|  |
| --- |
| **Machine learning documentation of startup prediction system** |
|  |
| GitHub repo: https://github.com/AhmadAmr/start-up-prediction-system |
|  |

|  |  |  |
| --- | --- | --- |
| Faculty of computers & information  KFS University  Predicting startup success with machine learning  **By**   * Yussef Raouf Abdelmassih   Advisors/Co Advisors: DR. Amr Abo Hany / ENG. Muatesm Daraz |  |  |

April 2021

Abstract

Start-ups are becoming the motor that moves our economy. Google, Apple, or more recently Airbnb and Uber are companies with tremendous impact in worldwide economy, social interactions and government. Over the past decade, both in the US, Europe, and the world, there has been an exponential growth in start-up formation. Thus, it seems a relevant challenge understanding what makes this type of high-risk ventures successful and as such, attractive to investors and entrepreneurs. Success for a start-up is defined here as the event that gives a large sum of money to the company’s founders, investors and early employees. The ability to predict success is an invaluable competitive advantage for investors to be one step ahead of competition.

We explored the world’s largest structured database for start-ups – provided by the website CrunchBase.com, with the objective of building a predictive model, through supervised learning, to accurately classify which start-ups going to be successful and which are not. Most of the studies regarding the prediction of processes of M&A or an alternative definition of a company’s success tend to focus on traditional management metrics provided by financial reports and thus using a low number of observations compared with the present study. As technologies of information evolve it became possible to achieve highly reliable results in data analysis by manipulating it with complex machine learning algorithms or data mining techniques to define features and characterize robust models.

Further developments on previous studies such as the development of new features and a new definition for the target variable were applied. Using Random Forests on our dataset, a general model (as including all categorical features) achieved a True Positive Rate (TPR) of 96%, which is the highest recorded with this data source, and a False Positive Rate (FPR) of 4%. The author also generated models per each category and country to provide results comparable with previous studies the values achieved ranged between 75% and 96%.

**Keywords**

positive rate (TPR), false positive rate (FPR), Naïve Bayes (NB), ROC (Receiver Operating Characteristic)

INDEX

Table of Contents

**1.Introduction4**

1.1Objectives 5

1.1.1 Technical objectives 5

**2.Data analysis 5**

2.1 Data Mining5

2.2 Machine learning5

**3.Methodology 6**

3.1 Data Preprocessing7

3.2 Data cleaning 7 3.2.1Data selection ……………………………………………………………………………………9 3.2.2 Data transformation …………………………………………………………………………...9

3.3 Experiment setup……………………………………………………………………………………….10 3.3.1Evaluation metrices……………………………………………………………………………10

1. **INTRODUCTION:**

**“A start-up can be defined as a human institution created to develop new products and/or services under extreme uncertainty conditions.”**

Predicting the success of a start-up is commonly defined as two-way strategy that makes a large amount of money to its founders, investors and first employees, as a company can either have an IPO (Initial Public Offering) by going to a public stock market (i.e. Facebook going public, allowing everyone to invest in the company by buying shares being sold by its insiders in the U.S stock market) or, be acquired by or merged (M&A) with another company (i.e. Microsoft acquiring LinkedIn for $26B) where those who have previously invested receive immediate cash in return for their shares. This process is often denominated as an exit strategy. This study will therefore just focusing of the percentage of which category of startup has more chance to be success than other categories in same area or country.

With a focus on how a start-up or an investor could explore all this knowledge for a better decision making in investment strategy and monetary gain, the study intends, by applying data mining and machine learning techniques, to create a predictive model that has as the dependent variable a label to classify whether a new start-up (**can be**) successful or not.

To generate the predictive model, two supervised machine learning algorithms were tested: Naïve Bayes and Random Forests. All these algorithms fit the characteristics of the dataset (875 features and more than 57805 observations),

* 1. **OBJECTIVES:**

The present work has as the main objective, the development of a predictive model to classify a start-up/company can be successful or not (binary classification).

* + 1. **Technical Objectives:**

During a first phase of Data Analysis, a full understanding of the CrunchBase database is expected, followed by the process of Data cleaning (missing values, duplicates, redundant data). Having a full database ready to be filtered it fundamental to define the scope of data to be used in the model and to be able to do an explorative analysis of key features. Transformation of data will be made by defining and creating new features which will generate the final dataset to be used in the learning task.

Followed by a second phase consisting on the Experiment Setup and its Results, where the experiment will be set up by applying different machine learning algorithms to generate the best possible model through supervised learning to try to outperform current state of art. The algorithms tested are Naïve Bayes (NB) and Random Forests (RF).

1. **Data Analysis:**

**2.1 Data Mining:**

Data Mining is a step in the KDD process that consists on applying data analysis and discovery algorithms to produce patterns (or models) over transformed data. Classification (as in the present study), regression or clustering are examples of common data analysis. The data-mining component of the KDD process often involves repeated iterative application of data-mining methods.

**2.2 Machine Learning:**

Machine learning can be divided in four different categories:

**supervised**, **unsupervised**, **semi supervised** and **reinforcement learning**. Being supervised and unsupervised learning the most widely used. **Supervised learning** algorithms make predictions based on a set of examples. A supervised learning algorithm is, having x input variables and an output variable y. The algorithm learns to map the function (y=f(x)) and can (correctly) predict/classify any new output y after getting new input data x. The possible answers from the output are known. All data is labelled, and the algorithms learn to predict the output from the input data. Supervised algorithms can be grouped into regression and classification problems: A regression function is a type of model when the output variable is a real value, i.e., 88, 130, 0%. A classification function generates models where the output is a category, i.e., “red”/ “blue” “not acquired”.

**Unsupervised** learning algorithm is when we only have input variables/features and no output (target variable). It is in the learning process that the algorithm will discover and classify possible outcomes. Here, we don’t know the possible answers. As all data is unlabeled, the algorithm should learn to create patterns from the input data. Typically, unsupervised learning can be grouped into clustering and association analysis. A clustering problem is the discovery of groups with heterogeneous characteristics between them and homogeneous characteristics between the observations of each group.

**3. METHODOLOGY :**

The methodology here applied (Figure 1– Methodology Overview) mirrors a loose interpretation of Knowledge Discovery in Databases (KDD) approach **(1) Selection** of data to be processed by defining relevant tables from the entire structured CrunchBase database; **(2) Preprocessing**, by cleaning, Selecting and Transforming data. At this stage we deal with missing values, outliers, discretization, and other common problems. An explorative analysis is made before further transformations; **(3) Experiment Setup**, where evaluation metrics are defined, and the major problems of the dataset - Imbalanced target classes and sparse data, are dealt with. these problems are only addressed at this stage. Several machine learning algorithms are chosen to test a binary classifier to classify the observations as either “successful” or “not-successful”. **(4) Experiment Results**, where we draw conclusions and interpret results.

Initial crunchbase table

Data Preprocessing

Cleaning

Initial crunchbase table

All features to binary

Problem1: Sparse data

Experiment setup

Transformation

Selection

Figure 3.1(explaining life cycle of dataset model)

Problem2: Imbalanced target class

Final dataset

TPR = 98.8 %

Random forest (RF)

Experiment Result

**3.1 Data Preprocessing:**

The data pre-processing consists in a 3-step process:

* **Data cleaning:** we aim to remove all redundant and irrelevant information from the database as well as duplicates, missing values and outliers. The explanation of this process is divided between specific changes in the ‘Companies’ table.
* **Data selection:** where the context of the study is defined to filter which data will be taken into the final dataset.
* **Data transformation:** consisting on the process of creating new variables.

**3.1.1 Data cleaning:**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Type** |

id Company id Nominal

Name company name Nominal

homepage URL company URL Nominal

permalink company link in CrunchBase Nominal

category\_list category of organization Nominal

funding\_total\_usd total fund Interval

status Operating or close Categorical

country\_code Country (i.e., USA – United States) Categorical

state\_code USA’s states (i.e., CA – California) Categorical

region Country’s region Categorical

city Country’s city Categorical

funding rounds Total funding rounds Ordinal

founded at Foundation date Date Time

first\_funding\_at Date of first fund Date Time

last\_funding\_at Date of last fund Date Time

**The first step** of pre-processing consists on treating all the irrelevant and redundant information present in tables. As a free-to-edit database with multiple purposes, the CrunchBase dataset has several columns (features) and instances (observations) whose context don’t match the objective of predicting a start-up’s success.

**From the ‘companies’ table:**

* Deleted region, city as they provide too much granularity.
* Deleted domain, homepage URL, name as irrelevant features**.**

**General changes:**

- Only a few duplicate instances were found in the database and all were removed.

**The second step** consists on eliminating noisy or unreliable data being the two most common cases of inconsistencies, Missing Values and Outliers. A Missing value (or missing data) is a variable that has no data value stored in an observation. Missing values are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data.

**From ‘companies table:**

* Deleted last\_funding\_at, first\_funding\_at, founded at, funding rounds, state code.

**3.1.2 Data selection:**

Before further advancements in the experiment setup of the dataset it is important to contextualize what will be the subject of study and filter data.

|  |  |  |
| --- | --- | --- |
| Feature | Description | Type |
| Categroy\_list | Company category | Nominal |
| Country\_code | Name of country | Categorical |
| Funding\_total\_usd | Amount of money | Interval |
| Status | Operating or closed | Categorical |

**3.1.3 Data transformation:**

Transforming data can be summarized as “the application of mathematical modification to the value of a variable” to extract more value than in its original state. In the present dissertation, the data transformation process achieved with two steps:

* **Changes in original data:**

Category: All companies were classified into one or more of 857 categories, it varies between “software”, “hardware”, “manufacturing”, “energy”, etc. Categories are merged into a column separated by the symbol “|”and sorted from A-Z.

to solve this, we used **one hot encoder** & **two categorial sklearn**, which give us a result all categories formed as binary data.

We now detail the process used to determine each company’s category. Originally organizations had between 1 up to 14 categories selected from a binary value list.

* **All Features to binary:**

as we see there is a must to our data set to change from actual form to binary form because this allow us to run our methods (Naïve Bayes & Random Forest).

So, with two methods in above section we transformed our **STATUS** and **COUNTRY\_CODE** to numeric data.

**3.2 Experiment Setup:**

**3.2.1 Evaluation Metrics:**

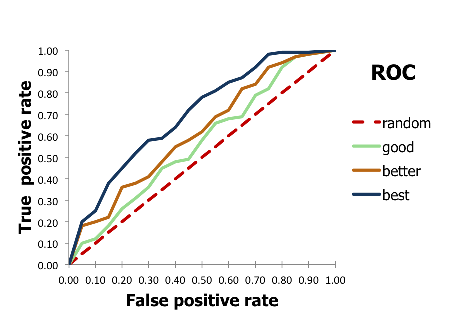
The classifier will have as its main evaluation metrics, True Positive Rate (TPR) and False Positive Rate (FPR). Not only are those standards for most binary classification tasks but they were also used in the work considered as state of art. Also, by using the same metrics we can perform a statistical comparison between the two approaches for the same problem.

**True Positive Rate (TPR = TP / (TP+FN))** or **Recall** can be defined as the percentage of all the successful companies correctly identified as successful. On the other side, **False Positive Rate (FPR = FP / (FP+TN))** can be understood as the percentage of all unsuccessful companies classified as successful. As an easy to understand metric, **TPR** clearly tells the predictive capability of the crucial aspect under study - classifying companies as successful with the features and methodology in-use.

|  |  |  |
| --- | --- | --- |
| Confusion matrix | 0, (predicted negative) | 1, (Predicted positive) |
| 0, (Actual negative) | True Negative (TN), company classified as not successful and it is not successful. | False Positive (FP), company classified as successful and it is not successful |
| 1, (Actual positive) | False Negative (FN), company classified as not successful and it is successful | True Positive (TP), company classified as successful and it is successful |

Precision will be shown as a support metric and can be defined as, “percentage of all successful companies correctly classified”. Although this metric is not the one used to compare results with previous studies it supports how well our instances are classified.

**Precision = (TP+TN) / (TP+FP+TN+FN)**

****As typical measure used in statistics to evaluate binary classifiers, ROC (Receiver Operating Characteristic) curve is a graphic plot that illustrates the predictive capability of a model by plotting both cumulative TPR and FPR at different thresholds. The area under the ROC curve (AUC) is a standard metric taken from the ROC curve as it clearly shows the trade-off between both main evaluation metrics, TPR and FPR.

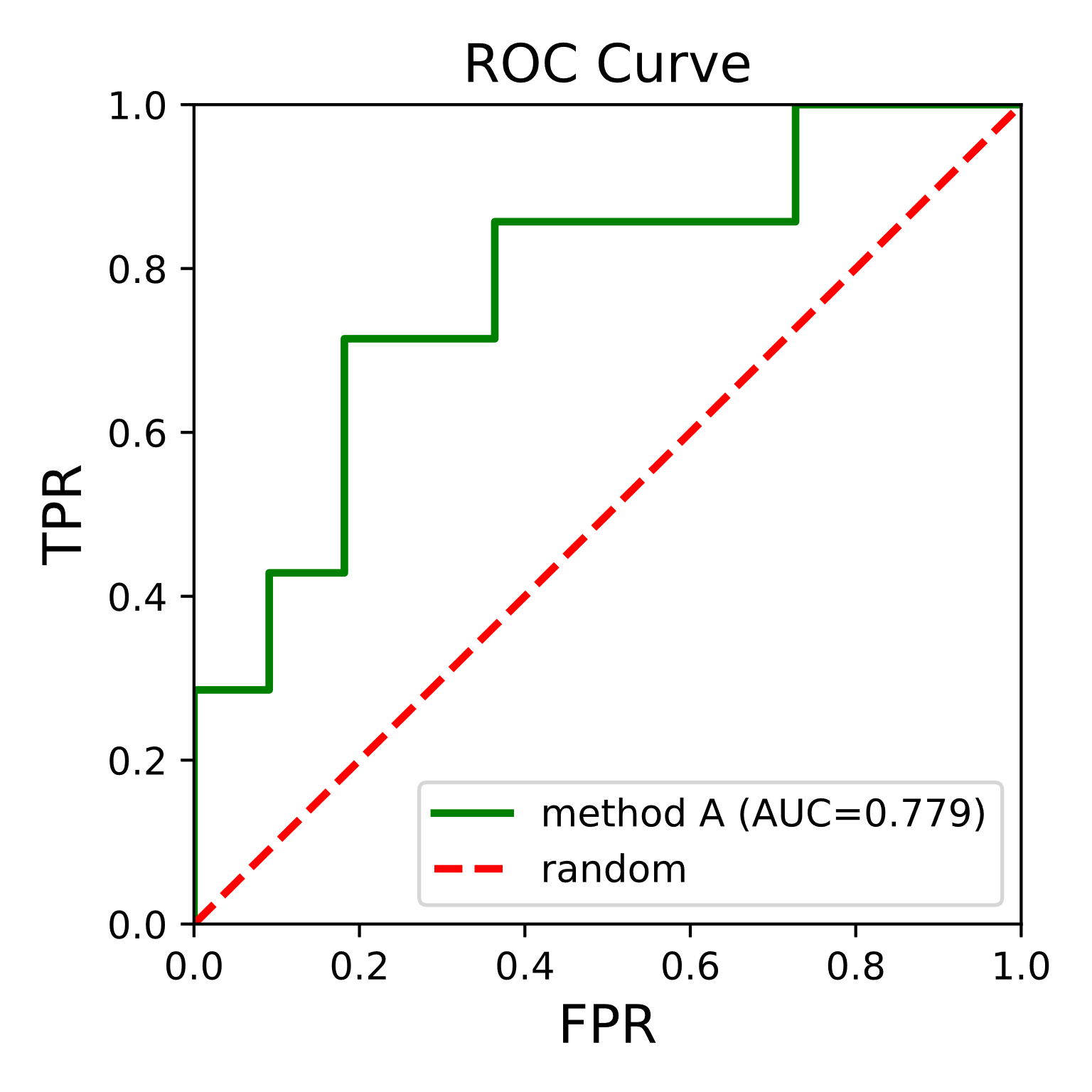


Figure 3.2 AUC (Trade between FPR & TPR)

Figure 3.2 ROC (Predictive capability)

**3.3.2. Problems with the dataset and solutions used:**

**3.3.2.1. Sparsity of the dataset**:

The first problem found in the present analysis was the sparsity of the CrunchBase database. As previously stated by Xiang et al., “despite its huge magnitude, the CrunchBase corpus is sparse with many missing attributes in the profiles” (Xiang et al., 2012). Due to its free-to-edit nature, anyone can create companies and fill its data without much control. This fact allied with its growing popularity creates an exponential growth in sparse data as more incomplete profiles are created than reviews are made.

Despite the ability of the machine learning algorithms used in this work to deal well with sparse data, the ambiguity in what is sparse and what is missing value in the present context motivated us to solve the problem in the following two steps:

* + A **binary feature** was created to support features with categorical data. These binary features meant to signal whether the observation had value in the feature.
  + A **multivalued column** was solved by running one hot encoder algorithm to separate them into standalone field with a True or false value.

**3.3.2.2. Imbalanced Classes:**

**“A dataset is imbalanced if the classes are not approximately equally represented.”**

Another problem faced when trying to create a good predictive model for the task at hands was the large class imbalance between successful and non-successful companies. After pre-processing, only 16,8% of the dataset consisted on unsuccessful companies. Most machine learning algorithms work best when the number of observations of each class is equal because when there is such disparity between classes the algorithms tend to classify the lowest represented class as the opposed. In the present study, if all observations were marked negative (unsuccessful) the model would still achieve around 77% of Accuracy, which wouldn’t be a better score than most models published in predicting success of a company.

Not only is “Accuracy” a dangerous metric to evaluate the quality of a model with a large imbalance of classes (ROC curve is more adequate) but also the problem of class imbalance can be tackled using different strategies such as over sampling the lowest represented class or under sampling the largest.

**SMOTE (Synthetic Minority Over-Sampling Technique)** is a technique that consists in an oversampling of the minority class. Meaning it will create new synthetic instances of the lowest represent class (in this case, unsuccessful companies) rather than by over-sampling with replacement.

After dealing with the sparsity, SMOTE was applied to our dataset before testing the different machine learning algorithms. With an increase of 400% of synthetic instances classified “unsuccessful” with 5-nearest neighbors, the classes become balanced:

|  |  |  |
| --- | --- | --- |
| Before SMOTE | Number of observations | % |
| 0 | **14 190** | **16%** |
| 1 | **72 398** | **84%** |

**86 588 100%**

|  |  |  |
| --- | --- | --- |
| After SMOTE 400% | Number of observations | % |
| 0 | **70 950** | **49%** |
| 1 | **72 398** | **51%** |

**143 348 100%**

**3.3.3. Machine learning algorithms:**

In the present work, we have a binary classification task - the target feature is either classified as “1” (for successful companies) or “0” (for not-successful companies). It is a type of supervised learning, a method of machine learning where the output categories are predefined. It is important to choose not only the algorithm that better fits the problem but also one which adapts well to the characteristics of the dataset:

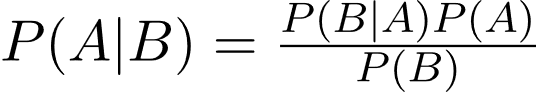
* 856 features
* 86 588observation
* No missing data

Different learning algorithms make different assumptions about the dataset and have different purposes. When testing the following algorithms, we intended to test its data with ML models that not only fit the nature of dataset but are also easy to understand and implement. It is equally important to test algorithms used with this dataset in previous works.

In the following section we will discuss the algorithm we used **Naïve Bayes and Random forest classification.**

**3.3.3.1. Naïve bayes:**

In [statistics](https://en.wikipedia.org/wiki/Statistics), **naive Bayes classifiers** are a family of simple "[probabilistic classifiers](https://en.wikipedia.org/wiki/Probabilistic_classification)" based on applying [Bayes' theorem](https://en.wikipedia.org/wiki/Bayes%27_theorem) with strong (naïve) [independence](https://en.wikipedia.org/wiki/Statistical_independence) assumptions between the features They are among the simplest [Bayesian network](https://en.wikipedia.org/wiki/Bayesian_network) models but coupled with [kernel density estimation](https://en.wikipedia.org/wiki/Kernel_density_estimation), they can achieve higher accuracy levels.

****Bayes’ Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes’ theorem is stated mathematically as the following equation:

**3.3.3.2. Random Forest:**

Random Forest (RF) is a collection of Decision Trees (DT). RF does not expect linear features. In its simplest form, it can be thought of using bagging on multiple tree classifier. However, since it is not possible to build multiple trees on the same data as it will get the same results, randomness of two types is introduced: each tree is built on slightly different rows, sampled with repetitions from the original (bagging), and each tree (or in some cases each branch decision) is built using a randomly selected subset of columns. The point of RF is to prevent overfitting which it does this by creating the random subsets of features and building smaller (shallow) trees using the subsets.

The main disadvantage of Random Forests compared with a simple Decision Tree is its interpretability as it is hard to see the relation between a dependent variable and the rule set created. A Random Forest must be a predictive tool and a descriptive one. It is easy to see its features importance but that might not be enough when the objective of the study is to understand the relationship between dependent and independent variables.

**3.4. EXPERIMENT RESULTS:**

**3.4.1. Evaluating Learning Algorithms:**

During a first stage of evaluation to see which ML algorithm better fits our problem, the accuracy of two algorithms was calculated.

The result:

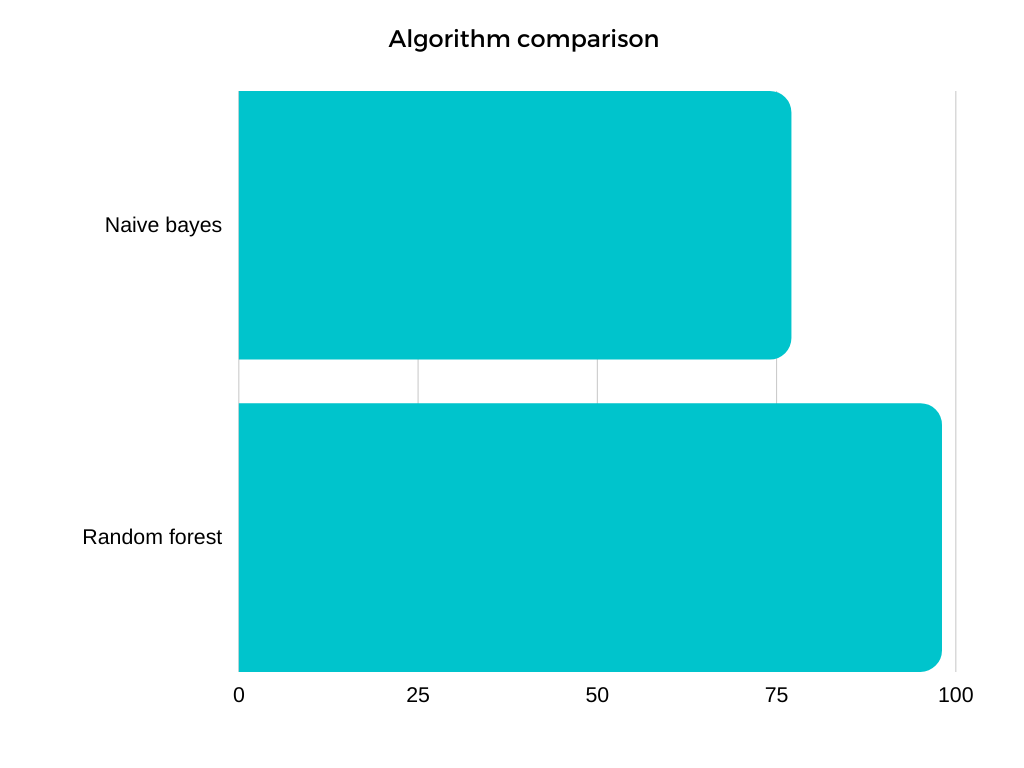
* **Naïve bayes: 0.77**
* **Random forest: 0.988**

Figure 3.3 Accuracy comparison

**4.Motivation and related work:**

**4.1.Motivation:**

All related work is goal was to predict the recent company is successful or not by categorize some features like funding rounds, salary per employee, number of employees, and the state or country the company located.

So, there was no related work until this work which its goal is to predict an unfinished company will success or not.

That was our goal here not to determine that n company is already success or not but to determine that n company which not in a market until now will success or not.

**4.2. Related work**:

**4.2.1. Web-based Startup Success Prediction (CIKM’18, October 22-26, 2018, Torino, Italy):**

They consider the problem of predicting the success of startup companies at their early development stages. They formulate the task as predicting whether a company that has already secured initial (seed or angel) funding will attract a further round of investment in a given period of time. They investigate the potential of using web-based open sources for the startup success prediction task and model the task using a very rich set of signals from such sources like **Crunchbase dataset.**

**4.2.2 Predicting Start-up Success with Machine Learning (NOVA Information Management School Universiade Nova de Lisbon 2017):**

They define Success for a start-up as the event that gives a large sum of money to the company’s founders, investors and early employees, specifically through a process of M&A (Merger and Acquisition) or an IPO (Initial Public Offering). The ability to predict success is an invaluable competitive advantage for venture capitals on the hunt for investments since first-rate targets are those who have the potential for growing rapidly soon, which ultimately, allows investors to be one step ahead of competition. they explored the world’s largest structured database for start-ups – provided by the website **CrunchBase.com.**

They use the same algorithm we used **Random forest classification** with following results:

* a True Positive Rate (TPR) of 94%
* False Positive Rate (FPR) of 8%
* Accuracy 94 %

**Note:** Their algorithm tested only in USA states not all the world.

**5.Conclusion:**

The main objective of the present study was to generate a model to classify successful companies or start-ups. By building a binary classifier to classify a company as successful or not-successful with **a True Positive Rate (TPR) of 96%** and **a False Positive Rate of 4%**. It is the highest reported using data from CrunchBase. The model can classify with high efficiency not only the total of successful companies in the dataset (TPR, recall) but also, from all the successfully-classified which are successful (Precision). The machine learning algorithm used is Random Forests which provides a fast and easy to interpret and implement model with positive results. It provided better results than Support Vector Machines and Logistic Regression. Both the alternative models were chosen due to their potential to fit in the size and nature of our dataset as a linear relation was expected.

To provide comparable results with previous studies, using CrunchBase data to predict company future behavior, the present study is comprised of a general model which contemplates both the category and Country of a company and a model per category. This approach is new and provides a new geographic baseline over the differences in company success based on category and country.

A Francisco Ramadas approached the problem by publishing performances of predictions per category and us states, achieved TPR 94 with Random forest. It should be noted that their best performances were achieved in companies located in USA of observations while the ones in the present study didn’t always follow that behavior. The model achieved TPRs ranging between 76% and 96%. Area under ROC is also higher than theirs – 93.2% vs 88%. Ultimately, the present study benefited from a larger dataset in some categories which proved essential to achieve higher results and from applied in all countries not just USA states.

**6.Refrences:**

* Predicting Start-up Success with Machine Learning (Francisco Ramadas da Silva Ribeiro Bento (M2013022) 2017 from https://run.unl.pt/bitstream/10362/33785/1/TGI0132.pdf
* Web-based Startup Success Prediction (CIKM’18, October 22-26, 2018, Torino, Italy) from https://sci-hub.se/10.1145/3269206.3272011
* Customer Stories | Crunchbase Data Solutions. (2017). Retrieved October 17, 2017.
* Gislason, P., Benediktsson, J., & Sveinsson, J. (2006). Random forests for land cover classification. Pattern Recognition Letters. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0167865505002242>
* C. E. Halabí and R. N. Lussier. A model for predicting small firm performance.