Determining Short Term Trends in Stock Prices Before a Quarterly Earnings Announcement

Ronnie Howard

1 INTRODUCTION

Investing in the stock market can be a lucrative way to invest your money for both the long term and short term. An investment in an S&P 500 Index Fund has historically grown 9% yearly on average which makes this an ideal investment for those who wish to passively invest.

In this project I am more concerned with short term investments. I wish to look at how recent trends in the stock market as a whole may be able to predict future growth in stocks. Specifically, I wish to investigate whether these recent trends can predict whether a stock's price will increase in price before a quarterly announcement. My experiment will look at a stock's price a week (5 days) before a quarterly earnings announcement and will then gauge whether the price increased or decreased the day before an announcement.

The **null hypothesis** is that there are no noticeable trends in stock growth based on the above parameters.

The **alternative** is that there is a trend in stock prices based on the above parameters.

The hope of this project is to be able to identify this trend and use it to inform short term investments.

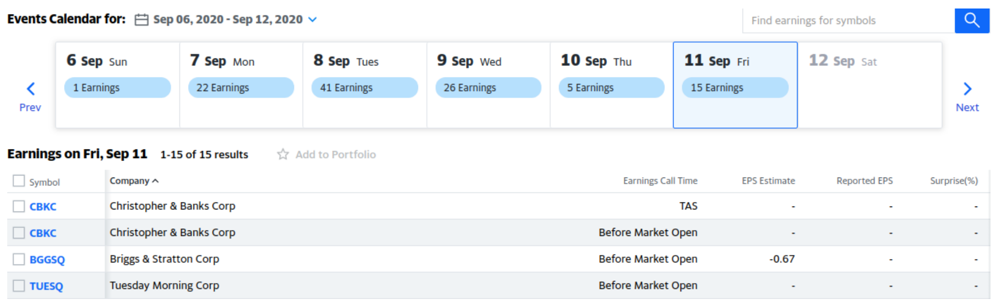
1.1 Audiences

This project may hold some significance to short term investors. These are people who try to make gains by holding a stock for only a short amount of time. This audience would be interested in this project as it may help them make informed decisions on which stocks they should buy in the short term.

2 DATA

There are two main sources that I will use to gather my data.

1. **Nasdaq's** website "old.nasdaq.com" will be used to gather an initial list of all of the major publicly traded stocks in the USA. I will specifically look at the 3 major indexes: Nasdaq, NYSE, and AMEX.
2. **Yahoo Finance** is part of Yahoo. This site provides historical information on a variety of stocks. It also provides calendars for upcoming earnings and other information. Yahoo Finance no longer has a public API. There is an unofficial Python API titled yFinance. This will be my main source for gathering information about stocks. While the yFinance API will be useful, some webscraping will need to occur in order to successfully gather all needed data.



Events Calendar for: Sep 06, 2020 - Sep 12, 2020

6 Sep Sun	7 Sep Mon	8 Sep Tues	9 Sep Wed	10 Sep Thu	11 Sep Fri	12 Sep Sat
1 Earnings	22 Earnings	41 Earnings	26 Earnings	5 Earnings	15 Earnings	

Earnings on Fri, Sep 11 1-15 of 15 results

Symbol	Company	Earnings Call Time	EPS Estimate	Reported EPS	Surprise(%)
CBKC	Christopher & Banks Corp	TAS	-	-	-
CBKC	Christopher & Banks Corp	Before Market Open	-	-	-
BGGSQ	Briggs & Stratton Corp	Before Market Open	-0.67	-	-
TUESQ	Tuesday Morning Corp	Before Market Open	-	-	-

Figure 1—Yahoo Finance Earning Calendar Example [1]

3 METHODOLOGY

3.1 Initial Data Wrangling

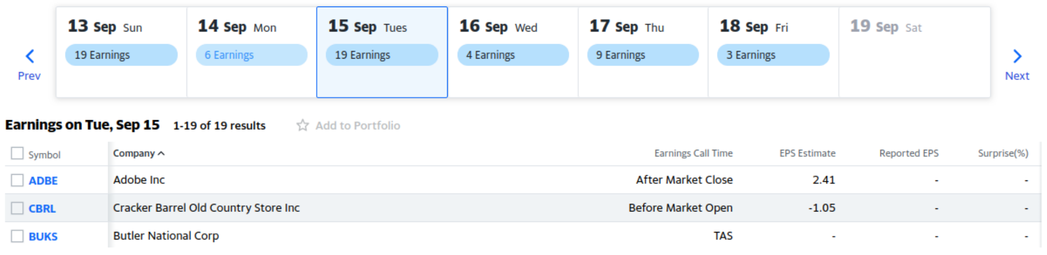
To initially gather my data, I needed a list of all publicly traded stocks. To keep things simple, I just downloaded *csv* files of the three main Stock Exchanges in the USA. I merged the 3 files together to create a list of the stocks as can be see in figure 2.

	Symbol	Name	LastSale	MarketCap	IPOyear	Sector	industry	Summary Quote	Exchange
0	GOED	1847 Goedeker Inc.	7.3500	$44.92M	2020.0	Consumer Services	Home Furnishings	https://old.nasdaq.com/symbol/goed	AMEX
1	XXII	22nd Century Group, Inc	0.6164	$85.59M	NaN	Consumer Non-Durables	Farming/Seeds/Milling	https://old.nasdaq.com/symbol/xxii	AMEX

Figure 2*—Initial dataframe of Stocks data.*

A lot of the date pulled from Nasdaq will be useful in my analysis but in order to effectively run ML algorithms on it, I needed more relevant features along with the dates for each past and future quarterly earnings. I checked out the yFinance API and determined that I could gather a lot of the features through it as well as the quarterly earnings. However, I quickly discovered that using yFinance to pull the earnings took dozens of hours to complete due to pulling 7000+ stocks one at a time. I determined that this problem needed a more efficient solution.

Luckily, Yahoo Finance's own website had a quarterly earnings calendar that could be used to gather this data. Using pandas *read_html* function, I was able to gather the data for albout 2000 living stocks in under 6 minutes. The other 5000 stocks were either dead or not available.



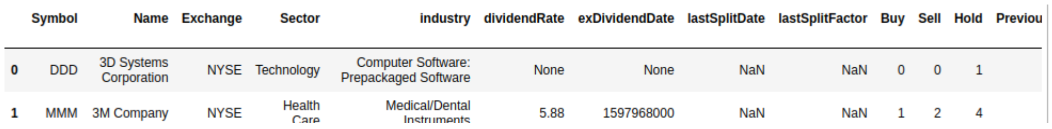
Prev	13 Sep Sun	14 Sep Mon	15 Sep Tues	16 Sep Wed	17 Sep Thu	18 Sep Fri	19 Sep Sat	Next
	19 Earnings	6 Earnings	19 Earnings	4 Earnings	9 Earnings	3 Earnings		

Earnings on Tue, Sep 15 1-19 of 19 results Add to Portfolio

Symbol	Company	Earnings Call Time	EPS Estimate	Reported EPS	Surprise(%)
ADBE	Adobe Inc	After Market Close	2.41	-	-
CBRL	Cracker Barrel Old Country Store Inc	Before Market Open	-1.05	-	-
BUKS	Butler National Corp	TAS	-	-	-

***Figure 3**—Yahoo Finance Earnings Calendar*

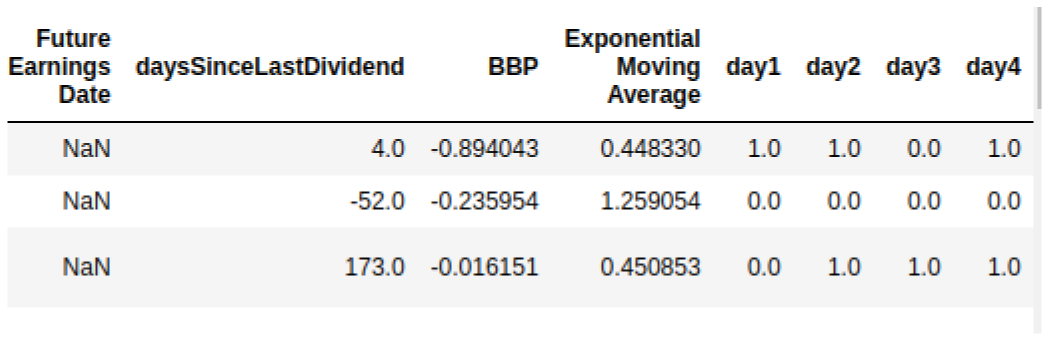
With a much smaller data set, I could use yFinance API to gather the rest of the features. Some of the features I decided to add initiall was **recent dividends, recent splits, and recent analyst recommendations.** The yFinance api allowed me to gather this data for each stock one at a time. For around 2000 stocks, this process took about two hours. I did learn that very few stocks had splits in the past 90 days but decided to keep it as a feature. The data can be seen in figure 4.



	Symbol	Name	Exchange	Sector	industry	dividendRate	exDividendDate	lastSplitDate	lastSplitFactor	Buy	Sell	Hold	Previou
0	DDD	3D Systems Corporation	NYSE	Technology	Computer Software: Prepackaged Software	None	None	NaN	NaN	0	0	1	
1	MMM	3M Company	NYSE	Health Care	Medical/Dental Instruments	5.88	1597968000	NaN	NaN	1	2	4	

***Figure 4**—Most features populated*

It is important to use some technical indicators to help determine trends. I decided to use Bollinger Bands and the Exponential moving average. The data needed to caluclate these could be pulled quickly with the yFinance API. After calculating these indicators, I also calculated whether the stock price increased or decreased in the 54days leading up to the quarter earnings. At this point my data was complete.



Future Earnings Date	daysSinceLastDividend	BBP	Exponential Moving Average	day1	day2	day3	day4
NaN	4.0	-0.894043	0.448330	1.0	1.0	0.0	1.0
NaN	-52.0	-0.235954	1.259054	0.0	0.0	0.0	0.0
NaN	173.0	-0.016151	0.450853	0.0	1.0	1.0	1.0

Figure 5*—The technical indicators are added along with whether the price increased leading up to the earnings*

Lastly, the data was cleaned to remove any rows that contained missing values in the Quarterly Earnings. With the data clean, we move on to training learners to make predictions. Bar Charts were used for the categorical data and all categorical features were found to be useful for the model. An example of one comparison can be seen below in figure 6.

3.3 Feature Interpretation

We now have a variety of features to use to train our models. The ouput variable of our model is going to be "day 4" in Figure 5 above as it represents whether the stock had increased the day before the quarterly earnings announcement. Some of our data is categorical while others are continuous, such as the technical indicators. Before training a model, I compared each feature with the outcome variable to determine any relationships.

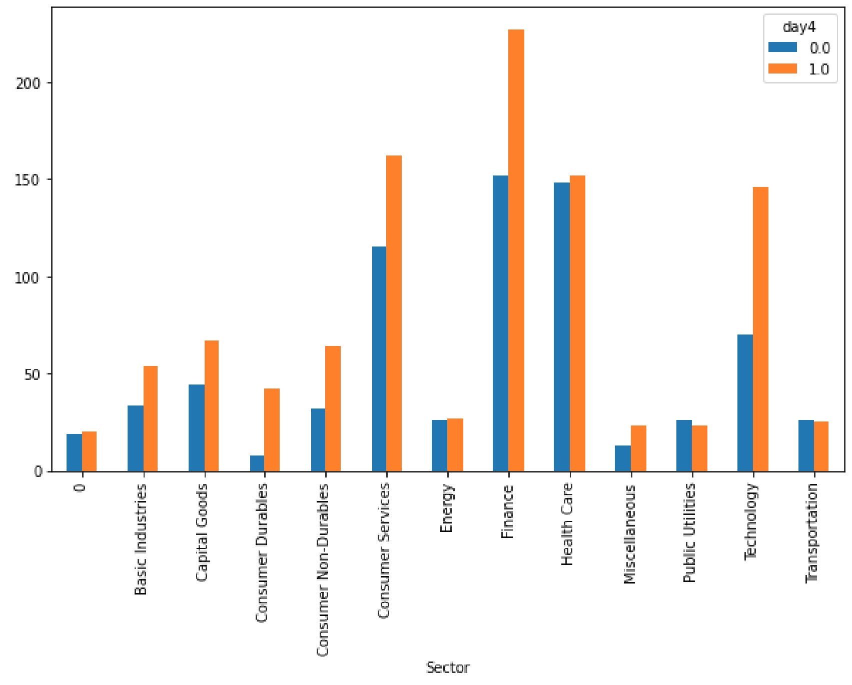


Figure 6—*Comparing features to the output variable. In this example the sectors are plotted to determine if they have any effect on the output variable. As can be seen, transportation stocks typically decrease in value, whereas technology stocks increase roughly 66% of the time.*

Continuous data was also compared using a scatter plot but no immediate patterns were noticeable by eye. Regardless, the continous data features were kept to be used in the analysis since there may be hidden patterns when used in conjunction with other categorical features.

3.3 Performing Classification

As a reminder, the purpose of this analysis is to determine whether we can determine how a stock will perform in the 4 days leading up to its quarterly earnings announcement. This is a classification problem. Therefore, a variety of classification techniques will be used.

The above data was split into a train set and a test set. While usually, it is best to use 20-30% of the data to test, I decided to use only the most recent two weeks of data for test. The reason this decision was made was to see how well the past 90 days could accurately help predict recent stock outcomes.

After splitting the data, I trained a Logistic Regression learner to use as a baseline. The learner was terribly inaccurate. It had an accuracy score of just 51% which is obviously not very useful. It does represent a baseline that should hopefully be easily beatable.

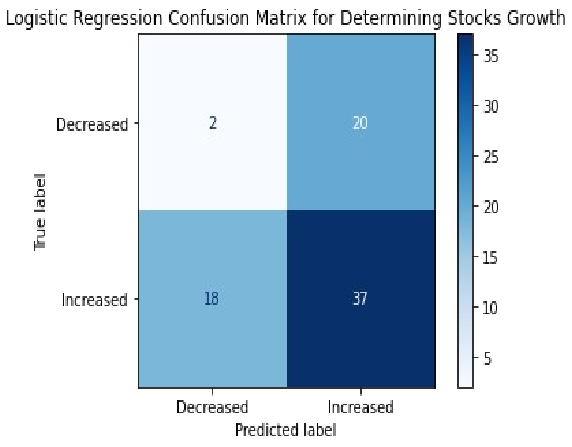


Figure 6—*Logistic Regression Confusion Matrix*

I then tried a KNN learner which gave an accuracy score of 79% which ultimately ended up being the best learner that I had. To build this model, I iterated through values of K from 1 to 100. The best value found for K was 51 at which point accuracy begins to go down. A chart depicting the accuracy from K = 1-100 is given in figure 7. The confusion matrix is given in figure 8.

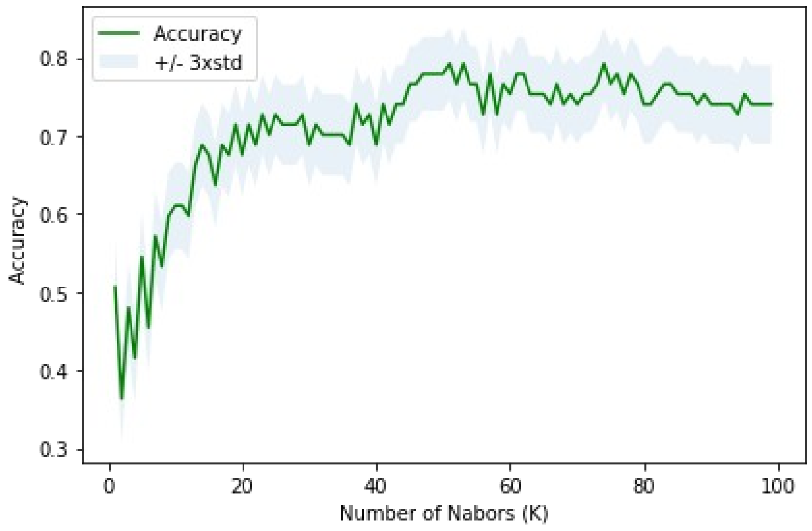


Figure 7—*Best Accuracy for Values of K*

Multiple other learners were created including an XGB learner, a Neural Network learner, a Bayes learner and a Random Forest learner. Of all of these, most got accuracy scores of 50% or less. The Random Forest learner got an accuracy score of 66%. For these reasons I believe the KNN learner is the best learner for the project as it currently stands.

4 RESULTS

As mentioned above, the KNN learner was chosen for the project due to having an accuracy that was greater than any 13% greater than the next best learner (Random Forest). Even so, the accuracy of KNN was less than 80%.

However, overall accuracy isn't the most useful metric for this project. This project was created to help inform the user on whether they should buy a stock. Therefore, we're not interested in the accuracy of when the learner tells us that the stock will decrease. We are much more concerned about how accurate the "increase" predictions are. This can be represented by the precision rate which is represented by:

$$Precision = \frac{TP}{TP + FP}$$

A high precision rate would help us feel more confident in accepting a "buy" output from our learner. Using the above formula we can calculate the precision of our KNN learner to be $Precision = \frac{49}{49+10} = 83\%$ This implies that if our learner tells us to buy a stock, then we can be around 83% confident that this stock will actually increase in price.

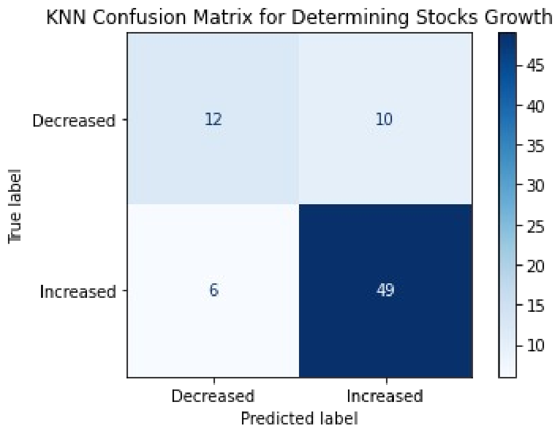


***Figure 8**—KNN Confusion Matrix*

5 Discussion

5.1 Observations

I reiterate my hypotheses:

The **null hypothesis** is that there are no noticeable trends in stock growth based on the above parameters.

The **alternative** is that there is a trend in stock prices based on the above parameters.

In the above results, it was determined that our KNN learner had an accuracy rate of 79% and a precision rate of 83%. This implies that there are trends that carry over in the stock market as a whole.

5.2 Recommendations:

Based on my analysis, I would present the following recommendations.

- It may be possible to use the above learner to make short term predictions about whether to buy a stock before quarterly earnings announcements. The KNN learner has a precision rate of 83% for the most recent two weeks of data.
- With that said, the stock market can be very volatile and recent trends in the stock market have been overwhelmingly bullish. Lots of growth has occurred in a short timespan which implies that the oppositie could hold true too. When a fluctuation in the other direction happens, this learner will most likely be very inaccurate.

5.3 Further Analysis:

Based on my analysis, I believe adding more technical indicators may help increase accuracy to some degree. Other categorical features such as volume could help improve accuracy as well. Further developments on this project would look at these other features to see if accuracy could be further enhanced.

6 CONCLUSION

I think it is fair to say that recent trends in the stock market can be captured by a KNN learning classifier. With that said, major disruptions in the market could make the classifier useless. Any use of the learner should be done in conjunction with other methods before attempting to invest in stocks. Further analysis will hopefully allow for an increase in accuracy even in major disruptions.

7 REFERENCES

1. Yahoo. (n.d.). Company Earnings Calendar. Retrieved September 12, 2020, from https://finance.yahoo.com/calendar/earnings?from=2020-09-06