Predicting an Increase or Decrease in a Company's Stock

DS5110: Big Data Analytics (Fall 2021) - Final Project Report

TJ McIntyre, Connor Killion, Anh Nguyen

Abstract

The face value metrics of a stock usually include statistics such as high/low, open/close price, market capitalization, and a few others, but making a decision about a stock does and should require much more in-depth research into the other underlying metrics related to the stock. The stock market as a whole is one of the hardest things to predict accurately on a consistent basis, not to mention predicting a day over day price change. With that being said, we thought we would examine the possibility of predicting if a stock's price would increase or decrease in a year compared to its price at the beginning of the year. For this project we obtained a combination of in-depth finance data from a Kaggle dataset. In addition, we did not just look solely into those aforementioned metrics, we also looked at the financial statement information of the companies whose stocks we were attempting to predict; as well as the metrics based on industry. We were able to obtain complete data for 2014 and 2015, so we wanted to try to predict a raise/fall of a stock year-over-year in 2015 after training models on the 2014 data. We experimented with logistic regression, gradient boosting, and random forest algorithms. We found that the random forest model performed the best, particularly in the energy industry. For our best models, the testing AUROC was around 0.65 with varying levels of precision. This came with overfitting in some cases, but since it is very hard to predict the stock market, we conclude that for the energy industry in this time period that this could be an investing approach.

Data & Models

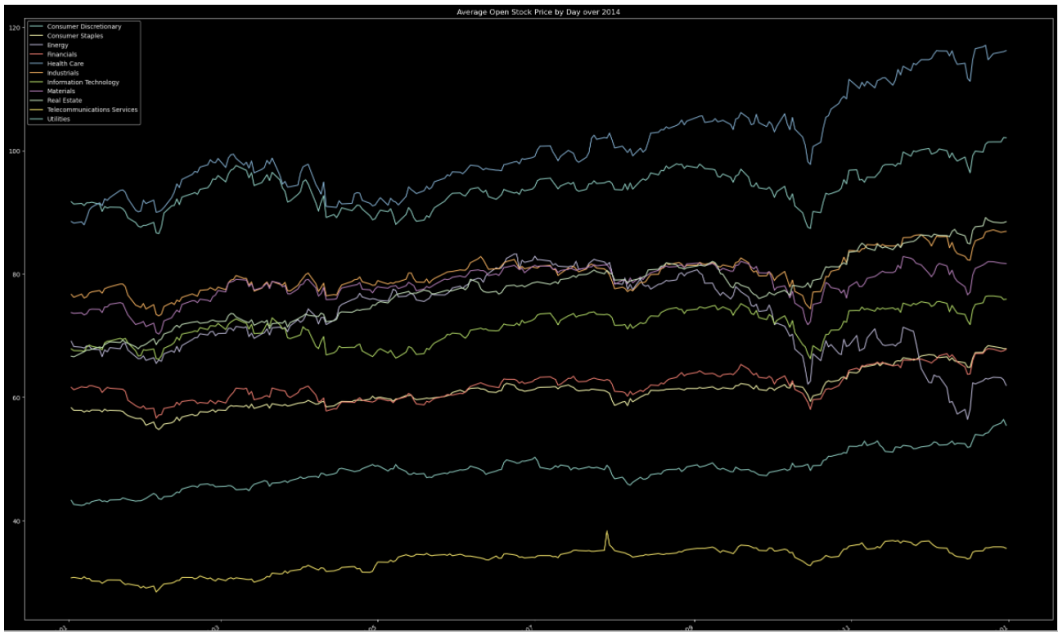
Our data is from a collection of Kaggle datasets. The data is in a few different files, but the main files we were concerned with were the price-adjusted-split, securities, and fundamentals files. We created the 'label' column into a binary variable which will indicate if the delta for the year is positive or negative. For example:

symbol	yearDelta	label	GICS Sector
ALXN	5.720000999999996	1	Health Care
GIS	4.329997999999996	1	Consumer Staples
K	6.829994999999997	1	Consumer Staples
LEN	4.099998999999997	1	Consumer Discreti...
SPGI	9.599998999999997	1	Financials

As you can see, ‘1’ will indicate the closing price for the year is greater than the beginning price. The data itself consists of over 70 numeric variables, so naturally we wanted to reduce this down to only the most important features. So, just as we did in class, we decided to compute the univariate AUC scores and only keep the metrics that had a value over 0.55. This left us with a major reduction in variables. The variables are: 'Non-Recurring Items', 'Liabilities', 'Goodwill', 'Deferred Liability Charges', 'Other Investing Activities', 'Total Current Assets', "Add'l income/expense items", 'Accounts Receivable', 'Other Current Assets', 'Total Current Liabilities', 'Net Borrowings', 'Interest Expense', 'Depreciation', 'Operating Margin', 'Sale and Purchase of Stock', 'Equity Earnings/Loss Unconsolidated Subsidiary', 'Intangible Assets', 'Fixed Assets', 'Deferred Asset Charges'. A sampling of some of these feature results is shown here:

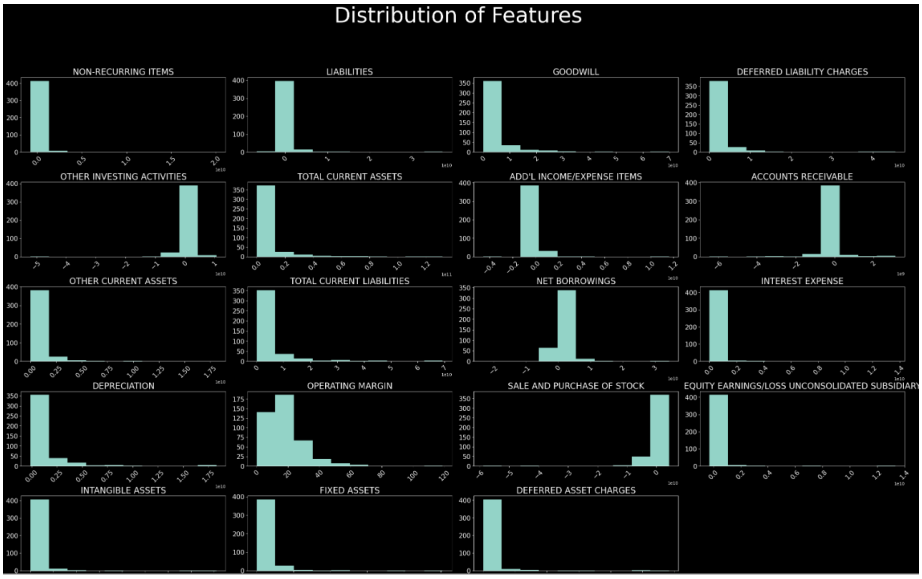
	Variable	AUROC	Weight
43	Non-Recurring Items	0.625042	-1.193668e-09
28	Liabilities	0.609423	6.856260e-12
20	Goodwill	0.608489	4.896798e-11
13	Deferred Liability Charges	0.607470	-3.701060e-11
51	Other Investing Activities	0.598132	-1.477484e-10
66	Total Current Assets	0.594737	-2.609071e-11
2	Add'l income/expense items	0.594312	-6.647816e-11
1	Accounts Receivable	0.590577	-2.041494e-10
47	Other Current Assets	0.583871	-6.303748e-11
67	Total Current Liabilities	0.577759	-3.053324e-11
33	Net Borrowings	0.576825	-1.806148e-11
25	Interest Expense	0.576231	5.301524e-10
14	Depreciation	0.574109	-1.292480e-10
45	Operating Margin	0.569864	1.293050e-02
61	Sale and Purchase of Stock	0.564261	1.259689e-10
18	Equity Earnings/Loss Unconsolidated Subsidiary	0.558744	-9.529001e-10
24	Intangible Assets	0.558574	1.690797e-10
19	Fixed Assets	0.557216	-9.909018e-12
12	Deferred Asset Charges	0.554669	-4.475488e-11

With our data assembled, we split the data between 2014 and 2015, which is our train and test set. First we wanted to visualize our response variable more. To do this we took the average open price of each ticker grouped by industry, and plotted them over the course of 2014.



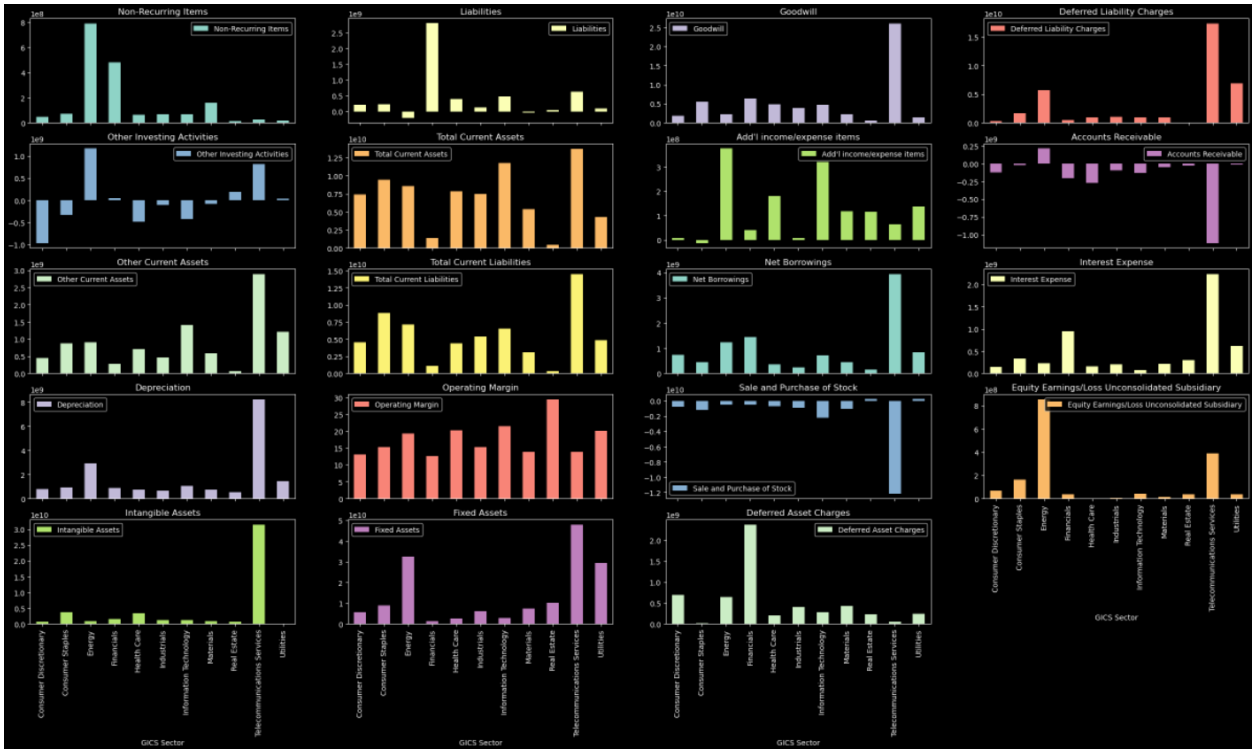
In the plot above, we can see that almost every industry had a positive trend in 2014. However, there is one industry that began dropping in Q3 of 2014 and continued this descending trend to the end of the year. This was the Energy industry, which is the light purple line on the plot.

As mentioned above, we only retained the numeric variables which had an univariate AUROC greater than 0.55. The following plots depict histograms of each of these 19 variables.



In the histograms above, we can clearly see that the distributions are skewed for most variables and several features have different scales. Thus, normalization of the numeric variables may be useful to look at.

The one categorical variable that we are interested in is “GICS Sector”, which tells us which industry the company is in. To better understand if the numeric variables differ across industries we created the plots below.



In the above plot, we can see that many of the variables have different outlooks in each industry and some variables contain both positive and negative numbers. Below is the basic summary report of all numeric variables in our dataset. The table shows the count, mean, standard deviation, minimum and maximum for each variable.

summary	count	mean	stddev	min	max
Non-Recurring Items	423	1.64451286052009 46E8	1.0817359403532982E9	-7.404E8	2.0058E10
Liabilities	423	4.8418768085106385E8	2.8837175937499924E9	-6.639E9	3.7103E10
Goodwill	423	3.8619079976359 34E9	7.678030390697382E9	0.0	6.9777E10
Deferred Liability Charges	423	1.69769515366430 26E9	4.76045068704802E9	0.0	4.5469E10
Other Investing Activities	423	-2.0524769267139 48E8	2.9749717417969 16E9	-5.2009E10	1.0578E10
Total Current Assets	423	6.9981514799054 37E9	1.3595987905503649E10	0.0	1.31839E11
Add'l income/expense items	423	1.1832659338061465E8	8.975300501353017E8	-4.577E9	1.1613E10
Accounts Receivable	423	-1.0975179196217494E8	6.1205299021330 34E8	-6.452E9	3.118E9
Other Current Assets	423	7.0864796690307 33E8	1.642181139410677E9	0.0	1.8096E10
Total Current Liabilities	423	4.8271032080378 25E9	9.10778467207754E9	0.0	6.9345E10
Net Borrowings	423	7.0652724822695 04E8	3.3415624885616846E9	-2.326E10	3.436E10
Interest Expense	423	3.1790018439716 31E8	9.483690718949759E8	0.0	1.369E10
Depreciation	423	1.0928565697399528E9	2.0964414359459646E9	-1.05E8	1.8273E10
Operating Margin	423	17.27659574468085	12.3573028754315	0.0	119.0
Sale and Purchase of Stock	423	-9.8844640189125 3E8	3.9431125785801315E9	-5.8852E10	4.282E9
Equity Earnings/Loss Unconsolidated Subsidiary	423	1.0882023167848 7E8	7.7842757143910 72E8	-2.9E8	1.3323E10
Intangible Assets	423	1.8307927777777777E9	5.991173353304622E9	0.0	8.1069E10
Fixed Assets	423	8.9551304775413 7E9	2.0115245649725044E10	0.0	2.52668E11
Deferred Asset Charges	423	6.0315188652482 27E8	2.6177297165452695E9	0.0	2.9166E10

Our EDA discovered that normalization may help and that the industry may play a key role. Thus, each of our models will be fit using the normalized data and standard data and then compared to see which is a more viable option.

We constructed a pipeline to prepare the data for all of our models on which the 2014 data was fit, trained, and assessed until we found our best model to make predictions on the 2015 data. The schema for our data ended up as seen below:

```
root
 |-- label: double (nullable = false)
 |-- features: vector (nullable = true)
 |-- features_norm: vector (nullable = true)
 |-- Non-Recurring Items: double (nullable = true)
 |-- Liabilities: double (nullable = true)
 |-- Goodwill: double (nullable = true)
 |-- Deferred Liability Charges: double (nullable = true)
 |-- Other Investing Activities: double (nullable = true)
 |-- Total Current Assets: double (nullable = true)
 |-- Add'l income/expense items: double (nullable = true)
 |-- Accounts Receivable: double (nullable = true)
 |-- Other Current Assets: double (nullable = true)
 |-- Total Current Liabilities: double (nullable = true)
 |-- Net Borrowings: double (nullable = true)
 |-- Interest Expense: double (nullable = true)
 |-- Depreciation: double (nullable = true)
 |-- Operating Margin: double (nullable = true)
 |-- Sale and Purchase of Stock: double (nullable = true)
 |-- Equity Earnings/Loss Unconsolidated Subsidiary: double (nullable = true)
 |-- Intangible Assets: double (nullable = true)
 |-- Fixed Assets: double (nullable = true)
 |-- Deferred Asset Charges: double (nullable = true)
 |-- GICS Sector: string (nullable = true)
```

Logistic Regression

The first model we tried was a simple logistic regression model. We split up the 2014 data using a 60/40 train/test split. Then we trained the model on the training part of the

data and tested it on the testing section of 2014 data. This test was done without the data being normalized. We then did a second logistic regression test in the same manner, but this time the features were normalized.

Random Forest

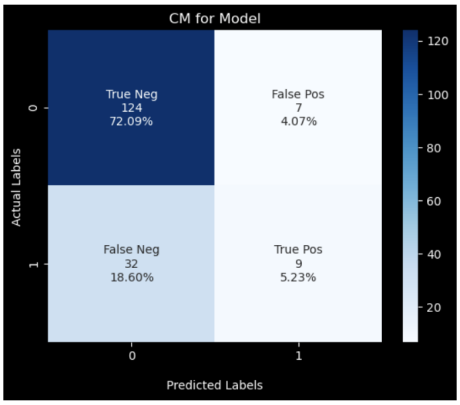
Random forest was the next model we tried, again beginning with the unnormalized data training and testing with the split 2014 data (60/40 train-test split). Next we fit a random forest on the normalized data. We found the normalized data with random forest to have strong AUC and the highest precision, but not necessarily the best model as we will introduce cross-validation and hyper-parameter tuning later in the paper.

Gradient Boosting

Our final test models were gradient boosting models in which we ran the test with the unnormalized and then normalized data. We used the GBTClassifier in pyspark. Both of these models performed worse than any of the other models we trained, thus we felt confident with not using this model to test on the 2015 data.

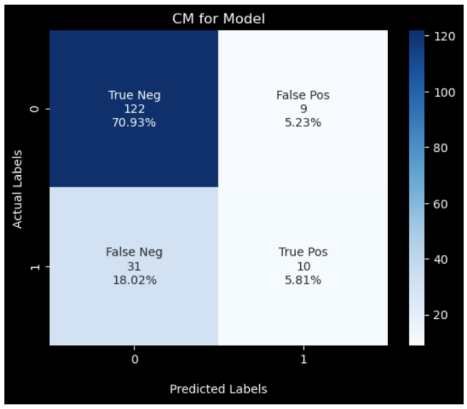
Results

Each of the models we evaluated primarily used metrics stemming from confusion matrices. In our preliminary analysis of the trained models, we used AUROC and Precision scores as the primary evaluation metrics. AUROC is consistently used in classification settings and precision was essential to our goal of choosing profitable stocks. Our initial benchmark models trained using a 60/40 train-test split can be seen below. We started with logistic regression on the standard data where we got an AUC of 0.7187 and a Precision of 0.5625.



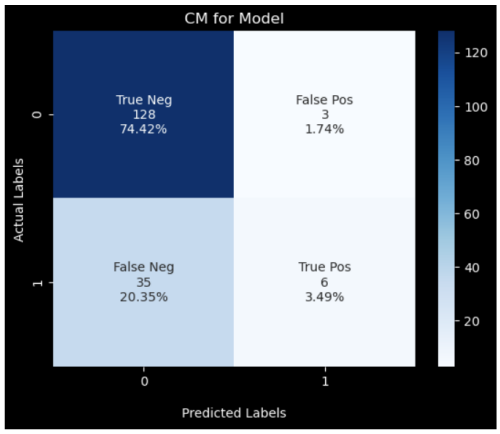
	Statistic	Measure
0	AUC	0.7187
1	Accuracy	0.7733
2	Precision	0.5625
3	Recall	0.2195
4	F1	0.3158

The normalized logistic regression model had an AUC of 0.7170 and precision of 0.5263.



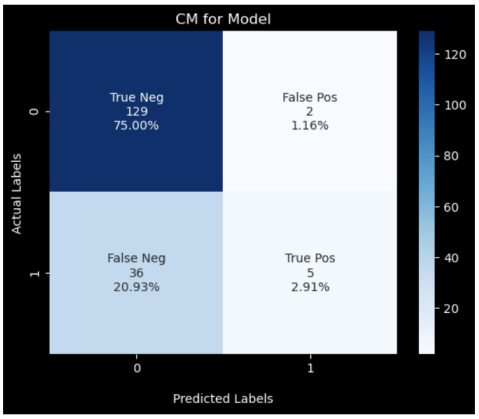
	Statistic	Measure
0	AUC	0.7170
1	Accuracy	0.7674
2	Precision	0.5263
3	Recall	0.2439
4	F1	0.3333

The random forest with standard data had an AUC of 0.7276 and a precision of 0.6667.



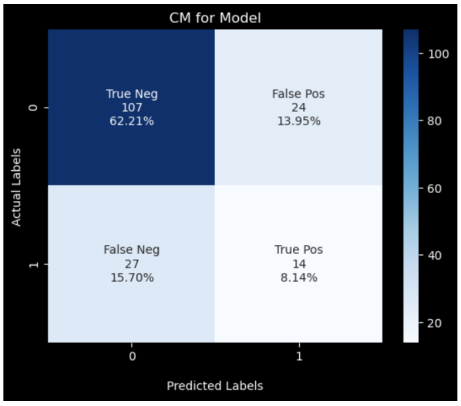
	Statistic	Measure
0	AUC	0.7276
1	Accuracy	0.7791
2	Precision	0.6667
3	Recall	0.1463
4	F1	0.2400

The random forest with normalized data yielded an AUC of 0.7213 and a precision of 0.7143.



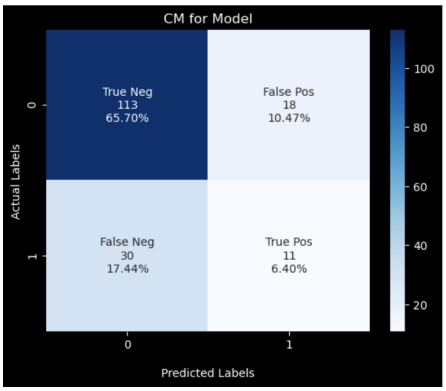
	Statistic	Measure
0	AUC	0.7213
1	Accuracy	0.7791
2	Precision	0.7143
3	Recall	0.1220
4	F1	0.2083

The next model was gradient boosting on standard data, which had an AUC of 0.6414 and a precision of 0.3684.



	Statistic	Measure
0	AUC	0.6414
1	Accuracy	0.7035
2	Precision	0.3684
3	Recall	0.3415
4	F1	0.3544

Finally, gradient boosting with normalized data yielded an AUC of 0.6745 and precision of 0.3793.

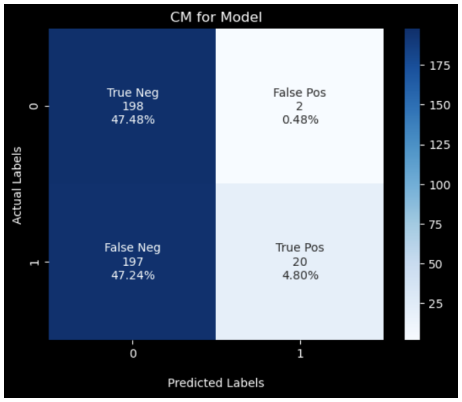


	Statistic	Measure
0	AUC	0.6745
1	Accuracy	0.7209
2	Precision	0.3793
3	Recall	0.2683
4	F1	0.3143

A summary table of the six trained models above, which were fit using a 60/40 train-test split on the in time data can be seen below. Using the output from the table below we will decide some models to test on the out of time 2015 testing dataset.

Model	AUROC	Precision
Logistic	0.7187	0.5625
Logistic (normalized)	0.7170	0.5263
Random Forest	0.7276	0.6667
Random Forest (normalized)	0.7213	0.7143
Gradient Boosted	0.6414	0.3684
Gradient Boosted (normalized)	0.6745	0.3793

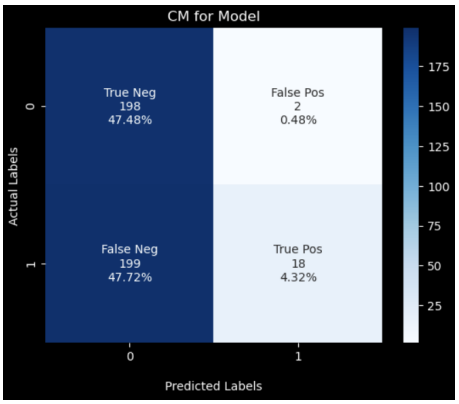
As you can see from our preliminary assessment, the normalized random forest model had high AUC and the highest precision, so we decided to see how it performed on the 2015 data.



	Statistic	Measure
0	AUC	0.6577
1	Accuracy	0.5228
2	Precision	0.9091
3	Recall	0.0922
4	F1	0.1674

Looking at the above results, the AUC came out to be 0.6577, telling us that the model may be slightly overfit since AUC decreased significantly. Nevertheless, we had previously stated that the stock market is extremely difficult to predict and the model performed better than 50/50 random guessing (AUC > .50).

Next, we tested the standard random forest model as it was the second best performing model on the in training dataset.



	Statistic	Measure
0	AUC	0.6561
1	Accuracy	0.5180
2	Precision	0.9000
3	Recall	0.0829
4	F1	0.1519

The random forest with the standard features had a testing AUC of 0.6561 and precision of 0.90. Since the normalized random forest had slightly higher AUC and Precision, we decided to use 5-fold cross validation and the following hyperparameter grid on the random forest with normalized features to train our final model.

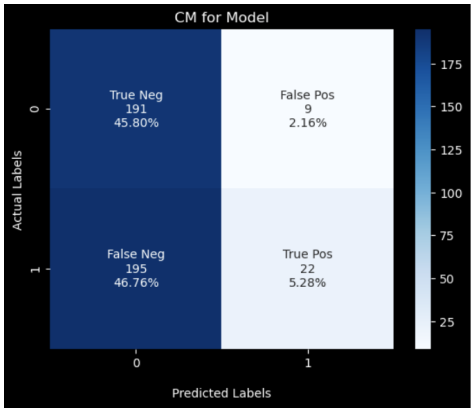
```
rf = RandomForestClassifier(featuresCol = 'features_norm', labelCol = 'label', seed = 42)

paramGrid = (ParamGridBuilder()
             .addGrid(rf.maxDepth, [2,4,6])
             .addGrid(rf.maxBins, [20, 40, 60])
             .addGrid(rf.numTrees, [20, 40, 60])
             .build())

cv = CrossValidator(estimator=rf, estimatorParamMaps=paramGrid, evaluator=evaluator, numFolds=5)

cvModel = cv.fit(training_df)
```

This will allow us to tune the parameters of the random forest more accurately to produce the best results across all of the observations in our in time dataset. Our cross-validation model had the following performance on the out of time data.



	Statistic	Measure
0	AUC	0.6220
1	Accuracy	0.5108
2	Precision	0.7097
3	Recall	0.1014
4	F1	0.1774

Despite the out of time testing precision and AUC of our cross-validated model being slightly lower than the random forests models shown above, we are more confident in the CV model since it was fit with many different train/test splits on the data and the results would be reproducible across many different seeds. An interesting finding from our CV model was that many of our predicted positives were in the Energy industry. The model predicted 24 of the 29 companies in the energy industry to have a positive year in 2015. Of those 24 positive predictions we only had 5 false positives, leading us to a precision of approximately 79% in the Energy sector compared to 71% overall.

All in all, the stock market is hard to predict, so the fact that we were able to be better than random guessing is significant. Precision is our key variable in our analysis when using this research as possible investment advice, because if we predict a company to be in the green, we want to make sure we are not getting false positives.

Conclusion

This project used pyspark in an attempt to build a model that will predict if a stock's price will go up or down in a year. Due to the overall complexity of accurately predicting the stock market consistently, the feasibility of our model exists to the extent that it is able to predict results over randomly guessing. Next steps for this project would consist of a couple aspects. The first aspect would be acquiring multiple years worth of data. Since we were looking at year-over-year delta, having only a year's worth of data to train certainly does not give the model a great picture of how the stock market can move over many years. One way we could gain more data points would be to collect the financial statements at a more frequent rate than yearly. The next aspect to improve upon is a less skewed data set. Our model was trained on stock data where about 75% of the outcomes were positive, which could have potentially affected how the model was trained. In this analysis we focused on financial statement line items, it could also be interesting to look at daily stock price changes as features in combination with these line items.