Data Analysis for USEN

(TITLE PAGE)

By

Sabaht Khurshid

Rana Abdul-Ahad

Supervisor

Dr. Ayesha Maqbool

Submitted to the faculty of Department of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology, in partial fulfillment for the requirements of B.E Degree in Computer Software Engineering

JULY 2020

Data Analysis for USEN

By

Sabaht Khurshid

Rana Abdul-Ahad

Supervisor

Dr. Ayesha Maqbool

Submitted to the faculty of Department of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology, in partial fulfillment for the requirements of B.E Degree in Computer Software Engineering

JULY 2020

DECLARATION

We hereby declare that no part of work introduced in this theory has been submitted on the side of another honor or capability in either this establishment or anyplace else. The thesis report has not been formerly published in any structure nor does it include any verbatim of the published resources which could be treated as violation of the international copyright decree. We also affirm that we do recognize the terms ‘plagiarism’ and ‘copyright’ and that in case of any copyright infringement or plagiarism established in thesis, we will be held fully accountable of the consequences of any such violation.

CERTIFICATE OF CORRECTIONS & APPROVAL

Confirmed that work contained in this proposition named, "Data Analysis for USEN ",did by Sabaht Khurshid and Rana Abdul Ahad under the oversight of Dr. Ayesha Maqbool for halfway satisfaction of Degree of Bachelors of Software Engineering, in Military College of Signals, National University of Sciences and Technology, Islamabad during the scholarly year 2019-2020 is right and affirmed. The material that has been utilized from different sources it has been appropriately recognized/alluded.

Approved by

Signature:

Supervisor: Dr. Ayesha Maqbool
MCS, Rawalpindi

Date:11-07-2020

Plagiarism Certificate (Turnitin Report)

This proposition has been checked for Plagiarism. Turnitin report supported by Supervisor is appended.

Signature of Student

Sabaht Khurshid

Rana Abdul Ahad

Signature of Supervisor

Acknowledgements

We are grateful to our Creator Allah Subhana-Watala to have guided us all through this work at each progression and for each new idea which Your arrangement in our psyche to improve it. We could have done nothing without Your extremely valuable assistance and direction. Whosoever helped us over the span of our postulation, regardless of whether our folks or some other individual was Your will, so without a doubt none be deserving of applause however You.

We are bountifully appreciative to our dearest guardians who raised us when we were not fit for strolling and kept on supporting us all through in each division of our lives.

We likewise want to communicate extraordinary gratitude to our supervisor Dr. Ayesha Maqbool for her assistance all through our proposal. We can securely say that we haven't took in some other data science courses and other skills required in such profundity than the ones which she has guided us. Her presence throughout the course of our project was like the base brick in the wall.

We likewise want to pay unique gratitude to Sir Afzal CEO of USEN for his enormous help and participation. Each time we stalled out in something; he concocted the arrangement. Without his assistance we wouldn't have had the option to finish our project. We value his understanding and direction all through the entire postulation.

At long last, we want to offer our thanks to all the people who have rendered significant help to our investigation.

Dedicated to our excellent guardians and revered kin whose gigantic help and collaboration drove us to this superb achievement.

Abstract

Sales predictions is a significant issue for various organizations associated with assembling, co ordinations, advertising, wholesaling and retailing. Modern brands are coordinating new advancements to record and use information to improve their proficiency. A colossal storehouse of Big information is created every day. While mechanical brands are producing exceptionally circulated information from different administrations and applications, a few difficulties in information examination require new ways to deal with help the huge information time. These difficulties for modern enormous information investigation are continuous examination and dynamic from huge heterogeneous information sources in mechanical space. The measures are compulsory to oblige process speed of exchange and to upgrade the normal development in information volume and client conduct. Hence, huge information investigation is an ebb and flow territory of innovative work.

The main objective of this project is to use approaches of data analytics, machine learning and processing of historical data for deriving valuable insights from data for USEN through predicting sales. These predictions lead to efficient decision making, enterprise planning and human behavior analysis. The repossessed data from USEN was gone under numerous data mining practices like pre-processing technique. We applied few exploratory analysis procedures on the data also. Several statistical values like p-value, range of the data were analyzed to see which one is the best possible model for the data. ARIMA model was found to be the best suited model as the data was found to be of time series nature. In order to cross validate ARIMA model its results were compared with results with another model. The model which we used for training and cross validation was LSTM, which applied the concept of deep learning and recurrent neural network. The end results were the forecasted values of the data given.

Key Words: *Big data, exploratory analysis, Autoregressive Integrated Moving Average (ARIMA), Long Short-Term Memory Networks (LSTM)*

Table of Contents

List of Figures

List of Tables

CHAPTER: INTRODUCTION

This chapter gives an overview and Introduction of the Data Analyst system for USEN. It gives a background and reason to develop the system as well as the solution we have managed to develop to the respective problem.

1.1 Overview

The system visualizes, analyzes and prepares the data. The system predicts the sales of a time series data by choosing the model that produces the more accurate sales. Model is tested through various means of testing using deep learning algorithms. The accurate model is used to forecast and the predicted sales are shown to clients to accept them.

1.2 Problem Statement

Industrial brands are integrating new technologies to record and utilize data to improve their efficiency. They are holders of big data but have no appropriate way to optimize their business.
Our aim is to play with their big data and produce the statistical models predicting various information for them. We want to refine the big data, apply certain algorithms, implementing data analysis techniques, producing statistical and prediction models. These models will help industrial brand to have a deep learning of their data and produce the desired results.

1.3 Approach

The project involves implementation of desktop application using python. Object classification and counting is done using TensorFlow. The application is integrated with MS SQL Server which provides for reliably storing the data in the database. The camera feed use

1.4 Scope

Product will help industrial brand to make more accurate, unbiased and unambigous insights. We will have a chance to work with big data and analyze its market value.

1.5 Objectives

The main objective of the system is to provide a system which performs following functions:

- Automatizing preparation, visualization and analysis of the data.

- Selecting model with correct parameter that forecast more accurate sales.
- Validate and test model with machine learning technique.
- Plotting the more accurate forecast values.

1.6 Deliverables

Table 1-1: Deliverables

Tasks	**Deliverables**
Literature Review	Literature Survey
Requirements Specification	Software Requirements Specification Document (SRS)
Detailed Design	Software Design Specification Document (SDS)
Implementation	Project demonstration
Testing	Evaluation plan and Testing plan
Training	Deployment plan
Deployment	Complete application with necessary documentation

LITERATURE REVIEW

2.1 Overview

Data is everywhere. The amount of data around us is mounting at a speedy rate and changing the way we live. An article by Forbes states, "*Data is increasing with each passing day. A time will come when it will be so huge in number that there will be 1.7 megabytes of new information created for every single human being with each passing moment".* This gives us awareness to know the essentials of the data at least.

Data Science

When we deal with unstructured and structured data, data cleansing, preparation, and analysis we are applying data science to the information.

Data Science is the blend of insights, arithmetic, programming, critical thinking, statistics, mathematics catching information in clever ways, the capacity to take a gander at things in an unexpected way, and the movement of purging, planning, and adjusting the data.

Big Data

Big Data is the colossal volumes of data that can't be prepared successfully with the existential conventional applications. The preparation of Big Data starts with the crude information that is not collected yet and is recurrently difficult to store in the memory of a computer.

Data Analytics

Data Analytics is the science of groping raw data to arrange that information. Applying an algorithmic or mechanical procedure to infer bits of knowledge and, for instance, going through a few informational indexes to search for significant connections between each other is basically work of data analytics.

Time series data

When data points are arranged in a sequence of consecutive order with dates mentioned it is called time series. The movement of selected data points, such as, sales of food items over a period being recorded at regular intervals is traced in time series information.

Time series data analysis and forecasting

The procedure for determining time series data to excerpt eloquent statistics, other characteristics, useful insights from the data is time series analysis. When we use previous observations to predict future values in a model it is basically time series forecasting.

Our data

The dataset we got from our client was a time series data with univariate qualities means we must work on a single variable of the dataset given at a time.

Autocorrelation in data

The values as a function of the time lag between them and the interconnection between them is called autocorrelation. It is the relation between the existing, previous and next row of data values.

Stationary data

If there is no change in the statistical values of a time series, then it is said to be stationary. If mean and variance are not changing and are constant, and covariance of the data does not dependent on time then time series is stationary

Seasonality in data

Periodic fluctuations or the pattern of the changings over a time period are referred to as seasonality in the data. An autocorrelation plot can be used to derive seasonality.

Modelling time series

Two models for modelling time series

1. Moving average
2. ARIMA

Moving Average

One of the ways to carry out time series modelling is moving average. The mean of all past values is cumulated as the next observation in this model. It uses good initial point. Moving average is used to speculate trends in the data. The window is defined that smooths the time series over the specified value, apply moving average and shows the different trends

ARIMA Model

ARIMA stands for auto-regressive integrated moving average. It's a way of modelling time series data for forecasting in such a way that it accounts for a pattern of growth/decline (autoregressive part), the rate of change of growth (integrated) and noise between consecutive time points (moving average).

ARIMA takes three parameters as its order p, q and d. p is defined from PACF correlation function and q from ACF correlation function.

The equation is as

$$ARIMA(p,q,d)X(P,Q,D)S$$

Its forecasting equation is:

$$\hat{Y}_t = Y_{t-12} + Y_{t-1} - Y_{t-13} - \theta_1 e_{t-1} - \Theta_1 e_{t-12} + \theta_1 \Theta_1 e_{t-13}$$

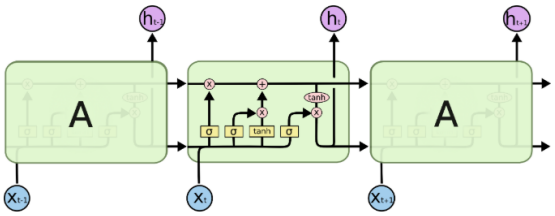
where θ_1 is the MA(1) coefficient and Θ_1 (*capital* theta-1) is the SMA(1) coefficient.

Deep learning techniques:

Recurrent neural systems are appropriate to manage learning issues where the dataset has a successive nature. They can review things from an earlier time, which is helpful for foreseeing time-subordinate targets. The regulated learning targets foreseeing valid or exact qualities from information. A preparation set of a yield (target) and some info factors is taken care of to a calculation that figures out how to foresee the objective qualities. The errand of the calculation is to convey top notch forecasts, without anyone else, separating the necessary information exclusively from the accessible information

Long Short-Term Memory

LSTM is a gated memory unit for neural systems. It has 3 doors that deal with the substance of the memory. These doors are basic calculated elements of weighted entireties, where the loads may be educated by backpropagation. The LSTM consummately fits into the neural system and its preparation procedure. It can realize what it needs to realize, recollect what it needs to recollect, and review what it needs to review, with no unique preparing or enhancement.



The repeating module in an LSTM contains four interacting layers.

2.2 Outline of Existing Works

In [4],[5] and [6] the authors have defined the terms of data science, data analytics, time series data, time series analysis and forecasting according to what the new world order.

In [7] the author has researched advanced deep learning techniques in forecasting time series data.

In [8] the author has shown how more advanced model than ARIMA can be used to predict more accurate values with a seasonal data.

In [8] the author has given the comparison of GRU and LSTM.

2.3 Comparative Review

More advanced version of ARIMA model can also b used to predict values like SARIMA. More refined predictions can be done using seasonality factor in it. It works more well with seasonality as compared to ARIMA. Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an expansion of ARIMA that unequivocally bolsters univariate time arrangement information with an occasional segment. It adds three new hyperparameters to determine the autoregression (AR), differencing (I) and moving normal (MA) for the occasional part of the arrangement, just as an extra boundary for the time of the irregularity.

The GRU is the more up to date age of Recurrent Neural systems and is truly like a LSTM. GRU has freed of the cell state and utilized the shrouded state to move data. It additionally just has two entryways, a reset door and update entryway.

CHAPTER: SOFTWARE REQUIREMENT DOCUMENT

3.1 INTRODUCTION

Software Requirements Specification (SRS), is a document describing the expected behavior of a software system. The aim of this document is to present detailed description and requirements of project Data Analysis Software for USEN which uses the machine learning, R and python analysis techniques. Our aim is to devise a software that can play with Big Data provided by the company.

Purpose

This document covers the software requirements and specifications of Data Analysis Software for USEN. The idea of the project is to analyze the Big Data and its consequences. This document describes the system development requirements and features of the data analysis software prototype, which can serve as a guide to data scientists, data analysts, as a software validation document for the prospective client and as a business value enhancer and analyzer.

Document Conventions

This section describes standards and typographical conventions followed when writing this SRS.
USEN: Universal Systems Engineering

3.2 OVERALL DESCRIPTION

Product Perspective

We are developing a software prototype in Python and R on specific machine learning models that uses statistical analysis and software engineering techniques. A bulk of data is there in industry, but companies don't have proper analysis to gain insight of that big data. We aim to analyze this data and produce certain models that help companies to optimize their sales and make most out of the big data.

Product Functions

The main features of Data Analysis Software are:

- **Data Collection:** Receiving datasets from USEN in a file of 8GB-32GB
- **Data Exploration:** Cleaning, filtering, and preparing datasets.
- **Analyzing Data:** Analyzation techniques for scrutinizing data behavior.

- **Visualizing Data:** Scree plot, box plot, scatter plot, histograms for analyzed data
- **Predicting Graphs/Models:** Designing graphs for making predictions using various prediction techniques of Machine Learning.
- **Deliverables:** Prediction graphs/models, software prototypes for implementation by USEN.

Product Features and Characteristics

Characteristics:

As we will be elucidating big data our software aims to accomplish certain characteristics:

- We will be starting with raw data and our software will lever the data programmatically for exploring it.
- We will be using a machine learning algorithm for hypothesis-free analysis. We will use the data to drive the analysis.
- As data sets are huge so we will be dealing many attributes serving us to attain more realistic predictions.
- Our software will give unambiguous, unbiased, non-hypothetical insights reciprocating the ambiguities of data predictions.

User Classes

This section describes the type of users for the Data Analysis Software:

Use Case Diagram

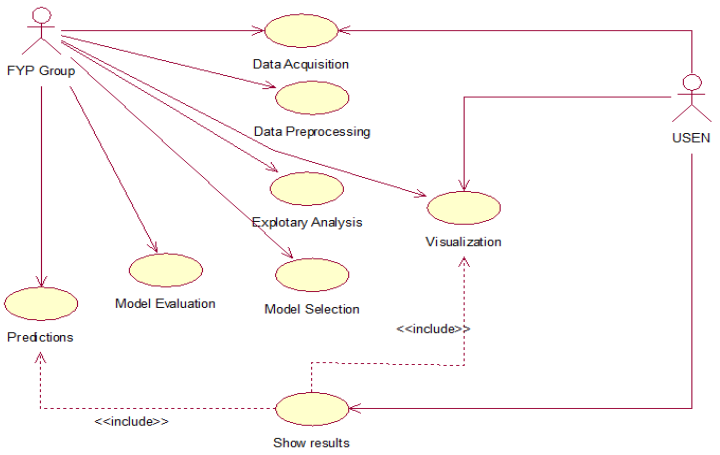


Figure 3.2.1 Use Case Diagram for the system

Activity Diagram

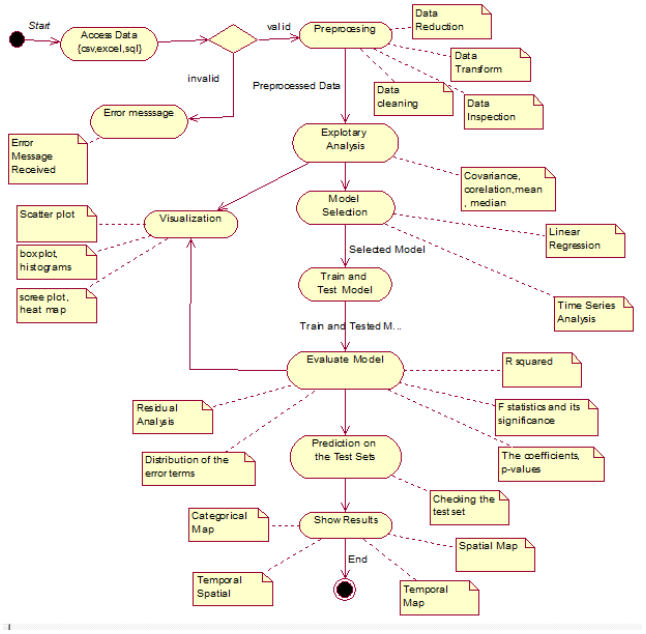


Figure 3.2.2 Activity Diagram

Description of Use Cases

Use case 1- Data Acquisition

Use case name	Data Acquisition
Primary actor	FYP Group
Secondary actor	USEN
Normal course	- Imports the data in csv, excel and valid only if successful acquisition
Alternate course	- Invalid acquisition, invalid file format, ask to import again
Pre-condition	Valid file format.
Post-condition	Successful acquisition, ready for preprocessing.

Use case 2- Data Preprocessing

Use case name	Data Preprocessing
Primary actor	FYP Group
Secondary actor	N/A
Normal course	- Cleans the data, remove null values, reduce data frame if required, transpose if required
Alternate course	- Data is clean, no null values, no reduction is required, no need of transpose
Pre-condition	Data is acquired successfully.
Post-condition	Successful preprocessing, ready for analysis.

Use case 3- Exploratory Analysis

Use case name	Exploratory Analysis
Primary actor	FYP Group
Secondary actor	N/A
Normal course	- Finds mean, median, variance, covariance, correlation, p-values separately for more clear processing - Finds all significant statistical values for selecting a good model
Alternate course	- Statistical values can be calculated collectively
Pre-condition	Data is clean. There are no null values and no noise which can create issues while calculating statistical values.
Post-condition	All important and required statistical values are calculated without any error and successfully, ready for visualization, technique selection.

Use case 4- Data Visualization

Use case name	Data Visualization
Primary actor	FYP Group
Secondary actor	USEN
Normal course	- Display scatter plot, box plot, histograms, line graph for better understanding of data and defining any outliers.
Alternate course	- Gives abnormalities in graphs when data is not clean and do not let to properly understand data.
Pre-condition	Data is acquired successfully. The data is clean and properly processed so that there are no abnormalities in data
Post-condition	Helps in understanding data hence helps in selection of model.

Use case 5- Model Selection

Use case name	Model selection
Primary actor	FYP Group
Secondary actor	USEN
Normal course	- selects models like ARIMA, ARMA, linear regression - define data is time series or some factor dependent - tell trend, seasonality, stationarity of data - remove trend, seasonality, stationarity for ARIMA and ARMA - selects the independent variable on which sales depend for linear regression
Alternate course	- model selected is not suitable for the data given
Pre-condition	Data is acquired successfully. The data is clean and properly processed so that there are no abnormalities in data. Data is visualized properly for defining the type of data.
Post-condition	Suitable model is selected ready for prediction

Use case 6- Evaluate Model

Use case name	Evaluate Model
Primary actor	FYP Group
Secondary actor	N/A
Normal course	- finds residual errors using r squared and f statistics - distribution of errors
Alternate course	- errors not calculated successfully
Pre-condition	Data is acquired successfully. The data is clean and properly processed so that there are no abnormalities in data. Data is visualized properly for defining the type of data. Suitable model for the data is selected.
Post-condition	Accurate Predictions are made.

Use case 7- Predictions

Use case name	Predictions
Primary actor	FYP Group
Normal course	- forecast the sales for some next days
Alternate course	- accurate sales are not predicted
Pre-condition	Data is acquired successfully. The data is clean and properly processed so that there are no abnormalities in data. Data is visualized properly for defining the type of data. Suitable model for the data is selected. Model is Evaluated
Post-condition	Accurate results are shown.

Use case 8- Show results

Use case name	Show results
Primary actor	FYP Group
Secondary actor	USEN
Normal course	- results of the of forecasted values are shown - display graphs of these predictions
Alternate course	- results are not shown.
Pre-condition	Data is acquired successfully. The data is clean and properly processed so that there are no abnormalities in data. Data is visualized properly for defining the type of data. Suitable model for the data is selected. Model is Evaluated. Predictions are made
Post-condition	Graphs are displayed.
Include	Visualization and model selection modules.

Operating Environment

Hardware

- The Data Analysis Software can run on a good laptop with 4GB RAM.

Software

- R Studio, R GUI
- Anaconda, Jupiter Notebook
- Excel, Microsoft SQL
- Libraries: Pandas

Design and Implementation Constraints

The software will be implemented using

- R and Python programming languages
- Data Analysis Techniques: Bayesian Statistics, Linear Regression, Classification, Clustering (K means), Principle Component Analysis
- Data Visualization Techniques: Scatter plot, histogram, heat map

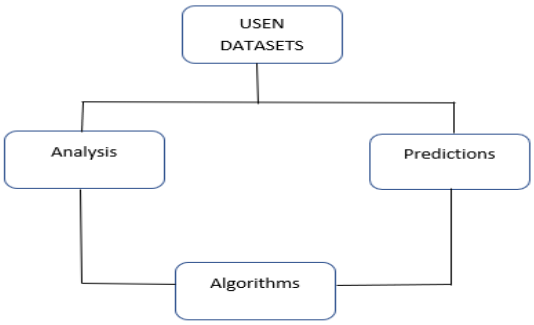


Figure 3.2.6.1 Design and Implementation

Assumptions and Dependencies

- Analysis (graphs, models, prediction) extracted from the data depends on the type of data given.
- The data shared by the company will not be used for other purposes and where required we will mask the data so that the reputation and confidentiality of the company of the company remain intact.
- The software will be accessible to only authorized persons.
- The predicted information will be confidential between company and project team.

3.3 EXTERNAL INTERFACE REQUIREMENTS

User Interfaces

The Data Analysis software will be able to:

- Access and read data whose analysis must be done (e.g. sales prediction, comparing predicted sales of system vs human behavioral analysis).
- Display visualized graphs of the analysis.

Hardware Interfaces

- This software can run on a laptop/desktop with core i5, 4GB RAM, 500 HDD.

Software Interfaces

- Database (datasets) provided by the required company.
- RStudio for analysis in R
    - R provides more specific statistical analysis, visualizations and models
    - Provides variety of statistical techniques and visualization Capabilities
    - It is extensible, open source, top notch graphical capabilities, easy to build publication quality plots.
- Anaconda, Jupiter Notebook for analysis in Python
    - It offers an ever-growing set of data management, analytical processing, and visualization libraries.
    - The Jupiter Notebooks make Python-based analysis more producible and repeatable.

3.4 SYSTEM FEATURES

The area sorts out the useful necessities for Data Analysis Software by framework includes, the significant administrations gave by the item.

Data Acquisition

Description and Priority

Acquiring Data includes anything that makes the software retrieve data including finding, accessing, acquiring, and moving data. It includes identification of an authenticated access to all related data, transportation of the data from different sources, and ways to subset and match the data to regions or times of interests. Data can be collected from either databases or will be provided either in form of text, CSV, SQL files by USEN with size from 8GB-32GB.

Priority: High

Stimulus/Response Sequences

Basic Data Flow
This function begins when the datasets are being fetched and are successfully running in the program.

Alternate Data Flow
The software notifies when the datasets are not acquired correctly.

Functional Requirements

REQ-1 The system shall be able to acquire and collect real time data or datasets provided by the organization.
REQ-2 The software shall access data using CSV, EXCEL, and SQL file.
REQ-3 The software shall be able to add, delete, manage and store the data.

Data Preparation

Description and Priority

Data readiness includes taking a gander at the information physically to comprehend its inclination, what it implies, its quality, and organization. It regularly takes a fundamental examination of information or tests of information to get this.

Priority: High

Stimulus/Response Sequences

Basic Data Flow
This function begins when the data is on hand.

Functional requirements

REQ-4 The system shall understand the nature of the data, its format, its meaning.

Data Pre-Processing

Description and Priority

This capacity incorporates cleaning information, subsetting or sifting information and making information that program can peruse and comprehend. On the off chance that there are various datasets included, this progression additionally incorporates coordination of information from

various information sources or streams. Information coordination lets you play with two datasets whose consolidated traits helps in forecast.

Priority: High

Stimulus/Response Sequences

Basic Data Flow

The software will clean and reduce the acquired data if required

Alternate Flow

The software will not be able to do any activity on data if it is not clean and filtered.

Functional Requirements

REQ-5 The software shall filter the data, remove noise, reduce if require.

REQ-6 The system shall be able to clean data, wrangle and munge data.

REQ-7 The software shall explore the data i.e. prepare, reduce integrate the data.

Exploratory Analysis

Description and Priority

This function involves selection of analytical techniques to use, building a model of the data, and analyzing results. The technique that produces most accurate results will be opted.

3.4.1.2 Stimulus/Response Sequences

3.4.1.2.1 Once data is clean, it will be analyzed based on some modern analysis techniques.

3.4.1.3 Functional Requirements

REQ-8 The software shall use numerical, categorical, and combination of both these analyses to analyze the data.

REQ-9 The software shall use non-hypothetical analysis to analyze.

REQ-10 The system shall be able to analyze the datasets and classify the results.

Visualization

Description and Priority

It includes evaluation of analytical results, presenting them in a visual way,

and creating reports that include an assessment of results with respect to success criteria. We will use scatter plot, box plot and scree plot to classify/ group data for meaningful visual display and results.

Stimulus/Response Sequences

Basic Data Flow
The visualized statistical graphs are displayed on screen.

Alternate Flow
Software notifies when it is not able to display results.

Functional Requirements

REQ-11 The software will use scatter plot, box plot, scree plot, histogram for visually displaying the results.
REQ-12 Software displays correct results and analysis on the screen.
REQ-13 The software shall classify data for meaningful results.

Predicting Graphs/Models

Description and Priority

Reporting experiences from examination and deciding activities from bits of knowledge dependent on the reason for what programming is planned (for example deals forecast, contrasting sales).

Stimulus/Response Sequences

The software shall produce accurate prediction graphs for predicting the scenarios stated by USEN.

Functional Requirements.

REQ-14 The software shall produce different statistical values of predicted model.
REQ-15 The software shall produce accurate statistical values for predicting data patterns.
REQ-16 The software shall produce reliable prediction graphs/models.

3.5 NON-FUNCTIONAL REQUIREMENTS

Performance Requirements

The software should be fast in terms of performance. To judge the software performance, we check its response time and efficiency.

Capacity

As we are using R, hence the software will be capable of handling data of 8 Gb or more.

Safety Requirements

We will make sure to avoid programming failures, software support errors and hardware failures.

Data Confidentiality

Data provided by the client must be kept confidential.

Database Backup

Proper validation of software after completion.

3.6 SOFTWARE QUALITY ATTRIBUTES

Accuracy

Our software will be able to make truthful predictions and accurate results. The prediction model that gives the more accurate value will be opted for making predictions.

Reliability

Predictions, Graphs, Models will be reliable so that industry can make their forecast of sales or anything on basis of these results. The software will run steadily with every one of the features mentioned above available and executing perfectly. The software will be tested completely, and all exceptions should be taken care of.

Computation

If extensive computational power is made available, we may deal with big datasets.

Usability

The successful software will predict and handle efficiently upcoming sales of a food item, a commercial item of a brand, compare system predicted sales vs sales from human behavioral analysis.

CHAPTER: SOFTWARE DESIGN DOCUMENT

4.1 INTRODUCTION

This chapter of report discusses the detailed design document of the system Data Analysis for USEN.

Purpose

This product software design document depicts the design, architecture and framework plan of Data Analysis for USEN. This report fills in as a guide for the engineers and as a product approval record for the customer. Archive incorporates classes and connections between them, use cases with expand depictions, succession outlines and different stream graphs.

Scope

Our aim is to play with the big data that a company holds and produce the statistical models predicting various information for them. We want to refine the big data, apply certain algorithms, implementing data analysis techniques, producing statistical and prediction models. These models will help industrial brand to have a deep learning of their data and produce the desired results.

Overview

This document is about the detailed architectural design of Data Analysis for USEN (Sales Predictor). The document is divided into various sections. Section 1 introduces the document and provides overview for executive purposes. Section 2 includes detailed description of the system with various diagrams and charts. This section includes all the architectural details of system under development. Section 3 describes all the modules and components of the system in detail. Section 4 compares this product to various other similar products available in the market. Section 5 throws light on the design decisions and tradeoffs. Section 6, pseudo code of all the components is provided.

Definitions and Acronyms

Table-4.1.1 Definitions and Acronyms

Scatter Plot, histogram, boxplot	R language
ARIMA (Auto regressive Integrated Moving Average)	Statistical Analysis
ARMA (Auto regressive moving Average)	Machine Learning (ML)

4.2 SYSTEM OVERVIEW

Overview of Modules

Following is the brief overview of all the modules for Data Analysis for USEN. Detailed description of these modules is presented in section 3.

1. **Data Acquisition Module**

   This module initiates the start of Sales Predictor i.e. it imports a data frame from an external source and passes this data to preprocessing module.

2. **Data Preprocessing Module**

   This module takes the data from data acquisition module and then the data is processed to remove null values, reduce data frame if required, log transformation if data is large and clean data.

3. **Exploratory Analysis**

   Significant statistical values are then calculated after preprocessing. These include median, mean variance etc.

4. **Visualize Data set**

   This module displays how data looks like using scatter plot, box plot. It helps to determine any outliers.

5. **Model Selection**

   This module selects model based on appearance and statistical values. If the data is time series it selects ARIMA and ARMA but if data depends on some other factor it selects linear regression.

6. **Train and Test Data**

   Converts data into train and test data.

7. **Evaluate Model Module**

   This module finds the residual errors, distribution of errors and show comparisons. If these values are ambiguous it asks to select model again.

8. **Forecasting Module**

   This module predicts the value on train data.

9. **Show Results Module**

   This module shows the predicted values on test data.

4.3 SYSTEM ARCHITECTURE

Architectural Design

This shows the collection of software components, interfaces, subsystems of Sales Predictor and how they interact with each other. These architectures define roles and responsibilities of high-level sub systems.

Top Level Architecture Diagram

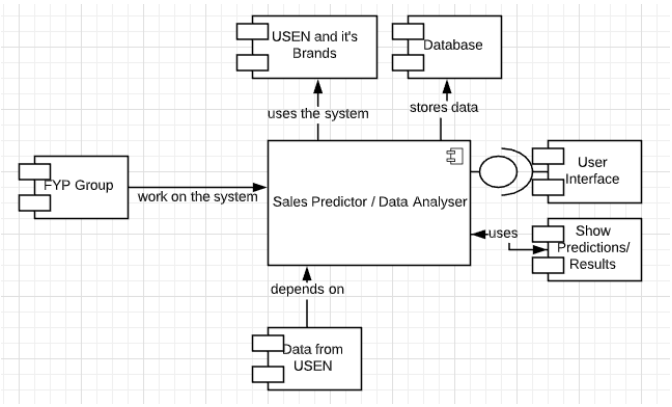


Figure 4.3.1 Top level Architecture Design

This figure shows the high-level sub systems of Sales Predictor, how they are associated with each other and depicts how components of sales predictor are wired together to form larger components or sales predictor. They explain the structure of the system.

Structure and Relationships

Sales predictor software first acquires the data through data acquisition module, clean the data, remove any abnormalities from the data using preprocessing module, calculates important statistical values of data with help of exploratory analysis module, display various graphs and on basis of these findings selects the model and perform prediction on train data using forecasting module. The predicted values are displayed after wards using show results module.

Block Diagram

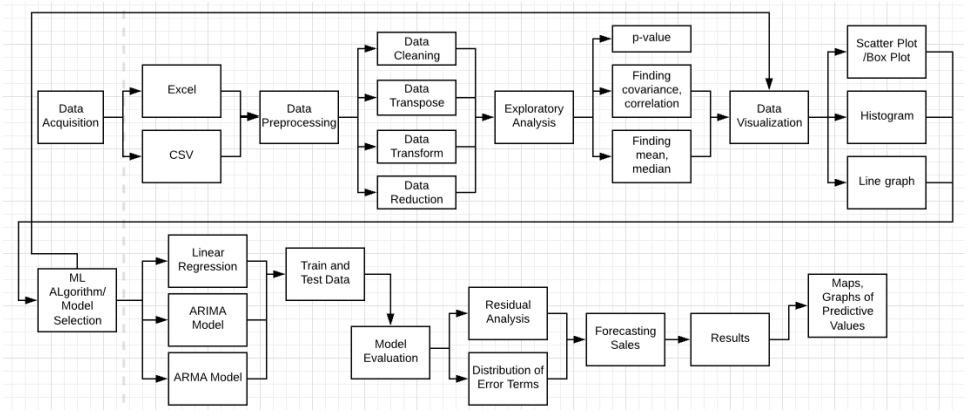


Figure 4.3.2 System Block Diagram

Data Flow Diagram Level 1

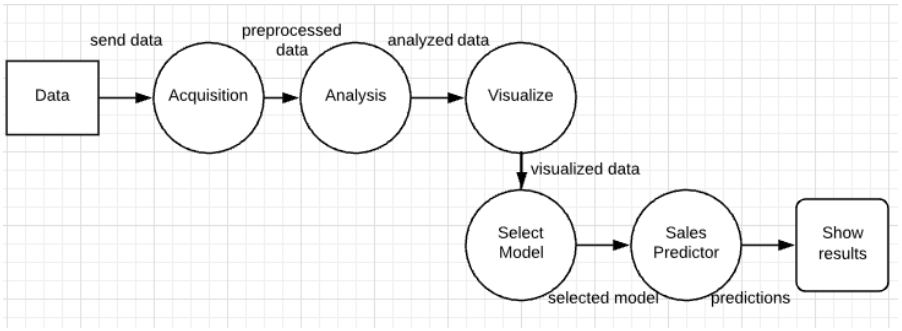


Figure 4.3.3 System Data Flow Diagram

Sequence Diagram

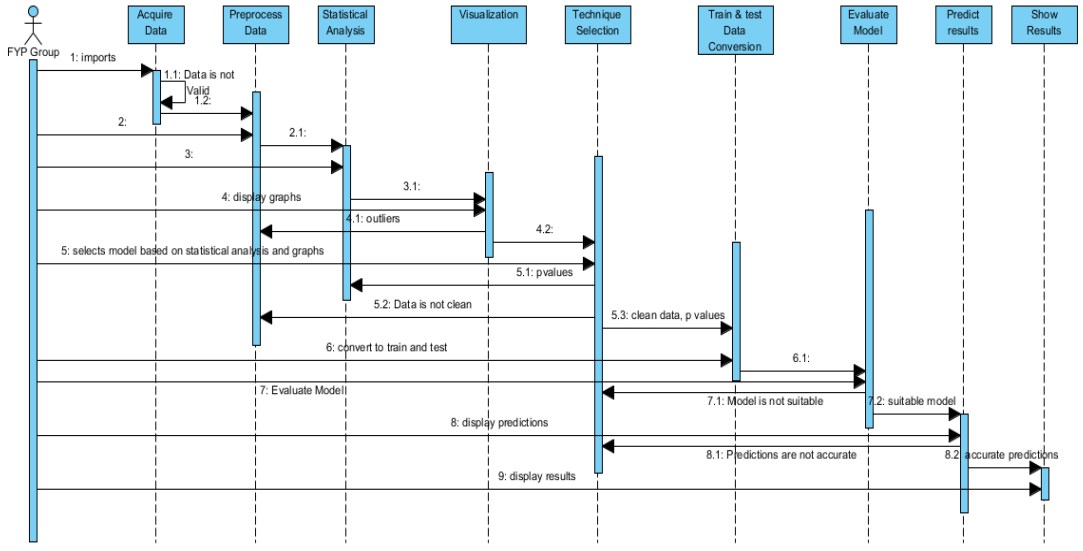


Figure 4.3.4 Sequence Diagram

State Transition Diagram

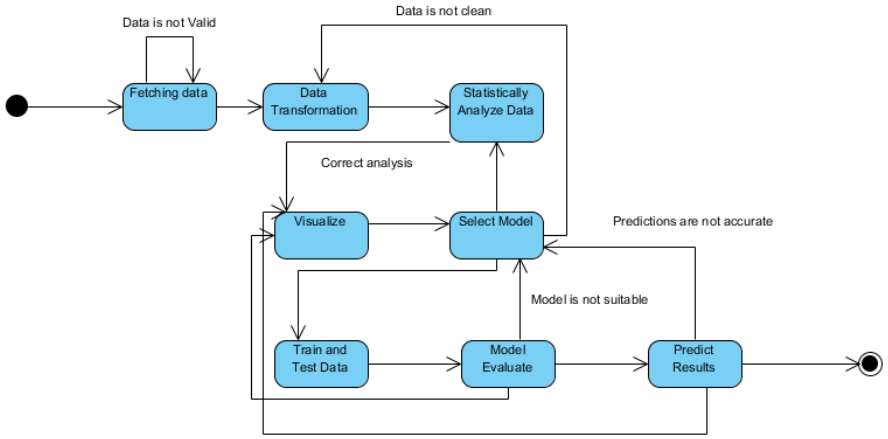


Figure 4.3.5 State Transition Diagram of Data Analysis for USEN (Sales Predictor)

Class Diagram

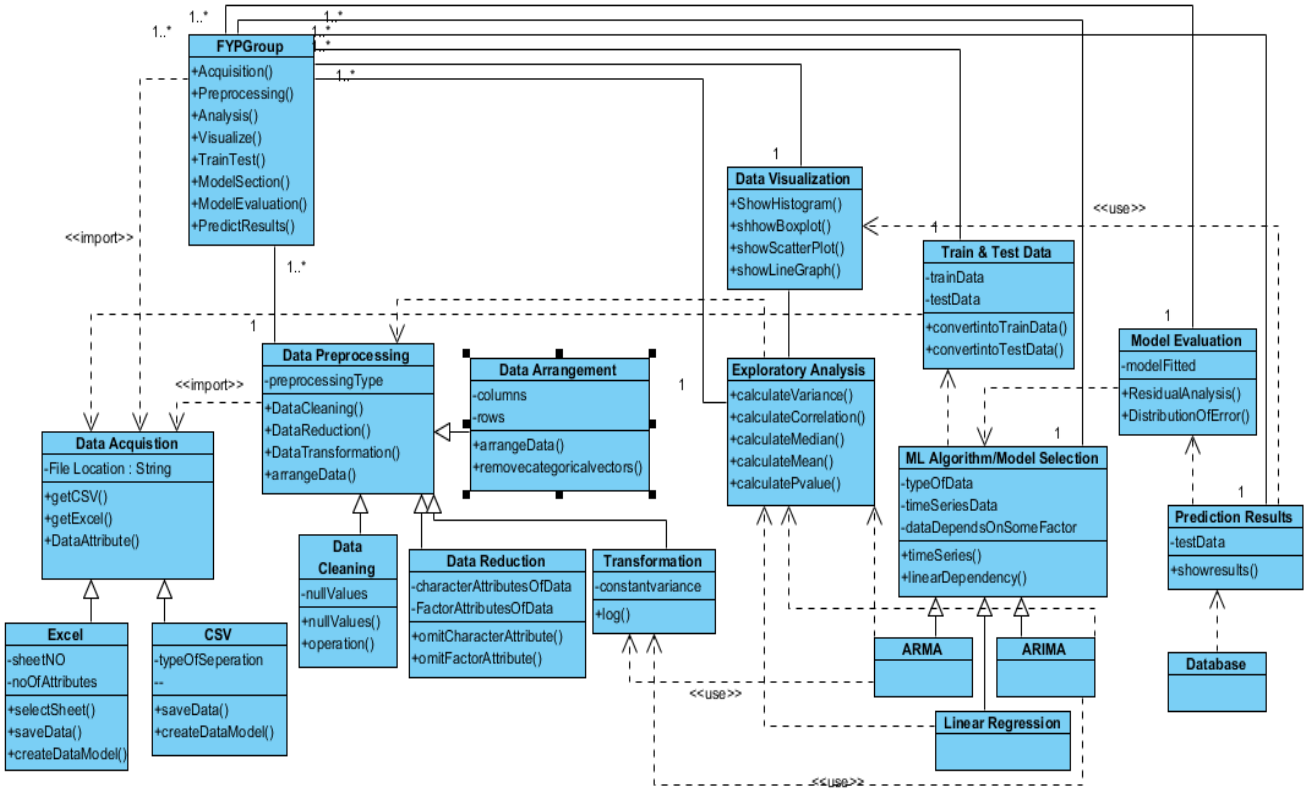


Figure 4.3.6 Class diagram

Description of Class Diagram

Class Name	Description
FYP Group	- This class does all the actual processing of the system. It gets data and perform all the other operations on data. It initiates the processing.

Data Acquisition	- This class imports the data.
Excel	- This imports the data only which has. xlsx extension.
CSV	- This imports the file with .csv extension.
Data Preprocessing	- This class prepares the data in a required format for further operations in the system. - This uses the R libraries like dplyr, tidyverse etc.
Data Cleaning	- It makes all null values in the data equal to 0.
Data Reducing	- This reduces the columns from the data that are not required further.
Data Arrangement	- This class arranges the data in a format so that we have only columns as observations.
Data Transformation	- If the data has large figures, we can convert the data into smaller values using log transformation.
Exploratory Analysis	- This class calculates all statistical values required further in the system. - This uses the R libraries like dplyr, tidyverse.
Data Visualization	- This class uses various spatial and temporal graphs for displaying data. - This use the R libraries like ggplot, ggtheme.
ML Algorithm Selection	- This gives the options of ARMA, ARIMA and linear regression models to be selected. - This uses the R libraries like tseries, tidyverse., rpart, aTSA etc.
Evaluate Model	- Finding the residual errors, distribution of errors will let you know the evaluation of model.
Predict Results	- This class gives the predictions. - This use the R libraries like forecast, tidyverse.
Database	- This class saves the prediction results.

Decomposition Description

Detailed Description of Components

Data Acquisition

Import Excel, CSV File

Identification	Name: Import Excel, CSV File Location: Data-Acquisition Module
Type	Component

Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Data Acquisition** **Requirement** The system shall be able to acquire real time data. [REQ-1,2] **Description** This feature enables the system to acquire data from different locations. This data will be fed into the system for further processing.
Function	This component acquires data in excel and csv format for further processing.
Subordinates	It has one subordinate: Preprocessing: Req. [REQ-5]
Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	N/A
Resources	**Hardware:** N/A **Software:** Data stores in some directory of computer.
Processing	This component will receive real time data which will be used for further processing.
Data	Local Disk for uploading sales data.

Data Preprocessing

Data Cleaning, Data Reduction, Data Arrangement, Data Normalization

Identification	Name: Data Cleaning, Data Reduction, Data Arrangement, Data Normalization Location: Preprocessing Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Preprocessing data** **Requirement** The system shall be able to make data clean, reduced, normalized and arranged for visualization. [REQ-5,6,7] **Description** This feature enables the system to prepare data for further processing.

Function	This component removes null values, reduce columns of data not required, log transformation etc.
Subordinates	It has one subordinate: Data Analysis: Req. [REQ-8]
Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	N/A
Resources	**Hardware:** N/A **Software:** Sales data
Processing	This component will prepare data for further processing.
Data	This component uses following information of the application: - Sales Data

Data Analysis

Statistical values

Identification	Name: Statistical Values Location: Data Analysis Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Exploratory Analysis** **Requirement** The system shall be able to find statistical values of data and analyze data statistically. [REQ-8] **Description** This feature enables the system provide non-hypothetical analysis of data.
Function	Finds mean, median, standard deviation, variance, covariance, correlation.
Subordinates	It has one subordinate: Data Visualization: [REQ-11]
Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	N/A
Resources	**Hardware:** N/A **Software:** Sales Data
Processing	This component will statically elaborate data

Data	This component uses following information of the application: - Sales Data

Data Visualization

Scatter plot, Box plot, Histogram, Line Graph

Identification	Name: Scatter plot, Box plot, Histogram, Line graph Location: Data Visualization Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Data Visualization** **Requirement** The system shall be able to show data in graphs and visual simulation. [REQ-11] **Description** This feature enables the system to display data in scatter plot, box plot, histogram and line graphs format.
Function	Visual Simulation of Data
Subordinates	It has one subordinate: Model Selection: [REQ-14]
Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	N/A
Resources	**Hardware:** N/A **Software:** Sales Data
Processing	Visually elaborates data
Data	This component uses following information of the application: - Sales Data

ML Algorithm Selection

ARIMA, ARMA, Linear Regression

Identification	Name: ARIMA, ARMA, Linear Regression Location: ML Algorithm Selection Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document:

	Model Selection **Requirement** The system shall be able to select model according to the data given. [REQ-14] **Description** This feature enables the system to selects the model and apply the selected mode
Function	Algortihm for prediction is selected
Subordinates	It has three subordinates: Train and test Data Model Evaluation: [REQ-15]
Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	N/A
Resources	**Hardware:** N/A **Software:** Sales Data
Processing	Selects suitable model for prediction
Data	This component uses following information of the application: - Sales Data, clean data, visualized and analyzed data

Evaluate Model

Residual errors

Identification	Name: Residual Errors Location: Data Evaluation Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Model Evaluation** **Requirement** The system shall be able to evaluate the model selected. [REQ-15] **Description** This feature enables the system to find errors compare them to show which model provides accurate predictions.
Function	Display and find residual errors.
Subordinates	It has one subordinate: Prediction: [REQ-16]

Dependencies	This component is independent module and runs in parallel to entire application.
Resources	**Hardware:** N/A **Software:** Sales Data
Processing	Evaluate model
Data	This component uses following information of the application: - Sales Data

Predictions

Results

Identification	Name: Results Location: Prediction Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Prediction** **Requirement** The system shall be able to predict next specified sales. [REQ-16] **Description** This feature enables the system to show accurate predictions
Function	Display and predict sales
Subordinates	N/A
Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	N/A
Resources	**Hardware:** N/A **Software:** Sales Data
Processing	Sales prediction
Data	This component uses following information of the application: - Sales Data

Activity Diagrams of Components

Following diagrams show the detailed working of each individual module of the system.

Import Module

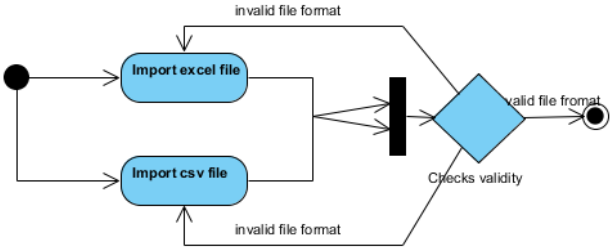


Figure 4.3.7 Activity diagram of import module

Preparation Module

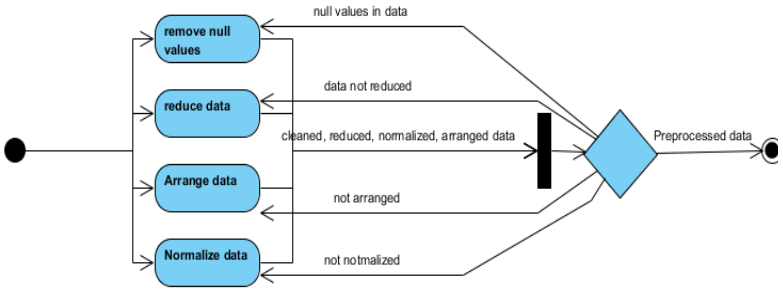


Figure 4.3.8 Activity diagram of preprocessing module

Analysis Module

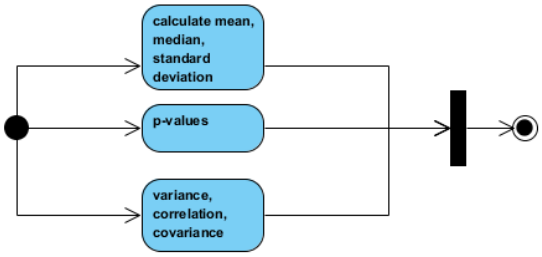


Figure 4.3.9 Activity diagram of Analysis module

Visualization Module

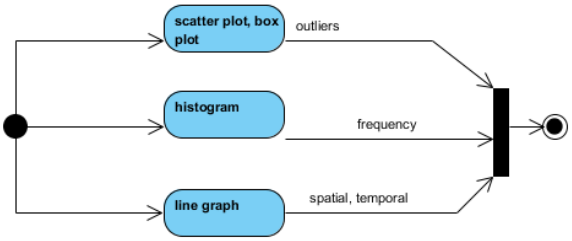


Figure 4.3.10 Activity diagram of visualization module

ML Algorithm Selection Module

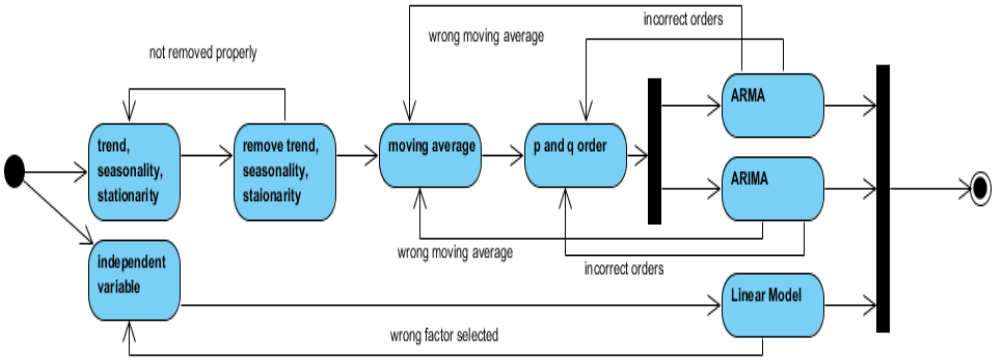


Figure 4.3.11 Activity diagram of ML selection module

Evaluation Module

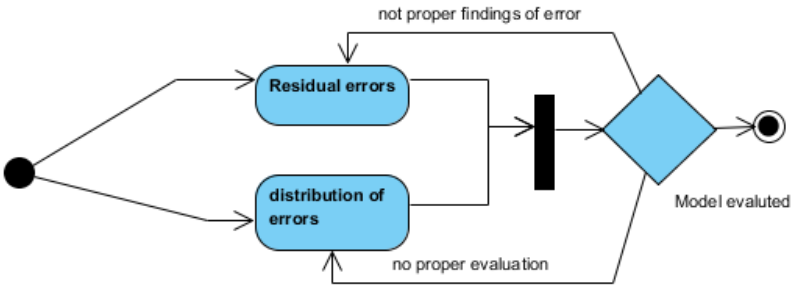


Figure 4.3.12 Activity diagram of Evaluation module

Train/Test Data Module

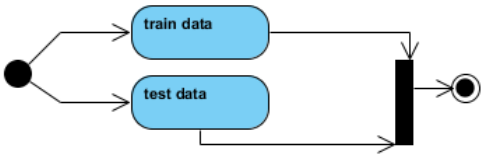


Figure 4.3.13 Activity Diagram of train/test data module

Prediction Module

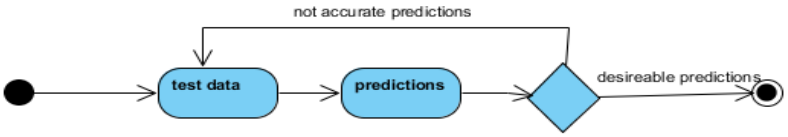
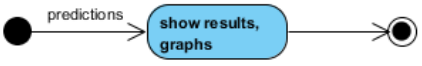


Figure 4.3.14 Activity Diagram of prediction module



Results

Figure 4.3.15 Activity diagram of results module

Data Flow Diagrams of Algorithms

Following coming sections will show detailed processing and low-level subsystems, calculations and decisions of models selected.

Data flow diagram of ARIMA level 1

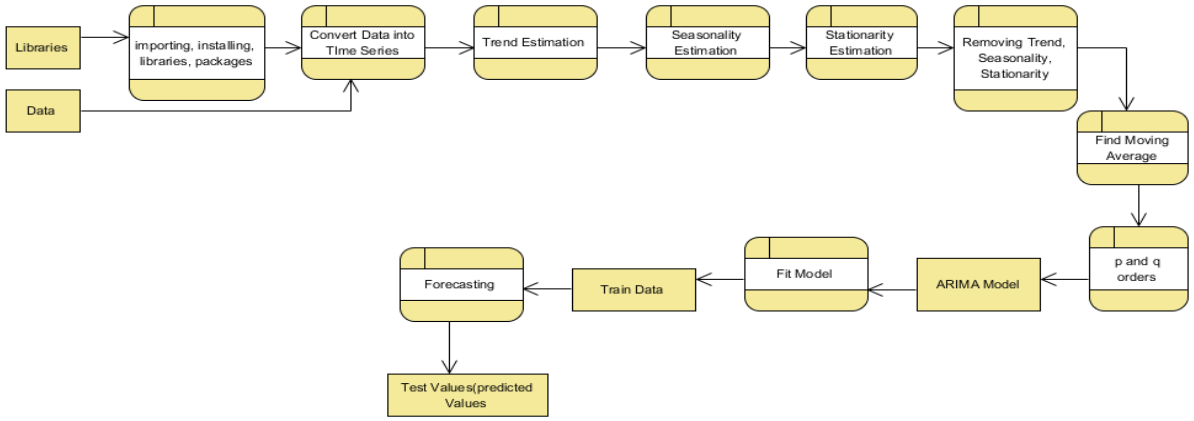
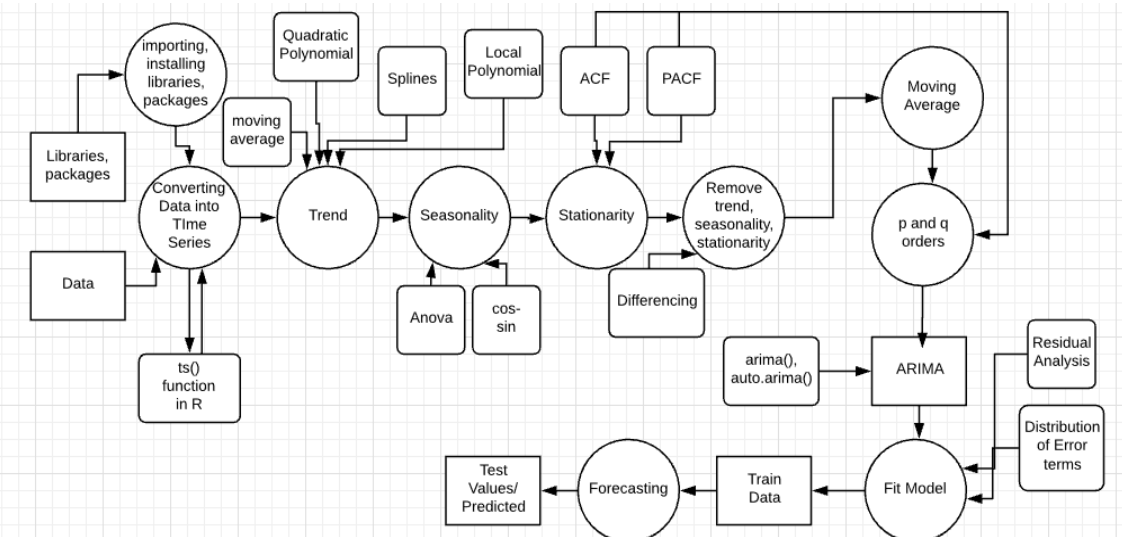


Figure 4.3.16 DFD 1.0 ARIMA

Data flow diagram of ARIMA level 2



Data flow diagram of ARMA level 1

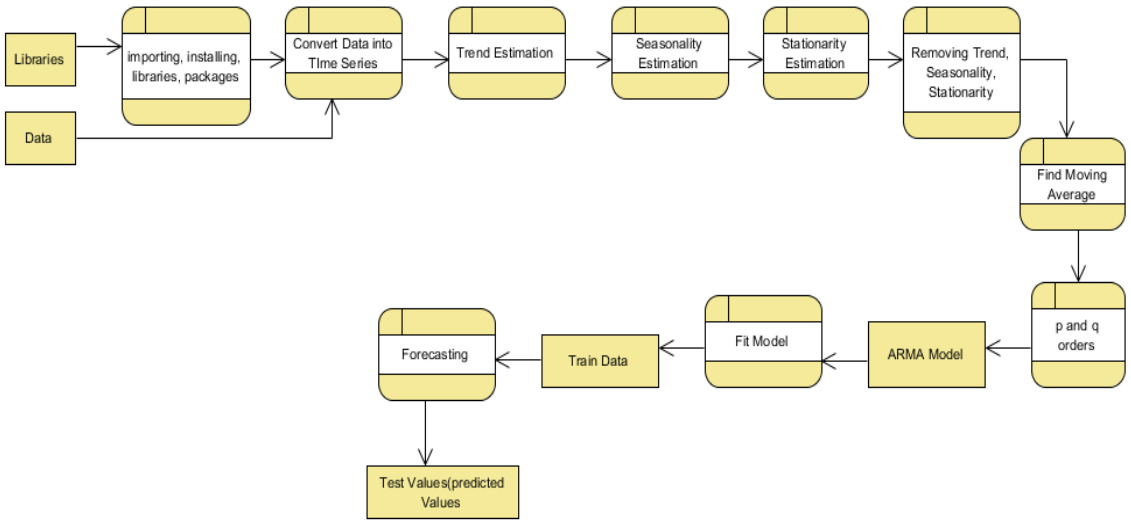


Figure 4.3.17 DFD 1.0 ARMA

Data flow diagram of ARMA level 2

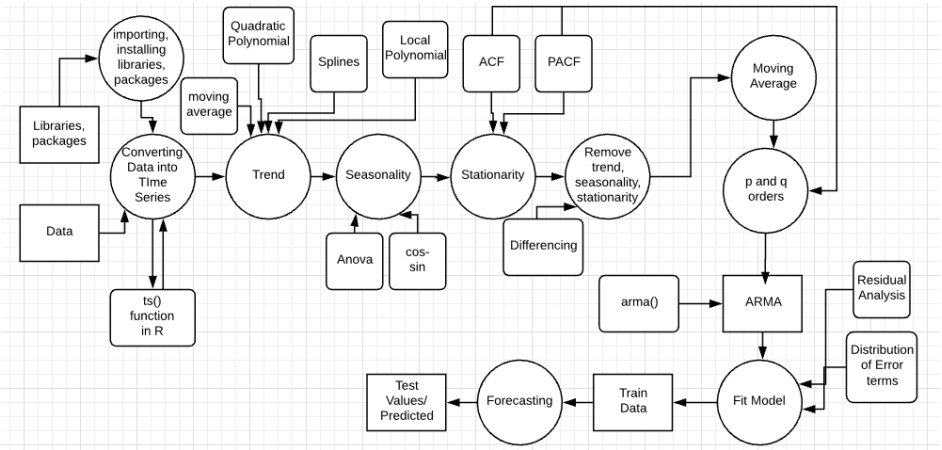


Figure 4.3.18 DFD 2.0 ARMA

Data flow diagram of Linear Regression level 1

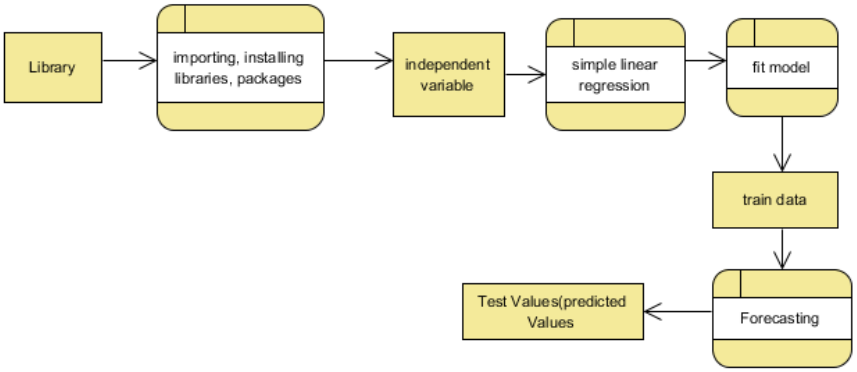


Figure 4.3.19 DFD 1.0 Linear Regression

Data flow diagram of Linear Regression level 2

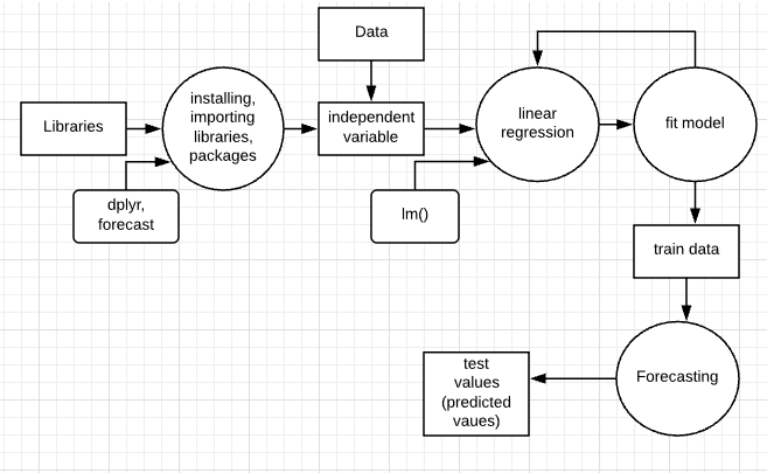


Figure 4.3.20 DFD 2.0 Linear Regression

Design Rationale

Sales predictor is component-based system that is driven by demand. Every component has been assigned with the responsibility to do a task. Sales predictor software first acquires the data through data acquisition module, clean the data, remove any abnormalities from the data using preprocessing module, calculates important statistical values of data with help of exploratory analysis module, display various graphs and on basis of these findings selects the model and perform prediction on train data using forecasting module. The predicted values are displayed after wards using show results module.

4.4 DATA DESIGN

Data Description

We got the information from USEN. The information is of FMCG (Fast Moving Consumer Goods) organization whose items are sold rapidly and at a generally minimal effort. Models incorporate non-tough family unit products, for example, bundled nourishments, refreshments, toiletries, over-the-counter medications, and different consumables. Our information is of sales of a milk item saled over a time of 9 years. The milk item is being sent in 143 distinct urban areas the nation over. Each locale has its own number of deals in 9 years. The data file is available in ".xlsx" and ".csv" format. 143 columns and 113 rows.

District Name	Tehsils	1	2	3	4	5	6	7	8	9	10	11	12
Attock	Attock	1384970	1471042	1291074	1173704	1541464	1032859	1025034	1142405	1150229	1455392	1643185	1799679
	Fateh Jang	1096237	1164365	1021916	929014.5	1220106	817532.8	811339.3	904240.8	910434.2	1151978	1300620	1424489
	Jand	784507.4	833262.1	731320.4	664836.8	873152.3	585056.3	580624.1	647107.8	651540	824397.6	930771.5	1019416
	Hassan Abdal	498318.4	529287.3	464534.1	422303.7	554625.5	371627.3	368811.9	411042.3	413857.6	523656.6	591225.2	647532.3
	Hazro	960722	1020428	895588.3	814171.2	1069278	716470.7	711042.8	792460	797887.8	1009572	1139840	1248396
	Pindi Gheb	673009.9	714835.4	627382.1	570347.4	749056.3	501905.7	498103.4	555138.1	558940.5	707230.8	798486.4	874532.7
Bahawalnagar	Bahawalnagar	1442803	1532469	1344986	1222715	1605832	1075989	1067837	1190109	1198260	1516166	1711800	1874829
	Chishtian	1223461	1299495	1140515	1036832	1361705	912411.7	905499.5	1009183	1016095	1285671	1451564	1589808
	Haroonabad	930308.5	988124.2	867236.7	788397	1035428	693789.4	688533.4	767373.1	772629.1	977612.3	1103756	1208875
	Fort Abbas	749646.3	796234.5	698822.9	635293.5	834352.1	559058.3	554823	618352.3	622587.6	787763.9	889410.9	974116.7
	Minchinabad	931777.6	989684.6	868606.2	789642	1037063	694885	689620.7	768584.9	773849.2	979156.1	1105499	1210784
Bahawalpur	Hasilpur	807130.6	857291.3	752409.9	684009	898331.8	601927.9	597367.9	665768.8	670328.8	848171.2	957612.6	1048814
	Khairpur Tamewali	464851.6	493740.6	433336.2	393942	517377.2	346669	344042.7	383436.9	386063.2	488488.1	551518.8	604044.4
	Yazman	1087033	1154589	1013336	921214.5	1209862	810668.8	804527.3	896648.8	902790.2	1142306	1289700	1412529
	Ahmadpur East	1909269	2027924	1779827	1618025	2125006	1423862	1413075	1574877	1585664	2006350	2265234	2480971
	Bahawalpur City	1206602	1281588	1124798	1022544	1342941	899838.7	893021.8	995276.2	1002093	1267955	1431562	1567901
	Bahawalpur Saddar	1017662	1080906	948667.5	862425	1132652	758934	753184.5	839427	845176.5	1069407	1207395	1322385
Bhakkar	Mankera	455067	483348	424215	385650	506487	339372	336801	375366	377937	478206	539910	591330
	Kalur Kot	615167	653397.8	573460.8	521328	684677.4	458768.6	455293.1	507425.9	510901.4	646446.7	729859.2	799369.6
	Bhakkar	1212554	1287911	1130347	1027589	1349566	904277.9	897427.3	1000186	1007037	1274210	1438624	1575636
	Darya Khan	638628.4	678317.2	595331.6	541210.5	710789.8	476265.2	472657.2	526778.2	530386.3	671101	757694.7	829856.1

Data Dictionary

Table 4.4.1 Data Dictionary

Name	Data Type	Nullable	Unique	Length
Districts	Character	True	Yes	10
Tehsils	Character	True	Yes	10
Advertisement Cost	Numeric	False	Yes	15
Attock	Numeric	False	Yes	20
Fateh Jang	Numeric	False	Yes	20
Jand	Numeric	False	Yes	20
Hassan Abdal	Numeric	False	Yes	20

4.5 COMPONENT DESIGN

Import Module

Importing, installing packages and libraries.

```
#Loading All required Libraries
library(dplyr) #For Data Manipulation
library(lubridate) #For managing Dates
library(ggplot2) #Creating Graphs

library(ggthemes) #For Look and Feel of Graphs
library(plyr)
library("ggpubr")
library("GGally")
library("Hmisc")
library("pastecs")
library("tsutils")
install.packages("psych")
```

Importing csv file

```
#read the data
sales<-read.csv('F:/FInal year Project/sales.csv')
```

OR

Importing excel file

```
sales <-  readxl::read_excel("F:/FInal year Project/SDS/salesdata.xlsx") # for aggrigation
```

Preparation Module

```
#**********Data arrange******#
AllCitySalesData= t(sales) # take transpose
```

```
#********************DATA CLEANING**********************#
# replace null values in the whole data with zero
sales[is.na(sales)]=0
```

```
#***************DaTA reduction********#
AllCitySalesData=AllCitySalesData[-c(1),]
AllCitySalesData = AllCitySalesData[-1, ]
#AllCitySalesData[1,1:ncol(AllCitySalesData)]=1;
colnames(AllCitySalesData) <- as.character(unlist(AllCitySalesData[1,]))
AllCitySalesData = AllCitySalesData[-1, ]
AllCitySalesDataDf=as.data.frame(AllCitySalesData)
View(AllCitySalesData)
```

Analysis module

```
describe(sales)
```

	vars	n	mean	sd	median	trimmed	mad	min
Shipments	1	113	57.0	32.76	57.0	57.0	41.51	1.00
Attock	2	113	988094.7	451685.73	847055.7	949241.0	455887.91	407540.19
Fateh.Jang	3	113	782100.7	357520.10	670464.9	751347.1	360846.23	322577.84
Jand	4	113	559699.8	255854.46	479809.2	537691.4	258234.76	230848.50
Hassan.Abdal	5	113	355520.8	162518.52	304774.4	341541.1	164030.48	146634.76
Hazro	6	113	685418.6	313324.03	587583.1	658466.7	316238.99	282701.28

Sales Data

```
#**********************EXPOLATORY ANALYSIS**************#
#--------------------Attock--------------------#
#----statistical values-----#
mean(sales$Attock)  #mean
summary(sales$Attock) #summary
sd(sales$Attock) #standard deviation
var(sales$Attock)#variance
describe(sales$Attock)
stat.desc(sales$Attock)
```

```
> mean(sales$Attock)  #mean
[1] 988094.7
> summary(sales$Attock) #summary
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 407540  596043  847056  988095 1323525 2058271
> sd(sales$Attock) #standard deviation
[1] 451685.7
> var(sales$Attock)#variance
[1] 2.0402e+11
> describe(sales$Attock)
   vars   n     mean       sd   median trimmed      mad      min     max   range skew kurtosis
X1    1 113 988094.7 451685.7 847055.7  949241 455887.9 407540.2 2058271 1650731 0.59    -0.89
          se
X1 42491.02
> stat.desc(sales$Attock)
     nbr.val     nbr.null       nbr.na          min          max        range          sum
1.130000e+02 0.000000e+00 0.000000e+00 4.075402e+05 2.058271e+06 1.650731e+06 1.116547e+08
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
8.470557e+05 9.880947e+05 4.249102e+04 8.419051e+04 2.040200e+11 4.516857e+05 4.571280e-01
```

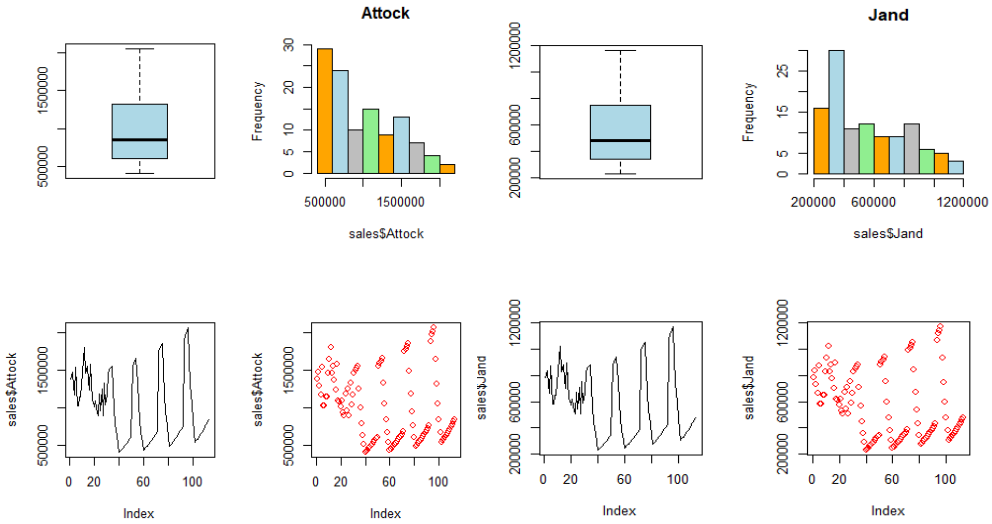
```
#--------------------sales$Fateh.Jang----------------------#
#----statistical values-----#
mean(sales$Fateh.Jang)  #mean
summary(sales$Fateh.Jang) #summary
sd(sales$Fateh.Jang) #standard deviation
var(sales$Fateh.Jang)#variance
describe(sales$Fateh.Jang)
```

```
 mean(sales$Fateh.Jang)  #mean
L] 782100.6
 summary(sales$Fateh.Jang) #summary
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
322578  471782  670465  782101 1047601 1629171
 sd(sales$Fateh.Jang) #standard deviation
L] 357520.1
 var(sales$Fateh.Jang)#variance
L] 127820622183
 describe(sales$Fateh.Jang)
  vars   n     mean       sd   median  trimmed      mad      min     max   range skew kurtosis
L    1 113 782100.7 357520.1 670464.9 751347.1 360846.2 322577.8 1629171 1306593 0.59    -0.89
          se
L 33632.66
```

```
#--------------------sales$Jand----------------------#
#----statistical values-----#
mean(sales$Jand)  #mean
summary(sales$Jand) #summary
sd(sales$Jand) #standard deviation
var(sales$Jand)#variance
describe(sales$Jand)
```

```
> mean(sales$Jand)  #mean
[1] 559699.8
> summary(sales$Jand) #summary
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 230849  337625  479809  559700  749702 1165895
> sd(sales$Jand) #standard deviation
[1] 255854.5
> var(sales$Jand)#variance
[1] 65461503804
> describe(sales$Jand)
   vars   n     mean       sd   median  trimmed      mad      min     max  range skew kurtosis
X1    1 113 559699.8 255854.5 479809.2 537691.4 258234.8 230848.5 1165894 935046 0.59    -0.89
          se
X1 24068.76
```

Visualization module



Model selection

In this section we are interested in the type of data given to us. AS FMCG products are sold in shipments and over a period so we considered the data as time series.

Most interestingly our data sales also depend on a independent variable therefore we will apply linear regression.

Time series data

When data points are arranged in a sequence of consecutive order with dates mentioned it is called time series. The movement of selected data points, such as, sales of food items over a period being recorded at regular intervals is traced in time series information.

ARIMA

```
#converting into time series
tssales<-ts(saleslog,frequency=12,start=1)
#plotting time series
ts.plot(tssales)
```

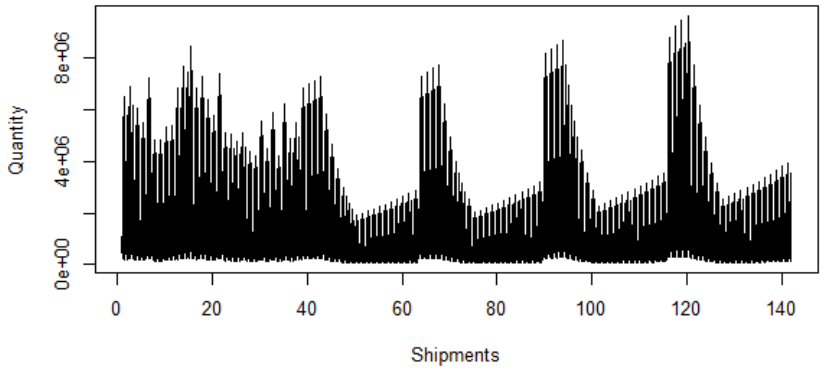
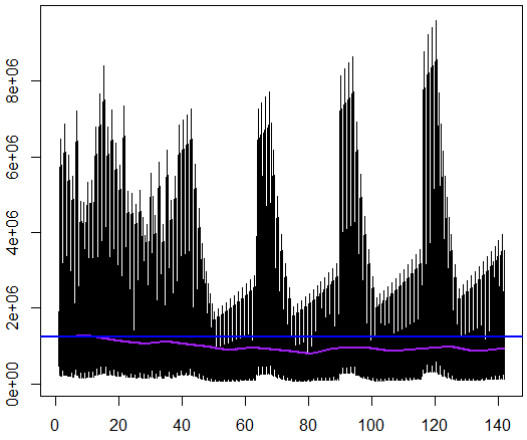


Figure Sales Data into Time Series Data

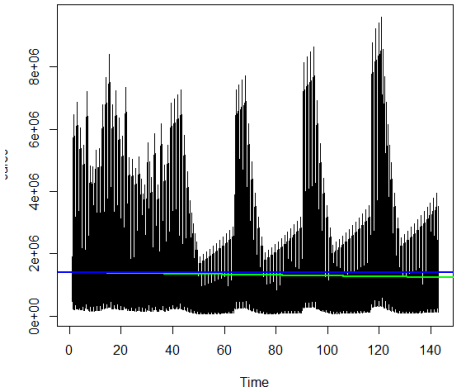
Trend Estimation

```
#**********************moving average**********************#

## Create equally spaced time points for fitting trends
#equally spaced time points because it is a monthly data in tl
timepts <- c(1:length(tssales))
timepts <- c(timepts - min(timepts))/max(timepts)
## Fit a moving average
mavfit = ksmooth(timepts, tssales, kernel = "box") ##kernel re
#case of krenel regression, constant kernel #thus it gives us
salesfitmav = ts(mavfit$y,frequency=12,start=25)
plot(salesfitmav)
## Is there a trend?
ts.plot(tssales,ylab="sales")
lines(salesfitmav,lwd=2,col="blue")#fitted trend
abline(salesfitmav[1],0,lwd=2,col="red")#constant line
```




```
## Fit a parametric quadratic polynomial
x1 = timepts
x2 = timepts^2
lmfit = lm(tssales~x1+x2)
summary(lmfit)
## Is there a trend?
salesfitlm = ts(fitted(lmfit),start=1,frequency=12)
ts.plot(tssales,ylab="sales")
lines(salesfitlm,lwd=2,col="green")
abline(salesfitlm[1],0,lwd=2,col="blue")
```

Figure Trend from Moving Average



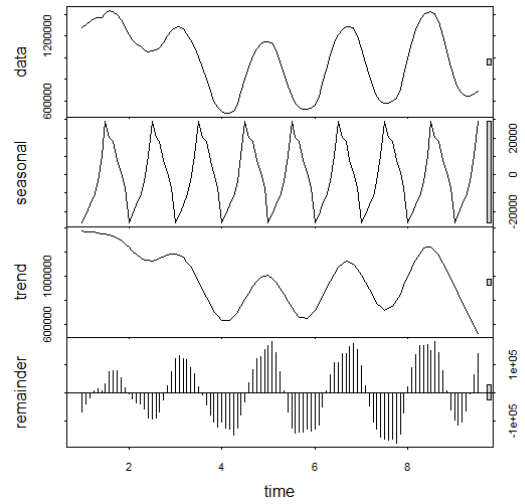

```
library(TSA)

## Estimate seasonality using ANOVA approach
month = season(tssales)#create 12 dummy variables
## Drop January (model with intercept)
model1 = lm(tssales~month)
summary(model1)
## All seasonal mean effects (model without interce
model2 = lm(tssales~month-1)
summary(model2)
```

Trend from quadratic polynomial Seasonality

Stationarity

```
##### Staionary tests
adf.test(AllCitySalesDataDf$Attock)
kpss.test(AllCitySalesDataDf$Attock)
```

```
#checking auto corelation
acf(attock_stationary,lag.max = 20)
pacf(attock_stationary,lag.max = 20)
```




```
## Differencing to Remove Trend
diff.ts.sales = diff(saleslog,1)
acf(as.vector(diff.ts.sales))
pacf(as.vector(diff.ts.sales),lag.max=12)
```

```
#remove seanolaity
timeseriesseasonallyadjusted <- attock_ts_t- f$seasonal
```

Trend, Seasonality, stationarity Remove trend, seasonality, stationarity

Moving average and p, q orders

```
CleanMovAvg10=ts(na.omit(AttockCitySalesDf$MovAvg10), frequency = 12)
decomp=stl(CleanMovAvg10,s.window="periodic")
deSeCleanMovAvg10<-seasadj(decomp)
plot(decomp)
adf.test(CleanMovAvg10) # alternate Stationary
acf(CleanMovAvg10)
pacf(CleanMovAvg10)
diff1MovAvg10= diff(log(deSeCleanMovAvg10), differences = 1)
plot(diff1MovAvg10)
adf.test(diff1MovAvg10)
auto.arima(deseasinalData,seasonal = FALSE)
fitMovAvg10auto<-auto.arima(deseasinalData,seasonal = FALSE)
```

```
arima(deseasinalData,seasonal = FALSE)
fitMovAvg10auto<-arima(deseasinalData,seasonal = FALSE)
```

ARMA

```
arima.sim(deseasinalData,seasonal = FALSE)
fitMovAvg10auto<-arima.sim(deseasinalData,seasonal = FALSE)
```

Vector dependent

In statistics, straight relapse is a direct way to deal with displaying the connection between a scalar reaction (or ward variable) and at least one logical factor (or autonomous factors). The instance of one illustrative variable is called straightforward direct relapse.

Linear Regression

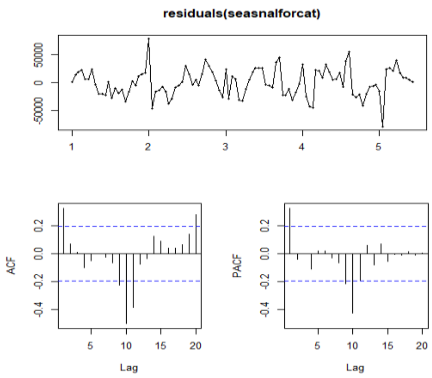
```
fit<-with(sales,lm(Attock~AdvertisementCost))
summary(fit)
prediction<-predict.lm(fit,sales)
```

Train and Test Data

```
#-----------seggregating data---------#
ind<-sample(2,nrow(sales),replace=TRUE,prob=c(0.7,0.3))
tsales<-sales[ind==1,]
vsales<-sales[ind==2,]
View(tsales)
View(vsales)
matplot(tsales,type="l")
matplot(vsales,type="l")
```

Model Evaluation

```
tsdisplay(residuals(seasnalforcat),lag.max=20,Main="seasonal Model")
```

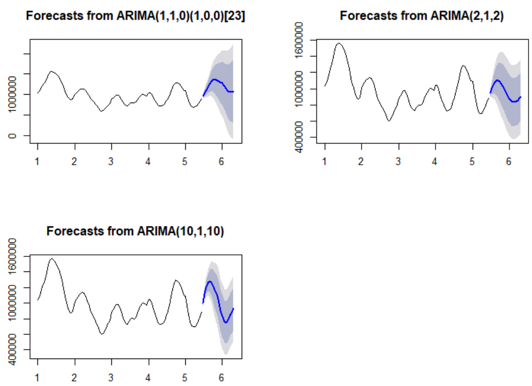


residual errors

Forecasting

```
testDataMovAvg10<- window(ts(deseasinalData),start=90) # data for testing
fiTestdataMovAvg10 <- arima(ts(deseasinalData[-c(90:133)]),order=c(1,1,3))
fcastfitTestMovAvg10 <- forecast::forecast(fiTestdataMovAvg10,h=23)
```

Results



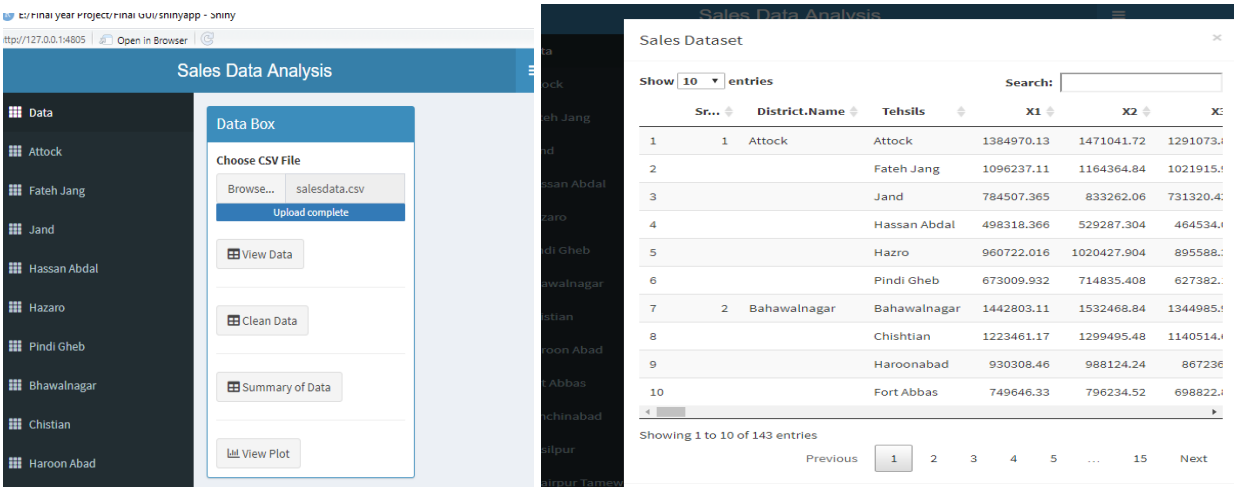
Forecasted values using different orders for ARIMA

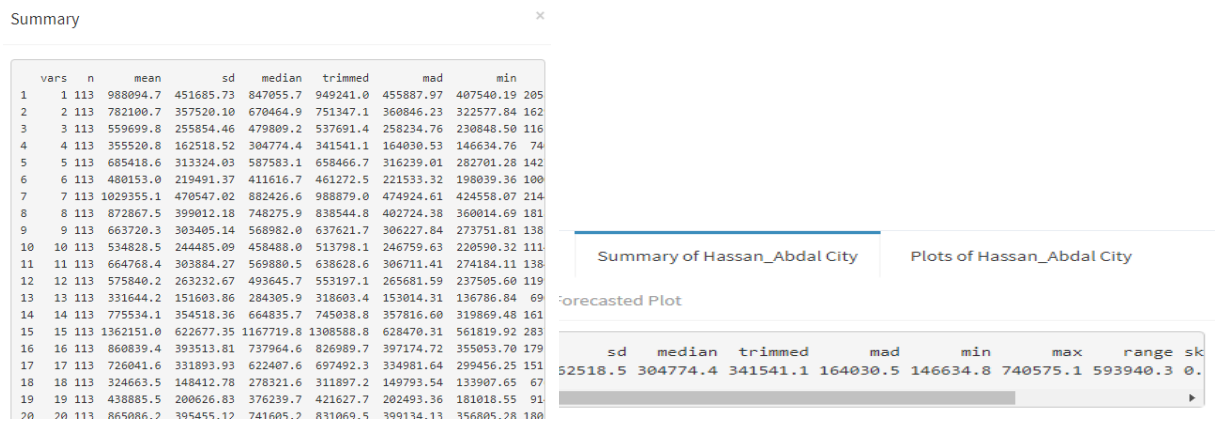
4.6 HUMAN INTERFACE DESIGN

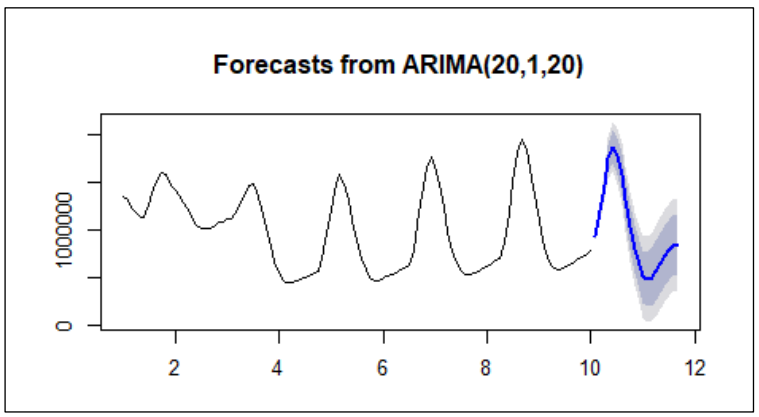
Overview of User Interface

Utilized GUI segments are menus, submenus, catches, text boxes, check boxes, down drop records, connections, and tables. The main methods for access to the whole information, by all clients, is through this UI.

Screen Images







CHAPTER: TESTING AND EVALUATION

5.1 INTRODUCTION

This test plan section portrays the fitting of the system, procedure and philosophies used to execute and oversee testing of the Data Analysis System for USEN. This test plan will guarantee that the application meets the client's necessities at an authorized level. Testing will be sought both manually and automatic. In the manual testing the tester accepts the command per the activity of an end-customer and tests the item to perceive any astounding behaviors or bug. Each module will be attempted freely and from that point onward will be composed with various modules. Unit and integration testing will be done after manual testing. For each module black box testing is done for merged modules Acceptance Testing is done.

The test scope incorporates the Testing of every single utilitarian prerequisite, application and execution and use cases necessities recorded in the necessity report.

Programming testing, dependent upon the testing methodology used, can be executed at whatever point in the improvement system. Regardless, after requirements are gathered and the coding is most of the effort is required at this point.

Test Items:

In the light of the sales prediction model and plan portrayal, various parameters and non-functional situations will be tried. The requirements defined in software requirements specification and the design described in Software Design Document will be tried.

Features tested:

Following features are tested:

- Ability to import the necessary libraries for importing dataset, processing, mathematical operations, models plotting and models forecasting.
- Ability to import the target data set.
- Ability to preprocess, visualize and analyze the dataset
- Ability to preprocess, visualize and analyze the data of each individual variable of the dataset.
- Ability to select correct moving average to smooth the data to required and realistic extent.
- Ability to select correct p values (autoregression) and q value (moving average) for ARI
- Ability to remove seasonality from the data.
- Ability to make predictions with seasonality added back

- Ability to test ARIMA model with actual values.
- Ability to test ARIMA model with L-Jung Box test.
- Ability to compare ARIMA predictions with deep learning predictions.
- Ability to grab client's approval on Forecasted values.
- Ability to run GUI properly.

Approaches:

Acceptance test: The system should receive feedback of the client. The system receives feedback on forecasted values.

Regression testing: if the system is changed in any point of the software lifecycle, it should be flexible enough to accommodate changes. Seasonality is added back after making predictions on non-seasonal data.

Unit Testing

In unit testing, singular units/parts of a product are tried. The intention is to approve that every unit of the product proceeds as planned. It is done at code level for explicit programming blunders.

White Box Testing

In white box testing, analyzer sidesteps the UI. The analyzer picks the sources of info and yields, and they are tried straightforwardly at code level and results are contrasted agreeing with necessities. In this kind of testing the code and structure of the program are known to the analyzer. The experiments created will make each condition be executed in any event once, so for this to happen we are Alternative Path Testing. As the usefulness of program is straightforward, this strategy will be anything but difficult to apply.

Black Box Testing

In discovery testing experiments are gotten from framework prerequisites and particulars. It includes going through each conceivable information/yield to check its outcomes.

Integration Testing

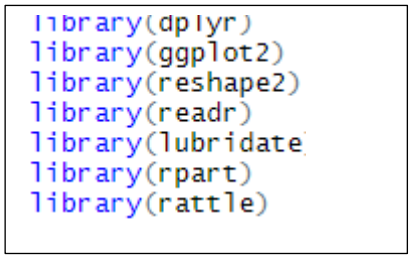
In coordination testing we test all the past modules after their combination. It is done to guarantee that the modules are practically regularly when joined.

Test Deliverables:

Test Case Name	Importing required libraries.
Test Case Number	1
Description	Testing feature to import libraries into the system.
Preconditions	The user must have R and RStudio installed on the system
Input Values	R statements for the relevant libraries i.e. dyplr, lattice, shinny etc.

Valid Inputs	Syntactically correct python statements.
Steps	Select the R statements and execute them.
Expected Output	Execution successful. Libraries Imported
Actual Output	Execution successful. Libraries Imported.

Output



Test Case Name	Importing the valid data.
Test Case Number	2
Description	Testing feature to import target dataset into the system.
Preconditions	The user must have R and RStudio installed on the system and the data must be of .csv file extension locates in specified folder
Input Values	Function with dataset name with .csv extension as argument.
Valid Inputs	Enter the valid filename and execute the function.
Steps	Write dataset name within the argument section of the read function and execute.
Expected Output	Execution successful. Data set uploaded.
Actual Output	Execution successful. Data set uploaded.

Output:

```
read.csv("E:/FInal year Project/SDS/salesdata.csv")
```

Test Case Name	Importing the invalid data.
Test Case Number	3
Description	Testing feature to import target dataset with different extension into the system.
Preconditions	The user must have R and RStudio installed on the system and the data is not of .csv file extension.
Input Values	Function with dataset name with .txt extension as argument.
Valid Inputs	Enter the invalid filename and execute the function.
Steps	Write dataset name within the argument section of the read function and execute.
Expected Output	Execution successful. Data not found.
Actual Output	Execution successful. Data not found.

Test Case Name	Preprocessing, Visualizing and Analyzing dataset
Test Case Number	4
Description	Testing feature to prepare, visualize, analyze dataset.
Testing technique	Unit testing

Preconditions	The user must have R and RStudio installed on the system and the data is already uploaded.
Input Values	R statements to prepare, plot and analyze dataset.
Valid Inputs	Enter the required R statements.
Steps	Write the statements to prepare data like making null values 0, to visualize data like ggplot() and to analyze like describe() in R.
Expected Output	Execution successful. Data is preprocessed, visualized and analyzed.
Actual Output	Execution successful. Data set uploaded. Data is preprocessed, visualized and analyzed.

Output:

Sales Processed Dataset

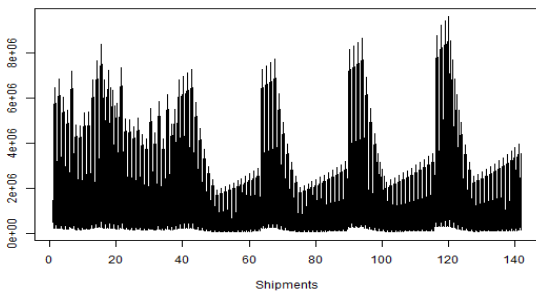
Show 10 entries Search:

	Attock	Fateh Jang	Jand	Hassan Abdal	Hazro
X1	1384970.1	1096237.1	784507.4	498318.4	960722.0
X2	1471041.7	1164364.8	833262.1	529287.3	1020427.9
X3	1291073.9	1021915.9	731320.4	464534.1	895588.3
X4	1173703.5	929014.5	664836.8	422303.7	814171.2
X5	1541463.9	1220105.7	873152.3	554625.5	1069278.2
X6	1032859.1	817532.8	585056.3	371627.3	716470.7

Summary

	vars	n	mean	sd	median	trimmed	mad
1	1	113	988094.7	451685.73	847055.7	949241.0	455887.97
2	2	113	782100.7	357520.10	670464.9	751347.1	360846.23
3	3	113	559699.8	255854.46	479809.2	537691.4	258234.76
4	4	113	355520.8	162518.52	304774.4	341541.1	164030.53
5	5	113	685418.6	313324.03	587583.1	658466.7	316239.01
6	6	113	480153.0	219491.37	411616.7	461272.5	221533.32
7	7	113	1029355.1	470547.02	882426.6	988879.0	474924.61
8	8	113	872867.5	399012.18	748275.9	838544.8	402724.38
9	9	113	663720.3	303405.14	568982.0	637621.7	306227.84
10	10	113	534828.5	244485.09	458488.0	513798.1	246759.63
11	11	113	664768.4	303884.27	569880.5	638628.6	306711.41
12	12	113	575840.2	263232.67	493645.7	553197.1	265681.59

Plot

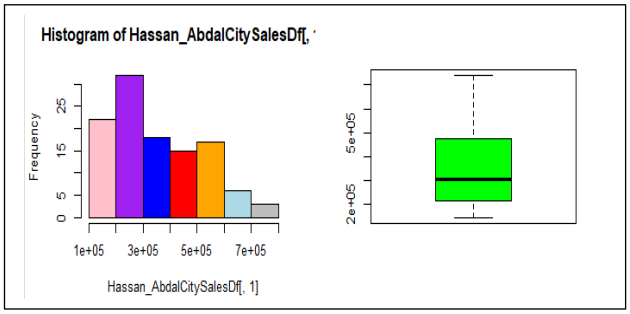


Test Case Name	Preprocessing, Visualizing and Analyzing dataset of individual variable.
Test Case Number	5
Description	Testing feature to prepare, visualize, analyze dataset of individual variable.
Testing technique	Unit testing
Preconditions	The user must have R and RStudio installed on the system and the data is already uploaded.
Input Values	R statements to prepare, plot and analyze dataset of individual variable.
Valid Inputs	Enter the required R statements.
Steps	Write the statements to prepare data like making null values 0, to visualize data like ggplot() and to analyze like describe() in R.
Expected Output	Execution successful. Data of individual variable is preprocessed, visualized and analyzed.
Actual Output	Execution successful. Data of individual variable set uploaded. Data is preprocessed, visualized and analyzed.

Output:

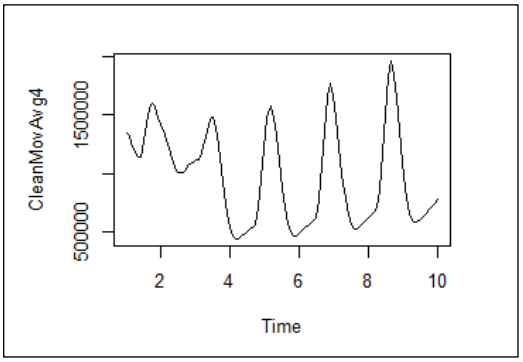
```
   vars   n     mean       sd   median  trimmed      mad      min      max
X1    1 113 355520.8 162518.5 304774.4 341541.1 164030.5 146634.8 740575.1
      range skew kurtosis       se
X1 593940.3 0.59    -0.89 15288.46
```

sales
498318.4
529287.3



Test Case Name	Selecting correct moving average
Test Case Number	6
Description	Testing feature to choose correct moving average.
Testing technique	Unit testing
Preconditions	The user must have R and RStudio installed on the system and the data is already uploaded. Data is prepared.
Input Values	R statements to select and apply moving average.
Valid Inputs	The R statement for smoothing data.
Steps	The desired moving average to achieve the smoothness of the data so that no important value of data is omitted is 4.
Expected Output	Execution successful. Data is smoothed with all important values in consideration.
Actual Output	Execution successful. Data is smoothed with all important values in consideration.

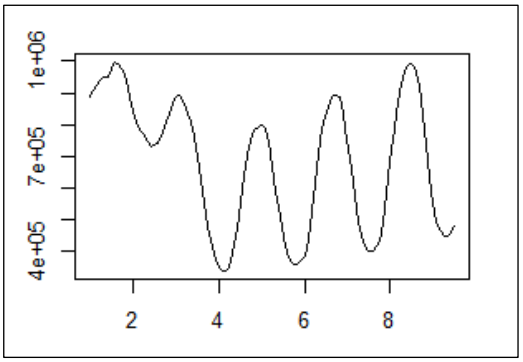
Output:



Test Case Name	Selecting incorrect moving average
Test Case Number	7
Description	Testing feature to choose incorrect moving average.
Testing technique	Unit testing
Preconditions	The user must have R and RStudio installed on the system and the data is already uploaded. Data is prepared.
Input Values	R statements to select and apply moving average.
Valid Inputs	The R statement for smoothing data.
Steps	The moving average with which data is not smoothed within the desired range and many significant values are omitted is 10
Expected Output	Execution successful. Data is smoothed with many important values not in consideration.

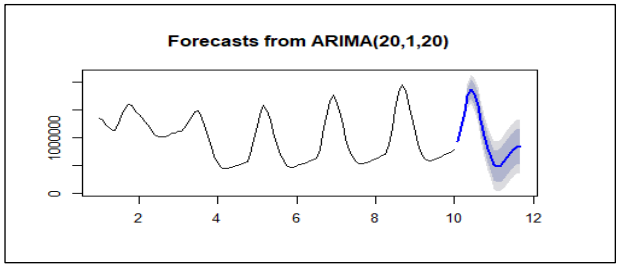
	Overly smoothed data.
Actual Output	Execution successful. Data is smoothed with many important values not in consideration. Overly smoothed data.

Output:



Test Case Name	Selecting correct p value (autoregression) and q value (moving average) for ARIMA
Test Case Number	8
Description	Testing feature to select p and q orders for ARIMA.
Testing Technique	White box Testing.
Preconditions	The data file is uploaded in the system. The data of the specific city is cleaned and smoothed.
Input Values	The data needs to be smoothed.
Valid Inputs	Cleaned and correctly smoothed data.
Steps	The value for p is 20 and for q is 20 that needs to be written in arima () function of R language.
Expected Output	Execution successful. ARIMA Model is more accurate. The more realistic forecasted values. The narrow spread with 80% and 90% of data closed to accuracy.
Actual Output	Execution successful. ARIMA Model is more accurate. The more realistic forecasted values. The narrow spread with 80% and 90% of data closed to accuracy.

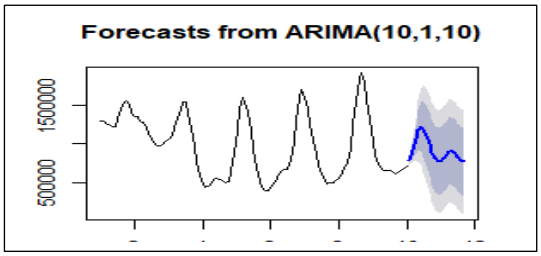
Output



Test Case Name	Selecting incorrect p value (autoregression) and q value (moving average) for ARIMA
Test Case Number	9
Description	Testing feature to see the behavior of ARIMA with incorrect p and q orders.
Testing Technique	White box Testing.
Preconditions	The data file is uploaded in the system. The data of the specific city is cleaned and smoothed.
Input Values	The data needs to be smoothed.
Valid Inputs	Cleaned and correctly smoothed data.
Steps	The value for p is 10 and for q is 10 that needs to be written in arima() function of R

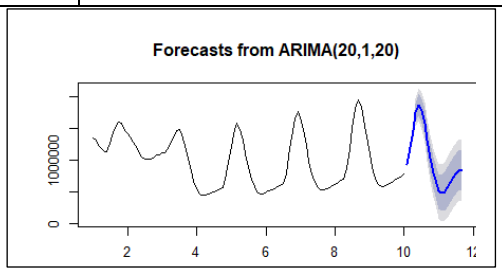
	language.
Expected Output	Execution successful. ARIMA Model is less accurate. Unrealistic forecasted values. The wider spread with 80% and 90% of data far away from accuracy.
Actual Output	Execution successful. ARIMA Model is less accurate. Unrealistic forecasted values. The wider spread with 80% and 90% of data far away from accuracy.

Output



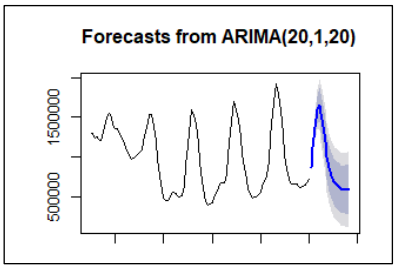
Test Case Name	Remove seasonality from the data
Test Case Number	10
Description	Testing feature to see the behavior ARIMA Model without seasonality.
Testing Technique	Regression Testing.
Preconditions	The data file is uploaded in the system. The data of the specific city is cleaned and smoothed. P and q values are correct.
Input Values	The data needs to be smoothed. ARIMA has correct p and q values.
Valid Inputs	Cleaned and correctly smoothed data. Correct p and q values
Steps	Decompose the data using R function, see if there is seasonality in data, if it is then remove the seasonality with R statements.
Expected Output	Execution successful. De-seasonal data. Less accurate ARIMA model.
Actual Output	Execution successful. De-seasonal data. Less accurate ARIMA model.

Output



Test Case Name	Predictions with seasonal data.
Test Case Number	11
Description	Testing feature to see the behavior ARIMA Model with seasonality.
Testing Technique	Regression Testing.
Preconditions	The data file is uploaded in the system. The data of the specific city is cleaned and smoothed. P and q values are correct.
Input Values	The data needs to be smoothed. ARIMA has correct p and q values.
Valid Inputs	Cleaned and correctly smoothed data. Correct p and q values
Steps	Decompose the data using R function, if seasonality is being removed add it back to the data.
Expected Output	Execution successful. Seasonal data. More accurate ARIMA model.
Actual Output	Execution successful. Seasonal data. More accurate ARIMA model.

Output



Test Case Name	Testing the ARIMA Model with actual values
Test Case Number	12
Description	Testing feature to see the accuracy of forecasted values from ARIMA
Testing Technique	Integration Testing.
Preconditions	The data file is uploaded in the system. The data of the specific city is cleaned and smoothed. P and q values are correct. Seasonality is added back.
Input Values	The data needs to be smoothed. ARIMA has correct p and q values. Data is seasonal.
Valid Inputs	Cleaned and correctly smoothed data. Correct p and q values. Seasonal data.
Steps	Use the arima function in R with all the correct parameters. Use R forecast statements to predict the future values on the train data. Compare the actual test values and train forecasted values.
Expected Output	Execution successful. Seasonal data. Forecasted values approximately near to the actual values for testing.
Actual Output	Execution successful. Seasonal data. Forecasted values approximately near to the actual values for testing.

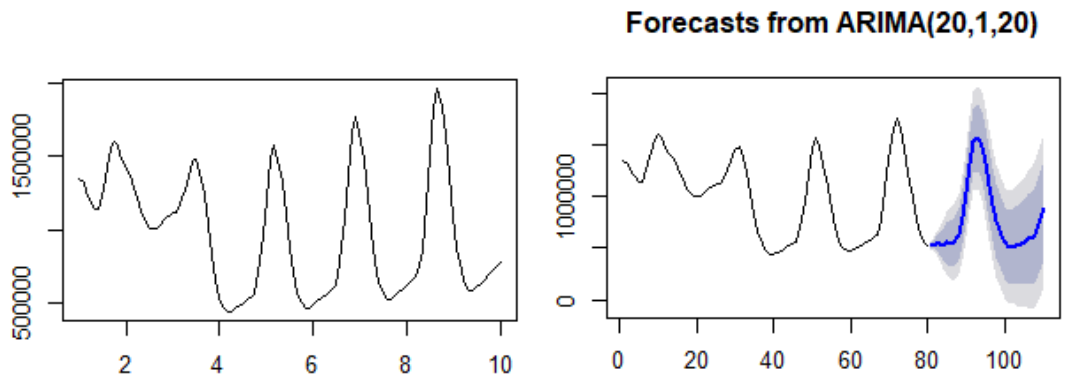
Output

	Actual Values
1	528000.0
2	550000.0
3	572916.7
4	596788.2
5	621654.4
6	647556.6
7	674538.1
8	702643.9
9	867512.3
10	1177415.7

Forecasted Values

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
81	524979.9	501537.8	548421.9	489128.286	560831.4
82	545565.8	483826.2	607305.3	451143.274	639988.3
83	541953.6	438483.6	645423.6	383709.873	700197.4
84	533078.9	379802.4	686355.4	298662.730	767495.1
85	550366.1	357822.1	742910.0	255895.547	844836.6
86	545076.7	326265.9	763887.5	210434.534	879718.9
87	550981.9	311769.2	790194.6	185137.609	916826.1
88	634342.4	379757.2	888927.6	244987.897	1023696.8
89	803541.3	535796.5	1071286.0	394060.978	1213021.6
90	1039968.4	758806.2	1321130.6	609967.912	1469968.9
91	1323207.1	1026950.8	1619463.5	870122.114	1776292.1

Actual Value



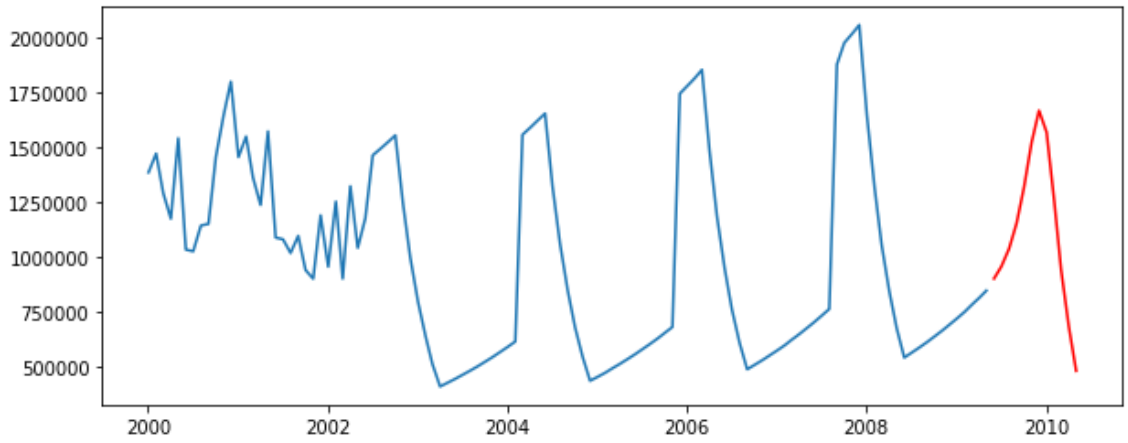
Test Case Name	Testing the ARIMA Model with L-Jung box test
Test Case Number	13
Description	Testing feature to see the accuracy of forecasted values from ARIMA and if there is any autocorrelation between existing, previous and next values.
Testing Technique	Integration Testing.
Preconditions	The data file is uploaded in the system. The data of the specific city is cleaned and smoothed. P and q values are correct. Seasonality is added back.
Input Values	The data needs to be smoothed. ARIMA has correct p and q values. Data is seasonal.
Valid Inputs	Cleaned and correctly smoothed data. Correct p and q values. Seasonal data.
Steps	Use the ARIMA function in R with all the correct parameters. Use R forecast statements to predict the future values on the train data. Apply the L-Jung box test to see the p-values. If p-value is greater than 0.05 then there is no autocorrelation between existing, previous and next values.
Expected Output	Execution successful. Seasonal data. P-values greater than 0.05.
Actual Output	Execution successful. Seasonal data. P-values greater than 0.05.

Output:

```
        Box-Ljung test

data:  fitMovAvg45$residuals
X-squared = 0.356, df = 5, p-value = 0.9965
```

Test Case Name	Applying recurrent neural network to see the accuracy of our model
Test Case Number	14
Description	Testing feature to see the accuracy of forecasted values from ARIMA and if there is any autocorrelation between existing, previous and next values.
Testing Technique	Integration Testing.
Preconditions	The data file is uploaded in the system.
Input Values	Data file is uploaded. Libraries for running LSTM recurrent network are imported and installed in python.
Valid Inputs	Data files. Required libraries installed and imported.
Steps	Use various deep learning statements to predict the next values. Using lstm() function in python to make predictions.
Expected Output	Execution successful. Plotting the predicted values. These values are similar to forecasted values from ARIMA.
Actual Output	Execution successful. Seasonal data. Plotting the predicted values. These values are similar to forecasted values from ARIMA.

Output:



Test Case Name	Clients approval for forecasted values.
Test Case Number	15
Description	Testing feature to see whether client agrees on the forecasted values or not.
Testing Technique	Acceptance Testing.
Preconditions	The data file is uploaded in the system. The system is running orderly.
Input Values	The graph of the forecasted values is plotted.
Valid Inputs	Plotted forecasted values.
Steps	Run the commands and model you have chosen to show the forecast to the client.
Expected Output	Execution successful. The client appreciates and accepts the forecast.
Actual Output	Execution successful. The client appreciates and accepts the forecast.

Test Case Name	GUI.
Test Case Number	16
Description	Testing feature for user interface
Testing Technique	Black Box Testing.
Preconditions	The system is correctly working.
Input Values	The program is initiated.
Valid Inputs	Program is initialized with correct parameters.
Steps	Create and display the main screen.
Expected Output	Execution successful. System runs perfectly and everything is displayed orderly
Actual Output	Execution successful. System runs perfectly and everything is displayed orderly

5.2 RISK AND CONTEGENCIES

Efforts have been made to remove all failures but there are certain unpredictable factors such as incorrect input data, model selection is quite difficult and may sometimes to lead to errors if parameters are not chosen correctly. Also, the selection of model varies data to data. To cover all these issues error handling was done but there may still be unforeseeable circumstances that may happen.

Schedule Risk

In order to complete the project on time, as the project might get behind schedule, we increased the no.-of- hours/day to work on the project.

5.3 FUTURE WORK

There is no boundary to the innovations and the data that is increasing exponentially. There is always and remain the need to analyze data and make useful business insights to have accurate predictions. This project can be used as a basis to understand and add features to make it into an even bigger and complex system. It can be used to play with more It could also be commercialized. In future accuracy could be improved further and scope could be extended to make it more comprehensive.
Following additional things can be done in future.

1. A more complex and large datasets can be used.
2. Data sets with multiple variables can be used.
3. Multiple variables dependency and their forecasting can be done.
4. The model changes with the type of dataset, hence more research on models in R, machine learning techniques and deep learning can be done.
5. Interface of the system can be made more efficient.
6. More intelligent business insights can be predicted.

5.4 CONCLUSION

Overview

In conclusion, this project provides sales forecast. Forecasted values are 75% accurate. This helps to make very useful and important sales production, business and sales marketing decisions.

Objectives Achieved

- Automatizing preparation, visualization and analysis of the data.
- Correct model chosen.
- Model cross validated and tested.
- More accurate forecasted values

APPENDIX A

Proposal for Data Analysis For USEN

Brief Description of the Project/Thesis with Salient Space:

Industrial brands are integrating new technologies to record and utilize data to improve their efficiency. They are holders of big data but have no appropriate way to optimize their business. Our aim is to play with their big data and produce the statistical models predicting various information for them. We want to refine the big data, apply certain algorithms, implementing data analysis techniques, producing statistical and prediction models. These models will help industrial brand to have a deep learning of their data and produce the desired results.

Scope of Work:

- Product will help industrial brand to make more accurate, unbiased and unambigous insights
- We will have a chance to work with big data and analyze its market value.

Academic Objective:

- Understanding and working with BigData, deep learning , data science and data analysis.
- To go through the process of professional project development.

Application /End Goal Objective:

- Brief statistical models of the data gathered imitating the information required by the client (industry) to predict the sales of their any food item within a given period.

Previous Work Done on The Subject:

- Big Data is changing the way the industries work. The industrial brand has the raw data only but no apposite analysis to produce the desire results neither any statistical model to make their understanding easy.

Material Resources Required:

- Data from the industry, A good working laptop

No of Students Required: 2

Special Skills Required:

- Data analysis techniques
- Data Science Education
- Deep learning and Handling BigData
- Statistiscal information
- Pyhton based programming
- R programming

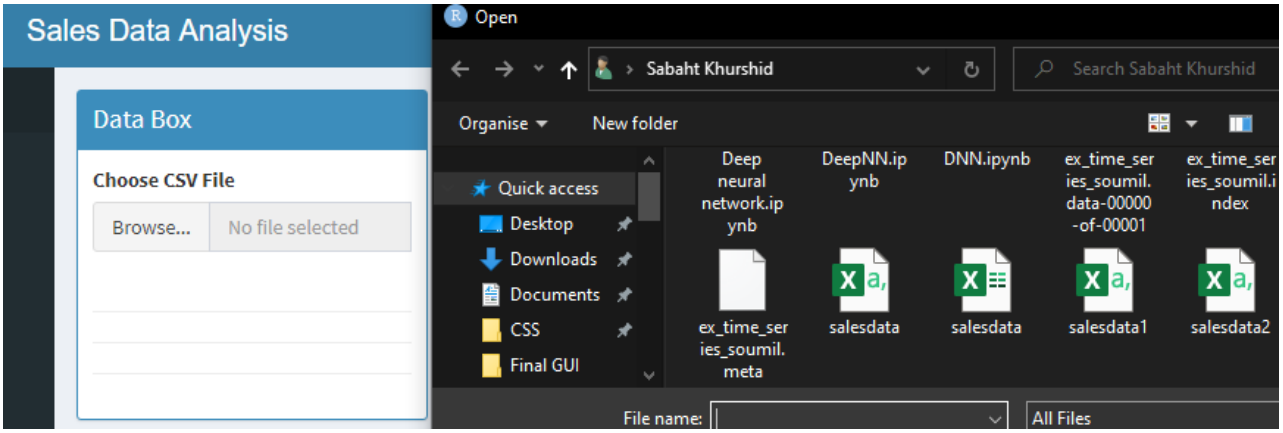
APPENDIX B

USER MANUAL

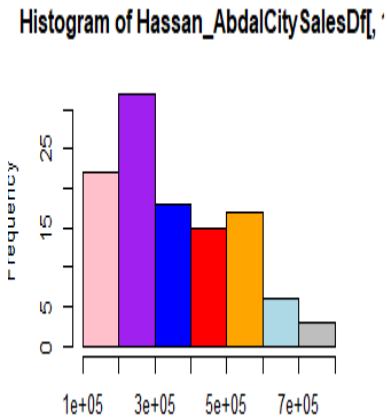
System Summary

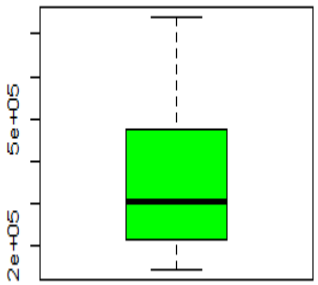
The framework utilizes the best boundaries for the model and makes forecast. The plots of preprocessed information, investigated information, and envisioned information.

Import data set.



Upload, preprocess, analyze and visualize dataset.





Sales Processed Dataset

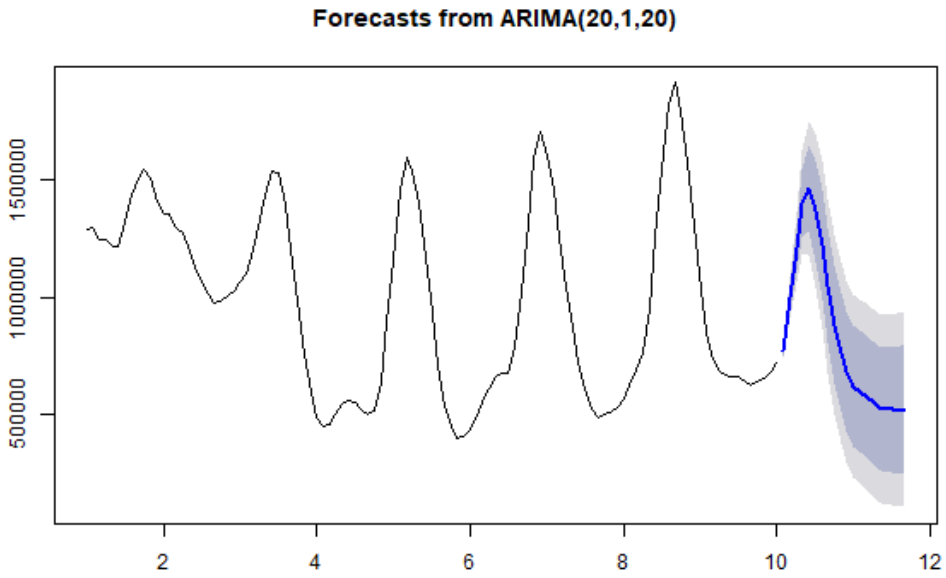
Show 10 entries Search:

	Attock	Fateh Jang	Jand	Hassan Abdal	Hazro
X1	1384970.1	1096237.1	784507.4	498318.4	960722.0
X2	1471041.7	1164364.8	833262.1	529287.3	1020427.
X3	1291073.9	1021915.9	731320.4	464534.1	895588.3
X4	1173703.5	929014.5	664836.8	422303.7	814171.2
X5	1541463.9	1220105.7	873152.3	554625.5	1069278.

Forecast values.

Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
773987.3	752135.2	795839.4	740567.4	807407.2
996043.6	938971.6	1053115.6	908759.5	1083327.7
1212620.6	1115571.4	1309669.8	1064196.6	1361044.6
1402030.2	1257191.0	1546869.4	1180517.8	1623542.7
1464326.2	1279575.6	1649076.9	1181774.5	1746878.0
1368804.3	1155082.1	1582526.4	1041944.4	1695664.1
1224520.3	991215.6	1457825.0	867711.6	1581329.0
1050265.7	806264.9	1294266.5	677098.6	1423432.7
884237.6	634476.6	1133998.6	502261.1	1266214.1
767295.7	514687.1	1019904.2	380964.2	1153627.1
683469.1	429152.0	937786.3	294524.6	1072413.7

Plot forecasted values.



CUSTOMER'S FEEDBACK

Feedback #1

Forecast is according to what was expected. The wide spread of the window is realistic to the state of business.

Feedback #2

From forecasted values following things can be interpreted:

1. Stock in
2. Stock Out

3. Regional buying of retailers
4. Seasonal variations effect on product
5. Any socio-economic activities.
6. Warehouse management

Feedback #3

When forecasted demand is low, variations are higher so the business management should cater risk according to the population.

APPENDIX C

Executive Summary

Product Description/Objectives

The curiosity to know future has always put humans in a struggle to achieve precise details to successfully peek into it. The commercial industries with more clever, sharp and efficient business intelligence prosper speedily as compare to other industries with less efficient business intelligence. It is the need of the day for these units to know about their future insights in order to manage product's productivity, sale and marketing.

Sales predictions is a significant issue for various organizations associated with assembling, co ordinations, advertising, wholesaling and retailing. Modern brands are coordinating new advancements to record and use information to improve their proficiency.

Our system successfully makes valuable predictions with most efficient model for time series that leads to very significant insights.

Methodology Adopted

The details which we have grown through to enable Data Analysis system to make more accurate predictions will be discussed in this section.

Preprocessing, analyzing and visualizing dataset

The given data was prepared and statistically analyzed to remove any stationarity in the data if there was. The data is stationary if p-value is >0.05 then stationary otherwise not.

kpss.test(AllCitySalesDataDf$Hazro)

After efficaciously preprocessing, analyzing and preparing data we went into the procedure of selecting best possible model for the type of data that we have i.e. time series.

Selecting model

There are numerous models available to forecast values when data is time series. Through thorough research we thought to go with ARIMA that works best for time series data.

Smoothing data

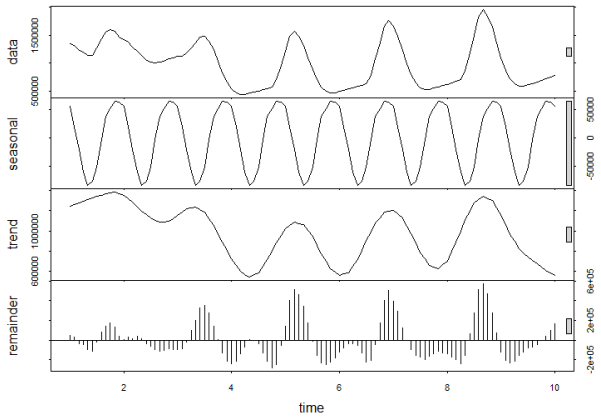
The data need to be free of any outlier or any insignificant value that does not contribute towards predictions. For this we need to smooth data over a correct window called moving average. We tested the data for a couple of values for moving average and concluded 4 is our desired value.

AttockCitySalesDf$MovAvg4=ma(AttockCitySalesDf$cleanData,order=4)

Seasonality in the data

We also need to go for checking seasonal component in the data to see whether we can go for data with seasonal data or not as ARIMA is good for both seasonal and non-seasonal data.

```
decomp=stl(CleanMovAvg4,s.window="periodic")
deseasinalData<-seasadj(decomp) # removing seasonality
plot(decomp)
```



If we find that we can make good spread predictions, we can add seasonality back. We will check for it later

Trend in data

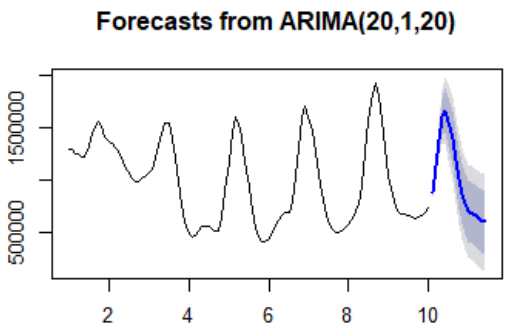
Our data has a trend with some patterns repeated hence we are good to go.

Autoregressive and Moving average component

Now to decide p (autoregressive) and q (moving average) component we look for acf and pacf graphs. Through various speculations we come to find p=20,d=1, q=20 will be good for our predictions.

Fitting and forecasting

The ARIMA model parameters we have decided and the respective changes in the data are good to forecast values and fit the model.



Adding seasonality back

When seasonality is added back it shows less accurate predictions hence we decided to go with ARIMA with non-seasonal data.

Model Evaluation and testing

We adopted the approaches to evaluate and test model

1. Comparison with actual values.

	Actual Values
1	528000.0
2	550000.0
3	572916.7
4	596788.2
5	621654.4
6	647556.6
7	674538.1
8	702643.9
9	867512.3
10	1177415.7

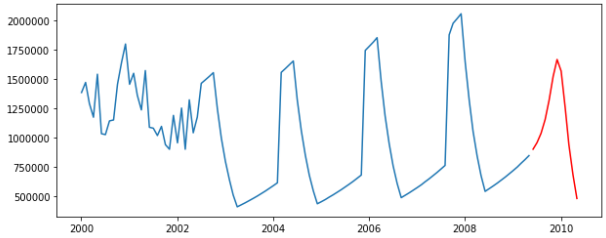
Forecasted Values

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
81	524979.9	501537.8	548421.9	489128.286	560831.4
82	545565.8	483826.2	607305.3	451143.274	639988.3
83	541953.6	438483.6	645423.6	383709.873	700197.4
84	533078.9	379802.4	686355.4	298662.730	767495.1
85	550366.1	357822.1	742910.0	255895.547	844836.6
86	545076.7	326265.9	763887.5	210434.534	879718.9
87	550981.9	311769.2	790194.6	185137.609	916826.1
88	634342.4	379757.2	888927.6	244987.897	1023696.8
89	803541.3	535796.5	1071286.0	394060.978	1213021.6
90	1039968.4	758806.2	1321130.6	609967.912	1469968.9
91	1333307.1	1026950.8	1619463.5	870123.114	1776392.1

2. L-Jung Box test

P-value greater than 0.05 shows that data is not autocorrelated and is best for making predictions.

3. Deep learning LSTM



The graph very much clearly shows that our predictions with ARIMA and LSTM are quite similar.

Results

After assessment and testing model through different propelled approaches we reasoned that ARIMA model we utilized and fitted to foresee the **sales** is increasingly exact and proficient. Thus, we anticipated the offer of every single other city with this model and introduced to the customer. Our customer was a lot of happy with the forecasts and give us certain input of utilizing the model further.

Conclusions

Even though we utilized ARIMA, known to be the best for time series data there are further developed models that help to have progressively sensible and exact figure. SARIMA for occasional information can likewise be applied. For testing and assessment GRU a bigger number of simples than LSTM can be utilized.

REFERENCES

[1] Junaid Umar, Danyal Farukh, Shaheed Hussain, "Crime Pattern Analysis and Prediction System", Military Colleg of Signals, 2017.

[2] Data Science: R Basics | edX, Edx.org, https://www.edx.org/course/data-science-r-basics-2

[3] Mansoor Ahmed, Nashrah Khan, Aniqa Tariq, Ali Sultan,"Missing Data Prediction from Wireless Network Sensors",2019

[4] "Data Science vs. Big Data vs. Data Analytics", 2020

[5] "Chapter 8 ARIMA Models | Forecasting: Principles and Practice."

[6] "The Complete Guide to Time Series Analysis and Forecasting", Medium,2019

[7] Petneházi, Gábor, "Recurrent Neural Networks for Time Series Forecasting", 2018.

[8] Jason Brownlee," A Gentle Introduction to SARIMA for Time Series Forecasting in Python"

[9] "The Complete Guide to Time Series Analysis and Forecasting",2020

[10] Dawes, W.N., 1994, " A Numerical Study of the Interaction of Transonic Compressor Rotor Over Tip Leakage Vortex with the Following Stator Blade Row", ASME Paper No. 94-GT-156

[11] Sharma, O,P., Renaud, E., Butler, T.L., Milasps, K., Dring, R.P., and Joslyn, H.D., 1988, " Rotor- Stator Interaction in Multi- Stage- Axial Flow Turbines", AIAA Paper No. 88-3013

[12] Busby, J.A., Davis, R.L., Dorney, D.J., Dunn, M.G., Haldeman , C.W., Abhari, R.S., Venable, B.L., and Delany, R.A., 1999, " Influence of Vane-Blade Spacing on Transonic Turbine Stage Aerodynamics: Part II- Time- Resolved Data and Analysis:, ASME Journal of Turbomachinery , Vol. 121, pp. 673-682

[13] Collar (1947), "The Expanding Domain of Aeroelasticity," *Journal of the Royal Aeronautical Society* 51-1.

[14] Bell Loyed, "Three dimensional Unsteady flow analysis in vibrating Turbine Cascades", Durham University, 1999.

[15] Jeff Green, PHD Thesis, "Controlling Forced Response of a High Pressure Turbine Blade", Royal Institute of Technology, Stockholm, 200

PLAGIRISM REPORT

Data Analysis for USEN (Plagiarism Report)
ORIGINALITY REPORT

7%	1%	1%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	1%
2	Submitted to Higher Education Commission Pakistan Student Paper	1%
3	machinelearningmastery.com Internet Source	<1%
4	people.duke.edu Internet Source	<1%
5	Submitted to Ain Shams University Student Paper	<1%
6	Submitted to Westcliff University Student Paper	<1%
7	Submitted to Central Queensland University Student Paper	<1%
8	Submitted to Trinity College Dublin Student Paper	<1%
9	Submitted to Universiti Teknikal Malaysia	

	Melaka Student Paper	<1%
10	Submitted to Colorado State University, Global Campus Student Paper	<1%
11	Submitted to Informatics Education Limited Student Paper	<1%
12	Submitted to Amity University Student Paper	<1%
13	Submitted to University of Denver Student Paper	<1%
14	Submitted to Federation University Student Paper	<1%
15	Submitted to Manchester Metropolitan University Student Paper	<1%
16	Submitted to Colorado Technical University Online Student Paper	<1%
17	Submitted to Middlesex University Student Paper	<1%
18	P. Muneeshwari, G. Athisha. "Extended artificial immune system-based optimized access control for big data on a cloud environment", International Journal of Communication	<1%

Systems, 2019
Publication

19	Submitted to University of West Florida Student Paper	<1%
20	Submitted to The Stockholm School of Economics in Riga Student Paper	<1%
21	Submitted to Misr International University Student Paper	<1%
22	Manohar Swamynathan. "Mastering Machine Learning with Python in Six Steps", Springer Science and Business Media LLC, 2019 Publication	<1%
23	Submitted to City of Glasgow College Student Paper	<1%
24	Submitted to Staffordshire University Student Paper	<1%
25	Submitted to London School of Commerce Student Paper	<1%
26	Submitted to (school name not available) Student Paper	<1%
27	Pankaj Jalote. "An Integrated Approach to Software Engineering", Springer Science and Business Media LLC, 1991 Publication	<1%

28	Submitted to University of Sunderland Student Paper	<1%
29	tuhin2nitdgp.wordpress.com Internet Source	<1%
30	Submitted to Cranfield University Student Paper	<1%
31	Submitted to Eiffel Corporation Student Paper	<1%
32	Scott M. Eisenkop, Nick M. Spirtos. "Division of pedicles by stapling during cytoreductive surgery for ovarian cancer", Gynecologic Oncology, 2005 Publication	<1%
33	Submitted to 65046 Student Paper	<1%
34	Submitted to International School of Management and Technology Student Paper	<1%
35	Submitted to Bocconi University Student Paper	<1%
36	Stefania Loredana Nita, Marius Mihailescu. "Chapter 8 Debugging Techniques Used in Big Data", Springer Science and Business Media LLC, 2017 Publication	<1%