ICINDEKILER

1. SPRINT 1

The sales table is a critical part of any business, as it records all of the transactions that have taken place within a given period of time. This table is usually used to track the revenue generated by the business, as well as to identify any trends or patterns that may be emerging in the market.

However, it's important to note that almost half of the sales table is often empty. This can be due to a variety of factors, such as a slow period for the business or a lack of sufficient data. Regardless of the reason, it's important for businesses to be aware of this issue and to work towards filling in these empty values.

In addition to the sales table, there are several other tables that are commonly used in businesses to track various types of data. For example, the returns list table is used to track all of the products that have been returned by customers. Similarly, the product reviews table is used to track customer feedback and ratings on various products.

Another table used by the firm is the Amazon Campaigns table, which is used to track the effectiveness of marketing campaigns on Amazon. Finally, the other promotions table is used to track any other promotional activities that the business may be engaged in, such as discounts or special offers.

It's worth noting that many rows in these tables may also contain null values. This can be due to a variety of reasons, such as missing data or incomplete information. As with the empty values in the sales table, it's important for businesses to be aware of these null values and to work towards filling them in as much as possible. This can help to ensure that the data in these tables is accurate and up to date, which can be critical for making informed business decisions.

FEATRUE (COLUMN)	ACTIONS TAKEN
	• Yellow filled cells mean that we dropped the related feature or column.
	• Purple filled cells mean that we added the related feature as a new column.

Table 1:Our Methodology by feature engineering

The methodology we followed during this period examining the datas row by row and at the end get an insight or pre-look about our feature analysis. At this point we marked the features from which we have taken the following actions on the table. If you do not see any table below the explanation, that means we left the table with its main existing columns.

1.1 SALES TABLE

When we analyzed the sales table, the raw data we received contained 190 thousand rows. Almost more than half of these lines consisted of blank lines. As a result of the analysis, all of these blank lines were removed from the data set. Then, data in the form of day, month and year were obtained from the date columns. And values like TotalWeight,DispatchUnitCost were dropped because they include excessively blank or zero rows.

When we analyzed the sales table, the raw data we received contained 190 thousand rows. Almost more than half of these lines consisted of blank lines. As a result of the analysis, all of

these blank lines were removed from the data set. Then, data in the form of day, month and year were obtained from the date columns. And values like TotalWeight,DispatchUnitCost were dropped because they didn't give us an insight. After these processes, new features such as TP_per_Quantity,ST_per_Quantity,UK_CPI,Turkey_CPI,Net_Profit,Day_Time have been added as required.

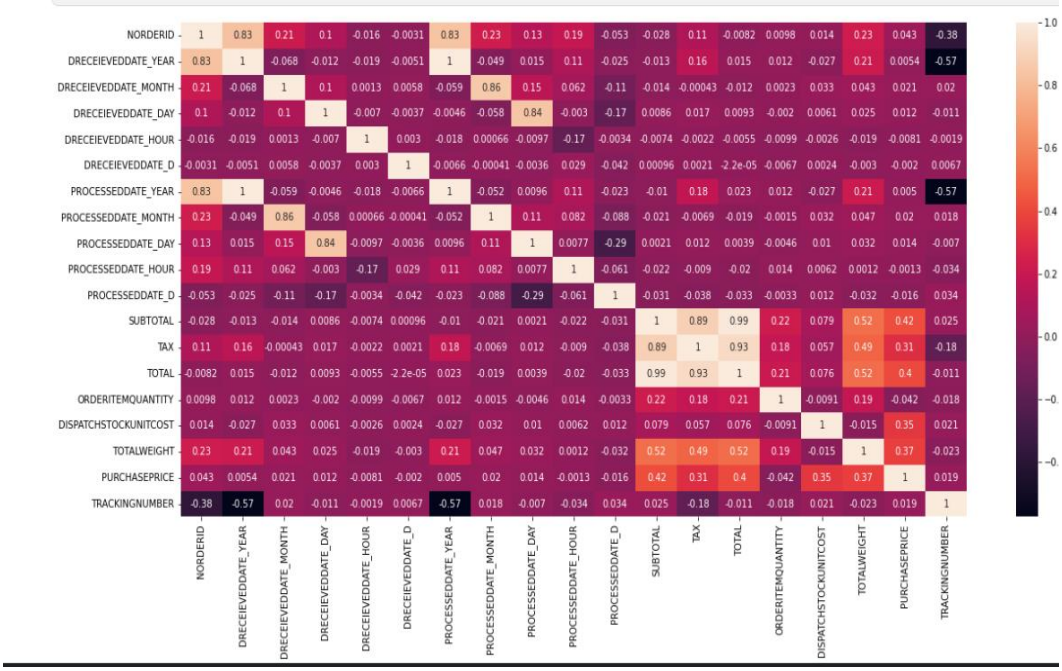


Table 2: Correlation Map of Sales Table

FEATURE (COLUMN)	THE FUNCTION OF COLUMN
nOrderId	• The uniqiue ID of each order taken by the company.

Company	• The company which shops or buys items.
Customer ID	• The ID number of customer
dReceievedDate	• The Date on which orders taken
Country	• Country of the customer
status	• Status of Orders (paid or unpaid)
Processed	• processing of the order if it is true the order is delivered to customer.
ProcessedDate	• The date of the delivery process begins
Source	• The source of orders taken(Amazon,Ebay etc.)
Currency	• The currency on that an order transacted
Subtotal	• The Price without tax
Tax	• The amount of tax a customer paid on an order
Total	• The Total price a customer paid on an order
OrderItemSKU	• Unique item number which is on sale
OrderItemTitle	• The ItemTitle which a customer sees on sale platform
ItemCategory	• Brands which items on sale
DispatchStockUnitCost	• Delivery Cost of an item
OrderItemQuantity	• Quantity of items order per nOrderID
TotalWeight	• The weight of dipatched goods.
PurchasePrice	• Purchase price of items sold
TrackingNumber	• Delivery tracking number of items sold
PostalService	• The postal service over it delivery sent

FEATURE (COLUMN)	ACTIONS TAKEN

dReceievedDate	The column is splitted up in to the following columns.To get insight about sales date based on time. The main column also stays. •DRECEIEVEDDATE_YEAR •DRECEIEVEDDATE_MONTH •DRECEIEVEDDATE_DAY •DRECEIEVEDDATE_HOUR •DRECEIEVEDDATE_WEEKDAY
ProcessedDate	It is also one splitted up following columns.The main ProcessedDateColumn also stays. •PROCESSEDDATE_MONTH •PROCESSEDDATE_YEAR •PROCESSEDDATE_DAY •PROCESSEDDATE_HOUR •PROCESSEDDATE_WEEKDAY
DispatchStockUnitCost	• There was lots of zero values so we decided to drop it. Also it didn't give so much insight.
TotalWeight	• This column did not give us the expected research perspective.
TrackingNumber	• It was determined that there was a delivery tracking number, but the process of obtaining the desired information did not take place.
Processed	• After dropping false values then the column was dropped.
TP_Per_Quantity	• We would like to know the price that a customer paid per quantity with tax.
ST_Per_Quantity	• We would like to know the price that a customer paid per quantity with without tax.
UK_CPI	• To use it on analysis as on the project instructions
Turkey_CPI	• To use it on analysis as on the project instructions
Net_Profit	• To know the net profit of the seller per sale (Subtotal-PurchasePrice)
Day_Time	• 24 hours of a day splitted up 4 parts like 'AfterNoon','Night','Early Morning' and 'Morning'

status	• There was no different values as 'PAID' because of that we have dropped this.

Table 3: Actions Taken on Sales Table

1.1.1 Recommendations for Sales Table

When the analysis phase of the sales table was over, it was seen that most of the rows were filled with blank data. Although the exact reason for this is not known, it is thought that if it is a technical malfunction, the cause of the malfunction must be found. In addition, it has been determined that some zero values on data such as on columns DispatchUnitCost, TotalWeight recommended to be recorded completely. In addition, it is recommended to the company that the process date data with the missing date is required to be filled in correctly.

Finally, unidentified characters were seen in a few product titles in the table, and it was determined that these characters belonged to the Swedish language. In this regard, it can be eliminated with this character with a renewal to be made within the system.

1.2 RETURNS TABLE

The raw form of the data in the returns table contained approximately 15 thousand rows of data, and it was seen that more than half of this data was filled with null raws, as in the Sales table, in this context, these rows were removed from the data set.

The remaining columns in this table were found to contain important data and we did not need to add any new columns.

FEATRUE (COLUMN)	**THE FUNCTION OF COLUMN**
Type	• The reason of return
nOrderId	• The Id of Order returned
cPostCode	• Postcode of customer who makes item return
Customer ID	• The ID of customer
ItemNumber	• The number of item Returned
ItemTitle	• The title of item which customer returned
dReceievedDate	• The receviedate of return request
cCountry	• Country of the customer
cCountryCode	• The code of country from which rreturn made
cCurrency	• The currency of returned order
source	• The platform return made

	(amazon,ebay,etc)
subsource	• The sub platform returns made
Return Date	• The date on that return of item realized
ReturnQty	• The quantity of item returned to company by customer
Category	• The reason of return category
ResendOrExchangeQty	• Resended or Exchanged Item Quantity from the company to customer
RMA Actioned	• The status of the return process
Refund Amount	• The amount money which refunded to customer
Return Reason	• Explaination from a customer about the return

Table 5: Raw Features of Returns Table

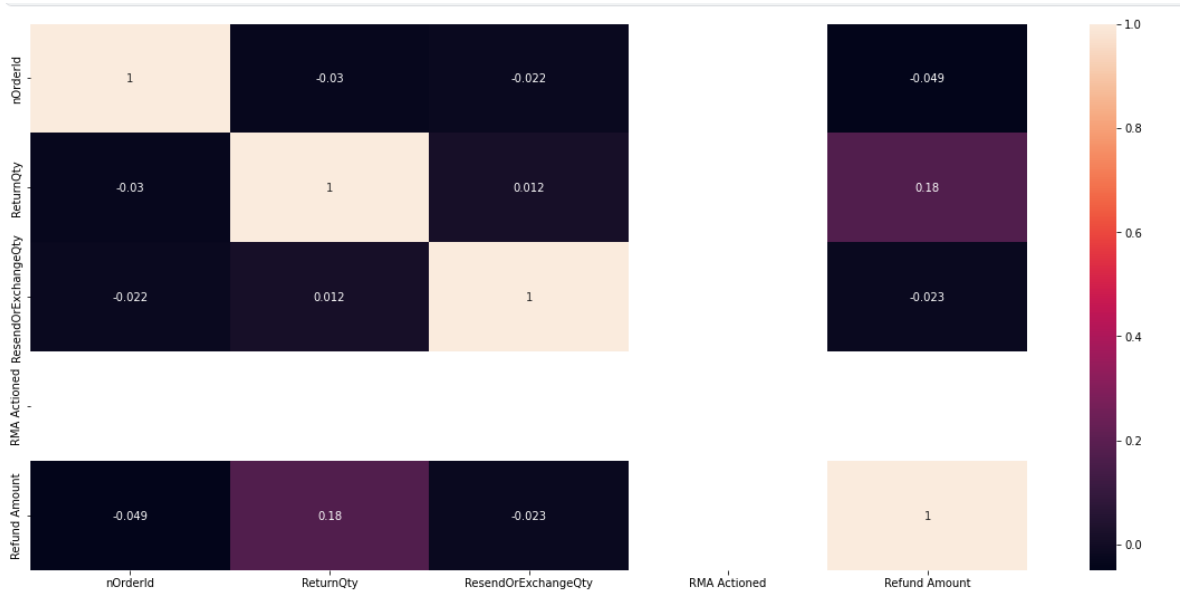


Table 6: Correlation Map of Returns Table

1.2.1 Recommendations For Returns Table

Again, as in the previous sales table, in this returns data set, it is recommended to the company that these lines, which mostly consist of blank rows, are likely to be filled if it is possible. Otherwise, these data cells will not be used for analysis.

It was seen that the returns made in terms of the Return Date column have dates, most of which show the return date of products recalled from customers. However, the same is not valid for orders shown as Refund, therefore it was thought that for the company it is proper to create a separate RefundDate column for Refund orders and add it to the sales system.

The Return Reason and Category columns are considered to be the reason for similar customer returns. But these columns contain half empty values. In order for this data to be evaluated better, the customer's reasons for return is recommended to be

FEATURE (COLUMN)	THE FUNCTION OF COLUMN
Reporting starts	• That column is mentioning about the date on which reporting starts.
Reporting ends	• That column contains the date on which the reporting ends.

recorded fully and correctly. For this, necessary arrangements must be made on the platform where the sale is made.

1.3 OTHER CAMPAIGNS

In the examinations made in the other campaigns table, it was seen that many of the examined columns contained exactly the same values. These are: Website with Adds to cart, Adds to cart, Checkouts initiated with Website checkouts initiated, Website purchases with Purchases, Purchases Conversion Value with Website purchases conversion value, are dropped.

The completely empty Meta Add to Cart, Meta Purchases, Meta Purchase Conversion Value columns were removed from the dataset.

Campaign name	• The campaign name which made
Campaign delivery	• Mentions the sitiation of campaign
Ad set budget	• This tells us the amount of the money that the seller spent on budget type basis.
Ad set budget type	• In this column it is mentioned that on which basis the seller spent the ad budget. (Daily or Lifetime)
Attribution setting	• The purpose of this column is telling the type of the ads. (28-day click or 1-day view,7-day click or 1-day view)
Results	• Number of Results which was aimed in Result indicator.
Result indicator	• The purpose of add set
Reach	• The number the ad reached
Impressions	• Total impression the ad taken(click,view etc.)
Cost per results	• Shows the cost per result
Amount spent (GBP)	• Result * Cost Per Result
Ends	• Shows the ende date of campaign
Frequency	• the average number of times users see the ad.
Unique link clicks	• the number of people who clicked
Landing page views	• people landing on ad's destination URL
Link clicks	• the number of clicks on links within the ad that led to destinations
Cost per landing page view (GBP)	• the total amount spent divided by the

	amount of landing page views.
Adds to cart	• allows customers to choose items to purchase without actually completing the payment.
Website adds to cart	• allows customers to choose items to purchase without actually completing the payment.
Meta add to cart	• allows customers to choose items to purchase without actually completing the payment on Facebook or Instagram
Checkouts initiated	• The number of purchase launch events tracked by the pixel or Conversions API on the website and attributed to the ads.
Website checkouts initiated	• The number of purchase launch events tracked by the pixel or Conversions API on the website and attributed to the ads.
Meta checkouts initiated	• *The number of initiate checkout events attributed to the ads*
Purchases	• *The number of purchases made within Meta technologies (such as Pages or Messenger) and attributed to the ads*
Website purchases	• *The number of total purchases made within the website.*
Meta purchases	• *The number of total purchases made within Meta Technologies.*
Purchases Conversion Value	• tracks the total value of purchases made from your advertising efforts
Website purchases conversion value	• tracks the total value of purchases made from your advertising efforts

FEATURE (COLUMN)	ACTIONS TAKEN
Meta purchase conversion value	• The total value of website purchases conversions.
Purchase ROAS (return on ad spend)	• the total revenue generated from your Facebook ads (your return) divided by your total ad spend.
Website purchase ROAS (return on advertising spend)	• the total revenue generated from your Facebook ads (your return) divided by your total ad spend

Table 7: Raw Features of Other Campaigns

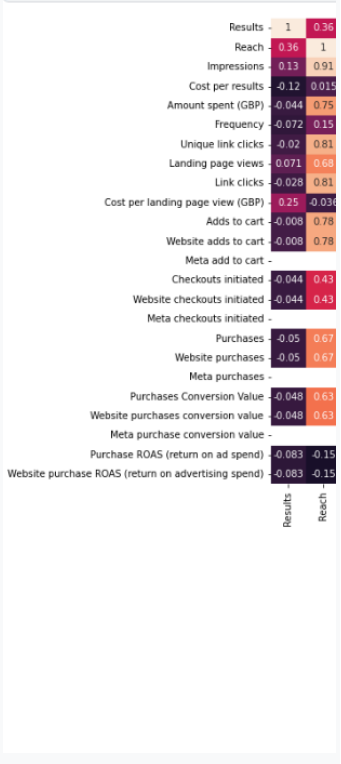


Table 8:Correlation Map of Other Campaigns

Table 9:Actions Taken on Other Campaigns

Adds to cart	• It contains same values with WebSite Adds to cart. It is duplicated so it is dropped.
Meta add to cart	• It is completely null.So it is dropped.
Checkouts initiated	• It contains same values withWebsite checkouts initiated. It is duplicated so it is dropped.
Meta checkouts initiated	• It is completely null.So it is dropped.
Purchases	• It contains same values withWebsite purchases. It is duplicated so it is dropped.
Meta purchases	• It is completely null.So it is dropped.
Purchases Conversion Value	• It contains same values withWebsite purchases conversion value. It is duplicated so it is dropped.
Meta purchase conversion value	• It is completely null.So it is dropped.
Purchase ROAS (return on ad spend)	• It contains same values withWebsite purchase ROAS (return on advertising spend). It is duplicated so it is dropped.

1.3.1 Recommendations For Other Campaigns

After examining the Other Campaigns table, it is thought that if columns such as Meta Add to Cart, Meta Checksout Initiated, Meta Purchases are completely blank, this data behooves to be filled completely.

1.4 AMAZON CAMPAIGNS

When the table was examined, it was seen that the data rows were empty in the same way and these rows were dropped. The other columns have been including significant values. Besides that to get an idea about columns as Type,Campaign Binding Strategy, Portfolio,ROAS discussion goes further. At the and we have taken the necessary answers. Consequently we have decided,it is recomended that the dataset stay the same with existing columns.

FEATURE (COLUMN)	**THE FUNCTION OF COLUMN**
State	• That column shows the status of the campanígn
Campaigns	• This is the name of the campaign
Status	• The actual status of the campaign it includes valuable information.
Type	• Sponsored Products (SP), Sponsored Brands (SB), Sponsored Brand Video (SBV), Sponsored Display (SD),
Targeting	• choosing the specific keywords and products you wish to target and set bids accordingly.
Campaign bidding strategy	• When an Amazon customer performs a search for a product, the sellers with the highest bids on relevant keywords win the auction, and their product ads get listed in their chosen placement
Start date	• Start date of campaign
End date	• End date of campaign
Portfolio	• If you have an existing Portfolio in your account, you can optionally associate the campaign being created to a particular portfolio

Budget(GBP)	• A daily budget
Top-of-search IS	• the percentage of top-of-search impressions your campaign received out of the total top-of-search impressions it was eligible to serve on
Cost type	• CPC is the cost per click that an ad receives.
Impressions	• measure the number of times Amazon shows shoppers your Ad, regardless of whether they clicked on it or not.
Clicks	• Total number of clicks on ad
CTR	• the ratio between how many people have clicked on your Ad and the number of people who have seen it:
Spend(GBP)	• Total Spend for campaign
CPC(GBP)	• Cost per Click
Orders	• The number of orders taken
Sales(GBP)	• The amount of sales made through campaign
ACOS	• It compares the amount spent on PPC campaigns to the amount earned, and it helps determine if your brand generated campaigns that were cost-efficient.
ROAS	• ROAS (Return on advertising spend) is a metric that allows sellers to calculate the amount of income (or loss) from each invested dollar and evaluate the productivity of a particular ad campaign or even a keyword.
NTB orders	• The number of first-time orders for products on Amazon within the brand over a one-year lookback window

% of orders NTB	• The percent of first-time orders for products on Amazon within the brand over a one-year lookback window
NTB sales(GBP)	• The total amount of NTB sales
% of sales NTB	• The percentage of NTB sales
Viewable impressions	• This means that almost the number of measurable impressions.
VCPM(GBP)	• Viewable CPM represents the cost to serve only viewable impressions, which can be compared directly with the CPM that you have paid to serve all of your ad impressions.

Table 40: Raw Data of Amazon Campaigns

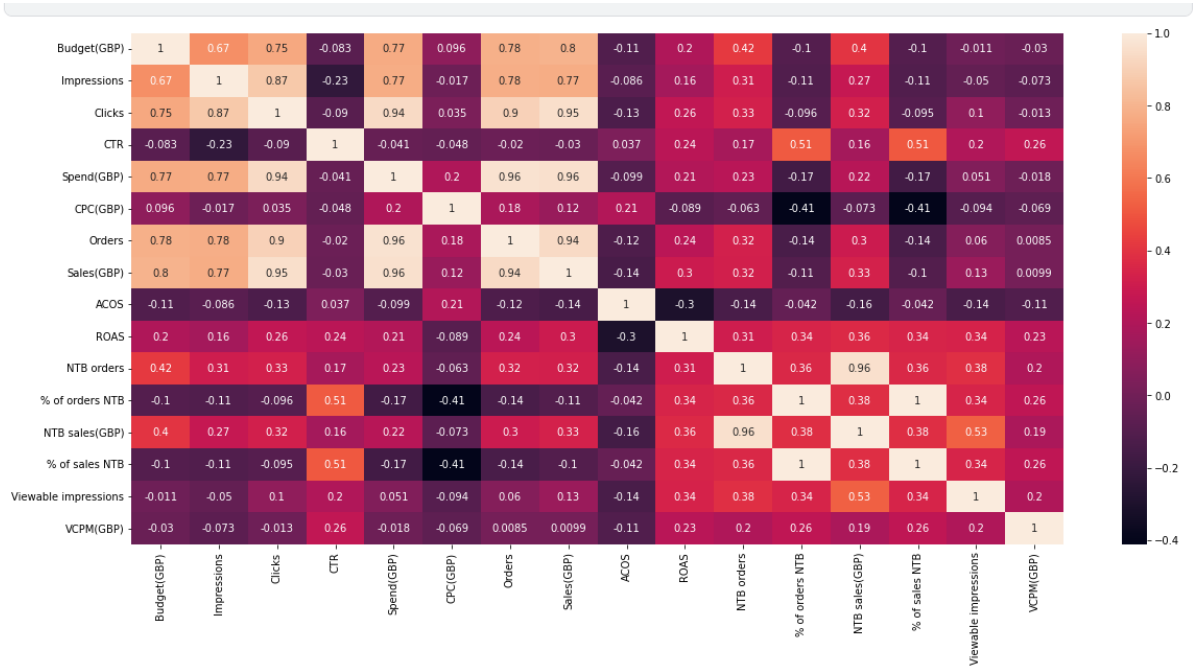


Table 5: Correlation Map of Amazon Campaigns

1.4.1 Recommendatins For Amazon Campaigns

After examining the Amazon Campaigns table, it was seen that a large part of it consisted of empty rows. It was suggested to the company that it is recommended to be filled in a more appropriate way in the same way.

1.5 PRODUCT REVIEWS

As seen in the data set received in the product comments table, empty data lines were seen as in the other tables. These were removed from the dataset.

FEATURE (COLUMN)	THE FUNCTION OF COLUMN
order_id	• The order id of the comment made
review_title	• The review title(empty)
comments	• The comments made by customers
rating	• The rating made by customers 1 to 5. (1:lowest-5:greatest)
status	• The status of review active or inactive.
date_created	• The date review created
sku	• The unique number of item
Customer ID	• The ID of customer who made the review
address	• Adress of customer (empty)
product_sku	• The unique number of item
product_name	• The name of product

FEATURE (COLUMN)	ACTIONS TAKEN
sku	• We dropped the column because it has same values with 'product_sku'.
product_link	• The link of review made
video_review_prompt_id	• Full of 199
tags	• Tags of reviews
reply	• Replies to review made
reply_private	• Private reply made
reply_date	• The date of reply
published_images	• The images published by making review
unpublished_images	• Unpublished images by making review
published_videos	• Published videos by making review
unpublished_videos	• Unpublished videos by making review
source	• The source in that the comment made
location	• Location of customer who made the review
timeago	• The time indicator for how long time ago the comment or review made
video_first_campaign	• It is almost impossible to get insight from this column

Table 6:Raw data of product reviews

address	• It is totally an empty column we can drop it
tags	• There are few values but we can drop it.
reply	• We can drop it. There are lots of null values.
reply_private	• It is empty.
published_images	• There are few values but it won't give us insight.The links on this column are not available.
unpublished_images	• There are few values but it won't give us insight.The links on this column are not available.
published_videos	• There are few values but it won't give us insight.The links on this column are not available.
unpublished_videos	• There are few values but it won't give us insight.The links on this column are not available.
timeago	• There are few values but it won't give us insight. Because it has a lot of null rows.
video_first_campaign	• There are few values but it won't give us insight. Because it has a lot of null rows.
	•

Table 13:Actions Taken on Product Reviews

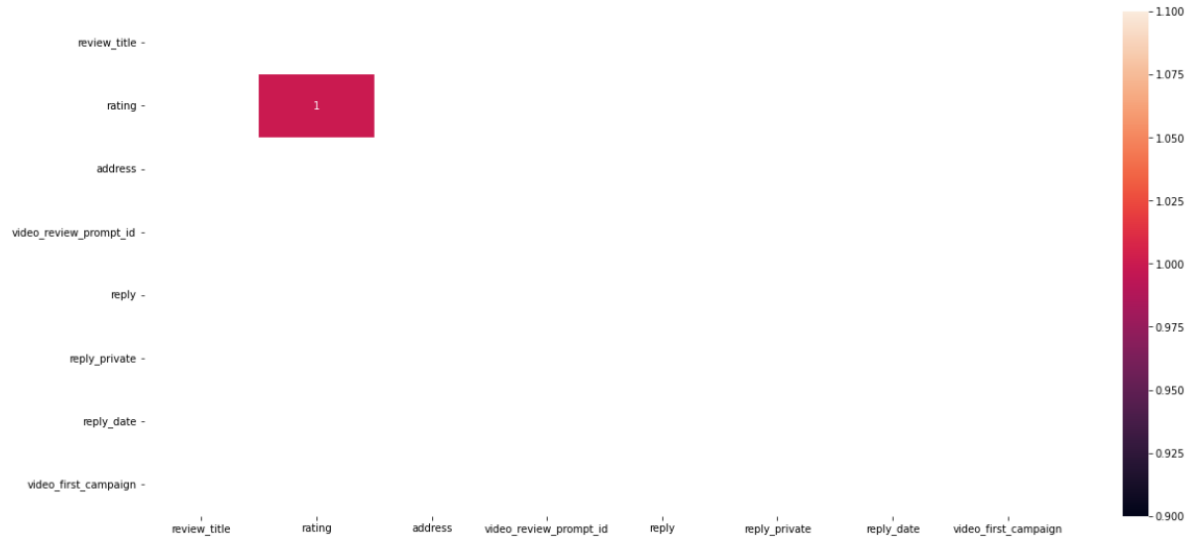


Table 14: Correlation Map of Reviews Table

1.5.1 Recommendations For Product Reviews Table

The opinions obtained as a result of the investigations are as follows:

Columns mentioned or dropped above must be taken into account to be filled if the sales system allows it to fill. Additonally, the links of videos and images those provided on the Reviews Table, it is recommended to make on work. The relations between tables could't be provided lack of table contents.

2. SPRINT 2

2.1 FORECAST FUTURE SALES

Time series is a sequence of observations recorded at regular time intervals. Depending on the frequency of observations, a time series may typically be hourly, daily, monthly, quarterly and annual.

To gain some useful insights from time series data, you must decompose the time series and look for some basic components such as trend seasonality, cyclic behavior, and irregular fluctuations. Based on some of these behaviors, we are deciding on which model to choose for time series modelling.

Stationary means that the statistical properties of a process generating a time series do not change over time. It is statistical properties (mean, variance, standard deviation) remain constant over time.

We can assume the series to be stationary if it has constant statistical properties over time

- constant mean
- constant variance
- an autocovariance that does not depend on time

We check the stationary using the:

Plotting Rolling Statistic: plot the moving average or moving variance and see if it varies time.

Dickey- Fuller Test: if the test statistic is less than the critical value, we can reject the null hypothesis and say that the series is stationary.

Dickey fuller test results:

Test Statistic	1.705736
p-value	0.998143
#Lags Used	10.000000
Number of Observations Used	26.000000
Critical Value (1%)	-3.711212
Critical Value (5%)	-2.981247
Critical Value (10%)	-2.630095

Since the p-value, is not less than 0.05 we fail to reject the null hypothesis. This means the time series is non-stationary.

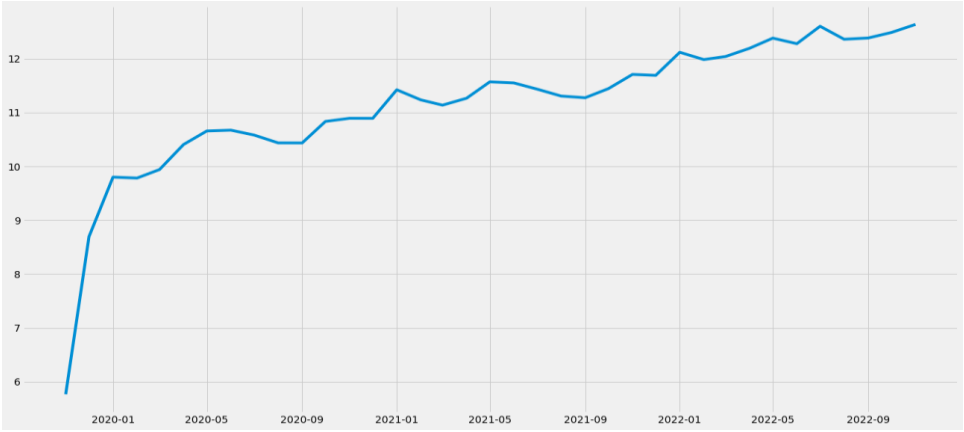
In the other words, it has some time-dependent structure and does not have constant variance over time.

Make a Time series Stationary

- Take a log transform
- Moving average
- Exponentially weighted moving average
- Difference

- Decomposition

1.5.2 Log Transform



1.5.3 Moving Average

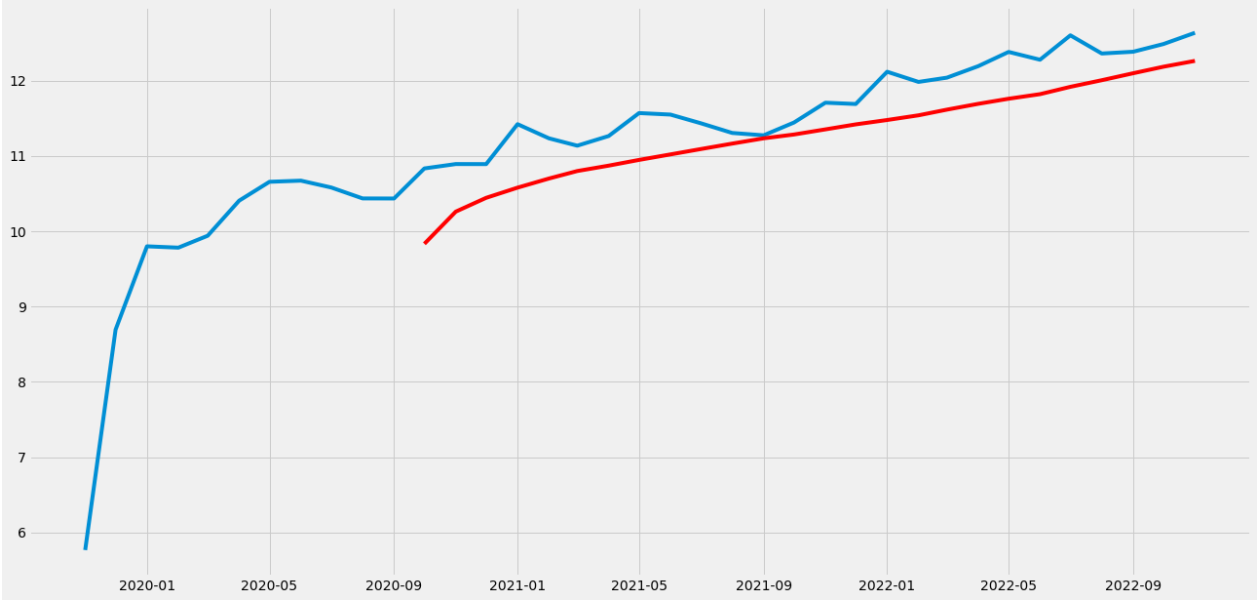


Figure 1-2

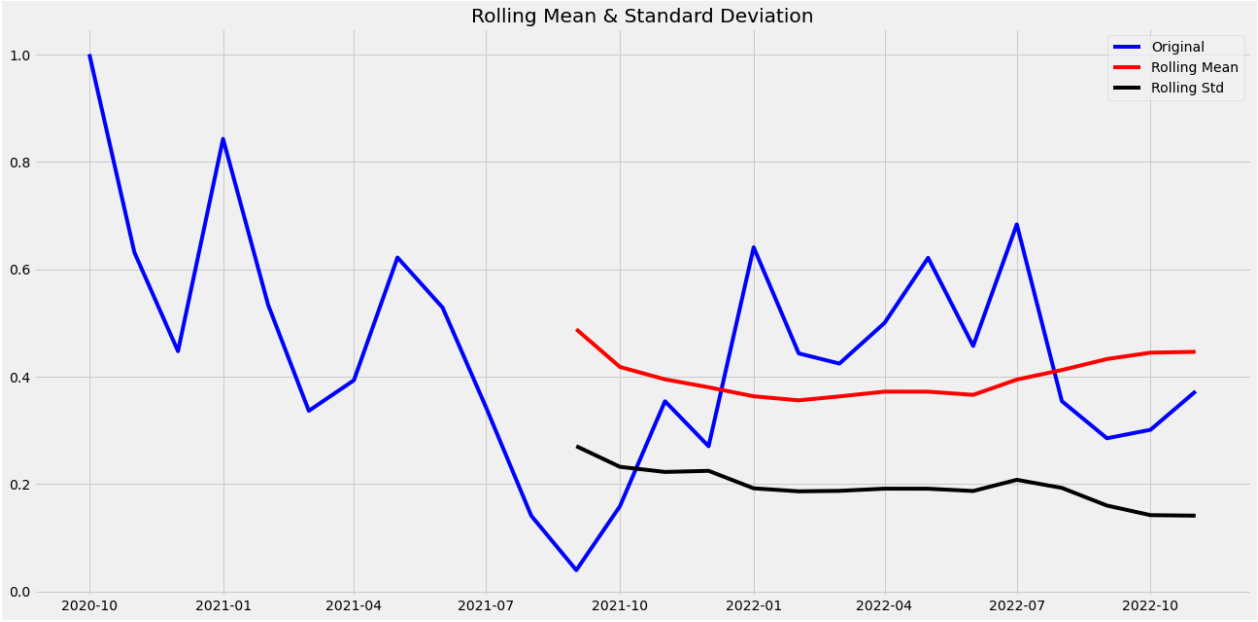


Figure 1-3

Results of Dickey-Fuller Test:

Test Statistic	-3.064405
p-value	0.029302
#Lags Used	9.000000

Number of Observations Used 16.000000

Critical Value (1%) -3.924019

Critical Value (5%) -3.068498

Critical Value (10%) -2.673893

1.5.4 Exponentially weighted moving average

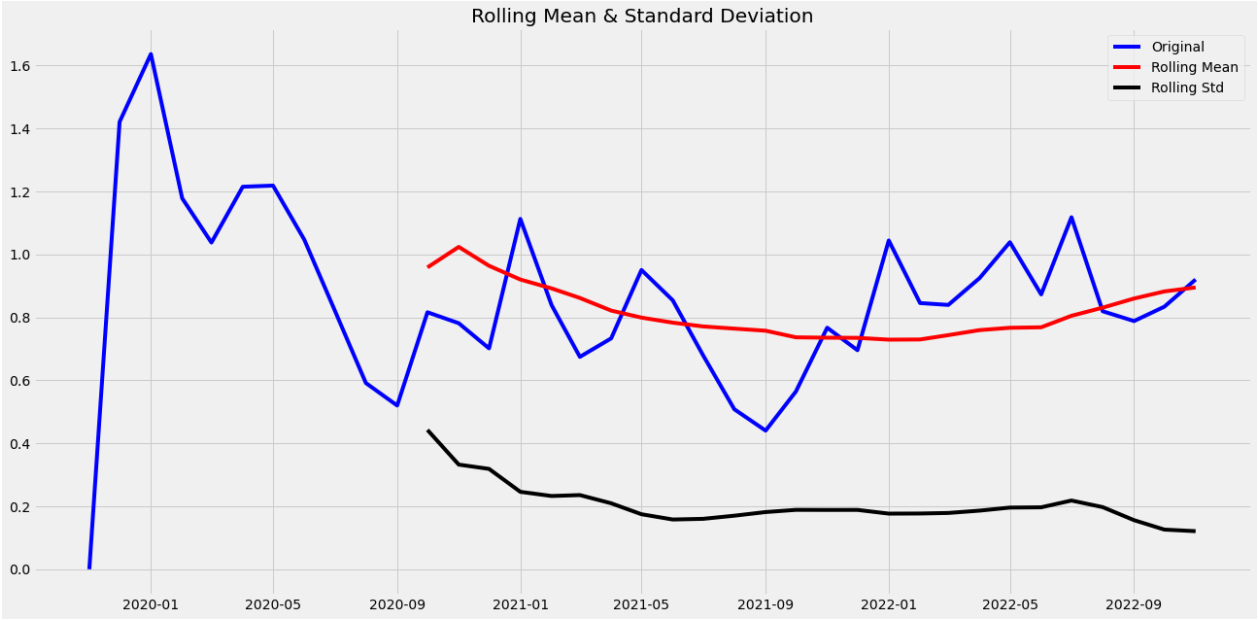


Figure 1-4

Results of Dickey-Fuller Test:

Test Statistic -4.934859

p-value 0.000030

#Lags Used 0.000000

Number of Observations Used 36.000000

Critical Value (1%) -3.626652

Critical Value (5%) -2.945951

Critical Value (10%) -2.611671

1.5.5 Difference

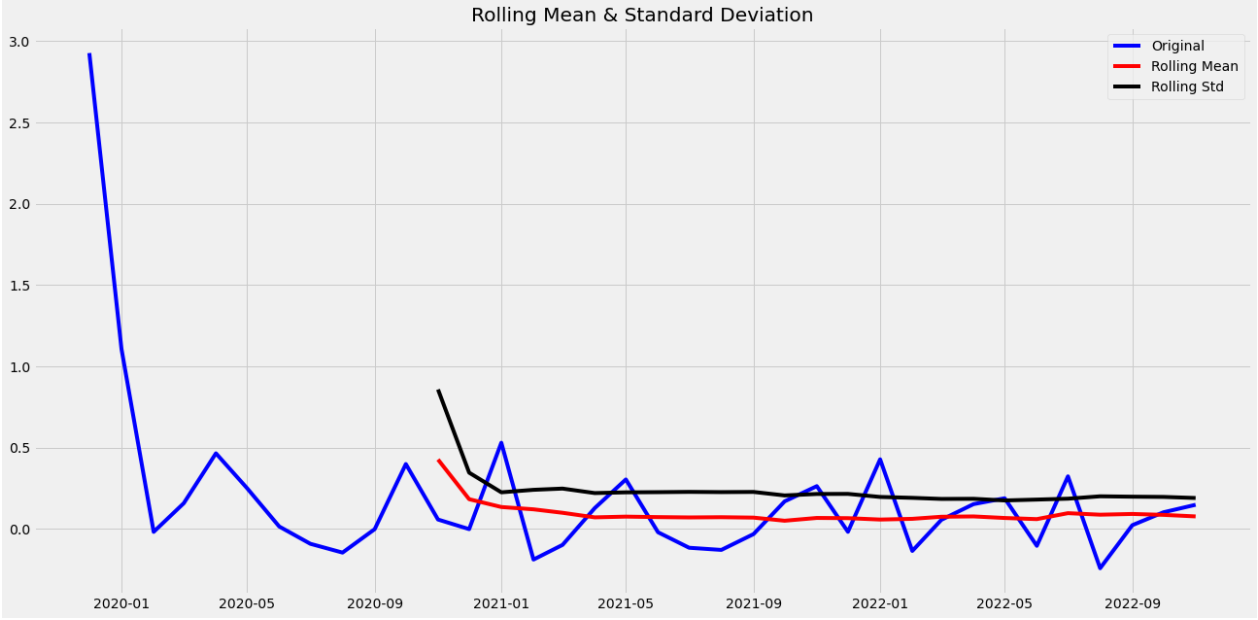


Figure 1-5

Results of Dickey-Fuller Test:

Test Statistic -1.049268e+01

p-value 1.136276e-18

#Lags Used 0.000000e+00

Number of Observations Used 3.500000e+01

Critical Value (1%)	-3.632743e+00
Critical Value (5%)	-2.948510e+00
Critical Value (10%)	-2.613017e+00

1.5.6 Decomposition

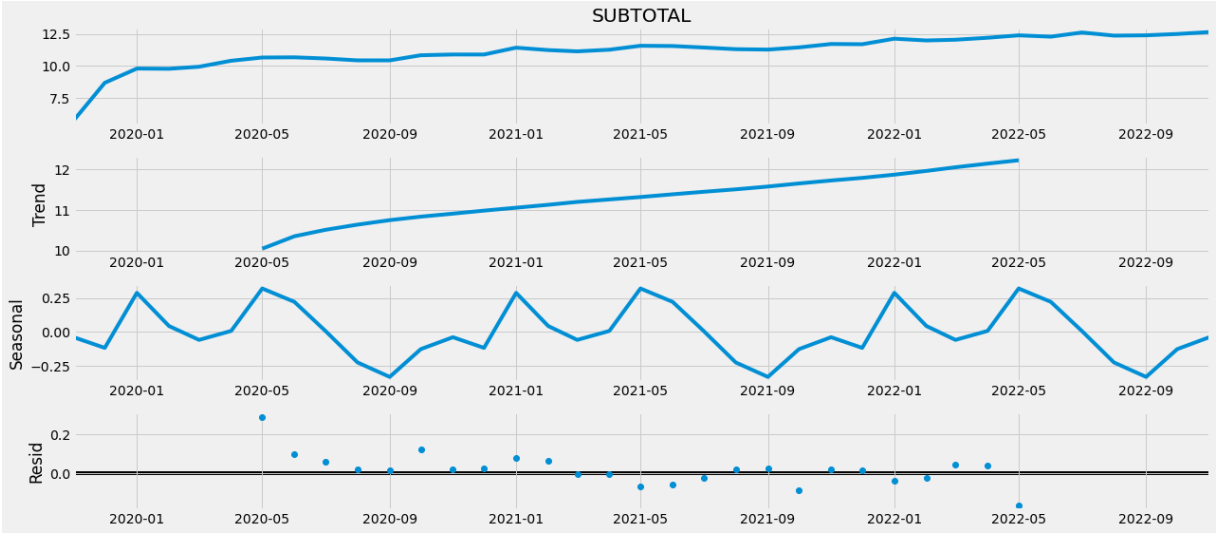


Figure 1-6

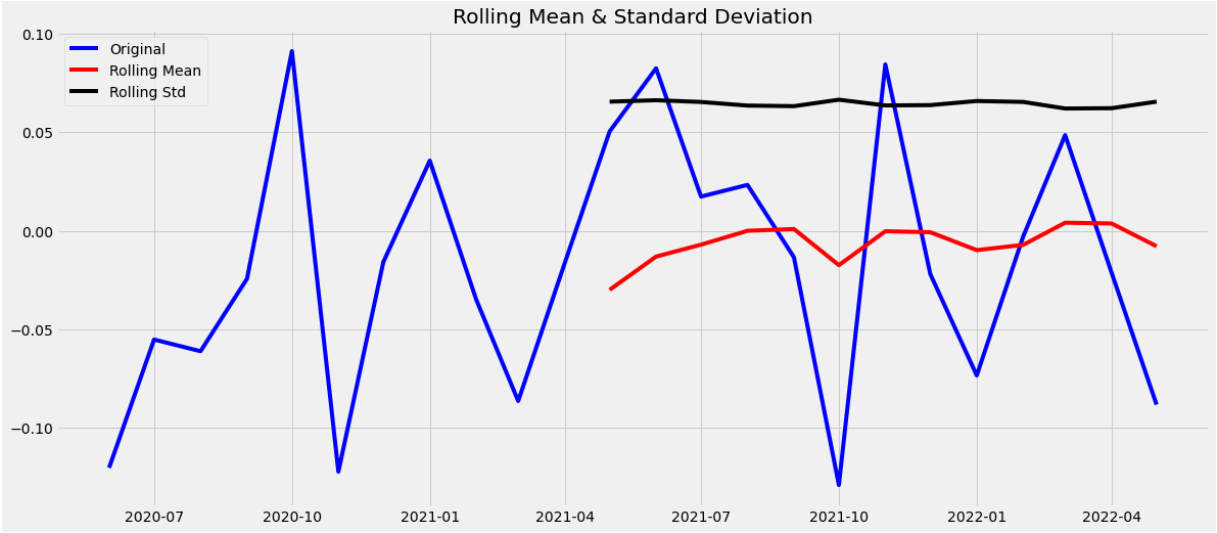


Figure 1-7

Results of Dickey-Fuller Test:

Test Statistic -5.061432

p-value 0.000017

#Lags Used 0.000000

Number of Observations Used 23.000000

Critical Value (1%) -3.752928

Critical Value (5%) -2.998500

Critical Value (10%) -2.638967

1.6 ARIMA

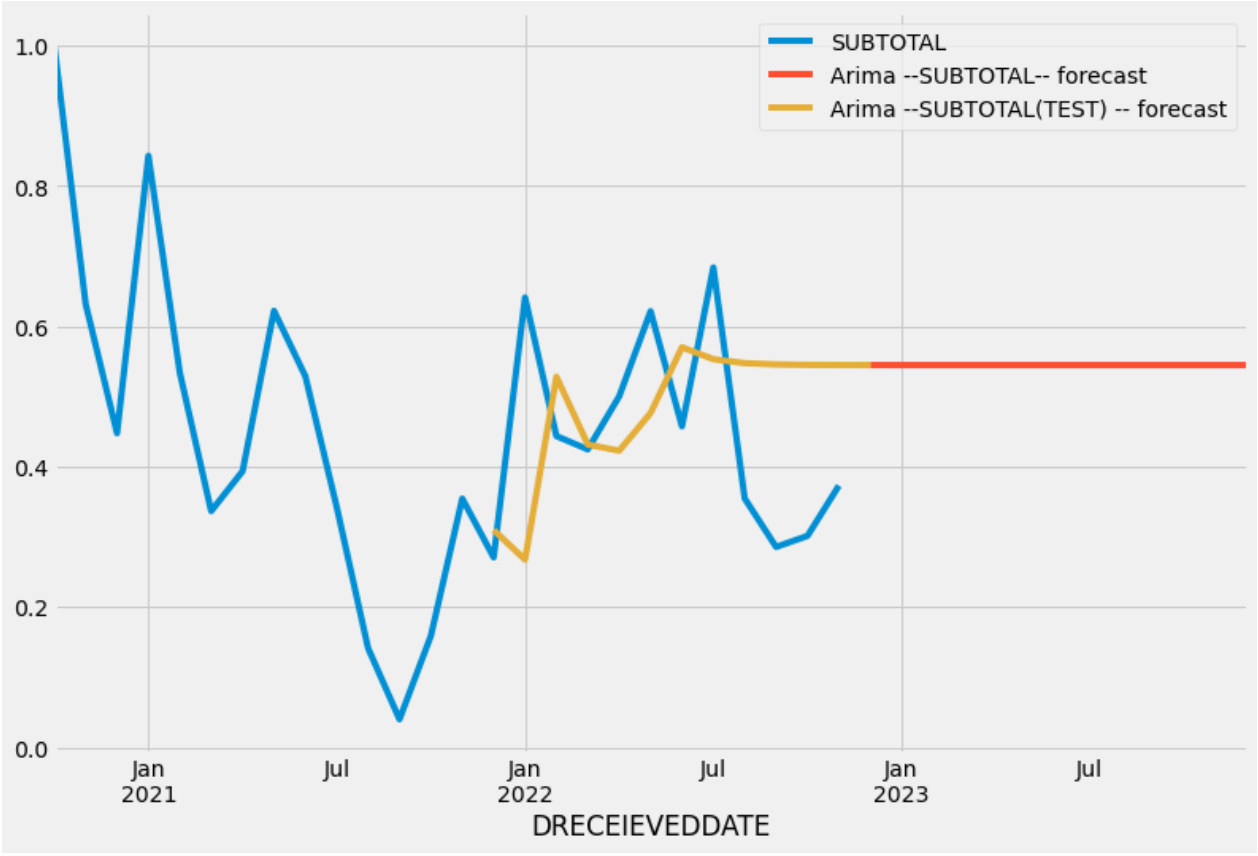


Figure 1-8

r2_score: -1.0514512338788533

mae: 0.18534852722341932

mse: 0.03726852842423262

rmse: 0.19305058514346085

1.7 SARIMAX MODEL

The SARIMAX model (Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors) is an improved version of the ARIMA model. ARIMA incorporates an

autoregressive integrated moving average, whereas SARIMAX incorporates seasonal effects and eXogenous factors in addition to the autoregressive and moving average components.

Trend Elements

There are three trend elements that require configuration.

They are the same as the ARIMA model; specifically:

- **p**: Trend autoregression order.
- **d**: Trend difference order.
- **q**: Trend moving average order.

1.8 DATA SET PREPARATION FOR THE MODEL

First of all the required libraries were imported from the python packages.

```
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
pd.set_option('display.float_format', lambda x: '%.2f' % x)
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from pmdarima import auto_arima
from sklearn.metrics import mean_squared_error
from statsmodels.tools.eval_measures import rmse
import pmdarima as pm
import warnings
warnings.filterwarnings("ignore")

warnings.filterwarnings("ignore")
```

Our main dataframe has been read by pandas library and defined as ‘df’ variable. Index column for SARIMAX model must be date or datetime column. The model makes the forecasting on datetime basis.

```
df = pd.read_csv('/kaggle/input/satislar.csv',parse_dates = True, index_col = 'DRECEIEVEDDATE')
```

Numeric columns like ’SUBTOTAL’ was selected to build SARIMAX model. The reason of seperation of dataframes, is the way of working SARIMAX model. Because the model requires only one dimensional dataframes.

```
df_subtotal = df[['SUBTOTAL']]
```

```
df_subtotal.plot(figsize=(12,8));
```

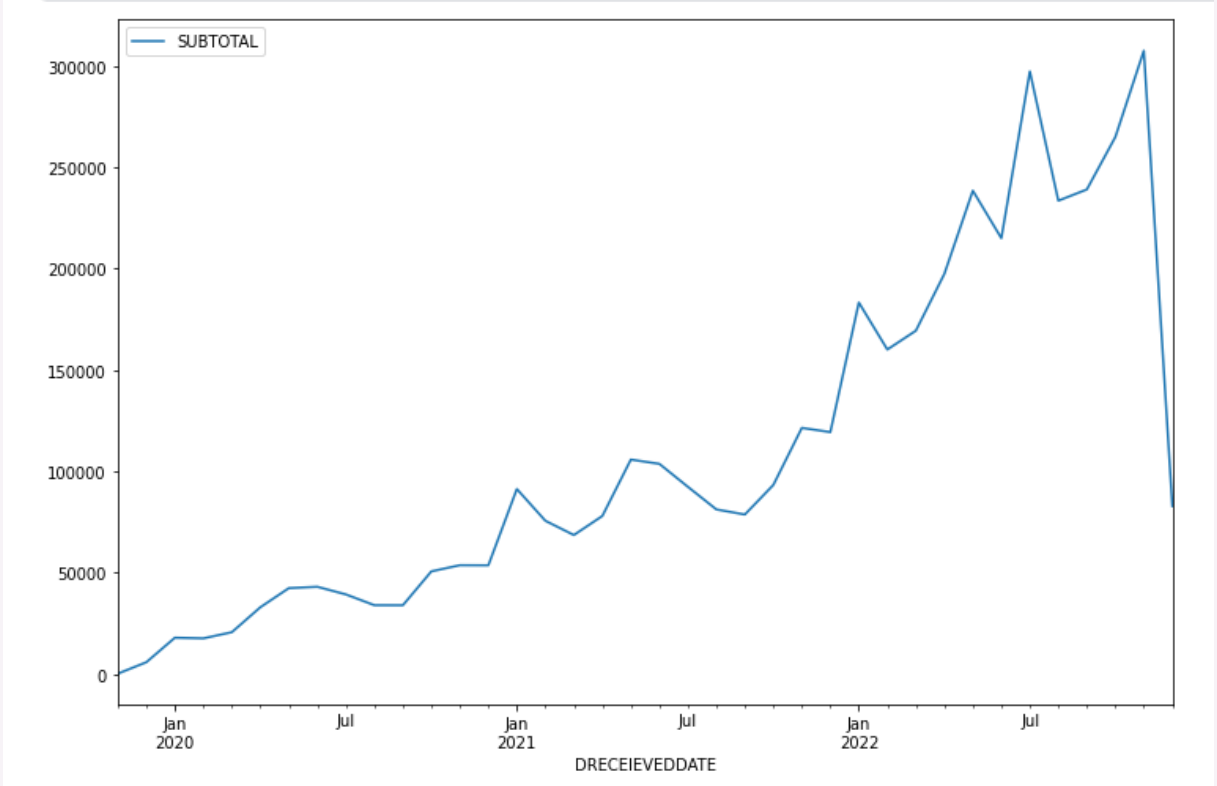


Figure 1-9

```
df['DRECEIEVEDDATE'].max()
```

Result:
```
' 9.12.2022 '
```

As the plot shows above, the data is slumping at the last value on ‘DRECEIVEDATE’ the reason for that, is the last month has not been completed. The last day of the last month was shown as ’09.12.2022’. In this case, it was decided to drop the data of the last month, as the Sarimax model would be more difficult to predict.

Below the last value of the dataframe were dropped.

```
df_subtotal.drop(df_subtotal.index[-1],inplace=True)
```

1.9 MODELLING

1.Subtotal

Subtotal value mentions the amount without tax on sales. In other words, it may be called the sales price without tax.

The dataframe fixed on the first day of the month and sum of ‘SUBTOTAL’ values were resampled on dataframe. So it is calculated monthly ‘SUBTOTAL’ values.

```
df_subtotal = df_subtotal.resample('MS').sum()
```

Result of Resampling(SUBTOTAL):

DRECEIEVEDDATE	SUBTOTAL
2022-08-01	233532.68
2022-09-01	239066.49
2022-10-01	264925.63
2022-11-01	307485.95
2022-12-01	82943.53

Figure 1-10

```
Sarimax_model = auto_arima(df_subtotal,
                      start_P=1,
                      start_q=1,
                      max_p=3,
                      max_q=3,
                      m=12,
                      seasonal=True,
                      d=None,
                      D=1,
                      trace=True,
                      error_action='ignore',
                      suppress_warnings=True,
                      stepwise=True)
Sarimax_model.summary()
```

‘Auto_Arima’ function is the function under the SARIMAX library to give the best parameters for the future forcasting.

Start_p refers to the starting value of P , the order of the auto-regressive portion of the seasonal model.

Star_q refers to The starting value of q , the order of the moving-average portion of the seasonal model.

Max_p refers to The maximum value of p, inclusive. Must be a positive integer greater than start_p.

Max_q refers to The maximum value of Q, inclusive. Must be a positive integer greater than start_q.

Parameter ‘M’ refers to The period for seasonal differencing, to the number of periods in each season. For example, m is 4 for quarterly data, 12 for monthly data, or 1 for annual (non-seasonal) data. Default is 1. Note that if m == 1 (i.e., is non-seasonal), seasonal will be set to False. For more information on setting this parameter, see Setting m.

Parameter ‘seasonal’ refers to Whether to fit a seasonal ARIMA. Default is True. Note that if seasonal is True and m == 1, seasonal will be set to False.

Parameter ‘d’ The order of first-differencing. If None (by default), the value will automatically be selected based on the results of the test (i.e., either the Kwiatkowski–Phillips–Schmidt–Shin, Augmented Dickey-Fuller or the Phillips–Perron test will be conducted to find the most probable value). Must be a positive integer or None. Note that if d is None, the runtime could be significantly longer.

Parameter ‘D’ refers to The order of the seasonal differencing. If None (by default, the value will automatically be selected based on the results of the seasonal_test. Must be a positive integer or None.

Parameter ‘trace’ refers to Whether to print status on the fits. A value of False will print no debugging information. A value of True will print some. Integer values exceeding 1 will print increasing amounts of debug information at each fit.

Parameter ‘error_action’ refers to If unable to fit an ARIMA for whatever reason, this controls the error-handling behavior. Model fits can fail for linear algebra errors, convergence errors, or any number of problems related to stationarity or input data.

- ‘warn’: Warns when an error is encountered (default)
- ‘raise’: Raises when an error is encountered
- ‘ignore’: Ignores errors (not recommended)
- ‘trace’: Logs the entire error stacktrace and continues the

search. This is the best option when trying to determine why a model is failing.

Parameter ‘suppress_warnings’ refers to Many warnings might be thrown inside of statsmodels. If suppress_warnings is True, all of the warnings coming from ARIMA will be

squelched. Note that this will not suppress UserWarnings created by bad argument combinations.

Parameter ‘stepwise’ refers to The stepwise algorithm can be significantly faster than fitting all (or a random subset of) hyper-parameter combinations and is less likely to over-fit the model.

Results of auto_arima_subtotal:

```
Performing stepwise search to minimize aic
 ARIMA(2,1,1)(1,1,1)[12]             : AIC=560.159, Time=0.44 sec
 ARIMA(0,1,0)(0,1,0)[12]             : AIC=558.022, Time=0.03 sec
 ARIMA(1,1,0)(1,1,0)[12]             : AIC=554.870, Time=0.05 sec
 ARIMA(0,1,1)(0,1,1)[12]             : AIC=556.640, Time=0.08 sec
 ARIMA(1,1,0)(0,1,0)[12]             : AIC=553.415, Time=0.02 sec
 ARIMA(1,1,0)(0,1,1)[12]             : AIC=555.123, Time=0.04 sec
 ARIMA(1,1,0)(1,1,1)[12]             : AIC=556.385, Time=0.09 sec
 ARIMA(2,1,0)(0,1,0)[12]             : AIC=555.301, Time=0.02 sec
 ARIMA(1,1,1)(0,1,0)[12]             : AIC=555.478, Time=0.03 sec
 ARIMA(0,1,1)(0,1,0)[12]             : AIC=555.007, Time=0.02 sec
 ARIMA(2,1,1)(0,1,0)[12]             : AIC=557.349, Time=0.08 sec
 ARIMA(1,1,0)(0,1,0)[12] intercept   : AIC=552.107, Time=0.02 sec
 ARIMA(1,1,0)(1,1,0)[12] intercept   : AIC=554.092, Time=0.07 sec
 ARIMA(1,1,0)(0,1,1)[12] intercept   : AIC=554.062, Time=0.05 sec
 ARIMA(1,1,0)(1,1,1)[12] intercept   : AIC=556.061, Time=0.08 sec
 ARIMA(0,1,0)(0,1,0)[12] intercept   : AIC=558.637, Time=0.02 sec
 ARIMA(2,1,0)(0,1,0)[12] intercept   : AIC=552.898, Time=0.04 sec
 ARIMA(1,1,1)(0,1,0)[12] intercept   : AIC=553.885, Time=0.04 sec
 ARIMA(0,1,1)(0,1,0)[12] intercept   : AIC=557.750, Time=0.03 sec
 ARIMA(2,1,1)(0,1,0)[12] intercept   : AIC=554.932, Time=0.04 sec

Best model:  ARIMA(1,1,0)(0,1,0)[12] intercept
Total fit time: 1.302 seconds
```

Figure 1-11

Model: SARIMAX(1, 1, 0)x(0, 1, 0, 12)

1.9.1 Fitting the model_subtotal

With the code below, it is build to fit ‘SARIMA’ model.

```
model_subtotal = SARIMAX(df_subtotal,order=(1, 1, 0),
          seasonal_order=(0, 1, 0, 12),
           enforce_stationarity=False,
           enforce_invertibility=False)
results_subtotal = model_subtotal.fit()
```

Wıth the code below, the forecast is operated for the next 12 months on dataset. It is given the start and end dates over the lenght of the dataset and the forecasting was made with predict function. Additonally, it is created a test model to get insight for the model performance.

```
forecast_subtotal = results_subtotal.predict(start =
len(df_subtotal),end=len(df_subtotal)+12,typ='levels').rename('Arima --SUBTOTAL-- forecast')

forecast_subtotal_test = results_subtotal.predict(start = len(df_subtotal)-
12,end=len(df_subtotal),typ='levels').rename('Arima --SUBTOTAL(TEST) -- forecast')
```

The code belove is plotting the previous subtoal values and forecasted subtotal values.

```
df_subtotal.plot(figsize=(12,8),legend=True)

forecast_subtotal.plot(legend=True);

forecast_subtotal_test.plot(legend=True)
```

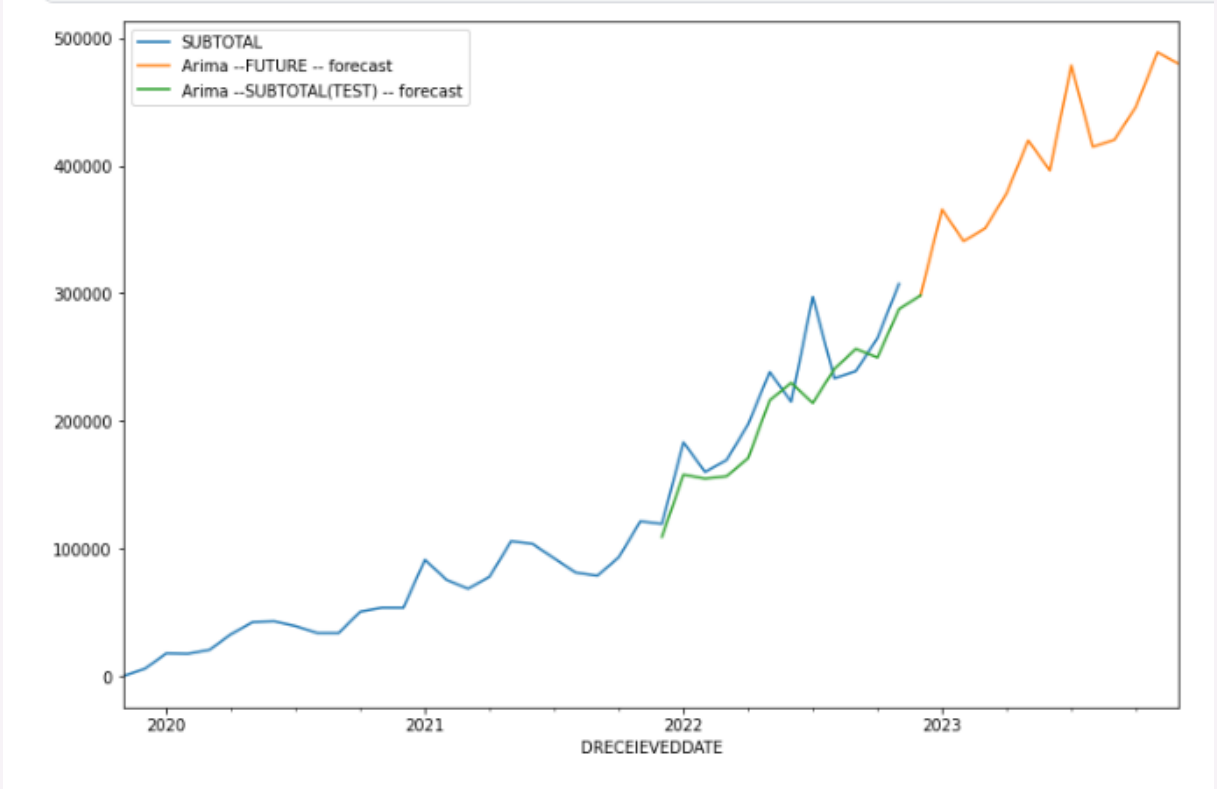


Figure 1-12

```
print(forecast_subtotal)
```

```
2022-12-01   298409.66
2023-01-01   365681.39
2023-02-01   340872.26
2023-03-01   350945.58
2023-04-01   378563.40
2023-05-01   419793.29
2023-06-01   396282.23
2023-07-01   478561.14
2023-08-01   414794.80
2023-09-01   420338.97
2023-10-01   446193.09
2023-11-01   488755.85
2023-12-01   479678.37
Freq: MS, Name: Arima --SUBTOTAL-- forecast, dtype: float64
```

Figure 1-13

```
print(forecast_subtotal_test)
```

```
2021-12-01   109290.80
2022-01-01   158028.46
2022-02-01   154909.87
2022-03-01   156737.48
2022-04-01   170910.98
2022-05-01   216211.74
2022-06-01   229940.32
2022-07-01   214067.09
2022-08-01   240752.40
2022-09-01   256517.69
2022-10-01   249746.02
2022-11-01   287598.45
2022-12-01   298409.66
Freq: MS, Name: Arima --SUBTOTAL(TEST) -- forecast, dtype: float64
```

Figure 1-14

1.9.2 Test Forecasting of the model

32]:

	SUBTOTAL	Arima --SUBTOTAL(TEST) -- forecast	Forecast Difference Percentage
2021-12-01	119423.94	109290.80	8.49
2022-01-01	183305.02	158028.46	13.79
2022-02-01	160140.10	154909.87	3.27
2022-03-01	169416.10	156737.48	7.48
2022-04-01	197420.56	170910.98	13.43
2022-05-01	238462.96	216211.74	9.33
2022-06-01	215042.82	229940.32	6.93
2022-07-01	297277.64	214067.09	27.99
2022-08-01	233532.68	240752.40	3.09
2022-09-01	239066.49	256517.69	7.30
2022-10-01	264925.63	249746.02	5.73
2022-11-01	307485.95	287598.45	6.47
2022-12-01	298409.66	298409.66	0.00

Figure 1-15

```
df_subtotal_tail['Forecast Difference Percentage'].mean()
```

Result:

```
8.714654823863587
```

When the performance of the model was evaluated, it was observed that there was an average of 9% difference between the actual and predictive values of the model.

```
eval_metric(df_subtotal_tail['SUBTOTAL'],df_subtotal_tail['Arima --SUBTOTAL(TEST) -- forecast'])
```

Result:

```
r2_score: 0.746786698282228
mae: 19994.263328004698
mse: 788930435.8409036
rmse: 28087.905508259308
```

Considering the predictive power and R2 scores of the model, the model made predictions with an r2 score of 0.75. R2 score close to 1 indicates that the prediction is accurate and strong

1.3 Future Forecasting of the model

The future predictions made by the SARIMAX modeling are given as follows.

2022-12-01	298409.66
2023-01-01	365681.39
2023-02-01	340872.26
2023-03-01	350945.58
2023-04-01	378563.40
2023-05-01	419793.29
2023-06-01	396282.23
2023-07-01	478561.14
2023-08-01	414794.80
2023-09-01	420338.97
2023-10-01	446193.09
2023-11-01	488755.85
2023-12-01	479678.37

Figure 1-16

It is also obtained from the model evaluation results that the model will have a certain margin of error on the predictions made. The estimation covers December of 2022 and December of 2023.

1.10 TIME SERIES WITH PROPHET

Prophet follows the sklearn model API. Here we create an instance of the Prophet class and then call its fit and predict methods.

The input to Prophet is always a dataframe with two columns: "ds" and "y". The ds (datestamp) column should be of a format expected by Pandas, ideally YYYY-MM-DD for a date or YYYY-MM-DD HH:MM:SS for a timestamp. The y column must be numeric, and represents the measurement we wish to forecast.

Visit: https://facebook.github.io/prophet/docs/quick_start.html#python-api

1.10.1 Model

In this model, we will predict daily sales. We will not include November 2019, December 2019 and December 2022 because they do not have enough data. The first two months in the dataset have missing daily sales values. On the other hand, we do not have data for the second half of December 2022. Therefore, their daily sales will be misleading for our prediction. Also, by excluding December 2022, we will have an opportunity to compare

forecasted sales with the actual values. So, in this model, we will use data beginnig from January 2020 until the end of Nvember 2022.

Since we have Subtotal values until 11 December 2022, we can check the success of our Prophet model by using these actual sales values.

Sales (SUBTOTAL) in Original Data Forecast

	DRECEIEVEDDATE	SUBTOTAL
1069	2022-11-24	11,750.3
1070	2022-11-25	13,555.0
1071	2022-11-26	12,421.8
1072	2022-11-27	16,071.2
1073	2022-11-28	12,841.4
1074	2022-11-29	13,902.4
1075	2022-11-30	12,908.5
1076	2022-12-01	4,726.3
1077	2022-12-02	4,306.7
1078	2022-12-03	3,962.2
1079	2022-12-04	5,714.3
1080	2022-12-05	5,679.4
1081	2022-12-06	9,353.8
1082	2022-12-07	15,697.3
1083	2022-12-08	7,662.0
1084	2022-12-09	7,173.6
1085	2022-12-10	7,943.2
1086	2022-12-11	10,724.8

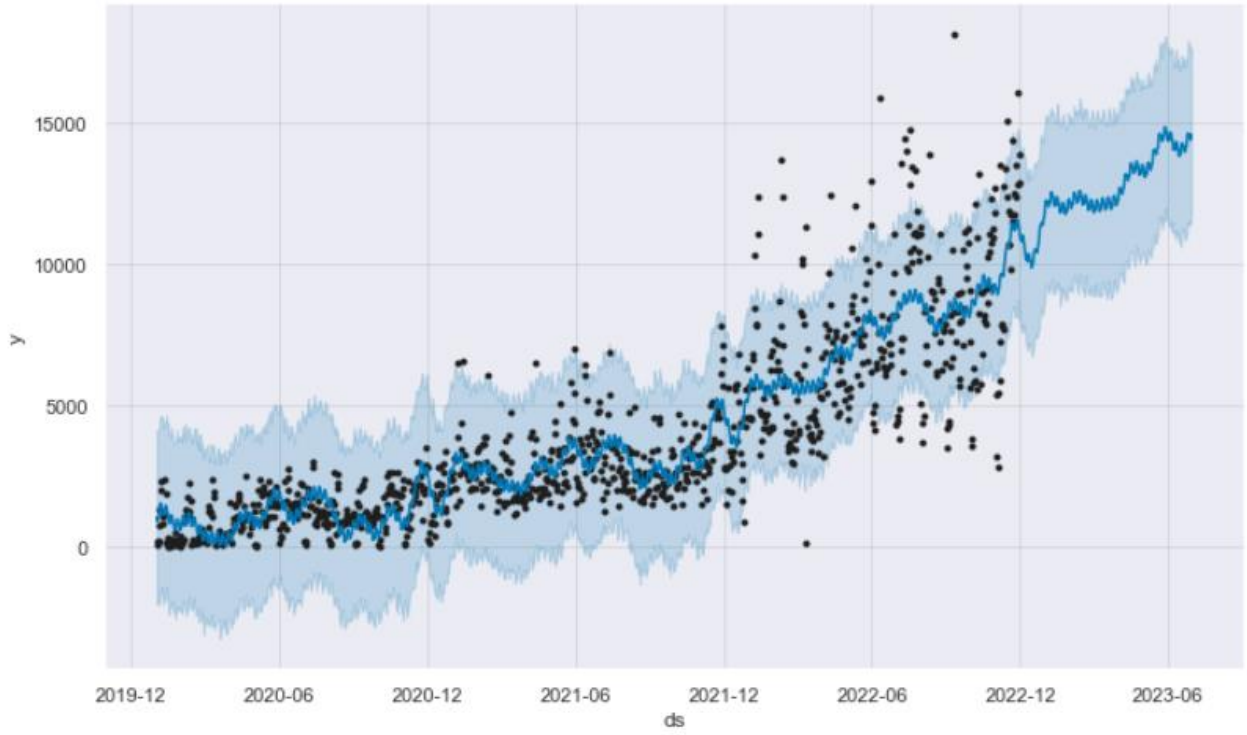
	ds	yhat
1050	2022-11-24	11,206.5
1051	2022-11-25	11,342.7
1052	2022-11-26	11,298.7
1053	2022-11-27	11,608.2
1054	2022-11-28	11,579.4
1055	2022-11-29	11,395.1
1056	2022-11-30	11,242.7
1057	2022-12-01	10,933.3
1058	2022-12-02	10,959.1
1059	2022-12-03	10,816.6
1060	2022-12-04	11,041.7
1061	2022-12-05	10,944.4
1062	2022-12-06	10,709.3
1063	2022-12-07	10,524.8
1064	2022-12-08	10,202.6
1065	2022-12-09	10,235.4
1066	2022-12-10	10,119.3
1067	2022-12-11	10,389.5

Let's see the sales forecasting in June. The last 10 days in June as an example:

```
# sales forecasting in June
forecast_values.tail(10)
```

	ds	yhat
1259	2023-06-21	14,167.1
1260	2023-06-22	13,988.8
1261	2023-06-23	14,162.1
1262	2023-06-24	14,179.8
1263	2023-06-25	14,574.2
1264	2023-06-26	14,651.2
1265	2023-06-27	14,591.6
1266	2023-06-28	14,579.7
1267	2023-06-29	14,423.6
1268	2023-06-30	14,612.2

1.10.2 Forecasting of Daily Sales from December 2022 until the end of June 2023 in Graph:



The plot above shows the predictions in the actual term (January 2020-December 2022) as weel as the following term beginning from December 2022 until the end of June (see the trend without values).

1.11 ANOVA TEST

Is there a difference between the average monthly sales amounts on the sales platforms?

Introduction

There are some statistical tests to see if there is a difference between monthly sales averages. Since there are more than 2 groups here, we decided to apply ANOVA Test. The one-way ANOVA tests the null hypothesis that two or more groups have the same population mean. The test is applied to samples from two or more groups, possibly with differing sizes.

The ANOVA test tells if there is a difference between the averages. If there is, it does not say between which groups.

In order to apply ANOVA test to a data, the following 6 conditions must be fulfilled:

1. The dependent variable must be continuous.
2. The independent variable must be categorical.
3. The number of groups must be at least 2.
4. The sample must have a normal distribution.
5. The variance of the groups must be homogeneous.
6. Observations should be randomly selected.

Here, the top 5 platforms with the most sales were selected.

```
df_anova['source'].value_counts(dropna = False)
✓ 0.4s

AMAZON            54814
WOOCOMMERCE       21328
EBAY               8790
AMAZON FBA         5110
ETSY               1777
WAYFAIRCHANNEL      355
OnBuy.com            40
DIRECT               35
Name: source, dtype: int64
```

Figure 2-1

The sum of each platform's monthly sales is assigned to a list.

```
g1A = list(df1['AMAZON'])
✓ 0.1s

g1A
✓ 0.2s

[179724.93,
 152342.31,
 146539.83,
 179833.04,
 225011.2,
 205314.3,
 237088.51,
 197429.13,
 198763.44,
 233401.61,
 283746.97,
 153866.4]
```

Figure 2-2

Shapiro-Wilk test for Normality

The Shapiro-Wilk test tests the null hypothesis that the data was drawn from a normal distribution.

```
normallik = stats.shapiro(g1A)
print(normallik)
✓ 0.2s
ShapiroResult(statistic=0.9487230777740479, pvalue=0.6183980703353882)

normallik = stats.shapiro(g2Web)
print(normallik)
✓ 0.1s
ShapiroResult(statistic=0.9115359783172607, pvalue=0.22323347628116608)

normallik = stats.shapiro(g3Ebay)
print(normallik)
✓ 0.2s
ShapiroResult(statistic=0.9361796379089355, pvalue=0.4502195417881012)

normallik = stats.shapiro(g4Etsy)
print(normallik)
✓ 0.2s
ShapiroResult(statistic=0.862890362739563, pvalue=0.05314469709992409)

normallik = stats.shapiro(g5Afba)
print(normallik)
✓ 0.1s
ShapiroResult(statistic=0.8857870101928711, pvalue=0.10399952530860901)
```

Figure 2-3

Since the p-value for all groups is >0.05, the null hypothesis cannot be rejected (normality assumption is valid). There is a normal distribution.

1.12 Homogeneity with Bartlett's Test for Equal Variances

Bartlett's test tests the null hypothesis that all input samples are from populations with equal variances. For samples from significantly non-normal populations, Levene's test levene is more robust.

```
homojenlik = stats.bartlett(g1A, g2Web, g3Ebay, g4Etsy, g5Afba)
homojenlik
✓ 0.3s
BartlettResult(statistic=123.75598136635764, pvalue=8.418384350614998e-26)
```

Figure 2-4

Here, it was concluded that the group variances were not homogeneous because the p-value was less than 0.05.

Let's look at the top 3 platforms with high sales

```
homojenlik = stats.bartlett(g1A, g2Web, g3Ebay)
homojenlik
✓ 0.2s
BartlettResult(statistic=33.48274323125633, pvalue=5.361850340681563e-08)
```

Figure 2-5

p-value again less than 0.05. These 5 groups do not meet the homogeneity of ANOVA test conditions. Let's look at the first two platforms.

```
homojenlik = stats.bartlett(g1A, g2Web)
homojenlik
✓ 0.2s
BartlettResult(statistic=1.2252643297967067, pvalue=0.26832999390167367)
```

Figure 2-6

Since the P-value is >0.05, the variances of Amazon and Website groups are homogenous. If we apply one-way ANOVA test to these two groups

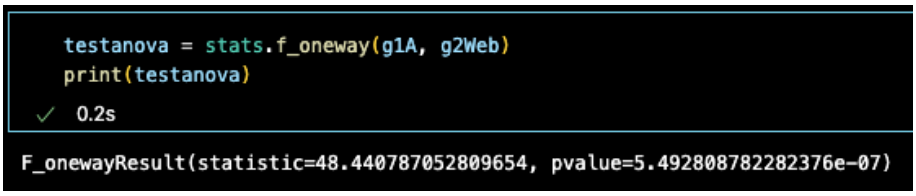


Figure 2-7

The one-way ANOVA tests the null hypothesis that two or more groups have the same population mean. P-value < 0.05, the h_0 hypothesis is rejected. This means that the groups do not have the same population mean.

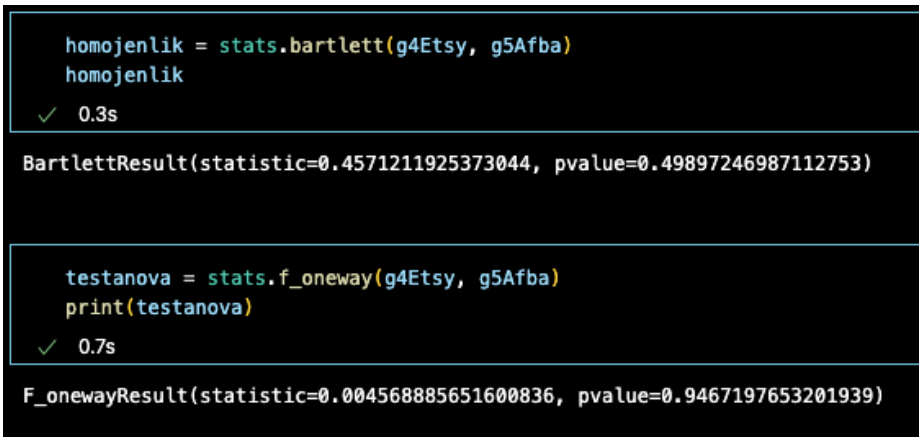


Figure 2-8

When we compare Etsy and Amazon FBA groups, one-way ANOVA test shows us that the average monthly sales of these two platforms are close to each other.

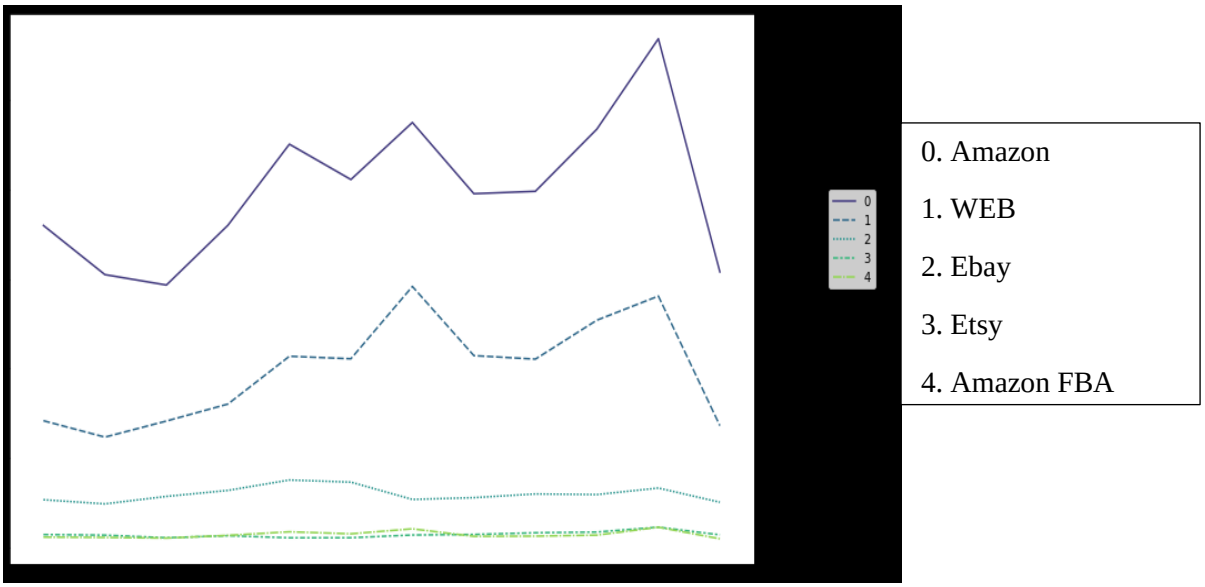


Figure 2-9

1.13 INCOME ANALYSIS WITH TABLEAU

Determining Whether There Are Breaks In the Monthly Turnover Values

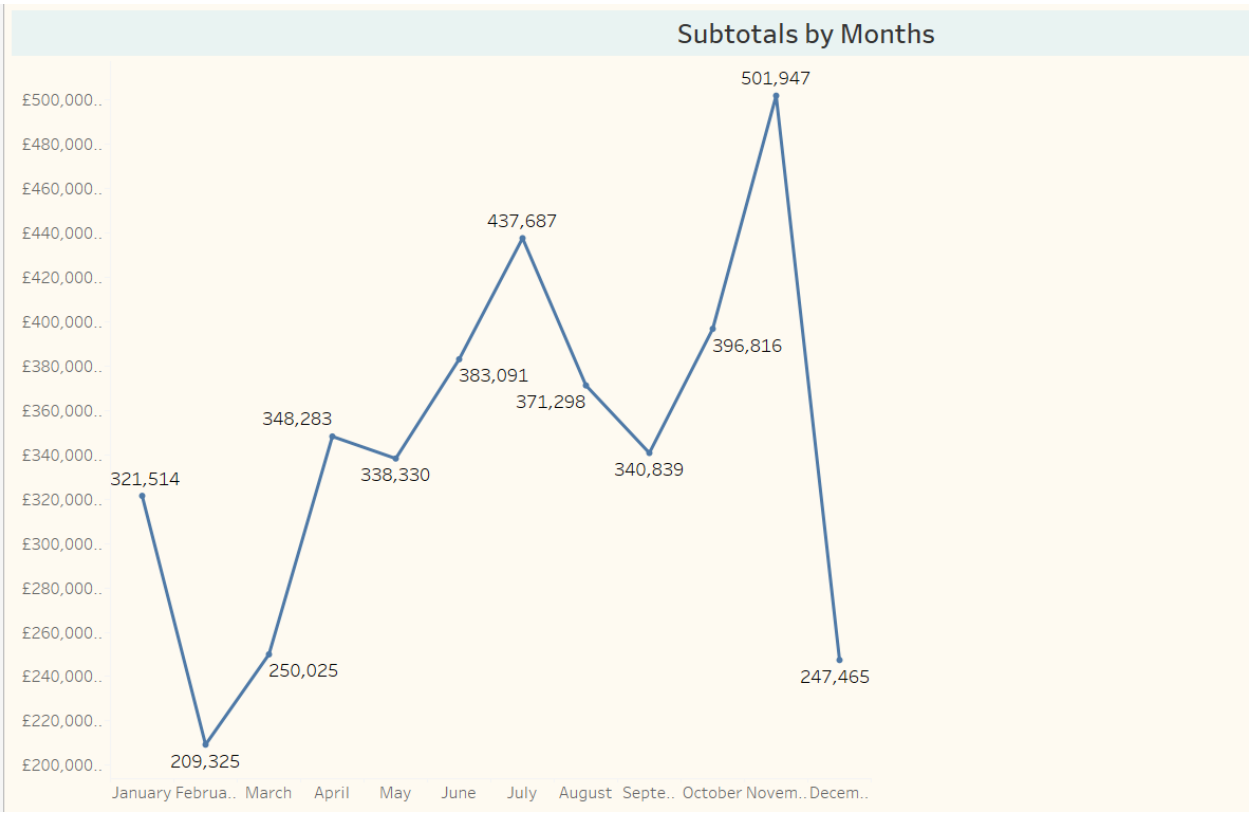


Figure 3-1

It is observed that there are decreases in the product turnover in January. In February and March, it is observed that the rise continues, but between April and May there is a situation that can be said to be partially sellable. After May, it is seen that the rise continues until July. A decrease is observed in July and August. However, the rise continued from September to December. It is seen that there is a serious decrease in December. The reason for this is thought to be due to the fact that the dataset we have contains the latest data dated December 9th. In fact, since there is no data on the whole month, a healthy evaluation cannot be made.

Comment: The reason for the decrease in July and August; It is considered that the reason for the rise from September to December may be due to the high season.

1.13.1 Determining Whether There is a Break in The Turnover Values of The Months of Each Year

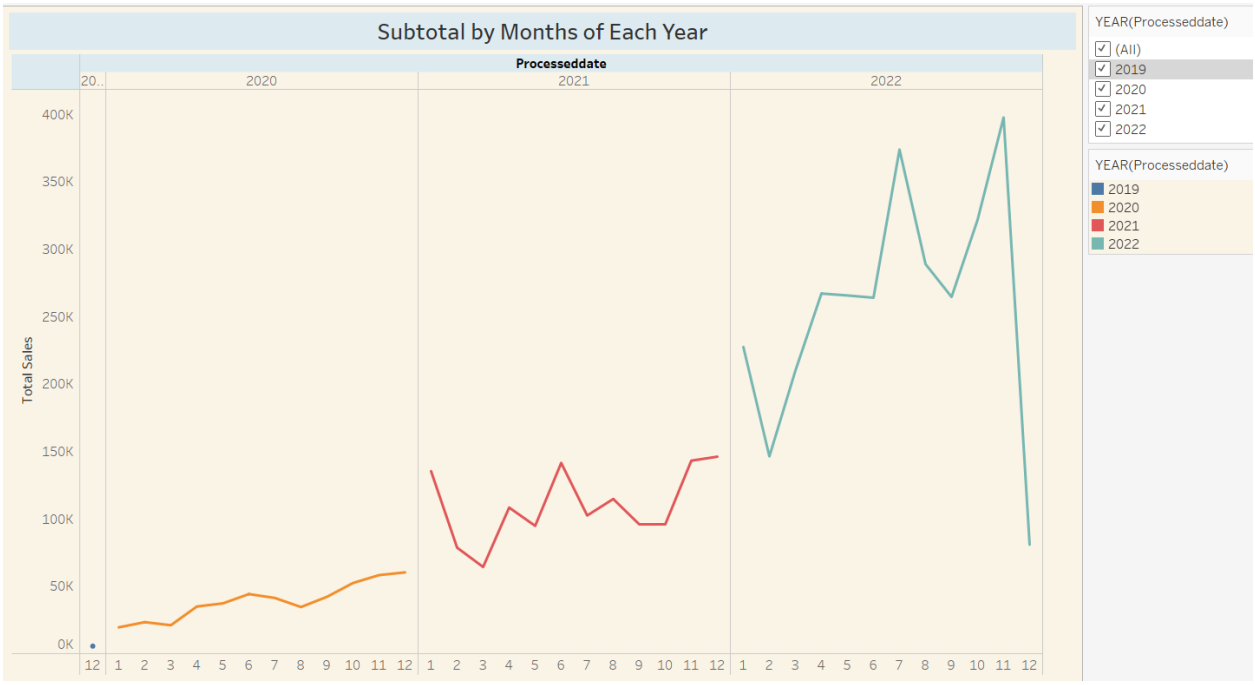


Figure 3-2

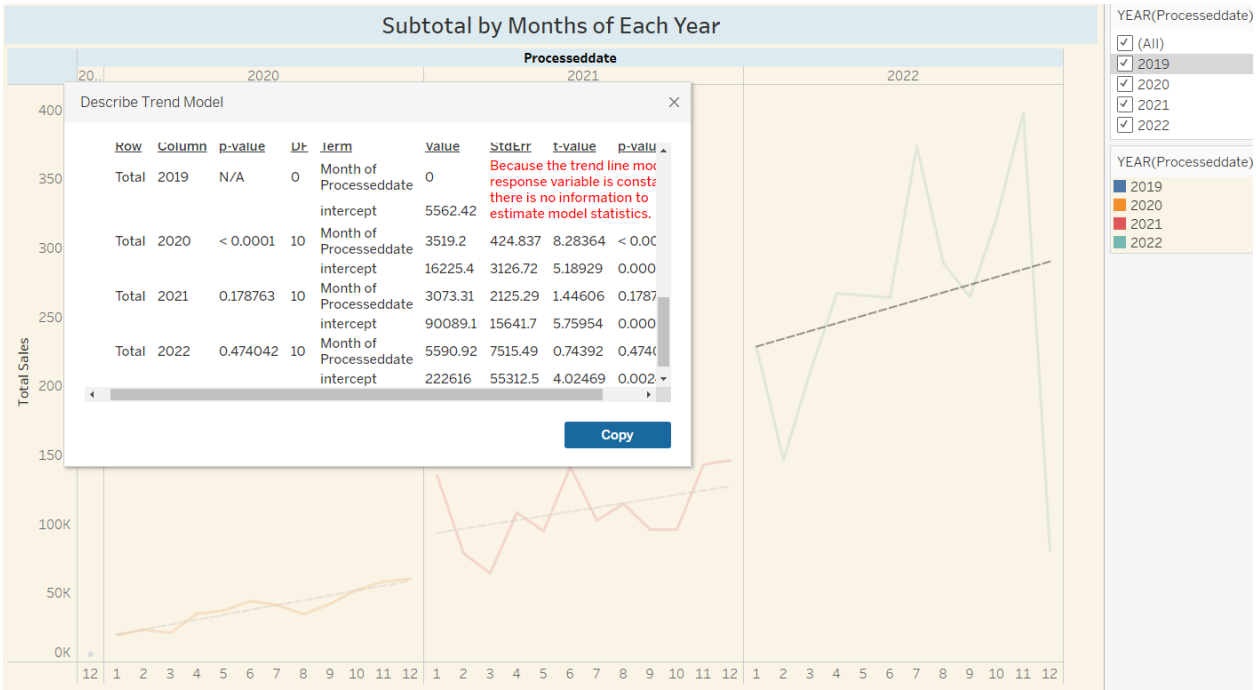


Figure 3-3

Since sales started in December 2019, 2019 was not included in the evaluation. When the graph is analyzed, while a linear sales trend is observed in 2020, slight ups and downs are observed in 2021. However, although there are sharp ups and downs in sales for 2022, there is a similarity between 2021

and 2022. When we examine the p-value values for 2021 and 2022, it is understood that there is no significant difference between the sales of these two years.

1.13.2 Causality Analysis: Effect of War on Sales, Inflation in Turkey, Expectation in Turkish exchange rate, Covid-19 etc.)

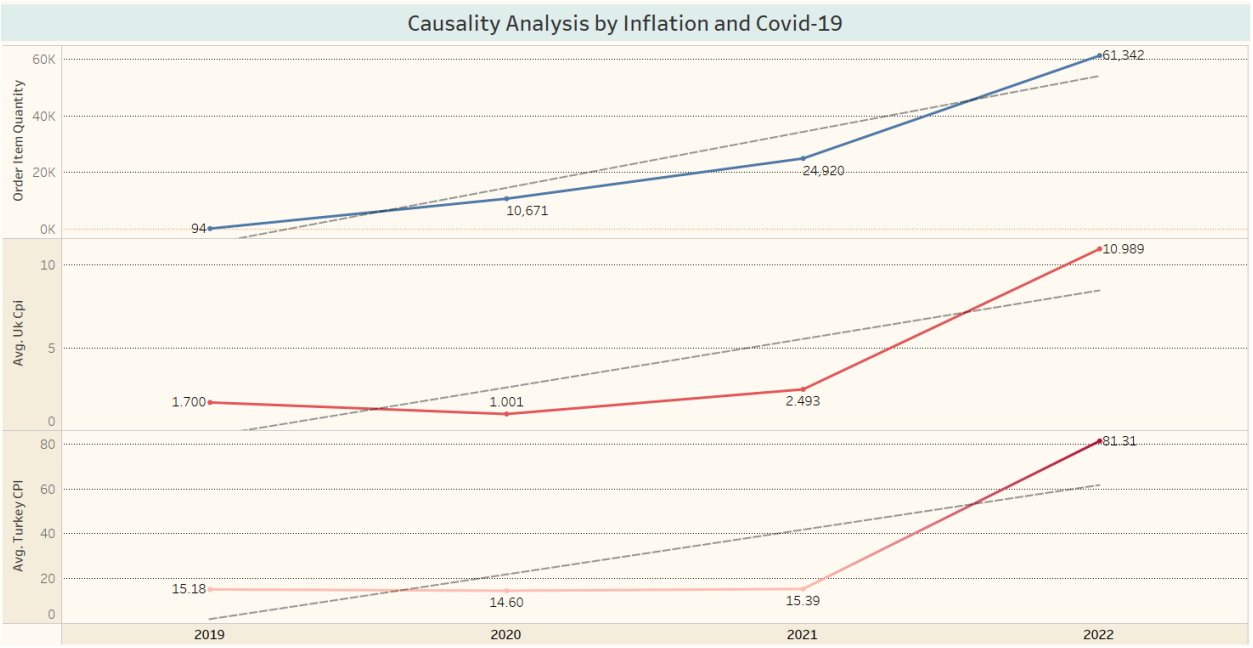


Figure 3-4

Individual trend lines:

Panes		Line		Coefficients				
Row	Column	p-value	DF	Term	Value	StdErr	t-value	p-value
Orderitqnt	Year of Processeddate	0.0438951	2	Year of Processeddate	19799.3	4290.75	4.61441	0.0438951
				intercept	-3.99802e+07	8.66946e+06	-4.61162	0.0439449
Uk Cpi	Year of Processeddate	0.188187	2	Year of Processeddate	2.93606	1.49329	1.96616	0.188187
				intercept	-5928.26	3017.2	-1.96482	0.188376
Turkey CPI	Year of Processeddate	0.223835	2	Year of Processeddate	19.9171	11.4409	1.74086	0.223835
				intercept	-40210.9	23116.4	-1.73949	0.224077

Figure 3-5

As can be seen in the above trend-line chart, the p-value of the order item quantity is 0.043, the p-value of the United Kingdom's inflation feature is 0.18, and the p-value of Turkey's inflation feature is 0.22.

According to this result, it is understood that although there is no significant difference between the annual average inflation values between Turkey and the United Kingdom, there is a significant difference between the orders received over the years.

1.13.3 Is there a difference between the average monthly sales amounts on the sales platforms? (A/B Analysis)

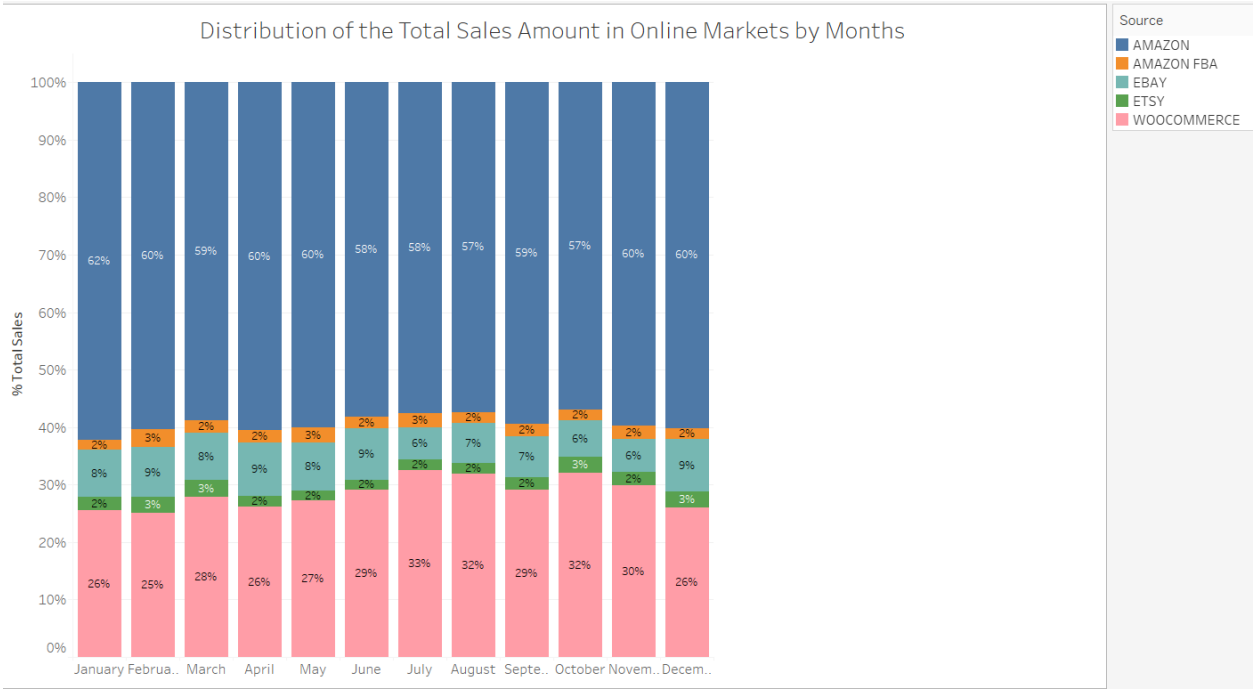


Figure 3-6

In the A-B Analysis; It is seen that the highest monthly sales in online markets are made on Amazon with a range of 57%-62%, Woocommerce as the second with a range of 26-33%, and EBAY as the third with a range of 6%-9%. Monthly sales on Amazon FBA and ETSY vary between 2% and 3%.

1.13.4 In the light of Time Series Analysis, the maximum profitability that can be obtained in the sales to be made within the next six months

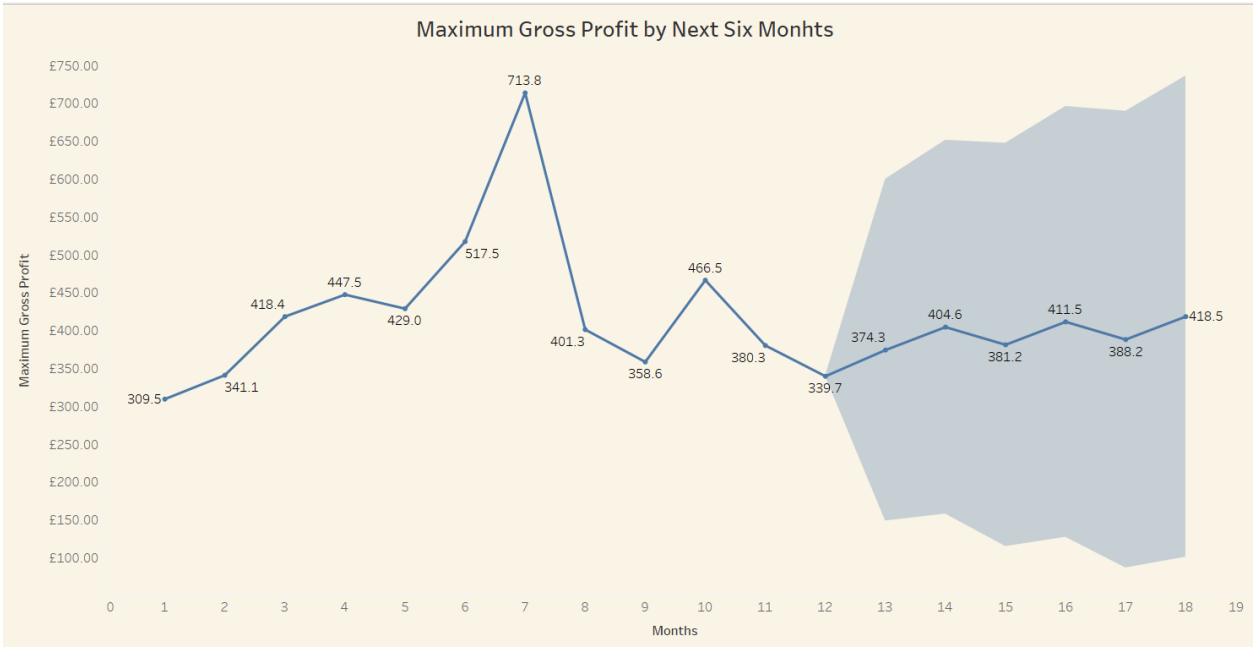


Figure 3-7

1.13.5 Describe Forecasts

Options Used to Create Forecasts

Time series: Dreceieveddate Month
Measures: Max. Profit
Forecast forward: 6 periods (13 – 18)
Forecast based on: 1 – 12
Ignore last: No periods ignored
Seasonal pattern: 2 period cycle

Max. Profit

Initial	Change From Initial	Seasonal Effect		Contribution		
13	13 – 18	High	Low	Trend	Season	Quality
£374.27 ± 60.3%	11.8%	18 3.8%	17 -2.9%	16.3%	83.7%	Poor

Figure 3-8

Maximum profitability is expected to continue "poorly" over the next 6 months.

Similarity of sales volumes of sales platforms (Effect Size Analysis)

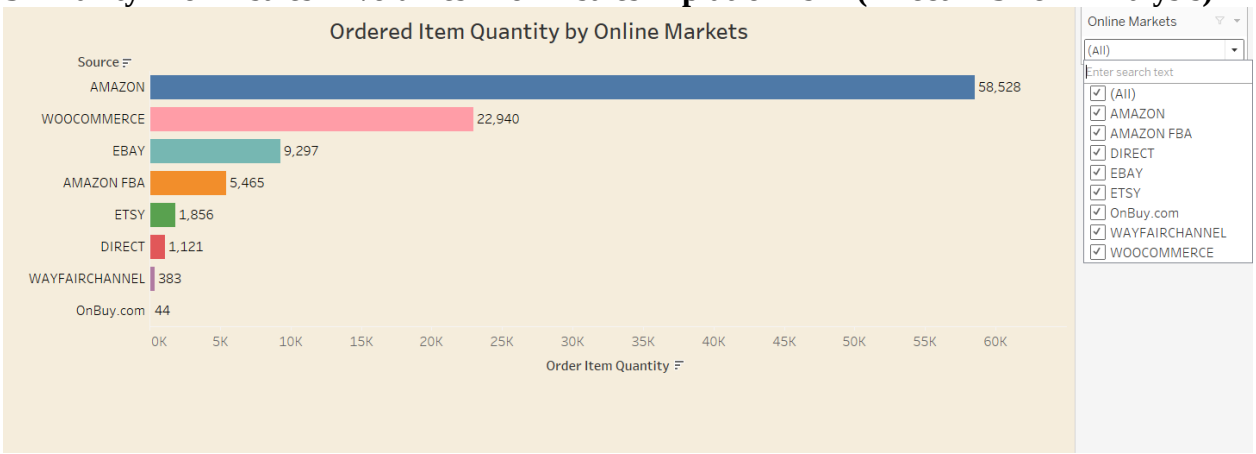


Figure 3-9

In online markets; Amazon is in the first place with 58528 sales, in the second place is Woocommerce with 22940 sales, followed by EBAY, Amazon FBA, ETSY and others. Sales volumes differ according to sales platforms.

1.13.6 UK Monthly Sales Changes

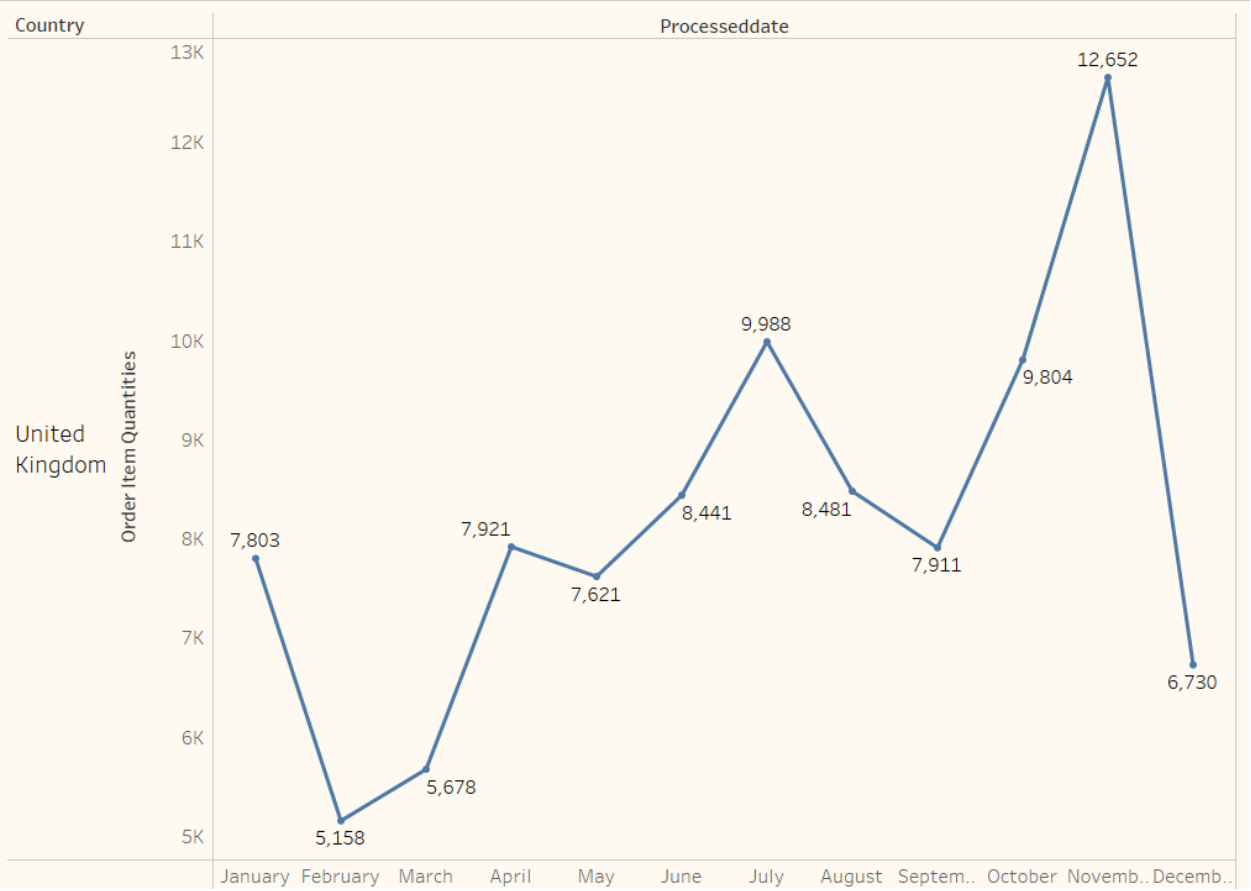


Figure 3-10

Ingilteredeki urun satislari ile aylik ciro arasinda benzerlik gozlemlenmektedir. Ocak ayinda dusus yasandigi, subat ve mart aylarinda ise, yukselisin devam ettigi ancak nisan ve mayis aylari arasinda kismen satabil denilebilecek bir durum yasandigi gozlenmektedir. Mayis ayindan sonra, temmuz ayina kadar yukselisin devam ettigi fakat temmuz ve agustos aylarinda dusus yasandigi gorulmektedir. Buna mukabil, eylul ayindan itibaren aralik ayina kadar yukselis devam ettigi gorulmektedir. Aralik ayinda ise, ciddi bir dusus yasandigi gozlenmektedir.

Comment: The reason for the decrease in July and August; since it is a holiday season, it is considered that the reason for the rise from September to December may be due to the high season. The reason for the decrease in sales in December is considered to be due to the fact that the dataset we have contains the latest data dated December 9th. In fact, since there is no data on the whole month, a healthy evaluation cannot be made.

1.13.7 Monthly Sales Trends (Line Graph)

Figure 3-11

Describe Trend Model

A linear trend model is computed for sum of Total given Processeddate Month.

Model formula:	(Month of Processeddate+intercept)
Number of modelled observations:	12
Number of filtered observations:	0
Model degrees of freedom:	2
Residual degrees of freedom (DF):	10
SSE (sum squared error):	8.52835e+10
MSE (mean squared error):	8.52835e+09
R-Squared:	0.204904
Standard error:	92349.1
p-value (significance):	0.139502

Individual trend lines:

Panes		Line		Coefficients				
Row	Column	p-value	DF	Term	Value	StdErr	t-value	p-value
Total	Month of Processeddate	0.139502	10	Month of Processeddate	12397.4	7722.62	1.60533	0.139502
				intercept	328004	56836.9	5.77096	0.0001799

Figure 3-12

The P-value is seen as 0.139. Since this value is greater than 0.05, it shows that there is no significant difference between monthly sales trends.

1.13.8 Distribution of Orders, Sales and Profits in the UK

Figure 3-13

1.13.9 Distribution of Order, Sales and Profits in the Top 8 Countries Except UK

Figure 3-14

There are eight countries where the company receives the highest number of orders after the UK. Since the order numbers of other countries are 1, they are not shown in this chart. In these countries; when the number of orders, sales and gross profits are examined, it is seen that there is a correct ratio in countries other than the first 3 countries. For example, although the number of orders in Ireland is more than Germany and Italy, it is understood that the subtotal and profit are less than these countries. From these results, it can be understood that the carpet preference of the countries varies according to the price and the type of carpet.

The Relationship Between Ad Spend and Revenue

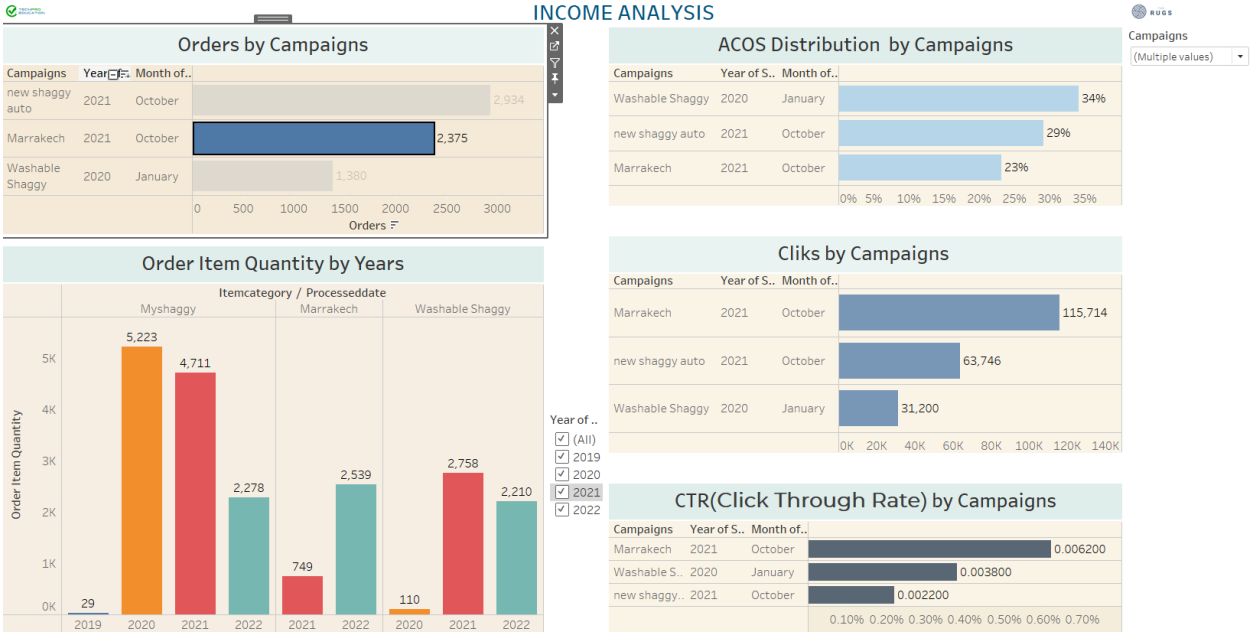


Figure 3-15

Recommendation: First of all, we need to be able to establish a relationship between the 'Sales and Amazon campaigns' tables so that we can measure the relationship between advertising expenditures and revenue. First of all, it is necessary to know which product category the advertisement application is applied to. Since this match could not be made properly, a relationship was tried to be established between the campaign names and the product category names in the sales table.
Therefore, it is evaluated that a healthy analysis can be made by adding the ASIN number in the "Sales" table to the "Amazon campaigns" table.

Comment: It is seen that the advertising campaign for the New Shaggy product was made in October 2021 and 2934 units were sold. When the sales volumes for the years are considered, it is seen that it lags behind the previous year and there is a decrease of approximately 50% in 2022 compared to 2021.

However, when the CTR ratio of the product is considered, it is seen that it is 0.0022 and the sales amount is 2934. Although we do not know the profit margin of the company, it can be said that the sales increased after the advertisement application.

4. Conclusions and recommendations

- We need to be able to establish a relationship between the 'Sales and Amazon campaigns' tables so that we can measure the relationship between advertising expenditures and revenue. First of all, it is necessary to know which product category the advertisement application is applied to. Since this match could not be made properly, a relationship was tried to be established between the campaign names and

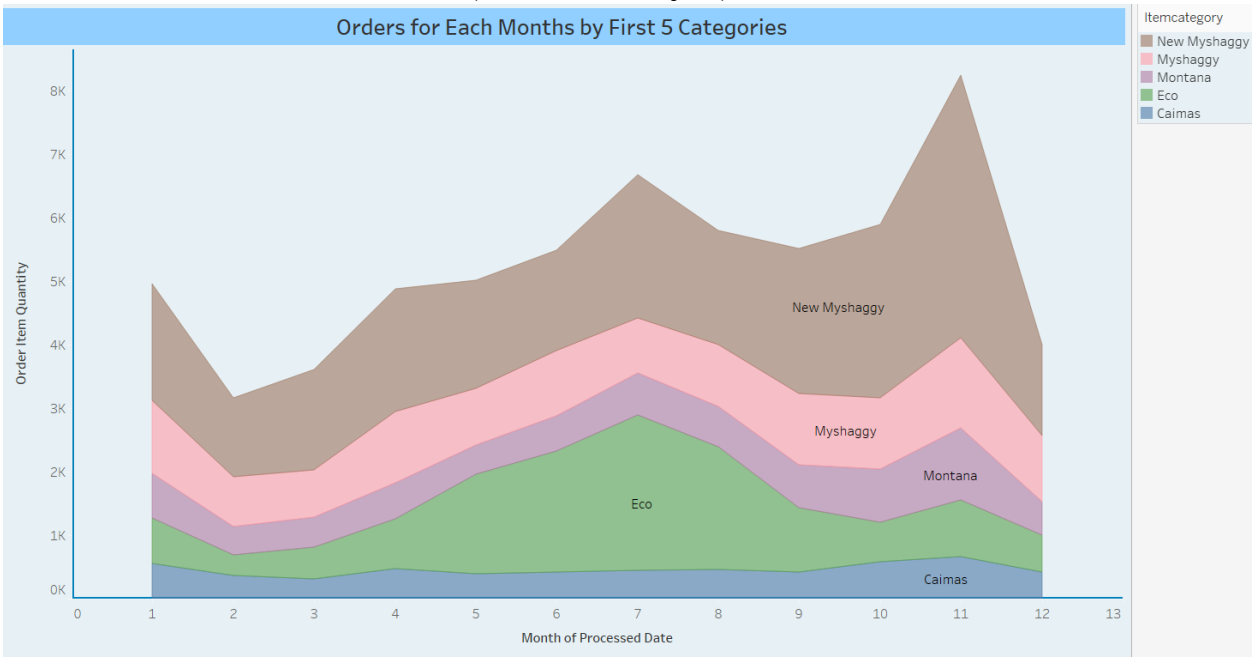
the product category names in the sales table. Therefore, it is evaluated that a healthy analysis can be made by adding the ASIN number in the "Sales" table to the "Amazon campaigns" table.

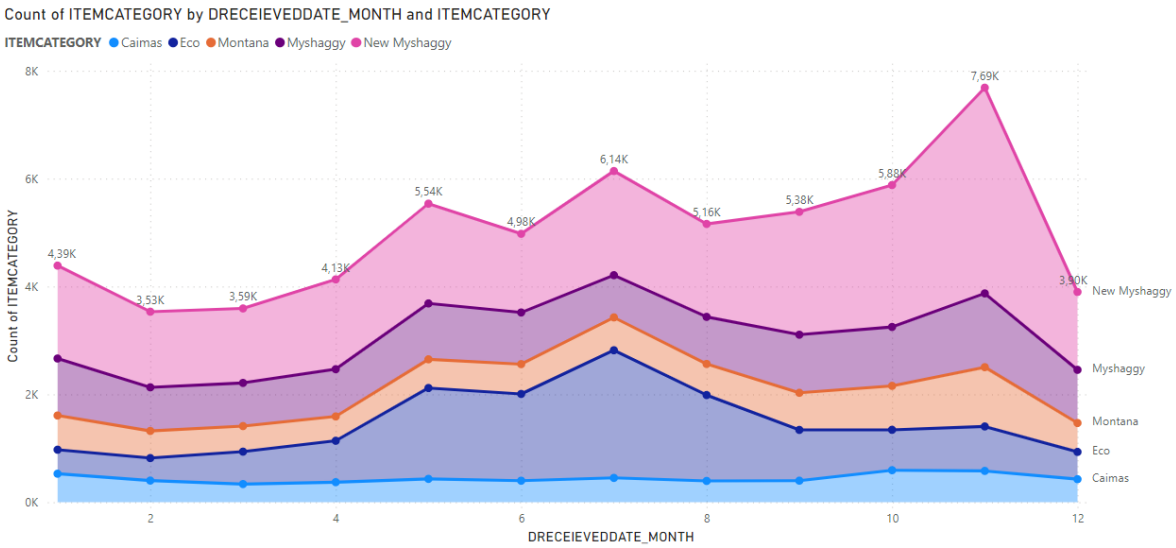
- In some products, it was observed that advertising did not have a positive effect on sales. For example, the My Shaggy model. It is considered that a review of the advertising strategy may be useful.
- It has been observed that Amazon and website sales rates are considerably higher than other platforms. In this context, it is thought that it may be useful to review sales strategies in order to increase the amount of sales on other online platforms.
- According to our data, the profit per product is 33 pounds in UK, 40 pounds in Germany, 53 pounds in Italy, 42 pounds in France and 55 pounds in Spain. Increasing sales to these countries will also increase profitability.
- According to the 6-month and 1-year forecast analyses, it is seen that the sales trend will continue upwards.
- When the total sales and order status in Amazon are evaluated, it is considered that it may be beneficial for the company to make sales on Amazon in other European countries.
- Sales are mostly made on the amazon platform. In this case, sales can be focused on sales platforms such as Etsy, Ebay.
- In general, sales are highest between June and September.

3. SPRINT 3

1.14 PRODUCT ANALYSIS

Which Product is Sold and When (Seasonal Analysis)

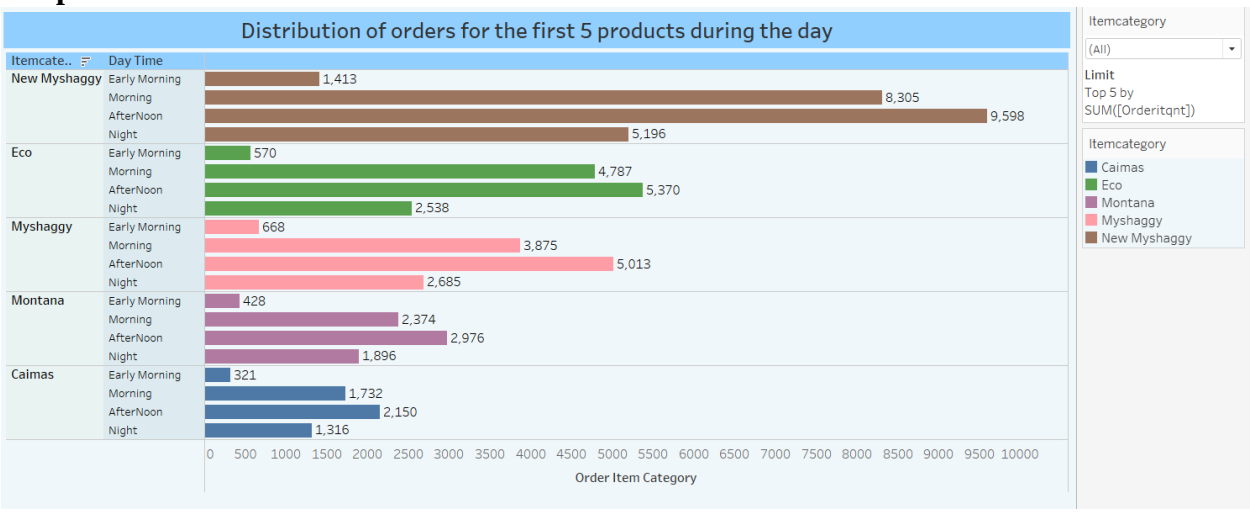


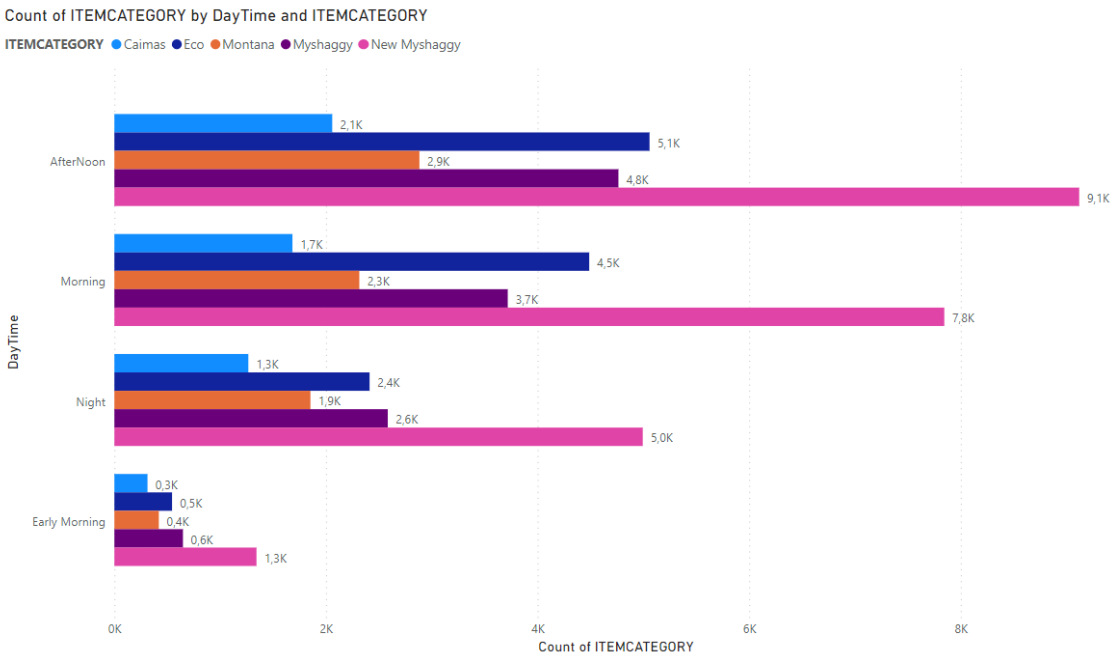


In this analysis, five best-selling products were examined. When the 5 best selling seasonal products are examined, it is seen that the sales trend of the product “Caimas” increased in September, peaked in November and started to decline in December.

When we look at the "New Myshaggy" in the first row, it is observed that there is a decrease from January to February, but there is a rise from February to July, there is a decrease in July-August, but a sharp upward trend is observed again from September. It is understood that this situation is the same for the products in the 2nd, 3rd and 4th rows.

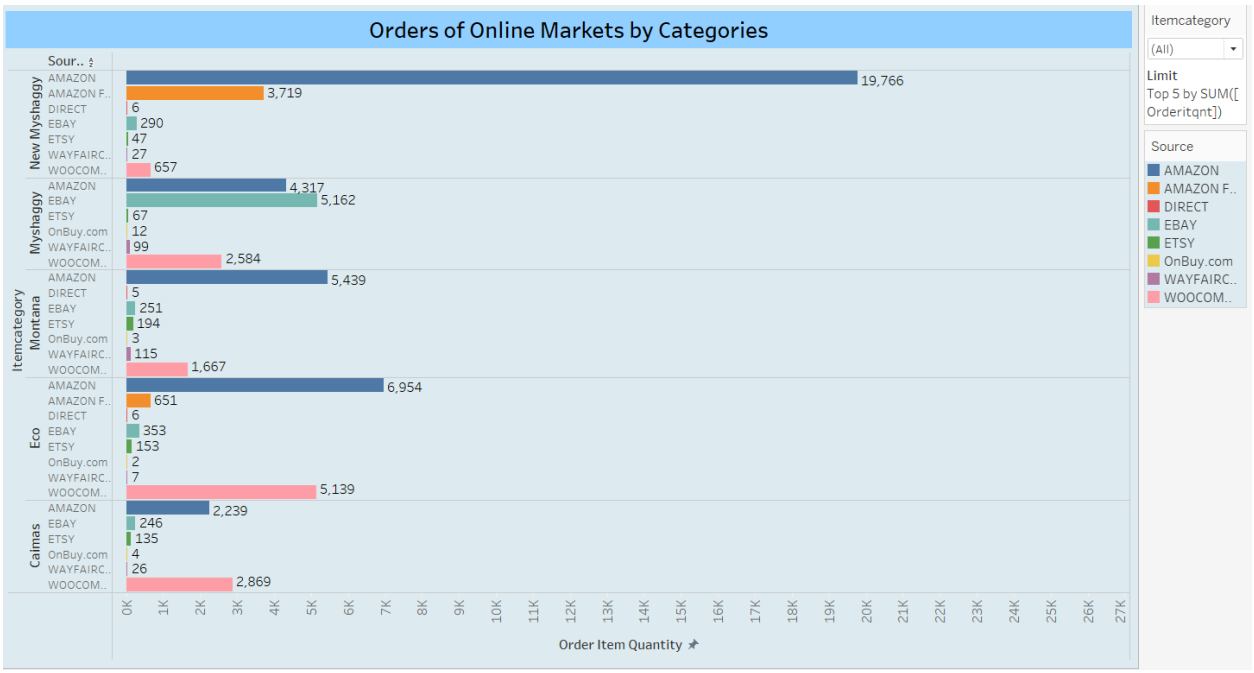
Graph-2

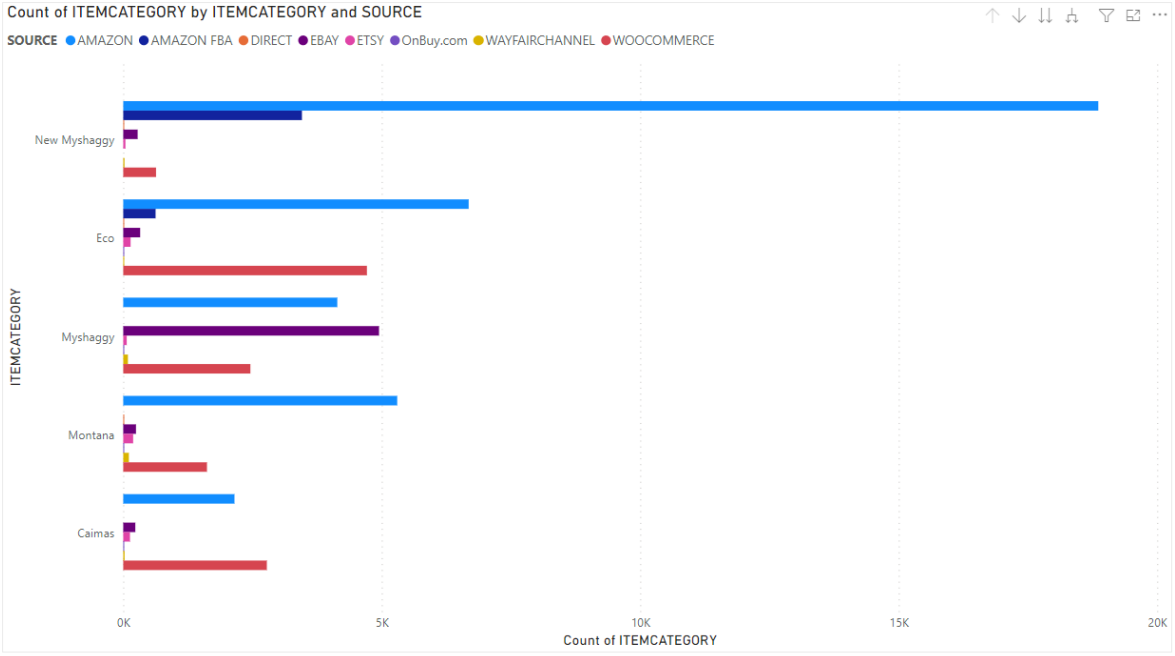




When the products in the first 5 categories are examined in which time period during the day; it is seen that the most sales are made in the afternoon, in the morning, in the evening and in the early morning, according to the order.

1.14.1 Which Product is Sold on Which Platform and How Many?

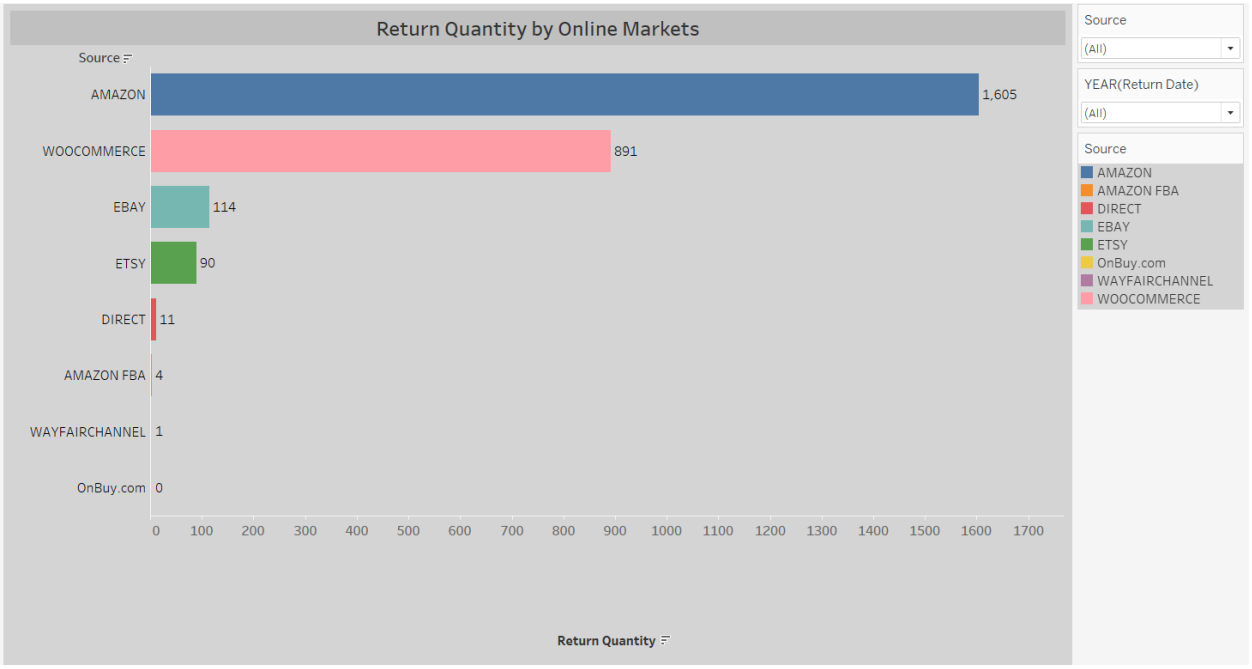


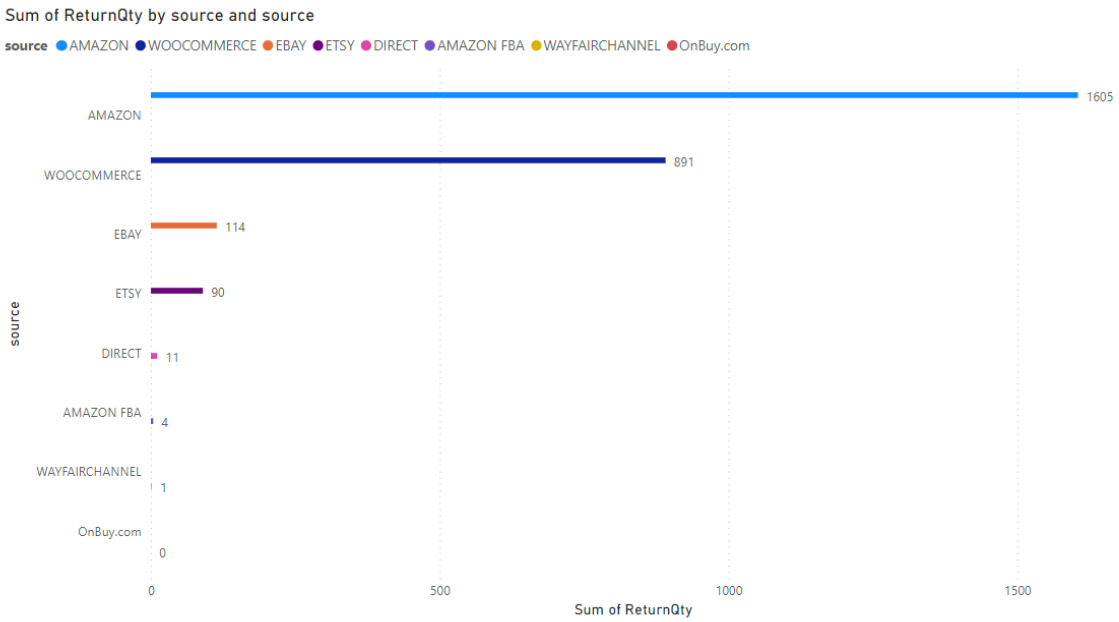


When the 5 best-selling products in the product category are examined, it is seen that the best-selling “New Myshaggy” is sold on Amazon, “Mysaggy” in the 2nd place is sold on EBAY, Montana in the 3rd place and Eco in the 4th place, in the 5th place and in the Amazon. It is seen that the “Caimas” type is sold at Woocommerce.

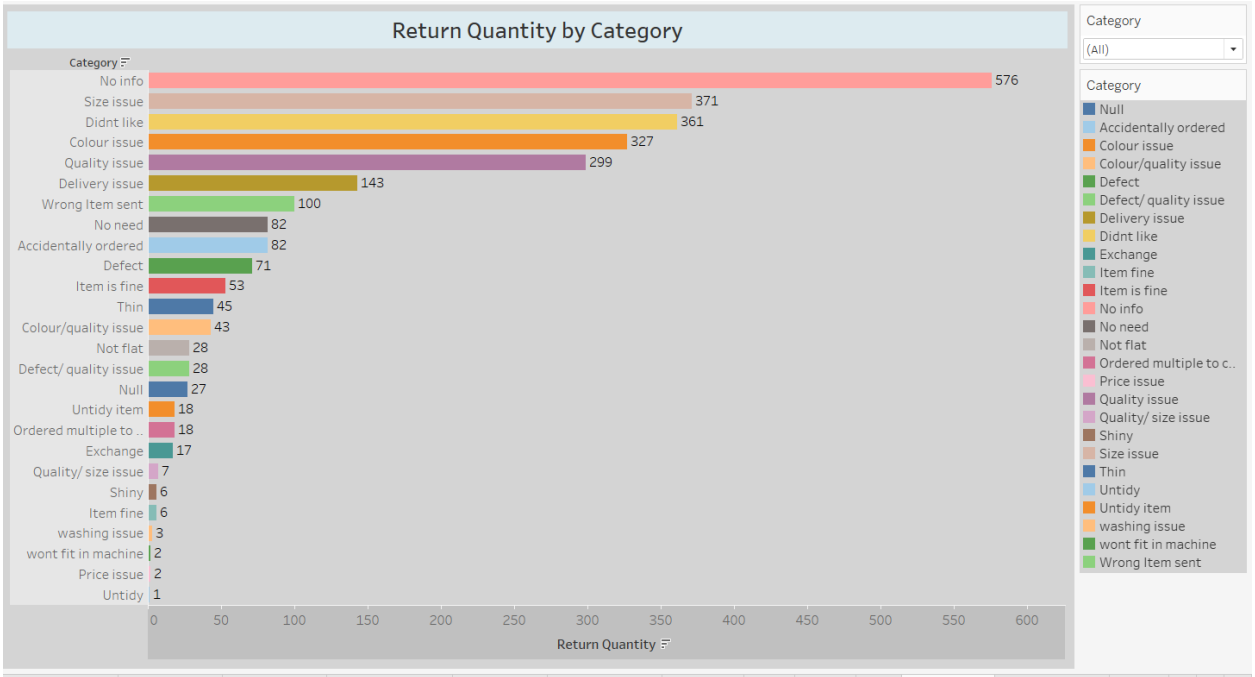
According to these results, the following can be said, the sales of each product on online platforms may differ.

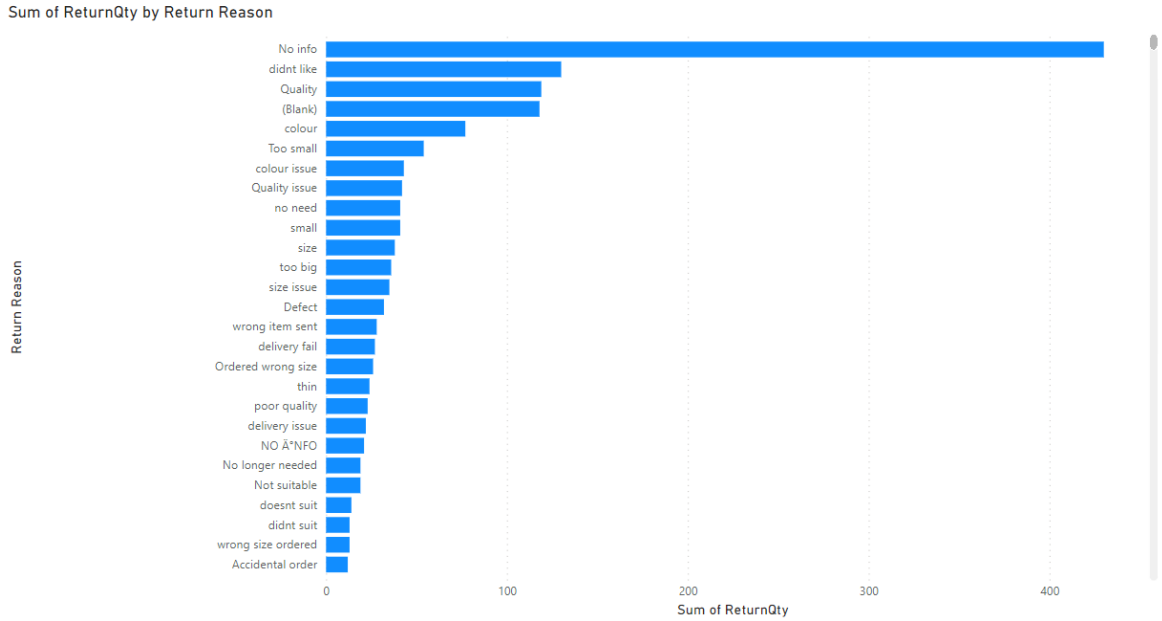
Analysis of Returned Products and Reasons for Return





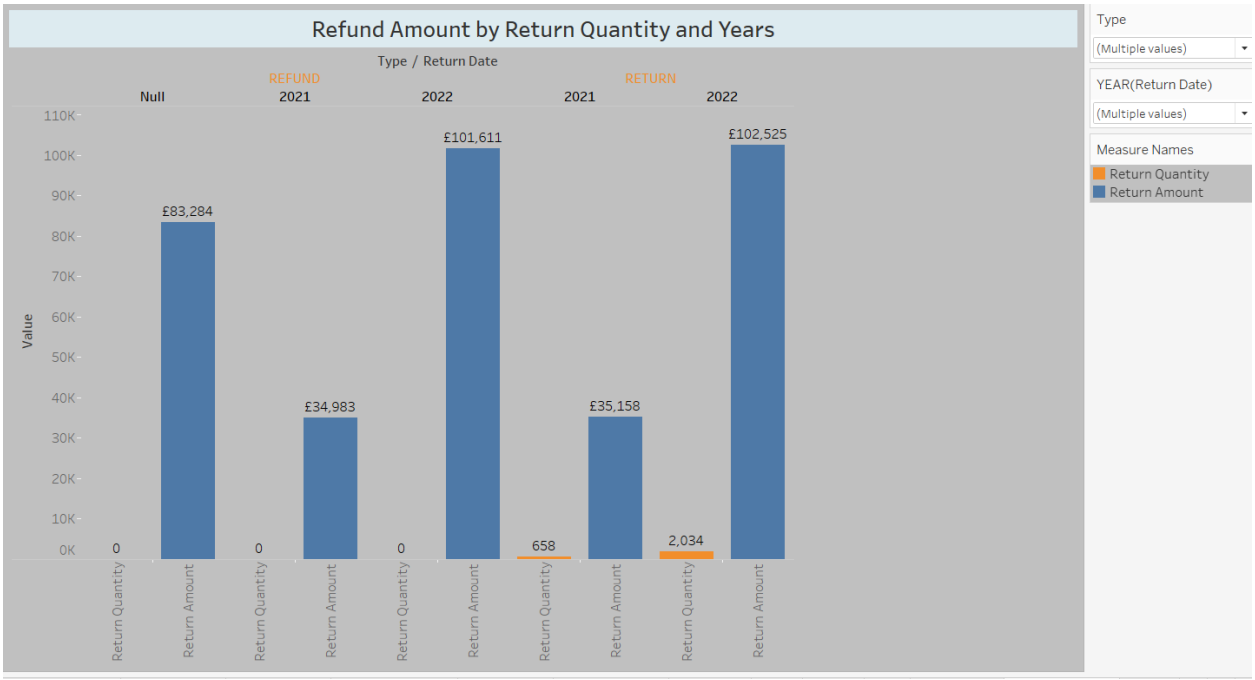
It is seen that the most returned products are from Amazon and then from Woocommerce. It is understood that this result is directly proportional to the orders received from online platforms.





Considering the number and justifications of the returned products, the customer returned the first 576 products without giving any reason. 2. The customer returned 361 items in the order due to the size of the product. 3. If the next 327 items are returned, the product was made because the product was not liked. Returns in the 4th, 5th and 6th rows were made due to quality, color and distribution problems.

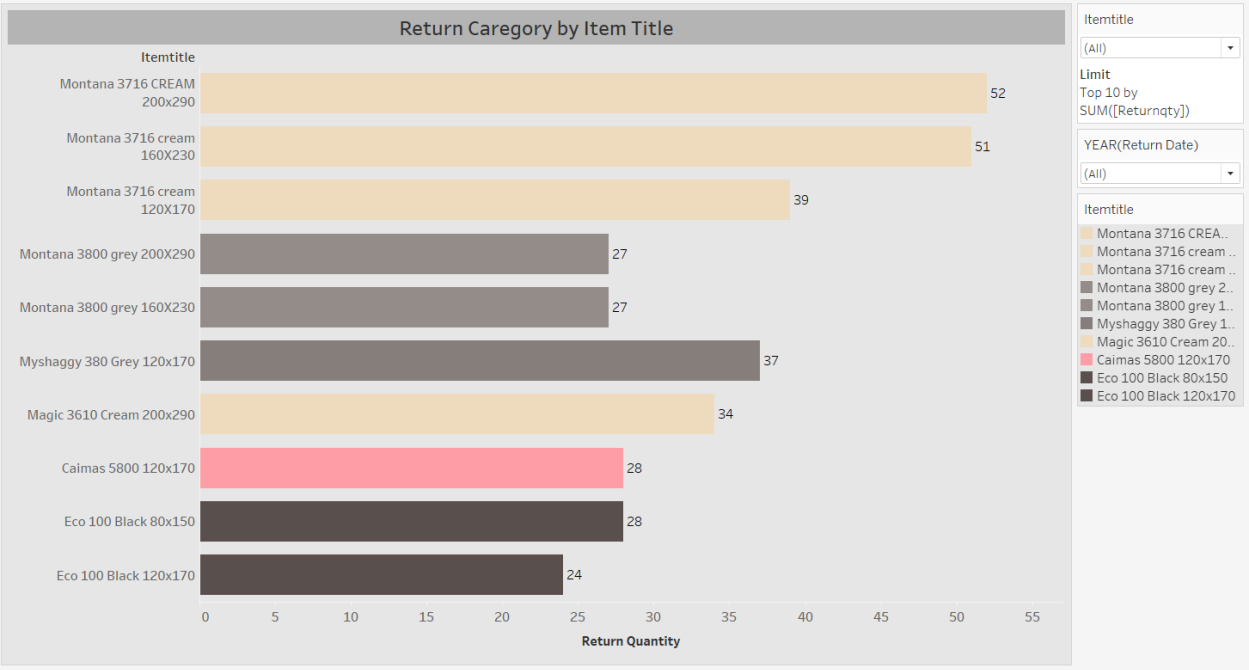
Regarding these results, this can be said; The customer can return the product for different reasons. Therefore, the seller should review the accuracy of the product information and the ability of the images to reflect the product in order to ensure customer satisfaction. Every stage of online sales should be done with care and customer relations should be well managed.

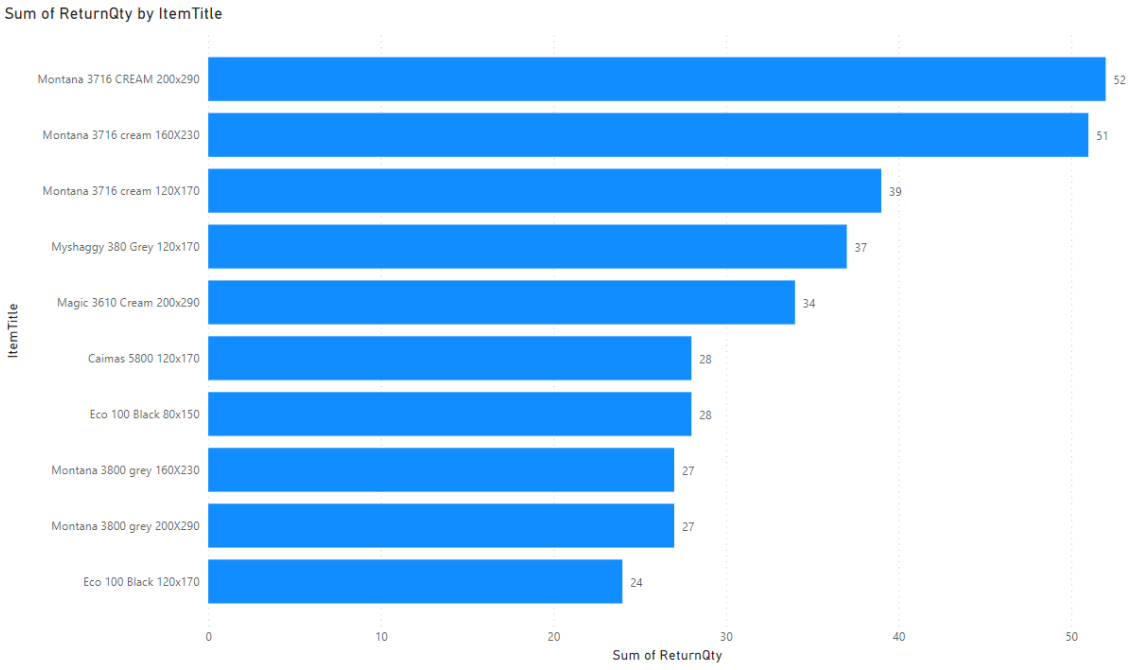


When the types of returned products are examined by years, it is seen that the refund amount in the years 2021 and 2022 is 101611 £ and the year 2022 at the most, in an unspecified year.

When the return status of return type products is examined, it is seen that 658 products were returned in 2021 and 2034 products in 2022. It is understood that these returns are directly proportional to the year in which the turnover is made.

Note: In Dataset, if the type of returned products is non-refundable, the return quantity is zero(0).

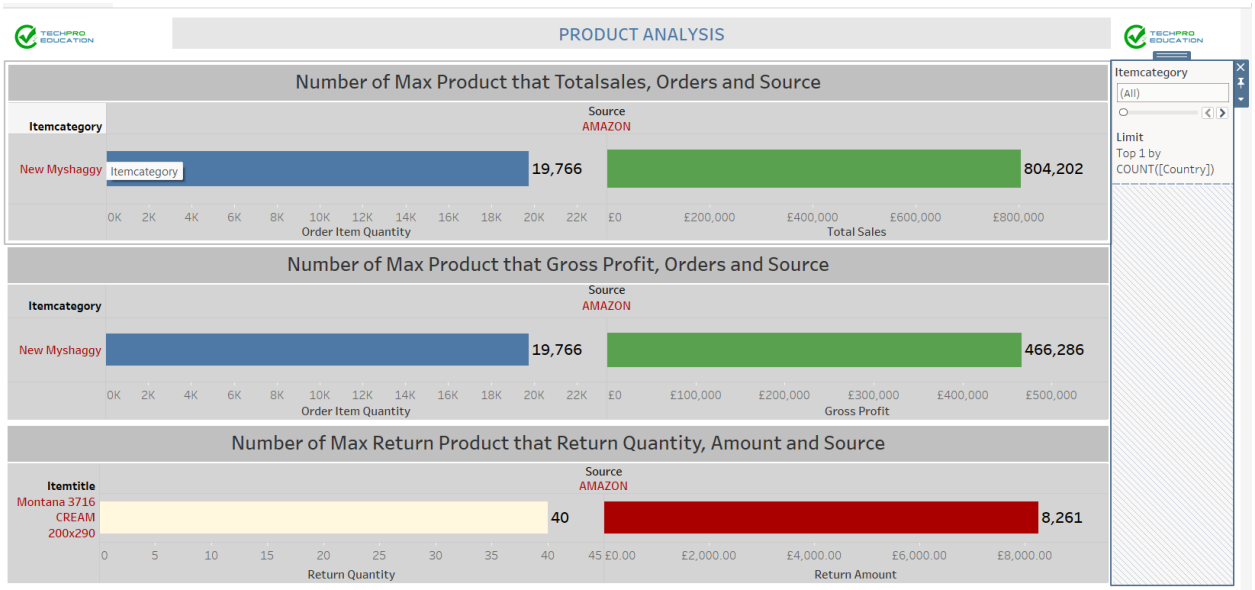




When the 10 most returned product categories are examined, it is seen that the large sizes (120x170, 160x230 and 200x290) of the Montana3716 cream and Montana 3800 gray categories are returned.

When the other returned categories are examined, it is seen that cream, gray and black colors are returned.

It is considered that it would be beneficial for the company to review the sales of these color products.



It is seen that 19766 units were sold from the amazon platform in the “New Myshaggy” tour, with a total turnover of 466286 pounds.

It is seen that the highest gross profit was obtained from the “New Myshaggy” category sold on amazon and £466286.

It is understood that the most returns from the products are made in the size of “Montana 3716 cream 200x290” and the cost of this is £ 8261.

1.15 Market Basket Analysis with Apriori

What is Apriori?

Apriori is a popular algorithm for extracting frequent itemsets with applications in association rule learning. The apriori algorithm has been designed to operate on databases containing transactions, such as purchases by customers of a store. An itemset is considered as "frequent" if it meets a user-specified support threshold. For instance, if the support threshold is set to 0.5 (50%), a frequent itemset is defined as a set of items that occur together in at least 50% of all transactions in the database.

In the shopping made in 2022, we selected the customers who bought more than one product. 1641 different products have been added to the basket in 2570 different shopping.

```
    teveri = te.fit_transform(sepet2)

    sepet2 = pd.DataFrame(teveri, columns=te.columns_)
    print(sepet2)
```

[21]

```
Output exceeds the size limit. Open the full output data in a text editor
      382 Blush Pink 160x220   382 Blush Pink 200x280   382 Brown 140x200  \
0                      False                    False              False
1                      False                    False              False
2                      False                    False              False
3                      False                    False              False
4                      False                    False              False
...                      ...                      ...                ...
2565                   False                    False              False
2566                   False                    False              False
2567                   False                    False              False
2568                   False                    False              False
2569                   False                    False              False

      382 Brown 200x280   382 Brown 70x140   382 Brown 70x250  \
0                 False              False              False
1                 False              False              False
2                 False              False              False
3                 False              False              False
4                 False              False              False
...                 ...                ...                ...
2565              False              False              False
2566              False              False              False
2567              False              False              False
2568              False              False              False
2569              False              False              False
```

```
df1 = apriori(sepet2, min_support = 0.0025, use_colnames=True)
print(df1)
```

```
      support                                           itemsets
0    0.011673                                        ( Bedroom)
1    0.003113                              ( Caimas 2790 160x230)
2    0.002724                                ( Caimas 2971 45x75)
3    0.003113                                ( Caimas 2980 40x60)
4    0.004669                           ( EFES 7437 grey 160X230)
..        ...                                               ...
221  0.011673  (THE RUGS Ultra Soft Area Rug - Modern Luxury ...
222  0.011673  (THE RUGS Ultra Soft Area Rug - Modern Luxury ...
223  0.011673  (THE RUGS Ultra Soft Area Rug - Modern Luxury ...
224  0.011673  ( Grey - Dark Grey), THE RUGS Ultra Soft Area ...
225  0.011673  ( Grey - Dark Grey),  Grey Plain Pattern Rugs ...

[226 rows x 2 columns]
```

We selected the minimum support value as 0.0025 and the minimum threshold of our confidence metric as 0.25. And we got the following results:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Grey - Dark Grey))	(Bedroom)	0.011673	0.011673	0.011673	1.000000	85.666667	0.011537	inf
1	(Bedroom)	(Grey - Dark Grey))	0.011673	0.011673	0.011673	1.000000	85.666667	0.011537	inf
2	(Bedroom)	(Grey Plain Pattern Rugs for Living Room)	0.011673	0.011673	0.011673	1.000000	85.666667	0.011537	inf
3	(Grey Plain Pattern Rugs for Living Room)	(Bedroom)	0.011673	0.011673	0.011673	1.000000	85.666667	0.011537	inf
4	(Bedroom)	(Kids Room (80x150 cm)	0.011673	0.011673	0.011673	1.000000	85.666667	0.011537	inf
5	(Kids Room (80x150 cm)	(Bedroom)	0.011673	0.011673	0.011673	1.000000	85.666667	0.011537	inf
6	(THE RUGS Ultra Soft Area Rug - Modern Luxury ...	(Bedroom)	0.011673	0.011673	0.011673	1.000000	85.666667	0.011537	inf
7	(Bedroom)	(THE RUGS Ultra Soft Area Rug - Modern Luxury ...	0.011673	0.011673	0.011673	1.000000	85.666667	0.011537	inf
8	(Eco 100 Black 120x170)	(Eco 100 Black 200x290)	0.005837	0.010895	0.003502	0.600000	55.071429	0.003438	2.472763
9	(Eco 100 Black 200x290)	(Eco 100 Black 120x170)	0.010895	0.005837	0.003502	0.321429	55.071429	0.003438	1.465083
10	(Eco 100 Black 120x170)	(Eco 100 Black 160x230)	0.007393	0.010117	0.002724	0.368421	36.417004	0.002649	1.567315
11	(Eco 100 Black 160x230)	(Eco 100 Black 120x170)	0.010117	0.007393	0.002724	0.269231	36.417004	0.002649	1.358304
12	(Eco 100 Black 160x230)	(Eco 100 Black 200x290)	0.010117	0.010895	0.003113	0.307692	28.241758	0.003003	1.428707
13	(Eco 100 Black 200x290)	(Eco 100 Black 160x230)	0.010895	0.010117	0.003113	0.285714	28.241758	0.003003	1.385837
14	(Grey - Dark Grey))	(Grey Plain Pattern Rugs for Living Room)	0.011673	0.011673	0.011673	1.000000	85.666667	0.011537	inf
15	(Grey Plain Pattern Rugs for Living Room)	(Grey - Dark Grey))	0.011673	0.011673	0.011673	1.000000	85.666667	0.011537	inf

We achieved the same result by using FP-Growth instead of Apriori. FP-Growth runs 6x faster.

In general, the algorithm has been designed to operate on databases containing transactions, such as purchases by customers of a store. An itemset is considered as "frequent" if it meets a user-specified support threshold. For instance, if the support threshold is set to 0.5 (50%), a frequent itemset is defined as a set of items that occur together in at least 50% of all transactions in the database.

In particular, and what makes it different from the Apriori frequent pattern mining algorithm, FP-Growth is an frequent pattern mining algorithm that does not require candidate generation. Internally, it uses a so-called FP-tree (frequent pattern tree) datastrucure without generating the candidate sets explicitely, which makes is particularly attractive for large datasets.

```
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.25)
rules
```

Python

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Myshaggy 380 D.Grey 140x200)	(Myshaggy 380 D.Grey 80x150)	0.007782	0.022179	0.003502	0.450000	20.289474	0.003329	1.777856
1	(Myshaggy 380 Duckegg 80x150)	(Myshaggy 380 Duckegg 60x110)	0.009728	0.009339	0.003113	0.320000	34.266667	0.003022	1.456855
2	(Myshaggy 380 Duckegg 60x110)	(Myshaggy 380 Duckegg 80x150)	0.009339	0.009728	0.003113	0.333333	34.266667	0.003022	1.485409
3	(Myshaggy 380 Beige 80x150)	(Myshaggy 380 Beige 60x110)	0.021401	0.010895	0.007393	0.345455	31.707792	0.007160	1.511133
4	(Myshaggy 380 Beige 60x110)	(Myshaggy 380 Beige 80x150)	0.010895	0.021401	0.007393	0.678571	31.707792	0.007160	3.044531
...	...	...	...	...	...	...	...	...	...
201	(Eco 100 Black 200x290)	(Eco 100 Black 120x170)	0.010895	0.005837	0.003502	0.321429	55.071429	0.003438	1.465083
202	(Myshaggy 380 Grey 140x200)	(Myshaggy 380 Grey 120x170)	0.012840	0.017510	0.003502	0.272727	15.575758	0.003277	1.350924
203	(Magic 3610 Cream 200x290)	(Magic 3610 Cream 80x150)	0.007393	0.004669	0.002724	0.368421	78.903509	0.002689	1.575940
204	(Magic 3610 Cream 80x150)	(Magic 3610 Cream 200x290)	0.004669	0.007393	0.002724	0.583333	78.903509	0.002689	2.382257
205	(Montana 3716 cream 80X150)	(Montana 3716 cream 160X230)	0.003113	0.012840	0.002724	0.875000	68.143939	0.002684	7.897276

206 rows × 9 columns

We read the table as:

Probability of getting (Myshaggy 380 D.Grey 140x200) product : **0,007782** [antecident support]

Probability of getting (Myshaggy 380 D.Grey 80x150) product: **0,022179** [consequent support]

Probability of getting both: **0,003502** [support]

The probability that the second will be added to the basket when the 1st is bought: **0,45** [confidence]

1.16 COMPUTER VISION

Data Set Preparation

The pre-trained YOLOV5 model was chosen to set up this machine learning model. The main reason for using this model is that it is stable and easy to set up compared to many Computer Vision models**.**

First, the carpet pictures were grouped as 30 pictures for each color group from the-rugs.com website. The main purpose here was to create an object detection model based on the state colors.

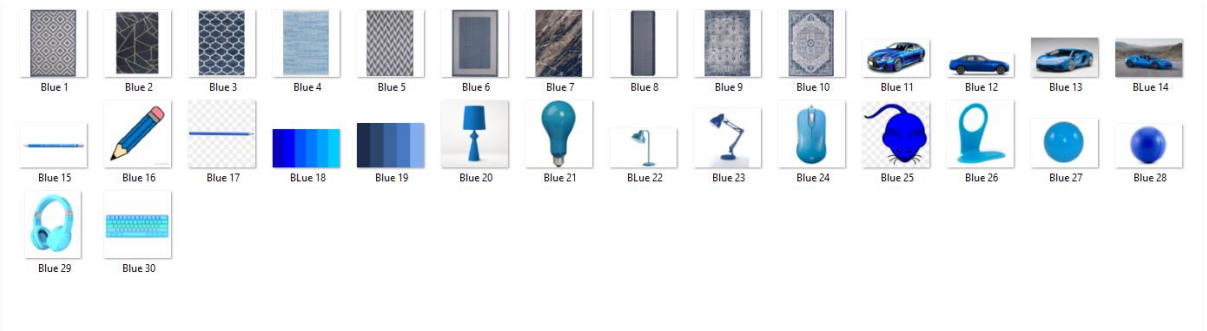
There were about 14 color categories for different types of carpet models on the web-site.

These are:

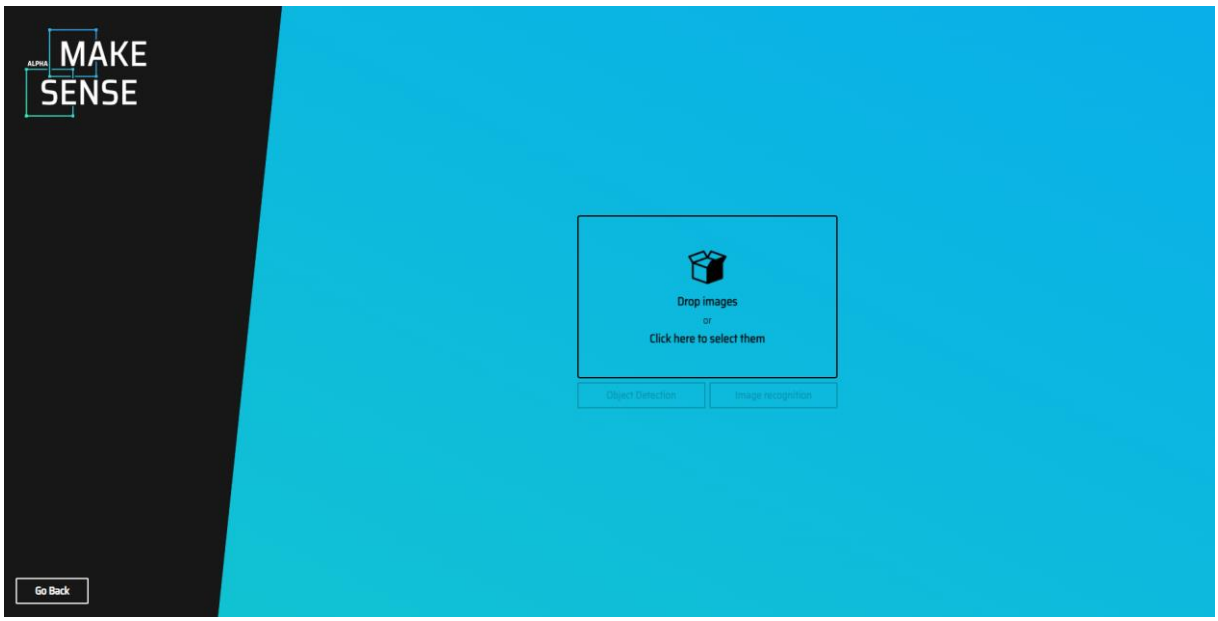
- Beige
- Black
- Blue
- Brown

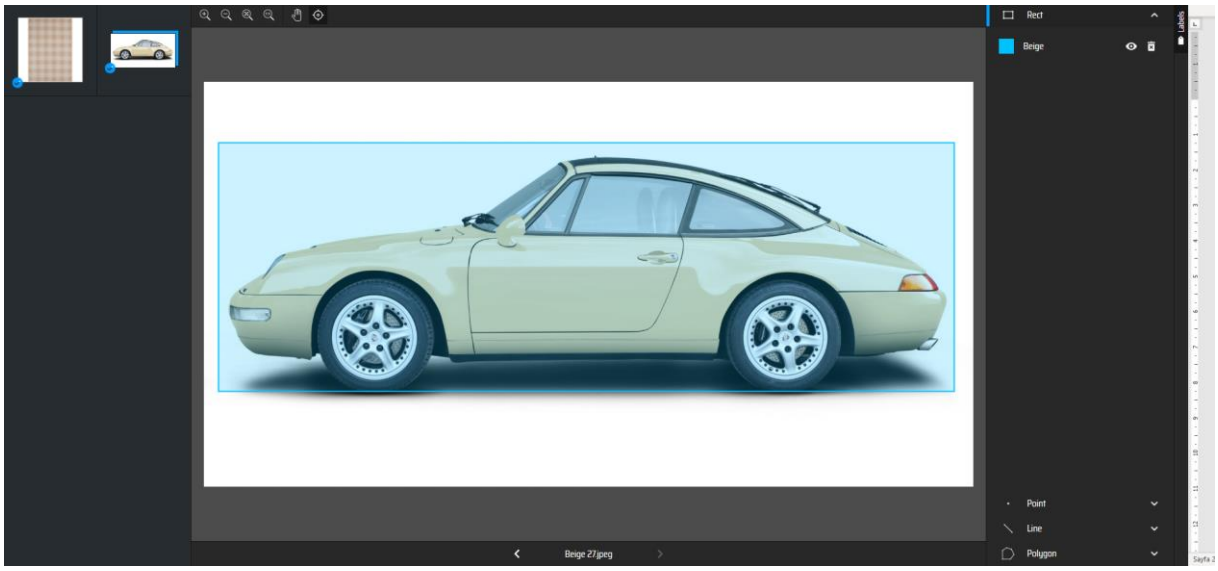
- Duck Egg
- Green
- Grey
- Multicolor
- Orange
- Pink
- Purple
- Red
- White
- Yellow

But in these color categories; There was not enough carpet model on sale to train our model. In this context, it has been determined that the Yolo object detection model can be used to detect the color of different objects other than carpets. Objects other than carpet, such as car, pen, etc. defined by color in to the YOLOV5 model. Insufficient data in the dataset has been there by completed.

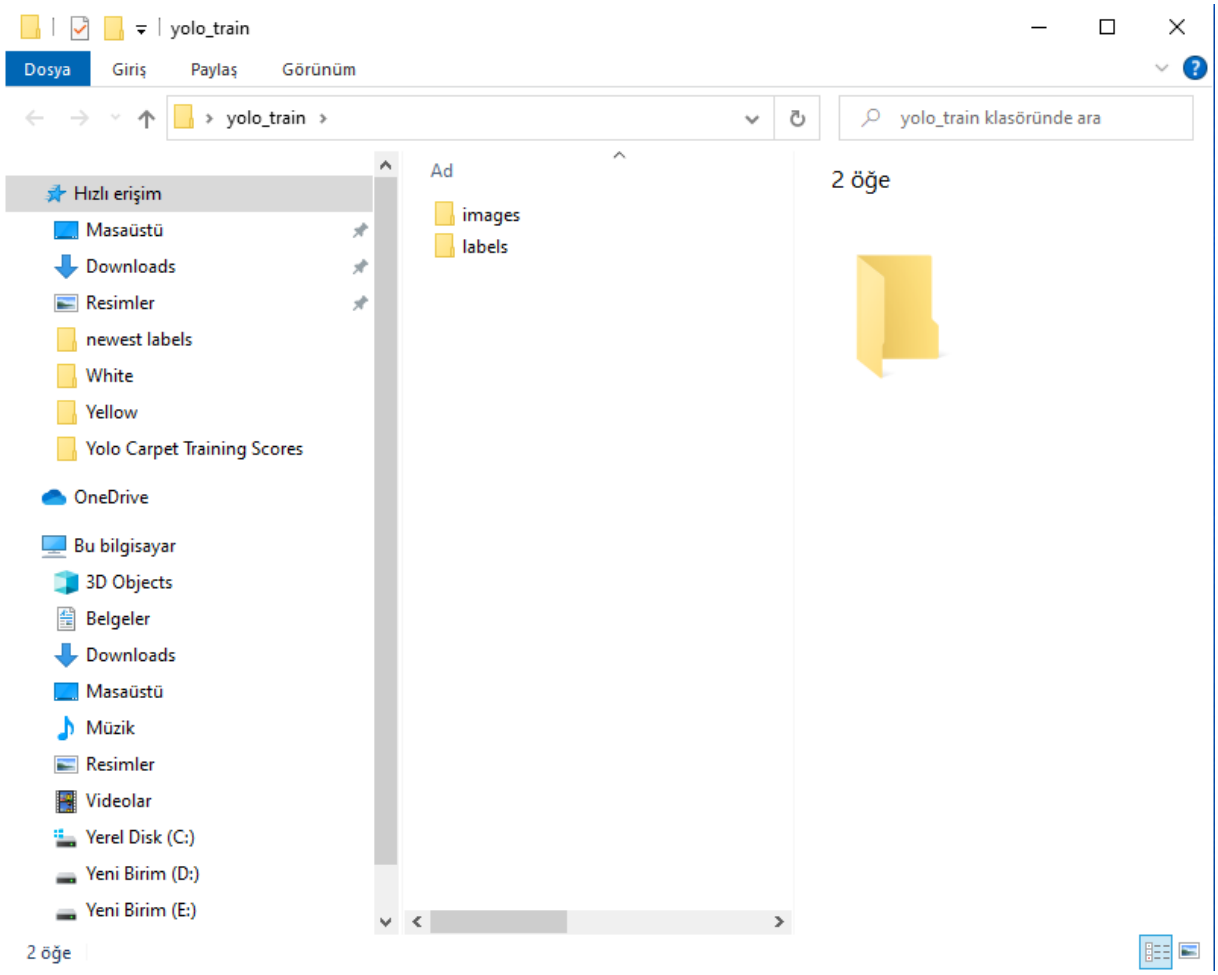


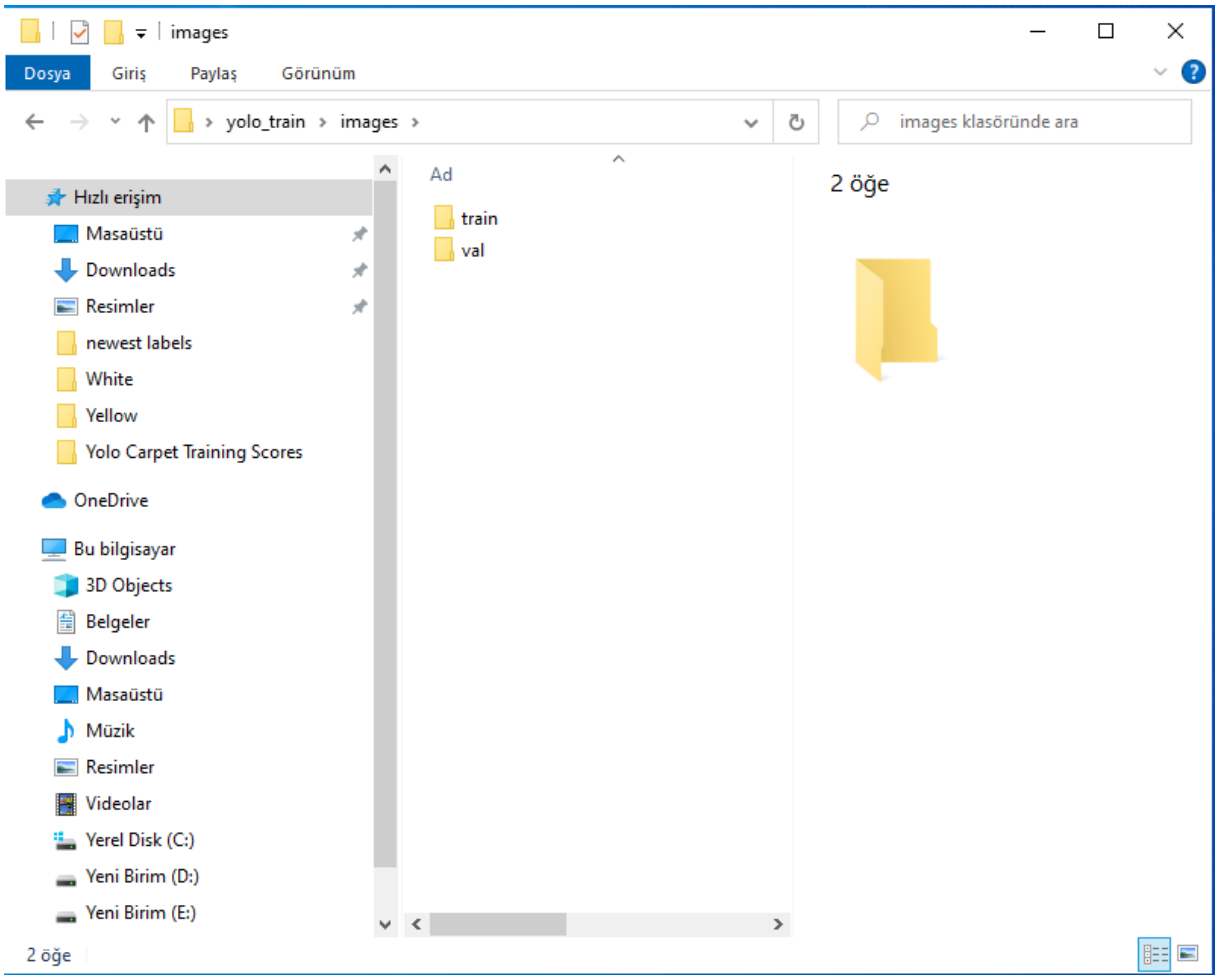
After reaching the sufficient number of images for the model in the data set, the images were labeled via makesenai.com.

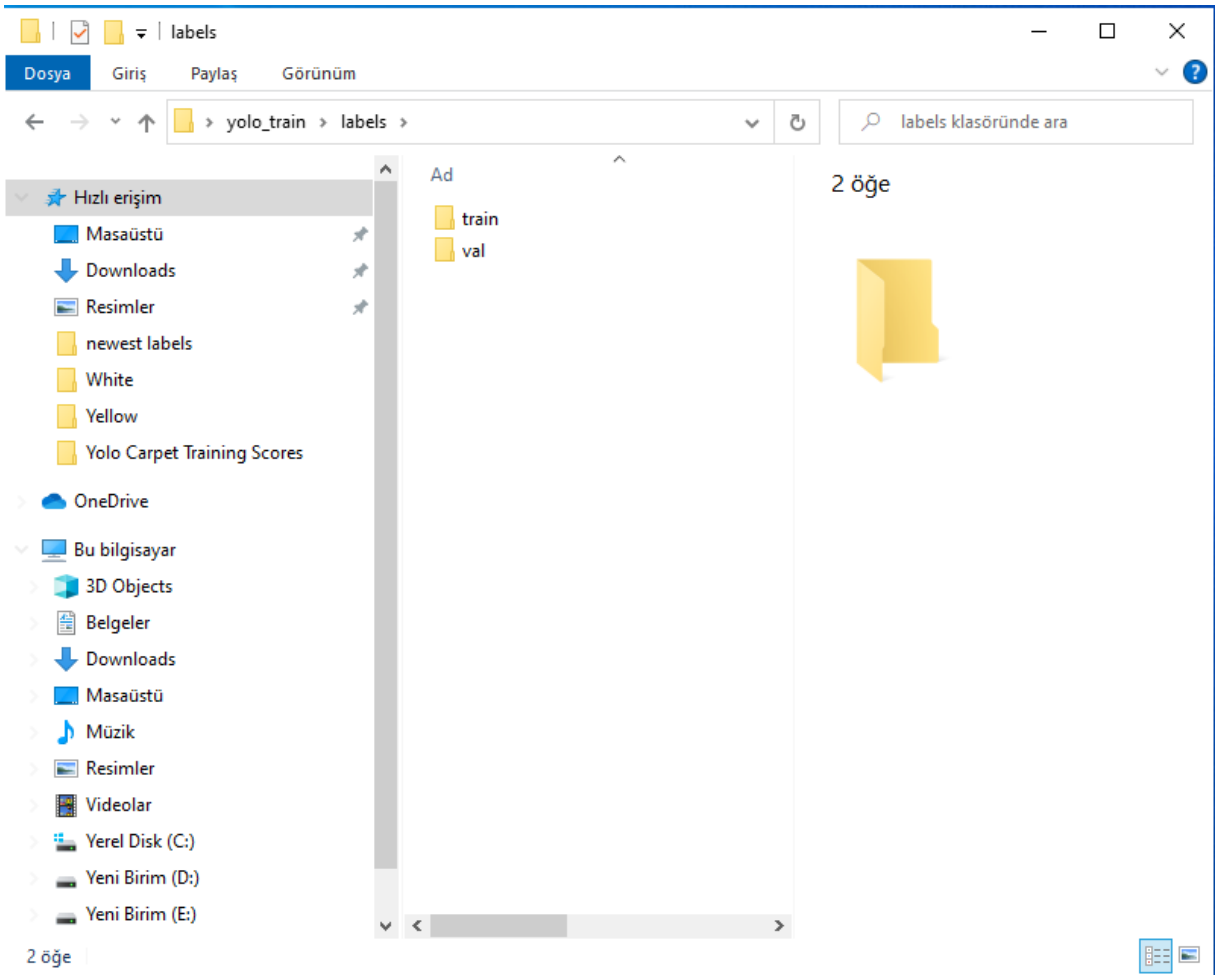




After labeling, labels and images were divided into two as train and validation. Here, 80% of the images and labels were allocated as test and 20% as validation.




images
Dosya
Giriş
Paylaş
Görünüm
yolo_train
images
images klasöründe ara
Hızlı erişim
Masaüstü
Downloads
Resimler
newest labels
White
Yellow
Yolo Carpet Training Scores
OneDrive
Bu bilgisayar
3D Objects
Belgeler
Downloads
Masaüstü
Müzik
Resimler
Videolar
Yerel Disk (C:)
Yeni Birim (D:)
Yeni Birim (E:)
Ad
train
val
2 öğe
2 öğe


labels
Dosya
Giriş
Paylaş
Görünüm
yolo_train
labels
labels klasöründe ara
Hızlı erişim
Masaüstü
Downloads
Resimler
newest labels
White
Yellow
Yolo Carpet Training Scores
OneDrive
Bu bilgisayar
3D Objects
Belgeler
Downloads
Masaüstü
Müzik
Resimler
Videolar
Yerel Disk (C:)
Yeni Birim (D:)
Yeni Birim (E:)
Ad
train
val
2 öğe
2 öğe

Modelling

The YOLOv5 GitHub repositor was used to set up the model. This respository was opened via Google Colab and the necessary downloads were made for the model. Then, the coco128.yaml file, which will make the necessary classification for the model, has been edited as follows.

```yaml
# YOLOv5 🚀 by Ultralytics, GPL-3.0 license
# COCO128 dataset https://www.kaggle.com/ultralytics/coco128 (first 128 images from COCO train2017) by Ultralytics
# Example usage: python train.py --data coco128.yaml
# parent
# ├── yolov5
# └── datasets
#     └── coco128  ← downloads here (7 MB)


# Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3) list: [path/to/imgs1, path/to/imgs2, ..]
path: ../datasets/coco128  # dataset root dir
train: /content/drive/MyDrive/yolo_train/images/train # train images (relative to 'path') 128 images
val: /content/drive/MyDrive/yolo_train/images/val  # val images (relative to 'path') 128 images
test:  # test images (optional)

# Classes
names:
  0: Beige
  1: Black
  2: Blue
  3: Brown
  4: Duck Egg
  5: Green
  6: Grey
  7: Multicolor
  8: Orange
  9: Pink
  10: Purple
  11: Red
  12: White
  13: Yellow


# Download script/URL (optional)
download: https://ultralytics.com/assets/coco128.zip

```

After editing the Coco128.yaml file, the necessary validation process for the model was performed. Along with this, the model recognized which object categories be detected.

```
!python val.py --weights yolov5s.pt --data coco128.yaml --img 640 --half
```

```
!python val.py --weights yolov5s.pt --data coco128.yaml --img 640 --half

val: data=/content/yolov5/data/coco128.yaml, weights=['yolov5s.pt'], batch_size=32, imgsz=640, conf_thres=0.001, iou_thres=0.6, max_det=300, task=val, device=, workers=8, single_cls=False,
YOLOv5 🚀 v7.0-72-g064365d Python-3.8.10 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15110MiB)

Downloading https://github.com/ultralytics/yolov5/releases/download/v7.0/yolov5s.pt to yolov5s.pt...
100% 14.1M/14.1M [00:00<00:00, 326MB/s]

Fusing layers...
YOLOv5s summary: 213 layers, 7225885 parameters, 0 gradients
Traceback (most recent call last):
  File "val.py", line 408, in <module>
    main(opt)
  File "val.py", line 379, in main
    run(**vars(opt))
  File "/usr/local/lib/python3.8/dist-packages/torch/autograd/grad_mode.py", line 27, in decorate_context
    return func(*args, **kwargs)
  File "val.py", line 169, in run
    assert ncm == nc, f'{weights} ({ncm} classes) trained on different --data than what you passed ({nc} ' \
AssertionError: ['yolov5s.pt'] (80 classes) trained on different --data than what you passed (14 classes). Pass correct combination of --weights and --data that are trained together.
```

The model was trained with the following code block.

```
!python train.py --img 640 --batch 16 --epochs 70 --data coco128.yaml --weights yolov5s.pt --cache
```

```
70 epochs completed in 0.087 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.5MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.5MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7047883 parameters, 0 gradients, 15.9 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 2/2 [00:00<00:00,  2.12it/s]
                   all         55         63      0.547      0.456      0.512      0.375
                 Beige         55          9       0.77      0.778      0.787      0.618
                 Black         55          7          0          0     0.0598      0.025
                  Blue         55          5          1      0.958      0.995      0.629
                 Brown         55          3      0.267      0.333      0.352      0.144
              Duck Egg         55          4          1          0     0.0234     0.0211
                 Green         55          5      0.546          1      0.747      0.415
                  Grey         55          5          0          0      0.221       0.16
            Multicolor         55          3          1      0.646      0.736      0.632
                Orange         55          3      0.674      0.667       0.72      0.627
                  Pink         55          5      0.321        0.4      0.271      0.199
                Purple         55          3          1          0      0.665      0.649
                   Red         55          3      0.646          1      0.913      0.723
                 White         55          3          0          0     0.0384     0.0197
                Yellow         55          5       0.43        0.6      0.639      0.389
Results saved to runs/train/exp
```

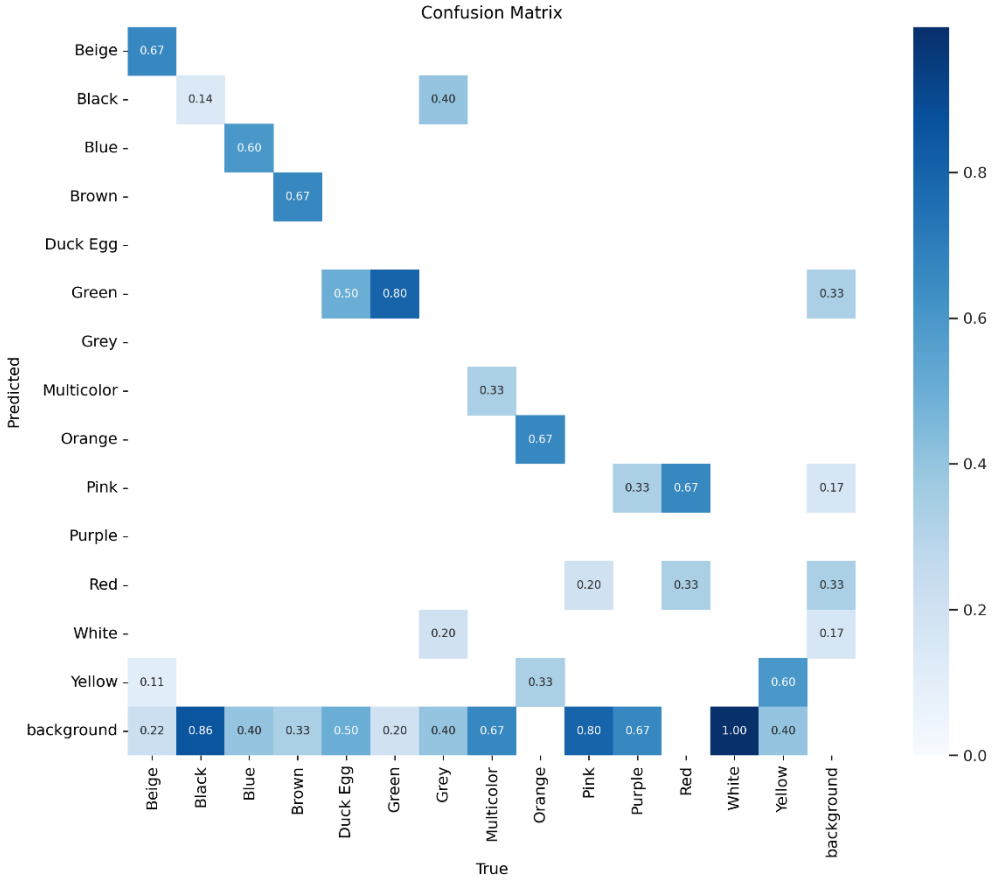
Model Testing

The detect command was used to see how well the model could detect. Below, we have seen that the model predicts beige color for a product in the orange category.

```
!python detect.py --weights runs/train/exp/weights/best.pt --img 640 --conf 0.25 --source /content/Orange3.jpeg

detect: weights=['runs/train/exp/weights/best.pt'], source=/content/Orange3.jpeg, data=data/coco128.yaml, imgsz=[640, 640], conf_thres=0.25,
YOLOv5 🚀 v7.0-72-g064365d Python-3.8.10 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15110MiB)

Fusing layers...
Model summary: 157 layers, 7047883 parameters, 0 gradients, 15.9 GFLOPs
image 1/1 /content/Orange3.jpeg: 640x640 1 Beige, 12.8ms
Speed: 0.7ms pre-process, 12.8ms inference, 1.5ms NMS per image at shape (1, 3, 640, 640)
Results saved to runs/detect/exp2
```

Model Evaluation

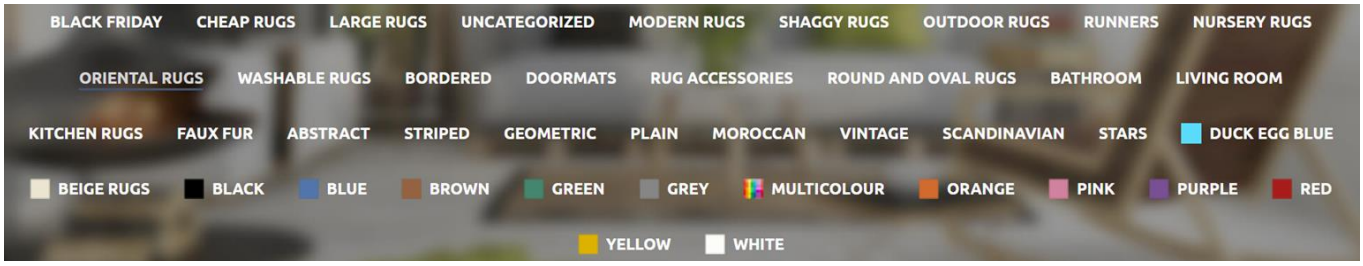
The model achieved an average confidence score of 0.42. This means that our model has difficulty distinguishing between colors. The conclusion to be drawn from here is the distinction made according to the colors of the carpets; does not give good predictive power to the model.



Classification of Rugs based on Styles

In this model, we applied *YOLOv5* to classify rugs based on the styles defined on the company's website. Among them we selected 8 classes:

1-Bordered
2-Doormats
3-Faux Fur
4-Geometric
5-Oriental
6-Round
7-Shaggy
8-Striped



After downloading rugs from each category, they were labeled through *Makesense.ai*. In this model, we used 147 photos in total by separating them into the train (115) and validation (32) sets. Also, we saved a few photos and two YouTube videos to test our model.

We trained our model with 200 epochs and 8 batch sizes.

```
# Validate YOLOv5s on COCO val
!python val.py --weights yolov5s.pt --data coco128.yaml --img 640 --half
```

```
# Train YOLOv5s on COCO128 for 200 epochs
!python train.py --img 640 --batch 8 --epochs 200 --data coco128.yaml --weights yolov5s.pt --cache
```

Although the overall train score is satisfactory, to receive a higher score, the model can be reset by using more samples, especially for Faux Fur and Shaggy classes.

```
200 epochs completed in 0.145 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.5MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.5MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7031701 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 2/2
                   all         32         32      0.922      0.899      0.939      0.868
                 Round         32          4      0.933          1      0.995      0.995
             Geometric         32          4      0.907          1      0.995      0.946
              Bordered         32          4      0.748          1      0.995      0.921
              Faux Fur         32          4          1      0.696      0.995      0.681
              Doormats         32          4      0.974          1      0.995      0.971
              Oriental         32          4      0.929          1      0.995      0.971
               Striped         32          4      0.952          1      0.995      0.913
                Shaggy         32          4      0.934        0.5       0.55       0.55
Results saved to runs/train/exp
```

Test results are shown below:

Faux Fur:

Geometric:

Doormats:

Oriental:

Round:

Shaggy:

Internship Computer Vision Report

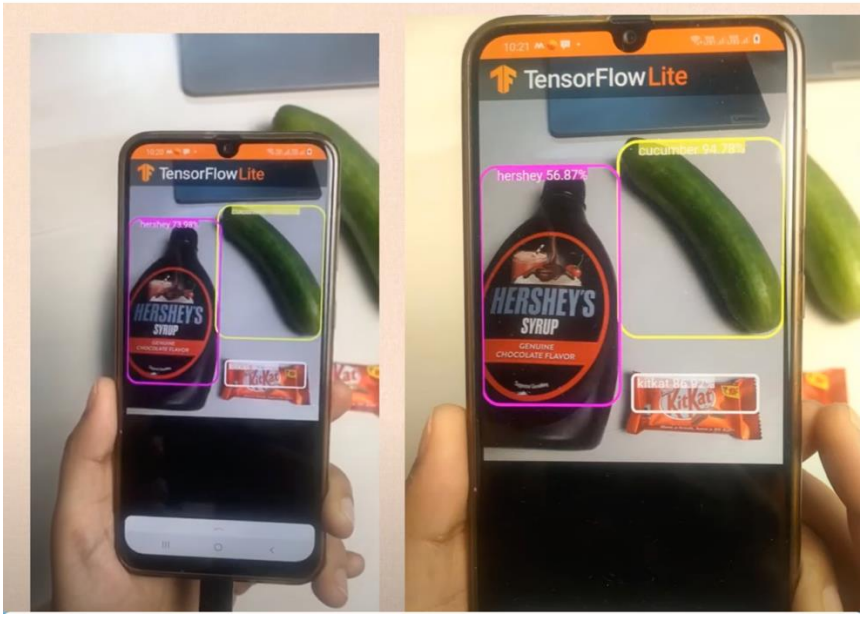
We want to make an app for a company. The purpose of making an app is that if the company wants to physically open a store, they can use our app.

Our app can be used in two ways.

1. When a customer comes to the store, the employee will be able to give the customer detailed information about the carpet. The employee will open the app on the phone and point it at the carpet the customer wants. Our model will recognize the carpet and show the carpet category and detailed features about the carpet. The employee will present these features to the customer. We expect the customer to be satisfied with this detailed and accurate information.
2. As another usage purpose, if the store is too big, customers entering the store can download and use the application from Google Play and AppStore.

   While walking around the store, they will be able to show the carpet they like to the APP and get detailed information about that carpet.
   To make it attractive for customers to download the APP, the carpet like button can be put in the APP, and the customer can press the like button after showing the carpet to the APP. With the APP, extra discounts can be offered to customers who like carpets.
   In this way, the Firm can employ fewer employees in the store.

The general outline of our model will be like this.

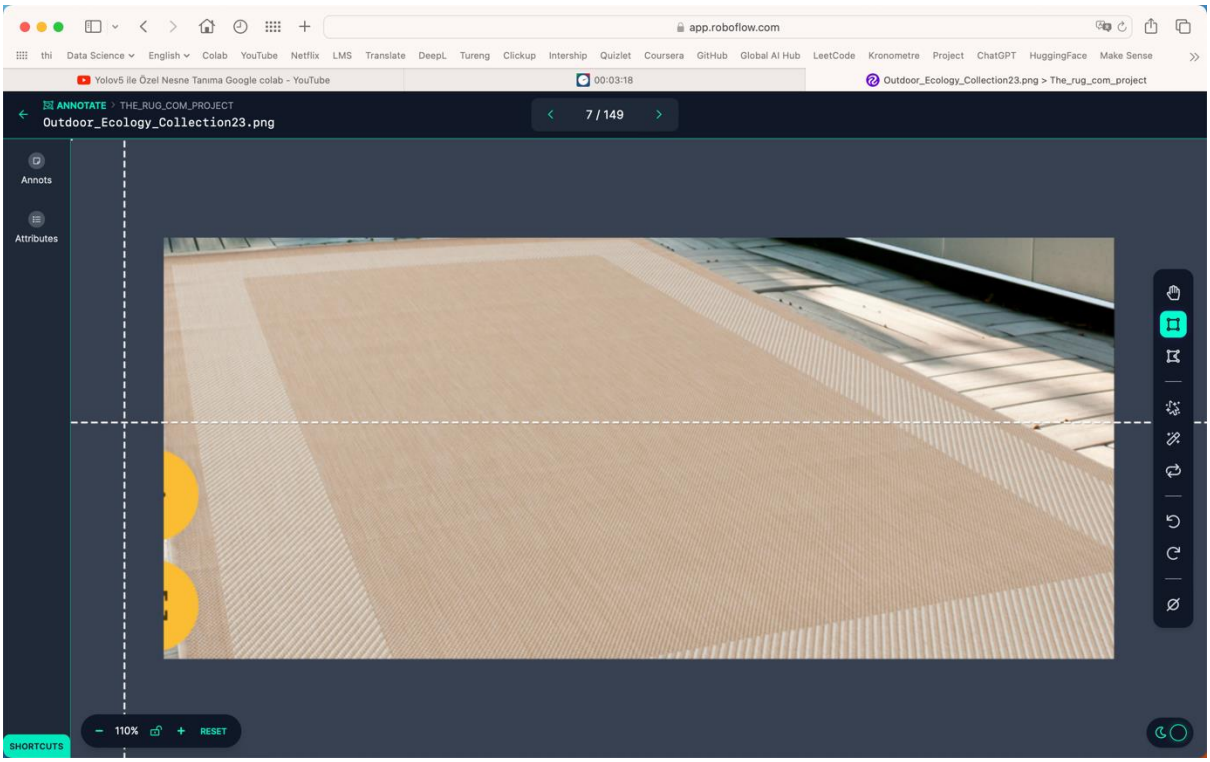


Model-related phases

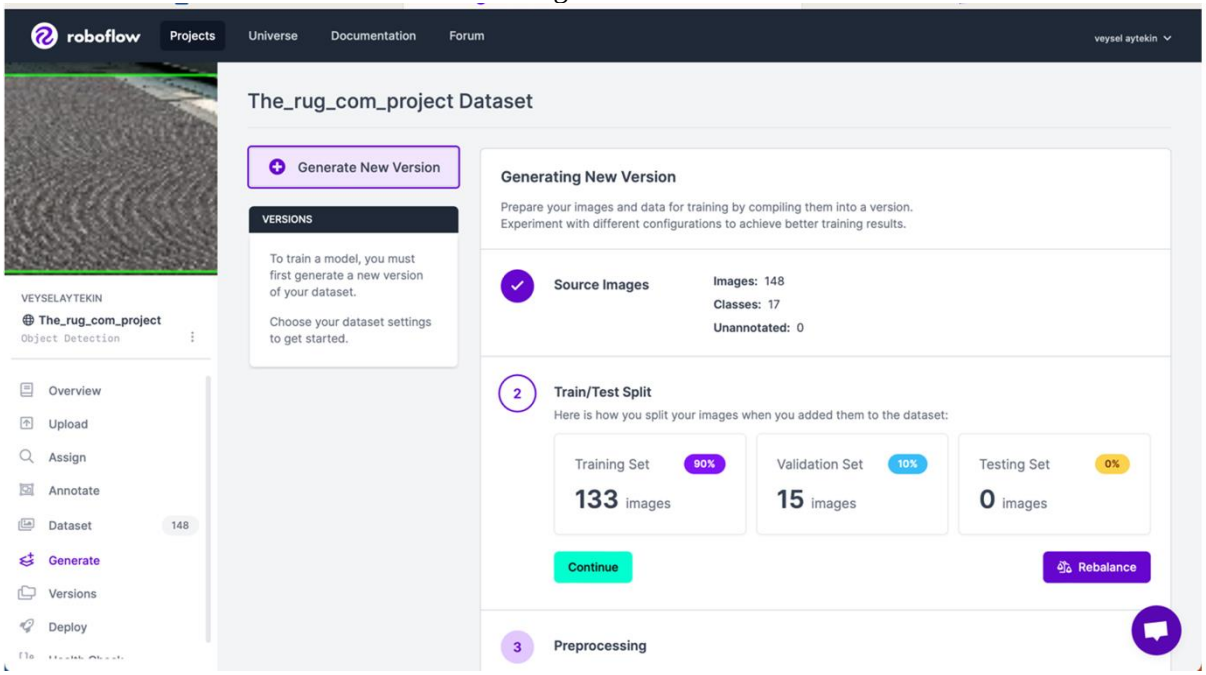
1. I downloaded a total of 149 carpet images from the website and categorized them into 16 different categories

1- Derby Doormat
2- Doormats_Coir_Collection
3- Eco Barrier
4- shaggy_Moraccan_Design
5- shaggy_Geometric_Design
6- shaggy_Diamond_Design
7- Round_Geometric_Collection
8- Round_shaggy_Collection
9- Outdoor_Ecology_Collection
10- Outdoor_Magic_Collection
11- oriental_Atlas_Collection
12- oriental_Caimas_Collection
13- oriental_Carina_Collection
14- oriental_Lara_Collection
15- oriental_Marrakech_Collection
16- oriental_Montana_Collection

I did the labeling through the Roboflow site. When I downloaded the images from the site, I had previously determined which category I was going to use. When I downloaded the images, I gave each image the category names I had determined. This way, when the model makes predictions, I can easily see whether it guessed correctly or not by looking at these names.



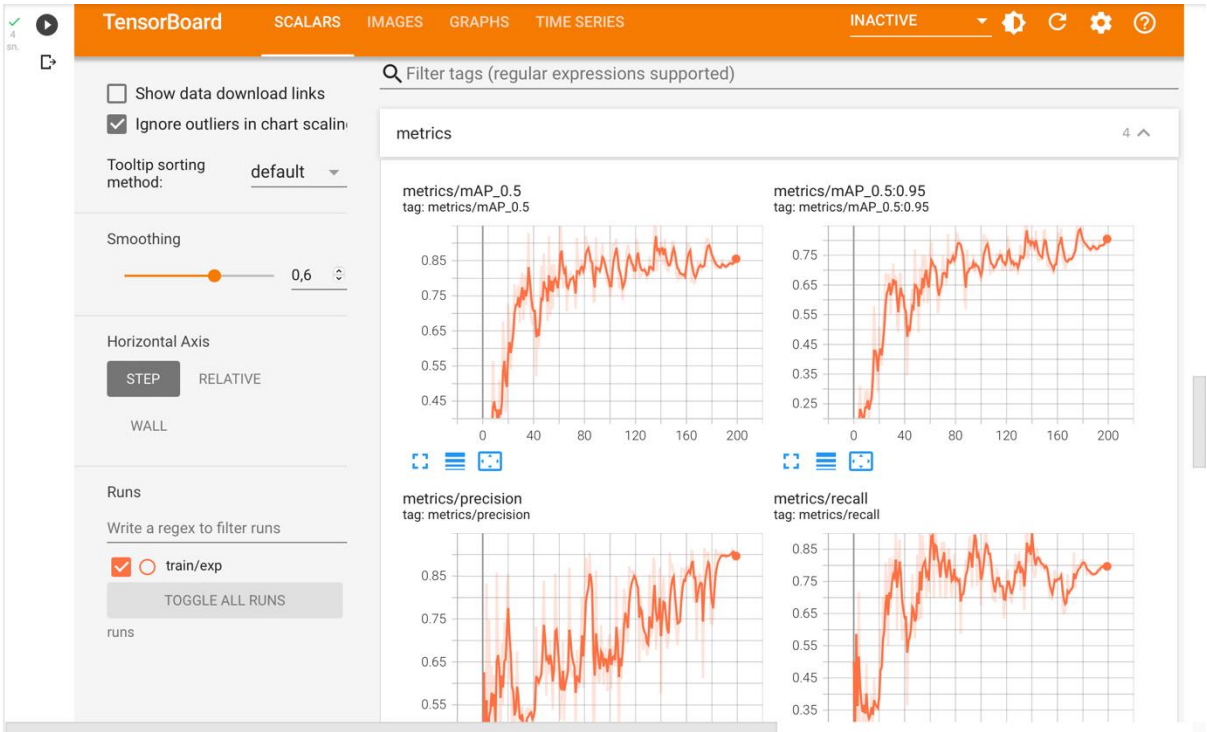
This is how Roboflow allocated the number of images.

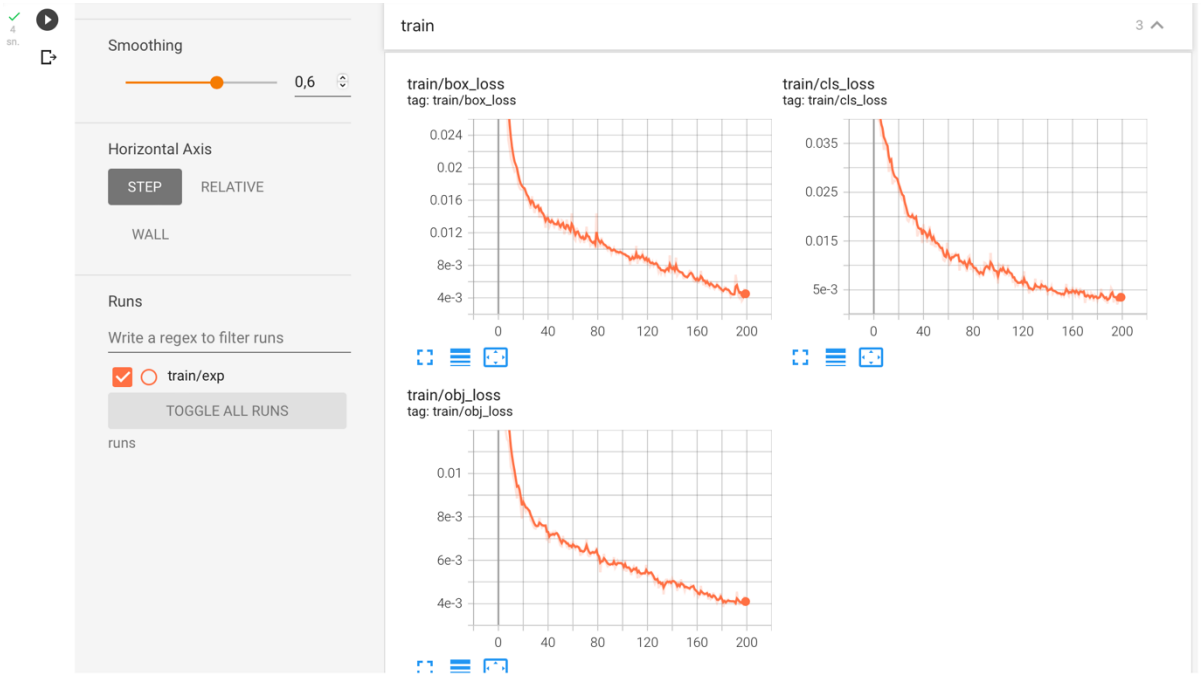


We are training our YOLO5 model.

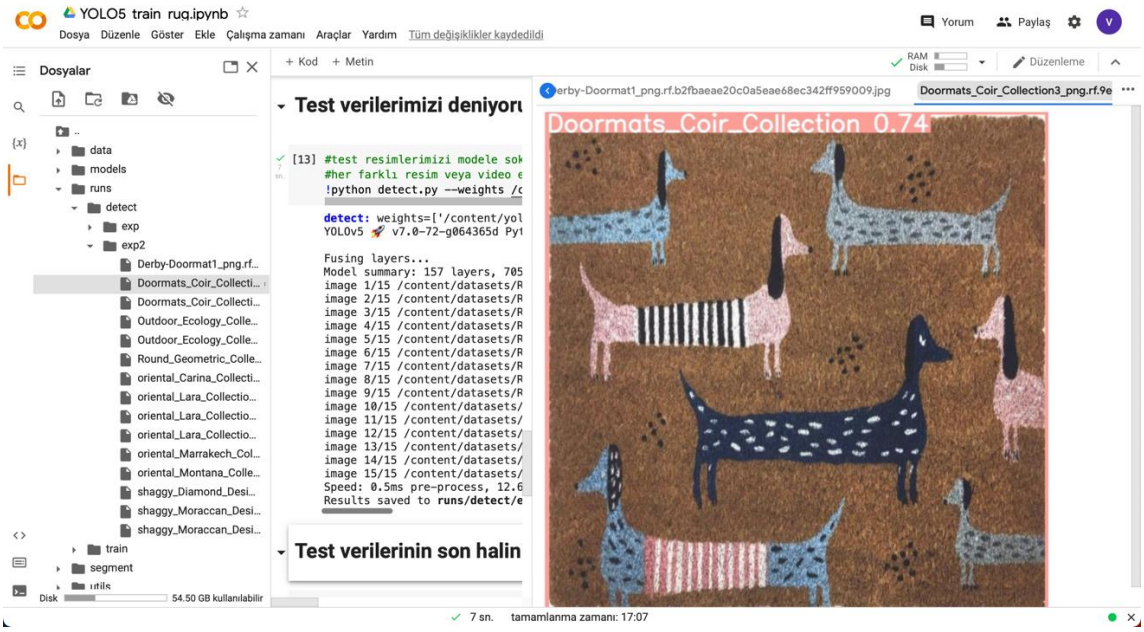
```
!python train.py --img 640  --batch 16 --epochs 200 --data {dataset.location}/data.yaml --weights yolov5s.pt --cache
```

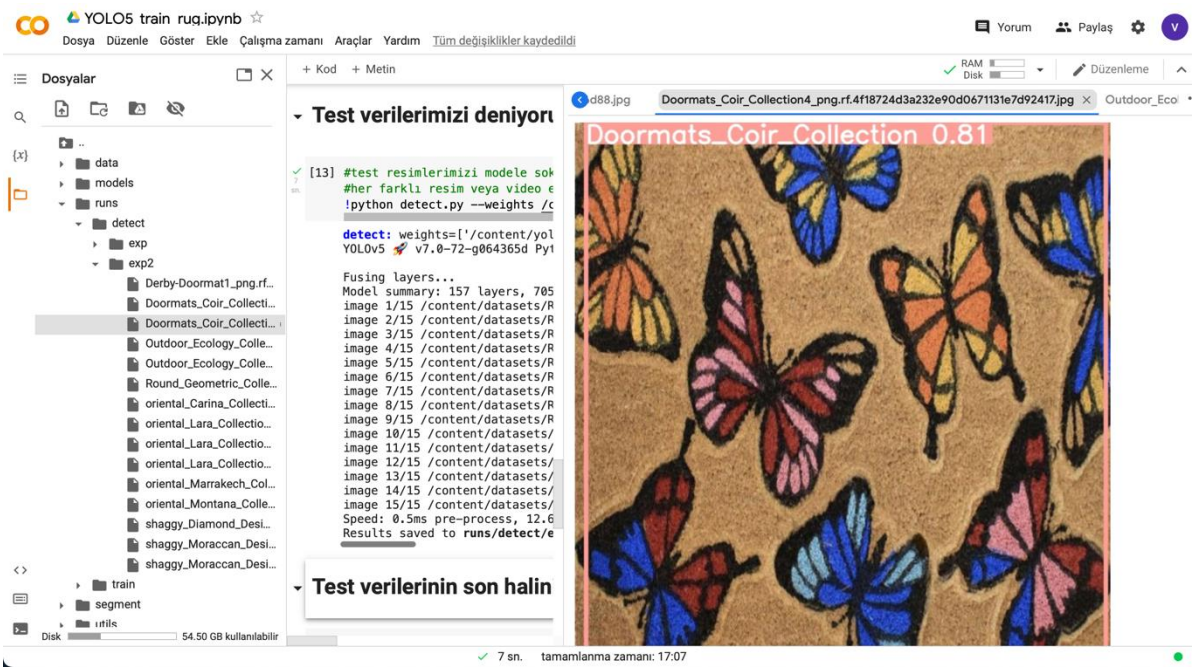
```
      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
    199/199      3.75G   0.004482   0.004023     0.0035         42        640: 100% 25/25 [00:05<00:00,  4.54it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 1/1 [00:00<00:00,  7.12it/
                   all         15         15      0.892      0.794      0.862      0.811

200 epochs completed in 0.333 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.5MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.5MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7053277 parameters, 0 gradients, 15.9 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 1/1 [00:00<00:00,  6.86it/
                   all         15         15      0.881      0.858      0.978      0.892
          Derby Doormat         15          1          1          0      0.995      0.895
Doormats_Coir_Collection         15          2      0.634      0.903      0.828      0.745
Outdoor_Ecology_Collection         15          2      0.875          1      0.995      0.947
Round_Geometric_Collection         15          1      0.859          1      0.995      0.895
oriental_Carina_Collection         15          1      0.894          1      0.995      0.995
oriental_Lara_Collection         15          3          1      0.674      0.995      0.931
oriental_Marrakech_Collection         15          1      0.933          1      0.995      0.796
oriental_Montana_Collection         15          1      0.817          1      0.995      0.895
 shaggy_Diamond_Design         15          1      0.869          1      0.995      0.995
shaggy_Moraccan_Design         15          2       0.93          1      0.995      0.822
Results saved to runs/train/exp
```

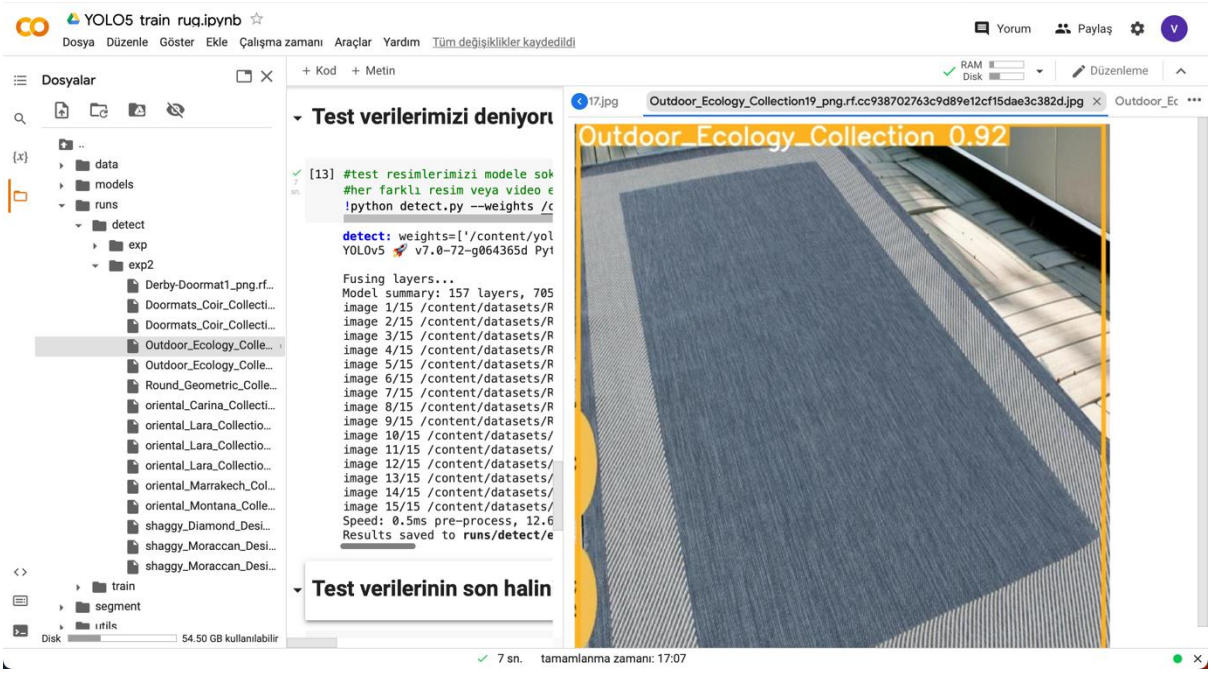


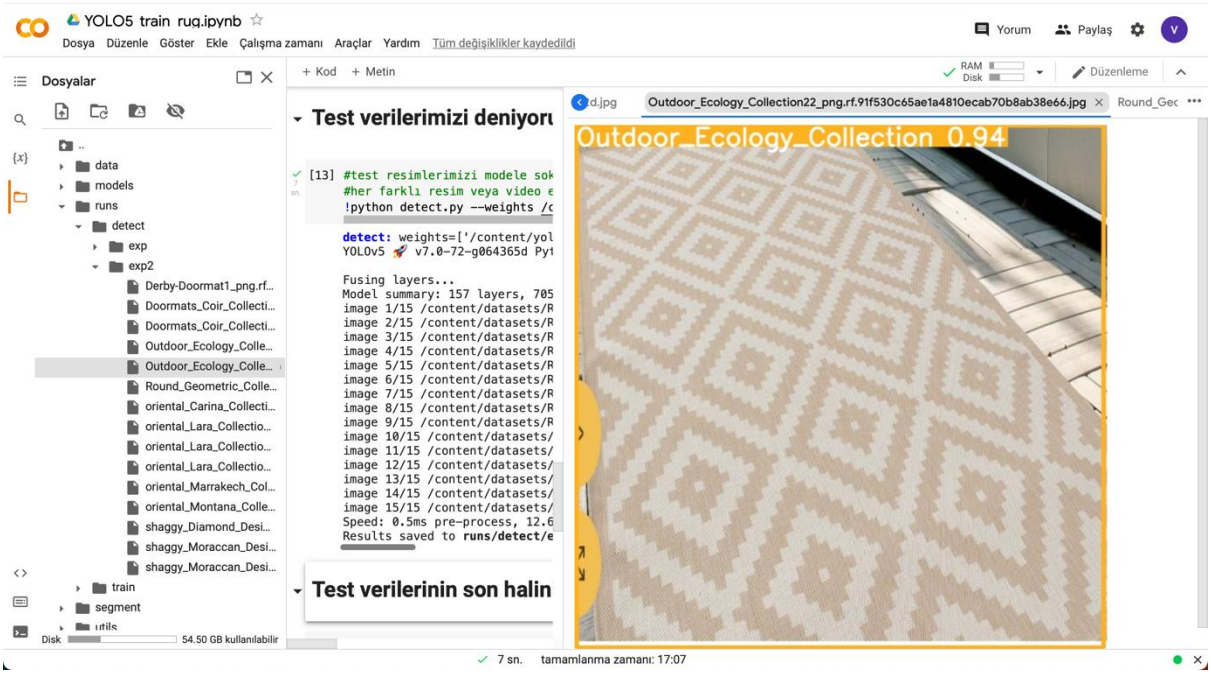

Smoothing
0,6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
train/exp
TOGGLE ALL RUNS
runs
train
3
train/box_loss
tag: train/box_loss
0.024
0.02
0.016
0.012
8e-3
4e-3
0
40
80
120
160
200
train/cls_loss
tag: train/cls_loss
0.035
0.025
0.015
5e-3
0
40
80
120
160
200
train/obj_loss
tag: train/obj_loss
0.01
8e-3
6e-3
4e-3
0
40
80
120
160
200

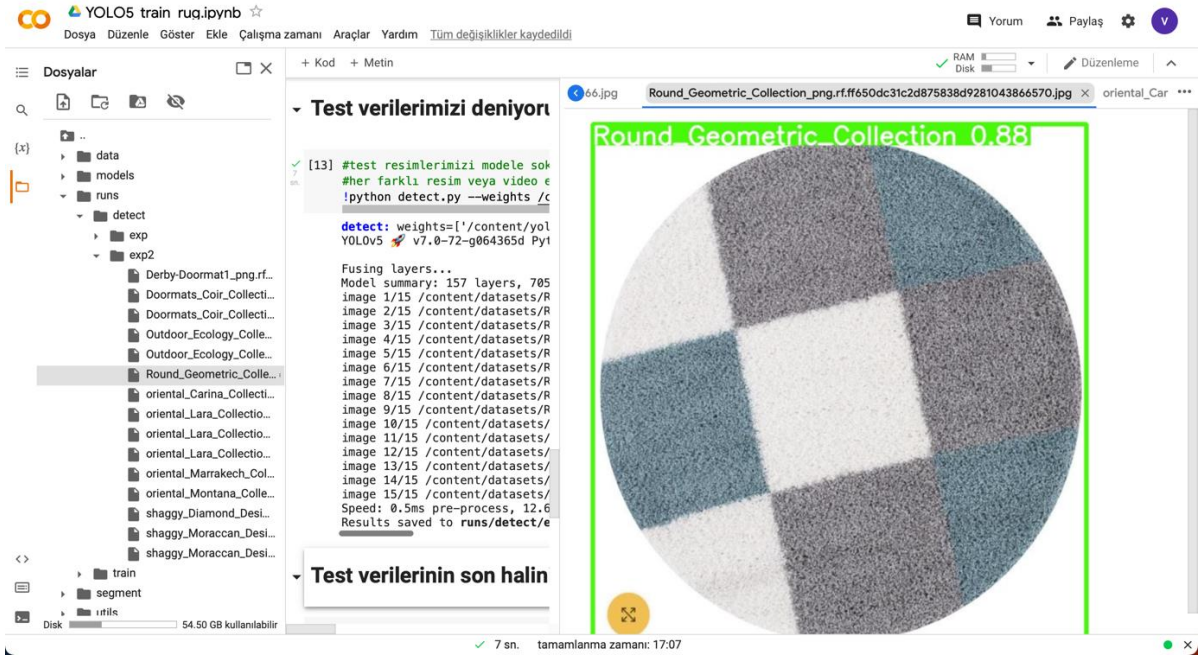
1.17 OUR PREDICTION

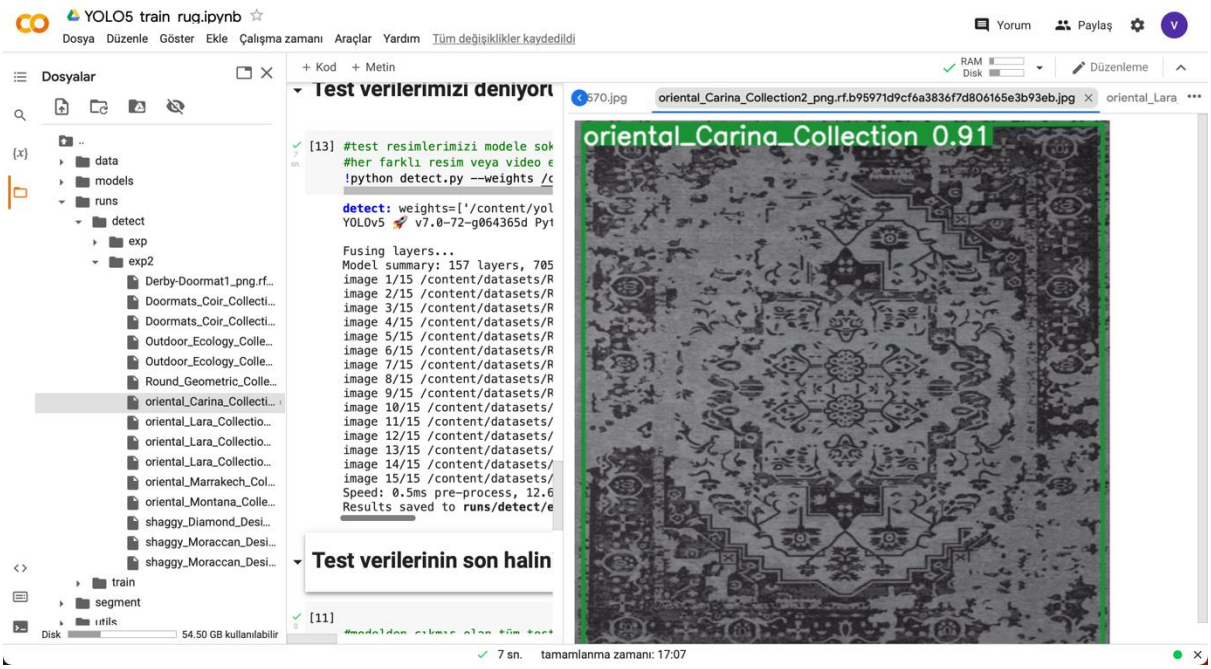


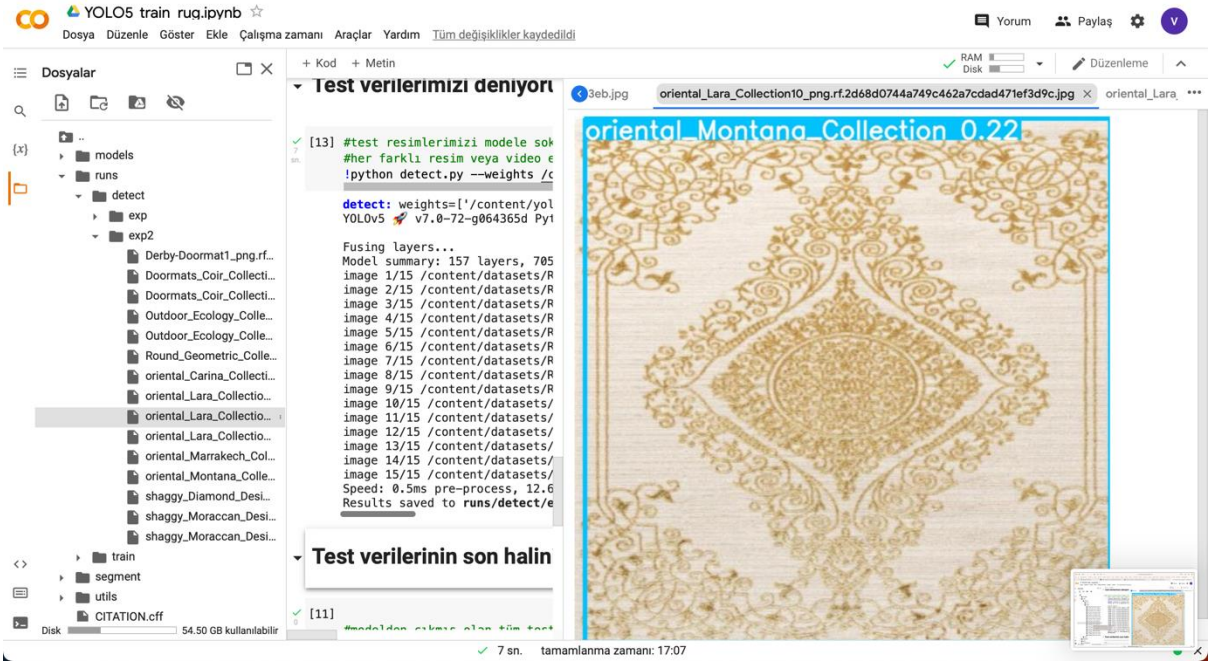

YOLO5 train ruq.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Düzenleme
Test verilerimizi deniyoru
[13] #test resimlerimizi modele sok
#her farklı resim veya video e
!python detect.py --weights /c
detect: weights=['/content/yol
YOLOv5 v7.0-72-g064365d Pyt
Fusing layers...
Model summary: 157 layers, 705
image 1/15 /content/datasets/R
image 2/15 /content/datasets/R
image 3/15 /content/datasets/R
image 4/15 /content/datasets/R
image 5/15 /content/datasets/R
image 6/15 /content/datasets/R
image 7/15 /content/datasets/R
image 8/15 /content/datasets/R
image 9/15 /content/datasets/R
image 10/15 /content/datasets/
image 11/15 /content/datasets/
image 12/15 /content/datasets/
image 13/15 /content/datasets/
image 14/15 /content/datasets/
image 15/15 /content/datasets/
Speed: 0.5ms pre-process, 12.6
Results saved to runs/detect/e
Test verilerinin son halin
d88.jpg
Doormats_Coir_Collection4_png.rf.4f18724d3a232e90d0671131e7d92417.jpg
Outdoor_Eco
Doormats_Coir_Collection 0.81
data
models
runs
detect
exp
exp2
Derby-Doormat1_png.rf...
Doormats_Coir_Collecti...
Doormats_Coir_Collecti...
Outdoor_Ecology_Colle...
Outdoor_Ecology_Colle...
Round_Geometric_Colle...
oriental_Carina_Collecti...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Marrakech_Col...
oriental_Montana_Colle...
shaggy_Diamond_Desi...
shaggy_Moraccan_Desi...
shaggy_Moraccan_Desi...
train
segment
utils
Disk
54.50 GB kullanılabilir
7 sn.
tamamlanma zamanı: 17:07

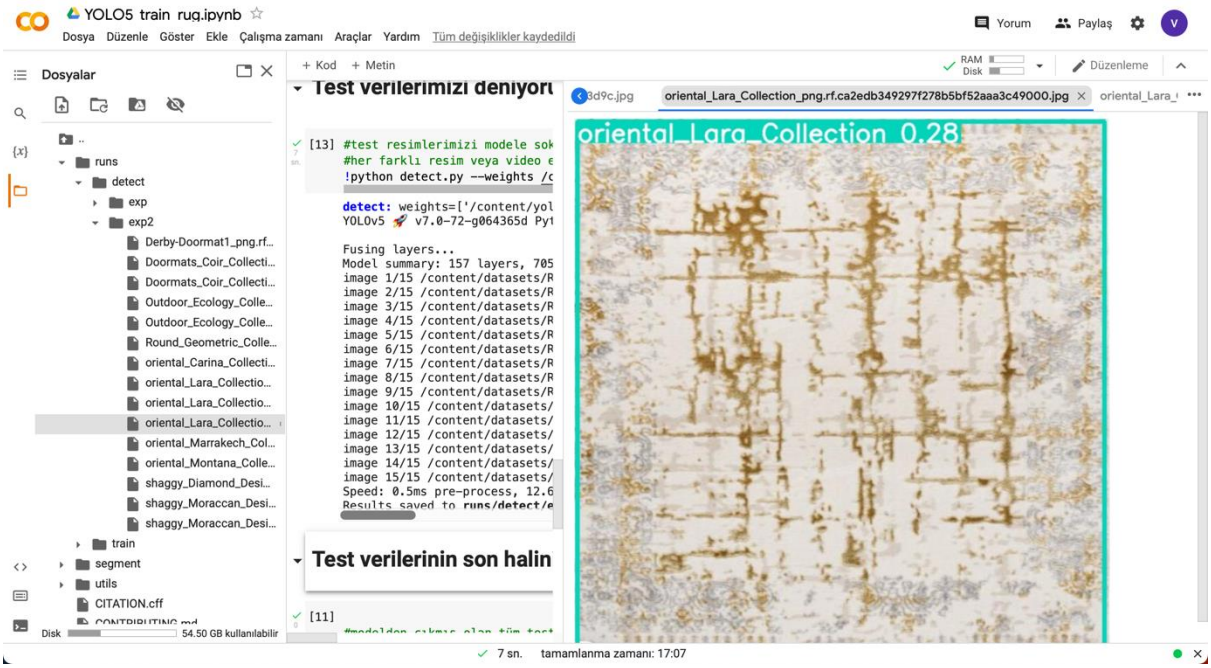

YOLO5 train ruq.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Düzenleme
Test verilerimizi deniyoru
[13] #test resimlerimizi modele sok
#her farklı resim veya video e
!python detect.py --weights /c
detect: weights=['/content/yol
YOLOv5 v7.0-72-g064365d Pyt
Fusing layers...
Model summary: 157 layers, 705
image 1/15 /content/datasets/R
image 2/15 /content/datasets/R
image 3/15 /content/datasets/R
image 4/15 /content/datasets/R
image 5/15 /content/datasets/R
image 6/15 /content/datasets/R
image 7/15 /content/datasets/R
image 8/15 /content/datasets/R
image 9/15 /content/datasets/R
image 10/15 /content/datasets/
image 11/15 /content/datasets/
image 12/15 /content/datasets/
image 13/15 /content/datasets/
image 14/15 /content/datasets/
image 15/15 /content/datasets/
Speed: 0.5ms pre-process, 12.6
Results saved to runs/detect/e
Test verilerinin son halin
17.jpg
Outdoor_Ecology_Collection19_png.rf.cc938702763c9d89e12cf15dae3c382d.jpg
Outdoor_Ec
Outdoor_Ecology_Collection 0.92
data
models
runs
detect
exp
exp2
Derby-Doormat1_png.rf...
Doormats_Coir_Collecti...
Doormats_Coir_Collecti...
Outdoor_Ecology_Colle...
Outdoor_Ecology_Colle...
Round_Geometric_Colle...
oriental_Carina_Collecti...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Marrakech_Col...
oriental_Montana_Colle...
shaggy_Diamond_Desi...
shaggy_Moraccan_Desi...
shaggy_Moraccan_Desi...
train
segment
utils
Disk
54.50 GB kullanılabilir
7 sn.
tamamlanma zamanı: 17:07

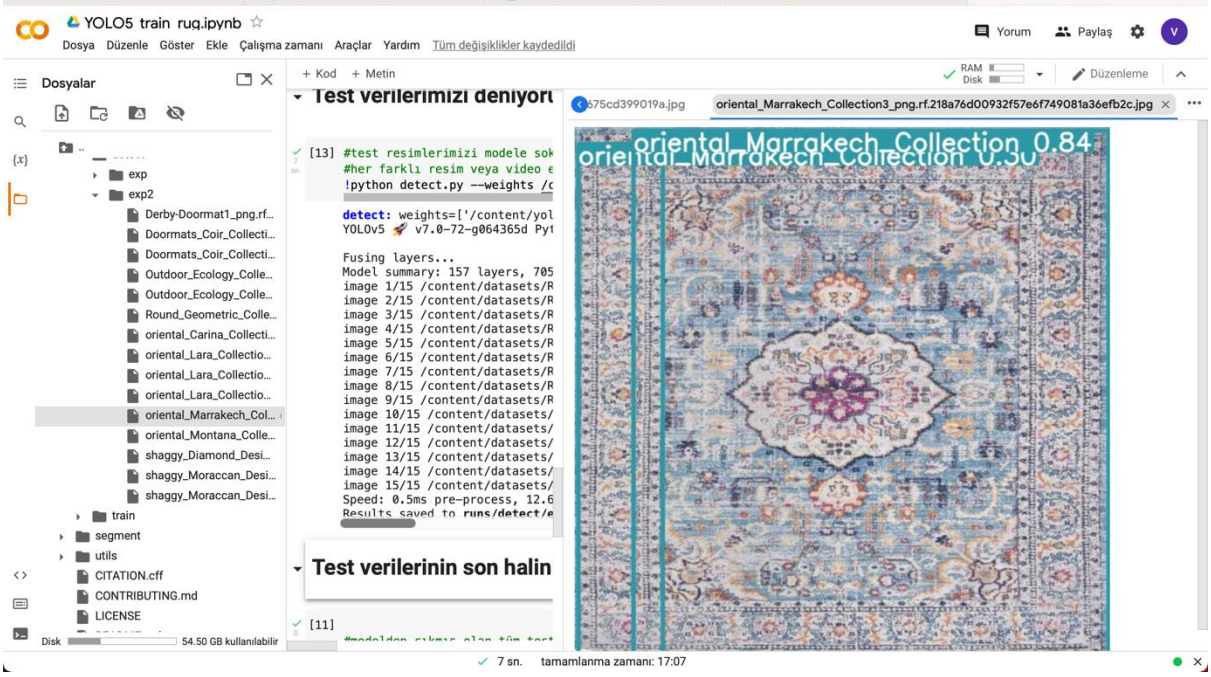

YOLO5 train rug.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Düzenleme
Test verilerimizi deniyoru
[13] #test resimlerimizi modele sok
#her farklı resim veya video e
!python detect.py --weights /c
detect: weights=['/content/yol
YOLOv5 v7.0-72-g064365d Pyt
Fusing layers...
Model summary: 157 layers, 705
image 1/15 /content/datasets/R
image 2/15 /content/datasets/R
image 3/15 /content/datasets/R
image 4/15 /content/datasets/R
image 5/15 /content/datasets/R
image 6/15 /content/datasets/R
image 7/15 /content/datasets/R
image 8/15 /content/datasets/R
image 9/15 /content/datasets/R
image 10/15 /content/datasets/
image 11/15 /content/datasets/
image 12/15 /content/datasets/
image 13/15 /content/datasets/
image 14/15 /content/datasets/
image 15/15 /content/datasets/
Speed: 0.5ms pre-process, 12.6
Results saved to runs/detect/e
Test verilerinin son halin
Outdoor_Ecology_Collection22_png.rf.91f530c65ae1a4810ecab70b8ab38e66.jpg
Round_Gec
Outdoor_Ecology_Collection 0.94
data
models
runs
detect
exp
exp2
Derby-Doormat1_png.rf...
Doormats_Coir_Collecti...
Doormats_Coir_Collecti...
Outdoor_Ecology_Colle...
Outdoor_Ecology_Colle...
Round_Geometric_Colle...
oriental_Carina_Collecti...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Marrakech_Col...
oriental_Montana_Colle...
shaggy_Diamond_Desi...
shaggy_Moraccan_Desi...
shaggy_Moraccan_Desi...
train
segment
utils
Disk
54.50 GB kullanılabilir
7 sn. tamamlanma zamanı: 17:07

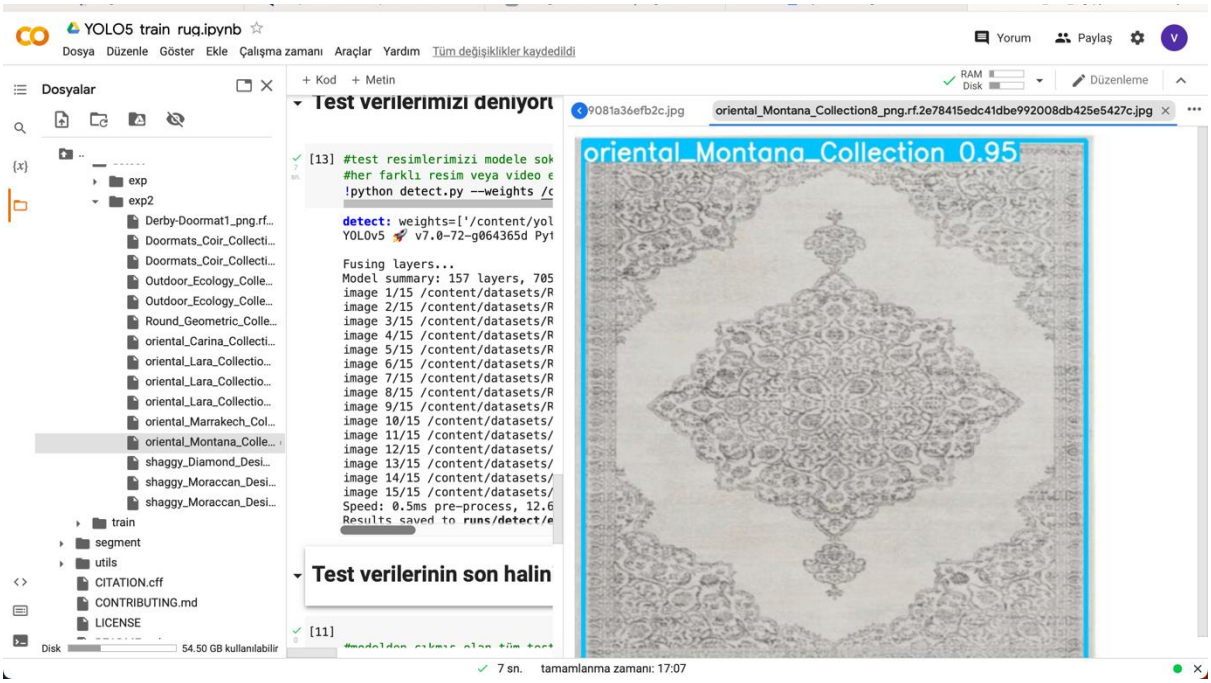

YOLO5 train rug.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Düzenleme
Test verilerimizi deniyoru
[13] #test resimlerimizi modele sok
#her farklı resim veya video e
!python detect.py --weights /c
detect: weights=['/content/yol
YOLOv5 v7.0-72-g064365d Pyt
Fusing layers...
Model summary: 157 layers, 705
image 1/15 /content/datasets/R
image 2/15 /content/datasets/R
image 3/15 /content/datasets/R
image 4/15 /content/datasets/R
image 5/15 /content/datasets/R
image 6/15 /content/datasets/R
image 7/15 /content/datasets/R
image 8/15 /content/datasets/R
image 9/15 /content/datasets/R
image 10/15 /content/datasets/
image 11/15 /content/datasets/
image 12/15 /content/datasets/
image 13/15 /content/datasets/
image 14/15 /content/datasets/
image 15/15 /content/datasets/
Speed: 0.5ms pre-process, 12.6
Results saved to runs/detect/e
Test verilerinin son halin
66.jpg
Round_Geometric_Collection_png.rf.ff650dc31c2d875838d9281043866570.jpg
oriental_Car
Round_Geometric_Collection 0.88
data
models
runs
detect
exp
exp2
Derby-Doormat1_png.rf...
Doormats_Coir_Collecti...
Doormats_Coir_Collecti...
Outdoor_Ecology_Colle...
Outdoor_Ecology_Colle...
Round_Geometric_Colle...
oriental_Carina_Collecti...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Marrakech_Col...
oriental_Montana_Colle...
shaggy_Diamond_Desi...
shaggy_Moraccan_Desi...
shaggy_Moraccan_Desi...
train
segment
utils
Disk
54.50 GB kullanılabilir
7 sn. tamamlanma zamanı: 17:07

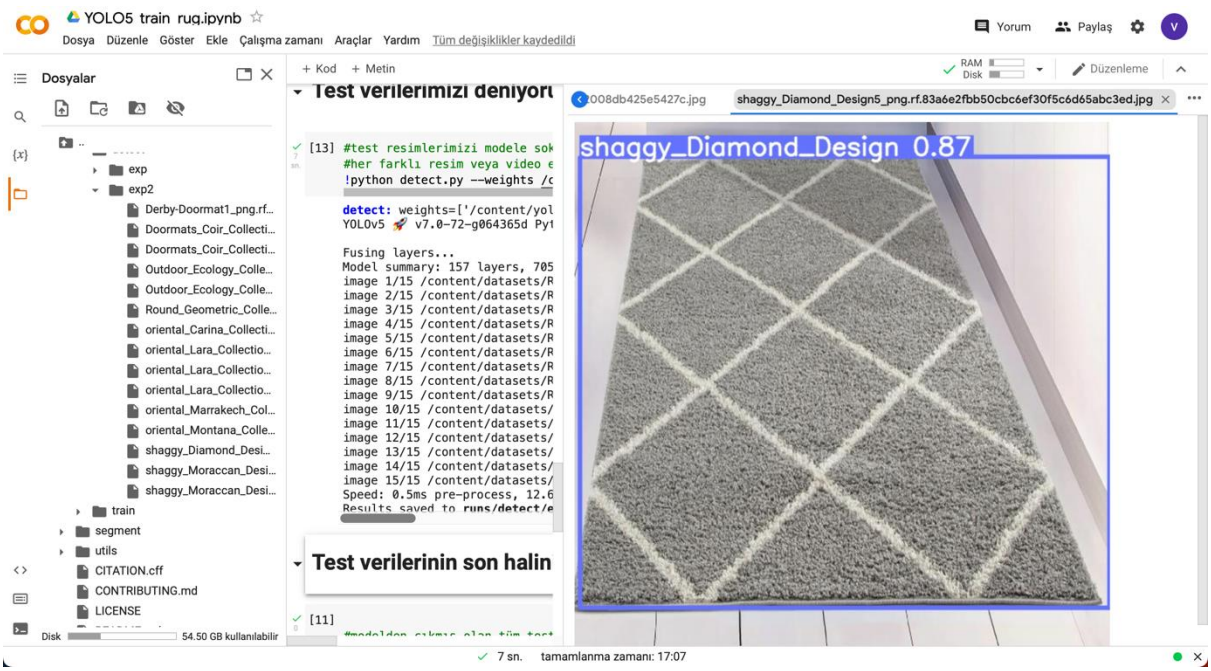

YOLO5 train ruq.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Düzenleme
Test verilerimizi deniyoruz
[13] #test resimlerimizi modele sok
#her farklı resim veya video e
!python detect.py --weights /c
detect: weights=['/content/yol
YOLOv5 v7.0-72-g064365d Pyt
Fusing layers...
Model summary: 157 layers, 705
image 1/15 /content/datasets/R
image 2/15 /content/datasets/R
image 3/15 /content/datasets/R
image 4/15 /content/datasets/R
image 5/15 /content/datasets/R
image 6/15 /content/datasets/R
image 7/15 /content/datasets/R
image 8/15 /content/datasets/R
image 9/15 /content/datasets/R
image 10/15 /content/datasets/
image 11/15 /content/datasets/
image 12/15 /content/datasets/
image 13/15 /content/datasets/
image 14/15 /content/datasets/
image 15/15 /content/datasets/
Speed: 0.5ms pre-process, 12.6
Results saved to runs/detect/e
Test verilerinin son halin
[11]
570.jpg
oriental_Carina_Collection2_png.rf.b95971d9cf6a3836f7d806165e3b93eb.jpg
oriental_Lara
oriental_Carina_Collection 0.91
data
models
runs
detect
exp
exp2
Derby-Doormat1_png.rf...
Doormats_Coir_Collecti...
Doormats_Coir_Collecti...
Outdoor_Ecology_Colle...
Outdoor_Ecology_Colle...
Round_Geometric_Colle...
oriental_Carina_Collecti...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Marrakech_Col...
oriental_Montana_Colle...
shaggy_Diamond_Desi...
shaggy_Moraccan_Desi...
shaggy_Moraccan_Desi...
train
segment
Disk
54.50 GB kullanılabilir
7 sn.
tamamlanma zamanı: 17:07

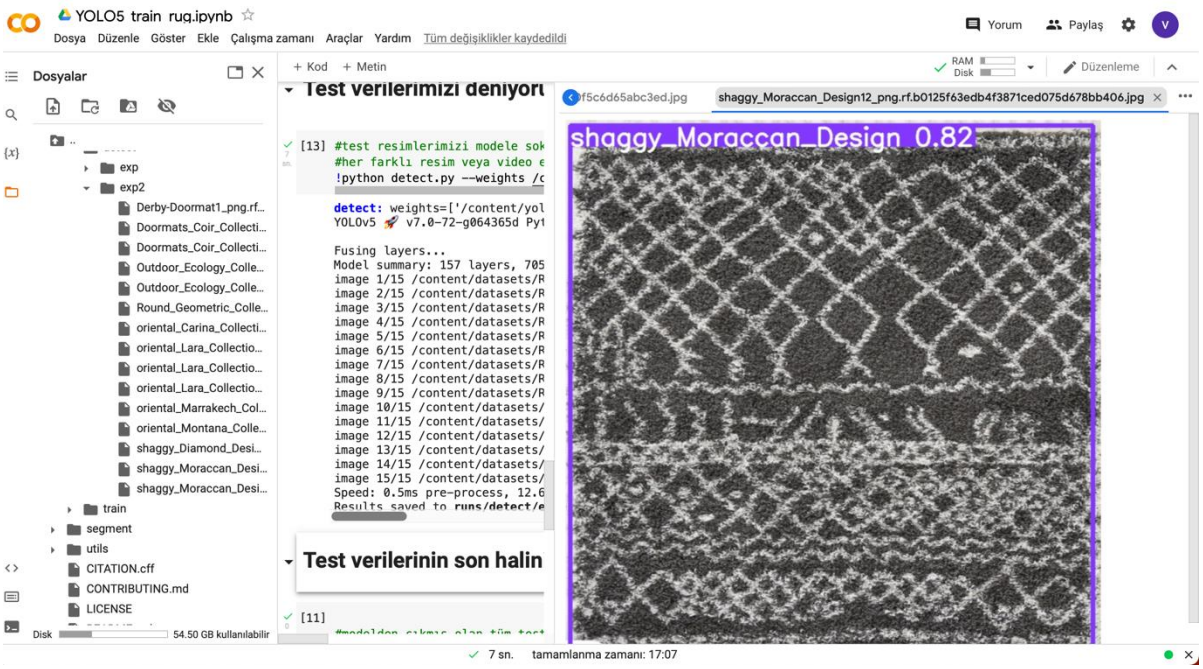

YOLO5 train ruq.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Düzenleme
Test verilerimizi deniyoruz
[13] #test resimlerimizi modele sok
#her farklı resim veya video e
!python detect.py --weights /c
detect: weights=['/content/yol
YOLOv5 v7.0-72-g064365d Pyt
Fusing layers...
Model summary: 157 layers, 705
image 1/15 /content/datasets/R
image 2/15 /content/datasets/R
image 3/15 /content/datasets/R
image 4/15 /content/datasets/R
image 5/15 /content/datasets/R
image 6/15 /content/datasets/R
image 7/15 /content/datasets/R
image 8/15 /content/datasets/R
image 9/15 /content/datasets/R
image 10/15 /content/datasets/
image 11/15 /content/datasets/
image 12/15 /content/datasets/
image 13/15 /content/datasets/
image 14/15 /content/datasets/
image 15/15 /content/datasets/
Speed: 0.5ms pre-process, 12.6
Results saved to runs/detect/e
Test verilerinin son halin
[11]
3eb.jpg
oriental_Lara_Collection10_png.rf.2d68d0744a749c462a7cdad471ef3d9c.jpg
oriental_Lara
oriental_Montana_Collection 0.22
models
runs
detect
exp
exp2
Derby-Doormat1_png.rf...
Doormats_Coir_Collecti...
Doormats_Coir_Collecti...
Outdoor_Ecology_Colle...
Outdoor_Ecology_Colle...
Round_Geometric_Colle...
oriental_Carina_Collecti...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Marrakech_Col...
oriental_Montana_Colle...
shaggy_Diamond_Desi...
shaggy_Moraccan_Desi...
shaggy_Moraccan_Desi...
train
segment
utils
CITATION.cff
Disk
54.50 GB kullanılabilir
7 sn.
tamamlanma zamanı: 17:07

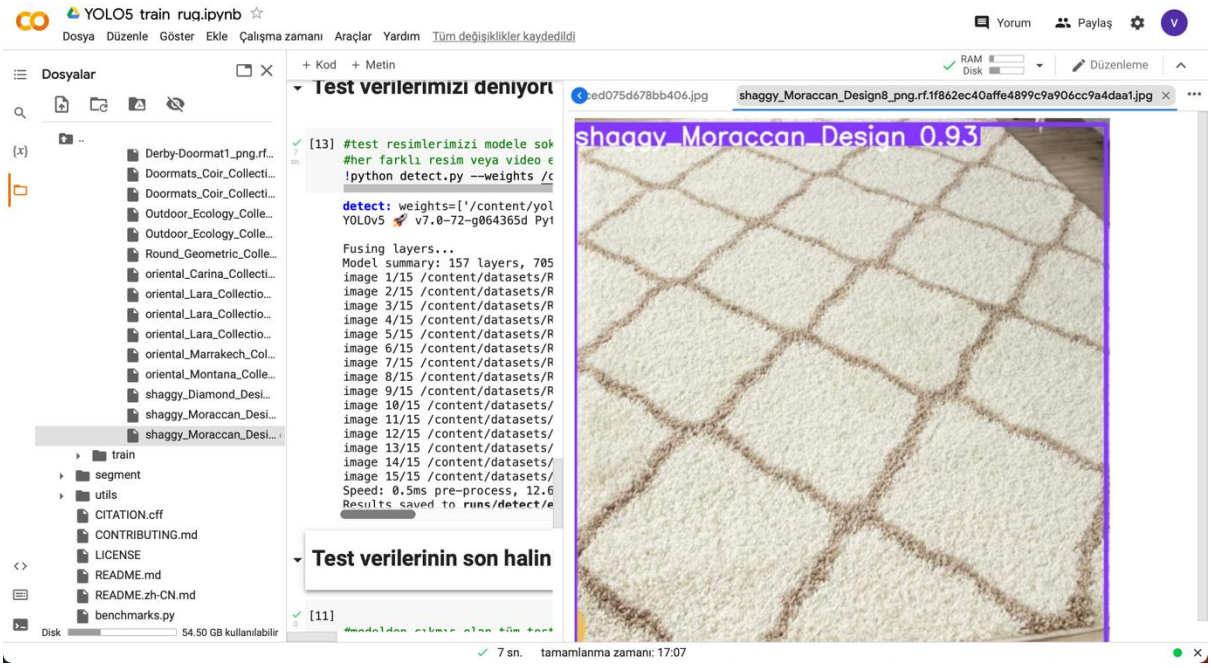

YOLO5 train ruq.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Düzenleme
Test verilerimizi deniyor
oriental_Lara_Collection_png.rf.ca2edb349297f278b5bf52aaa3c49000.jpg
oriental_Lara_Collection_0.28
[13] #test resimlerimizi modele sok
#her farklı resim veya video e
!python detect.py --weights /c
detect: weights=['/content/yol
YOLOv5 v7.0-72-g064365d Pyt
Fusing layers...
Model summary: 157 layers, 705
image 1/15 /content/datasets/R
image 2/15 /content/datasets/R
image 3/15 /content/datasets/R
image 4/15 /content/datasets/R
image 5/15 /content/datasets/R
image 6/15 /content/datasets/R
image 7/15 /content/datasets/R
image 8/15 /content/datasets/R
image 9/15 /content/datasets/R
image 10/15 /content/datasets/
image 11/15 /content/datasets/
image 12/15 /content/datasets/
image 13/15 /content/datasets/
image 14/15 /content/datasets/
image 15/15 /content/datasets/
Speed: 0.5ms pre-process, 12.6
Results saved to runs/detect/e
Test verilerinin son halin
[11]
runs
detect
exp
exp2
Derby-Doormat1_png.rf...
Doormats_Coir_Collecti...
Doormats_Coir_Collecti...
Outdoor_Ecology_Colle...
Outdoor_Ecology_Colle...
Round_Geometric_Colle...
oriental_Carina_Collecti...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Marrakech_Col...
oriental_Montana_Colle...
shaggy_Diamond_Desi...
shaggy_Moraccan_Desi...
shaggy_Moraccan_Desi...
train
segment
utils
CITATION.cff
Disk
54.50 GB kullanılabilir
7 sn. tamamlanma zamanı: 17:07


YOLO5 train ruq.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Düzenleme
Test verilerimizi deniyor
oriental_Marrakech_Collection3_png.rf.218a76d00932f57e6f749081a36efb2c.jpg
oriental_Marrakech_Collection_0.84
[13] #test resimlerimizi modele sok
#her farklı resim veya video e
!python detect.py --weights /c
detect: weights=['/content/yol
YOLOv5 v7.0-72-g064365d Pyt
Fusing layers...
Model summary: 157 layers, 705
image 1/15 /content/datasets/R
image 2/15 /content/datasets/R
image 3/15 /content/datasets/R
image 4/15 /content/datasets/R
image 5/15 /content/datasets/R
image 6/15 /content/datasets/R
image 7/15 /content/datasets/R
image 8/15 /content/datasets/R
image 9/15 /content/datasets/R
image 10/15 /content/datasets/
image 11/15 /content/datasets/
image 12/15 /content/datasets/
image 13/15 /content/datasets/
image 14/15 /content/datasets/
image 15/15 /content/datasets/
Speed: 0.5ms pre-process, 12.6
Results saved to runs/detect/e
Test verilerinin son halin
[11]
exp
exp2
Derby-Doormat1_png.rf...
Doormats_Coir_Collecti...
Doormats_Coir_Collecti...
Outdoor_Ecology_Colle...
Outdoor_Ecology_Colle...
Round_Geometric_Colle...
oriental_Carina_Collecti...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Marrakech_Col...
oriental_Montana_Colle...
shaggy_Diamond_Desi...
shaggy_Moraccan_Desi...
shaggy_Moraccan_Desi...
train
segment
utils
CITATION.cff
CONTRIBUTING.md
LICENSE
Disk
54.50 GB kullanılabilir
7 sn. tamamlanma zamanı: 17:07


YOLO5 train ruq.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Düzenleme
Test verilerimizi deniyorı
[13] #test resimlerimizi modele sok
#her farklı resim veya video e
!python detect.py --weights /c
detect: weights=['/content/yol
YOLOv5 v7.0-72-g064365d Pyt
Fusing layers...
Model summary: 157 layers, 705
image 1/15 /content/datasets/R
image 2/15 /content/datasets/R
image 3/15 /content/datasets/R
image 4/15 /content/datasets/R
image 5/15 /content/datasets/R
image 6/15 /content/datasets/R
image 7/15 /content/datasets/R
image 8/15 /content/datasets/R
image 9/15 /content/datasets/R
image 10/15 /content/datasets/
image 11/15 /content/datasets/
image 12/15 /content/datasets/
image 13/15 /content/datasets/
image 14/15 /content/datasets/
image 15/15 /content/datasets/
Speed: 0.5ms pre-process, 12.6
Results saved to runs/detect/e
Test verilerinin son halin
[11]
exp
exp2
Derby-Doormat1_png.rf...
Doormats_Coir_Collecti...
Doormats_Coir_Collecti...
Outdoor_Ecology_Colle...
Outdoor_Ecology_Colle...
Round_Geometric_Colle...
oriental_Carina_Collecti...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Marrakech_Col...
oriental_Montana_Colle...
shaggy_Diamond_Desi...
shaggy_Moraccan_Desi...
shaggy_Moraccan_Desi...
train
segment
utils
CITATION.cff
CONTRIBUTING.md
LICENSE
Disk
54.50 GB kullanılabilir
9081a36efb2c.jpg
oriental_Montana_Collection8_png.rf.2e78415edc41dbe992008db425e5427c.jpg
oriental_Montana_Collection 0.95
7 sn. tamamlanma zamanı: 17:07


YOLO5 train ruq.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Düzenleme
Test verilerimizi deniyorı
[13] #test resimlerimizi modele sok
#her farklı resim veya video e
!python detect.py --weights /c
detect: weights=['/content/yol
YOLOv5 v7.0-72-g064365d Pyt
Fusing layers...
Model summary: 157 layers, 705
image 1/15 /content/datasets/R
image 2/15 /content/datasets/R
image 3/15 /content/datasets/R
image 4/15 /content/datasets/R
image 5/15 /content/datasets/R
image 6/15 /content/datasets/R
image 7/15 /content/datasets/R
image 8/15 /content/datasets/R
image 9/15 /content/datasets/R
image 10/15 /content/datasets/
image 11/15 /content/datasets/
image 12/15 /content/datasets/
image 13/15 /content/datasets/
image 14/15 /content/datasets/
image 15/15 /content/datasets/
Speed: 0.5ms pre-process, 12.6
Results saved to runs/detect/e
Test verilerinin son halin
[11]
exp
exp2
Derby-Doormat1_png.rf...
Doormats_Coir_Collecti...
Doormats_Coir_Collecti...
Outdoor_Ecology_Colle...
Outdoor_Ecology_Colle...
Round_Geometric_Colle...
oriental_Carina_Collecti...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Marrakech_Col...
oriental_Montana_Colle...
shaggy_Diamond_Desi...
shaggy_Moraccan_Desi...
shaggy_Moraccan_Desi...
train
segment
utils
CITATION.cff
CONTRIBUTING.md
LICENSE
Disk
54.50 GB kullanılabilir
008db425e5427c.jpg
shaggy_Diamond_Design5_png.rf.83a6e2fbb50cbc6ef30f5c6d65abc3ed.jpg
shaggy_Diamond_Design 0.87
7 sn. tamamlanma zamanı: 17:07


YOLO5 train rug.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Düzenleme
Test verilerimizi deniyoruz
shaggy_Moraccan_Design12_png.rf.b0125f63edb4f3871ced075d678bb406.jpg
shaggy_Moraccan_Design_0.82
[13] #test resimlerimizi modele sok
#her farklı resim veya video e
!python detect.py --weights /c
detect: weights=['/content/yol
YOLOv5 v7.0-72-g064365d Pyt
Fusing layers...
Model summary: 157 layers, 705
image 1/15 /content/datasets/R
image 2/15 /content/datasets/R
image 3/15 /content/datasets/R
image 4/15 /content/datasets/R
image 5/15 /content/datasets/R
image 6/15 /content/datasets/R
image 7/15 /content/datasets/R
image 8/15 /content/datasets/R
image 9/15 /content/datasets/R
image 10/15 /content/datasets/
image 11/15 /content/datasets/
image 12/15 /content/datasets/
image 13/15 /content/datasets/
image 14/15 /content/datasets/
image 15/15 /content/datasets/
Speed: 0.5ms pre-process, 12.6
Results saved to runs/detect/e
Test verilerinin son halin
[11]
exp
exp2
Derby-Doormat1_png.rf...
Doormats_Coir_Collecti...
Doormats_Coir_Collecti...
Outdoor_Ecology_Colle...
Outdoor_Ecology_Colle...
Round_Geometric_Colle...
oriental_Carina_Collecti...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Marrakech_Col...
oriental_Montana_Colle...
shaggy_Diamond_Desi...
shaggy_Moraccan_Desi...
shaggy_Moraccan_Desi...
train
segment
utils
CITATION.cff
CONTRIBUTING.md
LICENSE
Disk
54.50 GB kullanılabilir
7 sn. tamamlanma zamanı: 17:07


YOLO5 train rug.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Düzenleme
Test verilerimizi deniyoruz
ced075d678bb406.jpg
shaggy_Moraccan_Design8_png.rf.1f862ec40affe4899c9a906cc9a4daa1.jpg
shaggy_Moraccan_Design_0.93
[13] #test resimlerimizi modele sok
#her farklı resim veya video e
!python detect.py --weights /c
detect: weights=['/content/yol
YOLOv5 v7.0-72-g064365d Pyt
Fusing layers...
Model summary: 157 layers, 705
image 1/15 /content/datasets/R
image 2/15 /content/datasets/R
image 3/15 /content/datasets/R
image 4/15 /content/datasets/R
image 5/15 /content/datasets/R
image 6/15 /content/datasets/R
image 7/15 /content/datasets/R
image 8/15 /content/datasets/R
image 9/15 /content/datasets/R
image 10/15 /content/datasets/
image 11/15 /content/datasets/
image 12/15 /content/datasets/
image 13/15 /content/datasets/
image 14/15 /content/datasets/
image 15/15 /content/datasets/
Speed: 0.5ms pre-process, 12.6
Results saved to runs/detect/e
Test verilerinin son halin
[11]
Derby-Doormat1_png.rf...
Doormats_Coir_Collecti...
Doormats_Coir_Collecti...
Outdoor_Ecology_Colle...
Outdoor_Ecology_Colle...
Round_Geometric_Colle...
oriental_Carina_Collecti...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Lara_Collectio...
oriental_Marrakech_Col...
oriental_Montana_Colle...
shaggy_Diamond_Desi...
shaggy_Moraccan_Desi...
shaggy_Moraccan_Desi...
train
segment
utils
CITATION.cff
CONTRIBUTING.md
LICENSE
README.md
README.zh-CN.md
benchmarks.py
Disk
54.50 GB kullanılabilir
7 sn. tamamlanma zamanı: 17:07

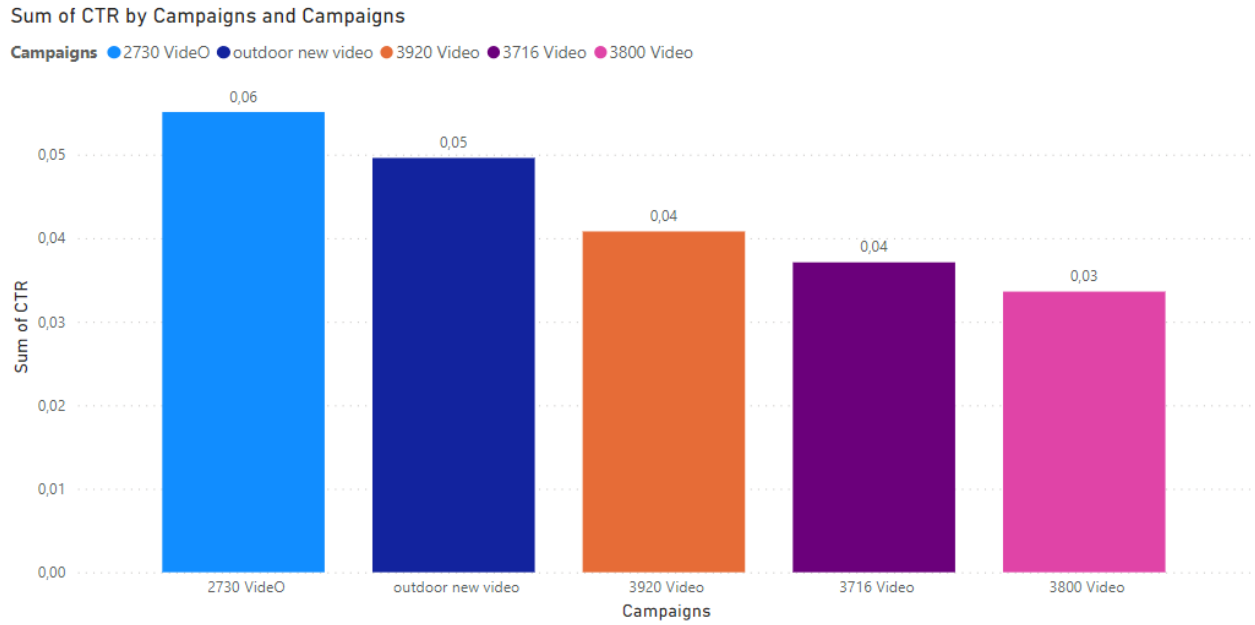
1.18 Recommendations Sprint-3

- The five best-selling models are New Myshaggy, MyShaggy, Eco, and Caimas. These models should be given priority in sales.
- Advertising campaigns can be placed between 12 and 18 o'clock due to the high sales in the afternoons.
- When examining the 10 most frequently returned product categories, it is apparent that the large sizes (120x170, 160x230, and 200x290) of the Montana3716 cream and Montana3800 gray categories are being returned. Upon further investigation of other returned categories, it becomes clear that cream, gray, and black colors are also being returned. Therefore, it would be beneficial for the company to review the sales of these color products.
- Same color, same category and different size products; They are the most sold together products. For example MyShaggy D.Grey 140*200 and MyShaggy D.Grey 80*150 are the most sale products. In this case, customers may order products in different sizes and return products that do not fit them.
- Carpets can be reviewed in terms of color and category.

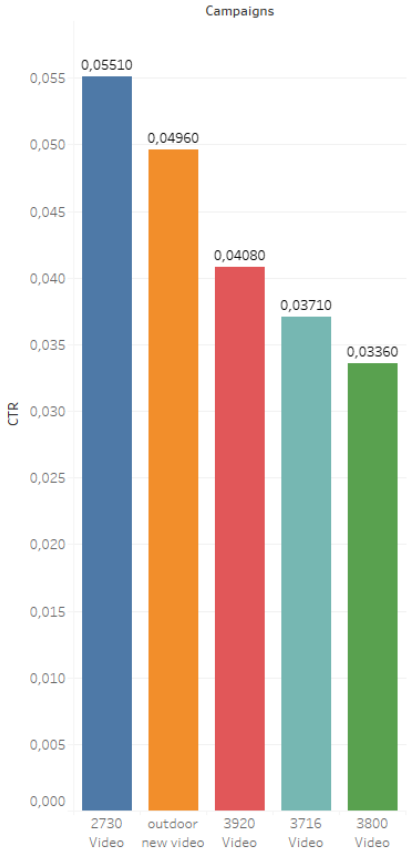
4. SPRINT 4

1.19 Amazon Campaingns

The TOP 5 of CTR

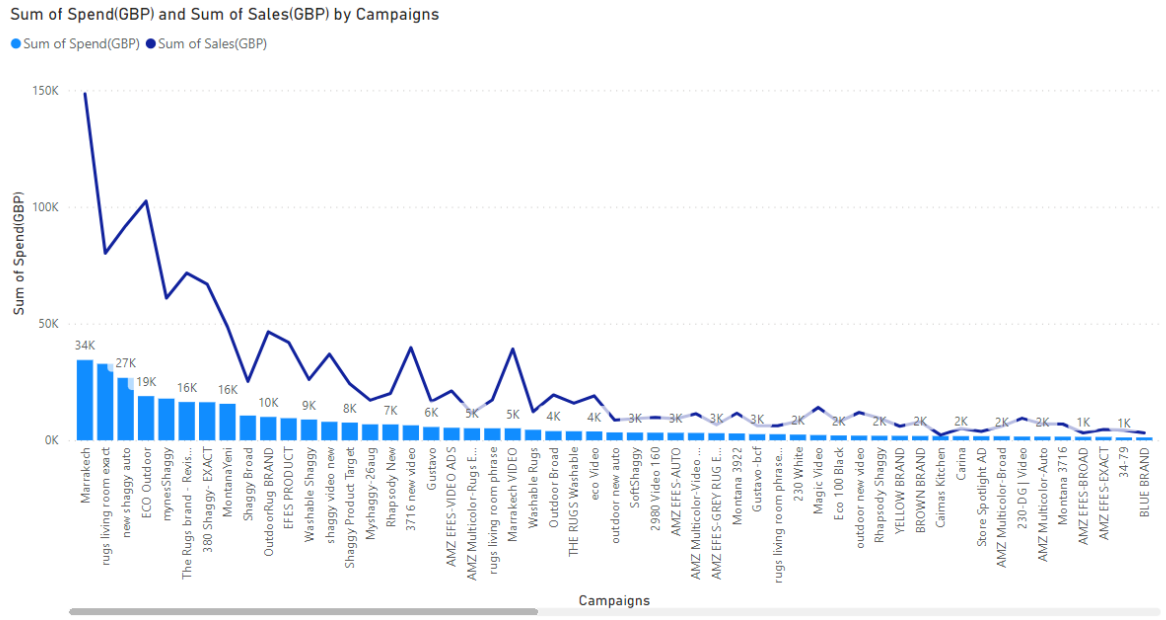


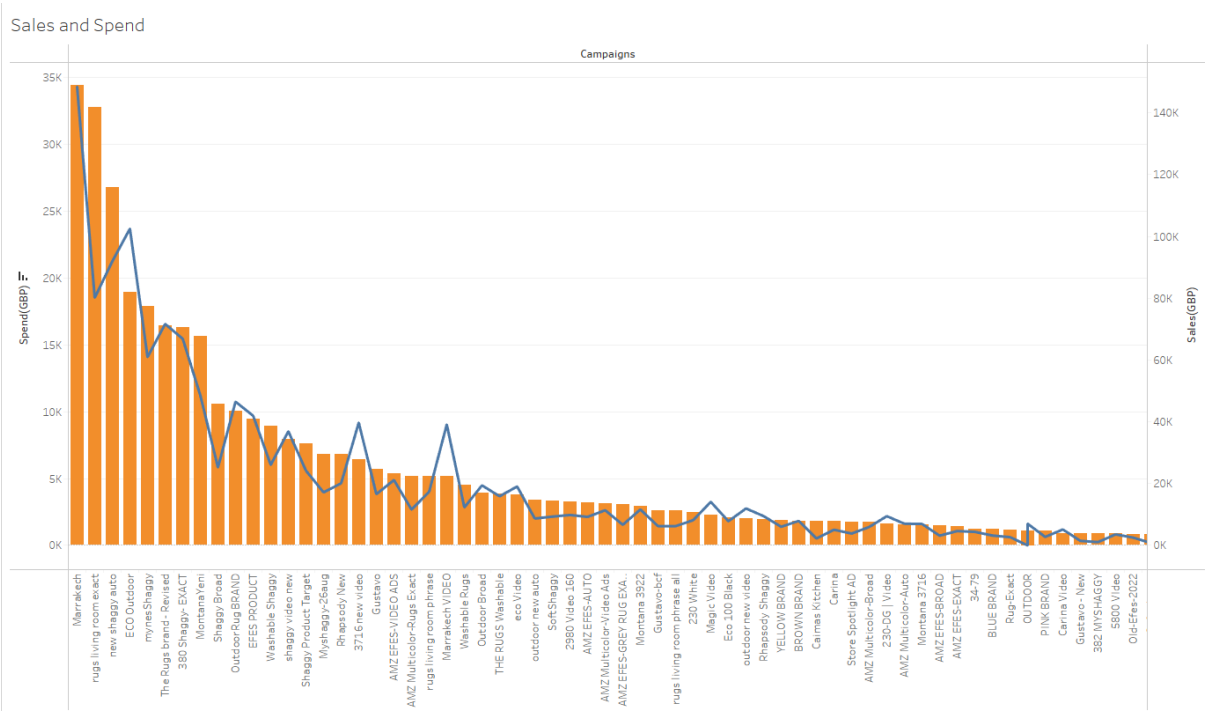
CTR



As it is seen above, the highest number of CTR belongs to ‘2730 video’ campaign among of the Amazon Campaigns. The graph also mentions another top 5 CTR rates of campaigns.

The relation between Spend and Sales

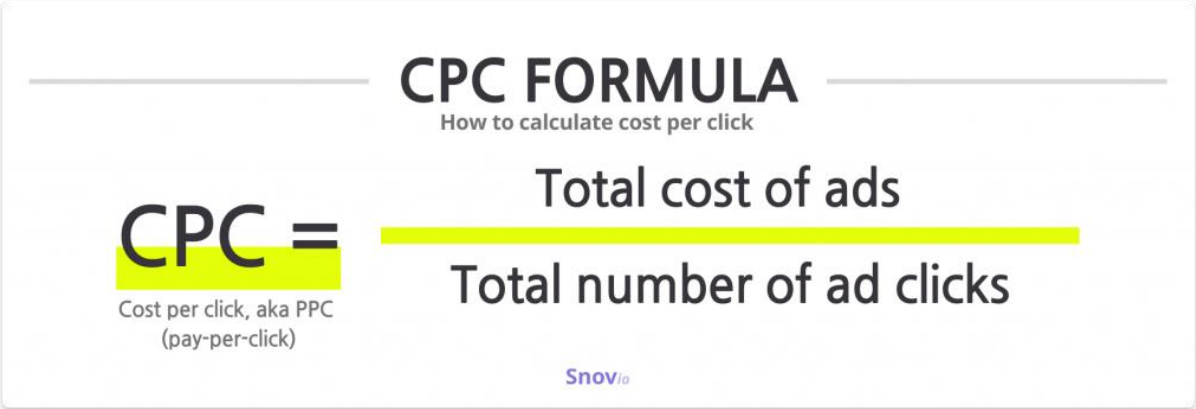


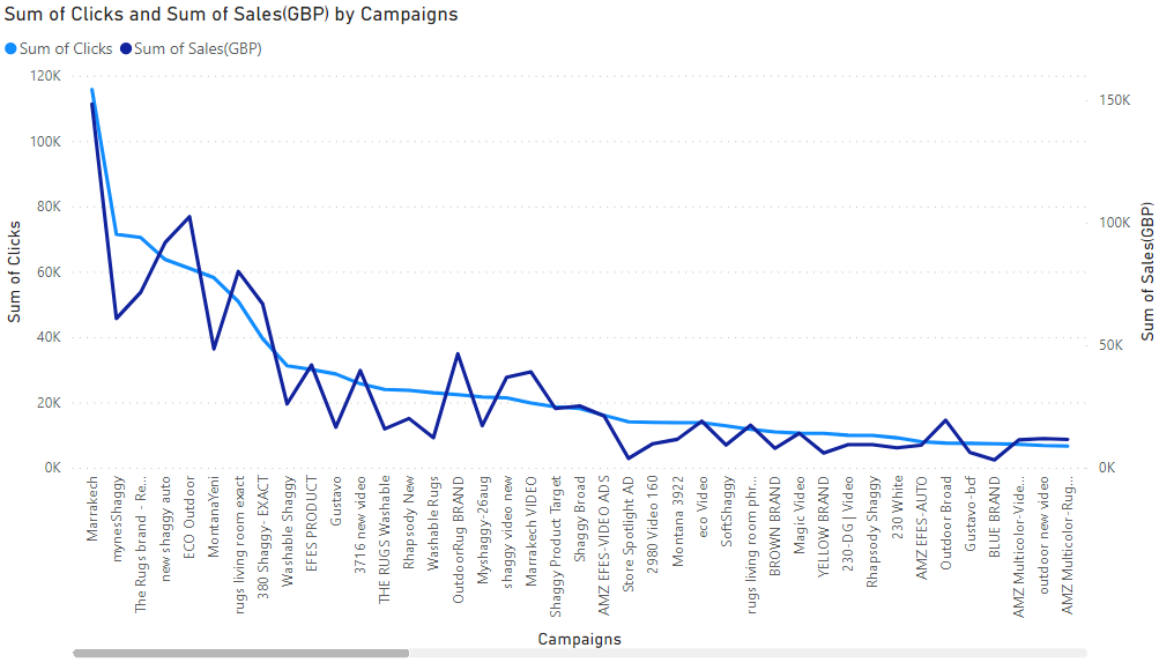


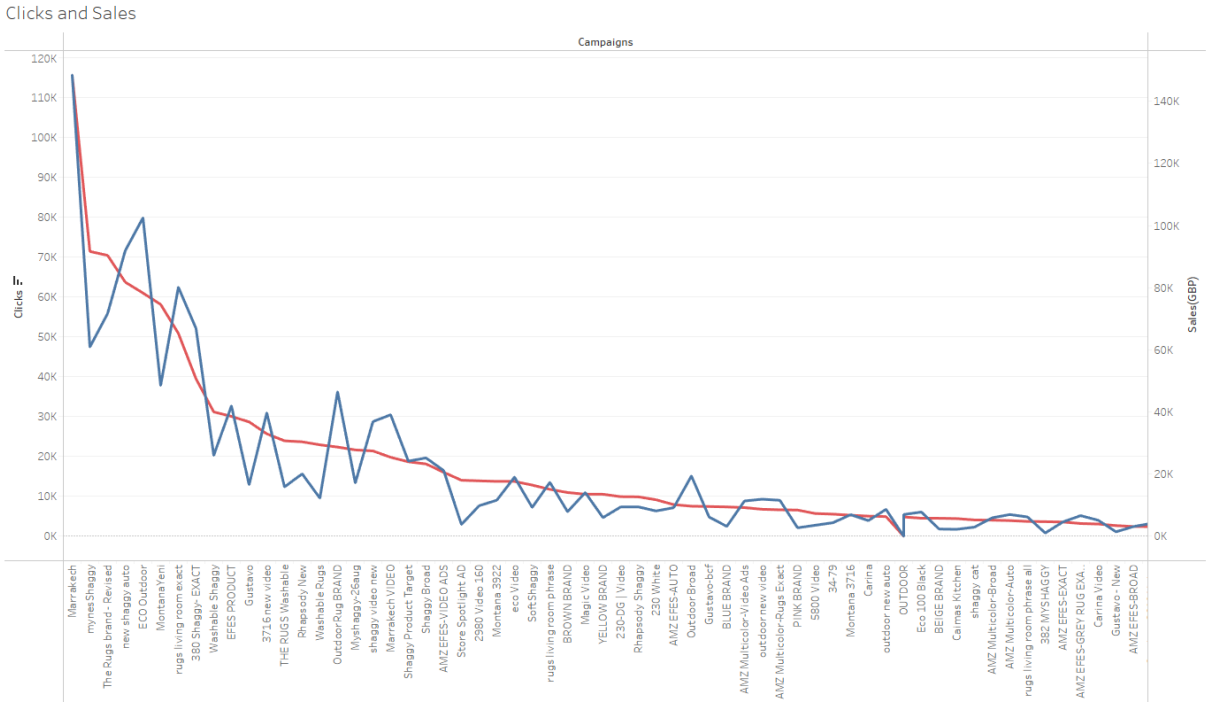
The graph above it is aimed to measure the statistical relation between sales and the amount which spent for the ads.

For almost 34.000£ was spent for the Marrakech campaign. On the other hand, sales of ‘Marrakech Campaign’ approximately 150 thousand dollars were achieved. Referring to this chart, it can be said that there is a relatively positive relationship between advertising expenditure and income.

The Relation between CPC and Income

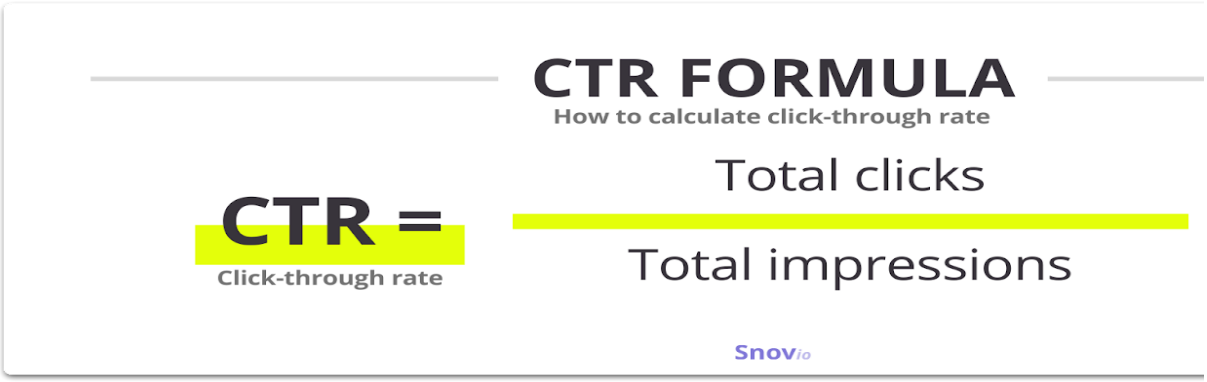


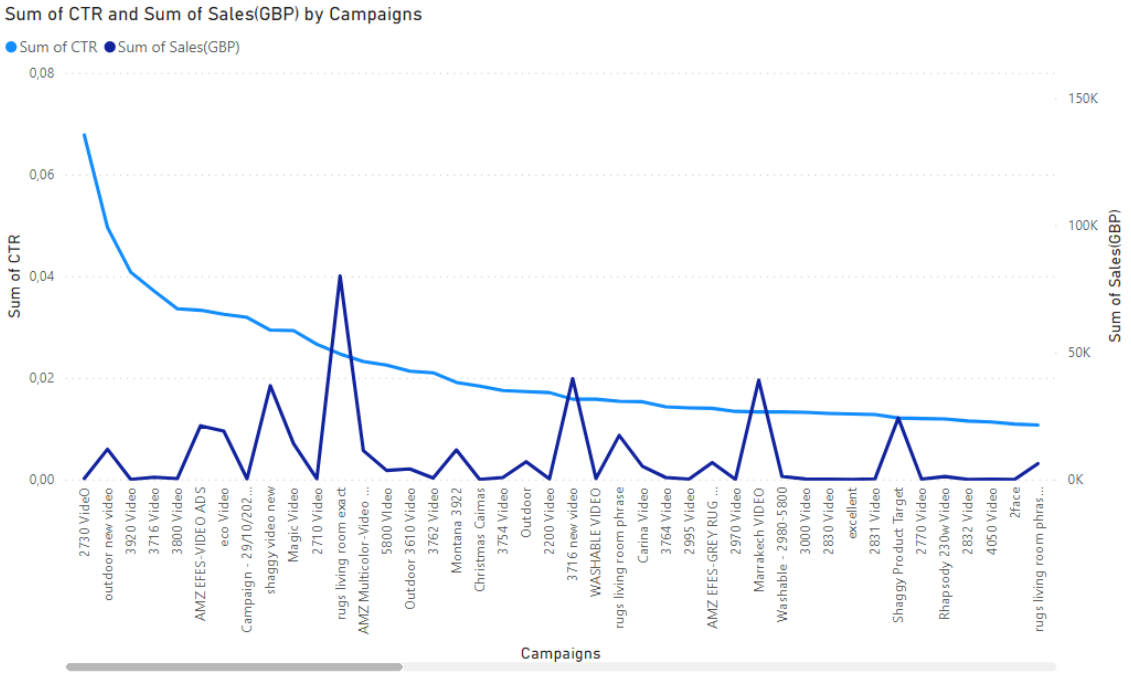


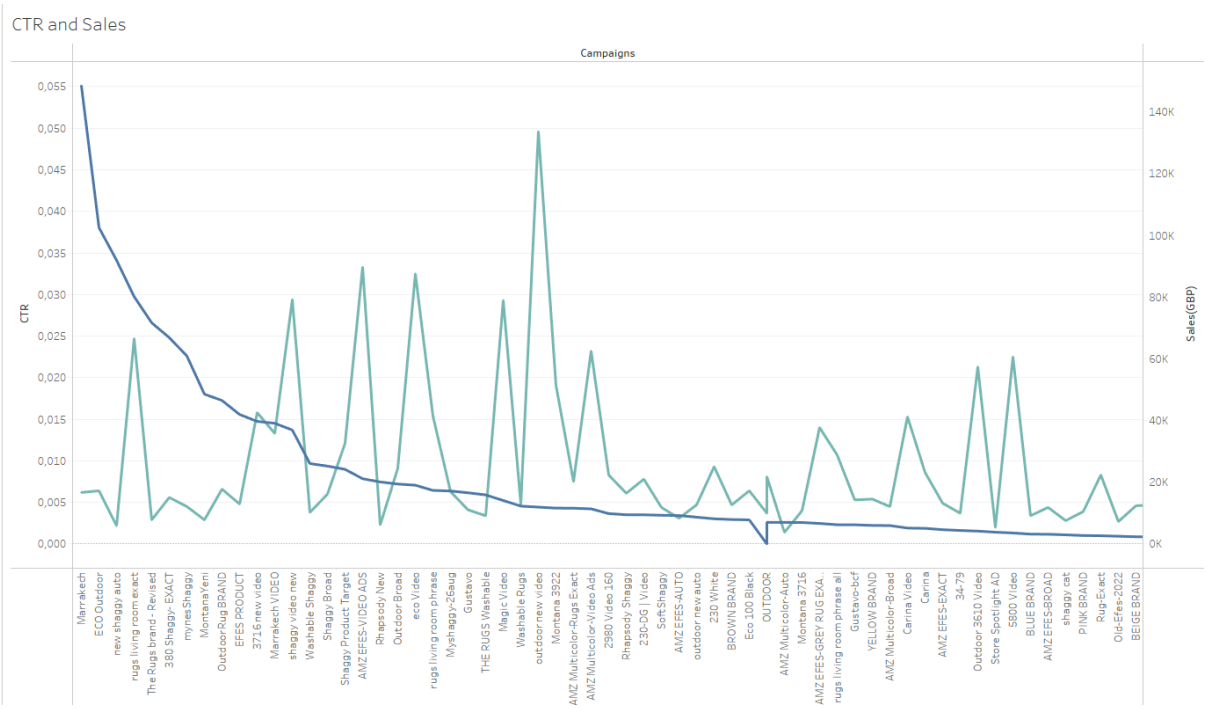


The graph above tell us the relation between CPC and Income at the end of Amazon campaigns. From this graph the idea can come out, that the relation between CPC and Income of a campaign is relative positive .

The Relation between CTR and Sales(GBP)

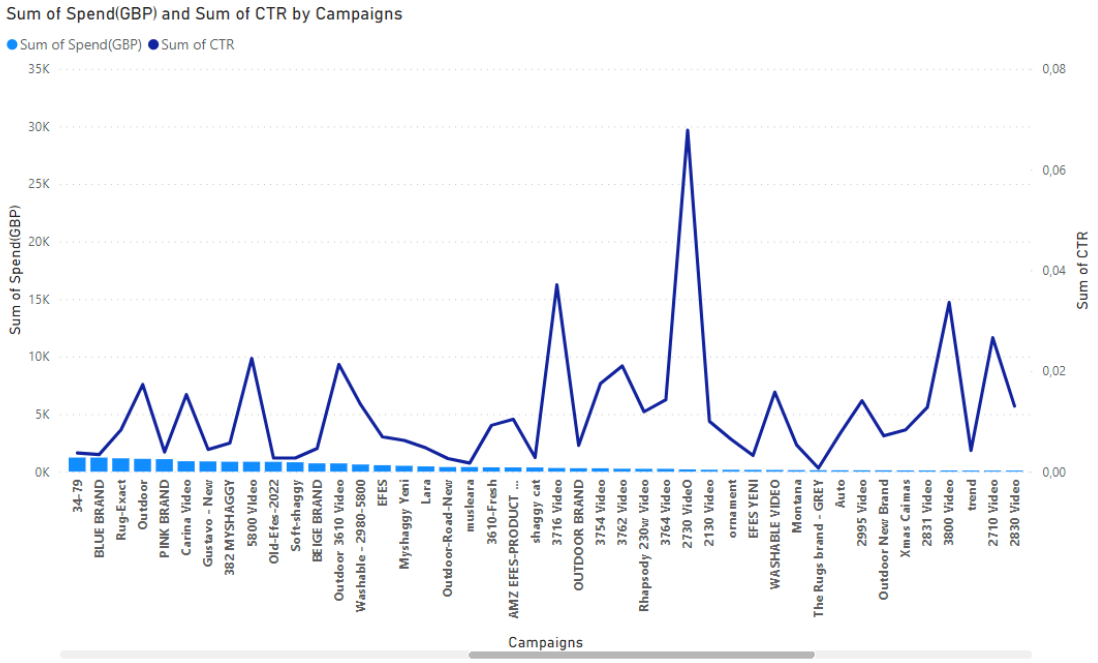


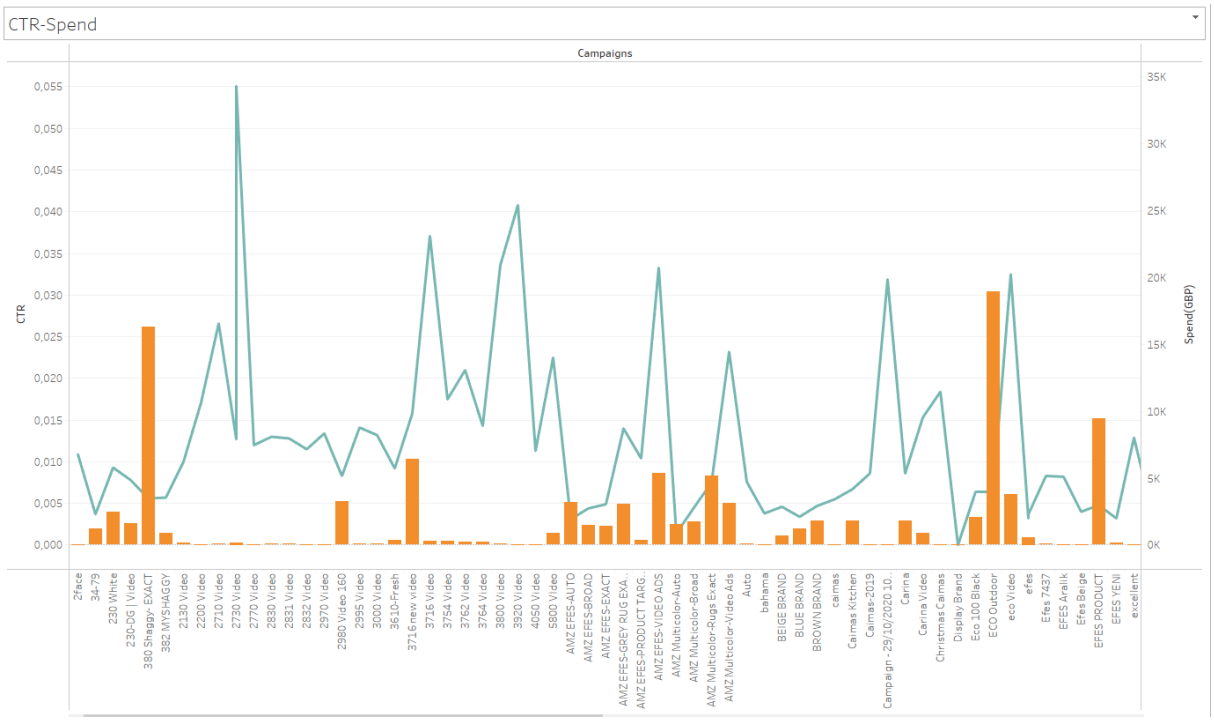




It can be easily said that the relation between CTR and Sales(GBP) is neither positive nor negative . Even if the campaign ‘2730’ have the highest CTR rate, but it has almost 360 pounds sales amount. In compare to ‘Rugs Living room campaign’ despite it has lower CTR rate but almost 80K pounds sales amount.

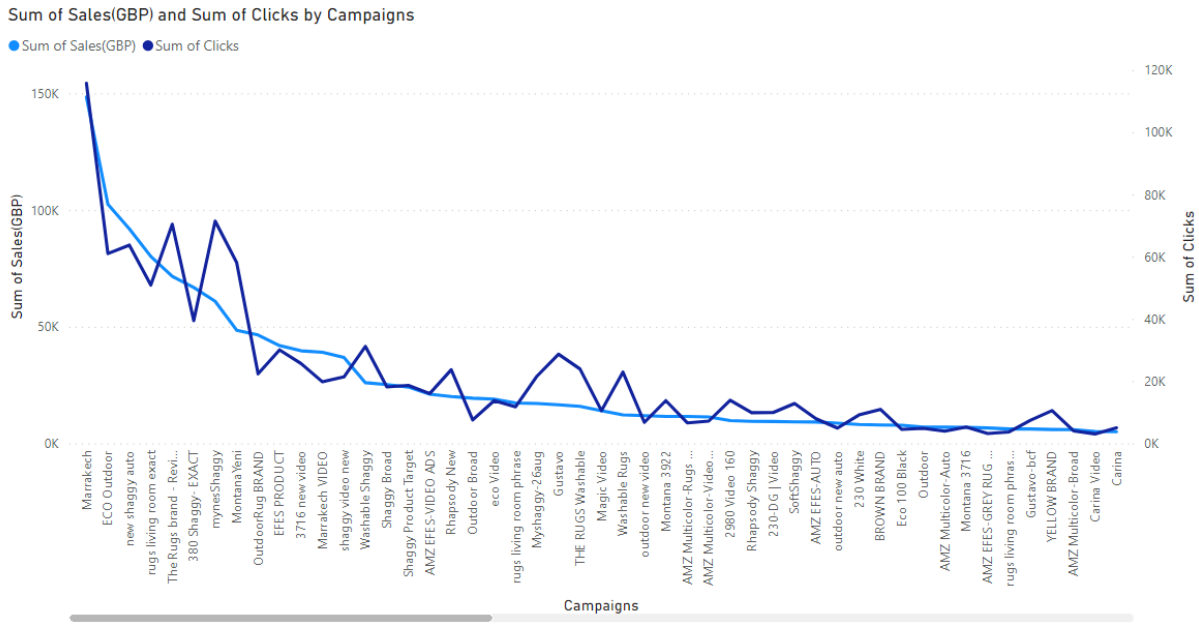
The relation between Spent(GBP) and CTR(clicks)

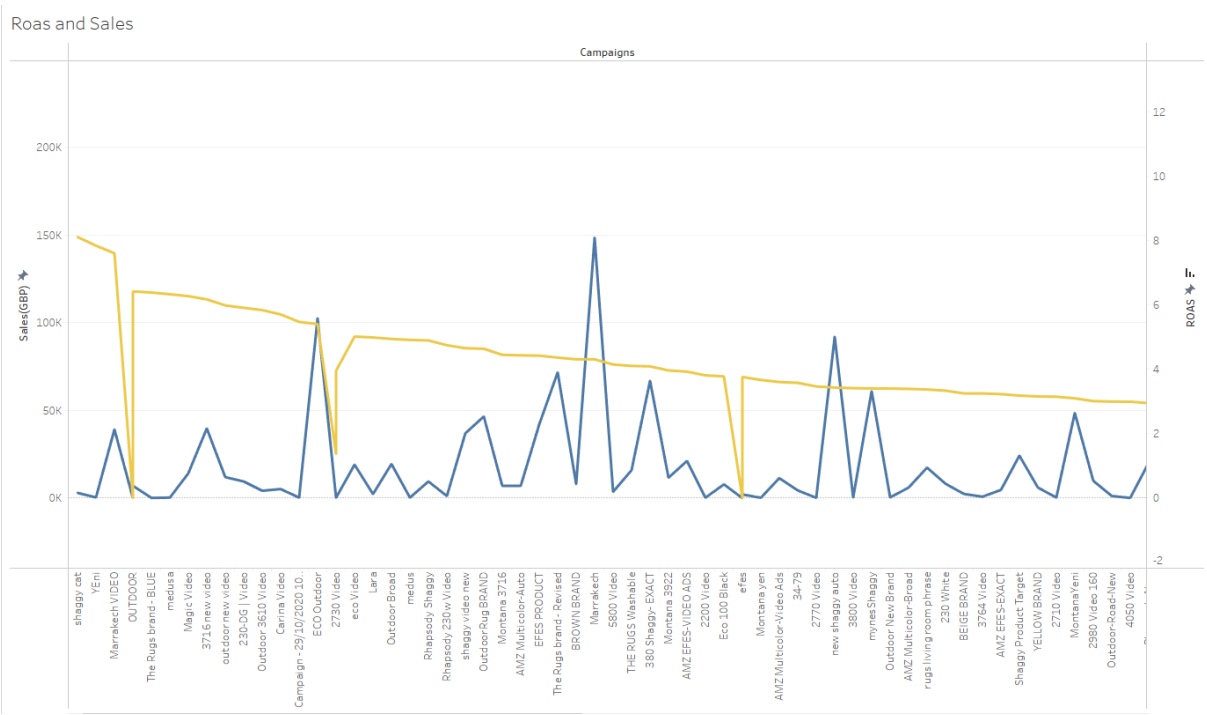




As can be seen from this graph, it can be said that a high CTR ratio reduces advertising expenditures. While the CTR value of the campaign 2730 is 0.07, the total amount spent on the advertisement is seen as 193 pounds.

1.20 The Relation between Sales(GBP) and Clicks

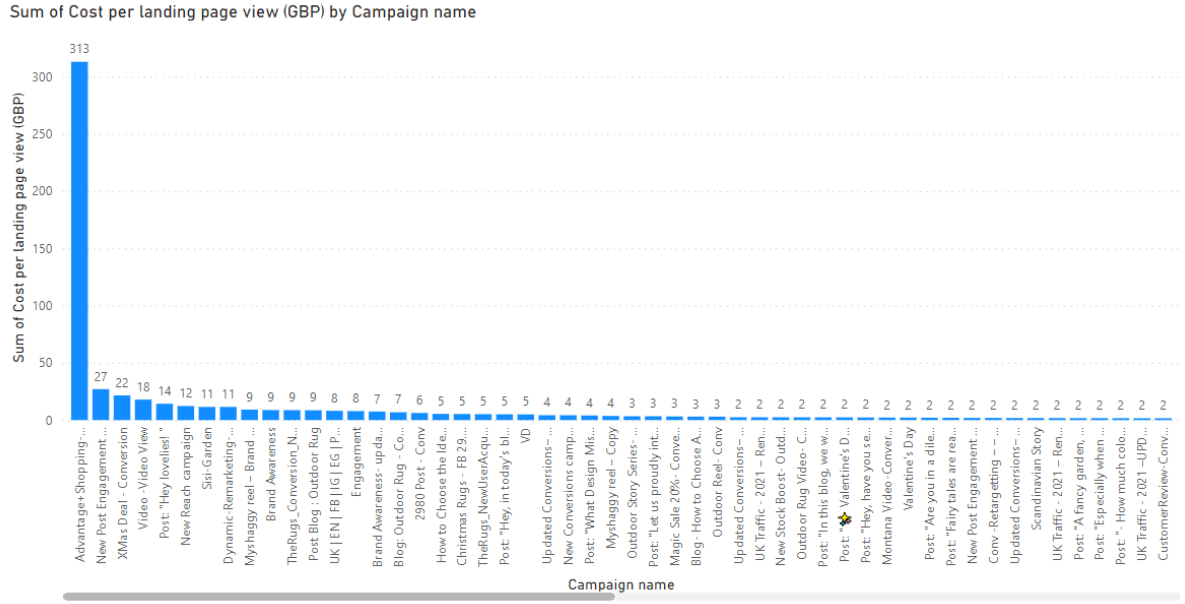


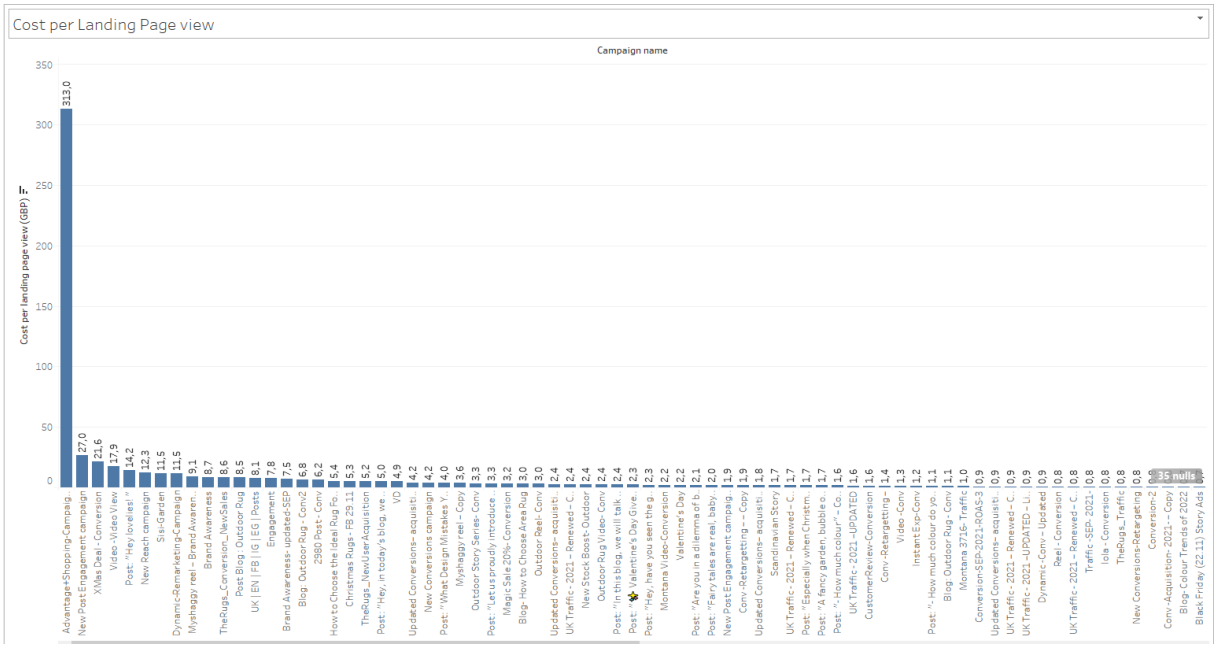


The result to be drawn from the chart above will be that the product with more clicks will sell more. Therefore, it can be said that there is a positive relationship between clicks and revenues.

1.21 Other Campaigns

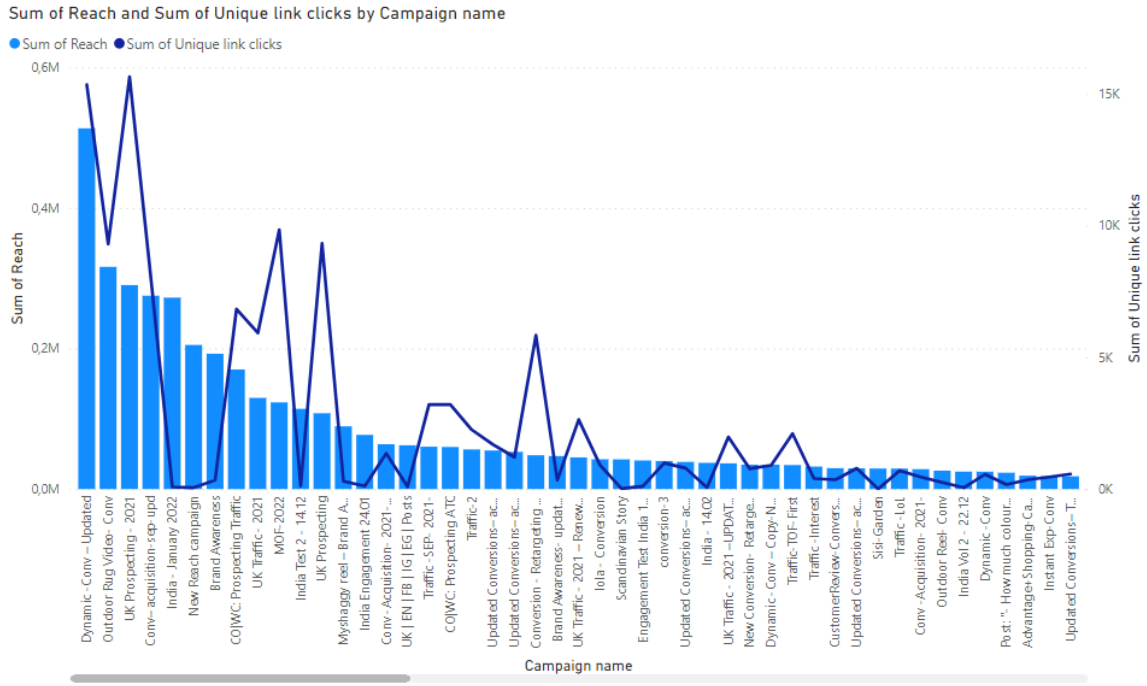
Total Cost per Landing Page View





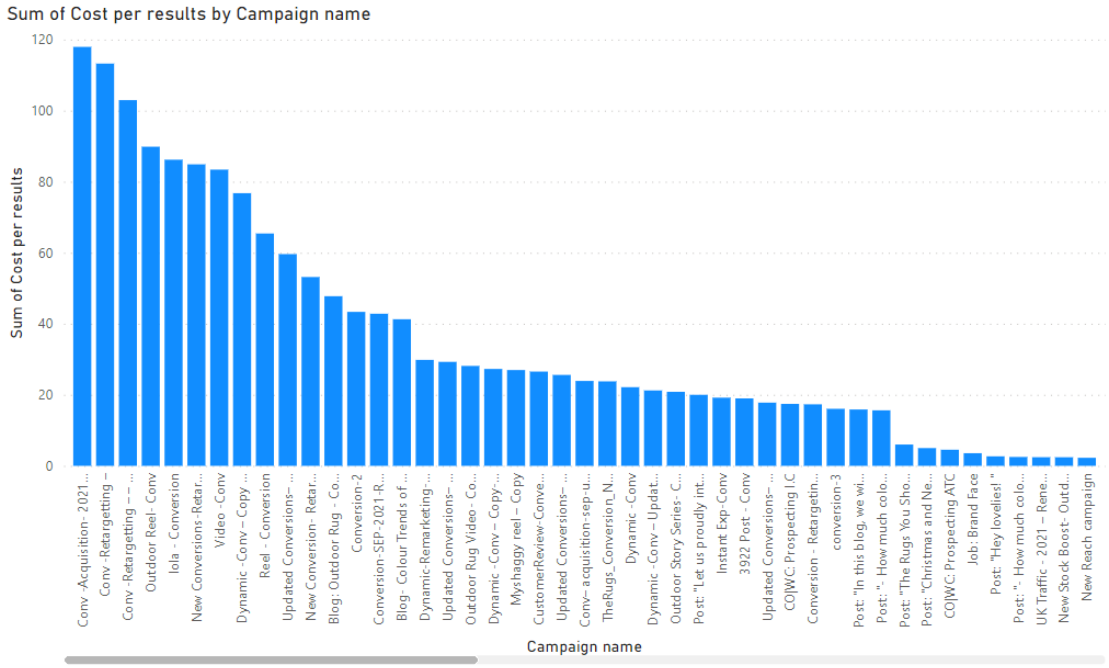
The chart above shows the total spend of landing page views. In this context, the Advantage Shopping campaign has been the landing page with the highest tracking cost in total.

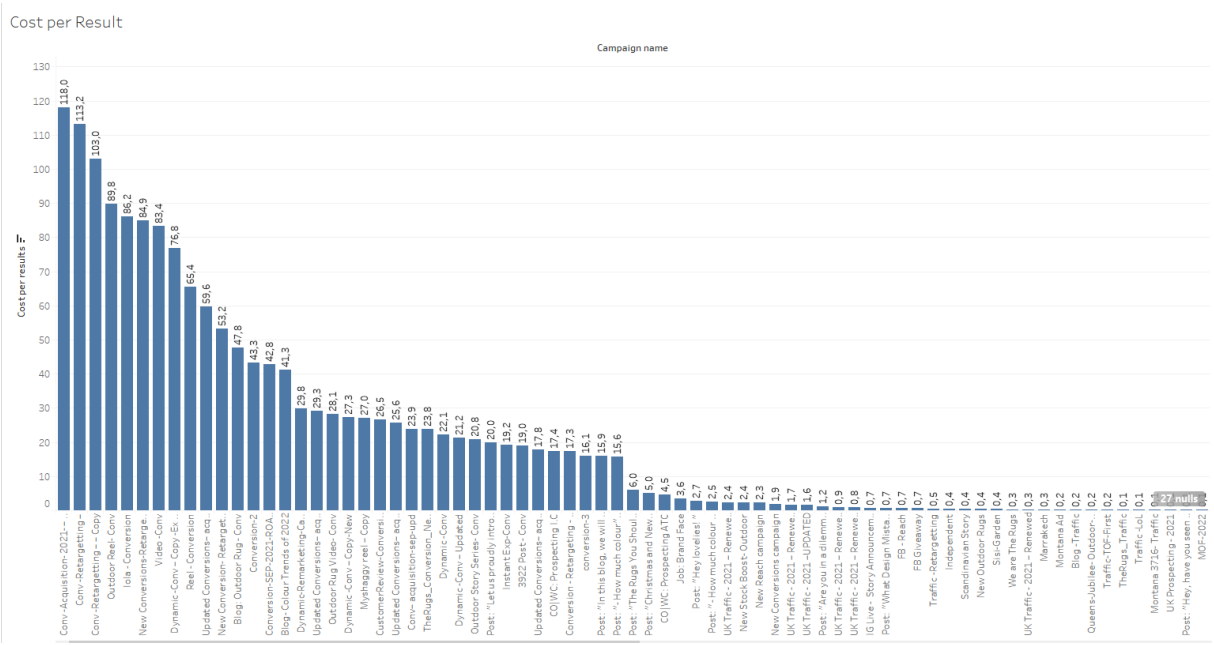
Reach and Unique Link Clicks (A/B)



The graphic above is intended to show the relationship between ad reachs and clicks. The conclusion to be drawn from this graph is that there is a relatively positive relationship between interaction and clicking. Although the number of people reached in the India January, New Reach and Brand Awareness campaigns in the ad set is thousands; The number of clicks is very low. It is recommended to improve the content of these campaigns.

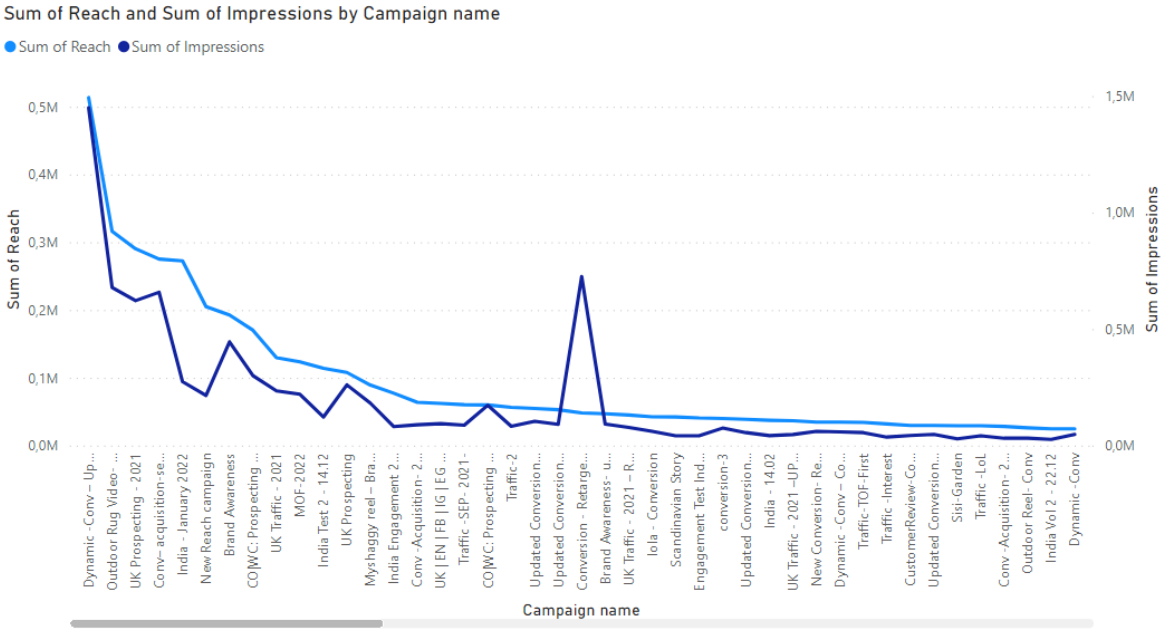
Cost per Results

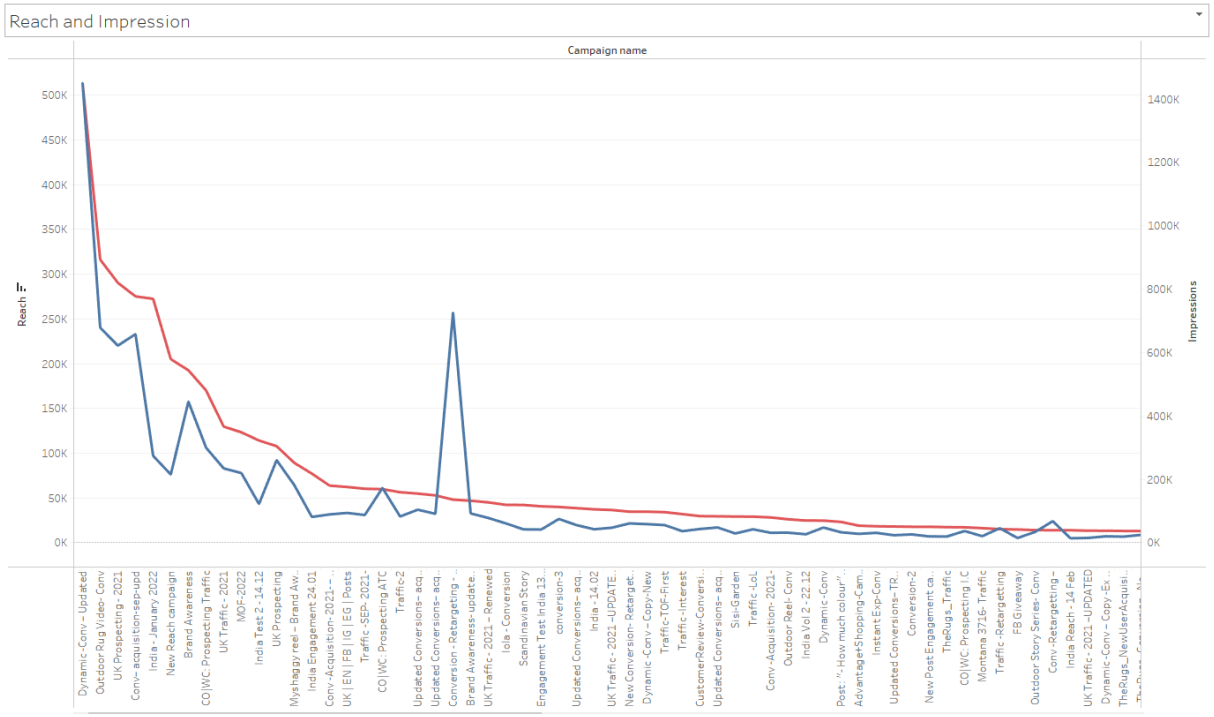




The above chart shows the CPR figures of the campaigns. It belongs mostly to the CTR Conv-Acquisition campaign.

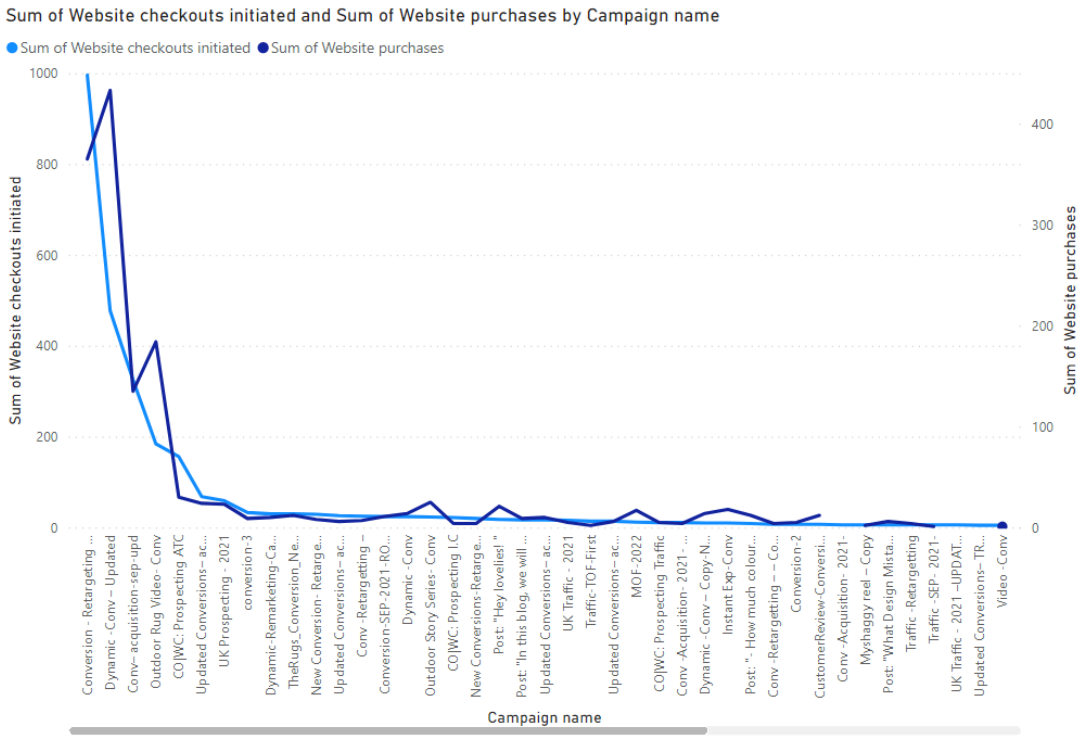
1.22 Reach and Impression(A/B)

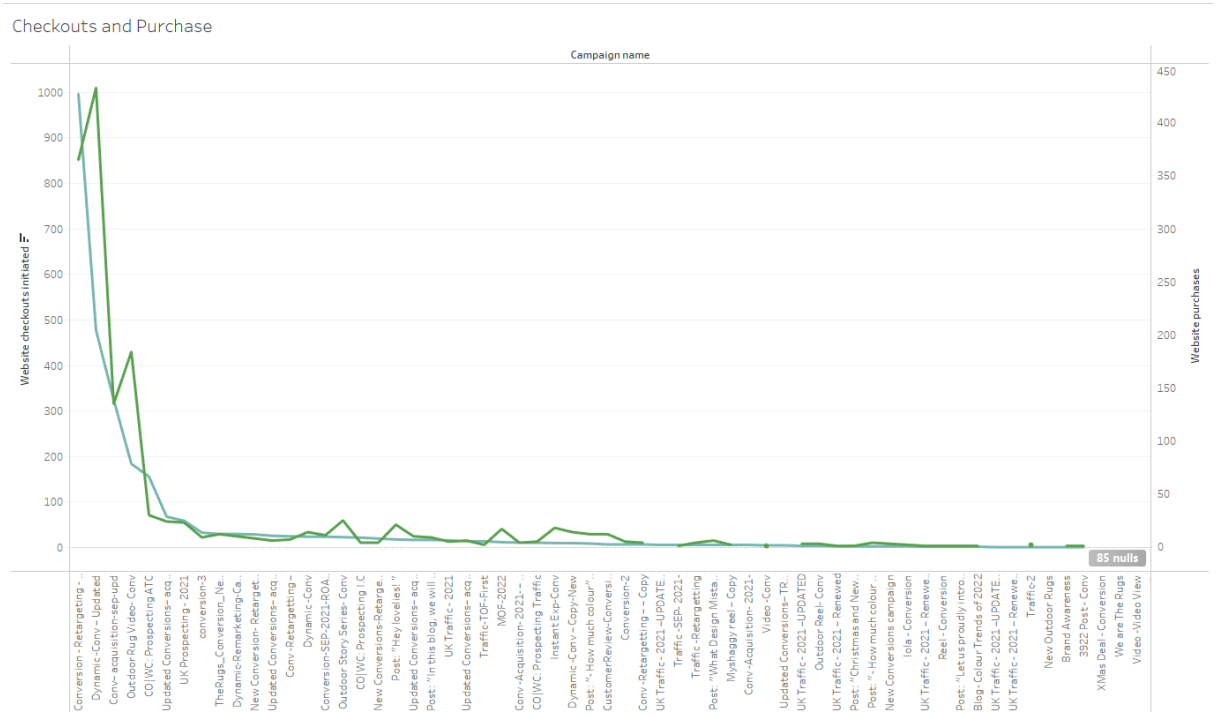




In the image above, you can see the relationship between interaction and the number of people reached. What can be said in this case is that there is a positive relationship between reach and impression.

Check-outs and Purchase

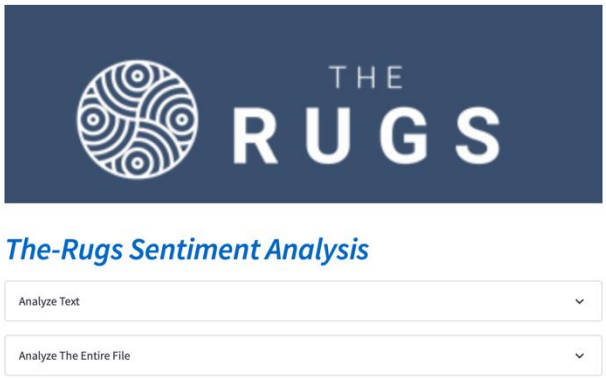




In the graphic above, the relationship between the products placed in the basket and waiting and the products purchased is presented. The result obtained here is that there is a positive relationship between these two.

5. SPRINT 5

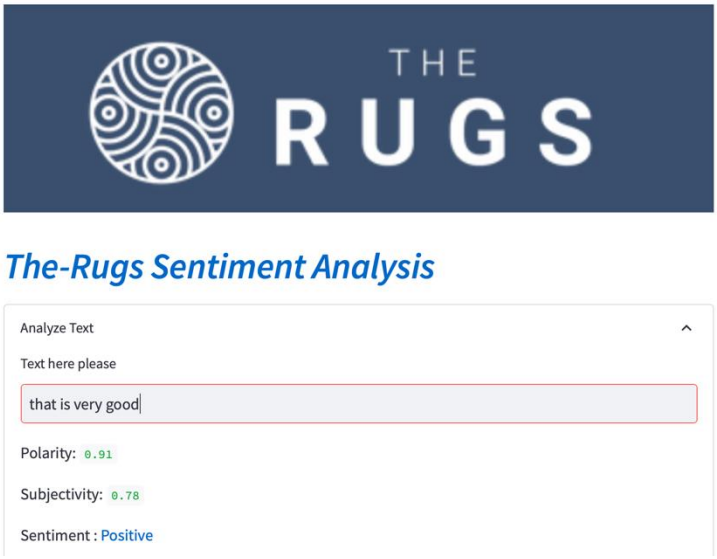
1.23 Sentiment Analysis APP



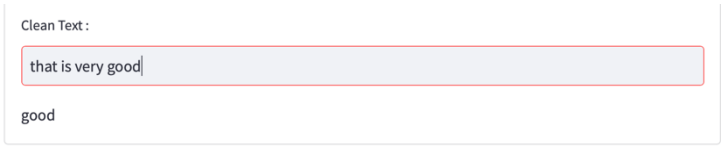
We have an interface in this way.

We can do 3 operations here.

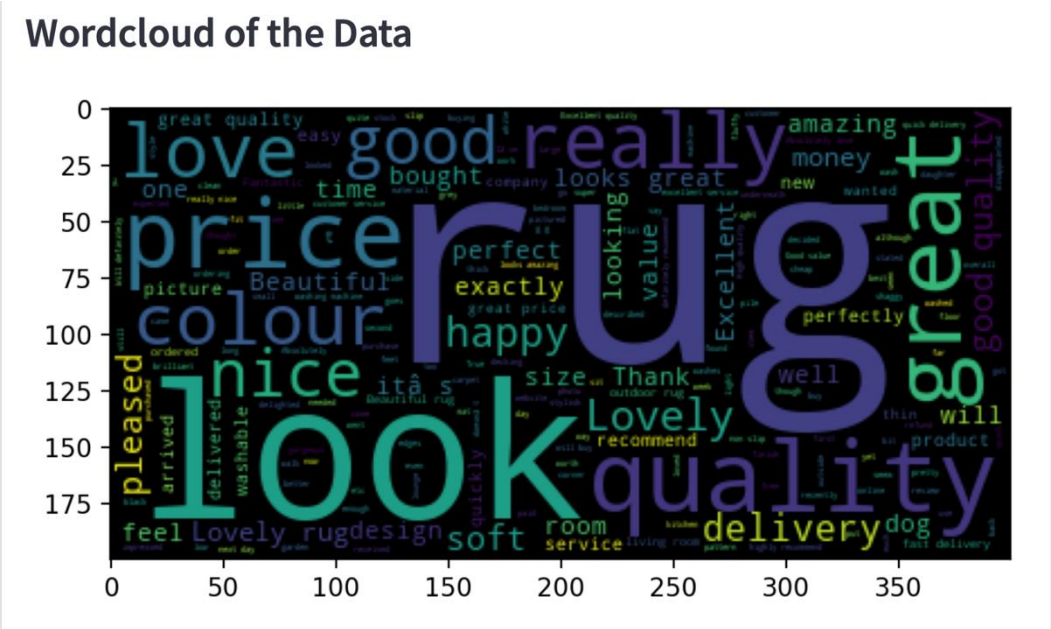
If a sentence is entered manually, it analyzes the sentence.



And if you are wondering which words are actually more important for our application, and which words are evaluated according to a manually entered text, this can be viewed on the 'Clean Text' line.



When we add the EXCEL file that contains the comments collectively, a WordCloud is created immediately according to the words in these comments.



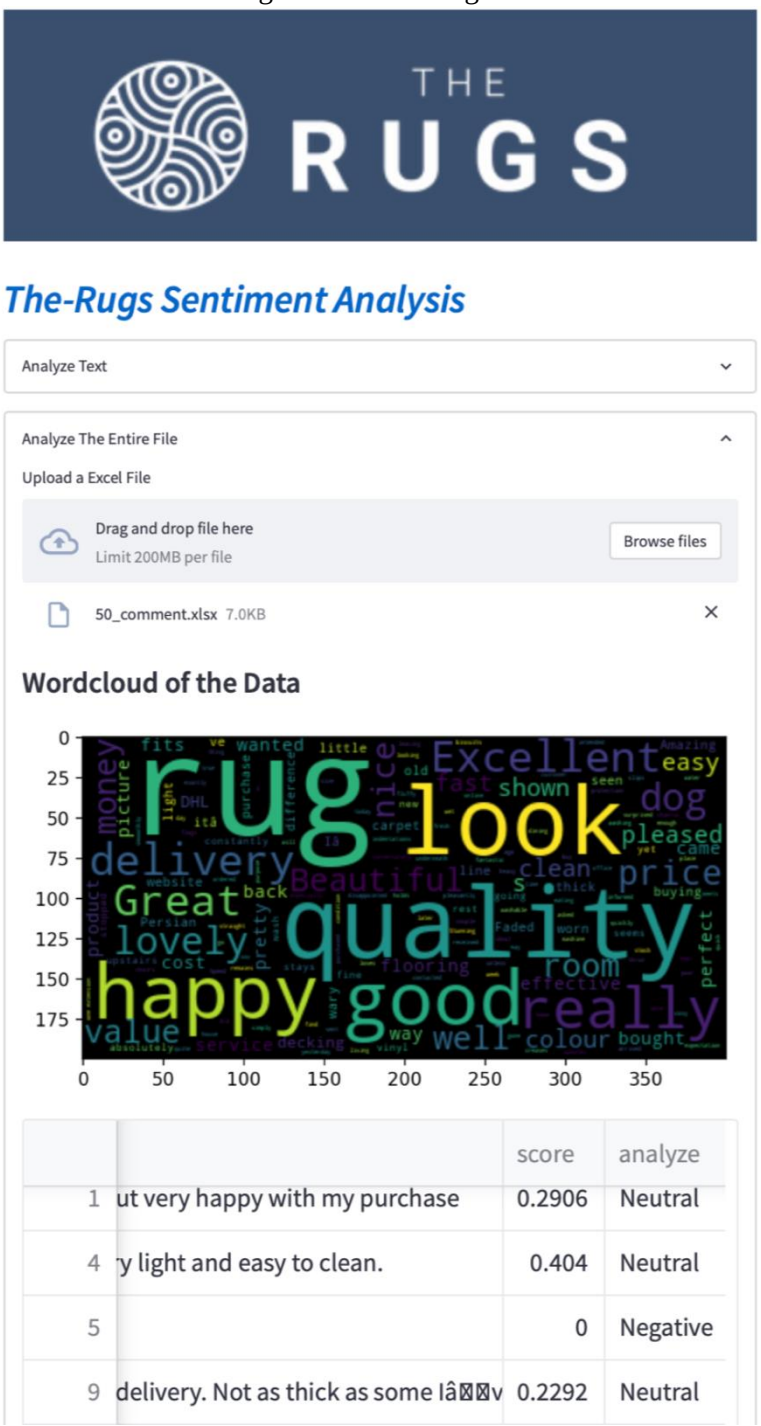
And it analyzes all the comments one by one and gives the sentiment results and scores to them.

	COMMENTS	score	analyze
0	Very pleased with the quality, look, and very good price too.	0.78	Positive
1	Beautiful rug was a little wary of buying on line but very happy with my purchase	0.2906	Neutral
2	Amazing rug, looks like an old Persian carpet, very light and easy to clean.	0.404	Neutral
3	"Faded , worn out. "	0	Negative
4	Really nice rug, colour as shown on website, fast delivery. Not as thick as some lâ⊠⊠v	0.2292	Neutral
5	Didn't wash it yet, but quality seems good and the dog stopped constantly going ups	0.2788	Neutral
6	I have vinyl flooring in my conservatory and the chairs were leaving heavy indentatio	0.5	Positive
7	I bought a rug from the rugs about a week ago and received it A day later the rug cam	0.16	Neutral
8	Excellent quality and fantastic service	0.7	Positive
9	Very happy with thisg	1	Positive

Download data as CSV

In this way, negative comments can be seen immediately and an immediate reaction can be given. In this way, customer satisfaction is increased.

This is how the APP looks when a file is dropped. I shrunk the screen to fit them all in the screenshot. That's why some of the comments are not visible. But in normal working time, it is perfectly fine and the comment on each line can be read regardless of its length.



This way you can see the results of the comments and if desired, these results can be downloaded as a CSV file.

Parts of the APP Project that need to be completed:

- I used a ready-made library called Textblob for Sentiment analysis. But in the text tests I made afterwards, I saw that he made obvious analysis mistakes in some places.

- As a solution to this, I want to include Bert model from Hugging Face in App instead of Textblob.

We can only export Excel files to the APP:

- I want to write another function here and read and evaluate whichever Excel and CSV formats are discarded.
- We can only download the results in CSV format if desired. I want to add one more option here. If desired, it should be downloadable as Excel.

1.23.1 Sentiment Analysis with Vader and Roberta

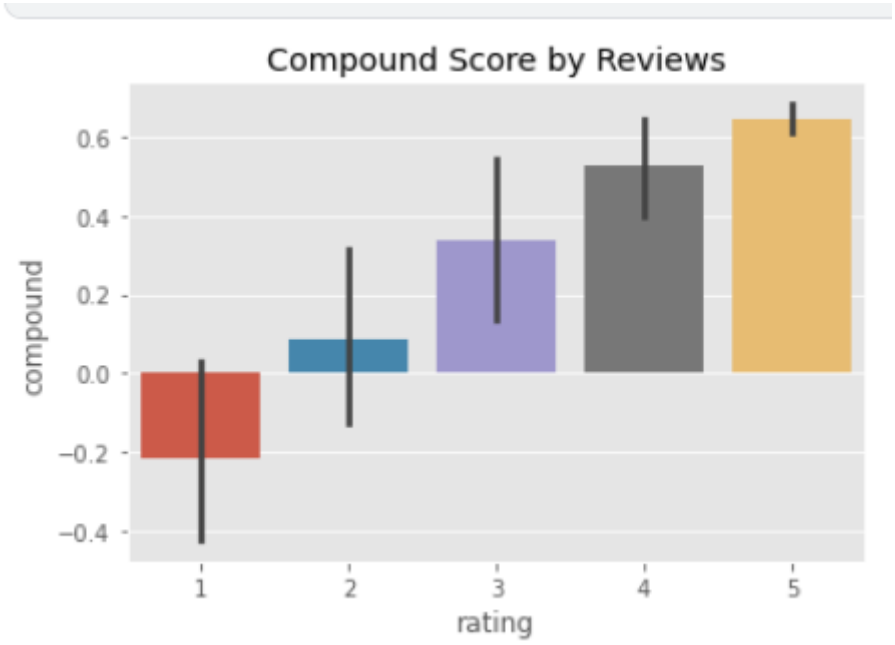
In this Sentiment Analysis we implemented a Vader and Roberta pretrained models on the 'Comments' data set. The results are shown as it follows.

Amount of Ratings

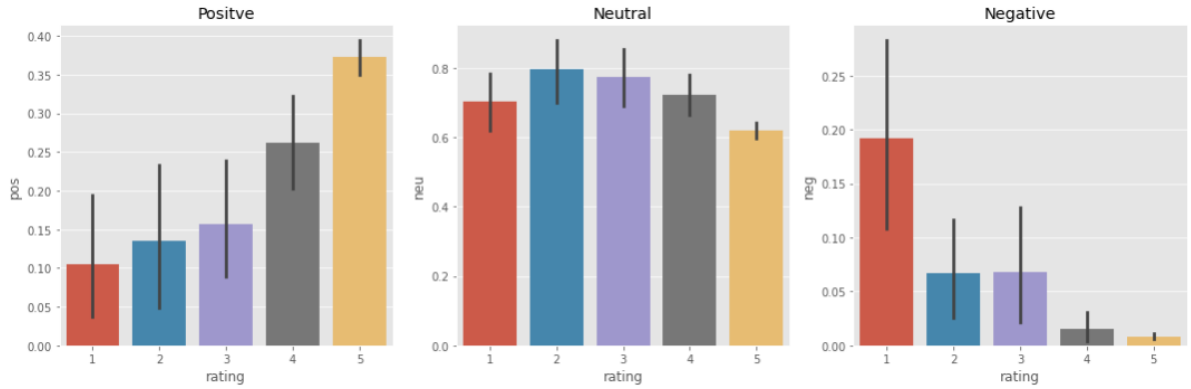
As it is seen above the most of the ratings accumulated on 5. That means we have lots of ‘5’ ratings.

Vader Sentiment Analysis

Vader is pre-trained ‘NLP’ library which is used to analyze text or sentence. The main purpose of Vader is Classifying the polarity of a given text at the document, sentence, or feature/aspect level—whether the expressed opinion in a document, sentence, or entity feature/aspect is positive, negative, or neutral—is a fundamental task in sentiment analysis. Advanced sentiment classification " polarity" considers emotional states such as ‘positive’,’negative’ , ‘neutral and ‘compound’.



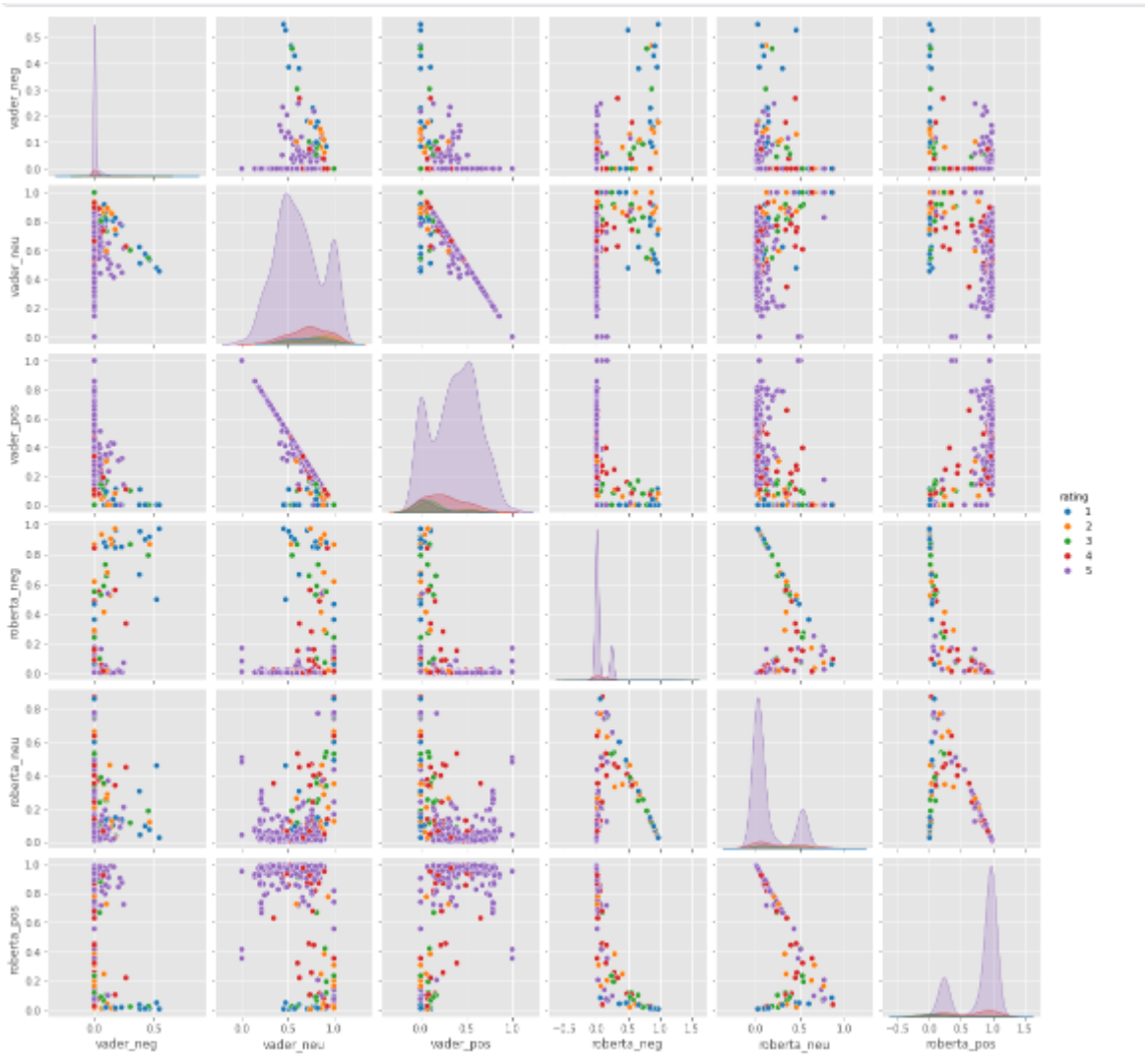
As it is shown above, the data set is the data set consists of relatively positive comments according to the vader library.



The data above has proved, the sentiment classification sentiment shows that the classification is working quite properly.

Sentiment Analysis with Roberta

Roberta is one of the pre-trained ‘NLP’ models to implement sentiment analysis on text and Word. Roberta is similar to Vader in several points much more better than Vader.

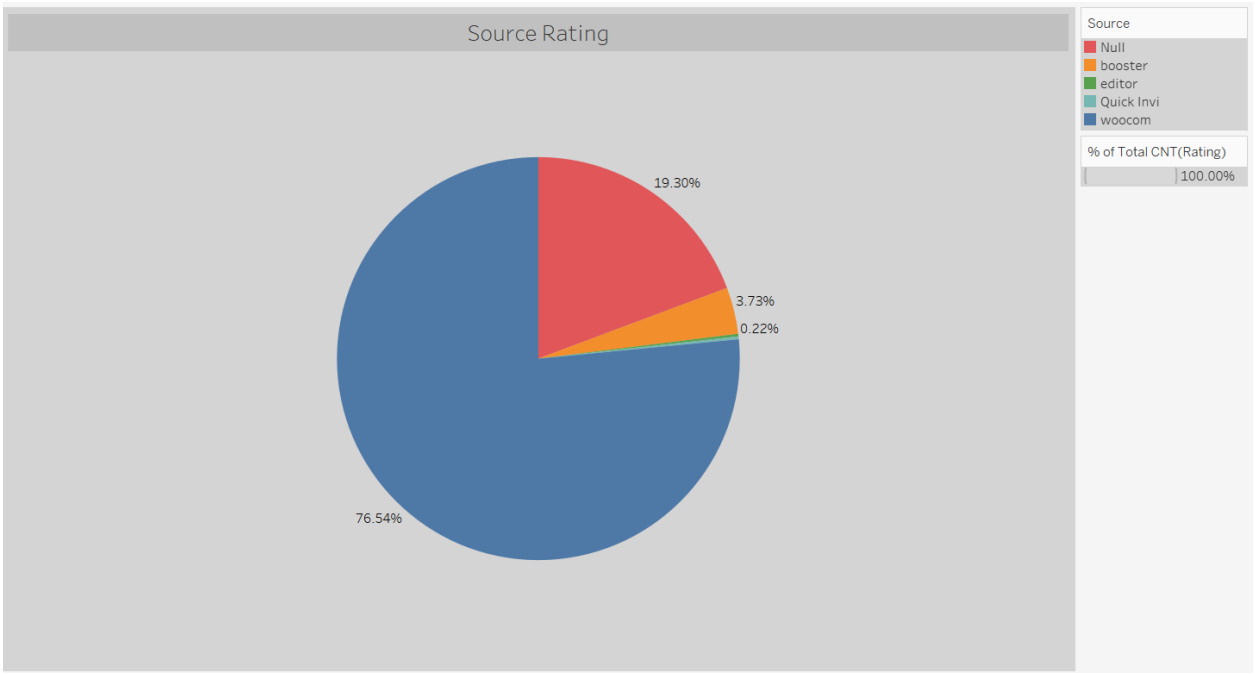


Comparing to Vader Roberta gave us better results on sentiment analysis.

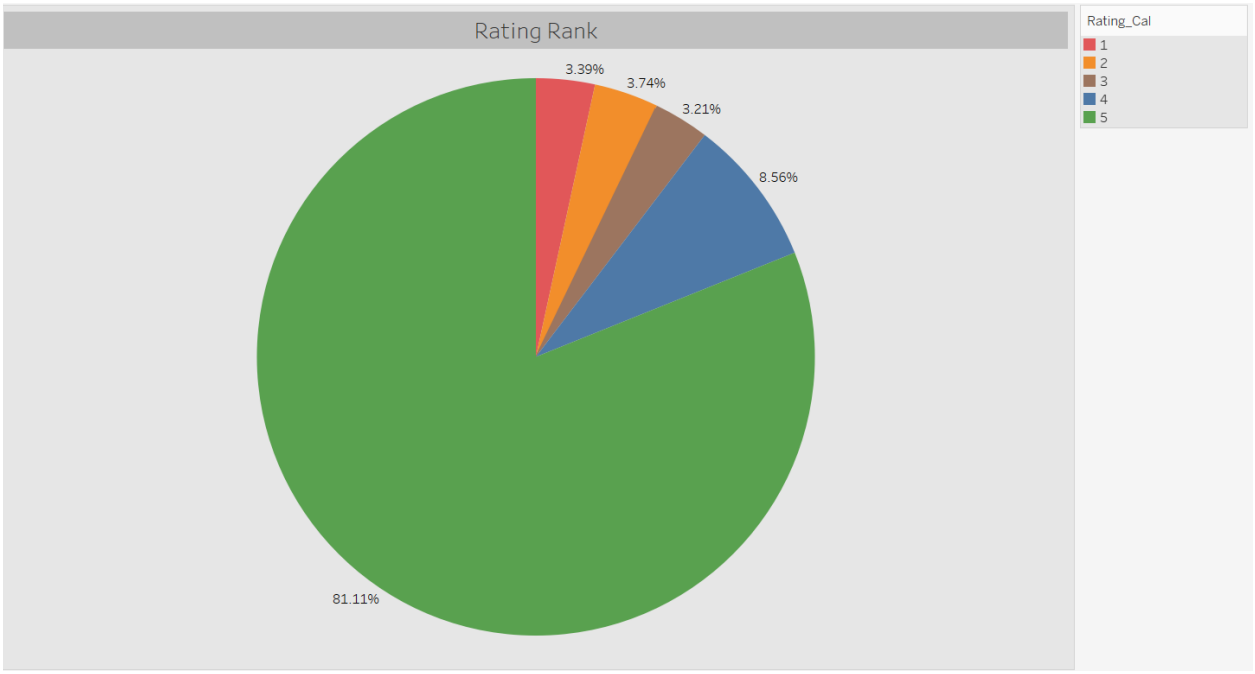
1.23.2 Recommendations

The most important suggestion to be made to the company in this section is to update the tabs for website comments and make it mandatory for customers to comment in the comment tabs.

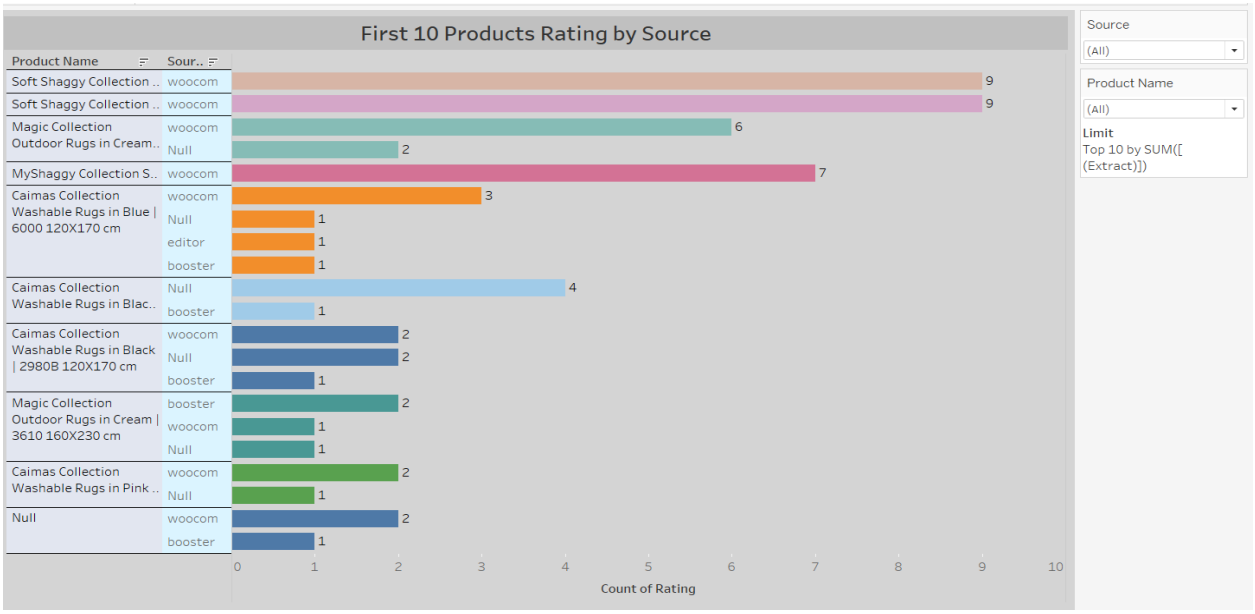
1.23.3 CUSTOMER ANALYSIS



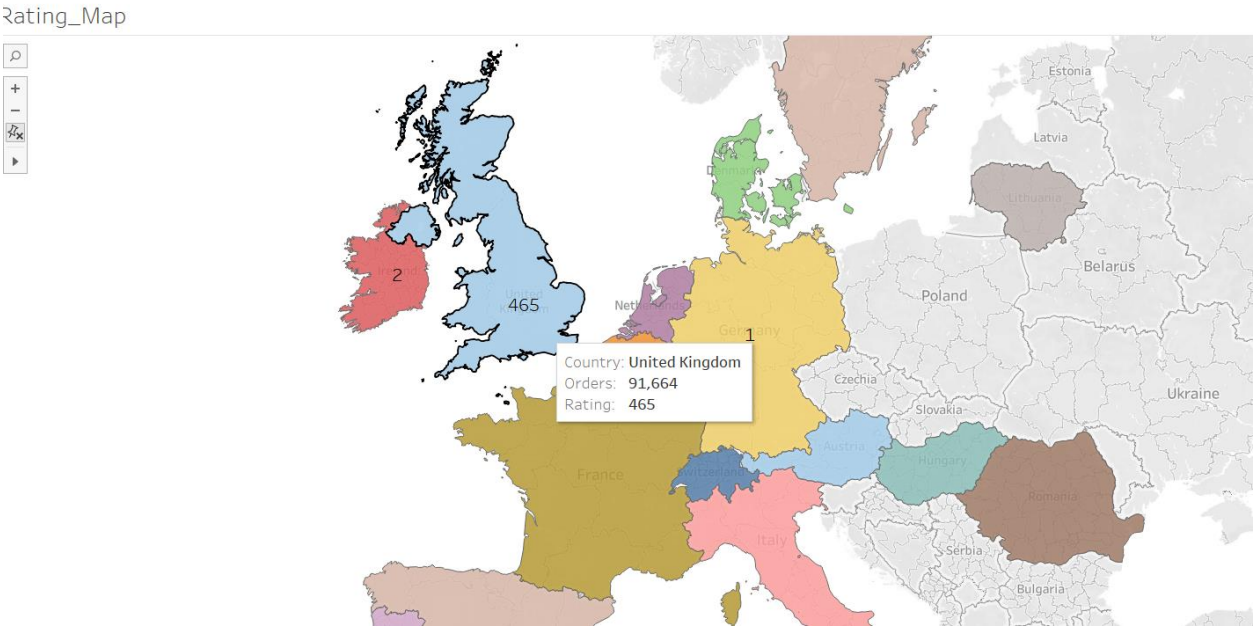
The highest rating was made from the woocommerce platform with 76.54%.



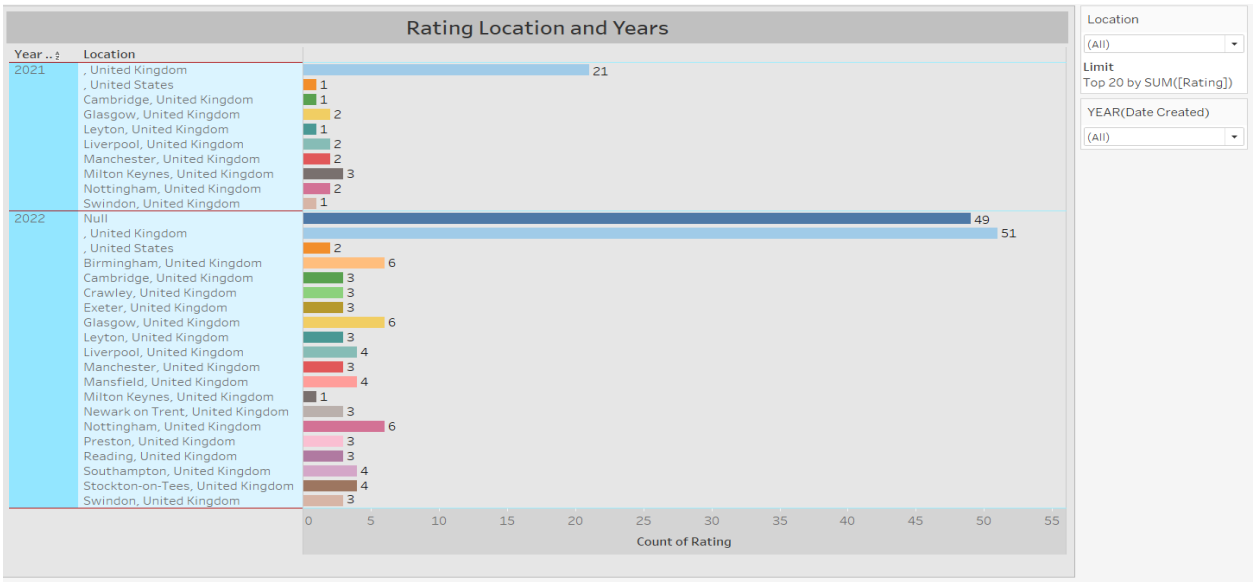
It is seen that the highest rating is made as “5” with a rate of 81.11%. At the rate of 3.39%, the lowest (1) rating was taken.



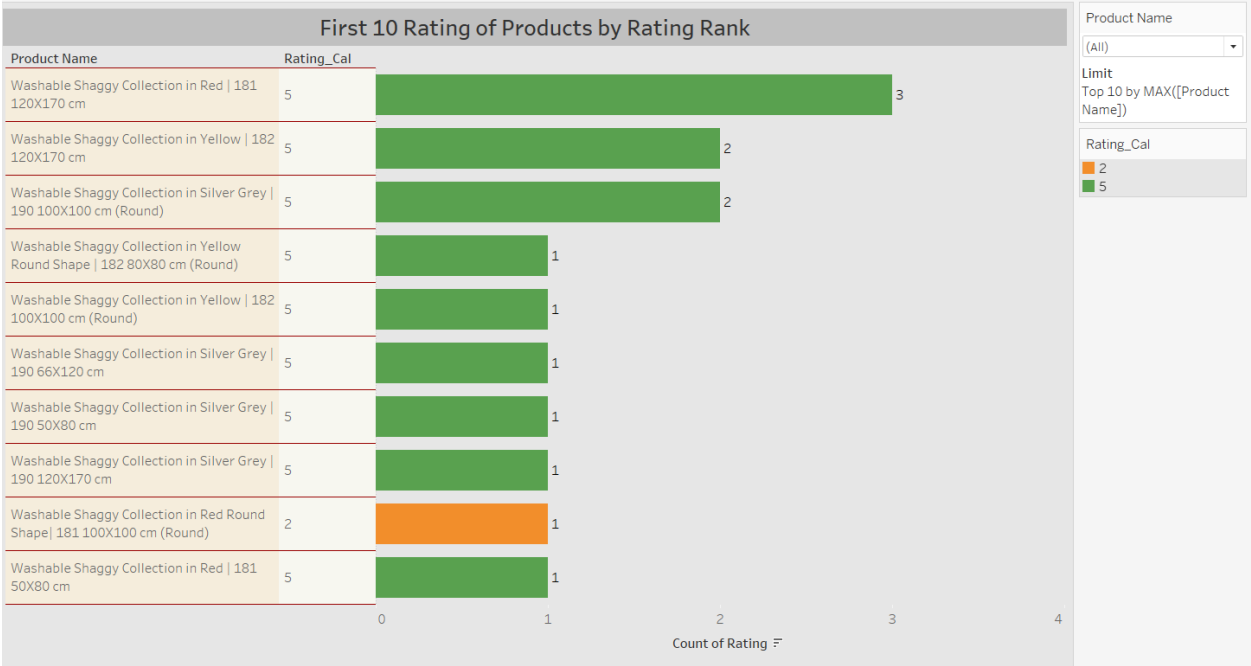
When the 10 product categories with the highest ratings are examined, it is seen that the “Soft Shaggy Collection” brand is the product with the highest rating (9) from the woocommers platform. When the whole analysis is examined, although the products with high ratings seem to be in the first place, in fact, it is seen that the products with a low rating are in the first place.



When the rating situation on the basis of countries is examined, the highest rating is seen in the UK. This is similar to the order numbers.



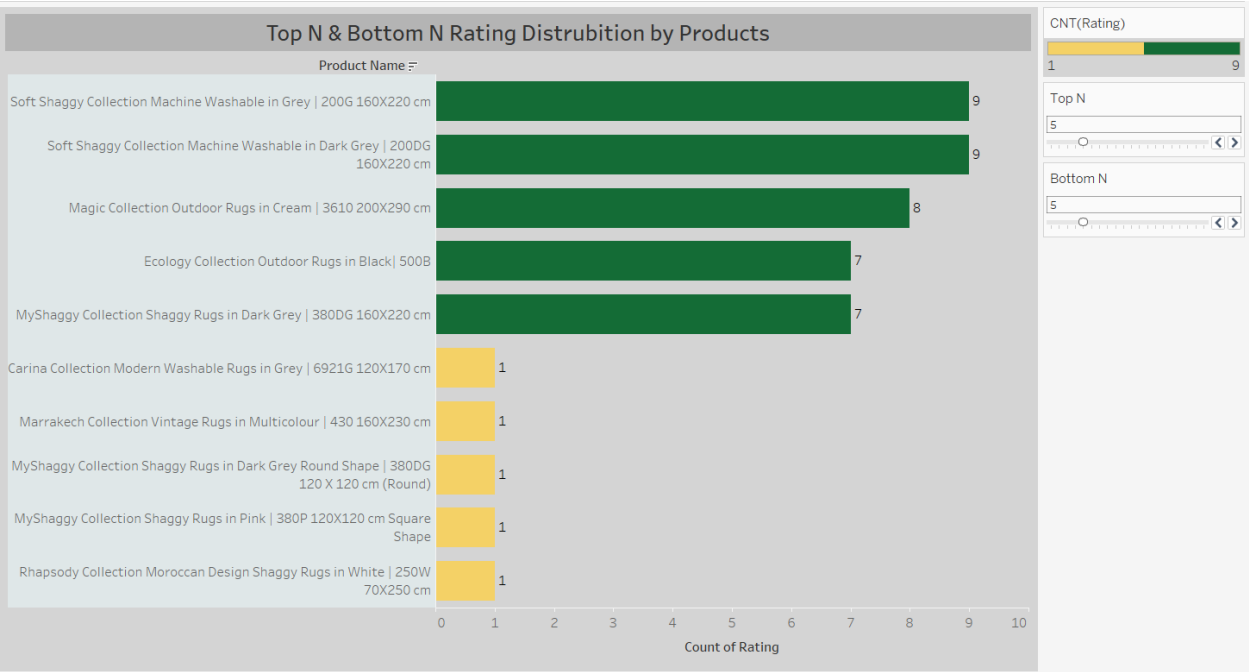
Considering the number of ratings according to years and location, it is seen that more ratings are received in 2022 than in other years. In addition, it is understood that the highest rating is received from United Kingdom (London).



When the rating status of the first 10 products according to their rating is examined; “Washable Shaggy Collection in Red | It is seen that the 181 120X170 cm” product has 3 ratings at the 5 level and is in the first place.

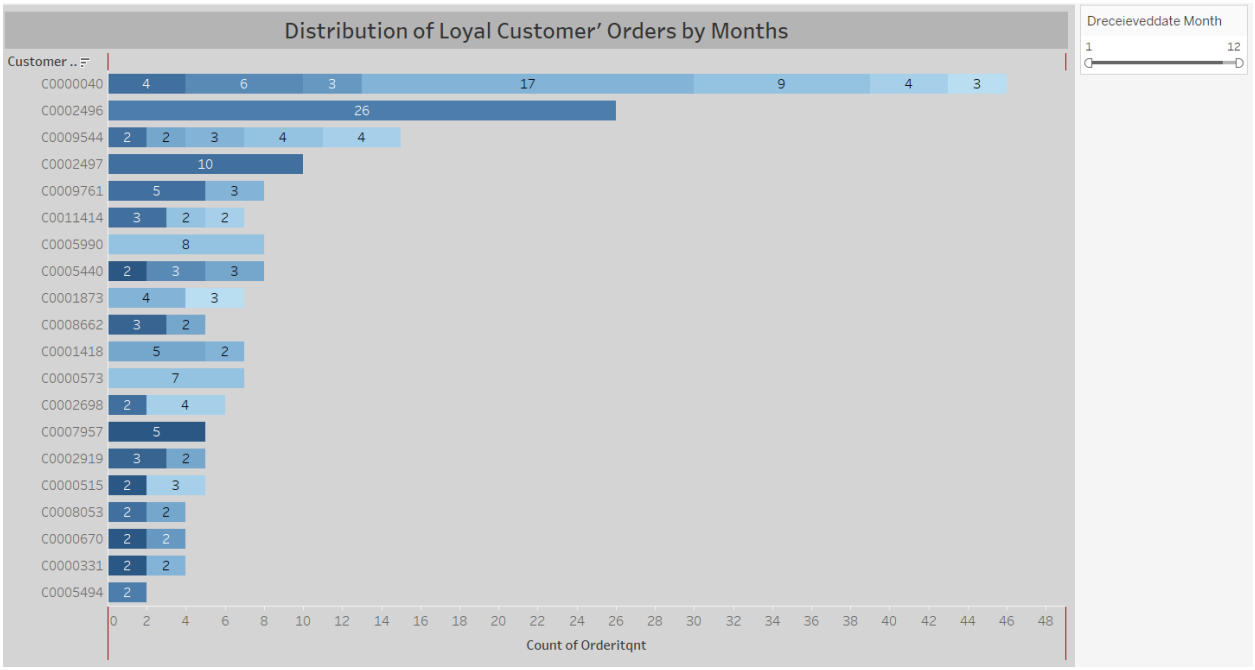
Again, “Washable Shaggy Collection in Red Round Shape| 181 100X100 cm (Round)” product, on the other hand, is seen to be in the top 10 product category, despite receiving a rating of 1 at the level of 2.

Although there is a rating of 2 for the product, it is seen that the order of the product is not affected much.



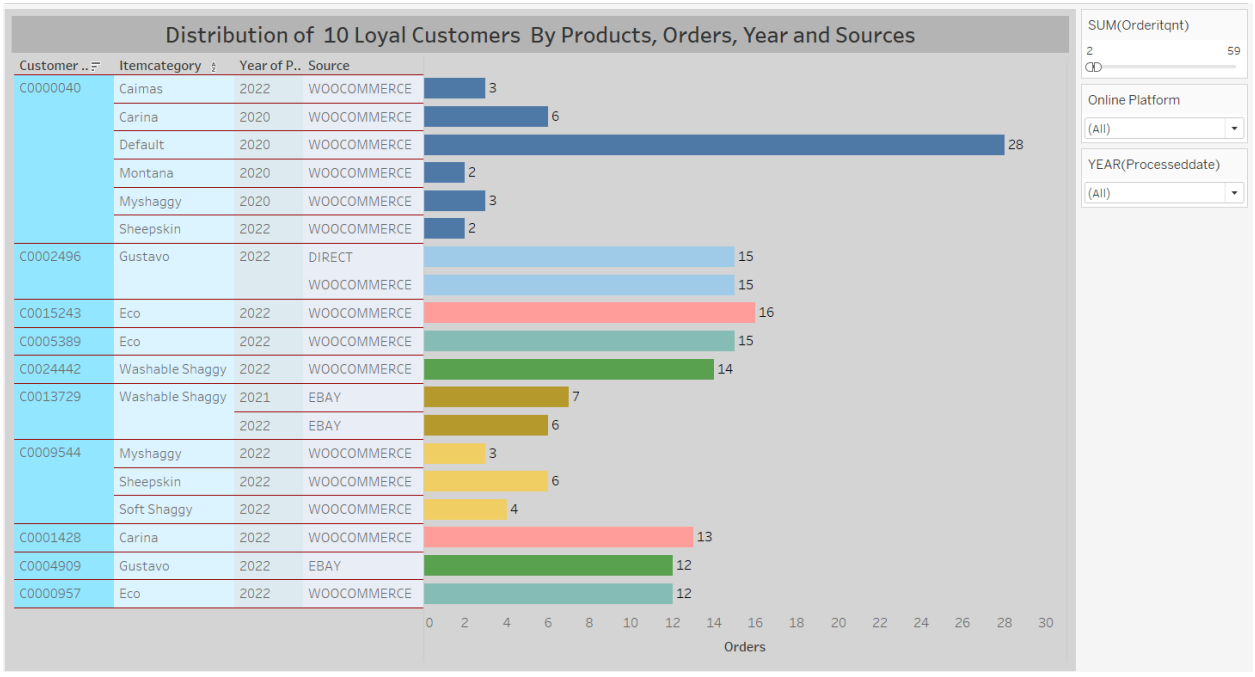
According to the rating status; when the first and last 5 products are examined, “Soft Shaggy Collection Machine Washable in Gray | 200G 160X220 cm” and “Soft Shaggy Collection Machine Washable in Dark Gray | It is seen that 200DG 160X220 cm” is the product with the highest rating, with 9 ratings.

“Rhapsody Collection Moroccan Design Shaggy Rugs in White | 250W 70X250 cm” and “MyShaggy Collection Shaggy Rugs in Pink | It is understood that 380P 120X120 cm Square Shape is the product with the least rating with 1 rating.



When the first 20 loyal customers are examined, it is seen that the first customer (C0000040) placed a total of 46 orders at 7 different times. It is understood that the second customer (C0002496) ordered 26 pieces at once. The reason why 1 person orders so much in a product category is not well understood.

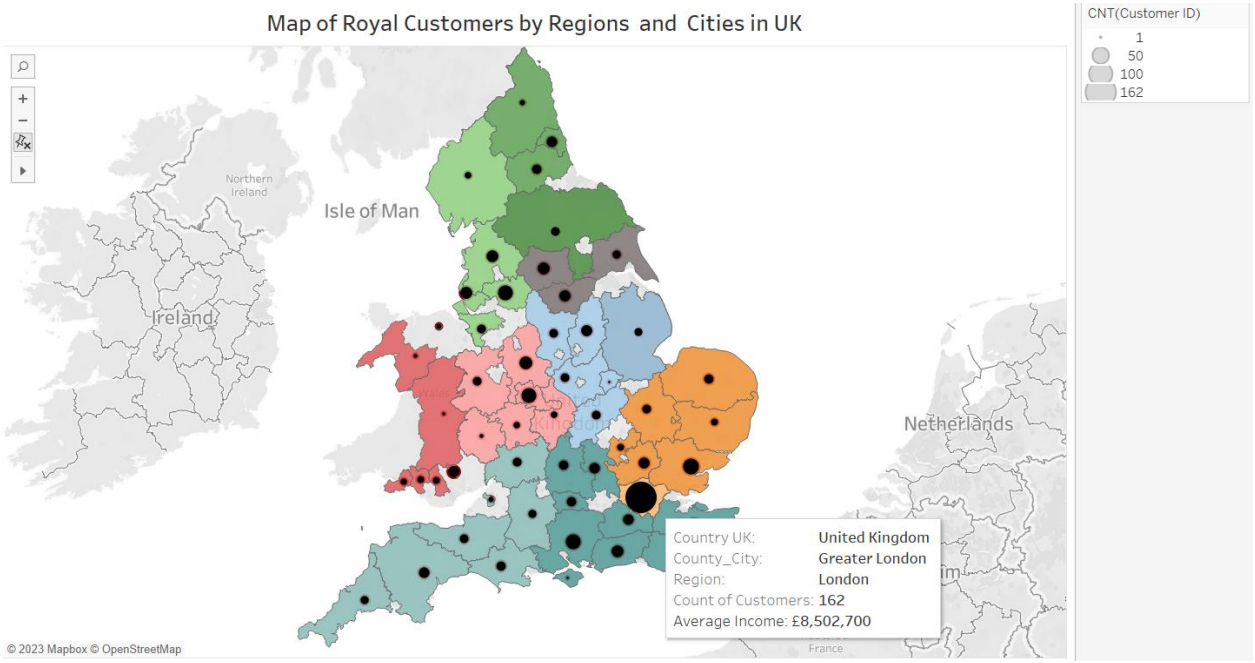
*Amazon customers are not included in the loyal customer list.



The first customer, in 2020 and 2022, using the woocommers platform; A total of 44 orders were placed in the “caimas, carina, myshaggy, montana, sheepskin and default” models. It is noteworthy here that he did not place an order in 2021. In addition, it is seen that the customer orders all these models from the same platform.

If the second customer (C0002496); It is seen that in 2022, 30 orders were placed in total from woocommerce and direct platforms.

It is understood that loyal customers place their orders from different platforms and different carpet models.



Looking at the number of loyal customers in the regions and cities on the map, it is seen that the most loyal customers are in London and then in the south east region.

In addition, when the relationship between the number of loyal customers and their average income is examined, it can be said that there is no difference.

Since the most loyal customers are in the Greater Londen and South Eash regions and the average income levels are similar, the firm may benefit from applying more advertising in these regions.

1.23.4 Clustering Of Customers

Under this title, we made analysis according to the tax amount which paid by customers. The main logic of this clustering analysis to divide customers after ‘Subtotal’ and ‘Tax’ amount.

At first the data set is prepared to analysis. We named it as ‘df_subtotal’.

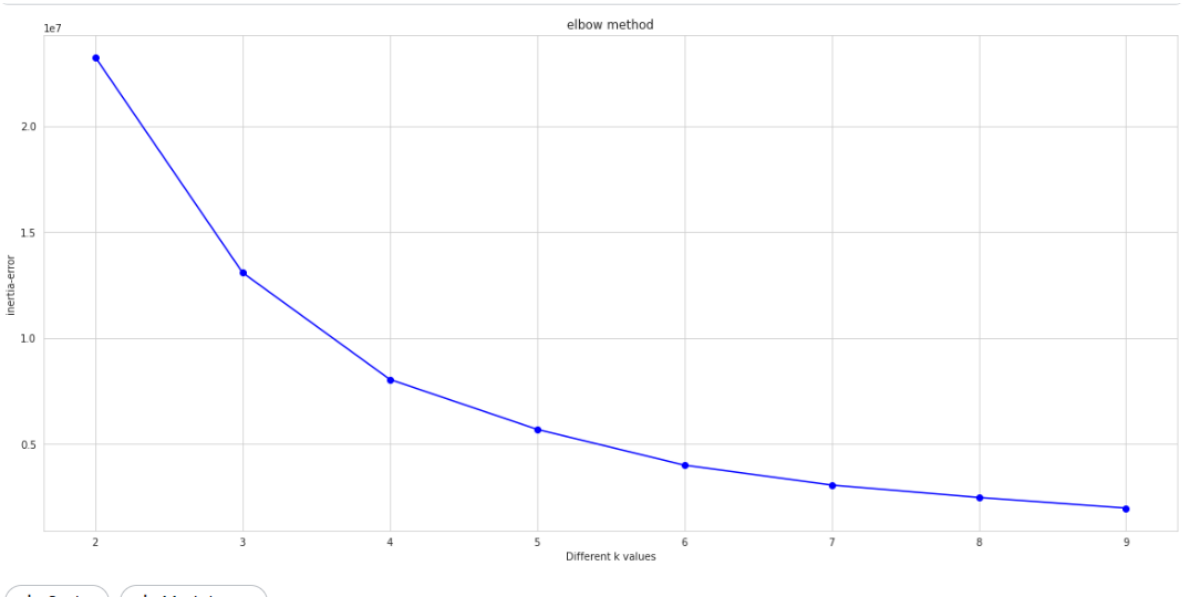
```
df2.set_index('CUSTOMER_ID',inplace=True)
df_subtotal = df2[['SUBTOTAL','TAX']]
```

In this step the target variable ‘Tax’ is dropped.

```
X = df_subtotal.drop('TAX',axis=1)
```

In this step, the k means model determines the best clustering value. At the end of the clustering the elbow is

```
from sklearn.cluster import KMeans
ssd = []
K = range(2,10)
for k in K:
    model = KMeans(n_clusters =k, random_state=42)
    model.fit(X)
    ssd.append(model.inertia_)
```

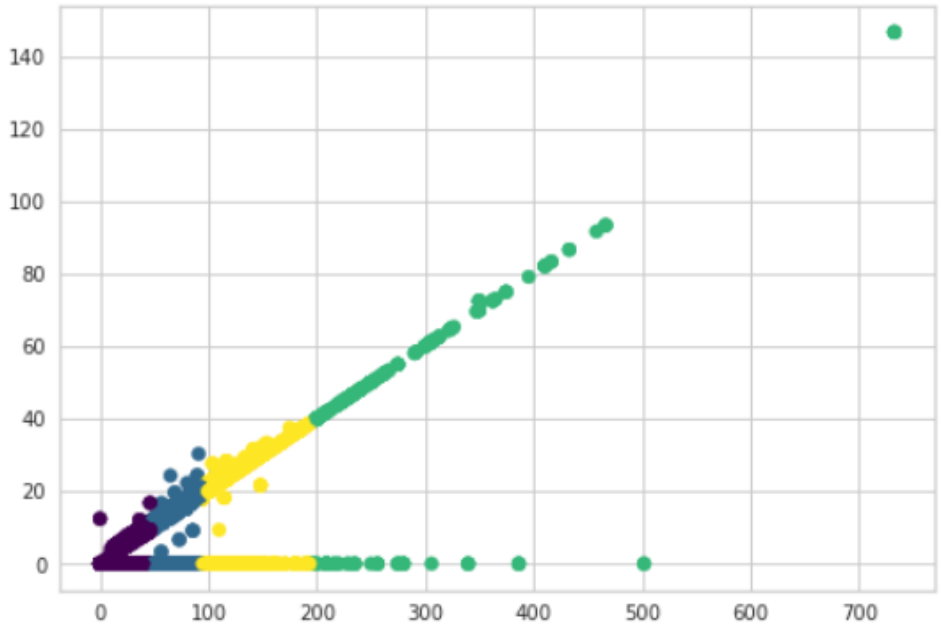


With this code below the model predicted the ‘Tax’ class of the customers.

```
from sklearn.cluster import KMeans

K_means_model = KMeans(n_clusters=4, random_state=42)
K_means_model.fit_predict(X)
```

With the code below and the graph it is seen that the more paid the customers that paid much more tax.

```
plt.scatter(df_subtotal["SUBTOTAL"],df_subtotal['TAX'],c =
df_subtotal.predicted_clusters,cmap='viridis');
```


140
120
100
80
60
40
20
0
0
100
200
300
400
500
600
700

1.23.5 COHORT ANALYSIS WITH PYTHON AND POWER BI

What is Cohort Analysis?

Cohort analysis is a kind of behavioral analytics that breaks the data in a data set into related groups before analysis. These groups, or cohorts, usually share common characteristics or experiences within a defined time-span. Cohort analysis allows a company to "see patterns clearly across the life-cycle of a customer (or user). By seeing these patterns of time, a company can adapt and tailor its service to those specific cohorts.

Types of Cohorts

Time-Based Cohorts

Time-based cohorts are customers who signed up for a product or service during a particular time frame. Analyzing these cohorts shows the customers' behavior depending on the time they started using a company's products or services. The time may be monthly or quarterly, depending on the sales cycle of a company.

Segment-Based Cohorts

It groups customers by the type of product or level of service they signed up for. Customers who signed up for basic level services might have different needs than those who signed up for advanced services. Understanding the needs of the various cohorts can help a company design tailor-made services or products for particular segments.

Size -Based Cohorts

Size-based cohorts refer to the various sizes of customers who purchase a company's products or services. The customers may be small and startup businesses, middle-sized businesses, and enterprise-level businesses. Comparing the different categories of customers based on their size reveals where the largest purchases come from.

Cohort Analysis with and without Amazon Sales

In this analysis, we will initially use the entire sales data. However, since it includes Amazon sales and represents all Amazon customers as one (C0000001), we will eliminate Amazon sales and customer "C0000001" in our second analysis.

Customers who have purchased multiple times:

```
In [9]: df['CUSTOMER_ID'].value_counts(dropna=False)
```

```
Out[9]: C0000001    54672
        C0000040       47
        C0002496       26
        C0009544       16
        C0002497       10
        C0001873        9
        C0005494        8
        C0009761        8
        C0002698        8
        C0005440        8
        C0008662        8
        C0002919        8
        C0011414        8
        C0008053        8
        C0000331        8
        C0007957        8
        C0005990        8
        C0001418        7
        C0000573        7
```

1.23.6 Insights:

- Out of 92249 records, 54672 orders come from customer "C0000001" representing Amazon customers.

- The second best customer is "C0000040". But when we check the customer "C0000040" from the original Sales data, almost half of the company columns have "Student". So, it is a group of people but not a specific customer.

Company	Customer ID
	C0000040
	C0000040
Student	C0000040
Student	C0000040
Student	C0000040
Student	C0000040
Student	C0000040
Student	C0000040
Student	C0000040
Student	C0000040
Student	C0000040
Student	C0000040
	C0000040
	C0000040
	C0000040
	C0000040
Student	C0000040
Student	C0000040
Student	C0000040
Student	C0000040
Student	C0000040
	C0000040
Student	C0000040
Student	C0000040
	C0000040
	C0000040

Our first customers (in November 2019):

```
In [12]: df[(df['DRECEIEVEDDATE_YEAR']==2019) & (df['DRECEIEVEDDATE_MONTH']==11)][['CUSTOMER_ID','DRECEIEVEDDATE']]
```

Out[12]:

	CUSTOMER_ID	DRECEIEVEDDATE
519	C0000001	2019-11-12
520	C0000001	2019-11-12
740	C0000001	2019-11-12
2950	C0001525	2019-11-12
15086	C0005444	2019-11-12
37745	C0000001	2019-11-12

1.23.7 Insights:

- In the beginning, we had only three different customers: “C0000001” , “C0001525 ”, “C0005444 ” and one of them represents Amazon customers.

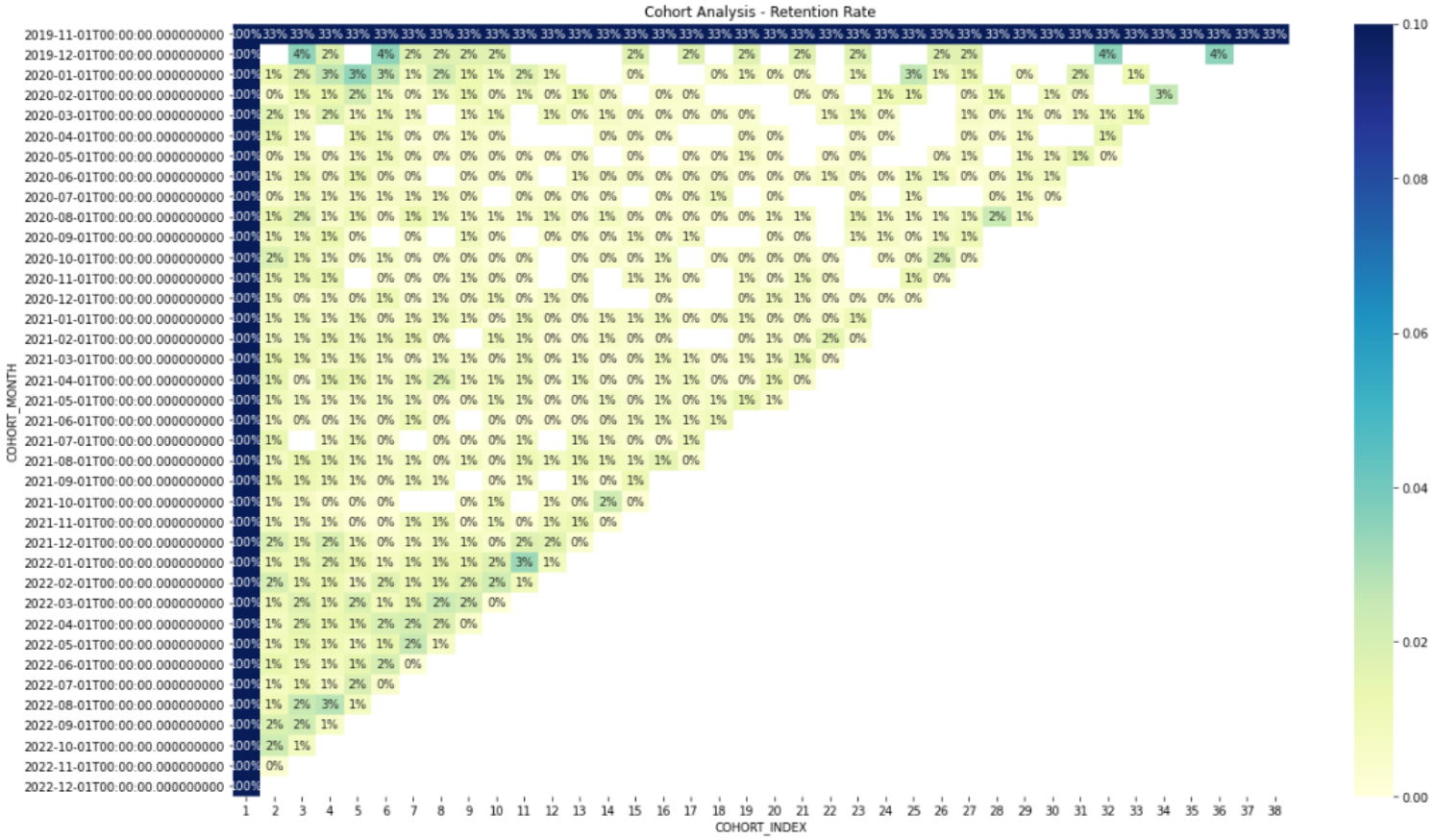
- As seen above, there are a few Amazon records in the first month. So, if we exclude Amazon sales, we will lose only one customer (C0000001) in Cohort Analysis in November 2019. Other months will not be affected because the first record for each customer is included in this analysis. Apart from that, Cohort Analysis without Amazon Sales will not make any difference and will resemble the tables and graphs below.

Create a Cohort Index for 38 months to see customer life-cycle

COHORT_INDEX	0	1	2	3	4	5	6	7	8	9	...	28	29	30	31	32	33	34	35	36	37
COHORT_MONTH																					
2019-11-01	3.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2019-12-01	57.0	NaN	2.0	1.0	NaN	2.0	1.0	1.0	1.0	1.0	...	NaN	NaN	NaN	2.0	NaN	NaN	NaN	2.0	NaN	NaN
2020-01-01	233.0	3.0	4.0	6.0	8.0	6.0	2.0	5.0	3.0	2.0	...	1.0	NaN	4.0	NaN	3.0	NaN	NaN	NaN	NaN	NaN
2020-02-01	233.0	1.0	3.0	3.0	5.0	2.0	1.0	2.0	3.0	1.0	...	NaN	2.0	1.0	NaN	NaN	6.0	NaN	NaN	NaN	NaN
2020-03-01	310.0	5.0	2.0	5.0	2.0	3.0	2.0	NaN	2.0	3.0	...	2.0	1.0	2.0	3.0	4.0	NaN	NaN	NaN	NaN	NaN
2020-04-01	499.0	7.0	3.0	NaN	6.0	5.0	1.0	1.0	3.0	2.0	...	3.0	NaN	NaN	7.0	NaN	NaN	NaN	NaN	NaN	NaN
2020-05-01	612.0	3.0	4.0	2.0	4.0	6.0	3.0	1.0	3.0	2.0	...	4.0	4.0	9.0	1.0	NaN	NaN	NaN	NaN	NaN	NaN
2020-06-01	610.0	4.0	7.0	3.0	8.0	3.0	2.0	NaN	2.0	2.0	...	5.0	5.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Insights:

- As seen above, the company had only 3 customers in the beginning and it could achieve to retain only one customer among them, namely, Amazon customers (C0000001). In other words, other customers were one-time customers and left the company in the following month. So, the retention rate is 33 % and the churn rate is 66 % for November 2019 customers.

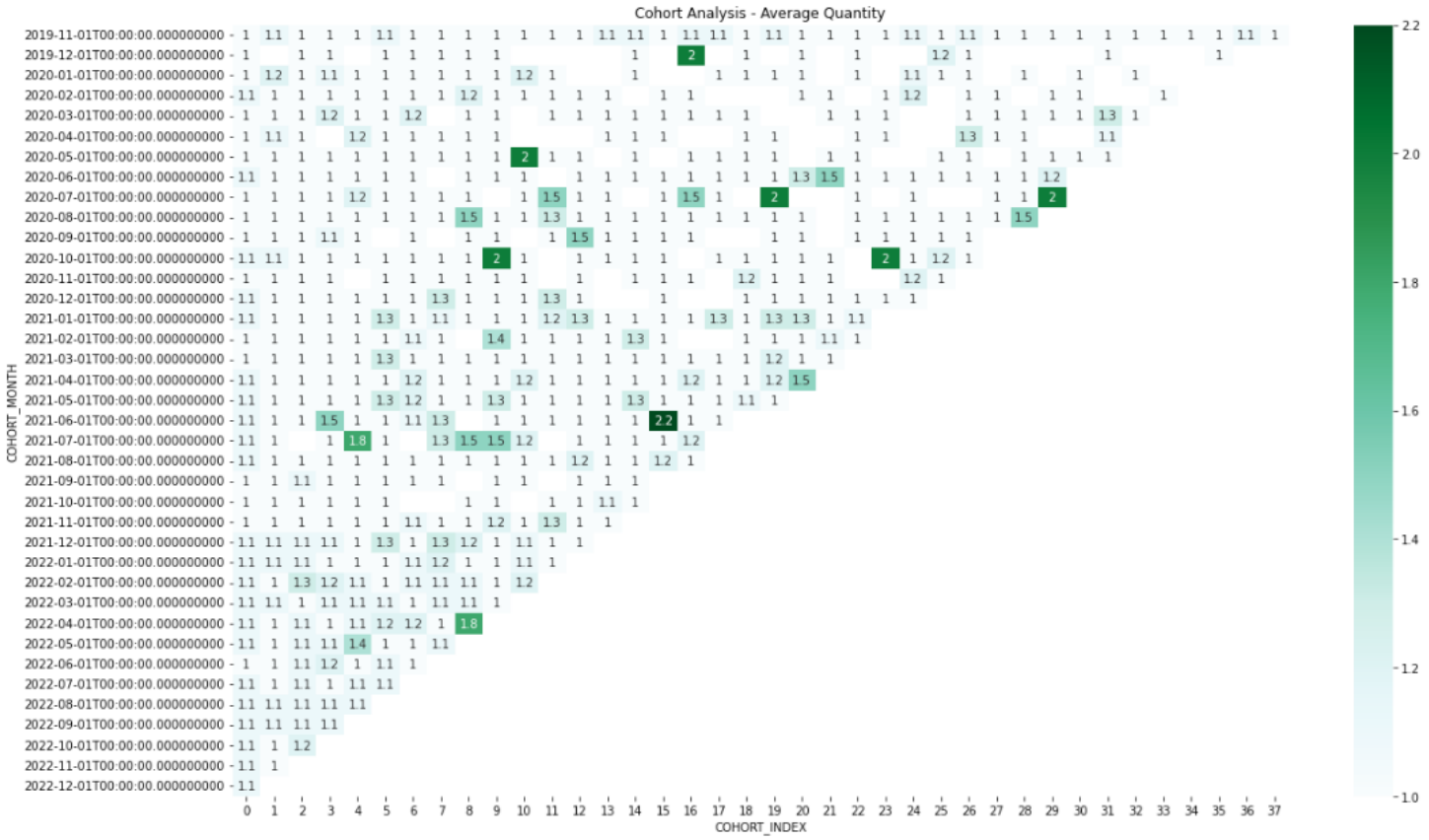


Insights:

- Retention rates in general are very low regardless of month and year. Rarely, 3% and 4% of our customers continue shopping. The majority of these are customers who bought products from the company in December 2019. The customers who met the company during this period seem to be the most loyal customers.

- One reason why the company has low retention rates or high churn rates is the lack of variety of products sold by the company since it is a rug company. Also, they sell a product (rugs and carpets) that customers can use for many years when they buy them. So it takes a long time for a customer to return. As a result, Cohort Analysis does not give valuable insights for the company unless it diversifies the products it sells.

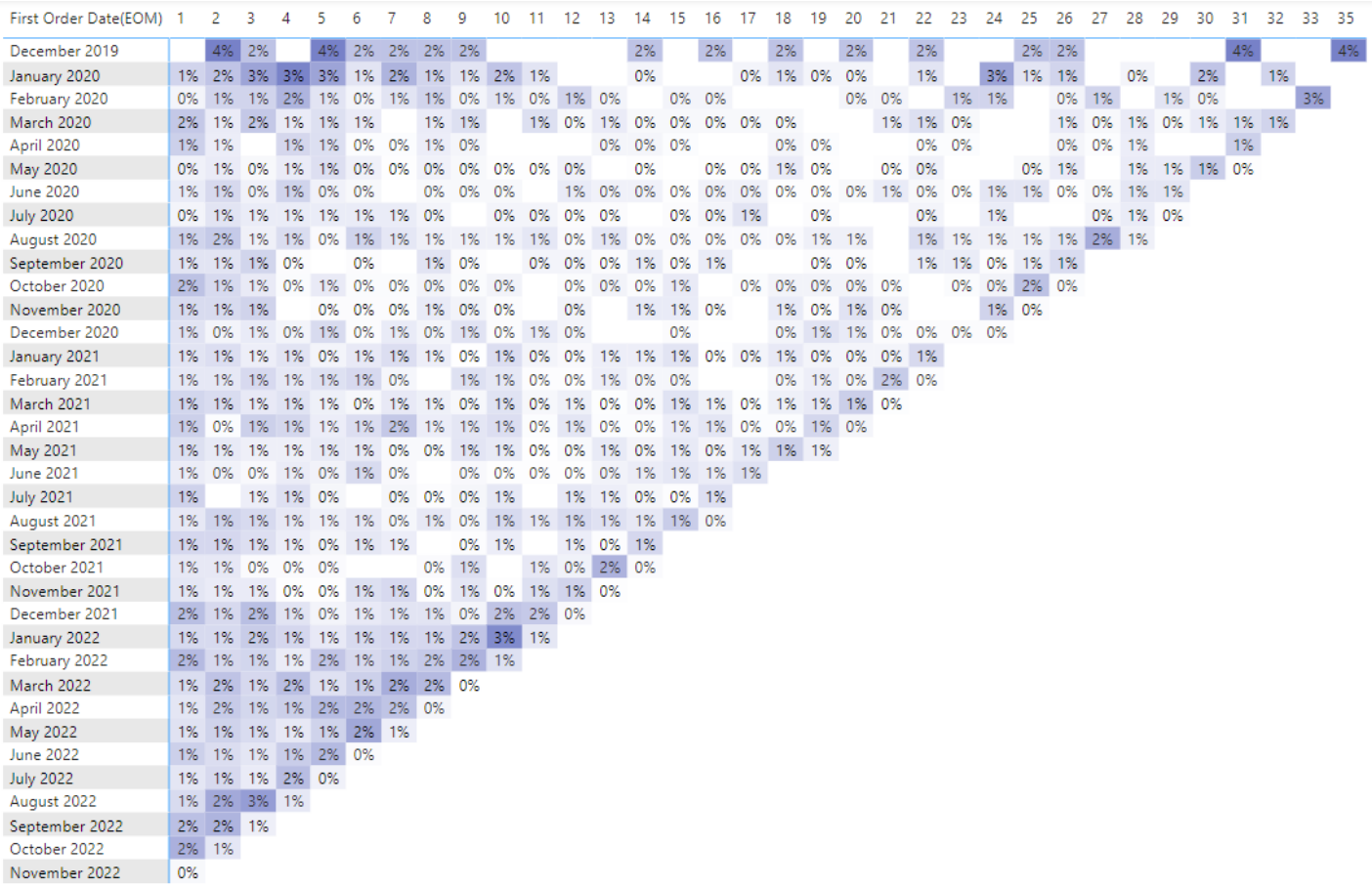
Average Quantity Sold by These Customers



Insights:

- The graph above reveals that loyal customers buy between 1 - 2.2 items on average. The reason is that a customer can buy a limited number of rugs and use them for a long time.

1.23.8 Cohort Analysis without Amazon Sales by using Power BI



First Order Date(EOM)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	35
December 2019		4%	2%		4%	2%	2%	2%	2%					2%		2%		2%		2%		2%			2%	2%					4%			4%
January 2020	1%	2%	3%	3%	3%	1%	2%	1%	1%	2%	1%			0%			0%	1%	0%	0%		1%		3%	1%	1%		0%		2%		1%		
February 2020	0%	1%	1%	2%	1%	0%	1%	1%	0%	1%	0%	1%	0%		0%	0%				0%	0%		1%	1%		0%	1%		1%	0%			3%	
March 2020	2%	1%	2%	1%	1%	1%		1%	1%		1%	0%	1%	0%	0%	0%	0%	0%			1%	1%	0%			1%	0%	1%	0%	1%	1%	1%		
April 2020	1%	1%		1%	1%	0%	0%	1%	0%				0%	0%	0%			0%	0%			0%	0%			0%	0%	1%			1%			
May 2020	0%	1%	0%	1%	1%	0%	0%	0%	0%	0%	0%	0%		0%		0%	0%	1%	0%		0%	0%			0%	1%		1%	1%	1%	0%			
June 2020	1%	1%	0%	1%	0%	0%		0%	0%	0%		1%	0%	0%	0%	0%	0%	0%	0%	0%	1%	0%	0%	1%	1%	0%	0%	1%	1%					
July 2020	0%	1%	1%	1%	1%	1%	1%	0%		0%	0%	0%	0%		0%	0%	1%		0%			0%		1%			0%	1%	0%					
August 2020	1%	2%	1%	1%	0%	1%	1%	1%	1%	1%	1%	0%	1%	0%	0%	0%	0%	0%	1%	1%		1%	1%	1%	1%	1%	2%	1%						
September 2020	1%	1%	1%	0%		0%		1%	0%		0%	0%	0%	1%	0%	1%			0%	0%		1%	1%	0%	1%	1%								
October 2020	2%	1%	1%	0%	1%	0%	0%	0%	0%	0%		0%	0%	0%	1%		0%	0%	0%	0%	0%		0%	0%	2%	0%								
November 2020	1%	1%	1%		0%	0%	0%	1%	0%	0%		0%		1%	1%	0%		1%	0%	1%	0%			1%	0%									
December 2020	1%	0%	1%	0%	1%	0%	1%	0%	1%	0%	1%	0%			0%			0%	1%	1%	0%	0%	0%	0%										
January 2021	1%	1%	1%	1%	0%	1%	1%	1%	0%	1%	0%	0%	1%	1%	1%	0%	0%	1%	0%	0%	0%	1%												
February 2021	1%	1%	1%	1%	1%	1%	0%		1%	1%	0%	0%	1%	0%	0%			0%	1%	0%	2%	0%												
March 2021	1%	1%	1%	1%	1%	0%	1%	1%	0%	1%	0%	1%	0%	0%	1%	1%	0%	1%	1%	1%	0%													
April 2021	1%	0%	1%	1%	1%	1%	2%	1%	1%	1%	0%	1%	0%	0%	1%	1%	0%	0%	1%	0%														
May 2021	1%	1%	1%	1%	1%	1%	0%	0%	1%	1%	0%	0%	1%	0%	1%	0%	1%	1%	1%															
June 2021	1%	0%	0%	1%	0%	1%	0%		0%	0%	0%	0%	0%	1%	1%	1%	1%																	
July 2021	1%		1%	1%	0%		0%	0%	0%	1%		1%	1%	0%	0%	1%																		
August 2021	1%	1%	1%	1%	1%	1%	0%	1%	0%	1%	1%	1%	1%	1%	1%	0%																		
September 2021	1%	1%	1%	1%	0%	1%	1%		0%	1%		1%	0%	1%																				
October 2021	1%	1%	0%	0%	0%			0%	1%		1%	0%	2%	0%																				
November 2021	1%	1%	1%	0%	0%	1%	1%	0%	1%	0%	1%	1%	0%																					
December 2021	2%	1%	2%	1%	0%	1%	1%	1%	0%	2%	2%	0%																						
January 2022	1%	1%	2%	1%	1%	1%	1%	1%	2%	3%	1%																							
February 2022	2%	1%	1%	1%	2%	1%	1%	2%	2%	1%																								
March 2022	1%	2%	1%	2%	1%	1%	2%	2%	0%																									
April 2022	1%	2%	1%	1%	2%	2%	2%	0%																										
May 2022	1%	1%	1%	1%	1%	2%	1%																											
June 2022	1%	1%	1%	1%	2%	0%																												
July 2022	1%	1%	1%	2%	0%																													
August 2022	1%	2%	3%	1%																														
September 2022	2%	2%	1%																															
October 2022	2%	1%																																
November 2022	0%																																	

Insights:

- When we exclude Amazon Sales and the first month the customers buy for the first time in our company, we see that November 2019 customers disappear. The reason is that no other customer continues shopping except Amazon customers.
- When we check the months for above 1 %, we see that customers return mostly around November. Campaigns before Christmas and Black Friday may cause to return customers during this time of the year. The cell before the last one for each month represents November and we can detect a cross line for November across the graph.
- Lastly, December 2019 and January 2020 customers seem the most loyal customers.

Word Cloud

Expressions containing unnecessary characters and numbers in product comments have been deleted.

```
for i in range(len(yorum)):
    duzenle = re.sub('[^a-zA-Z]',' ',yorum['comments'][i])
    duzenle = duzenle.lower()
    duzenle = duzenle.split()
    duzenle = [lema.lemmatize(kelime) for kelime in duzenle if not kelime in set(stopwords.words('english'))]
    duzenle = ' '.join(duzenle)

    temiz.append(duzenle)

print(temiz)
```

```
['happy', 'nan', 'quality meet expectation', 'picture crease came quite quickly', 'beautiful rug', 'nan', 'pretty cos
y picture fit place soo well happy', 'rug look fresh clean look lovely dining room', 'nan', 'lovely rug great quality
good price', 'nan', 'nan', 'delivery quick cost effective', 'speed delivery product cost effective', 'nan', 'pleasant
ly surprised quality rug purchased nice rug fast delivery update', 'nan', 'simply beautiful', 'excellent product look
great flag extension', 'stunning rug exactly looking', 'great rug perfect wanted go', 'really happy rug excellent qua
lity money excellent delivery time', 'nan', 'lovely rug good quality', 'poor quality really disappointed sent straigh
t back', 'pretty fit well chaotic home office look online pleased washable great value may buy couple room house dog
find way', 'nan', 'nan', 'look nice slip decking hold water decking underneath remains wet unless move rug', 'nan',
'nan', 'bought living room lovely especially dog easy enough throw washing mashine well dog love eating biscuit', 're
ally happy rug ordered yesterday arrived today dhl dhl kept informed way gone smooothly excellent value money rug pur
pose intended colour true one shown website maybe slightly darker overall happy customer thank service', 'nan', 'good
value money', 'im happy order rug look amazing living room', 'lovely rug good quality', 'lovely product great quality
young child long something got spilled easy clean damage rug would highly recommend', 'pleased rug ordered bargain so
ft overall great quality', 'lighter expected problem love love washed great value rug', 'nice little rug used front f
ireplace stove wanted rug show dirt great value', 'shipping communication arriving easy quick faultless start finish
came rolled folded half sceptical would unfold creased middle surprise unrolled immediately hoovered fluff come flat
packaging transport hoovered became fluffy instantly left hour settle came home bump creased flattened completely cor
ner fluffy underfoot day wish could leave photo review look feel soft perfect u great month old play shaggy push toy
car along soft enough newborn lay bargain totally worth paying especially x cm buy rug future wish bought sooner', 'n
```

```
yorum1 = pd.DataFrame(list(zip(yorum['comments'],temiz)),columns = ["OrijinalYorum", "TemizYorum"])
yorum1
```

	OrijinalYorum	TemizYorum
0	Very happy with thisğŸ˜Š	happy
1	nan	nan
2	Quality meets expectation	quality meet expectation
3	Just as the picture, creases came out quite qu...	picture crease came quite quickly
4	Beautiful Rug!!	beautiful rug
...	...	...
556	Absolutely lovely at extremely low price A* hi...	absolutely lovely extremely low price highly r...

Figure 1.1.

The number of times each word was used was checked.

```
print(frekans)
kelimeler = dict(frekans.values)
print(kelimeler)
```

```
        Kelimeler  Frekans
0           happy     45.0
2         quality    143.0
3            meet      2.0
4     expectation      2.0
5         picture     18.0
...           ...      ...
1167      updated      1.0
1168      reflect      1.0
1169   reflective      1.0
1170         luxe      1.0
1171       modern      1.0

[1170 rows x 2 columns]
{'happy': 45.0, 'quality': 143.0, 'meet': 2.0, 'expectation': 2.0, 'picture': 18.0, 'crease': 3.0, 'came': 8.0, 'quit
e': 11.0, 'quickly': 18.0, 'beautiful': 38.0, 'rug': 342.0, 'pretty': 7.0, 'cosy': 1.0, 'fit': 12.0, 'place': 5.0, 's
oo': 2.0, 'well': 33.0, 'look': 113.0, 'fresh': 1.0, 'clean': 13.0, 'lovely': 80.0, 'dining': 1.0, 'room': 45.0, 'gre
at': 135.0, 'good': 97.0, 'price': 78.0, 'delivery': 73.0, 'quick': 28.0, 'cost': 4.0, 'effective': 3.0, 'speed': 3.
0, 'product': 27.0, 'pleasantly': 4.0, 'surprised': 5.0, 'purchased': 9.0, 'nice': 58.0, 'fast': 17.0, 'update': 1.0,
```

A word cloud was created using a mask. A speech bubble was preferred because the comments were visualised.

```
resim = np.array(Image.open("comment.png"))

plt.figure(figsize=(15,15))
cloud = WordCloud(background_color='White', colormap='turbo_r', mask=resim, contour_width=1, contour_color =
plt.imshow(cloud)
plt.show()
```

The words used in the comments are mostly words with positive connotations. This is an indication that customers are satisfied with the products they buy.

1.23.9 Customer Segmentation by RFM

One of the most popular, easy-to-use, and effective segmentation methods to enable marketers to analyze customer behavior is RFM analysis. RFM stands for Recency, Frequency, and Monetary value, each corresponding to some key customer trait. These RFM metrics are important indicators of a customer's behavior because frequency and monetary value affects a customer's lifetime value, and recency affects retention, a measure of engagement.

RFM factors illustrate these facts:

- the more recent the purchase, the more responsive the customer is to promotions
- the more frequently the customer buys, the more engaged and satisfied they are
- monetary value differentiates heavy spenders from low-value purchasers

RFM values were calculated for each customer.

```
RFM = r.merge(f, on='customer_id').merge(m, on='customer_id')
RFM = RFM.reset_index()
RFM = RFM.rename(columns={'customer_id':'Customer','dreceievedda
RFM
```

	Customer	Recency	Frequency	Monetary
0	C0000002	727	1	99.60
1	C0000003	935	1	15.90
2	C0000004	919	1	15.90
3	C0000005	961	1	13.90
4	C0000006	932	1	13.52
...	...	...	...	...
31425	C0032495	104	1	69.95
31426	C0032496	239	1	139.90
31427	C0032497	280	1	111.92
31428	C0032498	55	1	69.95
31429	UNKNOWN	417	2	63.94

31430 rows × 4 columns

Frequency and **Recency** is low because the products sold by the company are products that are used for a long time, such as carpets. For example, you can't expect a customer to purchase a rug on a monthly basis. In this case, a marketer could give more weight to **Monetary** and **Recency** aspects rather than. Therefore, we will not use **Frequency** scores when calculating the RFM score. Instead, customers with a Frequency value greater than 1 were included in the analysis.

```
RFM[(RFM.Frequency > 1)]
```

	Customer	Recency	Frequency	Monetary
17	C0000019	737	3	217.98
19	C0000021	897	2	68.01
29	C0000031	155	2	113.83
31	C0000033	19	2	70.80
38	C0000040	88	6	297.38
...	...	...	...	...
30811	C0031877	1	2	149.75
30892	C0031960	147	2	159.84
30977	C0032045	535	2	159.80
31397	C0032467	428	2	59.80
31429	UNKNOWN	417	2	63.94

2755 rows × 4 columns

```
RFM[RFM["Frequency"]>=2]
```

	Customer	Recency	Frequency	Monetary	Labels	Recency_score	Frequency_score	Monetary_score	RFM_SCORE
17	C0000019	737	3	217.98	1	1	5	5	15
19	C0000021	897	2	68.01	2	1	5	4	14
29	C0000031	155	2	113.83	0	3	5	5	35
31	C0000033	19	2	70.80	0	5	5	4	54
38	C0000040	88	6	297.38	0	4	5	5	45
...	...	...	...	...	...	...	...	...	...
30811	C0031877	1	2	149.75	0	5	5	5	55
30892	C0031960	147	2	159.84	0	3	5	5	35
30977	C0032045	535	2	159.80	1	2	5	5	25
31397	C0032467	428	2	59.80	3	2	5	3	23
31429	UNKNOWN	417	2	63.94	3	2	5	4	24

2755 rows × 9 columns

According to RFM scores, customers were divided into 10 different segments.

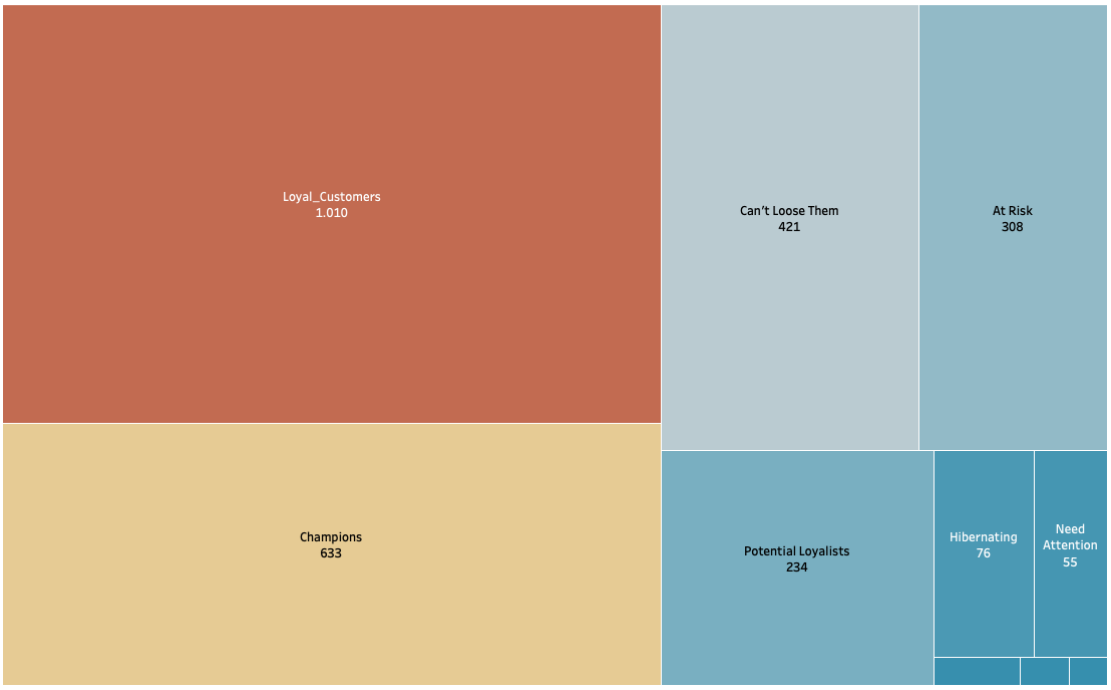
These are can’t loose them, loyal customers, champions, at risk, need attention, potential loyalists, hibernating, about to sleep, promising and new customers.

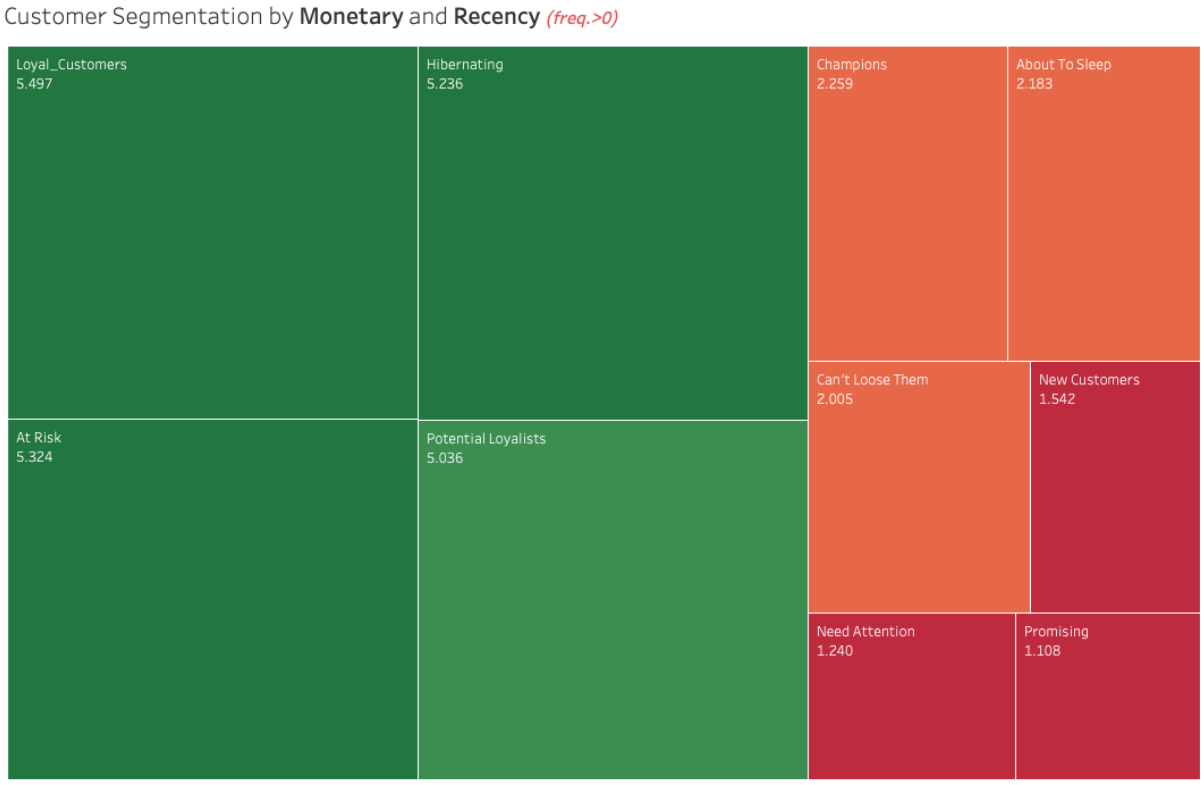
```
# RFM segmentlerinin oluşturulması
seg_map = {
    r'[1-2][1-2]': 'Hibernating',
    r'[1-2][3-4]': 'At Risk',
    r'[1-2]5': "Can't Loose Them",
    r'3[1-2]': 'About To Sleep',
    r'33': 'Need Attention',
    r'[3-4][4-5]': 'Loyal_Customers',
    r'41': 'Promising',
    r'51': 'New Customers',
    r'[4-5][2-3]': 'Potential Loyalists',
    r'5[4-5]': 'Champions'
}

# RFM skorlarını isimlendirelim
LyRFM['Segment'] = LyRFM['RFM_SCORE'].replace(seg_map, regex=True)

LyRFM.sample(11)
```

```
/var/folders/zq/ly5y6s0n65j9c368qkgnz3780000gn/T/ipykernel_9526/4010685409.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  LyRFM['Segment'] = LyRFM['RFM_SCORE'].replace(seg_map, regex=True)
```

	Customer	Recency	Frequency	Monetary	Labels	Recency_score	Frequency_score	Monetary_score	RFM_SCORE	Segment
3997	C0004786	128	2	82.36	0	4	5	4	44	Loyal_Customers
7919	C0008760	80	2	154.80	0	4	5	5	45	Loyal_Customers
211	C0000218	560	2	299.80	1	2	5	5	25	Can't Loose Them
1149	C0001917	8	2	115.33	0	5	5	5	55	Champions
7295	C0008130	61	4	154.41	0	4	5	5	45	Loyal_Customers
12674	C0013585	13	2	129.89	0	5	5	5	55	Champions
5420	C0006231	14	2	63.66	0	5	5	4	54	Champions
9727	C0010585	74	2	84.85	0	4	5	4	44	Loyal_Customers
2714	C0003491	27	2	568.07	0	5	5	5	55	Champions
2251	C0003025	620	2	181.45	1	2	5	5	25	Can't Loose Them
26092	C0027133	144	2	119.80	0	4	5	5	45	Loyal_Customers

Customer Segmentation by **Monetary** and **Recency** *(freq.>1)*





All customers are included in the final picture.

1.23.10Recommendations

- Different complementary products that can be sold with carpets can be added to the product list.
- For customer segmentation, information such as age, gender and zip code can be obtained from a customer.
- It is suggested that the location of the 'Subscribe' button can be placed at the top of the website page.
- Since most loyal customers are in the Greater London and South East region and because of the similar average income; The company can apply more advertisements in these regions.

The recommendations for the customers are composed as follows:

- **Champions** are your best customers, who bought most recently, most often, and are heavy spenders. Reward these customers. They can become early adopters for new products and will help promote your brand.
- **Potential Loyalists** are your recent customers with average frequency and who spent a good amount. Offer membership or loyalty programs or recommend related products to upsell them and help them become your Loyalists or Champions.
- **New Customers** are your customers who have a high overall RFM score but are not frequent shoppers. Start building relationships with these customers by providing onboarding support and special offers to increase their visits.
- **At Risk Customers** are your customers who purchased often and spent big amounts, but haven't purchased recently. Send them personalized reactivation campaigns to reconnect, and offer renewals and helpful products to encourage another purchase.
- **Can't Lose Them** are customers who used to visit and purchase quite often, but haven't been visiting recently. Bring them back with relevant promotions, and run surveys to find out what went wrong and avoid losing them to a competitor.