Predicting a Startup's Acquisition Status

Abstract

This project predicts a startup's acquisition status based on its financial statistics. In order to overcome the main challenge of biased data without under/oversampling the data, a novel ensemble model used. The resulting model combines a high precision model with a high accuracy model trained on a dataset transformed by the first model. Preliminary experiments suggest that this new model has the potential to yield higher precision predictions while preserving performance with respect to accuracy and weighted recall.

1 Introduction

The goal of this project is to predict a former startup's acquisition status based on a company's financial statistics. While the area of using machine learning to predict IPO underpricing has been well-researched, this topic has been surprisingly understudied. The results of this project may be of particular interest to investors as well as job applicants to pre-IPO companies as it can be extended to look at the likelihood of the prospective company being acquired, closed or reaching and IPO. The results of this project may also give insight to which features have the most influence on the predictions.

The resulting algorithm takes in a startup's financial statistics such as total funding dollars, funding dates, number of funding rounds, and headquarter location as inputs. The algorithm then predicts whether the startup has been closed, acquired, is operating, or has reached and IPO. The main challenge for this problem is dealing with an imbalanced dataset where one class is overrepresented, but under/oversampling cannot be used as a technique to balance the data. In order to address this, an ensemble-based technique that combines the results of a high precision anomaly detection algorithm (QDA) with a random forest classifier.

2 Related Work

Not a lot of work has been done on this specific type of problem. However, there are many ways to address biased data such as using bias-resilient models, over/undersampling the data, Boosting, etc.[2] that are typically taught in introductory classes. More recent research suggest that anomaly detection techniques[3] [5] trained for each individual class can also be promising. This paper builds off of these techniques by trying to apply an anomaly detection models in a novel way to modify a training set to be more balanced.

3 Dataset and Features

The dataset used for this project is a Kaggle dataset sourced from Crunchbase called 'Crunchbase 2013 - Companies, Investors, etc.'[4] There are $n = 17,727$ samples and each row of the dataset contains a startup's information. Specifically, these are: company name, website, sector category, funding received, headquarter location (city and state names), funding rounds, founding date, first and last funding dates, and last milestone date. Each row is also labeled with the company's status ('Acquired', 'Closed', 'IPO', 'Operating'). The dataset labels show that the dataset is

extremely biased. As shown in Table 1, the 'Operating' class is extremely over-represented and the other classes are under-represented.

IPO	Closed	Acquired	Operating
1.9%	3.1%	9.4%	85.6%

Table 1: The distribution of classes for the dataset show that it is biased, with the 'Operating' class over-represented and the other classes under-represented.

The data is pre-processed to transform the qualitative features into more meaningful ones. In particular, the following two transformations are performed on the dataset:

- All date features are converted from strings into two integers corresponding to month and year
- All headquarter location features are converted from strings into floats correspond to longitude, lattitude, and city/state 'importance', which are given by the GeoPy API [1]

After centering and scaling all of the features, the dataset is then shuffled to remove any biases in the order they appear, and split 60/20/20 to form the training, validation, and test sets used for the project.

4 Method

4.1 Initial Attempt - Logistic Regression

This is a multi-class classification problem (with only a small number of classes), so it initially seemed reasonable to apply a basic one-versus-all classification technique such as logistic regression to the problem. However, the resulting model performed poorly because logistic regression is succeptible to the biased data. Furthermore, balancing the data either by over or under-sampling creates a model that would not be applicaable to real applications. This is because logistic regression is derived from a maximum likelihood interpretation where

$$P(y = 1|x; \theta) = h_\theta(x)$$
$$P(y = 0|x; \theta) = 1 - h_\theta(x)$$

so the maximum likelihood of the parameters generating the predictions is given by maximizing

$$\begin{aligned} L(\theta) &= p(\vec{y}|X; \theta) \\ &= \prod_{i=1}^{n} p(y^{(i)}|x^{(i)}; \theta) \\ &= \prod_{i=1}^{n} (h_\theta(x^{(i)}))^{y^{(i)}} (1 - h_\theta(x^{(i)}))^{1-y^{(i)}} \end{aligned}$$

which is equivalent to maximizing the log likelihood

$$\log(L(\theta)) = \sum_{i=1}^{n} y^{(i)} \log(h(x^{(i)})) + (1 - y^{(i)}) \log(1 - h(x^{(i)})).$$

Maximizing the log likelihood is exactly how the parameters for logistic regression are obtained. However the derivation shows that the model maximizes the performance of the model over the distribution of the training set, so it would not perform as well on a test set with a different distribution.

4.2 Ensemble Technique

The underlying challenge with the dataset is the over-representation of 'Operating' classes. Any model can obtain a high accuracy and recall by over-prediction 'Operating', but to the detriment of precision. An ensemble technique is used to attempt to address this. The general idea of the technique is to chain together two models that are trained with a specialized goal. The two models can be described as follows:

1. An anomaly detection model that can identify *with high precision* members of the over-represented class

2. A classification model that has accuracy when the input data is sampled from a *more balanced* distribution of classes

To obtain these models, the following procedure is done:

1. Randomly split the training data into two sets T_1, T_2
2. Train M_1 using quadratic discriminant analysis (QDA) trained on T_1 to identify members of the over-represented class with high precision (i.e., has a high threshold)
3. Obtain a modified second training set T_2' by applying M_1 as a 'sieve' on T_2 to remove datapoints that are predicted to be in the over-represented class
4. Train M_2 as a random forest (RF) classifer using T_2'

Notice that applying M_1 to T_2 changes the distribution of the data M_2 is trained on, and M_2 is optimized for data with a similar distribution to T_2'. It is extremely important to split the training data into two separate sets because the M_1 will overfit T_1, so even if T_1 and T_2 are sampled from the same distribution, applying M_1 to the two training sets will give T_1' and T_2' that have different distributions, and an RF classifier trained on T_1' will be suboptimal on T_2'.

Predictions on input x, are given by the following formula:

$$\text{output} = \begin{cases} \text{'Operating' if } M_1(x) = \text{ 'Operating'} \\ M_2(x) \text{ otherwise} \end{cases}$$

The final model is summarized in Figure 1 below.

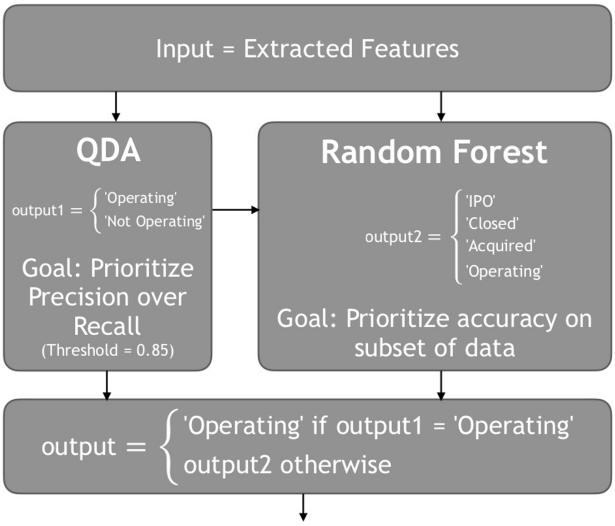

Figure 1: Information flow diagram for a prediction on the input.

5 Experiments/Results/Discussion

5.1 Performance Metrics

The performance of the different models are compared by examining the accuracy of their predictions on the validation set. Since this is a classification problem with a small number of classes, a reasonable definition of accuracy would be the ratio of correct classifications to total number of corrections. That is, the accuracy of an algorithm h is given by

$$\text{accuracy}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[h(x^{(i)}) == y^{(i)}].$$

Furthermore, since the dataset is so biased, the weighted precision, weighted recall, and weighted F_1 scores are also considered, which are given by the following formulas:

$$\begin{aligned}
\text{TP}_c &= \sum_{i=1}^{n} \mathbb{1}[y^{(i)} = c \text{ and } h(x^{(i)}) = c] \\
\text{FP}_c &= \sum_{i=1}^{n} \mathbb{1}[y^{(i)} \neq c \text{ and } h(x^{(i)}) = c] \\
\text{FN}_c &= \sum_{i=1}^{n} \mathbb{1}[y^{(i)} = c \text{ and } h(x^{(i)}) \neq c] \\
\text{Precision}_c &= \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \\
\text{Recall}_c &= \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \\
\text{weighted precision} &= \sum_{c \in C} \frac{\text{\# samples in } c}{n} \cdot \text{Precision}_c \\
\text{weighted recall} &= \sum_{c \in C} \frac{\text{\# samples in } c}{n} \cdot \text{Recall}_c \\
\text{weighted } F_1 &= \sum_{c \in C} \frac{\text{\# samples in } c}{n} \cdot 2\frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}
\end{aligned}$$

where $C = \{\text{'Operating', 'IPO', 'Closed', 'Acquired'}\}$ is the set of all classes.

5.2 Baseline

A trivial model that makes a constant prediction is used to estabnlish a baseline. Since the 'Operating' class is by far, the most frequent, this is the prediction that the baseline model makes. The baseline model performs extremely well with respect to accuracy and recall, but has low precision.

5.3 Comparison of Models

In order to show that the two-step ensemble technique improves precision without detriment to accuracy and recall, its performance is compared to the performances of the QDA (for multi-class classification) and RF classifiers that it is composed of. The subsections below outline how the models are optimized so that the comparison is fair.

5.3.1 Feature Selection

In order to reduce overfitting and to find the most optimal selection of features for QDA and RF classifiers, each model is trained on different combinations of the features. The strategy for picking feature combinations follows the forward selection algorithm, which greedily adds features that give the best performance with respect to accuracy when applied to the validation set [(2)]. Using this technique, we see that features such as number of funding rounds, total funding, and headquarter location importance are consistently chosen, which suggests that they are strong predictors for a startup's acquisition status.

The features used for each sub-model of the two-step ensemble technique are the ones obtained for the individual sub-models due to the limitations of time. However, this ensures that the performance when the features are selected to optimize specifically for the two-step ensemble technique when would be even better, so the results show a lower bound on the optimal performance.

5.3.2 Parameter Tuning

For the two-step ensemble technique, the threshold for deciding whether or not a sample is in the 'Operating' class is tuned to optimize performance with respect to accuracy on the validation set. The resulting threshold chosen was 0.8.

For the RF classifier, the hyperparameter of the number of decision trees used is also tuned by optimizing with respect to accuracy on the validation set.

5.4 Results

Figure 2 compares the training, validation, and test errors of the different models with respect to the chosen performance metrics. We see that the two-step ensemble technique which combines a high precision model with a high accuracy model gives a higher weighted precision on the test set without sacrificing accuracy or weighted recall when compared to the other models. While the increase in performance appears to be somewhat small, they are more significant when compared within each class.

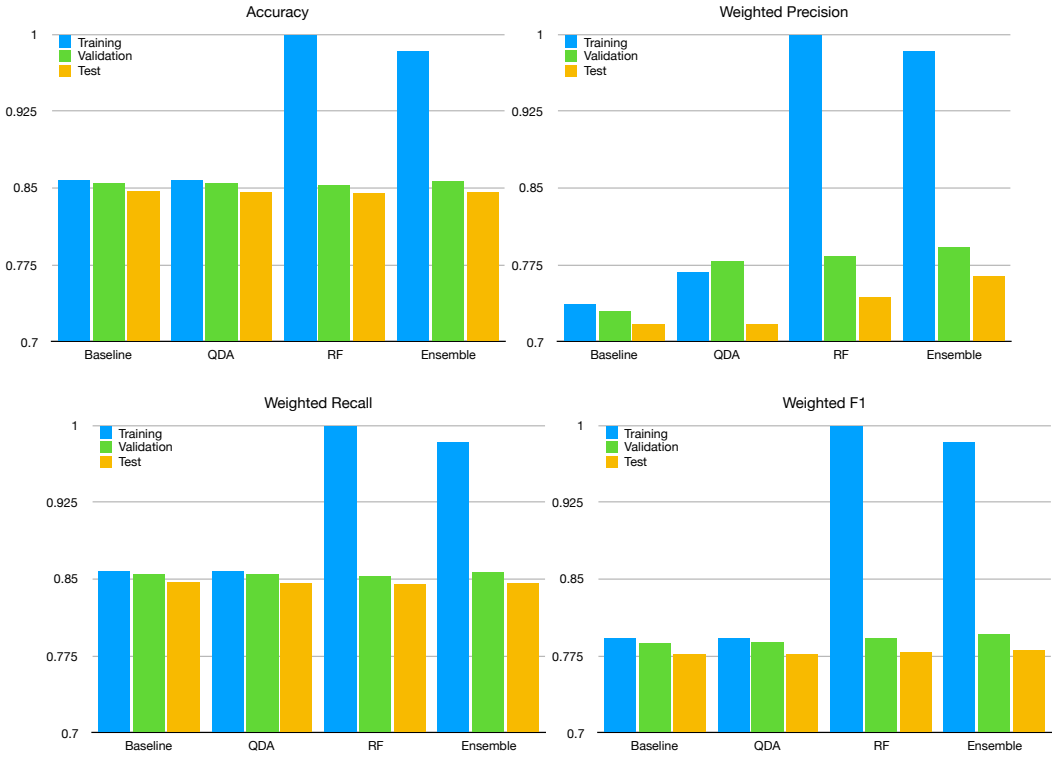

Figure 2: Comparison of performance between the different models on the training, validation, and test sets.

However, note the disporportionately high difference between training, validation and test performance on the RF model. This suggests that there is a possibility that the model was not correctly tuned (despite steps taken as outlined the parameter tuning section). There is a possibility that the two-step ensemble technique may be outperformed by a properly tuned RF model, which should be addressed in future work.

6 Conclusion and Future Work

While this novel technique seems somewhat promising, the results obtained are extremely preliminary and a lot of further work can be done. The following are suggestions of where future work can take off from.

6.1 Better Model Tuning

This two-step ensemble technique is not limited to using QDA and RF classifiers. We can explore how other models can be combined. Furthermore, RF models are high variance and dependent on the output of the QDA classifiers.

6.2 Better Comparisons to Other Techniques

While this technique seems somewhat promising when compared to using QDA and RF alone, there are many other techniques to account for biased data such as Boosting. One area of further exploration would be to see how this technique compares to many others that specifically target biased data and whether there are situations where this technique would be prefered.

6.3 Rigorous Derivation of Results

There are many points this paper makes that can be more rigorously derived. For example, further work should rigorously explain why splitting the test set for training the two models will yield better results. It would also be interesting to see if the improvement in performance is guaranteed using this technique or if there are specific criteria that would then guarantee performance improvement.

References

[1] Geopy: Geocoding library for python, 2006–2018. URL https://github.com/geopy/geopy/.

[2] Gareth James, Daniela Witten, T. H. R. T. *An Introduction to Statistical Learning*. 2013.

[3] Hejazi, M. and Singh, Y. P. One-class support vector machines approach to anomaly detection. *Applied Artificial Intelligence*, 27(5):351–366, 2013.