A Thesis/Project/Dissertation Report
On

STOCK FORETELL

STOCK PREDICTION USING ARIMA MODEL

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE & ENGINEERING

GALGOTIAS UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under the supervision of:
Dr. Vipin Rai
(Associate Professor of Galgotias University)

Summitted By:

Yogendra Kumar 18SCSE1010304

Kartik P Singh 18SCSE1010308

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA

DECEMBER,2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"STOCK FORETELL"** in partial fulfillment of the requirements for the award of the Bachelors of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Name **"Dr. Vipin Rai"** Designation, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering, Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by us for the award of any other degree of this or any other places.

Yogendra Kumar 18SCSE1010304

Kartik P Singh 18SCSE1010308

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor Name

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **"Yogendra Kumar (18SCSE1010304)"** and **"Kartik P Singh (18SCSE1010308)"** have been held on _________________ and their work is recommended for the award of B. Tech (CSE).

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date: December, 2021

Place: Greater Noida

ACKNOWLEDGEMENT

Apart from the efforts of our, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful

completion of this project.

We would like to show my greatest appreciation to "**Dr. Vipin Rai"** and also our dean "**Dr. Munish Sabharwal"**. We can't say thank you enough for his tremendous support and help. We feel motivated and encouraged every time, we attend his meeting. Without his encouragement and guidance this project would not have materialized.

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Abstract

- Stock price prediction is an important topic in finance and economics which has spurred the interest of researchers over the years to develop better predictive models. The autoregressive integrated moving average (ARIMA) models have been explored in literature for time series prediction. This paper presents extensive process of building stock price predictive model using the ARIMA model.

- Published stock data obtained from New York Stock Exchange (NYSE) and Bombay Stock Exchange (BSE) are used with stock price predictive model developed. Results obtained revealed that the ARIMA model has a strong potential for short-term prediction and can compete agree with existing techniques for stock price prediction.

- The ARIMA model has been widely utilized in banking and economics since it is recognized to be reliable, efficient, and capable of predicting short-term share market movements.

List of Tables

List of Figures

Table of Contents

Chapter-1

Introduction

Auto Regressive Integrated Moving Average (ARIMA) model is among one of the more popular and widely used statistical methods for time-series forecasting. It is a class of statistical algorithms that captures the standard temporal dependencies that is unique to a time series data. In this post, I will introduce you to the basic principles of ARIMA and present a hands-on tutorial to develop ARIMA for time-series forecasting in Python.

we dig right into ARIMA's formal mathematical definition, let me introduce you to the concept of **stationarity**. Stationarity simply means observations that do not depend on time. For data that depends on time (eg. seasonal rainfall), the stationarity condition may not hold as different timing will yield different values for these observations.

Another important concept to understanding ARIMA is **autocorrelation**. How does it different from the typical correlation? First of all, correlation relates two different sets of observations (eg. between housing prices and the number of available public amenities) while autocorrelation relates the same set of observation but across different timing (eg. between rainfall in the summer *versus* that in the fall)

1.2 Formulation of Problem:

Numerous forecasting models have been proposed to find an effective method that can be applied to practical situations. These techniques mostly rely on complex statistics, artificial intelligence techniques, and large amounts of meteorological and topographic data. Ideally, these methods minimize the risk of failure within the energy system and forecast its reliability by modeling or simulating future scenarios. The available prediction models can be classified into three main categories:

(i) Qualitative techniques, (ii) Quantitative techniques, and (iii) Artificial neural networks (ANNs).

(i) Qualitative techniques are based on expert opinion and/or personal judgment.

(ii) Quantitative techniques are based on mathematical models, which can be further classified as time series or causal forecasting techniques. Causal forecasting is used to identify relationships between dependent and independent variables. The quality of causal forecasting models depends on the accuracy of the input factors. Due to the high fluctuation of factors affecting solar radiation, however, the availability and accuracy of these models are questionable. Time series forecasting models collect observations over a designated period of time, where every observation represents a specific time (t) and then predicts future outputs according to previous events. Compared with causal forecasting, time series forecasting is flexible and requires fewer data inputs; thus the technique is easier to implement and does not require much cost. However, the main limitation of time series forecasting is the lack of a deterministic cause. To overcome this limitation, model developers usually depend on large numbers of inputs or stochastic events.

(iii) Artificial neural networks (ANNs), which have been used in several studies [12–14]. On the other hand, the authors in References [15–18] showed that a radial basis function neural networks (RBF-NN) can be applied to a wide range of nonlinear equation sets. The authors et al. proposed the RBF-NN for nonlinear mapping, which is exploited to solve a nonlinear equation set of load flow analysis. While, Referencess applied the RBF-NN technique into microgrids.

Despite the benefits of ANN, however, a previous study [19] investigated the performance of the time series auto-regressive integrated moving average (ARIMA) model in comparison with ANN models and found that the former generally performs better than the latter due to the effect of weather conditions, such as clouds. Solar radiation concentration is partially dependent on various weather, location, and time factors; thus, it displays a type of serial correlation, which suggests that time series forecasting is appropriate for solar radiation forecasting.

The unknown working principle or Symmetry 2019, 11, 240 3 of 17 “black-box” of neural networks limits their applicability in predicting solar radiation. The Numerical Weather Prediction (NWP) is widely available in meteorological organizations. However, NWP is highly dependent on air quality and hydrological characteristics, which strongly vary with time and are sensitive to location [20]. The direct implementation of NWP in solar radiation forecasting has been criticized [21]. Therefore, we considered developing a time series forecasting technique in this study because of its convenience and accurate prediction, low data input requirement, and simple computational process.

The forecast procedure can provide a rapid and standard way to generate forecasts for many time series in a single step. In the past, hundreds of series were forecasted at a time, with the series organized into separate variables or across groups. ARIMA is regarded as a smooth technique, and it is applicable when the data is reasonably long and the correlation between past observations is stable [22]. Several studies in the literature have used ARMA and ARIMA models for solar radiation prediction [23–26].

The ARMA and ARIMA models have also been compared in terms of the goodness-of-fit values produced by the log-likelihood function. As a result, the best statistical models and corresponding parameters for solar radiation prediction can be determined comprehensively. Many feasible comparisons have been conducted for solar radiation prediction. In previous work, the prediction task of many models lacked adequacy and timing in terms of data collection.

Here, a time series ARIMA model is built to forecast the daily and monthly solar radiation of Seoul, South Korea in consideration of the accuracy, suitability, adequacy, and timeliness of the collected data, which have been obtained from KMS over 37 years. The reliability, accuracy, suitability, and performance of the model are investigated in comparison with those of established tests, such as standardized residual, ACF, and PACF. Finally, the obtained results are compared with those forecasted by the Monte Carlo method.

1.2.1 Tools and Technology Used:

1.2.1.1 Hardware Requirements: Processor RAM Disk Space Pentium II, Pentium III, Pentium IV or higher 64 Mb or Higher 130Mb, Minimum 8 Gb RAM.

1.2.1.2 Software Requirements: Operating System DatabaseWin-98, Win-XP, Linux or any other higher version MS Access.

1.2.1.3 Others Tools:

- Google Collab
- Anaconda
- Azure
- python.

Chapter-2

Literature Survey

Kalid Yunus et al [1] presents this paper, a modified auto regressive integrated moving average (ARIMA) modeling technique which could capture time correlation and possibility distribution of determined wind-pace timecollection records is offered. The technique introduces frequency decomposition (splitting the wind-speed information into high frequency (HF) low-frequency (LF) components), shifting, and limiting further to differencing and energy transformation that are used within the trendy ARIMA modeling system.

Vaccaro et al [2] the paper proposes hybrid architecture for electricity price forecasting. The proposed architecture combines the reward of the easy-to-use and comparatively simple to- tune Auto regressive Integrated Moving Average (ARIMA) model and the approximation power of local learning techniques. The architecture is robust and more accurate than the individual forecasting methodologies on which it is based, since it combines a reliable built-in linear model (ARIMA) with an adaptive dynamic corrector (Lazy Learning algorithm). The corrector model is sequentially updated, in order to adjust the whole architecture to varying market conditions. Detailed simulation studies show the effectiveness of the proposed hybrid learning methods for forecasting the volatile Hourly Ontario Energy Prices (HOEPs) of the Ontario, Canada, and electricity market.

Guoqiang Liu et al [3] in the software program reliability boom section, the character of the failure records in an experience, decided by means of the software testing method. A hybrid version is proposed for medium and lengthy-term software program failure time forecasting on this paper. The hybrid version consists of two techniques, Singular Spectrum Analysis (SSA) and ARIMA on this version, the time series of software failure time are firstly decomposed into numerous sub-series corresponding to some tendentious and oscillation (periodic or quasiperiodic) components and noise by using the usage of SSA after which every sub-series is expected, respectively, through the best ARIMA version, and lastly a correction system is conducted for the sum of the prediction results to ensure the residual to be a natural random collection.

Takaomi HIRATA et al [4] Time series records examine and prediction may be very essential to the observe of nonlinear phenomenon. Studies of time series prediction have protected records when you consider that last century, linear models along with ARIMA, and nonlinear models consisting of multi-layer perception (MLP) are well-known. Because the nation-of-art approach, a deep belief net (DBN) using a couple of Restricted Boltzmann machines (RBMs) changed into proposed these days. In this have a look at advise a novel prediction method which composes not simplest a form of DBN with RBM and MLP however additionally ARIMA. Prediction experiments for the time collection of the actual records and chaotic time collection were performed, and effects confirmed the effectiveness of the proposed method.

Ling Wang et al [5] presents Based totally on the evaluation of the measured put on records and the wear characteristics in terms the wheels of Guangzhou Metro Line1, the accumulative put on prediction approach of metro wheels based totally on the ARIMA (p, d, q) model is proposed in this paper. According to the time series modeling method of the ARIMA (p, d, q) model, the stationarity evaluation and transformation of the metro wheel Z. Asha Farhath et al, International Journal of Computer Science and Mobile Computing, Vol.5 Issue.8, August- 2016, pg. 104-109 © 2016, IJCSMC All Rights Reserved 106 wear records are described at first. Then, with the application of the AIC criterion and the most likelihood

Estimation method, the model order is determined and the version parameters are derived for the ARIMA (p, d, q) model in the end, the flange thickness and the diameter of the metro wheels will be anticipated by means of this advanced ARIMA(p, d, q) model. The effects show that the proposed prediction method is straightforward and effective for the quick-term prediction of the metro wheel.

Theresa Hoang Diem Ngo et al [6] a time series is a put of ethics of a exacting variable that happen over a age of time in a positive pattern. The most ordinary patterns are rising or falling trend, cycle, seasonality, and uneven fluctuations. To replica a time sequence event as an occupation of its history values, analysts recognize the outline with the supposition that the pattern motivation keep on in the future.

Chapter 3

System Requirements

3.1 Introduction:-

Image processing is a technique than can convert an image into digital form and perform different kinds of operation on it for getting better image and useful information. Image processing technique used two types of method. These are analog and digital image processing. Analog technique can be used for hard copies and digital technique used for manipulating digital image. The purpose of image processing is divided into five groups. These are: visualization, image sharpening, image retrieval, measurement of pattern, image recognition. Visualization observe invisible object. Image sharpening that makes better image. Image retrieval finds interesting image. Measurement of pattern measures different objects of an image. Image recognition finds the difference of an image.

3.1.1 Machine Learning Algorithms:-

One of the most important applications of artificial intelligence is machine learning.

It provides the application that can automatically learn and improve from experience

without being apparently

programmed. The learning process starts with observations or data. Such as, we can assume a good decision based on direct experience or instruction. The basic aim is to allow the device without human interruption. Mostly machine learning algorithm is classified into two types. These are supervised and unsupervised learning.

a) Supervised Machine Learning Algorithms:

Supervised learning algorithms able to do different analysis with new data based on what it learned from the past and can also predict future event. The supervised learning algorithms create deduced function for predicting the starting analysis of known training data and output values. After some effective training the system makes a target for any new inputs. The system is able to compare its output with correct output and also find error for modification.

b) Unsupervised Machine Learning Algorithms:

Unsupervised machine learning algorithms are used for training unclassified and those data which are not leveled. Unsupervised learning is able to describe a secret shape from unleveled data. This system can’t provide proper output but it is able to take important decision from data set for describing secret shape from unleveled data.

c) Semi-supervised Machine Learning Algorithms:

Semi-supervised machine algorithms lies between supervised and unsupervised learning. For training they use both leveled and unleveled data but in this training data there are a small amount of leveled data and a huge amount of unleveled data. By using this method the systems are able to develop learning exactitude. Normally semi-supervised learning algorithms are used when leveled data need proficient and relevant resource for training.

3.1.2 <u>DEEP LEARNING:-</u>

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It’s achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

Deep learning achieves recognition accuracy at higher levels than ever before. This helps consumer electronics meet user expectations, and it is crucial for safety-critical applications like driverless cars. Recent advances in deep learning have improved to the point where deep learning outperforms humans in some tasks like classifying objects in images.

 Benefits of Deep Learning:-

- Has best-in-class performance on problems that significantly outperforms other solutions in multiple domains. This includes speech, language, vision, playing games like Go

etc. This isn’t by a little bit, but by a significant amount.

- Reduces the need for feature engineering, one of the most time-consuming parts of machine learning practice.

- Is an architecture that can be adapted to new problems relatively easily e.g. Vision, time series, language etc., are

   using techniques like convolutional neural networks, recurrent neural networks, long short-term memory etc.

- <u>DISADVANTAGE:-</u>

- Requires a large amount of data — if you only have thousands of examples, deep learning is unlikely to outperform other approaches.

- Is extremely computationally expensive to train. The most complex models take weeks to train using hundreds of machines equipped with expensive GPUs.

- Do not have much in the way of strong theoretical foundation. This leads to thenext disadvantage.

- Determining the topology/flavor/training method/hyper parameters for deep learning is a black art with no theory to guide you.

- What is learned is not easy to comprehend. Other classifiers (e.g. decision trees, logistic regression etc) make it much easier to understand what’s going on.

3.1.3 CONVOLUTIONAL NEURAL NETWORK (CNN):-

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery. CNNs use a variation of multilayer perceptions designed to require minimal pre-processing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage. They have applications in image and video recognition, recommender systems, image classification, medical image analysis, and natural language processing.

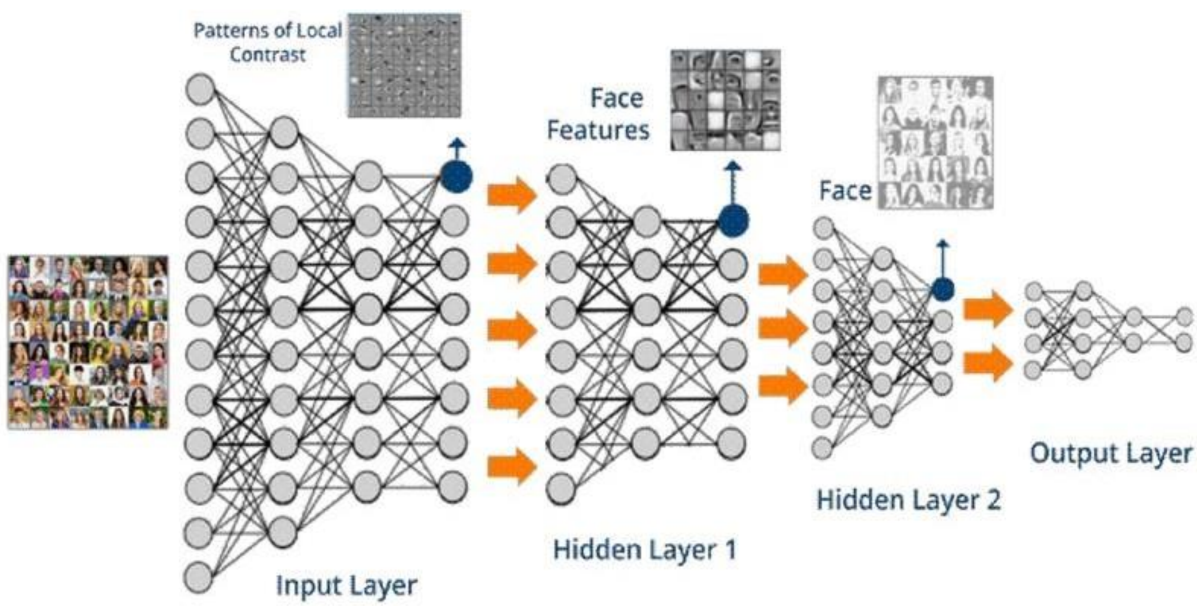


Fig-1 Structure of CNN

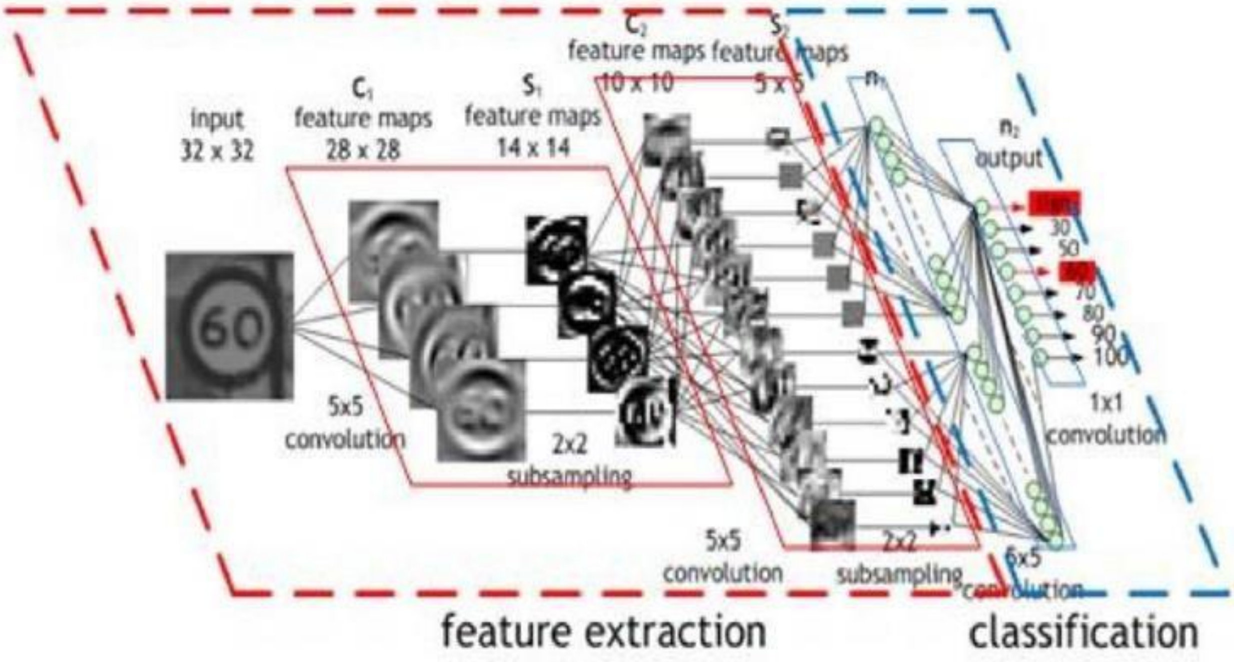


Fig-2 Example of CNN

3.2 PREREQUISITES OF THE PROJECT:

1. Keras2.0.2:-

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano.It was developed with a focus on enabling fast experimentation. Keras is an open source neural network library written in Python.It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or MXNet. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is François Chollet, a Google engineer.

2. Tensorflow1.1.0:-

TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains.

3. <u>Pandas 0.19.1:-</u>

Pandas is an open source, BSD-licensed library providing high- performance, easy to-use data structures and data analysis tools for thePython programming language.

Pandas is aNumFOCUS sponsored project. pandas is an open source, BSD licensed library providing high-performance, easy-to- use data structures and data analysis tools for the Python programming language. pandas is a NumFOCUS sponsored project.

4. <u>Opencv2-python 3.2.0:-</u>

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. Python is a general-purpose programming language started by Guido van Rossum that became very popular very quickly, mainly because of its simplicity and code readability. It enables the programmer to express ideas in fewer lines of code without reducing readability.

OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib. OpenCV supports a wide variety of programming languages such as C++, Python, Java, etc., and is available on different platforms including Windows, Linux, OS X, Android, and iOS. Interfaces for high-speed GPU operations based on CUDA and OpenCL are also under active development. OpenCV-Python is the Python API for OpenCV, combining the best qualities of the OpenCV C++ API and the Python language.

5 .PyCharm3.5:-

PyCharm is an Integrated Development Environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains.It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django. We have used PyCharm as the IDE in our project.

3.3 FUNCTIONAL REQUIREMENTS:-

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases.

Here, the system has to perform the following tasks:

- Take the real time input of the person from the web cam.
- Identify the face and extract the facial features
- Based on the trained data, classify the emotion/gender and if any object seen.
- The recognized emotion is given as output in the form of a speech

3.4 NON-FUNCTIONAL REOUIREMENTS:-

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements". Some of the quality attributes are as follows:

3.4.1 ACCESSIBILITY:-

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible.

The system will be accessible to a lot of people as it will be incorporated with virtualAssociates and home service robots.

3.4.2 MAINTAINABILITY:-

In software engineering, maintainability is the ease with which a software productcan be modified in order to:

- Correct defects
- Meet new requirements

Since the project will be implemented using python libraries it is easier to add or modifythe code since it won't have a larger code.

3.4.3 SCALABILITY:-

System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added. System can work normally under situations such as low bandwidth and large number of users

3.4.4 PORTABILITY:-

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed under different operation conditions provided it meet its minimum configurations. Only system files and dependant assemblies would have to be configured in such case.

3.5 HARDWARE REQUIREMENTS:-

Processor : Any Processor above 500 MHz

RAM : 512Mb

Hard Disk : 10 GB

Input device : Standard Keyboard & Mouse,

Webcam Outputdevice : VGA and High Resolution Monitor

3.6 SOFTWARE REQUIREMENTS:-

Operating system : Windows XP or

above Front End : Python

Chapter-4

Project Design

Data Flow Diagram:

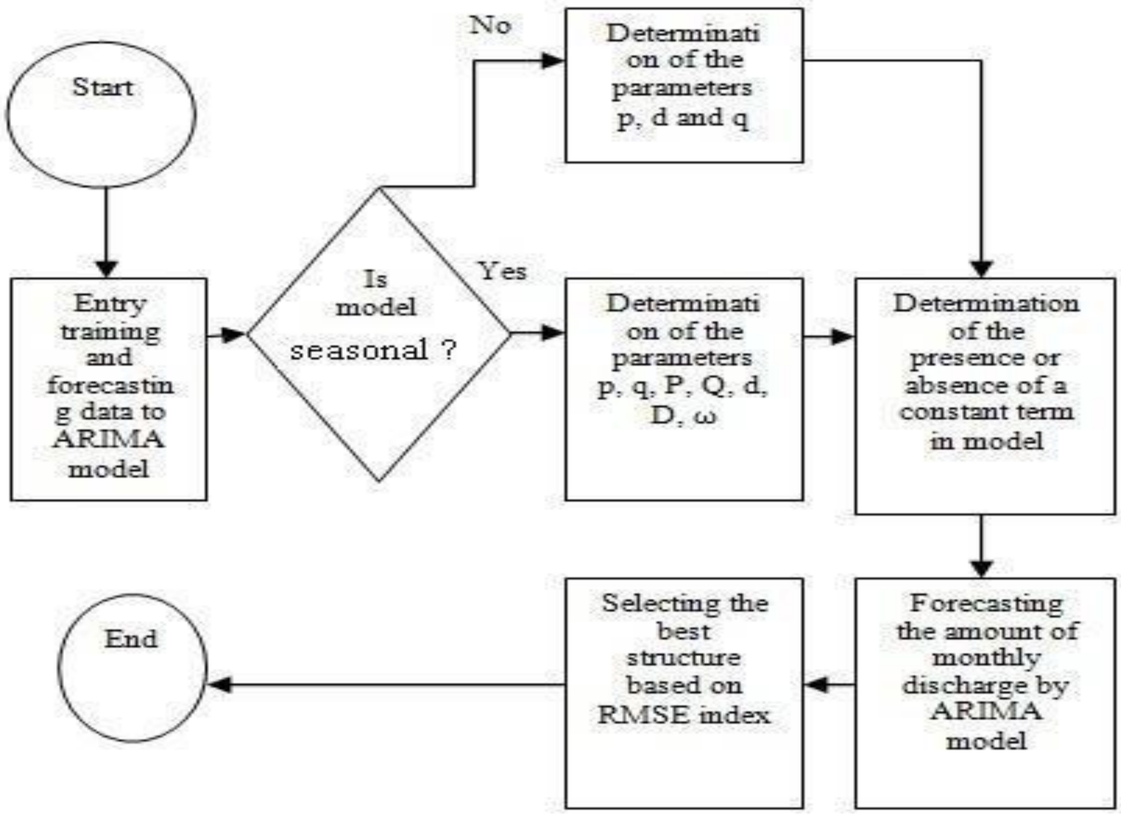
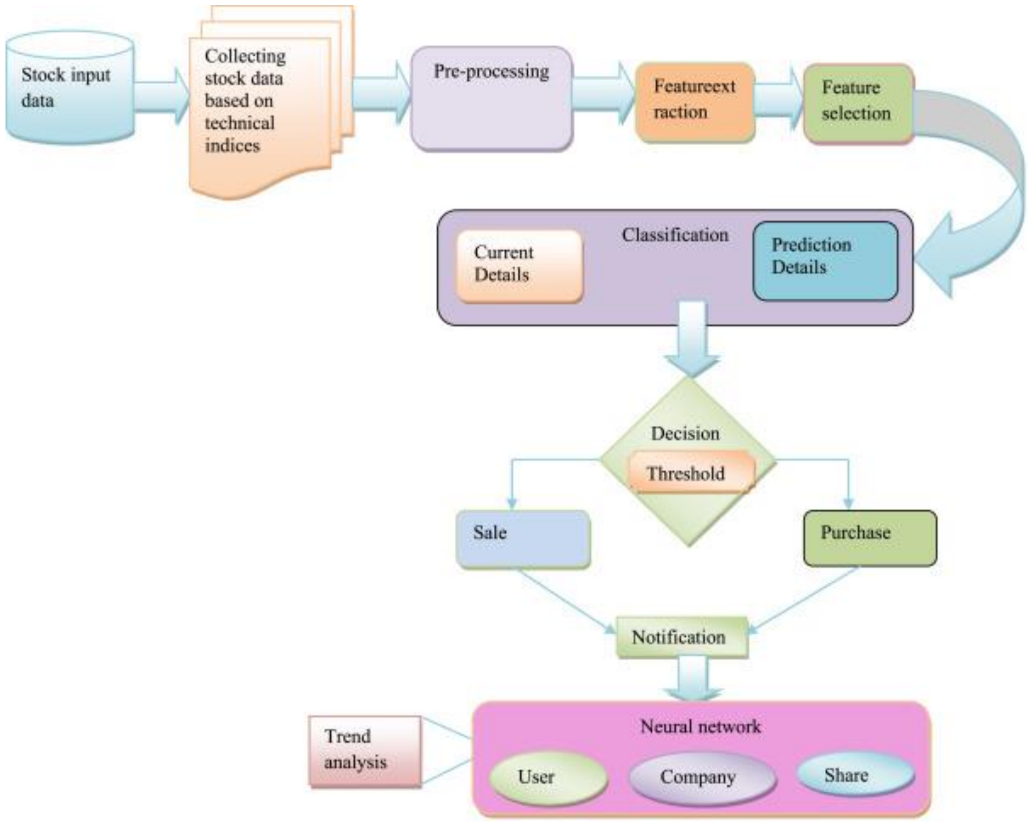


Fig [3]

UMLDiagram:



Fig[4]

Chapter-5

Modules Description

ARIMA (p, d, q) (P, D, Q) m,

- p — the number of autoregressive
- d — degree of differencing
- q — the number of moving average terms
- m — refers to the number of periods in each season
- (P, D, Q)— represents the (p,d,q) for the seasonal part of the time series

Seasonal differencing takes into account the seasons and differences the current value and it's value in the previous season eg: Difference for the month may would be value in May 2018— value in may 2017.

- In Purely seasonal AR model, ACF decays slowly while PACF cuts off to zero
- AR models are used when seasonal auto-correlation is positive
- In Purely seasonal MA model, ACF cuts off to zero and vice versa
- MA models are used when seasonal auto-correlation is negative

Final steps

- **Step 1 — Check stationarity:** If a time series has a trend or seasonality component, it must be made stationary before we can use ARIMA to forecast. .
- **Step 2 — Difference:** If the time series is not stationary, it needs to be stationarized through differencing. Take the first difference, then check for stationarity. Take as many differences as it takes. Make sure you check seasonal differencing as well.
- **Step 3 — Filter out a validation sample:** This will be used to validate how accurate our model is. Use train test validation split to achieve this
- **Step 4 — Select AR and MA terms:** Use the ACF and PACF to decide whether to include an AR term(s), MA term(s), or both.
- **Step 5 — Build the model:** Build the model and set the number of periods to forecast to N (depends on your needs).
- **Step 6 — Validate model:** Compare the predicted values to the actuals in the validation sample.

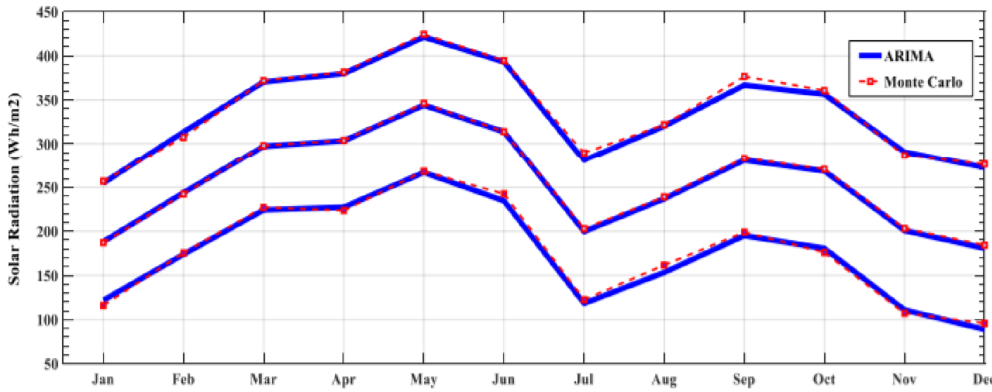


Fig.[5] Month Data

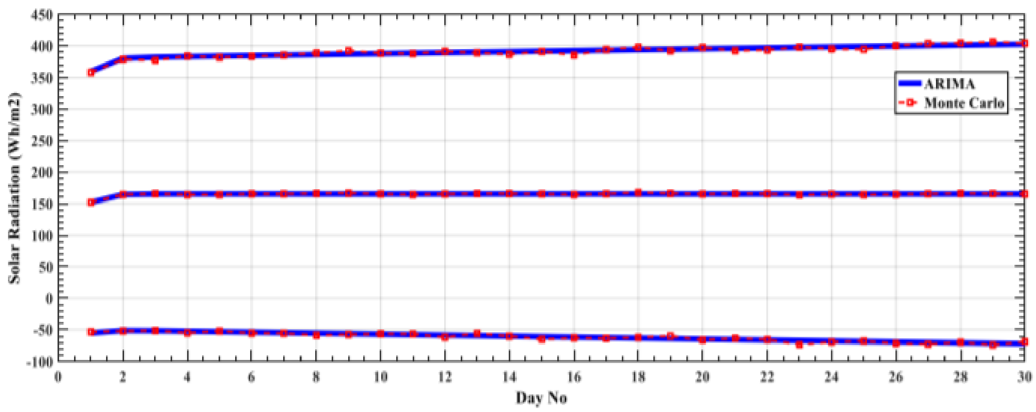


Fig.[6] Daywise Data

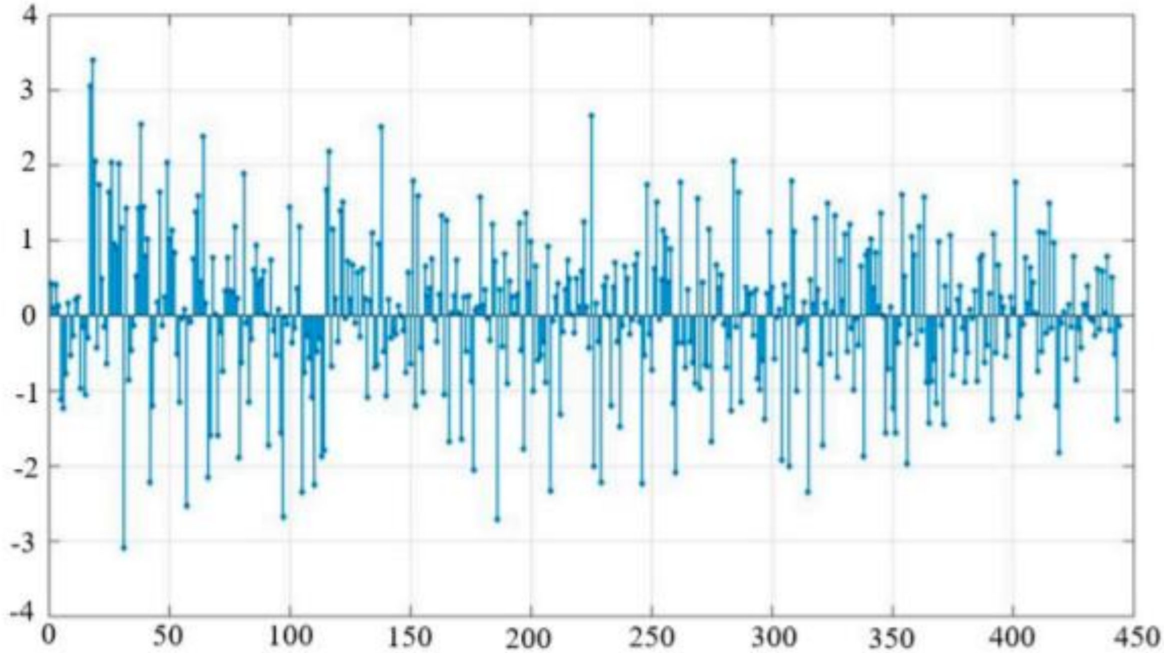


Fig.[7] No. of months

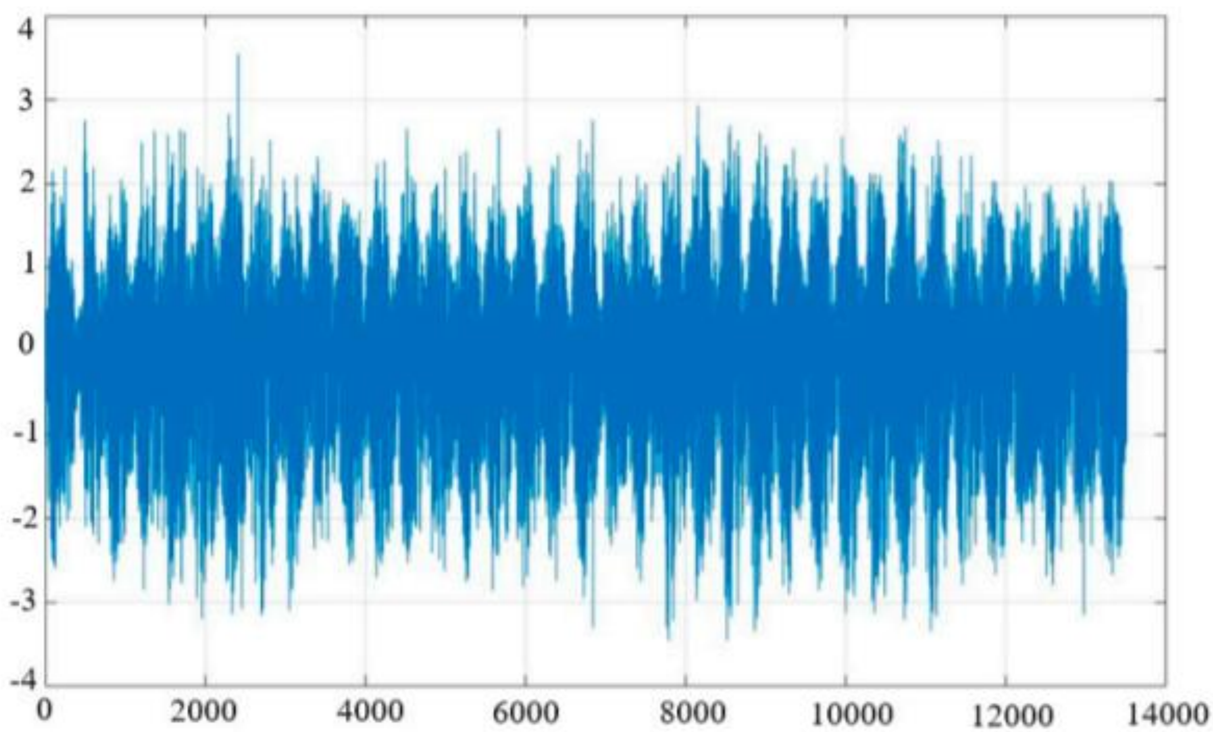


Fig.[8] No. of days

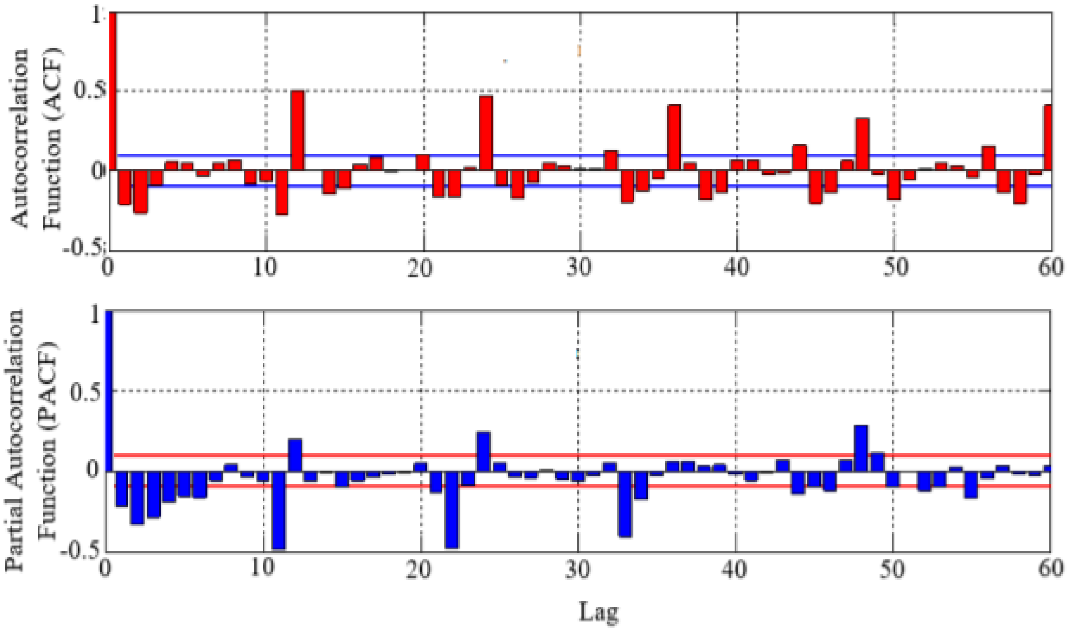


Fig.[9] ACF

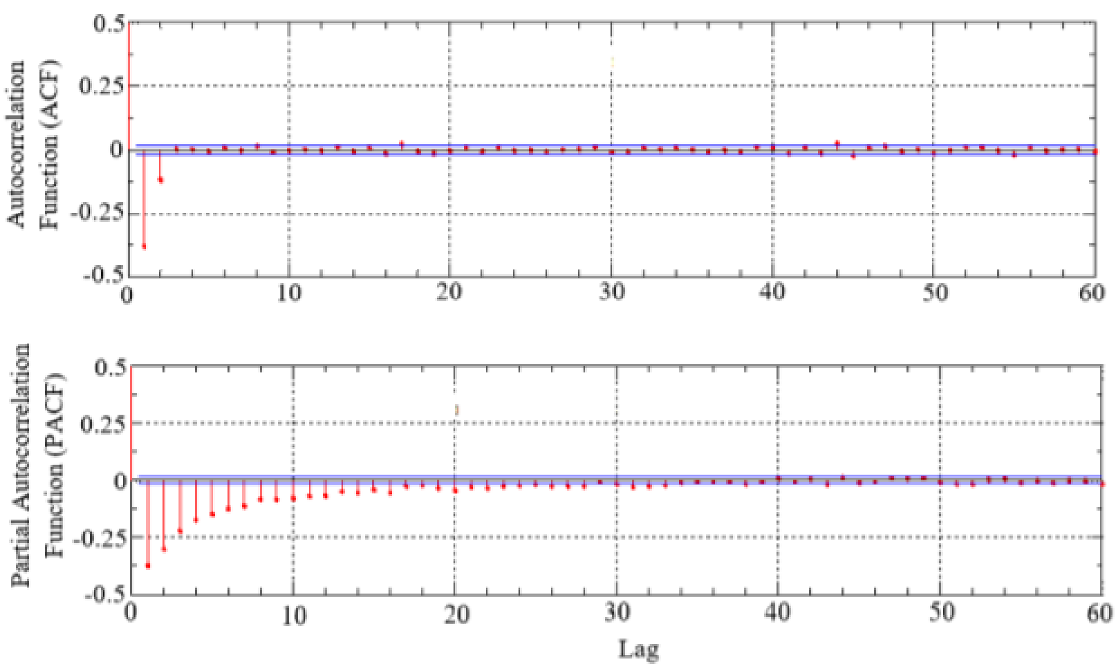


Fig.[10] PACF

<matplotlib.axes._subplots.AxesSubplot at 0x1c21ac3050>

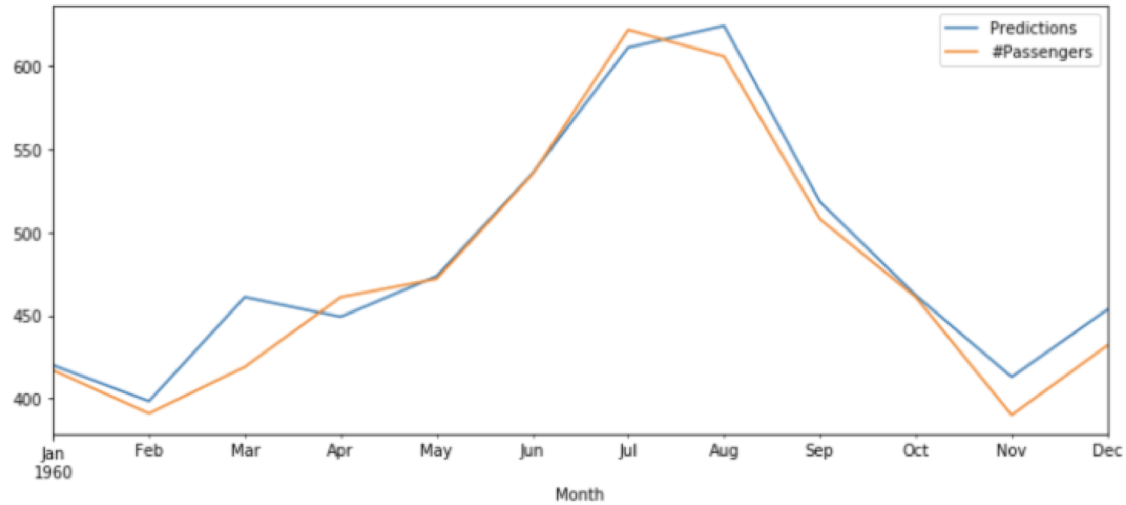


Fig.[11] Arima Pridiction

Chapter-6

Results

The parameters of that ARIMA model can be used as a predictive model for making forecasts for future values of the time series once the best-suited model is selected for time series data.

The d-value effects the prediction intervals —the prediction intervals increases in size with higher values of ‘d’. The prediction intervals will all be essentially the same when d=0 because the long-term forecast standard deviation will go to the standard deviation of the historical data.

There is a function called predict() which is used for predictions from the results of various model fitting functions. It takes an argument n.ahead() specifying how many time steps ahead to predict.

Predict (fitARIMA, n.ahead=5)

Forecast. ARIMA ()

function in the forecast R package can also be used to forecast for future values of the time series. Here we can also specify the confidence level for prediction intervals by using the level argument.

futureVal <- forecast.Arima(fitARIMA,h=10, level=c(99.5))

plot.forecast(futureVal)

We need to make sure that the forecast errors are not correlated, normally distributed with mean zero and constant variance. We can use the diagnostic measure to find out the appropriate model with best possible forecast values.

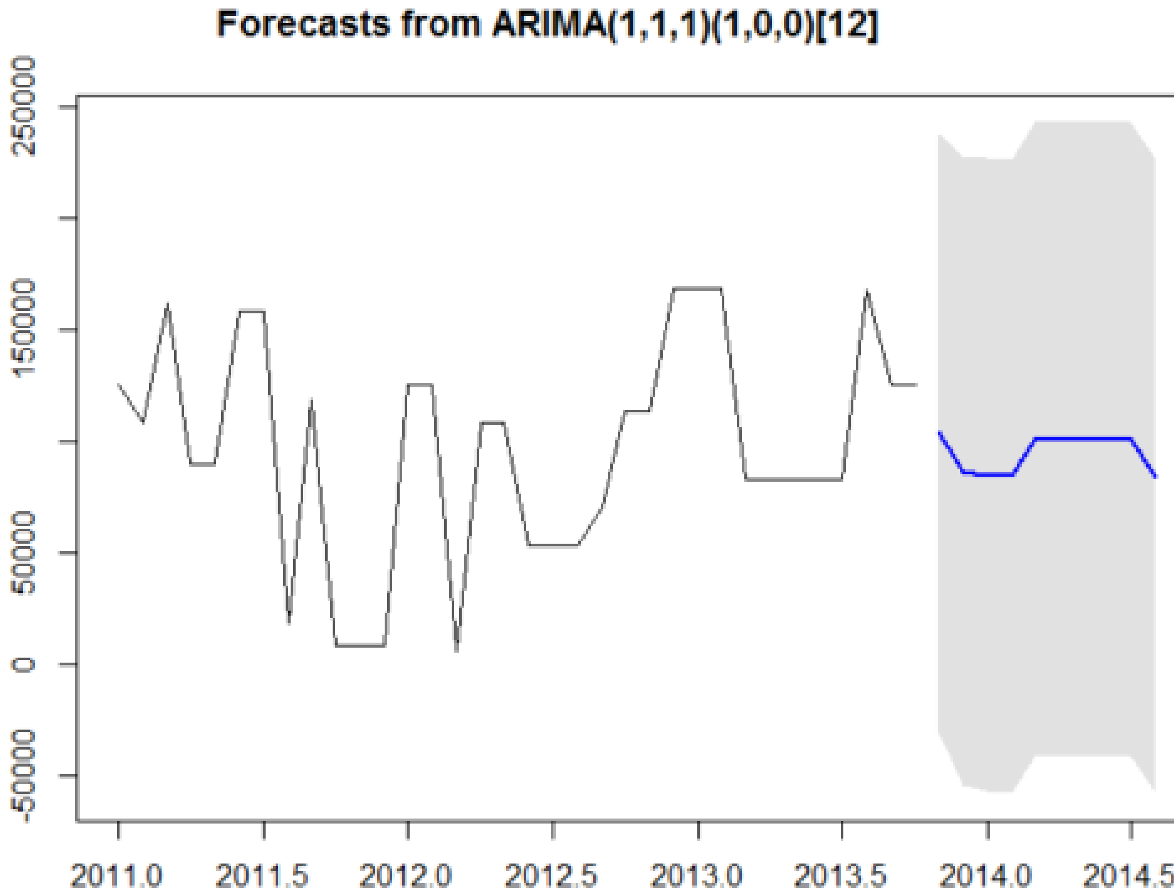


Fig.[12] forecast from Arima

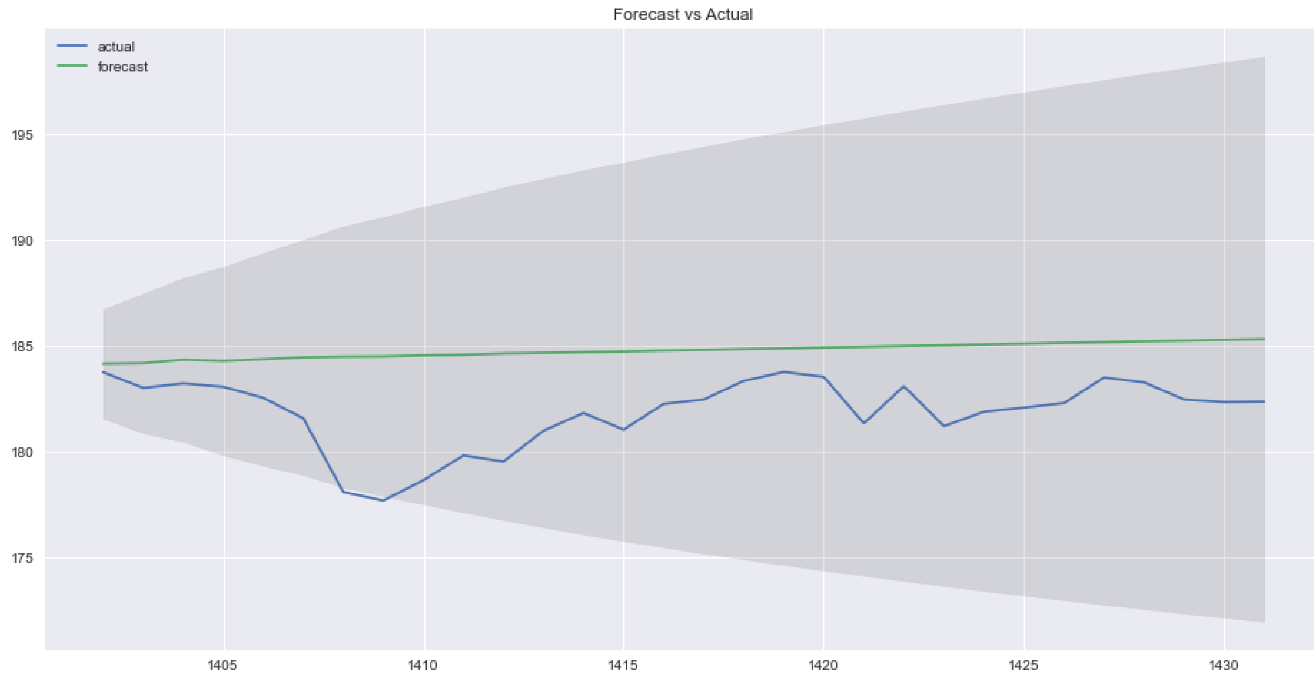


Fig.[13] Result of Arima

```
Performing stepwise search to minimize aic
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=6040.470, Time=1.15 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=6096.252, Time=0.18 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=6039.073, Time=0.34 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=6039.508, Time=0.35 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=6096.214, Time=0.12 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=6040.138, Time=0.52 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=6041.460, Time=1.24 sec
 ARIMA(1,1,0)(0,0,0)[0]             : AIC=6039.893, Time=0.16 sec

Best model:  ARIMA(1,1,0)(0,0,0)[0] intercept
Total fit time: 4.090 seconds
```

Fig.[14] Result of Auto Arima

The forecasts are shown as a blue line, with the 80% prediction intervals as a dark shaded area, and the 95% prediction intervals as a light shaded area.

This is the overall process by which we can analyze time series data and forecast values from existing series using ARIMA.

Chapter-7

Conclusion and Future Work

6.1 Conclusion:- Demand forecasting is an important function of managing supply chain. Its integration with other business functions makes it one of the most important planning processes business can deploy for future. In this context, we developed an ARIMA model to model the demand forecasting of the finished product in a food manufacturing by using Box–Jenkins time series approach. The historical demand data were used to develop several models and the adequate one was selected according to four performance criteria: SBC, AIC, standard error, and maximum likelihood. The model that we selected and which minimizes the four previous criteria is ARIMA (1, 0, 1). The results obtained proves that this model can be used for modeling and forecasting the future demand in this food manufacturing; these results will provide to managers of this manufacturing reliable guidelines in making decisions.

6.2 Future Work:- As future work, we will develop other models by using a combination of qualitative and quantitative techniques to generate reliable forecasts and increase the forecast accuracy. We will also try neural network approach to compare it with ARIMA's results in order to confirm the ANN's strength in the food company. Furthermore, we will make an ARIMA-radial basis function (RBF) combination always to achieve the same goal: high accuracy.

Chapter-8

References

[1]. Zen, Zibe and Ken “Stocks Market Prediction Using SVM”, 6 International Conference on Information Management Innovation and Industrial Engineering, 2013.

[2]. Wei Huang, Nkamori and Wang “Forecasting stock market movement direction with SVM”, Computer and Operation Research, Volume 32 Issue 10 October 2010.

[3]. N Ancona, Classification Properties of SVM for Regression

Technical Report, RIIESI/CNR-Nr.02/99.

[4]. K Kim “Financial time series forecasting using ARIMA”, Neurocomputing vol. 55 ,2011.

[5]. A V Dev and T A max, “Forecasting of stock price” int J computer algorithm, vol-2 2014.

[6]. Chen, S et al “The time series forecasting from the aspect of network” arXiv preprint arXiv vol-5 pp, 2016.

[7]. Devi, B. Uma, D. Sunder, and P. Alli “An Effective Time Series Analysis for Stock Trend Prediction Using Arima Model for nifty Midcap-50”.

[8]. Box, George and George C Tiao , “International analysis with application to economic and environmental problems” journal of the American Statistical Association 70,2016.

[9]. Yadav, A.K.; Chandel, S. Solar radiation prediction using Artificial Neural Network techniques: A review. Renew. Sustain. Energy Rev. 2014, 33, 772–781.

[10]. Benmouiza, K.; Cheknane, A. Forecasting hourly global solar radiation using hybrid k-means and nonlinear autoregressive neural network models. Energy Convers. Manag. 2013, 75, 561–569.

[11]. Beaumont, C.; Makridakis, S.; Wheelwright, S.C.; McGee, V.E. Forecasting: Methods and Applications. J. Oper. Res. Soc. 1984, 35, 79.

[12]. Koca, A.; Öztop, H.F.; Varol, Y.; Koca, G.O. Estimation of solar radiation using artificial neural networks with different input parameters for Mediterranean region of Anatolia in Turkey. Expert Syst. Appl. 2011, 38, 8756–8762.

[13]. Voyant, C.; Muselli, M.; Paoli, C.; Nivet, M.-L. Optimization of an artificial neural network dedicated to the multivariate forecasting of daily global radiation. Energy 2011, 36, 348–359.

[14]. Ramedani, Z.; Omid, M.; Keyhani, A.; Shamshirband, S.; Khoshnevisan, B. Potential of radial basis function based support vector regression for global solar radiation prediction. Renew. Sustain. Energy Rev. 2014, 39, 1005–1011.

[15]. Baghaee, H.R.; Mirsalim, M.; Gharehpetian, G.B.; Talebi, H.A. Generalized three phase robust load-flow for radial and meshed power systems with and without uncertainty in energy resources using dynamic radial basis functions neural networks. J. Clean. Prod. 2018, 174, 96–113.

[16]. Baghaee, H.R.; Mirsalim, M.; Gharehpetian, G.B.; Talebi, H.A. Three-phase AC/DC power-flow for balanced/unbalanced microgrids including wind/solar, droop-controlled and electronically-coupled distributed energy resources using radial basis function neural networks. IET Power Electron. 2017, 10, 313–328.

[17]. Baghaee, H.R.; Mirsalim, M.; Gharehpetian, G.B.; Bagahee, H.R.; Gharehpetian, G.B.B. Power Calculation using RBF Neural Networks to Improve Power Sharing of Hierarchical Control Scheme in Multi-DER Microgrids. IEEE J. Emerg. Sel. Top. Power Electron. 2016, 4, 1. [18]. Baghaee, H.R.; Mirsalim, M.; Gharehpetan, G.B.; Talebi, H.A. Nonlinear Load Sharing and Voltage Compensation of Microgrids Based on Harmonic Power-Flow Calculations Using Radial Basis Function Neural Networks. IEEE Syst. J. 2018, 12, 1–11.

[19]. Reikard, G. Predicting solar radiation at high resolutions: A comparison of time series forecasts. Sol. Energy 2009, 83, 342–349.

[20]. Bauer, P.; Thorpe, A.; Brunet, G. The quiet revolution of numerical weather prediction. Nature 2015, 525, 47–55.

[21]. Shamim, M.A.; Remesan, R.; Bray, M.; Han, D. An improved technique for global solar radiation estimation using numerical weather prediction. J. Atmos. Sol. -Terr. Phys. 2015, 129, 13–22.

[22]. Farhath, Z.A.; Arputhamary, B.; Arockiam, D.L. A Survey on ARIMA Forecasting Using Time Series Model. Int. J. Comput. Sci. Mobile Comput. 2016, 5, 104–109. Symmetry 2019, 11, 240 16 of 17.

[23]. Colak, I.; Yesilbudak, M.; Genc, N.; Bayindir, R. Multi-period Prediction of Solar Radiation Using ARMA and ARIMA Models. In Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Institute of Electrical and Electronics Engineers (IEEE), Miami, FL, USA, 9–11 December 2015; pp. 1045–1049.

[24]. Ferrari, S.; Lazzaroni, M.; Piuri, V.; Cristaldi, L.; Faifer, M. Statistical models approach for solar radiation prediction. In Proceedings of the 2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Institute of Electrical and Electronics Engineers (IEEE), Minneapolis, MN, USA, 6–9 May 2013; pp. 1734–1739.

[25]. Boualit, S.B.; Mellit, A. SARIMA-SVM hybrid model for the prediction of daily global solar radiation time series. In Proceedings of the 2016 International Renewable and Sustainable Energy Conference (IRSEC), Institute of Electrical and Electronics Engineers (IEEE), Marrakesh, Morocco, 14–17 November 2016; pp. 712–717.

[26]. Alsharif, M.H.; Kim, J.; Kim, J.H. Green and sustainable cellular base stations: An overview and future research directions. Energies 2017, 10, 587.

[27]. Kim, K.-G. Evolution of Climate Resilience and Low-Carbon Smart City Planning: A Process. In Urban Governance and Informal Settlements; Springer Nature: Berlin, Germany, 2017; pp. 1–76. 3. Lee, J.-S.; Kim, J.-W. South Korea’s urban green energy strategies: Policy framework and local responses under the green growth. Cities 2016, 54, 20–27.

[28]. Alsharif, M.H. A solar energy solution for sustainable third generation mobile networks. Energies 2017, 10, 429.

[29]. Kwon, T.-H. Is the renewable portfolio standard an effective energy policy?: Early evidence from South Korea. Util. Policy 2015, 36, 46–51.

[30]. Korean Energy Agency (KEA). Annual Report 2015. Available online: http://www.energy.or.kr/renew_ eng/resources/resources_view.aspx?no=12&page=1 (accessed on 22 January 2019).

[31]. Sindhu, S.; Nehra, V.; Luthra, S. Solar energy deployment for sustainable future of India: Hybrid SWOC-AHP analysis. Renew. Sustain. Energy Rev. 2017, 72, 1138–1151.

[32]. Park, E.; Yoo, K.; Ohm, J.Y.; Kwon, S.J. Case study: Renewable electricity generation systems on Geoje Island in South Korea. J. Renew. Sustain. Energy 2016, 8, 015904.