Project Proxy – Predicting Success of Early Stage Startups

Tsz Chun Ho & Yiğit Ihlamur
Vela Partners

July 1, 2022

Abstract

In this report we use XGBoost classifiers and neural networks to distinguish between companies invested by failed and successful investors, and to distinguish between companies invested by failed and brand investors. We find that the most important attributes are 'city' and 'top_success_prev_comp_flag'. The code can be found at https://github.com/aaronhtc/Proxy.

1 Introduction

Predicting future outcomes in early-stage investing is more difficult than other investment fields. There are less numbers and more qualitative features. With the proliferation of data about private companies and advances in machine learning, we can build models to predict the future performance of a startup much better than relying on our intuitions. Prior work at Vela proved that investors of a startup are the most important predictive feature of future success. Our thesis is that this feature is a critical proxy signal that we should deep dive and understand the correlation of other attributes deeper. In this project, our goal is to build a model to predict if successful investors are going to invest in a startup.

2 Data

We are given 3 datasets which detail attributes of companies invested by failed, successful and brand investors respectively. Successful investors have a relatively successful history of investments based on previous quantitative analysis, while failed investors have a relatively unsuccessful history of investments. Brand investors are the ones that have a well-known brand in the market (most of them are also successful investors). The datasets are an extraction of investments from a subjective list of investors.

Each dataset consists of 3 data sheets: List, Academic and Work. The 'List' sheet has companies with their main attributes. The 'Academic' sheet contains a mostly academic background of the founders. The 'Work' sheet lists the companies that the founders worked before and what their titles were. The organization name is the unique field that can be used to connect the 3 data sheets.

For each dataset, we merge the data sheets into a single data frame, only including the companies which have data in all 3 data sheets. Then, we merge the data frames for the 3 datasets. To distinguish between the data, we add the response variable 'success_flag', which is set to be 0 for companies invested by a failed investor (4989 instances), 1 for companies invested by a successful investor (2030 instances), and 2 for companies invested by a brand investor (1430 instances). There are 8449 organizations in total. The explanatory variables that we will consider are shown below:

- 'city': The city that the organization is based on.
- 'category_groups_list': The broad categories that each organization belongs to.
- 'universities_of_founders': The universities attended by the founders.
- 'degrees_of_founders': The university degrees obtained by the founders.
- 'gender_of_founders': The genders of the founders.

- ‘prev_companies_of_founders’: The companies that the founders worked before.

3 Feature Engineering and Exploratory Analysis

Following previous work, we use the QS World University Rankings 2022 dataset, where an overall score between 20 and 100 is assigned to each university. We map each university attended by the founders to the overall score of the corresponding university in the QS Rankings. For the universities that don’t appear exactly the same as those in the QS Rankings, we get the closest match (if it exists) using the Difflib library. If a closest match does not exist, the overall score is set to 20. For each organization, we can now compute the number of universities attended by the founders (‘num_universities’), the maximum score of the universities attended by the founders (‘maximum_founders_university_score’), the minimum score of the universities attended by the founders (‘minimum_founders_university_score’), and the average score of the universities attended by the founders (‘average_founders_university_score’). The summaries are shown in Figure 1. We see that all of the features are generally higher for companies invested by successful and brand investors.

We also use the ‘degrees_of_founders’ attribute to generate the ‘doctoral’ attribute, which is equal to 1 if any founder in the organization has a doctoral or post-doctoral degree, and is equal to 0 otherwise. 1063 companies have ‘doctoral’ equal to 1. The bar plot in Figure 2 shows that companies that are invested by brand investors have more founders with doctoral or postdoctoral degrees. However, this phenomenon is not apparent for companies invested by successful investors.

Next, we obtain the number of founders (‘num_founders’) from the attribute ‘gender_of_founders’. ‘gender_of_founders’ has 6 missing data, and for these columns we set ‘num_founders’ to be the number of universities attended by the founders. We also replace the ‘num_universities’ feature by ‘university_founder_proportion’, which is simply the ‘num_universities’ column divided by the ‘num_founders’ column. Figure 3 shows that the distribution of the number of founders are similar for all success flags. We also notice that for most organizations, the number of universities attended is equal to the number of founders. 823 organizations have number of universities greater than the number of founders, whereas no organizations have number of universities smaller than the number of founders. We keep the ‘university_founder_proportion’ feature as the features ‘maximum_founders_university_score’ and ‘minimum_founders_university_score’ depend on it. We also obtain the proportion of male founders (‘male_founders_proportion’) for each organization. The box plot shows that the organizations in the dataset are dominated by male founders. We also see that companies invested by successful and brand investors have a higher proportion of male founders overall.

Using the method in previous work, we use the attribute ‘prev_companies_of_founders’ to get the successful previous employers of the founders with more than 25 former employees in the dataset. Here we say that a previous employer is successful if it has a successfulness greater than 0.5, where successfulness is defined as the number of former employees working at a company in the dataset invested by a successful or brand investor, divided by the total number of former employees in the dataset.

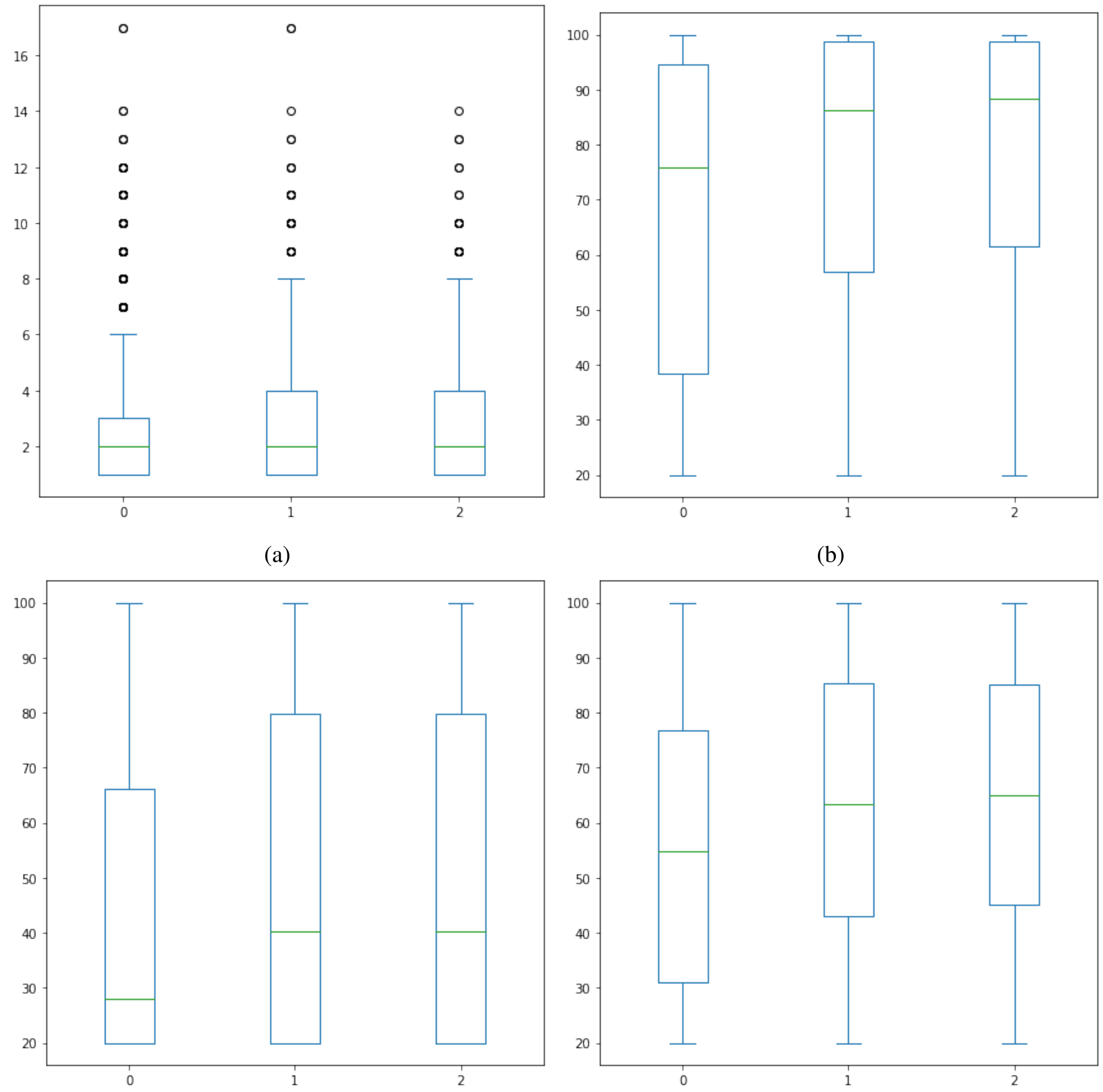
Most successful previous companies with more than 25 former employees, in decreasing order of successfulness: Pilot, Embark, Compass, Blend, Coinbase, Dropbox, Sun Microsystems, Sprig, VMware, Pioneer Fund.

We then create the feature ‘top_success_prev_comp_flag’, which is equal to 1 if any founder of the company was previously employed at a successful previous company with more than 25 former employees in the dataset, and is equal to 0 otherwise. We note that this induces a slight bias in the test performance of the models learned in the next section as the feature depends on the success flags of the whole dataset. The approach which does not induce such bias is to split the dataset into training, validation and test sets before generating the feature with the training dataset. For simplicity We do not do this here. The plots for the counts and proportions of companies grouped by ‘top_success_prev_comp_flag’ and ‘success_flag’ are shown in Figure 4.

We also obtain the top 10 most popular previous employers, where the popularity of each previous employer is quantified by counting the number of instances in the dataset.

Most popular previous companies, in decreasing order of popularity: Google, Microsoft, Meta, Yahoo, Techstars, Amazon, Apple, Stanford University, McKinsey & Company, Twitter.

We create the feature ‘top_popular_prev_comp_flag’, which is equal to 1 if any founder of the company was previously employed at one of the top 10 most popular previous companies, and is equal to 0 otherwise. Figure 5 shows that companies with founders who were previously employed at the most popular companies are more likely to attract successful and brand investors.



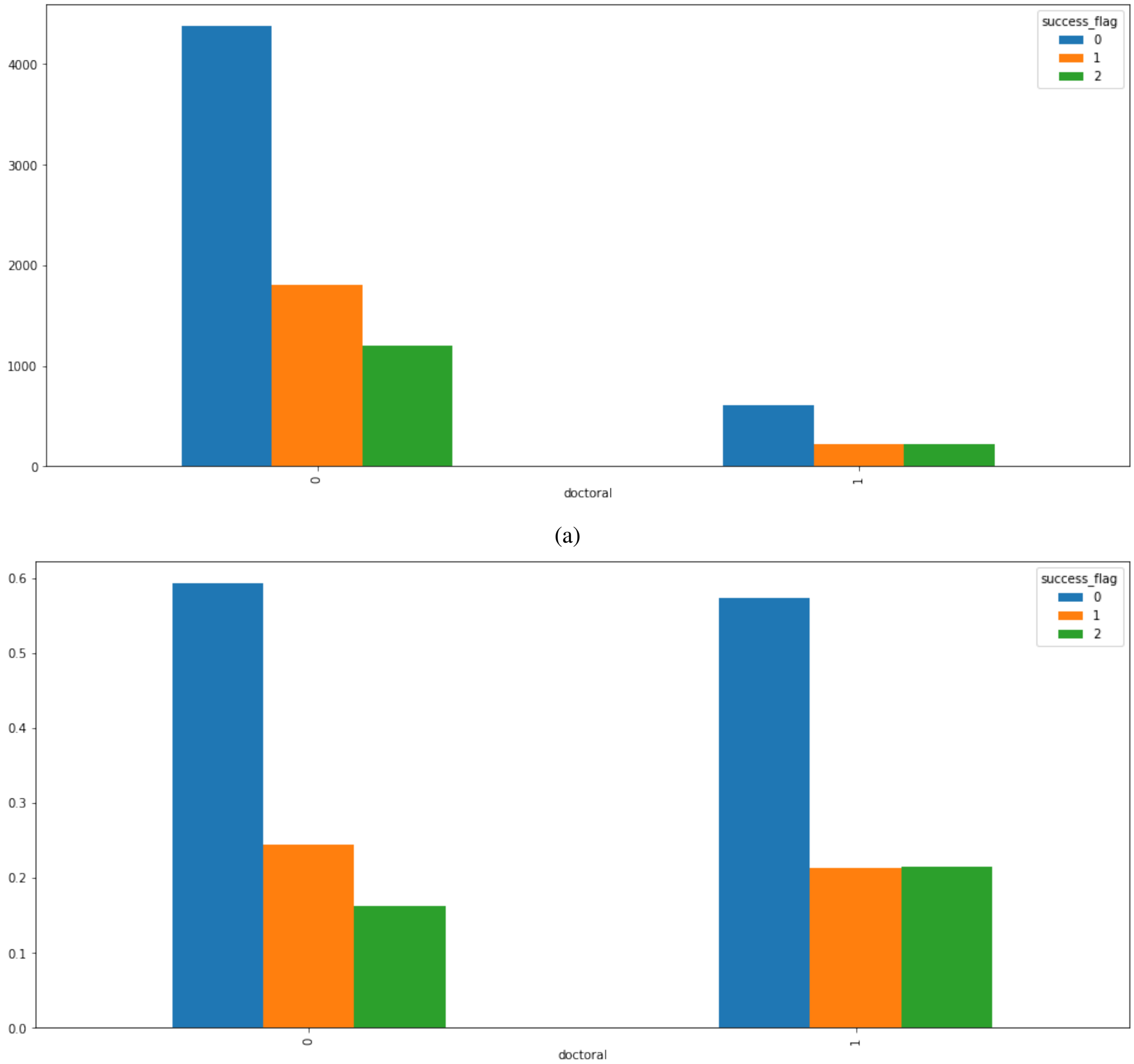
(a)

(b)

(c)

(d)

Figure 1: Box plots, grouped by success flag, of (a) the number of universities attended by the founders, (b) the maximum score of the universities attended by the founders, (c) the minimum score of the universities attended by the founders, and (d) the average score of the universities attended by the founders.



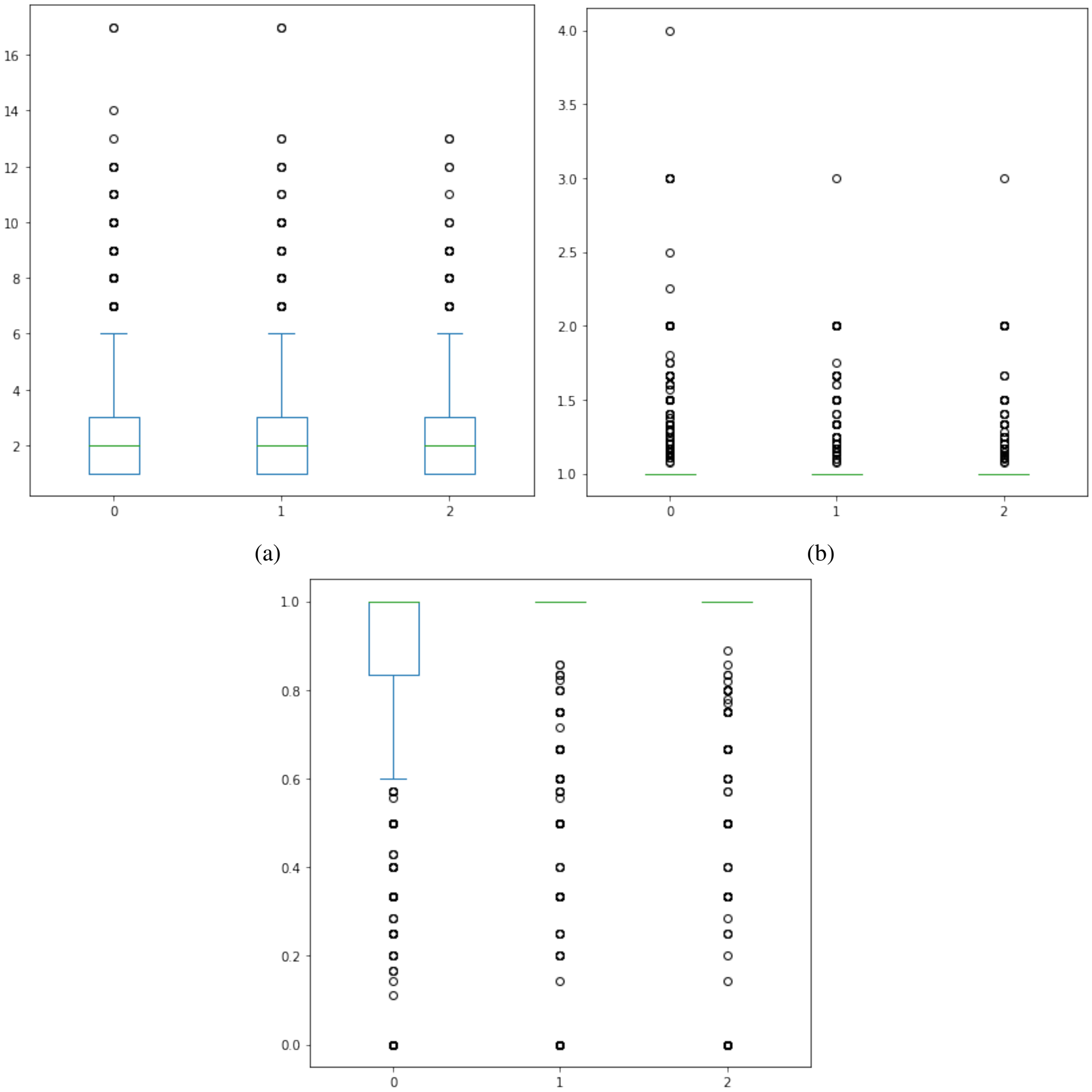
(a)

(b)

Figure 2: Bar plots, grouped by ‘doctoral’ value, of (a) the number of companies for each success flag, and (b) the proportion of companies for each success flag.

To extract the features from the ‘category_groups_list’ attribute, we create 47 columns where each column corresponds to a category group. For each column, the feature is set to 1 if the company belongs to the category group, and is set to 0 otherwise. Most companies belong to more than 1 category group. The bar plots are shown in Figure 6, and we can see a variation in the proportion of companies invested by successful and brand investors for each category group.

Lastly, we modify the ‘city’ feature by keeping the cities with more than 200 instances (which are San Francisco, New York, Los Angeles, Palo Alto and Boston) as categories, and group all the other cities into the other_cities category. Figure 7 shows that the 5 major cities have a significantly larger proportion of companies that are invested by successful investors. San Francisco and Palo Alto also have a significantly larger proportion of companies that are invested by brand investors.

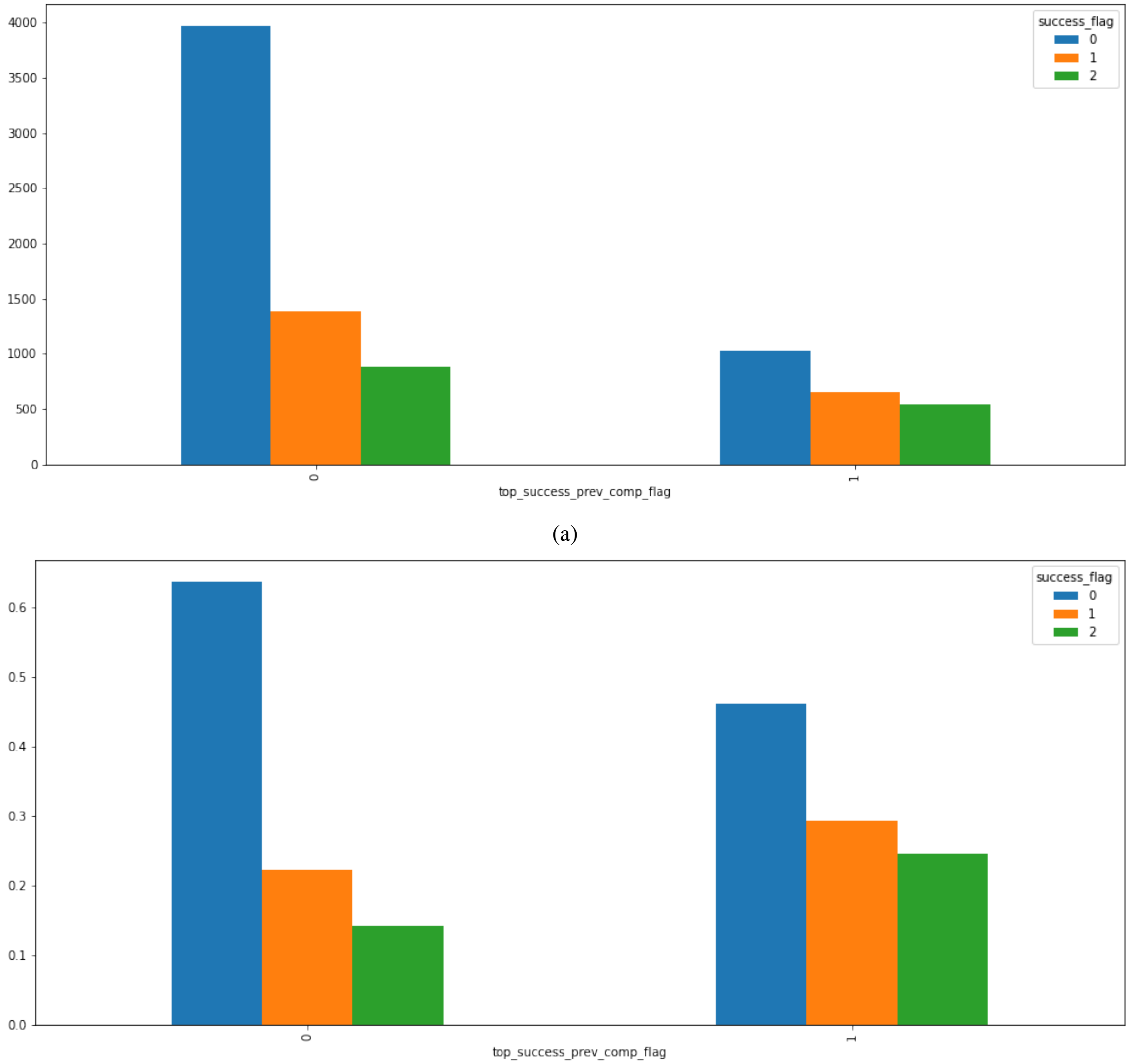


(a)

(b)

(c)

Figure 3: Box plots, grouped by success flag, of (a) the number of founders in the organization, (b) the number of universities attended divided by the number of founders, and (c) the proportion of male founders.



(a)

(b)

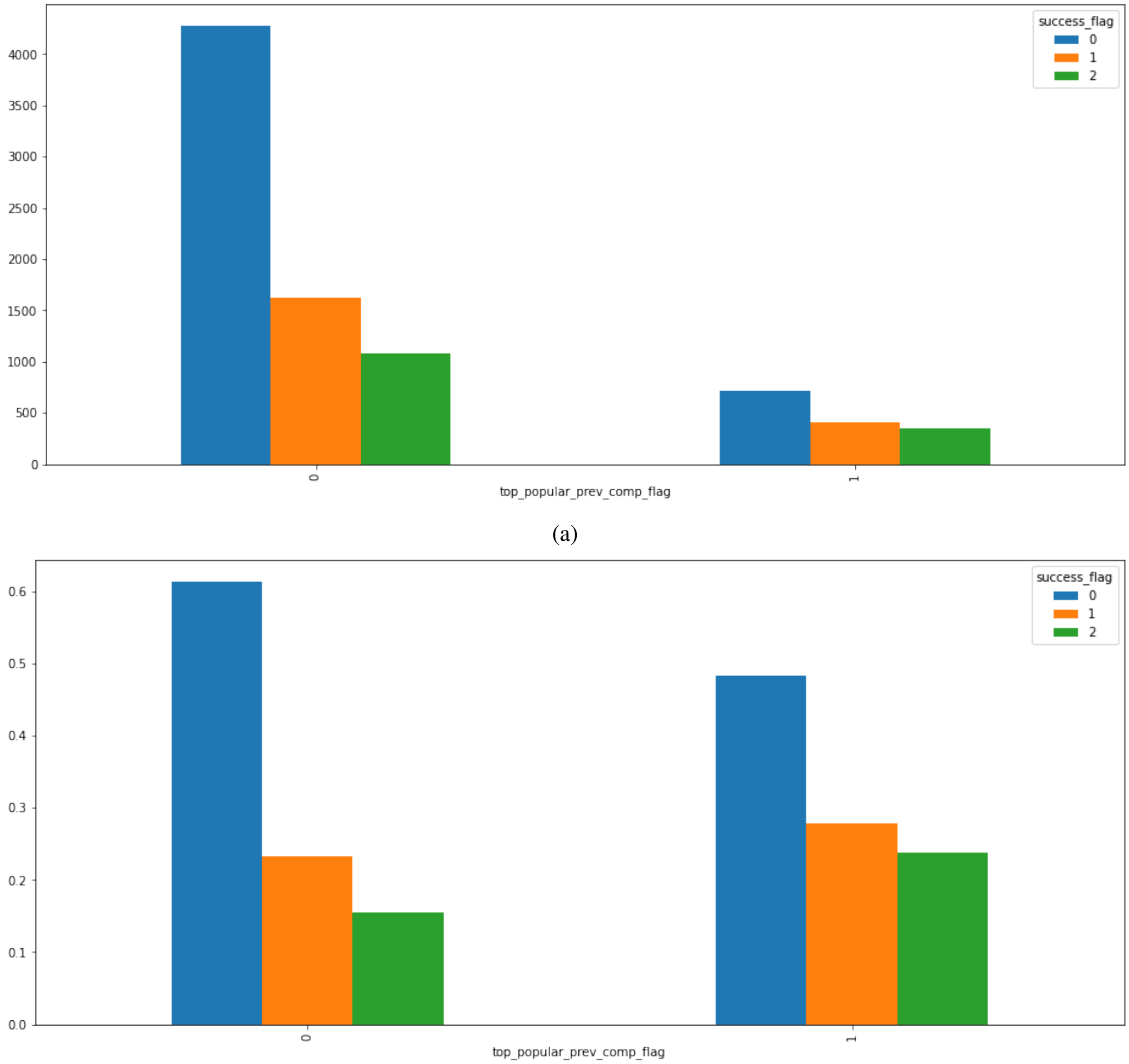
Figure 4: Bar plots, grouped by 'top_success_prev_comp_flag' value, of (a) the number of companies for each success flag, and (b) the proportion of companies for each success flag.

4 Model Training and Evaluation

Our objective is to classify between companies invested by failed and successful investors, and to classify between companies invested by failed and brand investors. We are interested in generating models with high precision and recall, with precision taking priority.

We train models separately over 2 datasets: the dataset which only includes companies invested by failed or successful investors (we call it the failed/successful dataset), and the dataset which only includes companies invested by failed or brand investors (the failed/brand dataset). For the failed/brand dataset, we change the success flag for companies invested by brand investors from 2 to 1. For each dataset, 20% of the datapoints are used for testing.

We consider 2 distinct families of models for classification: the XGBoost classifier and the neural network.



(a)

(b)

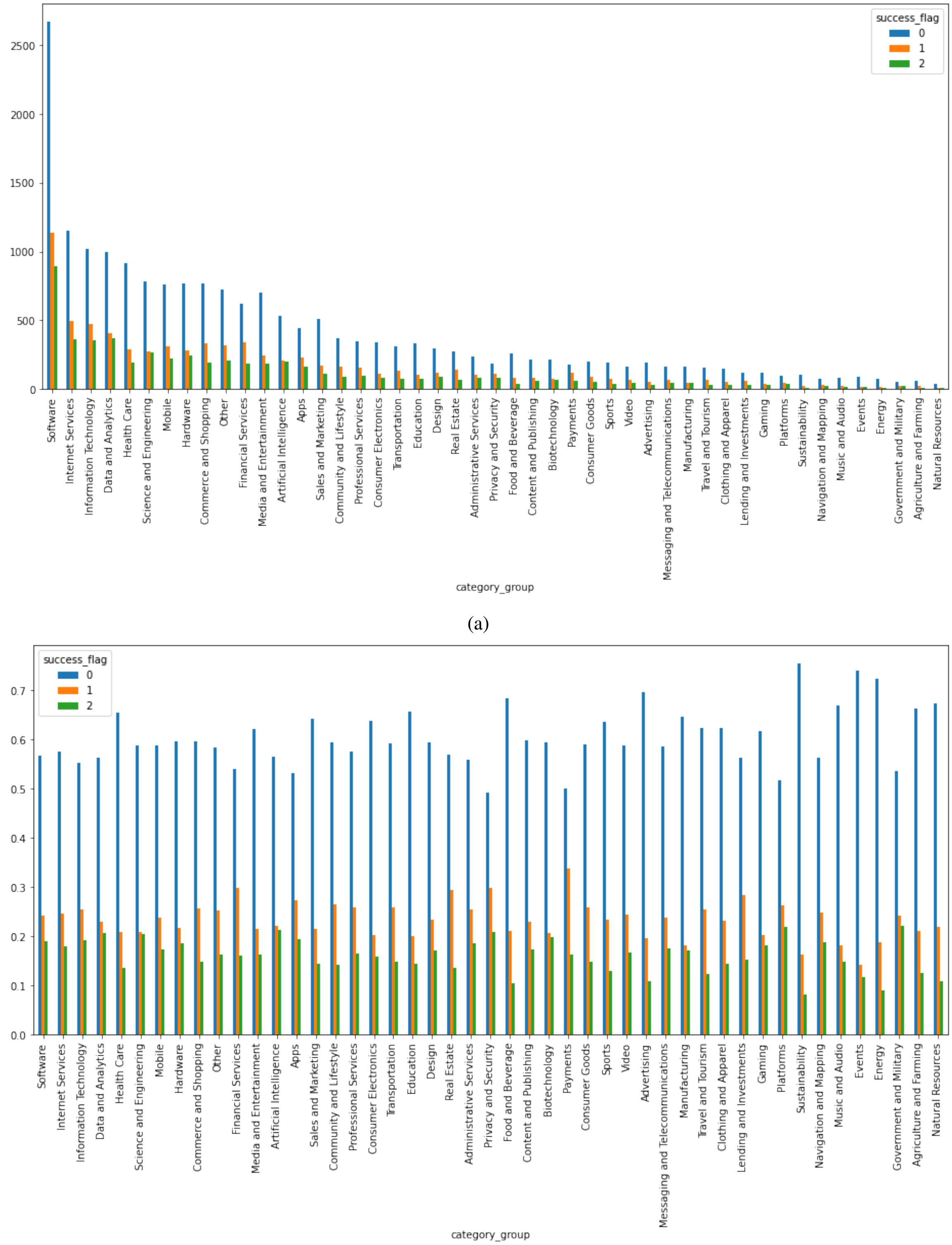
Figure 5: Bar plots, grouped by ‘top_popular_prev_comp_flag’ value, of (a) the number of companies for each success flag, and (b) the proportion of companies for each success flag.

4.1 XGBoost Classifier

We use GridSearchCV for hyperparameter tuning, which performs 5-fold cross validation for each choice of hyperparameters. The hyperparameters that are considered are shown below:

- ‘max_depth’: 3, 6
- ‘n_estimators’: 100, 500
- ‘learning_rate’: 0.1, 0.3
- ‘colsample_bytree’: 0.5, 0.75, 1.0
- ‘subsample’: 0.6, 0.8, 1

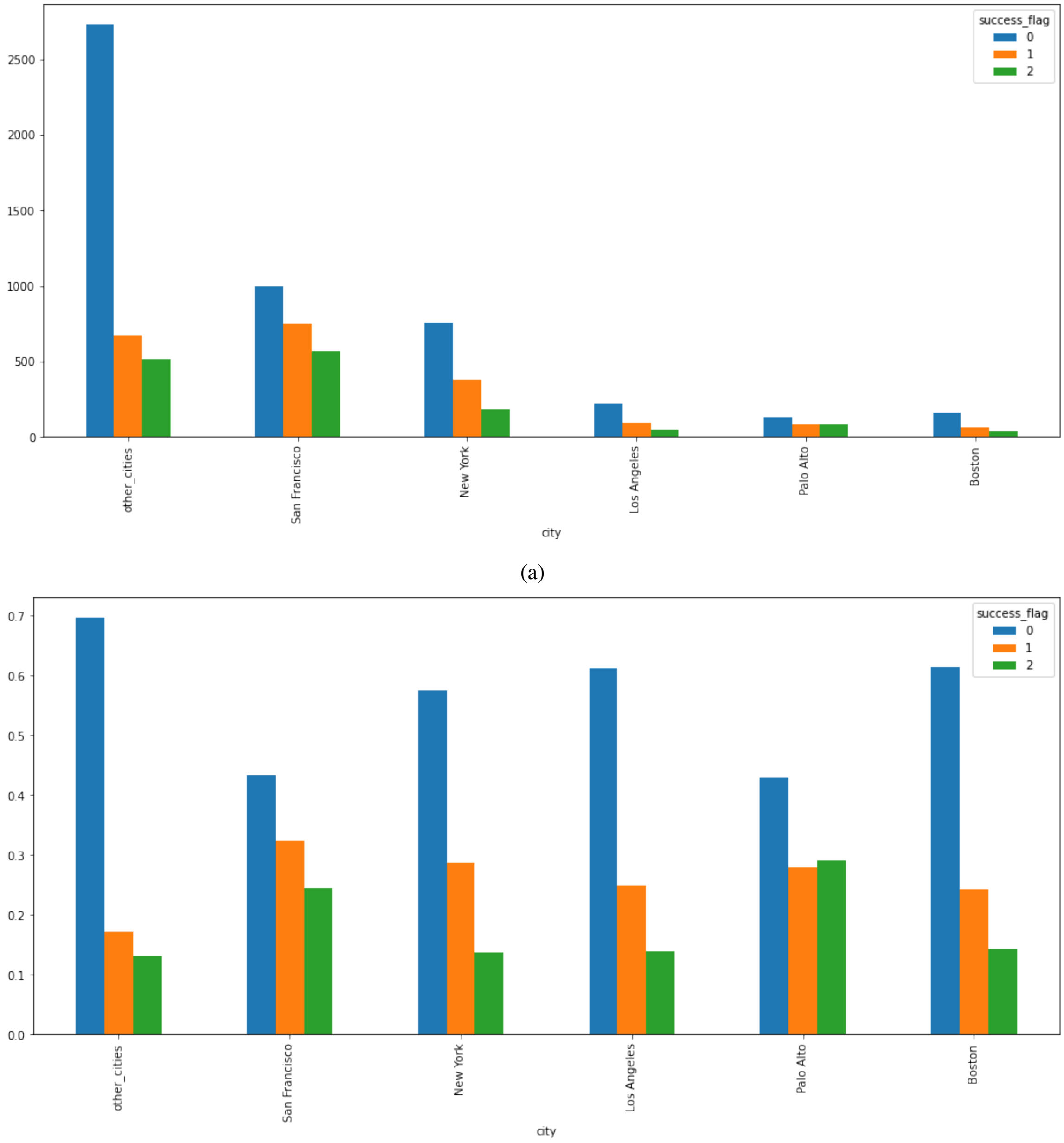
Default arguments are used for the other hyperparameters. We choose the hyperparameters which result in the highest precision score, which we found to be ’colsample_bytree’$=0.75$, ’learning_rate’$=0.1$, ’max_depth’$=3$, ’n_estimators’$=100$ and ’subsample’$=1$ for the failed/successful dataset, and ’colsample_bytree’$=0.5$, ’learning_rate’$=0.1$, ’max_depth’$=3$, ’n_estimators’$=100$ and ’subsample’$=0.6$ for the failed/brand dataset.



(a)

(b)

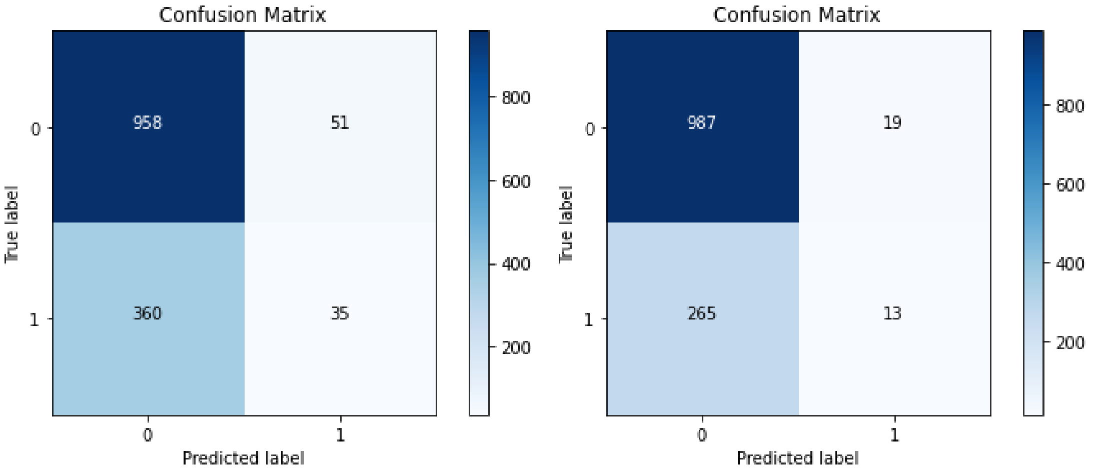
Figure 6: Bar plots, grouped by category group, of (a) the number of companies for each success flag, and (b) the proportion of companies for each success flag.



(a)

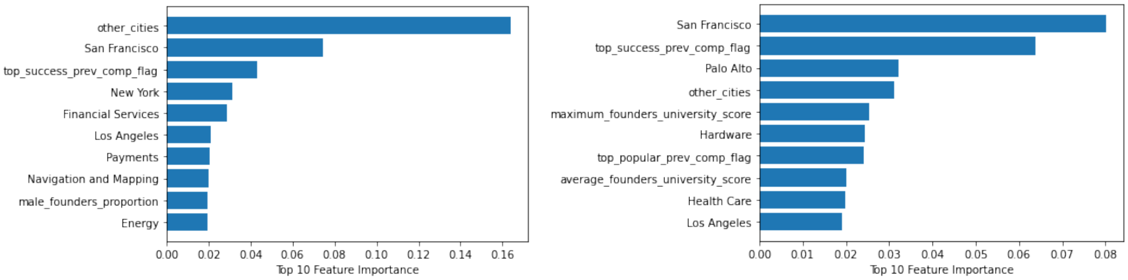
(b)

Figure 7: Bar plots, grouped by city, of (a) the number of companies for each success flag, and (b) the proportion of companies for each success flag.



(a) (b)

Figure 8: Confusion matrices over the test sets of XGBoost classifiers trained on (a) the failed/successful dataset and (b) the failed/brand dataset.



(a) (b)

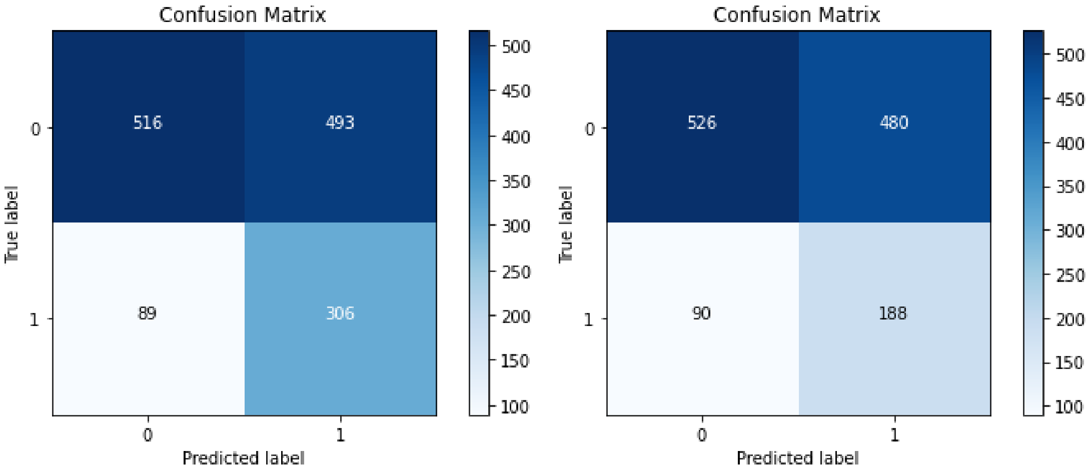
Figure 9: Plots of top 10 feature importances (based on importance_type = 'weight') of XGBoost classifiers trained on (a) the failed/successful dataset and (b) the failed/brand dataset.

We then evaluate the models on the test sets. For the failed/successful dataset, the test precision and recall are 40.7% and 8.9% respectively, while for the failed/brand dataset, the test precision and recall are 40.6% and 4.7% respectively. The confusion matrices are shown in Figure 8. We also plot the feature importances in Figure 9, and we see that for both datasets, the most important attributes are 'city' and 'top_success_prev_comp_flag'.

Despite having high precision scores, this method has some major drawbacks: Firstly, it is not flexible as we only output 1 model from each dataset, and we cannot control the number of companies which are predicted to be invested by successful/brand investors. This means that we cannot modify the model based on the precision-recall tradeoff. For the models obtained, very few instances are predicted to be invested by successful/brand investors, and are much less than the actual number of companies that are invested by successful/brand investors. Secondly, although the hyperparameters are chosen to maximize precision, the gradient boosted trees themselves are not designed to optimize precision, so this approach may not the most suitable for our goals.

4.2 Neural Network

After standardizing the data, we use a multi-layer perceptron with 3 hidden layers. The first hidden layer has 50 neurons and ReLU activation, the second hidden layer has 25 neurons and ReLU activation, and the third hidden layer has 12 neurons, ReLU activation, and dropout with dropout rate 0.5. The output layer has 1 neuron and sigmoid activation. We use the binary cross entropy loss and Adam as the optimizer. In the first stage, 10% of the datapoints in the training set is used as a validation set, and the rest of the training set datapoints are used for training. We track the validation loss over 100 epochs, and estimate the number of



(a)

(b)

Figure 10: Confusion matrices over the test sets of neural networks trained on (a) the failed/successful dataset and (b) the failed/brand dataset.

epochs until the validation loss stops decreasing, which we find to be 40. In the second stage, we train a new model over the entire training set for 40 epochs, and evaluate it on the test set.

We notice that the resulting neural networks classify almost all companies to to be invested by failed investors. However, this can be easily modified by shifting the decision boundary. Namely, the neural network outputs a value between 0 and 1, which represents the likeliness that a company is invested by a successful/brand investor. For classification which maximizes accuracy, the threshold is set to 0.5, i.e. companies with output less than 0.5 are predicted to be invested by a failed investor. By decreasing the threshold, we generally obtain models with lower precision but higher recall. This threshold can be chosen arbitrarily. Here we set the threshold to be 0.3. For the failed/successful dataset, the test precision and recall are 38.3% and 77.5% respectively, while for the failed/brand dataset, the test precision and recall are 28.1% and 67.6% respectively. In particular, for the label corresponding to companies invested by successful/brand investors, the predictions are significantly better than random chance. The confusion matrices are shown in Figure 10.

5 Conclusion

In this report we use XGBoost classifiers and neural networks to distinguish between companies invested by failed and successful investors, and to distinguish between companies invested by failed and brand investors. We find that the most important attributes are ‘city’ and ‘top_success_prev_comp_flag’. Further improvements may be made by extracting a larger quantity and quality of features from the original dataset, and by tuning the hyperparameters of the neural network.