# Crunchbase Rank Predictions using Linear and Ensemble Regression Methods

Group 8

Abdullah BILDIR

Aybike CANBEK

Buse Ceren GÖZTEPE

Kerem Ali KAYNAK

EC48W

27.05.2019

## ABSTRACT

In this paper, a predicted ranking system for each start-up company is presented using several machine learning and imputation methods. Essential and useful features such as Minimum Estimated Revenue (MER), Total Funding Amount (TFA), Number of Investors were selected to design a proper model which resembles the ranking system used by Crunchbase. The total sample size is trivial and redundant, although, the train test split is 80 to 20. The results indicate that out data set doesn't allow for predictive capabilities, yet provides insightful information about importance of future.

## 1. INTRODUCTION

Over the past few years, computational systems and technology have transferred models rapidly from being established by rules of static by means of creating substitutes to traditional decision support systems in favor of data analysts and investigators so that we could solve the problem of stable model depending on market structure. Especially machine-learning algorithms allow us to predict models more consistently since the previous ones were resistless to human-kind feelings causing inefficient outcomes. New models like genetic-algorithm-based systems, neuro-fuzzy models bring better and sufficient achievements about performance. Nowadays people are trying to develop new applications and algorithms to ease human living and new start-ups emerged. Some of these start-ups are successful but some are not. At that point people need a kind of program to forecast future of their projects whether it will worth it or not.

## 2. MODEL FEATURES

The indicators that is used by CrunchBase (CB) consists of several different measures such as equity and funding amounts, IPO information, investor types and numbers and other scores for identifying an appropriate CB rank for start-up companies. Even though there are tens of features to define a unique rank for each company, most of them might be ineffective and useless for prospective shareholders who are determined to invest according to their beliefs. In order to solve this problem, most of the features are eliminated and the other features which are supposed to be effective are selected. These features are minimum estimated revenue (in USD), last funding amount (in USD), total funding amount (in USD), number of investors, money raised at IPO, valuation at IPO and trend score (last 90 days). These are the features that we selected to use in our rank predicting model because they have richer and more accurate data compared to other indicators and have wider acceptance. Below the explanations of these indicators can be find.

## 3. METHODS

### 3.1. Libraries

3.1.1. **Numpy:** Used for data manipulation purposes.

3.1.2. **Pandas:** Used for data manipulation purposes in addition to statistical calculations, reading and formatting of data as data frames.

3.1.3. **MathPlotLib:** Used to plot accuracy curves, feature importances and linear regression.

3.1.4. **Scikit-Learn:** Used to train and test Machine Learning models, specifically Random Forest Regressor and Linear Regression Model. In addition, sklearn.metrics is utilized to measure accuracy metrics such as explained variance score, mean squared error.

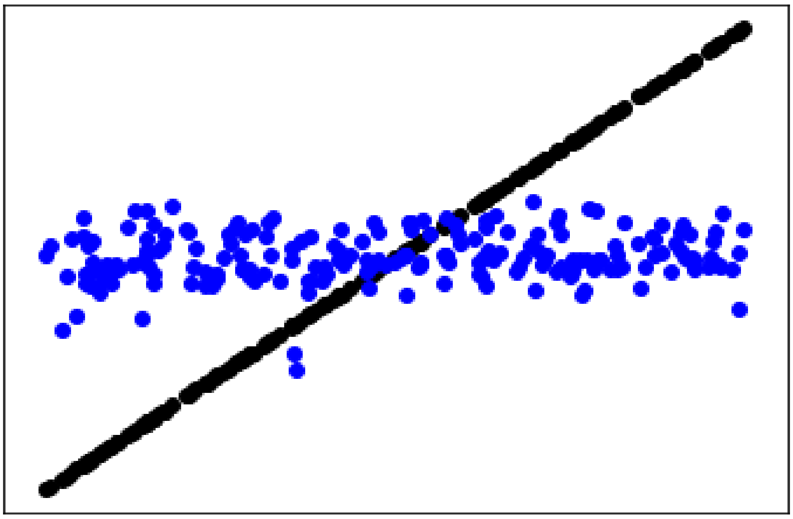
3.1.5. **Collections:** Used in K-Nearest Neighbor Imputation to handle missing values.

3.1.6. **SciPy:** Used in K-Nearest Neighbor Imputation to handle missing values.

## 3.2. ML Methods

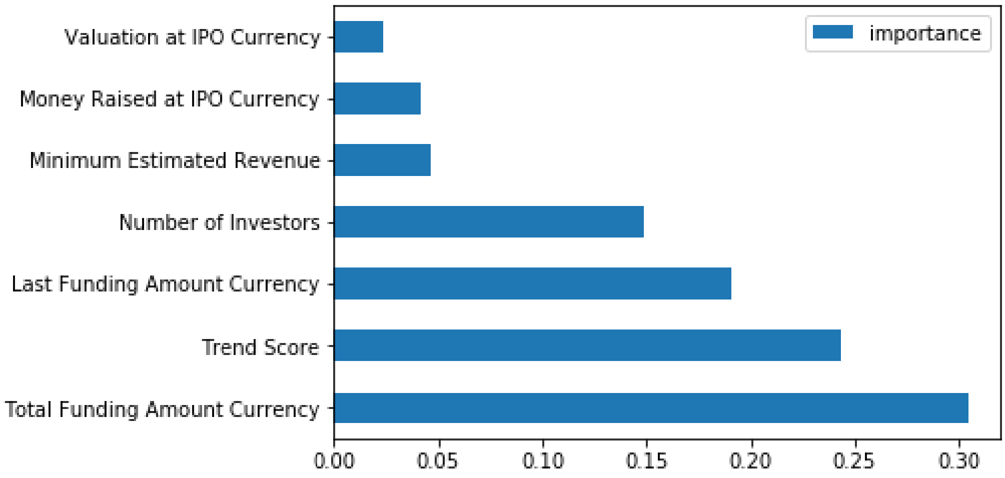
### 3.2.1. Linear Regression:

The dataset we have in hand contains a vast amount of missing values, therefore before fitting the training data into a linear regression model, we need to impute or discard missing data points. Our process of training the model consisted of first structuring the linear regression model, then imputing the dataset and measuring accuracy to decide on the imputation method. We tried a couple of different imputation methods, listed in section 4.3 and saw that for the Linear Regression model, imputing missing data points using Global Median Imputation yields the highest $R^2$ score and the lowest Mean Square Error. Therefore, we proceeded with this method. Linear Regression is an algorithm that fits a line through the data and tries to minimize the sum of residual errors. The line fit is the line that has the lowest cumulative difference compared to the actual data. The cost function is minimized with the utilization of the Gradient Descent Algorithm. The plot of our Linear Regression predictions and the line fit can be seen below.



### 3.2.2. Random Forest Regressor:

After Linear Regression, we decided to compare the results with an ensemble method. Random Forest Regressor is a powerful tool that utilizes a “forest” of decision trees created by a predetermined random subset of features. Then, taking into account the importance of these features, the model aggregates the decision trees to make predictions. Our only hyperparameter in this model is the n_estimators. By tuning this specific hyperparameter, we tried to maximize our accuracy measures. The take of the Random Forest Regressor is that it not only makes powerful predictions but it also provides feature importances, which allows for dimensionality reduction. Using iterations, we found that n=5500 maximizes our accuracy measures. Our feature importance table and bar graph can be seen below.



### 3.3. Imputation Methods

**3.3.1. Global Mean Imputation:** The global mean of each feature is substituted for the missing data points.

**3.3.2. Global Mode Imputation:** The global mode of each feature is substituted for the missing data points. For our case, some features' modes are "NaN", therefore we later eliminated this method. Normally in some cases, this method yields higher accuracy compared to global mean imputation.

**3.3.3. Global Median Imputation:** The global median of each feature is substituted for the missing data points.

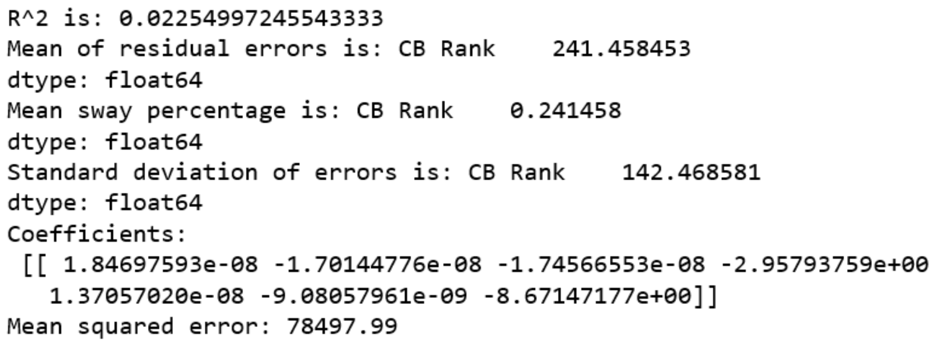
**3.3.4. K-Nearest Neighbour Imputation:** The missing data points are imputed through the usage of a KNN algorithm that takes into account all the features except the one being imputed.

**3.3.5. Row Deleting Using Threshold of Non-NA Features:** This method removes all rows that are under a certain threshold of existing features.

# 4. EVALUATION

### 4.1. Linear Regression Model:

First, the dataset is imported. Since the dataset contains missing values, we have to impute or delete missing data points. To decide on which method to use to handle missing data, we utilized two metrics: The mean squared error and more importantly the explained variance score ($R^2$ score). According to these metrics, we saw that utilizing solely the global median imputation provides the best results. After applying global median imputation, we fit our data to the model for training purposes. Our test-train split is 20/80 and the split is shuffled, therefore unbiased. Our error outputs and coefficients can be seen below.

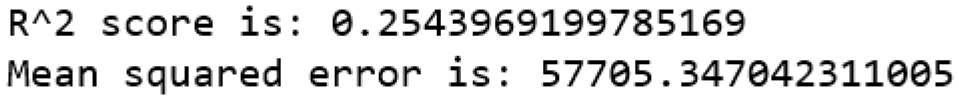


```
R^2 is: 0.022549972455433333
Mean of residual errors is: CB Rank    241.458453
dtype: float64
Mean sway percentage is: CB Rank    0.241458
dtype: float64
Standard deviation of errors is: CB Rank    142.468581
dtype: float64
Coefficients:
 [[ 1.84697593e-08 -1.70144776e-08 -1.74566553e-08 -2.95793759e+00
   1.37057020e-08 -9.08057961e-09 -8.67147177e+00]]
Mean squared error: 78497.99
```

Since no imputation method provides a comforting explained variance percentage, linear regression does not seem like a good fit for our dataset. Our model is hardly a better estimator than the global mean, therefore we will try an ensemble method next.

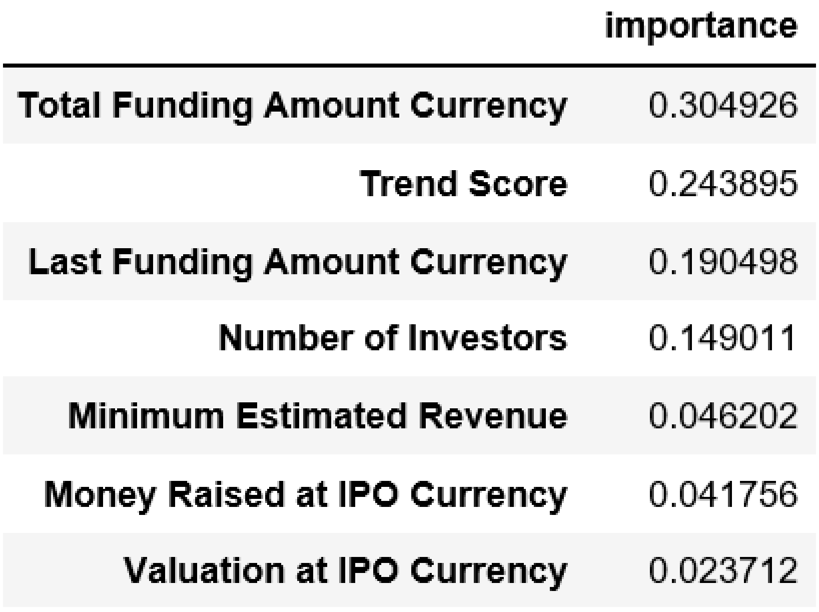
### 4.2. Random Forest Regressor Model:

First, we imported the clean dataset again to re-test the imputation methods. Since we are testing ranks, percentage sway and percentage error measures do not provide a concrete basis for accuracy metrics. Considering these factors, our baseline performance metric for this model is the explained variance score. Iterating through imputation methods, we discovered that firstly imputing the missing data points using KNN imputation, then deleting rows under a certain threshold of existing data and finally imputing the global mean for still existing "NaN" values yields the highest R^2 score. This method and the random forest regressor also yield the highest performance in terms of error metrics and explained variance throughout all of our research, although still not very high. Our error outputs can be seen below.



```
R^2 score is: 0.2543969199785169
Mean squared error is: 57705.347042311005
```

The biggest take of this project for us is the feature importance order provided by the highest performing model throughout our trials. Even though our model does not seem to be performing well in terms of making pinpoint predictions, the feature importances hardly fluctuate throughout our iterations yielding a useful guideline on our business problem: Which aspect of the start-up should the investor focus on? The feature importance table and the bar graph are presented below.



| | importance |
|---|---|
| **Total Funding Amount Currency** | 0.304926 |
| **Trend Score** | 0.243895 |
| **Last Funding Amount Currency** | 0.190498 |
| **Number of Investors** | 0.149011 |
| **Minimum Estimated Revenue** | 0.046202 |
| **Money Raised at IPO Currency** | 0.041756 |
| **Valuation at IPO Currency** | 0.023712 |

## 5. CONCLUSION

In this paper, we tried to establish a rank predicting model by using linear and ensemble regression. Our dataset contains one thousand companies and our test-train split is 20/80. Because there are lots of missing values, we trained our data with two regression models as the linear regression model and the random forest regressor model to find more definite ranks. In linear regression model, applying global median imputation provides more accurate results. Also, in random forest regressor model, first using KNN imputation and then applying the global mean imputation just after deleting the rows provides the highest $R^2$ value. However, in both methods, the accuracy measures are not sufficient enough and the label-feature correlation cannot provide a reliable result due to the undesirable number of missing values.

## 6. REFERENCES

API Reference: Metrics. (n.d.). Retrieved May 22, 2019, from https://scikit-learn.org/stable/modules/classes.html#sklearn-metrics-metrics

API Reference: Model Selection. (n.d.). Retrieved May 22, 2019, from https://scikit-learn.org/stable/modules/classes.html#module-sklearn.model_selection

Ensemble methods. (n.d.). Retrieved May 22, 2019, from https://scikit-learn.org/stable/modules/ensemble.html