Stock market trend analysis and prediction

Sanjey Selvan Sumathi[1] and Sangeetha Viswanathan Sakthivel[2]

[1]Department of ECE, University of California San Diego

***Abstract*—Accurate prediction of stock prices is a very daunting task. In this project, we will analyze and predict whether to buy/sell/do nothing to a stock based on the stock series data and a number of technical indicators. We use the aplha vantage API which provides the data we require. We will begin by creating the dataset and exploring the features in the dataset namely the type of the variables (categorical/ numerical), input and target variables, missing data, outliers, etc. Then we preprocess the data by applying preprocessing steps that transforms the data into meaningful features. Next, we analyze the features using univariate analysis, graphical plot analysis, multivariate analysis, dimensionality reduction, etc. Further we create different regression/ deep learning models to predict the buying and selling of stocks. Lastly, we compare the performances of different models and make enhancements accordingly.**

I. Introduction

Predicting stock market prices has always been a challenging problem and one that is so much demand. Share markets are volatile and many techniques like technical analysis, time series analysis, fundamental analysis, and statistical analysis etc. are used to predict the stock prices. There is no proved method yet for stock prediction and it is a challenging problem. We modify the problem of stock market prediction to stock market trend classification. Using the trend we suggest whether to buy, sell a stock or do nothing. We create our own dataset comprising of stock time series data and few technical indicators like SMA, EMA, etc using the Alpha Vantage apis. We then explore the features in the dataset and preprocess the data. In particular, we go through some domain knowledge on how each technical indicator affects the buying and selling of stocks and transform them accordingly. We further use concepts of mean, standard deviations to convert certain numerical features to categorical features. Once the different features are collected and transformed, some exploratory data analysis is performed on the data. This includes univariate analysis, multivariate analysis, graphical analysis, etc. Based on the analysis, feature selection is done and data is made ready to be used for training models. Different models like xgboost, logistic regression, etc are trained and evaluated. Since it is a classification problem, classification metrics like confusion matrix, accuracy, etc are calculated and the models are compared. Our models give significant performance on the classification and can be used to make stock market decisions.

II. Dataset Preparation

The right features can only be defined in the context of both the intention of our model and the data for this project; since data and models are so diverse in this field of stock prediction. We define our problem statement as "to predict the weekly stock trend i.e given the past data we try to predict the trend for the 7 days in future". Stock traders call this swing trading. Our goal is to predict if the stock price in the next 7 days is **bullish**(increase $> 2\%$), **bearish** (decrease $<2\%$) or **sideways** (within +/- 2%). Hence our target variable which is to be predicted has 3 categories So, we have to select the right set of feature for this. The nice thing about technical stock data is it complete. There are no incomplete/missing data. We used an API provided by **alpha vantage** to get the historical technical stock data. We read a few swing trading literature to select the appropriate features for this problem. We have have predicted the weekly trend of Apple Inc (**AAPL**) in this project. Fig.1 depicts the historical stock price of AAPL.

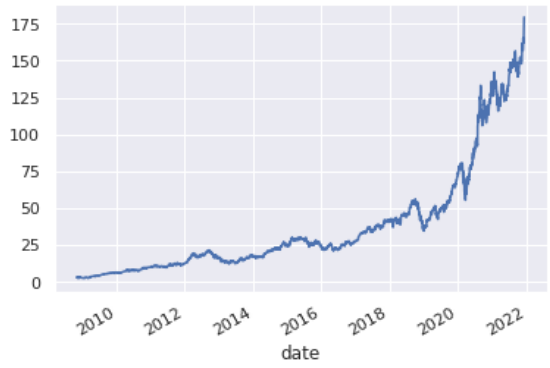


Fig. 1: Historical AAPL stock price

III. Feature Engineering

In this stage, the stock market series data and the technical indicators data are transformed based on domain knowledge into meaningful format. Firstly, the simple moving average and exponential moving average indicators crossing over the closing price can mean a change in trend i.e. a rise or fall can occur. These two features are transformed by dividing by the closing price, a division result of 1 indicates that the 2 features cross over. Secondly, RSI is a momentum indicator commonly used by traders to determine the strength of price changes in the market. A RSI value of greater than 70 indicates the stock is over brought, less than 30 means its under brought. This numerical feature can be converted to categorical feature based on this range. MACD indicator, is an oscillator and momentum indicator that's commonly used by traders for both buy and sell stocks. As is the case with the SMA crossover or the EMA crossover, when the MACD line crosses above the signal line,

it's a bullish crossover and vice versa the bearish behavior. Thus the MACD feature transformed in a similar manner to the SMA and EMA. Bollinger bands are a set of three trend lines i.e. SMA, lower and upper bands used to predicted the buying and selling of stocks. The closer the center line is to the upper band, indicates that the stock is over brought and the one closer to lower band is over sold. This features is transformed accordingly is converted to a categorical variable by setting the threshold based on 1 sd from the mean. We further normalize the data using standard scaler transformation i.e. we convert the data to standard normal distribution by subtracting mean and dividing by standard deviation.

$$Z = (X - \mu)/\sigma$$

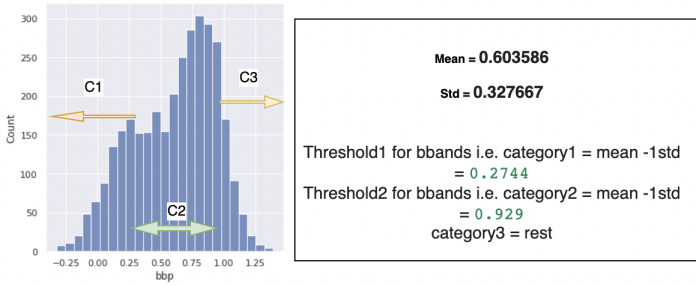


Fig. 2: Transformation of BBands to categorical feature

IV. Analyzing Relationships Between Variables

A. pairplot

Scatterplots are arguably one of the most useful visualizations when it comes to data. Scatterplots are useful for many reasons: like correlation matrices, it allows you to quickly understand a relationship between two variables, it's useful for identifying outliers, We have used sns.pairplot() to plot the density of three target categories for each features in Fig 3. This narrates an important story of how distinct is the three categories for each feature. For example for the feature "volume", the three categories are heavily overlapped. This makes this feature unusable. Whereas other features has less overlap in density for the three categories.

B. correlation matrix

Correlation matrix because it's the fastest way to develop a general understanding of all of the variables. To review, correlation is a measurement that describes the relationship between two variables. Thus, a correlation matrix is a table that shows the correlation coefficients between many variables. I used sns.heatmap() to plot a correlation matrix of all of the variables in the dataset in Fig 4.

C. Feature selection

With the help of correlation matrix and pair plot we selected the required features. There was a heavy overlap in density plot of the three categories for the features "volume" and "obv". And looking at the correlation matrix these two features are less related to the target as expected. Hence we dropped those from our data set. Another interesting observation from the correlation matrix is that the features "EMA" and "SMA" have correlation value of 0.98. This makes sense as both are moving average calculators. Since "SMA" has more correlation to the target we dropped "EMA" from our dataset.

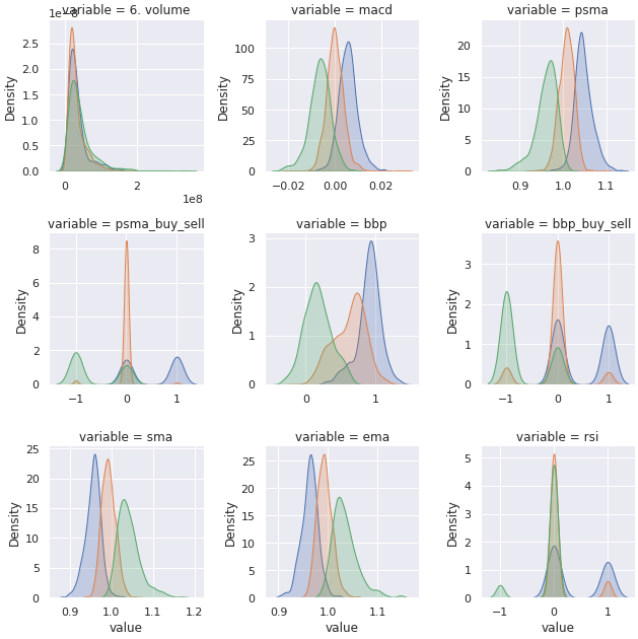


Fig. 3: Distribution of target w.r.t all variables

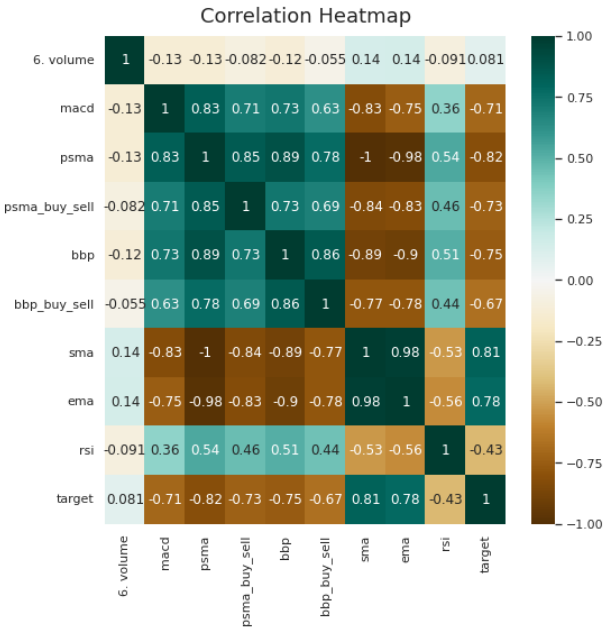


Fig. 4: correlation matrix

V. Results

In this section, we will show the result of the models. For this the data is split into train (75%) and test sets (25%). It

can be seen from Fig3 that Logistic Regression, XGBoost and SVC perform significantly well followed by RandomForest. Logistic regression predicts the probability of an outcome that can only have two values (i.e. a dichotomy). The prediction is based on the use of one or several predictors (numerical and categorical). A linear regression is not appropriate for predicting the value of a binary variable for two reasons. This was concluded based on taking accuracy as the classification metric. We further analyze the other metrics for the Top 4 models in the following steps.

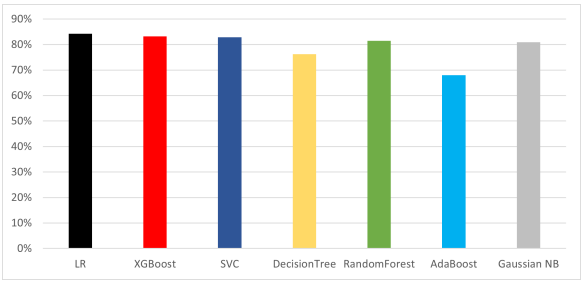


Fig. 5: Performance evaluations for different models

A. Logistic Regression

Logistic regression predicts the probability of an outcome that can only have two values (i.e. a dichotomy). The prediction is based on the use of one or several predictors (numerical and categorical). A linear regression is not appropriate for predicting the value of a binary variable for two reasons. This was concluded based on taking accuracy as the classification metric.

B. Support Vector Classification

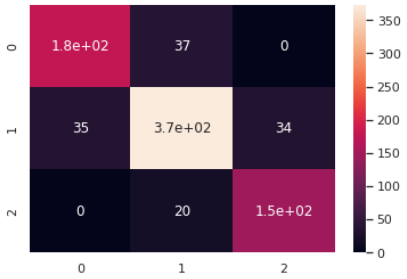
The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is.

C. Random Forest

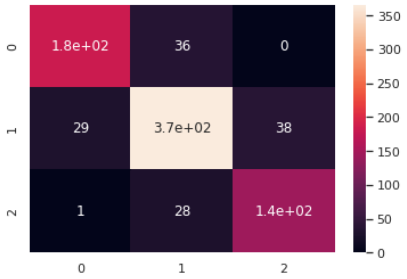
A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

D. XGBoost

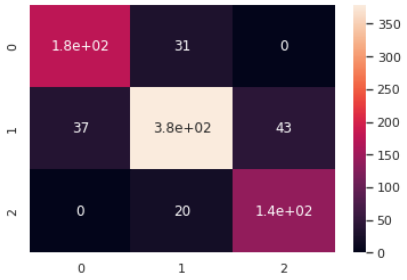
XGBoost is a tree based ensemble machine learning algorithm which is a scalable machine learning system for tree boosting. XGBoost stands for Extreme Gradient Boosting. It uses more accurate approximations to find the best tree model. We can see the performance of Logistic Regression, Random Forest, SVC and XGBoost in Fig 4. LR gives more number of false positives compared to the other models. In stock market, it is important to reduce false positives i.e. predicting buy or sell when its not needed or predicting the opposite i.e. buy when sell and vice versa. All 4 models perform in a similar manner. There are a total of 850 samples. Among that there are



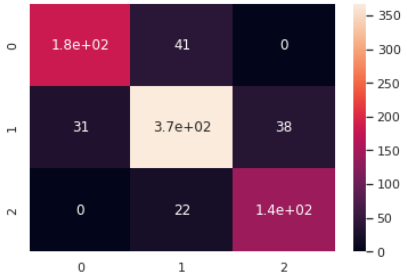
(a) Logistic Regression



(b) Random Forest



(c) SVC



(d) XGBoost

Fig. 6: Confusion matrix for different models

around 700 true positives i.e. 700/850 = 82.3%. The number of data falsely classified into buy or sell is around 35 for either cases i.e. 8% totally. The data that must be bought or sold which is not classified into any class is around 60 i.e. 7%. Thus, it can be seen that our model performs significantly well.

VI. Limitations

The stock data is dependent on both technical indicators (like EMA,SMA,etc.) and fundamental indicators (like news, earnings report, company's stability, etc). In this project we have only used technical indicators. If the stock movement

is driven by any fundamental indicator, then our model will malfunction as it is not aware of these indicators. For example, covid related news can cause stock price to deprecate and our model will never be able to predict it.

VII. Future Enhancements

Support and resistance are used by traders to refer to price levels on charts that prevent the price of an asset from getting pushed in a certain direction. Resistance is the price level at which supply (selling power) is strong enough to prevent the price from rising further. Support is the price level at which demand (buying power) is strong enough to prevent the price from declining further. Predicting support and resistance can be formulated as a regression problem on its own. Our intention is to extend this project to predict support and resistance levels.

References

[1] https://www.alphavantage.co/documentation/

[2] Short-term stock market price trend prediction using a comprehensive deep learning system Jingyi Shen M. Omair Shafiq

[3] Stock Closing Price Prediction using Machine Learning Techniques Author links open overlay panel MeharVijhaDeekshaChandolabVinay AnandTikkiwal ArunKumarc