SALES PREDICTION IN TIME SERIES FORECASTING USING ARIMA MODEL

KALAIVANI.S (19PCS007)

A PROJECT REPORT SUBMITTED TO

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND HIGHER EDUCATION FOR WOMEN

COIMBATORE – 641043

DEPARTMENT OF COMPUTER SCIENCE

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE

MASTER'S DEGREE IN COMPUTER SCIENCE

APRIL – 2021

SALES PREDICTION IN TIME SERIES FORECASTING USING ARIMA MODEL

KALAIVANI.S (19PCS007)

A PROJECT REPORT SUBMITTED TO

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND HIGHER EDUCATION FOR WOMEN

COIMBATORE – 641043

DEPARTMENT OF COMPUTER SCIENCE

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE

MASTER'S DEGREE IN COMPUTER SCIENCE

APRIL – 2021

SIGNATURE OF THE SUPERVISOR **SIGNATURE OF THE HEAD OF THE DEPARTMENT**

VIVA-VOCE EXAMINATION HELD ON_____

SIGNATURE OF EXAMINER

DECLARATION

DECLARATION

We hereby declare that this project work entitled **'SALES PREDICTION IN TIME SERIES FORECASTING USING ARIMA MODEL'** submitted to **Avinashilingam Institute for Home Science and Higher Education for Women , Coimbatore** in partial fulfillment of the degree of Master of Computer Science is a record of original work done by us under the guidance of **Mrs.Dr.N.VALLIAMMAL Ph.D, Assistant professor(SS) Department of Computer Science,** Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore and this project has not formed the basis for the award of any Degree/Diploma/Associate Ship/Fellowship or similar title to any candidate of this University.

SIGNATURE OF THE GUIDE **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **God Almighty**, for his constant love and grace that he showered upon us.

We would like to express our deep sense of reverential gratitude and sincere thanks to **Prof. S.P. Thyagarajan, Chancellor,** Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for his support and encouragement during the course of our study.

We owe our great deal of gratitude to **Dr. Premavathy Vijayan M.Sc., M.Ed., Dip.Spl.Edn., M.Phil., Ph.D., Vice Chancellor**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for extending all resources that facilitated the conduct of the present study.

We express our gratitude to **Dr. S. Kowsalya, M.Sc., M.Phil., Ph.D., Registrar**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing all facilities necessary for the study.

We are also thankful to **Dr. K. Udaya Chandrika, M.Sc., M.Phil., Ph.D., Dean School of Physical Sciences & Computational Sciences,** Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for granting the facility required.

We wish to place our deep sense of gratitude to **Dr. Vasantha kalyani devid, M.Sc., M.Phil., Ph.D., Professor and Head, Department of Computer Science,** for providing all the facilities required to complete the project.

We express our honorable thanks to our project coordinator **Dr. G.Sudhamathy M.C.A., Ph.D.Assistant Professor (SS), Department of Computer Science,** for her kind valuable advice and knowledgeable suggestions which helped us to complete our project successfully.

We heartily thank our esteemed project guide **Dr. N. Valliammal, M.Sc.,M.Phil., Ph.D., Assistant Professor (SS), Department of Computer Science**, for imparting the tremendous assistance and well-timed support for triumph of our project.

We would like to express our sincere gratitude to all the staff members of the Department of Computer Science, for their constant encouragement and for the opportunity to do our project in this esteemed university. Last yet importantly, we would like to thank our parents, family members, friends and all well-wishers for their kind inspiration, blessings and encouragement during the completion of the course of project.

ABSTRACT

ABSTRACT

The project named "**SALES PREDICTION IN TIME SERIES FORECASTING USING ARIMA MODEL**" has attempted to develop a ARIMA Model to predict. A time series analysis model involves using historical data to forecast the future. **Sales forecasting** is the process of estimating future revenue by predicting the amount of product or services a **sales** unit (which can be an individual salesperson, a **sales** team, or a company). By past values we can predict future values, for prediction we need a dataset. Dataset is collected from Kaggle. By ARIMA model and SARIMAX we are going to predict the sales forecasting. Based on level of sales increase, we can select the best month out of it. The problem statement is to extract the sales forecasting by using the data science techniques and describe the subject of a document. ARIMA (Autoregressive Integrated Moving Average), SARIMAX Model is used in this project to predict the future values. A comparative study has been conducted to decide which month is best for this project to train the behaviour features of sales is stationary or non-stationary. The aim of this project is to predict a system which can perform early prediction of sales forecasting for a superstore. This project aims to predict the furniture sales and office supplies, With the help of ARIMA Model and SARIMAX, one can understand the sales how differ in every month by plotting of graphs. Combination for seasonal ARIMA is used for parameter Selection of our furniture's sales ARIMA Time Series Model. The **best ARIMA model** have been selected by using the criteria such as AIC (Akaike Information Criterion). To select the **best ARIMA model** the data split into two periods, as estimation period and validation period. The **model** for which the values of criteria are smallest is considered as the **best model**. The main objective of this project is to predict the sales forecasting using ARIMA, SARIMAX Models. The system which helps user to know about sales forecasting using ARIMA and SARIMAX model in an effective manner.

.

TABLE OF CONTENTS

TABLE OF CONTENTS

INTRODUCTION

I INTRODUCTION

1.1. Problem Definition

The purpose of the project is to predict the sales forecasting which is used to solve the amount of product or services a **sales** unit. As per the present system the User can predict the sales forecasting by applying ARIMA and SARIMAX models. Users have to download the software and packages prompt. Based on some dynamic values this software will process and display result to User. The problem statement is to extract the sales forecasting by using the data science techniques and describe the subject of a document

1.2. Overview of Project

Time Series is a series of observations taken at specified time intervals usually equal intervals. Analysis of the series helps us to predict future values based on previous observed values. In Time series, we have only 2 variables, time & the variable we want to forecast. Time series data can be analysed in order to extract meaningful statistics and other characteristics. It's used in at least the 4 scenarios: **a) Business Forecasting, b) Understand past behaviour, c) Plan the future, d) Evaluate current accomplishment.** By past values we can predict future values, for prediction we need a dataset. Dataset is collected from a Kaggle and it contain 21 attributes of sales. By building a Arima model we are going to predict the sales forecasting. Based on level of sales increase, we can select the best month out of it. They are 11 different classical time series forecasting methods; they are:

- Autoregression (AR)
- Moving Average (MA)
- Autoregressive Moving Average (ARMA)
- Autoregressive Integrated Moving Average (ARIMA)
- Seasonal Autoregressive Integrated Moving-Average (SARIMA)
- Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors (SARIMAX)
- Vector Autoregression (VAR)
- Vector Autoregression Moving-Average (VARMA)
- Vector Autoregression Moving-Average with Exogenous Regressors (VARMAX)
- Simple Exponential Smoothing (SES)
- Holt Winter’s Exponential Smoothing (HWES)

Above models we are going to use ARIMA and SARIMAX model to predict. The best ARIMA model have been selected by using the criteria such as AIC, AICc, SIC, AME, RMSE and MAPE etc. To select the best ARIMA model the data split into two periods, estimation period and validation period and select the optimal parameter values for our ARIMA(p,d,q) (P,D,Q)s time series model. We will use a "grid search" to iteratively explore different combinations of parameters. For each combination of parameters, we fit a new seasonal ARIMA model with the SARIMAX () function from the stats models module and assess its overall quality. Once we have explored the entire landscape of parameters, our optimal set of parameters will be the one that yields the best performance for our criteria of interest. The model for which the values of criteria are smallest is considered as the best model. The Akaike Information Criteria (AIC) is a widely used measure of a statistical model. It basically quantifies 1) the goodness of fit, and 2) the simplicity/parsimony, of the model into a single statistic. When comparing two models, the one with the lower AIC is generally "better".

SYSTEM CONFIGURATION

2.SOFTWARE & HARDWARE SPECIFICATION

Hardware Specification :

CPU Type : Intel I3 Processor

Clock Speed : 3.0 Ghz

Ram Size : 4 GB

Hard Disk Capacity : 500 GB

Monitor Type : 15 Inch Color Monitor

Keyboard Type : Standard Keyboard

Software Specification :

Operating System : Windows 10, Mac, Linux, Etc. (Any One)

Language : PYTHON 3.7

IDE : Spyder(anaconda3)

Documentation : Ms-office

2.3.1 Python Programming

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted**: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- **Python is Interactive**: You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented**: Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language**: Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

PYTHON FEATURES

Some of the features of Python are

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

USES OF PYTHON

Python are widely used for

- System programming
- GUI Programming
- Internet scripting
- Database Programming

- Gaming, Robotics
- Mozilla
- Government and non-profit organization

PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. The fast edit-test-debug cycle makes this simple approach very effective.

2.3.2 About the Software

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package. Beyond its many built-in features, its abilities can be extended even further via its plugin system and API. Furthermore, Spyder can also be used as a PyQt5 extension library, allowing developers to build upon its functionality and embed its components, such as the interactive console, in their own PyQt software.

There are currently **118932 packages** here. **There** are 250+ **libraries** in **python**. Most of the Python programs are submitted to the Python Package Index aka PyPI[1] which acts as a central repository.

Matplotlib - Matplotlib helps with data analysing, and is a numerical plotting library.

Pandas

Like we've said before, Pandas is a must for data-science. It provides fast, expressive, and flexible data structures to easily (and intuitively) work with structured (tabular, multidimensional, potentially heterogeneous) and time-series data.

Requests - Requests is a Python Library that lets you send HTTP/1.1 requests, add headers, form data, multipart files, and parameters with simple Python dictionaries. It also lets you access the response data in the same way.

NumPy - It has advanced math functions and a rudimentary scientific computing package.

SciPy - Next up is SciPy, one of the libraries we have been talking so much about. It has a number of user-friendly and efficient numerical routines. These include routines for optimization and numerical integration.

Spyder features:

- An editor with syntax highlighting, introspection, code completion, Support for multiple I Python consoles
- The ability to explore and edit variables from a GUI
- A Help pane able to retrieve and render rich text documentation on functions, classes and methods automatically or on-demand
- A debugger linked to IPdb, for step-by-step execution
- Static code analysis, powered by Pylint, A run-time Profiler, to benchmark code.
- Project support, allowing work on multiple development efforts simultaneously
- A built-in file explorer, for interacting with the file system and managing projects
- A "Find in Files" feature, allowing full regular expression search over a specified scope
- An online help browser, allowing users to search and view Python and package documentation inside the IDE

SYSTEM STUDY AND ANALYSIS

3.SYSTEM STUDY AND ANALYSIS

3.1. Existing System

Any forecast can be termed as an indicator of what is likely to happen in a specified future time frame in a particular field. Therefore, the sales forecast indicates as to how much of a particular product is likely to be sold in a specified future period in a specified market at specified price. Accurate sales forecasting is essential for a business to enable it to produce the required quantity at the right time. Some firms manufacture on the order basis, but in general, firm produces the material in advance to meet the future demand.

Drawbacks:

- **Time-Intensive Completion** - The time spent forecasting is less time spent selling. In more data-driven processes, a company often has marketing, IT and sales staff involved in building a system to collect and interpret data.
- **Expensive Technology Tools** - Manual processes are not as technology-oriented, but computer tools such as spreadsheets are commonly used.
- **Internal Bias -** The challenge for company marketing and sales reps in preparing forecasts is that internal bias is hard to avoid. Sales reps look better and tend to earn more commission when they achieve high sales goals. When sales forecasts are high, companies could invest too much in inventory and resources in preparation for selling activities.

3.2. Proposed System

The drawbacks, which are faced during existing system, can be eradicated by using the proposed applications. Proposed to make the new system extremely user friendly. Using this application, user can check the sales forecasting how it varies by time.

Benefits

(i) It forms a basis of sales budget, production budget natural budget etc, It helps in deciding policies.

(ii) It helps in taking decision about the plant expansion and changes in production mix or should it divert its resource for manufacturing other products.

SYSTEM DESIGN

4. SYSTEM DESIGN

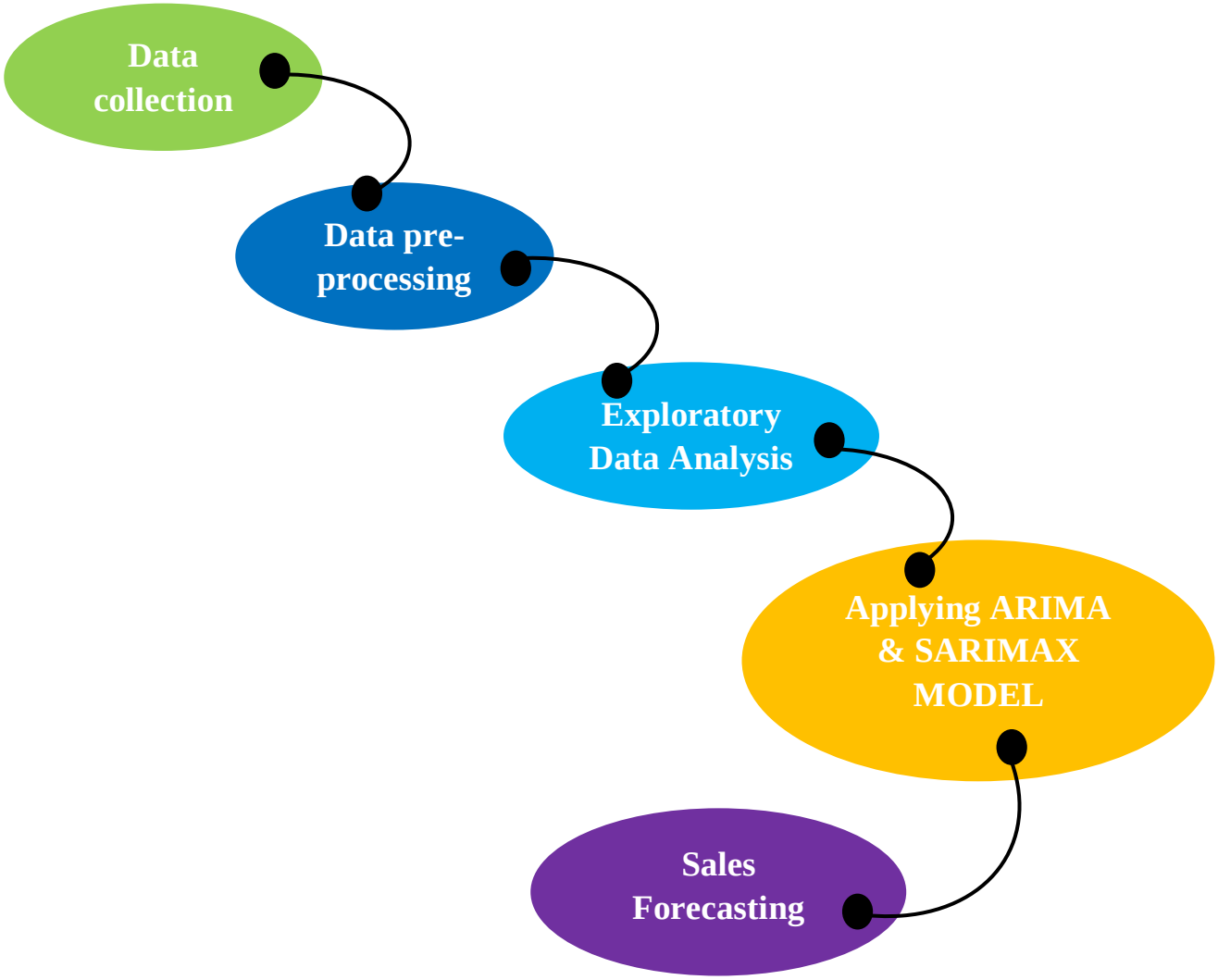
4.1. Output Design

The output generated by the system is often regarded as the criteria for evaluating the performance of the system. For the proposed system, it is necessary that the output should be compatible with the existing manual reports. The outputs have been formatted with this consideration in mind. The outputs are obtained after all the phase, from the system can be displayed or can be produced in the soft copy. The soft copy is highly preferred since it can be used by the controller section for future reference and it can be used for maintaining the record. In our project we designed various outputs such as data that is processed, visualised, compared using various models such as ARIMA, SARIMA. Finally, the report shows the output as that sales how differ in every month.

SYSTEM DEVELOPMENT

5. SYSTEM DEVELOPMENT

5.1 METHODOLOGY



5.2. Modules

This project consists of various modules such as,

5.2.1. Data Collection.

5.2.2. Data Pre-processing.

5.2.3. Exploratory Data Analysis.

5.2.4. Applying Model (ARIMA) 'Auto Regressive Integrated Moving Average', and (SARIMAX) 'Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors'.

5.2.5. Sales Forecasting.

5.3. Module Description

5.3.1. Data Collection

Data collection is defined as the procedure of collecting, measuring and analysing accurate insights for research using standard validated techniques. In this project the sales dataset is collected from Kaggle. Superstore sales dataset contain 9994 instances and 21 attributes of sales.

5.3.2. Data Pre-processing

Data pre-processing is a data mining technique which is used to transform the raw data in a useful and efficient format. In data pre-processing method we have used data cleaning method to Deal with missing data, Remove Noise from data, remove outliers from data, Dealing with Duplicate data, Remove inconsistencies from data.

5.2.3. Exploratory Data Analysis

Exploratory data analysis (EDA) is used by data scientists to analyse and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers we need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions. It can also help determine if the statistical techniques we are considering for data analysis are appropriate. The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

5.2.4. Build The (ARIMA) Auto Regressive Integrated Moving Average Model and (SARIMAX) Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors

ARIMA (Auto Regressive Integrated Moving Average) is a combination of 2 models AR (Auto Regressive) & MA (Moving Average). It has 3 hyperparameters - P (auto regressive lags), d (order of differentiation), Q (moving avg.) which respectively comes from the AR, I & MA components. The AR part is correlation between previous & current time periods. To smooth

out the noise, the MA part is used. The I part binds together the AR & MA parts. Before building a ARIMA model we have to check the sales dataset is Stationarity**. By using value count command, we can know the data's, in that, we are going to select the first two attributes which has highest sales.** Before applying any statistical model on a Time Series, the series has to be stationary, which means that, over different time periods, It should have constant mean, It should have constant variance or standard deviation, Auto-covariance should not depend on time. Trend & Seasonality are two reasons why a Time Series is not stationary & hence need to be corrected. There are 2 ways to check for Stationarity of a TS:

1. Rolling Statistics - Plot the moving average or moving standard deviation to see if it varies with time. It's a visual technique.

2. ADCF Test - Augmented Dickey–Fuller test is used to gives us various values that can help in identifying stationarity. The Null hypothesis says that a TS is non-stationary. It comprises of a **Test Statistics** & some **critical values** for some confidence levels. If the Test statistics is less than the critical values, we can reject the null hypothesis & say that the series is stationary. THE ADCF test also gives us a **p-value**. According to the null hypothesis, lower values of p is better. By applying these techniques, we can predict the future sales of selected attributes. Here I have used Rolling Statistics to check the data is stationary or non-stationary. There are many guidelines and best practices to achieve this goal, yet the correct parametrization of ARIMA models can be a painstaking manual process that requires domain expertise and time. We will use a "grid search" to iteratively explore different combinations of parameters. For each combination of parameters, we fit a new seasonal ARIMA model with the SARIMAX () function from the stats models module and assess its overall quality. Once we have explored the entire landscape of parameters, our optimal set of parameters will be the one that yields the best performance for our criteria of interest. Both AIC measures how well a model fits the data while taking into account the overall complexity of the model. A model that fits the data very well while using lots of features will be assigned a larger AIC score than a model that uses fewer features to achieve the same goodness-of-fit. Therefore, we are interested in finding the model that yields the lowest AIC value. The output of our code suggests that SARIMAX (1, 1, 1)x(1, 1, 1, 12) yields the lowest AIC value of 277.78. We should therefore consider this to be optimal option out of all the models we have considered.

5.3.5. Forecasting

Sales forecasting is the process of estimating future sales. Accurate sales forecasts enable companies to make informed business decisions and predict short-term and long-term performance. Companies can base their forecasts on past sales data, industry-wide comparisons, and economic trends. It is easier for established companies to predict future sales based on years of past business data. Sales forecasting gives insight into how a company should manage its workforce, cash flow, and resources. In addition to helping a company allocate its internal resources effectively, predictive sales data is important for businesses when looking to acquire investment capital. Sales forecasting allows companies to, predict achievable sales revenue, efficiently allocate resources, Plan for future growth. Therefore, forecasting is shown in graphical plot.

Table 1: Advantages and Disadvantages of ARIMA Models.

Model	Pros	Cons
Linear Regression	Ability to handle different time series components and features. High interpretability.	Sensitive to outliers. Strong assumptions.
Exponential Smoothing	Ability to handle variable level, trend, and seasonality components. Automated optimization.	Sensitive to outliers. Narrow confidence intervals.
ARIMA (Autoregressive Integrated Moving Average)	High interpretability. Realistic confidence intervals. Unbiased forecasts.	Requires more data. Strong restrictions and assumptions. Hard to automate.
Dynamic Linear Model	High interpretability. More transparent than other models. Deals well with uncertainty. Control the variance of the components.	Higher holdout error. Higher training and evaluation time.
Neural Network Model	Less restrictions and assumptions. Ability to handle complex nonlinear patterns. High predictive power. Can be easily automated.	Low interpretability. Difficult to derive confidence intervals for the forecasts. Requires more data.

SYSTEM IMPLEMENTATION

6. SYSTEM IMPLEMENTATION

6.1.1 PLANNING AND SCOPING

The scoping phase involves mapping the shape of your business and forecasting process to the various Prophecy 'dimensions', which are:

- Forecasted measures, additional (e.g. financial) measures, solve formulae
- Time periods - history and forecast horizons
- Forecast versions (e.g. rolling 12 versions, version incremented monthly)
- Security design by customer, product and measure
- The business process and reporting structure required to support the forecasting cycle

Most forecasting requirements and processes can be mapped to these elements. Once they are agreed the implementation can move on to:

- Sourcing product and customer hierarchies from existing business systems, Sourcing historical actuals from existing business systems.
- Where relevant, sourcing future product standard costs and customer prices from existing business systems. Note that all of the above will be re-used to feed the monthly actuals update process - only the time period for the data imports differs.

6.1.2. MODEL PREPARTION

The build process means building the ARIMA Model using AR, MV which is 'the other half' of the ARIMA. Because the imports are scripted it is trivial to re-run the build process any number of times, until it is fully validated.

6.1.3. PSEUDO CODE AND MODEL FITTING

The ARIMA forecasting equation for a stationary time series is a linear (i.e., regression-type) equation in which the predictors consist of lags of the dependent variable and/or lags of the forecast errors. That is: **Predicted value of Y = a constant and/or a weighted sum of one or more recent values of Y and/or a weighted sum of one or more recent values of the errors.**

If the predictors consist only of lagged values of Y, it is a pure autoregressive ("self-regressed") model, which is just a special case of a regression model and which could be fitted with standard regression software. If some of the predictors are lags of the errors, an ARIMA model it is NOT a linear regression model, because there is no way to specify "last period's error" as an independent variable: the errors must be computed on a period-to-period basis when the model is fitted to the data. From a technical standpoint, the problem with using lagged errors as predictors is that the model's predictions are not linear functions of the coefficients, even though they are linear functions of the past data. So, coefficients in ARIMA models that include lagged errors must be estimated by nonlinear optimization methods ("hill-climbing") rather than by just solving a system of equations.Lags of the stationaries series in the forecasting equation are called "autoregressive" terms, lags of the forecast errors are called "moving average" terms, and a time series which needs to be differenced to be made stationary is said to be an "integrated" version of a stationary series. **Random-walk and random-trend models, autoregressive models, and exponential smoothing models are all special cases of ARIMA models.**

6.1.4 General Notation for ARIMA and SARIMA Models

- **ARIMA:**

ARIMA is an acronym for Auto Regressive Integrated Moving-Average. The order of an ARIMA model is usually denoted by the notation ARIMA (p,d,q), where p is the order of the autoregressive part d is the order of the differencing q is the order of the moving-average process If no differencing is done (d = 0), the models are usually referred to as ARMA(p,q) models. The final model in the preceding example is an ARIMA (1,1,1) model since the IDENTIFY statement specified d = 1, and the final ESTIMATE statement specified p = 1 and q = 1.The forecasting equation is constructed as follows. First, let y denote the d^{th} difference of Y, which means:

If d=0: $y_t = Y_t$

If d=1: $y_t = Y_t - Y_{t-1}$

If d=2: $y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$

Note that the second difference of Y (the d=2 case) is not the difference from 2 periods ago. In terms of y, the general forecasting equation is:

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

Here the moving average parameters (θ's) are defined so that their signs are negative in the equation, following the convention introduced by Box and Jenkins. Some authors and software define them so that they have plus signs instead. When actual numbers are plugged into the equation, there is no ambiguity, but it's important to know which convention your software uses when you are reading the output. Often the parameters are denoted there by AR(1), AR(2), …, and MA(1), MA(2), … etc.. To identify the appropriate ARIMA model for Y, we begin by determining the order of differencing (d) needing to stationarize the series and remove the gross features of seasonality, perhaps in conjunction with a variance-stabilizing transformation such as logging or deflating. However, the stationarized series may still have autocorrelated errors, suggesting that some number of AR terms ($p \geq 1$) and/or some number MA terms ($q \geq 1$) are also needed in the forecasting equation.

- **SARIMAX**

A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models we have seen so far. It is written as follows:

ARIMA	(p,d,q)	(P,D,Q)m
	↑↑	↑↑
	Non-seasonal part of the model	Seasonal part of of the model

The seasonal part of the model consists of terms that are similar to the non-seasonal components of the model, but involve backshifts of the seasonal period. For example, an ARIMA(1,1,1)(1,1,1)44 model (without a constant) is for quarterly data (m=4m=4), and can be written as

$(1-\phi 1B)\ (1-\Phi 1B4)(1-B)(1-B4)yt=(1+\theta 1B)\ (1+\Theta 1B4)\varepsilon t.(1-\phi 1B)\ (1-\Phi 1B4)(1-B)(1-B4)yt=(1+\theta 1B)\ (1+\Theta 1B4)\varepsilon t.$

The additional seasonal terms are simply multiplied by the non-seasonal terms. We will use a "grid search" to iteratively explore different combinations of parameters. For each

combination of parameters, we fit a new seasonal ARIMA model with the SARIMAX() function from the statsmodels module and assess its overall quality. Once we have explored the entire landscape of parameters, our optimal set of parameters will be the one that yields the best performance for our criteria of interest. combinations of parameters and uses the SARIMAX function from statsmodels to fit the corresponding Seasonal ARIMA model. Here, the order argument specifies the (p, d, q) parameters, while the seasonal_order argument specifies the (P, D, Q, S) seasonal component of the Seasonal ARIMA model. After fitting each SARIMAX ()model, the code prints out its respective AIC score. The output of our code suggests that SARIMAX(1, 1, 1)x(1, 1, 1, 12) yields the lowest AIC value of 277.78. We should therefore consider this to be optimal option out of all the models we have considered. The summary attribute that results from the output of SARIMAX returns a significant amount of information, but we'll focus our attention on the table of coefficients. When fitting seasonal ARIMA models (and any other models for that matter), it is important to run model diagnostics to ensure that none of the assumptions made by the model have been violated. ur primary concern is to ensure that the residuals of our model are uncorrelated and normally distributed with zero-mean. If the seasonal ARIMA model does not satisfy these properties, it is a good indication that it can be further improved.

In this case, our model diagnostics suggests that the model residuals are normally distributed based on the following:

- In the top right plot, we see that the red KDE line follows closely with the N(0,1) line (where N(0,1)) is the standard notation for a normal distribution with mean 0 and standard deviation of 1). This is a good indication that the residuals are normally distributed.
- The qq-plot on the bottom left shows that the ordered distribution of residuals (blue dots) follows the linear trend of the samples taken from a standard normal distribution with N(0, 1). Again, this is a strong indication that the residuals are normally distributed.
- The residuals over time (top left plot) don't display any obvious seasonality and appear to be white noise. This is confirmed by the autocorrelation (i.e. correlogram) plot on the bottom right, which shows that the time series residuals have low correlation with lagged versions of itself.

Those observations lead us to conclude that our model produces a satisfactory fit that could help us understand our time series data and forecast future values. Although we have a satisfactory fit, some parameters of our seasonal ARIMA model could be changed to improve our model fit. For example, our grid search only considered a restricted set of parameter combinations, so we may find better models if we widened the grid search. After model fitting, trend and pattern visualize the final forecasting. Good to see that the sales for both furniture and office supplies have been linearly increasing over time and will be keep growing, although office supplies' growth seems slightly stronger. The worst month for furniture is April, the worst month for office supplies is February. The best month for furniture is December, and the best month for office supplies is October.

Table 2: Shows the Pseudo code of Sales forecasting using ARIMA, SARIMAX Model

	PSEUDO CODE FOR ARIIMA MODEL
Step 1:	Importing all necessary libraries.
Step 2:	Importing the dataset and visualise the data by reading the attributes.
Step 3:	Calling the highest sales values by using value_count() command to check those datas are stationary or not.
Step 4:	Creating the ARIMA, SARIMAX model to check the data is stationary or not and based on AIC value, the model is selected out of it.
Step 5:	After the AIC value outcome, here the Arima model is selected.Then the ARIMA Model is fitted for both furniture sale and office supplies attributes in the dataset.
Step 6:	Getting the plot visualization of trend and pattern we can get how the sales is forecasting.
Step 7:	By Using ARIMA Model the best month is selected and explored.
Step 8:	End the process.

CONCLUSION

7. CONCLUSION

Sales prediction is rather a regression problem than a time series problem. The use of regression approaches for sales forecasting can often give us better results compared to time series methods. One of the main assumptions of regression methods is that the patterns in the historical data will be repeated in future. The accuracy on the validation set is an important indicator for choosing an optimal number of iterations of ARIMA Models. The effect of ARIMA generalization consists in the fact of capturing the patterns in the whole set of data. This effect can be used to make sales prediction when there is a small number of historical data for specific sales time series in the case when a new product or store is launched. In stacking approach, the results of multiple model predictions on the validation set are treated as input regressors for the next level models. The **best ARIMA model** have been selected by using the criteria such as AIC, AICc, SIC, AME, RMSE and MAPE etc. To select the **best ARIMA model** the data split into two periods, viz. estimation period and validation period. The **model** for which the values of criteria are smallest is considered as the **best model**. Here AIC Criteria is used and it produces the output yields the lowest AIC value of 297.78. The summary attribute that results from the output of SARIMAX returns a significant amount of information, but we'll focus our attention on the table of coefficients. Here, each weight has a p-value lower or close to 0.05, so it is reasonable to retain all of them in our model. Therefore, we should consider this to be optimal option. Combination for seasonal ARIMA is used for parameter Selection of our furniture's sales ARIMA Time Series Model. Although we have a satisfactory fit, some parameters of our seasonal ARIMA model could be changed to improve our model fit. For example, our grid search only considered a restricted set of parameter combinations, so we may find better models if we widened the grid search. Using stacking makes it possible to take into account the differences in the results for multiple models with different sets of parameters and improve accuracy on the validation and on the out-of-sample data sets.

8. SCOPE FOR FUTURE ENHANCEMENT

- Demand forecasting is an important function of managing supply chain.
- Its integration with other business functions makes it one of the most important planning processes business can deploy for future.
- The historical demand data were used to develop several models and the adequate one was selected according to four performance criteria. The model that we selected and which minimizes the four previous criteria is ARIMA (1, 0, 1).
- The results obtained proves that this model can be used for modelling and forecasting the future demand in this Sales Forecasting; these results will provide to managers of this manufacturing reliable guidelines in making decisions.
- As future work, we will develop other models by using a combination of qualitative and quantitative techniques to generate reliable forecasts and increase the forecast accuracy. We will also try neural network approach to compare it with ARIMA's results.
- Furthermore, we will make an ARIMA-radial basis function (RBF) combination always to achieve the same goal: high result.

BIBLIOGRAPHY

RESEARCH PAPERS:

- Giuseppe Nunnari and Valeria Nunnari, "Forecasting Monthly Sales Retail Time Series: A Case Study", *Proc. of IEEE Conf. on Business Informatics (CBI)*, July 2017.
- Smt. Rajeshwari and M. Shettar, "Emerging Trends of E-Commerce in India: An Empirical Study", International Journal of Business and Management Invention ISSN (Online), vol. 5, no. 9, pp. 2319-8028, September. 2016.
- Q. Huang and F. Zhou, "Research on retailer data clustering algorithm based on spark", AIP Conference Proceedings, vol. 1820, no. 1, pp. 080022, 2017, March.
- Ho, SL, Xie, M, Goh, TN. A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction.

REFERENCE BOOK:

- **John T. Mentzer - University of Tennessee, Knoxville, USA, Tennessee Department of Mental Health and Mental Retardation**

- **Mark A. Moon - University of Tennessee-Knoxville, Tennessee, University of Tennessee-Knoxville, USA**
- **http://sk.sagepub.com/books/sales-forecasting-management-2e**

REFERENCE WEBSITES:

- **https://towardsdatascience.com/5-machine-learning-techniques-for-sales-forecasting-598e4984b109**
- **https://otexts.com/fpp2/seasonal-arima.html**
- **https://www.wisdomgeek.com/development/machine-learning/sarima-forecast-seasonal-data-using-python/**

CODING:

```
import warnings

import itertools

import numpy as np

import matplotlib.pyplot as plt

warnings.filterwarnings("ignore")

plt.style.use('fivethirtyeight')

import pandas as pd

import statsmodels.api as sm

import matplotlib

matplotlib.rcParams['axes.labelsize'] = 14

matplotlib.rcParams['xtick.labelsize'] = 12

matplotlib.rcParams['ytick.labelsize'] = 12

matplotlib.rcParams['text.color'] = 'k'

df = pd.read_csv("C:/Users/Kalaivani/Desktop/Final/Sample-Superstore.csv")

df.head()

df.tail()

df.shape

df.columns
```

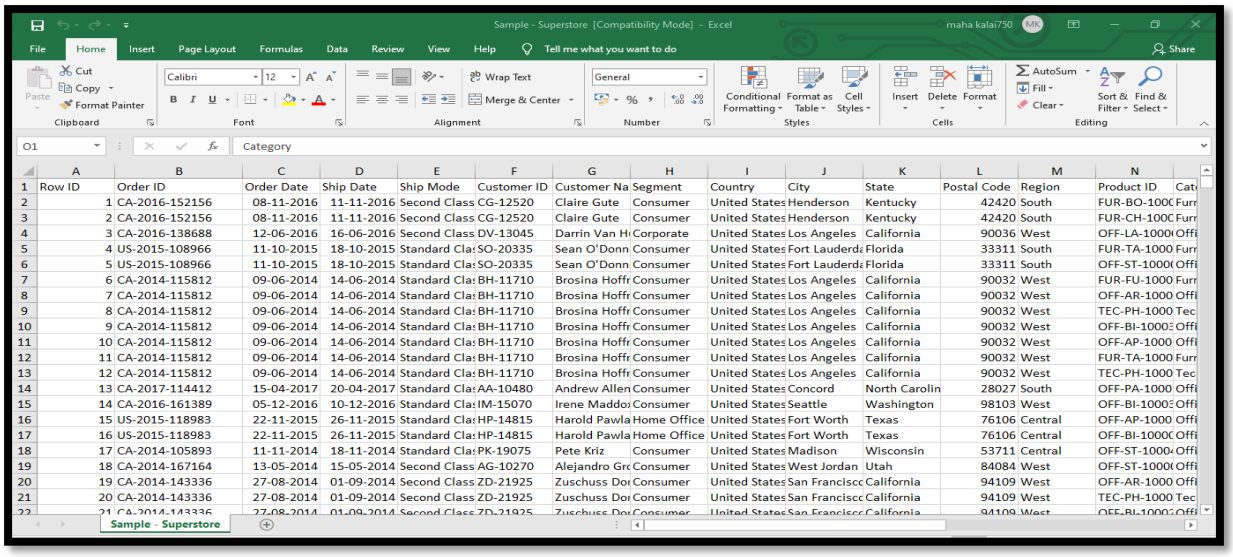
```
df.dtypes

df.isnull().sum()

df=df.drop('Row ID',axis=1)

df.head()

df['Category'].unique()

df['Category'].value_counts()

df['Sub-Category'].nunique()

df['Sub-Category'].value_counts()

plt.figure(figsize=(16,8))

plt.bar('Sub-Category','Category',data=df,color='b',width=0.8)

plt.show()


plt.figure(figsize=(12,10))

df['Sub-Category'].value_counts().plot.pie(autopct="%1.1f%%")

plt.show()

#The store has wide variety of Office Supplies especially in Binders and Paper department.

df.groupby('Sub-Category')['Profit','Sales'].agg(['sum']).plot.bar()

plt.title('Total Profit and Sales per Sub-Category')

# plt.legend('Profit')

# plt.legend('Sales')

plt.show()

furniture = df.loc[df['Category'] == 'Furniture']

furniture['Order Date'].min(), furniture['Order Date'].max()

cols = ['Row ID', 'Order ID', 'Ship Date', 'Ship Mode', 'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State', 'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category', 'Product Name', 'Quantity', 'Discount', 'Profit']

furniture.drop(cols, axis=1, inplace=True)

furniture = furniture.sort_values('Order Date')

furniture.isnull().sum()
```

```
furniture = furniture.groupby('Order Date')['Sales'].sum().reset_index()
furniture = furniture.set_index('Order Date')
furniture.index
df.index = pd.to_datetime(df.index, unit='s')
y = furniture['Sales'].resample('MS').mean()
y['2017':]
y.plot(figsize=(15, 6))
plt.show()
from pylab import rcParams
rcParams['figure.figsize'] = 18, 8
decomposition = sm.tsa.seasonal_decompose(y, model='additive')
fig = decomposition.plot()
plt.show()
p = d = q = range(0, 2)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]
print('Examples of parameter combinations for Seasonal ARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))
for param in pdq:
for param_seasonal in seasonal_pdq:
        try:
        mod = sm.tsa.statespace.SARIMAX(y,
                                        order=param,
                                        seasonal_order=param_seasonal,
                                        enforce_stationarity=False,
                                        enforce_invertibility=False)
```

```
            results = mod.fit()
            print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic))
            except:
            continue
            mod = sm.tsa.statespace.SARIMAX(y,
                                            order=(1, 1, 1),
                                            seasonal_order=(1, 1, 0, 12),
                                            enforce_stationarity=False,
                                            enforce_invertibility=False)
results = mod.fit()
print(results.summary().tables[1])
results.plot_diagnostics(figsize=(16, 8))
plt.show()pred_uc = results.get_forecast(steps=100)
pred_ci = pred_uc.conf_int()
ax = y.plot(label='observed', figsize=(14, 7))
pred_uc.predicted_mean.plot(ax=ax, label='Forecast')
ax.fill_between(pred_ci.index,
                pred_ci.iloc[:, 0],
                pred_ci.iloc[:, 1], color='k', alpha=.25)
ax.set_xlabel('Date')
ax.set_ylabel('Furniture Sales')
plt.legend()
plt.show()
furniture = df.loc[df['Category'] == 'Furniture']
office = df.loc[df['Category'] == 'Office Supplies']
furniture.shape, office.shape
cols = ['Row ID', 'Order ID', 'Ship Date', 'Ship Mode', 'Customer ID', 'Customer Name',
'Segment', 'Country', 'City', 'State', 'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-
Category', 'Product Name', 'Quantity', 'Discount', 'Profit']
furniture.drop(cols, axis=1, inplace=True)
```

```
office.drop(cols, axis=1, inplace=True)

furniture = furniture.sort_values('Order Date')

office = office.sort_values('Order Date')

furniture = furniture.groupby('Order Date')['Sales'].sum().reset_index()

office = office.groupby('Order Date')['Sales'].sum().reset_index()

furniture = furniture.set_index('Order Date')

office = office.set_index('Order Date')

y_furniture = furniture['Sales'].resample('MS').mean()

y_office = office['Sales'].resample('MS').mean()

furniture = pd.DataFrame({'Order Date':y_furniture.index, 'Sales':y_furniture.values})

office = pd.DataFrame({'Order Date': y_office.index, 'Sales': y_office.values})

store = furniture.merge(office, how='inner', on='Order Date')

store.rename(columns={'Sales_x': 'furniture_sales', 'Sales_y': 'office_sales'},
inplace=True)

store.head()

plt.figure(figsize=(20, 8))

plt.plot(store['Order Date'], store['furniture_sales'], 'b-', label = 'furniture')

plt.plot(store['Order Date'], store['office_sales'], 'r-', label = 'office supplies')

plt.xlabel('Date'); plt.ylabel('Sales'); plt.title('Sales of Furniture and Office Supplies')

plt.legend();

first_date = store.ix[np.min(list(np.where(store['office_sales'] >
store['furniture_sales'])[0])), 'Order Date']

print("Office supplies first time produced higher sales than furniture is
{}.".format(first_date.date()))
```

SCREENSHOTS :



Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Na	Segment	Country	City	State	Postal Code	Region	Product ID
1	CA-2016-152156	08-11-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FUR-BO-1000
2	CA-2016-152156	08-11-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FUR-CH-1000
3	CA-2016-138688	12-06-2016	16-06-2016	Second Class	DV-13045	Darrin Van H	Corporate	United States	Los Angeles	California	90036	West	OFF-LA-1000
4	US-2015-108966	11-10-2015	18-10-2015	Standard Cla	SO-20335	Sean O'Donn	Consumer	United States	Fort Lauderda	Florida	33311	South	FUR-TA-1000
5	US-2015-108966	11-10-2015	18-10-2015	Standard Cla	SO-20335	Sean O'Donn	Consumer	United States	Fort Lauderda	Florida	33311	South	OFF-ST-1000
6	CA-2014-115812	09-06-2014	14-06-2014	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	FUR-FU-1000
7	CA-2014-115812	09-06-2014	14-06-2014	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	OFF-AR-1000
8	CA-2014-115812	09-06-2014	14-06-2014	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	TEC-PH-1000
9	CA-2014-115812	09-06-2014	14-06-2014	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	OFF-BI-1000
10	CA-2014-115812	09-06-2014	14-06-2014	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	OFF-AP-1000
11	CA-2014-115812	09-06-2014	14-06-2014	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	FUR-TA-1000
12	CA-2014-115812	09-06-2014	14-06-2014	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	TEC-PH-1000
13	CA-2017-114412	15-04-2017	20-04-2017	Standard Cla	AA-10480	Andrew Allen	Consumer	United States	Concord	North Carolin	28027	South	OFF-PA-1000
14	CA-2016-161389	05-12-2016	10-12-2016	Standard Cla	IM-15070	Irene Maddo	Consumer	United States	Seattle	Washington	98103	West	OFF-BI-1000
15	US-2015-118983	22-11-2015	26-11-2015	Standard Cla	HP-14815	Harold Pawla	Home Office	United States	Fort Worth	Texas	76106	Central	OFF-AP-1000
16	US-2015-118983	22-11-2015	26-11-2015	Standard Cla	HP-14815	Harold Pawla	Home Office	United States	Fort Worth	Texas	76106	Central	OFF-BI-1000
17	CA-2014-105893	11-11-2014	18-11-2014	Standard Cla	PK-19075	Pete Kriz	Consumer	United States	Madison	Wisconsin	53711	Central	OFF-ST-1000
18	CA-2014-167164	13-05-2014	15-05-2014	Second Class	AG-10270	Alejandro Gr	Consumer	United States	West Jordan	Utah	84084	West	OFF-ST-1000
19	CA-2014-143336	27-08-2014	01-09-2014	Second Class	ZD-21925	Zuschuss Do	Consumer	United States	San Francisc	California	94109	West	OFF-AR-1000
20	CA-2014-143336	27-08-2014	01-09-2014	Second Class	ZD-21925	Zuschuss Do	Consumer	United States	San Francisc	California	94109	West	TEC-PH-1000

Figure 1: Dataset of Superstore Sales.

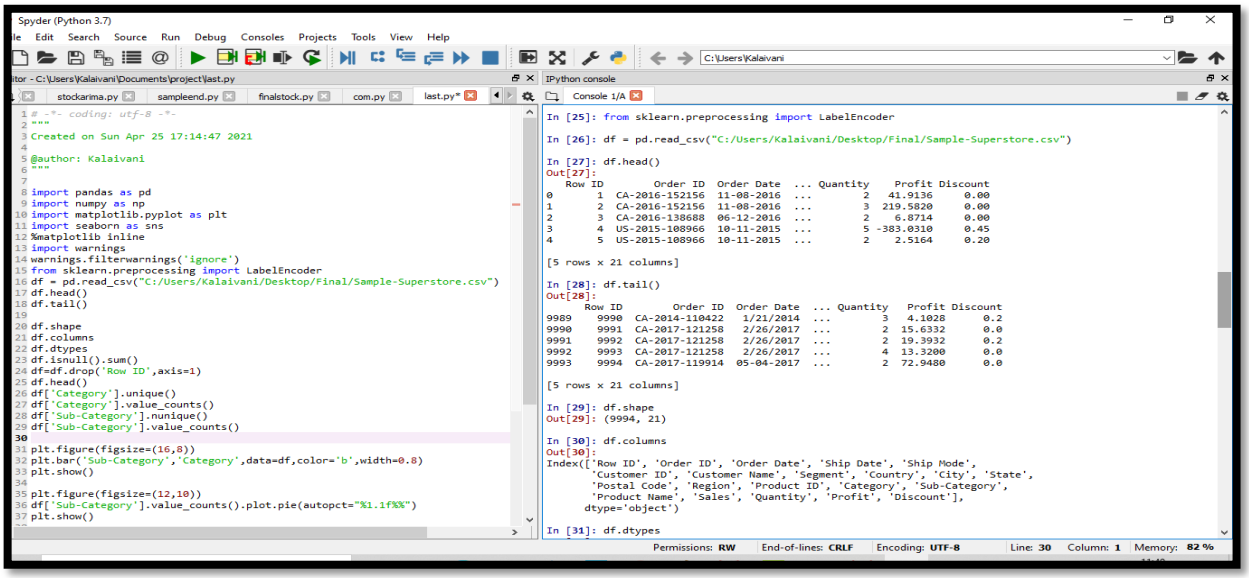


Figure 2: Step of Pre-processing**.**

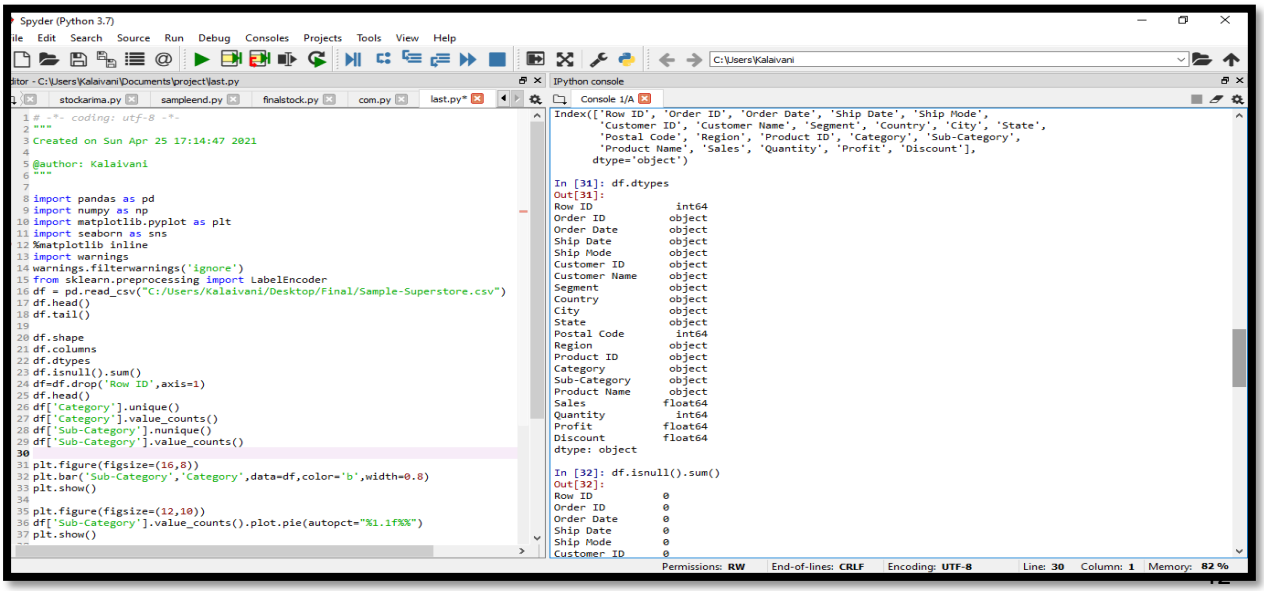


Figure 3: Checks is there null value.

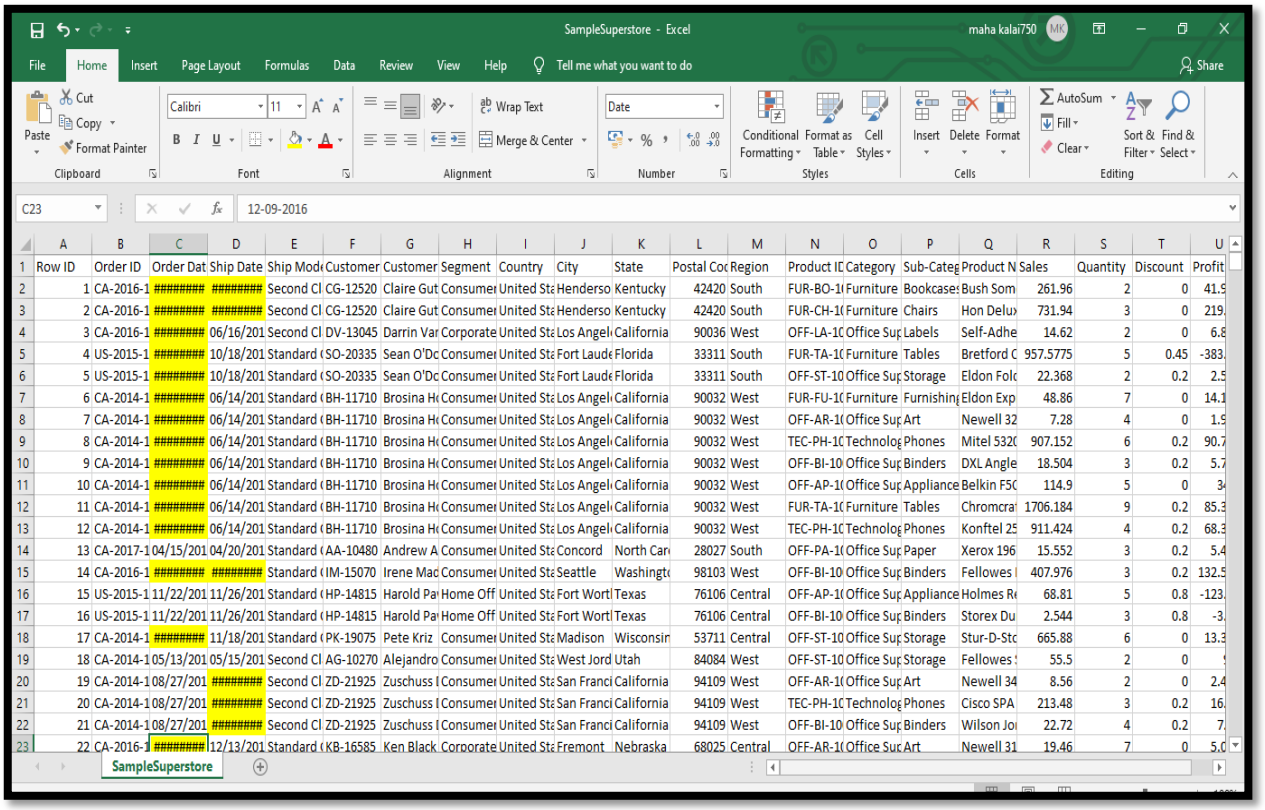


Figure 4: Before Data Cleaning

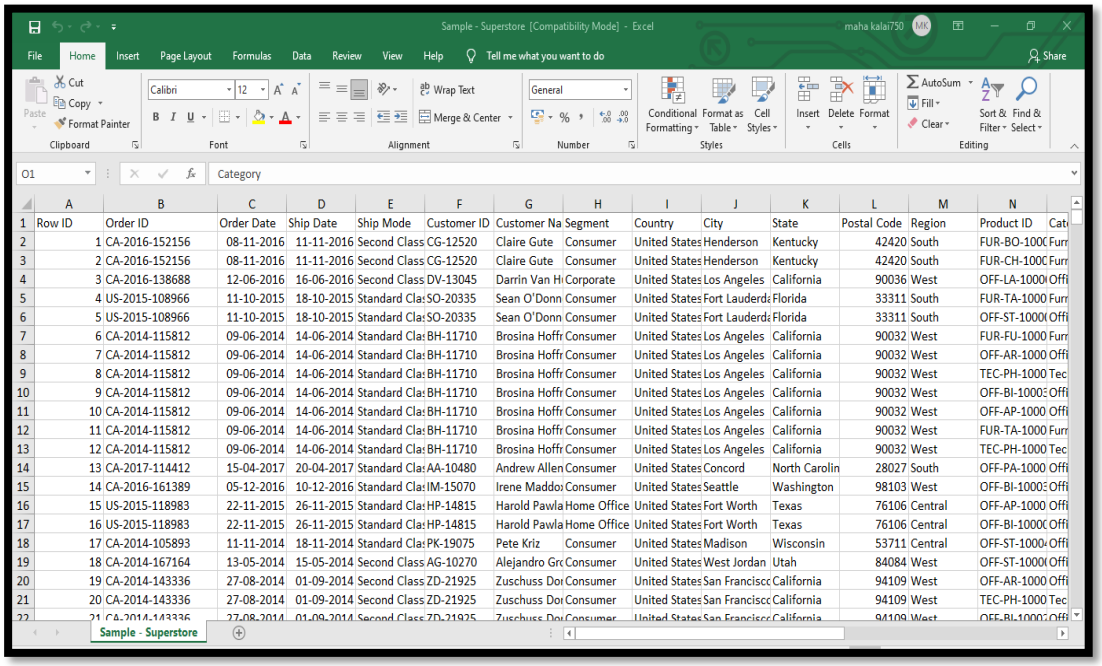


Figure 5: After Data Cleaning

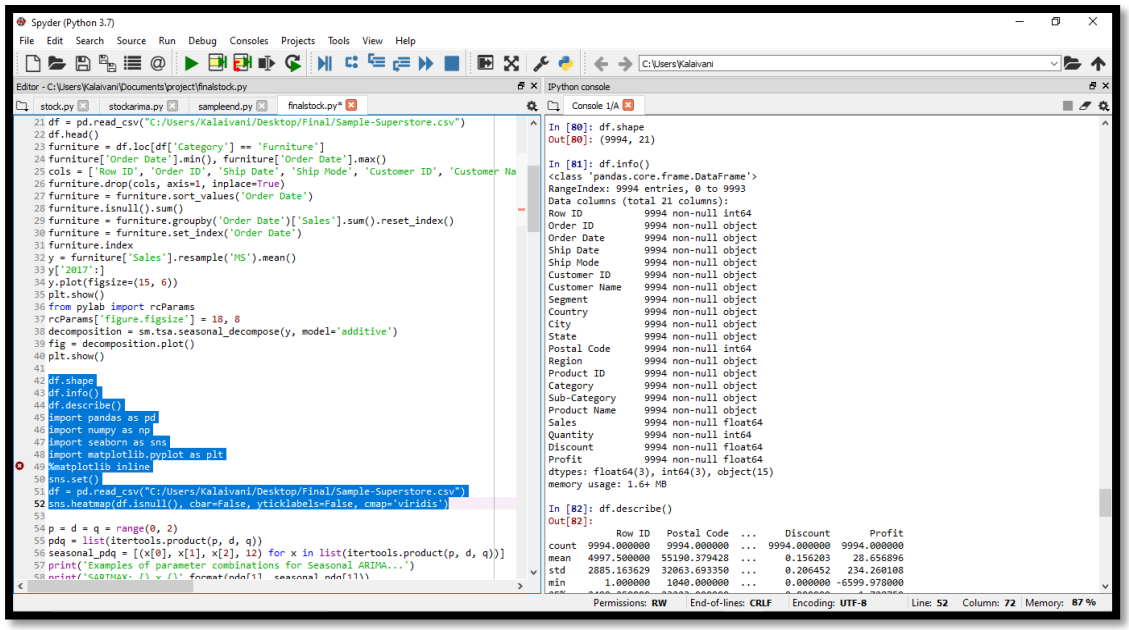


Figure 6: Exploratory Data Analysis

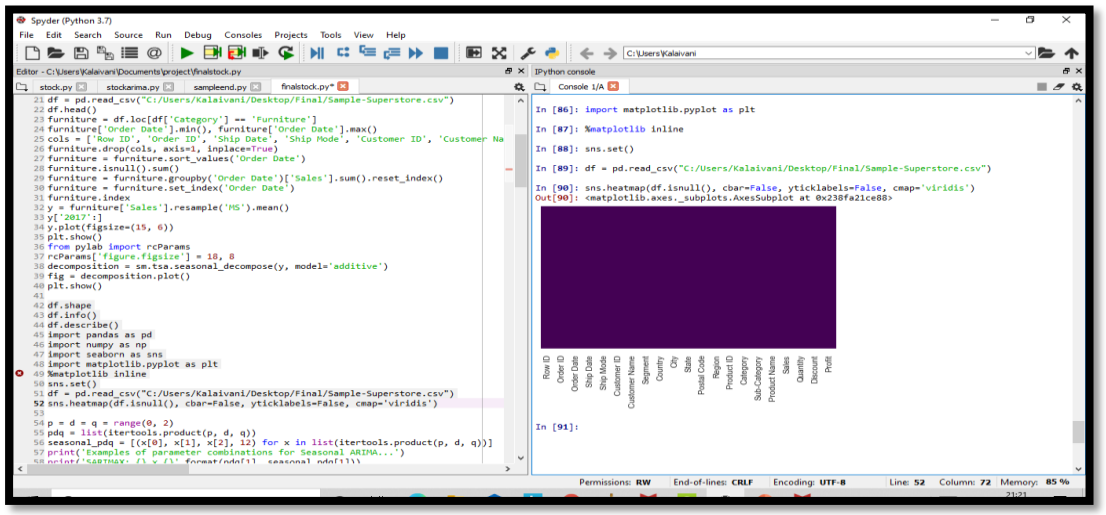
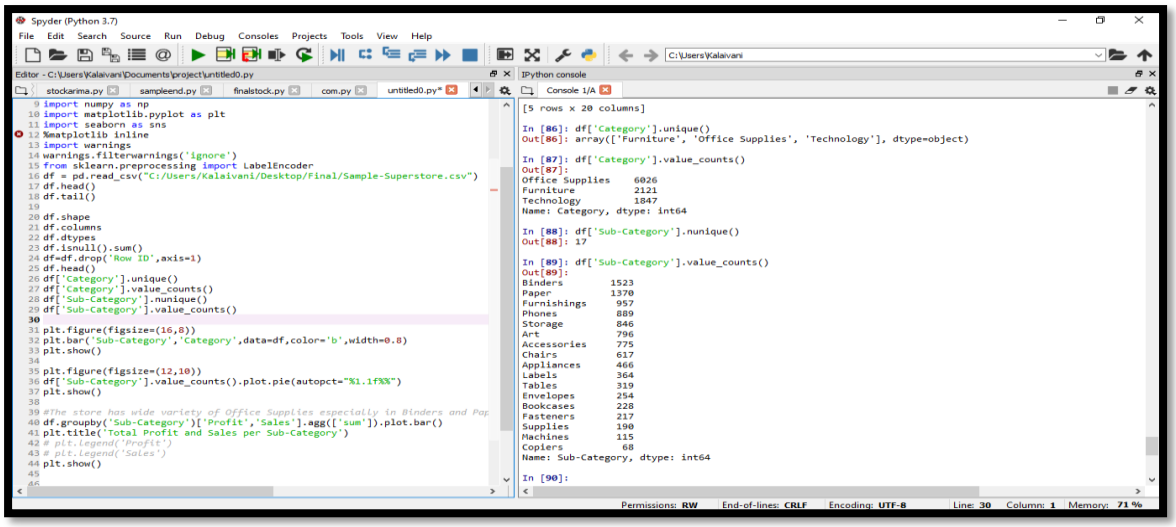


Figure 7: Missing values



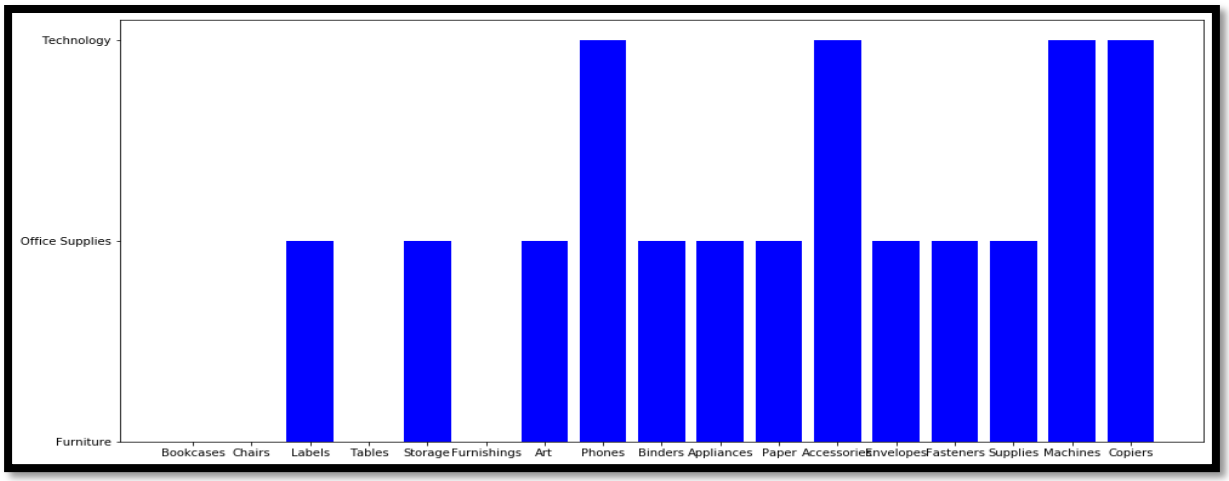


Figure 8: Shows the highest sales and graph represents the data.

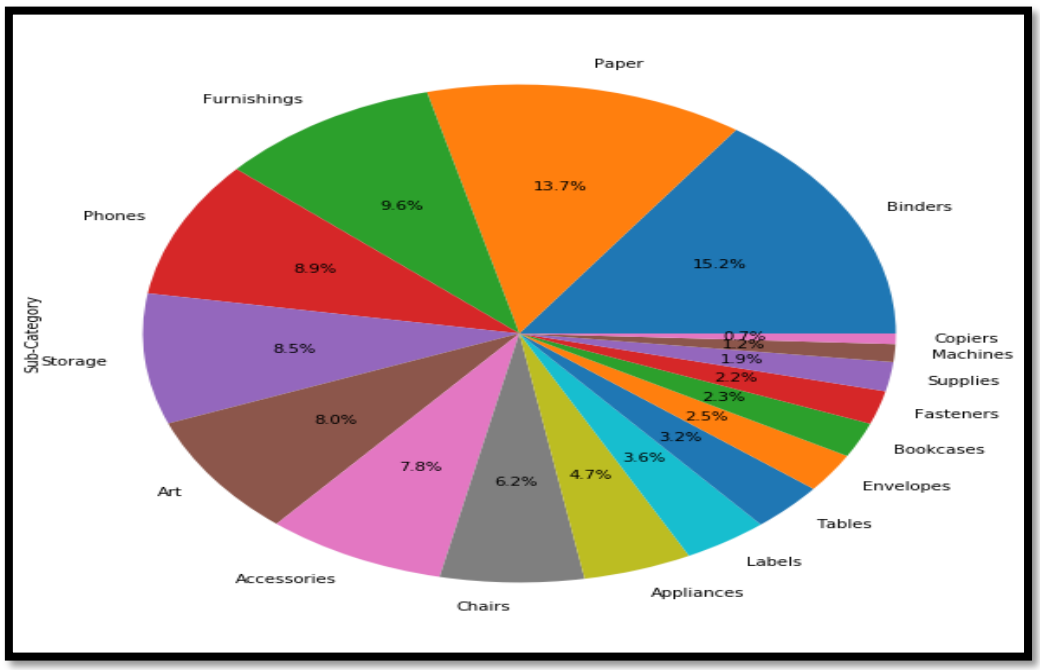


Figure 9: Presents the graphical method of sub-category.

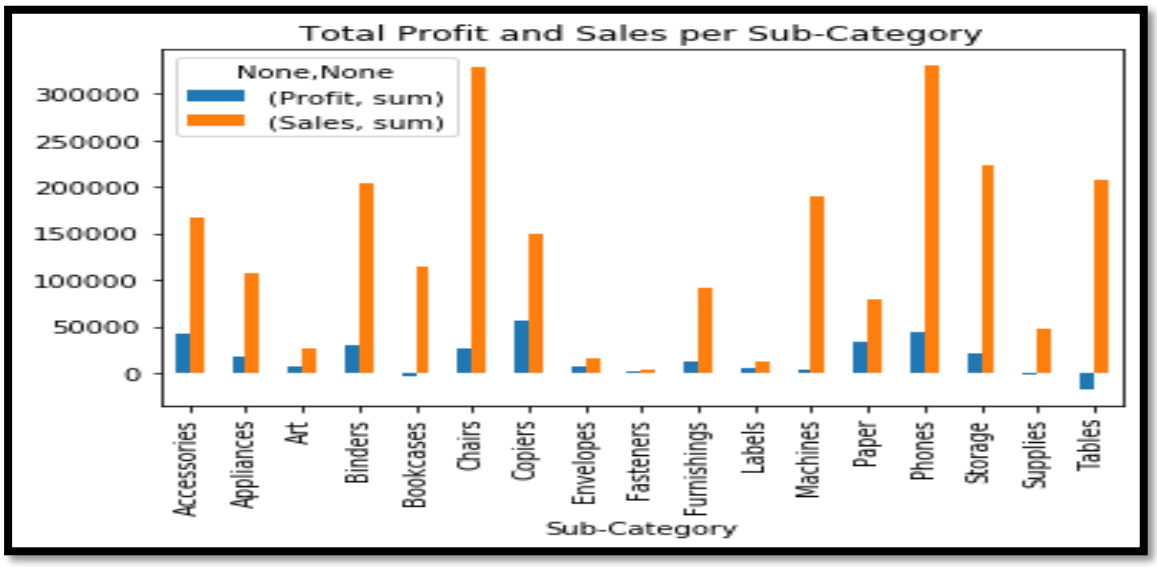
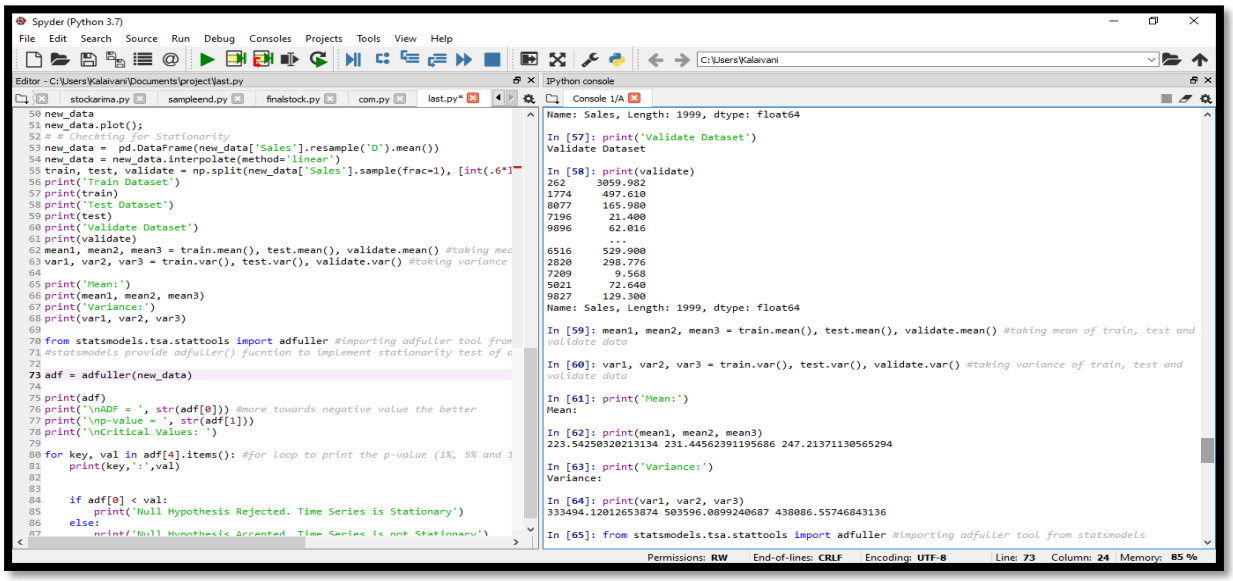


Figure 10: Total Profit and sales per sub-category.



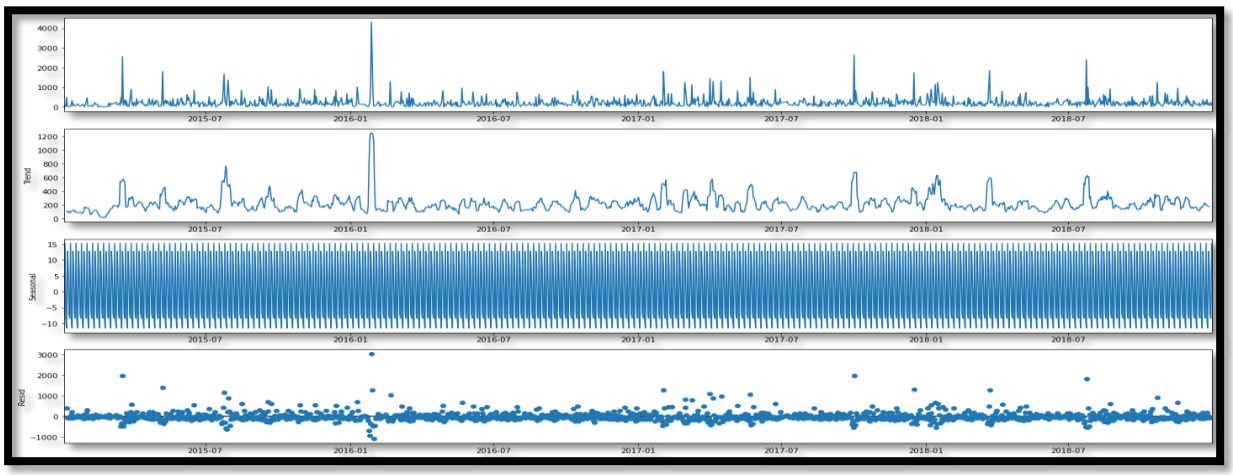


Figure 11: Rolling statistics of Sales Dataset that provides that dataset is stationery.

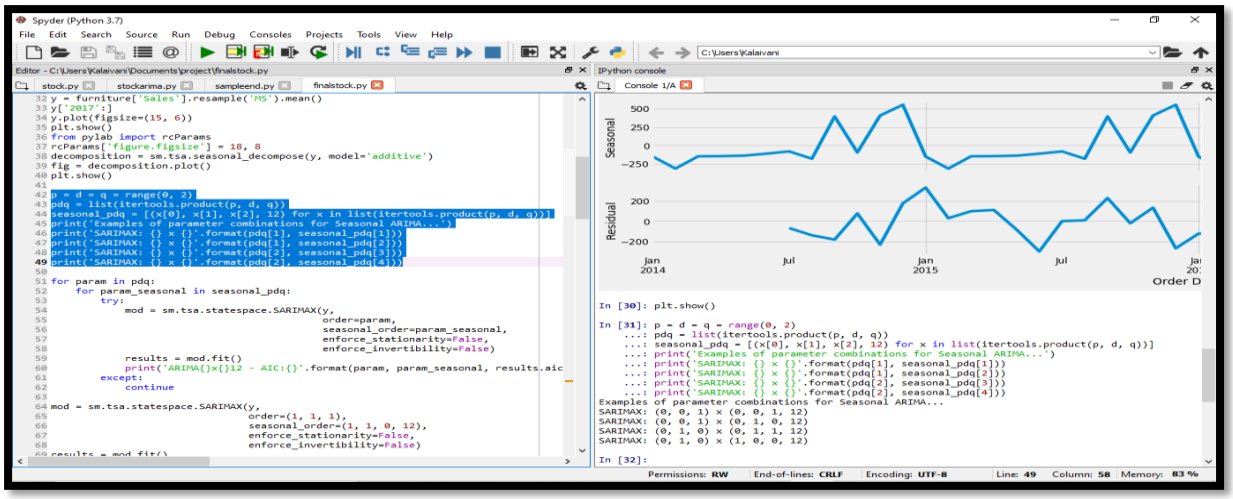
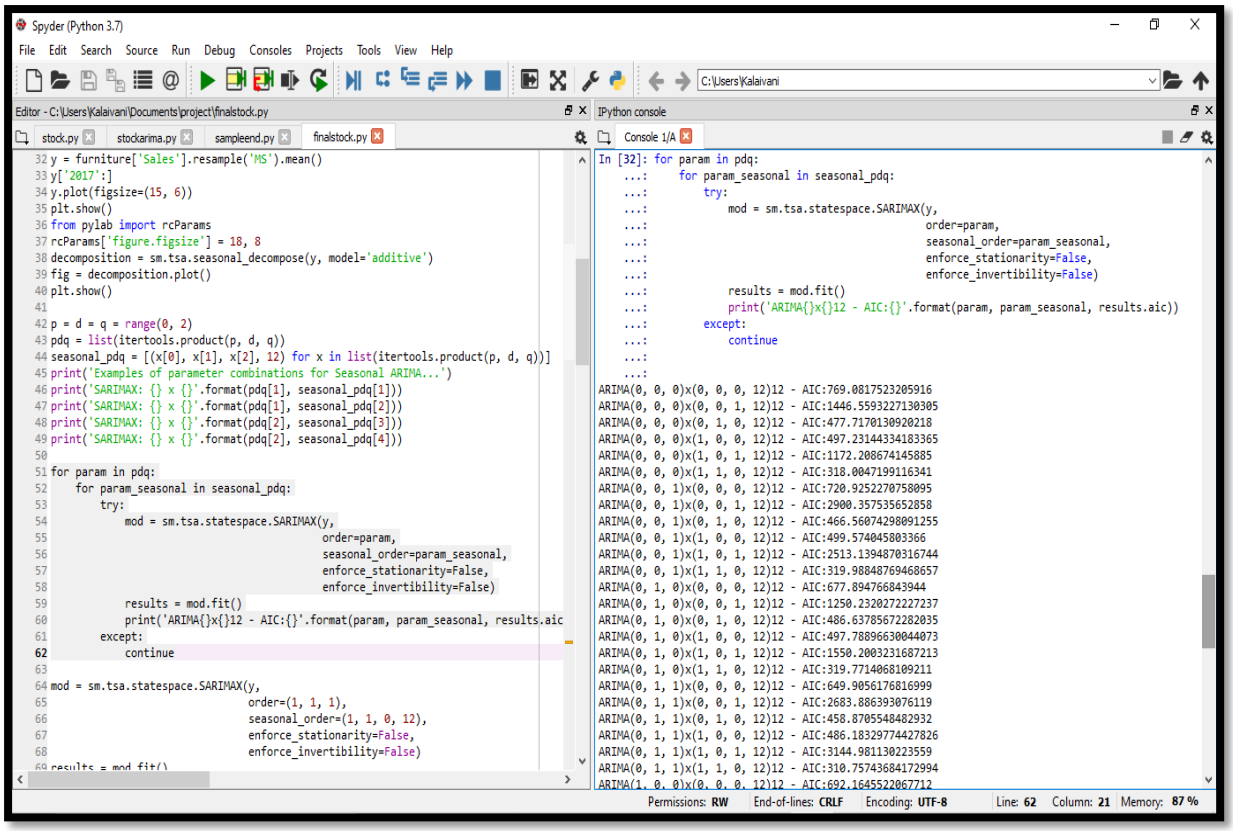


Figure 12: Parameter Selection for the ARIMA Time Series Model



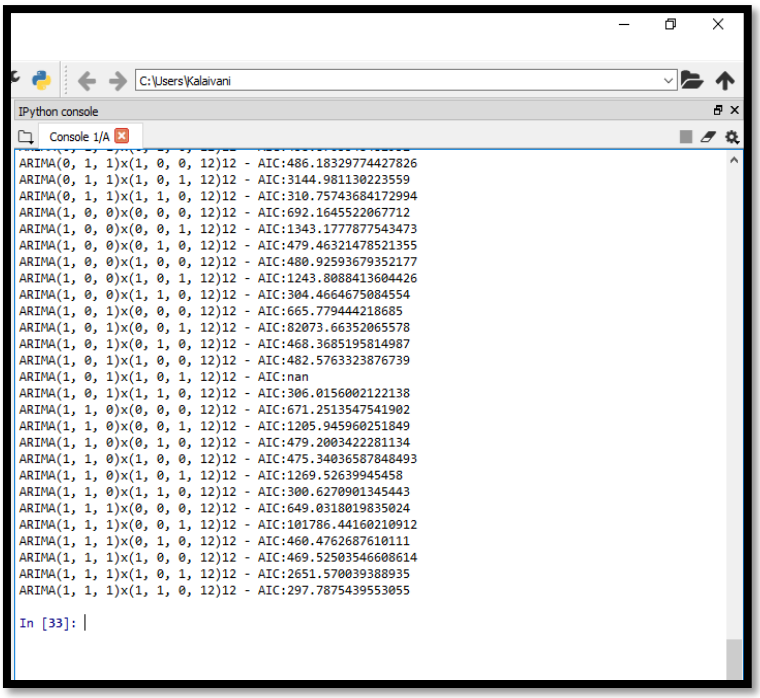
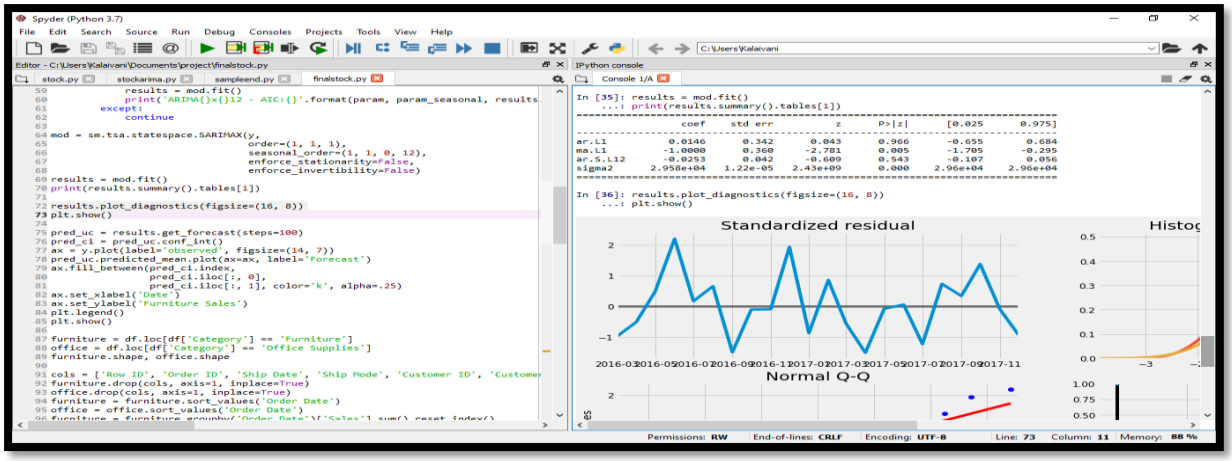


Figure 13: Result of Time series forecasting with SARIMAX (1, 1, 1)x(1, 1, 1, 12) yields the lowest AIC value of 277.78.



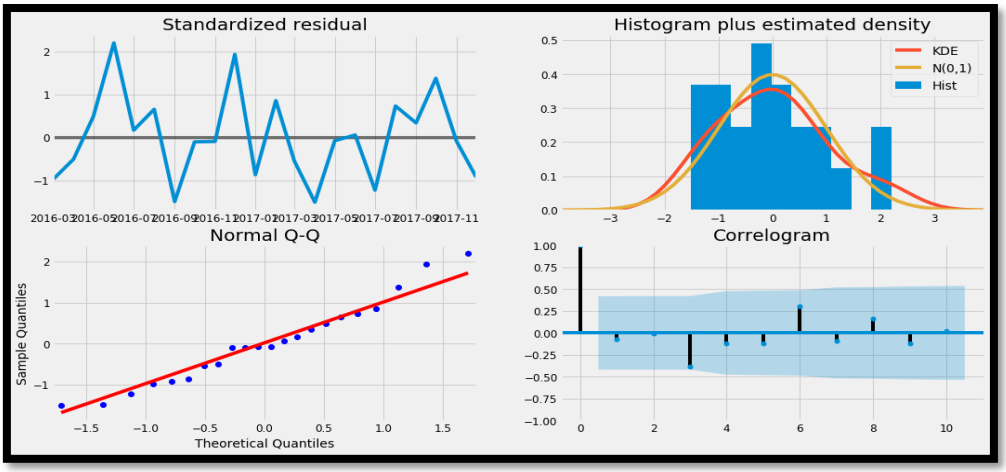


Figure 14: Fitting ARIMA Model provides Standardized Residual and Histogram density

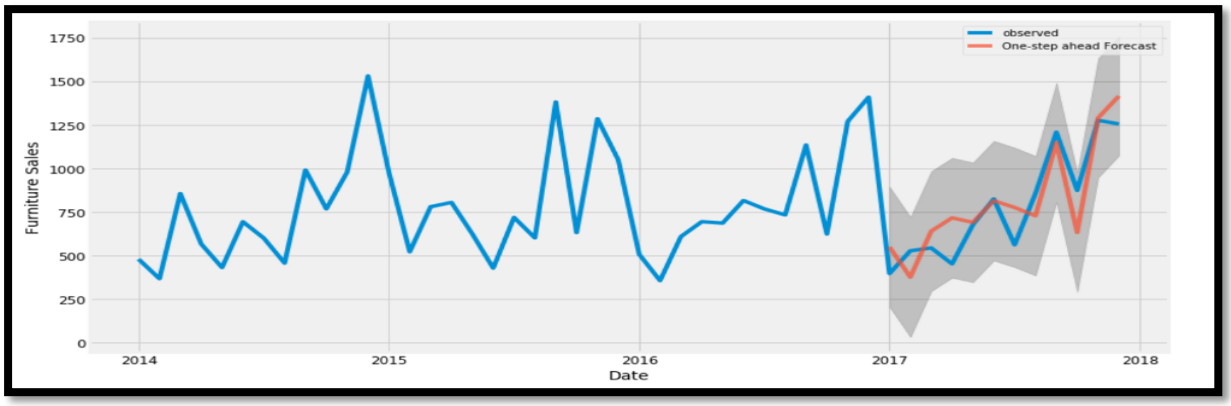
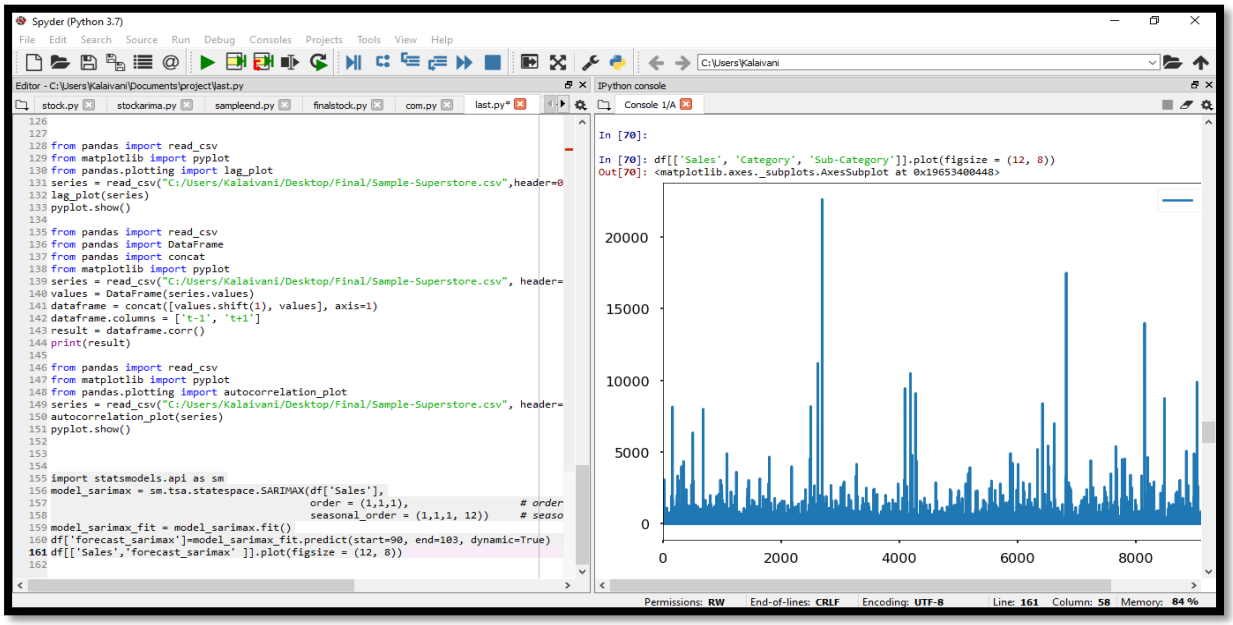


Figure 15: Visualization of forecasts



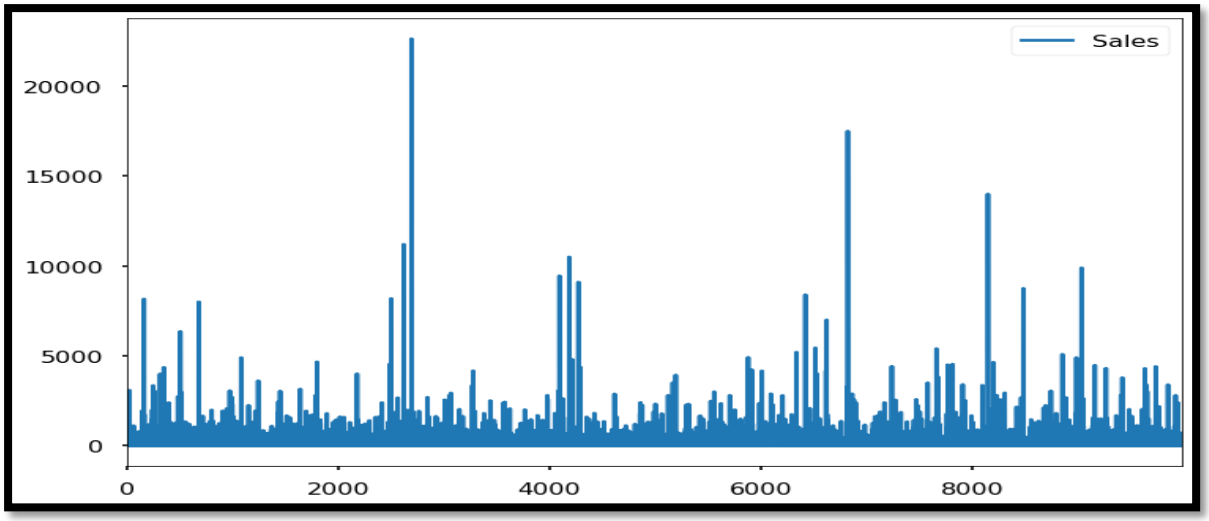


Figure 16: Visualization of SARIMAX

Figure 17: Producing and visualizing forecasts

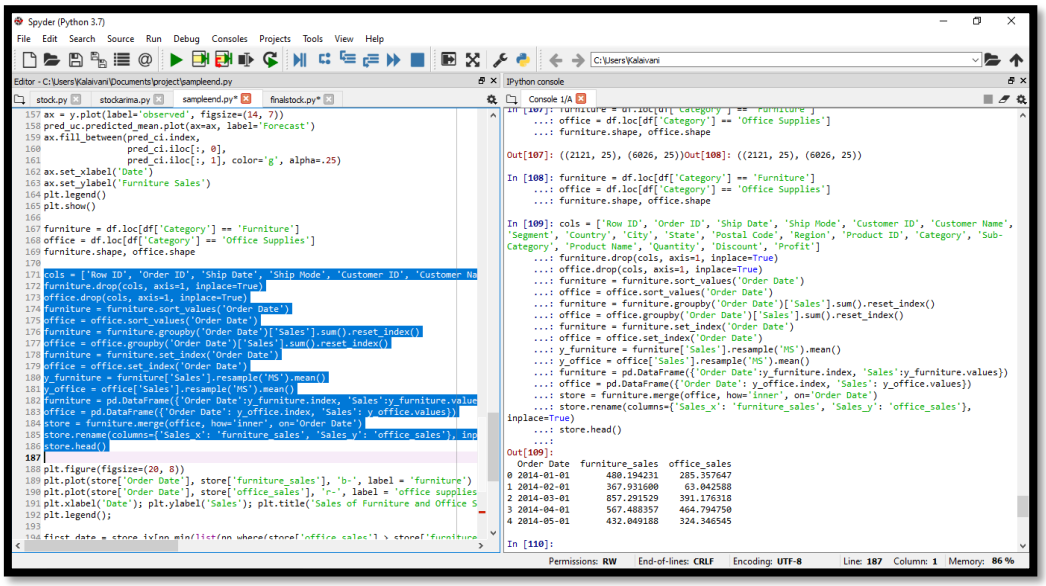


Figure 18: Time series of furniture vs. Office supplies (Data Exploration)

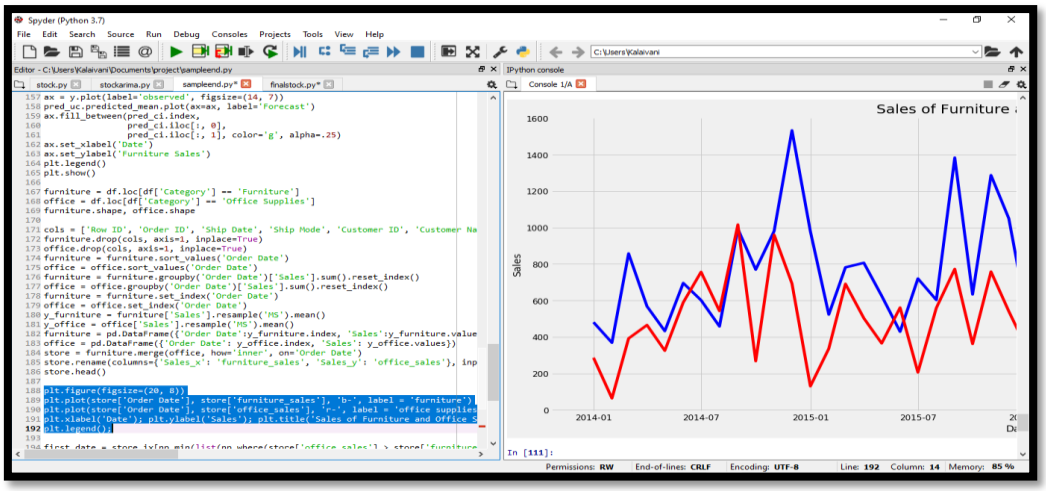


Figure 19: Sales of Furniture and Office supplies

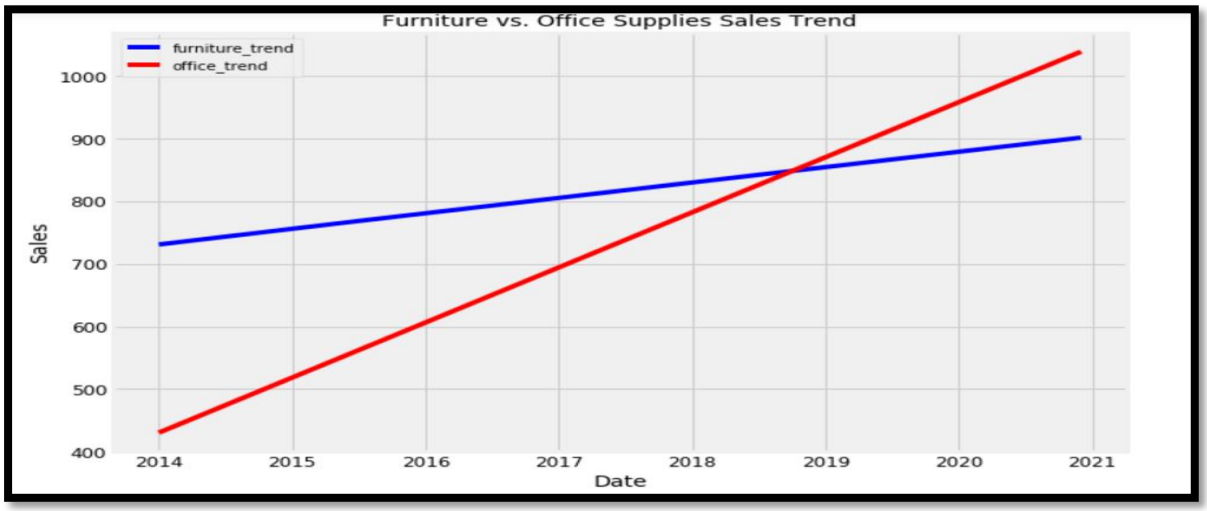


Figure 20: Sales Trend of Furniture vs Office Supplies

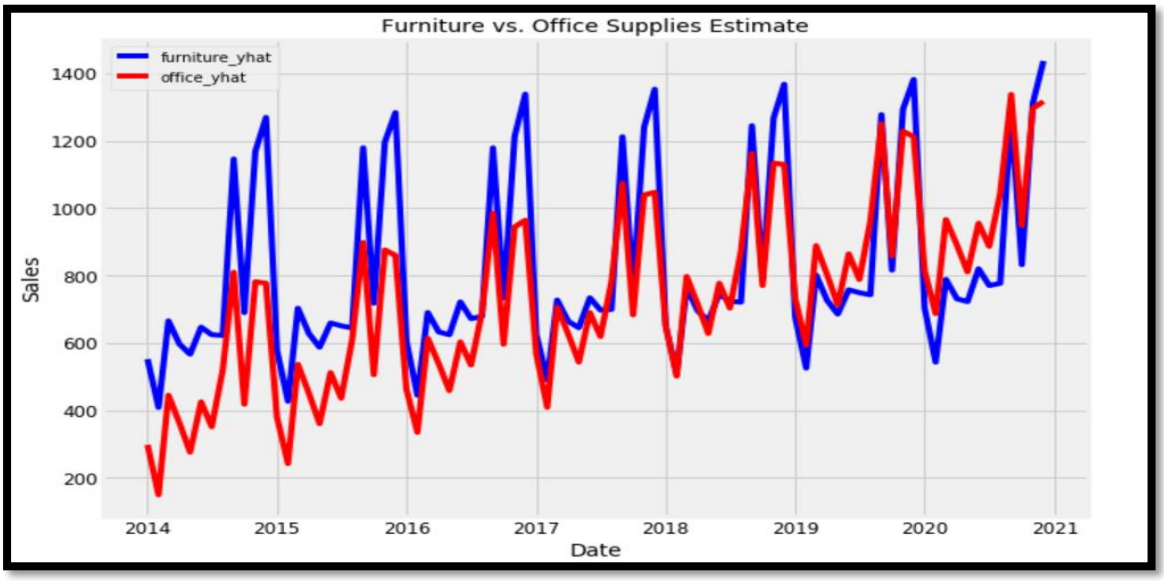


Figure 21: Trend And Forecast Visualization of Furniture and Office supplies estimation.

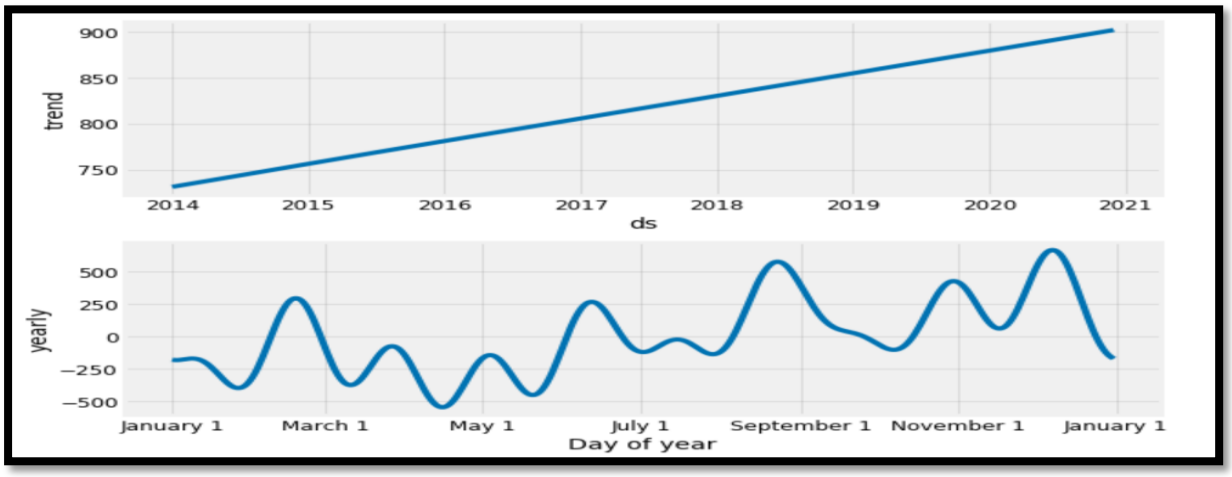


Figure 22: Shows the Trends and Patterns.

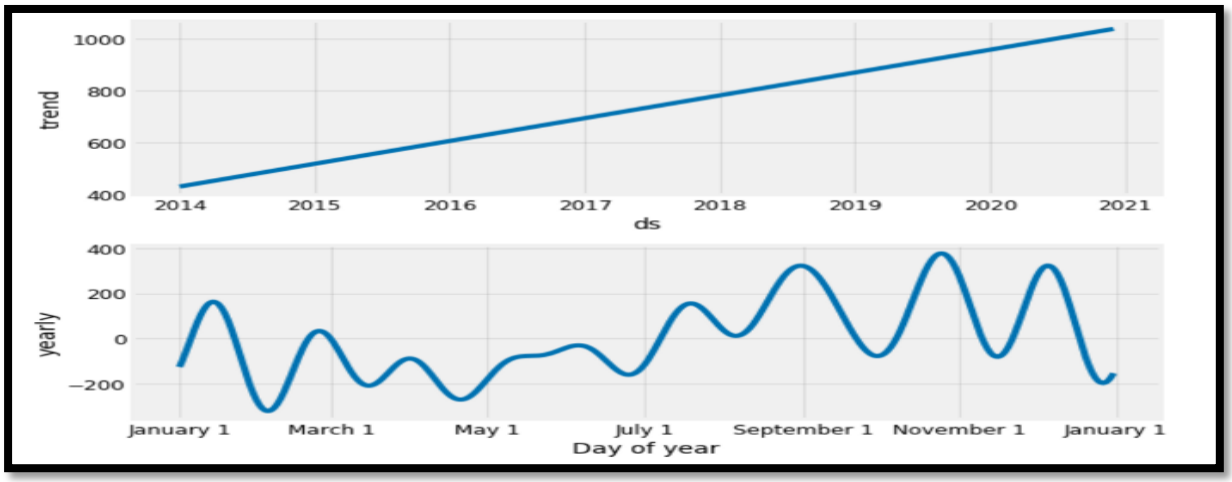


Figure 23: Final Result of sales forecasting.