the future work that can be done in order to achieve higher prediction score of our machine learning models.

Chapter 2

Literature & Related Work

Contents

2.1 Machine Learning Prediction of Companies' Business Success

Prediction of new entrepreneurs or new start-ups in whether they have the potential of receiving a next big funding or not has earned a lot of attention the last years. There has been a rapid growth in the number of new entrepreneurs and predicting their funding receival success is a very important and interesting task. Finding out what makes some entrepreneurs become more successful in receiving funding and some others not is very important for the investors, in order to decide on whether to invest or not to a new startup.

As Machine Learning becomes a popular tool nowadays we are trying to use it in combination with some important information about the funding receival success of entrepreneurs and make predictions for the success of a new entrepreneur in receiving funding or not, based on the given data.

Chenchen P., Yuan G. and Yuzi L. (Cs229.stanford.edu, 2018) [4] have done a similar work to ours, but with the intention of predicting companies' funding receival success and not entrepreneurs'. They state in their research "Machine Learning Prediction of Companies' Business Success", that they used data from Crunchbase to build a machine learning model through supervised learning in order to predict which start-ups have the potential of being successful with their next venture. They explored K-Nearest Neighbors (KNN) model on this task and compared it with Logistic Regression (LR) and Random Forests (RF) model in previous works. They used F1 score as the metric and found that KNN model had a better performance on this task, which achieved 44.45% of F1 score and 73.70% of accuracy.

Bento (Run.unl.pt, 2018) [6] and (Xiang et al. 2012) [8] have also experimented upon CrunchBase data. Bento built a Random Forests model to predict which start-ups have success in receiving funding and which do not using M&A or metrics from financial reports. The model they built to predict whether a company would be successful or not successful based on the funding receival had a True Positive Rate (TPR) of 94.1% (the highest reported using data from CrunchBase) and a False Positive Rate of 7.8%. Xiang [8] used CrunchBase data together with profiles and news articles from TechCrunch to predict company acquisitions. (Liang and Yuan, 2012) [10] tried to find general rules for companies seeking investment, involving investors' preference to invest using descriptive data mining with CrunchBase. (Liang and Yuan, 2016) [22] used social network features to build a prediction model based on Crunchbase data. Some other analysts, like (Wei et al. 2008) [20] focused more on M&A events prediction. Also, based on the publication of (Yang and Berger, 2017) [21], "Relation between start-ups' online social media presence and fundraising", it is explained that new start-up companies were able to benefit from communicating on social media platforms. Start-ups, which were active in Facebook and Twitter social media, received larger amount of funding in total. Furthermore, it was examined that as their business expanded, they committed even more into online social networking. It confirmed the idea that businesses are using social media consciously.

Even though most of the studies have the funding receival success prediction of companies as their main topic, these approaches have a significant impact on the prediction of entrepreneurial funding receival success. To clarify, even though we are based on different techniques and steps of studies on companies' funding receival success prediction, in this thesis we are focusing on success prediction of entrepreneurs receiving a funding or not.

2.2 Similar Studies

Besides the prediction of entrepreneurial and companies' funding receival success using Machine Learning there have been made many other studies upon the characteristics of entrepreneurs who successfully received funding or not.

In the paper "Facial Structure and Entrepreneurship", (Nicolaou N. et al.)[13], they discuss how facial characteristics act as a sociable indication that affects the entrepreneur's actions with others and examine whether the fWHR, fWHR-lower, cheekbone prominence and facial symmetry are associated with entrepreneurship engagement. It is stated that as a result the cheekbone prominence and facial symmetry increased the likelihood of entrepreneurship engagement, while the fWHR and fWHR-lower were not associated with entrepreneurship.

Furthermore, in the paper "Entrepreneurship and Emotions", (Shane S. et al.)[19], they discuss whether entrepreneurs are more likely than non-entrepreneurs to exhibit positive emotions. They specifically examined whether social entrepreneurs are more likely than other entrepreneurs to exhibit positive emotions and whether serial entrepreneurs are less likely to exhibit positive emotions. Using a two-study design with four samples they found that entrepreneurs have more positive emotions in contrast to non-entrepreneurs. They further showed that social entrepreneurs experience more positive emotions compared to other entrepreneurs.

All these examined characteristics in the two above papers are included in the dataset in which we are based in order to make our prediction.

2.3 Data Collection

Data Collection in such studies is one of the most critical steps, in getting a complete and accurate prediction.

In the paper (Cs229.stanford.edu, 2018) [4], there has been done similar research to ours, but with companies' characteristics instead of entrepreneurs'. The dataset they used was extracted from Crunchbase Data Export as our dataset, but in contrast of our 600K rows, and its' part of 50K rows, it contained 60K+ companies' information updated to December 2015. There were four data files, named "company", “investments”, “rounds” and “acquisition” to choose from but the “company” file contained most comprehensive information of the companies, while other files contain more detailed information regarding the investment operations, in contrast to our single file containing all the useful characteristics of 600K persons (entrepreneurs and non-entrepreneurs included).

At first our dataset had a lot of data that could be used to predict the success of an entrepreneur receiving funding or not, but not all of it could be used. In the 50K rows dataset sample, a part of the 600K rows dataset, only 11,427 were entrepreneurs that could be used for training our models and most of them had more than 40%-50% missing values in the features given. So, after the preprocessing and the selection of useful features we ended up with 3.2K rows of data from the 50K dataset. As for the 600K rows dataset, 156.7K were entrepreneurs and after the preprocessing we ended up with 42.4K rows of useful prediction data.

Chapter 3

Methodology

Contents

3.1 Methodology Overview

Our research methodology is built upon 4 important pillars: Data Collection, Data Analysis and Preprocessing, Feature Selection, and the machine learning models' training and testing. An overview of our architecture is shown in Figure 3.1.

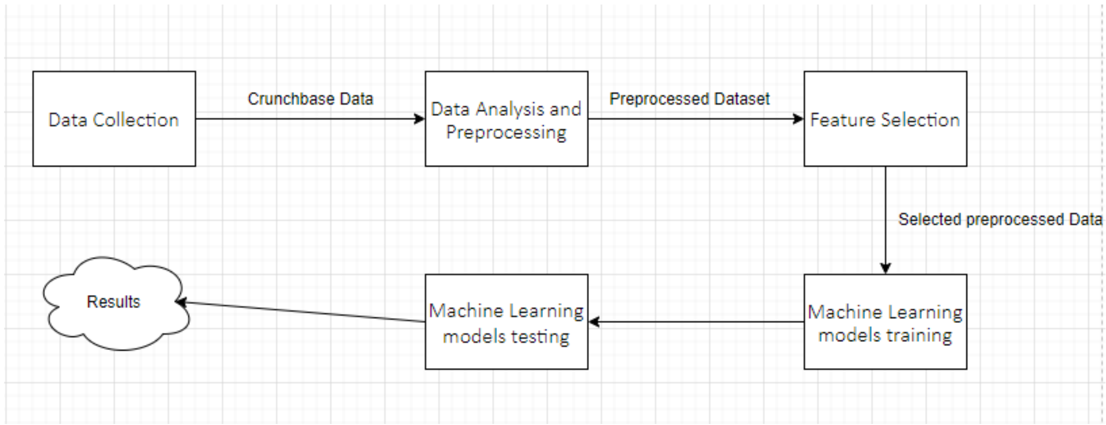

Figure 3.1: An overview of our methodology

Firstly, we had to collect our data. Then, we had to examine our data and preprocess it by handling the empty values, the alphabetical features, the outliers, scaling them, and selecting only the most essential features. In this research, we are focusing on training the different machine learning models and choosing the best parameters for them. By achieving these steps, we will increase the prediction score of the models. In conclusion, we want to use these models in order to predict correctly whether a person will receive a funding or not, based on some information that the person will give about himself.

3.2 Data Analysis

After collecting an efficient amount of data from Crunchbase dataset and joining these data with the emotion and face characteristics of other studies (Nicolaou N. et al.; Shane S. et al.)[13,19] we proceed in our analysis. Our main goal is the prediction of receiving a funding or not, only for the entrepreneurs. Our initial dataset includes also information of non-entrepreneurs in it. So, in order to make a correct data analysis for the entrepreneurs we remove all the non-entrepreneurs from the dataset and we are left with 11K data from the initial 50K, part of 600K rows original dataset. An overview of the entrepreneurs’ data is shown below in figures 3.2, 3.3, 3.4, 3.5, 3.6 and 3.7.

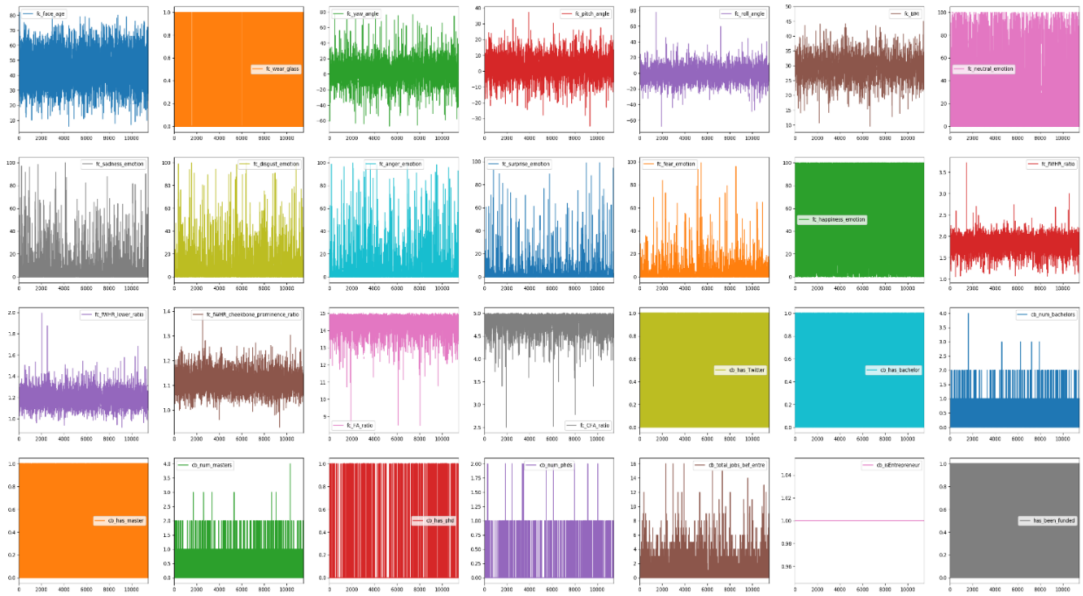

Figure 3.2: An overview of the entrepreneurs’ numerical data

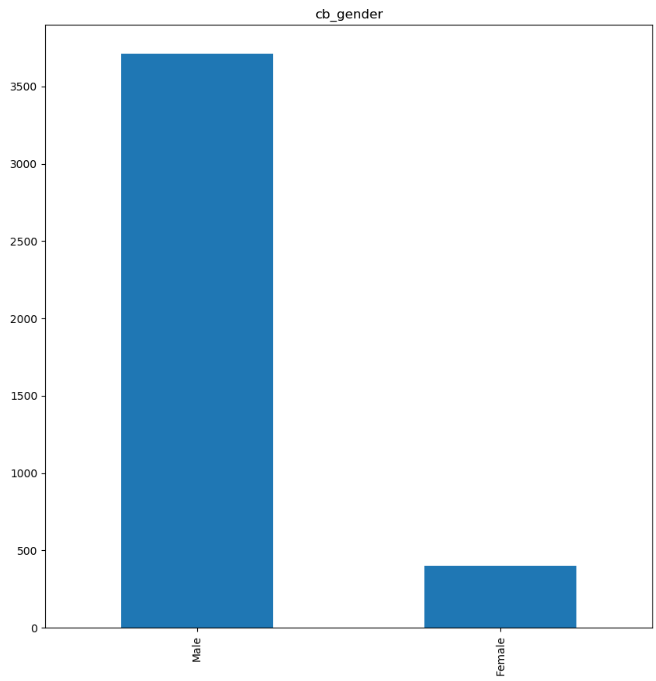

Figure 3.3: An overview of the entrepreneurs’ gender

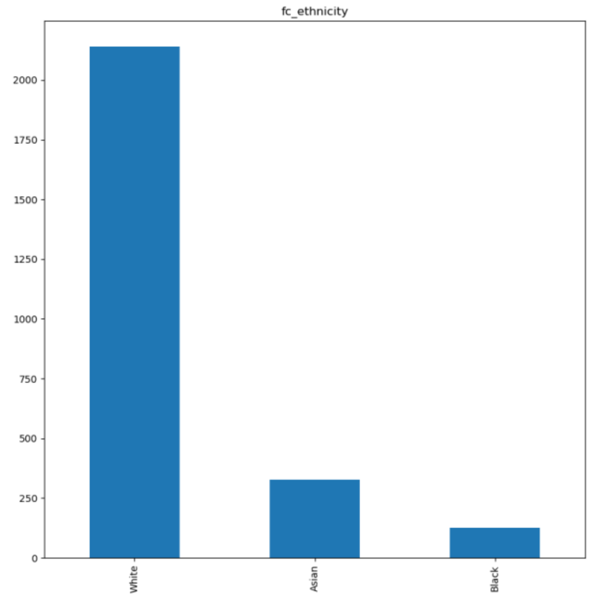

Figure 3.4: An overview of the entrepreneurs’ ethnicity

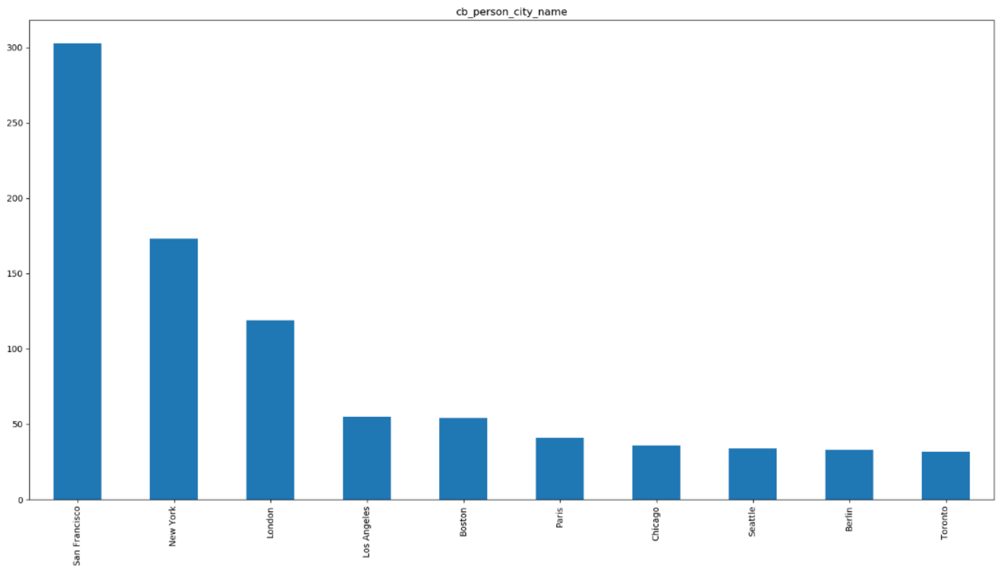

Figure 3.5: An overview of the entrepreneurs’ top 10 cities

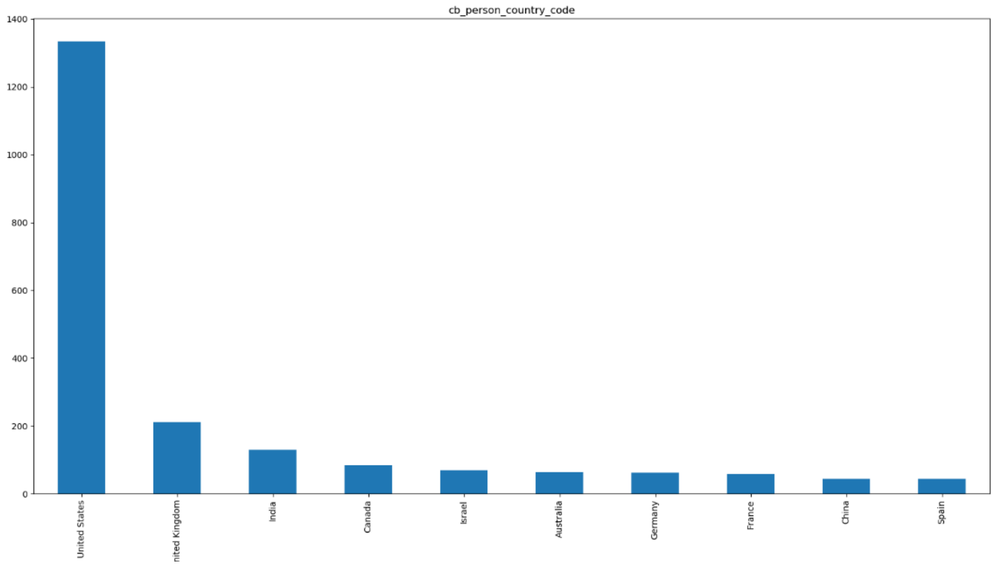

Figure 3.6: An overview of the entrepreneurs’ top 10 countries

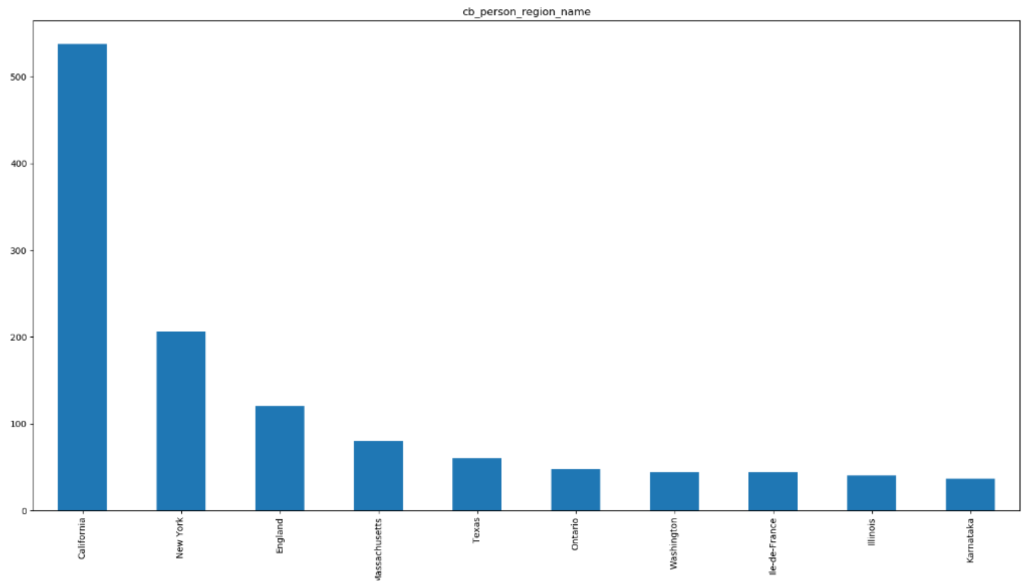

Figure 3.7: An overview of the entrepreneurs’ top 10 regions

From the above figures we noticed that there was a lot of missing data both on the numerical and alphabetical features as the sum of the plotted bars on the y-axis metrics range did not reach the expected 11K sum for all the entrepreneurs. So, our next step was to visualize these missing data.

3.2.1 Dataset Overview

The dataset consists the following columns:
person_uuid: a special user id for each person (alphanumeric format)
fc_face_age: each person's age (integer)
fc_wear_glass: if a person wears glasses or not (0 = doesn't wear glasses, 1 = wears)
fc_ethnicity: the ethnicity of the person (in string format)
fc_yaw_angle: the metric of the yaw's angle (in float format)
fc_pitch_angle: the metric of the pitch's angle (in float format)
fc_roll_angle: the metric of the roll's angle (in float format)
fc_BMI: the BMI metric of the person (in float format)
fc_neutral_emotion: the metric of neutral emotion from 0-100 (in float format)
fc_sadness_emotion: the metric of sadness emotion from 0-100 (in float format)
fc_disgust_emotion: the metric of disgust emotion from 0-100 (in float format)
fc_anger_emotion: the metric of anger emotion from 0-100 (in float format)
fc_surprise_emotion: the metric of surprise emotion from 0-100 (in float format)
fc_fear_emotion: the metric of fear emotion from 0-100 (in float format)
fc_happiness_emotion: the metric of happiness emotion from 0-100 (in float format)
fc_fWHR_ratio: the metric of face's width and height ratio (in float format)
fc_fWHR_lower_ratio: the metric of lower face's width and height ratio (in float format)
fc_fWHR_cheekbone_prominence_ratio: the metric of cheekbone's width and height ratio (in float format)
fc_FA_ratio: the FA metric ratio (in float format)
fc_CFA_ratio: the CFA metric ratio (in float format)
cb_gender: the person's gender (in string format)
cb_born_on: the person's birthday (in string format e.g. '1/1/1977')
cb_person_country_code: the origin country of the person (in string format)

cb_person_region_name: the region of the person (in string format)

cb_person_city_name: the current city of the person (in string format)

cb_has_Twitter: if a person has Twitter account or not (0 = doesn't have, 1 = has)

cb_has_bachelor: if a person has bachelor's degree or not (0 = doesn't have, 1 = has)

cb_num_bachelors: the number of bachelor's degrees a person has (integer)

cb_has_master: if a person has master's degree or not (0 = doesn't have, 1 = has)

cb_num_masters: the number of master's degrees a person has (integer)

cb_has_phd: if a person has PhD degree or not (0 = doesn't have, 1 = has)

cb_num_phds: the number of PhD degrees a person has (integer)

cb_total_jobs_bef_entre: the number of jobs a person had before becoming entrepreneur (integer)

cb_isEntrepreneur: whether a person is entrepreneur or not (0 = is not, 1 = is)

num_founded_companies: the number of companies the person founded (integer)

num_closed_companies: the number of companies owned by the person and closed (integer)

num_sold_companies: the number of companies owned by the person and sold (integer)

longest_survival_years_founded_companies: the maximum number of years a company founded by the person survived (integer)

num_funding_rounds: the number of funding rounds that a person received (integer)

num_missing_rounds: the number of funding rounds that a person missed (integer)

total_funding_amount: the total amount in US dollars that the person received from fundings (integer)

num_companies_with_ipo: the number of the companies with IPO that the person owned/owns (integer)

biggest_ipo_founded_companies: the biggest IPO that a company owned by the person ever had (integer)

'person_uuid' was not included in the below data analysis, considering it was a unique id for every row, thing that would not help in the prediction.

3.2.2 Missing data

The dataset included a lot of missing values. In order to analyze the missing data and decide on whether a feature's missing data could be predicted or not we had to visualize it. We decided to not predict features whose missing data was greater than 60% of the data and try to predict the missing values that we did not drop. The initial dataset's missing data was as shown below in figure 3.8.

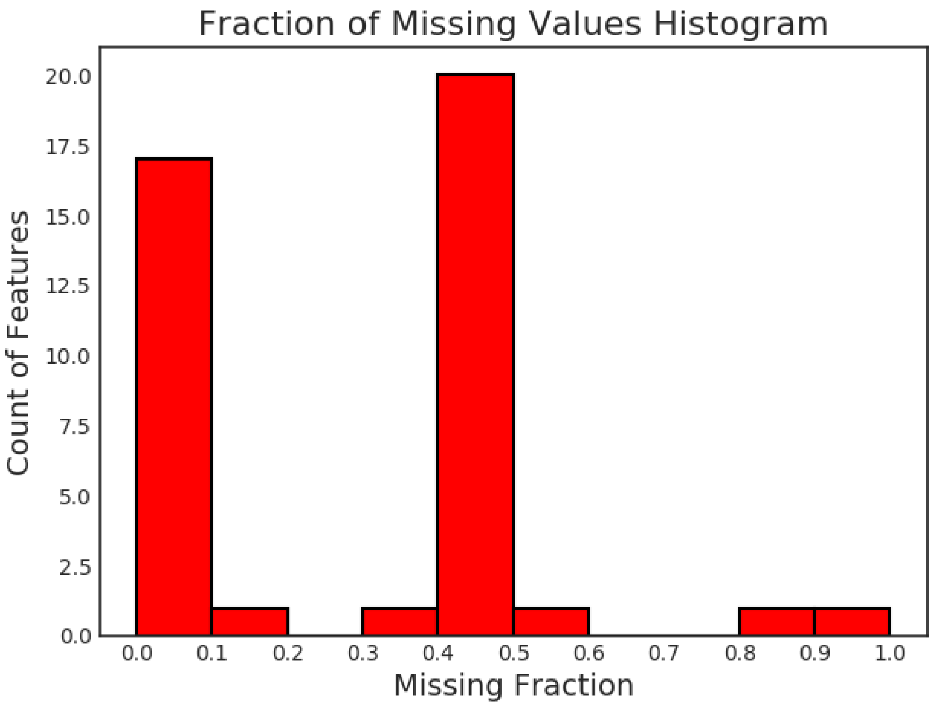

Figure 3.8: An overview of missing data of initial dataset

From the above figure 3.8 we noticed that about 16 features had less than 10% missing values, 1 had between 10-20%, 1 had between 30-40%, 20 features had between 40-50% missing values, 1 between 50-60%, 1 between 80-90% and 1 more than 90%.

The 2 features with missing data greater than 60% were: 'cb_born_on' and 'biggest_ipo_founded_companies'.

While the top 10 features with missing values were:

Features	missing_fraction
biggest_ipo_founded_companies	0.991279
cb_born_on	0.828028
fc_BMI	0.559606
fc_fear_emotion	0.441441
fc_wear_glass	0.441441
fc_CFA_ratio	0.441441
fc_FA_ratio	0.441441
fc_fWHR_cheekbone_prominence_ratio	0.441441
fc_fWHR_lower_ratio	0.441441
fc_fWHR_ratio	0.441441

Table 3.1: Top 10 missing data features

So, the dataset had many problematic missing data mainly between 40-50% on 20 features which we decided to try and predict before dropping those 20 features out of the dataset. As we will explain below the outcome of dropping some of the rows including missing data of these features rather than predicting those, was better.
Our next step was to visualize the unique values of each feature.

3.2.3 Unique data

We wanted to see how many unique data each feature had so we could have a general image over the data description of each feature that would help us later at trying to find the outliers of the dataset.

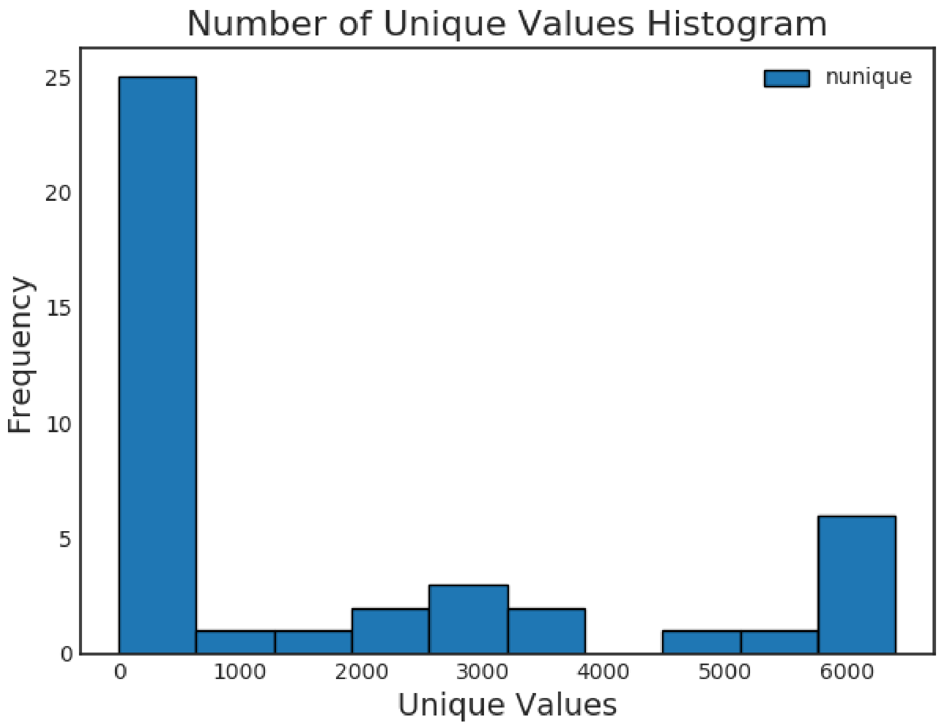

Figure 3.9: An overview of unique data of initial dataset

Based on the above figure we noticed that the unique values were very interesting as more than 5 features had about 6000 unique data and 11 features had about 1000-5000 unique values, while the other 25 features had 0 – 1000 unique values. These meant that the data prediction would be challenging to us, especially for those features with a lot of unique values.

Also, the only feature which had only 1 single unique value was the ‘cb_isEntrepreneur’ feature, which was expected to be like that as far as its’ value was always ‘1’ because of the dataset including only entrepreneurs.

3.2.4 Correlation between features

As for the next step we decided to check the correlation between all the features as well as the correlations between features that were greater than a specific threshold.

So, we found pairs of collinear features based on the Pearson correlation coefficient and if they were above the chosen threshold, we decided to remove one feature of the pair from the dataset.

For a correlation threshold of 0.85 we found the below features:

corr_feature	corr_value	drop_feature
fc_FA_ratio	0.958914	fc_CFA_ratio
cb_has_bachelor	0.965812	cb_num_bachelors
cb_has_master	0.951869	cb_num_masters
cb_has_phd	0.984038	cb_num_phds

Table 3.2: Features with correlation greater than 0.85

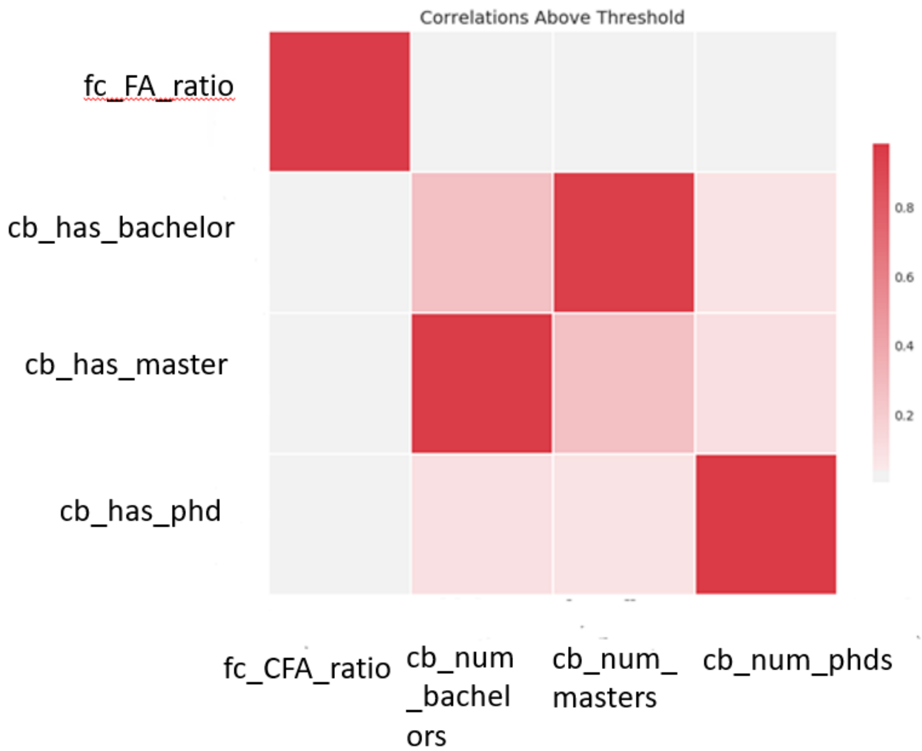

Figure 3.10: Heat-map of features with correlation greater than 0.85

Also, we visualized all the pairs of collinear features based on the Pearson correlation coefficient. This is shown in figure 3.11 below.

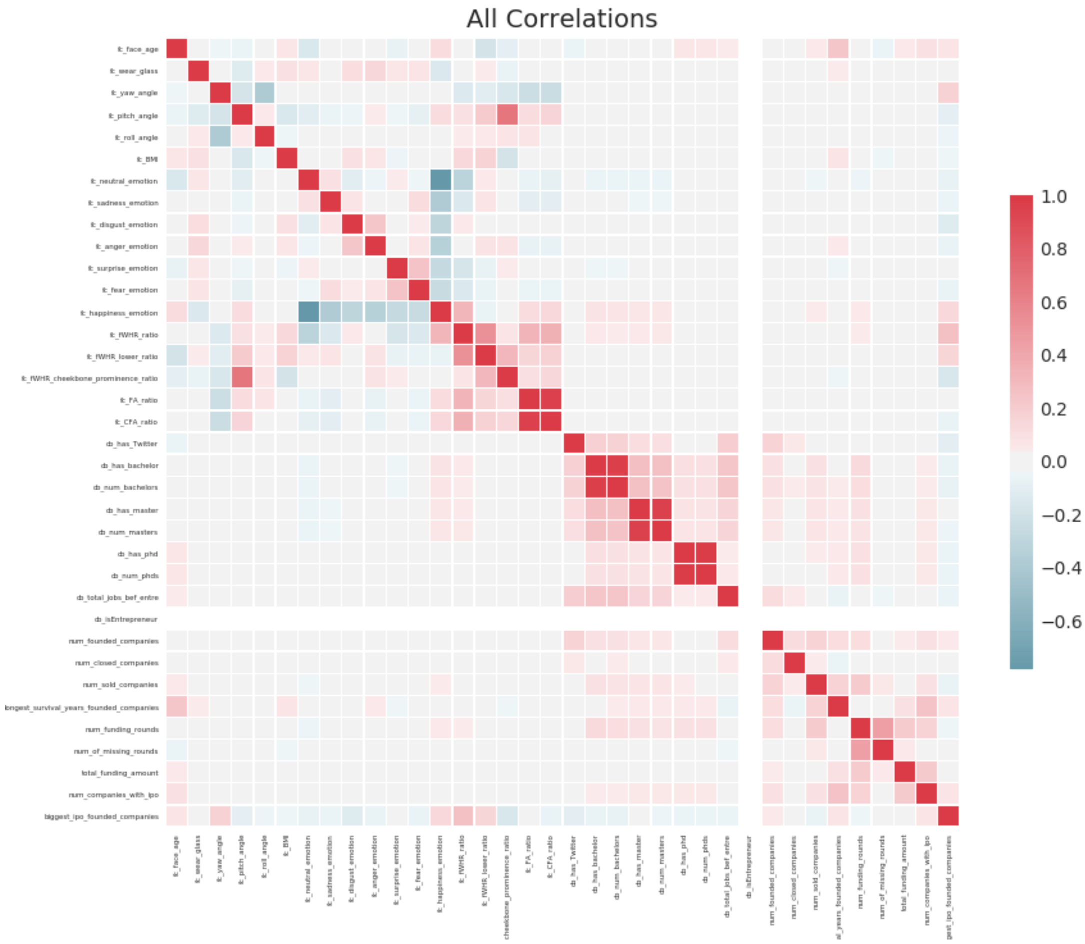

Figure 3.11: Heat-map of all collinear pairs of features

3.3 Data Preprocessing

After analyzing the data, we decided to start the data preprocessing stage. As we stated in the beginning from 50K data we reached to 11K entrepreneurs and after the data preprocessing stage our data dropped to 3.2K. First, we used data analysis results from above to determine the not important features and drop them, as well the obvious not needed features and the dependent features as explained below.

3.3.1 Labeling data

One very important part of getting started was labeling out data. The Crunchbase dataset, including extra data from other studies (Nicolaou N. et al.;Shane S. et al.)[13,19], given to us had unlabeled data, thing that would not help us for the funding receival prediction. So, after a lot of investigation trying to find what an important funding is considered, we ended up with many different answers but as we mentioned before none of them is considered to be accepted. Until this day we cannot say what an important funding is considered. So, what we did was to keep things simple and decided to define as a funding receival success of an entrepreneur the possibility to get a funding in a funding round or not. So, we labeled data from dataset as ‘1’ which meant successfully received funding, if the entrepreneur had at least one num_funding_rounds and ‘0’ which meant not success in receiving funding, if the entrepreneur did not have even one num_funding_rounds.

3.3.2 Drop features

Based on the missing values we dropped the ‘cb_born_on’ and ‘cb_person_region_name’ since the missing data of those features was too much to try and predict.
Also, we continued by dropping the ‘person_uuid’ as far as it was obvious that it was not needed for the prediction of the target feature as far as a specific ID would not help to predict whether someone would successfully get a funding or not.

Finally, we decided on dropping the dependent features of the dataset. As explained by (Sarikas, 2020) [17] the dependent variable (sometimes known as the responding variable) is what is being studied and measured in the experiment or in other words the variables that are highly related to the target value**.** It is what changes as a result of the changes to the independent variable.

So ‘cb_isEntrepreneur’, ‘num_founded_companies’, ‘num_closed_companies’, ‘num_sold_companies’, ‘longest_survival_years_founded_companies’, ‘num_funding_rounds’, ‘num_of_missing_rounds’, ‘total_funding_amount’, ‘num_companies_with_ipo’, ‘biggest_ipo_founded_companies’ were dependent features which had direct relation with the target value and therefore were dropped from the dataset.

3.3.3 Remove outliers

Another important step was the removal of data known as outliers that would affect our final prediction negatively. We noticed from the data description of figures 3.2-3.7 that there were some specific data out of the usual range of the general data. So, as a first step, we decided to remove these data by setting a threshold on each characteristic.

First, we removed all the data where the fc_face_age was less than 18, as usually a person younger that the age of 18 cannot successfully get a big funding. We also removed the persons, whose fc_face_age was on that 1% of the data that was unique from the other, e.g. very old people.

Then based on the data description the normal fc_yaw_angle range was -58 to 75 so every data that did not belong in this characteristic’s range was removed. Also, the 1% unique data of fc_roll_angle was removed.

We also removed data where: fc_BMI was lower than 12, fc_sadness_emotion higher than 98, fc_disgust_emotion higher than 95, fc_fWHR_ratio higher than 3, fc_fWHR_lower_ratio higher than 1.9, fc_FA_ratio lower than 11, fc_CFA_ratio lower than 3, cb_num_bachelors higher than 3, cb_num_masters higher than 3 and finally the 1% of unique data of cb_total_jobs_bef_entre.

We did many tests to see what data ranges were the best to keep and we ended up with the above. We did not remove alphabetical data as outliers, but only numerical.

Fortunately, we soon noticed that all the removed data was affecting the prediction negatively. The reason was that the row containing each outlier on a specific feature also contained information that was a very important part of data for prediction and by removing that row the prediction's F1 score was decreasing by at least 5%.
Finally, we decided not to remove any outliers as it would impact negatively the outcome even in the big dataset of 600K rows.

3.3.4 One-hot encoding

The next step was to handle the alphabetical data of the dataset, such as: fc_ethnicity, cb_gender, cb_person_country_code, cb_person_region_name, cb_person_city_name. In order to make these data useful for prediction we had to convert them to a format acceptable from the model to predict and string was not one of them. So, we used one-hot-encoder method. As Pandas documentation explains, 'get_dummies()' is used to separate each string in the caller series at the passed separator. A data frame is returned with all the possible values after splitting every string. If the text value in original data frame at same index contains the string (Column name/ Split values) then the value at that position is 1 otherwise, 0. So as a result we end up with a new column for each unique row of the one-hot-encoded data and that is why our dataset's columns after this step were about 4K.

3.3.5 Fix missing data

As we mentioned above, we removed from our dataset 2 features with missing values more than 60% of the data. But we still had a dataset with many missing values that could reach up to 50% missing data on 20 features at least.
That meant that we had to try and fix these missing values, either by dropping or predicting them.
Our first try was replacing these missing values with a '-1', thing that would help us keep all the data to train our models, without dropping them, but it was not a good approach as the prediction result had a very low F1 score and accuracy.

After that we tried replacing these missing data with the mean of each feature's data, which resulted in a better approach. This helped us keep all the data needed and it was a good approach as the values that we replaced were not the same in all features, thing that gave us a better prediction considering the metrics. We also tried to replace all the missing data with the median of each feature's data, which had same results as the replacement with mean as far as it kept all the data and the replaced values were not the same in all features.

An even better approach came later, after the use of the Iterative Imputer, which as explained from the sklearn documentation, it models each feature with missing values as a function of other features and uses that estimate for imputation. It does so in an iterated round-robin fashion: at each step, a feature column is designated as output y and the other feature columns are treated as inputs X. A regressor is fit on (X, y) for known y. Then, the regressor is used to predict the missing values of y. This is done for each feature in an iterative fashion, and then is repeated for max_iter imputation rounds. So, we ended up with having all the data to train our models and the replaced missing data were different based on a specific prediction model. We also tried the Simple Imputer technique, but it was not as good as the Iterative Imputer because it is similar to mean and median replacement mentioned above.

Finally, we tried the simplest way of fixing these missing values by dropping all the data rows that included missing values in them. The data left to use for prediction was less than the other techniques that had tried but the prediction's F1 score, which was our main metric, was higher in comparison with the other strategies.

Furthermore, we tried to predict all the missing values using different interpolation techniques such as Akima and Pchip. Akima uses differentiable sub-splines, while Pchip uses monotonic cubic splines to find the values of missing points, but the prediction of these missing values seemed not to depend on a specific function as the results were not better than those of the mean and median replacement.

So, we decided to drop all rows that had missing values and get a better prediction rather than training our models with a lot of data which resulted to be misleading.

All the model's results are shown in the tables 3.3-3.7 below, where different missing values handling techniques are applied. As we mentioned above, the deletion of all rows including missing values had the best performance as shown in table 3.7.

Comparison (with all steps done) by Dropping Na:

Machine Learning Algorithms	precision	recall	f1_score	accuracy
Logistic Regression	0.64	0.64	0.64	0.64
Linear Discriminant Analysis	0.66	0.66	0.66	0.66
K-Nearest Neighbours	0.54	0.54	0.54	0.54
Gaussian Naive Bayes	0.62	0.61	0.6	0.61
Decision Tree	0.5	0.5	0.5	0.5
Support Vector Classification	0.66	0.66	0.66	0.66
Gradient Boosting	0.6	0.6	0.6	0.6
Random Forest	0.59	0.59	0.59	0.59
Neural Network	0.64	0.64	0.64	0.64

Table 3.3: Dropping Na strategy’s model’s metrics

Comparison (with all steps done) by replacing Na with -1:

Machine Learning Algorithms	precision	recall	f1_score	accuracy
Logistic Regression	0.58	0.58	0.58	0.58
Linear Discriminant Analysis	0.58	0.58	0.58	0.58
K-Nearest Neighbours	0.55	0.55	0.54	0.55
Gaussian Naive Bayes	0.52	0.52	0.52	0.52
Decision Tree	0.52	0.52	0.52	0.52
Support Vector Classification	0.59	0.59	0.59	0.59
Gradient Boosting	0.58	0.57	0.57	0.57
Random Forest	0.55	0.55	0.54	0.55
Neural Network	0.58	0.57	0.57	0.57

Table 3.4: Replacing Na’s with -1 strategy’s model’s metrics

Comparison (with all steps done) by replacing Na with mean value of feature:

Machine Learning Algorithms	precision	recall	f1_score	accuracy
Logistic Regression	0.58	0.58	0.58	0.58
Linear Discriminant Analysis	0.58	0.58	0.58	0.58
K-Nearest Neighbours	0.56	0.56	0.56	0.56
Gaussian Naive Bayes	0.52	0.52	0.52	0.52
Decision Tree	0.5	0.5	0.5	0.5
Support Vector Classification	0.58	0.58	0.58	0.58
Gradient Boosting	0.58	0.58	0.58	0.58
Random Forest	0.54	0.54	0.53	0.54
Neural Network	0.59	0.59	0.58	0.59

Table 3.5: Replacing Na's with mean value of feature strategy's model's metrics

Comparison (with all steps done) by replacing Na with Iterative Imputer prediction:

Machine Learning Algorithms	precision	recall	f1_score	accuracy
Logistic Regression	0.61	0.61	0.61	0.61
Linear Discriminant Analysis	0.61	0.61	0.61	0.61
K-Nearest Neighbours	0.54	0.54	0.54	0.54
Gaussian Naive Bayes	0.58	0.58	0.58	0.58
Decision Tree	0.52	0.52	0.52	0.52
Support Vector Classification	0.58	0.58	0.58	0.58
Gradient Boosting	0.58	0.57	0.57	0.57
Random Forest	0.52	0.52	0.52	0.52
Neural Network	0.59	0.59	0.59	0.59

Table 3.6: Replacing Na's with Iterative Imputer prediction metrics

Comparison of best models' results in each missing value handling technique:

Missing value handling strategy	precision	recall	f1_score	accuracy
Dropping Na	0.66	0.66	0.66	0.66
Replacing with other value (-1 or 'Na')	0.59	0.59	0.59	0.59
Replacing with mean value of feature	0.59	0.59	0.58	0.59
Predicting with Iterative Imputer	0.61	0.61	0.61	0.61

Table 3.7: Comparison of best models' metrics using different missing values handling strategies

3.3.6 Scaling data

As for a last step of data preprocessing what we did was to scale the newly modified dataset. As (Medium, 2017) [2] mentions on his publication, most of the times, your dataset will contain features which have different values, units and range. But since, most of the machine learning algorithms use Euclidian distance between two data points in their computations, this is a problem. If left alone, these algorithms only take in consideration the values of these features neglecting their units. The results between different units would vary a lot. The features with higher values will have a greater impact in the distance calculations than features with low values.

So, all the features' values need to be scaled. For scaling the data we used the simple min-max normalization, the simplest method that consists in rescaling the range of features in [0, 1].

3.4 Feature Selection

A very important step on the improvement of the scoring was the feature selection phase, in which we decided to remove all features with zero and low importance, with cumulative importance threshold of 0.90.

3.4.1 Zero Importance Features

As a first step we decided to find all the features of zero importance using the feature_selector library. The library also made available the one_hot_encoding of alphabetic features. The used method relied on a machine learning model to identify features to remove. It required a supervised learning problem with labels. The method worked by finding feature importance using a gradient boosting machine implemented in the LightGBM library.

After the training we reached in the conclusion of 3351 features with zero importance after one-hot encoding. The original features were 42 while the one-hot features were 3319. There were many features of zero importance mainly one-hot features.

The top 10 features importance’s are shown below:

Feature	Importance	Normalized Importance	Cumulative Importance
fc_face_age	2.3	0.589744	0.589744
num_funding_rounds	1.0	0.256410	0.846154
fc_neutral_emotion	0.4	0.102564	0.948718
fc_yaw_angle	0.2	0.051282	1
cb_person_region_name_Valle Del Cauca	0.0	0.000000	1
cb_person_region_name_Tokyo	0.0	0.000000	1
cb_person_region_name_Toscana	0.0	0.000000	1
cb_person_region_name_Tunis	0.0	0.000000	1
cb_person_region_name_Udmurt	0.0	0.000000	1
cb_person_region_name_Umbria	0.0	0.000000	1

Table 3.8: Top 10 feature importances

3.4.2 Low Importance Features

In the next step, we decided to find the low importance features using again the gradient boosting machine, but first the low importance features method had to be executed. We found the low importance features that did not need to reach a specified cumulative total feature importance. For example, if we passed as cumulative importance the 0.99, this would find the lowest importance features that are not needed to reach 99% of the total feature importance. We set our cumulative importance threshold on 0.90.
So, we found that 2 features were required for cumulative importance of 0.90 after one-hot encoding. 3353 features did not contribute to cumulative importance of 0.90.
This time the lowest importance features were: ‘fc_yaw_angle’, ’fc_BMI’, ’num_closed_companies’, ‘fc_fWHR_cheekbone_prominence_ratio’, ’fc_pitch_angle’.

As we noticed in the below figure 3.12 not many of the features had great importance on the initial dataset.

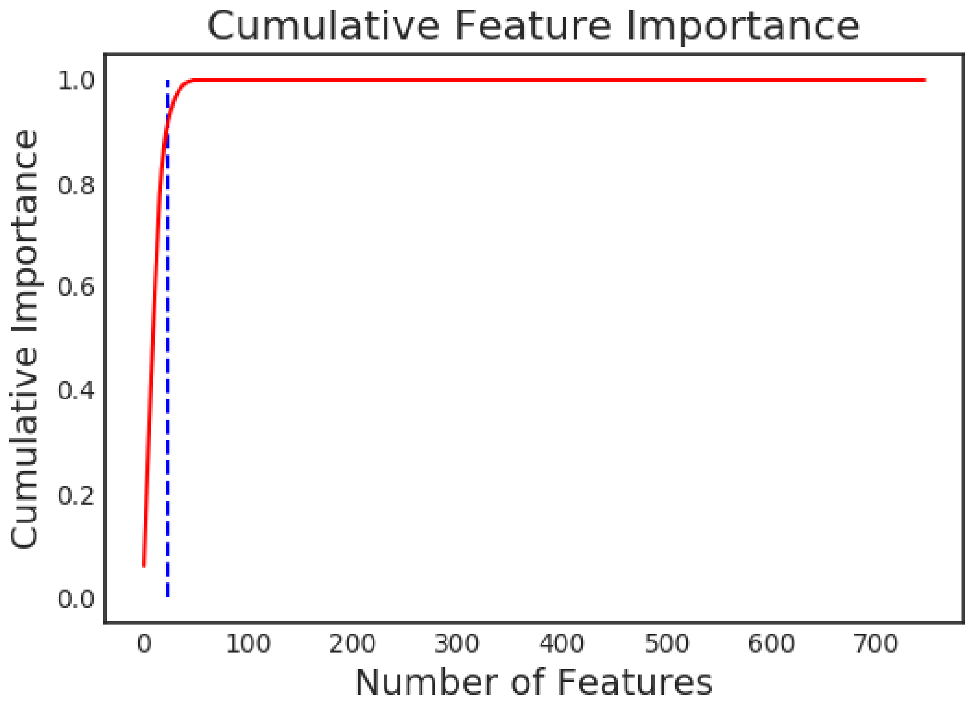

Figure 3.12: Feature importance of initial dataset

3.4.3 Features Importance's using LassoCV and Extra Trees Classifier

We also tried finding the feature importance with other different techniques.
By using LassoCV we managed to keep 14 features out of the 26 that we inserted as input. As we noticed from the below figure 3.13 the features that were selected had a very low importance which could not get higher than 0.15.

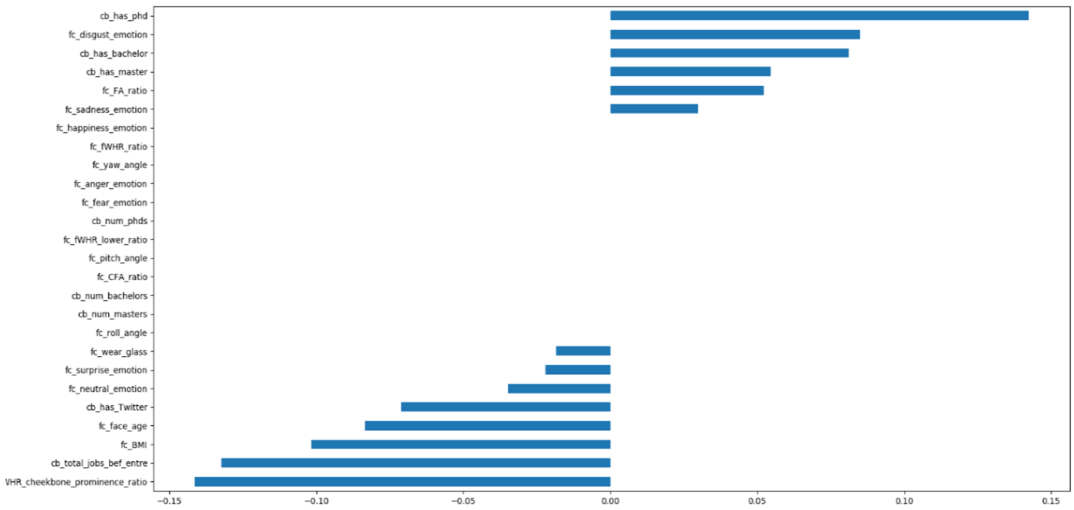

Figure 3.13: Feature importance of initial dataset using LassoCV

We did not stop there as we also tested using the Extra Trees Classifier technique, but the results were the same with the feature importance as shown in the below figure 3.14, where we notice that they did not pass the 0.06 importance.

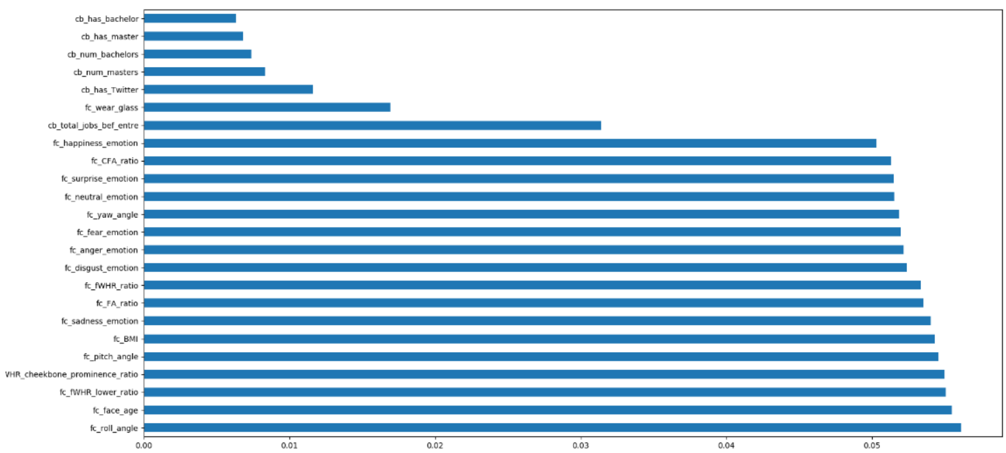

Figure 3.14: Feature importance of initial dataset using Extra Trees Classifier

Furthermore, features' importance was tried to be visualized with LinearSVC too, but the results were similar as the above, very low importance for each feature. Variance Threshold was also tried in order to remove the features with a variance below a specific threshold but the variance of each feature with its' importance was not very related, so their removal did not achieve a better result. At last, we tried RFE (Recursive Feature Elimination) in order to select features recursively by considering smaller and smaller sets of features and train the model on each iteration but this technique requires to know the number features we want to keep. To mention here that RFE is a very time consuming and high complexity algorithm and we could not try all the possible combination of the features after One Hot Encoding technique, which returned approximately 4K features, as it would require a long time to execute.

In conclusion, the time restriction and the features' importance of the initial dataset not being high enough to determine which were better to keep or not, made us continue to use our first choice which was to remove all features with low and zero importance.

3.5 Machine Learning

After all the preprocessing of the data done, the next step was to train and test specific machine learning algorithms for prediction. The training and test phase were done using k-fold cross validation of 8 parts in order to estimate the skill of each model on unseen data as well as calibration of the results by using CalibratedClassifier. The implementation of the machine learning models is not part of this research, but we must understand how they work from an abstract point of view and understand their parameters' role and the effect they have on prediction, so we get the best results out of them.

The machine learning algorithms used are:

- Logistic Regression
- Linear Discriminant Analysis
- K-Nearest Neighbors
- Gaussian Naïve Bayes
- Decision Tree
- Support Vector Classification

- Gradient Boosting
- Random Forest
- Neural network

What all the selected algorithms have in common is their ability to be used for good classification machine learning problems and they were chosen after a lot of research. There are other ML algorithms that we decided not to use, because of the nature of our problem, which is a prediction classification problem with 2 classes. In general learning can be supervised, semi-supervised or unsupervised, but in our case, we are using supervised learning. Supervised learning means that the accuracy of the model is highly correlated with the input we provide to the model.

In this phase, we also used Over-Sampling strategy in order to ensure that the training phase would be made upon a well-balanced dataset part of the cross validation. Under-Sampling of the majority class was also tried but Over-Sampling of the minority class had better results as it did not remove data like Under-Sampling removed those of the majority class, but in contrast it duplicated even more those of the minority class.

3.5.1 Machine Learning algorithms' hyper-parameter tuning

In order to achieve the best prediction results of the above machine learning algorithms we had to understand and find the best parameters for each model as mentioned above. As a first step, after understanding the use of each parameter we tried to manually experiment with them and find the best parameters based on the prediction's F1 score, which was our main metric. Then we decided to stop searching manually, as each model's parameters would depend on the nature of the data and it would not be dynamically effective for every dataset.

So, as a next step, we decided to use parameter hyper-tuning available from the GridSearchCV function of the model_selection library of sklearn. Based on Wikipedia (Hyperparameter optimization, 2019) [9] in machine learning, hyper-tuning parameters for models is the process of choosing a set of optimal parameters for a learning algorithm's prediction.

Hyper-tuning of parameters was executed upon Logistic Regression, Random Forest, Gradient Boosting, Support Vector, K-Nearest Neighbors, Decision Tree and Neural

Network models. We could not find the best parameters of Gaussian Naïve Bayes algorithm as the parameters were limited and of Linear Discriminant Analysis because of rounding error bugs on the computation of its' weighted covariance matrix.

Considering the above machine learning algorithms, each of them had a specific parameter grid with different trial values for each parameter. The hyper-tuning was done on a 10 times k-fold cross-validation method and the best parameters were based on the best F1 score result.

To mention here that Over-Sampling was used for correct training upon a well-balanced dataset part.

Below we show all the parameter grids for each model, including the parameter and the trial values for each of them.

Logistic Regression:

Parameter	Values
solver	newton-cg, lbfgs, liblinear, sag, saga
penalty	l1,l2

Table 3.9: Hyper-tuning parameters for Logistic Regression

Random Forest Classifier:

Parameter	Values
n_estimators	100,200,300,500,600,700,800
class_weight	balanced,balanced_subsample
Criterion	gini,entropy

Table 3.10: Hyper-tuning parameters for Random Forest

Gradient Boosting:

Parameter	Values
learning_rate	0.2,0.4,0.6,0.7
n_estimators	100,200,300,500
Loss	deviance,exponential
Criterion	friedman_mse,mse,mae

Table 3.11: Hyper-tuning parameters for Gradient Boosting

Support Vector:

Parameter	Values
C	1,10,100,1000
kernel	linear,rbf
gamma	0.001, 0.0001
decision_function_shape	ovo,ovr

Table 3.12: Hyper-tuning parameters for Support Vector

K-Nearest Neighbors:

Parameter	Values
n_neighbors	1,5,10,20,25,30,40,50,60,70,100
algorithm	auto,ball_tree,kd_tree,brute
weights	uniform,distance

Table 3.13: Hyper-tuning parameters for K-Nearest Neighbors

Decision Tree:

Parameter	Values
class_weight	balanced
criterion	gini,entropy
splitter	best,random

Table 3.14: Hyper-tuning parameters for Decision Tree

Neural Network:

Parameter	Values
epochs	10,50,100,150

Table 3.15: Hyper-tuning parameters for Neural Network

The only supported hyper-tuning parameter for keras Neural Network based on GridSearchCV was “epochs”.

3.5.2 Selected Metrics

Metric selection was a very important part of our prediction’s evaluation.
As a first step we decided that accuracy, was a usual metric that would be used for this kind of problems so depending on that our first results were satisfactorily. But soon we realized that we could not depend on accuracy as it is used when True Positives and True Negatives are more important than False Negatives and False Positives and when the data is perfectly balanced. For us all True Positives, True Negatives, False Positives and False Negatives were equally important and our data was not balanced so the use of accuracy was not the best choice.
So, we decided on F1 score to be our main metric, the harmonic mean of Precision and Recall and gives a better measure of the incorrectly classified cases than the accuracy metric.

$$F1 = 2 * \left(\frac{Precision * Recall}{Precision + Recall}\right)$$

Precision is the measure of the correctly identified positive cases from all the predicted positive cases. Thus, it is useful when the cost of False Positives is high.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} = \frac{True\ Positives}{Total\ Predicted\ Positives}$$

Recall is the measure of the correctly identified positive cases from all the actual positive cases. It is important when the cost of False Negatives is high.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} = \frac{True\ Positives}{Total\ Actual\ Positives}$$

In summary, accuracy metric would always be satisfactory as its' prediction success would be based only on positive samples and would not count the negative samples prediction. F1 score metric would be based on both so it would be more accurate and informative for us to evaluate the models.
To mention here the following definitions:
True Positive => correct prediction of the positive class ('1') made by the model.
True Negative => correct prediction of the negative class ('0') made by the model.
False Positive => incorrect prediction of the positive class ('1') made by the model.
False Negative => incorrect prediction of the negative class ('0') made by the model.

Chapter 4

Evaluation

Contents

4.1 Experiments and Results

This section presents the results of our study on the 50K rows dataset as well as on the 600K rows dataset and provides a summary table 4.10, which contains the accuracy, precision, recall, and F1-score for each machine learning model we tried on the small dataset and a summary table 4.20, which contains the same metrics for the big dataset. Also, we contain tables for each machine learning model's metrics changes on each step starting from the baseline of prediction for both datasets. Furthermore, we create a visual comparison of each model's metrics for both datasets to give a better understanding of the Precision and Recall impact on accuracy and F1 score outcome. Finally, we extract some of the most important features of entrepreneurs who have the potential of receiving a funding based on our model's prediction. From the tables below 4.1 – 4.9 and 4.11-4.19, we can notice that almost in each algorithm the metrics improve after every step in both datasets. We started measuring the score of the baseline case, where only missing values were dropped and the conversion of alphabetical data to numerical data took place. Then continued by adding the feature selection preprocessing part and after that we added the scaling of data's values. At, last hyper-tuning of models' parameters was added, thing that completed the whole preprocessing phase of the dataset. The steps of handling the prediction problem were a very important part as they had to be executed with a specific row in order to get a good result. We tried to change the execution order and the metrics were much lower than the ones below.

From the table 4.10 we can notice that Linear Discriminant analysis had the best outcome in the small dataset, which reached 66% F1 score, followed by Logistic Regression with 65%, SVC with 64% and Random Forest with 64%. All the metrics displayed in the table are the weighted_avg of the gathered metrics.

Meanwhile, from the table 4.20, we can notice that Neural Network had the best outcome in the big dataset, which reached 59% F1 score, followed by Logistic Regression, Gradient Boosting and Linear Discriminant analysis with 58% F1 score.

But what these scores actually represent?

A 66% F1 score represents a harmonic mean of precision and recall of this problem. F1 score is a mean measurement that actually gives more weight, or importance, to the lower values. Also, based on the below metrics precision and recall are in the range of 66%-67%, considering the Linear Discriminant Analysis model, so the above 66% F1 score seems logical and correct.

As we mentioned before recall is the metric that represents all the relevant, positive, cases and the model's ability to find all the data points of interest in a dataset. So, in simple words the metric of all entrepreneurs who successfully received funding. So, our intuition tells us that we should maximize this ability. But even if we have a 100% recall that would not mean that it is correct as all cases of the dataset would be labeled as entrepreneurs who successfully received funding, but who actually did not, so the prediction would always be a successful funding receival for each entrepreneur case. This problem is solved by precision, the ability to identify only the relevant data points of interest of those found by the recall. In other words, precision finds all the entrepreneurs who truly received funding out of all entrepreneurs who were labeled to successfully receive funding. As with most concepts in data science, there is a trade-off in the metrics we choose to maximize. In the case of recall, when we increase the recall, we decrease the precision. But as we explained we want to avoid this imbalance so, we are trying to achieve the highest scored balance between recall and precision, or in other words the highest F1 score, which shows us a percentage of how many entrepreneurial funding receival results were predicted correctly. Based on these we managed to achieve our best F1 score of 69%.

Logistic Regression:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.60	0.60	0.60	0.60
Dropped NAs, one-hot-encoded & feature selection	0.63	0.62	0.62	0.62
Dropped NAs, one-hot-encoded, feature selection & scaling	0.64	0.64	0.64	0.64
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.66	0.65	0.65	0.65

Table 4.1: Metrics' changes of Logistic Regression based on each step on small dataset

Linear Discriminant Analysis:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.61	0.61	0.61	0.61
Dropped NAs, one-hot-encoded & feature selection	0.64	0.63	0.63	0.63
Dropped NAs, one-hot-encoded, feature selection & scaling	0.65	0.65	0.65	0.65
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.67	0.66	0.66	0.66

Table 4.2: Metrics' changes of Linear Discriminant Analysis based on each step on small dataset

K-Nearest Neighbors:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.48	0.47	0.47	0.47
Dropped NAs, one-hot-encoded & feature selection	0.47	0.47	0.47	0.47
Dropped NAs, one-hot-encoded, feature selection & scaling	0.57	0.57	0.57	0.57
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.58	0.58	0.58	0.58

Table 4.3: Metrics' changes of K-Nearest Neighbors based on each step on small dataset

Gaussian Naive Bayes:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.58	0.52	0.47	0.52
Dropped NAs, one-hot-encoded & feature selection	0.66	0.63	0.62	0.63
Dropped NAs, one-hot-encoded, feature selection & scaling	0.64	0.61	0.60	0.61
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.66	0.63	0.62	0.62

Table 4.4: Metrics' changes of Gaussian Naive Bayes based on each step on small dataset

Decision Tree:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.50	0.50	0.50	0.50
Dropped NAs, one-hot-encoded & feature selection	0.48	0.48	0.48	0.48
Dropped NAs, one-hot-encoded, feature selection & scaling	0.51	0.51	0.50	0.51
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.50	0.50	0.50	0.50

Table 4.5: Metrics' changes of Decision Tree based on each step on small dataset

Support Vector:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.52	0.50	0.50	0.50
Dropped NAs, one-hot-encoded & feature selection	0.52	0.50	0.50	0.50
Dropped NAs, one-hot-encoded, feature selection & scaling	0.62	0.61	0.62	0.61
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.64	0.64	0.64	0.64

Table 4.6: Metrics' changes of Support Vector based on each step on small dataset

Gradient Boosting:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.59	0.59	0.59	0.59
Dropped NAs, one-hot-encoded & feature selection	0.59	0.58	0.58	0.58
Dropped NAs, one-hot-encoded, feature selection & scaling	0.60	0.59	0.59	0.59
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.60	0.60	0.60	0.60

Table 4.7: Metrics' changes of Gradient Boosting based on each step on small dataset

Random Forest:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.57	0.57	0.57	0.57
Dropped NAs, one-hot-encoded & feature selection	0.57	0.57	0.57	0.57
Dropped NAs, one-hot-encoded, feature selection & scaling	0.57	0.57	0.57	0.57
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.64	0.64	0.64	0.64

Table 4.8: Metrics' changes of Random Forest based on each step on small dataset

Neural Network:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.54	0.53	0.52	0.53
Dropped NAs, one-hot-encoded & feature selection	0.51	0.50	0.50	0.50
Dropped NAs, one-hot-encoded, feature selection & scaling	0.58	0.58	0.58	0.58
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.58	0.58	0.58	0.58

Table 4.9: Metrics' changes of Neural Network based on each step on small dataset

Furthermore, a file with the prediction and prediction probability of each dataset's row for the small dataset was created and the best result was 92% prediction probability, achieved by Neural Network, followed by 80%, achieved by Linear Discriminant Analysis and Logistic Regression. The prediction probability is the likelihood that each prediction belongs to the class predicted for that specific input, that in this case was each entrepreneur of the dataset separately. So, a 92% prediction probability meant that the result predicted for that entrepreneur, which was getting a funding, was correct by 92%.

Final Comparison:

Machine Learning Algorithms	precision	recall	F1 score	accuracy
Logistic Regression	0.66	0.65	0.65	0.65
Linear Discriminant Analysis	0.67	0.66	0.66	0.66
K-Nearest Neighbors	0.58	0.58	0.58	0.58
Gaussian Naive Bayes	0.66	0.63	0.62	0.62
Decision Tree	0.5	0.5	0.5	0.5
Support Vector Classification	0.64	0.64	0.64	0.64
Gradient Boosting	0.6	0.6	0.6	0.6
Random Forest	0.64	0.64	0.64	0.64
Neural Network	0.58	0.58	0.58	0.58

Table 4.10: Comparison of all models' metrics after all steps on small dataset

The model with the lowest F1-score for the small dataset was the Decision Tree, with 50%. The explanation of this fact may be the dataset's features which are very abstract to define whether an entrepreneur will receive funding or not with a high prediction probability. Most of them are general features which do not have correlation with the entrepreneurial funding receival success and cannot help in the prediction of funding receival or not.

The following graphs present the visual comparison of every model's metrics in the small dataset.

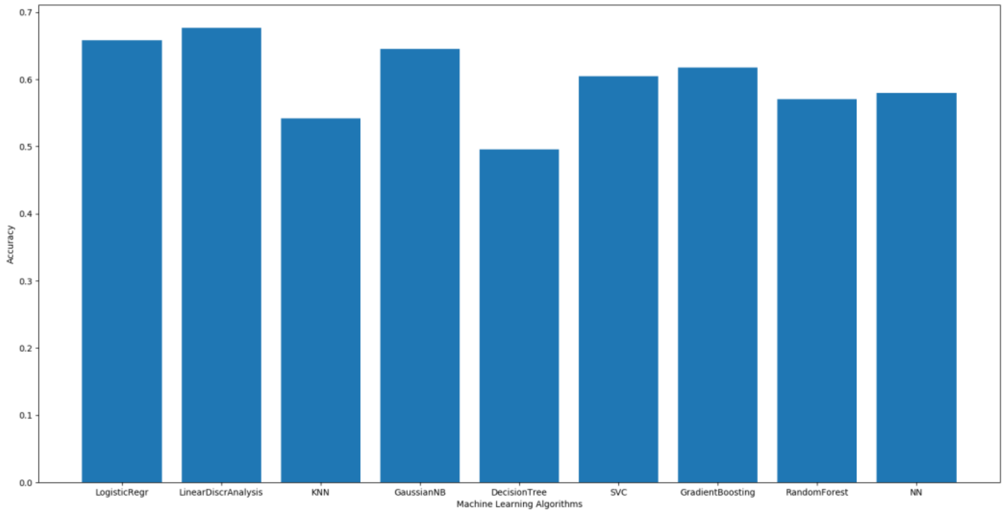

Figure 4.1: Accuracy comparison graph of all models on small dataset

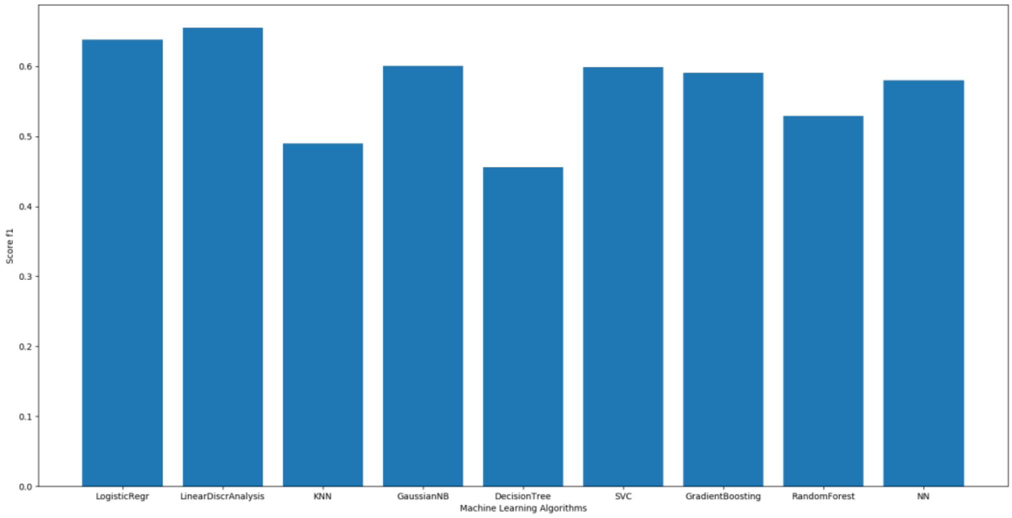

Figure 4.2: F1 score comparison graph of all models on small dataset

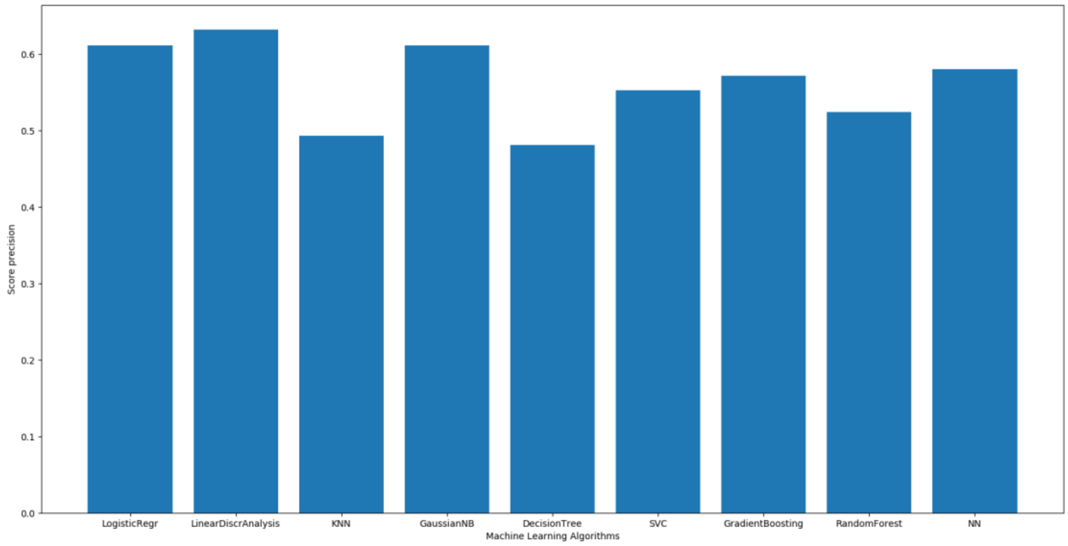

Figure 4.3: Precision comparison graph of all models on small dataset

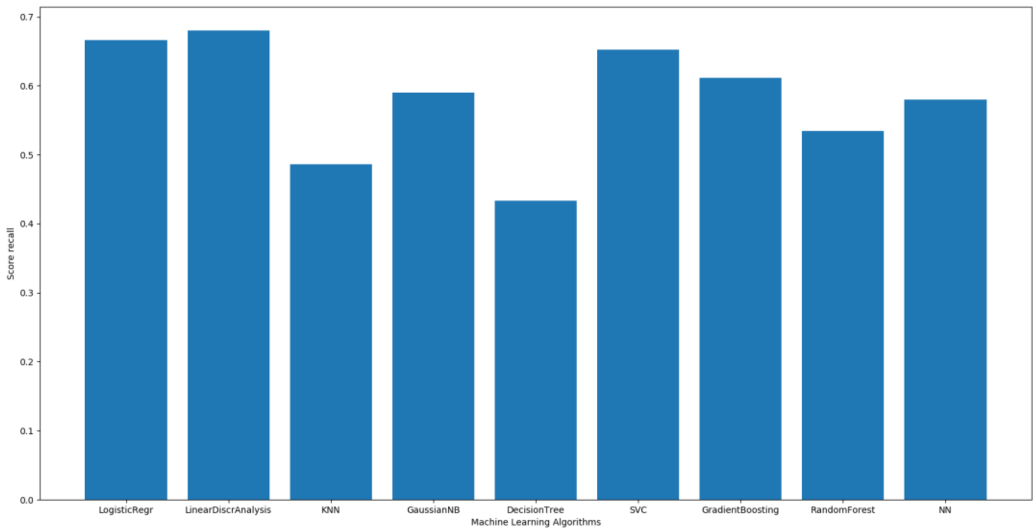

Figure 4.4: Recall comparison graph of all models on small dataset

Below the big dataset's results are displayed:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.57	0.57	0.57	0.57
Dropped NAs, one-hot-encoded & feature selection	0.57	0.57	0.57	0.57
Dropped NAs, one-hot-encoded, feature selection & scaling	0.58	0.58	0.58	0.58
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.58	0.58	0.58	0.58

Table 4.11: Metrics' changes of Logistic Regression based on each step on big dataset

Linear Discriminant Analysis:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.57	0.57	0.57	0.57
Dropped NAs, one-hot-encoded & feature selection	0.58	0.58	0.58	0.58
Dropped NAs, one-hot-encoded, feature selection & scaling	0.58	0.58	0.58	0.58
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.58	0.58	0.58	0.58

Table 4.12: Metrics' changes of Linear Discriminant Analysis based on each step on big dataset

K-Nearest Neighbors:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.51	0.51	0.51	0.51
Dropped NAs, one-hot-encoded & feature selection	0.52	0.52	0.52	0.52
Dropped NAs, one-hot-encoded, feature selection & scaling	0.54	0.54	0.54	0.54
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.54	0.54	0.54	0.54

Table 4.13: Metrics' changes of K-Nearest Neighbors based on each step on big dataset

Gaussian Naive Bayes:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.54	0.54	0.53	0.54
Dropped NAs, one-hot-encoded & feature selection	0.55	0.55	0.53	0.55
Dropped NAs, one-hot-encoded, feature selection & scaling	0.55	0.55	0.53	0.55
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.55	0.55	0.52	0.55

Table 4.14: Metrics' changes of Gaussian Naive Bayes based on each step on big dataset

Decision Tree:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.5	0.5	0.5	0.5
Dropped NAs, one-hot-encoded & feature selection	0.52	0.52	0.52	0.52
Dropped NAs, one-hot-encoded, feature selection & scaling	0.5	0.5	0.5	0.5
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.52	0.52	0.52	0.52

Table 4.15: Metrics' changes of Decision Tree based on each step on big dataset

Support Vector:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.52	0.52	0.52	0.52
Dropped NAs, one-hot-encoded & feature selection	0.52	0.52	0.52	0.52
Dropped NAs, one-hot-encoded, feature selection & scaling	0.58	0.58	0.58	0.58
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.56	0.56	0.56	0.56

Table 4.16: Metrics' changes of Support Vector based on each step on big dataset

Gradient Boosting:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.57	0.57	0.57	0.57
Dropped NAs, one-hot-encoded & feature selection	0.58	0.58	0.58	0.58
Dropped NAs, one-hot-encoded, feature selection & scaling	0.58	0.58	0.58	0.58
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.58	0.58	0.58	0.58

Table 4.17: Metrics' changes of Gradient Boosting based on each step on big dataset

Random Forest:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.56	0.56	0.56	0.56
Dropped NAs, one-hot-encoded & feature selection	0.57	0.57	0.56	0.57
Dropped NAs, one-hot-encoded, feature selection & scaling	0.57	0.57	0.57	0.57
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.57	0.57	0.57	0.57

Table 4.18: Metrics' changes of Random Forest based on each step on big dataset

Neural Network:

Steps	precision	recall	f1_score	accuracy
Dropped NAs & one-hot-encoded	0.56	0.56	0.56	0.56
Dropped NAs, one-hot-encoded & feature selection	0.54	0.54	0.54	0.54
Dropped NAs, one-hot-encoded, feature selection & scaling	0.57	0.57	0.57	0.57
Dropped NAs, one-hot-encoded, feature selection, scaling & hyperturning ml alg parameters	0.59	0.59	0.59	0.59

Table 4.19: Metrics' changes of Neural Network based on each step on big dataset

Final Comparison:

Machine Learning Algorithms	precision	recall	f1_score	accuracy
Logistic Regression	0.58	0.58	0.58	0.58
Linear Discriminant Analysis	0.58	0.58	0.58	0.58
K-Nearest Neighbors	0.54	0.54	0.54	0.54
Gaussian Naive Bayes	0.55	0.55	0.52	0.55
Decision Tree	0.52	0.52	0.52	0.52
Support Vector Classification	0.56	0.56	0.56	0.56
Gradient Boosting	0.58	0.58	0.58	0.58
Random Forest	0.57	0.57	0.57	0.57
Neural Network	0.59	0.59	0.59	0.59

Table 4.20: Comparison of all models' metrics after all steps on big dataset

The model with the lowest F1-score this time was the Decision Tree and Gaussian Naïve Bayes, with 52%, an improved lowest score in comparison with the small dataset's lowest score. This happened since more data were used so the recall in this big dataset was higher than then recall in the smaller dataset, as there existed more data points of interest or in other words more entrepreneurs who received a funding.

The following graphs present the visual comparison of every model's metrics in the big dataset.

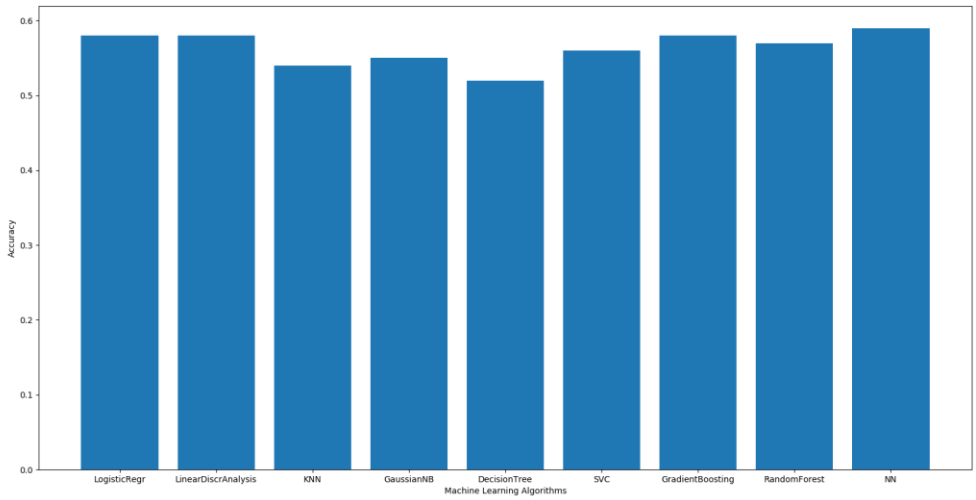

Figure 4.5: Accuracy comparison graph of all models on big dataset

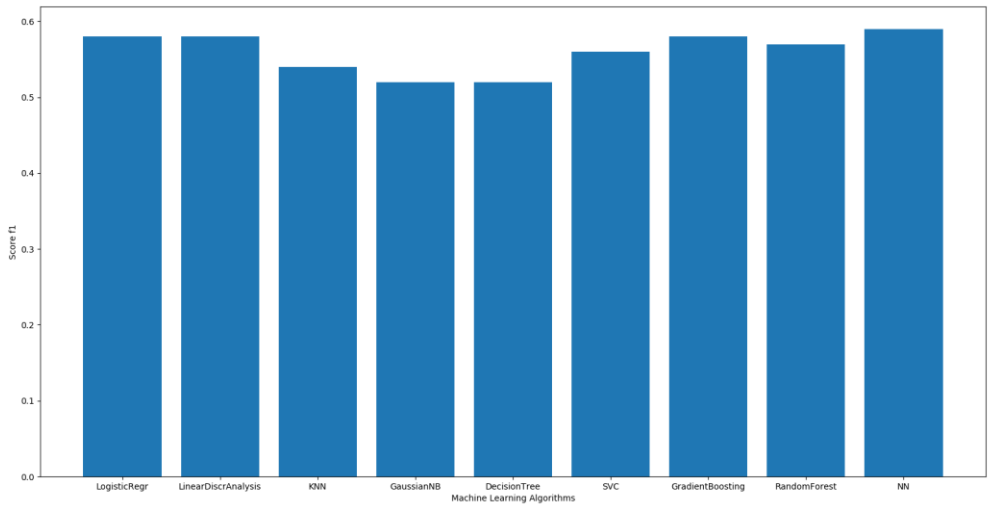

Figure 4.6: F1 score comparison graph of all models on big dataset

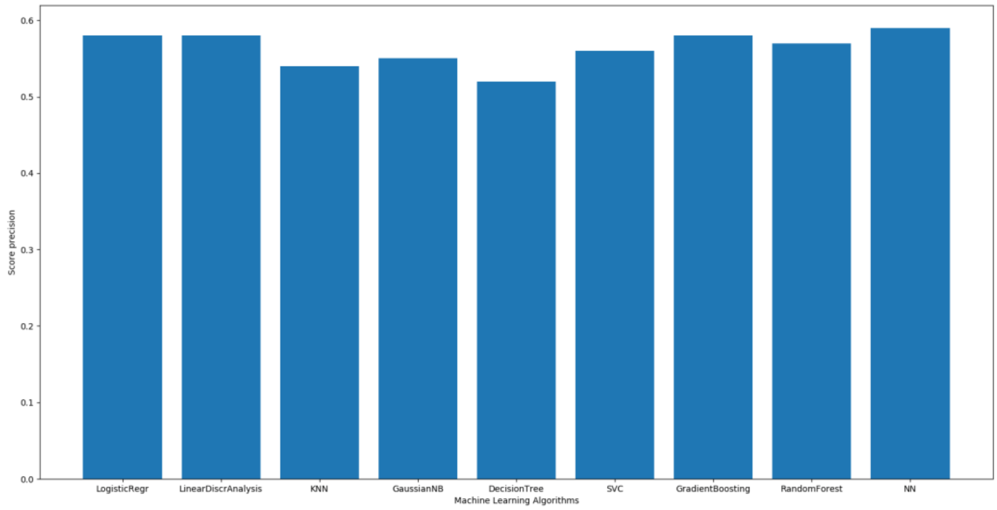

Figure 4.7: Precision comparison graph of all models on big dataset

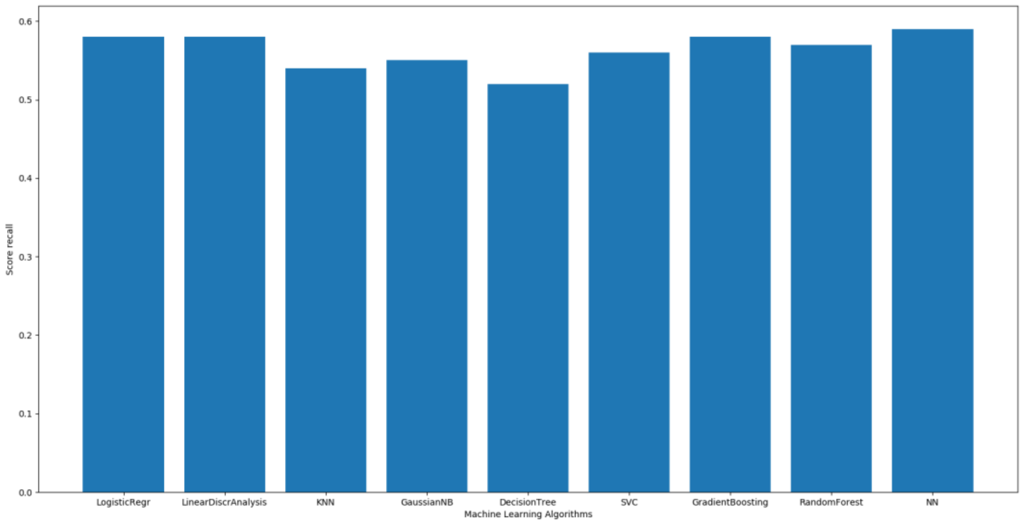

Figure 4.8: Recall comparison graph of all models on big dataset

From the above figures we notice that the Precision, Recall, Accuracy and F1 score are all balanced for each algorithm, in every case, something that shows that our prediction's score is logical and correct. Also, the executions were made many times and the best F1 score that was achieved was by Linear Discriminant Analysis on the small dataset, 69%.

Moreover, we also visualized the ROC (Receiver Operator Characteristic) curve for each machine learning model on the small dataset in order to show the diagnostic ability of each binary classifier. The ROC curve shows the trade-off between sensitivity (or TPR) and specificity (1 – FPR). Classifiers that give curves closer to the top-left corner indicate a better performance. As a baseline, a random classifier is expected to give points lying along the diagonal (FPR = TPR). The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test. So, based on this we can see from the figures 4.9-4.12 below that Logistic Regression and Linear Discriminant Analysis had the best performance in comparison with the other models. Also, Neural Network's ROC curve is visualized separately as Neural network was executed on a separate script than the other models.

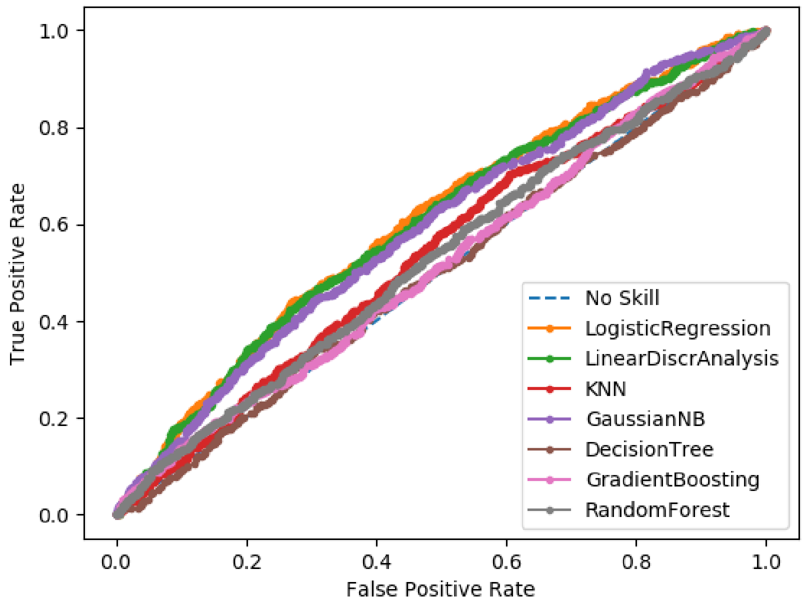

Figure 4.9: ROC curves of all machine learning models

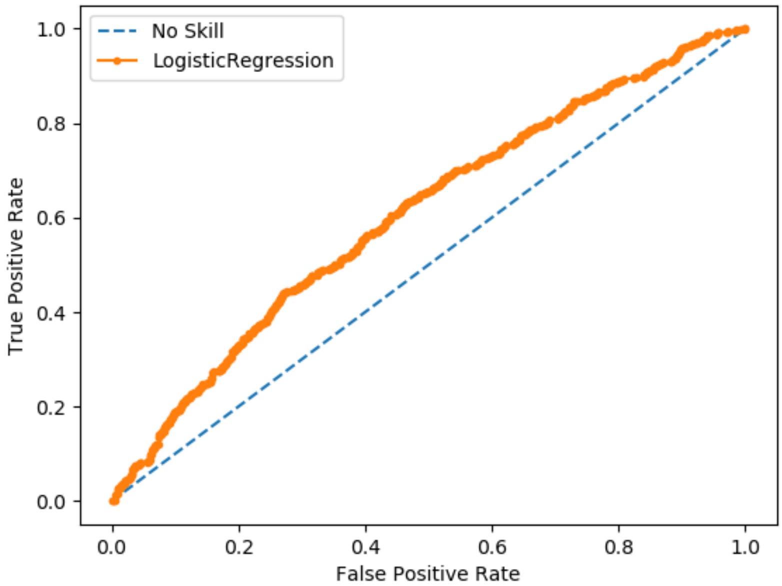

Figure 4.10: ROC curve of Logistic Regression model

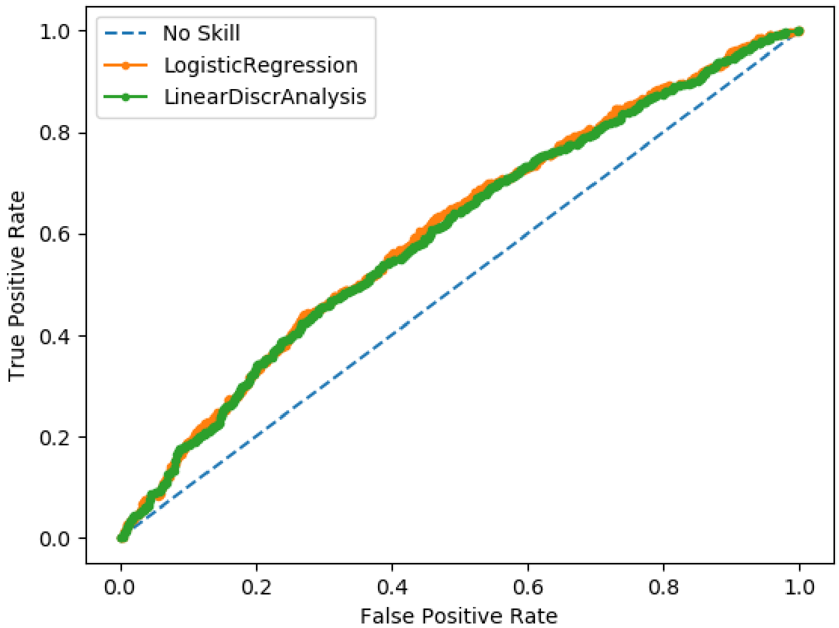

Figure 4.11: ROC curves of Logistic Regression and Linear Discriminant Analysis models

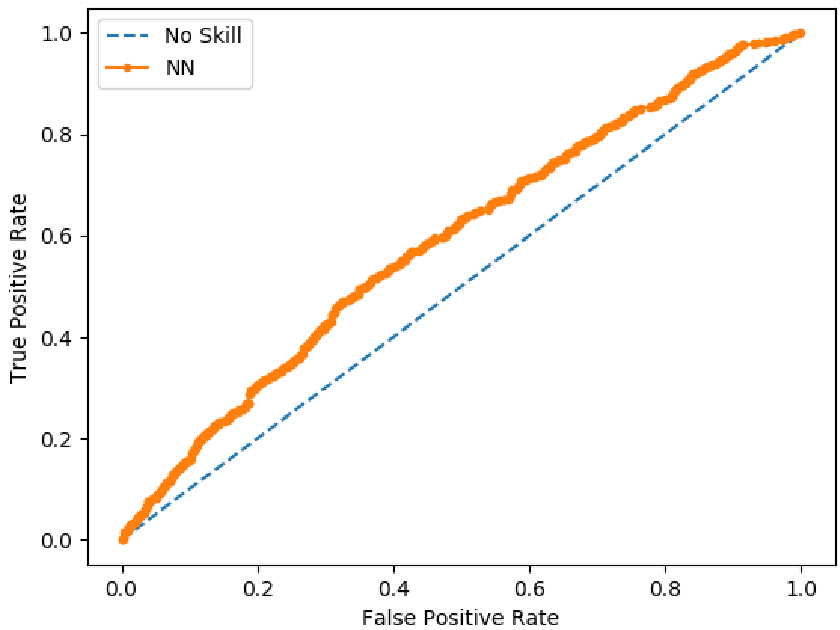

Figure 4.12: ROC curve of Neural Network

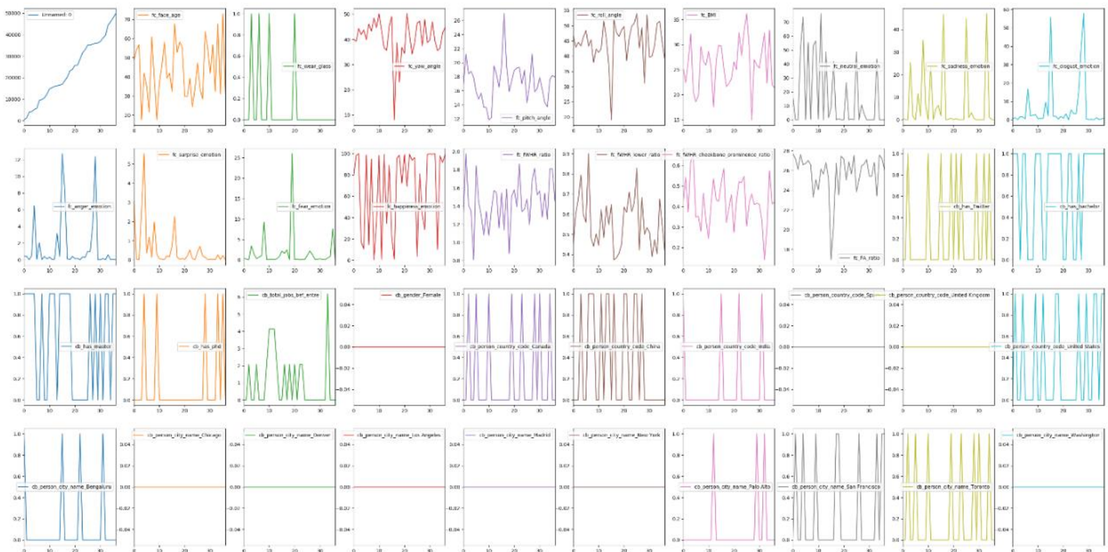

Figure 4.13: Characteristics of successful entrepreneurs based on best model prediction

Based on the above figure 4.13, we found that entrepreneurs who are more possible to receive a funding have some specific characteristics based on the best model's prediction with the highest F1 score made upon the given Crunchbase dataset and extra data about emotion and face characteristics examined in other similar studies (Nicolaou N. et al.; Shane S. et al.)[13,19]. With a 69% confidence we can say that:

- Age of an entrepreneur, who has the potential of receiving funding, varies from 20-70 years old.
- ¼ of entrepreneurs, who have the potential of receiving funding, wear glasses.
- More than ½ of entrepreneurs, who have the potential of receiving funding, have high neutral emotion, that does not pass 70/100 score.
- Entrepreneurs, who have the potential of receiving funding, have low sadness emotion, that does not pass 40/100 score.
- Entrepreneurs, who have the potential of receiving funding, have very low disgust emotion, that does not pass 20/100 score, with some exceptions that can reach up to 50/100.
- Entrepreneurs, who have the potential of receiving funding, have very low anger emotion, that does not pass 22/100 score.
- Entrepreneurs, who have the potential of receiving funding, have very low surprise emotion, that does not pass 10/100 score.
- Entrepreneurs, who have the potential of receiving funding, have very low fear emotion, that does not pass 25/100 score.
- Entrepreneurs, who have the potential of receiving funding, have high happiness emotion, that usually reaches up to 100.
- Entrepreneurs', who have the potential of receiving funding, face fWHR_ratio is between 1-2, higher than entrepreneurs' who do not receive funding. The facial width-to-height ratio (fWHR) is the ratio of the distance between the left and right zygon and the distance between upper lip and mid-brow.
- Entrepreneurs', who have the potential of receiving funding, face fWHR_lower_ratio is between 0,4- 0,9 higher than entrepreneurs' who do not receive funding. fWHR_lower_ratio is the ratio of the bizygomatic width divided by the distance between the mean eye height and the bottom of the chin.
- Entrepreneurs', who have the potential of receiving funding, face fWHR_cheekbone_prominence_ratio is between 0,2- 0,6 insignificantly higher than entrepreneurs' who do not receive funding.

fWHR_cheekbone_prominence_ratio is “the ratio of cheekbone width divided by jaw width” (Nicolaou N. et al.)[13].

- ½ of successful entrepreneurs, who have the potential of receiving funding, have Twitter.
- Most of entrepreneurs, who have the potential of receiving funding, usually have bachelor’s and master’s degrees.
- Not many entrepreneurs, who have the potential of receiving funding, have PhD degree. Less than ¼ of them do.
- Most of entrepreneurs, who have the potential of receiving funding, have more than 2 previous jobs but can reach up to 6.
- Based on the model predictions most entrepreneurs, who have the potential of receiving funding, are of male gender.
- Most entrepreneurs, who have the potential of receiving funding, are originated from developed countries such as Canada, China and United States, while a few of them are originated from countries like India.
- Based on the above countries most entrepreneurs, who have the potential of receiving funding, are originated from cities such as Toronto and San Francisco, while a few of them are originated from cities like Bengaluru and Palo Alto.

Finally, based on all the above findings the results reveal with a 69% confidence that a male entrepreneur older than 20 years old, showing a happy emotion and at the same time being fearless, calm and not surprised, with not absolutely symmetric face characteristics but higher ratios than the entrepreneurs who do not receive funding, actively using social media, having bachelor’s and master’s degree, having some previous job experience and whose origin is from developed countries like Canada, United States or China and especially from cities like Toronto and San Francisco has more chances of succeeding with their next venture and get an important funding than an entrepreneur who does not have the above characteristics.

This result shows that all the previous studies, that concentrated only on some characteristics, like social media engagement, financial growth though years and facial characteristics of entrepreneurs in order to predict whether funding receival will be successful or not, were only based on one small part of the total factors that impact the funding receival success. We give a more complete picture of the factors that influence the funding receival decision and believe that the impact factor of each characteristic must be different and very important to the decision made by the investors whether to give a funding or not. Furthermore, we make a funding receival prediction based on all these characteristics, which we believe to be more accurate than the predictions made using only specific parts of characteristics. Finally, we understand the importance of the characteristics used for prediction in other studies, but we believe that a whole picture of all the factors that impact the funding receival success would be more useful information for new entrepreneurs and more accurate to predict whether a new entrepreneur will receive funding or not.

4.2 Prediction

As a final step we managed to save each model after training and testing phase as a .sav file, except from the Neural Network, which was not saved due to KerasClassifier function use, which does not support this saving option. In order to make this possible we used joblib library to save the model and then in another script, used only for the prediction of user input, we load the ready model, fit it in the data and predict the result of the user's input.

The user's input consists of the following 21 inputs displayed in table 4.21 below:

Input	Description
fc_face_age	each person’s age (integer)
fc_wear_glass	if a person wears glasses or not (0 = doesn’t wear glasses, 1 = wears)
fc_ethnicity	the ethnicity of the person (in string format)
fc_BMI	the BMI metric of the person (in float format) (not necessary if connected with face++ and image is inserted)
fc_neutral_emotion	the metric of neutral emotion from 0-100 (in float format)
fc_sadness_emotion	the metric of sadness emotion from 0-100 (in float format)
fc_disgust_emotion	the metric of disgust emotion from 0-100 (in float format)
fc_anger_emotion	the metric of anger emotion from 0-100 (in float format)
fc_surprise_emotion	the metric of surprise emotion from 0-100 (in float format)
fc_fear_emotion	the metric of fear emotion from 0-100 (in float format)
cb_gender	the person’s gender (in string format)
cb_person_country_code	the metric of fear emotion from 0-100 (in float format)
cb_person_city_name	the current city of the person (in string format, joined with ‘_’, in case of 2 names)
cb_has_Twitter	if a person has Twitter account or not (0 = doesn’t have, 1 = has)
cb_has_bachelor	if a person has bachelor’s degree or not (0 = doesn’t have, 1 = has)
cb_num_bachelors	the number of bachelor’s degrees a person has (integer)
cb_has_master	if a person has master’s degree or not (0 = doesn’t have, 1 = has)
cb_num_masters	the number of master’s degrees a person has (integer)
cb_has_phd	if a person has PhD degree or not (0 = doesn’t have, 1 = has)
cb_num_phds	the number of PhD degrees a person has (integer)
cb_total_jobs_bef_entre	the number of jobs a person had before becoming entrepreneur (integer)

Table 4.21: Prediction tool’s required user input

The output is a 0 or 1 (0 = unsuccessful in receiving funding, 1=successful in receiving funding) for each model, including each model’s prediction probability or in other words, confidence.

Furthermore, using Django Web Framework, we created a very simple website in order to create a friendly user environment as an online tool where the users can enter the required input, submit it, and wait some time for the predictions to be displayed. Some figures of this are provided below:

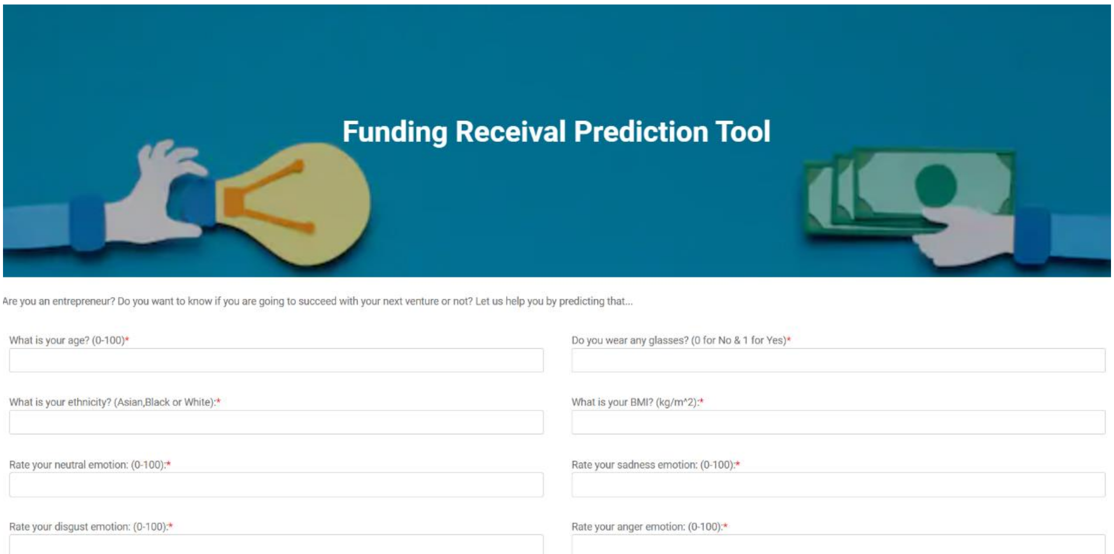

Figure 4.14: Part of Web Application display incomplete

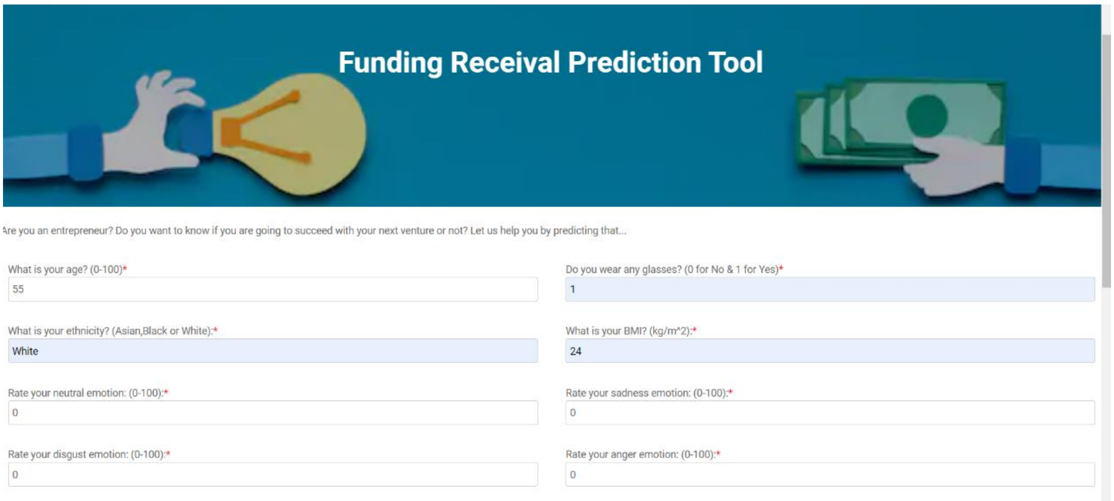

Figure 4.15: Part of Web Application display completed

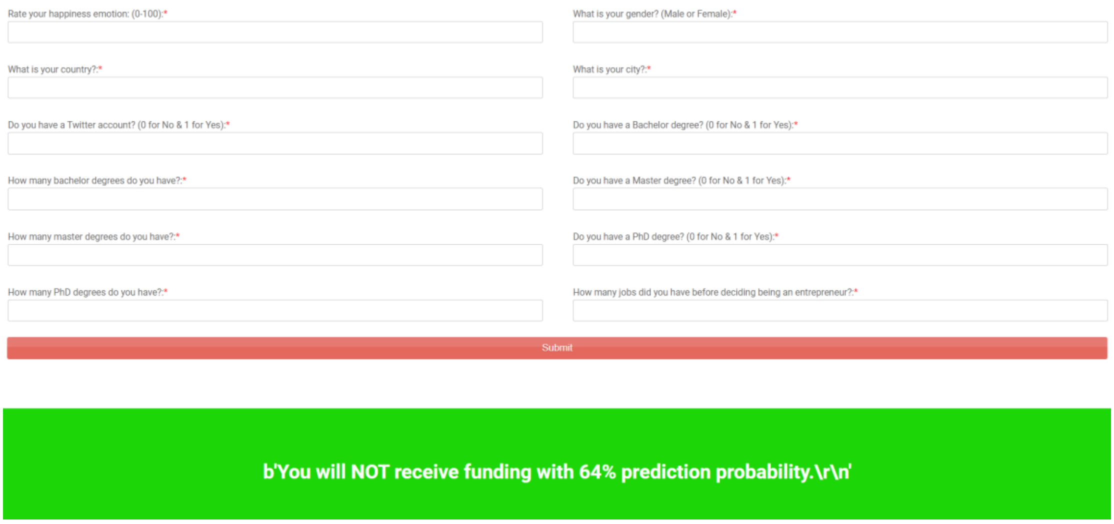

Figure 4.16: Part of Web Application display completed and final result

Also, the architecture of this online prediction tool is displayed below in figure 4.17, even though, due to limited time Face++ tool which would take as input a user's photo and calculate face ratios and emotions, was not used. Once the required input is inserted by the user, including here the image which will be processed by Face++ and Face-to-BMI extraction tool, it is sent to Django server where our prediction script is executed. After it is finished all the results are filtered by choosing the best prediction score based on the results, receive funding or not, and it is displayed to the user.

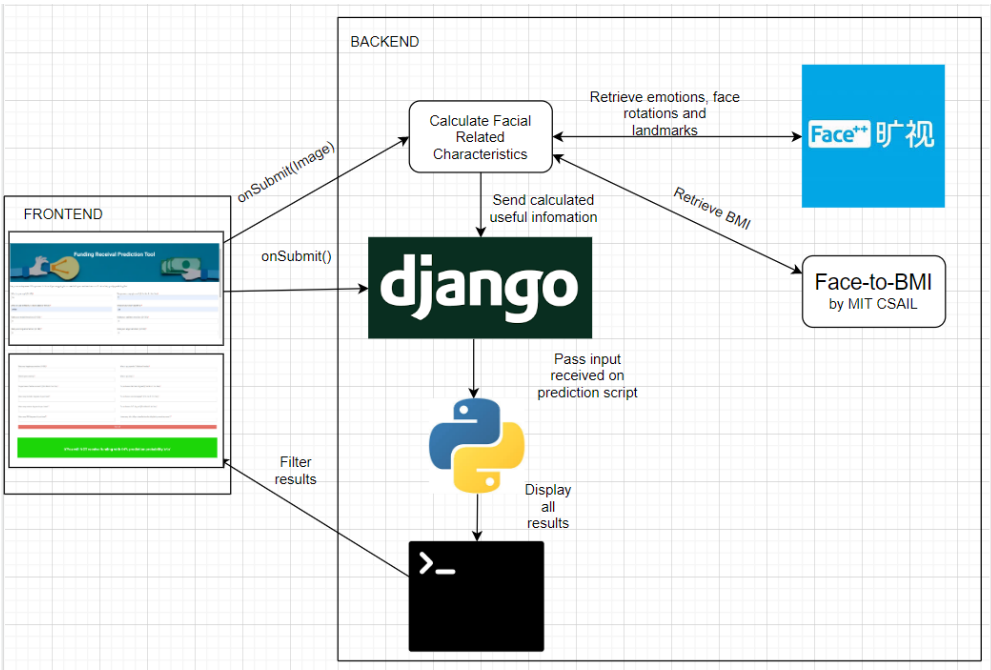

Figure 4.17: Overview of prediction tool's architecture

Chapter 5

Conclusion

Contents

5.1 Conclusion

Taking everything into consideration, the goal of this study, to predict whether an entrepreneur will successfully receive a funding or not based on specific information about them, is achieved and we offer some valuable information which will help analysts to address this very challenging problem. Specifically, the extraction of the characteristics of entrepreneurs, who were predicted to successfully receive funding, will help analysts to try and experiment more on the part of their data which are similar or have a relation with these characteristics. They will have a specific range of features in which they will focus, rather than the whole dataset with numberless features. Considering the analysts who work with the similar Crunchbase dataset, they will now have a features' baseline to define when an entrepreneur has the potential of receiving funding and may try other techniques to increase or even limit even more these features' range for a more precise funding receival success definition. Furthermore, the created tool can predict with a relatively high probability if a person has the potential of receiving funding or not just by submitting some specific information about themselves that will be requested.

The variety of dataset preprocessing methods that we used and the amount of our different experimented models give a trendsetting outcome. The experiments we cover suggest the use of different models and the selection of their appropriate parameters to achieve the highest possible score of this pediction. Moreover, we figure out that data is the most important part when it comes to this kind of classification problems and that the dataset's impact on prediction is much higher than the machine learning models themselves.

Finally, we managed to make a possible prediction of a person receiving a funding or not, with 69% confidence and managed to extract some of the most important characteristics of entrepreneurs who were predicted to successfully receive funding based on the given dataset, a thing that we hope to be used by other analysts on this issue.

5.2 Future Work

This study approaches the entrepreneurial funding receival success classification challenge with an incomplete dataset. There may exist some datasets with more complete information and better related to the final classification result which may make the prediction more accurate by using the same techniques. Moreover, there may also be some better preprocessing techniques of the given dataset from Crunchbase which will select some more useful features for the prediction and will manage to find and remove the real outliers. Furthermore, a great future work would be the implementation of a more suitable Neural Network model, which could give as a result a more precise prediction of whether an entrepreneur will receive funding or not.

In addition, the connection of the web application with the Face++ and Face-to-BMI tools must be finalized in order for the application to be more user-friendly by making the input process easier for the users. Finally, finding the importance of each extracted characteristic and the impact that it has on entrepreneurial funding receival success would be a very interesting and informative future work. The outcome of a more precise prediction will be extremely valuable for the community and will help us understand further what it takes for an entrepreneur to successfully receive a funding or not.

Bibliography

[1] Amanda Bowman. 2020. Essential Facts and Statistics Every Entrepreneur Must Know. (March 2020). Retrieved April 27, 2020 from https://www.crowdspring.com/blog/entrepreneur-statistics/

[2] Asaithambi, S. (2017, December 22). Why, How and When to Scale your Features. Retrieved from https://medium.com/greyatom/why-how-and-when-to-scale-your-features-4b30ab09db5e

[3] Barbara Orser and Lorraine Dyke. 2009. The Influence of Gender and Occupational-Role on Entrepreneurs' and Corporate Managers' Success Criteria. *Journal of Small Business & Entrepreneurship* 22, 3 (2009), 327–353. DOI:http://dx.doi.org/10.1080/08276331.2009.10593459

[4] Chenchen Pan, Yuan Gao & Yuzi Luo. Machine Learning Prediction of Companies' Business Success. Retrieved April 7, 2020 from http://cs229.stanford.edu/proj2018/report/88.pdf

[5] Dominika Wach, Ute Stephan, and Marjan Gorgievski. 2016. More than money: Developing an integrative multi-factorial measure of entrepreneurial success. *International Small Business Journal: Researching Entrepreneurship* 34, 8 (2016), 1098–1121. DOI:http://dx.doi.org/10.1177/0266242615608469

[6] Francisco Ramadas da Silva Ribeiro Bento. Anon. Predicting Start-up Success with Machine Learning. Retrieved April 7, 2020 fromhttps://run.unl.pt/bitstream/10362/33785/1/TGI0132.pd

[7] Groco Staff Writer / About Author More posts by Groco Staff Writer and More posts by Groco Staff Writer. 2020. The Qualities That Define A Successful Entrepreneur - Advisors to the Ultra-Affluent. (January 2020). Retrieved April 7, 2020 from https://groco.com/readingroom/lead-successful-entrepreneu

[8] Guang Xiang, Zeyu Zheng, Miaomiao Wen, Jason I Hong, Carolyn Penstein Rosé, and Chao Liu. A supervised approach to predict company acquisition with factual and

topic features using profiles and news articles on techcrunch. Retrieved April 7, 2020 from https://www.cs.cmu.edu/~guangx/papers/icwsm12-short.pdf

[9] Hyperparameter optimization. (2020, March 31). Retrieved from https://en.wikipedia.org/wiki/Hyperparameter_optimization

[10] Liang Yuxian Eugene and Soe-Tsyr Daphne Yuan. Where's the money? the social behavior of investors in Facebook's small world. In Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining Retrieved April 7, 2020 from https://dl.acm.org/doi/10.1109/ASONAM.2012.36

[11] Marjan J. Gorgievski, M.Evelina Ascalon, and Ute Stephan. 2011. Small Business Owners Success Criteria, a Values Approach to Personal Differences. *Journal of Small Business Management* 49, 2 (2011), 207–232. DOI:http://dx.doi.org/10.1111/j.1540-627x.2011.00322.x

[12] Matt Mansfield. 2019. STARTUP STATISTICS - The Numbers You Need to Know. (March 2019). Retrieved April 23, 2020 from https://smallbiztrends.com/2019/03/startup-statistics-small-business.html

[13] Nicolaou N., Stefanidis, D., Pallis G., Dikaiakos M. & Charitonons S. "Facial Structure and Entrepreneurship".

[14] Pablo Angel, Anna Jenkins, and Anna Stephens. 2018. Understanding entrepreneurial success: A phenomenographic approach. *International Small Business Journal: Researching Entrepreneurship* 36, 6 (2018), 611–636. DOI:http://dx.doi.org/10.1177/0266242618768662

[15] Rachana Chattopadhyay and Anjali Ghosh. 2002. Predicting Entrepreneurial Success. *The Journal of Entrepreneurship* 11, 1 (2002), 21–31. DOI:http://dx.doi.org/10.1177/097135570201100102

[16] Rosemary Fisher, Alex Maritz, and Antonio Lobo. 2014. Evaluating entrepreneurs' perception of success. *International Journal of Entrepreneurial Behavior & Research* 20, 5 (2014), 478–492. DOI:http://dx.doi.org/10.1108/ijebr-10-2013-0157

[17] Sarikas, C. (n.d.). Independent and Dependent Variables: Which Is Which? Retrieved April 7, 2020 from https://blog.prepscholar.com/independent-and-dependent

[18] Scott Shane. 2016. Why Small Business Failure Rates are Declining. (January 2016). Retrieved April 7, 2020 from https://www.entrepreneur.com/article/254871

[19] Shane S., Nicolaou N., Stefanidis, D., Pallis G., Dikaiakos M. & Conley M. "Trust me, I'm an Entrepreneur".

[20] Wei, C.-P., Jiang, Y.-S., & Yang, C.-S. (2009). Patent Analysis for Supporting Merger and Acquisition (M&A) Prediction: A Data Mining Approach. *Designing E-Business Systems. Markets, Services, and Networks Lecture Notes in Business Information Processing*, 187–200. doi: 10.1007/978-3-642-01256-3_16

[21] Yang, S., & Berger, R. (2017). Relation between start-ups' online social media presence and fundraising. *Journal of Science and Technology Policy Management*, *8*(2), 161–180. doi: 10.1108/jstpm-09-2016-0022

[22] Yuxian Eugene Liang and Soe-Tsyr Daphne Yuan. 2016. Predicting investor funding behavior using crunchbase social network features. *Internet Research* 26, 1 (2016), 74–100. DOI:http://dx.doi.org/10.1108/intr-09-2014-0231

Appendices

Appendix A

Prediction and prediction's probability of each model of first 25 rows of dataset

Using Logistic Regression:

fc_face_a	fc_wear_g	fc_yaw_ar	fc_pitch_a	fc_roll_an	fc_BMI	fc_neutral	fc_sadnes	fc_disgust	fc_anger_	fc_surprise	fc_fear_er	fc_happine	fc_fWHR_	fc_fWHR_	fc_fWHR_	fc_FA_rati	cb_has_Tv	cb_has_ba	cb_has_m	cb_has_ph	cb_total_j	fc_ethnicit
63.51351	0	47.97563	15.82107	31.09296	26.9286	4.288043	0.616686	57.73439	12.37004	0.134753	0.138448	28.568	1.816469	0.499616	0.409285	26.66667	0	1	1	1	0	1
41.89189	0	39.77374	17.86155	39.60341	25.28989	0.002	0.026627	0.10421	0.006182	0.02533	0.220684	99.632	1.565043	0.510756	0.410734	25.41935	0	1	1	0	0	1
48.64865	0	40.14834	16.82048	46.71509	25.42773	16.32516	0.479289	0.532746	0.463683	3.068927	0.468433	78.796	1.321764	0.453867	0.418118	27.69892	0	1	1	0	0	1
52.7027	0	39.34365	3.750567	44.8497	18.40223	0.256003	39.7725	12.33611	0.96137	0.210742	1.29912	48.41	1.698139	0.254286	0.04084	25.80645	1	1	1	0	2.0625	1
35.13514	0	39.92195	15.80728	48.40365	18.55731	0.018	0.019172	0.515732	0.054612	0.358666	0.283142	98.799	1.845071	0.559557	0.348527	26.70968	0	1	0	0	2.0625	1
36.48649	0	44.25272	10.6059	41.37976	12.11134	91.60092	1.073608	4.868089	0.445136	0.068896	0.070785	2.247	1.472393	0.618297	0.375812	26.70968	1	1	1	0	2.0625	1
33.78378	1	35.73287	10.17791	43.99707	22.65697	9.398094	23.83985	25.19965	32.71749	0.458971	5.577474	6.958	2.45354	0.626433	0.123967	27.44086	1	1	1	0	4.125	1
31.08108	0	44.71365	17.93355	39.06246	21.32411	0.007	0.047929	0.807095	0.007213	0.00304	0.019778	99.16	1.451535	0.41718	0.420301	26.10753	0	1	1	0	0	1
68.91892	0	45.92811	20.39031	45.29115	28.15079	0.01	0.010651	0.286045	1.121083	0.010132	0.033311	98.581	1.688681	0.50468	0.712741	23.91398	0	1	1	1	0	1
32.43243	0	37.73901	22.61389	59.75627	4.724495	0.001	0.00213	0.00319	0.002061	0.004053	0.012492	99.976	1.605175	0.416101	0.534566	24.30108	1	1	1	0	0	1
72.97297	0	42.61537	18.157	44.30455	21.93202	0.011	0.115029	0.517859	0.058733	0.223913	7.56467	91.849	1.80741	0.542278	0.412515	27.22581	0	1	1	1	0	1
41.89189	0	43.75879	17.93374	46.27681	20.5663	73.55274	1.491122	1.488712	6.460654	5.529945	1.457347	10.518	0.808363	0.596442	0.656757	26.58065	0	1	1	1	0	0
56.75676	0	44.33182	18.3944	44.47155	26.43422	0.002	0.00213	0.029774	0.002061	0.002026	0.005205	99.959	1.407562	0.675519	0.371963	26.10753	1	1	1	0	2.0625	1
56.75676	0	46.98705	16.15931	40.03707	28.72837	0.062001	0.066035	0.240321	0.142196	0.031409	0.062458	99.421	1.27852	0.38871	0.331795	25.41935	0	1	0	0	0	1
17.56757	0	44.66806	13.58256	42.27606	25.87458	56.57257	6.050762	2.769005	0.111284	0.290783	0.112424	34.64	1.341842	0.502262	0.331625	25.07527	0	1	0	1	2.0625	0
32.43243	0	49.11308	19.49369	19.07987	29.76	8.738087	1.679643	55.56832	12.7111	0.724425	2.101702	22.358	1.546249	0.646875	0.515507	16.94624	1	1	1	0	0	0
67.56757	0	35.53933	13.68329	50.74581	22.26295	43.49743	47.07687	1.170766	0.620305	0.258361	0.429917	9.931	1.253029	0.403424	0.296991	23.39785	0	1	1	1	6.1875	1
36.48649	1	50.19969	11.84745	41.19052	28.63155	0.496005	0.055385	0.611435	0.322517	0.052685	0.077031	98.437	1.075628	0.445038	0.243283	24.04301	1	1	1	0	4.125	0

cb_gender	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	is_Success	is_success	is_not_succ
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.797374	0.202626
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.791605	0.208395
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.786611	0.213389
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0.764494	0.235506
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.76237	0.23763
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0.760995	0.239005
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0.749948	0.250052
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.746475	0.253525
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0.745793	0.254207
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0.743165	0.256835
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.742388	0.257612
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.740847	0.259153
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.736181	0.263819
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.734167	0.265833
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.732555	0.267445
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.730058	0.269942
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.729524	0.270476
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.729468	0.270532
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0.728807	0.271193
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.726716	0.273284
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.72588	0.27412
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.723407	0.276593
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.722472	0.277528
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.720326	0.279674

Using Linear Discriminant Analysis:

fc_face_a	fc_wear_g	fc_yaw_ar	fc_pitch_a	fc_roll_an	fc_BMI	fc_neutral	fc_sadnes	fc_disgust	fc_anger_	fc_surprise	fc_fear_er	fc_happine	fc_fWHR_	fc_fWHR_	fc_fWHR_	fc_FA_rati	cb_has_Tv	cb_has_ba	cb_has_m	cb_has_ph	cb_total_j	fc_ethnicit
48.64865	0	40.14834	16.82048	46.71509	25.42773	16.32516	0.479289	0.532746	0.463683	3.068927	0.468433	78.796	1.321764	0.453867	0.418118	27.69892	0	1	1	0	0	1
52.7027	0	39.34365	3.750567	44.8497	18.40223	0.256003	39.7725	12.33611	0.96137	0.210742	1.29912	48.41	1.698139	0.254286	0.04084	25.80645	1	1	1	0	2.0625	1
41.89189	0	39.77374	17.86155	39.60341	25.28989	0.002	0.026627	0.10421	0.006182	0.02533	0.220684	99.632	1.565043	0.510756	0.410734	25.41935	0	1	1	0	0	1
63.51351	0	47.97563	15.82107	31.09296	26.9286	4.288043	0.616686	57.73439	12.37004	0.134753	0.138448	28.568	1.816469	0.499616	0.409285	26.66667	0	1	1	1	0	1
35.13514	0	39.92195	15.80728	48.40365	18.55731	0.018	0.019172	0.515732	0.054612	0.358666	0.283142	98.799	1.845071	0.559557	0.348527	26.70968	0	1	0	0	2.0625	1
32.43243	0	37.73901	22.61389	59.75627	4.724495	0.001	0.00213	0.00319	0.002061	0.004053	0.012492	99.976	1.605175	0.416101	0.534566	24.30108	1	1	1	0	0	1
36.48649	0	44.25272	10.6059	41.37976	12.11134	91.60092	1.073608	4.868089	0.445136	0.068896	0.070785	2.247	1.472393	0.618297	0.375812	26.70968	1	1	1	0	2.0625	1
31.08108	0	44.71365	17.93355	39.06246	21.32411	0.007	0.047929	0.807095	0.007213	0.00304	0.019778	99.16	1.451535	0.41718	0.420301	26.10753	0	1	1	0	0	1
33.78378	1	35.73287	10.17791	43.99707	22.65697	9.398094	23.83985	25.19965	32.71749	0.458971	5.577474	6.958	2.45354	0.626433	0.123967	27.44086	1	1	1	0	4.125	1
68.91892	0	45.92811	20.39031	45.29115	28.15079	0.01	0.010651	0.286045	1.121083	0.010132	0.033311	98.581	1.688681	0.50468	0.712741	23.91398	0	1	1	1	0	1
72.97297	0	42.61537	18.157	44.30455	21.93202	0.011	0.115029	0.517859	0.058733	0.223913	7.56467	91.849	1.80741	0.542278	0.412515	27.22581	0	1	1	1	0	1
56.75676	0	46.98705	16.15931	40.03707	28.72837	0.062001	0.066035	0.240321	0.142196	0.031409	0.062458	99.421	1.27852	0.38871	0.331795	25.41935	0	1	0	0	0	1
52.7027	0	38.64682	19.41054	47.7733	21.6	0.009	0.020237	1.924692	0.121588	0.145898	2.445219	95.551	1.584987	0.382965	0.321878	26.70968	0	1	1	0	0	1
67.56757	0	35.53933	13.68329	50.74581	22.26295	43.49743	47.07687	1.170766	0.620305	0.258361	0.429917	9.931	1.253029	0.403424	0.296991	23.39785	0	1	1	1	6.1875	1
41.89189	0	43.75879	17.93374	46.27681	20.5663	73.55274	1.491122	1.488712	6.460654	5.529945	1.457347	10.518	0.808363	0.596442	0.656757	26.58065	0	1	1	1	0	0
17.56757	0	44.66806	13.58256	42.27606	25.87458	56.57257	6.050762	2.769005	0.111284	0.290783	0.112424	34.64	1.341842	0.502262	0.331625	25.07527	0	1	0	1	2.0625	0
55.40541	0	33.24832	9.692594	44.61087	35.46815	32.70233	6.151945	6.703459	1.991777	1.95848	2.012179	49.42	1.563702	0.656513	0.199575	24.90323	0	0	0	1	0	1
32.43243	0	49.11308	19.49369	19.07987	29.76	8.738087	1.679643	55.56832	12.7111	0.724425	2.101702	22.358	1.546249	0.646875	0.515507	16.94624	1	1	1	0	0	0
39.18919	0	41.93828	19.32564	45.8991	32.35535	0.002	0.00213	0.002127	0.005152	0.517736	0.002082	99.477	1.86513	0.709593	0.520579	26.88172	0	1	0	0	2.0625	1
41.89189	0	42.64756	17.92046	41.99459	28.58374	0.01	0.010651	0.287109	0.042247	0.017224	0.027065	99.626	1.74791	0.449433	0.418114	27.35484	0	1	0	0	0	1
56.75676	0	44.33182	18.3944	44.47155	26.43422	0.002	0.00213	0.029774	0.002061	0.002026	0.005205	99.959	1.407562	0.675519	0.371963	26.10753	1	1	1	0	2.0625	1
33.78378	0	47.2437	16.73995	42.59874	29.29198	0.01	0.919171	15.24973	0.9861	0.708214	2.151668	81.063	1.196929	0.421706	0.41852	25.07527	0	1	1	0	0	0
35.13514	0	40.40858	18.40629	44.25497	21.02599	4.917049	0.262012	11.86078	0.391555	0.118542	0.18425	83.01	1.784913	0.662684	0.380308	26.70968	1	1	0	0	4.125	0
24.32432	0	40.11885	19.03573	44.55663	25.51312	33.83534	37.62528	1.736477	0.390524	0.383996	0.394525	28.07	1.140583	0.733477	0.636391	27.95699	1	1	0	0	0	1

cb_gender	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	is_Success	is_success	is_not_succ
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.77593	0.22407
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0.775906	0.224094
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.763806	0.236194
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.759907	0.240093
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.736479	0.263521
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0.73599	0.26401
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0.733565	0.266435
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.731942	0.268058
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0.730711	0.269289
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0.730345	0.269655
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.726884	0.273116
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.724991	0.275009
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.720441	0.279559
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.714591	0.285409
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.712837	0.287163
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.712306	0.287694
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.70973	0.29027
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.708498	0.291502
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.707387	0.292613
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0.707135	0.292865
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.705969	0.294031
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.705674	0.294326
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.704465	0.295535
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0.703985	0.296015

Using k-Nearest Neighbors:

fc_face_a	fc_wear_g	fc_yaw_ar	fc_pitch_a	fc_roll_an	fc_BMI	fc_neutral	fc_sadnes	fc_disgust_	fc_anger_	fc_surprise	fc_fear_er	fc_happine	fc_fWHR_	fc_fWHR_	fc_fWHR_	fc_FA_rati	cb_has_Tv	cb_has_ba	cb_has_m	cb_has_ph	cb_total_j	fc_ethnicit
56.75676	0	45.23512	15.65843	42.6652	33.03624	71.63472	18.5272	3.666486	0.214325	0.97772	4.14511	2.367	1.46802	0.586387	0.400213	26.36559	0	0	0	0	0	1
35.13514	0	27.60313	20.98532	53.46447	15.53058	0.003	0.012781	0.061675	0.022669	0.00304	0.113465	99.793	1.70352	0.420439	0.682064	24	0	0	0	0	0	1
63.51351	0	47.97563	15.82107	31.09296	26.9286	4.288043	0.616686	57.73439	12.37004	0.134753	0.138448	28.568	1.816469	0.499616	0.409285	26.66667	0	1	1	1	0	1
50	1	35.90019	16.03284	46.86674	29.74971	0.016	0.023432	16.12595	3.398283	0.080041	1.303284	80.168	1.661062	0.455209	0.390443	27.6129	1	0	0	0	0	1
58.10811	0	38.94758	17.22172	42.48908	35.06508	0	0.003195	1.601429	0.200929	0.029382	1.306407	97.012	1.750036	0.53148	0.410471	25.93548	0	0	0	0	0	1
35.13514	0	41.84411	23.1302	43.79082	18.39477	0.005	0.135266	0.574218	0.113345	0.005066	0.200906	99.02	1.623929	0.566968	0.591222	24.43011	1	1	0	0	2.0625	1
66.21622	1	46.80332	15.92596	41.92673	27.07649	0.213002	0.436686	9.638349	56.32	0.659581	7.452246	27.844	1.350943	0.44418	0.435295	26.32258	0	0	0	0	0	1
60.81081	0	27.76567	20.1015	47.61172	27.00344	0.279003	0.233254	1.111217	0.086554	0.021277	0.03227	98.32	1.211018	0.492854	0.443414	20.90323	0	1	0	0	0	1
63.51351	0	52.09106	19.19749	50.33808	20.49999	88.84089	1.221655	2.228815	0.290575	0.157043	0.145735	7.34	0.657549	0.352031	0.420705	21.33333	0	0	0	0	2.0625	1
51.35135	0	39.1193	16.30347	38.85214	22.57905	29.5743	57.55413	9.629842	0.291605	0.029382	2.809556	4.323	1.682547	0.689564	0.352399	24.47312	0	0	0	0	0	1
54.05405	0	40.44548	18.31198	41.35111	19.10034	18.93619	0.626271	1.100584	0.382281	0.408312	0.49758	78.188	1.531428	0.487825	0.443284	26.92473	0	1	0	0	10.3125	1
50	0	32.97989	18.96569	57.51223	24.9383	0.04	0.042603	1.628013	0.55745	0.285717	1.014938	96.592	1.558731	0.802078	0.6124	24.90323	0	1	1	0	10.3125	1
43.24324	0	42.14305	17.72197	42.56174	27.88734	73.93774	9.105433	6.026095	5.263321	2.20063	1.671785	2.961	1.544459	0.655307	0.421273	26.53763	0	0	0	0	2.0625	1
32.43243	0	35.27321	19.74822	50.29026	27.24854	0.024	0.017041	1.586542	0.054612	0.016211	0.016655	98.383	1.814935	0.416249	0.431764	27.65591	1	1	1	0	2.0625	1
55.40541	0	40.6458	18.03503	41.64015	26.26871	0.833008	27.26837	0.263715	0.200929	0.19757	2.540988	70.484	1.439025	0.504318	0.435699	27.1828	1	1	0	0	0	1
72.97297	0	40.16381	20.27829	44.17937	21.33777	92.67893	4.961178	0.259461	0.085524	0.068896	0.102014	2.169	1.312231	0.571418	0.392959	27.09677	0	0	0	0	0	1
36.48649	0	41.82279	19.73289	43.96033	34.03716	83.36483	10.04697	1.524867	0.166926	0.11449	0.293551	5.211	1.72758	0.610956	0.482022	27.6129	1	1	0	0	0	1
58.10811	0	39.59125	20.24326	43.80247	31.79862	0.138001	0.118225	0.281792	0.015456	0.015198	0.015614	99.441	1.767241	0.655416	0.345213	27.13978	0	0	0	0	0	1
33.78378	0	47.2437	16.73995	42.59874	29.29198	0.01	0.919171	15.24973	0.9861	0.708214	2.151668	81.063	1.196929	0.421706	0.41852	25.07527	0	1	1	0	0	0
47.2973	0	46.45462	14.2558	53.93814	36.17164	48.75749	44.85829	3.160324	0.999495	0.408312	1.127362	3.698	1.476368	0.829108	0.303512	23.82796	0	0	0	0	0	0
28.37838	0	36.83707	18.88687	47.1186	34.08459	0.043	0.018106	0.280729	0.016487	0.016211	0.016655	99.628	1.760219	0.504519	0.425759	26.36559	0	0	0	0	0	1
63.51351	0	37.60167	19.63982	47.00925	23.67017	24.48724	0.771123	0.287109	0.27821	1.185422	0.310207	72.781	1.648011	0.575569	0.544152	27.35484	0	1	1	0	0	1
70.27027	1	39.96274	18.84283	47.597	36.74988	0.03	0.017041	1.211174	0.687282	0.057751	0.114506	97.98	1.610598	0.555817	0.503493	27.26882	0	1	0	1	0	1
58.10811	0	48.84816	13.27167	40.36808	24.29696	61.98962	20.0311	1.154815	0.365795	3.382	2.75959	11.773	0.874428	0.598118	0.538427	22.7957	0	0	0	0	0	0

cb_gender	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	is_Success	is_success	is_not_suc
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.605886	0.394114
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.605886	0.394114
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.605886	0.394114
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0.605886	0.394114
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.605886	0.394114
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.605886	0.394114
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.605886	0.394114
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0.605886	0.394114
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.605886	0.394114
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0.605886	0.394114
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.605886	0.394114
1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0.605886	0.394114
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.605886	0.394114
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.605886	0.394114
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0.605886	0.394114
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.605886	0.394114
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.605886	0.394114
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.605886	0.394114
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.605886	0.394114
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.605886	0.394114
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.605886	0.394114
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.605886	0.394114
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.605886	0.394114
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.605886	0.394114

Using Gaussian Naïve Bayes:

fc_face_a	fc_wear_g	fc_yaw_ar	fc_pitch_a	fc_roll_an	fc_BMI	fc_neutral	fc_sadness	fc_disgust	fc_anger_	fc_surprise	fc_fear_er	fc_happine	fc_fWHR_	fc_fWHR_	fc_fWHR_	fc_FA_rati	cb_has_Tv	cb_has_ba	cb_has_m	cb_has_ph	cb_total_j	fc_ethnicit
39.18919	0	42.58545	22.02966	40.30903	28.21349	0.517005	1.075738	0.387065	0.106132	0.104358	0.107219	97.8	1.437711	0.738681	0.715761	25.16129	1	0	0	0	0	1
48.64865	0	26.17953	17.05207	44.49192	26.15051	0.060001	0.673135	22.5944	32.41352	0.117529	7.714568	39.076	1.670484	0.818582	0.498567	24.25806	0	0	0	0	0	1
54.05405	0	46.66637	9.450767	48.48453	35.73228	19.23819	5.54165	1.182463	0.871725	0.366772	2.654453	70.689	1.599149	0.720756	0.233928	27.09677	1	0	0	0	2.0625	0
64.86486	0	24.27478	27.88811	44.74634	26.58286	0.512005	4.816326	6.815113	0.041216	0.02533	0.026024	88.467	1.777893	0.865302	0.521709	26.66667	0	0	0	0	2.0625	1
63.51351	0	47.97563	15.82107	31.09296	26.9286	4.288043	0.616686	57.73439	12.37004	0.134753	0.138448	28.568	1.816469	0.499616	0.409285	26.66667	0	1	1	1	0	1
35.13514	0	31.58189	19.41477	51.27728	28.03599	0.463005	0.102248	4.23007	0.390524	0.164135	0.243585	94.687	1.125618	0.439494	0.439004	25.72043	1	1	0	0	4.125	1
55.40541	1	48.93185	14.30836	49.14092	25.02704	57.45457	0.784969	0.651843	35.38213	4.544119	1.30953	1.115	1.580956	0.715785	0.318321	25.67742	0	0	1	0	2.0625	0
55.40541	0	33.24832	9.692594	44.61087	35.46815	32.70233	6.151945	6.703459	1.991777	1.95848	2.012179	49.42	1.563702	0.656513	0.199575	24.90323	0	0	0	1	0	1
56.75676	0	38.68881	16.43279	50.65179	25.64868	15.11915	0.082012	0.261588	0.079341	0.346508	2.039244	82.181	2.106498	0.704317	0.270484	27.56989	0	0	0	0	6.1875	1
28.37838	0	33.32393	18.71172	49.7001	31.99025	94.58595	0.167219	0.473198	2.343146	2.119576	0.200906	0.254	1.428722	0.67236	0.498353	26.19355	0	0	0	0	0	1
56.75676	0	46.98705	16.15931	40.03707	28.72837	0.062001	0.066035	0.240321	0.142196	0.031409	0.062458	99.421	1.27852	0.38871	0.331795	25.41935	0	1	0	0	0	1
40.54054	0	53.79814	16.15007	46.68482	33.76585	26.16726	62.4759	10.73043	0.337974	0.446813	1.921616	2.469	1.234542	0.766386	0.44345	22.75269	0	0	0	0	0	0
54.05405	1	28.64473	16.68267	46.08118	31.91074	41.83542	2.512541	3.040163	2.430731	40.95077	5.624317	4.767	1.438859	0.632675	0.517774	24.94624	1	1	0	0	2.0625	0
44.59459	1	48.2241	16.00087	32.26455	32.44014	0.909009	0.082012	14.82651	0.432771	0.207702	0.309166	84.148	1.56135	0.463172	0.327085	23.65591	0	1	0	0	0	1
43.24324	1	22.50924	17.57901	56.57717	23.30427	48.35848	8.576085	16.23654	4.622407	8.035542	10.95404	5.381	0.984152	0.247131	0.552749	21.89247	1	1	0	0	2.0625	0
44.59459	0	33.50345	17.5026	41.02846	25.45595	1.362014	3.396564	0.532746	0.323548	0.551171	0.398688	93.706	1.14527	0.610793	0.506542	24.73118	0	0	0	0	0	0
35.13514	0	40.40858	18.40629	44.25497	21.02599	4.917049	0.262012	11.86078	0.391555	0.118542	0.18425	83.01	1.784913	0.662684	0.380308	26.70968	1	1	0	0	4.125	0
45.94595	1	36.78452	19.68349	44.51896	26.59	0.247002	0.263077	1.478079	0.254511	45.49793	32.08557	22.14	1.464992	0.458157	0.355543	27.65591	0	0	0	0	0	1
52.7027	0	41.47136	8.210508	36.8591	31.02931	47.12847	50.28917	4.198169	0.293666	0.028369	0.319575	1.088	1.41688	0.482472	0.265867	25.07527	1	0	0	0	0	0
25.67568	0	30.11748	18.10858	48.3138	37.16091	81.01681	0.520828	1.07719	12.62661	1.090183	2.931349	1.336	1.716102	0.775916	0.407912	25.93548	1	0	1	0	2.0625	0
39.18919	0	41.93828	19.32564	45.8991	32.35535	0.002	0.00213	0.002127	0.005152	0.517736	0.002082	99.477	1.86513	0.709593	0.520579	26.88172	0	1	0	0	2.0625	1
48.64865	0	40.14834	16.82048	46.71509	25.42773	16.32516	0.479289	0.532746	0.463683	3.068927	0.468433	78.796	1.321764	0.453867	0.418118	27.69892	0	1	1	0	0	1
45.94595	0	37.62942	22.87489	41.44973	28.75341	0.006	0.028757	0.144618	0.041216	0.012158	0.298756	99.493	1.402154	0.517887	0.494166	26.66667	0	0	1	0	0	1
60.81081	1	45.19777	13.93185	42.85334	24.61603	1.573016	0.187455	25.96208	6.742986	0.877415	0.412221	66.031	1.372268	0.446844	0.290181	22.53763	1	0	0	0	0	0

cb_gender	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	is_Success	is_success	is_not_succe
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.552196	0.447804
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0.552196	0.447804

Using Decision Tree:

fc_face_a	fc_wear_g	fc_yaw_ar	fc_pitch_a	fc_roll_an	fc_BMI	fc_neutral	fc_sadnes	fc_disgust_	fc_anger_	fc_surprise	fc_fear_er	fc_happine	fc_fWHR_	fc_fWHR_	fc_fWHR_	fc_FA_rati	cb_has_Tv	cb_has_ba	cb_has_m	cb_has_ph	cb_total_j	fc_ethnicit
52.7027	1	29.29247	11.9453	42.66097	28.77664	1.531015	1.630649	2.252209	2.638873	2.5826	4.211732	85.664	1.664315	0.539142	0.269703	24.60215	0	1	0	0	0	0
48.64865	0	34.49918	24.10689	56.38443	30.19113	0.079001	0.204497	48.12263	14.22786	0.194531	0.857753	39.65	1.900584	0.79939	0.692717	25.07527	0	0	0	0	0	1
56.75676	0	35.75507	17.15446	37.28753	29.41962	0.002	0.029822	0.089323	0.013395	0.005066	0.268568	99.61	1.817646	0.502187	0.528979	24.21505	0	0	0	0	0	1
37.83784	0	29.71293	18.50206	47.50426	27.37328	90.72691	4.476563	2.602057	0.459562	0.070923	0.138448	1.974	1.050841	0.607271	0.502794	26.49462	0	0	0	0	0	1
44.59459	1	45.85768	8.491142	45.94263	29.08485	41.45441	2.721299	3.858955	0.655339	9.755925	6.141675	36.197	1.384925	0.572818	0.15012	25.76344	0	0	0	0	0	1
59.45946	0	41.75872	19.08878	38.21657	22.52631	0.019	1.727572	1.005944	0.02576	0.020264	0.430958	96.953	1.095147	0.3664	0.270156	24.60215	0	0	0	0	8.25	1
70.27027	1	43.50074	18.25079	48.39943	18.87631	88.43688	2.784139	0.798588	0.773836	3.302972	1.906001	2.357	1.052225	0.487077	0.39735	27.09677	0	0	0	1	0	1
36.48649	1	38.93765	22.40286	37.05151	24.11512	0.056001	0.675265	1.138865	0.955188	0.08612	11.87946	85.816	1.587081	0.689242	0.592365	25.50538	1	0	0	0	4.125	1
36.48649	0	42.73463	21.16853	45.05294	14.26218	0.014	0.014911	6.045236	4.459603	0.169201	0.950398	88.88	1.464793	0.674174	0.61781	27.22581	0	0	0	0	0	1
33.78378	0	46.83414	12.81296	51.13157	22.12516	0.026	0.033018	0.320073	0.020608	0.009119	0.009369	99.605	1.038988	0.409977	0.28621	25.37634	1	1	1	0	0	1
63.51351	1	34.07895	10.59209	45.26387	25.46934	60.62961	15.94862	8.581363	1.170543	0.168188	3.325873	11.829	1.465164	0.436686	0.272645	27.52688	0	1	0	0	0	1
59.45946	0	47.90983	12.50224	34.03568	21.2807	53.74254	6.057153	0.575281	0.461622	2.350581	3.590277	33.812	1.060564	0.447908	0.382963	25.97849	0	1	0	0	2.0625	1
50	1	42.30396	8.47479	43.70999	30.60261	32.98633	4.366859	1.410023	0.586302	14.34057	42.69401	5.852	1.705573	0.435946	0.118161	26.4086	0	0	0	1	0	1
62.16216	0	32.11726	19.39126	46.22635	29.77471	0.001	0.008521	0.081879	0.02473	0.00304	0.188414	99.707	1.781498	0.665449	0.553681	25.67742	1	1	0	0	4.125	1
63.51351	0	40.71403	24.61166	44.50954	17.50632	0.028	0.24923	11.65874	10.76879	0.08612	1.924738	76.389	1.980959	0.736713	0.890877	27.35484	1	1	0	0	2.0625	1
60.81081	0	41.3453	13.4046	42.29832	24.87447	7.584076	24.49488	1.710956	0.362703	0.573461	0.455941	66.454	1.883246	0.795826	0.279187	27.01075	0	0	0	0	2.0625	1
47.2973	0	33.2114	19.01178	65.44616	32.50797	0.113001	0.197041	0.309439	0.307061	0.117529	0.329985	98.68	1.883717	0.807842	0.53384	24.73118	0	0	0	0	0	1
47.2973	1	39.77311	10.00964	48.85735	32.11042	0.96601	96.1103	0.278602	0.269967	0.265454	0.463228	7.568	1.528245	0.595213	0.242809	26.45161	0	0	0	0	0	1
45.94595	1	43.84338	9.04466	44.77404	35.00243	66.27966	21.14625	4.420412	0.780018	4.024357	5.03201	0.146	1.351176	0.509362	0.309581	26.62366	1	1	0	0	0	1
67.56757	0	46.67256	8.334644	42.11777	25.90637	0.124001	1.844732	5.810232	0.350338	0.029382	0.269609	92.051	1.298649	0.395909	0.309219	25.63441	0	1	1	0	0	1
28.37838	1	48.31889	20.27269	42.17252	22.56881	0.193002	0.648638	9.939282	50.85987	0.140832	1.798782	38.626	1.179603	0.503225	0.586379	24.12903	1	0	0	0	0	0
83.78378	0	34.73302	13.48855	49.50135	38.23094	2.110021	2.385796	33.66191	3.557996	0.347521	12.57794	48.115	1.309888	0.415257	0.329926	26.88172	0	0	0	0	0	0
52.7027	0	41.47136	8.210508	36.8591	31.02931	47.12847	50.28917	4.198169	0.293666	0.028369	0.319575	1.088	1.41688	0.482472	0.265867	25.07527	1	0	0	0	0	0
43.24324	0	29.84271	16.17658	54.7368	31.00133	0.390004	1.418697	1.045289	0.034003	0.027356	0.077031	97.162	1.66328	0.729434	0.479687	25.37634	1	0	0	0	0	1

cb_gender	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	is_Success	is_success	is_not_succe
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0.541624	0.458376
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.541624	0.458376
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.541624	0.458376
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.541624	0.458376
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.541624	0.458376
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.541624	0.458376
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.541624	0.458376
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.538274	0.461726
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.538274	0.461726
1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.538274	0.461726
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.538274	0.461726
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.538274	0.461726
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0.538274	0.461726
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0.535928	0.464072
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.535928	0.464072
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.535928	0.464072
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.535928	0.464072
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.535928	0.464072
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.535928	0.464072
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.535928	0.464072
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0.535928	0.464072
0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0.535928	0.464072
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0.535928	0.464072
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0.535928	0.464072

Using Gradient Boosting:

fc_face_a	fc_wear_g	fc_yaw_ar	fc_pitch_a	fc_roll_an	fc_BMI	fc_neutral	fc_sadnes	fc_disgust	fc_anger_	fc_surprise	fc_fear_er	fc_happine	fc_fWHR_	fc_fWHR_	fc_fWHR_	fc_FA_rati	cb_has_Tv	cb_has_ba	cb_has_ma	cb_has_ph	cb_total_j	fc_ethnicit
58.10811	0	40.25593	7.01009	45.75143	46.88872	16.37216	63.28111	13.67489	6.934641	0.561303	2.784573	1.395	1.786611	0.60794	0.118538	27.52688	0	0	0	0	0	1
40.54054	0	84.39456	20.80898	36.55274	24.59442	26.31526	44.32042	0.773067	2.363754	3.885551	25.6441	0.583	0.324402	0.274377	0.272696	0	1	1	1	0	0	0
60.81081	0	32.31045	25.33066	61.80662	34.72039	0.055001	0.122485	17.56468	25.08424	0.142859	0.784885	58.073	2.07293	0.848285	0.73864	18.7957	0	1	0	0	0	1
22.97297	0	51.31054	15.39681	38.24705	31.60221	5.334053	0.314201	17.39348	0.30397	0.725438	0.307084	76.708	1.594692	0.764149	0.362814	21.41935	0	1	0	0	0	1
35.13514	0	40.40858	18.40629	44.25497	21.02599	4.917049	0.262012	11.86078	0.391555	0.118542	0.18425	83.01	1.784913	0.662684	0.380308	26.70968	1	1	0	0	4.125	0
47.2973	0	70.62559	10.51907	60.45092	25.65837	35.42135	14.2775	3.391074	2.064936	9.106475	26.98798	11.067	0.014091	0.158408	0.102649	24.55914	0	0	0	0	2.0625	1
62.16216	1	63.66671	5.248324	60.48709	28.99908	38.45638	0.138461	28.21748	0.63164	0.447826	0.160308	33.668	1.5307	0.376533	0.189662	12.30108	0	0	0	0	0	1
45.94595	0	37.62942	22.87489	41.44973	28.75341	0.006	0.028757	0.144618	0.041216	0.012158	0.298756	99.493	1.402154	0.517887	0.494166	26.66667	0	0	1	0	0	1
41.89189	0	42.16815	13.12275	44.84463	24.88654	0.025	0.085207	2.26497	0.281301	0.02533	1.631187	95.899	1.73132	0.361084	0.215907	27.6129	0	0	0	0	0	1
39.18919	0	39.71806	19.56705	46.43559	27.65386	0.006	0.006391	0.051042	0.006182	0.006079	0.056212	99.875	1.685373	0.476333	0.448513	27.65591	0	0	0	0	0	1
40.54054	0	44.5487	18.10518	40.7351	29.43259	21.04021	1.311123	3.565466	1.268431	7.926119	9.101129	56.58	1.434015	0.598877	0.339175	24.77419	0	0	1	0	2.0625	1
59.45946	0	36.97727	13.38412	42.5905	43.78641	0.03	2.753251	16.31841	0.314274	0.009119	2.886587	78.952	1.705045	0.490809	0.380901	27.01075	0	0	0	1	2.0625	1
55.40541	0	36.82137	17.40865	48.54959	23.13283	0.01	0.657159	1.815166	0.409072	0.06687	25.94285	72.281	1.508378	0.452007	0.45685	27.44086	1	1	0	0	0	1
35.13514	0	31.58189	19.41477	51.27728	28.03599	0.463005	0.102248	4.23007	0.390524	0.164135	0.243585	94.687	1.125618	0.439494	0.439004	25.72043	1	1	0	0	4.125	1
55.40541	0	33.24832	9.692594	44.61087	35.46815	32.70233	6.151945	6.703459	1.991777	1.95848	2.012179	49.42	1.563702	0.656513	0.199575	24.90323	0	0	0	1	0	1
40.54054	0	53.79814	16.15007	46.68482	33.76585	26.16726	62.4759	10.73043	0.337974	0.446813	1.921616	2.469	1.234542	0.766386	0.44345	22.75269	0	0	0	0	0	0
44.59459	0	38.69681	21.76949	35.37755	30.19214	0.005	0	0	0.00103	0	0	99.992	1.782523	0.476969	0.456882	25.16129	1	1	1	0	0	1
32.43243	0	49.11308	19.49369	19.07987	29.76	8.738087	1.679643	55.56832	12.7111	0.724425	2.101702	22.358	1.546249	0.646875	0.515507	16.94624	1	1	1	0	0	0
41.89189	1	28.14534	11.0811	63.16375	25.51534	23.37923	4.370054	11.42374	0.619275	0.850059	3.790142	56.693	1.123866	0.629661	0.27446	24.73118	0	1	0	0	0	1
63.51351	0	47.97563	15.82107	31.09296	26.9286	4.288043	0.616686	57.73439	12.37004	0.134753	0.138448	28.568	1.816469	0.499616	0.409285	26.66667	0	1	1	1	0	1
37.83784	0	39.64881	20.15016	45.76354	21.95111	2.177022	9.79987	0.957029	0.058733	0.057751	2.733566	84.982	1.135797	0.427397	0.494197	27.69892	1	1	1	0	8.25	1
50	0	40.94762	18.32433	48.54271	19.39282	0.154002	2.517867	4.057805	0.175169	0.183386	1.43965	91.932	1.76491	0.474743	0.434392	27.78495	0	0	1	0	4.125	1
39.18919	0	41.93828	19.32564	45.8991	32.35535	0.002	0.00213	0.002127	0.005152	0.517736	0.002082	99.477	1.86513	0.709593	0.520579	26.88172	0	1	0	0	2.0625	1
54.05405	1	0	9.190582	28.19947	25.97709	65.46065	6.849578	1.716273	1.581675	4.9808	20.26336	0.577	0.521389	0.20716	0.185498	23.22581	0	0	0	0	0	1

cb_gender	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	is_Success	is_success	is_not_succ
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0.603171	0.396829
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0.601939	0.398061
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.601329	0.398671
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.599671	0.400329
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.599006	0.400994
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.596647	0.403353
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.591723	0.408277
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.590439	0.409561
0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0.589941	0.410059
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.589197	0.410803
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.588925	0.411075
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.58826	0.41174
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.588157	0.411843
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.587494	0.412506
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.586762	0.413238
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.586192	0.413808
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.586074	0.413926
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.585902	0.414098
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.585254	0.414746
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.585175	0.414825
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.583111	0.416889
0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0.583019	0.416981
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.582966	0.417034
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.582666	0.417334

Using Random Forest:

fc_face_a	fc_wear_g	fc_yaw_ar	fc_pitch_a	fc_roll_an	fc_BMI	fc_neutral	fc_sadness	fc_disgust	fc_anger_	fc_surprise	fc_fear_er	fc_happine	fc_fWHR_	fc_fWHR_	fc_fWHR_	fc_FA_rati	cb_has_Tv	cb_has_ba	cb_has_m	cb_has_ph	cb_total_j	fc_ethnicit
35.13514	1	44.66873	18.17218	46.68747	23.84499	59.8456	9.451586	0.695441	0.348278	0.873362	23.82449	6.54	1.123888	0.478919	0.401421	26.10753	0	0	0	0	0	0
35.13514	0	27.60313	20.98532	53.46447	15.53058	0.003	0.012781	0.061675	0.022669	0.00304	0.113465	99.793	1.70352	0.420439	0.682064	24	0	0	0	0	0	1
60.81081	1	38.70074	18.09912	52.47659	24.26622	0.017	0.018106	29.46055	27.81585	0.02533	0.372664	44.883	1.473189	0.328253	0.438261	25.11828	0	0	0	0	0	1
51.35135	0	39.11596	19.64338	43.89515	27.34623	0.04	0.042603	0.975107	0.041216	0.040527	0.041638	98.885	1.341897	0.394865	0.493623	27.01075	1	1	0	0	2.0625	0
67.56757	0	40.01135	18.27152	43.67613	24.37776	0.028	0.11929	0.685871	0.122618	0.080041	2.838703	96.291	1.961355	0.377143	0.329257	27.35484	0	1	1	0	4.125	1
31.08108	0	36.54293	17.60196	51.39016	32.88912	0.007	1.118342	0.085069	0.007213	0.01925	0.85463	98.017	1.810186	0.675188	0.574229	27.56989	1	1	0	0	0	0
22.97297	0	45.54249	17.85504	46.47057	24.51889	0.671007	27.17571	21.54273	15.21912	0.053699	0.655806	38.102	1.238449	0.623769	0.49393	26.06452	0	0	0	0	0	0
56.75676	0	47.52456	21.10689	41.35708	22.3606	0.150002	8.862593	20.49957	4.238065	0.366772	10.62926	57.565	1.022497	0.31871	0.530828	25.97849	1	1	1	0	0	0
22.97297	0	41.25449	16.46805	46.35503	19.84593	96.94097	2.406033	0.444487	0.183413	0.06383	0.049966	0.094	1.312023	0.522425	0.501051	26.15054	1	0	0	0	0	1
50	0	41.19319	20.14967	45.9838	29.01982	0.014	0.254556	2.039536	0.518295	0.098279	5.181908	92.251	1.65423	0.399666	0.488919	27.39785	0	0	0	1	4.125	1
60.81081	0	47.35568	19.83056	39.17301	24.0785	0.001	0.003195	1.922566	0.296757	0.001013	0.116588	97.787	1.690736	0.462103	0.438971	25.67742	0	0	0	0	0	1
9.459459	0	47.3793	20.73255	50.92616	22.18493	0.004	0.006391	0.052105	0.011334	0.005066	0.018737	99.907	1.599416	0.633346	0.617362	24.55914	0	1	0	0	0	0
54.05405	0	52.00798	15.15053	42.89513	28.91285	2.317023	30.22292	9.683011	0.173108	0.161096	1.914329	58.035	1.471165	0.515078	0.369957	24.12903	1	1	0	0	4.125	1
43.24324	0	39.18316	19.26239	50.54192	21.17499	0.001	0.005325	1.388756	0.059764	0.002026	0.078072	98.552	1.678563	0.515897	0.457783	27.44086	1	0	0	0	2.0625	1
50	0	50.69322	1.975353	44.32776	26.04607	16.00916	4.354078	7.253219	0.649157	3.132757	10.09004	59.667	0.743778	0.341664	0.069292	25.03226	0	0	0	0	0	1
54.05405	0	41.95872	19.25663	44.17502	24.98276	9.629096	1.719051	20.61229	9.180929	0.074975	0.280019	60.12	1.843771	0.575736	0.46993	27.91398	0	1	1	0	2.0625	1
66.21622	0	36.5562	20.26693	53.59866	27.57513	0.098001	2.543429	4.485278	0.801657	0.082068	7.83532	84.91	1.545212	0.383708	0.406014	27.31183	0	1	0	0	0	1
35.13514	0	53.62534	17.4105	35.94675	24.51368	10.3271	1.251478	1.249455	1.210729	1.190488	1.22313	83.799	0.697373	0.466168	0.217278	21.46237	0	1	0	0	0	0
39.18919	0	42.80703	22.70055	46.28571	27.20152	72.52473	0.97775	10.13919	9.914579	0.588658	0.361214	6.472	1.426521	0.770953	0.538108	28	0	0	0	0	0	1
62.16216	0	18.10731	9.664392	47.64585	27.08935	3.477035	2.285678	7.508427	1.841338	0.719359	1.68844	83.196	1.014906	0.437477	0.291514	25.41935	0	0	0	0	0	1
35.13514	0	42.73626	21.92785	42.65429	30.36866	72.37472	4.823781	4.737295	13.04599	0.651476	5.042419	0.493	1.310024	0.53486	0.537352	26.36559	1	1	0	0	0	1
37.83784	1	44.65395	23.70304	36.08891	28.36388	0.008	0.008521	0.498719	0.026791	0.008105	0.035393	99.446	1.751595	0.77371	0.62794	23.6129	0	1	0	0	0	0
50	0	40.78078	19.13792	44.39781	34.07829	69.25769	0.54852	11.39397	2.734701	2.295869	0.536095	14.079	1.604041	0.654855	0.450255	27.78495	0	0	0	0	2.0625	1
67.56757	0	27.45111	17.98825	37.26411	34.17961	0.060001	0.038343	1.547197	0.162804	0.030395	0.031229	98.232	1.73134	0.431526	0.394037	22.66667	0	0	1	0	4.125	1

cb_gender	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	is_Success	is_success	is_not_succe
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.845264	0.154736
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.840735	0.159265
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.838532	0.161468
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.831638	0.168362
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.830876	0.169124
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.829063	0.170937
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.828162	0.171838
0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0.82745	0.17255
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.827367	0.172633
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.82697	0.17303
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.8257	0.1743
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.824903	0.175097
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.824801	0.175199
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.823466	0.176534
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.823137	0.176863
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.82271	0.17729
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.821955	0.178045
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.821836	0.178164
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0.821314	0.178686
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.820937	0.179063
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.820089	0.179911
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0.820031	0.179969
0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0.819955	0.180045
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.818978	0.181022

Using Neural Network:

fc_face_a	fc_wear_g	fc_yaw_ar	fc_pitch_a	fc_roll_an	fc_BMI	fc_neutral	fc_sadness	fc_disgust_	fc_anger_	fc_surprise	fc_fear_er	fc_happine	fc_fWHR_	fc_fWHR_	fc_fWHR_	fc_FA_rati	cb_has_Tv	cb_has_ba	cb_has_m	cb_has_ph	cb_total_j	cb_gender
39.18919	0	39.71806	19.56705	46.43559	27.65386	0.006	0.006391	0.051042	0.006182	0.006079	0.056212	99.875	1.685373	0.476333	0.448513	27.65591	0	0	0	0	0	0
41.89189	0	35.44862	19.40948	49.07172	25.30098	0.060001	0.42923	0.433853	0.023699	0.023303	0.154062	98.935	1.537595	0.388079	0.530959	26.27957	0	0	0	0	0	0
36.48649	0	40.82052	19.00706	40.65233	18.58316	0.078001	0.093728	5.284929	1.231337	0.841954	0.989955	91.886	1.407872	0.56756	0.451026	26.27957	1	0	0	0	0	0
39.18919	0	44.46272	20.64197	45.39333	19.50123	0.064001	1.15562	0.835806	1.631135	0.124621	0.293551	96.078	1.644007	0.51234	0.585657	26.62366	1	0	0	0	0	0
39.18919	1	42.37624	21.31807	44.86132	18.06043	0.001	0.00213	1.431291	0.218446	0.002026	0.233175	98.213	1.368126	0.447148	0.425905	27.44086	1	1	0	0	2.0625	0
58.10811	0	44.50632	21.17801	42.76405	23.26344	0.005	0.005325	0.793271	0.041216	0.006079	0.077031	99.123	1.588062	0.407338	0.436461	25.54839	0	0	0	0	0	0
43.24324	1	33.18288	15.09097	54.65628	22.37624	2.465025	1.180117	0.397699	0.38022	0.373864	0.682871	94.66	1.373785	0.363152	0.516503	23.22581	1	1	0	0	0	0
40.54054	0	34.75962	19.62331	47.27371	28.82221	0.235002	0.189586	11.56517	0.159713	0.050659	0.062458	88.444	1.420603	0.484318	0.373254	26.62366	1	1	0	0	0	0
62.16216	0	41.927	21.21954	43.98597	26.75774	0.021	0.05858	2.414904	0.679038	0.052685	0.358091	96.598	1.824949	0.518212	0.374092	24.60215	0	0	0	0	0	0
68.91892	0	41.87319	21.25607	36.52896	18.99667	0.036	0.575147	3.919567	2.044328	0.241137	1.052412	92.506	1.500474	0.438271	0.58658	24.12903	1	0	0	0	0	0
24.32432	0	34.15372	16.94966	49.8175	27.796	0.627006	0.111834	5.362555	0.334882	0.06383	0.065581	93.774	1.337476	0.611929	0.516212	25.63441	1	1	0	0	2.0625	0
27.02703	1	60.69071	8.008187	59.55692	27.32959	84.98685	1.449584	7.059687	4.718235	0.720372	0.740124	1.013	0.412628	0.22479	0.106483	19.13978	1	0	0	0	0	0
31.08108	0	36.54293	17.60196	51.39016	32.88912	0.007	1.118342	0.085069	0.007213	0.01925	0.85463	98.017	1.810186	0.675188	0.574229	27.56989	1	1	0	0	0	0
47.2973	1	42.64409	7.769119	50.49521	32.00941	10.64511	1.76059	3.338969	3.021154	1.496469	1.466715	78.744	1.600563	0.703602	0.294162	25.89247	1	1	0	0	2.0625	0
28.37838	1	41.39187	16.65395	42.88725	25.88201	0.140001	0.149112	0.632703	0.144257	0.141845	0.145735	98.707	1.567886	0.499762	0.363811	27.31183	0	1	0	0	2.0625	0
35.13514	0	39.92195	15.80728	48.40365	18.55731	0.018	0.019172	0.515732	0.054612	0.358666	0.283142	98.799	1.845071	0.559557	0.348527	26.70968	0	1	0	0	2.0625	0
56.75676	0	44.33182	18.3944	44.47155	26.43422	0.002	0.00213	0.029774	0.002061	0.002026	0.005205	99.959	1.407562	0.675519	0.371963	26.10753	1	1	1	0	2.0625	0
47.2973	1	42.1126	20.96732	45.97529	19.18542	0.183002	0.194911	10.18067	15.20778	4.927102	3.25717	67.309	1.591385	0.295835	0.448778	27.39785	1	1	0	0	4.125	0
44.59459	1	40.41543	20.23417	44.95909	26.85012	77.57678	7.208512	3.774949	1.617739	0.328271	1.951803	8.337	1.119761	0.488641	0.502193	27.31183	1	0	0	0	0	0
50	0	52.38238	14.14825	45.90407	29.17414	0.719007	0.080947	8.576047	0.335913	0.647423	0.125956	90.054	1.735178	0.635618	0.311836	25.54839	1	0	0	0	4.125	0
50	1	46.39071	16.73061	41.84745	22.39697	4.409044	4.322125	16.5598	3.787777	1.414401	0.981627	69.945	1.449969	0.391063	0.404936	24.08602	1	1	0	0	2.0625	0
31.08108	0	48.14258	20.72227	35.43295	24.40228	0.065001	0.069231	6.060123	0.5049	0.21986	0.843179	92.655	1.243221	0.566167	0.629112	22.83871	0	0	0	0	0	0
41.89189	0	39.77374	17.86155	39.60341	25.28989	0.002	0.026627	0.10421	0.006182	0.02533	0.220684	99.632	1.565043	0.510756	0.410734	25.41935	0	1	1	0	0	0
44.59459	1	42.519	18.80149	43.03715	19.72753	0.119001	0.126745	23.96189	4.443116	10.04468	2.633633	60.471	1.567986	0.439916	0.428002	27.26882	1	1	0	0	4.125	0

cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	cb_person	is_Success	is_success	is_not_succ
0	0	0	0	0	0	0	1	0	1	0.923704	0.076296
0	0	0	0	0	0	0	1	0	1	0.915674	0.084326
0	0	0	0	0	0	0	1	0	1	0.904951	0.095049
0	0	0	0	0	0	0	1	0	1	0.904437	0.095563
0	1	0	0	1	0	0	0	0	0	0.899604	0.100396
0	0	0	0	0	0	0	1	0	1	0.892557	0.107443
0	1	0	0	1	0	0	0	0	1	0.890546	0.109454
0	0	0	0	0	0	0	1	0	0	0.887895	0.112105
0	0	0	0	0	0	0	1	0	1	0.886569	0.113431
0	0	0	0	0	0	0	1	0	1	0.883806	0.116194
0	0	0	0	0	0	0	1	0	1	0.882774	0.117226
0	1	0	0	0	1	0	0	0	1	0.880905	0.119095
0	0	0	0	0	0	0	1	0	1	0.87642	0.12358
0	1	0	0	1	0	0	0	0	1	0.87303	0.12697
0	1	0	0	1	0	0	0	0	1	0.869399	0.130601
0	0	0	0	0	0	0	1	0	1	0.864277	0.135723
0	0	0	0	0	0	0	1	0	1	0.858497	0.141503
0	1	0	0	0	0	1	0	0	1	0.857251	0.142749
0	1	0	0	0	1	0	0	0	1	0.855417	0.144583
0	0	0	0	0	0	0	1	0	1	0.855131	0.144869
0	1	0	0	0	1	0	0	0	1	0.852604	0.147396
0	0	0	0	0	0	0	1	0	1	0.851828	0.148172
0	0	0	0	0	0	0	1	0	1	0.851118	0.148882
0	1	0	0	0	0	1	0	0	1	0.849852	0.150148