





Intensive Programs on Communications and Information Technology (3 Months)

Track

Big Data Science

Project Name

Stock Price Prediction

December 2018

**Project Title**

**Stock Price Prediction   
by Algorithms Trading**

**Team member**

Eng. Moahmed Saber

Eng. Amir Zaghloul

Eng. Yara Sabry

Eng. Amr Abd Alaziz

**Under The Supervision of**

Dr. Doaa Hassan

Eng. Sayed Omer

Acknowledgements:

At the beginning and at the end we all thank Allah for helping us to achieve this work and ask Allah to benefit us with what he taught us and teach us what will benefit us. Also the achievement of this piece of work wouldn’t be to see the light without the support, help and contribution of: Dr. Doaa Hassan the acting head of the computer science and information system department for his support and advice. Eng. Omer Sayed supervisor of our graduation project. All the faculty members who taught us during our study in the Intensive Programs on Communications and Information Technology (3 Months) in Track Big Data Science, Finally we thank our families for supporting us and helped us to reach so far and to have the tolerance to face difficulties and keep up.

Abstract:

The project proposes a machine learning model to predict stock market price, Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on a financial exchange. The successful prediction of a stock's future price will maximize investor’s gains. So as a chaotic system, the stock market contains patterns that can be analyzed with models. Algorithms can be designed to make predictions about the stock market by (ARIMA model), but It's better in this direction using (LSTM) and (RNN by using FBProphet Tools facebook).we can't implement these techniques because we have a limited time but don't mind trying. so we will use the main technique at the beginning like (ARIMA) in time series forecast.

- Feature Extraction and selection (Technical Indicators)

- Data Acquisition and Preprocessing (Stocks Historical data

- Testing model with new data optimizing and training

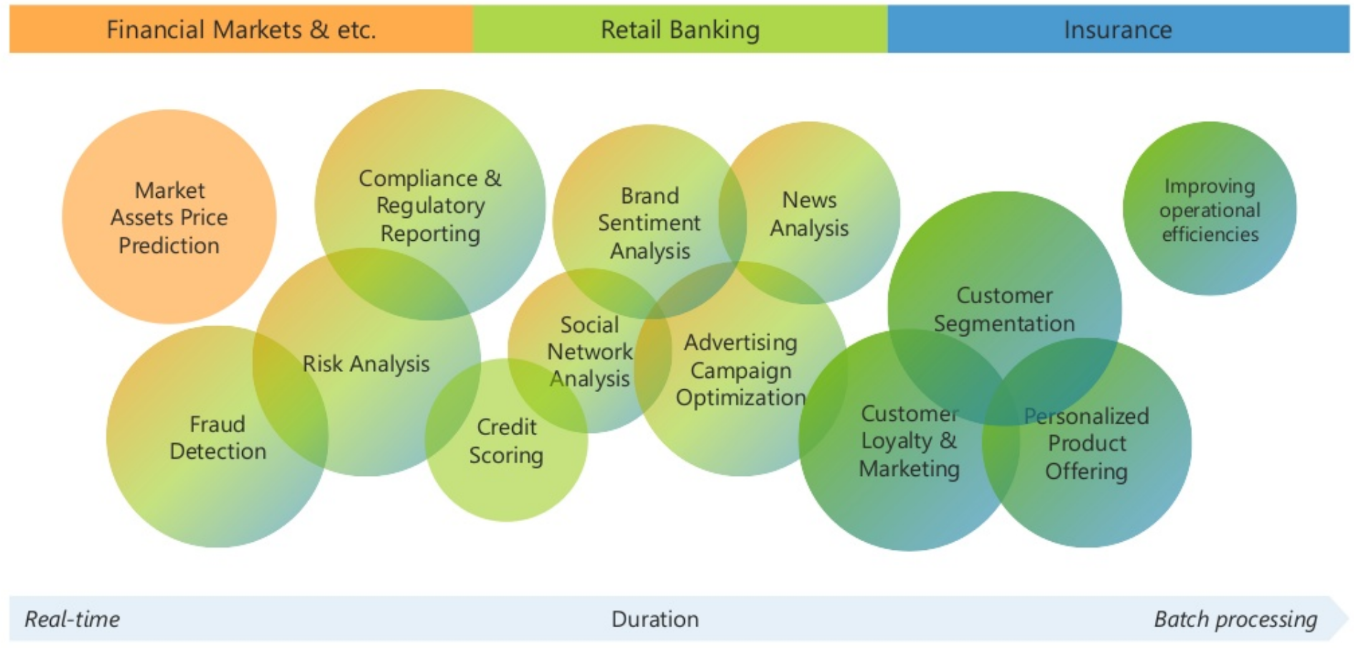
|  |  |  |
| --- | --- | --- |
| List of Figures | | |
| Figure 1 | Finance current Applications | 7 |
| Figure 2 | Technical analysis - Charting | 9 |
| Figure 3 | Technological methods - Machine Learning | 10 |
| Figure 4 | ARIMA Model for forecasting stock price in R | 12 |
| Figure 5 | Seasonal trend model | 13 |
| Figure 6 | Stationary series | 13 |
| Figure 7 | Long term trend with cyclical and seasonal | 14 |
| Figure 8 | Box-Jenkins Modeling Approach | 16 |
| Figure 9 | Framework architecture | 16 |
| Figure 10 | Visualization data frame and Candlestick plot | 17 |
| Figure 11 | Compare stocks with other companies and Spyder | 18 |
| Figure 12 | Non-Station and not Trend | 18 |
| Figure 13 | Moving Averages | 18 |
| Figure 14 | Interface Stock price by Shiny API | 25 |

|  |  |  |
| --- | --- | --- |
| List of Tables | | |
| Table 1 | **Symbol Dataset** |  |
| Table 2 | **Load Data about Apple company (AAPL)** | 17 |
| Table 3 | **Evaluation Algorithms Table** |  |

Table of Content

|  |  |  |  |
| --- | --- | --- | --- |
| Chapters | | | Page |
| Chapter One: Introduction | | |  |
| 1.1 | Machine Learning in Finance. | | 9 |
| 1.2 | Stock price prediction. | | 10 |
| 1.3 | Prediction methods | | 10 |
| 1.4 | Algorithmic Trading concept | | 13 |
| Chapter Two: Background and Theoretical Overview | | |  |
| 2.1 | ARIMA Model for forecasting Stock Price | | 14 |
|  | 2.1.1 | Auto Regression (AR) | 15 |
| 2.1.2 | Differencing (I) | 15 |
| 2.1.3 | Moving Average (MA) | 15 |
| Chapter Three: Proposed Project Implementation | | |  |
| 3.1 | Framework architecture | | 18 |
| 3.2 | Analysis of stock price data by Python | | 18 |
| 3.3 | Predict of stock price data by R | | 22 |
| 3.4 | Stock price by Shiny API | | 25 |
| 3.5 | Big Data by PySpark: | | 29 |
| Chapter Four: Conclusions | | | 32 |
| Reference | | | 33 |
|  | | | |

**Chapter 1 Introduction**

**1.1 Machine Learning in Finance**

**Figure (1)**

Machine learning has had fruitful applications in finance well before the advent of mobile banking apps, proficient chat bots, or search engines. Given high volume, accurate historical records, and quantitative nature of the finance world, few industries are better suited for artificial intelligence. There are more uses cases of machine learning in finance than ever before, a trend perpetuated by more accessible computing power and more accessible machine learning tools (such as Google’s Tensorflow). Today, machine learning has come to play an integral role in many phases of the financial ecosystem, from approving loans, to managing assets, to assessing risks. Yet, few technically-savvy professionals have an accurate view of just how many ways machine learning finds its way into their daily financial lives.

**Machine Learning in Finance current Applications:**

Portfolio Management Algorithmic Trading  
Fraud Detection Loan / Insurance Underwriting  
Customer Service Security financial  
Sentiment / News Analysis  
Sales / Recommendations of Financial Products

**1.2 Stock price prediction intro.**

Stock price prediction is the act of trying to determine the future value of a company [stock](https://www.wikiwand.com/en/Stock) or other instrument traded on an [exchange](https://www.wikiwand.com/en/Exchange_(organized_market)). The successful prediction of a stock's future price could yield significant profit. The [efficient-market hypothesis](https://www.wikiwand.com/en/Efficient-market_hypothesis) suggests that stock prices reflect all currently available information and any price changes that are not based on newly revealed information thus are inherently unpredictable. Others disagree and those with this viewpoint possess myriad methods and technologies which purportedly allow them to gain future price information.

While the [efficient market hypothesis](https://www.wikiwand.com/en/Efficient_market_hypothesis) finds favor among financial academics, its critics point to instances in which actual market experience differs from the prediction-of-unpredictability the hypothesis implies. A large industry has grown up around the implication proposition that some analysts can predict stocks better than others; ironically that would be impossible under the Efficient Markets Hypothesis if the stock prediction industry did not offer something its customers believed to be of value.

**1.3 Prediction methods.**

Prediction methodologies fall into three broad categories which can (and often do) overlap. They are fellow:

[**Fundamental** **analysis**](https://www.wikiwand.com/en/Fundamental_analysis)

[**Technical analysis**](https://www.wikiwand.com/en/Technical_analysis) (charting)

**Technological** **methods** (Machine Learning)

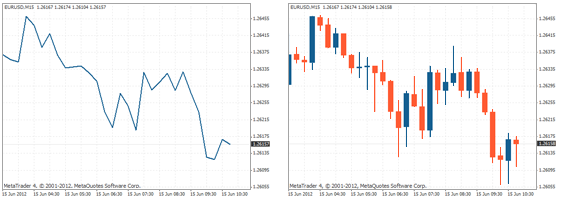
**1.3.1 Fundamental analysis:**



Fundamental Analysts are concerned with the company that underlies the stock itself. They evaluate a company's past performance as well as the credibility of its [accounts](https://www.wikiwand.com/en/Account_(accountancy)). Many performance ratios are created that aid the fundamental analyst with assessing the validity of a stock, such as the [P/E ratio](https://www.wikiwand.com/en/P/E_ratio). [Warren Buffett](https://www.wikiwand.com/en/Warren_Buffett) is perhaps the most famous of all Fundamental Analysts.

Fundamental analysis is built on the belief that human society needs capital to make progress and if a company operates well, it should be rewarded with additional capital and result in a surge in stock price. Fundamental analysis is widely used by fund managers as it is the most reasonable, objective and made from publicly available information like **financial statement analysis.**

Another meaning of fundamental analysis is [beyond](https://www.wikiwand.com/en/Text_mining) bottom-up company analysis, it refers to top-down analysis from first analyzing the global economy, followed by country analysis and then sector analysis, and finally the company level analysis.

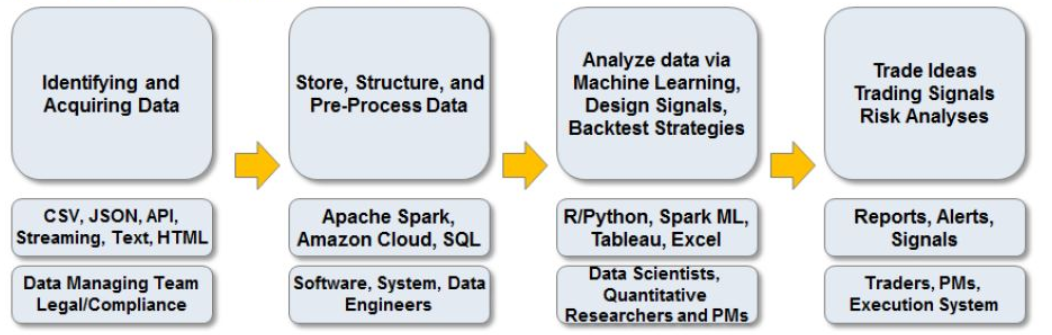
**1.3.2 Technical analysis (Charting)**

**Figure (2)**

Technical analysts or chartists are not concerned with any of the company's fundamentals. They seek to determine the future price of a stock based solely on the trends of the past price (a form of [**time series analysis**](https://www.wikiwand.com/en/Time_series_analysis)**).** Numerous patterns are employed such as the [head and shoulders](https://www.wikiwand.com/en/Head_and_shoulders_(chart_pattern)) or cup and saucer. Alongside the patterns, techniques are used such as the [exponential **moving average**](https://www.wikiwand.com/en/Exponential_moving_average) (EMA). **Candle stick** patterns, believed to have been first developed by Japanese rice merchants, are nowadays widely used by technical analysts.

**1.3.3 Technological** **methods (Machine Learning)**

With the advent of the [digital computer](https://www.wikiwand.com/en/Digital_computer), stock market prediction has since moved into the technological realm. The most prominent technique involves the use of [**artificial neural networks**](https://www.wikiwand.com/en/Artificial_neural_networks) (ANNs) and **Genetic Algorithms** (GA). Scholars found bacterial chemotaxis optimization method may perform better than GA. ANNs can be thought of as [mathematical function](https://www.wikiwand.com/en/Mathematical_function) approximations. The most common form of ANN in use for stock market prediction is the [feed forward network](https://www.wikiwand.com/en/Feedforward_neural_network) utilizing the [backward propagation of errors](https://www.wikiwand.com/en/Backpropagation) algorithm to update the network weights. These networks are commonly referred to as [**Backpropagation**](https://www.wikiwand.com/en/Backpropagation)**networks**. Another form of ANN that is more appropriate for stock prediction is the time [**recurrent neural network**](https://www.wikiwand.com/en/Recurrent_neural_network) (RNN) or [**time delay neural network**](https://www.wikiwand.com/en/Time_delay_neural_network) (TDNN). Examples of RNN and TDNN are the Elman, Jordan, and Elman-Jordan networks.



**Figure (3)**

For stock prediction with ANNs, there are usually two approaches taken for **forecasting different time horizons:** independent and joint. The independent approach employs a single ANN for each time horizon, for example, 1-day, 2-day, or 5-day. The advantage of this approach is that network forecasting error for one horizon won't impact the error for another horizon since each time horizon is typically a unique problem. The joint approach, however, incorporates multiple time horizons together so that they are determined simultaneously. In this approach, forecasting error for one time horizon may share its error with that of another horizon, which can decrease performance. There are also more parameters required for a joint model, which increases the risk of overfitting. of late, the majority of academic research groups studying ANNs for stock forecasting seem to be using an ensemble of independent ANNs methods more frequently, with greater success. An ensemble of ANNs would use low price and time lags to predict future lows, while another network would use lagged highs to predict future highs. The predicted low and high predictions are then used to form stop prices for buying or selling. Outputs from the individual "low" and "high" networks can also be input into a final network that would also incorporate volume, intermarket data or statistical summaries of prices, leading to a final ensemble output that would trigger buying, selling, or market directional change. A major finding with ANNs and stock prediction is that a classification approach (vs. function approximation) using outputs in the form of buy(y=+1) and sell(y=-1) results in better predictive reliability than a quantitative output such as low or high price. Since NNs require training and can have a large parameter space; it is useful to optimize the network for optimal predictive ability.

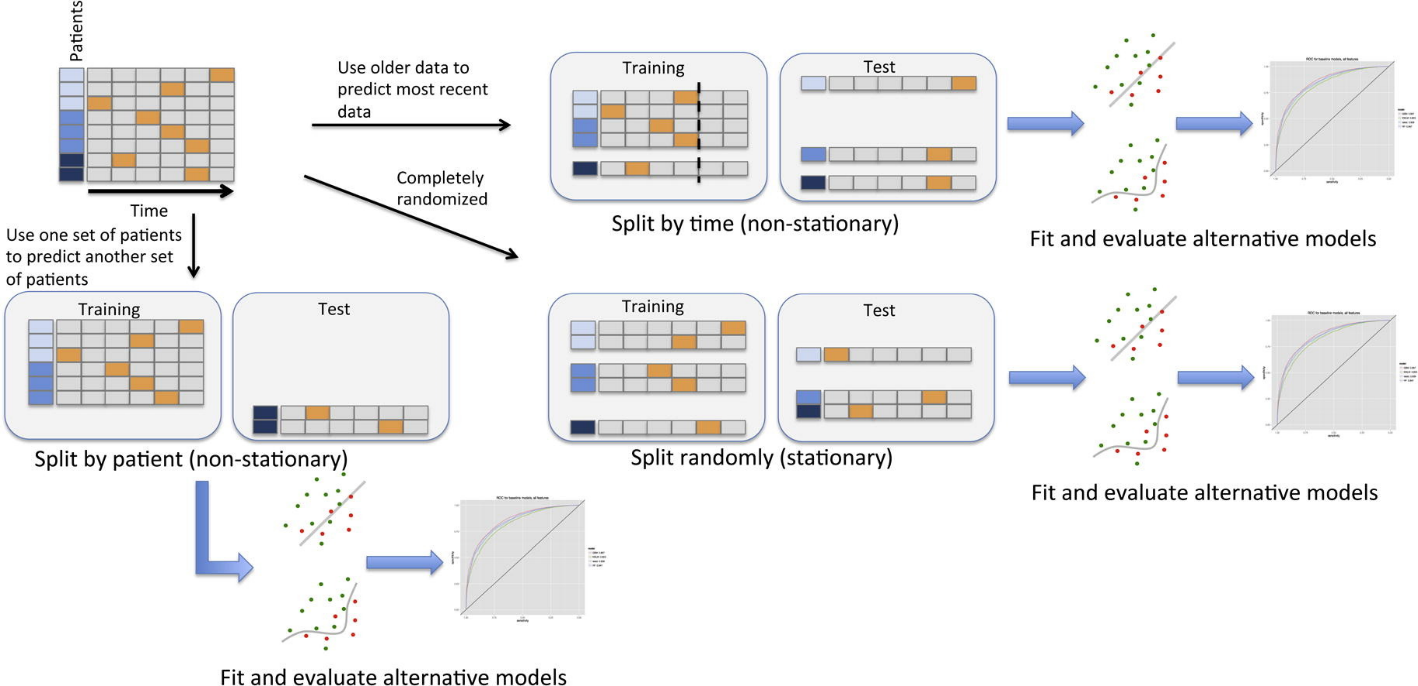
**1.4 Algorithmic Trading concept**

Algorithmic trading (automated trading, black-box trading or simply algo-trading) is the process of using computers programed to follow a defined set of instructions (an algorithm) for placing a trade in order to generate profits at a speed and frequency that is impossible for a human trader. The defined sets of rules are based on timing, price, quantity or any mathematical model. Apart from profit opportunities for the trader, algo-trading makes markets more liquid and makes trading more systematic by ruling out the impact of human emotions on trading activities.

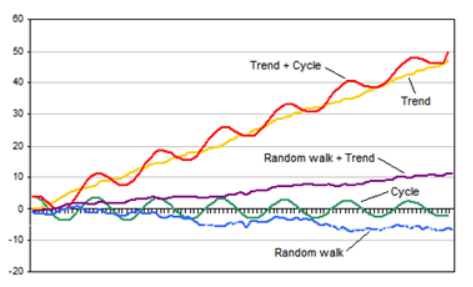
**Algorithmic Trading Strategies:**

1. Trend-following Strategies
2. Arbitrage Opportunities
3. Mathematical Model Based Strategies
4. Trading Range (Mean Reversion)
5. Volume Weighted Average Price (VWAP)
6. Time Weighted Average Price (TWAP)
7. Percentage of Volume (POV)
8. Implementation Shortfall

**Chapter 2 Background and Theoretical Overview**

**2.1 ARIMA Model for forecasting stock price in r (New Approached)**

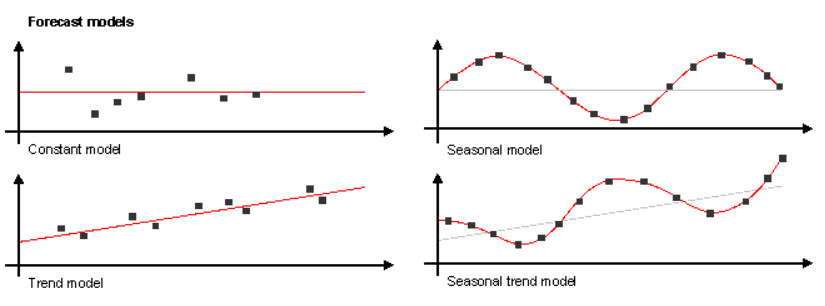
**Figure (4)**

Forecasting involves predicting values for a variable using its historical data points or it can also involve predicting the change in one variable given the change in the value of another variable. Forecasting approaches are primarily categorized into qualitative forecasting and quantitative forecasting. Time series forecasting falls under the category of quantitative forecasting wherein statistical principals and concepts are applied to a given historical data of a variable to forecast the future values of the same variable. Some time series forecasting techniques used **include**:

* Autoregressive Models (AR)
* Moving Average Models (MA)
* Seasonal Regression Models
* Distributed Lags Models

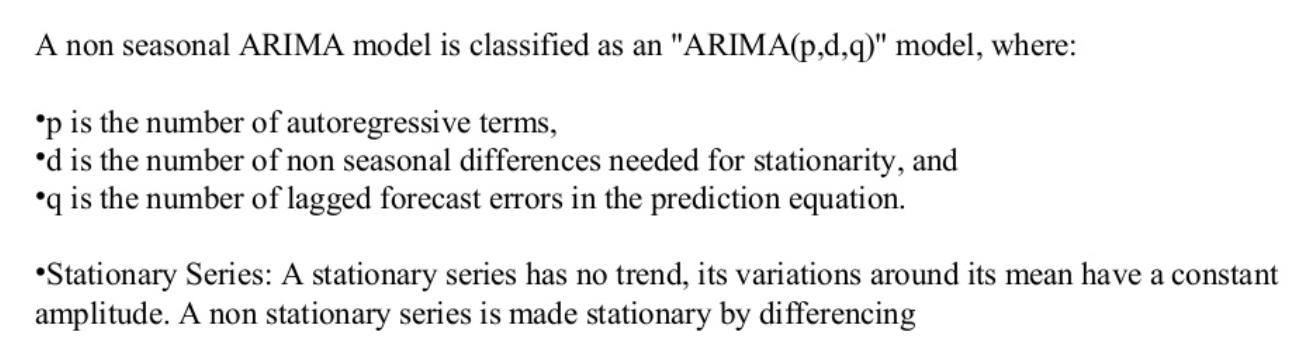
ARIMA stands for Autoregressive Integrated Moving Average. ARIMA is also known as Box-Jenkins approach. Box and Jenkins claimed that non-stationary data can be made stationary by differencing the series, . The general model for  is written as

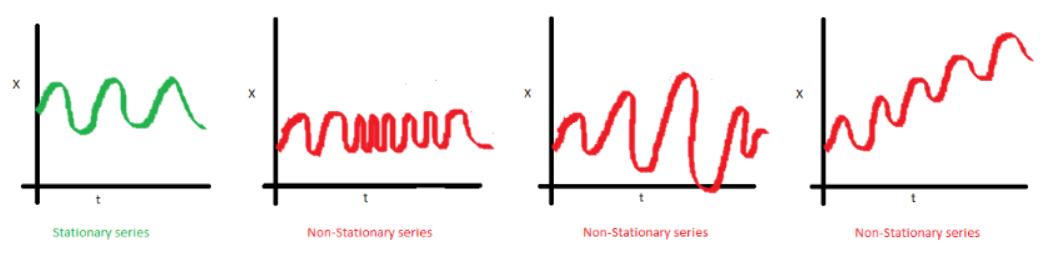


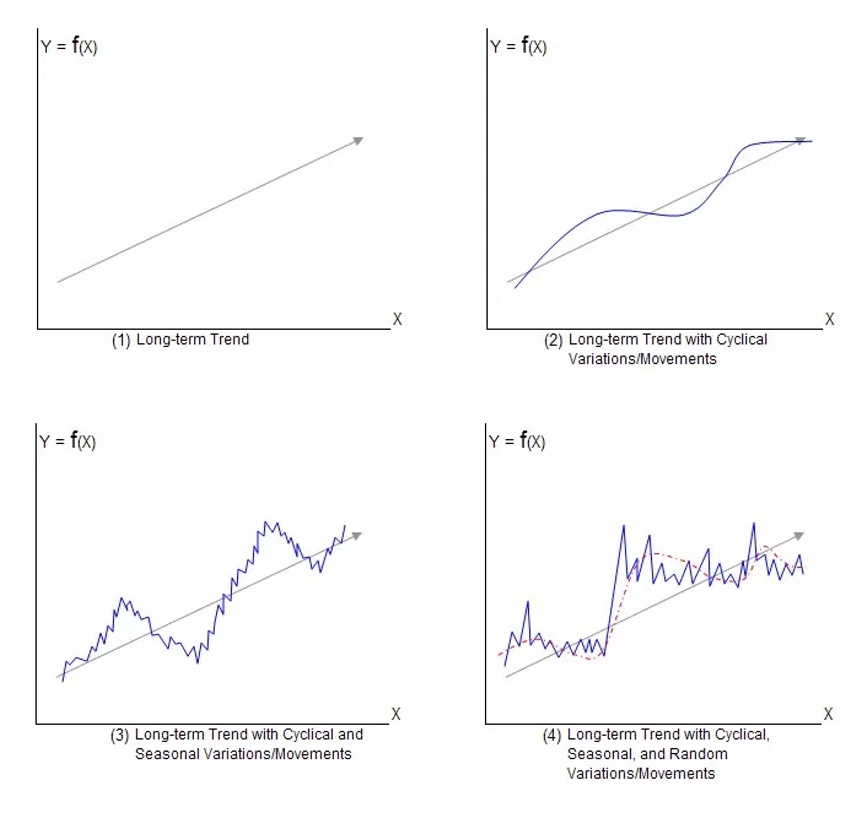
Where,  is the differenced time series value, ϕ and θ are unknown parameters and ϵ are independent identically distributed error terms with zero mean. Here,  is expressed in terms of its past values and the current and past values of error terms.

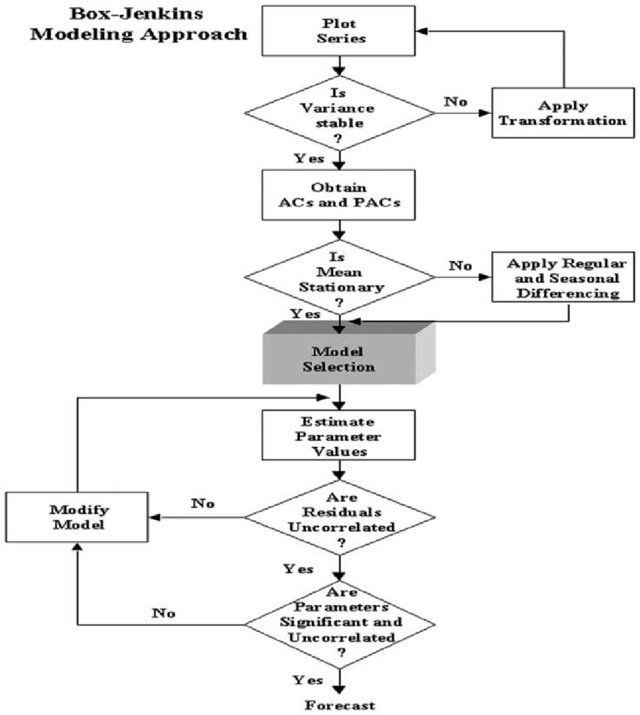
**The ARIMA model combines three basic methods:**

* **Auto Regression (AR)** – In auto-regression the values of a given time series data are regressed on their own lagged values, which is indicated by the “p” value in the model.
* **Differencing (I-for Integrated)** – This involves differencing the time series data to remove the trend and convert a non-stationary time series to a stationary one. This is indicated by the “d” value in the model. If d = 1, it looks at the difference between two time series entries, if d = 2 it looks at the differences of the differences obtained at d =1, and so forth.
* **Moving Average (MA)** – The moving average nature of the model is represented by the “q” value which is the number of lagged values of the error term.

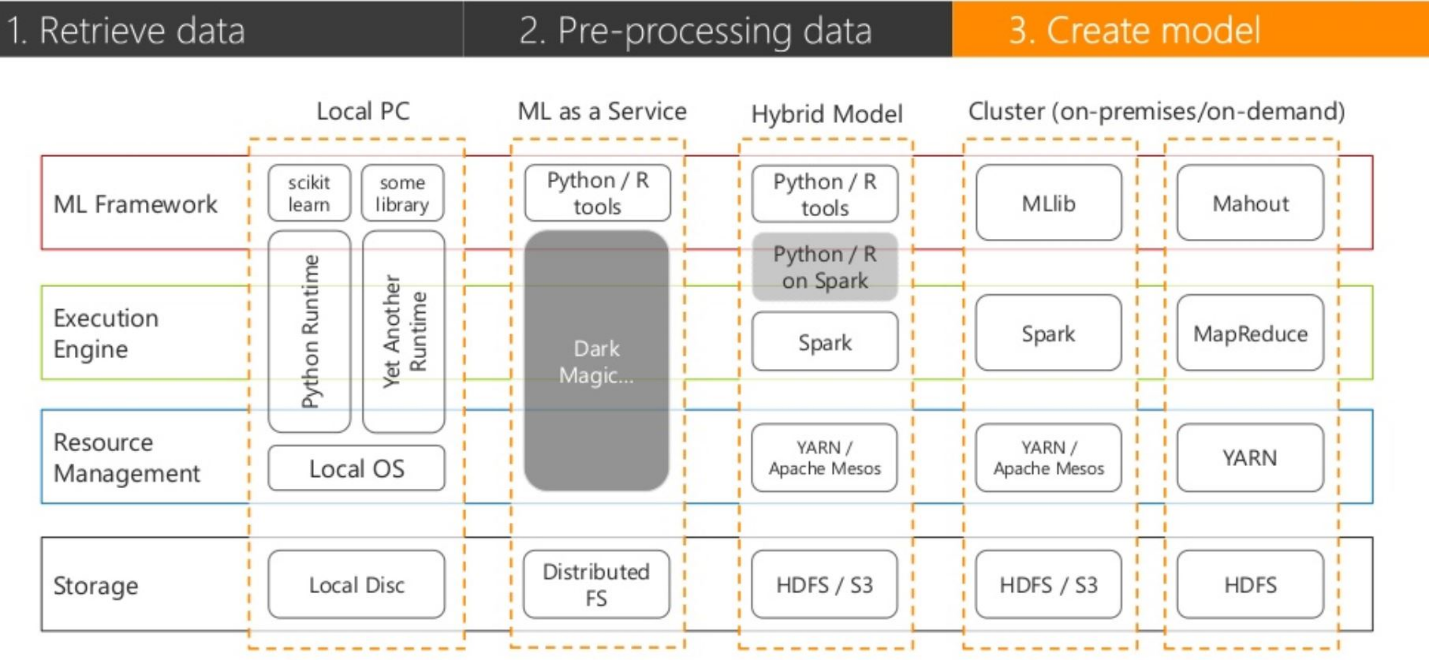






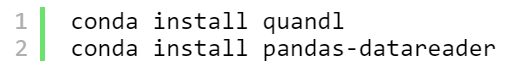


**Chapter 3 Proposed Project Implementation**

**3.1 Framework architecture**

**3.2 Analysis of stock price data by Python**

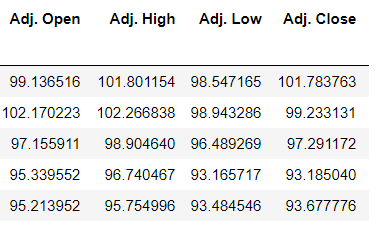
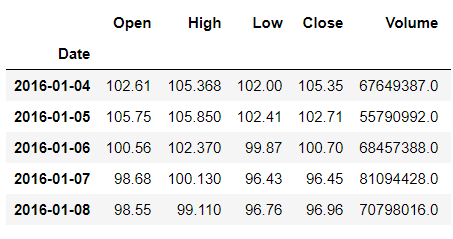
I will be using two packages, **quandl** and **pandas\_datareader**, which are not installed with [Anaconda](https://www.anaconda.com/) if you are using it. To install these packages, run the following at the appropriate command prompt:



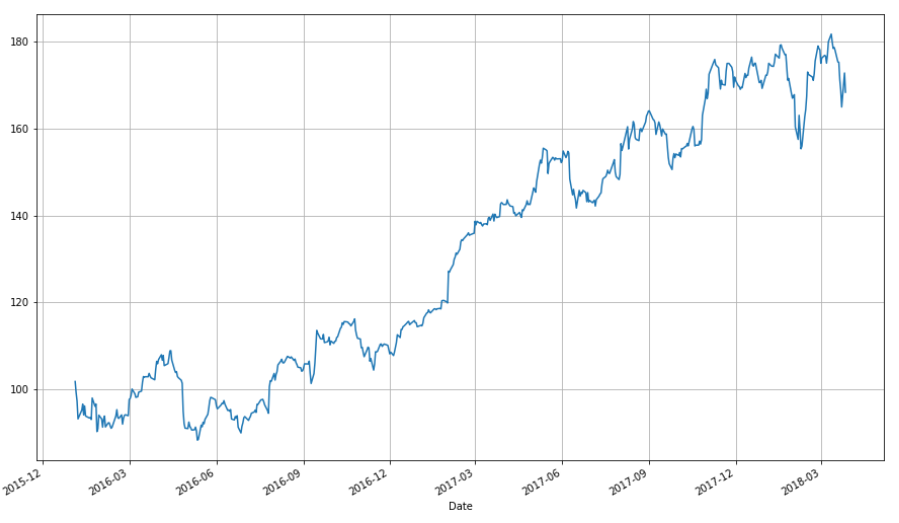
**Getting Data from Quandl**

Before we analyze stock data, we need to get it into some workable format. Stock data can be obtained from [**Yahoo! Finance**](http://finance.yahoo.com/), [**Google Finance**](http://finance.google.com/), or a number of other sources. These days I recommend getting data from [**Quandl**](https://www.quandl.com/), a provider of community-maintained financial and economic data. (**Yahoo! Finance** used to be the go-to source for good quality stock data

**Load Data (AAPL)**

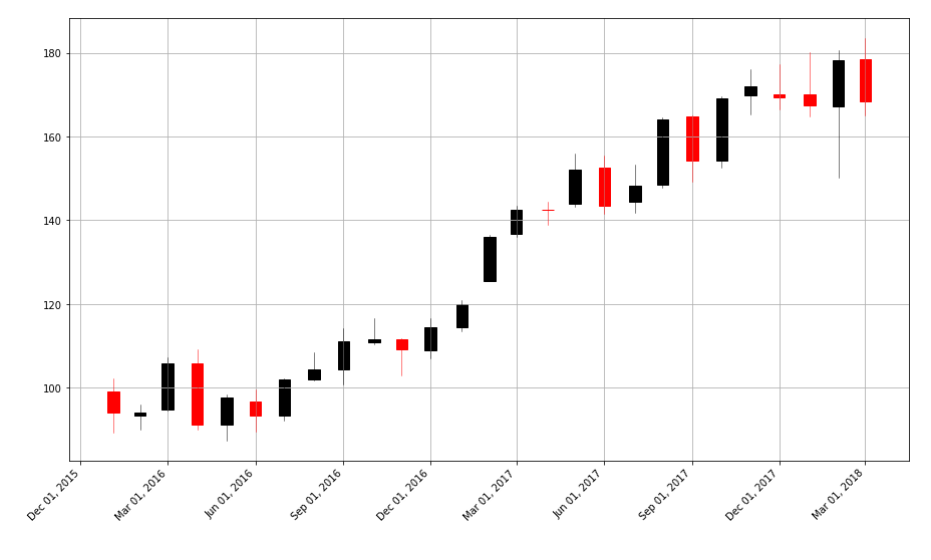


**Visualizing Stock Data**

Now that we have stock data we would like to visualize it. I first demonstrate how to do so using the matplotlib package. Notice that the apple DataFrame object has a convenience method, plot(), which makes creating plots easier.

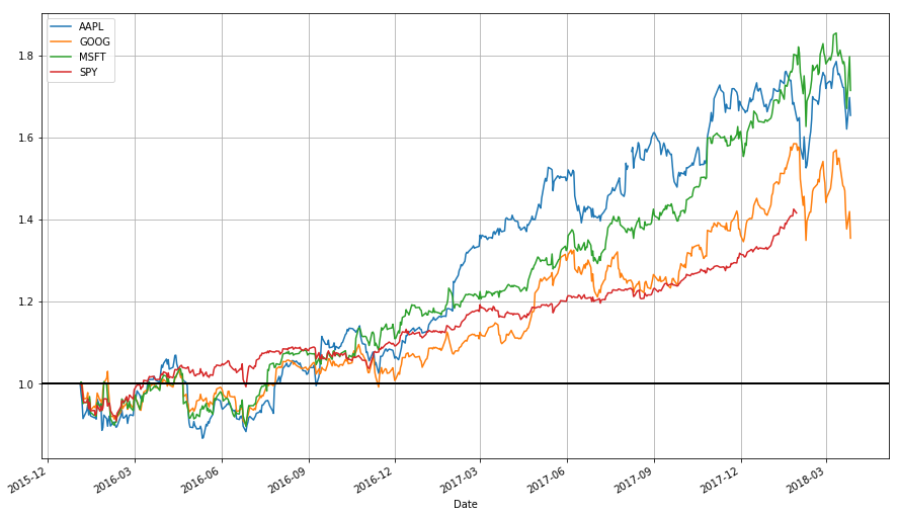
**Candlestick Plot**

A line chart is fine, but there are at least four variables involved for each date (open, high, low, and close), and we would like to have some visual way to see all four variables that does not require plotting four separate lines. Financial data is often plotted with a Japanese candlestick plot, so named because it was first created by 18th century Japanese rice traders. Such a chart can be created with matplotlib, though it requires considerable effort.

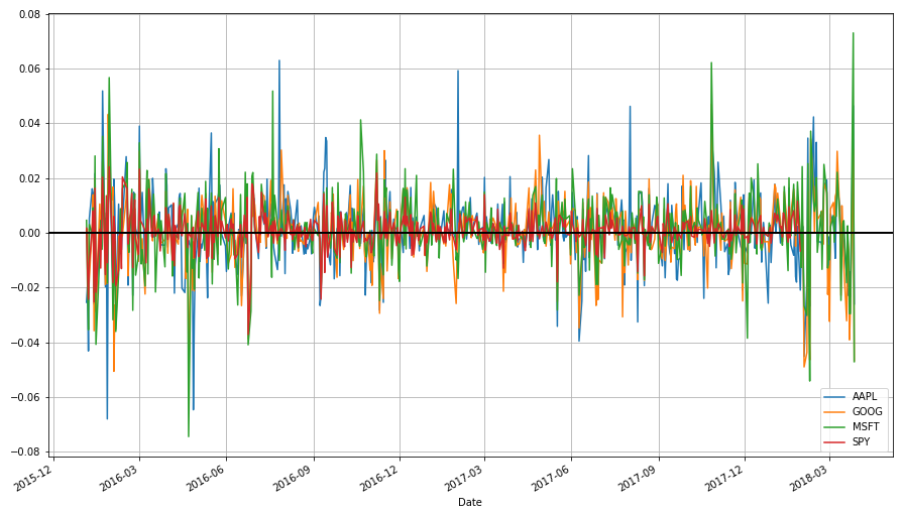


**Compare stocks with other companies and marking (Spyder):**

We may wish to plot multiple financial instruments together; we may want to compare stocks, compare them to the market, or look at other securities such as [exchange-traded funds (ETFs)](https://en.wikipedia.org/wiki/Exchange-traded_fund). Later, we will also want to see how to plot a financial instrument against some indicator, like a moving average. For this you would rather use a line chart than a candlestick chart



**Non-Station and not Trend**

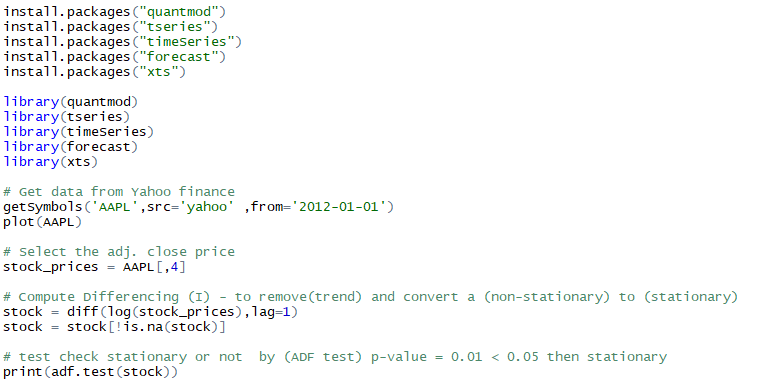


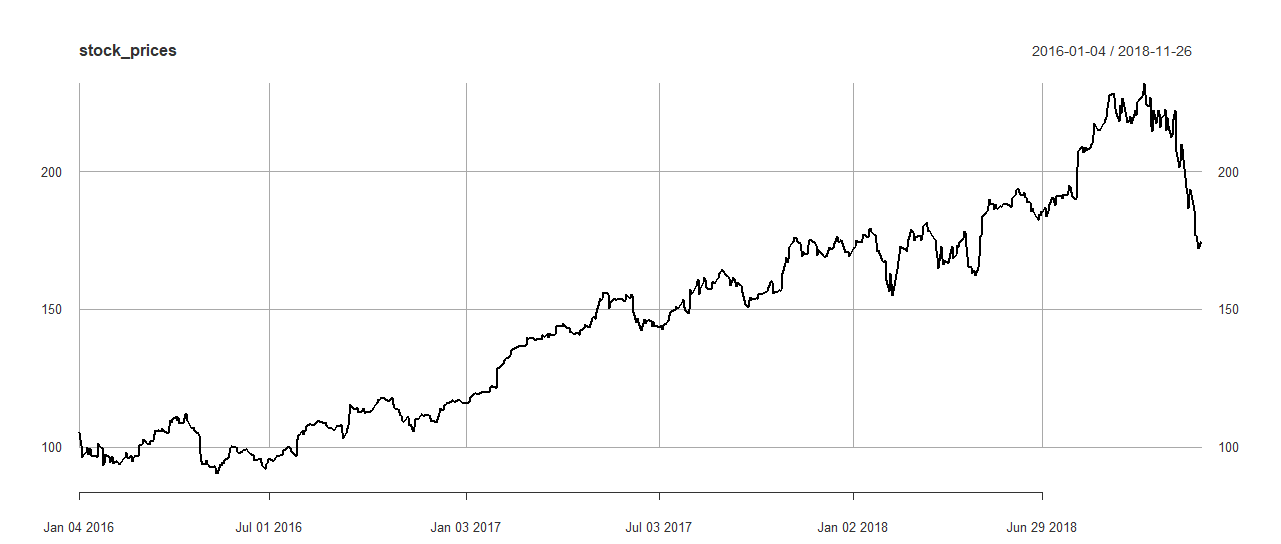
**Moving Averages**

Charts are very useful. In fact, some traders base their strategies almost entirely off charts (these are the "technicians", since trading strategies based off finding patterns in charts is a part of the trading doctrine known as technical analysis). Let's now consider how we can find trends in stocks.

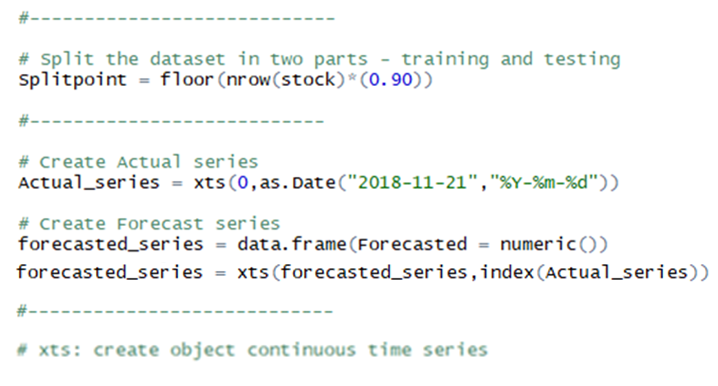
**3.3 Prediction of stock price data by R**

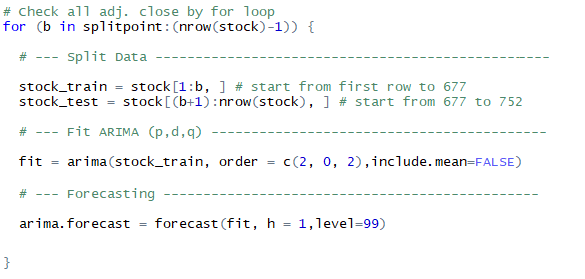
There are a number of packages available for time series analysis and forecasting. We load the relevant R package for time series analysis and pull the stock data from yahoo finance.

**Load Data and Plot**

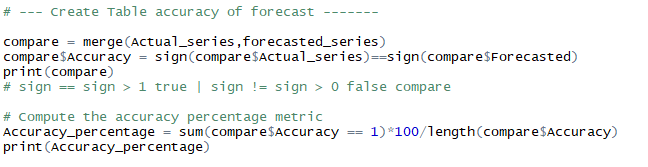
****

**Create Object series**

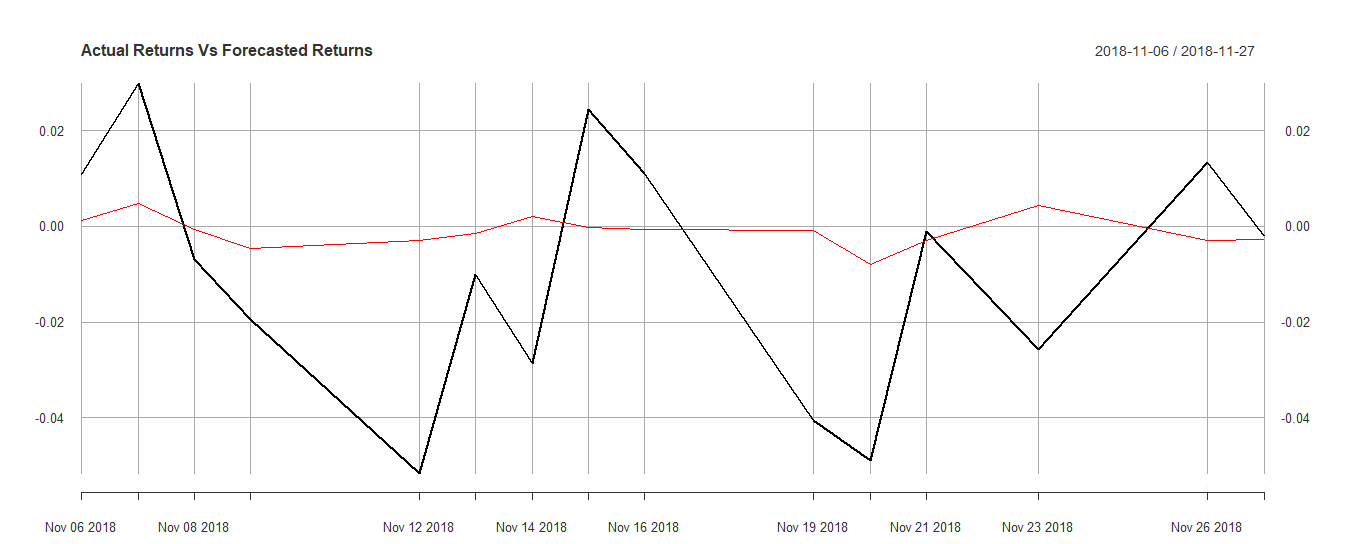
****

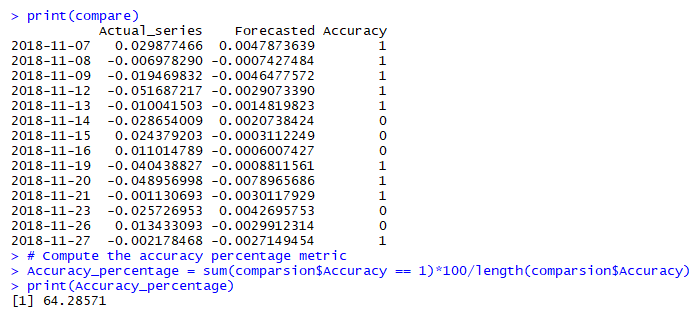
**ARIMA Model (Split, Fit and Forecast)**

**Create Table**

****

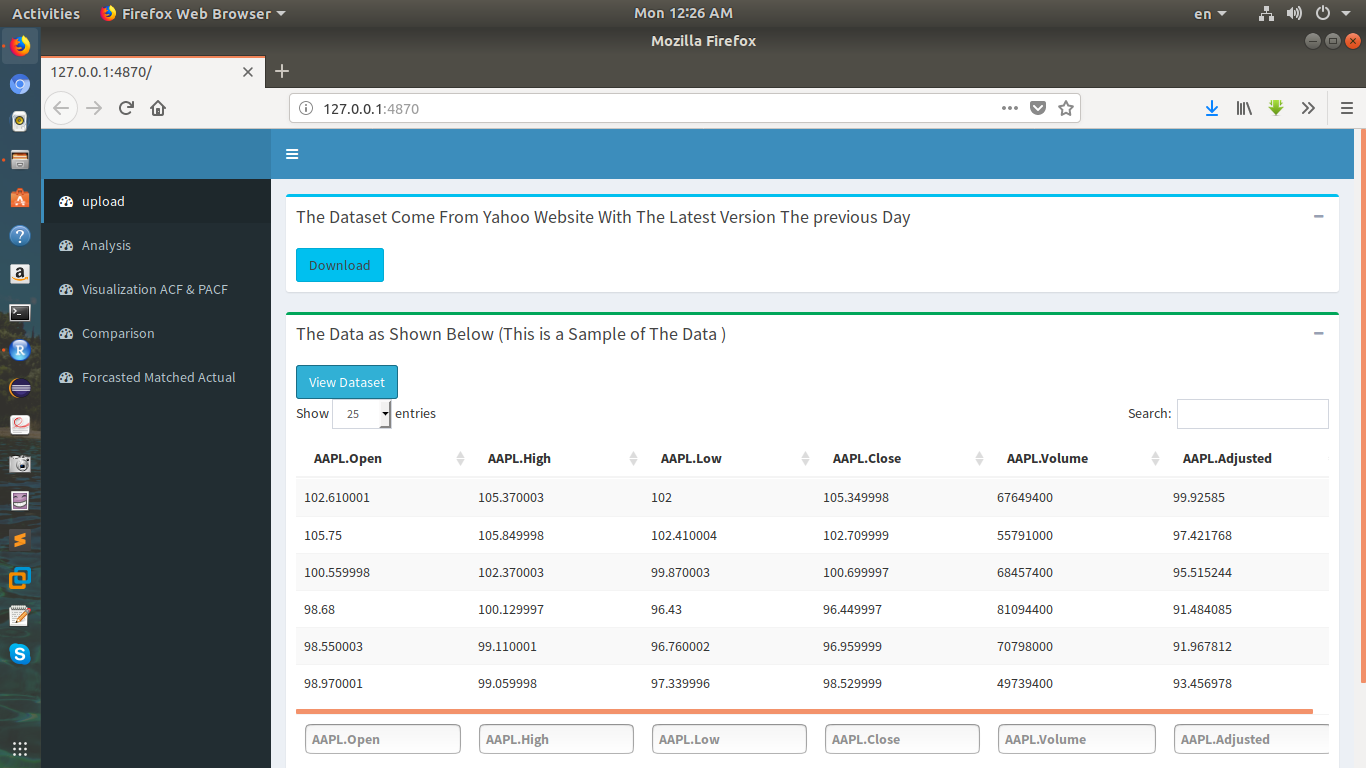
**Result**

****

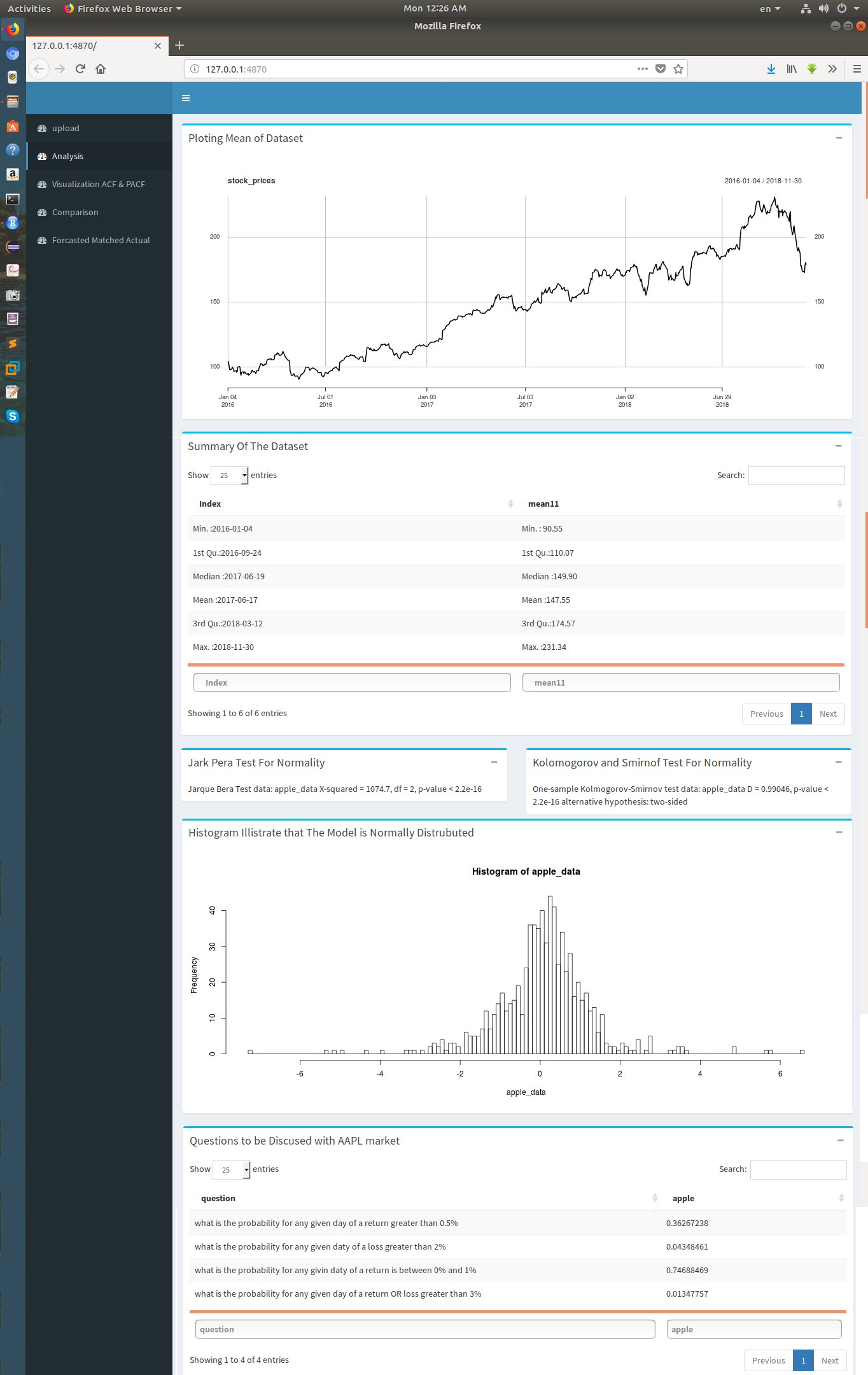
****

**3.4 Stock Price Prediction by Shiny R API**

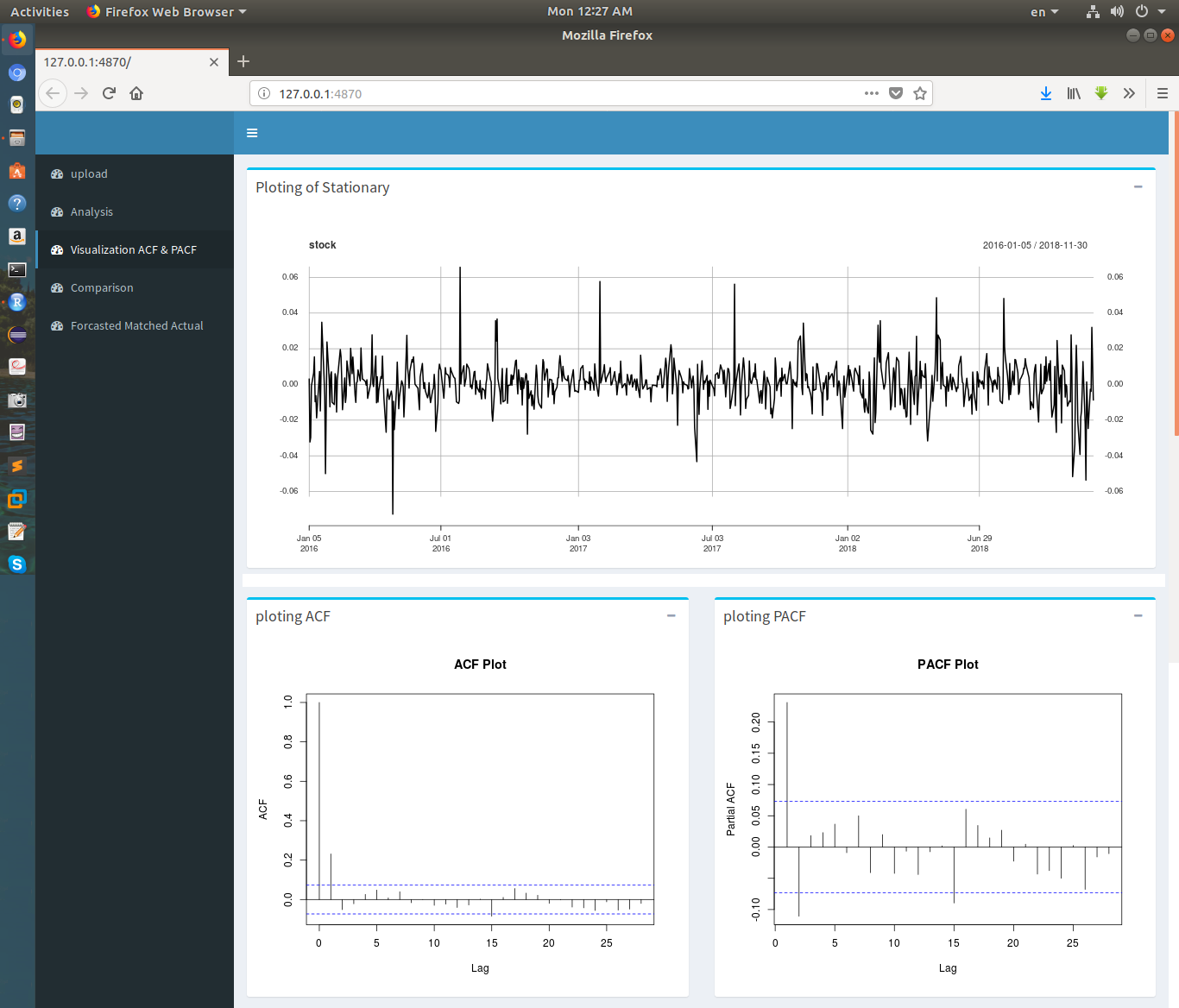
**Load Data**

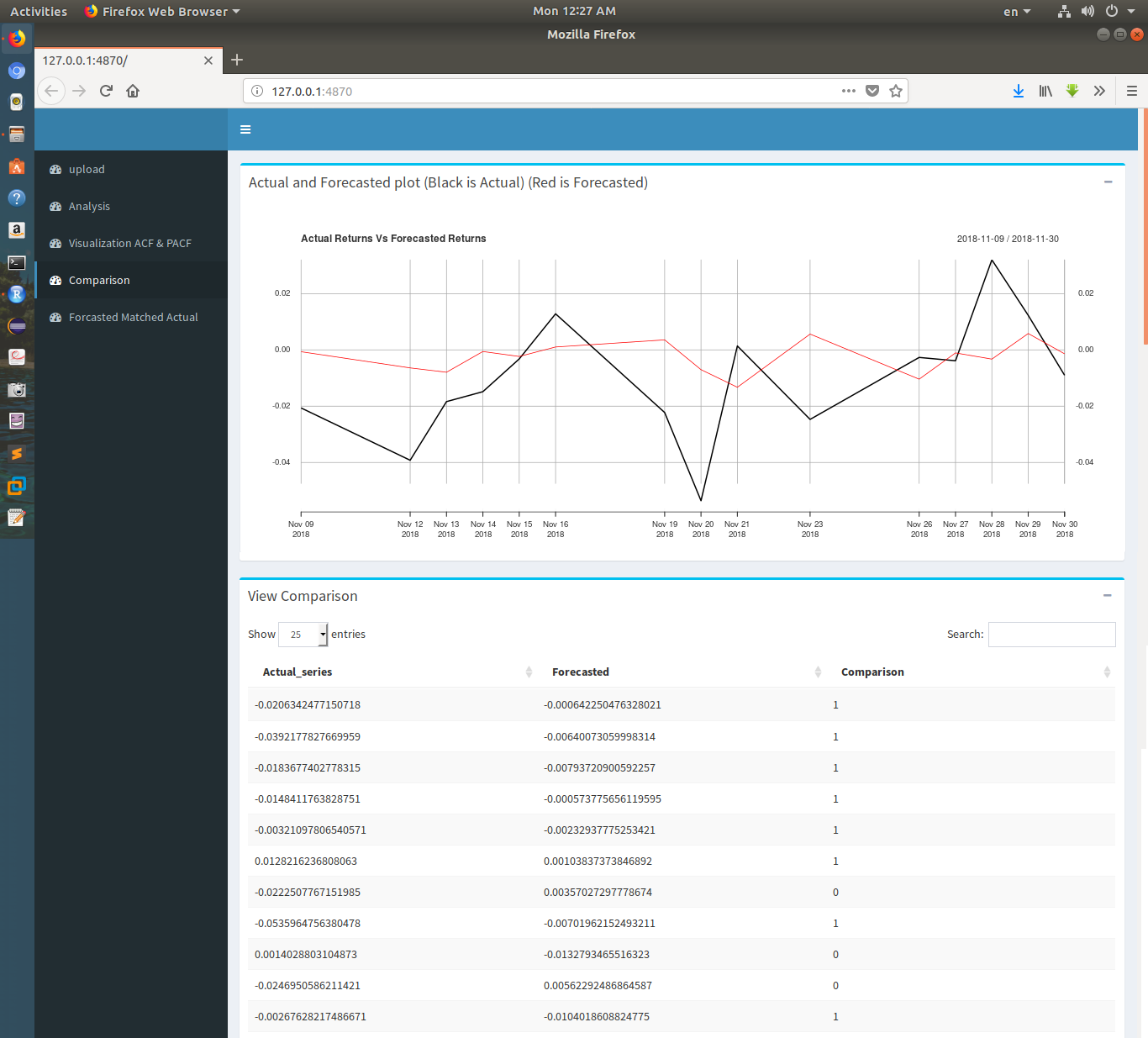
****

**Visualization analysis**

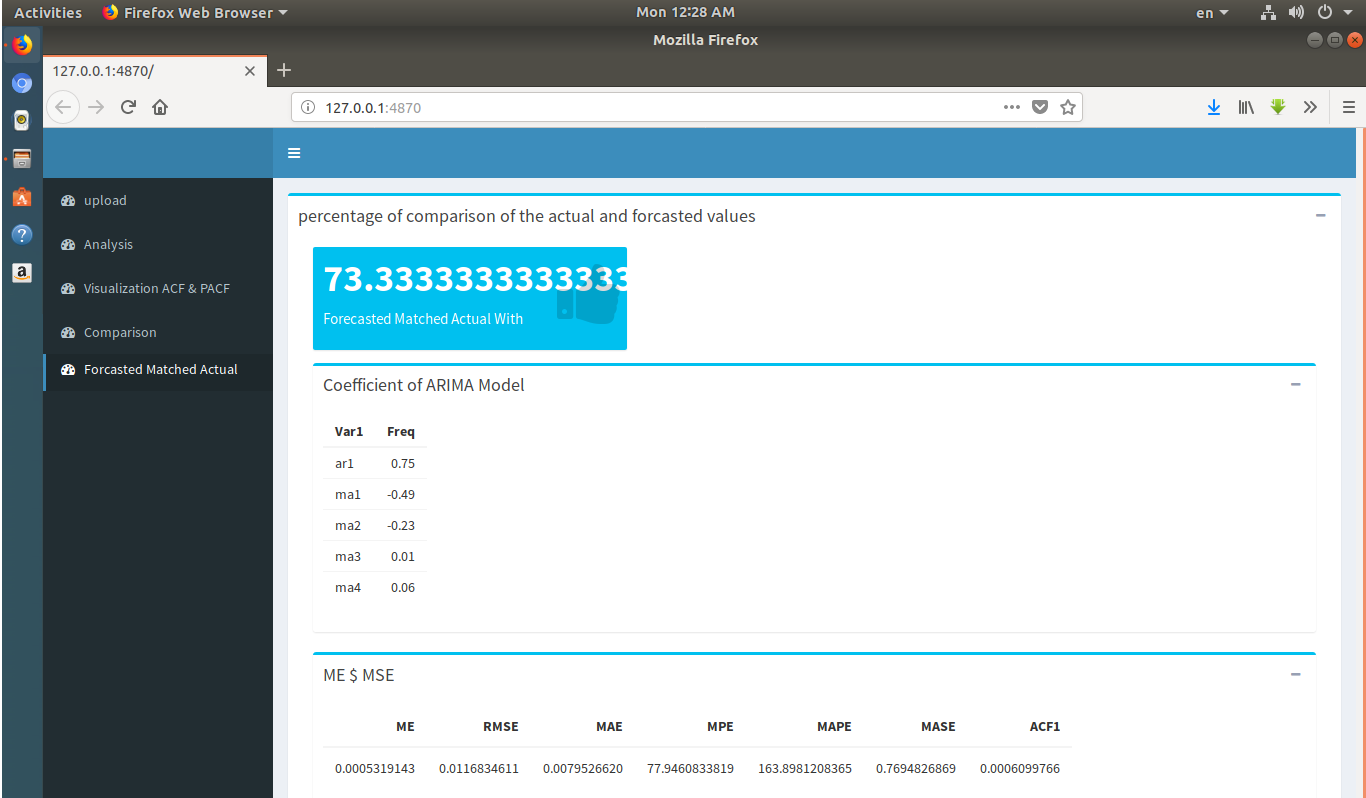
****

**Visualization (ACF – PACF)**

****

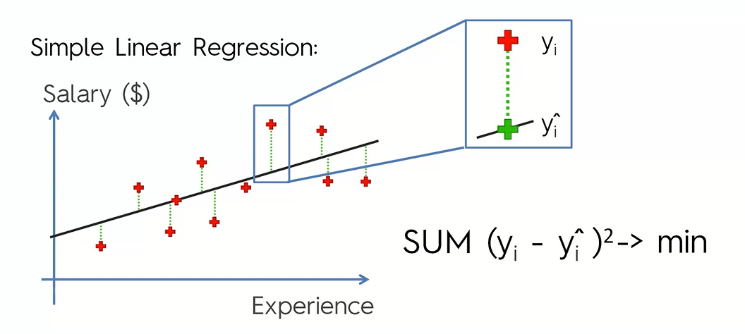
**Comparison (Actual vs Forecast)**

**Accuracy**

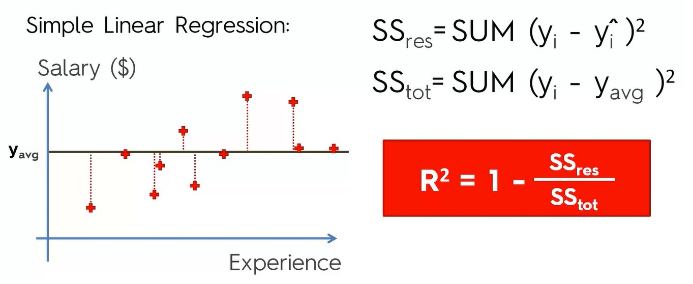
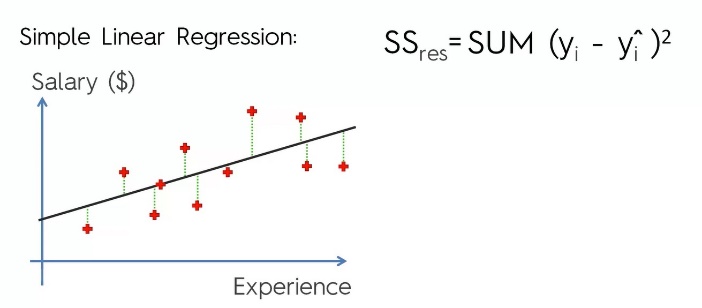
****

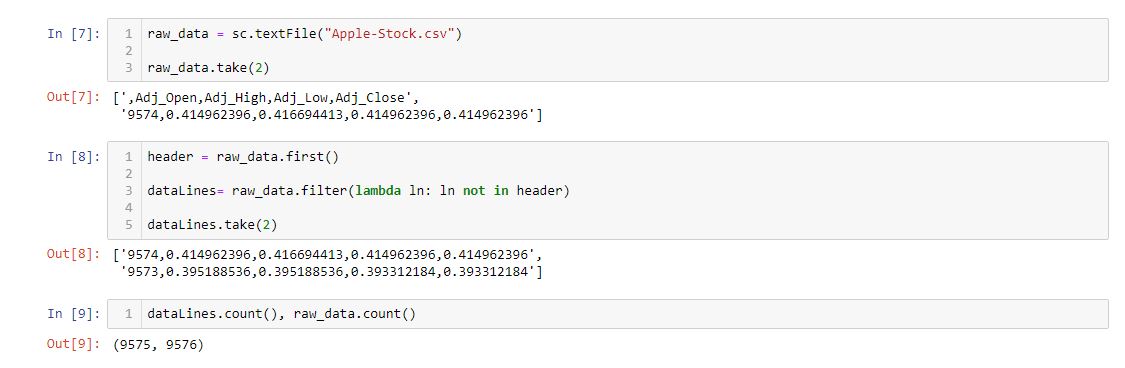
**ARIMA Accuracy**

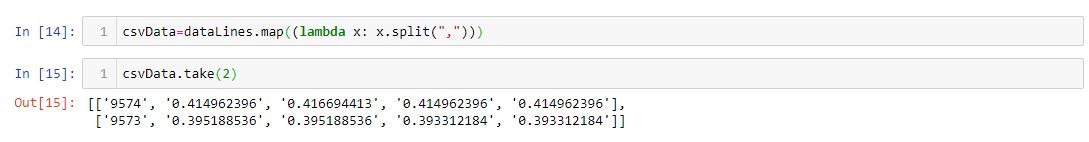
**3.4 Big Data by PySpark:**

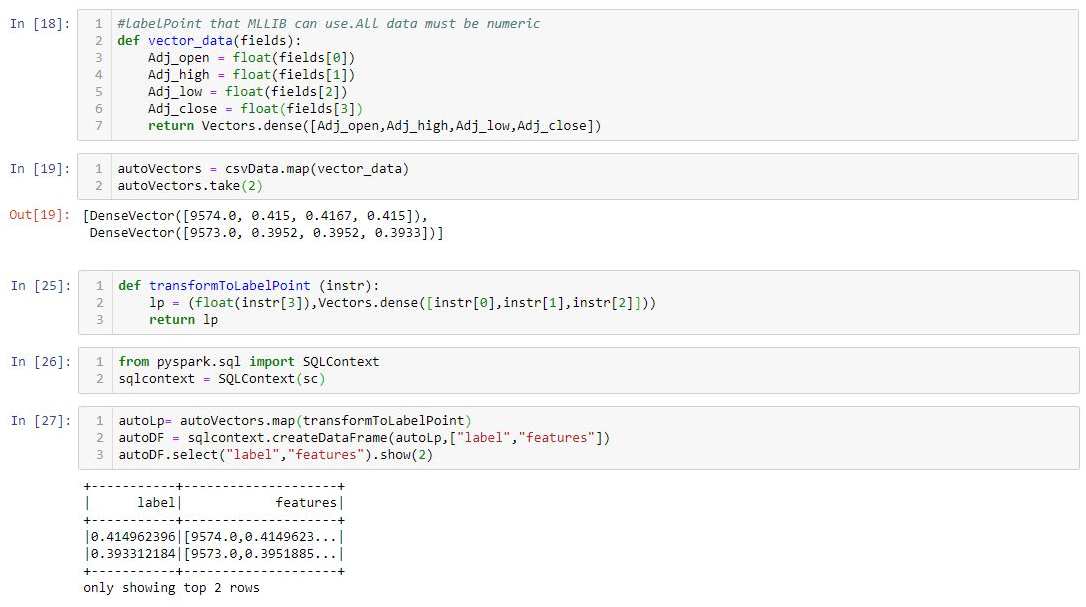
****

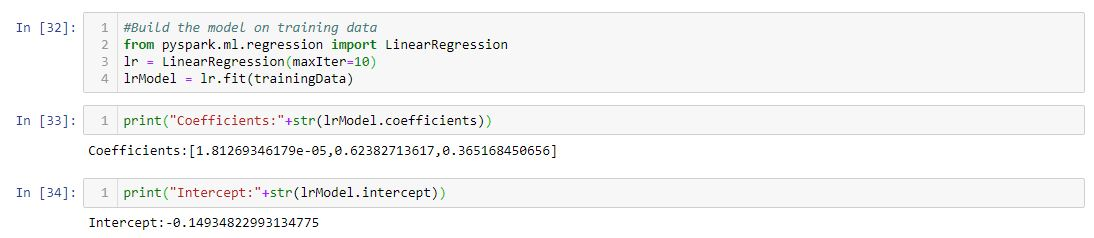
****

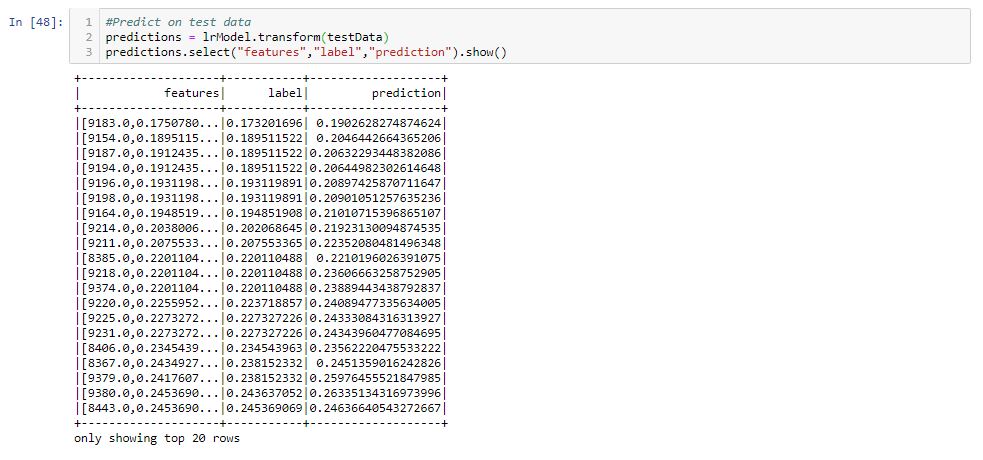
****

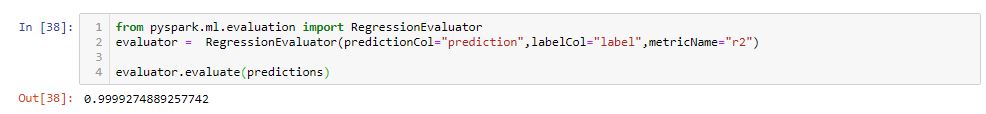
****

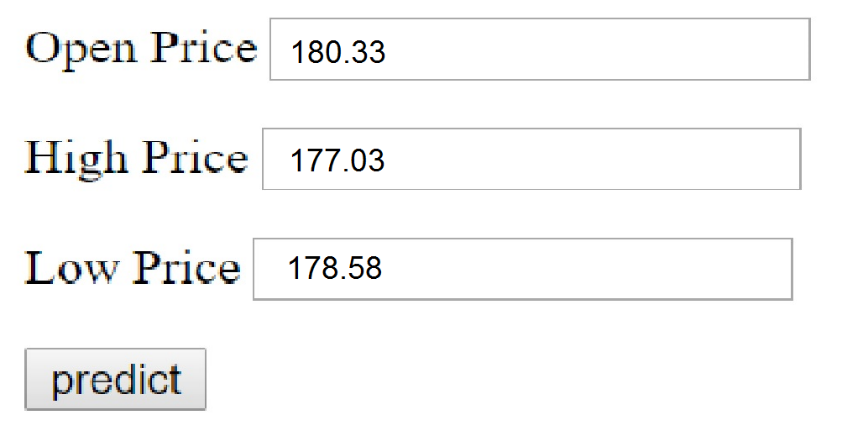
****

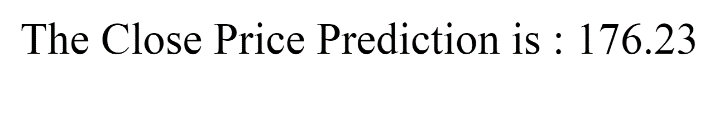
****

****

****

****

****

****

**Conclusions**

Algorithms can be designed to make predictions about the stock market by (ARIMA model), but It's better in this direction using (LSTM) and (RNN).we can't implement these techniques because we have a limited time but don't mind trying. so we will use the main technique at the beginning like (ARIMA) in time series forecast.

**Reference:**

* https://www.wikiwand.com/en/Stock\_market\_prediction
* https://emerj.com/ai-sector-overviews/machine-learning-in-finance/
* https://emerj.com/ai-podcast-interviews/machine-learning-not-a-crystal-ball-but-it-brings-clarity-to-investment-decisions/
* https://www.investopedia.com/articles/active-trading/101014/basics-algorithmic-trading-concepts-and-examples.asp
* https://www.r-bloggers.com/forecasting-stock-returns-using-arima-model/
* http://itfeature.com/time-series-analysis-and-forecasting/component-of-time-series-data
* https://www.sciencedirect.com/science/article/pii/S1532046415002282