Machine Learning vs Initial Public Offerings

AUTHORS: AMANDEEP KAUR, NITIN THOMAS, PRABHDYAL SINGH, RABIH RAYMOND ANTOUN, MATTHEW PHAN

Your best quote that reflects your approach… “It’s one small step for man, one giant leap for mankind.”

- NEIL ARMSTRONG

Project Goals

To use various machine learning methods in order to predict the returns on initial public offerings after 100 days to see if they are profitable to trade or invest in or not.

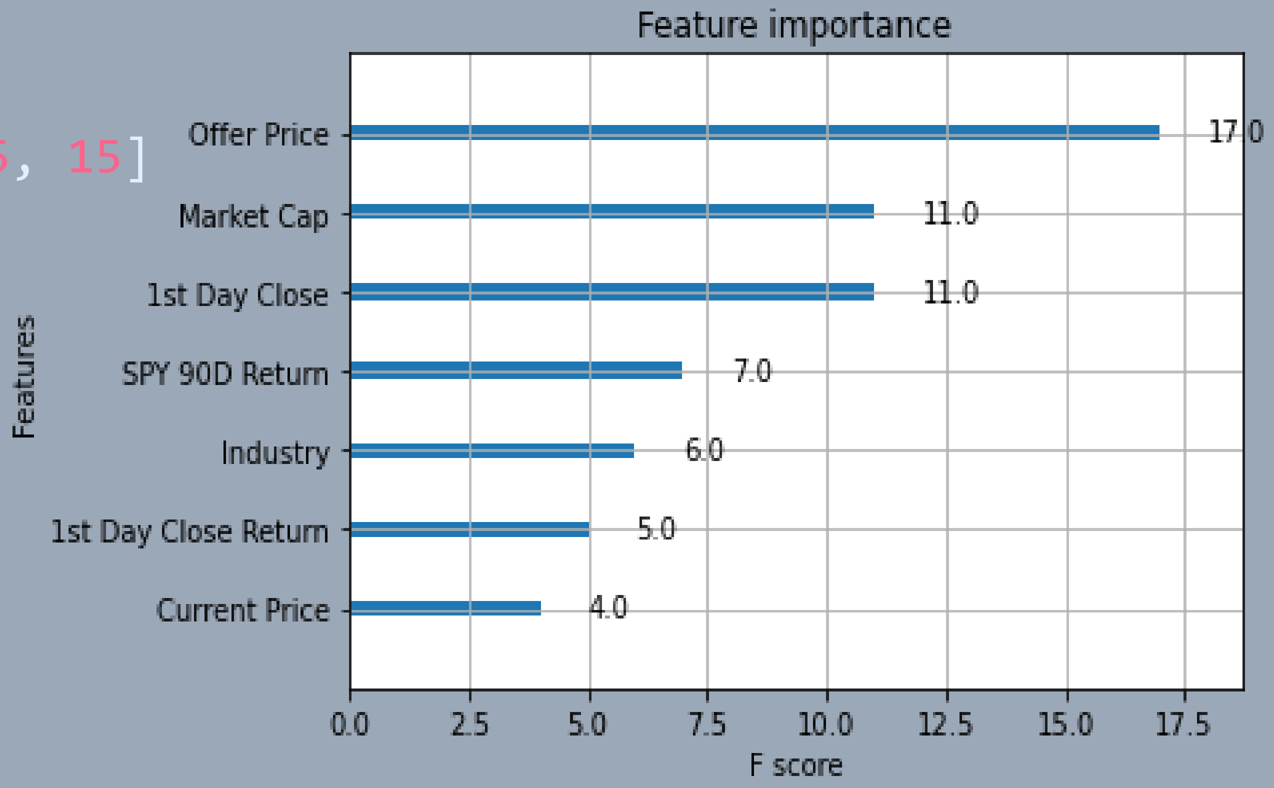
Different deep learning methods will provide a the option for a better choice for performance.

Approach

- Information started with gathering data from iposcoop.com and this was manually done into a csv file which was then imported into a pandas dataframe.
- Data cleaning involved removing the ttm column, making seperate dataframes of 3 different time points, and encoding the 100 day return target column into non negative values for processing.
- The machine learning libraries used were Adaboost, XGBoost, random forest, and logistic regression.
- Add more features to see if accuracy is better (market cap, 1st day close)

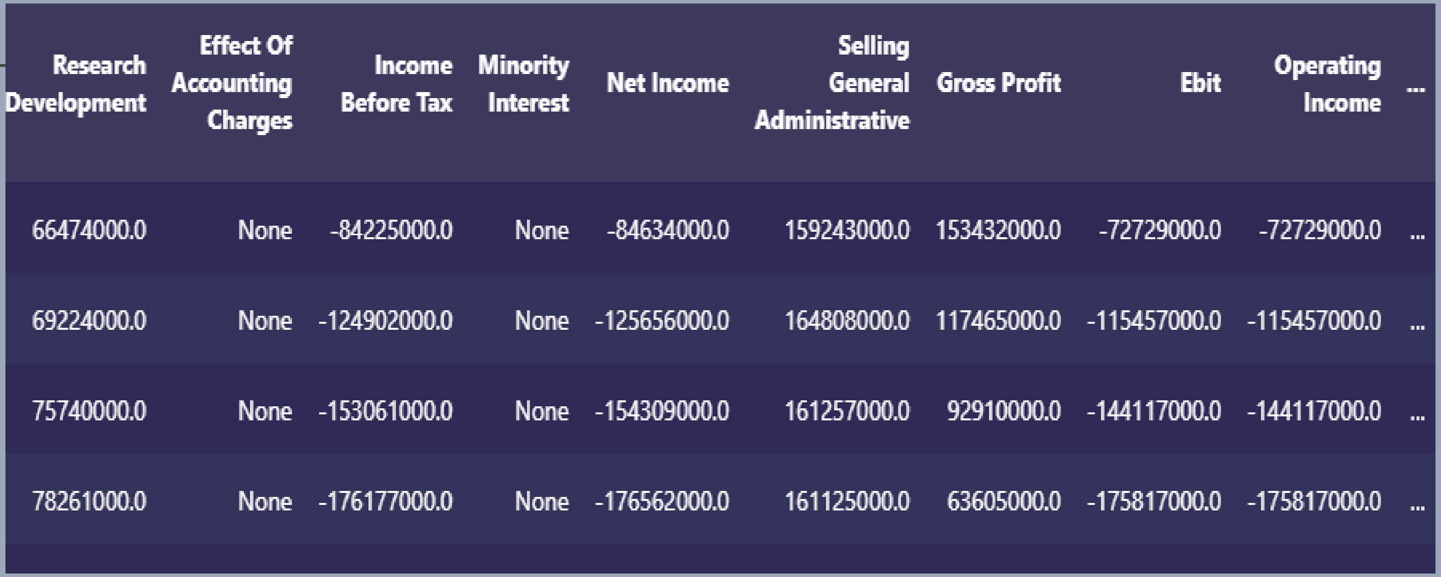
Code

```
# Visualizing the feature importance result as a bar graph

xgb.plot_importance(xg_reg)

plt.rcParams['figure.figsize'] = [15, 15]

plt.show()
```



Yahoo Data Scraper for more features

```
dfs = [] # list for each ticker's dataframe
for ticker in tickers:
    # get each financial statement
    pnl = ticker.financials
    bs = ticker.balancesheet
    cf = ticker.cashflow
    # concatenate into one dataframe
    fs = pd.concat([pnl, bs, cf])
    # make dataframe format nicer
    # Swap dates and columns
    data = fs.T
    # reset index (date) into a column
    data = data.reset_index()
    # Rename old index from '' to Date
    data.columns = ['Date', *data.columns[1:]]
   # Add ticker to dataframe
    data['Ticker'] = ticker.ticker
    dfs.append(data)
data.iloc[:,:3]# for display purposes
```



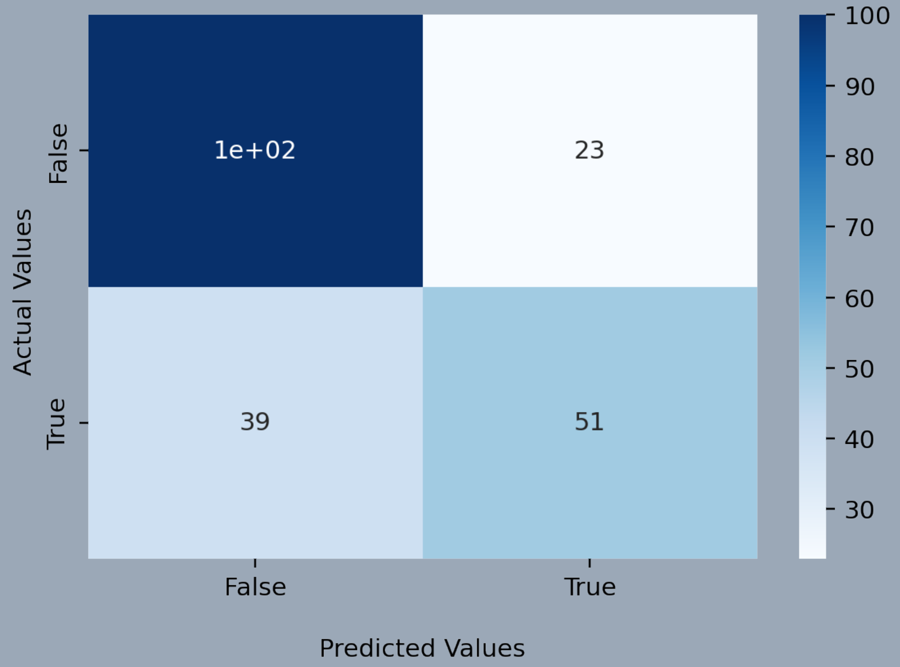
Research Development	Effect Of Accounting Charges	Income Before Tax	Minority Interest	Net Income	Selling General Administrative	Gross Profit	Ebit	Operating Income	...
66474000.0	None	-84225000.0	None	-84634000.0	159243000.0	153432000.0	-72729000.0	-72729000.0	...
69224000.0	None	-124902000.0	None	-125656000.0	164808000.0	117465000.0	-115457000.0	-115457000.0	...
75740000.0	None	-153061000.0	None	-154309000.0	161257000.0	92910000.0	-144117000.0	-144117000.0	...
78261000.0	None	-176177000.0	None	-176562000.0	161125000.0	63605000.0	-175817000.0	-175817000.0	...

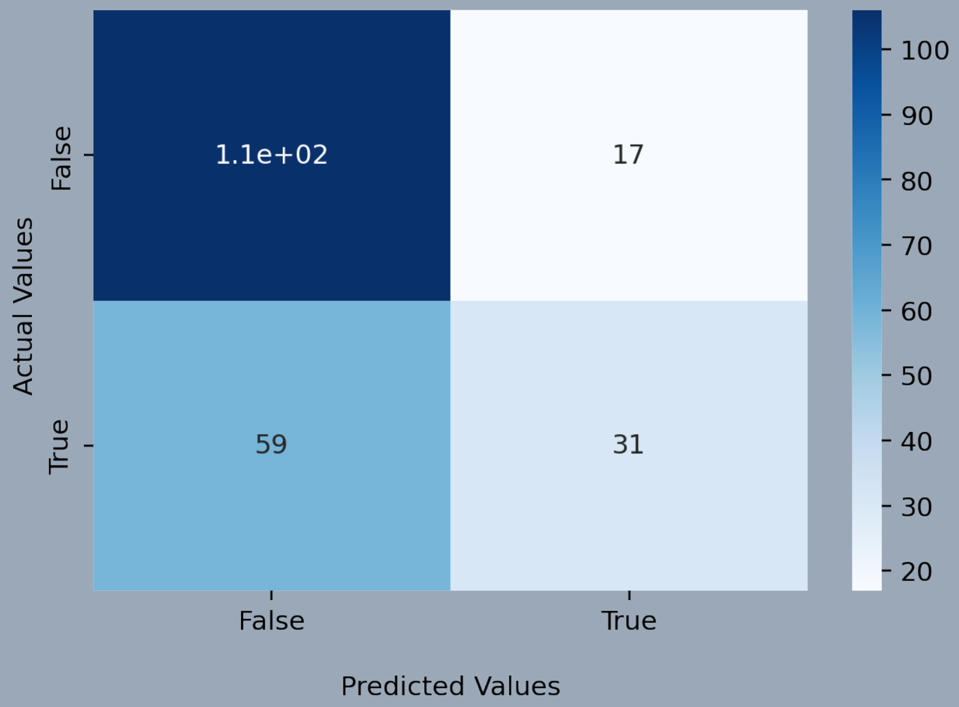
Difficulties

Dropping company symbol and company name, and use the encoded industry as a number, because there was problems reading the string values.

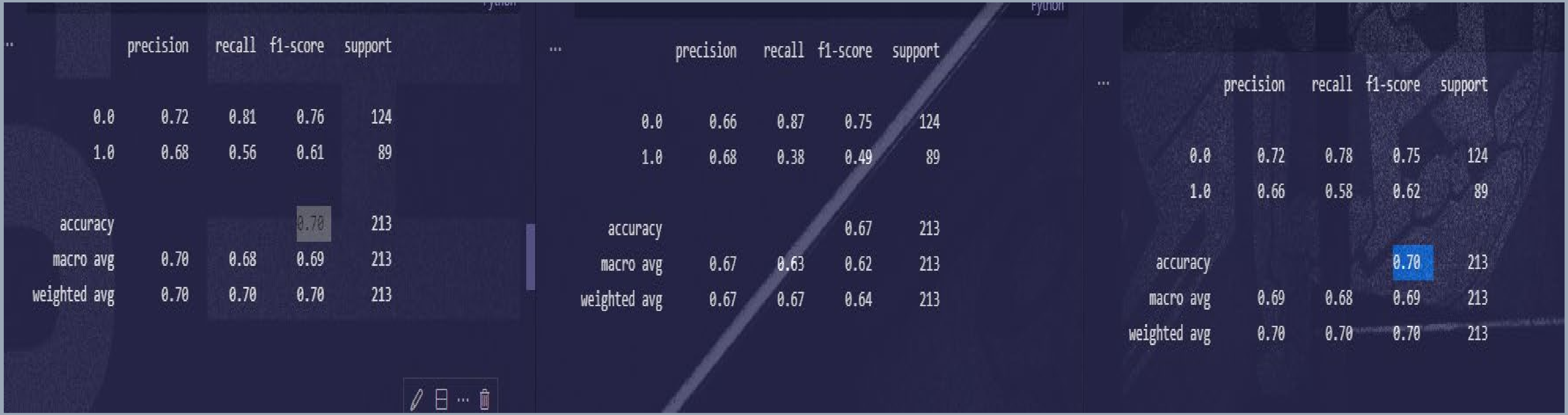
- Problems with data gathering, getting data older than 2018 and having it to work with the code was problematic, the old csv with dates all the way back to 2000 file wasn't computing properly with the notebook.

Outcome


AdaBoost Seaborn Confusion Matrix with labels


Logistic Regression Seaborn Confusion Matrix with labels

Adaboost----|----LogisticRegression----|----XG Boost



```
              precision    recall  f1-score   support

         0.0       0.72      0.81      0.76       124
         1.0       0.68      0.56      0.61        89

    accuracy                           0.70       213
   macro avg       0.70      0.68      0.69       213
weighted avg       0.70      0.70      0.70       213
```

```
              precision    recall  f1-score   support

         0.0       0.66      0.87      0.75       124
         1.0       0.68      0.38      0.49        89

    accuracy                           0.67       213
   macro avg       0.67      0.63      0.62       213
weighted avg       0.67      0.67      0.64       213
```

```
              precision    recall  f1-score   support

         0.0       0.72      0.78      0.75       124
         1.0       0.66      0.58      0.62        89

    accuracy                           0.70       213
   macro avg       0.69      0.68      0.69       213
weighted avg       0.70      0.70      0.70       213
```

Conclusion

It appears that XG Boost and Ada boost have the highest accuracy score and the offer price, market cap, and 1st day close are the top 3 feature importance

Accuracy scores of 0.7 were found

Next Steps

Get more data from years predating 2018 and covid

More machine learning models for comparison
(LSTM, Support Vector)

Paid subscription info/data

Additional Questions

Get more data that would particularly help our models for IPO’s

(R&D , working capital, growth rate, dividends, reinvestment rate discount model)