Can We Identify Which Companies' IPOs Will Do Well on Day One of Trading?

Ray Weng (rw525), Dhruv Girgenti (ddg35)

Introduction

An initial price offering (IPO) is the process by which a company issues shares of stock to the general public through the stock market. IPO performance can vary wildly based on the company–while some IPOs succeed greatly, like Snowflake, whose price more than doubled on the first day of trading, others can not do so well, and experience significant price drops on the first day of trading. Our project aims to determine which factors will lead to day one IPO success–and the extent of their influence–to identify which companies' IPOs are worth investing in. Such an analysis will hopefully yield better informed market decisions for newly public companies, lead to more market stability, and possibly make a good profit for those who follow buy recommendations from our models.

Dataset Description

Our dataset describes the performance of 3,762 companies with 1665 columns. It provides data for opening, closing, high and low prices, and volume for each day from opening day to day 262; these are all response variables, rather than predictive features. The dataset also provides other relevant information about each company, such as sector, industry, and country, and information about the day it IPOed, such as day of the week, month, and date. Other summary statistics were supplied, including the number of days in which a profit was realized over the course of 262 days, and the number of days that the stock performed better than SPY.

Many potentially useful features about company financials had blank/missing values, which we found by summing up all N/A values in the columns. These features could not be used because over 80% of the companies in our dataset had blank values for these columns. As a result, we were left with a limited number of features that had dense enough data to be used. In cleaning the data, we first removed all companies with fewer than 100 trades on the first day, as most of these companies had missing price data on the first day. Next, we created a new column that calculated the percentage change between closing price on day 1 and opening price on day 1. We then created another indicator variable for whether the price increased or decreased on day 1.

For the usable variables we identified (i.e. revenue, net income, sector, etc), we dropped all companies with these missing values. For other columns, like yearDifference (difference in year between company IPO year and year the company was founded), we transformed all negative-values to 0, as these were often incorrect. We actually noticed the dataset had some blatant errors in some columns. The columns for opening and lowest prices were actually swapped for all days, which we noticed by comparing them to values found on Yahoo Finance, and thus, we had to swap these columns. There were also some typos in the dataset, which we manually corrected after looking at histograms for certain features. In the end, we removed almost 48% of companies, leaving us with 1,973 examples, and 9 potentially useful predictive features. These features are as follows:

Features	Description
Year	the year the company IPOed
Day	the day of the week the company IPOed
Net income	net income for company
yearDifference	number of years between IPO year and year founded
USA Company	whether the company is US-based or not (one-hot encoding)
Sector	what sector the company is in (one-hot encoding)
companySize	company size, grouped by number of employees (one-hot encoding)
CEOAge	age of the CEO
Fiscal Month	fiscal month

Table 1: Features used in data analysis

In order to improve our understanding of our dataset, we chose to make several visualizations, illustrating various properties. Since our main interest is in determining what factors influence day 1 success, we first plotted a histogram of percent change in price on day 1 for every stock (Figure 1). The results show what we expected, which is that most stocks do not see much change in price on the first day. This makes sense because, over such a small range of time, there is very little opportunity for change, and many people may be hesitant to buy or sell a new stock. Although we are most interested in the data that falls to the left and right of this mean, these values will still help inform our analyses by offering insight into what features do not affect opening day success. We also wanted to supplement this graphic by providing a visualization to see how many stocks' prices increased on day 1 (Figure 2). From this, we see that most stocks close at a value that is equal to or less than their opening price.

Figure 1:

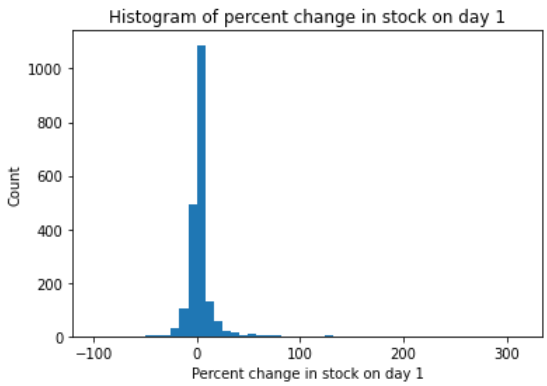
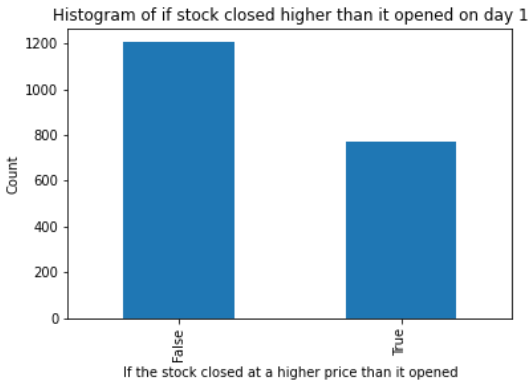


Figure 2:



We also chose to look at the distribution of sectors in our dataset (Figure 3), and where each company is located (Figure 4). In general, most companies are centered in the US, and four sectors dominate our dataset: consumer services, finance, technology, and healthcare. These together provide a better understanding of what kind of companies will play the biggest role in shaping our analyses.

Figure 3:

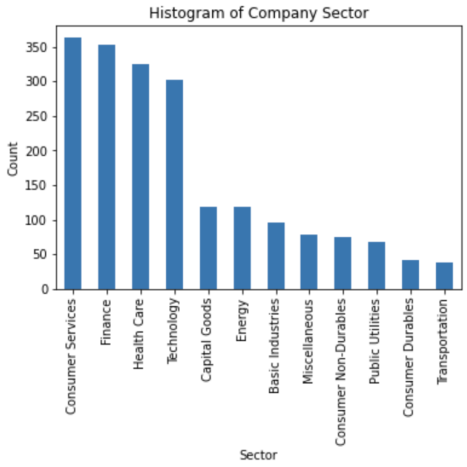
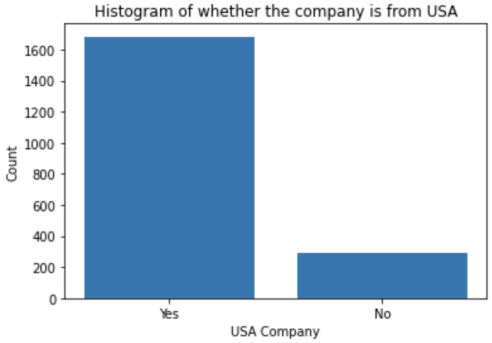


Figure 4:



Model Selection and Results

In order to solve this problem, we decided to treat it as a classification problem. Our response variable is a boolean value–True (1) or False (0)–on whether the company's stock closed at a higher price than its opening price on day 1 of trading. We decided to do classification instead of regression on stock price for two main reasons: first, that in preliminary data analysis, regression performed poorly given our dataset, and second, because classification corresponds nicely with how the prediction will be used in real life–if the prediction for a stock is 'true', then we should buy the stock at the start of the day, as we are predicting a profit if we sell at the end of the day. If the prediction is 'false', then we can pass on buying the stock.

We ultimately decided on testing three different classification models: logistic regression, random forests, and support vector machines (SVMs). These were chosen because they seemed like interesting classification models that could produce good accuracy for our use case. All three models were trained with 10-fold cross-validation, meaning we fit the model 10 times with 90% of the data in the training set and 10% of the data in the test set, to get an accurate measure of overfitting and true test error rate. The test error is then obtained by averaging the errors for these 10 iterations. Three main metrics were used to measure model performance:

1) total accuracy, which is the percentage of labels the model got right

2) accuracy on true labels, which is the percentage of 'buys' (label 1) the model got right

3) profit per stock buy, the amount of profit one would receive if one bought the stocks the model predicted would increase on the first day, averaged over the number of stocks the model recommended to buy

Metrics 2 and 3 are important because these are the actual accuracy measures for IPOs that are actionable. We care more about stocks that we actually buy than those we may have

missed. Metric 3 was calculated by summing up the difference between closing price and opening price for all companies in the test set that had a predicted label of True, and dividing by the total number of companies that had predicted label of True.

Logistic Regression

We decided to first explore the data with logistic regression. This model was chosen to serve as a baseline model, because it is commonly used for classification problems with binary categorical output. This model was trained with no normalization. The training accuracy was 0.551, and the testing accuracy was 0.550.

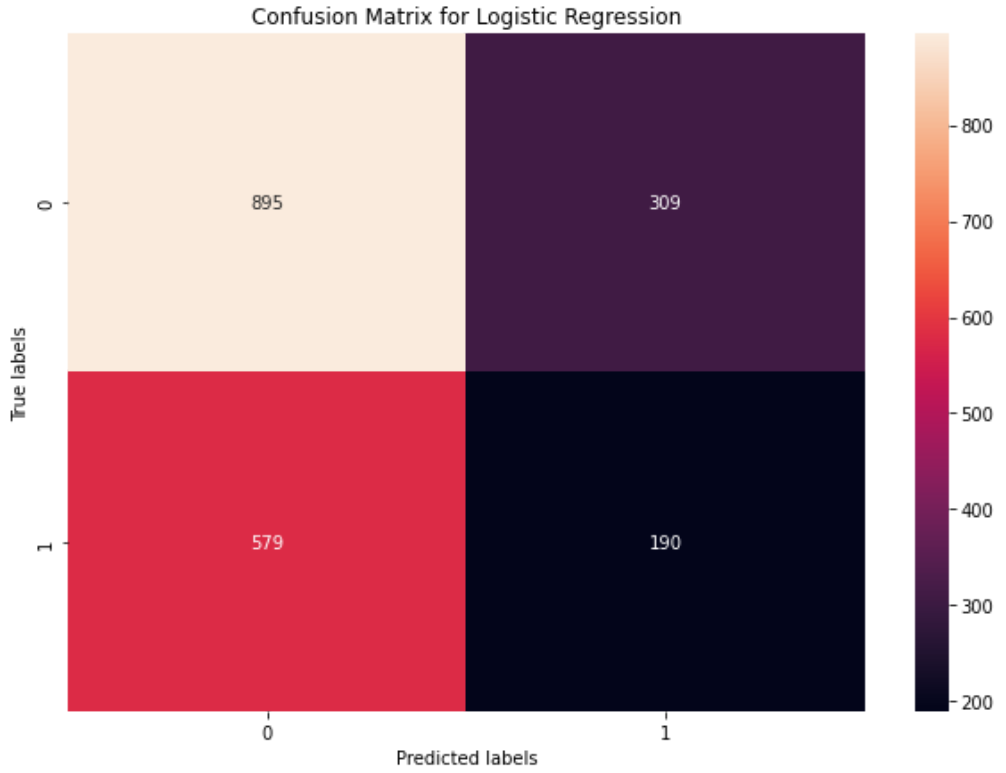


Figure 5: Confusion matrix for predictions of logistic regression (0 = False, 1 = True)

A confusion matrix was generated using the test set of all 10 iterations of the 10-fold cross-validation to summarize the predictions of the model. Looking at Figure 5, the accuracy of the model for stocks with predicted label 1 (stock would increase in price on day 1) was 190/(309+190) = 0.381. The profit per stock buy for this model was -0.02, meaning that if we followed this model, we would lose an average of 0.02 cents for each stock the model recommends. Obviously, these results were not great, so we tried two other models as well.

Random Forest Classifier

We tried the random forest classifier next. This model was trained at first with no max_depth, and n_estimators = 100. With that initial model, we noticed that it was severely overfitting, as the training accuracy was 1, while the testing error was hovering around 0.6. To deal with this overfitting, we did some hyperparameter tuning, doing a grid search cross-validation for the max_depth and n_estimators parameters. Ultimately, max_depth = 10

and n_estimators = 100 were the most optimal hyperparameters. The training accuracy for this model was 0.818, while the testing accuracy was 0.611. This model is still slightly overfitting, but decreasing the max_depth any lower would have caused underfitting, and the model would not have been able to generalize as well.

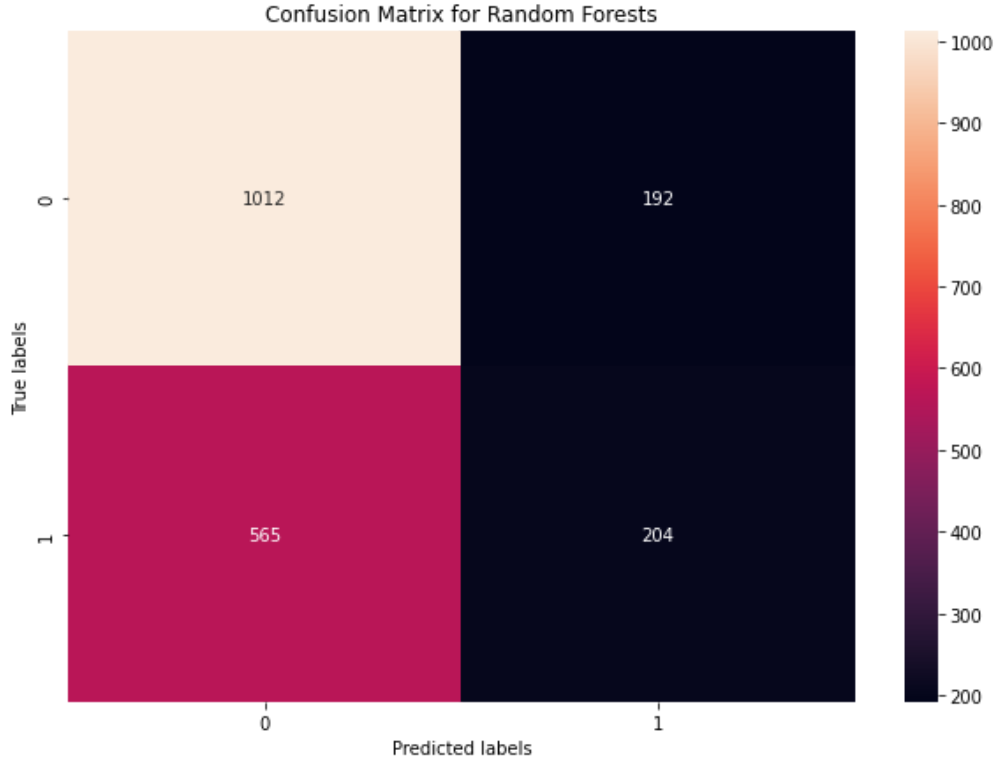


Figure 6: Confusion matrix for predictions of random forests (0 = False, 1 = True)

Looking at Figure 6, this model had an accuracy for stocks with predicted label 1 (stock would increase in price on day 1) of 204/(192+204) = 0.515. This is more promising, as it beats the baseline of 0.5 (random guessing). The average profit per stock buy recommendation was 0.33 cents.

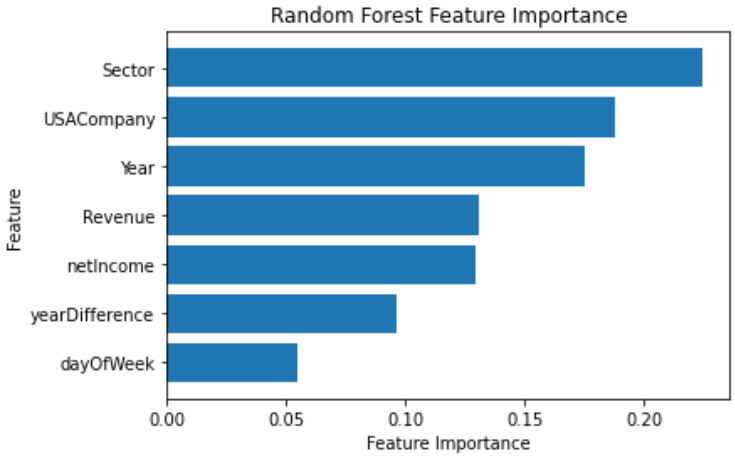


Figure 7: Feature importance according to random forests

From the random forest model, we also found which features it deemed the most important for predicting IPO success. Sector is the most important feature followed by USACompany and Year, which is surprising, as one would expect revenue and netIncome to be more important.

Support Vector Machine (SVM)

Lastly, we tried Support Vector Machines. We chose an SVM with an rbf kernel. Before training the model, we normalized the features using a standard scaler (scaling each feature to mean 0 and variance 1). Then, we did some hyperparameter tuning, again with grid search cross-validation for the parameters C and gamma. We searched through a range of 10^{-2} to 10^{10} for C and a range of 10^{-9} and 10^{3} for gamma. Ultimately, through this grid search, we landed on the parameters C = 1 and gamma = 0.01. The training accuracy for this model was 0.655, and the testing accuracy was 0.620.

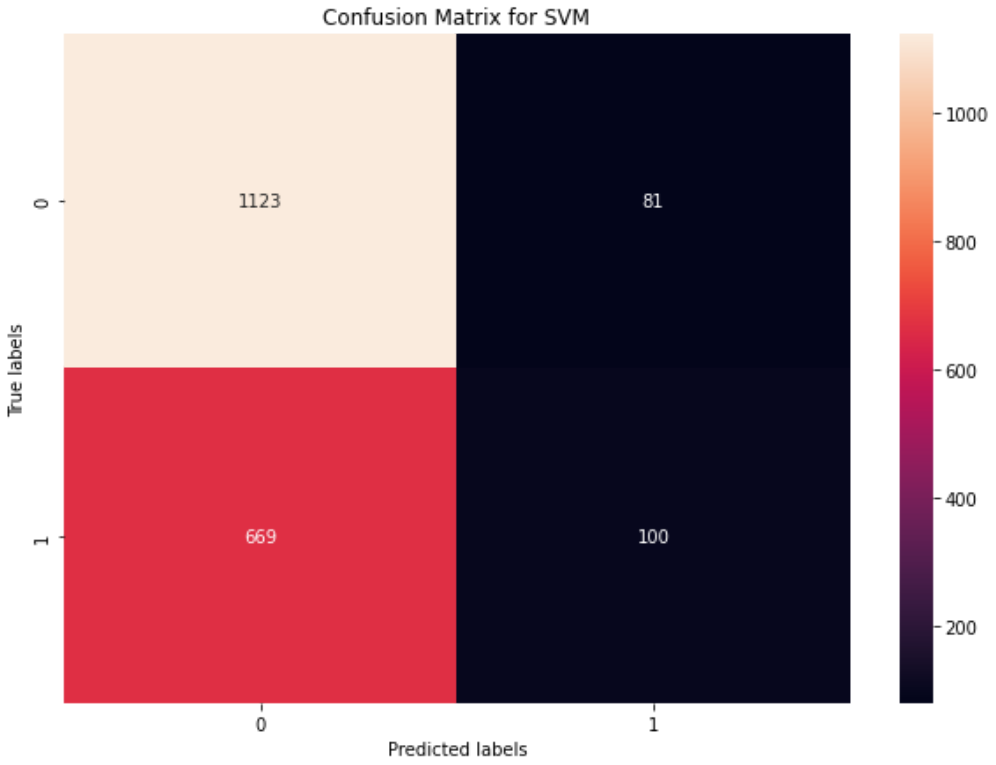


Figure 8: Confusion matrix for predictions of SVM (0 = False, 1 = True)

The accuracy of the model for IPOs labeled true was 100/(100+81) = 0.552. This model performed the best, as the average profit per stock buy was 0.74 cents.

A summary of the results is presented below:

	Logistic Regression	Random Forests	SVM w/ rbf kernel
Training Accuracy	0.551	0.818	0.655
Testing Accuracy	0.550	0.611	0.620
Testing Accuracy for Predicted Label 1	0.381	0.515	0.552
Average Profit Per Stock Recommendation	-0.02c	0.33c	0.74c

Table 2: Summary of accuracy metrics for all models

Discussion

Predicting IPO performance is inherently a hard problem–if people could accurately predict stock movements, everyone would be rich. As such, it's not unexpected that our best models performed only a little bit better than the baseline for random guessing (0.5). Looking at the results, logistic regression performed the worst. It did not have very good 'buy' signals and if we followed this model, then we would lose money on average. Random forests and SVM performed a lot better, as they had greater than 50% accuracy on buy signals, and returned a decent profit per IPO buy (0.33 cents and 0.74 cents, respectively). Looking at the confusion matrices, we can see that all models seemed to give a lot more predictions of 0 (stock will decrease on the first day) than 1 (stock will increase on the first day). This seemed to stem from both the fact that the dataset itself had a lot more IPOs that did worse on the first day than IPOs that did well, and the fact that more predictive features would have helped the models generalize better. Using the same methods described above, better accuracy could possibly be obtained if a better dataset was used. A lot of the data was missing and many of the potentially useful features about company fundamentals could not be used due to a massive amount of missing data. If the models had more features related to company fundamentals, previous funding rounds, or public sentiment before the IPO, they could have had higher accuracy. Even so, the random forests and SVM models, as is, show that it is possible to identify IPOs that will do well on day one of trading, with accuracy better than random guessing.

Since all models were trained with 10-fold cross-validation and with a decently large dataset, these models and their accuracy should be very trustworthy. I would definitely use either the random forests or SVM model in production for deciding whether or not to jump in on an IPO. These models do reasonably well in their 'buy' signals and on average, make a profit for IPOs it recommends. At the very least, I think these models could be used to supplement human analysis in evaluating IPOs.

Fairness/Weapon of Math Destruction

Prior to any analysis, we made sure to consider possible violations of fairness in both our dataset and our planned analysis. One potential consequence of our analysis regards the effect on people who will view it. On average, the people who will have access to this analysis (peers,

faculty) likely already have more accumulated wealth than the average person, in part because of better education. So, the added insight into the stock market provided by our analysis is less valuable for these people. Poorer people already have less involvement in the market, and could now fall further behind, since some richer people will use new insight to supplement, hence creating a disparate impact. However, our analysis and its results are legally fair, and have limited statistical bias, since our variables have nothing to do with race, gender, sexual orientation, etc. There are also no protected attributes to consider in our analysis. From this, we see that our dataset does not violate fairness, but our analysis possibly could.

Although there may be some overlooked factors, our analysis does not seem to be a weapon of math destruction. Since our outcomes solely measure the change in price, they are very concrete and easily measurable. This provides protection against self-fulfilling feedback loops, despite potential negative consequences on companies whose IPOs we predict to do poorly, as people may stray away from buying these stocks.

Conclusion

Overall, our analysis of this IPO dataset yielded some interesting results in regards to company IPOs. We found that the sector of the company, whether or not the company is US-based, and the year the company IPOed in are all important variables in determining whether or not an IPO does well on the first day, even more so than revenue or net income. We also found that it is possible to predict which companies will do well on the first day of trading after an IPO, with success better than random guessing. Both the random forest classifier and the SVM model had accuracies above 60%, and would have generated a positive profit if they were deployed to production. These models could be extremely useful to any investor or investment company hoping to generate a profit off of IPOs, either by investing in the IPOs the models deem likely to increase in price the first day, or just using the models to supplement manual analysis of companies.