Predicting IPO Day Success

Summary

We live in a world today where investing is not only becoming more widespread, but it is also becoming more accessible and democratized through a multitude of fintech start-ups. Almost anyone now can invest their money with a single touch on their smartphone. One of the types of investment that is on an exponential rise is IPO investing. According to Luisa Beltran from Barron's: "As of Dec. 7, 954 companies listed their shares in the U.S. this year, collecting $301 billion, Dealogic says. This is more than double the 399 companies that raised $146 billion for the same period a year earlier. This year's pace surpassed the high-water mark for IPOs, set at the dawning of the dot-com boom in 1996, when 848 companies in 1996 secured $79 billion."

This frenzy around investing in IPOs is two-fold:

1. Potential fast high returns if the company succeeds and grows quickly
2. High returns of the opening day of the IPO

While there are several external factors that impact the first bullet point, such as the state of the stock market, regulations, scandals, supply chain issues, etc… The second bullet point is primarily driven by the company's history and relevant data up until they go public. Therefore, it is possible to imagine that the success of an IPO on opening (whether the price is higher during the day than at the start of the day) is more predictable than the long-term success of the company. **The current project seeks to do just that!**

The Dataset

This IPO dataset contains the data about companies that did their initial public offering in the Indian capital market for the years 2010-2021. For each IPO, the following data is available:

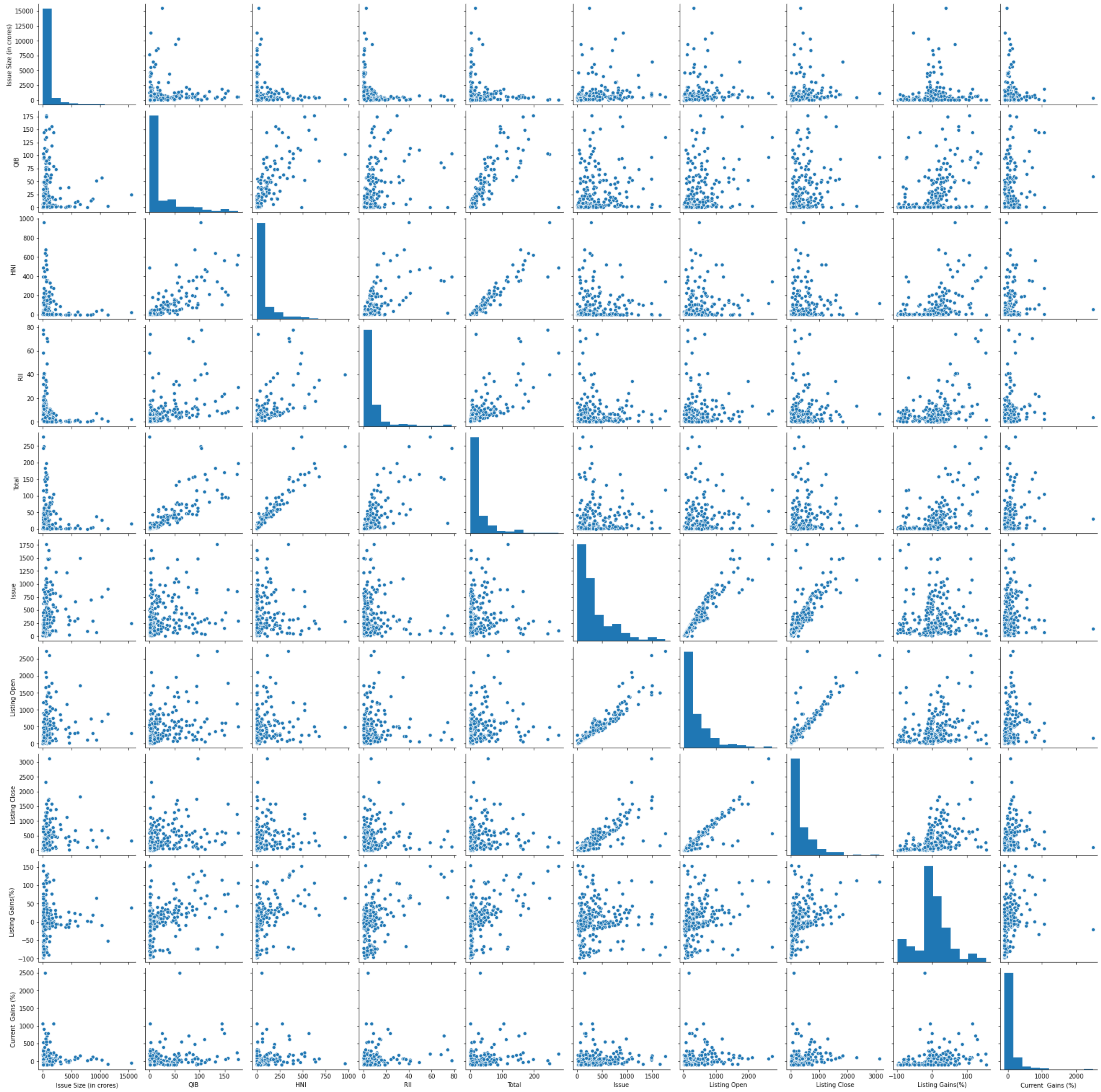
1. Issue Size (in crores) (The amount expected by company in turn selling their shares)
2. QIB (xTimes subscribed by Qualified Institutional Bidders)
3. HNI (xTimes subscribed by High Networth Individuals)
4. RII (xTimes subscribed by Retail Individual Investors)
5. Total Issue (Total subscription)
6. Listing Open (Open price on listing day in Indian Rupees)
7. Listing Close (Close price on listing day in Indian Rupees)
8. Listing Gains(%) (% change in listing price)
9. CMP (Current market price)
10. Current Gains (%) (% change in price comparing current market price to listing price)

Also, this dataset will help us to analyze the market sentiment right before the IPO listing by giving the price of the NIFTY index for the same period of time.

Link to dataset: https://www.kaggle.com/balabaskar/indian-ipo-dataset-2010-2021

Data Wrangling & EDA

Both datasets, the IPO data and the NIFTY data, were fairly well structured, and complete. No null values were noted in the entire datasets. The only correction that had to be done was to change the date columns to datetime format. Once that change was made the data was ready to be explored. Since all the fields were numerical (aside from date), an easy way to have a quick look at the correlation and distribution of the values was to use the pairplot module from the seaborn library. The following output was obtained:



Logical correlations are noted, e.g: listing open and close. Nonetheless, the most important observation is the histogram plots on the diagonal. It tells us that there are wide disparities in ranges of fields, as well as, generally skewed distributions of values. **Therefore, scaling will be important during modeling!**

Preprocessing

Before modeling, two more fields were engineered: 30-day and 60-day market trends. This was done using the NIFTY stock prices for 30 days and 60 days before each IPO listing date. An average of the day-over-day delta in NIFTY stock prices for these two time frames were calculated. To further ensure apples-to-apples comparison, each average was normalized by the mean of the time window of interest. The number obtained (for each IPO) served as 30-day and 60-day market trends.

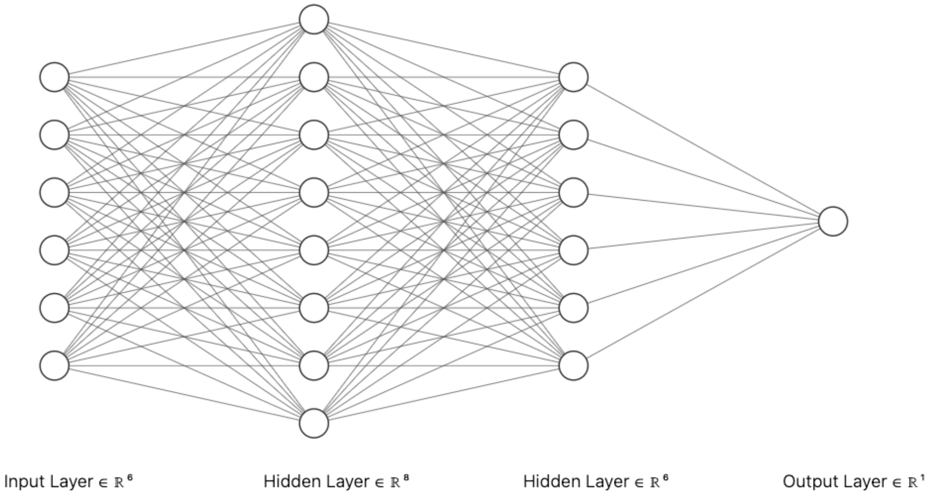
Finally, since the aim is to figure out if the IPO day will be successful or not, the “Listing Gains(%)” field was converted to a boolean value (1 if greater positive and 0 otherwise). This new engineered field was called “open_day_success”

The next steps were common neural network preprocessing steps:

1. The predictor variables were chosen:
    a. “Issue Size (in crores)”, "QIB", "RII", "Total", "Issue", "Listing Open","HNI", "m_trend_30", "m_trend_60"
2. The target variable was chosen: "open_day_success"
3. Because of the disparity in value ranges and the skewed distributions, the predictor variables were scaled using MinMaxScaler() from sklearn
4. Finally the dataset was split between training and testing, with one third reserved for testing

Since the training of a neural network can be a tedious task with large rounds of iterations, some quick modeling trials were done to see how the neural network behaved. For reproducibility, the random seed was fixed.

The chosen architecture for the neural network was as follows: 3 fully connected hidden layers. For each layer, rectified linear units was used as the activation function. To obtain the desired boolean output, a sigmoid function was applied for the output layer to get a prediction between 0-1. An sketch of the neural network architecture with an arbitrary number of nodes each layer is shown below:


Input Layer $\in \mathbb{R}^6$
Hidden Layer $\in \mathbb{R}^8$
Hidden Layer $\in \mathbb{R}^6$
Output Layer $\in \mathbb{R}^1$

The following model specifications were also utilized: loss function ='binary_crossentropy', optimizer='adam' and evaluation metric='accuracy'.

The above neural network was built using Keras and various combinations of hyperparameters, such as number of nodes in each layer, epochs and batch_size were randomly attempted to get a feel of how the model was performing. This exercise helped define the ranges of nodes, epochs and batch_size to perform grid search and cross-validation on!

Modeling

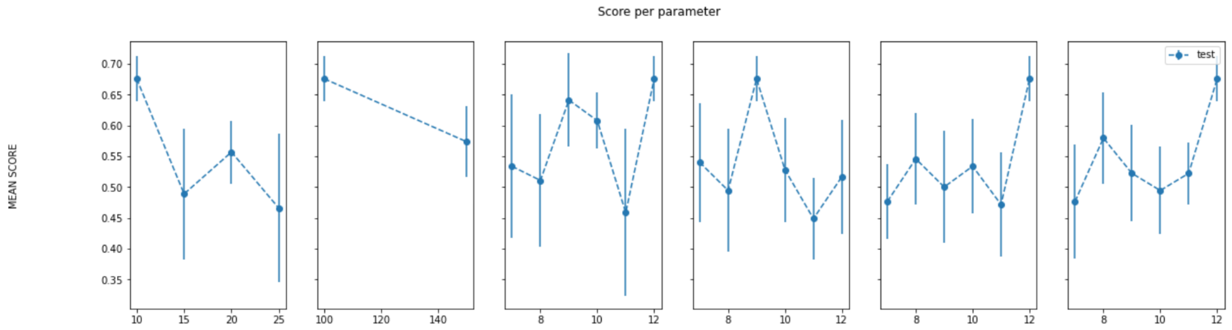
With the ranges for the hyperparameters chosen, a grid search 5-fold cross-validation using GridSearchCV in sklearn was conducted to find the best hyperparameters for this model. The chosen ranges were as follows:

1. Batch sizes: 10, 15, 20, 25
2. Number of epochs: 100, 150
3. Number of neurons in input layer: 7, 8, 9, 10, 11, 12
4. Number of neurons in 1st hidden layer: 7, 8, 9, 10, 11, 12
5. Number of neurons in 2nd hidden layer: 7, 8, 9, 10, 11, 12
6. Number of neurons in 3rd hidden layer: 7, 8, 9, 10, 11, 12

The modeling took ~12 hours to run and result in the following choice of best parameters for the neural network using accuracy as a testing metric:

- **'batch_size': 10**
- **'nb_epoch': 100**
- **'neurons_0': 12**
- **'neurons_1': 9**
- **'neurons_2': 12**
- **'neurons_3': 12**

By all other hyperparameters at their best values, and varying only one parameter at a time, we can observe the changes in performance (mean score) of each parameter as they were iterated over. This is done in the plot below:

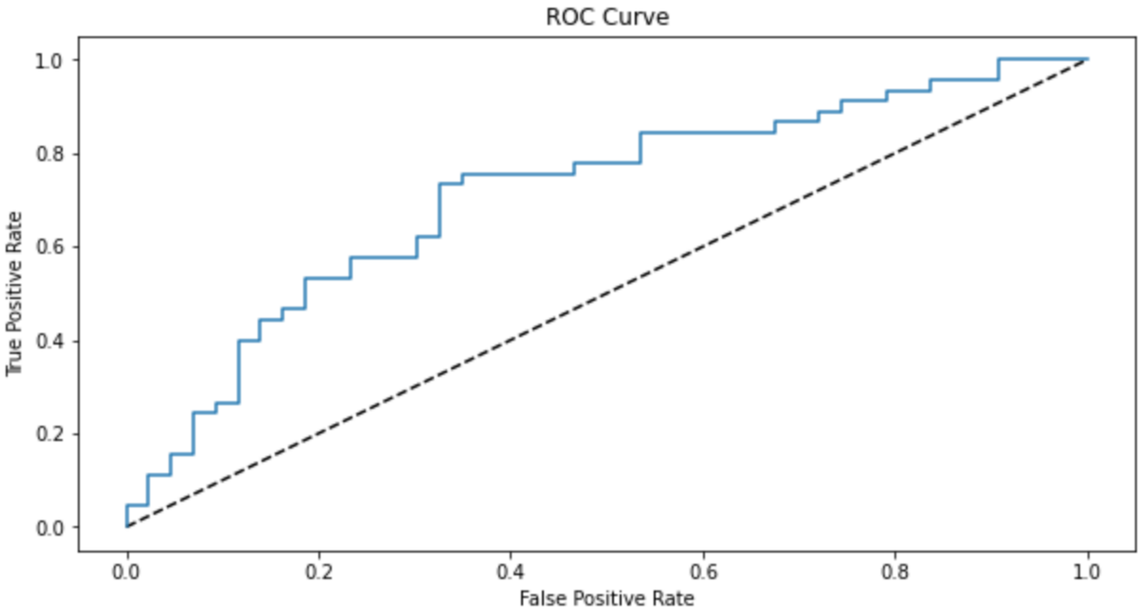


With the current architechure, increasing batch_size and epochs seemed to have decreased the performance of the NN. On the other hand, increasing the number of nodes in each hidden layers seemed to have improved the results, except for the second layer where 9 nodes was found to work best.

Results & Conclusion

The best hyperparameters obtained from the grid search and cross-validation modeling exercise were use to retrain the optimized neural network model, which was called opt_model. As is commonly used for problems with boolean outputs, the Area Under Curve (AUC) score and ROC Curve were employed as metrics for the success of this modeling effort. The confusion matrix was also studied.

On testing the model on the testing set which it has never seen before, the model performed relatively well with an AUC score greater than 0.7! This positive outcome can be visualized in the ROC curve below:

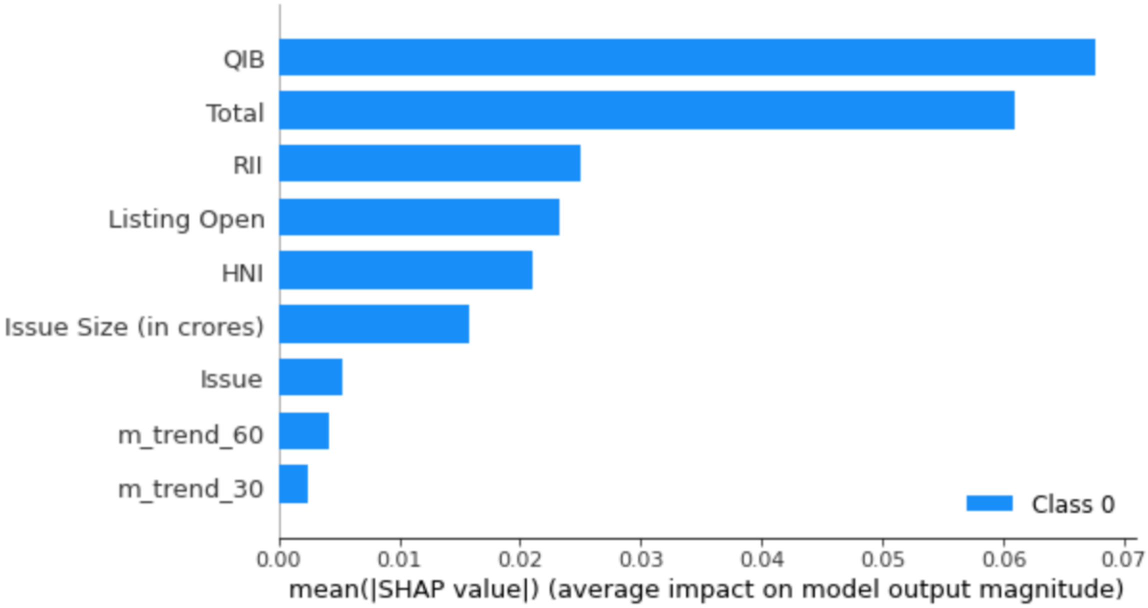


The following confusion matrix was obtained:

	Precision	Recall	F1-score	Support
0	0.63	0.67	0.65	43
1	0.67	0.62	0.64	45
Accuracy			0.65	88
Macro Avg	0.65	0.65	0.65	88
Weighted Avg	0.65	0.65	0.65	88

The model looks well balanced between precision and recall, which shows that the model will not be biased and will perform agnostically on new data. The overall is only 0.65, but given the small training set, and limited computational power, these results are deemed satisfactory.

A further investigation is conducted to understand what are the driver factors in this neural network. The SHAP library was used to this end and the importance of each feature in the optimized trained model was calculated. The output of the SHAP library is a simple bar chart showcasing the relative impact of each field on the model:

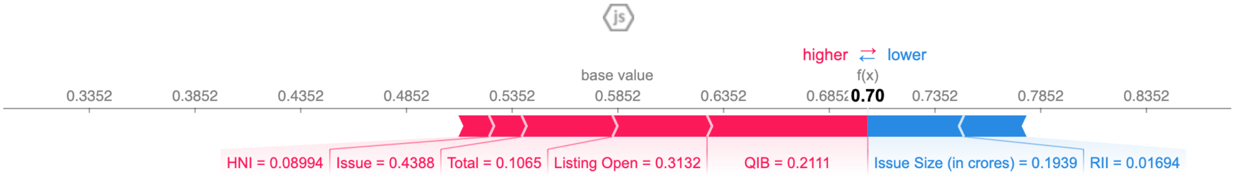


By checking the feature importance, we find that the following two features are the dominating factors in predicting whether a IPO will be successful on its open day:

1. QIB: Qualified institutional buyer
2. Total: Total subscription from all categories

Both of these categories make sense to be drivers of IPO success. The presence of QIBs is favorable to the success of an IPO because it builds confidence in the company going public. Similarly, the total amount of subscription towards the IPO has a similar effect. **Interestingly, both market trends have the least impact on whether an IPO will be successful on its open day!**

We double clicked on a specific testing data point to see how the different features influenced the final prediction for that data point using SHAP's force plot:



The randomly chosen example shows most factors contributed to pushing the prediction towards 1, while two factors opposed that prediction. The final prediction was a probability of 0.7 which therefore translated into success on IPO day!