**CS - 1: Quantitative Analysis and Modeling for S&P 500**

**Aakriti Sinha**

**9th October 2020**

**Introduction**

The meteoric increase in compute power and advances in Machine Learning have given rise to a variety of use-cases for mechanical/algorithmic trading. Quantitative funds across the world use a plethora of techniques to forecast market prices, volumes and general market behaviour. S&P 500 is one of the world’s leading benchmark indices consisting of 500 publicly listed companies. This study will be restricted to data from these companies’ price-volume data (as traded on the New York Stock Exchange).

**Objectives**

There are 3 main objectives -

**1. Volatility Index** - Out of all the 500 stocks in the dataset, establish a weekly volatility index which ranks stocks on the basis of intraday price movements. (Weekly volatility Index implies that it is to be calculated on a weekly time frame and both intraday as well as weekly change in price needs to be used in calculating volatility)

a. Give an exploratory analysis on any one stock describing it’s key statistical tendencies.

b. The index should rank the stocks from most to least volatile in the selected time frame.

c. The output needs to be grouped weekly showing the Top 10 Most and Least Volatile stocks. Both your code and output will be evaluated.

**2. Pair Trading** - The concept of pair trading suggests that there are stocks whose prices move together (could have an inverse relationship). The objective is to identify the 5 strongest pairs for every year in the dataset (eg. 5 strongest pairs for 2014, 2015 and so on).

**3. Binary Classification** - Given a stock and it’s data, you have to predict whether it will close lower than it opened (red) or higher than it opened (green)

**Data**

The given data has been downloaded from Kaggle and is very clean. The dataset contains the following columns -

● Date - The day when trading took place.

● Open - Opening price

● High - Highest price level reached during the day

● Low - Lowest price level reached during the day

● Close - Closing price

● Volume - Number of stocks traded on that day

● Name - Name or Ticker Symbol of the stock

A detailed metadata description is available in a separate report named Metadata\_Description.pdf located at /docs/.

**Approach**

OBJECTIVE 1: The approach for the first objective is statistical in nature. Volatility can be measured by the standard deviation of returns over a chosen period of time. Historical Volatility (HV) is calculated as:

HV = (Std dev of Price Returns) \* (Square root (T))

where T is the number of periods in chosen time frame. Price Returns can be calculated as the natural logarithm of price change over a single period. For weekly VIX, price returns are calculated using daily price change. The number of periods in a week is 5 (counting only weekdays as the stock market is closed on weekends).

Once the volatility index is prepared, the next step is to rank stocks. The ranking needs to be done on a weekly basis to identify the top ten most and least volatile stocks each week. We use the average volatility of a stock on the last day of week – Friday – to calculate the rank.

The final section of the first objective is related to the statistical exploration of any one stock’s performance in the market over the given time period. We have chosen to study Starbucks Corp. The Starbucks stock is examined through various market measures:

* Trend of closing price
* Trend of volume
* Frequency distribution of opening price, closing price, high , low and volume
* Daily logarithmic price returns
  + Log(closei – closei-1)
* Daily return or Percentage change in daily closing price
  + (closei – closei-1)/closei-1
* Cumulative daily return
  + Cumulative of daily logarithmic price returns
* Moving average for 10, 20 and 50 days
* Auto-correlation with a lag time of 5 days
* Weekly Volatility Index (as calculated above)

All these parameters were also plotted on a graph to enable visual analysis too.

Risk analysis was subsequently conducted for Starbuck Corp.

* Expected return rate
  + Average daily return for the time period
* Total return rate gained from the stock
  + (Daily Return last day of time period – Daily Return first day of time period) / Daily Return first day of time period)
* Total risk
  + Standard deviation in daily returns
* Sharpe ratio
  + ((Average return rate – Risk-free return rate) / standard deviation of returns) \* sqrt(252)
  + Risk-free rate is the lowest risk offered by any investment portfolio in the market. It is usually assumed to be 0.01.
  + The ratio is annualized by multiplying by square root of average number of business days a year.
  + The standard deviation represents the volatility of the stock i.e. how much the return deviate from expected value. The returns could be higher or lower than expected value.
  + The higher the Sharpe ratio, more stable or less volatile a stock is said to be. A value above 1 is said to be good. If a stock has Sharpe ratio above 2, it is believed to be very good with low volatility. Excellent stocks reach values above 3. However, a negative value denotes high volatility, which shows that the stock is not reliable.
* Sortino Ratio
  + ((Average return rate – Risk-free return rate) / standard deviation of downside returns) \* sqrt(252)
  + While similar to Sharpe ratio, sortino ratio is often studied to understand the downside risk. Deviations in return on the upper side of the expected values are welcomed by investors and is considered good risk. Downside risk denotes loss in value. Thus, sortino ratio penalizes only the harmful downside risk.
  + The higher the Sortino ratio, the less risky the stock is. A value above 1 is said to be good. If a stock has Sharpe ratio above 2, it is believed to be very good with low downside risk. Excellent stocks reach values above 3. However, a negative value denotes high risk.

OBJECTIVE 2: The second objective required to identify the strongest pairs of stocks of each year. Pairs trading is widely followed as an investing or trading strategy. It revolves around the concept to simultaneous movement of a pair of stocks. Certain stocks are affected almost identically by market conditions. These often belong to the same portfolio or industries. Traders look for deviations in their movements i.e. difference in their stock prices. When deviation in trends of a pair is large, there is opportunity to trade.

Pair trading analysis involves looking for these very deviations. Machine learning techniques can help find patterns that people may miss. In our case study, we are using these techniques to identify such pairs of stocks that generally move together in the market. A huge correlation matrix is built with each pair of stocks trading in a year. Each pair of stock is measured with the Pearson’s correlation coefficient. Each pair is thus sorted in descending order to identify the tem most and least correlated pair. This procedure is repeated for each year. Finally, the top ten and the bottom ten from the ranks of each year are reported.

OBJECTIVE 3: The third objective is to predict whether a given stock will close at a price higher or lower than the opening price on a given day. The prediction is supposed to be 0 or red if it is lower, 1 or green if it is higher and 0.5 for no confidence.

The prediction is done by multinomial classification. First the data was selected for building a classifier model. A random stock name and date were selected. Only the data for that particular stock till that particular date was used for the purpose. Thus look ahead bias was avoided.

The data was processed to create date features and categorical variables. Missing rows were found. There are only 11 rows with missing values. 8 rows have all data missing for open, high and close. So they will be dropped. Out of remaining 3 rows, two are missing Name tag, so they will be dropped. The remaining 1 row is also dropped as inferring opening value from high, low, close is unreliable due to volatile nature of stock market data.

Next, the data was split into training and test datasets. The split ratio was 80:20. The last 20% rows were held out as future dates.

On the training dataset, the target and new features were created.

Target : closing price higher or lower than opening price. Variable needs to be created. 2 - higher, 1 - no change, 0 – lower

Features: open, high, low, close, volume, price return, volatility, year, month, week, dayofweek, mon-fri, Stock High minus Low price (H-L), Stock Close minus Low price C-L), Stock High minus Close price (H-C), MA for 10, 20, 50 days, std dev for 7 days, today's close compared with 1 or 2 previous days'.

Next, feature selection was attempted. Features were scaled using MinMax scaler, bringing numerical columns in the range of 0 and 1. PCA and factor analysis was attempted but a bug was encountered. Refer <https://github.com/scikit-learn/scikit-learn/pull/9105> discussing an open bug related to incorrect calculation of explained variance. Thus, dimension reduction was skipped for the time being.

We trained eight classifier models:

* Multinomial logistic classifier
* K Nearest Neighbours
  + 3 neighbours
  + 5 neighbours
  + 7 neighbours
* Gaussian Naive Bayes
* Decision Tree Classifier (CART)
* Random Forest Classifier
* Gradient Boosting Classifier

The baseline prediction was estimated with a dummy classifier using stratified strategy to make a naïve prediction. A pipeline was created to prevent leakage of training and test data. The prediction probabilities were binned as:

* (0 – 0.40] probability 🡪 0
* (0.40 – 0.60] probability 🡪 1
* (0.60 – 1] probability 🡪 2.

Model selection was done on the basis of f1 score to combat class imbalance through 5 fold Time Series cross validation, which helps avoid look ahead bias too. The best model was the Random Forest Classifier pruned to maximum 3 branches.

The trained model was then tested over the partitioned testing dataset and metrics were examined gain to check for overfitting.

**Findings**

**Objective 1:**

The Weekly volatility Index was successfully used to rank the top 10 most and least volatile stocks of each week. The report is placed as worksheet named Top\_10\_Volatile\_Stocks in Output\_Report.xlsx at / reports/.

The exploration of stock market performance of Starbucks Corp is visualised through graphs present in Starbuck\_Stock\_Trend\_Figures.pdf at /reports/figures/. Some findings re:

* The company’s meteoric rise in closing prices since 2013 slowed down a little for a year in 2014-2015 but nearly double the next year. However, after 3 years of sharp growth, the company saw a stable period till 2018.
* However, volume has seen sharp peaks in 2018.
* Stock prices tend to be in 55-60 or 35-40 range
* Majority of the time, stock volume has been around 10,000,000
* Daily returns rate tends to be in the +/-0.25 range and has an almost normal distribution.

Risk Analysis:

The calculated values for risk analysis for Starbuks can be found in Risk Analysis for SBUX.csv at /reports/. The stock performance is judged to be poor. Sharpe’s ratio indicates high voltily and Sortino ratio indicates high downside risk.

**Objective 2:** The strongest pairs of each pair were found. The report is placed in worksheet named ‘Pairs\_Trading\_Strongest\_Pairs’ in Output\_Report.xlsx at /reports/.

**Objective 3:** The trained model has 23 features. The baseline accuracy was 0.48 and f1 score was 0.34. The best model was the random Forest Classifier, which achieved an accuracy of 0.83 and f1-score of 0.61. Thus RFC showed marked improvement over baseline.

* The validation model showed slightly lower accuracy (0.81), which is expected and, also, desirable to prevent over-fitting.
* The recall is higher than the precision. This is also desirable as we aimed to prevent false negatives.
* Support for class 1 is very low, which shows that the model has “no confidence” on very few cases.
* The importance of each feature was also identified. The most important features are closing price on the previous two days, daily return (percentage change in closing price), and proximity of closing price to high and low prices of the day.

**Challenges**

The main challenge was to ensure that no look ahead bias enters the model as the data was a time series. Another challenge was heavy class imbalance. The predicted output was a range of probabilities. It was converted into a category. In the first iteration, the output was cut equally into three parts. The lowest range was assigned the value of 0, the middle range was assigned 1, and highest range was assigned 2. They could not be assigned the values of 0, 0.5, and 1 as categorical variable cannot hold float values. However, with this division, when classification metrics were examined, it was found that while accuracy is very high, precision and f1-score are undefined. This happened because, due to majority of zero cases, the model became heavily biased towards zero prediction. A different demarcation of probability led to better results.

The task of finding weekly and annual ranking from daily level data was also a challenge. The data had to be grouped very carefully to attain the proper results.

**Conclusion**

This project attempted to analyse the historic stock market data of S&P 500 companies and used several techniques to help investors make better informed decisions. It was demonstrated with a deep dive into the stock market performance of Starbucks Corp, which revealed the highly volatile and risky nature of its stocks. A classifier model was also built that could predict with more than 80% accuracy whether the closing price of a given stock would be higher or lower than the opening price on a given day. The model showed that today’s closing price is influenced by the previous two days prices, percentage change in price observed yesterday over the day before, and how near yesterday’s close price is to the high and low prices of the day. These factors can give important pointers to investors studying the stock market.

**Retrospective**

Many more functionalities can be added to each part of this project. The exploration of a single stock’s performance can be made dynamic so that users can choose the stock they want to study. An interactive dashboard can be built using Plotly Dash for a more comprehensive visual analysis. The classifier model can be optimized for hyper-parameters for better performance. There is an urgent need to look in to the reasons behind failure of dimensionality reduction techniques. Decision trees can handle both large number of features and imbalanced data. However, the performance of logistic classifier could also have improved after reducing redundant factors. Time constraints also made for a messy coding structure, which has huge scope for improvement.