Stock Price Trend Forecasting
Gulay Samatli-Pac

Background:

Stock price prediction is a challenging problem that widely studied by researchers from various fields. The biggest challenge of studying the stock market is the volatility in prices caused by their sensitivity to financial and economical noises. This volatility makes it difficult to apply the traditional statistical techniques including time series and regression models. Recently, machine learning algorithms like support vector machine, random forest, reinforcement learning, deep learning or decision tree learning have become popular among researchers and financial institutions to explain the movement of stock market. This problem is challenging though appealing to many researchers and traders because even a slight improvement could make a significant increase in profit for the stock holder.

Goals:

In this project, I discuss various machine learning techniques which have commonly been applied for stock trading to predict the rise and fall of stock prices before the actual event of an increase or decrease in the stock price occurs. Some of the algorithms that will be discussed are Linear Regression, Logistic Regression, LASSO, Random Forest Gradient Boosting Method and XGBOOST. The main goal of this project is to study and apply the machine learning algorithms to predict the movement of the underlying stock. My attempt is to model whether the stock price will increase or decrease sometime in the future compared to its value on a given day. In particular, I plan to discuss tree based models like random forest, boosting along with logistic regression in detail to predict the movement of a single stock. While predicting the movement, I also explore various variables that might have an effect on stock market or that might be used as an indicator of stock price movement. I want to make a note that my primary goal is not creating a better algorithm which is more efficient to make predictions for this project.

Data Source and Data Wrangling:

To apply different models and to compare their efficiencies, only one stock, AT&T (T) is chosen from 1/1/2010 to 4/25/2017. The first data set contains daily stock information (Open, Low, High, Close, Volume, Adjusted Close, Dividend Date). It is extracted from Yahoo!Finance using pandas_datareader library. The dividends information is also extracted from Yahoo!Finance and the earning data is extracted from busystock.com.

The following features are some candidates to explore the movement of stock prices and predict the future prices:

- n-day price change eg. n in {1, 2, 5, 10, 20, 270}[1]. It can be net price change or percent change (variable abbreviation: PCh)
- Kaufman's efficiency ratio link, link2, link3 (variable abbreviation: Effr)

[1] n in {5, 10, 20, 270} represent one week, two weeks, one month and one year.

- n-day moving average (variable abbreviation: MA)
- n-day momentum (rate-of-change) (variable abbreviation: ROC)
- Stock moving direction i.e. 1 if closing price that day is higher than the day before, and −1 if the price is lower than the day before (variable abbreviation: PrDir)
- Calendar date eg. 2 for earning record date, 1 for ex-dividend date, 0 for regular date (variable abbreviation: Action)
- Day of the week (variable abbreviation: Day)
- n-day volume difference (variable abbreviation: Volumelagged).

The initial features selection for modeling part is based on the literature and expert recommendations.

The main data is loaded into pandas dataframes via pandas_readerlibrary. Thus, it is ready to use. There are no missing values in the data set except the holidays on which the stock market was closed. Thus I start preparing the data set for modeling part. The first thing before modeling is to do an exploratory data analysis. The next section uses visualisation and transformation to explore the data in a systematic way. Then, additional features are created. Since rolling features like moving average, momentum and volume difference exclude the first specific number of the series, the first year of the data removed before modeling.

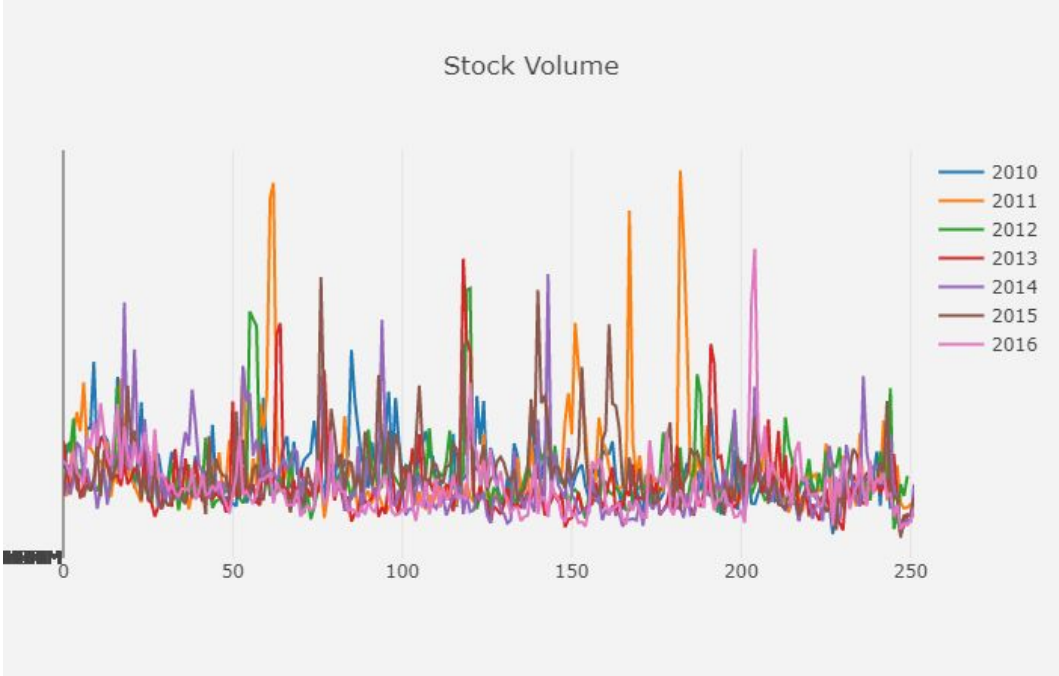
Exploratory Data Analysis:

I start exploring data with a time series visualization of closing price. It is clearly evident from below graph that there is an overall increasing trend in data along with some variations. The AT&T stock price shows overall increases about 30% from 2010 through 2017. The steepest increase in prices happens the first half of the 2012 and 2016. On the other hand, from the second half through 2016, the prices shows steady trend despite some ups and downs.

The following two plot shows the monthly and yearly rolling average of prices. Both plot support our previous observations.

Next, box plots of closing prices for each year is shown. This plot provides a bigger picture how closing prices behave with respect to each year. Except 2013 through 2015, A&T stock prices show an increasing trend with a highest performance in 2016 and lowest performance in 2015. Another observation seen in the plot is the dispersion of stock prices. While the stock prices show small variability and close to each other in most of the time, the dispersion is big in 2012 followed by 2016.

The last plot shows the volume of AT&T stocks for each year. The volume of stock looks stationary overall although it has some strikes in each year.



For more descriptive statistics, please refer [ATT-ExploratoryDataAnalysis.ipynb] notebook.

Modeling:

In this section,I evaluate the results generated on applying different methods: Linear and Logistic Regression, LASSO, Random Forest, Gradient Boosting and XGBoost. Linear Regression is applied to model the feature Close while the other methods are applied to the feature Price Direction. Before going in deep, feature scaling is applied to standardize the range of features of data. Then, the data is divided into into train (2/1/2011 to 4/22/2016) and test (4/25/2016 to 4/25/2017) subsets. To calculate the accuracy of model, a predicted value greater than 0.5 is assumed to have a fitted value equal to 1, otherwise 0.

Logistic Regression:

Logistic Regression to the response Price Direction produces the following results:

MSE = 0.4156, Accuracy = 0.5844

Coefficients (Logistic Regression)	Estimate	Pr(>|z|)	
MA10	**19.2139**	**3.10E-14**	*******
ROC20	**-5.35745**	**0.0197**	*****
PCh1	**-0.78296**	**1.62E-05**	*******
PCh20	**-5.48976**	**0.0179**	*****
Closelag5	**-4.37411**	**6.72E-07**	*******
Closelag10	**-3.59514**	**8.92E-06**	*******
Volumelag2	**0.17394**	**0.0127**	*****
Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05			

LASSO (Least Square Shrinkage and Selection Operator):

LASSO is a regression analysis method performing both variable selection and regularization to improve the prediction accuracy and interpretability of the statistical model it produces. It uses the shrinkage method. It fits a model containing all predictors using a technique that constrains or regularizes the coefficient estimates, or equivalently, that shrinks the coefficient estimates towards zero. By doing this, standard estimators can be improved, e.g. mean squared error (MSE). Moreover, it produces robust models when the number of features is 'large' (10 or more features). Thus, it helps overfitting and

decreases the computational challenges.

LASSO to the response Price Direction produces the following results:

MSE = 0.4033, Accuracy = 0.5967

Coefficients (LASSO)	Estimate	Pr(>|z|)	
MA10	**19.2139**	**3.10E-14**	***
ROC20	**-5.35745**	**0.0197**	*
PCh1	**-0.78296**	**1.62E-05**	***
PCh20	**-5.48976**	**0.0179**	*
Closelag5	**-4.37411**	**6.72E-07**	***
Closelag10	**-3.59514**	**8.92E-06**	***
Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05			

Random Forest:

Random Forest is a tree-based algorithm which involves building several decision trees using a random sample of *m* predictors from the full set of *p* predictors. Then it combines each tree output to improve the performance of the model. Advantages are as follows[2]:

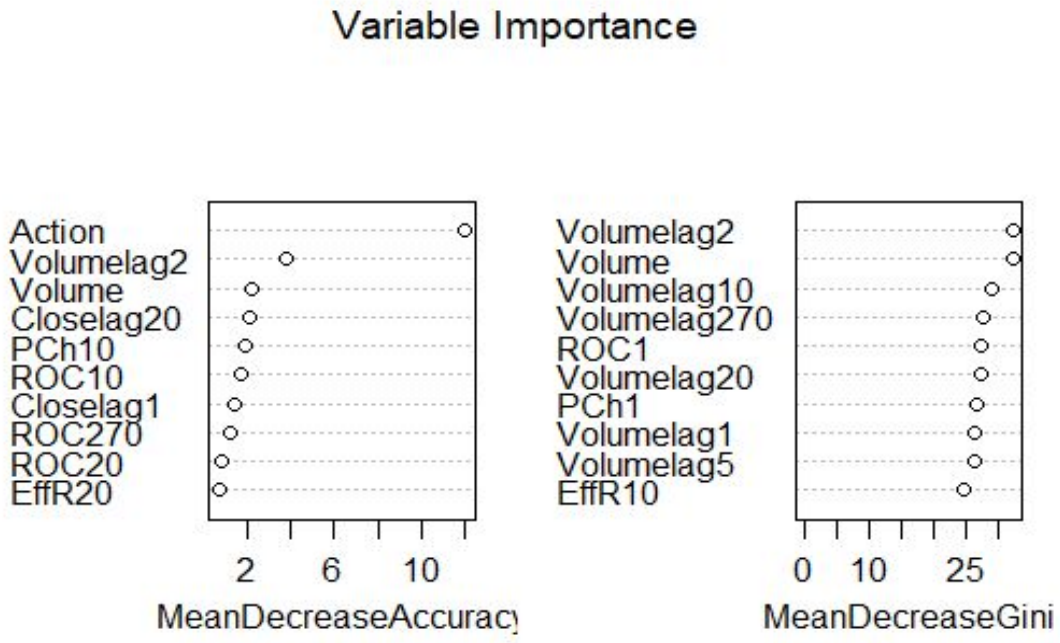
- It is robust to correlated predictors.
- It is used to solve both regression and classification problems.
- It can be also used to solve unsupervised ML problems.
- It can handle thousands of input variables without variable selection.
- It can be used as a feature selection tool using its variable importance plot.
- It takes care of missing data internally in an effective manner.

Disadvantages are as follows[3]:

- The model is difficult to interpret.
- It tends to return erratic predictions for observations out of range of training data. For example, the training data contains two variable x and y. The range of x variable is 30 to 70. If the test data has x = 200, random forest would give an unreliable prediction.
- It can take longer than expected time to computer a large number of trees.

[2] from https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/tutorial-random-forest-parameter-tuning-r/tutorial/)
[3] refer the footnote 2

The following plot provides variable importance with respect to mean decrease in accuracy and mean decrease in node impurity (gini):



In the above plot, the mean decrease in accuracy shows how much inclusion of the corresponding feature in the mode reduces the classification error whereas the mean decrease in node impurity tells how much the model fit decreases when you drop a variable, the greater the drop the more significant for the variable selection. For the above model, till Volumelag2 feature, the mean decrease in accuracy is almost stable but including Action feature decreases the accuracy significantly.

Accuracy measures for the random forest with the number of tree = 2000 (ntree = 500)

Accuracy = 0.5267 for ntree = 2000,
Accuracy = 0.5391 for ntree = 500

Gradient Boosting:

Boosting method sequentially fits multiple trees using the information from previously grown trees. In this method, smaller trees (eg than random forest) usually sufficient since the growth of a particular tree takes into account the other trees that have already been grown.

The relative influence matrix and accuracy for the gradient boosting model with the number of tree = 200, interaction depth = 4 is as follows:

	rel.inf
Volumelag2	***15.6294289***
Volume	***11.4313704***
Volumelag10	***7.6869236***
PCh1	***5.3991596***

Volumelag270	***5.1453495***
ROC1	***4.5640851***
EffR20	***4.1891484***
EffR10	***3.9478100***
Volumelag20	***3.8624858***
Closelag270	***3.7129478***
Closelag20	***3.6736036***
Action	***2.9238823***
PCh20	***2.7201664***
ROC270	***2.6147908***
EffR270	***2.5362866***
PCh10	***2.4304264***
Volumelag5	***2.1333986***
ROC20	***2.0066690***

Accuracy = 0.5514

XGBoost:

XGBoost is similar to the gradient boosting algorithms but mostly more efficient. It supports various objective functions, including regression, classification and ranking.

Accuracy for the XGBoost with the number of rounds = 100 as follows:

Accuracy = 0.5432

Linear Regression:

Simple Linear Regression to the response Close produces the following results:

	Estimate	Pr(>|t|)	
Volume	**-6.70E-03**	**0.01268**	*
Action	**-2.32E-02**	**0.00803**	**
MA10	**1.10E+00**	**<2E-16**	***
PCh1	**-5.19E-02**	**1.66E-12**	***
EffR20	**1.04E-02**	**0.03503**	*
Closelag1	**9.17E-01**	**0.00277**	**
Closelag2	**-6.56E-01**	**0.03329**	*
Closelag5	**-2.25E-01**	**1.73E-10**	***
Closelag10	**-1.91E-01**	**4.67E-09**	***
Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05			

Conclusion:

This study uses several statistical and machine learning models to predict the stock price movement. The primary goal while building a model is to produce a higher prediction accuracy instead of exploring most exploratory features.

It is known that traditional statistical models like linear or logistic regression are easy to apply and interpret. However they are not so powerful for predictions, also sensitive to any volatility. Machine learning algorithms, on the other hand, are powerful and robust for prediction but difficult to interpret the relation between response and features. For the data set selected for this study, the LASSO model with proposed parameters yields the highest accuracy followed by Logistic Regression, XGBOOST, Random Forest and Gradient Boost.

All the used algorithm in this study shows that some of the features are not statistically significant to predict the response. They create mostly computational difficulties like overfitting. For the **future studies**, the first step might be applying feature selection. Feature selection is one of the method that has a significant impact on a machine learning model's accuracy. Another future work may be to add new features to the current feature list. While doing this, it might be beneficial to keep in mind to include features that do not have significant correlation with each other. Including features related to the sector, market and some broader macroeconomic factors may also increase the predictability power of models. Here we have looked at price volatility and momentum for the particular stock and for the technology sector. For example, the stock's Price/Earnings ratio, its market cap, working capital ratio, cost-of-capital, can be some other company related features. While features like interest rate, inflation, GDP growth rate, can be macroeconomic factors.