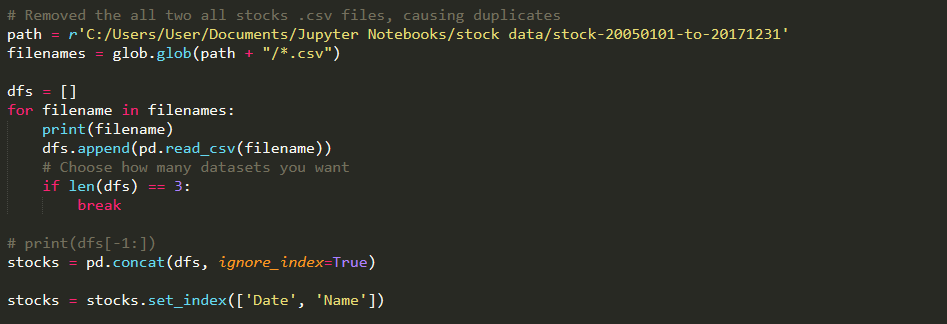
Introduction

This is a project aimed at delivering a report illustrating the tasks completed to perform algorithmic trading based on Machine Learning. This includes tasks such as experimenting with feature selection, using a range of machine learning method to select the best performing method, completing an evaluation using a number of stocks.

Data preparation

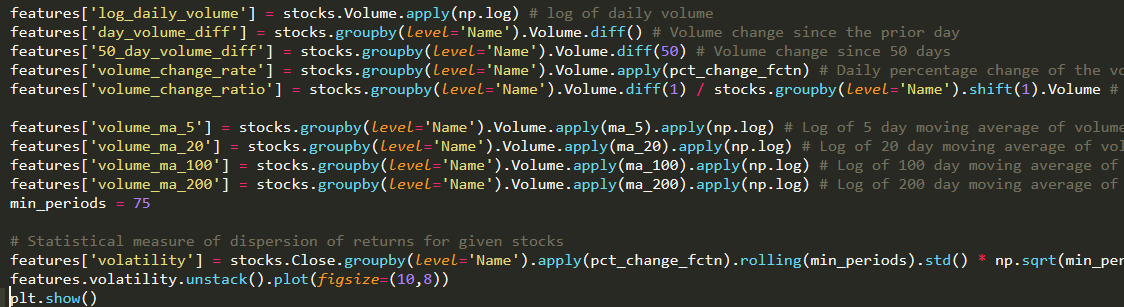
To begin with, I setup a process that would allow me to select as many or as few stock datasets that I could perform feature selection and evaluation on. This allowed me to stop graphs and matrixes becoming cluttered and unintelligible, but also retain a large amount of data to work with.



In the example above, I limited to 3 datasets at the beginning to test functions previously unknown to me and increased to 5 later. A multi-index of Date and Name (symbol) was used by the data frame to uniquely identify each instance, for who the data belonged to and the date occurred. I dropped missing data, for the first three datasets this was only 7 instances out of 3000~ so not a great loss, and each instance is likely recorded in the evening or night of the date stated.

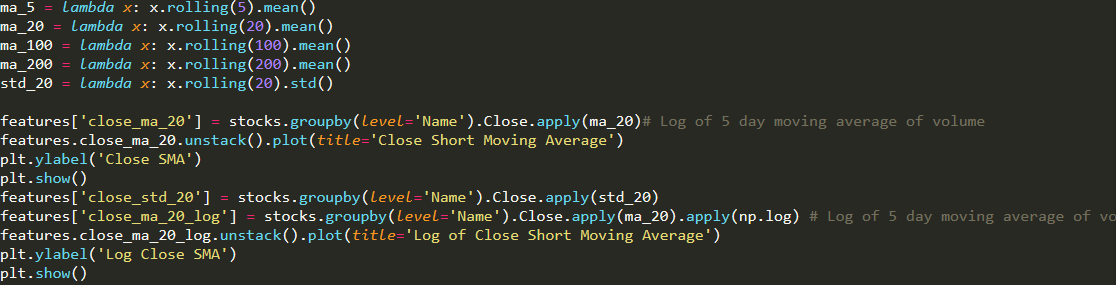
Feature engineering

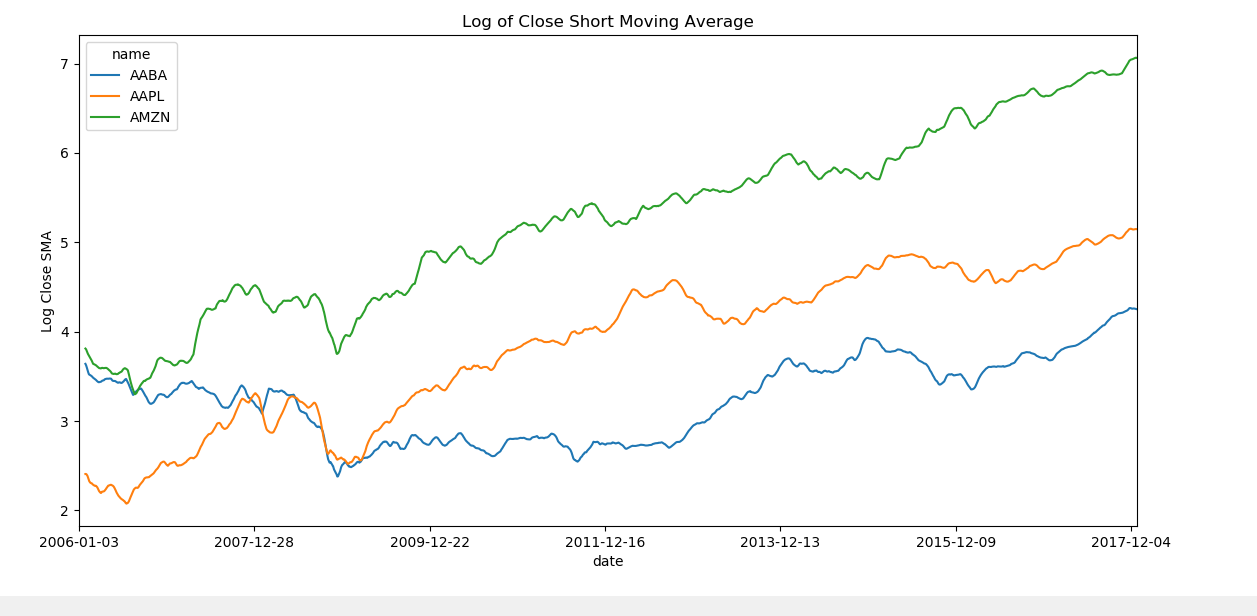
Due to having little to no knowledge of stocks, I initially used Open, Close and Volume as features, then used them to create as many features that could possibly be useful, including a moving average, volume change rate, and a few technical indicators for momentum volatility:



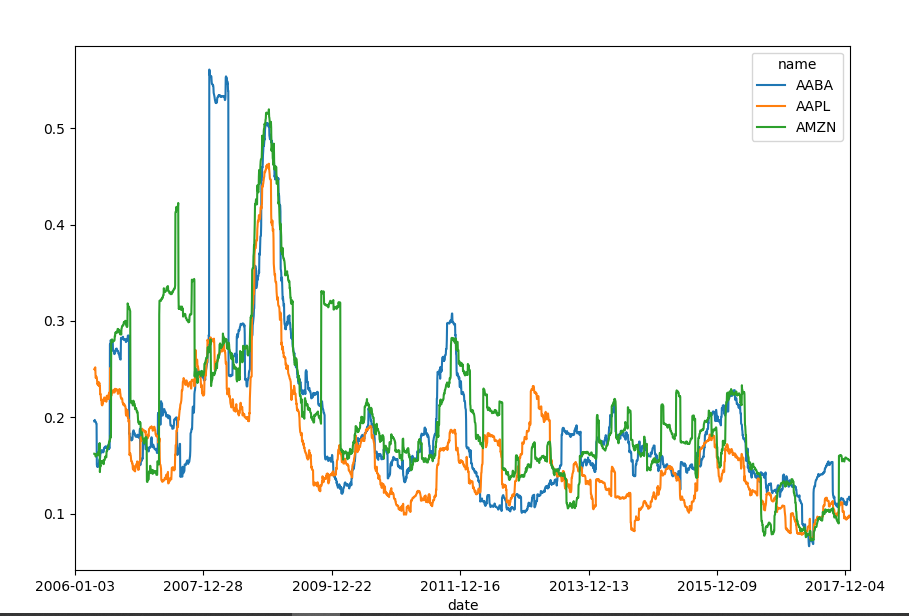
This feature engineering provided the basis to experiment with a range of potential factors and were extracted later using selection techniques.

For moving averages, I also took logs, for I read that it can improve the performance of predictive models.

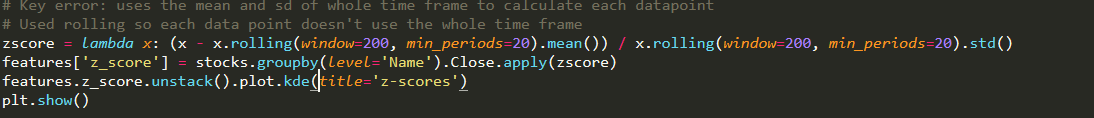


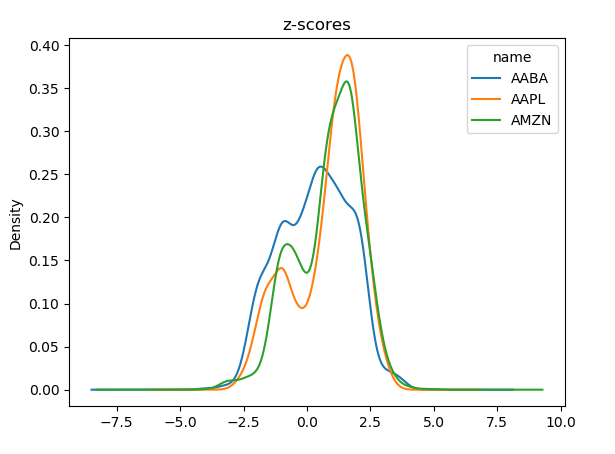


Volatility – a measure of the dispersion of returns, higher the volatility the riskier the security:



z-score to see the density of points that deviate from the mean of Close prices:

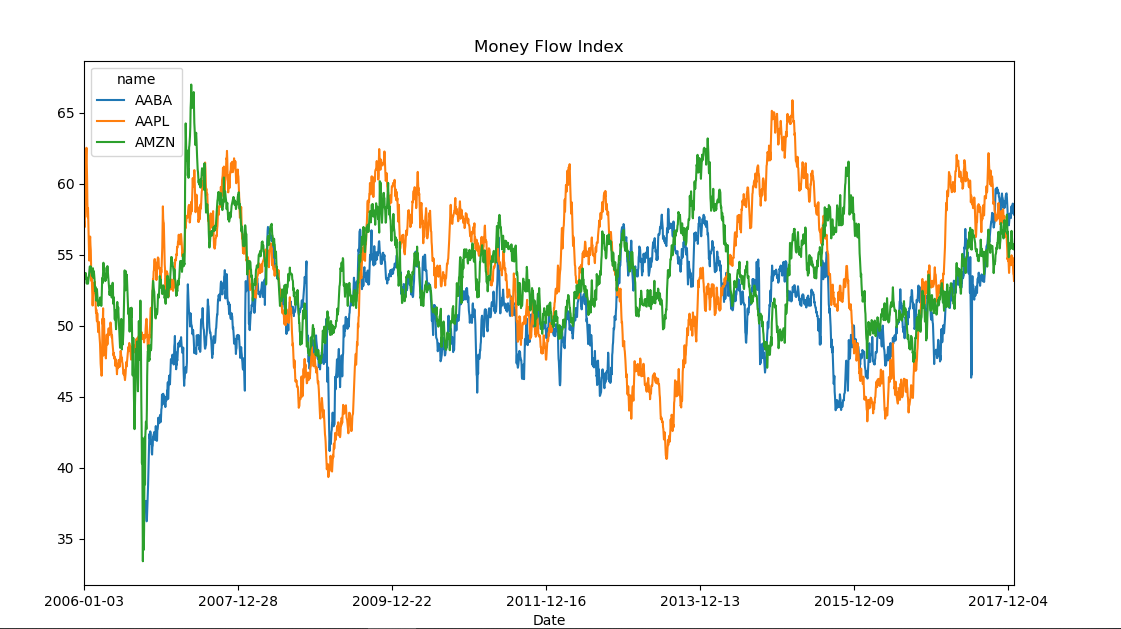




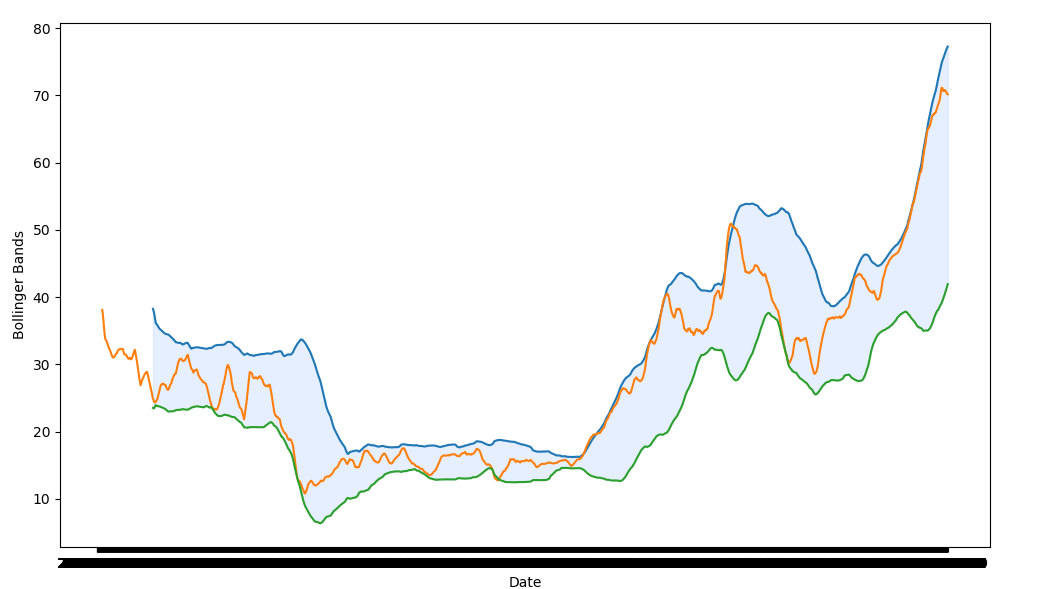
With Apple and Amazon recent rapid growth, it is expected that they’re closing stocks average above the mean.

In addition to that, I used ta-lib to create additional features, including money flow index, Bolling bands and WILL R.

Money flow index over 200 time periods:

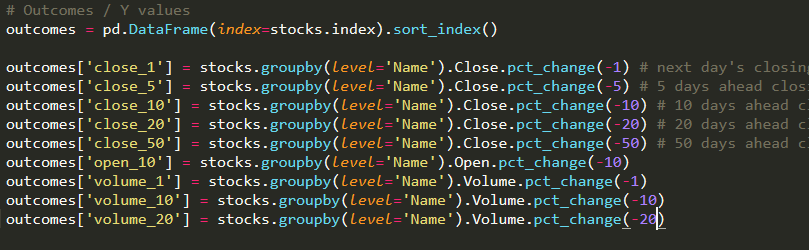


Here is the Bollinger bands figure for AABA stock over 20 periods of time. The top of the band represents the upper, the bottom green line represents the lower band, and the middle orange line represents the SMA of the close datapoints. using more than one dataset would cause errors for the fill\_between() function and looks messy. There is also an error with the date not displaying, however, the data is from 2006-01-01 to 2018-01-01.



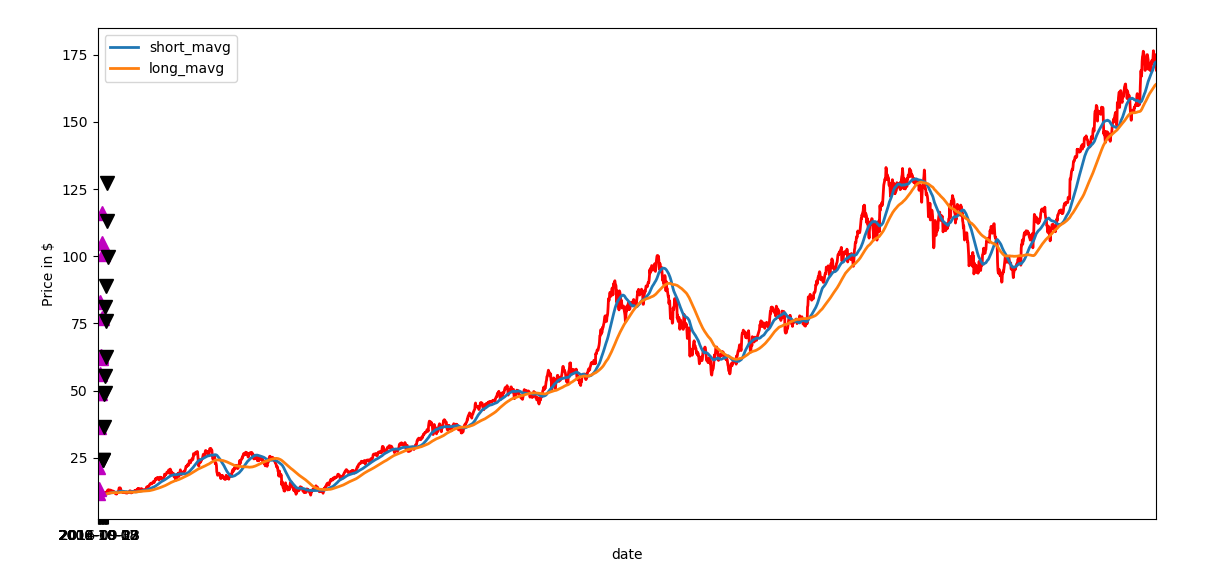
The upper and lower bands are typically 2 standard deviations +/-. The closer the prices move to the upper band, the more overbought the market and the closer to the lower band, the more oversold the stock is to the market.

For outcomes or y values, I put them into a different data frame to use for the predictive model.

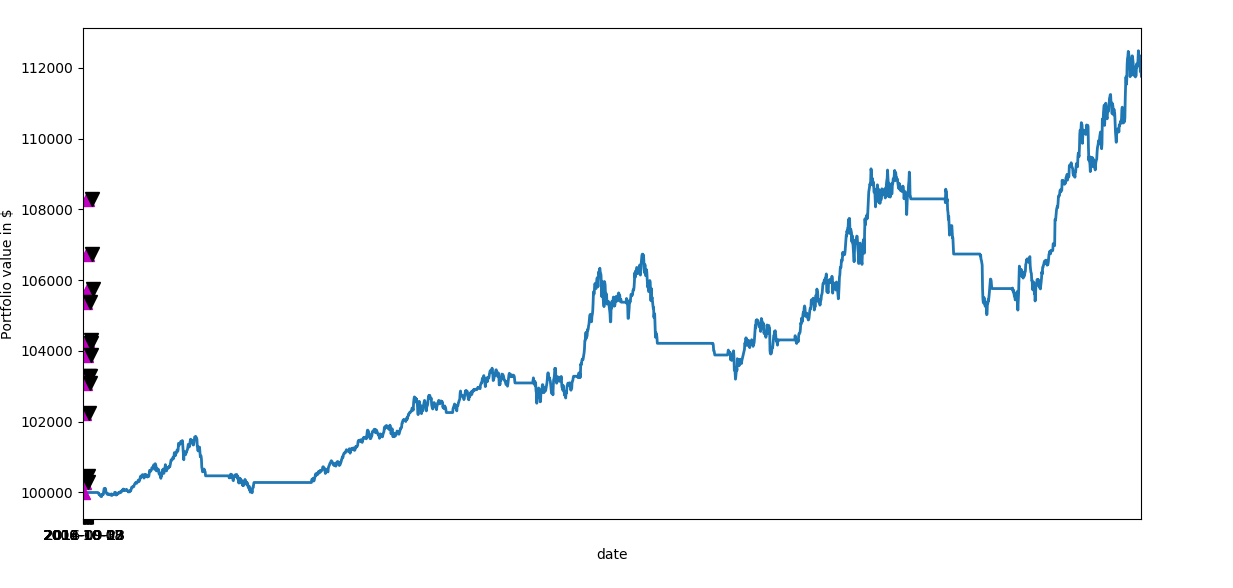


Trading strategy

I created a trading strategy mainly using the Amazon and Apple stocks. However, regrettably, these weren’t used for supervised training aside from OLS.



Backtesting:



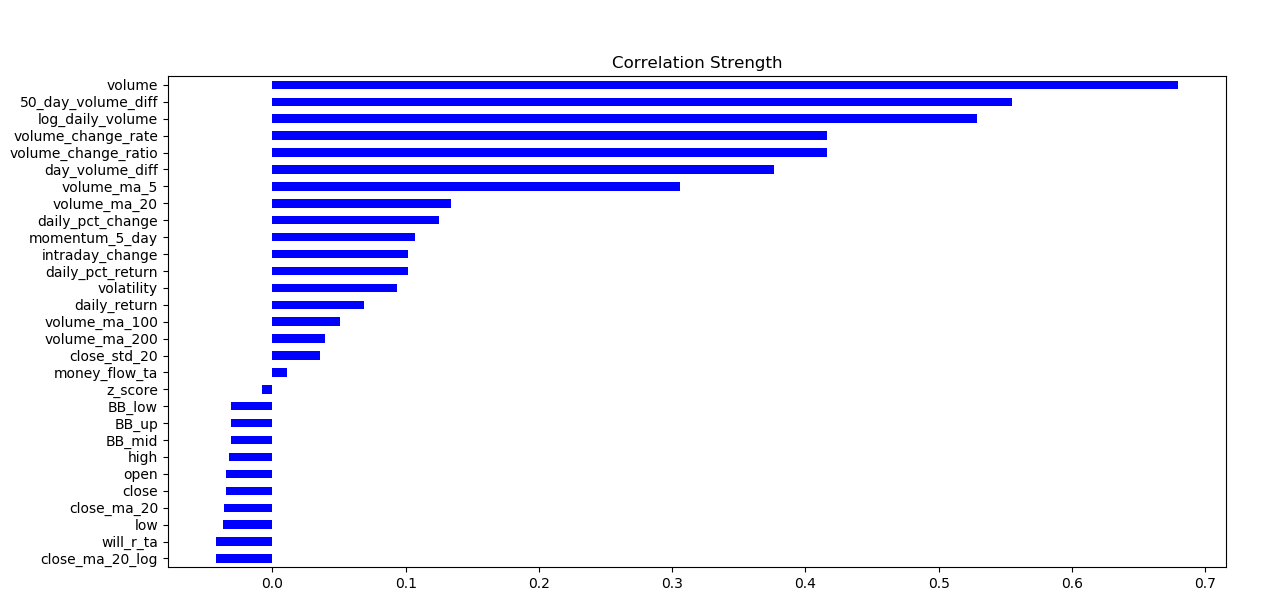




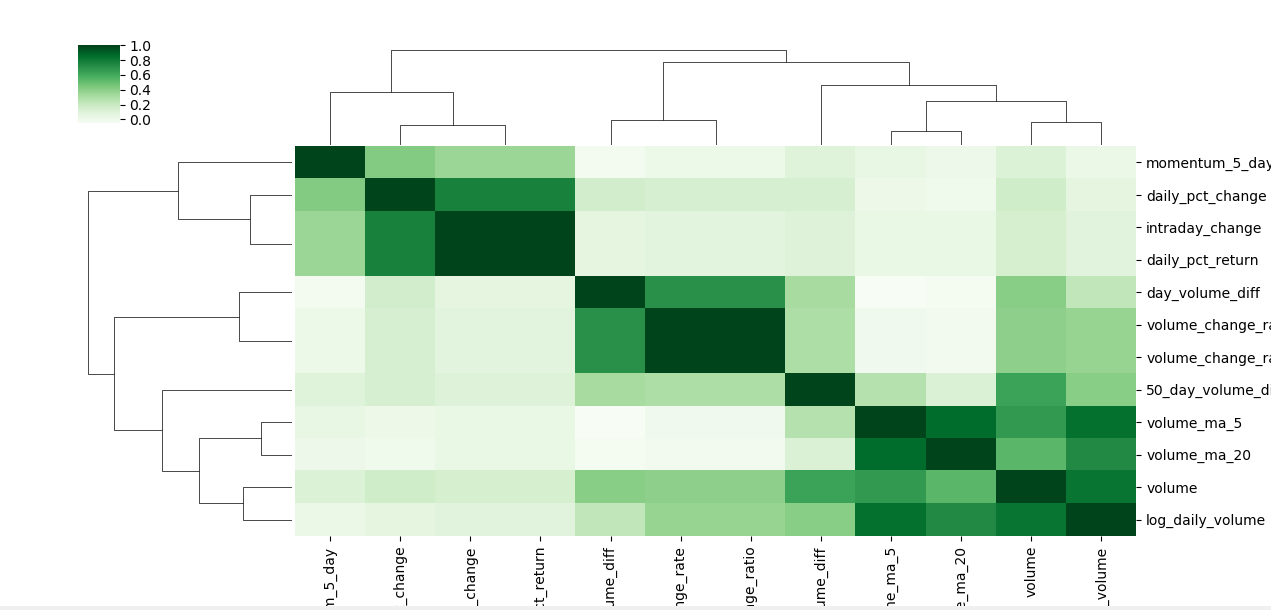
Feature selection

Before selection, I standardized data frame data. Instead of using a sklearn feature selection function, I performed two manual feature selection operations: one to measure feature correlation strength with the y variable and another to measure orthogonality between features. Correlation strength is important for how the features influence the outcome, and it is important to use features which don’t overlap or have minimal side effects and influence on each other.

I compared the features correlation strength with different datasets and different y outcomes, the strongest Pearson correlation was consistently the volume, and other various features are shown below, with volume change of 10 days in the future as y value, and this performed best on the Amazon stocks, a close second was for volume 20 days in ahead as the y value:

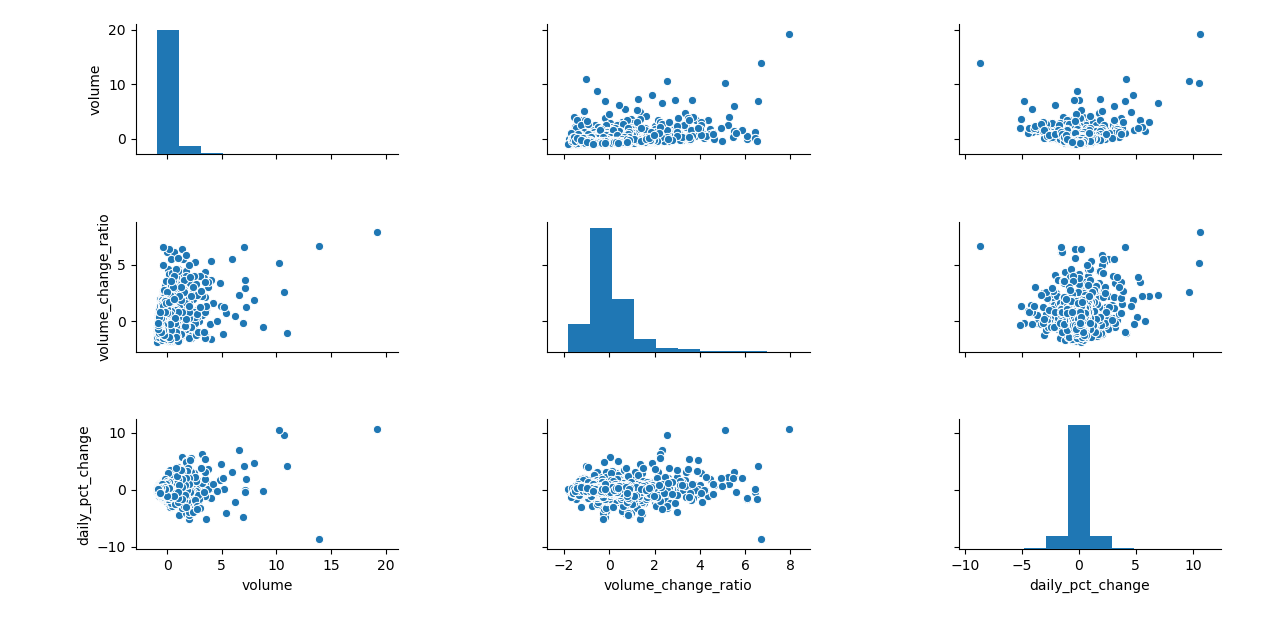


To measure orthogonality, I used a heatmap of a covariance matrix of the features. This illustrated the covariance between each of the features, so I could identify the positivity of their covariance. 1.0 is a strong positive covariance showing a possible linear relationship, and 0.0 is a weak or no correlation between the features.



To manually extract features with these two visual forms, I used a simple heuristic: I selected the strongest feature, then continually searched for the next feature in strength that didn’t have a strong correlation with the previously selected feature on the heatmap. This resulted in the selection of three features: volume, volume change ratio, and daily percentage change.

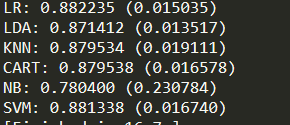
There are weak patterns among these features, nothing highly correlative as shown below. I can’t exploit the volume with parametric methods for it doesn’t have a Gaussian distribution.

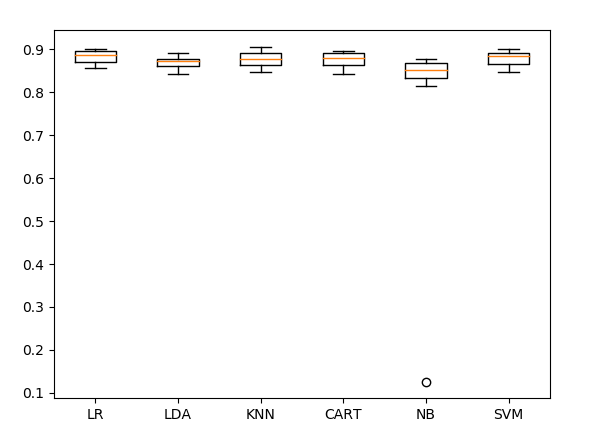


With having three features, I will avoid on the “curse of dimensionality”, which some algorithms are susceptible to e.g. KNN and clustering, where the volume of feature spaces increases exponentially, data becomes sparse in the space that it occupies which makes it difficult to achieve statistical significance with the models.

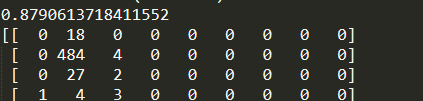
Model performance and Evaluation

The features performed well on the KNN, Logistic Regression, SVM, NB, Cart, and LDA, with default parameters.





I chose to evaluate the performance on the KNN model for it performs well with few features, and easy to understand. For default parameters, I received a KNN accuracy of 88% without the 10-fold cv harness, with 5 neighbours



After experimenting with different K values, on the KNN model without the harness, 2 neighbours performed (tied) best:



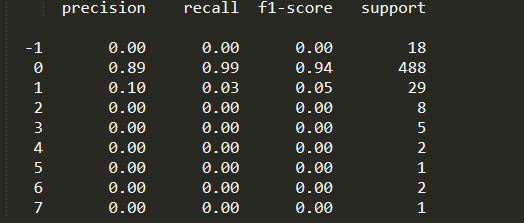
However, I am wary with such a low number of neighbours this could have overfitted on training data, so I returned to my k-fold harness and it did confirm lower accuracy with 2 neighbours:



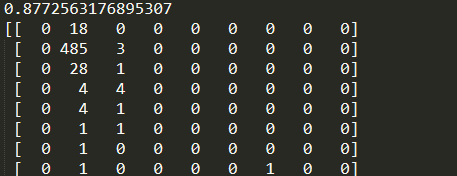
Performing evaluation using the k-fold harness, I discovered that k=19 produced the best results:



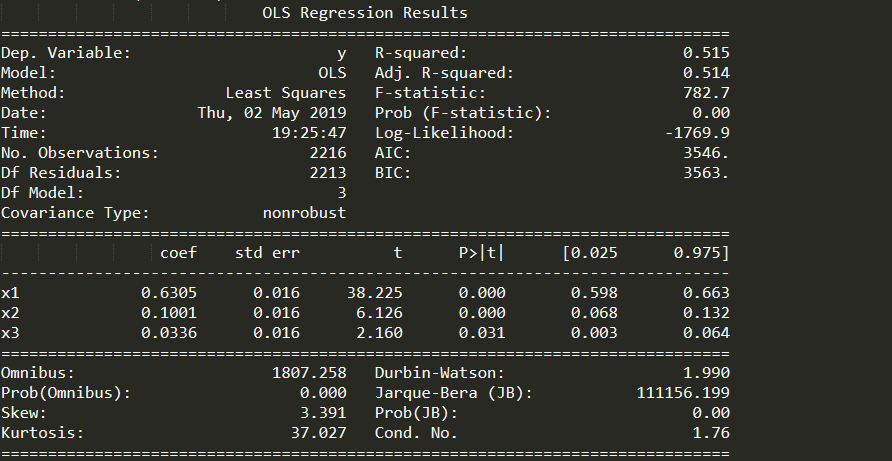
As seen above and below, there are additional dimensions that were added other than the three features used, at 0.00 these represent labels with no predicted samples. There must have been an error somewhere, maybe when encoding.



When ran on the unseen test data, the accuracy score is similar to previous, however with the higher k value of 19, it is not risking extreme overfitting.



Lastly, I perform OLS regression on the 3 selected features and volume in 10 days dependent variable:



The R-squared value is a goodness-of-fit measure that indicates a percentage of the variance in the dependent variable that the independent variables can explain. So, in this case, the R-squared value of 0.515 means that the independent variables can explain for over half of the dependent variable results.

All P values are below 0.05, so I can reject the null hypothesis because there is no correlation and I can claim there is statistical significance between the features and the response.