Machine Learning Prediction of Companies’ Business Success or failure

|  |  |  |
| --- | --- | --- |
|  | Holiness Supremacy Mhlanga  [holisupremacy@gmail.com](mailto:holisupremacy@gmail.com)  Project writeup  Abstract |  |

There are thousands of companies coming out worldwide each year. Over the past decades, there has been a rapid growth in the formation of new companies both in the US and China. Thus, it is an important and challenging task to understand what makes companies successful and to predict the success of a company. In this project, we used Crunchbase data to build a predictive model through supervised learning to classify which start-ups are successful and which aren’t. We explored K-Nearest Neighbors (KNN) model on this task, and compared it with Logistic Regression (LR) and Random Forests (RF) model in previous work. We used F1 score as the metric and found that KNN model has a better performance on this task, which achieves 44.45% of F1 score and 73.70% of accuracy.

# Introduction

Thousands of companies are emerging around the world each year. Among them, some are merged and acquired (M&A), or go to public (IPO), while others may vanish and disappear. What makes this difference and leads to the different endings for each company? How to predict the success of companies? If the investors can know how likely the company will achieve success given their current information, they can make a better decision on the investments. Therefore, in this project, given some key features of a company, we

want to predict the probability of its success. More specifically, the input features are of two types: text features (such as industry category list and location) and numerical features (such as the amount of money a company already raised). We then use Logistic Regression, Decision Trees, and K-Nearest Neighbors to output a predicted probability of success. Here we define the company success 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) [1]. Finally, we use F1 score as the metric to compare the performance of these three models.

# Related work

As Machine Learning becomes a more and more popular tool for researchers to utilize in the field of finance and investment, the researcher has found some related work to predict companies’ business success with Machine Learning and Crunchbase.

Indeed, these works propose a variety of efficient methods that we can use to predict the success of company. However, we notice that none of them implement K-nearest neighbor’s model. In this project, we aim to apply KNN model to solving this problem.

# Dataset and Features

The dataset we used was extracted from Crunchbase Data Export containing 60K+ companies’ information updated to December 2015. There were four data files, named “company”, “investments”, “rounds” and “acquisition”. The “company” file contains most comprehensive information of the companies, while other files contain more detailed information regarding the investment operations. Thus, we chose the file “company” as the base and extracted meaningful features from

3.1 Dataset Overview

The "company" dataset consists the following:

Number of Instances: 6819

Number of Attributes: 96

* Name: company’s name
* Homepage URL: the website of the company
* Category list: the industry category the company belongs to, including up to four subcategory divisions
* Funding\_total\_usd: the total amount of funding in all rounds of investments
* Status: the operation status of the company (0 = closed or operating, 1 = ipo or acquired)
* Country\_code: the country of company’s headquarters
* State code: the state of company’s headquarters
* Region: the region of company’s headquarters
* City: the city of company’s headquarters
* Funding rounds: total number of funding rounds
* Founded at: the date company founded (in string format ‘2007-01-01’)
* Founded at: the first time the company raised money (in string format ‘2008-03-19’)
* Founded at: the last time the company raised money (in string format ‘2008-03-19’)

Figure 1 displays some examples for each selected feature.

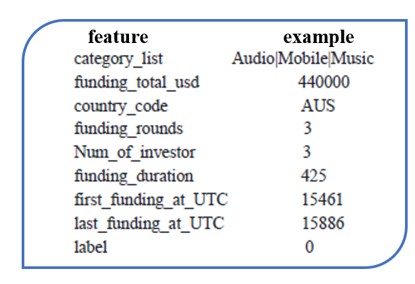


Figure 1: Selected Features and Corresponding Examples

* 1. Cleaning and Labeling

We labeled the company that has M&A with 1, otherwise 0. We plotted the amount of the 0 or 1 labeled data as Figure 1. As seen from Figure 2, the number of data labeled 0 to labeled 1 is over 8 to 1, which is quite imbalanced.

We noticed some skewness regarding the distribution of date of funding events in this dataset as shown in Figure 3. To reduce the bias in the old, invest events, we filtered data before 1990. We also

* 1. Feature Selection

We selected the most essential features to companies’ business success and end up with input features as: category, country, funding rounds, funding rounds, and the difference between when funding rounds and funding rounds.

The training set is composed of two parts. The first part of data is the numerical data: number of funding rounds and total funding. The second part of data is the date in string format, such as ‘first funding at’, ‘final funding at’ and ‘funded at’ columns. As there are too many missing data for ‘funded at’, we finally chose ‘first funding at’ and “final funding at’ columns, converted them from timestamp to numerical UTC format and calculated a ‘duration’ column with the subtracted data.

# Methods

The goal of this project is to make a binary prediction on the status of start-ups, whether they have gone through M&A or IPO. In this project, we explored Logistic Regression, Random Forests, and K Nearest Neighbors.

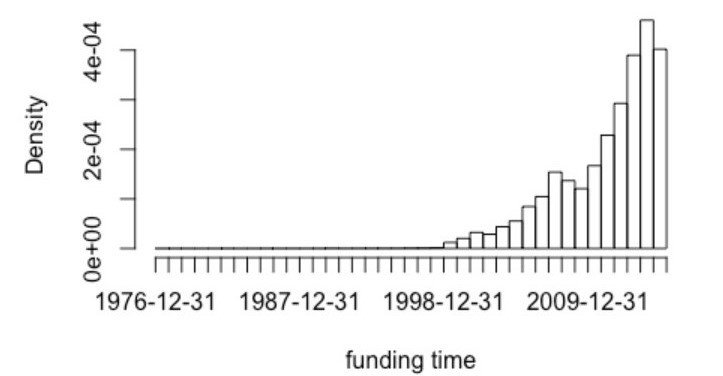
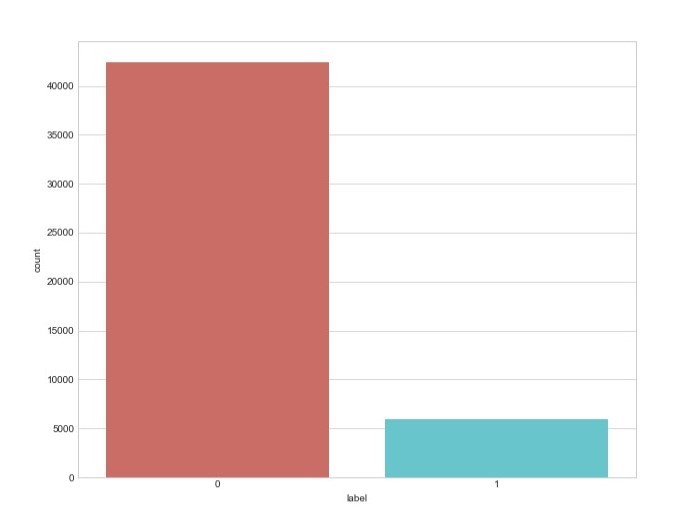
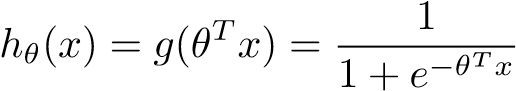


Figure 2: Imbalanced dataset: 1 = IPO or acquired, 0 = closed or Figure 3: Distribution of funding dates.

operating

4.1 Logistic Regression

Logistic regression is a simple algorithm that is commonly used in binary classification. Due to its efficiency, it is the first model we selected to do the classification. The hypothesis of Logistic Regression algorithm is as follows [4]:

 (1)

|  |  |
| --- | --- |
| The algorithm optimizes *θ* by maximizing the following log likelihood function: |  |
| *m*  *`*(*θ*) = X*y*(*I*) log*h*(*x*(*i*)) + (1 − *y*(*i*))log(*h*(1 − *x*(*i*))) | (2) |

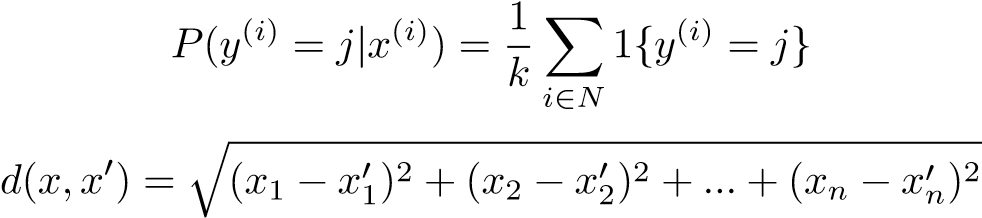
*i*=1

4.2 Random Forests

Random Forests construct a multitude of decision trees at training time and outputting the mode of the classification result of individual trees. At each split point in the decision tree, only a subset of features is selected to take into consideration by the algorithm. The candidate features are generated using bootstrap. Compared to an individual tree, bootstrapping mitigates the variance by averaging the results of a large number of decision trees.

4.3 K Nearest Neighbors

An instance is classified by a majority vote of its K nearest neighbors. The algorithm assigns class *j* to *x*(*I*) that maximizes:

(3)

(4)

4.4 Selected Metrics

In a confusion matrix, we describe the performance of a classification model. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (vice versa). There are four basic terms in a confusion matrix:

TP (true positive): an outcome where the model correctly predicts the positive class.

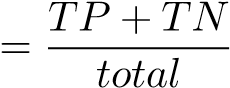
TN (true negative): an outcome where the model correctly predicts the negative class.

FP (false positive): an outcome where the model incorrectly predicts the positive class.

FN (false negative): an outcome where the model incorrectly predicts the negative class.

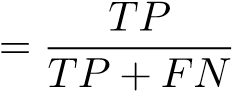
Here we select three metrics: accuracy, F1 score and AUC score.

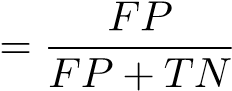
Accuracy: The proportion we have predicted right.

Accuracy  (5)

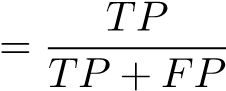
|  |  |  |  |
| --- | --- | --- | --- |
| Training Set (90%) | | Validation Set (5%) | Test Set (5%) |
| Original | 29428 | 1635 | 1635 |
| Up-sample | 50040 |

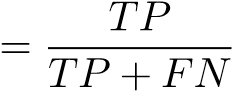
Table 1: Dataset split and up-sample

True Positive Rate (TPR)  (6)

False Positive Rate (FPR)  (7)

F1 Score:

Precision  (8)

Recall  (9)

|  |  |  |
| --- | --- | --- |
| AUC Score: Area under the ROC Curve. | Precision ∗ Recall  F1 = 2 ∗  Precision + Recall | (10) |
|  | Area under ROC Curve  AUC Score =  Total Area | (11) |

# Experiments and Results

5.1 Data Processing

To utilize more data in the training, we split the dataset into three parts: 95% data as training set, 5% as cross validation set and 5% as test set. Since the dataset is quite imbalanced, we up-sample the minority class (label = 1) in the training set to balance the data, but keep the cross validation set and test set untouched (see Table 1).

We also normalize all the numerical features, such as ‘funding rounds’ and ‘funding duration’, and use bag-of-words to encode the text features, such as ’category list’ and ’country code’.

5.2 Hyperparameter Tuning

After preprocessing the data, we concatenate the two types of features, and feed them to logistic regression model, random forest model and K-nearest neighbor’s model. For random forest and K-nearest neighbors’ model, we used random search to tune the hyperparameters. A list of hyperparameters and their associated range is summarized in the table below (see Table 2).

|  |  |
| --- | --- |
| Hyperparameters | Range |
| Number of trees (in RF) | 5-50 |
| K (number of neighbors) | 10-100 |

Table 2: Setting of hyperparameters tuning

5.3 Results

Table 3 shows the result of hyperparameter tuning.

|  |  |
| --- | --- |
| Hyperparameters | Value |
| Number of trees (in RF) | 25 |
| K (number of neighbors) | 92 |

Table 3: Summary of hyperparameters

We use accuracy, F1 score and AUC score to compare the performance of different models, but the F1 score is our primary metric. The figure below summarizes the results of each model on the validation set (see Table 4). We also plot the ROC curve to compare the three models with different thresholds (see Figure 4). We can know the K-Nearest Neighbors (KNN) model has better performance on this task. So, we use KNN model on the test, and achieve the results with 44.45% of F1 score and 73.70% of accuracy.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Classification Models |  | Evaluation Metrics | | |  |
| F1 Score | Accuracy | AUC Score | TPR | FPR |
| Logistic Regression | 44.22% | 72.54% | 79.00% | 69.80% | 26.96% |
| Support Vector Machine | 39.16% | 84.03% | 79.31% | 32.94% | 6.52% |
| Decision Tree | 46.44% | 73.33% | 79.89% | 74.12% | 26.81% |

Table 4: Metrics Results

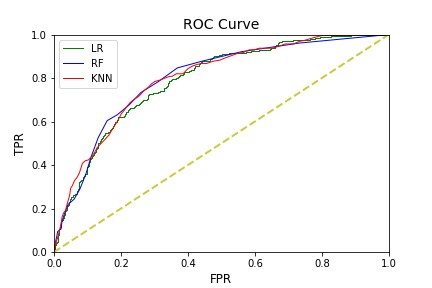
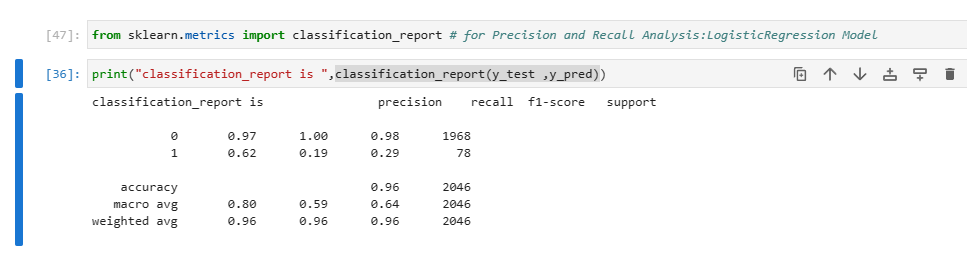
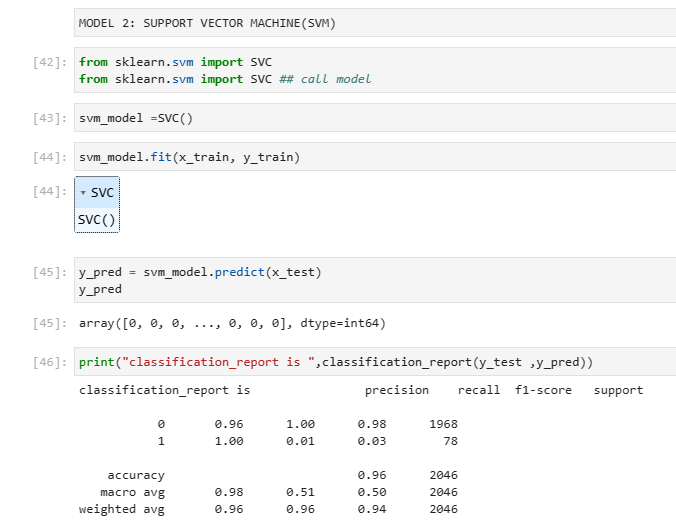


Figure 4: ROC Curve





# Conclusion and Discussion

From results above, we know in general, KNN model performs better. However, why Random forests has a higher accuracy compared with KNN? And how to choose the model based on the different investor’s preference (such as risk tolerance and investment budget)? We compare these two-model using confusion matrix (see Figure 5 and Figure 6). We can see that Random Forests model tries to predict more negative examples but achieve a higher true positive rate. In practice, if the investor has limit investment budget and wants to maximize the proportion of success among its portfolio, it would be better to choose Random Forests model instead of KNN model. However, if the investor has much investment money and want to maximize the number of successful companies it could invest, it would be better to choose KNN model, since KNN model has a higher recall.

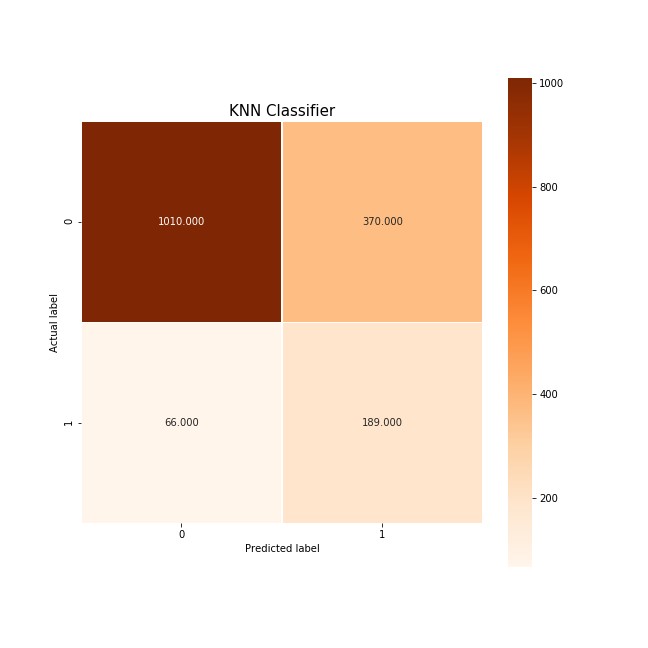
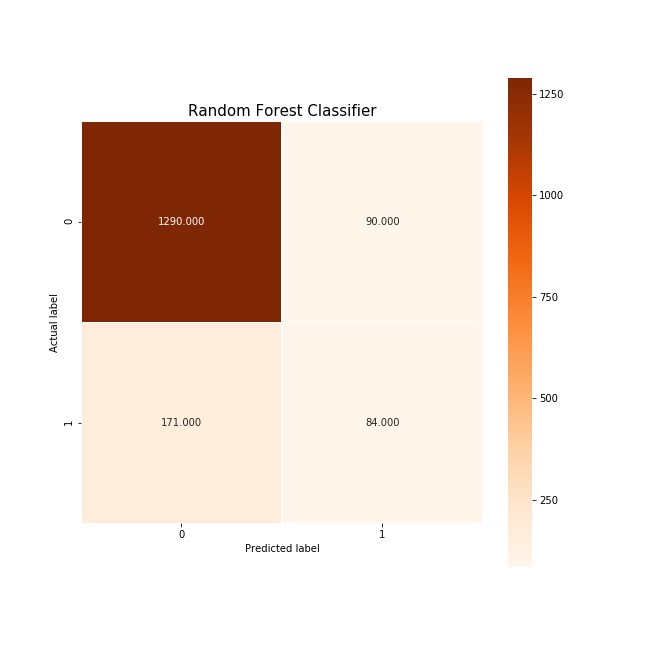


Figure 5: Confusion Matrix of Decision Tree Figure 6: Confusion Matrix ofSVM

# Future Work

In the future, we should include more features of the companies and examine which features are more significant than others. Also, we will try more complex models, such as Neural Network and pre-trained word embedding. Using kernel method to move the data to higher dimensional space is also a good direction. In addition, more new questions are to explore, such as predicting the total funding size for a company (regression problem).

1. Acknowledgments

Thank you to the INT354 staff for the assistance.

# References

1. <https://www.researchgate.net/publication/338289416_Review_of_bankruptcy_prediction_using_machine_learning_and_deep_learning_techniques>
2. <https://www.mdpi.com/1911-8074/15/1/35>

.

1. <https://core.ac.uk/download/pdf/130027032.pdf>
2. <https://www.aimspress.com/article/doi/10.3934/DSFE.2021010?viewType=HTML>.
3. <https://core.ac.uk/download/pdf/130027032.pdf>
4. <https://eecs.ku.edu/applying-ml-models-analysis-bankruptcy-prediction>