**PROJECT OVERVIEW**

This project has been learned and presented across varied financial stocks basics indicators and stock price anomaly detectors and predictors machine leaning algorithms.

How stock market world trade uses the graph and key indicators to analyse and take decisions. How. basic statistics concept implemented in python’s maths library help them to comprehend and anticipate the outcomes. It should be noted these concept are very old and in price prediction attempt which made is based on the assumption all of the factors are constant and only focus is given to Closing price.

However in reality there n- numbers of factors which impact the price of a stock. The factors here presented are based on historical time-series data only.

Even though most of the trading is being based algorithm and every tech company tries to make Holy –Grail of stock market where they can predict the prices. But in reality stock prices are always out-smart the Machine Leaning algorithm. One cannot deny the fact the with introduction of mathematical models and high computing really helped in trading but its ML speculation is limited to high frequency and short maturity trading. Where logical constraints and threshold has been introduced . But when it comes to Long maturity and very High Notional trading say trades with hindered of million with maturity of 20-30 years then comes the human intelligence.

Here is minuscule attempt is made in the step forward to learn and implement basics of machine learning.

**Problem Statement**

* The problem is here to know the basic of stocks indicators i.e. how ,why and when they are used.  
  namely Bollingder curves , Cumulative and Daily Return , Sharpe Ratio, CAPM model (Alpha and Beta of the stocks)
* Secondly is to see the anomaly in the prices of stock wherein actual historic stocks prices are analysed and bucketed under some thresholds and Outliners are shown as Anomalies.
* Lastly LSTM is being used to predict the stock price after failed attempt of Linear Regression and Knn model.

**Metrics**

Root Mean Square Error is being used as LSTM performance gauge . And Mean Square Error is being used for benchmark model linear parametric regression and K-Nearest Model

**Data Exploration**

Various Stocks data has been taken into consideration and compared with one another over different time periods. The comparison is made across only one attribute that is Adjusted Closing price.

To pull out the data there are various API available, however I find Yahoo Finance API more convenient and easy to use than others.



**Data Visualization**

**01-01\_Reading\_and\_plotting\_stock\_data**

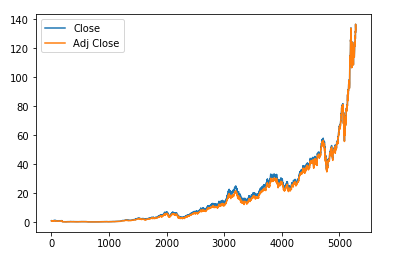
Stock values are stated in terms of the closing price and the adjusted closing price. The closing price is the raw price, which is just the cash value of the last transacted price before the market closes. The adjusted closing price factors in anything that might affect the stock price after the market closes.

A stock's price is typically affected by supply and demand of market participants. However, some [corporate actions](https://www.investopedia.com/terms/c/corporateaction.asp), such as [stock splits](https://www.investopedia.com/terms/s/stocksplit.asp), [dividends](https://www.investopedia.com/terms/d/dividend.asp), and [rights offerings](https://www.investopedia.com/terms/r/rightsoffering.asp), affect a stock's price.

**Stock Split** - A stock split is a way for a company to lower its per-price share. This may occur to make the stock more accessible (i.e. its price is too high), to make the shares more divisible, etc. If a stock in company undergoes a split, one wants to have the same total value, which is done just by splitting across more shares. In a price chart, this is dealt by an adjusted closing value. Going backwards in time, when encountering a split, the “actual” price is divided by the split factor to create the adjusted price. This adjusted price is easier to understand

**Dividends** - They are issued periodically and they have and effect on stock price. Consider a company that pays out Rs100/- dividend whose “true worth” is Rs 1000/- per share. What would be stock price a day before and on the day of dividend? As payout day gets closer, the share’s price will grow towards Rs 1100/- (adding dividend ,stockprice ), sharply dropping off Rs 1000/- after the payout. The reason because people know that they will get Rs 1100/- worth of the value on that day – the dividend and the share.

Adjustments allow investors to obtain an accurate record of the stock's performance. Investors should understand how corporate actions are accounted for in a stock's adjusted closing price. It is especially useful when examining historical returns because it gives analysts an accurate representation of the firm's equity value.



In this module, Difference between a closing price and Adjusted closing price of AAPL stock is being shown on the graph

we will see how to read data, select subsets of it and generate useful plots, using pandas and matplotlib.   
--> Read stock data from CSV files:

pandas.DataFrame

pandas.read\_csv

--> Select desired rows and columns:

Indexing and Slicing Data

Gotchas: Label-based slicing conventions

--> Visualize data by generating plots:

Plotting

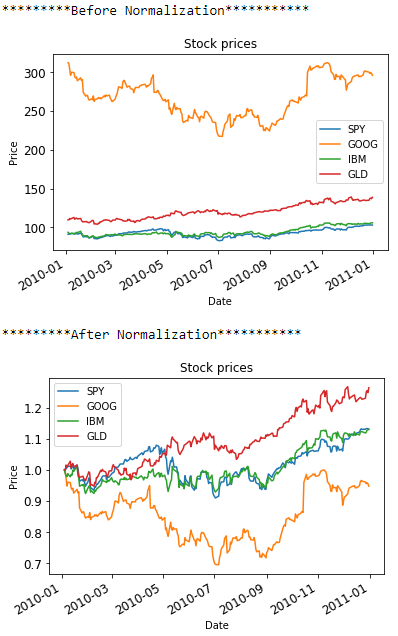
pandas.DataFrame.plot

matplotlib.pyplot.plot

**01-02\_Working\_with\_multiple\_stocks**

Before Normalization, notice that each of our indicators has a different range which makes it difficult to plug them into plots or learning models that is why there is a need to normalization.

The Normalized Price indicator graphs the price movement of an instrument using 100 as the base value for a user specified base date/time. The normalized value for each bar after the base date/time is the percent of the base price expressed as a whole number. (i.e. 100 times actual price divided by actual base price) This indicator shows the percentage move in price relative to some fixed starting point



Here's an overview of what we'll see in this module. Documentation links are for reference.

--> Read in multiple stocks:

- Create an empty pandas.DataFrame with dates as index: pandas.date\_range

This helps align stock data and orders it by trading date

- Drop missing date rows: pandas.DataFrame.dropna

Read in a reference stock (here SPY) and drop non-trading days using pandas.DataFrame.dropna

- Incrementally join data for each stock: pandas.DataFrame.join

--> Manipulate stock data:

- Index and select data by row (dates) and column (symbols)

- Plot multiple stocks at once (still using pandas.DataFrame.plot)

- Carry out arithmetic operations across stocks

**01-04\_Statistical\_analysis\_of\_time\_series**

Stock Indicators are measures to get an insight to trends in particular stocks over the time and compare performance with other stock prices.

Pandas makes it very convenient to compute various statistics on a dataframe:

Global statistics: mean, median, std, sum, etc.

Rolling statistics: rolling\_mean, rolling\_std, etc.

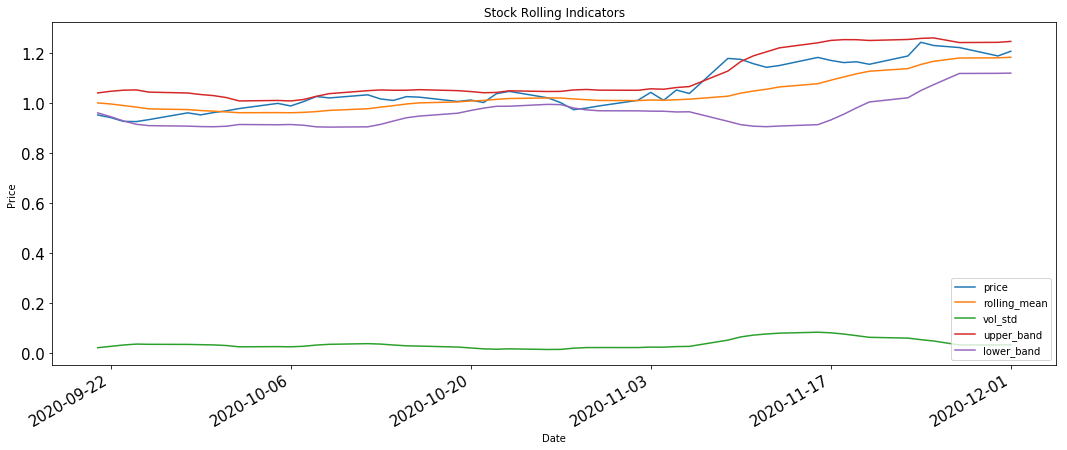
One can use these functions to analyse stock movement over time.

Focus here would be specifically,

**Bollinger Bands**: A way of quantifying how far stock price has deviated from some norm.

**Daily/Cumulative returns**: Day-to-day change in stock price and total periodic change in price.

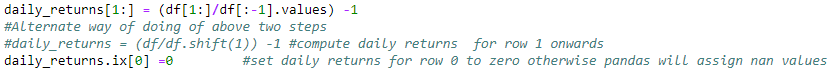
**Bollinger bands**- By using rolling std deviation above and below the global mean , one can an notion of the volatility on when the stock intersects the bands. A point of intersection , can indicate a trading opportunity. A double intersection (it dips through lower band and backs up thought it.) suggest a buy, otherwise a sell. Note: It could be misleading also, as one should not be blindly dependent on it.

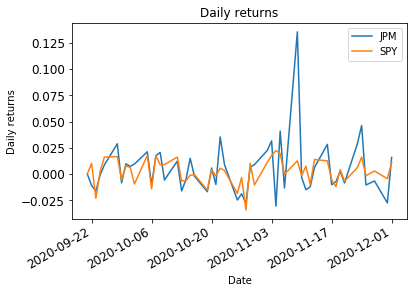


**Daily Return**

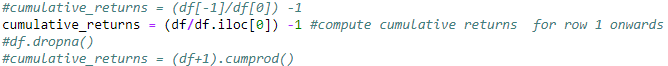
To put in simple words – how much did the price go up or down in a given day.

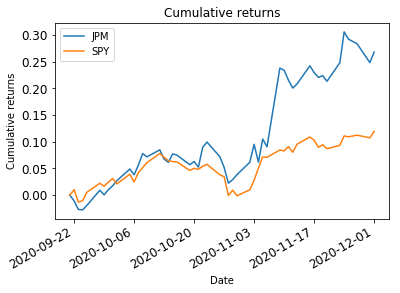
One can plot the daily return for a stock over time to get a sense of its growth trend; it will generally zig-zag around zero, though its mean will be above zero if the stock price is increasing over time and below zero otherwise.





**Cumulative Returns:** It is a measure of the stock price over a large period of time starting from some t0





**01-06\_Histograms\_and\_scatter\_plots**

**Histograms**:

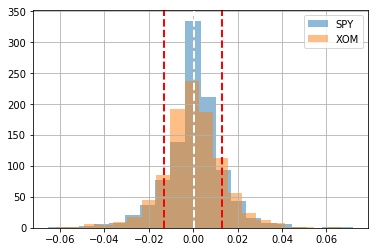
This is a bar graph that relates values to the frequency of their occurrences. For example, if there were 3 days in which one had 1% daily returns, one would have a 3-unit tall bar for x = 1%.

Below is the Normal/Gaussian Distribution histogram of SPY vs XOM normalizing it to [-1,1].

The white dashed line indicates mean and red dashed line indicates standard deviation. Standard deviation give a notion of how often and how far value deviates from the mean.

**XOM has a lower return because it has lower mean value than SPY**

**XOM has a higher volatility because it has larger standard deviation than SPY:**



**Kurtosis** is also an indicator that depicts how frequently large outliners of a histogram are present relative to a perfect normal distribution.

The great recession of 2008 is consequence of bank’s assumption that returns of bonds based on mortgages were normally distributed. Their claim based on perfect normal distribution that bonds have a very low probability of default. This epic blunder amplifies owing to actual amount of outlines present in data resulted due to massive numbers of borrowers defaulted in their home loans.

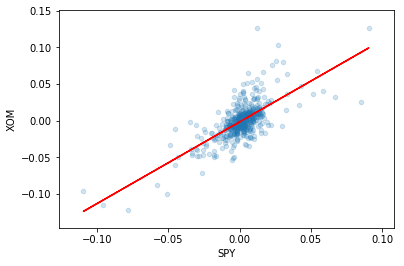
**Scatter Plot:**

A scatter plot help in visualizing differences between stock at particular point in time . Below plot shows a linear relationship. Linear Regression degree 1 is used to fit a line to the scatter plot.

Beta β -the slop of the line refers how stock reacts to the market. Here market is considered as benchmark SPY stock and stock in analysis is of XOM Exxon Mobil Corporation. If β = 1, it means that when the market goes up 1%, that stock will also go up 1%.

**Alpha** is a measure of residual risk of an investment relative to some market index. Alpha is the Y intercept of the best fit line mentioned above. An alpha of 1.0 means the investment outperformed its benchmark index by 1%. An alpha of -1.0 means the investment underperformed its benchmark index by 1%. If the alpha is zero, its return matched the benchmark.

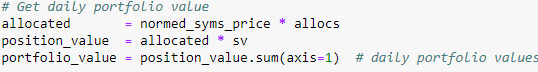
**Co-relation an another aspect which measures how tight the points are to the line and it varies [0,1]**

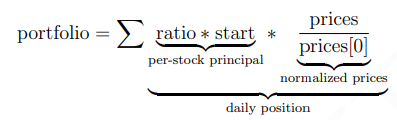


[**01-07\_Portfolio\_Assessment\_Sharpe\_ratio\_and\_other\_portfolio\_statistics**](https://viewooizm3ck72.udacity-student-workspaces.com/tree/CapstoneProject_MachineLearning_In_TradingAndInvestment/01-07_Portfolio_Assessment_Sharpe_ratio_and_other_portfolio_statistics)

Portfolios : It’s a weighted set of assets. For an example

* Principal starting capital is - $ 1000000
* Stocks = ['GOOG','AAPL','GLD','XOM']
* Allocation =[0.2,0.3,0.4,0.1]







Start Date: 2018-01-01 00:00:00

End Date: 2020-12-31 00:00:00

Symbols: ['GOOG', 'AAPL', 'GLD', 'XOM']

Allocations: [0.2, 0.3, 0.4, 0.1]

Sharpe Ratio: 1.18824855256

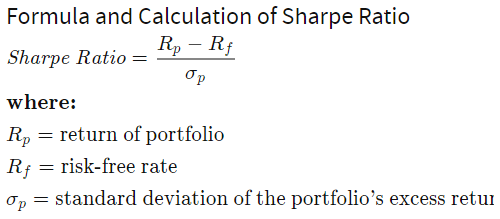
Volatility (stdev of daily returns): 0.0125579507936

Average Daily Return: 0.000939995556348

Cumulative Return: 0.916733939073

​

**Sharpe Ratio:**



The ratio is the average return earned in excess of the risk-free rate per unit of [volatility](https://www.investopedia.com/terms/v/volatility.asp) or total risk. Volatility is a measure of the price fluctuations of an asset or portfolio.

Subtracting the risk-free rate from the mean return allows an investor to better isolate the profits associated with risk-taking activities. [The risk-free rate of return](https://www.investopedia.com/articles/financial-theory/08/risk-free-rate-return.asp) is the return on an investment with zero risk, meaning it's the return investors could expect for taking no risk. The yield for a U.S. Treasury bond, for example, could be used as the risk-free rate.

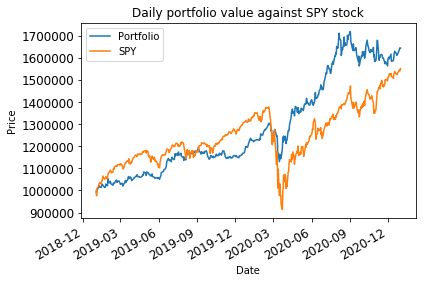
**01\_09\_Portfolio\_Optimization**

**Portfolio Optimization :**

Objective is to maximize return that could be daily return or cumulative return of the portfolio by optimizing allocation of given stocks. Instead of daily or cumulative return, Sharpe ratio is being chosen as performance parameter.

* Pick a minimize function to work with. – Sharpe ratio as performance parameter is picked for portfolio maximum return Sharpe ratio is multiplied with -1 because minimiser is used.
* Give an initial guess – Here consider equal allocation of stocks in portfolio.
* Set up ranges and constraints – For obvious reasons, one can not allocate beyond 100% of the total value.
* Call the optimiser function.

In below case leading stocks is being chosen and checked in CO-VID duration to see, which allocation would have resulted in maximum return.



IBM – International Business Machine

X- United States Steel Corporation

JPM – JP Morgan and Chase & Co

GLD – SPDR Gold trust

XOM- Exxon Mobil Corporation

AMZN- Amozon.com Inc

FB- Facebook Inc

AXY - Alterra Power Corp

Start Date: 2019-01-01 00:00:00

End Date: 2021-01-01 00:00:00

Symbols: ['IBM', 'X', 'JPM', 'GLD', 'XOM', 'AMZN', 'FB', 'AXY']

Allocations: [ 0. 0. 0.03 0.66 0. 0.26 0.04 0.01]

Sharpe Ratio: 1.68895425285

Volatility (stdev of daily returns): 0.00985025130724

Average Daily Return: 0.00104800879353

Cumulative Return: 0.656155034894

One Would have made profit if allocation is made 0.66 in GLD 0.26 in AMZN and rest in FB, JPM.

**02-04\_Capital\_Asset\_Pricing\_Model\_For\_Optimized\_Portfolio**

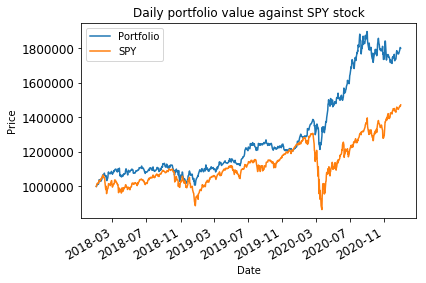
The CAPM ignite a debate about whether passive or active investment is better. Passive investing involves simply buying an index and holding it, letting the growth of the market bring in profits. Active investing is (obviously) more involved.

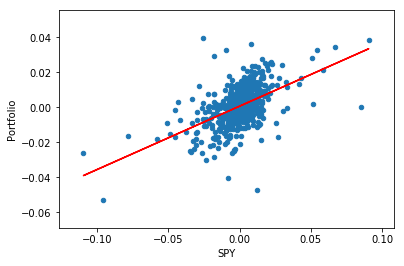
From previous section we know -

Alpha - This depicts how well it performs with respect to the comparing stock.

Beta - This denotes how much more reactive it is to the market than the comparing stock

The CAPM implies some things about a robust portfolio. In a downward market, we want β ≤ 1, so we don’t lose as much money as the market in general. Contrarily, we want β ≥ 1 in an upward market to outperform just tracking the market





Start Date: 2018-01-01 00:00:00

End Date: 2021-01-01 00:00:00

Symbols: ['AMZN', 'FB', 'AXY', 'GLD']

Allocations: [ 0.28 0. 0. 0.72]

Sharpe Ratio: 1.31914313069

Volatility (stdev of daily returns): 0.00994544335521

Average Daily Return: 0.000826448504365

Cumulative Return: 2018-01-02 0.000000

beta\_portfolio= 0.362138572388

alpha\_portfolio= 0.000604546822613

**BENCHMARK MODEL**

[**03-02\_Supervised\_Regression\_Price\_Prediction**](https://viewooizm3ck72.udacity-student-workspaces.com/tree/CapstoneProject_MachineLearning_In_TradingAndInvestment/03-02_Supervised_Regression_Price_Prediction)

There are many machine learning algorithm in supervised regression

-Linear Regression,

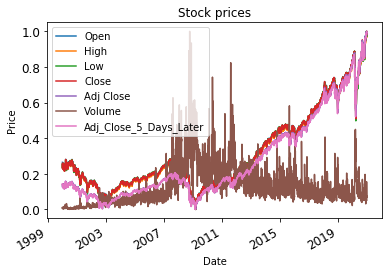
- K-nearest Neighbour(kNN)

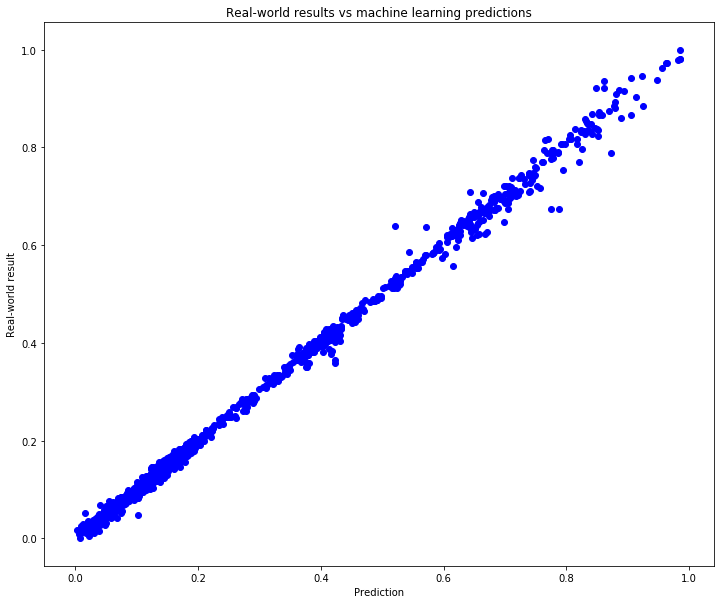
- Decision Tree and Decision Forest.

In supervised parametric model, First apply Linear regression where in one chose parameters to a line (m, the slope, and b, the y-intercept from the familiar equation of a line, y = mx + b) that best fit the data set. This “best fit” is minimizing the total vertical error between the line and the points.

In given case, create a dataframe or a feature in focus Adj\_Close Price -this is the column data for a particular stock and each row is the data at a specific point in time. For learning our model we create a test data such that we shifted the closing price by 5 days . That will become future price for our historical data.



****

****

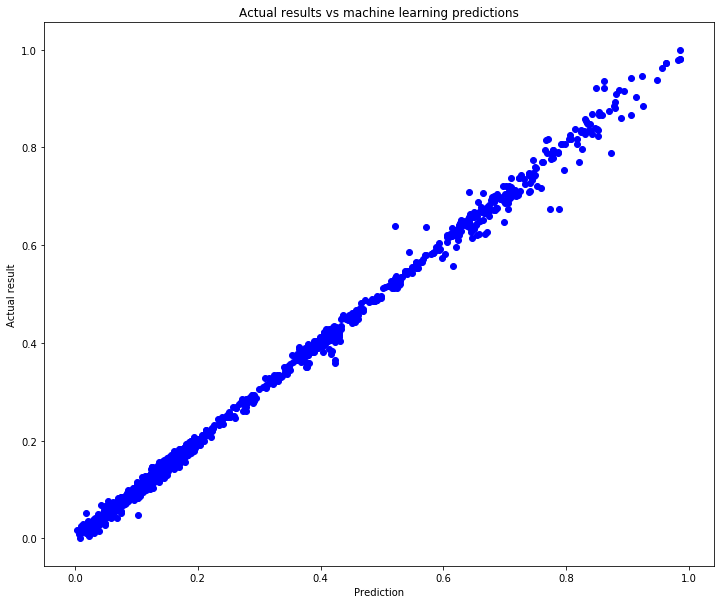
Score on training data

regr.score(X\_train, y\_train): 0.99744158312

Score on testing (unseen) data

regr.score(X\_test, y\_test): 1.00

Mean squared error: 0.000142232150903

****

Score - variance between prediction and actual results

linreg\_bagging.score(X\_test, y\_test): 1.00

Mean squared error: 0.000141811721453

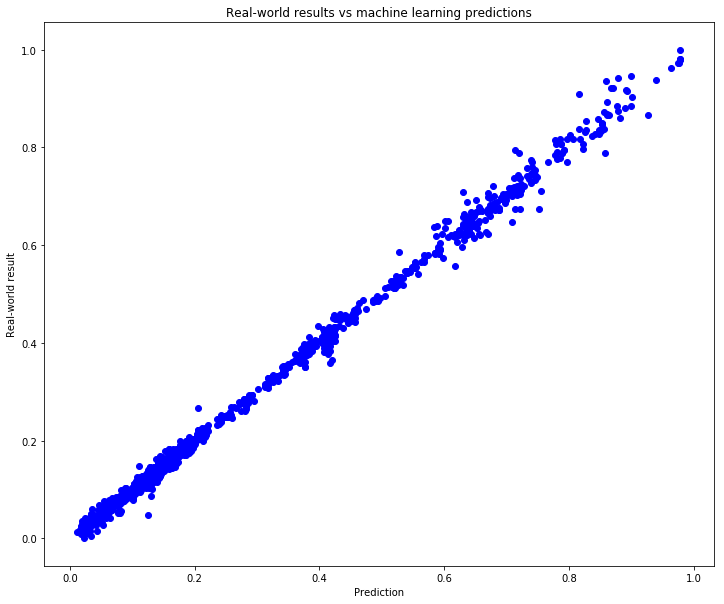
**K-Nearest Neighbour-**

It’s a data centric approach uses examples of data points to make an estimated prediction based on the distance of 1, 2, k of its nearest neighbour. This method has drawback that it take up lot of space.

Training KNeighborsRegressor...

best parameter: {'leaf\_size': 1, 'n\_neighbors': 15, 'weights': 'distance'}

best score: 0.997111509593



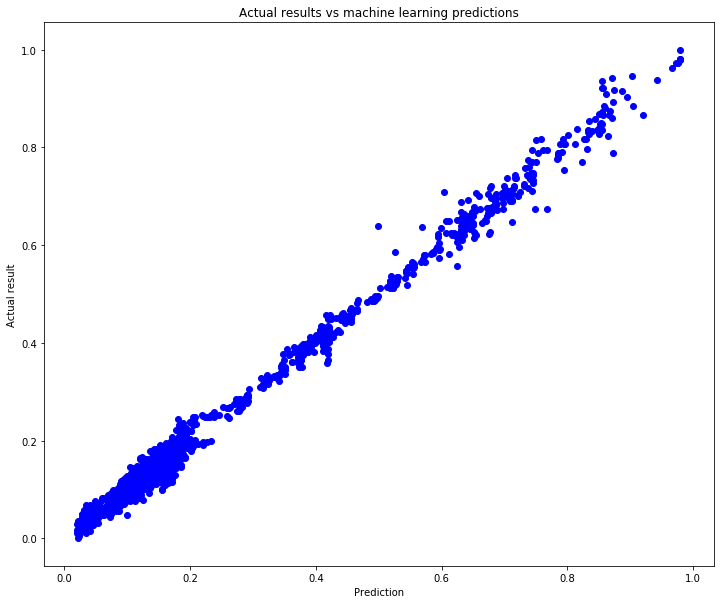
Score on training data

regr.score(X\_train, y\_train): 0.99805085107

Score on testing (unseen) data

regr.score(X\_test, y\_test): 1.00

Mean squared error: 0.000159780378048



Score - variance between prediction and actual results

linreg\_bagging.score(X\_test, y\_test): 0.99

Mean squared error: 0.000352615008086

**04-01\_Anomalies\_Detection**

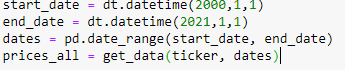
Long Short-Term Memory Autoencoder a method comes under un-supervised learning which inturn based on training using supervised learning methods. In below example , Attempt is to bring out anomalies in Stock benchmark prices SPY historical time series data.

Approach is to observe the data for a given duration , and consider that as the normal .

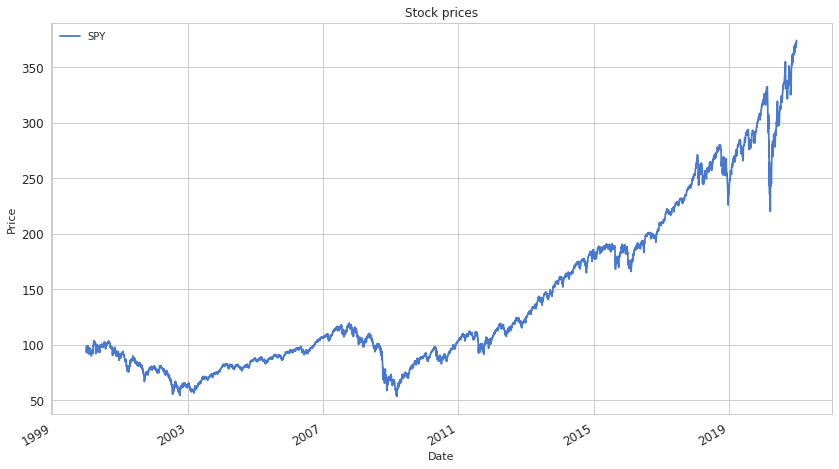
Utilize LSTM autoencoder on test data .

Name the data point as Anomaly if they breach threshold .

### Load and Inspect the data

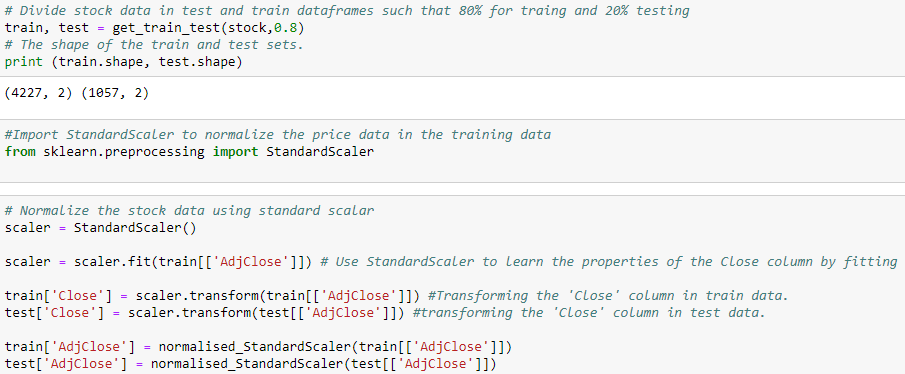


Visualize the SPY Stock Benchmark Data

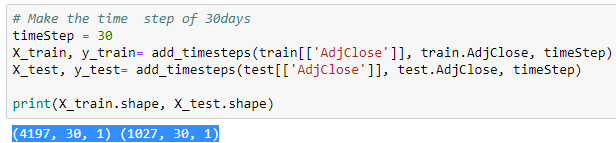


Data Pre-processing

* Train test split
* Apply Standard Scaler

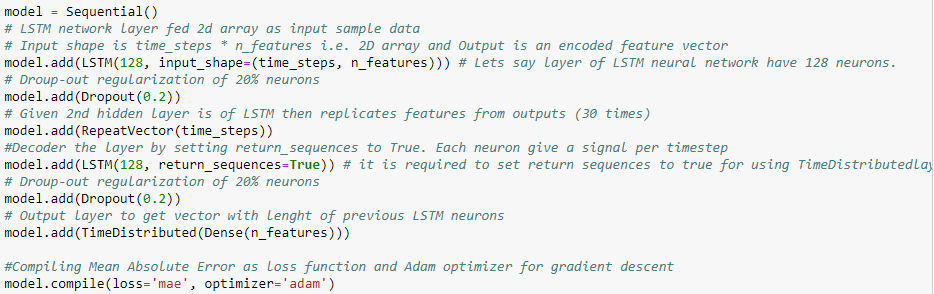


### Temporalize Data and Create Training and Test Splits

* Input data is converted into an 3d array and requirement is to have memory of 30 days in our network. That is why TIME\_STEP is taken as 30 days.
* 

**Build LSTM Auto-encoder Model**

In the reconstruction LSTM Autoencoder architecture, the parameters which are considered are input sequences with time steps (30) and one feature . The output is a sequence with time steps(30) and one feature.



Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

lstm (LSTM) (None, 128) 66560

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout (Dropout) (None, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

repeat\_vector (RepeatVector) (None, 30, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_1 (LSTM) (None, 30, 128) 131584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 30, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed (TimeDistri (None, 30, 1) 129

=================================================================

Total params: 198,273

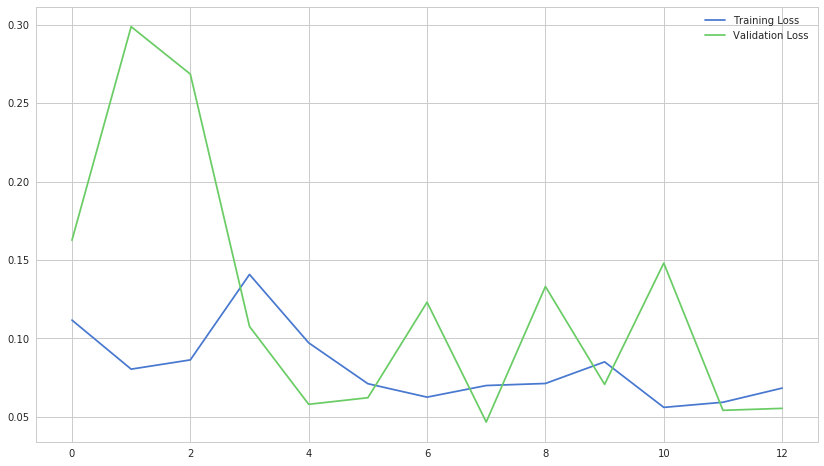
Trainable params: 198,273

Non-trainable params: 0

**Train Model**

### Plot Metrics and Evaluate the Model - If validation loss is lower than Training loss, it means the model is under fitting.

### - If validation loss is higher than Training loss, it means the model is overfitting,

****

**Mean Absolute Error Loss**

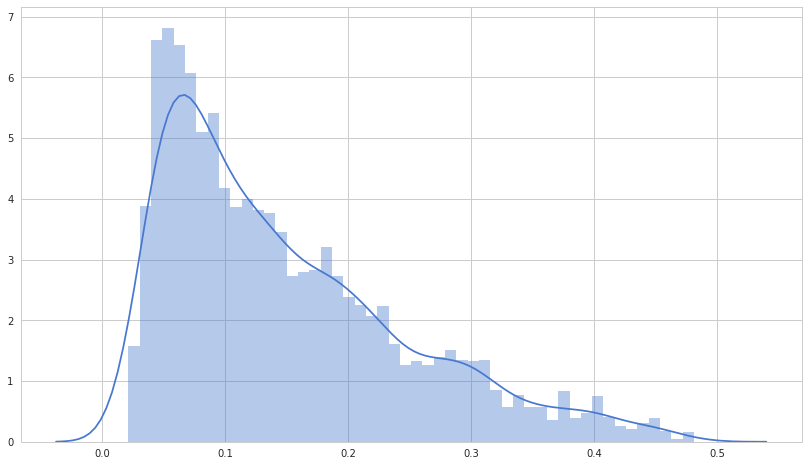
This is the sum of absolute differences between target and predicted variables. It measures the average magnitude of errors in a set of predictions , without considering their directions.

**Reconstruction Error Threshold :**

Lets set a threshold for the MAE Loss. Threshold is the value that decides a data point is an anomaly or not. If the reconstruction loss for a data point in the test set is greater than this reconstruction error threshold value then we will label this data point as an anomaly.

threshold = 0.55

With the **threshold**, the normal data was filtered and the model was retrained with the filtered dataset to obtain the final and optimal **threshold**. The **reconstruction errors** from the training data will be used obtain an optimal **threshold.**

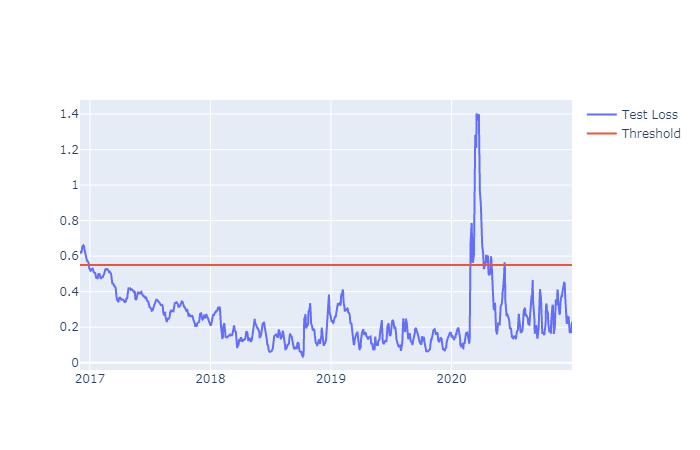


**TestLoss Vs Threshold**

Lets set a threshold for the MAE Loss. Theshold is the value that decides a data point is an anomaly or not.

If the reconstruction loss for a data point in the test set is greater than this reconstruction error threshold value then we will label this data point as an anomaly.

threshold = 0.55



**Plot Anomalies**

#### C:\Users\gargrc\AppData\Local\Temp\Downloads\CapstoneProject_MachineLearning_In_TradingAndInvestment\04-01_Anomalies_Detection\Anomalies On Plot.png

#### It was well observed that the SPY stock benchmark hit a 2020 low in March owing to COVID-19 pandemic uncertainties, but quickly reaccelerated to a high point later on bullish expectations with coronavirus vaccine news and lock-down relaxations.

### 05-01\_LongShortTermMemory\_PricePrediction

**LSTM Recurrent Neural Network**

Long-Short-Term Memory Recurrent Neural Network belongs to the family of deep learning algorithms.. It has an advantage over traditional neural networks due to its capability to process the entire sequence of data. LSTM models are extremely powerful time-series models. They can predict an arbitrary number of steps into the future. An LSTM module (or cell) has 5 essential components which allows it to model both long-term and short-term data.

The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell. The cell of the model is responsible for keeping track of the dependencies between the elements in the input sequence.

Cell state (ct) - This represents the internal memory of the cell which stores both short term memory and long-term memories

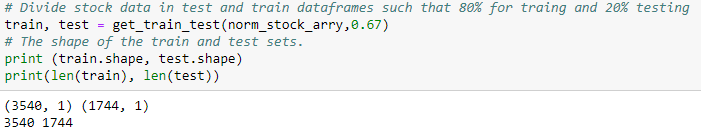
Hidden state (ht) - This is output state information calculated w.r.t. current input, previous hidden state and current cell input which you eventually use to predict the future stock market prices. Additionally, the hidden state can decide to only retrieve the short or long-term or both types of memory stored in the cell state to make the next prediction.

Input gate (it) - Decides how much information from current input flows to the cell state

Forget gate (ft) - Decides how much information from the current input and the previous cell state flows into the current cell state

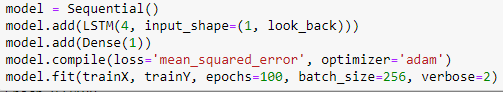
Output gate (ot) - Decides how much information from the current cell state flows into the hidden state, so that if needed LSTM can only pick the long-term memories or short-term memories and long-term memories

### Data Pre-processing

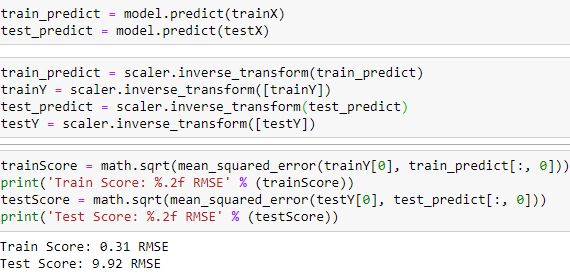


### Implementation

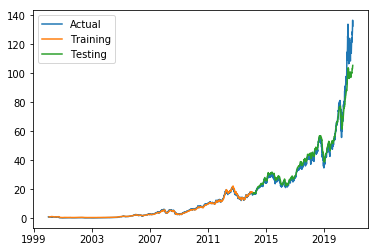
### Build LSTM Model



**Results**



### Visualize the Prediction



**Justification**With an attempt to predict the stock price has not been proven accurate the error score on testing data is very high as compared to train data. For refinement one can play with hyper parameters of LSTM model. But in above case chosen parameters were not proven fruitful

**Conclusion**Although LSTM did not prove for me to accurately predict future stock prices however , it helped to takeout the outliners from the stock time series data. This project laid a stepping stone towards my journey in learning *machine learning*