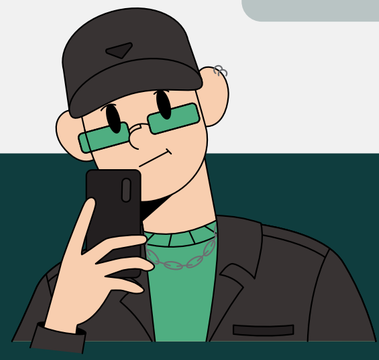
Rakamin Academy X KALBE Nutritionals

TIME SERIES FORECASTING AND CUSTOMER SEGMENTATION: LEVERAGING ARIMA AND K-MEANS IN MACHINE LEARNING

Data Scientist Intern

Presented by **Kevin Laurent Oktavian Putra**

FINAL PROJECT



ABOUT ME

Kevin Laurent Oktavian Putra

Highly motivated Statistics and Computer Science students who have an interest in data analysis, predictive modeling, research, and projects. Have one year of experience in statistical and technology research papers and publications. Active in organizations and events in administration and planning. Committed to utilizing statistical theories and methods to collect, analyze, interpret, and present data through R and Python Programming.

Let's get connect

LINKEDIN.COM/IN/KEVINLOP/

Time Series

CASE STUDY

Clustering

PROBLEM DEFINITION

IN TODAY'S DATA-DRIVEN BUSINESS LANDSCAPE, UNDERSTANDING AND LEVERAGING TIME SERIES DATA IS CRITICAL FOR COMPANIES TO MAKE INFORMED DECISIONS AND ENHANCE CUSTOMER EXPERIENCES. THIS PROJECT AIMS TO ADDRESS TWO SIGNIFICANT CHALLENGES:

TIME SERIES FORECASTING

Time series data, such as historical sales data or website traffic patterns, often exhibit complex temporal dependencies. Accurate predictions are crucial for optimizing inventory, resource allocation, and overall business performance. Using the ARIMA (AutoRegressive Integrated Moving Average) model, this project seeks to develop a machine learning solution that can forecast future values in time series data with precision.

CUSTOMER SEGMENTATION

Understanding the diverse needs and behaviors of customers is pivotal for creating tailored marketing strategies and improving customer satisfaction. Employing k-Means clustering, this project aims to group customers into distinct segments based on their purchasing patterns, demographics, or other relevant features. By doing so, it aims to provide businesses with actionable insights for personalized marketing and service offerings

SQL QUERIES

PostgreSQL

SQL QUERIES

1

WHAT IS THE AVERAGE AGE OF CUSTOMERS BASED ON THEIR MARITAL STATUS?

```sql
SELECT `Marital Status`, AVG(Age) AS 'Average Age'
FROM customer
GROUP BY `Marital Status`;
```

Marital Status	Average Age
	31.333333333333332
Married	43.03823529411765
Single	29.384615384615383

- People’s average age who’s not inputting the marital status / do not have marital status is 31 years old.
- Married people’s average age is around 43 years old.
- Single people’s average age is around 29 years old.

SQL QUERIES

2

WHAT IS THE AVERAGE AGE OF CUSTOMERS BASED ON THEIR GENDERS?

```sql
SELECT `Gender`, AVG(Age) AS 'Average Age'
FROM customer
GROUP BY `Gender`;
```

Gender	Average Age
0	40.32644628099174
1	39.141463414634146

- 0 for woman and 1 for man.
- Customer who is a woman has average age around 40 years old.
- Customer who is a man has an average age around 39 years old.

SQL QUERIES

3

WHAT IS THE STORE NAME WITH MOST QUANTITIES SOLD?

```sql
SELECT s.StoreName, SUM(t.Qty) AS TotalQuantity
FROM transaction t
JOIN store s ON t.StoreID = s.StoreID
GROUP BY s.StoreName
ORDER BY TotalQuantity DESC
LIMIT 1;
```

StoreName	TotalQuantity
Lingga	2777

- “Lingga” store is being the store with the most quantities sold with the total of 2777 items.

SQL QUERIES

4

WHAT IS THE BEST SELLING PRODUCTS BASED ON THE REVENUE OF TOTAL AMOUNT ITEMS?

```sql
SELECT p.`Product Name`, SUM(t.TotalAmount) AS TotalAmount
FROM transaction t
JOIN product p ON t.ProductID = p.ProductID
GROUP BY p.`Product Name`
ORDER BY TotalAmount DESC
LIMIT 1;
```

Product Name	TotalAmount
Cheese Stick	27615000

- The best selling product goes to “Cheese Stick” with the total revenue of Rp 27,615,000.00

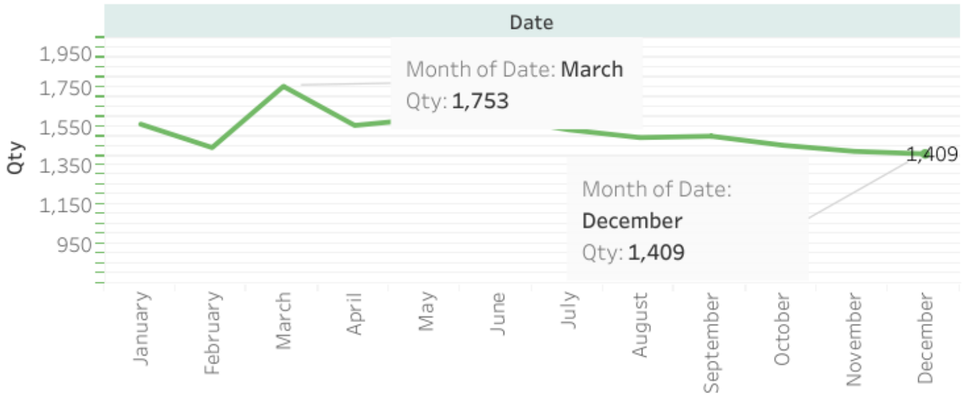
DATA VISUALIZATION

tableau®

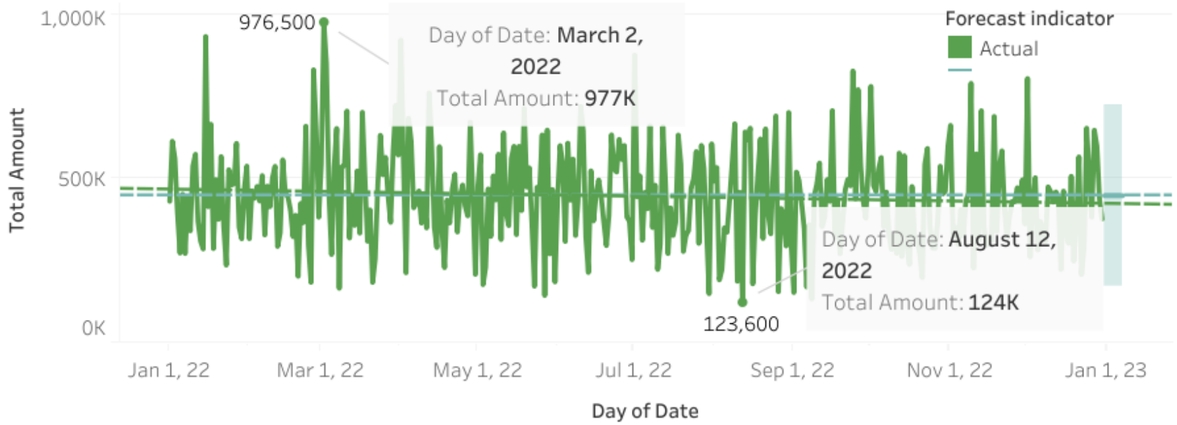
TABLEAU DASHBOARD

Product Sales Revenue Based on Total Amount and Quantity Across the Stores (2022)

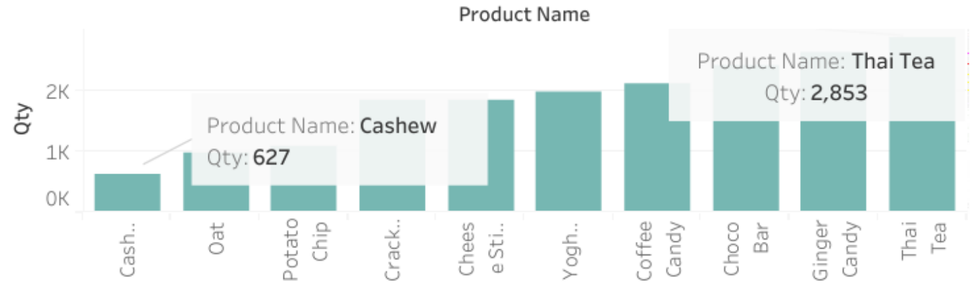
Montly Sales Transaction Based on Quantities (2022)



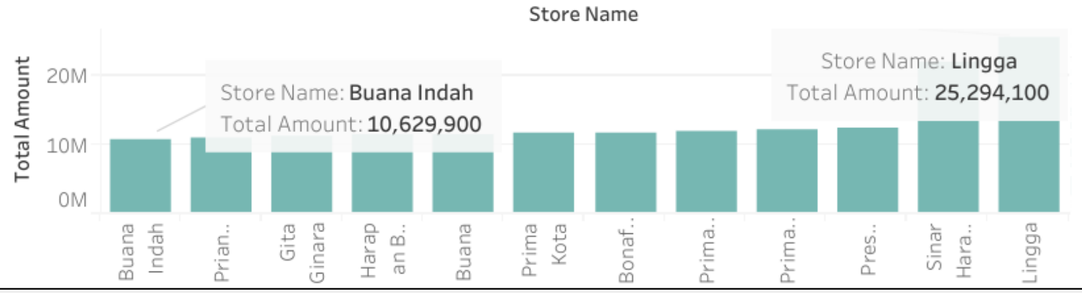
Daily Sales Transaction Based on Total Amount (2022)



Product Sales Based on Quantities (2022)



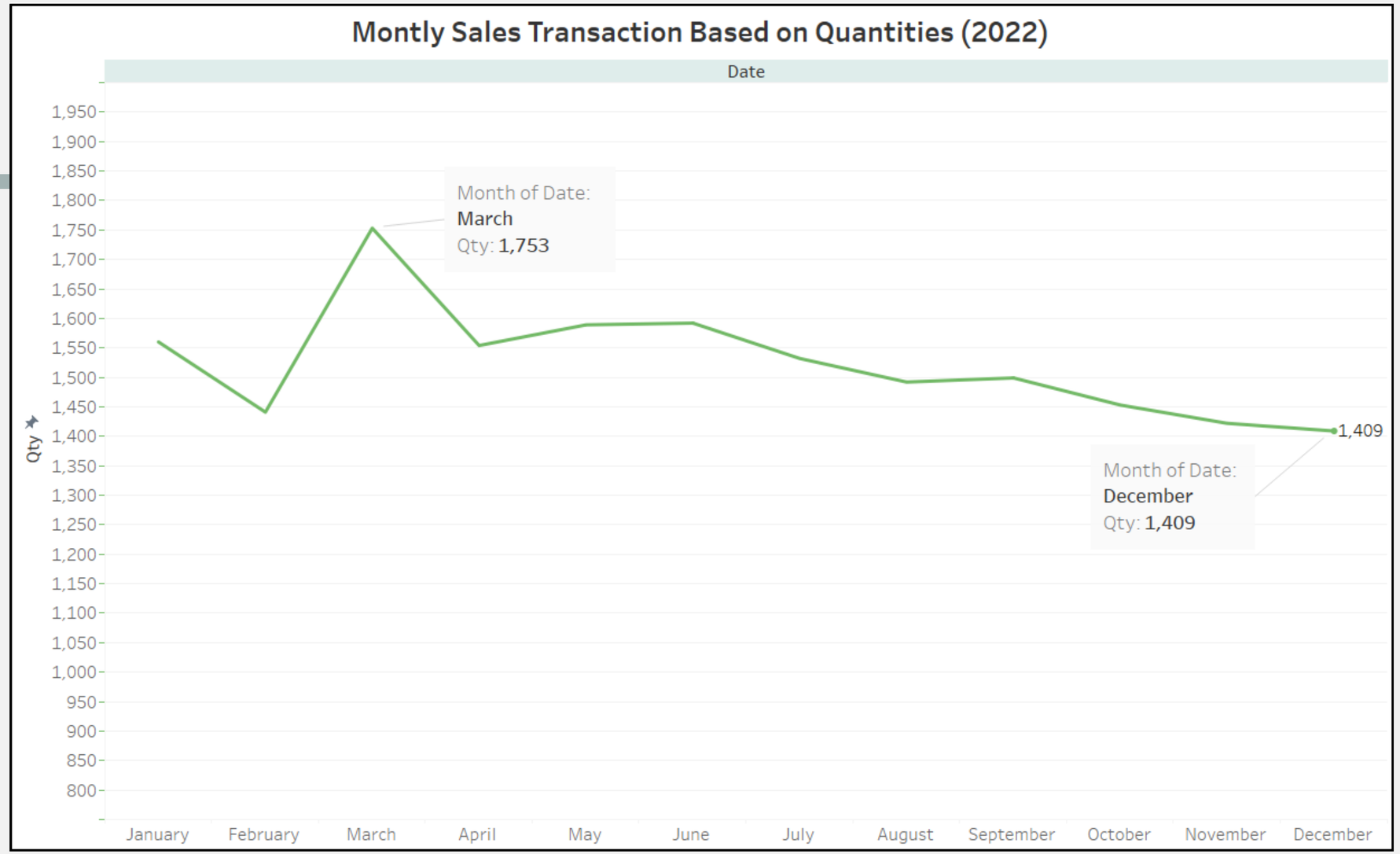
Store Sales Based on Total Amount of Revenue (2022)



DATA VISUALIZATION

Quantity vs Month

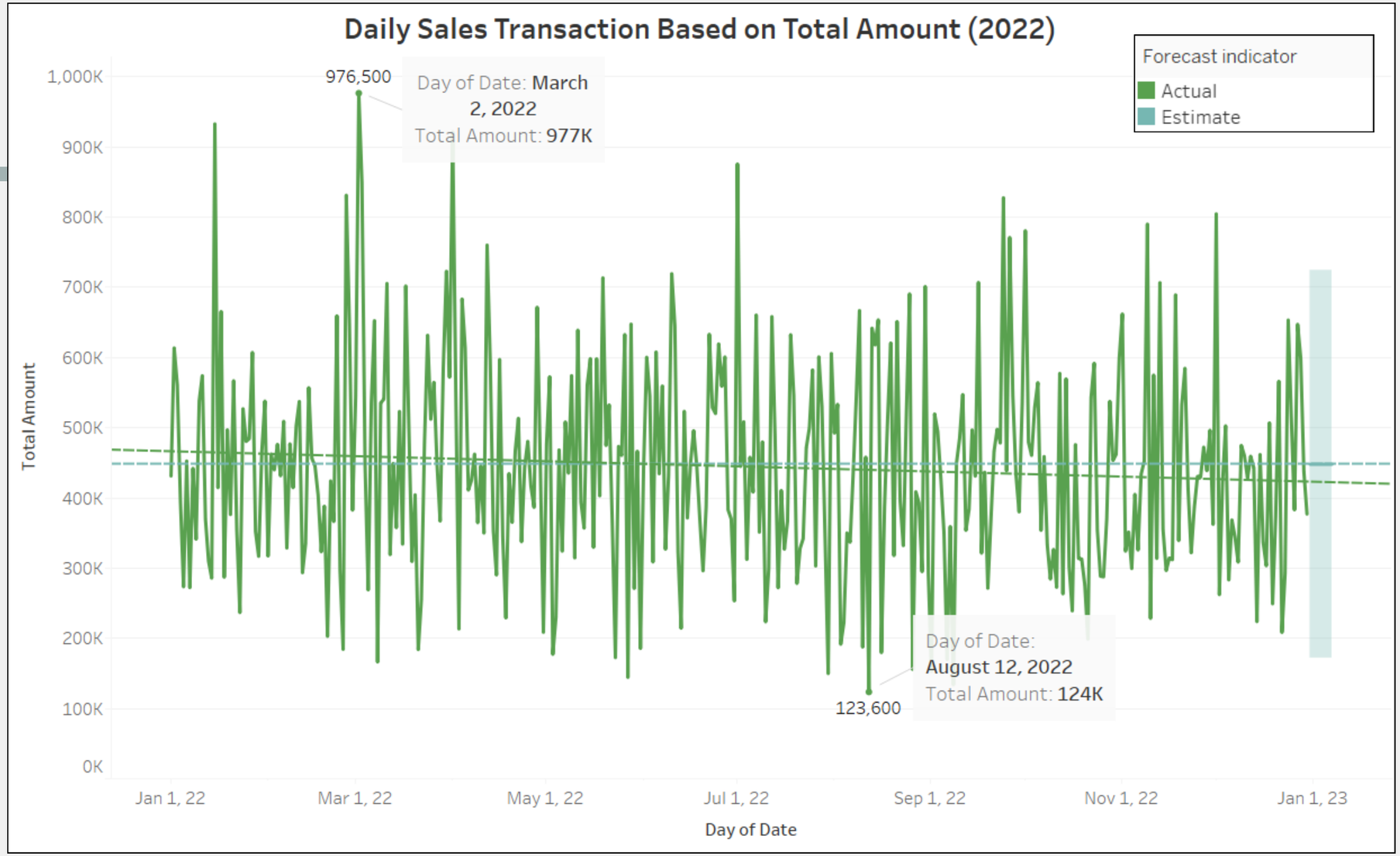
- Throughout the month of the year 2022, the sales are quite fluctuated.
- The highest peak is in March with the total quantities sold around 1,753 items.
- The lowest peak is in the last month of the year, December, with the total quantities of 1,409.
- The Q1 has the best sales than the other 3 quartal (Q2,Q3, and Q4).



DATA VISUALIZATION

Total Amount vs Sales Per Day

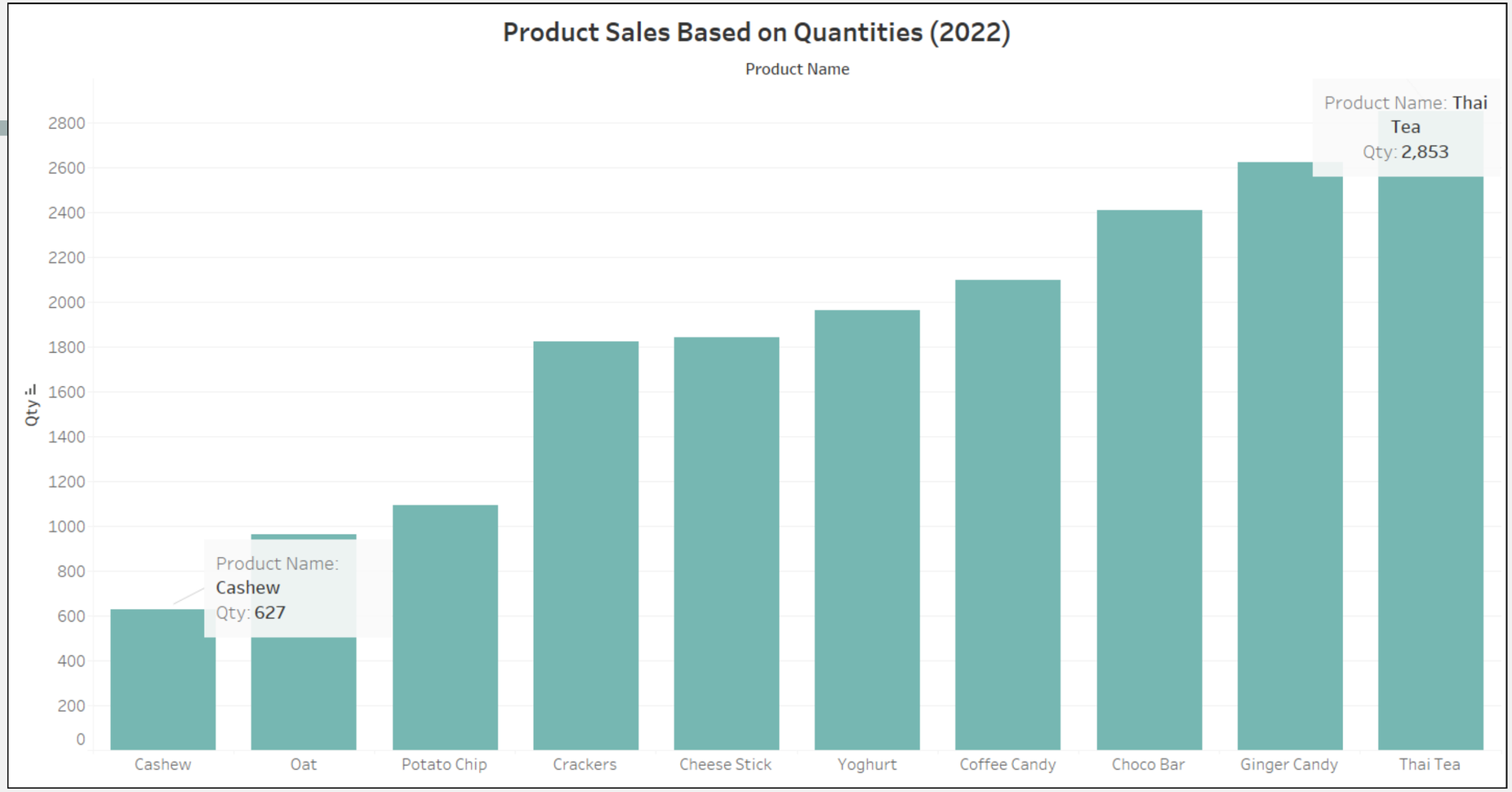
- Throughout the day of the year 2022, the sales are quite fluctuated.
- The highest peak is in March 2nd with the total amount of 977K.
- Unlike per month, the lowest peak was in August 12th with the total amount of 124K.
- The forecasting prediction shows that the sales would be around 450K while the actual one would drop down under 400K in the year of 2023.



DATA VISUALIZATION

Quantity vs Product Sales

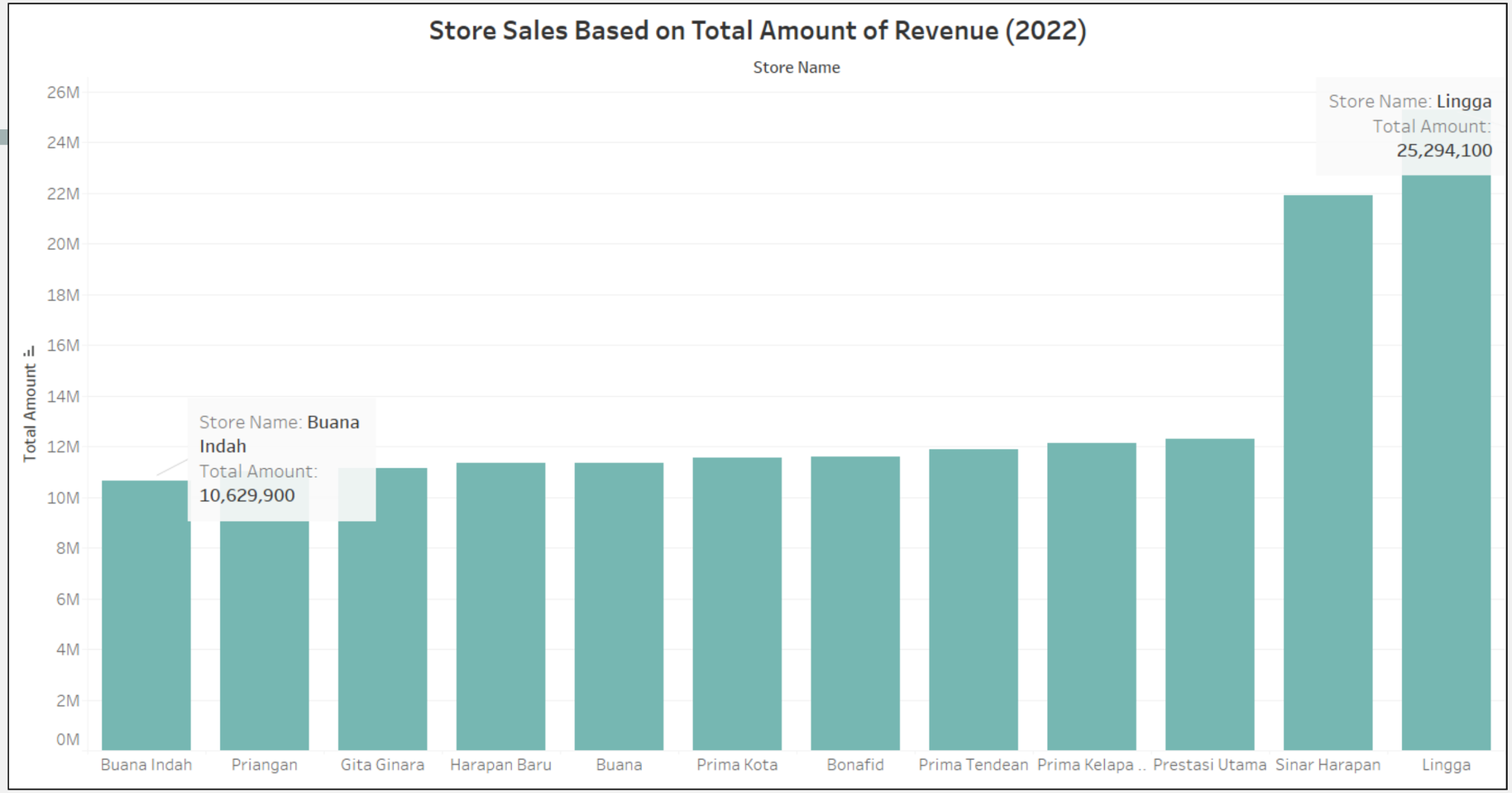
- The best selling product surprisingly goes to “Thai Tea” with the total of 2,853 products. With the average of man and woman is around 40 years old, it means that sweetened drinks are still favorable for people in 40's.
- Cashew being the lowest selling product with only 627 packages are able to be sold.



DATA VISUALIZATION

Store Sales Vs Total Amount

- “Lingga” store has the highest revenue of total amount of sales , which is Rp 25,294,100.00. It’s the highest along with “Sinar Harapan”.
- “Buana Indah” being the lowest for its revenue stream over the year with the total amount of Rp 10,629,900.00. Despite that, it’s revenue are not far with the other 9 stores.



python™

DATA

exploratory data analysis

PREPROCESSING

DATA PREPROCESSING

IMPORT LIBRARIES AND DATASETS

	CustomerID	Age	Gender	Marital Status	Income
0	1	55	1	Married	5,12
1	2	60	1	Married	6,23
2	3	32	1	Married	9,17

	StoreID	StoreName	GroupStore	Type	Latitude	Longitude
0	1	Prima Tendean	Prima	Modern Trade	-6,2	106,816666
1	2	Prima Kelapa Dua	Prima	Modern Trade	-6,914864	107,608238
2	3	Prima Kota	Prima	Modern Trade	-7,797068	110,370529

	ProductID	Product Name	Price
0	P1	Choco Bar	8800
1	P2	Ginger Candy	3200
2	P3	Crackers	7500

	TransactionID	CustomerID	Date	ProductID	Price	Qty	TotalAmount	StoreID
0	TR11369	328	01/01/2022	P3	7500	4	30000	12
1	TR16356	165	01/01/2022	P9	10000	7	70000	1
2	TR1984	183	01/01/2022	P1	8800	4	35200	4

DATA CLEANING (1)

FAMILIARIZE WITH THE DATASETS

```
RangeIndex: 447 entries, 0 to 446
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   CustomerID      447 non-null    int64
 1   Age             447 non-null    int64
 2   Gender          447 non-null    int64
 3   Marital Status  444 non-null    object
 4   Income          447 non-null    object
dtypes: int64(3), object(2)
```

```
RangeIndex: 10 entries, 0 to 9
Data columns (total 3 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   ProductID     10 non-null     object
 1   Product Name  10 non-null     object
 2   Price         10 non-null     int64
dtypes: int64(1), object(2)
```

```
RangeIndex: 14 entries, 0 to 13
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   StoreID     14 non-null     int64
 1   StoreName   14 non-null     object
 2   GroupStore  14 non-null     object
 3   Type        14 non-null     object
 4   Latitude    14 non-null     object
 5   Longitude   14 non-null     object
dtypes: int64(1), object(5)
```

```
RangeIndex: 5020 entries, 0 to 5019
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   TransactionID  5020 non-null   object
 1   CustomerID     5020 non-null   int64
 2   Date           5020 non-null   object
 3   ProductID      5020 non-null   object
 4   Price          5020 non-null   int64
 5   Qty            5020 non-null   int64
 6   TotalAmount    5020 non-null   int64
 7   StoreID        5020 non-null   int64
dtypes: int64(5), object(3)
```

- From the output, it can concluded that every dataset at least consists of integer value variables and object / string variables.
- For the 'product', 'store', and 'transaction' dataset, it can been that it is free from missing value. While in the 'customer' dataset, for 'Marital Status' it is detected to have 3 null values.

DATA CLEANING (2)

HANDLING WITH MISSING VALUES

```
CustomerID        0
Age               0
Gender            0
Marital Status    3
Income            0
dtype: int64
```

We wanted to fill the null values.

```
customer = customer.fillna(method='ffill')
```

```
CustomerID        0
Age               0
Gender            0
Marital Status    0
Income            0
dtype: int64
```

```
RangeIndex: 447 entries, 0 to 446
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   CustomerID      447 non-null    int64
 1   Age             447 non-null    int64
 2   Gender          447 non-null    int64
 3   Marital Status  447 non-null    object
 4   Income          447 non-null    object
dtypes: int64(3), object(2)
```

DATA CLEANING (3)

CHECK STRUCTURAL ERRORS

- The decimal number still using comma instead of dot.
- The type of numeric variables are still object, not float.
- And also, for the date format in the 'transaction' dataset, we would change the format so we can use it for the time series later on.

```
customer['Income'] = customer['Income'].str.replace(',', '.')
customer['Income'] = customer['Income'].astype(float)
store['Longitude'] = store['Longitude'].str.replace(',', '.')
store['Longitude'] = store['Longitude'].astype(float)
store['Latitude'] = store['Latitude'].str.replace(',', '.')
store['Latitude'] = store['Latitude'].astype(float)
transaction['Date'] = pd.to_datetime(transaction['Date'], format='%d/%m/%Y')
```

DATA COLLECTION

FOREIGN KEY AND PRIMARY KEY

Product

ProductID	Unique ID for every product
Product Name	Name of the Product
Price	The product's price

Store

StoreID	Unique ID for every store
StoreName	Name of the Store
GroupStore	Store's Group Name
Type	Modern / General Trade
Latitude	Store's Latitude
Longitude	Store's Longitude

Customer

CustomerID	Unique ID for every customer
Age	Customer's age
Gender	0 for female, 1 for male.
Marital Status	Married or Single
Income	Customer's income

Transaction

TransactionID	Unique ID for every transaction
CustomerID	Unique ID for every customer
Date	Transaction's date
ProductID	Unique ID for every product
Price	Product's price per item
Qty	Number of items purchased
TotalAmount	Price * Qty
StoreID	Unique ID for every store

DATA MERGING

MERGE DATASETS AND CHECK DATA IRREGULARITIES

Merging datasets are based on the foreign keys. It is known that transaction has foreign keys for every other datasets, so we would use transaction as the “main dataset”.

```
tr_customer = pd.merge(left = transaction,
                       right = customer,
                       left_on = 'CustomerID',
                       right_on = 'CustomerID',
                       how = 'left')

tr_cst_pro = pd.merge(left = tr_customer,
                      right = product,
                      left_on = ['ProductID', 'Price'],
                      right_on = ['ProductID', 'Price'],
                      how = 'left')

df_merge = pd.merge(left = tr_cst_pro,
                    right = store,
                    left_on = 'StoreID',
                    right_on = 'StoreID',
                    how = 'left')
```

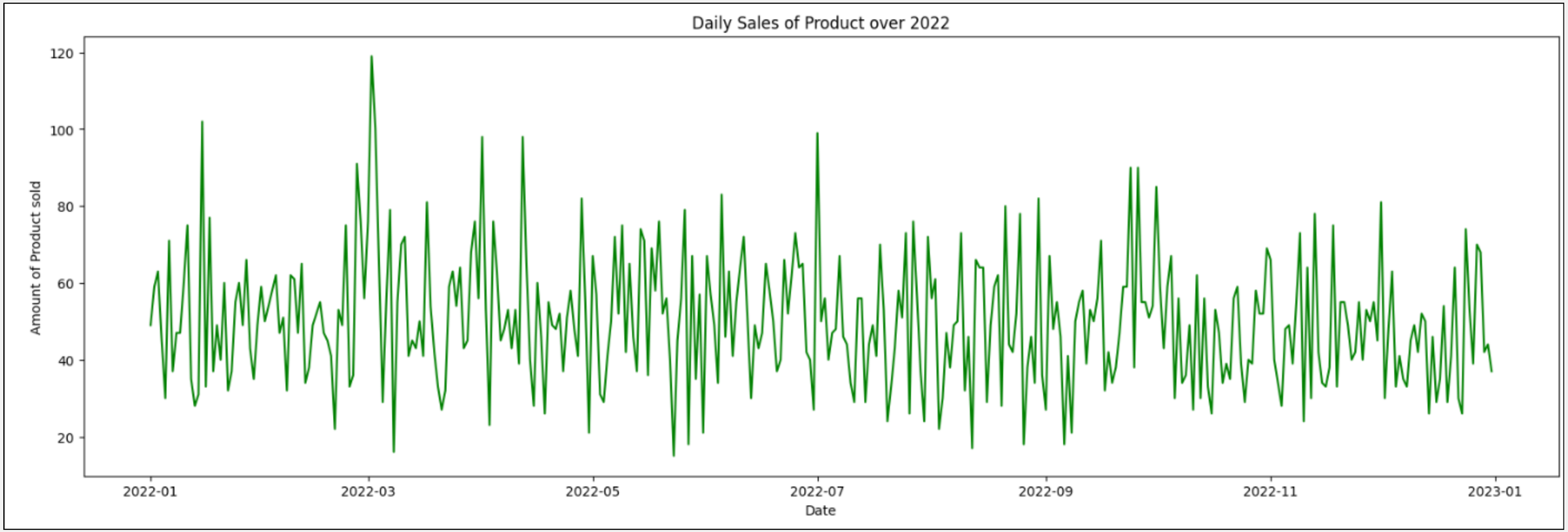
For data irregularities, we would drop the duplicated values to ensure there is no double or more same transactions being repeated.

```
df_merge = df_merge.drop_duplicates()
df_merge.duplicated().sum()
```

	TransactionID	CustomerID	Date	ProductID	Price	Qty	TotalAmount	StoreID	Age	Gender	MaritalStatus	Income	ProductName	StoreName	GroupStore	Type	Latitude	Longitude
0	TR11369	328	2022-01-01	P3	7500	4	30000	12	36	0	Married	10.53	Crackers	Prestasi Utama	Prestasi	General Trade	-2.990934	104.756554
1	TR16356	165	2022-01-01	P9	10000	7	70000	1	44	1	Married	14.58	Yoghurt	Prima Tendean	Prima	Modern Trade	-6.200000	106.816666
2	TR1984	183	2022-01-01	P1	8800	4	35200	4	27	1	Single	0.18	Choco Bar	Gita Ginara	Gita	General Trade	-6.966667	110.416664

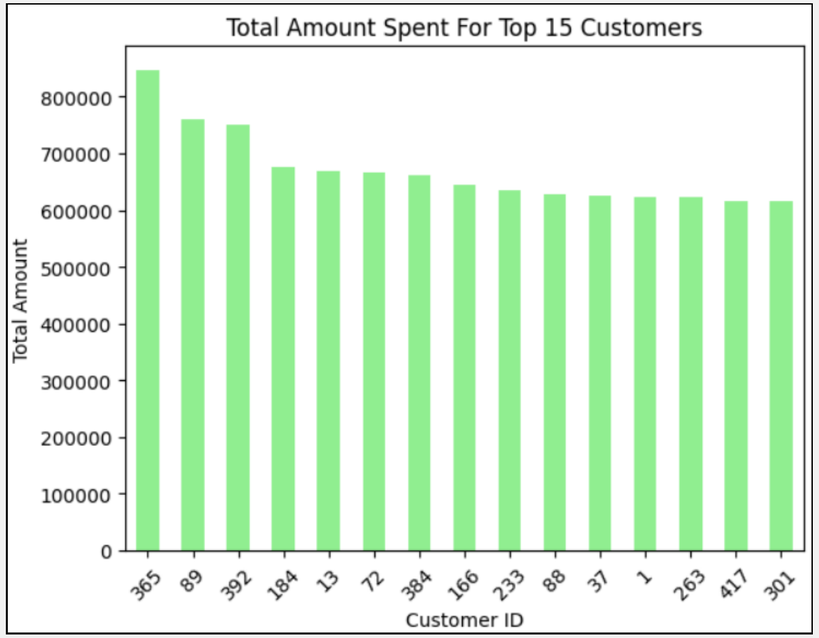
EXPLORATORY DATA ANALYSIS

Quantity vs Date

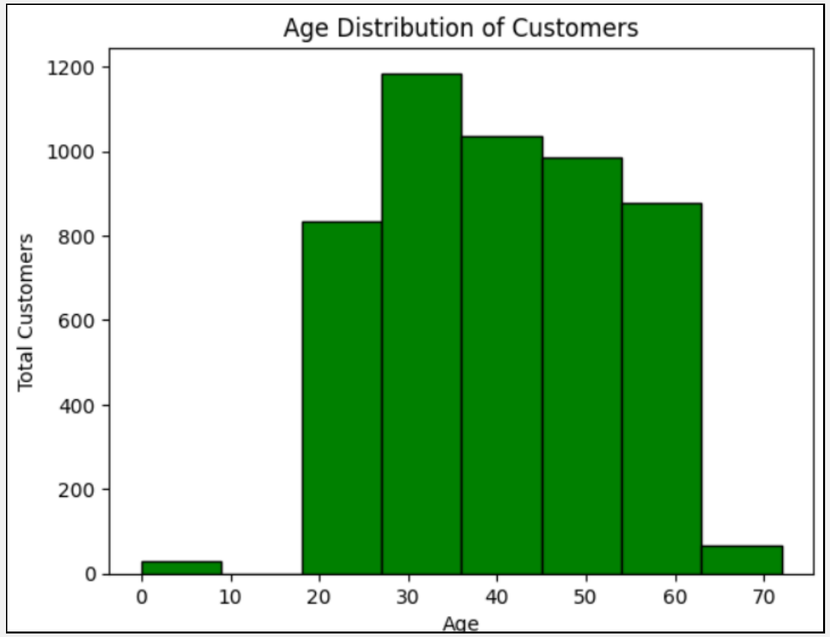


EXPLORATORY DATA ANALYSIS

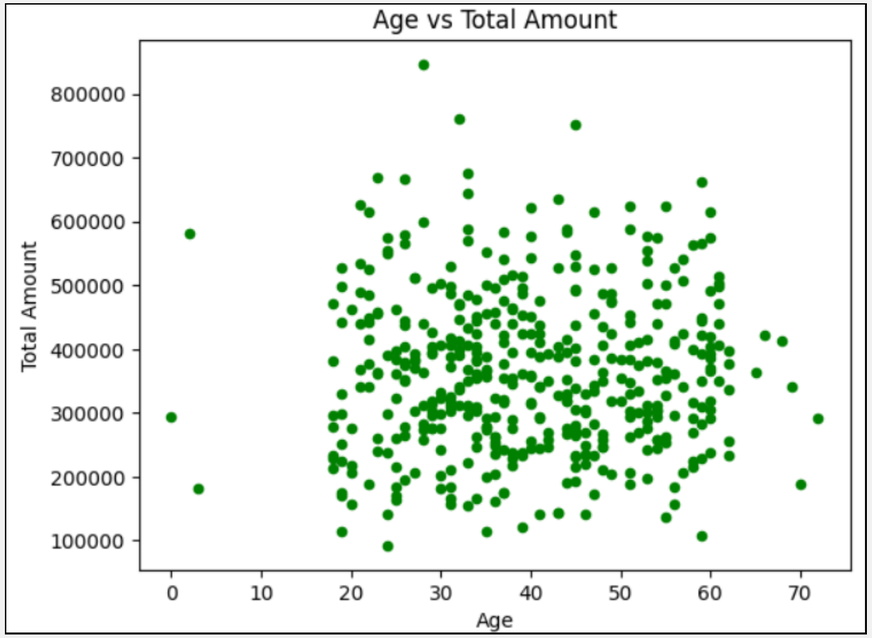
CustomerID and Age vs Total Amount



IT SHOWS THAT THERE IS A CUSTOMER THAT EXCEEDS RP 800,000.00 FOR IT'S TRANSACTIONS ON 2022.



WITH THE RANGE OF 8, MOSTLY THE CUSTOMERS IS AROUND 30 YEARS OLD.

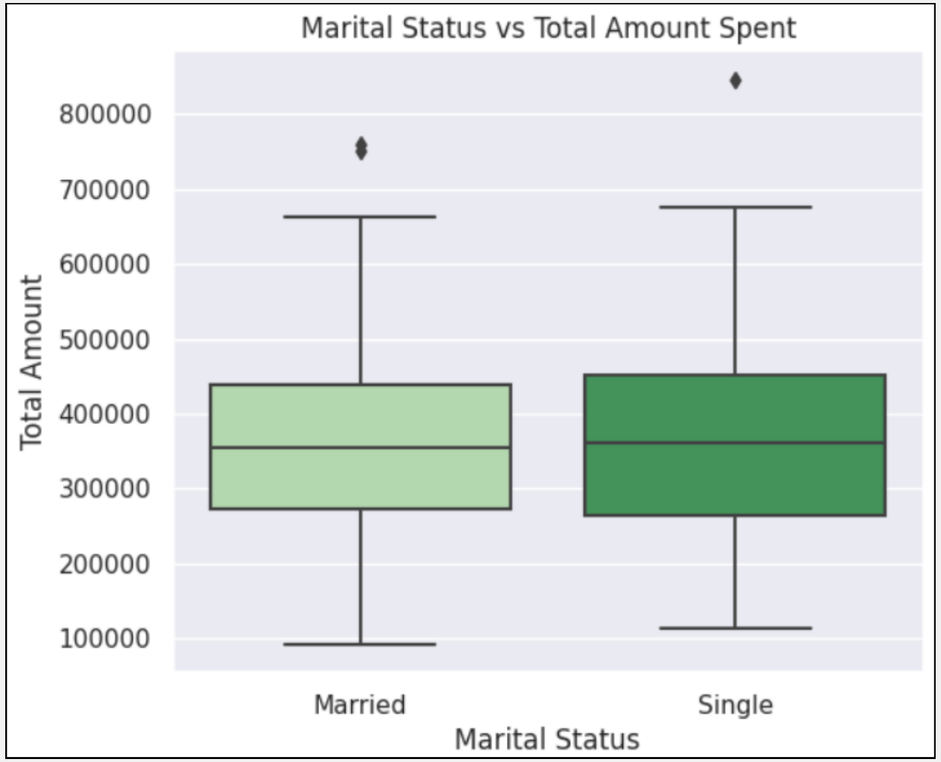


CUSTOMER WHO PURCHASED EXCEEDS 800K IS PART OF THE CUSTOMER WHO'S AGE IS AROUND 30 YEARS OLD.

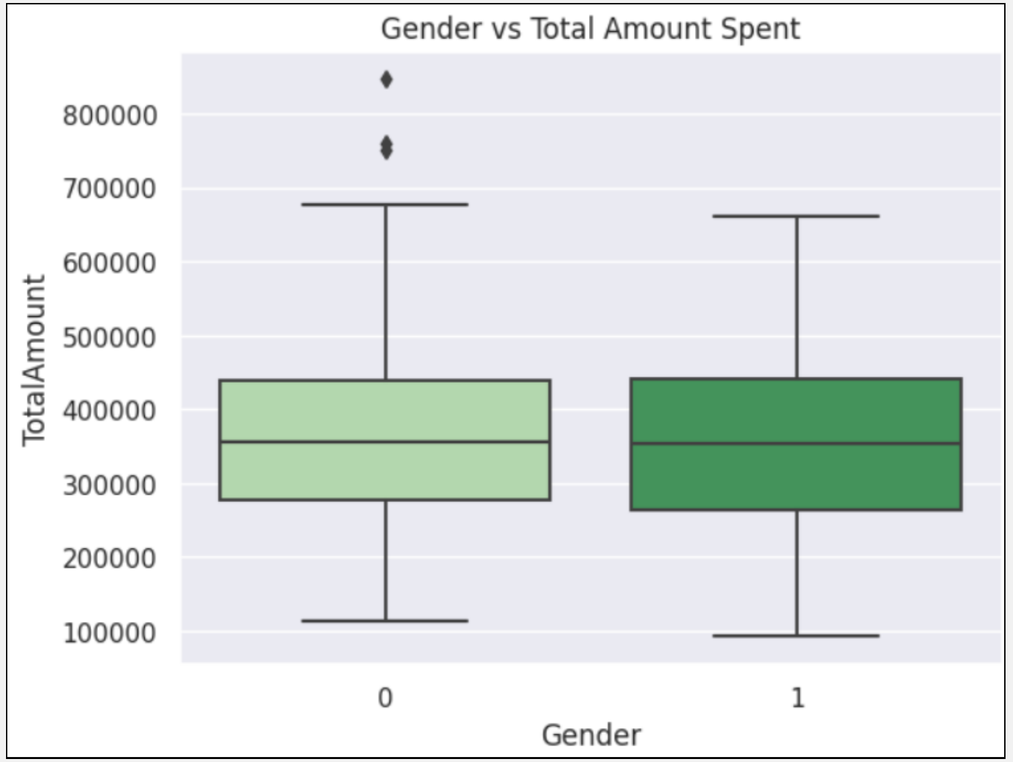
EXPLORATORY DATA ANALYSIS

BOXPLOT

*OUTLIERS ARE NOT GONNA BE REMOVED SINCE IT'S JUST TOTAL AMOUNT



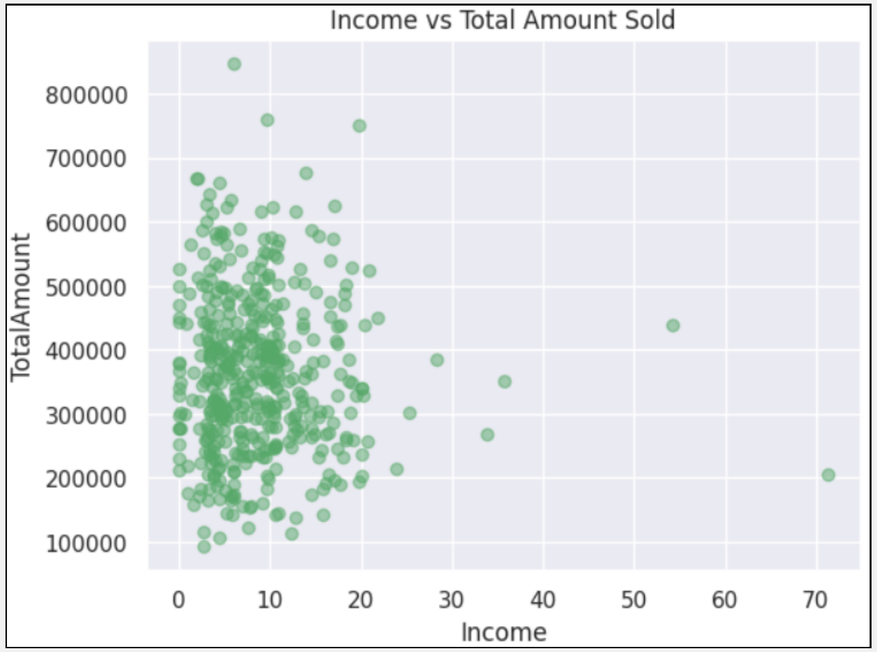
CUSTOMER WHO IS SINGLE TEND TO PURCHASE MORE THAN WHO HAS MARRIED.



FEMALE CUSTOMERS TEND TO PURCHASE MORE THAN MALE CUSTOMERS.

EXPLORATORY DATA ANALYSIS

COMPARISONS



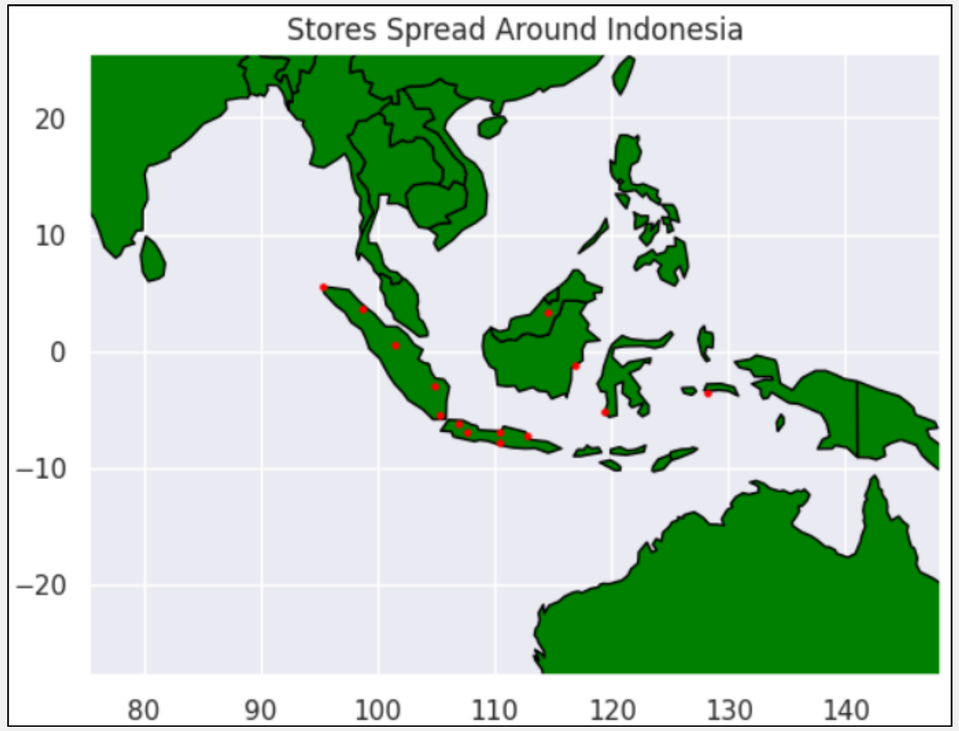
CUSTOMERS WITH LOWER INCOME TEND TO PURCHASE ITEMS MUCH MORE THAN WHO HAS HIGHER INCOME.

CHEAPER PRODUCT TENDS TO BE MORE FAVORABLE THAN PRODUCTS THAT COST MORE.

THE TOTAL AMOUNT OF SALES ON EVERY WEEK IS APPROXIMATELY THE SAME.

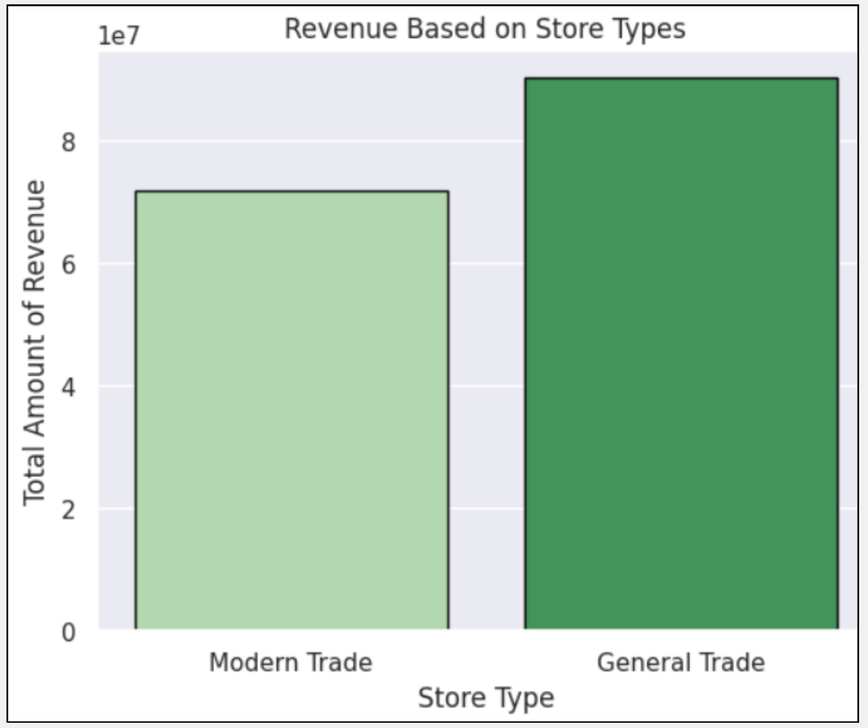
EXPLORATORY DATA ANALYSIS

STORE LOCATION & REVENUE



THE STORE LOCATION MOSTLY ON SUMATERA AND JAVA ISLAND. WHILE INI NTB, NTT, AND PAPUA ISLAND ARE STILL NOT OPERATED.

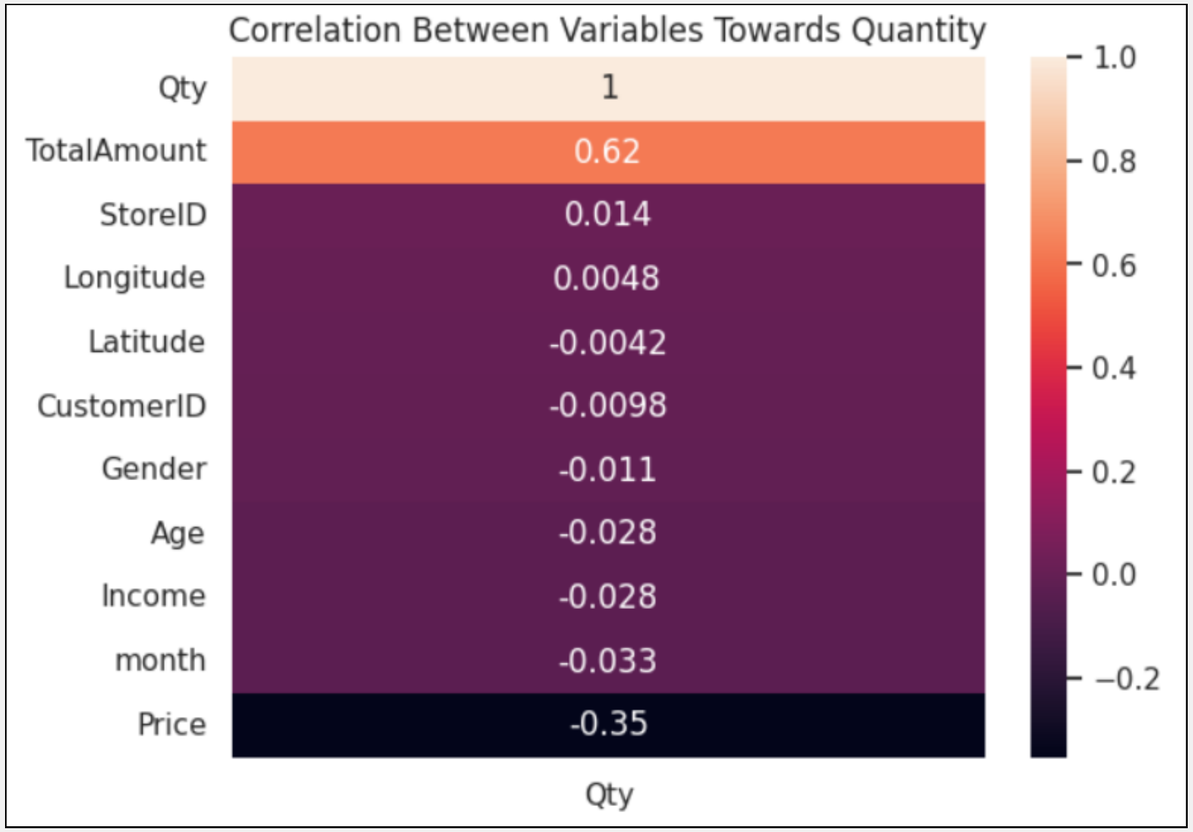
THE GENERAL TRADE STORE STILL LEADING IN TERMS OF REVENUE RATHER THAN MODERN TRADE.



GENERAL TRADE, DUE TO ITS ACCESSIBILITY AND PROXIMITY TO RESIDENTIAL ZONES, CAN BE MORE ADVANTAGEOUS FOR CLIENTS. MODERN TRADE, HOWEVER, TYPICALLY PROVIDES A MORE EXCELLENT CHOICE OF GOODS AND BETTER CUSTOMER SUPPORT.

EXPLORATORY DATA ANALYSIS

CORRELATIONS



- THE HIGHEST CORRELATION WITH QUANTITY GOES TO TOTALAMOUNT WITH THE SCORE OF 0.62. THIS SCORE IS ACTUALLY NOT REALLY HIGH YET NOT REALLY LOW.
- THE OTHER VARIABLES SEEMS TO HAVE A VERY LOW CORRELATION (ALMOST NO CORRELATION) WITH QUANTITY AND RELATIVELY NEGATIVE.

TIME

analysis & forecasting

SERIES

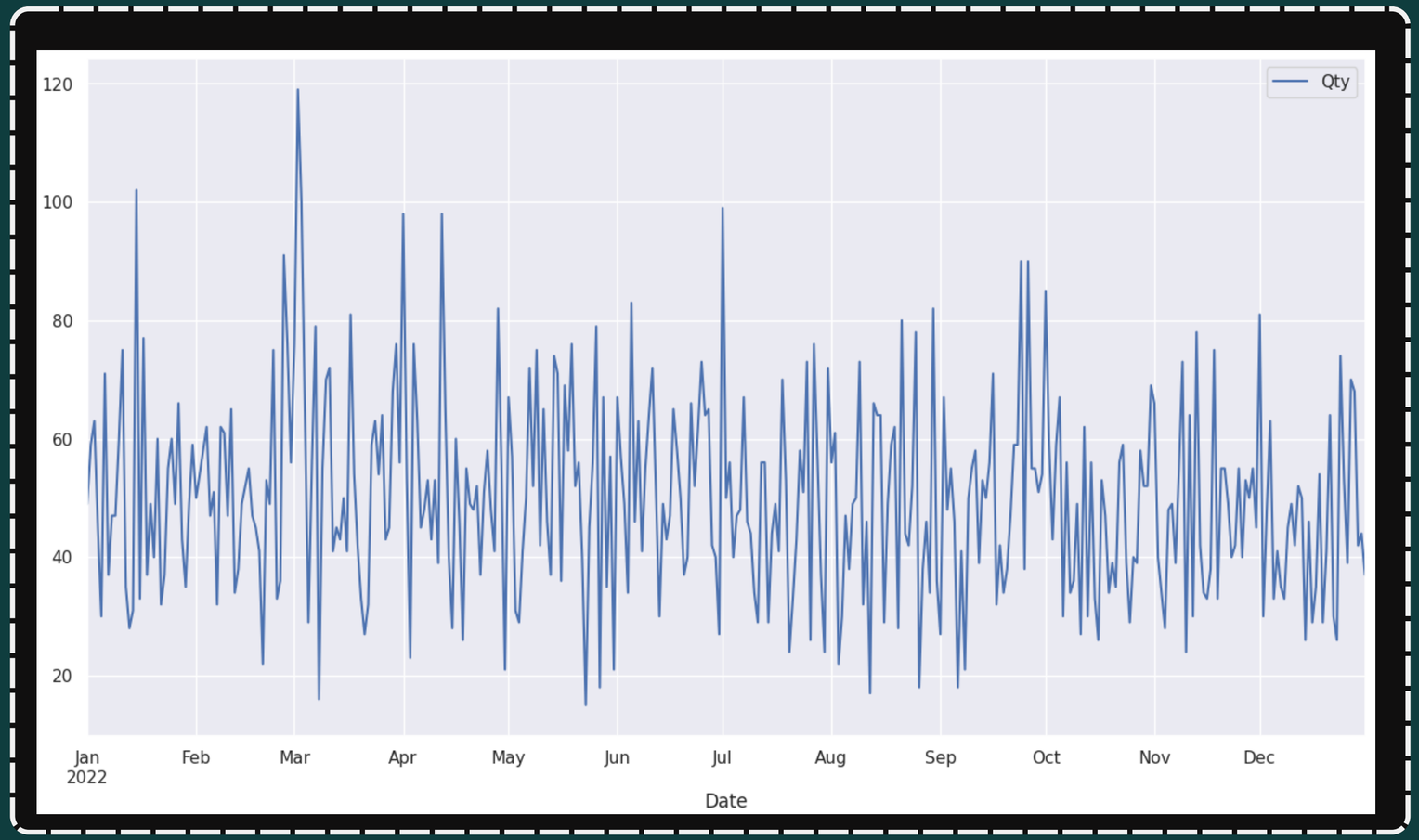
TIME SERIES : ARIMA

DATE AGGREGATIONS

AGGREGATION IS DONE BY GROUPING THE DATE OF 2022 AND SUM THE QUANTITY.

```
df_ts['Date'] = pd.to_datetime(df_ts['Date'])
df_ts.set_index('Date', inplace=True)
df_qty = df_ts.groupby('Date').agg({'Qty':'sum'})
```

THE QUANTITY TOWARDS THE Q4 ARE NOT AS HIGH AS THE Q1 WHICH WOULD BE AFFECTED BY SOME UNKOWN FACTORS, SUCH AS INFLATION RATE OR ECONOMIC RESESSION, AND MAY FROM THE MARKETING OF THE PRODUCT.



TIME SERIES : ARIMA

STATIONARITY

TIME SERIES DATA SHOULD BE STATIONARY

hipotesis

- H0: IT IS A STATEMENT ABOUT THE POPULATION THAT EITHER IS BELIEVED TO BE TRUE OR IS USED TO PUT FORTH AN ARGUMENT UNLESS IT CAN BE SHOWN TO BE INCORRECT BEYOND A REASONABLE DOUBT.
- H1:IT IS A CLAIM ABOUT THE POPULATION THAT IS CONTRADICTORY TO H0

USE ADFULLER TEST

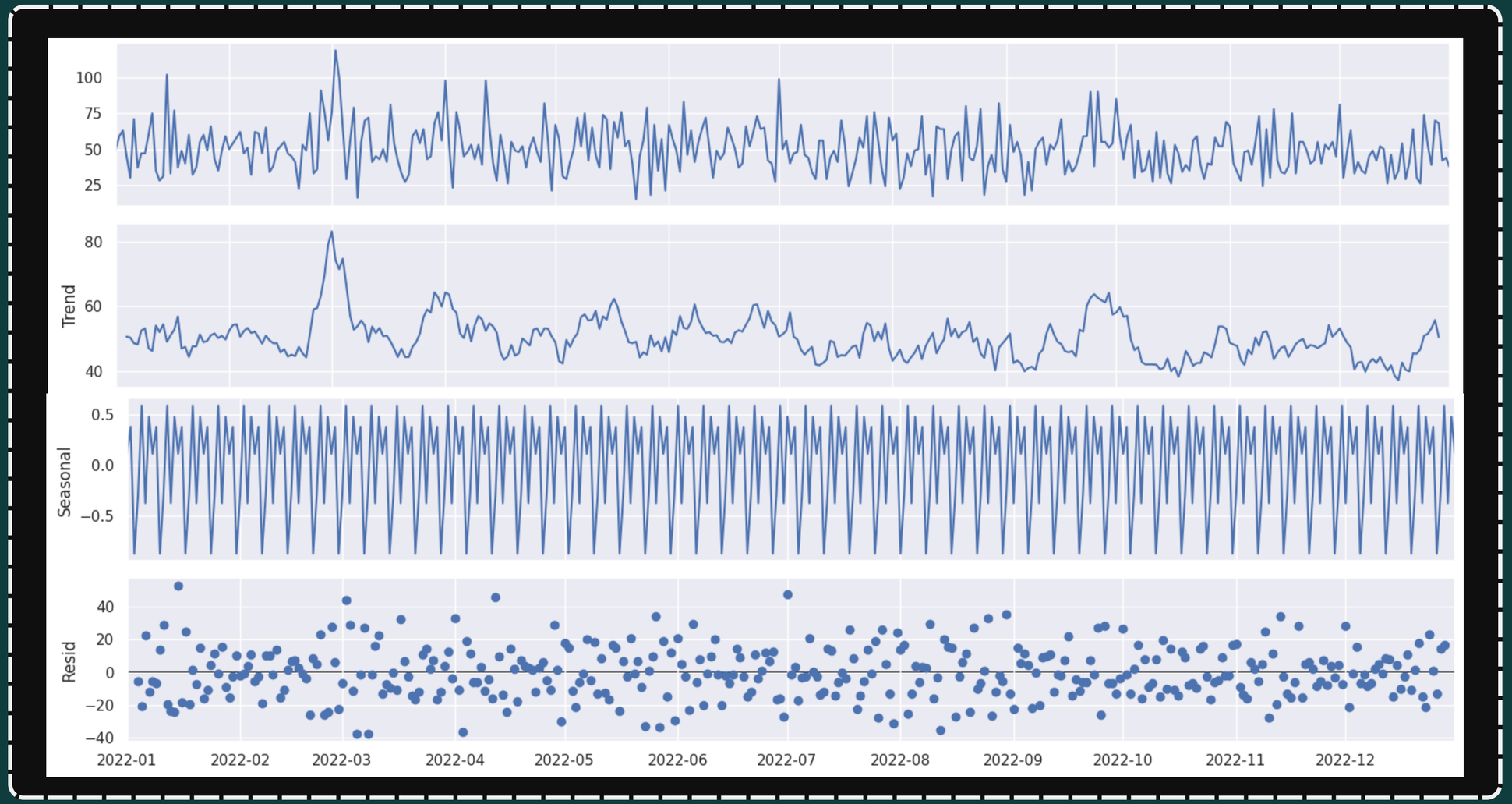
```
ADF Test Statistic : -19.448086319449082
p-value : 0.0
#Lags Used : 0
Number of Observations : 364
strong evidence against the null hypothesis(Ho), reject the null hypothesis. Data is stationary
```

kesimpulan

- P-VALUE FALLS UNDER ALPHA / SIGNIFICANCE LEVEL (5%), IT MEANS THAT THE DATA IS STATIONARY

TIME SERIES : ARIMA

SEASONALITY DIFFERENCES



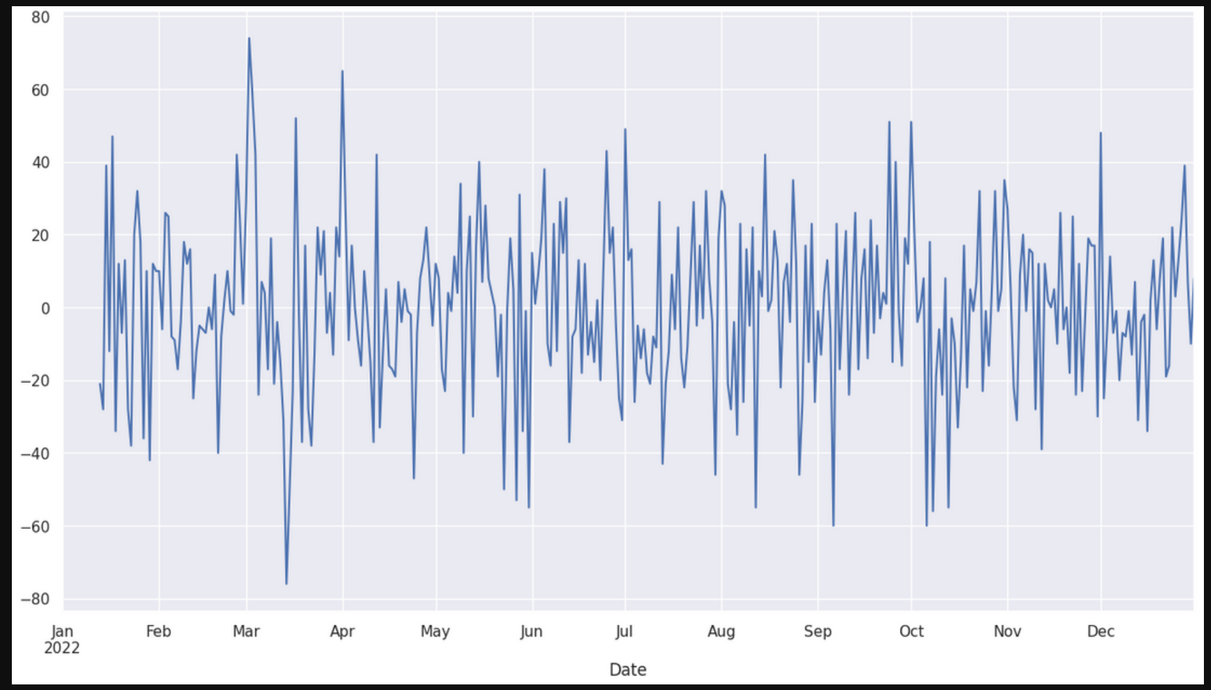
TIME SERIES : ARIMA

SEASONALITY DIFFERENCES

WE WOULD COMPARE BETWEEN THE QUANTITY FIRST DIFFERENCE AND SEASONAL FIRST DIFFERENCE.

```
ADF Test Statistic : -8.67956366792047
p-value : 4.322883783680308e-14
#Lags Used : 11
Number of Observations : 341
strong evidence against the null hypothesis(Ho), reject the null hypothesis. Data is stationary
```

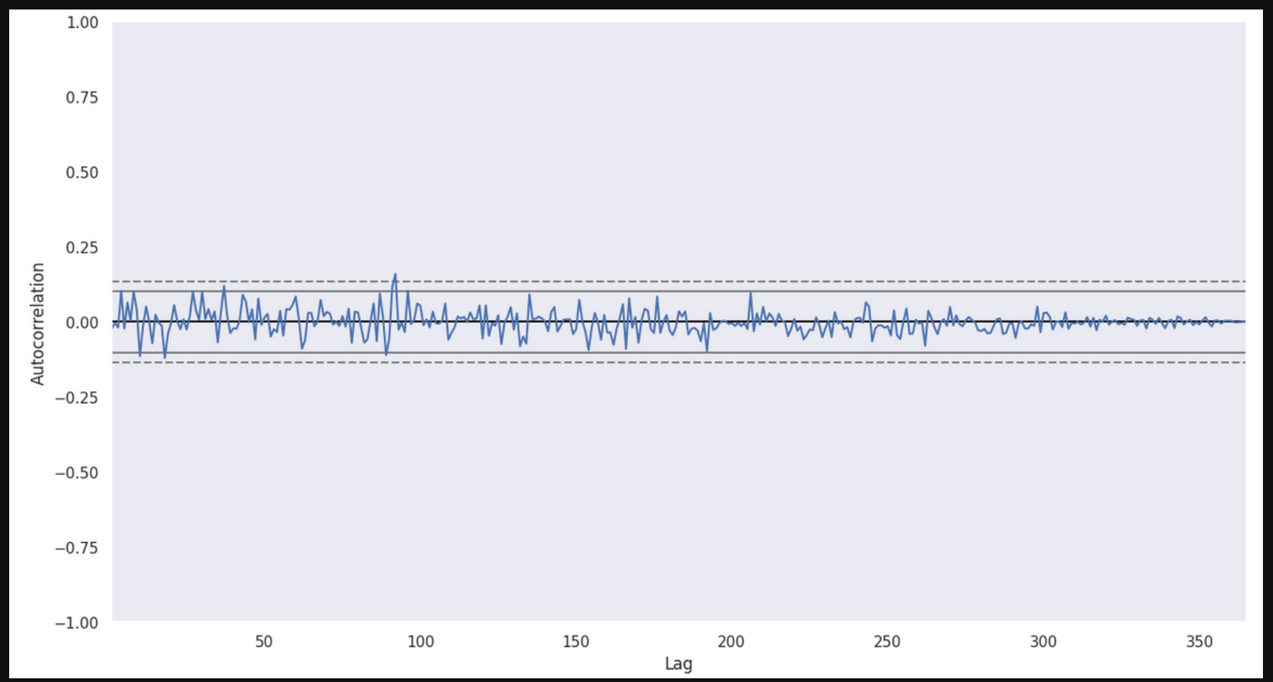
Date	Qty	Quantity First Difference	Seasonal First Difference
2022-01-01	49	NaN	NaN
2022-01-02	59	10.0	NaN
2022-01-03	63	4.0	NaN
2022-01-04	45	-18.0	NaN
2022-01-05	30	-15.0	NaN



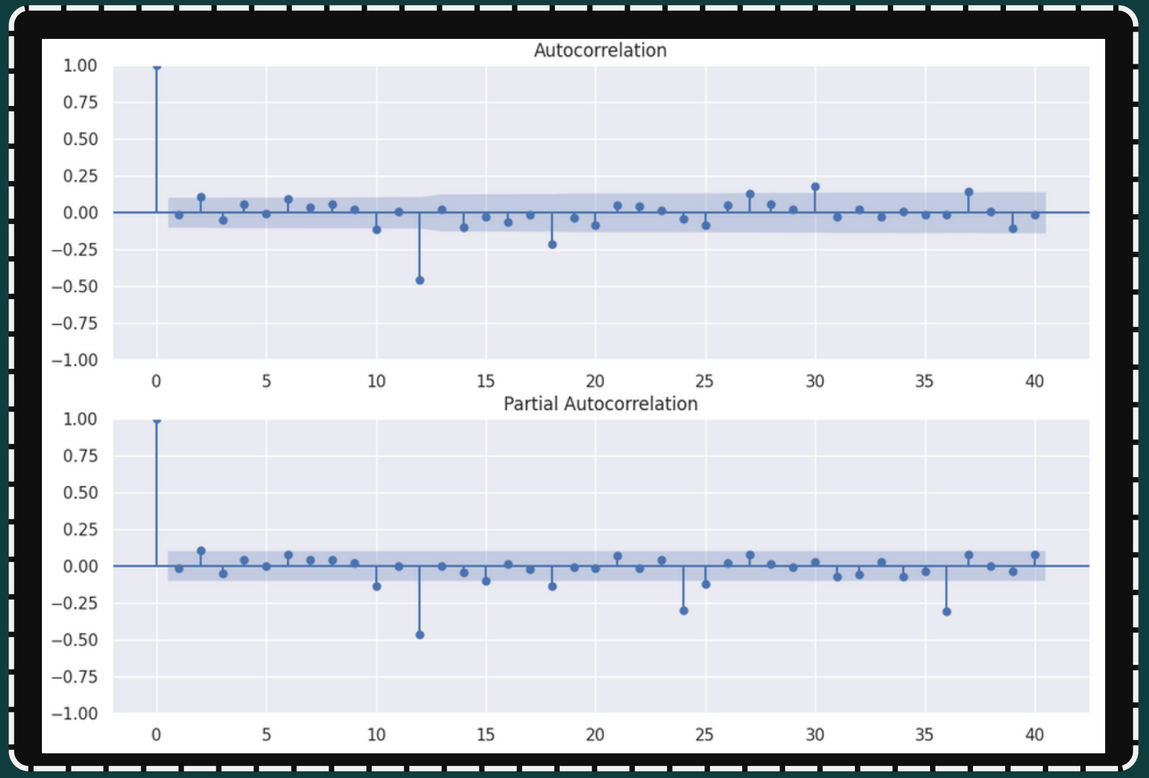
WITH THE FIRST DIFFERENCE AND SEASONAL FIRST DIFFERENCE, THE DATA IS STILL STATIONARY, WHICH MEANS THIS IS CONSISTENT.

TIME SERIES : ARIMA

AUTOCORRELATION



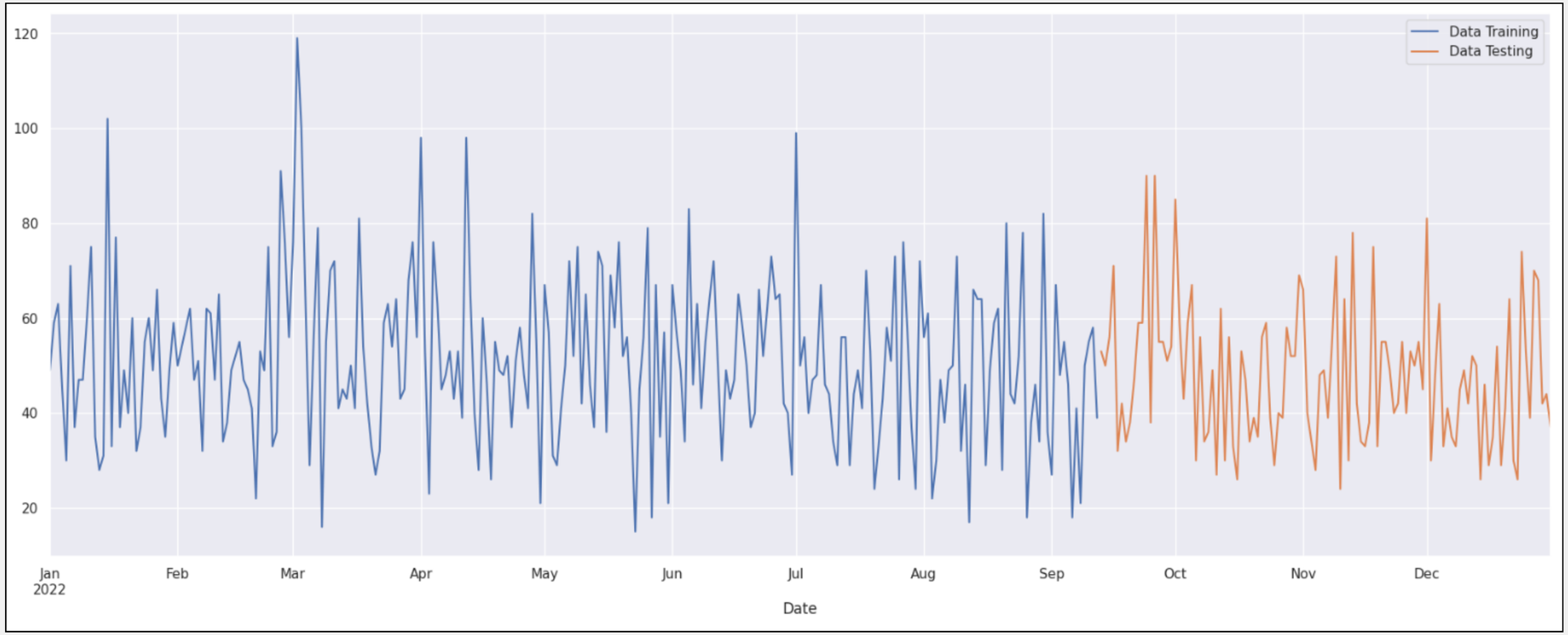
IT HAS POSITIVE CORRELATION AND IT HAS A LOW AUTOCORRELATION RATE.



FROM THE OUTPUT, THE THEORETICAL PACF DOES NOT SHUT OFF, BUT INSTEAD TAPERS TOWARD 0 IN SOME MANNER. A CLEARER PATTERN FOR THE MODEL IS IN THE ACF. THE ACF WILL HAVE NON-ZERO AUTOCORRELATIONS ONLY AT LAGS INVOLVED IN THE MODEL.

ARIMA MODEL

TRAIN AND TEST SPLIT



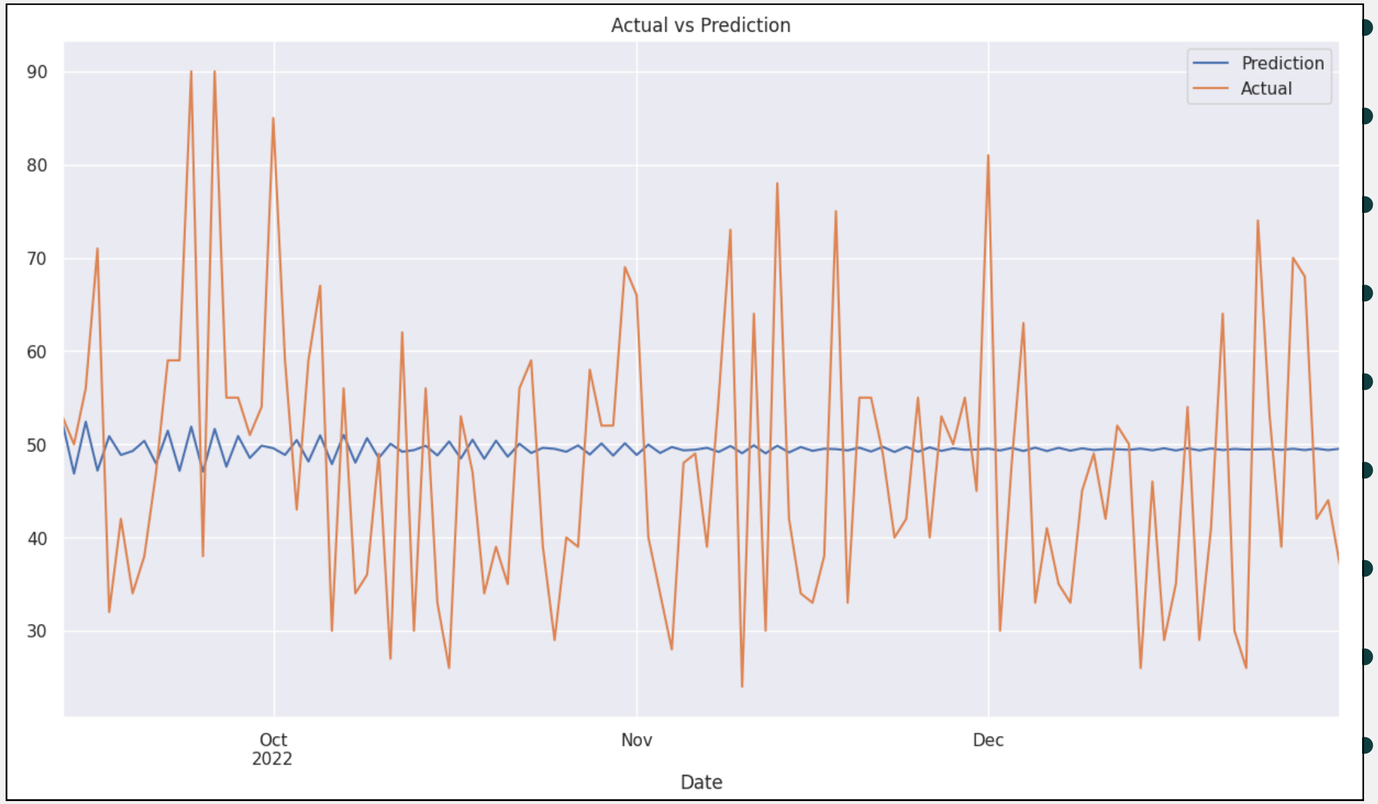
TRAIN AND TEST SPLIT IN TIME SERIES IS KINDA DIFFERENT FROM USUAL ONE. WE WOULD SPLIT IT BASED ON TIME / DATE. TRAINING DATA WOULD BE THE FIRST ONE AND THE REST GOES TO TESTING DATA.

ARIMA MODEL

MODEL FITTING

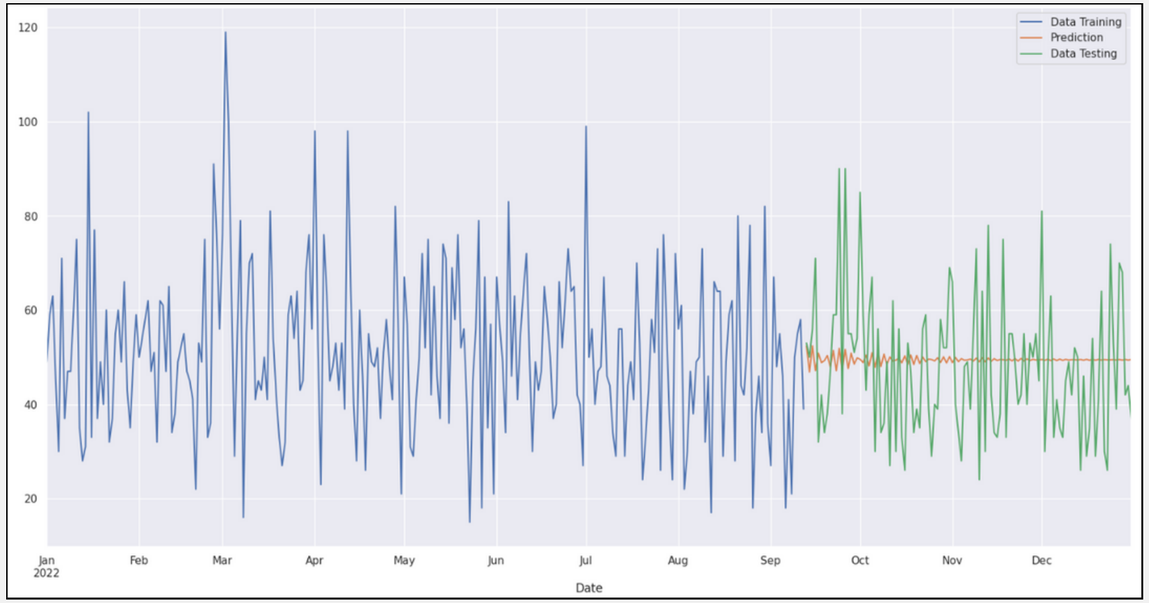
```
Best model:  ARIMA(4,1,3)(0,0,0)[0]
Total fit time: 39.444 seconds
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  255
Model:               SARIMAX(4, 1, 3)   Log Likelihood               -1081.451
Date:                Fri, 29 Sep 2023   AIC                           2178.903
Time:                        06:38:09   BIC                           2207.202
Sample:                    01-01-2022   HQIC                          2190.287
                         - 09-12-2022
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.8409      0.066    -27.766      0.000      -1.971      -1.711
ar.L2         -1.0066      0.113     -8.944      0.000      -1.227      -0.786
ar.L3         -0.1992      0.131     -1.520      0.128      -0.456       0.058
ar.L4         -0.1100      0.072     -1.528      0.127      -0.251       0.031
ma.L1          0.8733      0.047     18.592      0.000       0.781       0.965
ma.L2         -0.8844      0.035    -25.428      0.000      -0.953      -0.816
ma.L3         -0.9422      0.044    -21.191      0.000      -1.029      -0.855
sigma2       285.8242     22.208     12.870      0.000     242.297     329.351
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):                22.66
Prob(Q):                              0.91   Prob(JB):                         0.00
Heteroskedasticity (H):               0.80   Skew:                             0.60
Prob(H) (two-sided):                  0.32   Kurtosis:                         3.84
```

MODEL PREDICTING



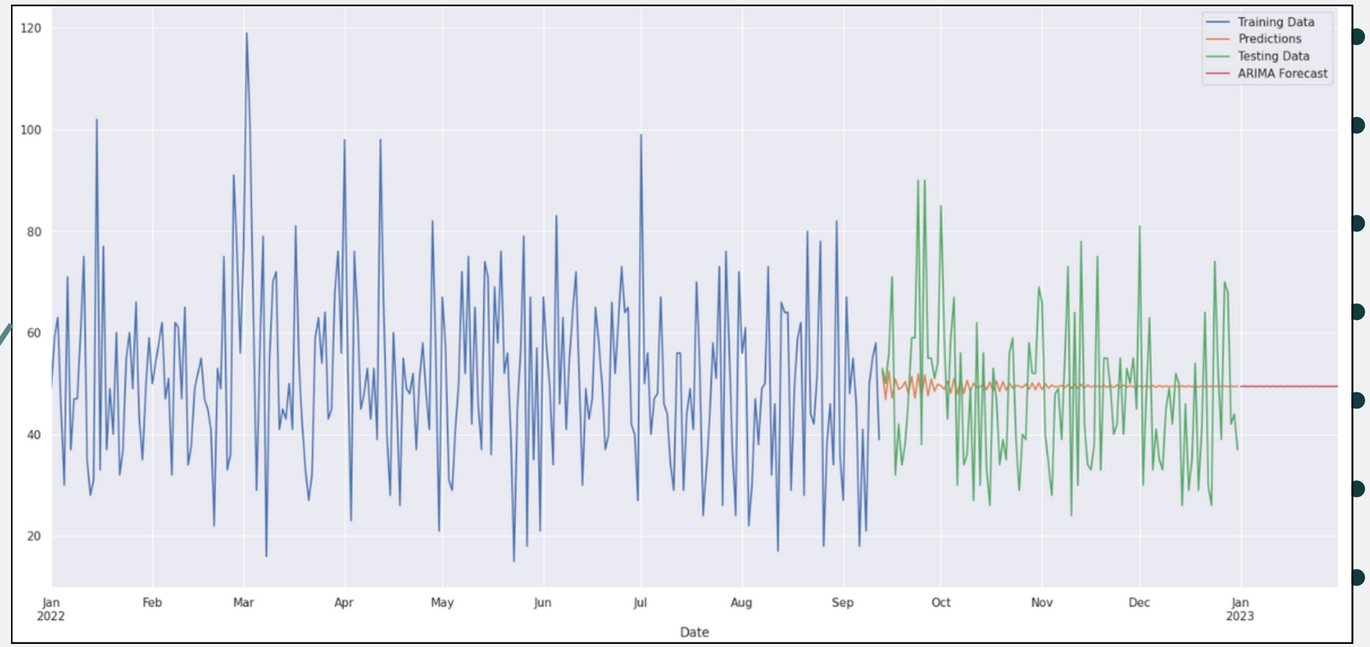
THE RESULT FOR THE MODEL PREDICTING ON TESTING DATA SHOWS A SIGNIFICANTLY DIFFERENT RESULTS. WHILE THE DATA ACTUAL DATA BEING FLUCTUATIVE, THE PREDICTION CONSTANTLY BEING AROUND 50.

ARIMA MODEL



PREDICTING TESTING DATA

- ON THE LEFT IS THE RESULT BETWEEN THE TRAINING DATA, TESTING DATA, AND PREDICTION
- BELOW IS THE RESULT WITH AN ADDITION OF ARIMA FORECASTING



- THE ARIMA FORECASTING RESULT SHOWS THE SAME TREND LINE AS THE PREDICTION WHICH IS AROUND 50.
- FUTURE WORKS WOULD BE ABLE TO IMPLEMENT HYPERPARAMETER TUNING OR OTHER METHODS TO BE ABLE TO SEE THE PREDICTION AND ARIMA FORECASTING RESULT IS AROUND THE SAME AS TESTING DATA

ARIMA MODEL

MODEL EVALUATION

```
MAPE: 28.63%
MAE: 12.05
MSE: 216.95
RMSE: 14.73
Explained Variance: 0.00
```

1. **MAPE (Mean Absolute Percentage Error): 28.63%**
   - MAPE measures the average percentage difference between predicted and actual values. A MAPE of 28.63% indicates that, on average, your model's predictions are off by almost 29%, which is quite high. It suggests that the model is not accurate in capturing the underlying patterns in the time series data.
2. **MAE (Mean Absolute Error): 12.05**
   - MAE represents the average magnitude of errors between predicted and actual values. An MAE of 12.05 suggests that, on average, your model's predictions are off by approximately 12.05 units. While this metric is more interpretable than MAPE, it still indicates a significant prediction error.
3. **MSE (Mean Squared Error): 216.95**
   - MSE measures the average squared difference between predicted and actual values. An MSE of 216.95 indicates that the model's errors are large and vary widely. This could be due to a lack of accuracy in capturing the time series patterns.
4. **RMSE (Root Mean Squared Error): 14.73**
   - RMSE is the square root of MSE and provides a measure of the average magnitude of errors in the same units as the original data. An RMSE of 14.73 suggests that, on average, your model's predictions deviate from the actual values by approximately 14.73 units.
5. **Explained Variance: 0.0**
   - Explained variance (R-squared) is a measure of how well the model explains the variance in the data. An explained variance of 0.0 indicates that the model is not explaining any of the variance in the data. In other words, the model is not capturing the underlying patterns or trends in the time series.

CUSTOMER

clustering with k-means

SEGMENTATION

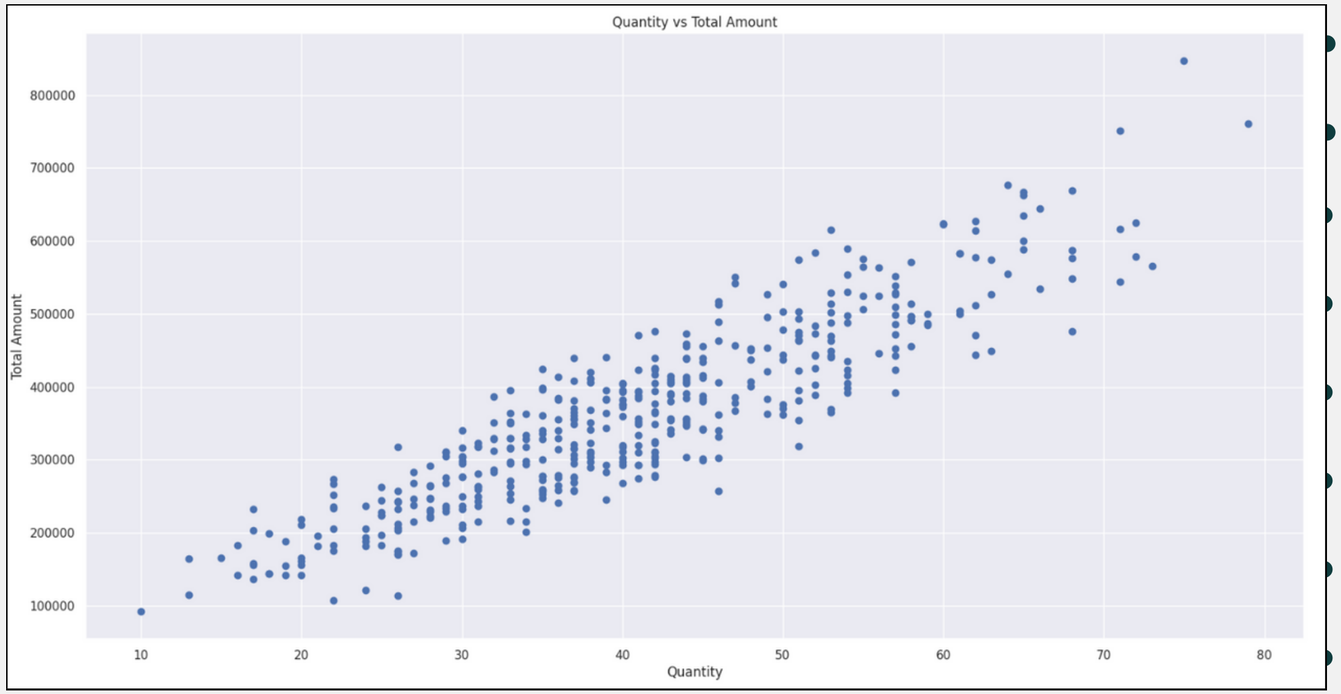
CLUSTERING

MERGING DATA

```
df_cluster = pd.merge(left = transaction,
                      right = customer,
                      left_on = 'CustomerID',
                      right_on = 'CustomerID',
                      how = 'left')

df_cluster = pd.merge(left = df_cluster,
                      right = product,
                      left_on = ['ProductID', 'Price'],
                      right_on = ['ProductID', 'Price'],
                      how = 'left')

df_cluster = pd.merge(left = df_cluster,
                      right = store,
                      left_on = 'StoreID',
                      right_on = 'StoreID',
                      how = 'left')

df_cluster = df_cluster.drop_duplicates()
```

AGGREGATING DATA

CustomerID	TransactionID	Qty	TotalAmount
1	17	60	623300
2	13	57	392300
3	15	56	446200

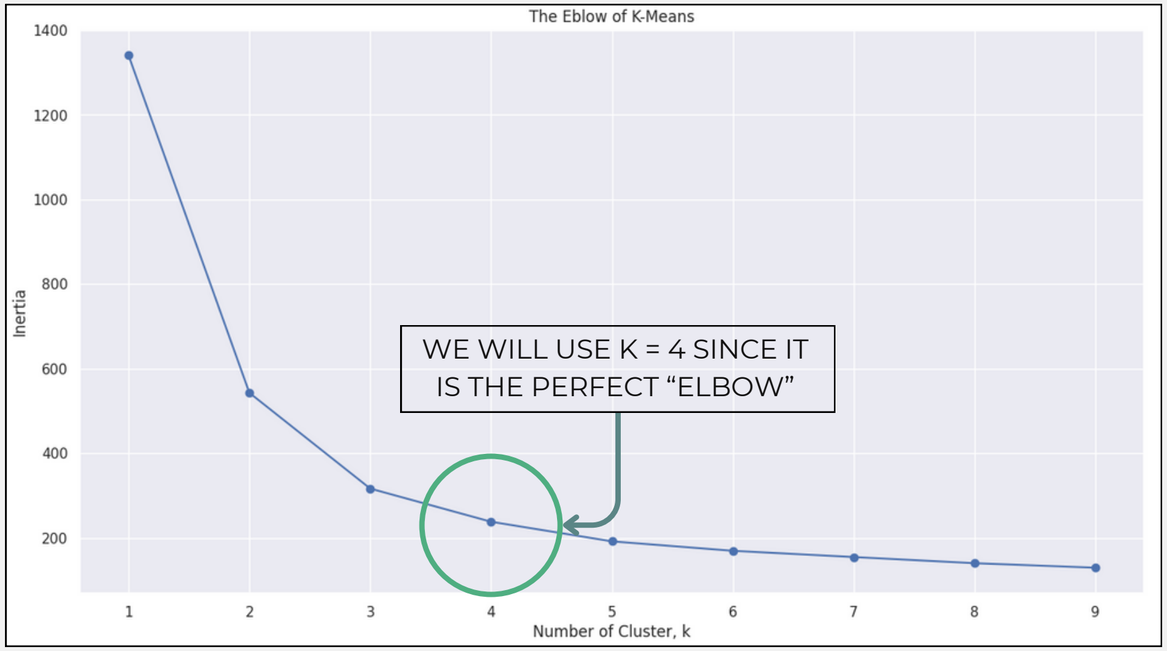


K-MEANS

STANDARDIZE DATA

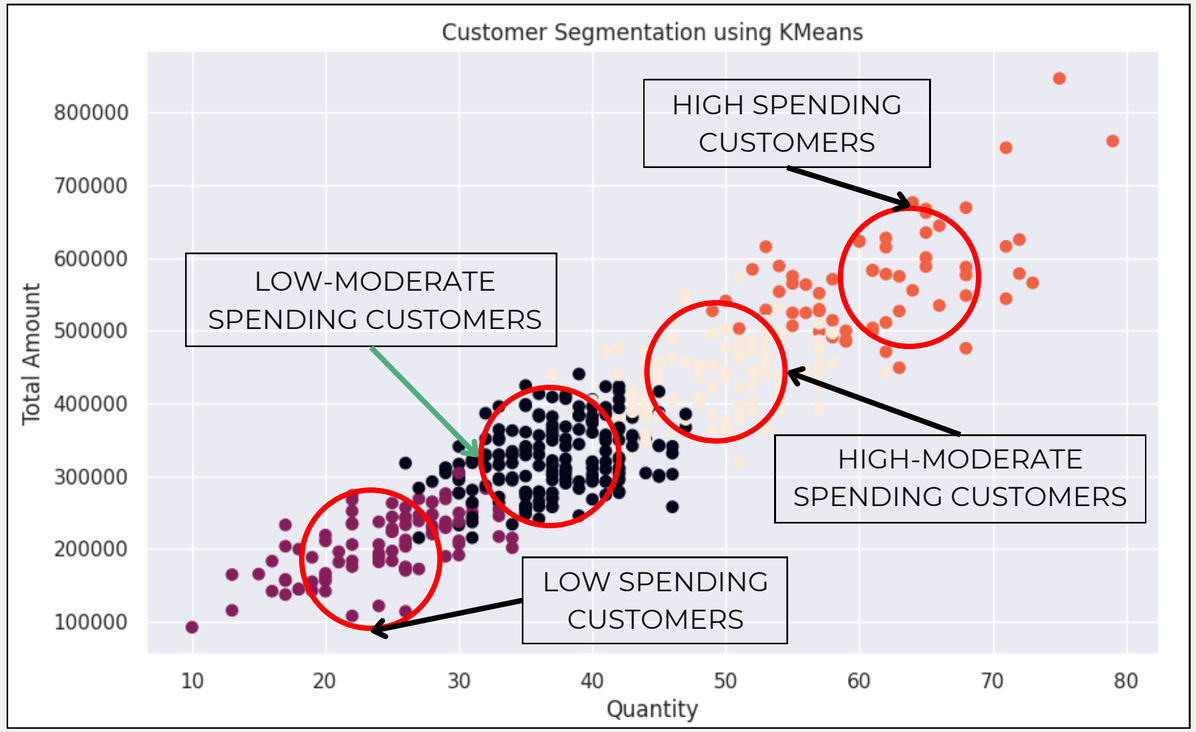
```
sc = StandardScaler()
sc.fit(df_cluster)
df_scaled = sc.transform(df_cluster)
```

FIND K USING ELBOW METHOD

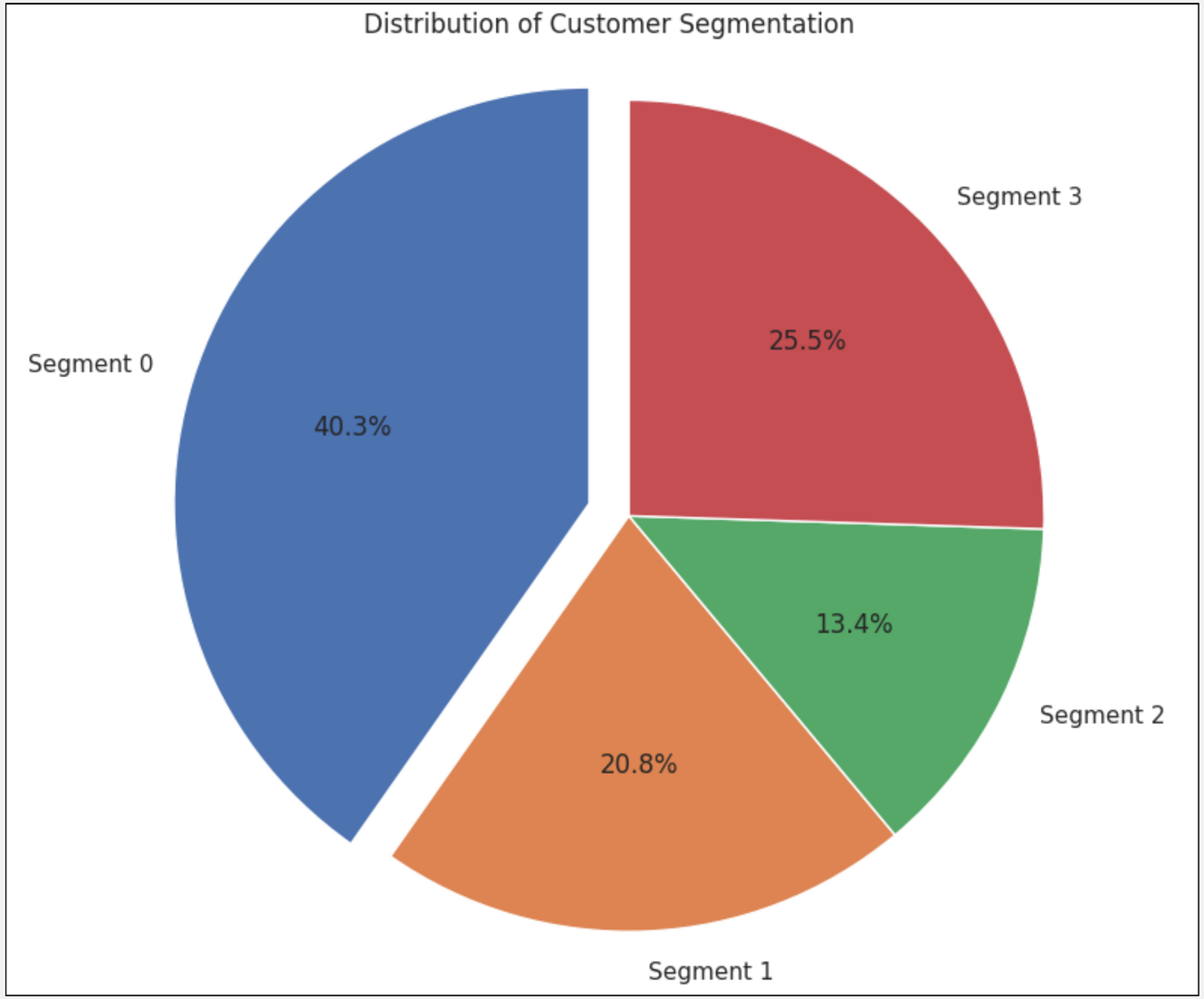


MODEL FITTING & PREDICTING

```
KMeans
KMeans(n_clusters=4, random_state=50)
```



CUSTOMER SEGMENTATION



pie chart report

- Low spending customer as “Segment 0” is being the highest than the other customer segmentations.
- High spending customers as “Segment 3” is being the second highest with a total of 25.5% out of the total.
- Meanwhile, the low-moderate and high-moderate customers are only respectively 13.4% and 20.8%.

conclusion

In summary, the provided metrics indicate that the ARIMA model is not performing well, and you should explore various strategies to improve its accuracy, including model selection, hyperparameter tuning, and data preprocessing. It's important to iterate and experiment with different approaches to find the best forecasting solution for the specific time series dataset.

DOCUMENTATION

GITHUB

YOUTUBE

just click the image for the link!!

Rakamin Academy X KALBE Nutritionals

for the opportunities

THANK YOU!

Presented by **Kevin Laurent Oktavian Putra**