Final Report - ORIE 4740 - Statistical Data Mining

Crowd Funded Startup Risk Prediction

Ali, Ibrahim (IAA37); Singh, Bakulesh (BS774); Tiwari, Ishaan (IT236)

Abstract:

This project was aimed at allowing people planning on a crowdfunded startup to be able to get an idea of the amount of funding they could expect based on features such as country of origin, name of the startup, and a broad classification of the domain (such as Art, Technology and Music) and also to be able to get a prediction on whether they would be successful based on data from past startups. We used various approaches for the regression and the classification problem starting with simple linear models and moving on to more complex models using feature transformation, splines and non-parametric models such as Random Forests and KNN. We have used approaches like regularization, subset selection and cross-validation to ensure that the chosen models generalize well to new data. The implementation was done using the R language on the RStudio IDE. For both the regression and classification problems we found that the 5 nearest neighbors model performed the best and had the lowest estimated test errors and thus the highest chances of generalizing to new data, the MAE for regression was ~$11,000 and the classification error rate for classification was ~18%.

1. Introduction:

The startup environment is one that is filled with extreme uncertainty. Starting a business can be a very anxious and nerve-wracking experience for startup owners as there is never a comfortable level of certitude with respect to a project's success, or even if it will secure enough funding to lift off. We aimed to tackle this problem to some extent with this project. Using a vast data set of crowd-funded startups from Kickstarter, we have attempted to build a model that classifies, with a certain level of confidence, whether a startup will be successful or not. For the ones that are successful, our model tries to give information about approximately how much funding they will receive based on predictors such as domain, campaign length and the number of backers.

1.1. Goals & Accomplishment

We had set out on this project with two very specific goals i.e. to predict the amount that a startup could expect in funding and the likelihood of the startup being successful along with the most important factors that determined success or failure. We were able to meet these goals only partially due to two reasons. Firstly, after trying out various methods we found that the predictive power of the features was very low for our goals. We inferred this from the high errors in prediction despite trying out very flexible models. Second, the dataset was very large and had ~400,000 observations and this made the task computationally intensive. We could only tune the model parameters to a very limited extent but did our best by trying out packages that supported parallel implementation of some models and also bootstrapping smaller datasets from the larger one.

The K-Nearest-Neighbors method was deemed to be the best for both regression and classification. The error in regression can seem high but, it provides a reasonable estimate considering the high variance in the pledged amount for different projects and the limited predictive power of the predictors available to us. Predicting the pledged amount within $11,000 of the actual amount would be better than having no idea about it.

We are satisfied by the prediction accuracy of our classification model and also that our chosen model can provide a probabilistic estimate for the success or failure. However, it does not allow us to determine the features that most affect this probability and thus would not allow startup owners to identify and work on their weak points.

1.2. Results and Conclusion

The K-Nearest Neighbor model was best suited for regression as well as classification. A prediction error of approximately $11,000 was achieved in the regression problem. In the classification problem, a misclassification error rate of approximately 18% was achieved. We felt that better results could have been achieved with more relevant predictors.

4. Cleaning/Preprocessing

There were a total of 378,661 observations and 185 variables in the dataset after data preprocessing while the raw data comprised. The data was first preprocessed in order to have clean data to conduct the analyses. Rows with missing data were removed.

The predictor ‘ID’ was likely to have no bearing on the response variable (pledged amount) as it was a random number assigned to each observation and hence that column was removed. The

columns ‘launched’ and ‘deadline’ were transformed into one column called the ‘Campaign Length’ because we felt that the campaign length was something that was more likely to impact the pledged amount and the success as opposed to the start and end dates. The column ‘Name’ could have been used to improve the predictions using NLP methods, however due to our lack of experience with these methods we reasoned that rather than removing it altogether we could transform it to a feature containing the length of the name in terms of characters. We removed the column containing the ‘currency’ because we had a feature containing the country, which would be highly correlated with it, and the amount in USD which would ensure that we would not be missing out on any relevant predictors. The column ‘Goal’ contained the funding goal of startups in different currencies in its observations which made it irrelevant in the presence of the column containing the goal amount in USD. Lastly, the column ‘usd.pledged’ was removed because we had another column that contained the amount that was actually pledged in USD but using more recent conversion rates. The column ‘category’ was favored over ‘main_category’ because the former is a subset of the latter and thus would be correlated to it but would contain more information. The categorical predictors namely ‘category’ and ‘country’ were then converted into dummy variables. The unused levels for these were also removed from the data as part of the preprocessing steps.

We then tried to visualize the data in low dimensions by trying PCA and seeing if the first 3 principal components explained a significant proportion of the variance explained (>80%). In order to comply with the package requirements, the ‘int’ type predictors were converted into ‘numeric’. The ‘state’ predictor was also removed thereby creating a new data set. However, this approach did not yield a favorable result since the ‘Proportion of Variance Explained’ (PVE) by each principal component was very low and 80% of the variance was explained by a total of 150

Principal Components (Figure 1). Therefore, the principal component analysis can only reduce dimensionality but cannot be used for visualization.

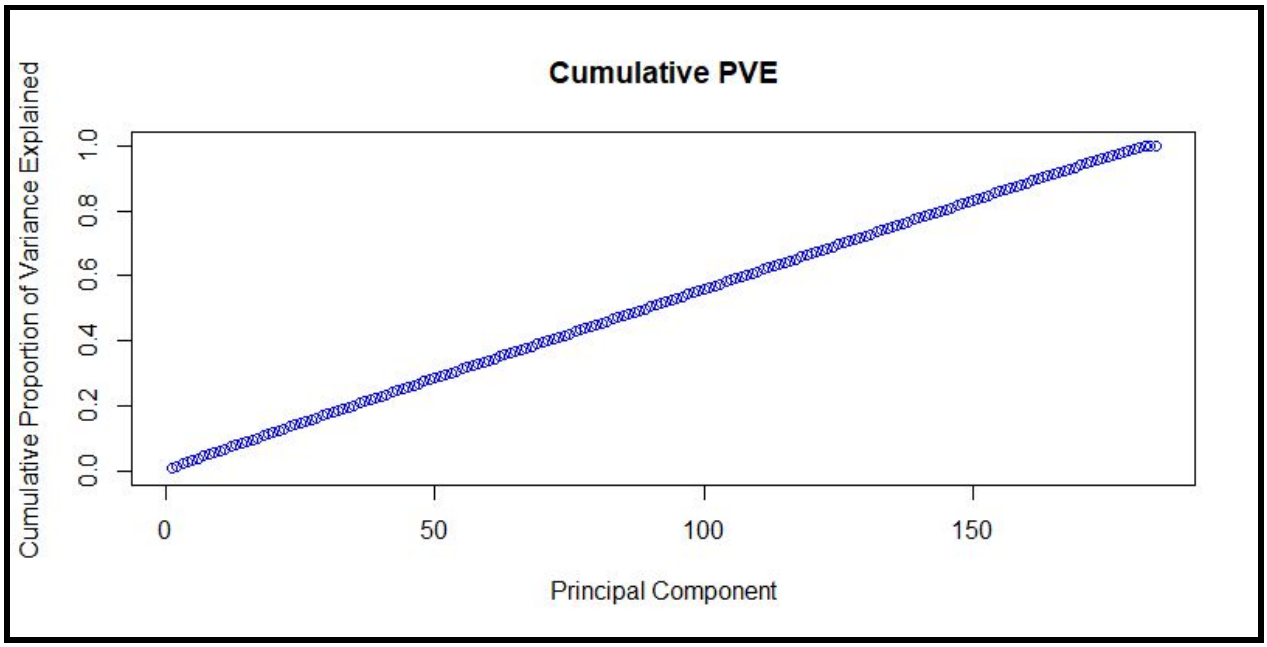

Figure 1

For regression, only the successful startups were considered as the amount of funding secured makes sense only for successful startups. The variance for the pledged amount for the whole data was calculated and observed to be $85,000.

For all the models, we utilized 80% of the observations for training the model and 20% of the observations as a validation set. We created two separate sets of training and test data and we normalized one of them. We used the normalized data to train and test models whose outcomes would get affected if the different predictors were on different scales and had differing variances such as linear regression, logistic regression, regularized regression, KNN, GAMs using regression splines. On models that are robust to such differences such as random forests and boosted trees we used the non-normalized data.

5. Exploratory Data Analysis

Fig.1 represents the number of crowdfunded startups listed on kickstarter from each country versus the name of the country. The team decided to take a look at the top contributors since there were a large number of countries. The table in the code showed that approximately 78% of the startups were from the United States, 9% of the startups were from Great Britain, and 4% of the startups were from Canada.

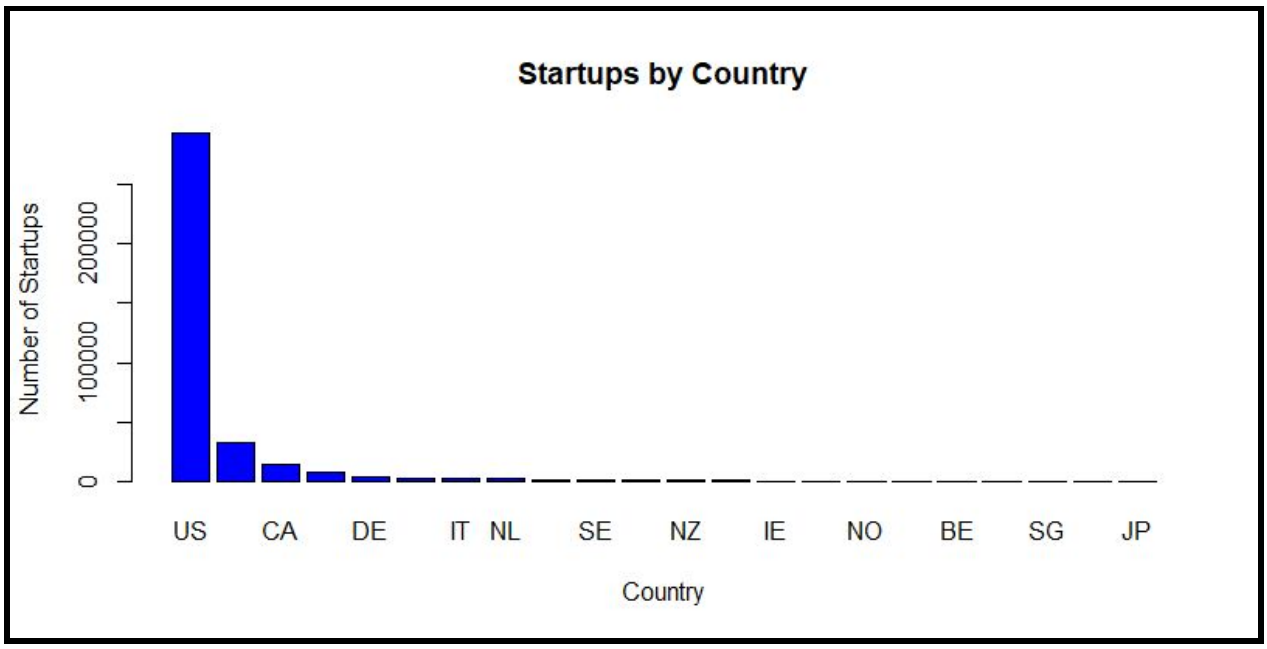

Figure 1

Fig. 2 represents the number of crowdfunded startups listed on kickstarter from each main category versus the name of the main category. The plot shows a more even spread of projects across categories as opposed to countries where the startups were primarily concentrated in the

US. The main categories that account for 80% of all startups were “Film and Video”, "Music”, "Publishing", "Games", "Technology", "Design", “Art” and “Food” . Furthermore, the 5 largest main categories in the US are "Film & Video", "Music", "Publishing", "Games" and "Art". Surprisingly, “Technology” based startups are less popular in the US than “Art” based ones.

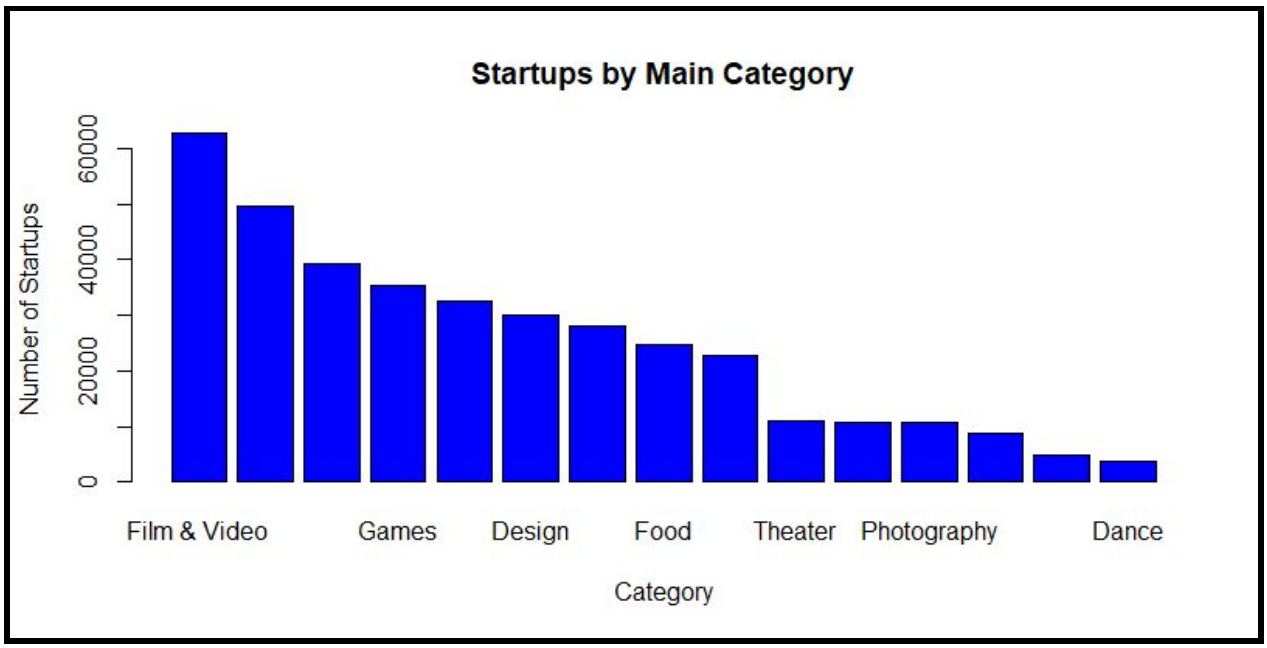

Figure 2

This basic analysis was done to get an idea of the data in terms of the countries where crowdfunded startups are most famous and in what areas.

6. Regression

The team first tried fitting a linear regression model to the dataset. Although some predictors did have a correlation with the pledged amount as suggested by the p-value (<2.2e-16) for the F-test, the linear model had an R-squared value of 0.5877 which conveys that the linear model

fails to explain a significant amount of the variance in the data. Thus we decided to move onto more flexible models by first selecting a subset of the predictors and regressing the pledged amount onto then using GAMs (with regression splines for the continuous predictors) and polynomial transformations.

Feature Selection was done using forward stepwise selection. We decided to use the features selected by the Bayesian Information Criterion (BIC). BIC selected a set of 13 features namely: the number of backers, funding goal, campaign length and the categories 3D Printing, Camera Equipment, Fabrication Tools, Flight, Gadgets, Gaming Hardware, Hardware, Product Design, Sound, Tabletop Games, Technology, Video Games and Wearables. Although Adjusted Rsq came up with 1 and Cp recommended a 6 predictor model we decided to go with a higher number because as it is we were not getting a good fit and did not want to reduce flexibility by too much.

We selected the degrees of freedom while fitting regression splines onto the 3 continuous variables among the 8 by using 10-fold cross validation over 20 different values for the degrees of freedom (Figure 3). Similarly, for polynomial transformations we again used 10-fold cross validation to tune the degree of polynomial (Figure 4). The lowest CV errors were obtained for 18 (~$12800) for the degrees of freedom of regression splines and for the degree 1 (~$13400) of the polynomial transformations of our predictors and we decided to used these values for our parameters.

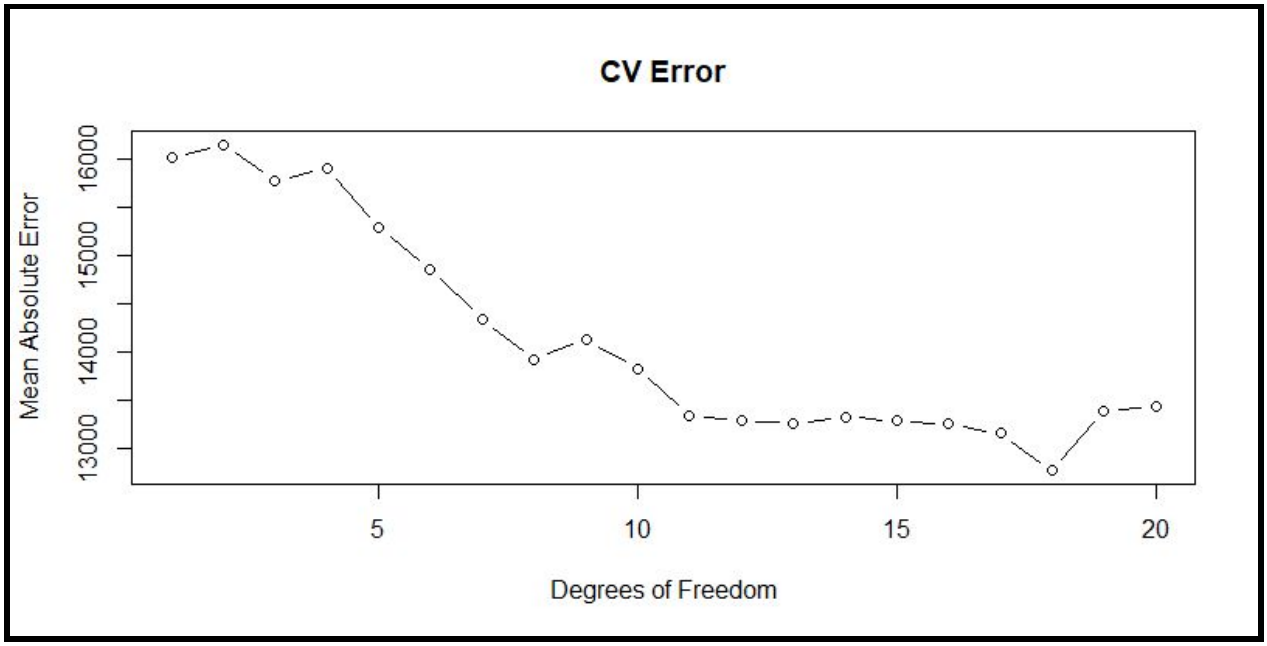

Figure 3

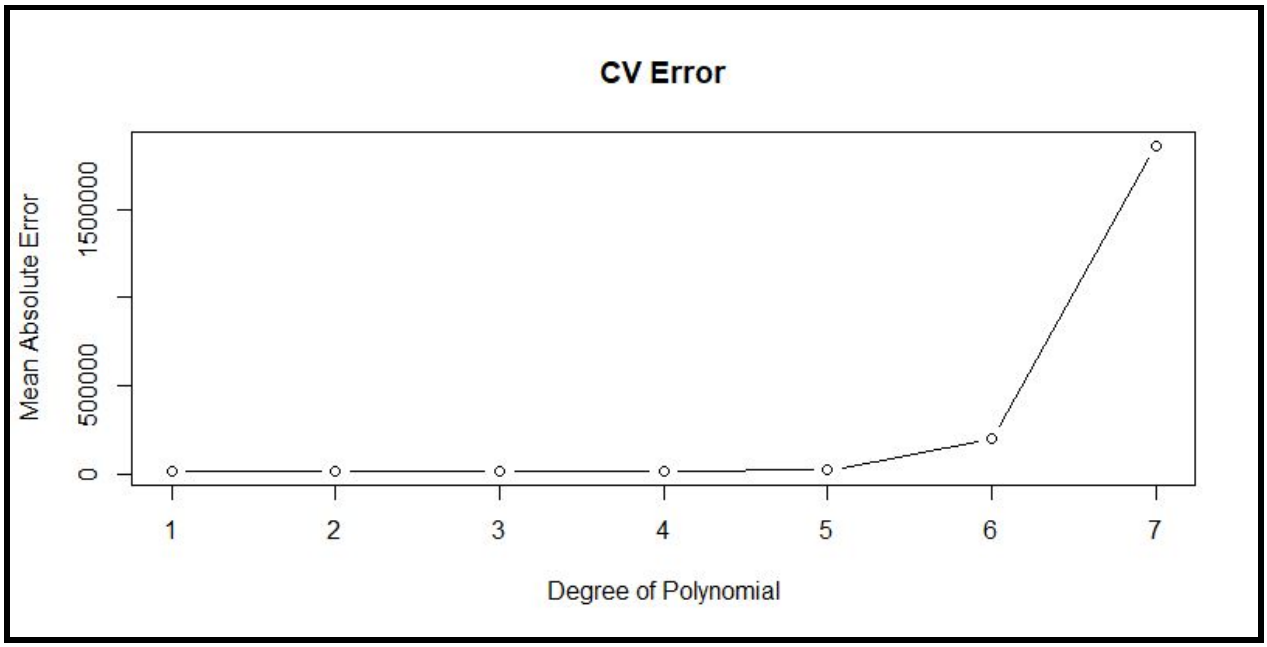

Figure 4

The mean absolute CV error for the tuned GAM was ~$12800 and that for the polynomial regressions model was $13400. So we concluded that the GAM was the better of the 2 due to its lower estimated test error.

We wanted to try even more flexible models to see if they would help us improve our predictions. So we started with KNN regression. We tuned the number of predictors using LOOCV on the training set and found that the 7 Nearest Neighbors model had the lowest estimated test error of ~$11000 (Figure 5).

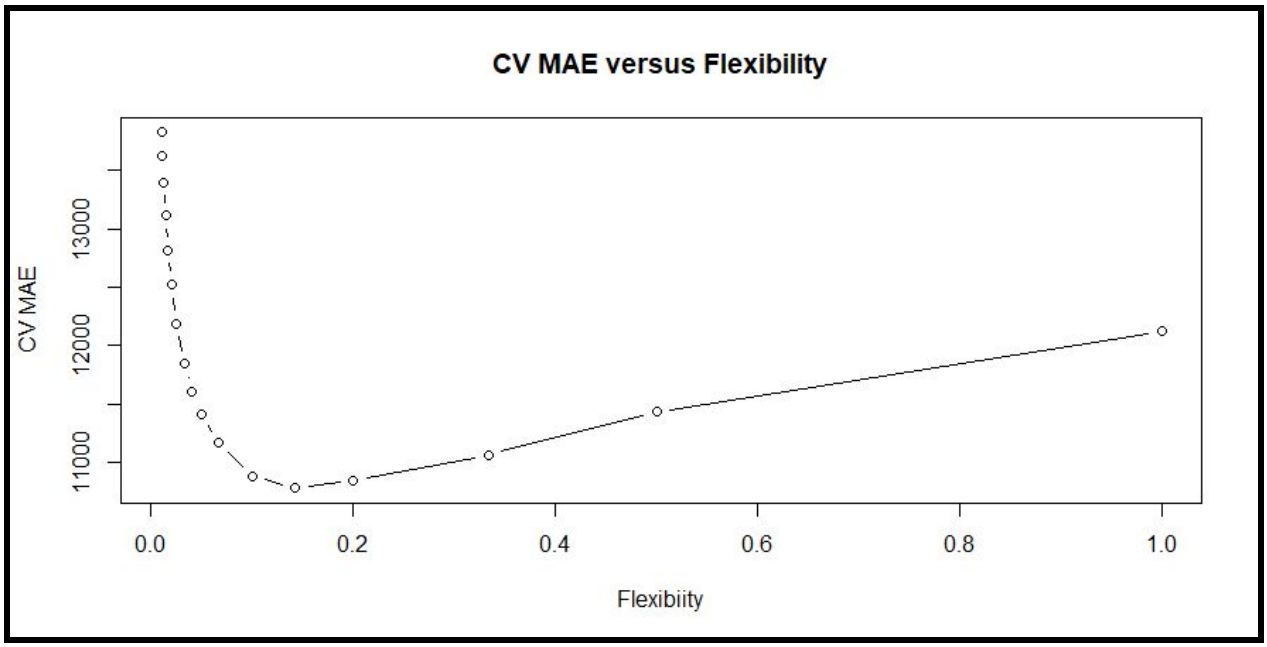

Figure 5

Next, the Random Forest method was applied to the dataset. A total of 500 trees were used as part of the analysis. It was found that running the model took a lot of time due to the massive number of data points. The oob MAE was ~$14,000 which is high compared with the other models and so we decided not to further tune the Random Forest due to computational limitations.

Finally, the boosting technique was used. The boosted tree was specified with a learning rate of 0.001, interaction depth of 4 and for the purpose of this analysis. The boosted tree was found to have an R-squared value of approximately 74% and the RSS value was $5.65*10^{14}$.

Classification

For classification, we decided to remove live projects since it did not make sense to include data points with labels that were not a success or failure yet. A value of '1' was assigned to successful startups and a value of '0' was assigned to the other categories () to convert it into a binary classification. Similar to the regression setting, 80% of the observations were used in the 'train set' and 20% of the observations were used in the 'test set'. However, there were not a lot of live projects in the dataset and the total number of data points we had were ~400000.

We started with trying out a linear classifier and chose the logistic regression for this. After training the model we performed 10-fold cross validation to estimate the test classification error rate for the model and it turned out to be ~2.8%. This was quite surprising because of the uncannily high accuracy and we thought this could be due to the particular random splits done in that run but we ran the cross validation several times and got similar results.

To avert the risk of overfitting we next tried out ridge and lasso regression by first tuning the penalty parameter using 10 fold cross validation. And since the cross validation had already been done while tuning the parameter and also since there were significantly more data points than regression which would make it computationally harder to do cross validation we decided to use the validation set approach. The error rates were ~28% and ~31% for the ridge and lasso regularized logistic regression, respectively.

To see if non-parametric models might perform any better we decided to try out KNN by tuning the number of neighbors (flexibility) using LOOCV (Figure 6) and found that the model with 5 nearest neighbors performed the best with an estimated test classification error rate of ~18%.

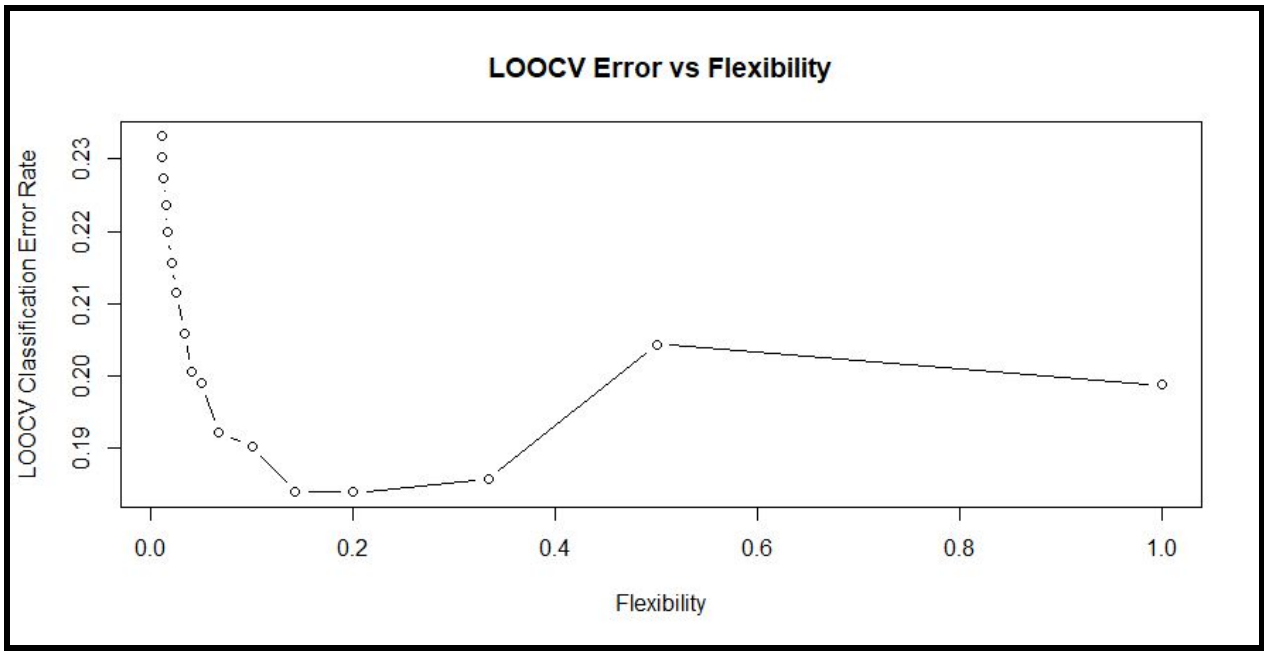

Figure 6

7. Conclusions

After our analysis we found that the K-Nearest Neighbor model was best suited for both our goals. We achieved a Mean Absolute Prediction Error of ~$11,000 in the regression problem and a misclassification error rate of ~18% for the classification problem. In addition to this the K-Nearest Neighbor classifier can also provide a probability estimate associated with each class (success or failure).

We believe that although this in itself does accomplish most of the goals we had set but the results could have been better with more predictors that had a stronger bearing on the success

of crowdfunded startups such as popularity (measured in views or likes) on social media platforms like YouTube or Instagram, the credentials of the team such as number of previous successful startups, education level or experience in the domain in which the startup is placed and mentions on startup websites like Crunchbase.