"Grocery Demand Prediction"

A Major Project Report
Submitted in the Partial Fulfillment

of

the Requirements for the Degree of
Bachelors in information technology Engineering

at

Everest Engineering College
Sanepa, Lalitpur
Affiliated to Pokhara University

By

Dipesh Magar	**(19120054)**
Jhak Prasad Pun	**(19120056)**
Manish Rai	**(19120060)**
Milan Bhattarai	**(19120063)**

July,2023

DECLARATION

We hereby declare that the report of the project entitled "**Grocery Demands Predictions**" which is being submitted to the Department of Computer and Information Technology Engineering, Everest Engineering College, Sanepa, in the partial fulfillment of the requirements for a ward of the Degree of Bachelor of Engineering in Information Technology Engineering, is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this document.

Dipesh Magar(19120054) ________________

Jhak Prasad Pun(19120056) ________________

Manish Rai(19120060) ________________

Milan Bhattarai(19120063) ________________

CERTIFICATE OF APPROVAL

The project report entitled "**Grocery Demands Predictions**", submitted by Dipesh Magar, Jhak Prasad Pun, Manish Rai and Milan Bhattarai in partial fulfillment of the requirement for the Bachelor's degree in information technology engineering has been accepted as a bonafide record of work independently carried out by the group in the department.

…………………………………….

Name of the External Examiner

Designation

Affiliation

…………………………………………

Santa Bahadur Basnet

Project Supervisor

Department of Computer and IT Engineering

…………………………………….

Nischal Regmi

Project Coordinator

Department of Computer and IT Engineering

…………………………………….

Anuj Ghimire

Head of Department

Department of Computer and IT Engineering

COPYRIGHT

ACKNOWLEDGEMENT

This project would not have been possible without the joint efforts of many individuals. It has been a pleasure for us to acknowledge the assistance and contributions that were very important and supportive throughout the project. We would like to thank them sincerely.

We are indebted to our project supervisor, **Santa Basnet** for his guidance, support, and follow-ups throughout the project development period which helped our project to grow and foster to a level we didn't think of reaching in such a short period of time. We also express our thanks to our Associate Prof. and project coordinator **Er. Nischal Regmi**, Department of Computer & IT, Everest Engineering College for his encouragement and guidance throughout the project.

Last, but not least we would like to thank our teachers, seniors, colleagues, and family members who have been knowingly or unknowingly part of our project with their views and instructions during the entire development time.

Contents

LIST OF FIGURES

LIST OF TABLE

Chapter 1: Introduction

1.1 Introduction

The grocery industry is constantly evolving and it is becoming increasingly important for retailers to accurately predict demand for products in order to optimize their inventory, reduce waste, and improve customer satisfaction. There has been an increasing demand for Ecommerce sites in the past decade. Online grocers, an offline grocery especially, have increased in popularity. The aim of this report is to create a "GROCERY DEMAND PREDICTION" that needs to be delivered to which customer and come up relevant predictions for the user and display the patterns for some business. This model predicts the grocery items to find which item is most demanded by customers or which item is less in demand. Focusing on grocery retailing has been increasing rapidly especially in developing countries like; Nepal due to the supermarket radical changes. For this project, we will be using different machine learning models, linear Regression Model, Arima Model, Naive Regression and Random forest for supervised learning, classification and regression.

Linear regression is a type of statistical analysis used to predict the relationship between two variables. It assumes a linear relationship between the independent variable and the dependent variable and aims to find the best-fitting line that describes the relationship. The line is determined by minimizing the sum of the squared differences between the predicted values and the actual values. In a simple linear regression, there is one independent variable and one dependent variable. The model estimates the slope and intercept of the line of best fit, which represents the relationship between the variables. The slope represents the change in the dependent variable for each unit change in the independent variable, while the intercept represents the predicted value of the dependent variable when the independent variable is zero.Linear regression is a quiet and the simplest statistical regression method used for predictive analysis in machine learning. Linear regression shows the linear relationship between the independent(predictor) variable i.e. X-axis and the dependent(output) variable i.e. Y-axis, called linear regression. If there is a single input variable X(independent variable), such linear regression is called simple linear regression.

To calculate best-fit line linear regression uses a traditional slope-intercept form which is given below,

$(Y_i = \beta_0 + \beta_1 X_i)$

where Y_i = Dependent variable, β_0 = constant/Intercept, β_1 = Slope/Intercept, X_i = Independent variable. This algorithm explains the linear relationship between the dependent(output) variable y and the independent(predictor) variable X using a straight-line $Y= B_0 + B_1 X$.

Naive regression refers to a basic form of regression analysis that assumes a simplistic relationship between a dependent variable and one or more independent variables. It is called "naive" because it overlooks important considerations and assumptions that are typically made in more sophisticated regression models. In naive regression, the relationship between the dependent variable and independent variable(s) is assumed to be linear, with no consideration for other factors that may influence the relationship. This means that the model does not account for things like outliers, non-linear relationships, multicollinearity, heteroscedasticity, or autocorrelation, among other potential issues.

Exponential Smoothing is a popular time series forecasting method used to make predictions based on past observations, giving more weight to recent data points while gradually reducing the importance of older observations. The basic idea behind exponential smoothing is to calculate a weighted average of past data, and the weights decrease exponentially as the data gets older.

An autoregressive integrated moving average, or ARIMA, is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends A statistical model is autoregressive if it predicts future values based on past values. For example, an ARIMA model might seek to predict a stock's future prices based on its past performance or forecast a company's earnings based on past periods.

1.2 Motivation:

This project helps us to analyze Business growth by knowing the future value of quantity and price. It will help in understanding the role of strategy for Business Ideas, Also analyzes to calculate the Business Revenue.

1.3 Objectives:

- Forecasting the future average value of quantity and price per month.
- Predicting next month's quantity, which product is needed in what amount.

1.4 Project Applications and Scope:

- This model is used to predict what customer demand will be for a product and price.

Chapter 2: Literature Review

“Predicting sales in a food store department using machine learning” (ROBERT SIWERZ and CHRISTOPHER DAHLEN) [1].In this paper they deal with the SVM (Support Vector Machines) they likewise utilize two more calculation is Multilayer Perceptron and Radial Basis Capacity Networks. There is a measurably huge distinction between the SVM, MLP and RFBN while foreseeing the deals in a food store division. The SVM performed lower mistake measures than the other two techniques. Since this investigation utilized constrained information, hence, one could scarcely make the determination that the SVM is consistently the most exact technique to use for deal expectation in a food store office.

“Predicting Online Grocery Ordering Intention” (Rohit Rathish and Yash Jahagirdar) [2]. In this paper they work on E-Commerce, Grocery, Recommendation System, Neural Networks. They mainly work on the neural network. The recommendation System which has been proposed, is based upon neural networks and is able to provide suitable predictions about the future orders of the customer.

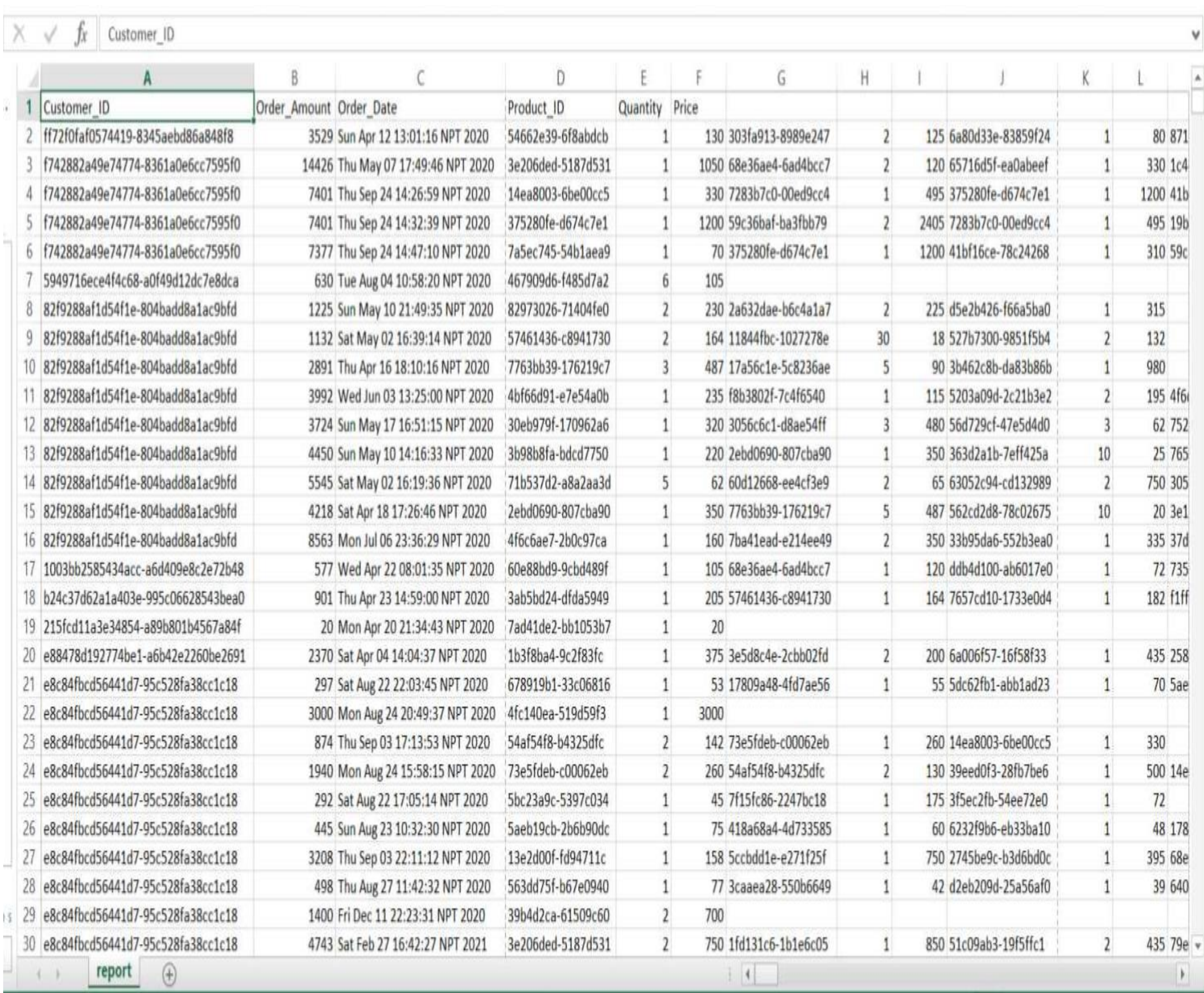
"Applied Machine Learning for Supermarket Sales Prediction"[3]. Sales forecasting is very crucial for every company, especially big ones and this process is very complex because there are lots of factors that should be taken into consideration. In order to implement achievable goals and successfully implement them, supermarket chains always want to forecast sales. In this paper, they used three machine learning algorithms (K-Nearest Neighbor, Random Forest and Gradient Boosting) for sales forecasting, RF performed better, as it had a lower mean absolute error than the other two models.

"A Review on Grocery Management System Using Machine Learning Algorithms"[4].The main goal of this paper is Analyzing the proper requirement of the consumer and suggesting him products accordingly. By comparing Apriori and Max-Miner Algorithm , Max-Miner Algorithm showed more accuracy than Apriori Algorithm. In this paper, we found that Random forest and support vector machine algorithms can be used to classify the products based on purchased product and not purchased product in the grocery list. Comparing the algorithm Support vector machine showed more accuracy than random forest algorithm. This algorithm is used to build a system.

Chapter 3: Methodology

3.1 Data acquisition:

At first, we collected the data from some grocery stores. We obtained nearly 40k data in the csv file. From that data we analyzed that data are in unmanaged form. In that dataset there are nearly 40k rows and nearly 350 columns.



Customer_ID

	A	B	C	D	E	F	G	H	I	J	K	L	
1	Customer_ID	Order_Amount	Order_Date	Product_ID	Quantity	Price							
2	ff72f0faf0574419-8345aebd86a848f8	3529	Sun Apr 12 13:01:16 NPT 2020	54662e39-6f8abdcb	1	130	303fa913-8989e247	2	125	6a80d33e-83859f24	1	80	871
3	f742882a49e74774-8361a0e6cc7595f0	14426	Thu May 07 17:49:46 NPT 2020	3e206ded-5187d531	1	1050	68e36ae4-6ad4bcc7	2	120	65716d5f-ea0abeef	1	330	1c4
4	f742882a49e74774-8361a0e6cc7595f0	7401	Thu Sep 24 14:26:59 NPT 2020	14ea8003-6be00cc5	1	330	7283b7c0-00ed9cc4	1	495	375280fe-d674c7e1	1	1200	41b
5	f742882a49e74774-8361a0e6cc7595f0	7401	Thu Sep 24 14:32:39 NPT 2020	375280fe-d674c7e1	1	1200	59c36baf-ba3fbb79	2	2405	7283b7c0-00ed9cc4	1	495	19b
6	f742882a49e74774-8361a0e6cc7595f0	7377	Thu Sep 24 14:47:10 NPT 2020	7a5ec745-54b1aea9	1	70	375280fe-d674c7e1	1	1200	41bf16ce-78c24268	1	310	59c
7	5949716ece4f4c68-a0f49d12dc7e8dca	630	Tue Aug 04 10:58:20 NPT 2020	467909d6-f485d7a2	6	105							
8	82f9288af1d54f1e-804badd8a1ac9bfd	1225	Sun May 10 21:49:35 NPT 2020	82973026-71404fe0	2	230	2a632dae-b6c4a1a7	2	225	d5e2b426-f66a5ba0	1	315	
9	82f9288af1d54f1e-804badd8a1ac9bfd	1132	Sat May 02 16:39:14 NPT 2020	57461436-c8941730	2	164	11844fbc-1027278e	30	18	527b7300-9851f5b4	2	132	
10	82f9288af1d54f1e-804badd8a1ac9bfd	2891	Thu Apr 16 18:10:16 NPT 2020	7763bb39-176219c7	3	487	17a56c1e-5c8236ae	5	90	3b462c8b-da83b86b	1	980	
11	82f9288af1d54f1e-804badd8a1ac9bfd	3992	Wed Jun 03 13:25:00 NPT 2020	4bf66d91-e7e54a0b	1	235	f8b3802f-7c4f6540	1	115	5203a09d-2c21b3e2	2	195	4f6
12	82f9288af1d54f1e-804badd8a1ac9bfd	3724	Sun May 17 16:51:15 NPT 2020	30eb979f-170962a6	1	320	3056c6c1-d8ae54ff	3	480	56d729cf-47e5d4d0	3	62	752
13	82f9288af1d54f1e-804badd8a1ac9bfd	4450	Sun May 10 14:16:33 NPT 2020	3b98b8fa-bdcd7750	1	220	2ebd0690-807cba90	1	350	363d2a1b-7eff425a	10	25	765
14	82f9288af1d54f1e-804badd8a1ac9bfd	5545	Sat May 02 16:19:36 NPT 2020	71b537d2-a8a2aa3d	5	62	60d12668-ee4cf3e9	2	65	63052c94-cd132989	2	750	305
15	82f9288af1d54f1e-804badd8a1ac9bfd	4218	Sat Apr 18 17:26:46 NPT 2020	2ebd0690-807cba90	1	350	7763bb39-176219c7	5	487	562cd2d8-78c02675	10	20	3e1
16	82f9288af1d54f1e-804badd8a1ac9bfd	8563	Mon Jul 06 23:36:29 NPT 2020	4f6c6ae7-2b0c97ca	1	160	7ba41ead-e214ee49	2	350	33b95da6-552b3ea0	1	335	37d
17	1003bb2585434acc-a6d409e8c2e72b48	577	Wed Apr 22 08:01:35 NPT 2020	60e88bd9-9cbd489f	1	105	68e36ae4-6ad4bcc7	1	120	ddb4d100-ab6017e0	1	72	735
18	b24c37d62a1a403e-995c06628543bea0	901	Thu Apr 23 14:59:00 NPT 2020	3ab5bd24-dfda5949	1	205	57461436-c8941730	1	164	7657cd10-1733e0d4	1	182	f1ff
19	215fcd11a3e34854-a89b801b4567a84f	20	Mon Apr 20 21:34:43 NPT 2020	7ad41de2-bb1053b7	1	20							
20	e88478d192774be1-a6b42e2260be2691	2370	Sat Apr 04 14:04:37 NPT 2020	1b3f8ba4-9c2f83fc	1	375	3e5d8c4e-2cbb02fd	2	200	6a006f57-16f58f33	1	435	258
21	e8c84fbcd56441d7-95c528fa38cc1c18	297	Sat Aug 22 22:03:45 NPT 2020	678919b1-33c06816	1	53	17809a48-4fd7ae56	1	55	5dc62fb1-abb1ad23	1	70	5ae
22	e8c84fbcd56441d7-95c528fa38cc1c18	3000	Mon Aug 24 20:49:37 NPT 2020	4fc140ea-519d59f3	1	3000							
23	e8c84fbcd56441d7-95c528fa38cc1c18	874	Thu Sep 03 17:13:53 NPT 2020	54af54f8-b4325dfc	2	142	73e5fdeb-c00062eb	1	260	14ea8003-6be00cc5	1	330	
24	e8c84fbcd56441d7-95c528fa38cc1c18	1940	Mon Aug 24 15:58:15 NPT 2020	73e5fdeb-c00062eb	2	260	54af54f8-b4325dfc	2	130	39eed0f3-28fb7be6	1	500	14e
25	e8c84fbcd56441d7-95c528fa38cc1c18	292	Sat Aug 22 17:05:14 NPT 2020	5bc23a9c-5397c034	1	45	7f15fc86-2247bc18	1	175	3f5ec2fb-54ee72e0	1	72	
26	e8c84fbcd56441d7-95c528fa38cc1c18	445	Sun Aug 23 10:32:30 NPT 2020	5aeb19cb-2b6b90dc	1	75	418a68a4-4d733585	1	60	6232f9b6-eb33ba10	1	48	178
27	e8c84fbcd56441d7-95c528fa38cc1c18	3208	Thu Sep 03 22:11:12 NPT 2020	13e2d00f-fd94711c	1	158	5ccbdd1e-e271f25f	1	750	2745be9c-b3d6bd0c	1	395	68e
28	e8c84fbcd56441d7-95c528fa38cc1c18	498	Thu Aug 27 11:42:32 NPT 2020	563dd75f-b67e0940	1	77	3caaea28-550b6649	1	42	d2eb209d-25a56af0	1	39	640
29	e8c84fbcd56441d7-95c528fa38cc1c18	1400	Fri Dec 11 22:23:31 NPT 2020	39b4d2ca-61509c60	2	700							
30	e8c84fbcd56441d7-95c528fa38cc1c18	4743	Sat Feb 27 16:42:27 NPT 2021	3e206ded-5187d531	2	750	1fd131c6-1b1e6c05	1	850	51c09ab3-19f5ffc1	2	435	79e

report

Table 1:Data set in CSV Format

3.2 Data Preprocessing:

After we obtained data, we analyzed data and then we sliced it into 5 chunks such as customer id, order amount, order date, product id, quantity and price. After that we obtained nearly 291k data.

```
In [11]: # collects all the bill items in the form of dictionary.
         orderBills = []
         a = 0
         for entry in dataContents:
             entries = entry.split(",")
             firstPart = [item.strip() for item in entries[0:3]]
             firstPart.pop(1) # Removes total quantity.
             secondPart = entries[3:]
             secondPart = [i for i in secondPart if len(i.strip()) > 0]
             sliced = list(divide_chunks(secondPart, 3))

             sliced = [bill_item for bill_item in sliced if len(bill_item) > 0]
             for orderQty in sliced:
                 if len(orderQty) == 3:
                     rowData = firstPart + orderQty
                     orderBills.append(rowData)

In [12]: len(orderBills)

Out[12]: 292160
```

Figure 1:Splitting Data into corresponding

Showing rows 0 - 24 (290079 total, Query took 0.0010 seconds.)

```
SELECT * FROM `bills`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> | Number of rows: 25 Filter rows: Search this table

Extra options

customer_id	order_date	product_id	quantity	price
ff72f0faf0574419-8345aebd86a848f8	2020-04-12 13:01:16	54662e39-6f8abdcb	1	130
ff72f0faf0574419-8345aebd86a848f8	2020-04-12 13:01:16	303fa913-8989e247	2	125
ff72f0faf0574419-8345aebd86a848f8	2020-04-12 13:01:16	6a80d33e-83859f24	1	80
ff72f0faf0574419-8345aebd86a848f8	2020-04-12 13:01:16	871c68b0-3c4bc1b0	2	820
ff72f0faf0574419-8345aebd86a848f8	2020-04-12 13:01:16	226e0f5e-8b946a36	2	160
ff72f0faf0574419-8345aebd86a848f8	2020-04-12 13:01:16	419f9d0e-a8fa881e	2	110
ff72f0faf0574419-8345aebd86a848f8	2020-04-12 13:01:16	2fae974c-c6fb6f9d	1	164
ff72f0faf0574419-8345aebd86a848f8	2020-04-12 13:01:16	1c8efa6a-db242122	1	325
ff72f0faf0574419-8345aebd86a848f8	2020-04-12 13:01:16	2b8ddc68-5e107d12	2	180
ff72f0faf0574419-8345aebd86a848f8	2020-04-12 13:01:16	7ad41de2-bb1053b7	2	20
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	3e206ded-5187d531	1	1050
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	68e36ae4-6ad4bcc7	2	120
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	65716d5f-ea0abeef	1	330
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	1c442ea5-345d0cf9	1	750
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	5203a09d-2c21b3e2	1	195
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	871c68b0-3c4bc1b0	1	820
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	6369ea72-ff33c95c	2	230
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	10209df8-d01bfba0	2	370
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	57461436-c8941730	3	164
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	22023381-46884c00	1	1470
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	1a1608d0-fd61fe4b	1	470
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	56e398b0-c6d14156	1	510
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	7803055f-1bc75cba	2	175
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	18d87992-0b9e4f1c	3	200
f742882a49e74774-8361a0e6cc7595f0	2020-05-07 17:49:46	39c195a9-86882936	3	78

Console

Table 2:Splitted Data in Order

After that we cleaned unmanaged data and then we analyzed data by using aggregate queries according to our project requirements.

Here are some of the examples of aggregate queries which we used during the data analyzed process.

```sql
SELECT
    DATE_FORMAT(order_date, '%Y-%m') as Date,
    sum(quantity) as quantity
FROM bills
GROUP BY YEAR(order_date),MONTH(order_date)
ORDER BY order_date
```

Figure 2: Query to select Date and Quantity

1	Date	quantity
2	2020-01	618
3	2020-02	666
4	2020-03	1021
5	2020-04	1267
6	2020-05	987
7	2020-06	642
8	2020-07	698
9	2020-08	870
10	2020-09	839
11	2020-10	834
12	2020-11	573
13	2020-12	611
14	2021-01	566
15	2021-02	502
16	2021-03	527
17	2021-04	777
18	2021-05	1488
19	2021-06	856
20	2021-07	656
21	2021-08	725
22	2021-09	671
23	2021-10	572
24	2021-11	672
25	2021-12	750

Table 3:Date and Quantity

```sql
SELECT
    DATE_FORMAT(order_date, '%Y-%m') as Date,
    sum(price) as price
FROM bills
GROUP BY YEAR(order_date),MONTH(order_date)
ORDER BY order_date
```

Figure 3:Date and Price

1	Date	price
2	2020-01	47001
3	2020-02	70609
4	2020-03	173928
5	2020-04	182180
6	2020-05	178814
7	2020-06	111961
8	2020-07	116041
9	2020-08	145128
10	2020-09	156981
11	2020-10	142440
12	2020-11	108011
13	2020-12	121915
14	2021-01	99701
15	2021-02	90387
16	2021-03	93751
17	2021-04	136531
18	2021-05	245964
19	2021-06	165766
20	2021-07	128742
21	2021-08	148853
22	2021-09	144979
23	2021-10	127395
24	2021-11	156685
25	2021-12	180000

Table 4:Date and Price

[A] Forecasting the average value of quantity and price per month since 2022

3.3 Data preparation and EDA:

Hereby, the datasets are prepared and preprocessed for Exploratory Data Analysis. The datasets are visualized and analyzed and the parameters or datasets that did not have much effect in our prediction model were neglected and removed. EDA was done to make it easier for us to find anomalies in the datasets or discover patterns in our datasets.

Step 1: we did the process of EDA (Exploratory Data Analysis). Here, we plot different graphs

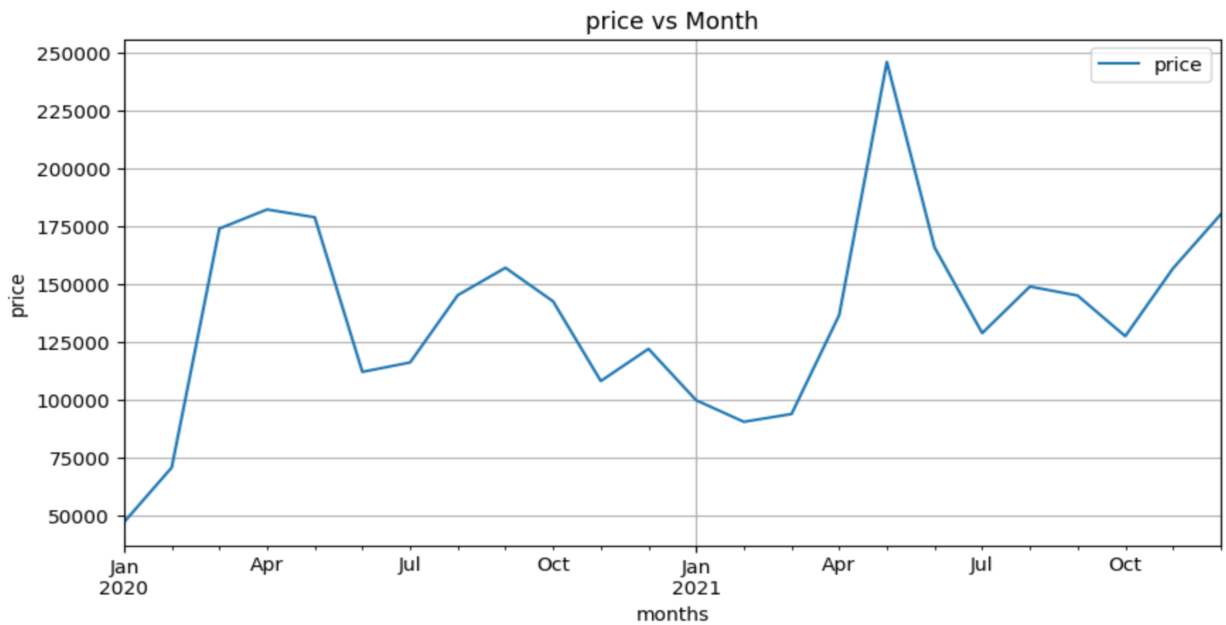


Figure 5:Price Vs month

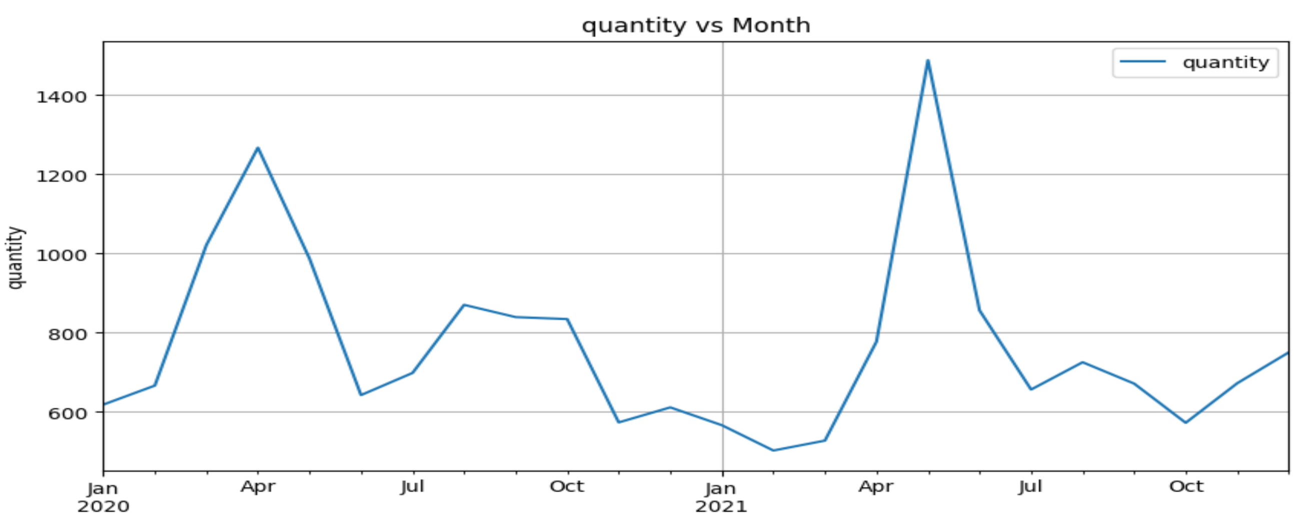


Figure 4:Quantiy VS month

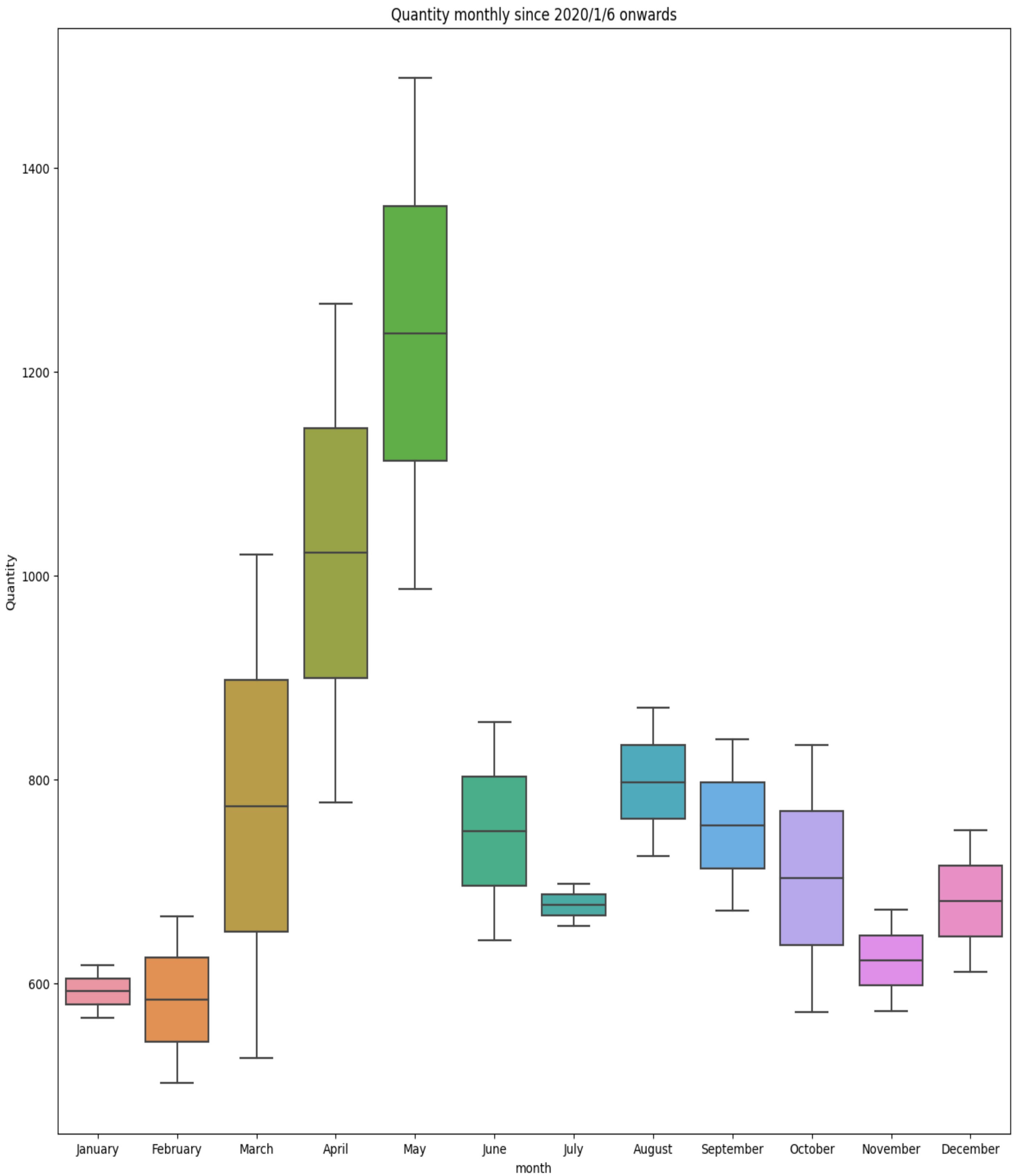


Figure 6:Box Plot(Quantity vs month)

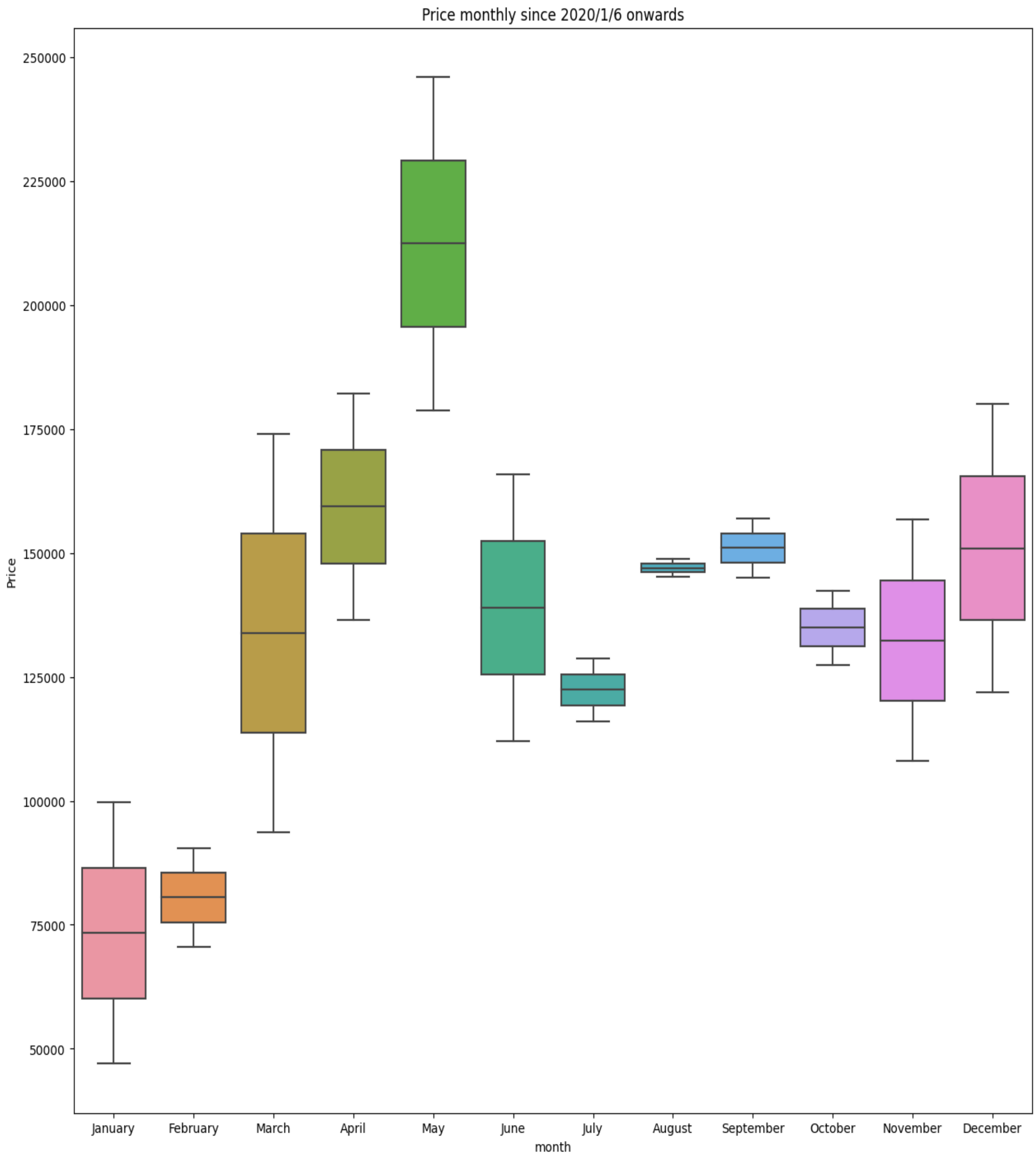


Figure 7:Box Plot (Price Vs month)

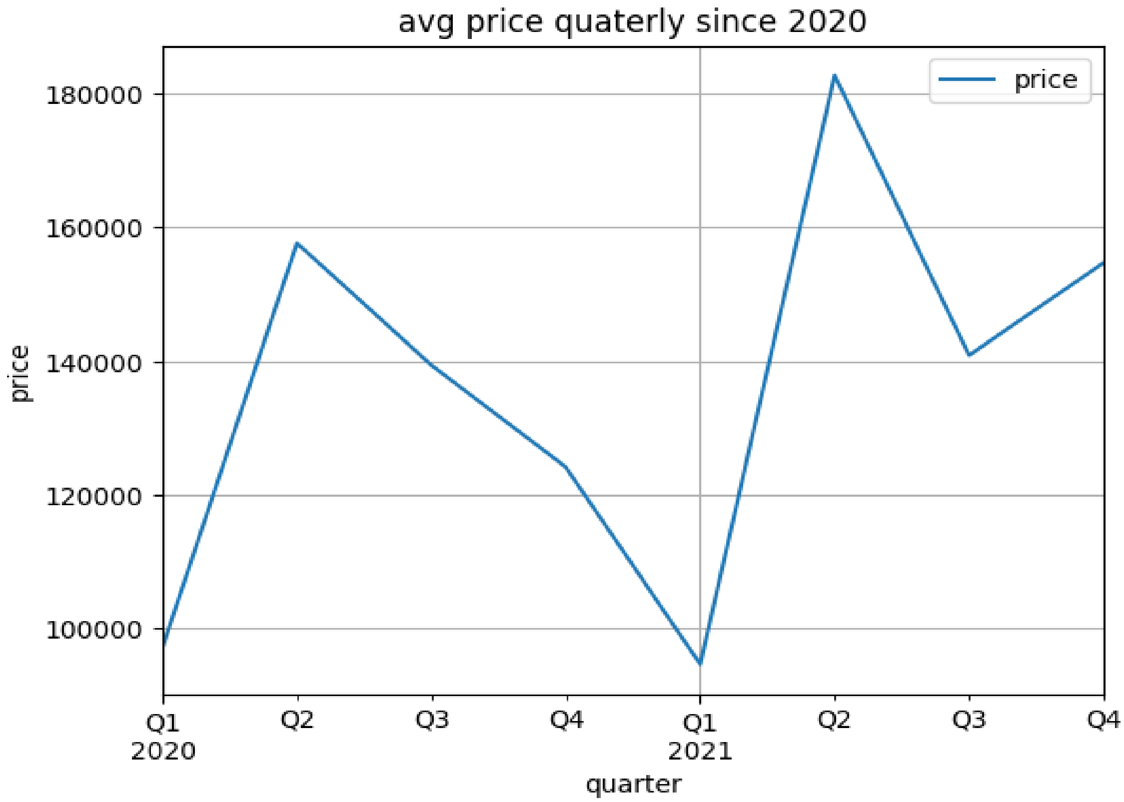


Figure 9:Average price quarterly

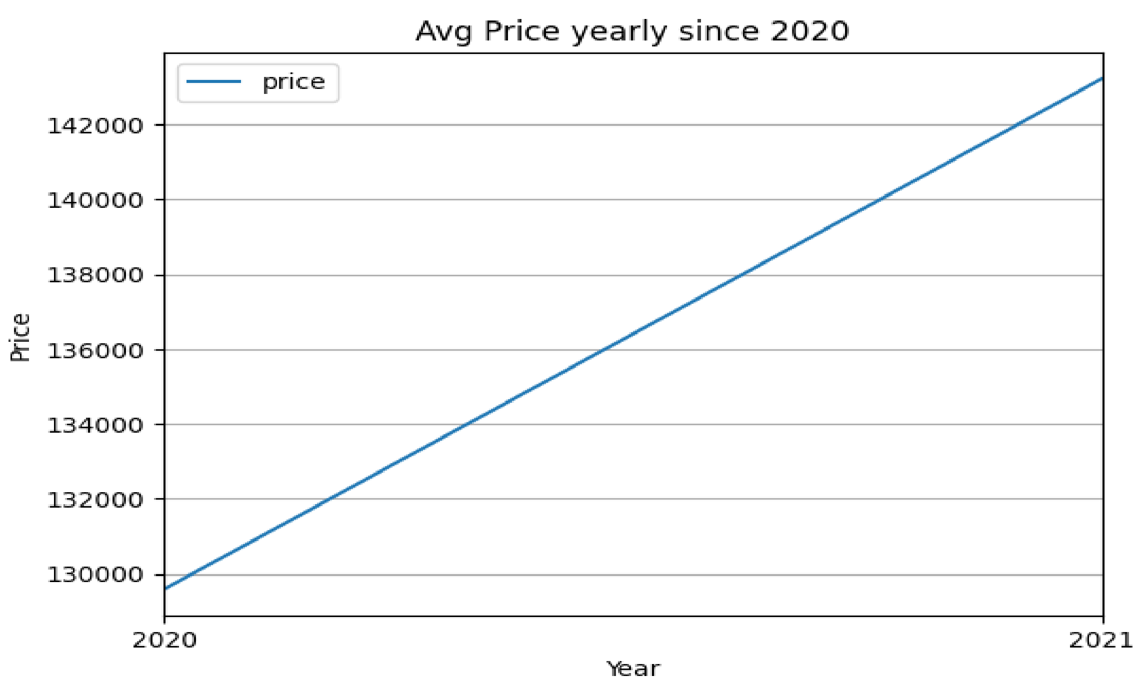


Figure 8:Average price yearly

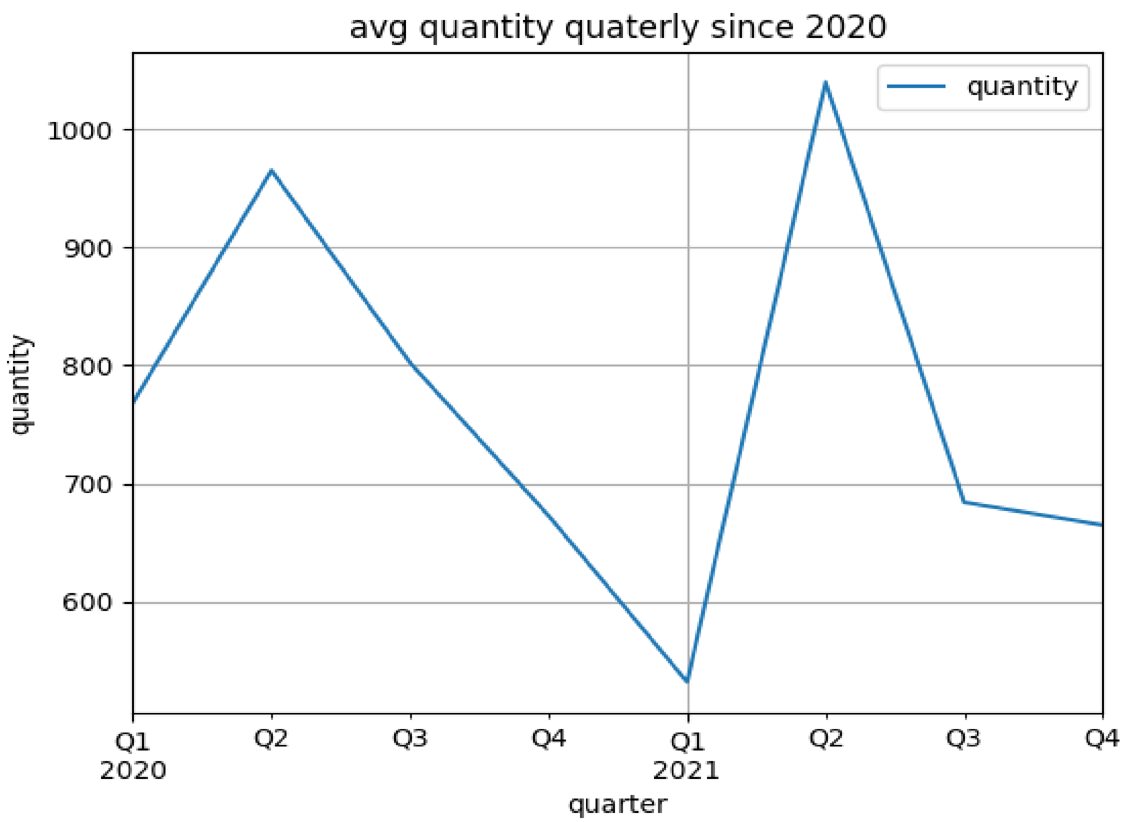


Figure 10:Average quantity quarterly

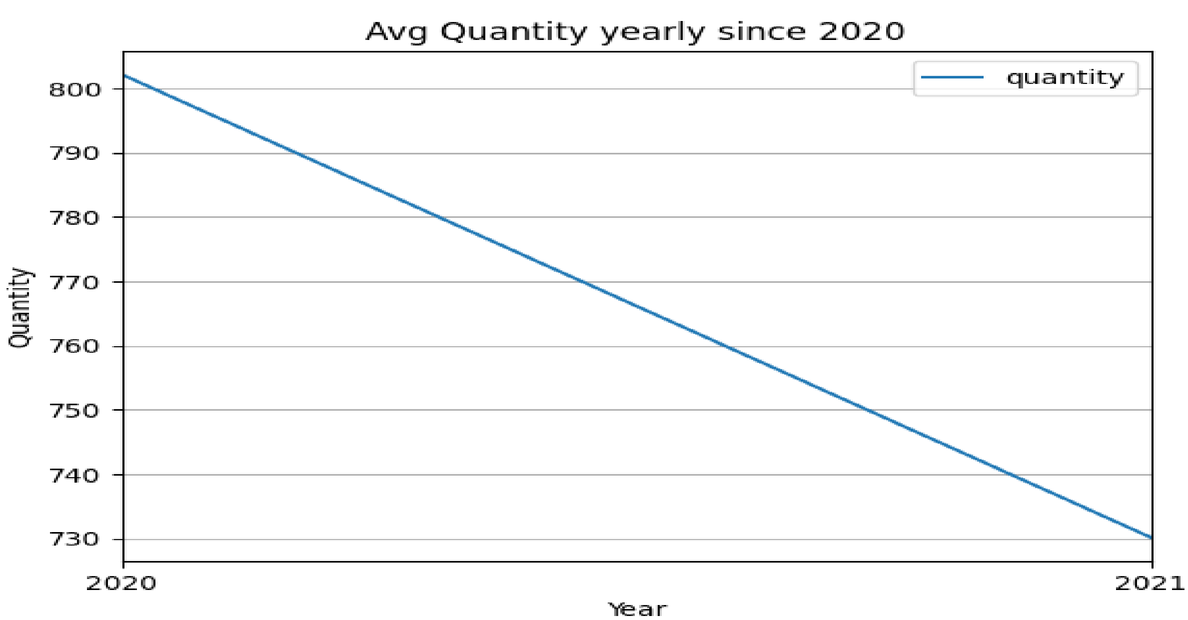


Figure 11:Average quantity yearly

Step 2: We Analysis the coefficient of variation.

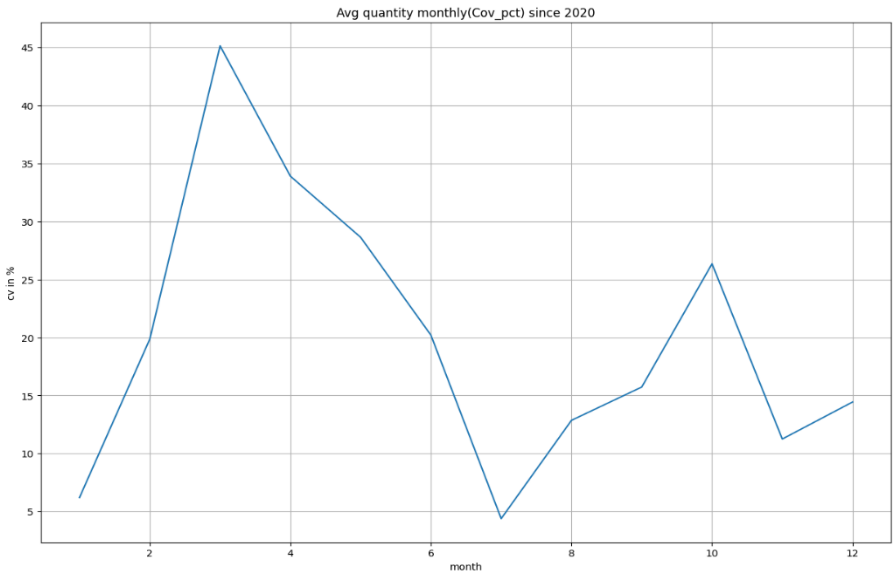


Figure 12:average quantity monthly(covt_pct)

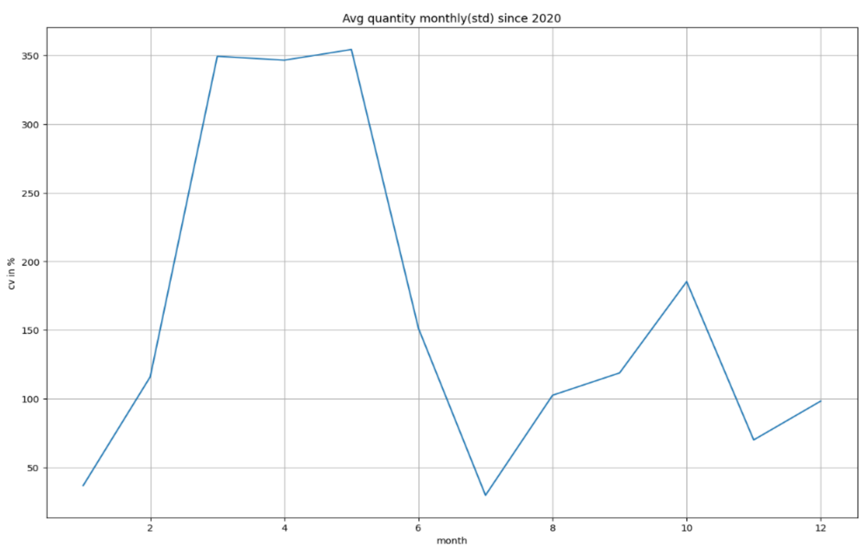


Figure 13:average quantity monthly(std)

Figure 15: Average monthly (Std)

Figure 14:Average monthly(cov_ppt)

Step 3: We splitted data into train and test.

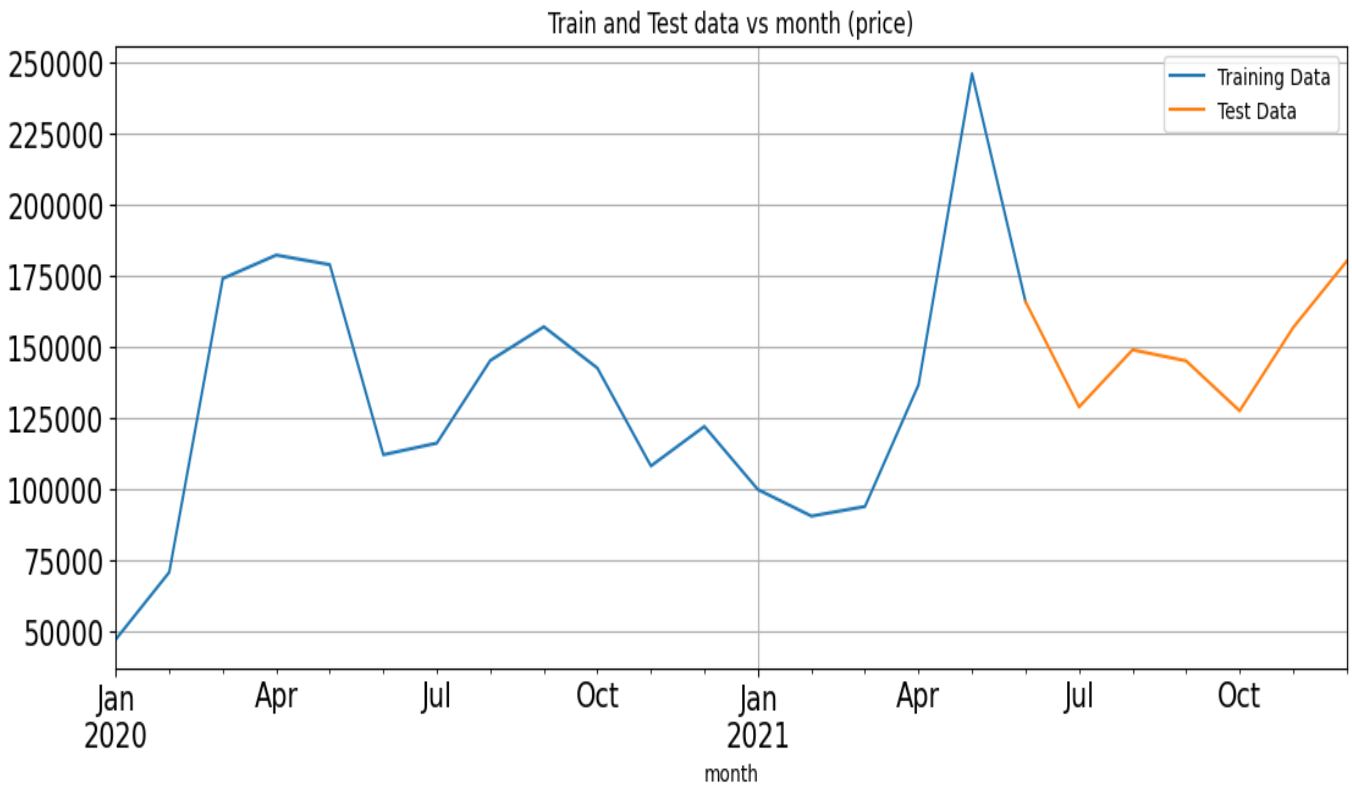


Figure 16: train and test data vs month for price

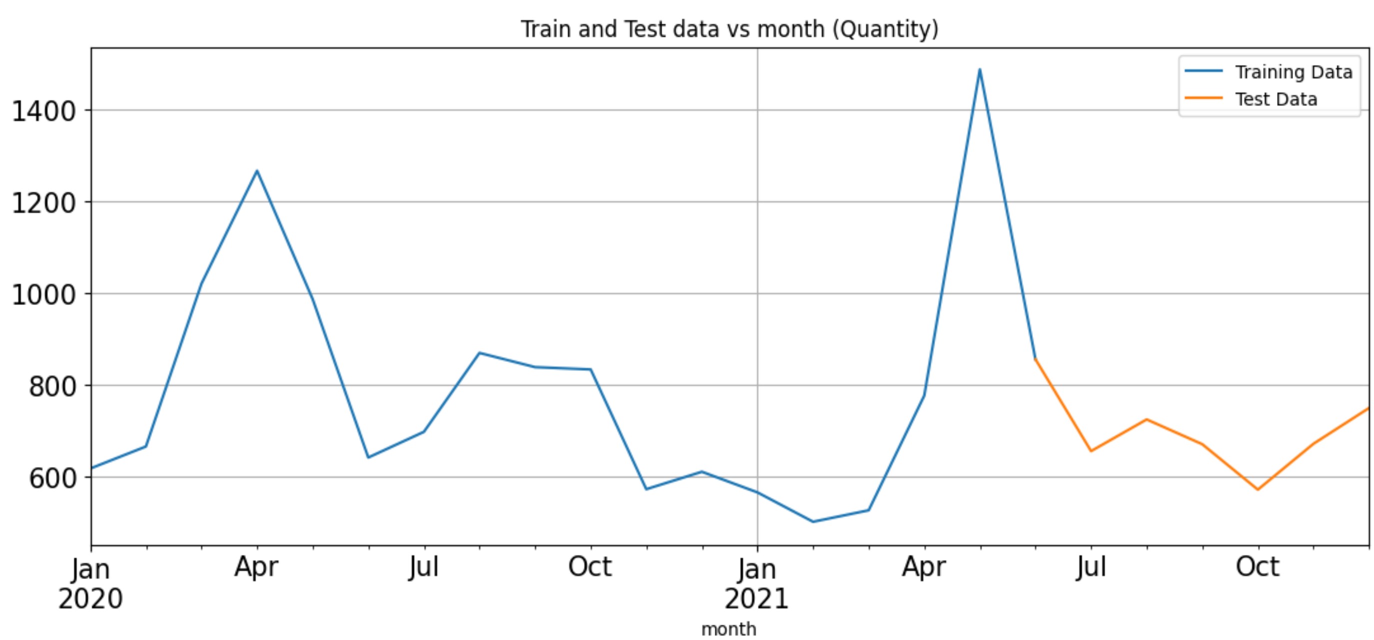


Figure 17:train and test data vs month for quantity

Step 4: We did the Model formation. Here, we apply Linear Regression, Naive Regression and ExponentialSmoothing.

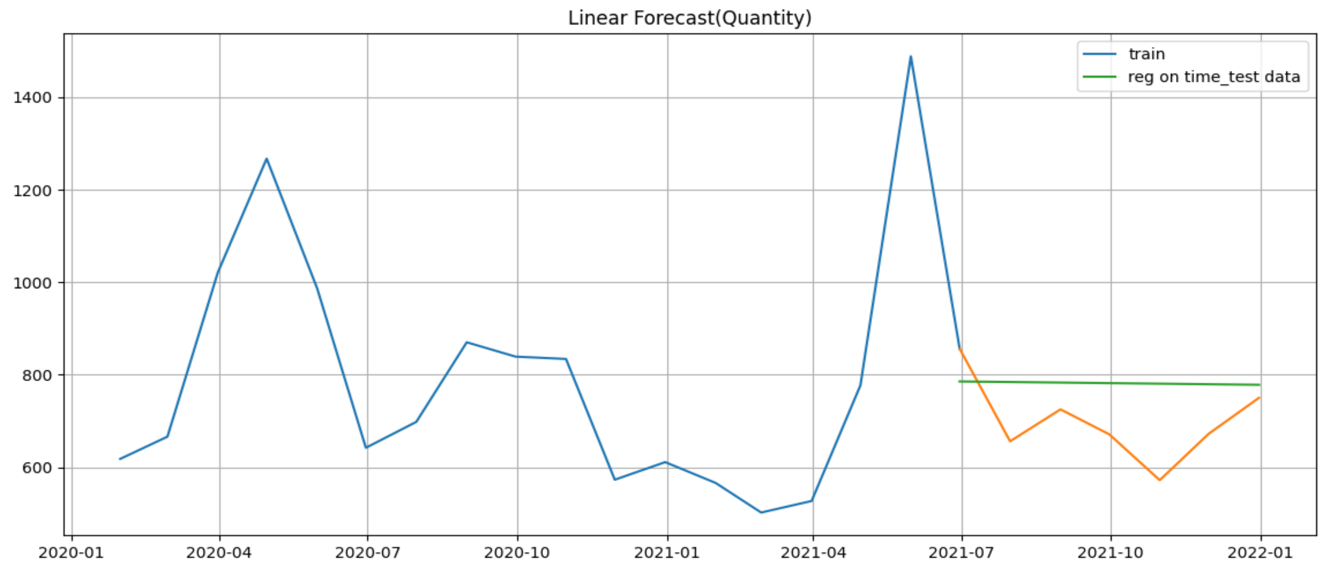


Figure 18:linear forecast for quantity

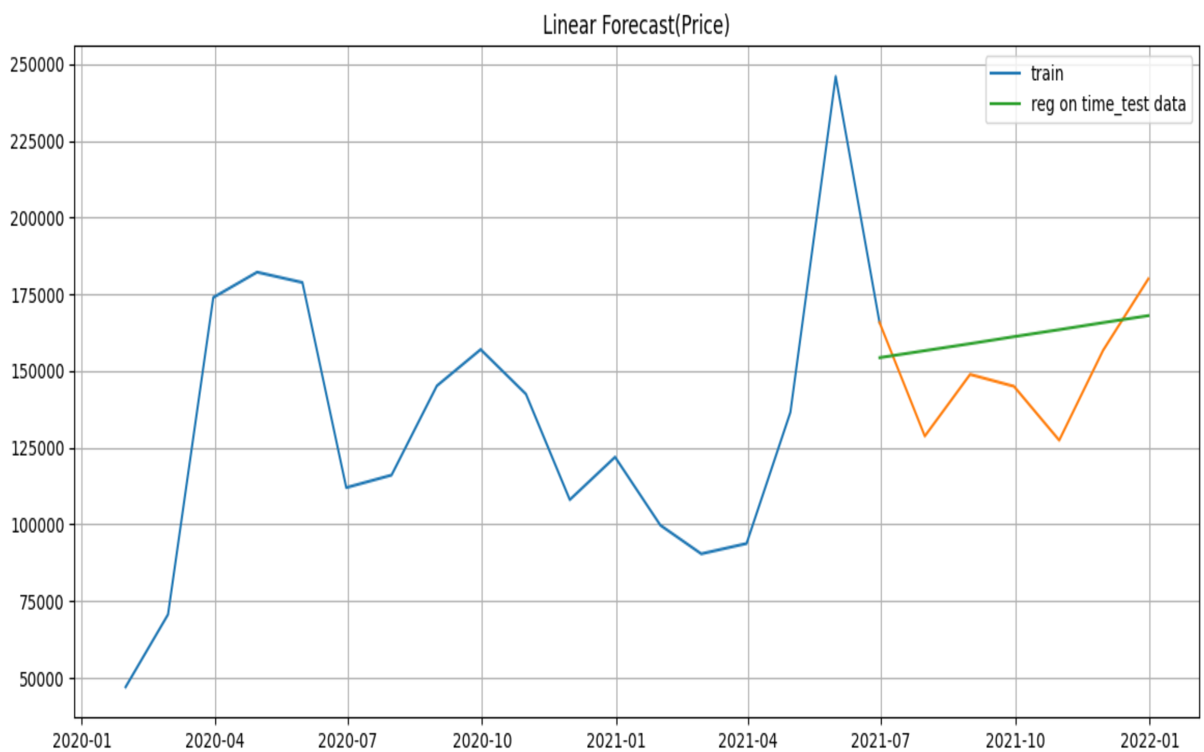


Figure 19:linear forecast for price

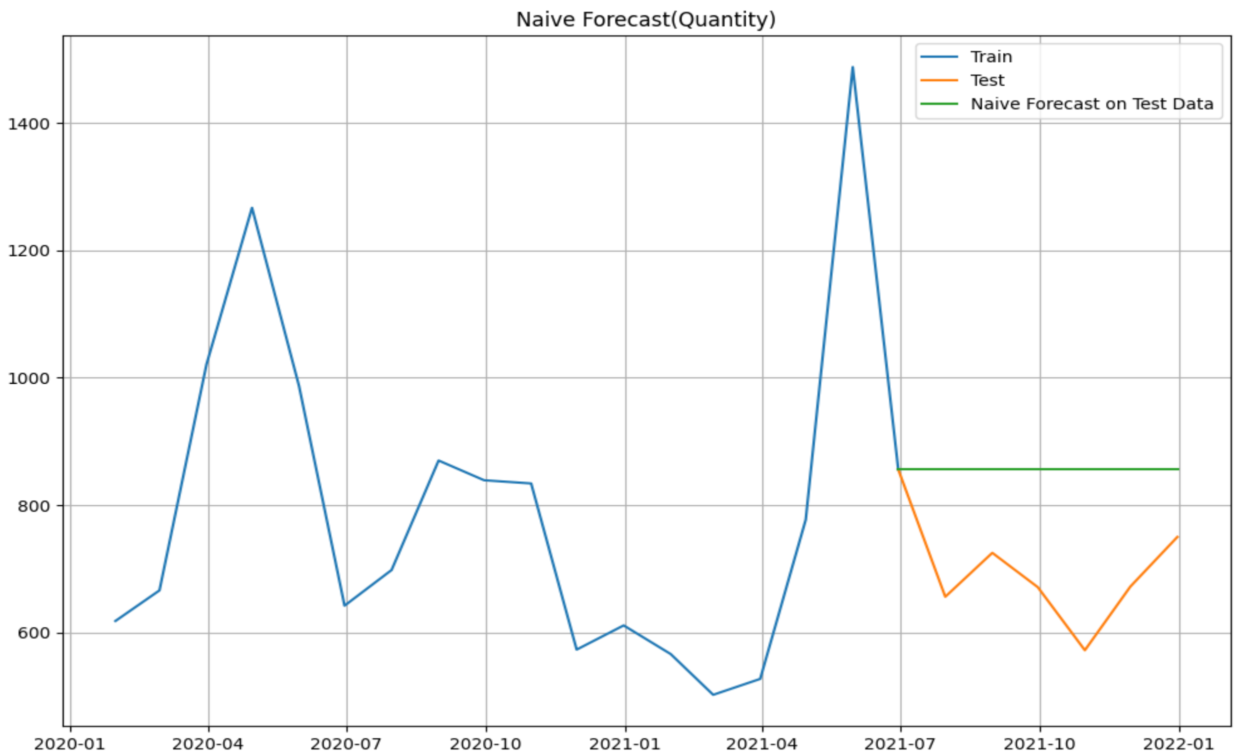


Figure 20:Naive forecast for quantity

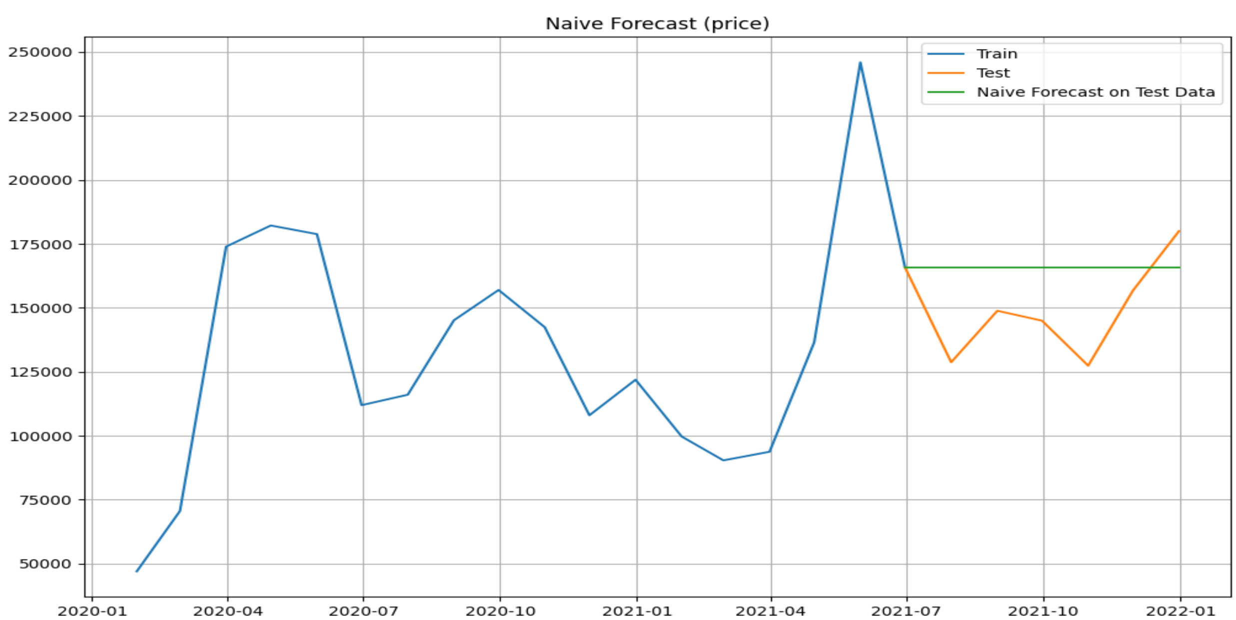


Figure 21:Naive Forecast for price

	Test Mape (%)
RegressionOnTime	15.50
NaiveModel	23.90
exponentialSmoothing	14.02

Figure 22:Test MAPE result

After comparing the above three models, we get the least MAPE value on ExponentialSmoothing. So, we choose ExponentialSmoothing for final forecasting.

Step 5: we did the final forecasting for price and quantity where we seen the following graph.

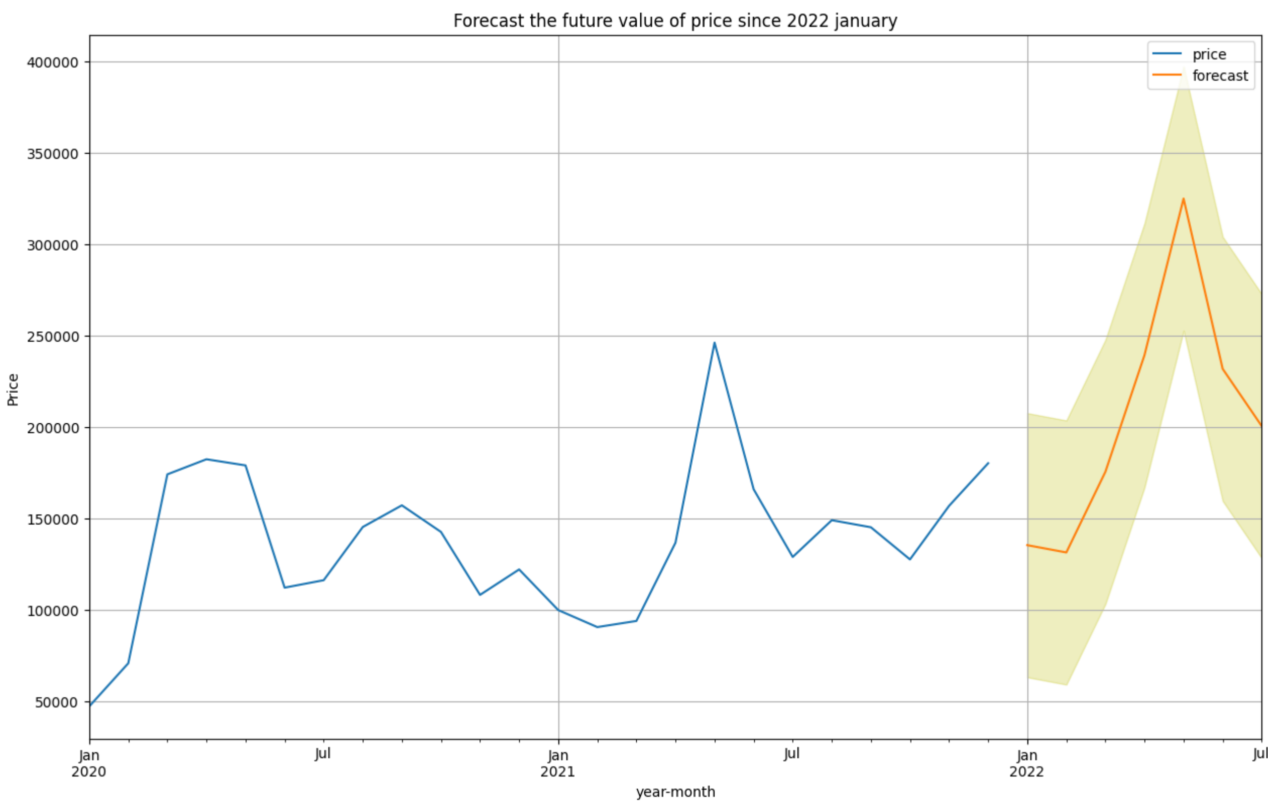


Figure 23:forecast the future value of price

[B] Predicting which quantity is required in what amount using ARIMA model.

Step1: We calculate Autocorrelation and Partial Autocorrelation of data set.

```
# Plot the ACF and PACF plots
plt.figure(figsize=(12, 6))
plt.subplot(211)
plot_acf(time_series, lags=30, ax=plt.gca(), alpha=0.05)
plt.subplot(212)
plot_pacf(time_series, lags=30, ax=plt.gca(), alpha=0.05)
plt.show()
```

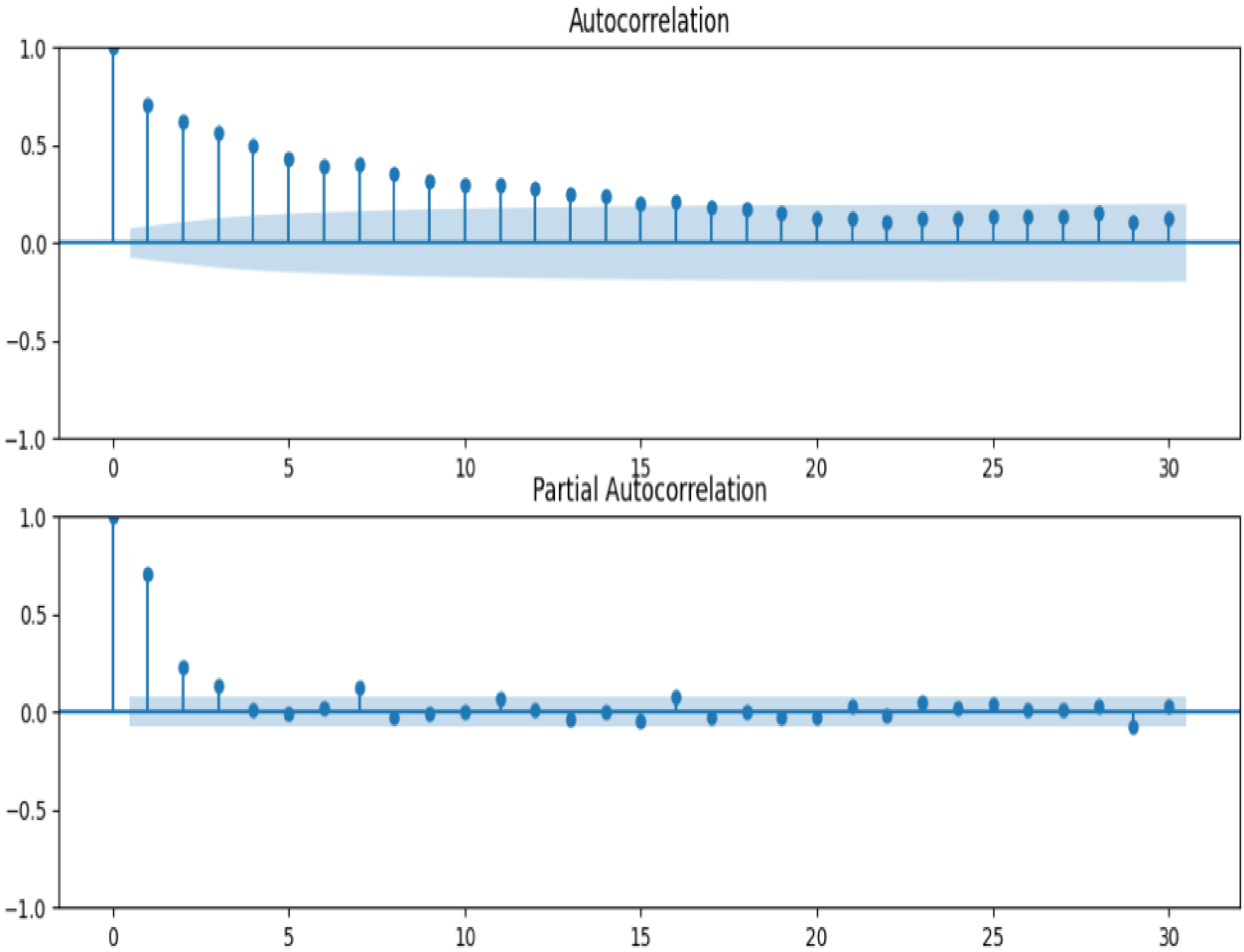
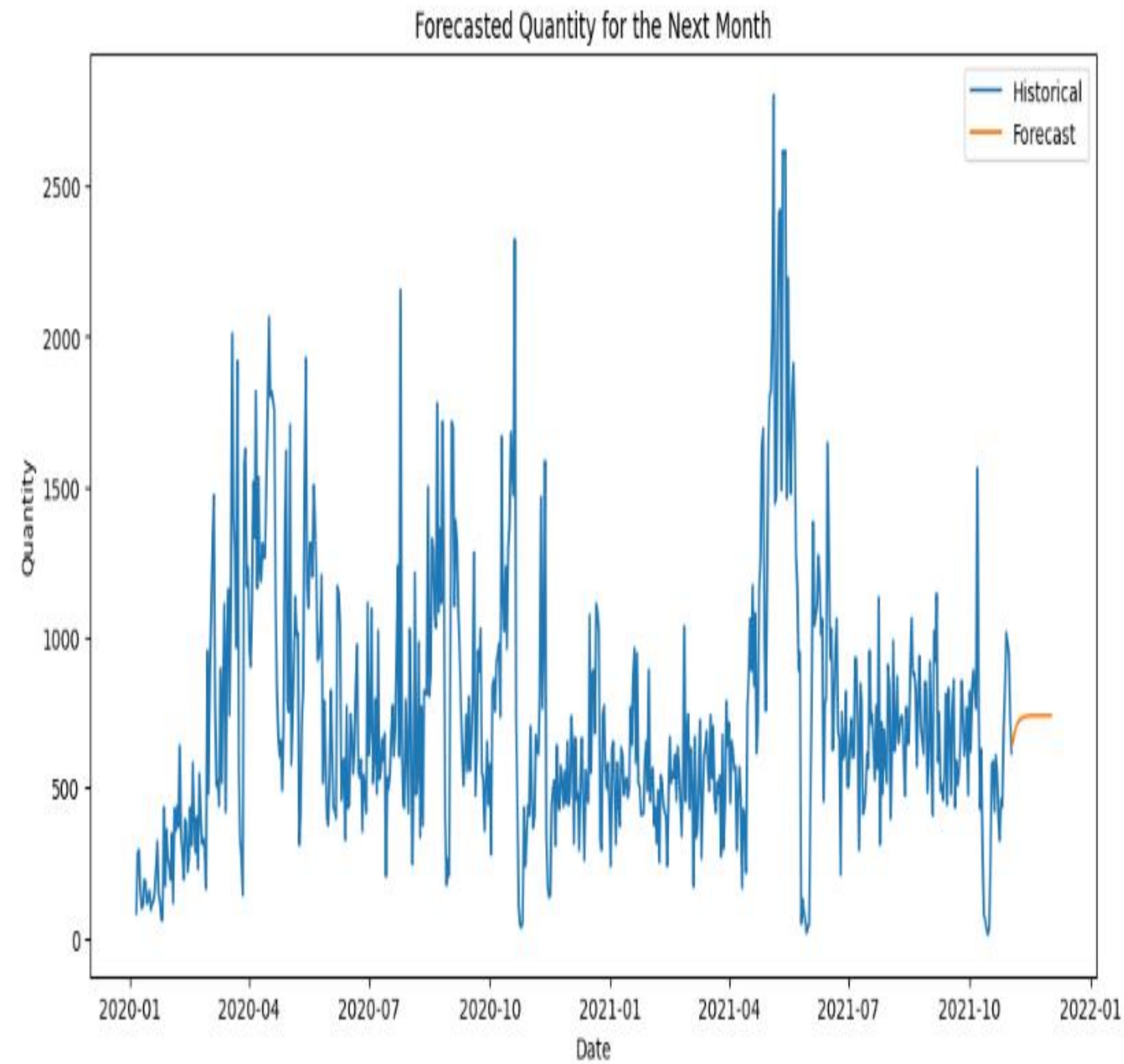


Figure 24:autocorrelation and partial autocorrelation

Step 2: We use ARIMA model to forecast the quantity for next month.




```
Total quantity for the next month: 21909
```

Figure 25:Total quantity for the next month

Step 3: We use table to show which quantity is required in what amount in ascending and descending order.

Descending Order:

Product ID	Quantity
7ad41de2-bb1053b7	12747
57461436-c8941730	11303
3d82cfe9-92ab1e78	7203
b2173f97-bae900c0	5899
fa12e293-40941901	5059
226e0f5e-8b946a36	4812
65716d5f-ea0abeef	4601
132dbfb2-98b798a9	4332

Table 5:Quantity order related to Product_ID

Chapter 4: Implementation Details

4.1 Software Requirement

The list of software that will be used to run this system is listed below:

- Any operating system (Linux, Windows, MacOS)
- Web Browser
- Code Editor (Jupyter Notebook)

4.2 Hardware Requirement

The list of hardware that will be used to run this system is listed below:

- General PC (min RAM 8GB, HDD 500GB, SSD adds value)

Chapter 5: Results and Analysis

For the first objective of our project that is Forecasting the future average value of quantity and price per month since 2022 we apply Linear regression, Naïve regression and ExponentialSmoothing and comparing the MAPE value of each where we found the least MAPE value of ExponentialSmoothing. So, we choose ExponentialSmoothing for the forecasting of future average value of quantity and price.

	Test Mape (%)
RegressionOnTime	15.50
NaiveModel	23.90
exponentialSmoothing	14.02

Figure 26:Test MAPE result

After comparing the above three models, we get the least MAPE value on ExponentialSmoothing. So, we choose ExponentialSmoothing for final forecasting.

For the second objective of our project, we use ARIMA model to predict which product is needed in what amount in coming next month. And show the required quantities in ascending and descending order.

We forecast the future average value of quantity and price per month since 2022 this helps the user to make business strategy accordingly.

Descending Order:

Product ID	Quantity
7ad41de2-bb1053b7	12747
57461436-c8941730	11303
3d82cfe9-92ab1e78	7203
b2173f97-bae900c0	5899
fa12e293-40941901	5059
226e0f5e-8b946a36	4812
65716d5f-ea0abeef	4601
132dbfb2-98b798a9	4332

Table 6: Predicted quantities of individual product for next month

Chapter 6: Conclusion

Grocery Demand Prediction is very crucial for every company, especially big ones and this process is overly complex because there are a lot of factors. In this study, we used three machine learning algorithms (Linear Regression, Navie Regression, Exponential Smoothing) for quantity forecasts, Exponential Smoothing performed better, as it had a lower MAPE value than the other two models.

The Grocery Demand Prediction which has been proposed is based upon ARIMA model and is able to provide suitable predictions about the future quantity of the customer.

References

1]Dahlen, R. S. (2016 6th). *Predicting sales in a food store department Using machine learning.* International Conference on Information and Communication Technology for Muslim World.

[2] Kumari, N. (n.d.). Predicting Online Grocery Ordering Intention. *Department of CSE, VIT University Chennai Campus, India*.

[3] Odegua, R. (n.d.). *Applied Machine Learning for Supermarket Sales Prediction.* Ekpoma, Edo state, Nigeria: Department of Computer Science Ambrose Alli University.

[4] Pawar, P. M. (n.d.). Retrieved from www.ijera.com