A PROJECT REPORT ON

"TIME SERIES FORECASTING USING MACHINE LEARNING WITH TENSORFLOW JS"

Submitted in partial fulfilment of the requirement of

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

Submitted By

KOMAL KUMARI (15107013)
VEDA DHANRAJ VERMA (15107029)

Under the supervision of

MR. DEEPAK KANT NETAM

(Assistant Professor)

DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF STUDIES ENGINEERING AND TECHNOLOGY
GURU GHASIDAS VISHWAVIDYALAYA BILASPUR, INDIA
(CENTRAL UNIVERSITY)
SESSION: 2018- 2019

DECLARATION

We, hereby declare that the work presented in this dissertation entitled "**TIME SERIES FORECASTING USING MACHINE LEARNING WITH TENSORFLOW.JS**" has been done by us under the supervision of Prof. Mr. Deepak Kant Netam, and this dissertation embodies our own work.

Name of team members

Komal Kumari (15107013)

Veda Dhanraj Verma (15107029)

DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF STUDIES ENGINEERING AND TECHNOLOGY,
GURU GHASIDAS VISHWAVIDYALAYA, BILASPUR
(CENTRAL UNIVERSITY)

CERTIFICATE

This is to certify that the project entitled "**TIME SERIES FORECASTING USING MACHINE LEARNING WITH TENSORFLOW JS**" carried out by "Komal Kumari (15107013) and Veda Dhanraj Verma (15107029)" under my supervision at department of Information Technology, School of Studies Engineering and Technology, Guru Ghasidas Vishwavidyalaya, Central University, Bilaspur.

To the best of my knowledge and belief the report

1. Embodies the work of the candidates themselves/him/herself,
2. Has duly been completed,
3. Fulfils the requirement of the B.Tech degree of the University

(Signature of the Supervisor)
Name: Deepak Kant Netam
Assistant Professor

CERTIFICATE BY THE EXAMINERS

The Project Report entitled "**TIME SERIES FORECASTING USING MACHINE LEARNING WITH TENSORFLOW.JS**" Submitted by Komal Kumari (15107013) Veda Dhanraj Verma (15107029) has been examined by the undersigned as a part of the examination and fulfills the requirement of Bachelor of Technology in Information Technology of Guru Ghasidas Vishwavidyalaya, BILASPUR.

(Signature)	*(Signature)*
Examiner 1	Examiner 2
Name	Name
Designation	Designation
Date:	Date:

ACKNOWLEDGEMENT

With great pleasure, we hereby extend our heartiest thanks and deep sense of indebtedness to Mr. DEEPAK KANT NETAM, Assistant Professor of Information Technology, ITGGV, Bilaspur (CG), for his guidance and everlasting help which has been instrumental in completion of this project. His guidance and support has made this whole project easy to comprehend.

Further we are also grateful to Head of the Department Mr. SANTOSH SONI and all other faculty members Dr. AMIT KUMAR KHASKALAM (Assistant Professor), Mr. SUHEL AHAMED (Assistant Professor), Ms. AKANKSHA GUPTA (Assistant Professor), Mr. ANAND PRAKASH RAWAL (Assistant Professor), Mr. ABHISHEK JAIN (Assistant Professor), Mr. PANKAJ CHANDRA (Assistant Professor), Mr. RAJESH MAHULE (Assistant Professor) and Mr. AGNIVESH PANDEY (Assistant Professor), for their teaching and help which became the basic foundation at every stage of the project work.

Further we are also thankful to Dr. V.D. RANGARI, Dean of Institute of Technology for his support in all academic activities.

We would like to thank to all our friends and parents for appreciation, help and encouragement provided by them throughout the project preparation time, without their support the project would not have completed.

KOMAL KUMARI (15107013)
VEDA DHANRAJ VERMA (15107029)

PREFACE

Financial Forecasting is one of the most trending topic of interest for most of the researchers because it uses the new technology i.e, machine learning models to predict the future aspects. The key relationship is between sales and assets. As sales change, so do assets. Increases in sales result in an increase in cash, accounts receivable, inventory, and possibly in fixed assets. These increases in assets must be funded and that is done by the help of forecasting.

This project revolves around the financial forecasting by using the ARIMA model which is the category of time series model which means the forecast mainly depends on the time factor and we have to predict the average cost of the thing from high price and low price of the value. And we used the tensorflow.js technology to visualize the graph on web-browser. All the aspects were seen on the web which gives the better experience to the client to see the trend and analyse the working of model graph.

TABLE OF CONTENTS

Chapter – I

Chapter Name

1.INTRODUCTION

1. Introduction

Financial forecasting is a technique which is used to predict the future outcome related to the sells, earning etc. It has become the most important thing in the business process and also for the finance company. Forecasting plays the important role for those person who have interest to invest in share market and also for the researchers. The researchers use the various machine learning models for forecasting. Financial forecasting is a method through which we can predict the future thing or aspects. We can also say that forecasting is the art of science and technology together which help in to predict the future outcomes.

But we can also say that the price change are unpredictable and forecast is just a hopeless thing because the price of anything may changes at any instant of time, it does not depends on single condition actually the price can depends on various components and situation. The price forecasting depends on the economic condition of the country, the budget of the country and also on the climate and many other bodies which indirectly affect the forecast.

For forecasting there are various models used to get the desired outcomes but that outcomes may or may not be the same with the actual outcome, it have some amount of deflection present in the result due to presence of noise and some external bodies. Noise gives us the amount of error in our result and to minimise this we should take the various models and compare there result with the other model and then have to select the best model to forecast it.

A futures contract is a binding obligation to make or take delivery of a specific quantity of a particular commodity at a certain future date. The types of commodities covered by the futures contracts are agricultural, petroleum, precious metals, foreign currencies, stock market averages, and interest rate instruments. The contractual obligation can be satisfied also by making an offsetting sale or purchase of an equivalent futures contract prior to the delivery date. Rather than trading the actual commodity, speculators can now trade the futures contract. The price of the futures contract varies in proportion to the price of the underlying commodity, but also other factors such as time remaining till delivery, interest rate, and expectations of future supply have some influence. We have two types of futures traders: hedgers and speculators. A hedger is the producer or the user of the underlying commodity. The hedger buys or sells futures contracts to protect himself against adverse price movements. For example, a corn farmer can utilize the futures market to sell his crop while it is still in the ground, by selling futures contracts. By the time his crop is harvested the price might have fallen, and this way the farmer has protected himself against such a decline. Similarly, a food processing plant can buy corn futures contracts months before it needs the corn, to protect itself against possible price increases in the future.

The speculator, on the other hand, buys and sells futures contracts for the purpose of profit. He thereby has the vital roles of accepting the risks that commodity producers and users wish to avoid, and of providing the market liquidity needed by these hedgers. The methods used by speculators to forecast and trade futures can be grouped into two main categories: fundamental analysis and technical analysis. Fundamental analysis is based upon the study of supply and demand. An increase in supply or a decrease in demand tends to depress the price of a commodity. Conversely, a decrease in supply or an increase in demand will raise the price.

Technical analysis is based on the analysing price pattern, technical analysts do not utilize any external economic data or any relevant news events. They assume that these factors get reflected in the past price pattern, and are therefore utilized in an indirect way. They believe that the recent price pattern to some extent can determine future price direction. Therefore, they study historical data to find correlations between certain patterns and subsequent market direction. Such pattern formations can be observed by looking at the price charts and performing a 'mental pattern recognition'. There are, however, more quantitative approaches in technical analysis.

The main one is the trend following approach, which includes the moving averages technique and the regression analysis. The moving averages technique is one of the most widely used technical indicators. In regression analysis, a line is fit to the most recent price data points. The slope of the line determines whether the market is in an uptrend or downtrend.
Another approach to financial forecasting is the ARIMA model. Here the time series is modelled as a linear system. Once the parameter is estimated, the model run and the forecast is obtained. This model is the part of time series in which the forecast depends upon the time and situation. Along with the arima model there are many model present in the machine learning like linear regression, moving average, auto regression model. We proceed our project with the arima model because this model works on the time-series and for predicting the average price value of something that depends upon the time and also on some external bodies, and we just neglecting the external things as a noise and we have to calculate less RMSE value from above models and those model gives the rmse value less then it will be best model for us.
In second part of our project is to deploy the machine learning model with the tensorflow.js which is a java-script library which help us to make connection between the model with the java-script. The use of tensorflow.js help us to view the things of the model graph on the web browser. With the help of tensorflow.js it gives a visualization of all the components on web which looks good and it also attract the viewers, because when we see the things on some notebook or any other IDE then it will not give us user-interface but by the use of tf.js all the things come visualize on google chrome.

Chapter – II

Background

2. BACKGROUND

2.1 What is Artificial Intelligence?

Artificial Intelligence is the integration of human intelligence along with the machine process especially computer systems. The process may include learning, reasoning and self-correction. The application of the artificial intelligence includes expert system, speech recognition and machine vision. Artificial Intelligence may be categorized as weak and strong. Weak artificial intelligence, were also known as narrow AI, and it is an AI system in which system is designed and trained for a particular work, example Virtual personal assistants.

Strong Artificial Intelligence were also known as artificial general intelligence, and is an artificial intelligence system with generalized human cognitive abilities. And also the strong AI is able to find the solution of a problem with human intervention. Due to the presence of hardware, software and any other cost for artificial intelligence may be expensive, many other peoples those who are including AI components in their standard offerings, as well as access to Artificial Intelligence as a Service system. The tools of the artificial intelligence give a range of new functionality to the business. The deep learning algorithm gives many of the most advanced artificial intelligence which is smart as the data they are given to the training. Because human brain can select the data which gone for the training.

Mainly the artificial intelligence may be categorized into four types: (1). Reactive Machine, (2). Limited Memory, (3). Theory of Mind, (4). Self-awareness. Some few example of the artificial intelligence are: Automation, Machine learning, Machine Vision, Natural Language processing, Robotics, Self-Driving car. And also some application of the AI are: AI in Healthcare, business, education, finance, law, manufacturing etc. Artificial intelligence and machine learning technologies have the potential to transform health care by deriving new and important insights from the vast amount of data generated during the delivery of health care every day. But in these days in medical the new device manufacturers uses these technologies to innovate the new product for better assist health care providers and improve patient care. FDA is considering a total product lifecycle-based regulatory framework for these technologies that would allow for modifications to be made from real-world learning and adaptation.

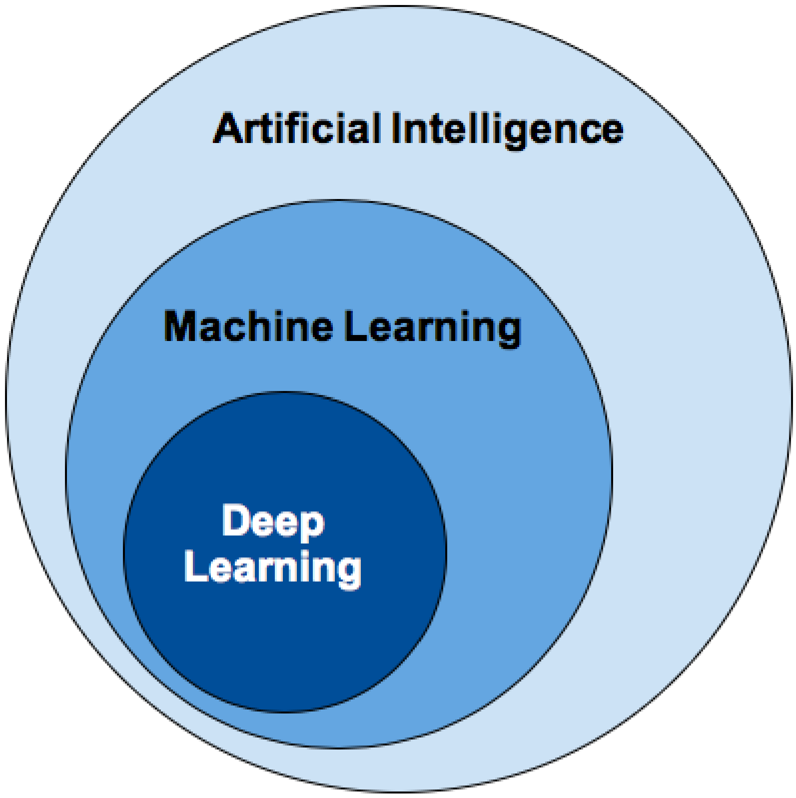
Computing systems engineered over the previous decades are mostly helpless in attempts to catalog and exploit this deluge of "unstructured data," an umbrella term which includes emails, text documents, tweets, photographs, audio and video content, and any other data that lives outside the orderly confines of a spreadsheet. Every minute, internet users generate roughly 400 hours of video, 18 million text message, and 187 million emails. This data is not only overwhelming, but also largely useless. Unless a person categorizes this information for a computer to understand (or analyzes it themselves), unstructured data by nature is incoherent to computers and, crucially, not actionable. The vast majority of data we store is unstructured or, as some would describe it, invisible. AI as a broad domain of machine techniques that are characteristic of human intelligence. He also sees AI as a generic term which refers to "smart machines that exhibit human traits like logical thinking, intuitive leaps, emotional intelligence, and empathy," he said. A subset of implementation tools, including machine learning, deep learning, and convolutional neural networks, are each specialized fields, but generally considered to be part of the computational toolset underneath the umbrella of AI. As long as insight dictates success, AI will continue to dominate the world of emerging technology because of its power gain insight from data to bolster efficiency, prediction, and effective decision-making. At its core, AI is about making our data visible and uncovering insights to help tackle the world’s most important problems. Data scientists at Leidos are continually developing expertise in this domain through internal research and development and center of excellence.

Machine Learning is an artificial intelligence technique that can be used to design and train software algorithms to learn from and act on data. Software developers can use machine learning to create an algorithm that is ‘locked’ so that its function does not change, or ‘adaptive’ so its behavior can change over time based on new data. Some real-world examples of artificial intelligence and machine learning technologies include:

- An imaging system that uses algorithms to give diagnostic information for skin cancer in patients.
- A smart electrocardiogram (ECG) device that estimates the probability of a heart attack.

2.2 What is Machine Learning?

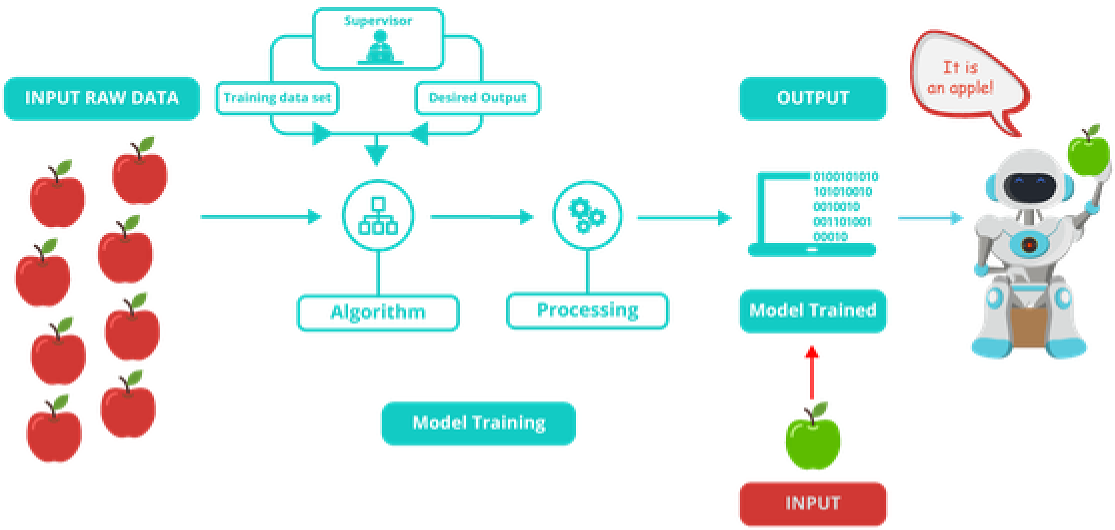
Machine learning, artificial intelligence (AI), and cognitive computing are dominating conversations about how emerging advanced analytics can provide businesses with a competitive advantage to the business. Now a day there is no any existing business leaders are facing new difficulties by the coming of new technology and unanticipated competitors. These businesses are looking at new strategies that can prepare them for the future. While a business can try different strategies, they all come back to a fundamental truth — you have to follow the data. In this chapter, we delve into what the value of machine learning can be to your business strategy. How should you think about machine learning? What can you offer the business based on advanced analytics technique that can be a game-changer?
Fundamentally the Machine learning is a building block of a mathematical models to help in understanding the data. Mainly the machine learning is nothing it is a simply a mathematics and all things are based on the calculation which are carried out to get the outcome. Machine Learning is type of AI that enables a system to learn from the given data which is store in the csv file or excel sheet file rather than from the explicit programming. Machine Learning use a very large number of algorithm from which it can learn from the data to improve, describe data and predict the outcomes. Machine Learning can give the outcome when we train our machine learning algorithm with desired amount of data. After the completion of training algorithm, then we can give some input to the algorithm then we get our output. For example, a predictive algorithm will create a predictive model. When we provide the predictive model with the data-set, we will receive a prediction based on the data-set that we used to train our model. Machine learning is now essential for creating analytics models. For example, when you visit an e-commerce site and start viewing products and reading reviews, you’re likely presented with other, similar products that you may find interesting. These recommendations aren’t hard coded by an army of developers. The suggestions are served to the site via a machine learning model. The model ingests your browsing history along with other shoppers’ browsing and purchasing data in order to present other similar products that you may want to purchase. With the help of the machine learning we can enable our model to train the data-set that we used before deploying.
Some of the machine learning models that we are using were online and continuously take the new data ingested. On the other hand, other models, called offline machine learning models, are derived from machine learning algorithms but, once deployed, do not change. This iterative process of online models leads to an improvement in the types of associations that are made between data elements. After the training of the models from the given data-set then that model is used in our physical world to take the real world values to predict the future outcomes.



It is not enough to simply ingest vast amounts of data. When we give the accurate machine learning models then it will require the source data-set be accurate and meaningful. In addition, these data sources are meaningful when combined with each other so that the model is accurate and trusted. You have to understand the origin of your data sources and whether they make sense when they're combined. When we are using some data-set for the operation then we should trust on our data-set and then after we have check the data-set for the noise, if there are noise present the data-set then that will hamper our result if we use the best to best model to predict the things. Data refinement provides the foundation for building analytical models that deliver results you can trust. The process of data refinement will help to ensure that your data is timely, clean, and well understood.
Mainly machine learning is classified into two categories (1.) Supervised Learning, (2.) Unsupervised Learning.
Supervised learning basically starts with the predefined set of data that is used and a certain understanding of how the data is classified. Supervised learning is intended to find patterns in data that can be applied to an analytics process. This data has labeled features that define the meaning of data. For example, there could be millions of images of animals and include an explanation of what each animal is and then you can create a machine learning application that distinguishes one animal from another. By labeling this data about types of animals, you may have hundreds of categories of different species. Because the attributes and the meaning of the data have been identified, it is well understood by the users that are training the modeled data so that it fits the details of the labels. When the label is continuous, it is a regression; when the data comes from a finite set of values, it known as classification. In essence, regression used for supervised learning helps you understand the correlation between variables. An example of supervised learning is weather forecasting. By using regression analysis, weather forecasting takes into account known historical weather patterns and the current conditions to provide a prediction on the weather. The algorithms are trained using preprocessed examples, and at this point, the performance of the algorithms is evaluated with test data. Occasionally, patterns that are identified in a subset of the data can't be detected in the larger

population of data. If the model is fit to only represent the patterns that exist in the training subset, you create a problem called overfitting. Overfitting means that your model is precisely tuned for our training data that we are using may cannot be used for that which have large number of unknown facts present or data. To protect against overfitting, testing needs to be done against unforeseen or unknown labeled data. Using unforeseen data for the test set can help you evaluate the accuracy of the model in predicting outcomes and results. Supervised training models have broad applicability to a variety of business problems, including fraud detection, recommendation solutions, speech recognition, or risk analysis.

Unsupervised learning is best suited when the problem requires a massive amount of data that is unlabeled. For example, social media applications, such as Twitter, Instagram, Snapchat, and so on all have large amounts of unlabeled data. Understanding the meaning behind this data requires algorithms that can-not be understand by the patterns or other. Therefore, the supervised learning conducts an iterative process of analyzing data without human intervention. Unsupervised learning is used with email spam-detecting technology. There are far too many variables in legitimate and spam emails for an analyst to flag unsolicited bulk email. Instead, machine learning classifiers based on clustering and association are applied in order to identify unwanted email. The unlabeled data creates the parameter values and classification of the data. In essence, this process adds labels to the data so that it becomes supervised. Unsupervised learning can determine the outcome when there is a massive amount of data. In this case, the developer doesn't know the context of the data being analyzed, so labeling isn't possible at this stage. Therefore, unsupervised learning can be used as the first step before passing the data to a supervised learning process. Unsupervised learning algorithms can help businesses understand large volumes of new, unlabeled data. Similarly to supervised learning (see the preceding section), these algorithms look for patterns in the data; however, the difference is that the data is not already understood. For example, in healthcare, collecting huge amounts of data about a specific disease can help practitioners gain insights into the patterns of symptoms and relate those to outcomes from patients. It would take too much time to label all the data sources associated with a disease such as diabetes. Therefore, an unsupervised learning approach can help determine outcomes more quickly than a supervised learning.



Chapter – III

Project Methodology

3. PROJECT METHODOLOGY

3.1. INTRODUCTION

Financial Forecasting or we say Time series forecasting is a technique for the prediction of events through a sequence of time. The technique is used across many fields of study, from the geology to behavior to economics. The techniques predict future events by analyzing the trends of the past, on the assumption that future trends will hold similar to historical trends. It is an important area of machine learning that is often neglected because there are so many prediction problems that involve a time component. These problems are neglected because it is this time component that makes time series problems more difficult to handle.

Financial Forecasting or Time series forecasting can be performed using various methodologies :

A. Neural Networks

B. SVM (Support Vector Machines)

C. Regression Techniques

3.1.1. DATASET SPECIFICATION

"A time series is a sequence of observations taken sequentially in time." A time series dataset is different. Time series adds an explicit order dependence between observations: a time dimension. This additional dimension is both a constraint and a structure that provides a source of additional information.

DATASET is taken from
TATAMOTORS STOCK PRICES FROM NSE (National Stock Exchange of India Ltd.)
The DATASET size is 247 X 10. Some part of data is displayed below :

	Date	Average Price
0	01-Jan-2018	430.49
1	02-Jan-2018	433.41
2	03-Jan-2018	435.06
3	04-Jan-2018	429.56
4	05-Jan-2018	432.92
5	08-Jan-2018	433.92
6	09-Jan-2018	438.96
7	10-Jan-2018	434.76
8	11-Jan-2018	435.64
9	12-Jan-2018	437.06

3.2. LITERATURE REVIEW

Some of the methods that have been tried for the task of time series forecasting with their error rates is mentioned below:

Method	RMSE
Simple Average :	134.70891439533298
Moving Average :	66.80933755238976
Holt's Linear Trend model :	77.2764142463477
AR(1,1,0)	8.2083
MA(0,1,1)	8.6071
ARIMA(1,1,1)	8.0267
AR(2,1,0)	15.3444
MA(0,1,2)	15.3632
ARIMA(2,1,2)	15.1693

3.3. PROPOSED METHODOLOGY

Time series analysis involves developing models that best capture or describe an observed time series in order to understand the underlying causes. This field of study seeks the "*why*" behind a time series dataset.

3.3.1. Dataset

The Dataset provided to us from Stock Price from NSE (National Stock Exchange of India Ltd.) consists of the following characteristics:

- Size: The dataset consists of 247 X 10, Rows and Columns
- Frequency: The real-time data collected is on a daily basis
- Properties: The data consists of date, closing price, opening price, stock exchanges, high price, low, price, etc.

3.3.2. Pre-processing Done

We did not include any other datasets apart from the given dataset, as said in the pre-final report, however we carried out some pre-processing steps. We extracted limited parameters provided in the data.

There were a number of columns in the dataset including closing price, opening price, stock exchanges, high price, low, price, etc. We are only extracting date and average price column which we will be using for time series forecasting.

After that we have converted our data from rows and columns format to a time series data which is efficient and beneficial for financial or time series forecasting. By converting to time series data we are using date as the dataset index inspite of serial numbers. Dataset is divided into two parts:

A. Training dataset: It will be used for training our machine learning model
B. Testing dataset: It will be used for testing and validating our model

3.3.3. Training Model

The model which we have chosen after reviewing other approaches and literature reviews for performing time series forecast is ARIMA Model which stands for Auto-Regressive Integrated Moving Average. In the training phase, training data is used for training the model, in our project the model is trained by learning stock prices on a daily basis.

3.3.4. Building UI using Tensorflow.js

Since python code is limited only to terminals. Therefore, for better visualization and user-friendly interactions we are developing a UI for displaying our deployed machine learning model into it using tensorflow.js

The dataset representations, pre-processing steps and the final predictions and forecasts are displayed in the UI.

Chapter – IV

Requirements

3. REQUIREMENTS

4.1 SOFTWARE REQUIREMENTS

1. Jupyter Notebook
2. Sublime Text
3. Google Chrome
4. TensorFlow.js tfvis visor (library)
5. Pandas Library of Python
6. Matplotlib python library

4.2 HARDWARE REQUIREMENTS

1. Laptop with any operating system (windows, linux or mac os)
2. RAM of minimum 4gb with memory space of 1 TB
3. Graphics of 1 GB

4.3 LANGUAGE USED

1. Python3 core
2. Machine Learning Models along with Artificial Intelligence
3. TensorFlow.js library
4. HTML / CSS

Chapter – V

Implementation

5. IMPLEMENTATION

5.1. Phases of Project

5.1.1. Phase I: Pre-processing of dataset using Pandas in Jupyter Lab

- Pandas is one of the most popular Python libraries for Data Science and Analytics.
- We have used Pandas for handling and performing pre-processing in data
- We convert the dataset with integral index to time series data with date as index

5.2.2. Phase II: Construction of ARIMA model using statsmodels in python

- Statsmodels.tsa (time series analysis) contains model classes and functions that are useful for time series analysis
- Statsmodels.tsa basic models include univariate autoregressive models (AR), vector autoregressive models (VAR) and univariate autoregressive moving average models (ARMA) and autoregressive integrated moving average models (ARIMA)

5.2.3. Phase III: Training and Testing dataset on the ARIMA model by statsmodels.tsa

- Training the model by the functions defined by model.fit() in statsmodels.tsa
- Testing the model by providing test dataset by model.predict() in statsmodels.tsa
- Visualising Data and Forecasts using matplotlib python library

5.2.4. Phase IV: Deploying ARIMA model in UI using Tensorflow.js

- TensorFlow.js is a library for developing and training ML models in JavaScript, and deploying in browser or on Node.js
- Run existing models: Use off-the-shelf JavaScript models or convert Python TensorFlow models to run in the browser or under Node.js.
- Develop ML with JavaScript: Build and train models directly in JavaScript using flexible and intuitive APIs.

5.2. Pandas Implementation

Pandas is a powerful Python data analysis toolkit. It is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

5.2.1. Converting to time series

The dataset is converted to time series data using the pandas read_csv () and Date is made the index of the series.

Data = pd.read_csv('tatavg.csv', parse_dates=['Date'], index_col='Date')

print (data.head())

```
            Average Price
Date
2018-01-01         430.49
2018-01-02         433.41
2018-01-03         435.06
2018-01-04         429.56
2018-01-05         432.92
```

5.2.2. Log transform

We have to take log transform of the time series data for further processing on it.

#Making the time series stationary by removing:

#1. Trend and 2. Stationarity

#1. Removing trend : Taking log transform and applying smoothing-moving average

ts_log = np.log(ts)

5.2.3. Differencing

Differencing is a method of transforming a time series dataset. It can be used to remove the series dependence on time, so-called temporal dependence. This includes structures like trends and seasonality.

#Plot after log transform

#plt.plot(ts_log)

#plotting log transform and moving average together

moving_avg = ts_log.rolling(window=12, center=False).mean()

```
plt.plot(ts_log)
plt.plot(moving_avg, color='red')
#Taking difference from the original series
ts_log_moving_avg_diff = ts_log – moving_avg
ts_log_moving_avg_diff.head(12)
```

5.3. ARIMA Implementation

5.3.1. Understanding ARIMA model

ARIMA stands for AutoRegressive Integrated Moving Average.

AR (Autoregression): A model that uses the dependent relationship between an observation and some number of lagged observations. P is a parameter of how many lagged observations to be taken in.

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

I (Integrated): A model that uses the differencing of raw observations (e.g. subtracting an observation from the previous time step). Differencing in statistics is a transformation applied to time-series data in order to make it stationary. This allows the properties do not depend on the time of observation, eliminating trend and seasonality and stabilizing the mean of the time series.

$$\begin{aligned} y_t^* &= y_t' - y_{t-1}' \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2} \end{aligned}$$

MA (Moving Average): A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations. Q is a parameter of how many lagged observations to be taken in. Contrary to the AR model, the finite MA model is always stationary.

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

Parameters of the ARIMA model:

- p (lag order): number of lag observations included in the model
- d (degree of differencing): number of times that the raw observations are differenced
- q (order of moving average): size of the moving average window

The forecasting equation is:

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \ldots - \theta_q e_{t-q}$$

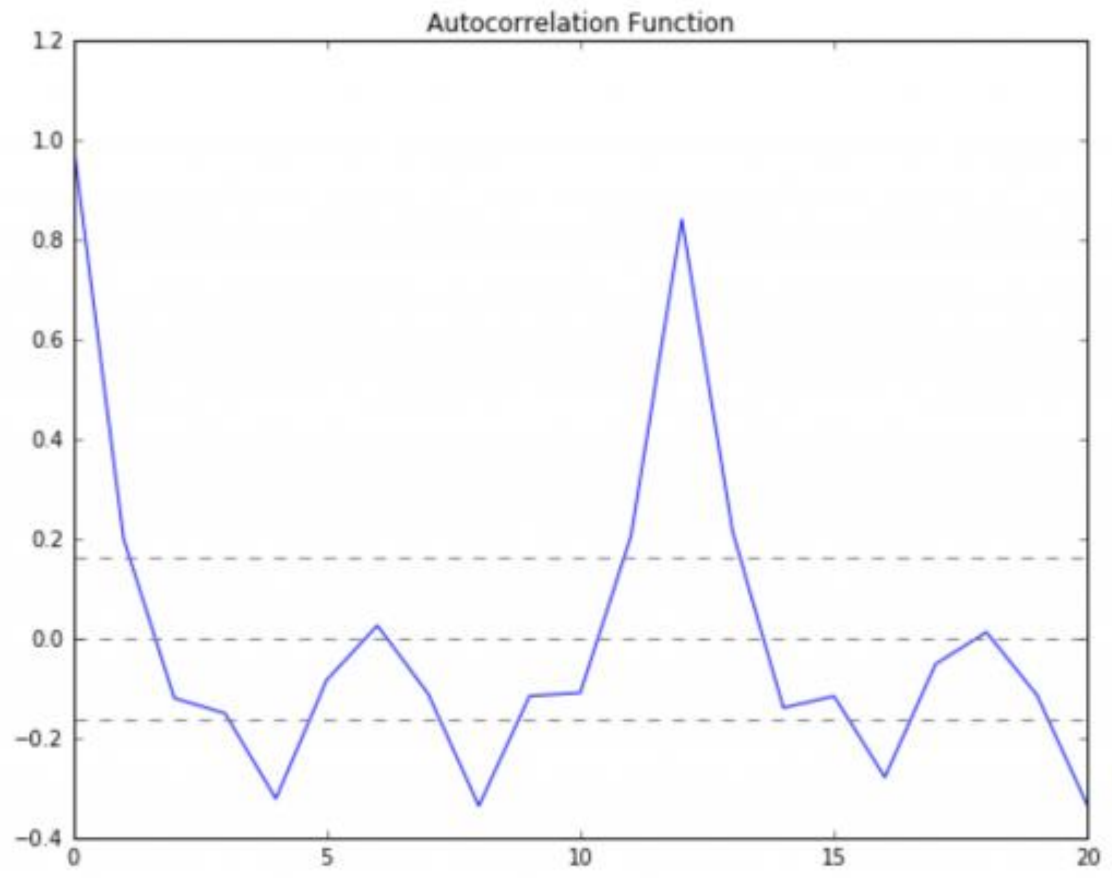
# ARIMA MODEL
from statsmodels.tsa.arima_model import ARIMA

# The ARIMA () function is imported from statsmodels.tsa
# the parameter selection will be performed using ACF and PACF plots
# order = (p, d, q) where,
# p is the number of AR terms i.e, autoregressive terms
# q is the number of MA terms i.e, moving average terms
# d is the order of differencing
model = ARIMA(ts_log, order = (2, 1, 2))

5.3.2. Determining model parameters

ACF and PACF plots of the differenced series are used to determine the parameters

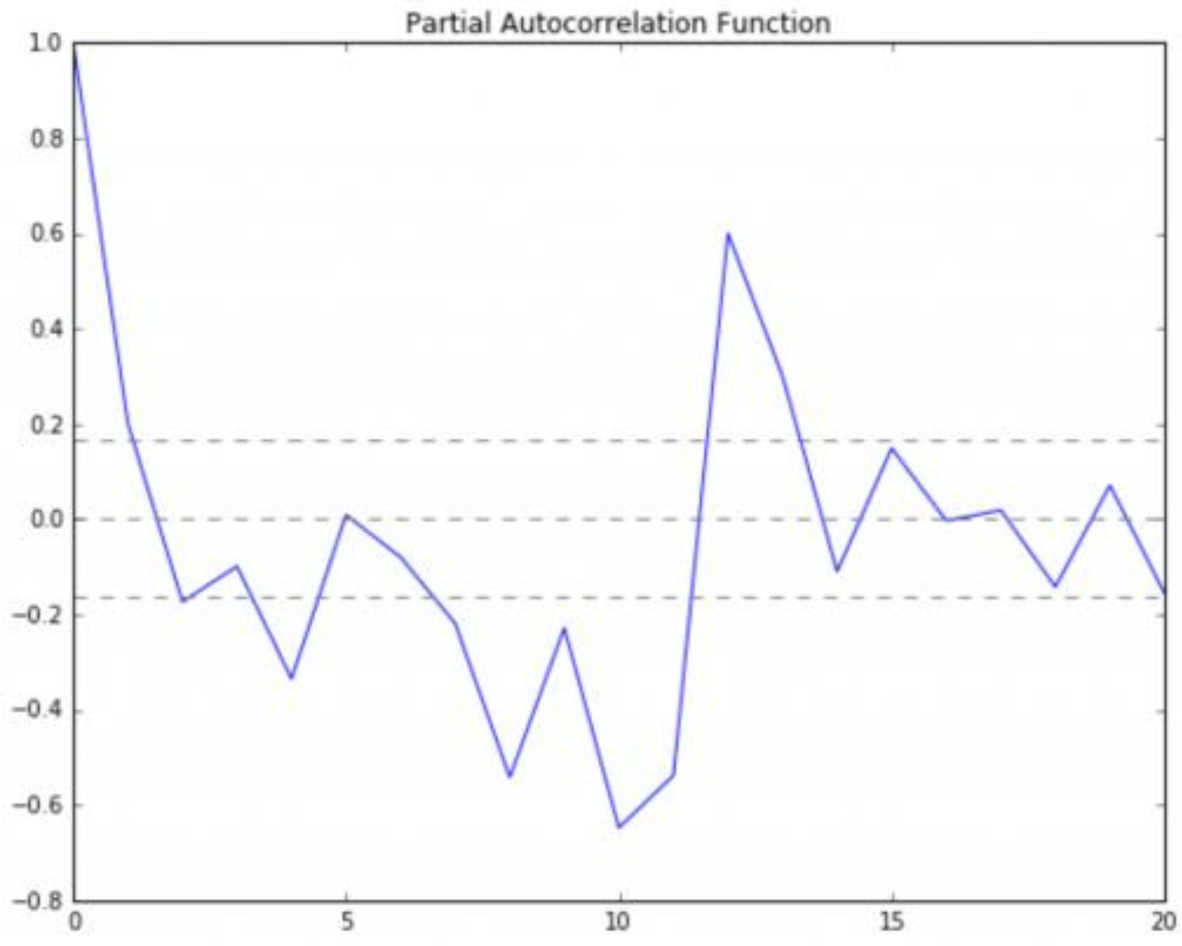
- Autocorrelation function (ACF) identifies the numbers of AR terms. It is merely a bar chart of the coefficients of correlation between a time series and lags of itself.



```
#ACF and PACF plots:
from statsmodels.tsa.stattools import acf, pacf
lag_acf = acf(ts_log_diff, nlags=20)
lag_pacf = pacf(ts_log_diff, nlags=20, method='ols')

#Plot ACF:
plt.subplot(121)
plt.plot(lag_acf)
plt.axhline(y=0,linestyle='--',color='gray')
plt.axhline(y=-1.96/np.sqrt(len(ts_log_diff)),linestyle='--',color='gray')
plt.axhline(y=1.96/np.sqrt(len(ts_log_diff)),linestyle='--',color='gray')
plt.title('Autocorrelation Function')
```

- Partial autocorrelation (PACF) identifies the numbers of MA terms It is a plot of the partial correlation coefficients between the series and lags of itself.



```
from statsmodels.graphics.tsaplots import plot_pacf
plot_pacf(ts_log_diff, lags = 20)
#Plot PACF:
plt.subplot(122)
plt.plot(lag_pacf)
plt.axhline(y=0,linestyle='--',color='gray')
plt.axhline(y=-1.96/np.sqrt(len(ts_log_diff)),linestyle='--',color='gray')
plt.axhline(y=1.96/np.sqrt(len(ts_log_diff)),linestyle='--',color='gray')
plt.title('Partial Autocorrelation Function')
plt.tight_layout()
```

5.3.3. Checking stationarity

It is very important to check whether the time series data is stationary or not because it is helpful in predicting true forecasts. There are various methods of checking the stationarity of the series which includes rolling statistics and dickey fuller test. We have chosen the graphical approach of rolling statistics in which we calculate the rolling mean and rolling standard deviation rather than dickey-fuller test which is an algebraic approach.

```
#Checking the stationarity of the time series data, techniques are -
#1) Rolling statistics
#2) Dickey-Fuller Test
#Clearly there is an increasing trend: taking the Rolling Statistics technique
from statsmodels.tsa.stattools import adfuller

#Creating a function for stationarity check

def test_stationarity(timeseries):
```

```
    #Determing rolling statistics
    rolmean = timeseries.rolling(window=12, center=False).mean()
    #rolmean = pd.rolling_mean(timeseries, window=12)
    #rolstd = pd.rolling_std(timeseries, window=12)
    rolstd = timeseries.rolling(window=12, center=False).std()

    #Plot rolling statistics:
    orig = plt.plot(timeseries, color='blue',label='Original')
    mean = plt.plot(rolmean, color='red', label='Rolling Mean')
    std = plt.plot(rolstd, color='black', label = 'Rolling Std')
    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show(block=False)

    #Perform Dickey-Fuller test:
    #print ('Results of Dickey-Fuller Test:')
    #dftest = adfuller(timeseries, autolag='AIC')
    #dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observations Used'])
    #for key,value in dftest[4].items():
     #   dfoutput['Critical Value (%s)'%key] = value
    #print (dfoutput)

#The function ends here
```

5.3.4. Developing and training model importing from statsmodels.tsa

- We have imported the ARIMA model from statsmodels.tsa which is a python module for models of time series analysis.
- Train the model with the dataset
- Plot the dataset

```
# there is a differencing or order 1 that’s why parameter d=1
# the model fitting is done through the training data
results_ARIMA = model.fit(disp=-1)
```

```
predictions_ARIMA_diff = pd.Series(results_ARIMA.fittedvalues, copy=True)
print (predictions_ARIMA_diff.head())

predictions_ARIMA_diff_cumsum = predictions_ARIMA_diff.cumsum()
print (predictions_ARIMA_diff_cumsum.head())

predictions_ARIMA = np.exp(predictions_ARIMA_log)
plt.plot(ts)
plt.plot(predictions_ARIMA)
plt.title('RMSE: %.4f'% np.sqrt(sum((predictions_ARIMA-ts)**2)/len(ts)))
```

5.4. Tensorflow.js Implementation

TensorFlow.js is a library for developing and training ML models in JavaScript, and deploying in browser or on Node.js. TensorFlow.js provides flexible building blocks for neural network programming in JavaScript. Works performed by Tensorflow.js are as follows:

- ❖ Run existing models - Use off-the-shelf JavaScript models or convert Python TensorFlow models to run in the browser or under Node.js.
- ❖ Retrain existing models - Retrain pre-existing ML models using your own data.
- ❖ Develop ML with JavaScript - Build and train models directly in JavaScript using flexible and intuitive APIs.

We have used Tensorflow.js for providing a UI to our machine learning model. The Dataset, Pre-processing steps, the forecast is display in the user-friendly and interacting interface usng tfvis visor.

- ➢ tfjs-vis provides some UI helpers to make it easier to render visualizations in an unobtrusive way
- ➢ We are constructing a visor with various tabs for efficiency including surface for visualizing time series analysis and forecast

```
<!DOCTYPE html>
<html>
<head>
```

```
<title> TensorFlow.js </title>

<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.7.3/p5.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.7.1/addons/p5.dom.min.js"></script>
<!--<script
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.7.1/addons/p5.sound.min.js"></script>-->
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@0.15.1/dist/tf.min.js"></script>
```

// Includng cdn for tfvis.visor library of tensorflow.js

```
<!--For visor -->
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs"> </script>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs-vis"></script>
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/d3/3.5.6/d3.min.js"
charset="utf-8"></script>
```

// Linking the Tenorflow.js javascript file with the main html file

```
<script type="text/javascript" src="complete.js"></script>
</head>
</body>
</html>
```

// Inside the javascript file

// setup() which is automatically triggered on page loading

```
function setup() {

}
```

// Making an HTML element a button whose onclick event will trigger the visor in our UI user interface

```
<<HTML element>>

    <section>
                <h2>Triggering the Visor</h2>
```

```
                <br><br>
                <button onclick="showVisor()">Show Visor</button>
                <br><br><br>
    </section>

<<-->>

//Triggering the visor with the following function

function showVisor() {

}
```

Chapter – VI

Validation

6. VALIDATION

6.1. Introduction

We have obtained a model for our time series that can now be used to produce forecasts. We start by comparing predicted values to real values of the time series, which will help us understand the accuracy of our forecasts.

It is important to evaluate forecast accuracy using genuine forecasts. Consequently, the size of the residuals is not a reliable indication of how large true forecast errors are likely to be.

6.2. Implementation

When choosing models, it is common practice to separate the available data into two portions, training and test data, where the training data is used to estimate any parameters of a forecasting method and the test data is used to evaluate its accuracy.

In our project of Time series forecasting using machine learning and tensorflow.js we have used pandas library for python for taking the input data which is divided in a ratio of:

70% Training Data

20% Testing Data

10% Validation Data

Validation Metric: RMSE

RMSE stands for Root Mean Squared Error

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results.

The formula is:

$$RMSE = \sqrt{\overline{(f - o)^2}}$$

Where,

f = forecasts (expected values or unknown results),

o = observed values (known results).

The bar above the squared differences is the mean (similar to x̄).

The same formula can be written with the following, slightly different, notation (Barnston, 1992):

$$\mathrm{RMSE}_{fo} = [\sum_{i=1}^{N} (z_{f_i} - z_{o_i})^2 / N]^{1/2}$$

Where,

Σ = summation ("add up")

(zfi – Zoi)Sup>2 = differences, squared

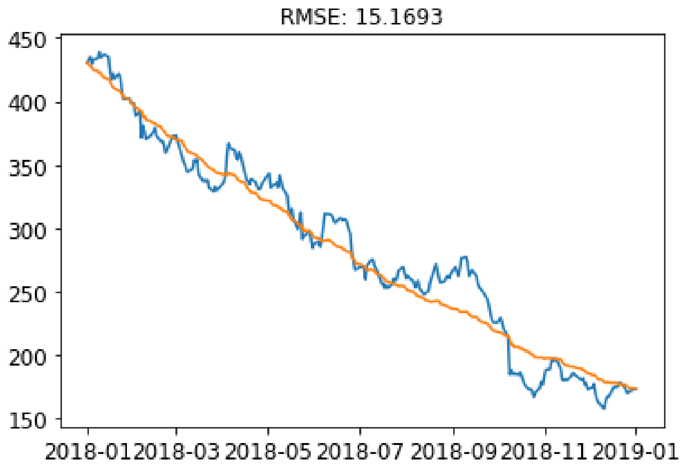
N = sample size

In our project, RMSE is calculated using numpy libraries in python between the actual time series and the predicted series.

The Validation section is mentioned as below with an RMSE value of 15.1693 on the TATAMOTORS average price using ARIMA model with model parameters, p=q=d=1.

```
In [22]: predictions_ARIMA = np.exp(predictions_ARIMA_log)
         plt.plot(ts)
         plt.plot(predictions_ARIMA)
         plt.title('RMSE: %.4f'% np.sqrt(sum((predictions_ARIMA-ts)**2)/len(ts)))

Out[22]: Text(0.5,1,'RMSE: 15.1693')
```



6.3. Importance

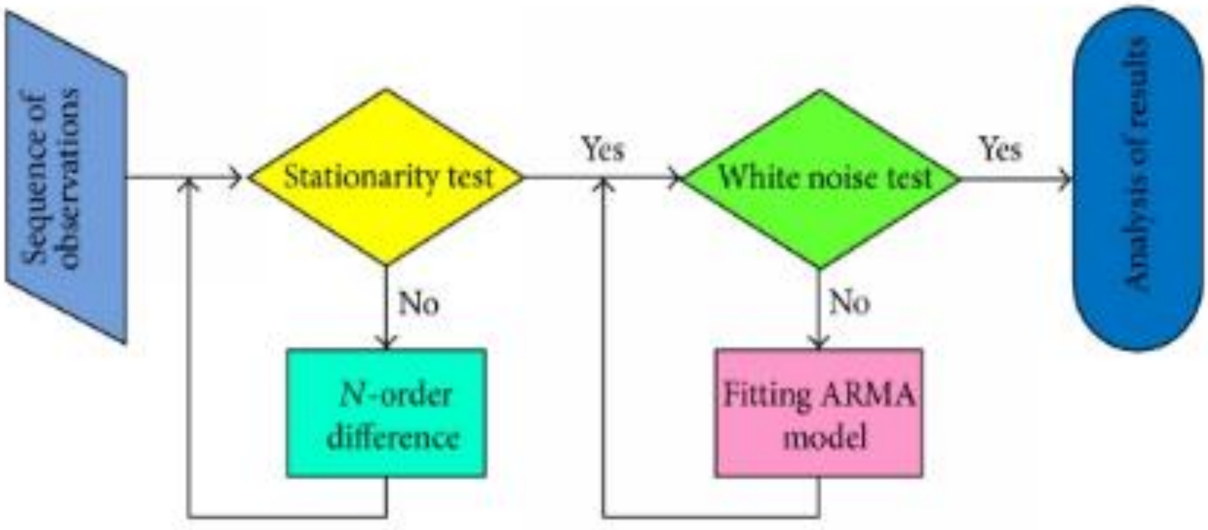
Validation is an important and essential step of constructing any model. It prevents overfitting in the standard model. When you use cross validation in machine learning, you verify how accurate your model is on multiple and different subsets of data. Therefore, you ensure that it generalizes well to the data that you collect in the future. It improves the accuracy of the model.

Chapter – VII

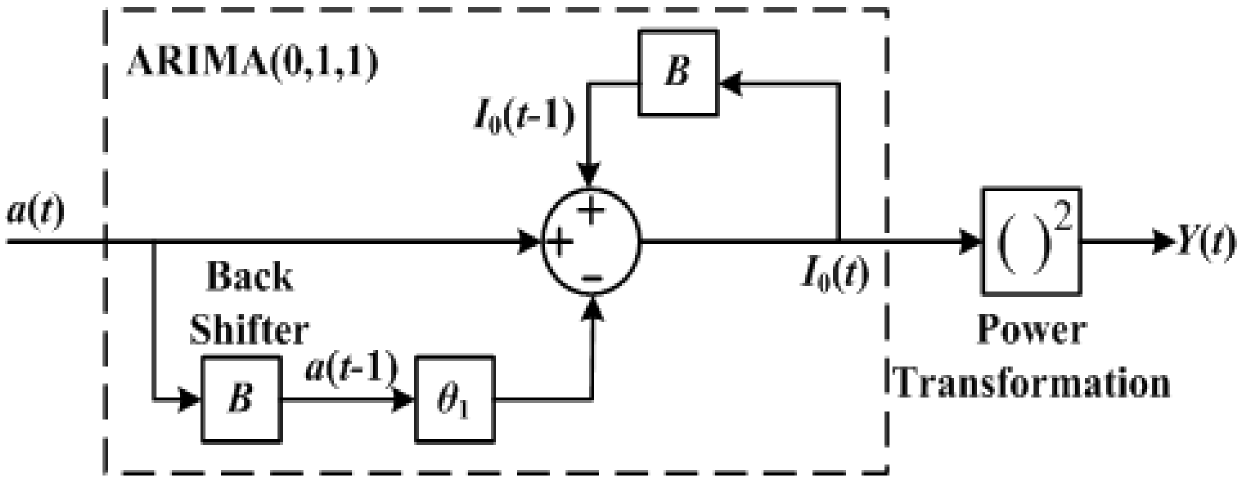
Diagrams

7. DIAGRAMS

7.1 ARIMA MODEL FLOW CHART

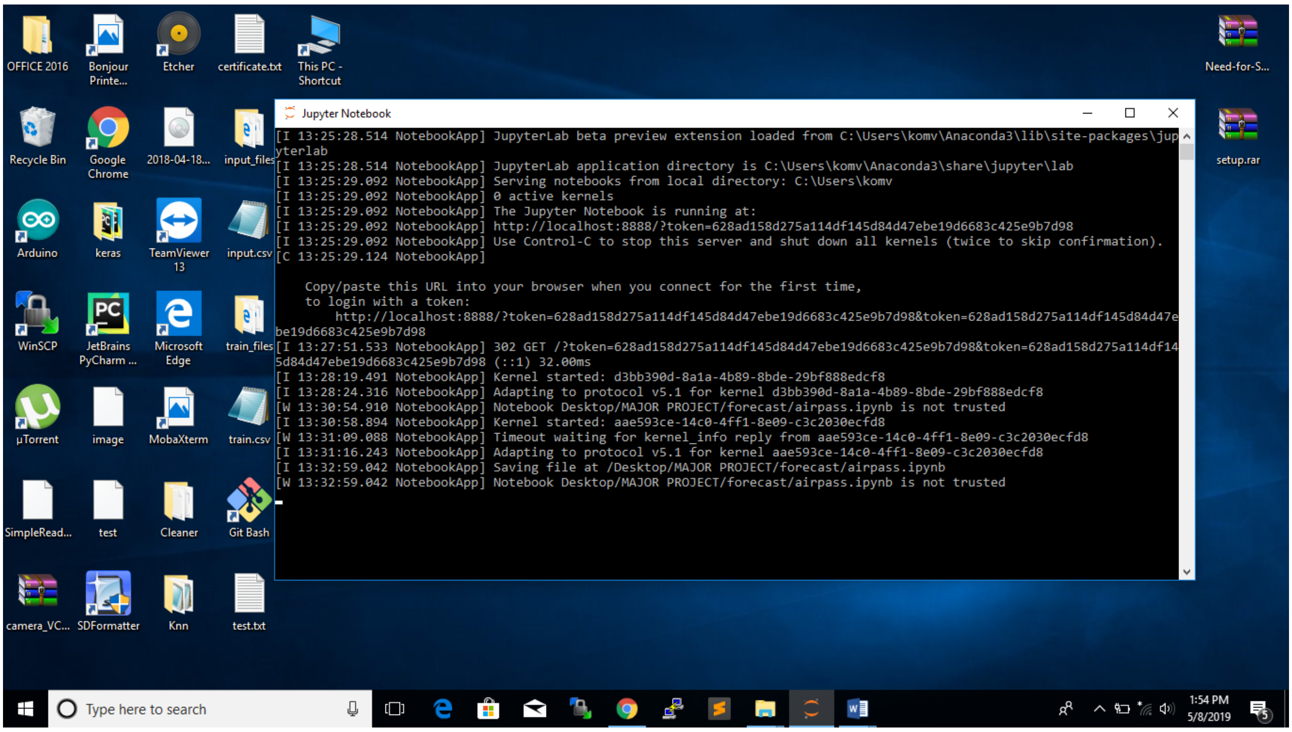


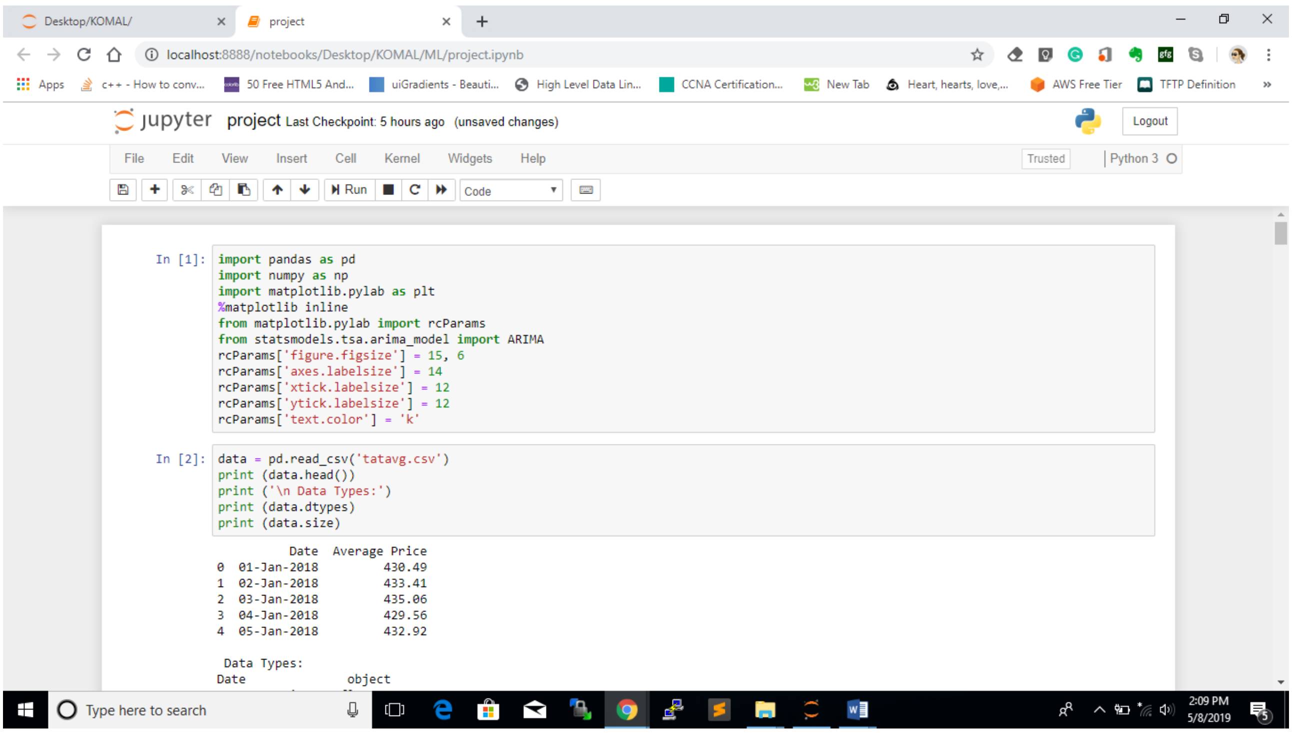
7.2 BLOCK DIAGRAM OF ARIMA MODEL

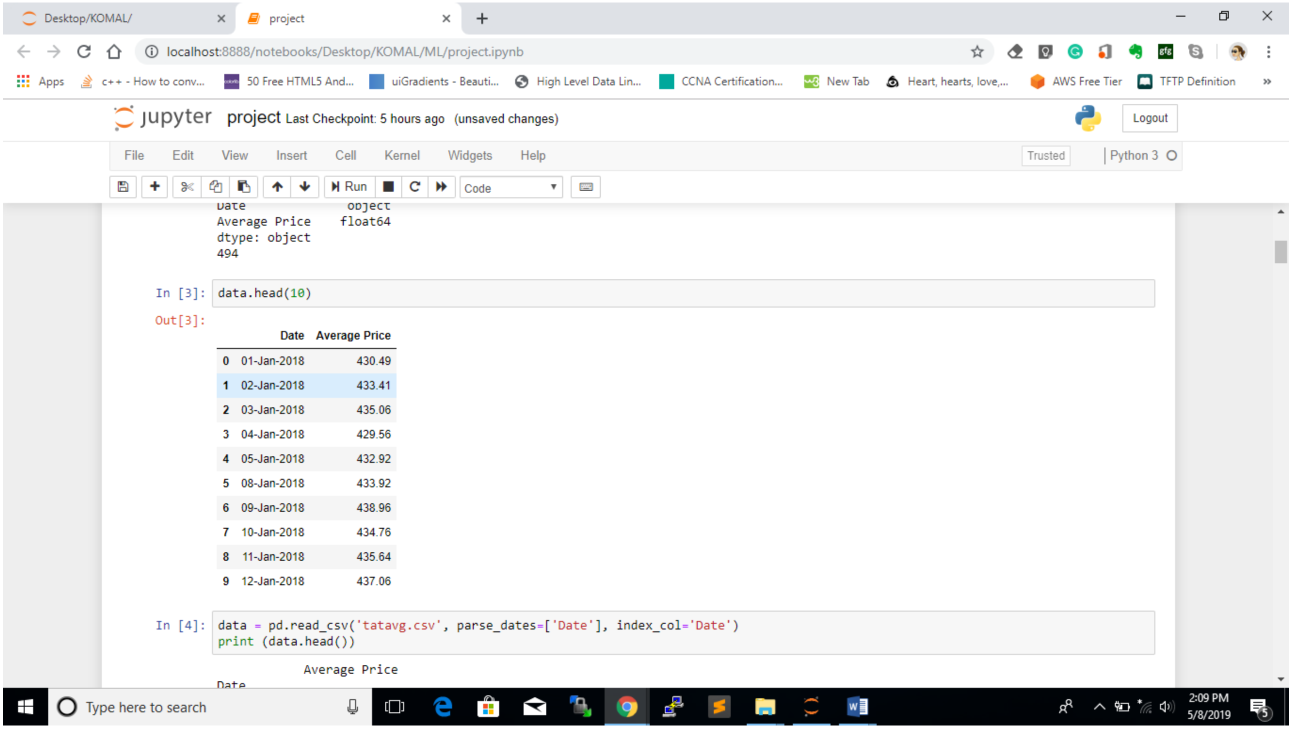


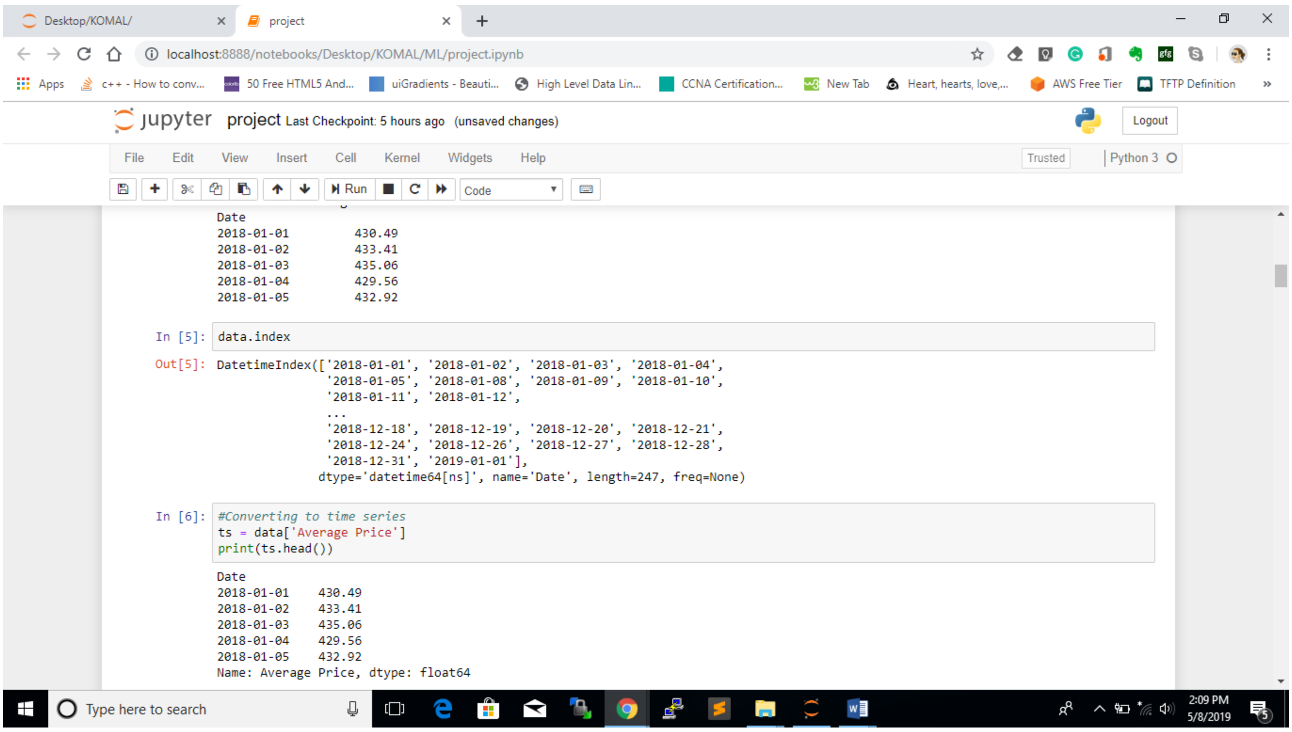
Chapter – VIII

Screenshots


Jupyter Notebook
[I 13:25:28.514 NotebookApp] JupyterLab beta preview extension loaded from C:\Users\komv\Anaconda3\lib\site-packages\jupyterlab
[I 13:25:28.514 NotebookApp] JupyterLab application directory is C:\Users\komv\Anaconda3\share\jupyter\lab
[I 13:25:29.092 NotebookApp] Serving notebooks from local directory: C:\Users\komv
[I 13:25:29.092 NotebookApp] 0 active kernels
[I 13:25:29.092 NotebookApp] The Jupyter Notebook is running at:
[I 13:25:29.092 NotebookApp] http://localhost:8888/?token=628ad158d275a114df145d84d47ebe19d6683c425e9b7d98
[I 13:25:29.092 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 13:25:29.124 NotebookApp]
Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=628ad158d275a114df145d84d47ebe19d6683c425e9b7d98&token=628ad158d275a114df145d84d47ebe19d6683c425e9b7d98
[I 13:27:51.533 NotebookApp] 302 GET /?token=628ad158d275a114df145d84d47ebe19d6683c425e9b7d98&token=628ad158d275a114df145d84d47ebe19d6683c425e9b7d98 (::1) 32.00ms
[I 13:28:19.491 NotebookApp] Kernel started: d3bb390d-8a1a-4b89-8bde-29bf888edcf8
[I 13:28:24.316 NotebookApp] Adapting to protocol v5.1 for kernel d3bb390d-8a1a-4b89-8bde-29bf888edcf8
[W 13:30:54.910 NotebookApp] Notebook Desktop/MAJOR PROJECT/forecast/airpass.ipynb is not trusted
[I 13:30:58.894 NotebookApp] Kernel started: aae593ce-14c0-4ff1-8e09-c3c2030ecfd8
[W 13:31:09.088 NotebookApp] Timeout waiting for kernel_info reply from aae593ce-14c0-4ff1-8e09-c3c2030ecfd8
[I 13:31:16.243 NotebookApp] Adapting to protocol v5.1 for kernel aae593ce-14c0-4ff1-8e09-c3c2030ecfd8
[I 13:32:59.042 NotebookApp] Saving file at /Desktop/MAJOR PROJECT/forecast/airpass.ipynb
[W 13:32:59.042 NotebookApp] Notebook Desktop/MAJOR PROJECT/forecast/airpass.ipynb is not trusted
Type here to search
1:54 PM
5/8/2019


localhost:8888/notebooks/Desktop/KOMAL/ML/project.ipynb
jupyter project Last Checkpoint: 5 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
In [1]:
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
%matplotlib inline
from matplotlib.pylab import rcParams
from statsmodels.tsa.arima_model import ARIMA
rcParams['figure.figsize'] = 15, 6
rcParams['axes.labelsize'] = 14
rcParams['xtick.labelsize'] = 12
rcParams['ytick.labelsize'] = 12
rcParams['text.color'] = 'k'
In [2]:
data = pd.read_csv('tatavg.csv')
print (data.head())
print ('\n Data Types:')
print (data.dtypes)
print (data.size)
Date Average Price
0 01-Jan-2018 430.49
1 02-Jan-2018 433.41
2 03-Jan-2018 435.06
3 04-Jan-2018 429.56
4 05-Jan-2018 432.92
Data Types:
Date object
Type here to search
2:09 PM
5/8/2019


Desktop/KOMAL/
project
localhost:8888/notebooks/Desktop/KOMAL/ML/project.ipynb
Apps
c++ - How to conv...
50 Free HTML5 And...
uiGradients - Beauti...
High Level Data Lin...
CCNA Certification...
New Tab
Heart, hearts, love,...
AWS Free Tier
TFTP Definition
jupyter
project Last Checkpoint: 5 hours ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Date object
Average Price float64
dtype: object
494
In [3]: data.head(10)
Out[3]:
Date Average Price
0 01-Jan-2018 430.49
1 02-Jan-2018 433.41
2 03-Jan-2018 435.06
3 04-Jan-2018 429.56
4 05-Jan-2018 432.92
5 08-Jan-2018 433.92
6 09-Jan-2018 438.96
7 10-Jan-2018 434.76
8 11-Jan-2018 435.64
9 12-Jan-2018 437.06
In [4]: data = pd.read_csv('tatavg.csv', parse_dates=['Date'], index_col='Date')
print (data.head())
Average Price
Date
Type here to search
2:09 PM
5/8/2019

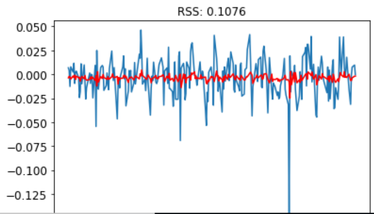

Desktop/KOMAL/
project
localhost:8888/notebooks/Desktop/KOMAL/ML/project.ipynb
jupyter
project Last Checkpoint: 5 hours ago (unsaved changes)
Logout
Trusted
Python 3
Date
2018-01-01 430.49
2018-01-02 433.41
2018-01-03 435.06
2018-01-04 429.56
2018-01-05 432.92
In [5]: data.index
Out[5]: DatetimeIndex(['2018-01-01', '2018-01-02', '2018-01-03', '2018-01-04',
'2018-01-05', '2018-01-08', '2018-01-09', '2018-01-10',
'2018-01-11', '2018-01-12',
...
'2018-12-18', '2018-12-19', '2018-12-20', '2018-12-21',
'2018-12-24', '2018-12-26', '2018-12-27', '2018-12-28',
'2018-12-31', '2019-01-01'],
dtype='datetime64[ns]', name='Date', length=247, freq=None)
In [6]: #Converting to time series
ts = data['Average Price']
print(ts.head())
Date
2018-01-01 430.49
2018-01-02 433.41
2018-01-03 435.06
2018-01-04 429.56
2018-01-05 432.92
Name: Average Price, dtype: float64
Type here to search
2:09 PM
5/8/2019

In [18]:

```
#ARIMA MODEL
model = ARIMA(ts_log, order=(2, 1, 2))
results_ARIMA = model.fit(disp=-1)
plt.plot(ts_log_diff)
plt.plot(results_ARIMA.fittedvalues, color='red')
plt.title('RSS: %.4f'% sum((results_ARIMA.fittedvalues-ts_log_diff)**2))
```

```
C:\Users\komv\Anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:225: ValueWarning: A date index has been provided,
but it has no associated frequency information and so will be ignored when e.g. forecasting.
  ' ignored when e.g. forecasting.', ValueWarning)
C:\Users\komv\Anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:225: ValueWarning: A date index has been provided,
but it has no associated frequency information and so will be ignored when e.g. forecasting.
  ' ignored when e.g. forecasting.', ValueWarning)
```

Out[18]: Text(0.5,1,'RSS: 0.1076')

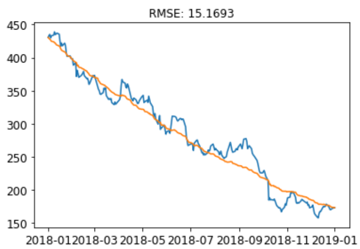


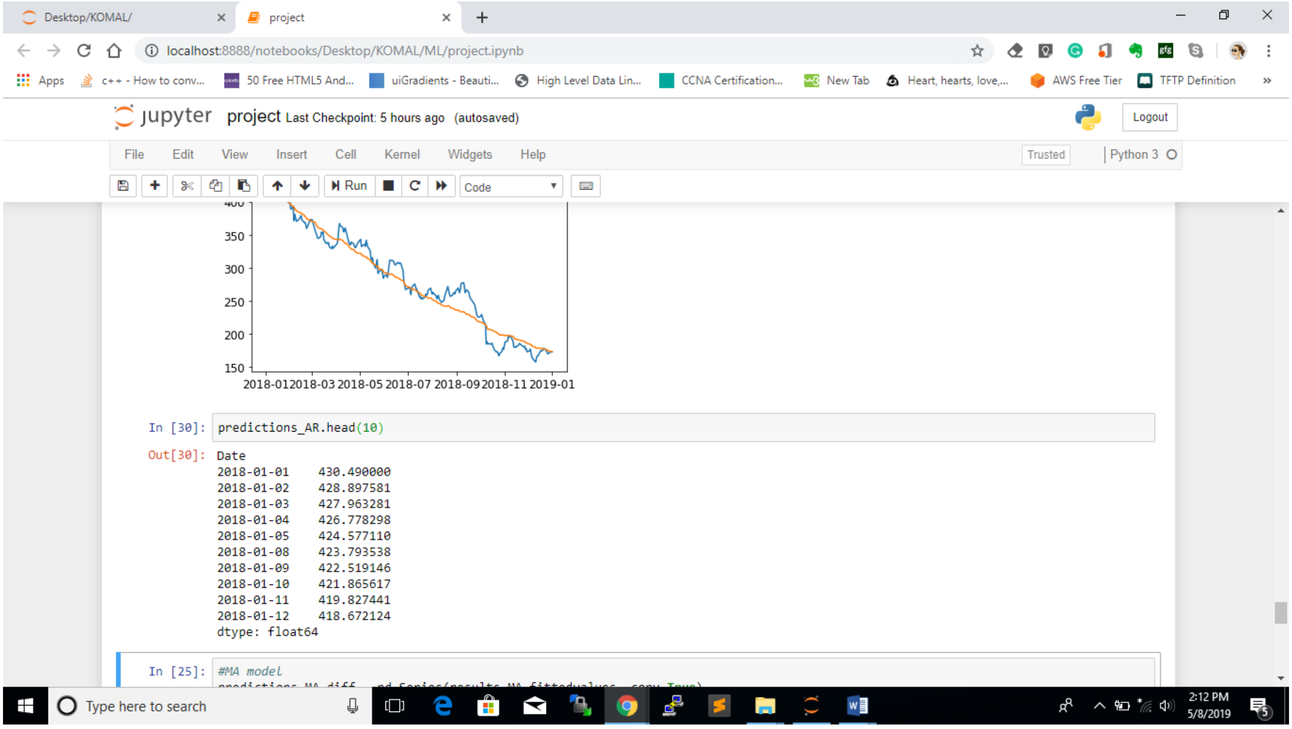

```
2018-01-01    6.064924
2018-01-02    6.061219
2018-01-03    6.059041
2018-01-04    6.056233
2018-01-05    6.051289
dtype: float64
```

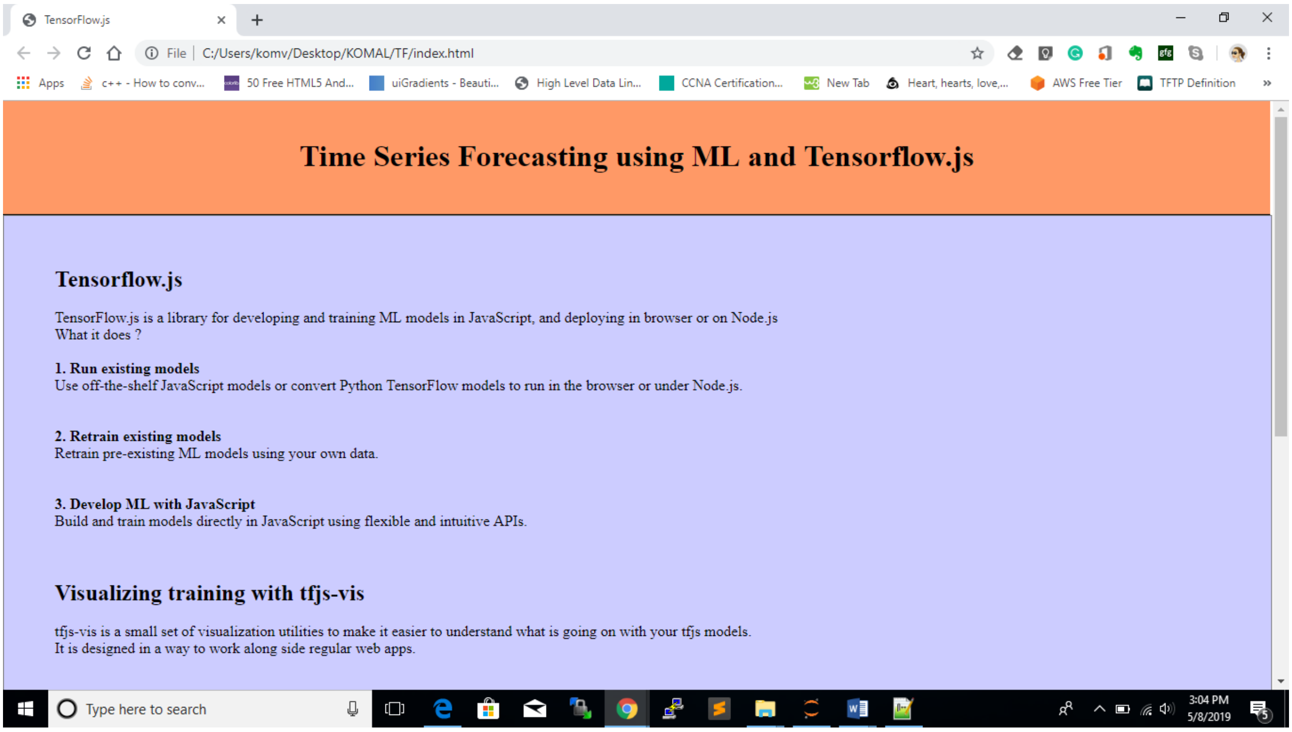
In [22]:

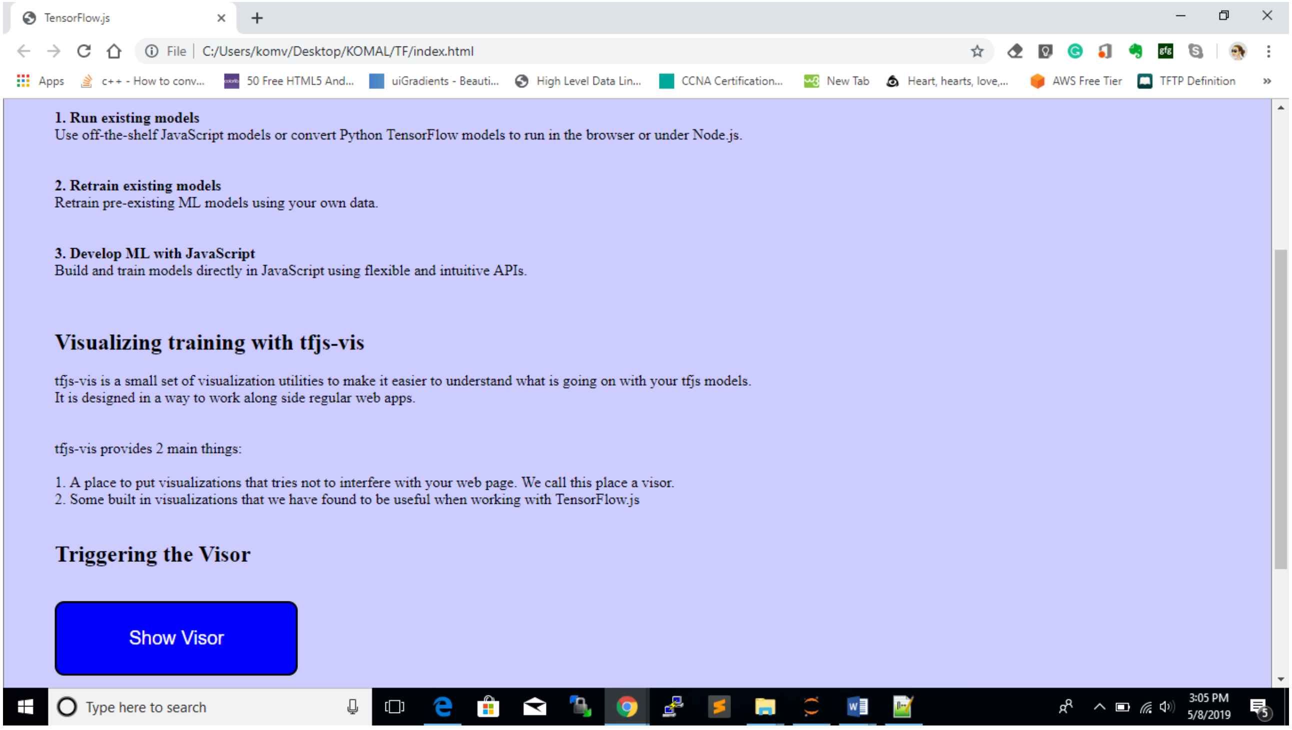
```
predictions_ARIMA = np.exp(predictions_ARIMA_log)
plt.plot(ts)
plt.plot(predictions_ARIMA)
plt.title('RMSE: %.4f'% np.sqrt(sum((predictions_ARIMA-ts)**2)/len(ts)))
```

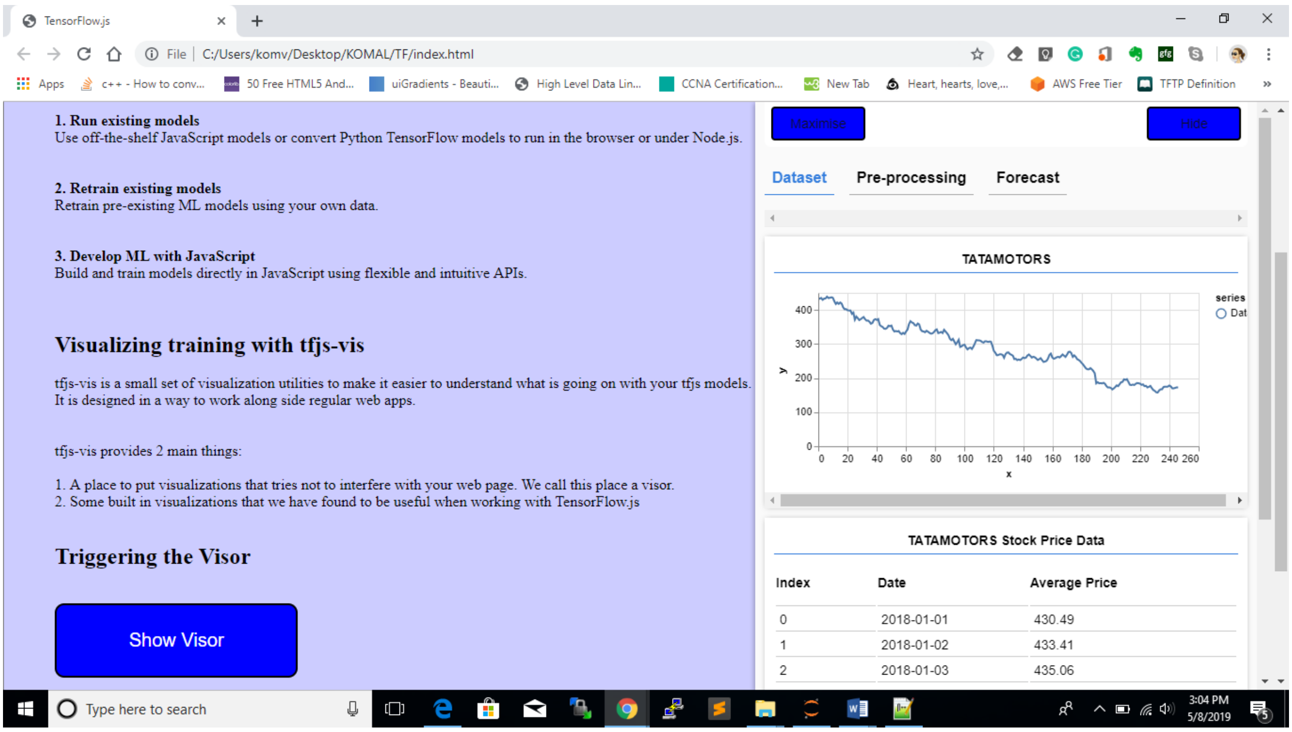
Out[22]: Text(0.5,1,'RMSE: 15.1693')

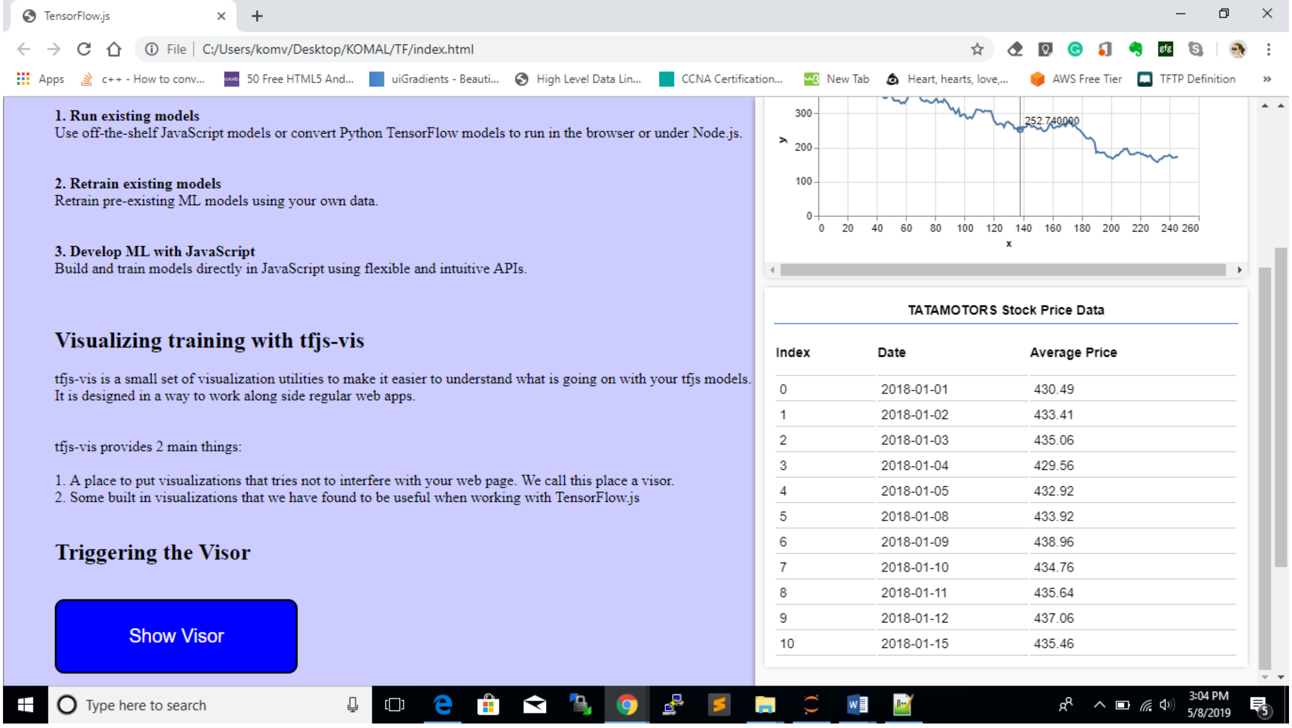


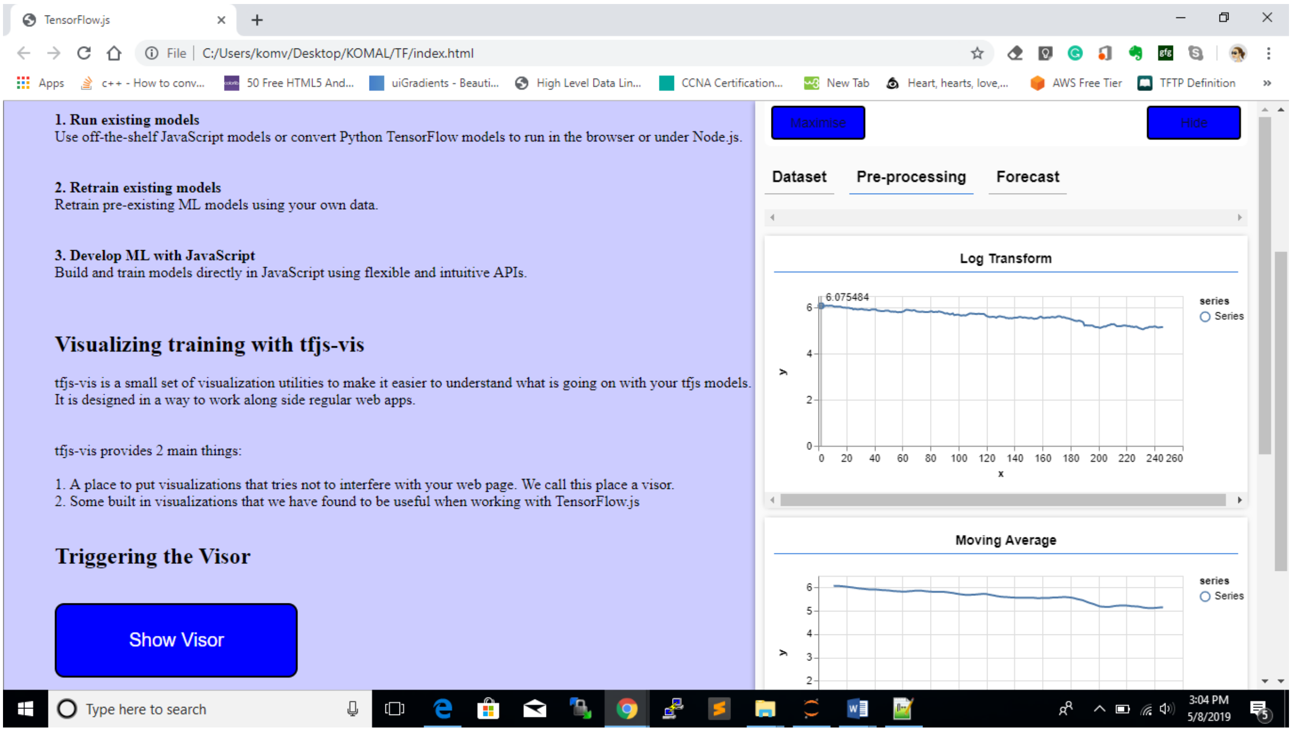

project Last Checkpoint: 5 hours ago (autosaved)
2018-01 2018-03 2018-05 2018-07 2018-09 2018-11 2019-01
In [30]: predictions_AR.head(10)
Out[30]: Date
2018-01-01 430.490000
2018-01-02 428.897581
2018-01-03 427.963281
2018-01-04 426.778298
2018-01-05 424.577110
2018-01-08 423.793538
2018-01-09 422.519146
2018-01-10 421.865617
2018-01-11 419.827441
2018-01-12 418.672124
dtype: float64
In [25]: #MA model

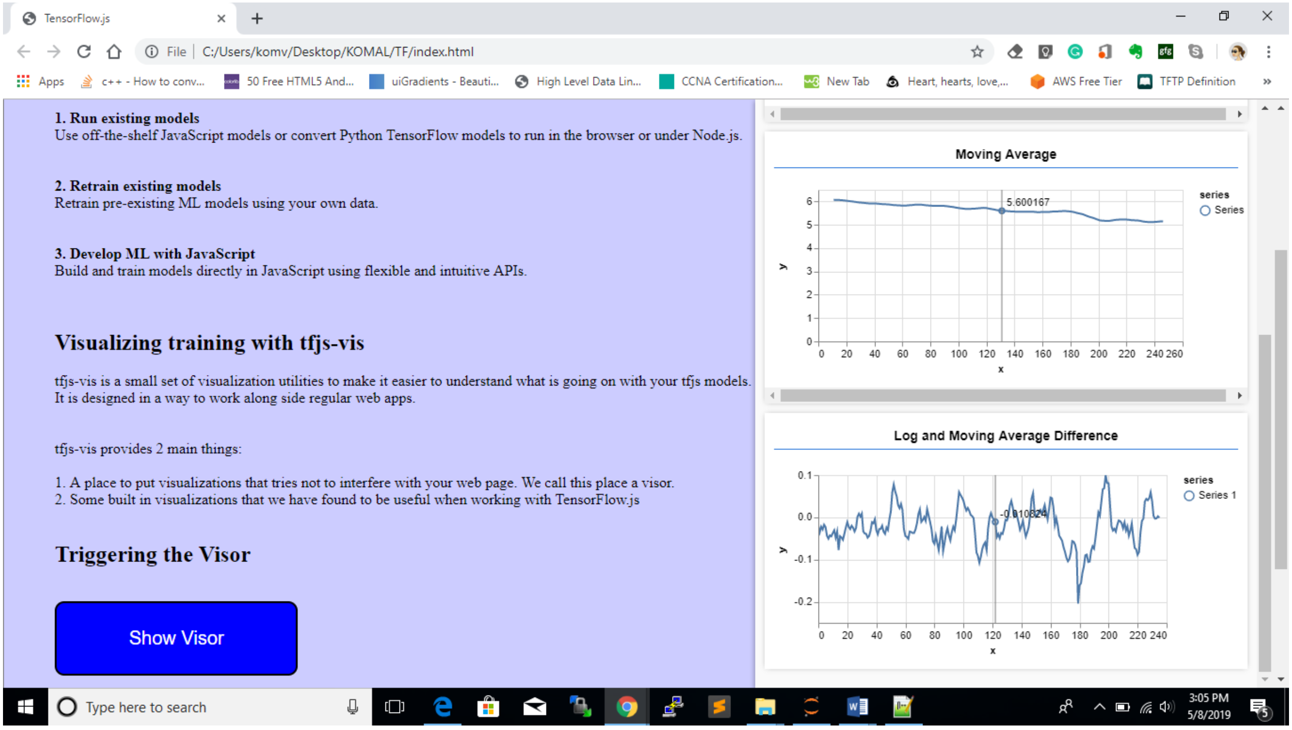

Time Series Forecasting using ML and Tensorflow.js
Tensorflow.js
TensorFlow.js is a library for developing and training ML models in JavaScript, and deploying in browser or on Node.js
What it does ?
1. Run existing models
Use off-the-shelf JavaScript models or convert Python TensorFlow models to run in the browser or under Node.js.
2. Retrain existing models
Retrain pre-existing ML models using your own data.
3. Develop ML with JavaScript
Build and train models directly in JavaScript using flexible and intuitive APIs.
Visualizing training with tfjs-vis
tfjs-vis is a small set of visualization utilities to make it easier to understand what is going on with your tfjs models.
It is designed in a way to work along side regular web apps.

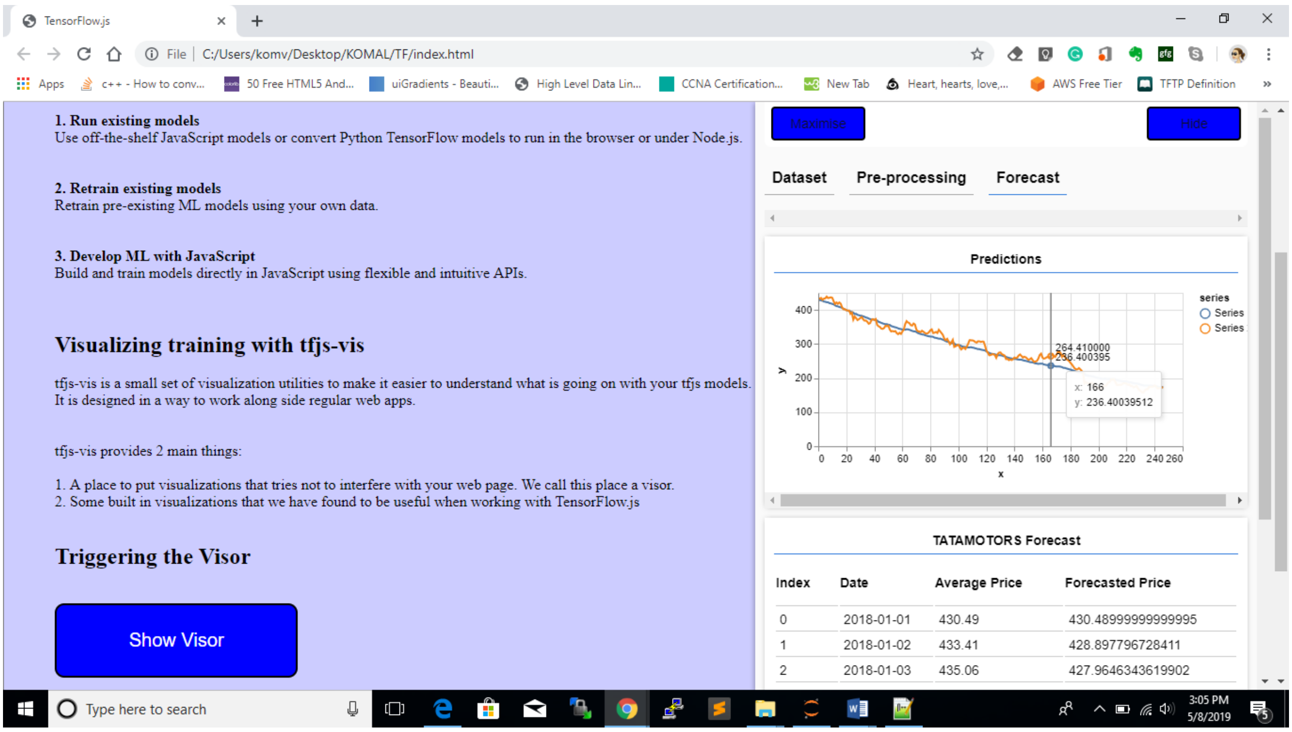

TensorFlow.js
C:/Users/komv/Desktop/KOMAL/TF/index.html
1. Run existing models
Use off-the-shelf JavaScript models or convert Python TensorFlow models to run in the browser or under Node.js.
2. Retrain existing models
Retrain pre-existing ML models using your own data.
3. Develop ML with JavaScript
Build and train models directly in JavaScript using flexible and intuitive APIs.
Visualizing training with tfjs-vis
tfjs-vis is a small set of visualization utilities to make it easier to understand what is going on with your tfjs models.
It is designed in a way to work along side regular web apps.
tfjs-vis provides 2 main things:
1. A place to put visualizations that tries not to interfere with your web page. We call this place a visor.
2. Some built in visualizations that we have found to be useful when working with TensorFlow.js
Triggering the Visor
Show Visor
Type here to search
3:05 PM
5/8/2019

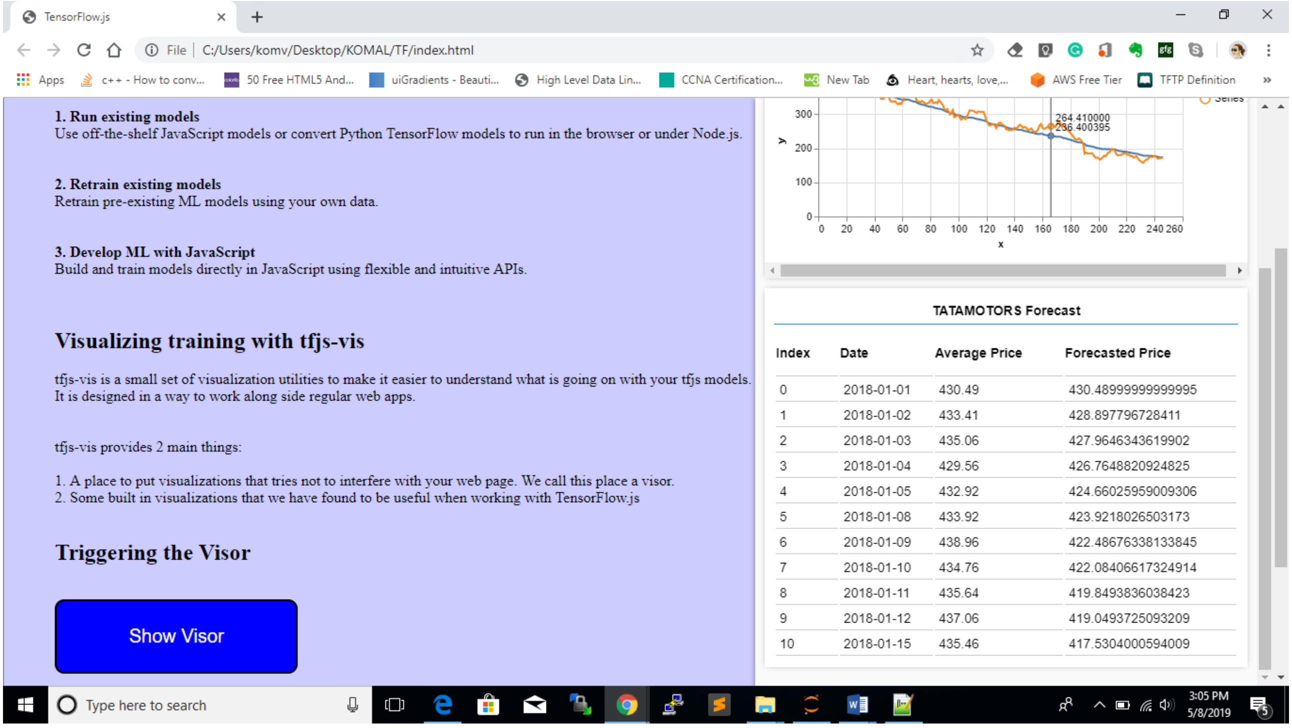

TensorFlow.js
C:/Users/komv/Desktop/KOMAL/TF/index.html
1. Run existing models
Use off-the-shelf JavaScript models or convert Python TensorFlow models to run in the browser or under Node.js.
2. Retrain existing models
Retrain pre-existing ML models using your own data.
3. Develop ML with JavaScript
Build and train models directly in JavaScript using flexible and intuitive APIs.
Visualizing training with tfjs-vis
tfjs-vis is a small set of visualization utilities to make it easier to understand what is going on with your tfjs models.
It is designed in a way to work along side regular web apps.
tfjs-vis provides 2 main things:
1. A place to put visualizations that tries not to interfere with your web page. We call this place a visor.
2. Some built in visualizations that we have found to be useful when working with TensorFlow.js
Triggering the Visor
Show Visor
Dataset
Pre-processing
Forecast
TATAMOTORS
series
Dat
y
x
TATAMOTORS Stock Price Data
Index Date Average Price
0 2018-01-01 430.49
1 2018-01-02 433.41
2 2018-01-03 435.06
Type here to search
3:04 PM
5/8/2019


TensorFlow.js
File | C:/Users/komv/Desktop/KOMAL/TF/index.html
Apps
c++ - How to conv...
50 Free HTML5 And...
uiGradients - Beauti...
High Level Data Lin...
CCNA Certification...
New Tab
Heart, hearts, love,...
AWS Free Tier
TFTP Definition
1. Run existing models
Use off-the-shelf JavaScript models or convert Python TensorFlow models to run in the browser or under Node.js.
2. Retrain existing models
Retrain pre-existing ML models using your own data.
3. Develop ML with JavaScript
Build and train models directly in JavaScript using flexible and intuitive APIs.
Visualizing training with tfjs-vis
tfjs-vis is a small set of visualization utilities to make it easier to understand what is going on with your tfjs models.
It is designed in a way to work along side regular web apps.
tfjs-vis provides 2 main things:
1. A place to put visualizations that tries not to interfere with your web page. We call this place a visor.
2. Some built in visualizations that we have found to be useful when working with TensorFlow.js
Triggering the Visor
Show Visor
300
200
100
0
y
252.740000
0 20 40 60 80 100 120 140 160 180 200 220 240 260
x
TATAMOTORS Stock Price Data
Index Date Average Price
0 2018-01-01 430.49
1 2018-01-02 433.41
2 2018-01-03 435.06
3 2018-01-04 429.56
4 2018-01-05 432.92
5 2018-01-08 433.92
6 2018-01-09 438.96
7 2018-01-10 434.76
8 2018-01-11 435.64
9 2018-01-12 437.06
10 2018-01-15 435.46
Type here to search
3:04 PM
5/8/2019


TensorFlow.js
File | C:/Users/komv/Desktop/KOMAL/TF/index.html
Apps
c++ - How to conv...
50 Free HTML5 And...
uiGradients - Beauti...
High Level Data Lin...
CCNA Certification...
New Tab
Heart, hearts, love,...
AWS Free Tier
TFTP Definition
1. Run existing models
Use off-the-shelf JavaScript models or convert Python TensorFlow models to run in the browser or under Node.js.
2. Retrain existing models
Retrain pre-existing ML models using your own data.
3. Develop ML with JavaScript
Build and train models directly in JavaScript using flexible and intuitive APIs.
Visualizing training with tfjs-vis
tfjs-vis is a small set of visualization utilities to make it easier to understand what is going on with your tfjs models.
It is designed in a way to work along side regular web apps.
tfjs-vis provides 2 main things:
1. A place to put visualizations that tries not to interfere with your web page. We call this place a visor.
2. Some built in visualizations that we have found to be useful when working with TensorFlow.js
Triggering the Visor
Show Visor
Maximize
Hide
Dataset
Pre-processing
Forecast
Log Transform
6.075484
series
Series
6
4
2
0
y
0 20 40 60 80 100 120 140 160 180 200 220 240 260
x
Moving Average
series
Series
6
5
4
3
2
y
Type here to search
3:04 PM
5/8/2019


TensorFlow.js
C:/Users/komv/Desktop/KOMAL/TF/index.html
1. Run existing models
Use off-the-shelf JavaScript models or convert Python TensorFlow models to run in the browser or under Node.js.
2. Retrain existing models
Retrain pre-existing ML models using your own data.
3. Develop ML with JavaScript
Build and train models directly in JavaScript using flexible and intuitive APIs.
Visualizing training with tfjs-vis
tfjs-vis is a small set of visualization utilities to make it easier to understand what is going on with your tfjs models. It is designed in a way to work along side regular web apps.
tfjs-vis provides 2 main things:
1. A place to put visualizations that tries not to interfere with your web page. We call this place a visor.
2. Some built in visualizations that we have found to be useful when working with TensorFlow.js
Triggering the Visor
Show Visor
Moving Average
5.600167
series
Series
Log and Moving Average Difference
series
Series 1
Type here to search
3:05 PM
5/8/2019


TensorFlow.js
C:/Users/komv/Desktop/KOMAL/TF/index.html
1. Run existing models
Use off-the-shelf JavaScript models or convert Python TensorFlow models to run in the browser or under Node.js.
2. Retrain existing models
Retrain pre-existing ML models using your own data.
3. Develop ML with JavaScript
Build and train models directly in JavaScript using flexible and intuitive APIs.
Visualizing training with tfjs-vis
tfjs-vis is a small set of visualization utilities to make it easier to understand what is going on with your tfjs models. It is designed in a way to work along side regular web apps.
tfjs-vis provides 2 main things:
1. A place to put visualizations that tries not to interfere with your web page. We call this place a visor.
2. Some built in visualizations that we have found to be useful when working with TensorFlow.js
Triggering the Visor
Show Visor
Dataset
Pre-processing
Forecast
Predictions
series
Series
Series
264.410000
236.400395
x: 166
y: 236.40039512
TATAMOTORS Forecast
Index Date Average Price Forecasted Price
0 2018-01-01 430.49 430.48999999999995
1 2018-01-02 433.41 428.897796728411
2 2018-01-03 435.06 427.9646343619902
Type here to search
3:05 PM
5/8/2019



1. Run existing models
Use off-the-shelf JavaScript models or convert Python TensorFlow models to run in the browser or under Node.js.

2. Retrain existing models
Retrain pre-existing ML models using your own data.

3. Develop ML with JavaScript
Build and train models directly in JavaScript using flexible and intuitive APIs.

Visualizing training with tfjs-vis

tfjs-vis is a small set of visualization utilities to make it easier to understand what is going on with your tfjs models. It is designed in a way to work along side regular web apps.

tfjs-vis provides 2 main things:

1. A place to put visualizations that tries not to interfere with your web page. We call this place a visor.
2. Some built in visualizations that we have found to be useful when working with TensorFlow.js

Triggering the Visor

Show Visor

TATAMOTORS Forecast

Index	Date	Average Price	Forecasted Price
0	2018-01-01	430.49	430.48999999999995
1	2018-01-02	433.41	428.897796728411
2	2018-01-03	435.06	427.9646343619902
3	2018-01-04	429.56	426.7648820924825
4	2018-01-05	432.92	424.66025959009306
5	2018-01-08	433.92	423.9218026503173
6	2018-01-09	438.96	422.48676338133845
7	2018-01-10	434.76	422.08406617324914
8	2018-01-11	435.64	419.8493836038423
9	2018-01-12	437.06	419.0493725093209
10	2018-01-15	435.46	417.5304000594009

Chapter – IX

Conclusion

9. CONCLUSION

After the completion of our project on Financial Forecasting using Machine Learning with Tensorflow.js we conclude that for completion of the project we used the concept of the machine learning with the ARIMA model. In this model we have calculated the three parameters p, d, q on the basis of the data-set that we use. The data-set that we use is of Tata-motors which have high and low price and we calculated the average price by using ARIMA model, AR model, AM model for checking the RMSE value, after checking this value for all the model we finally get the RMSE value of 15.1693 for the ARIMA model which have least as compared to all the above models and the graph obtained is smooth curve which justify that our model is good and showing the best fit.

Chapter – X

Advantages and Limitations

10. ADVANTAGES AND LIMITATIONS

10.1. Advantages:

1. The biggest advantage of using time series analysis – It can be used to understand the past as well as predict the future.
2. Extensively used in financial forecasting based on historical trends and patterns
3. Determines trends and patterns of future using graphs and other tools.
4. Data tendencies reporting from time series charts can be useful to managers when measurements show an increase or decrease in sales for a particular product or good.
5. The time series method is a useful tool to measure both financial and endogenous growth, according to Professor Hossein Arsham of the University of Baltimore.

10.2. Limitations:

1. Time series methods draw on vastly different areas in statistics, and lately, machine learning. You have to know a lot about all of these things, in general, to make sense of what you're doing. There is no real unification of the theory, either.
2. Most machine learning algorithms don't deal with time well.
3. This is hard stuff, and if you're not motivated by challenge, you can get overwhelmed. Also, there is, in some other areas of data science, the notion that all we use are ARIMA models and EWMA; while we do often use these tools, we also use RNN and LTSM networks and a whole lot of interesting things.

Chapter – XI

Future Scope

11. FUTURE SCOPE

As the scope of the project is concerned in future, the application has the tremendous scope as it is required for determining historic trends and seasonalities from the time perspective, define the patterns, and yield short or long-term predictions.

In future we plan to:

- Develop UI which accepts dynamic dataset for training and prediction
- Dynamic model which adjusts itself for accurate forecast according to the dynamic and real-time continuous data provided
- Launch this as an application software which will be beneficial for business forecasting in the financial field
- Provide an option for model selection and then continuing the forecast

This project opens up immense avenues of potential workflows. It would enable the model selection and forecasting after training with a single click.

With maximum types of trends and seasonalities covered and being trained, it would only be positive growth for this project but would enable this nation to empower and grow by managing the capital in advance through the financial forecast.

References

ONLINE REFERENCES

[1] Wikipedia
https://en.wikipedia.org/wiki/Time_series

[2] Time series forecasting
https://machinelearningmastery.com/time-series-forecasting/

[3] NSE – National Stock Exchange of India Ltd
https://www.nseindia.com/

[4] How to use Jupyter Notebooks – Codeacademy
https://www.codecademy.com/articles/how-to-use-jupyter-notebooks

[5] Pandas - 0.24.2 documentation
https://pandas.pydata.org/pandas-docs/stable/development/internals.html

[6] Python plotting — Matplotlib 3.0.3 documentation
https://matplotlib.org/

BOOKS AND RESEARCH PAPERS

[1] Artificial Intelligence – A Modern Approach
by Stuart J. Russell and Peter Norvig

[2] Artificial Intelligence
by Elaine Rich Senior Lecturer UTA, Kelvin Knight & Shivshankar N Nair

[3] The Elements of Statistical Learning
by Trevor Hastie, Robert Tibshirani and Jerome Friedman

[4] Stock Price Prediction Using the ARIMA Model
Published in 2014 under UKSim-AMSS 16th International Conference on Computer Modelling and Simulation
by Ayodele A. Adebiyi of School of Mathematic, Statistics & Computer Science
Aderemi O. Adewumi University of KwaZulu-Natal Durban, South Africa
Charles K. Ayo of Department of Computer & Information Sciences Covenant University
Ota, Nigeria

[5] Gold Price Forecasting Using ARIMA Model
Published on 2, March 2016 under Journal of Advanced Management Science Vol. 4
by Banhi Guha and Gautam Bandyopadhyay
Department of Management Studies, National Institute of Technology, Durgapur, India

Appendix