Final Report
Springboard Capstone 3
By Rime Saad

Table of Contents

Introduction:

This is a transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail.The company mainly sells unique all-occasion gifts and many customers of the company are wholesalers. The purpose of this project is to clean and study the data, create customer cohorts and perform customer segmentation as well as predict sales for the company.

This data set has been acquired from the UCI Machine Learning Repository and was made available by Dr Daqing Chen.

Data Wrangling:

An excel file was downloaded from the UCI Machine Learning site that has two sheets, the first has 2010 sales and the second 2011. The file was read using the panda's read_csv.

The data consisted of 1,067,371 rows and 8 columns. The columns included:

- InvoiceNumber: unique number for each customer transaction
- Stockcode: unique number for each product
- Description: product description
- Quantity
- Price
- InvoiceDate
- Customer ID
- Country

There are 5,835 customers and the store sells 4,615 products. There were also 243,007 missing Customer ID and some description values.

Countries:

Most of the sales were for customers in the UK, which is expected since the company is based in the UK but the company sells to 41 other countries.

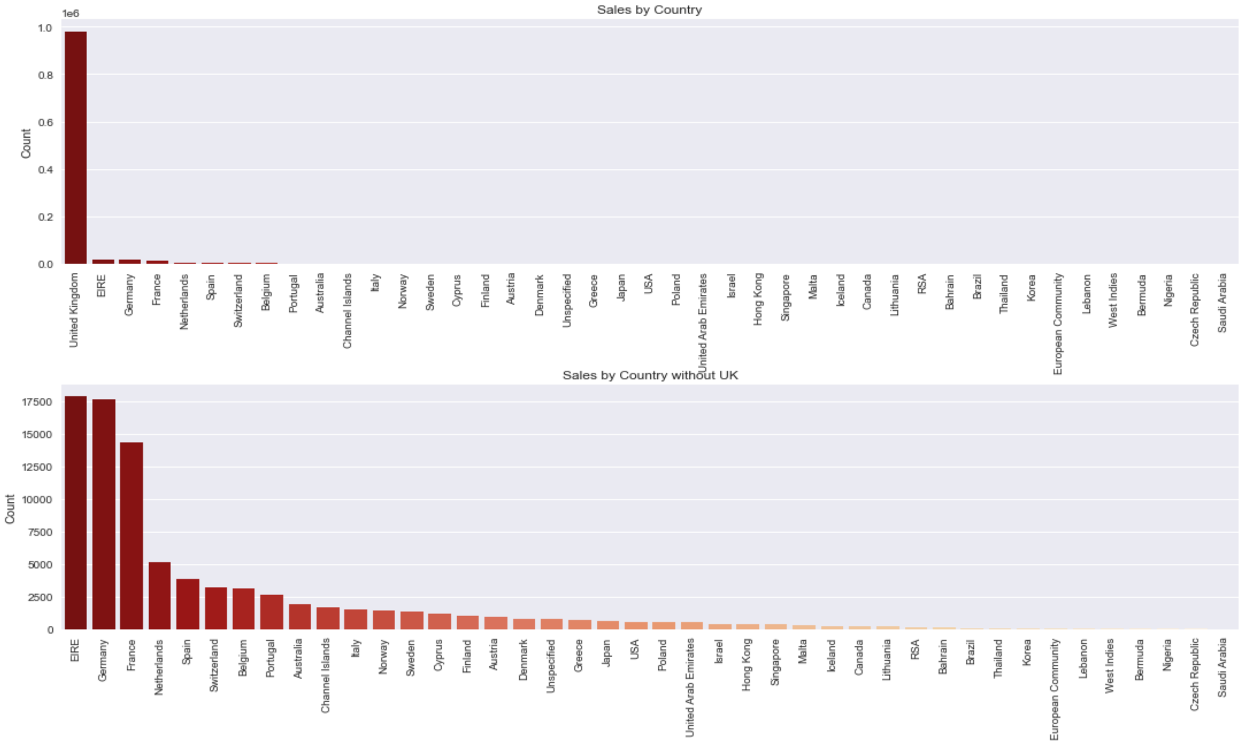


Figure 1: a. Sales by country, b. Sales by country without the UK

Price and Quantity:

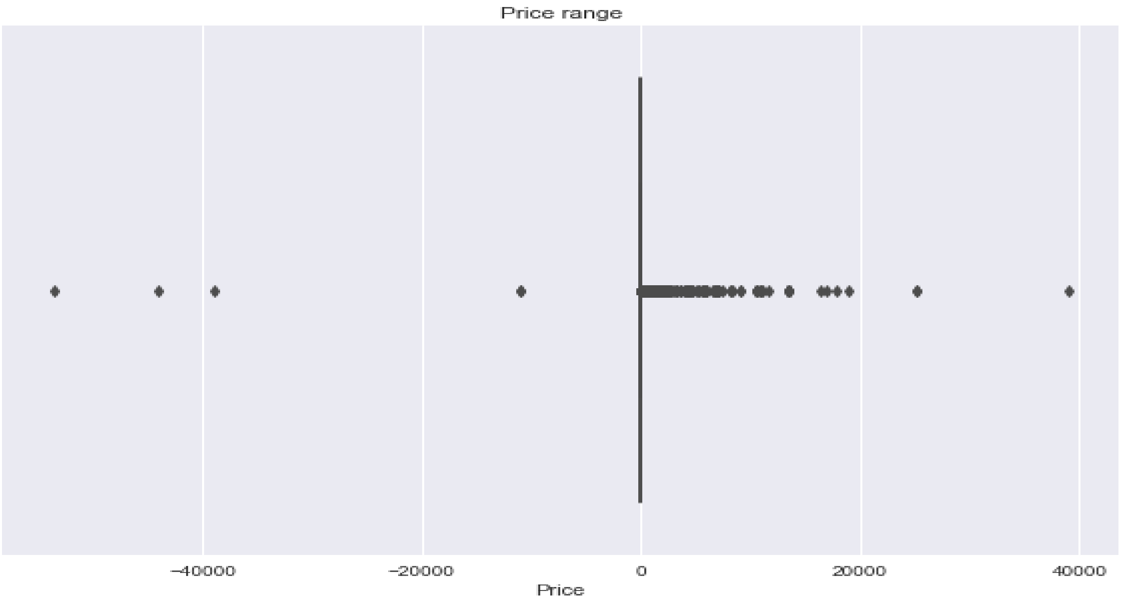


Figure 2: Price distribution

There were some negative prices which turned out to be adjusted bad debt and were deleted. Also, there were some prices that were zeroes and there were negative quantities. These turned out to be returned damaged items and were deleted.

Quantity:

There were more sales that had negative quantities, and further investigation shows that they were cancelled products, so a function was created to read the dataframe and see if there were duplicate orders made with positive quantities and these were cancelled. Then all entries with negative quantities were deleted because these are not sales.

StockCode:

Stockcode represents the unique product code. However, weird stock codes like postage, Manual, Amazon fee and Bank charges were found, and these are not sales so they were deleted.

Next thing to investigate is if there were any specialty products by seeing if there were any popular products.

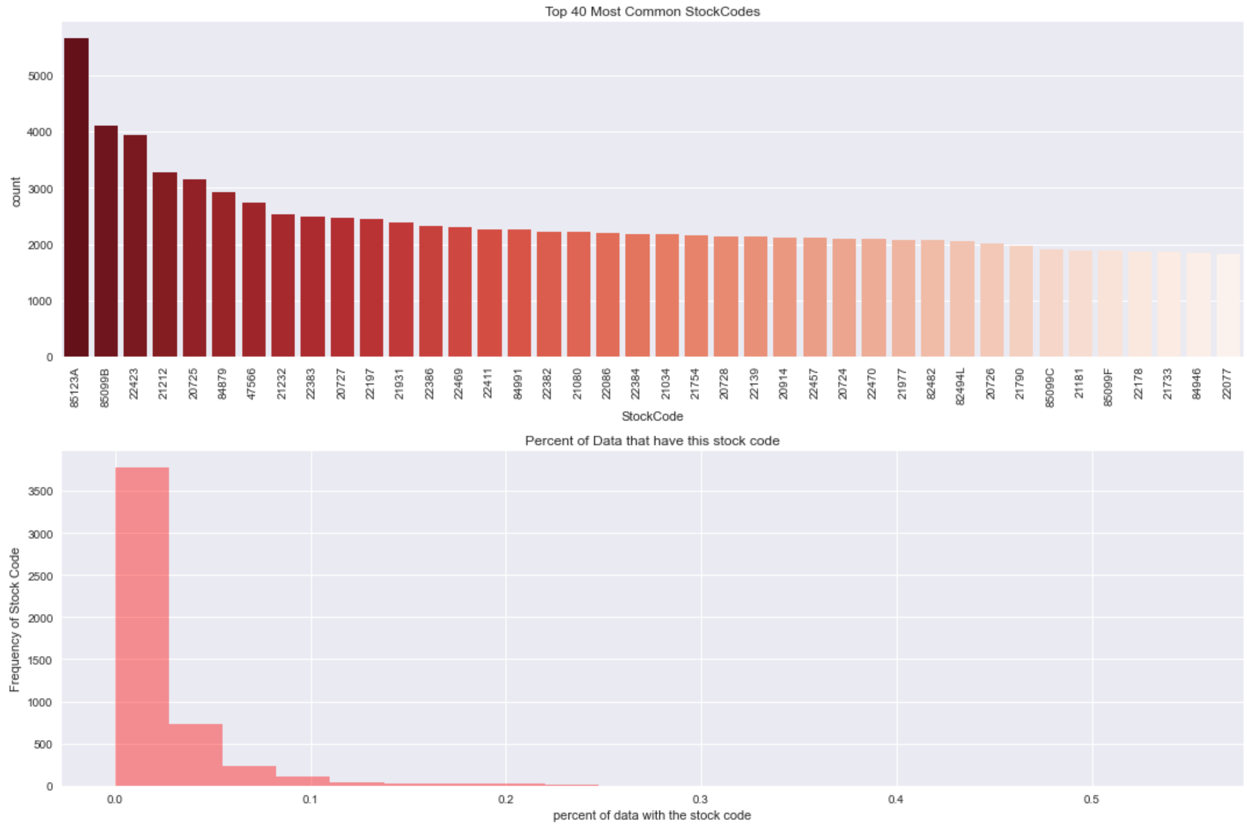


Figure 3: Most popular products by stock code

Most stockcodes are not very frequent, meaning that the seller sells many types of products and that there are no specialty products. But this could be misleading because the same product could have different stock code like for example if the same product had different color and different stock code, so I need to see if there are stock codes that differ only by a letter that could represent the same product. Next to investigate is the number of digits and letters in stock codes and see if there is a pattern and Gift vouchers, test products and a product called adjust. Test products and adjust were deleted.

Customer ID:

There were 30% of customer ID that were NANs, which were guests, since they were paying customers and many of them bought gift vouchers. These customers were saved in a different dataframe and a new dataframe was created without the unknown customers. These are the customers that were most active

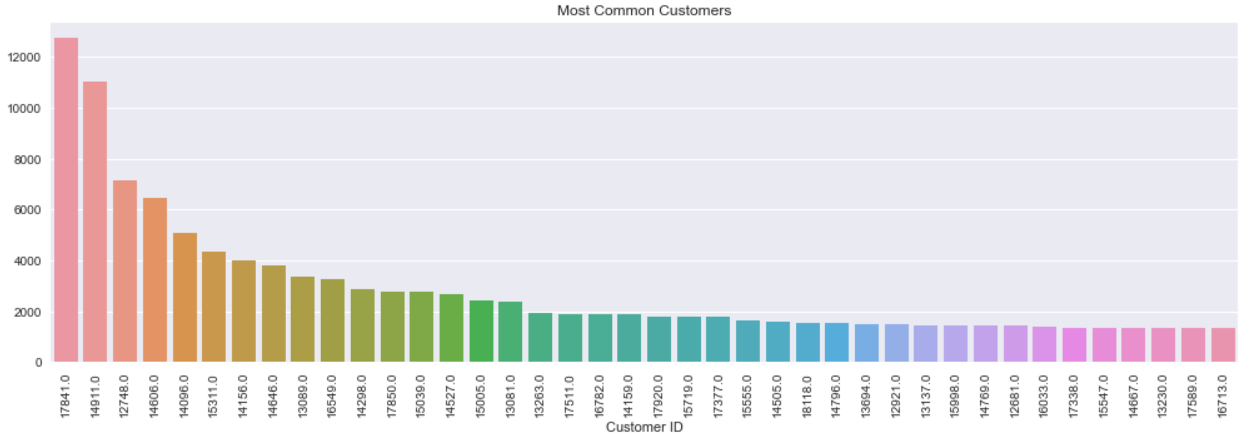


Figure 4: Most active customers by customer id

And these are the customers with the highest revenue

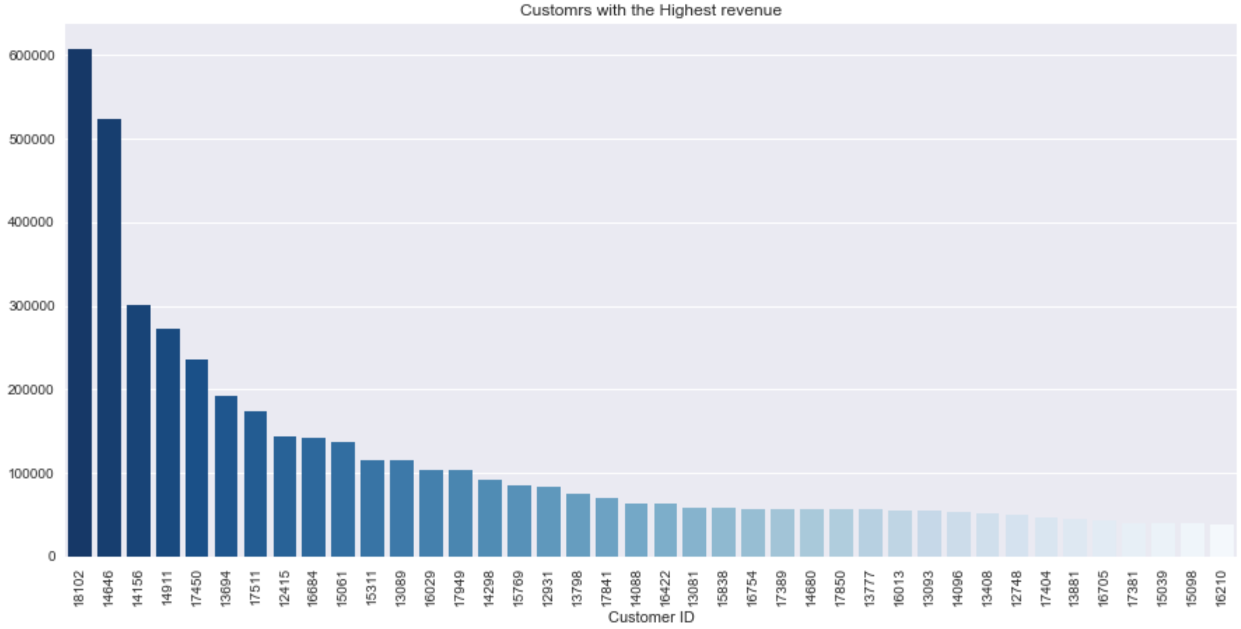


Figure 5: Customers that bring the most revenue

Created Features:

A few features were created:

1. Revenue = quantity * price

2. Year
3. Month
4. Week
5. Day
6. Quarter
7. Week day
8. Day of year

All except the first were extracted from InvoiceDate feature

Data Analysis:

The cleaned data that had all the known customers was read into a new notebook

Customer Cohorts:

Cohort analysis is a type of behavioral analytics, which is primarily identified by breaking down customers into related groups in order to gain a better understanding of their behaviors. In other words, a cohort is a group of people who have something in common during a specific time period. For example,if a company rolled out a new feature, they can analyse the effect of this feature on customer retention for example.

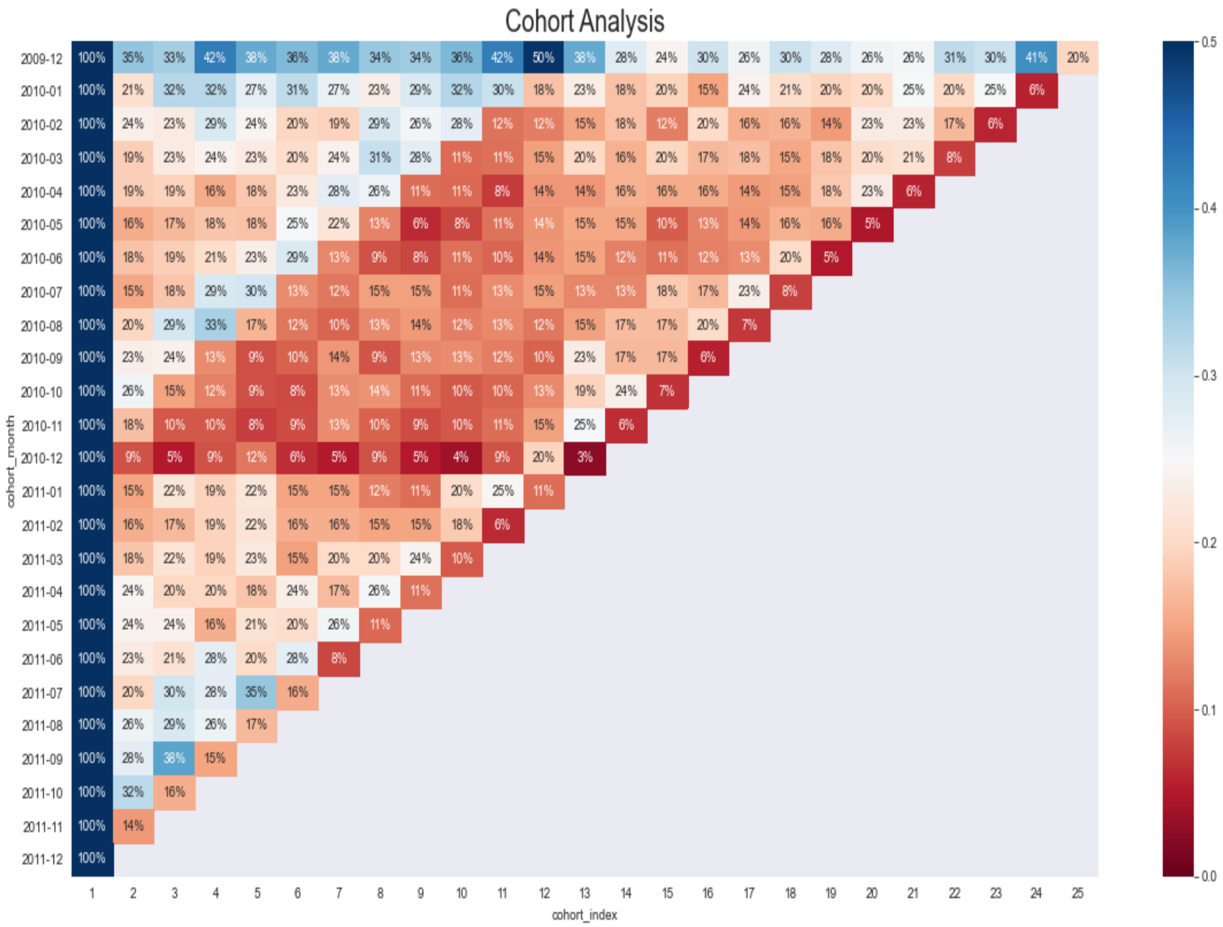
Some of the reasons for creating customer cohorts are:

- Know how user behaviors affect your business.
- Understand customer churn.
- Calculate customer lifetime value.
- Optimize your conversion funnel which describes the different stages in a buyer's journey leading up to a purchase.
- Create more effective customer engagement.

There are 5835 customers in the dataset, and since monthly cohorts are being created, the number of months between purchases are calculated for each customer and then customers are grouped into groups according to their first purchase month. Here new columns were created, namely cohort month, which is the month customers made their first purchase, and order month, which is the month of all their subsequent purchases, at which point the number of months between the cohort month and the order is obtained, called cohort_index. The cohort index in this case ranges from 1 month to 25 months since the data spans two years. This is how it looks like:

```
      cohort_month   cohort_index   customer_count
  0   2009-12             1               949
  1   2009-12             2               330
  2   2009-12             3               317
  3   2009-12             4               403
  4   2009-12             5               359
  5   2009-12             6               342
```

Then this dataframe is pivoted with the cohort month as index and column names as the cohort index that was just calculated and the count of customers as the values, from which percentage customer count is calculated by dividing by the total number of customers at start of the cohort. Then a heat map is created to be able to view



cohort_month	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
2009-12	100%	35%	33%	42%	38%	36%	38%	34%	34%	36%	42%	50%	38%	28%	24%	30%	26%	30%	28%	26%	26%	31%	30%	41%	20%
2010-01	100%	21%	32%	32%	27%	31%	27%	23%	29%	32%	30%	18%	23%	18%	20%	15%	24%	21%	20%	20%	25%	20%	25%	6%	
2010-02	100%	24%	23%	29%	24%	20%	19%	29%	26%	28%	12%	12%	15%	18%	12%	20%	16%	16%	14%	23%	23%	17%	6%		
2010-03	100%	19%	23%	24%	23%	20%	24%	31%	28%	11%	11%	15%	20%	16%	20%	17%	18%	15%	18%	20%	21%	8%			
2010-04	100%	19%	19%	16%	18%	23%	28%	26%	11%	11%	8%	14%	14%	16%	16%	16%	14%	15%	18%	23%	6%				
2010-05	100%	16%	17%	18%	18%	25%	22%	13%	6%	8%	11%	14%	15%	15%	10%	13%	14%	16%	16%	5%					
2010-06	100%	18%	19%	21%	23%	29%	13%	9%	8%	11%	10%	14%	15%	12%	11%	12%	13%	20%	5%						
2010-07	100%	15%	18%	29%	30%	13%	12%	15%	15%	11%	13%	15%	13%	13%	18%	17%	23%	8%							
2010-08	100%	20%	29%	33%	17%	12%	10%	13%	14%	12%	13%	12%	15%	17%	17%	20%	7%								
2010-09	100%	23%	24%	13%	9%	10%	14%	9%	13%	13%	12%	10%	23%	17%	17%	6%									
2010-10	100%	26%	15%	12%	9%	8%	13%	14%	11%	10%	10%	13%	19%	24%	7%										
2010-11	100%	18%	10%	10%	8%	9%	13%	10%	9%	10%	11%	15%	25%	6%											
2010-12	100%	9%	5%	9%	12%	6%	5%	9%	5%	4%	9%	20%	3%												
2011-01	100%	15%	22%	19%	22%	15%	15%	12%	11%	20%	25%	11%													
2011-02	100%	16%	17%	19%	22%	16%	16%	15%	15%	18%	6%														
2011-03	100%	18%	22%	19%	23%	15%	20%	20%	24%	10%															
2011-04	100%	24%	20%	20%	18%	24%	17%	26%	11%																
2011-05	100%	24%	24%	16%	21%	20%	26%	11%																	
2011-06	100%	23%	21%	28%	20%	28%	8%																		
2011-07	100%	20%	30%	28%	35%	16%																			
2011-08	100%	26%	29%	26%	17%																				
2011-09	100%	28%	38%	15%																					
2011-10	100%	32%	16%																						
2011-11	100%	14%																							
2011-12	100%																								

Figure 6: Monthly cohorts

The index can change according to the type of analysis being done, for example this analysis can be done as quarterly or daily customer retention.

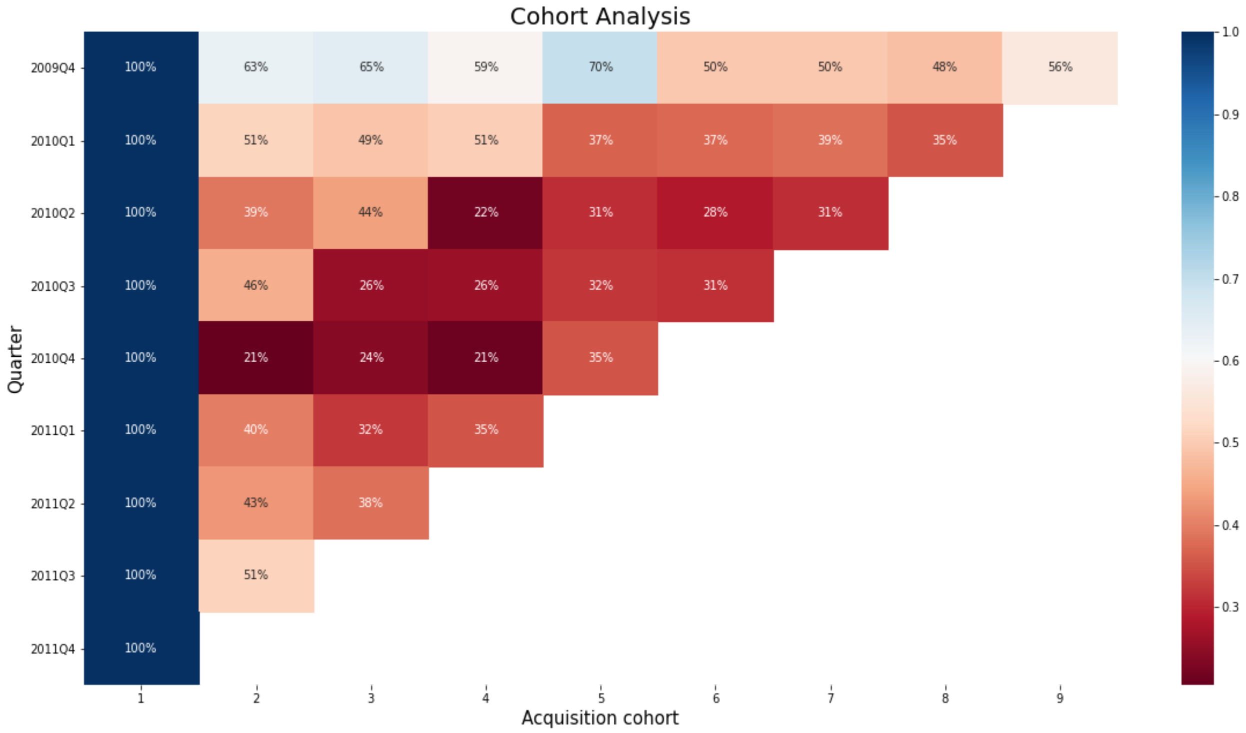


Figure 7: Quarterly Customer Retention

From figure 6 it looks like many customers did not return at the end of year 2010, maybe the store introduced a feature to their website that did not prove to be popular with the customers.

Recency, Frequency and Monetary Value or RFM analysis:

While cohort analysis gives businesses information on their customer behavior overtime and a better understanding of their retention rates, they also want to be able to segment the data by customers' behavior as well. There is a business saying that "80% of your business comes from 20% of your customers." So businesses need a method to be able to give their customers a quantitative value that is measurable and easy to understand. This is where RFM analysis comes in. It is a marketing technique used to rank and group customers based on their recency, frequency and monetary value of their transactions to identify the best customers to be able to perform targeted marketing campaigns.

Recency is the number of days since the last transaction, frequency means the number of transactions in the past period being investigated, and monetary value means how much the customer has spent in that period. Then Each customer is assigned a numerical score based on these and that is called the RFM score.

Usually the analysis is done on the current date looking back a specific period of time, since this data is from 2010-2011, I created a hypothetical date that is going to be the last date in the data set plus 1 more day. Next aggregate customers using the number of days since last purchase, the number of purchases a customer makes and the total amount a customer spends. Then the data is cut into 4 quarters with labels 1 to 4, with 1 being most recent and 4 being most frequent and highest monetary value. These three scores are then summed to get an RFM score

Customer ID	R	F	M	RFM_segment	RFM_score
12346	1	2	1	121	4
12347	4	4	4	444	12
12348	3	2	3	323	8
12349	4	4	4	444	12
12350	2	1	1	211	4

So top score would be 3 for RFM segment 111. Then we can sort by segment size to see which is the biggest segment, and get summary statistics

RFM_score	Recency	Frequency	MonetaryValue	MonetaryValue
	mean	mean	mean	count
3	543.28	9.66	172.87	532
4	375.96	16.9	242.98	563
5	313.52	24.7	411.21	632

Clustering:

Businesses use clustering to cluster customers according to their purchase history and past activities. This helps understand customers and what they like, also it is easier to target customers according to their behaviour, so if there is a cluster where customers are very active, they can be rewarded with discounts for example, or if there are customers that are less active, they can be given incentives to get them to buy more. I will use Kmeans and hierarchical clustering to cluster customers according to their purchase history and past activities.

The data that is going to be used is the dataframe with the following features: Recency, Frequency, Monetary Value

Kmeans:

Kmeans is a cluster based algorithm where the data is partitioned into samples such that similar instances are grouped together in the same partition.

Kmeans is the the easiest clustering algorithm to use and the most popular. There are three requirements for kmeans:

1. All the features have to have a symmetric distribution, meaning no skewness.
2. The variables have the same average value, meaning their means are the same, So they are standardized which ensures that each variable gets equal weight in the kmeans calculation
3. Variables have the same variance.

This is how the features looked like

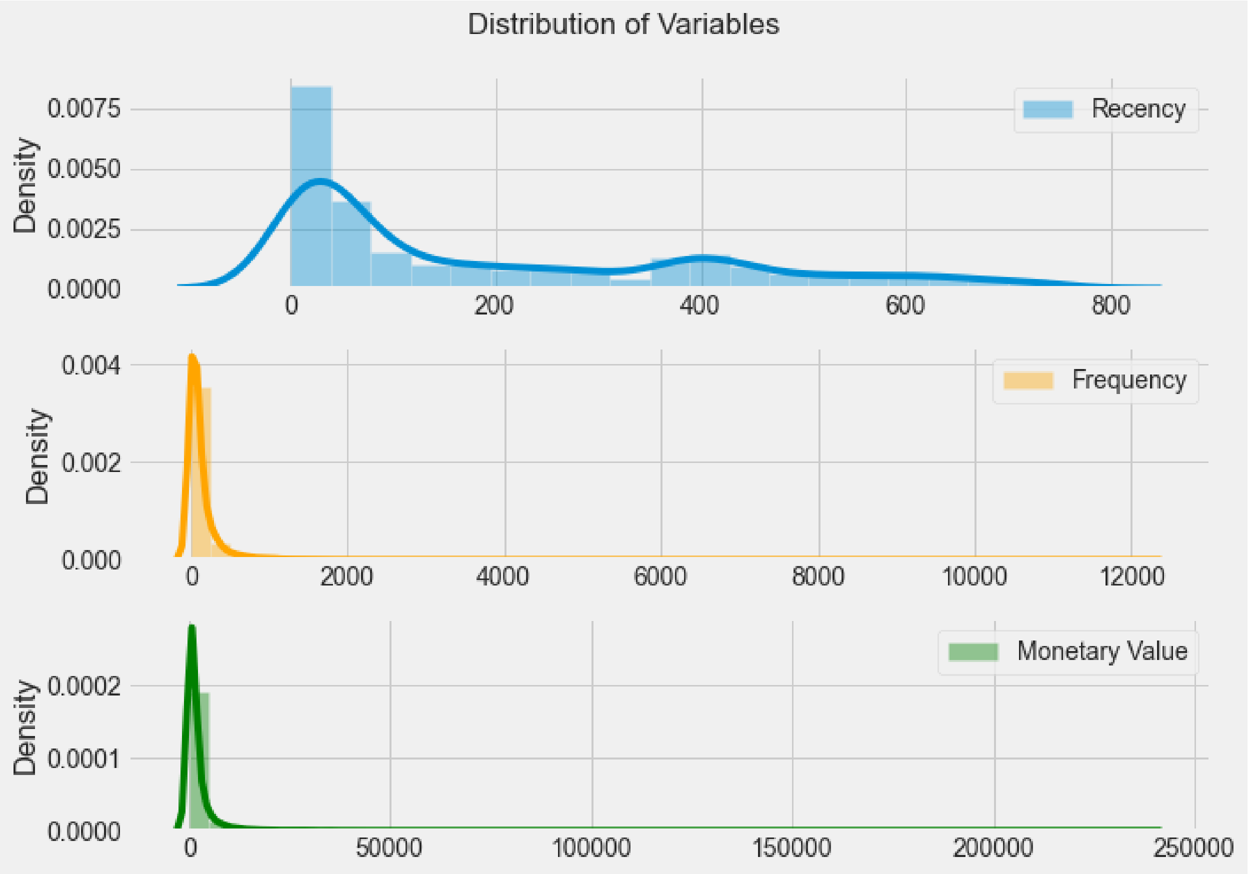


Figure 8: distribution of Recency, Frequency and Monetary Value

And this is their skew, where skew shows the asymmetry of a distribution. Normal distribution has skewness equal to 0. If the skewness is between -0.5 and 0.5, the data is fairly symmetrical

Recency	0.895357
Frequency	18.470250
MonetaryValue	19.263475

Skew is high so the data is highly skewed. So log transformation was performed on the data, but 1 was added to the datapoints beforehand to ensure that there are no zeroes. After performing the log transformation this is the skew

Recency	-0.483442
Frequency	0.034652

MonetaryValue   0.098873
Since skewness between 0.5 and -0.5 the variables now are normal. This is how the distribution looks like after transformation and standardizing

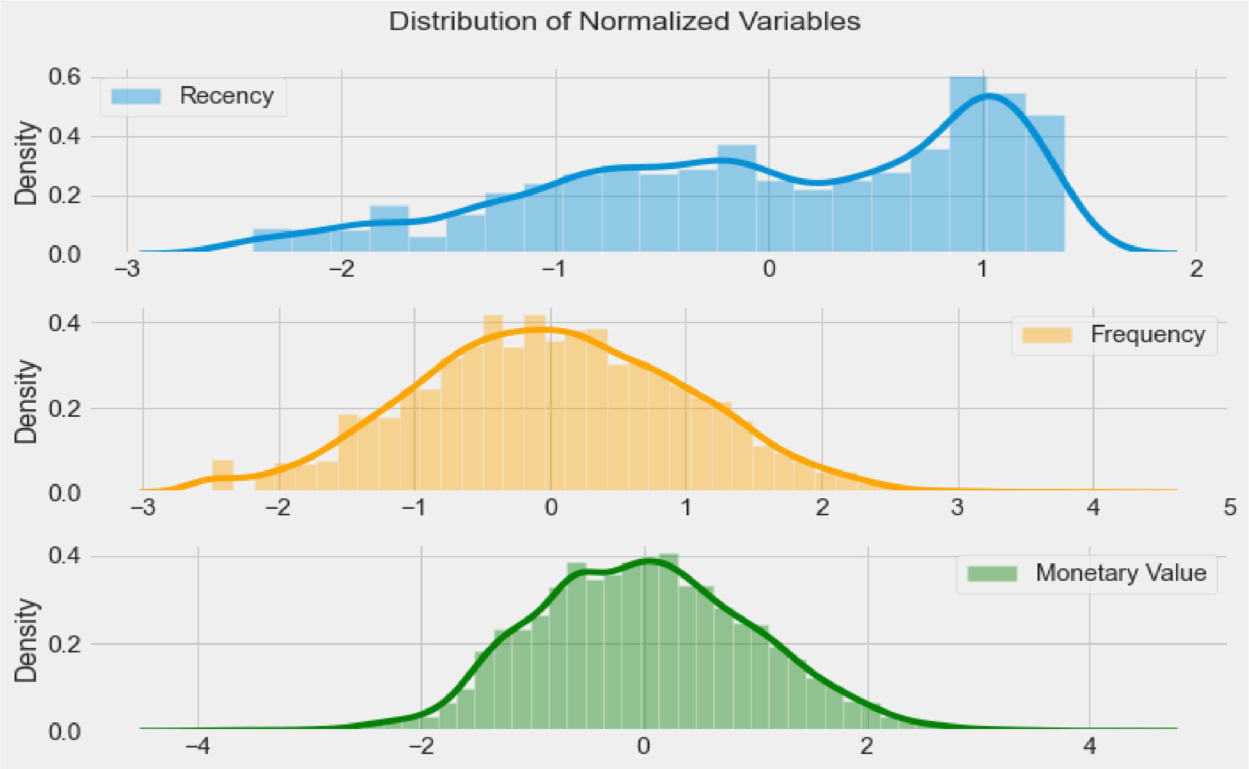


Figure 10: Distribution of variables after standardization

There are three metrics to measure how good the clusters are:

1. The elbow method, which measures the sum of squared distances of samples to the nearest cluster center and plots it to see a point that looks like an elbow.
2. The Silhouette Index measure the distance between each data point, the centroid of the cluster it was assigned to and the closest centroid belonging to another cluster, (formula is b-a/max(a,b)), values close to 0 indicate overlapping clusters, and closer to 1 indicate a better separated dense clusters
3. David Bouldin score evaluates intra-cluster similarity and inter-cluster differences. Lower value indicates a better model with better separation between clusters

2 to 10 clusters were tested to see which one gave better scores. This is the sum of squares graph

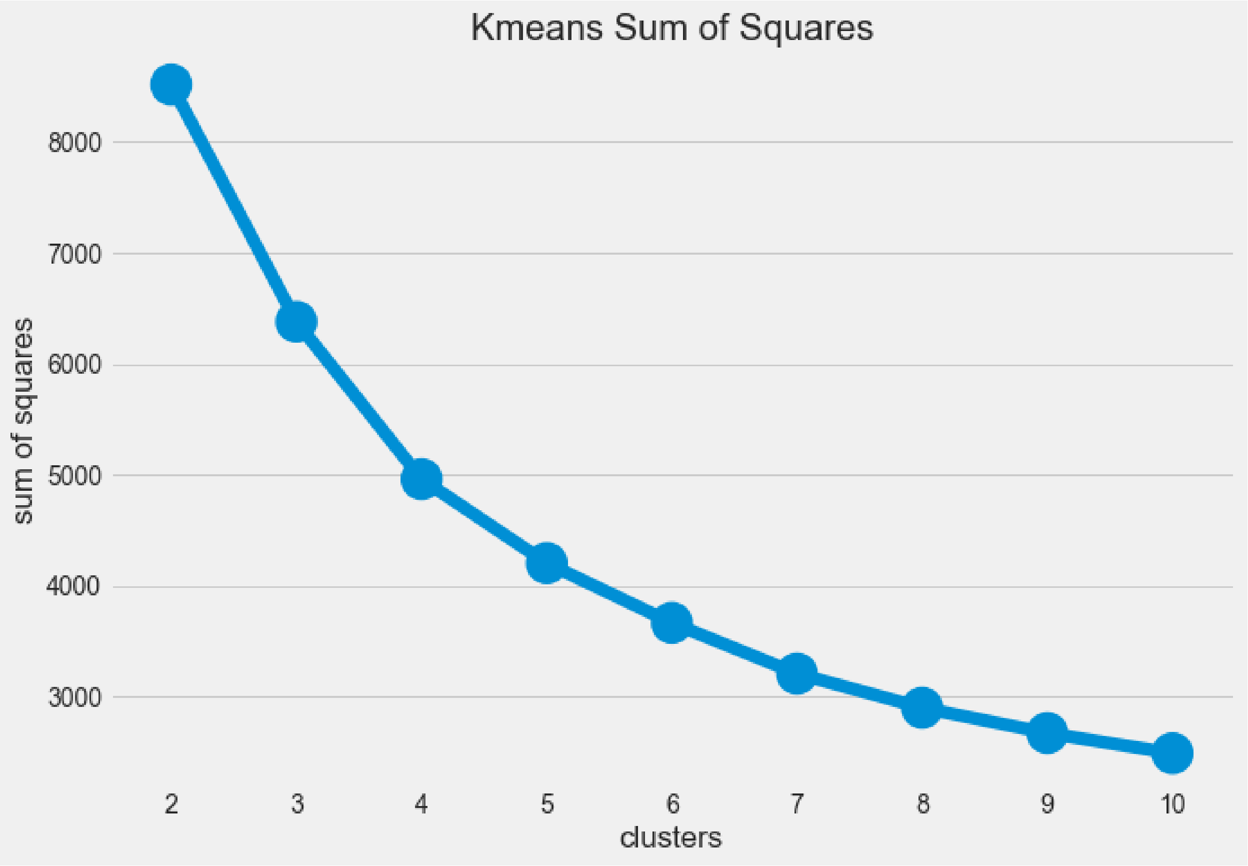


Figure 11: Kmeans sum of squares

There is a small elbow at 4 but results are not conclusive. The David Bouldin test, the lower the value the better

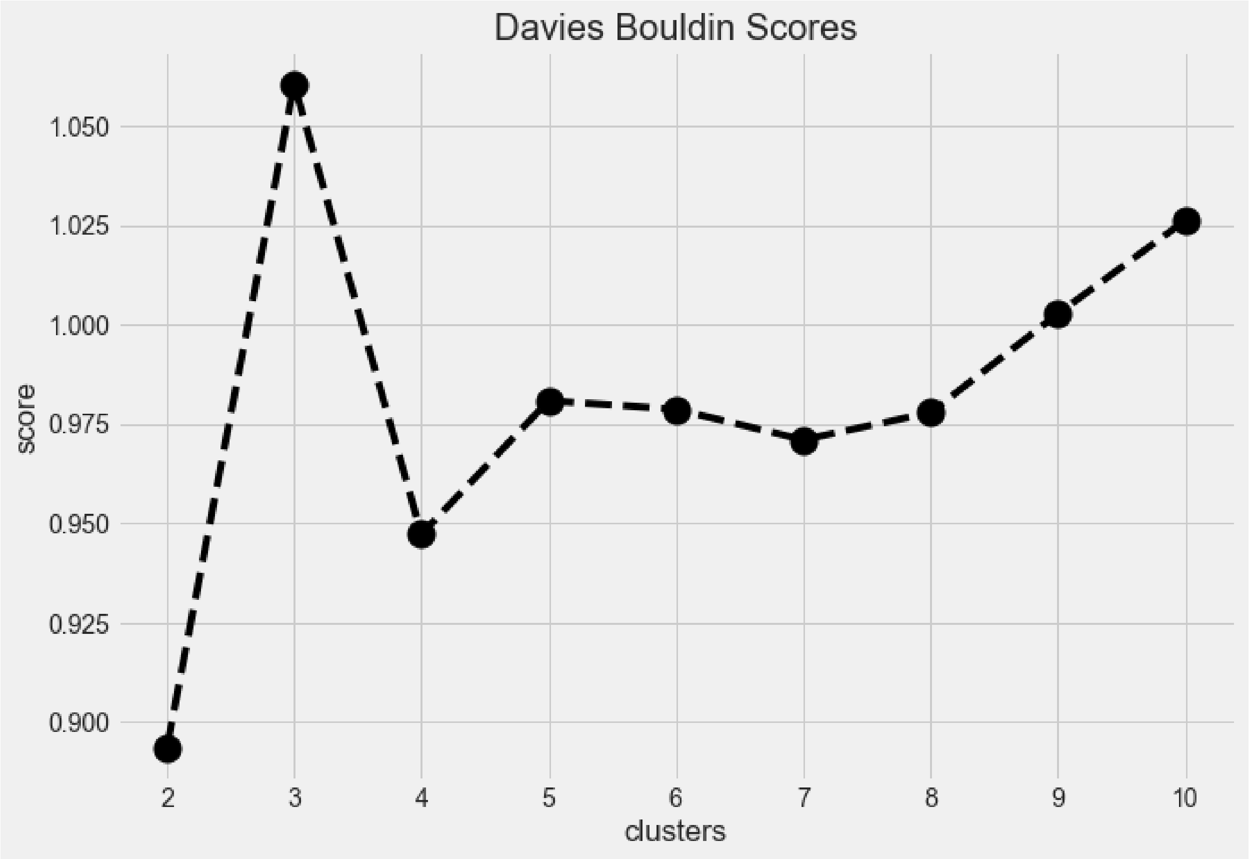


Figure 12: David Bouldin test

The lowest value is at 4, next is the Silhouette test, the closer the value to 1 the better

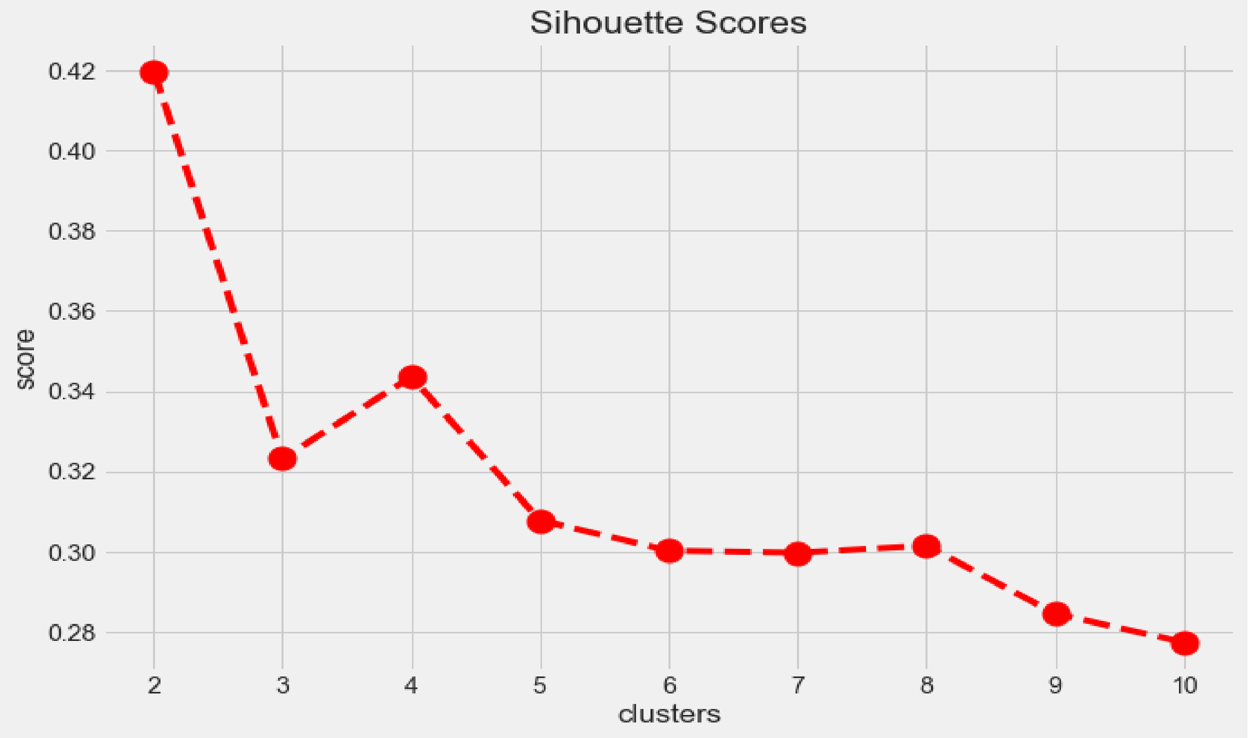


Figure 13: Silhouette score

The closest value is at 4 giving a score of 0.34. The Silhouette scores were low for all the clusters indicating that there is a great separation between them.

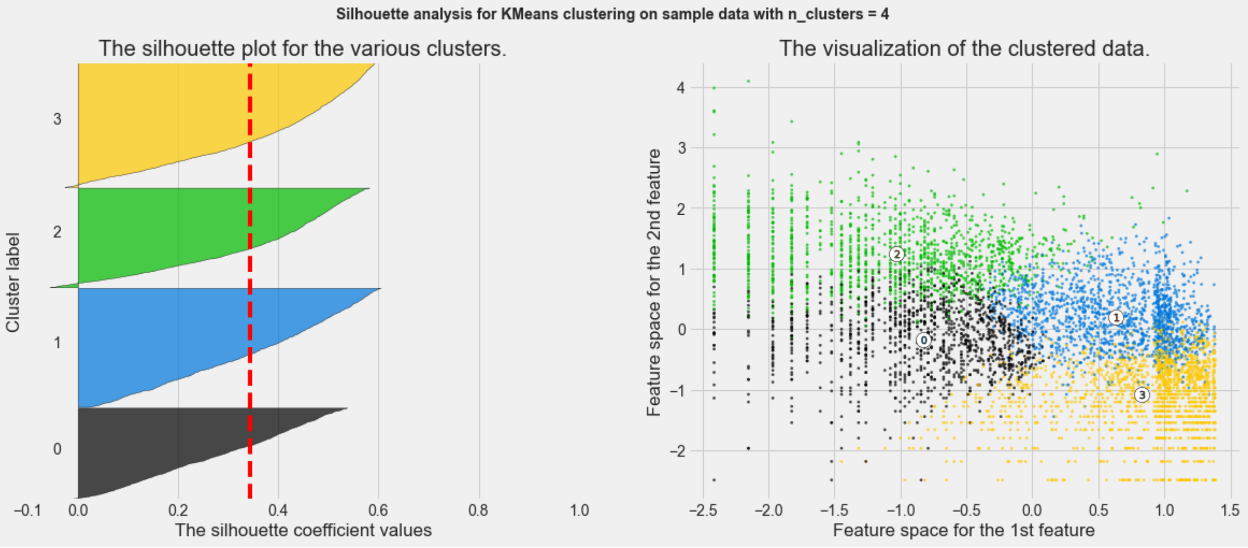


Figure 14: How data is divided in the 4 clusters.

The 4 clusters gave the best distribution of data, but all 10 clusters are above silhouette average. These are the descriptive statistics for the 4 clusters

```
        Recency    Frequency  MonetaryValue
        mean       mean       mean       count
Group
0       31.0       55.0       770.0      1207
1       279.0      88.0       1310.0     1568
2       29.0       395.0      6434.0     1311
3       386.0      15.0       235.0      1645
```

The group that spent the most and was the most active and most recent was group 2 which makes it the most important group. Group 1 has the second highest value, but they are not very active so the company could offer them some promotions and specials to get them to become more active. Group 3 is the least active group and they spend the least so the company could send them more offers and promotions to get them to become more active.

To be able to visualize the segments PCA was used to reduce the dimensionality of the data. This is how the clusters look like

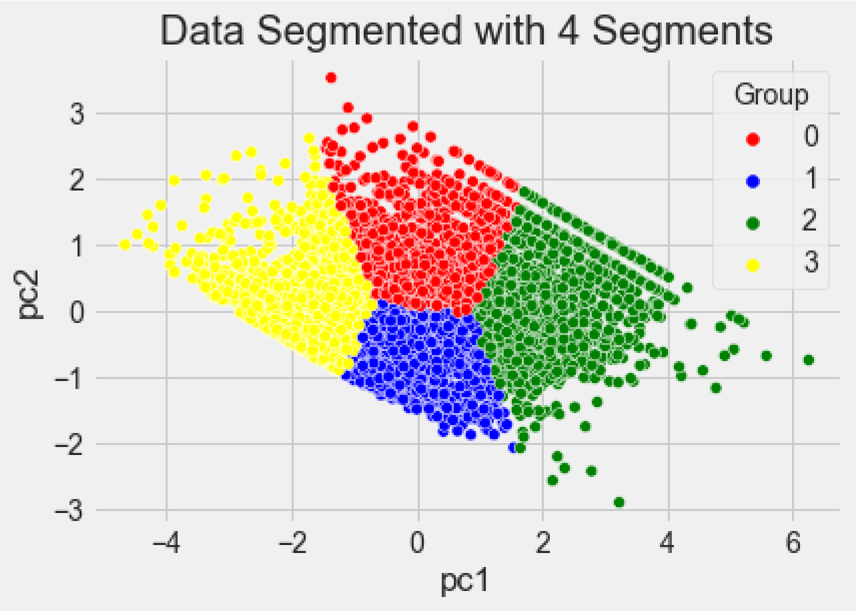


Figure 15: visualizing the 4 segments using pca

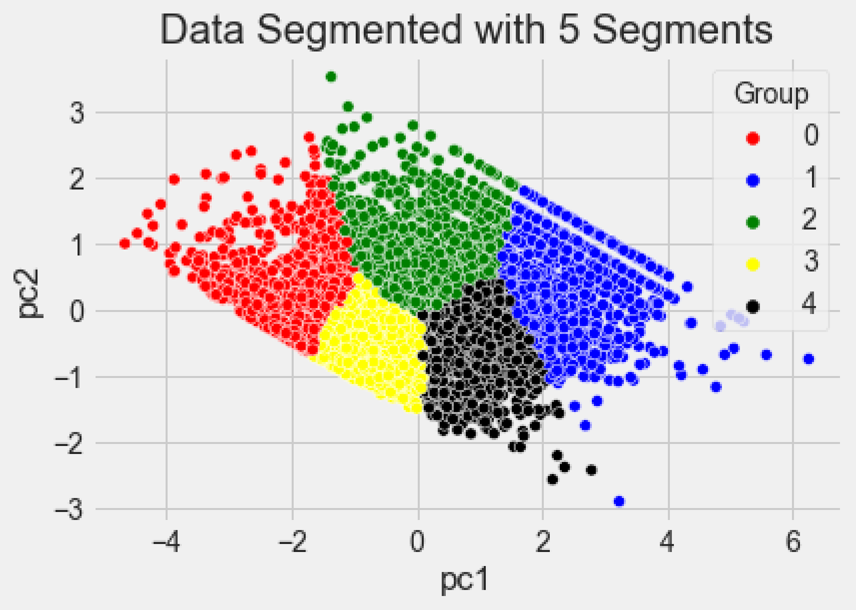


Figure 15: visualizing the 4 segments using pca

The clusters are very close together and that is why the silhouette scores were small, also having 5 clusters seems to further segment segment 1 and 2. This is further explored using Hierarchical clustering.

Hierarchical Clustering:

To get more insight on the clusters, hierarchical clustering was used. Hierarchical clustering also groups customers according to their behavior. It starts by making each point, or customer, its own segment, then it combines the closest segments using the euclidean distance between points, and continues to merge segments until there is one big segment. A dendrogram is produced.

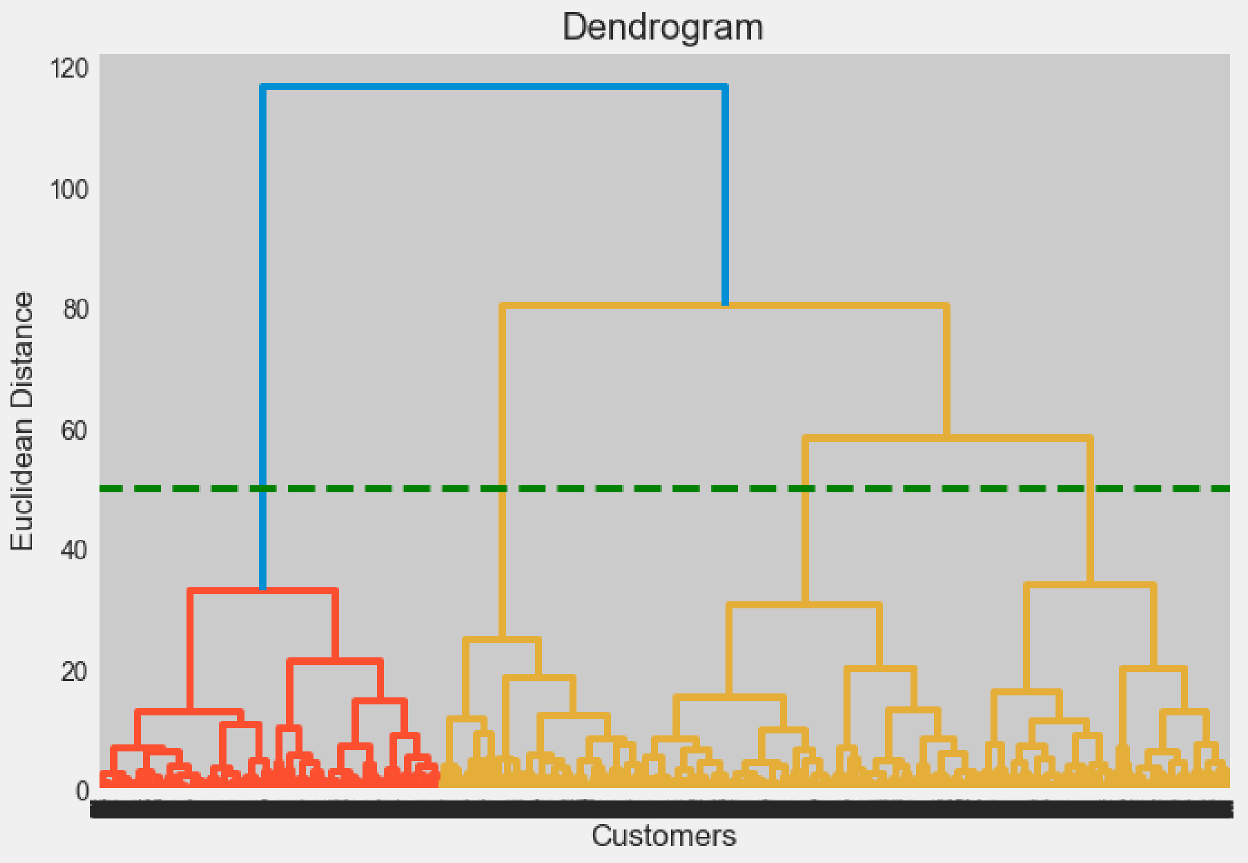


Figure 16: Hierarchical clustering dendrogram

From the dendrogram, a line is drawn at the longest cluster line which is the blue line and the number of lines are counted, each representing a cluster. Here, there are 4 clusters

So for 4 clusters now the customer behaviour looks like:

Group	Recency	Frequency	MonetaryValue	
	mean	mean	mean	count
1	31.0	55.0	770.0	1207
2	279.0	88.0	1310.0	1568
3	29.0	395.0	6434.0	1311
4	386.01	5.0	235.0	1645

Exploring Sales:

The dataframe with the missing customer ids was added to the rest of the data for sales prediction. Then some exploration was done to see how sales look like for the store.

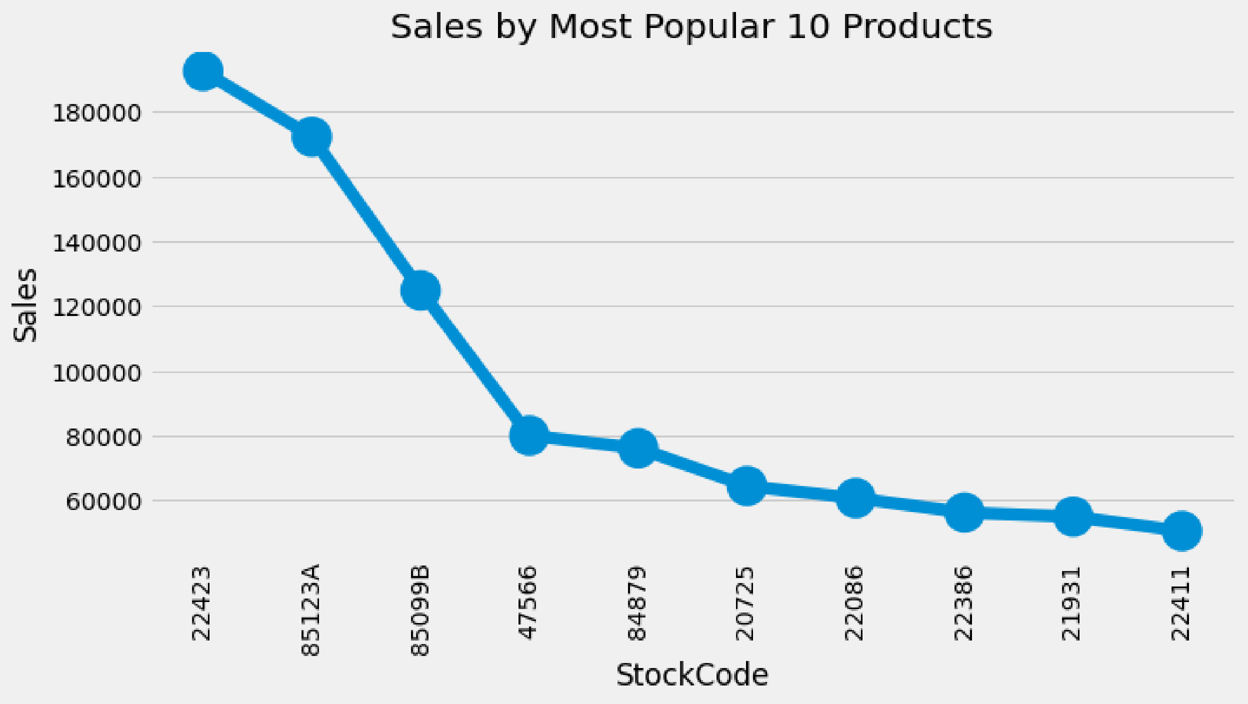


Figure 17: Sales by top 10 products

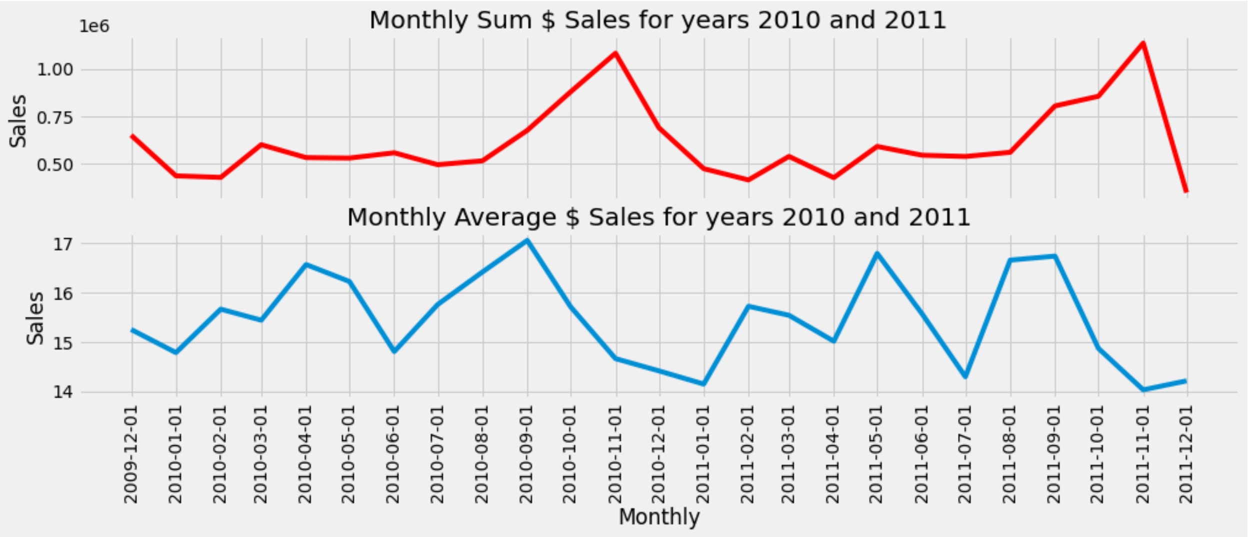


Figure 18: Monthly sum and average sales

In September 2010 there was a spike in average sales which then went down but in sum of sales continued to rise, which means that the number of orders continue to increase for products of similar value. There is also an increase in sales in November which reflects the Christmas rush.

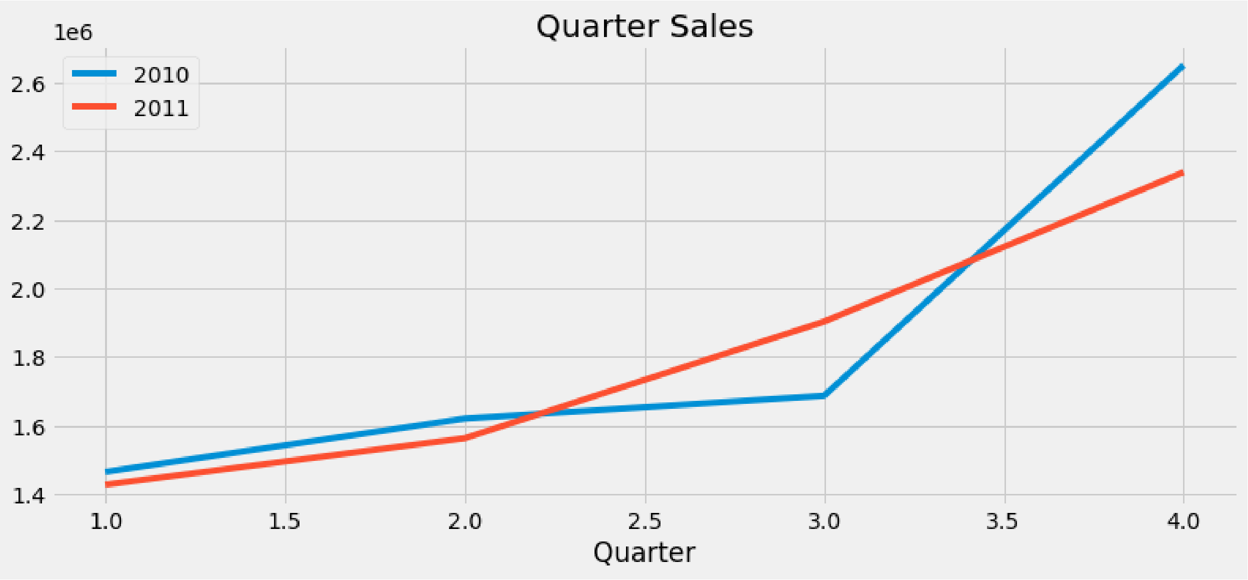


Figure 19: Quarterly sales

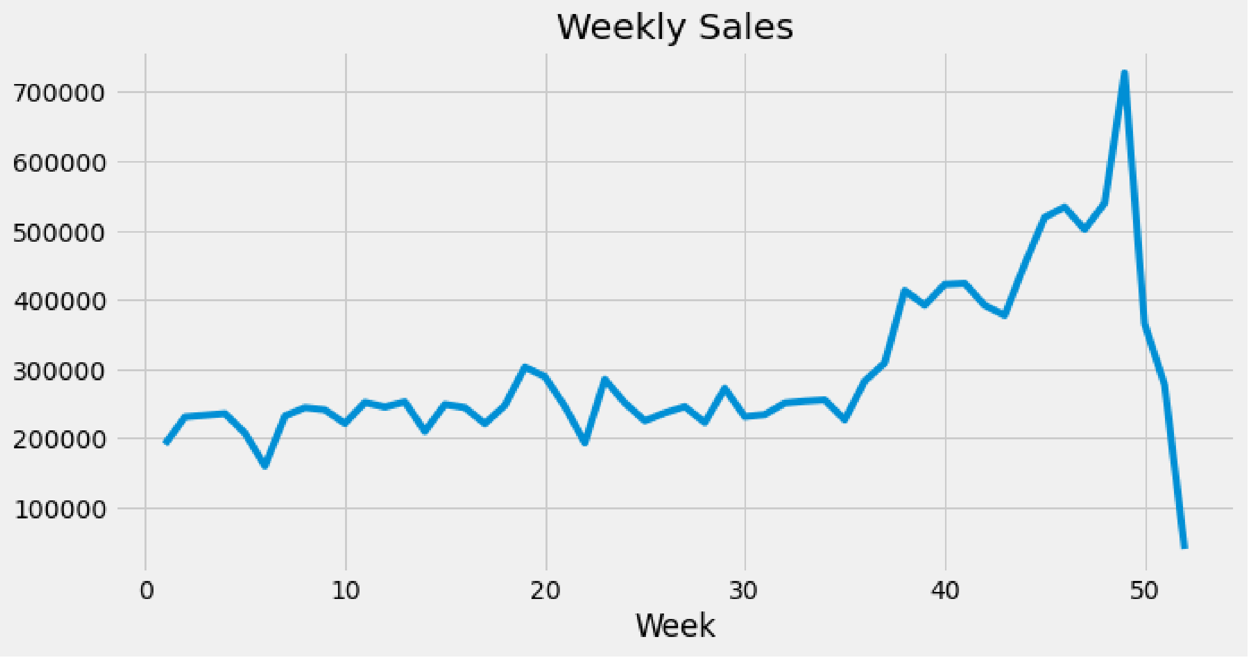


Figure 20: Weekly sales

Again the Christmas rush is evident in the increase in sales for the last quarter and the last week of the year.

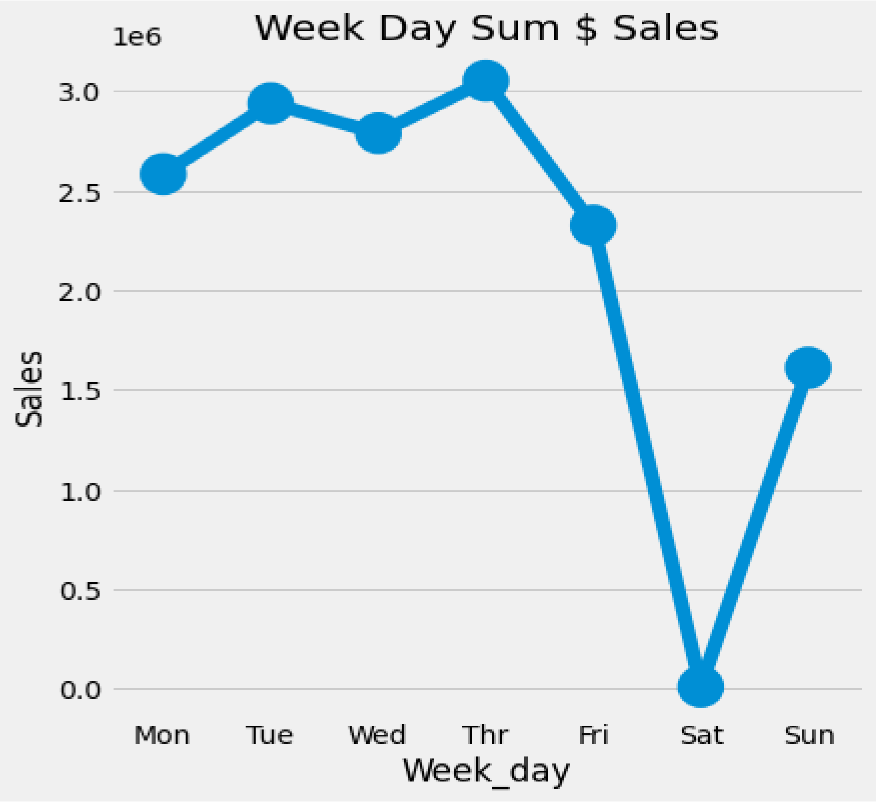
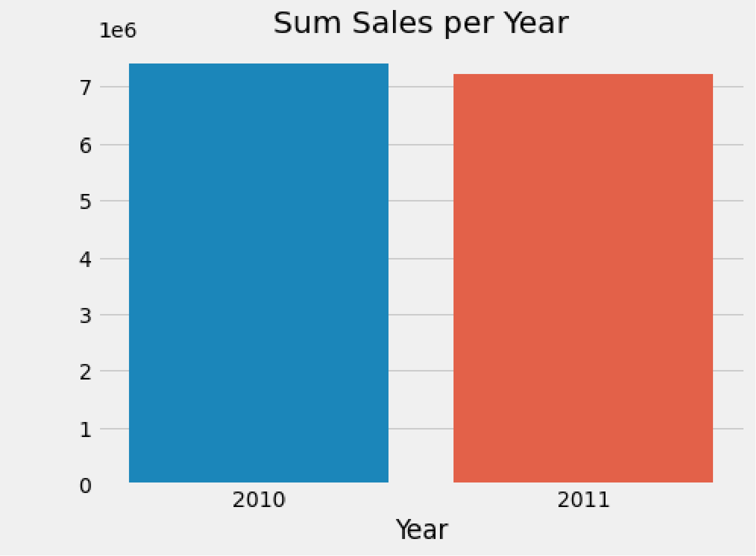
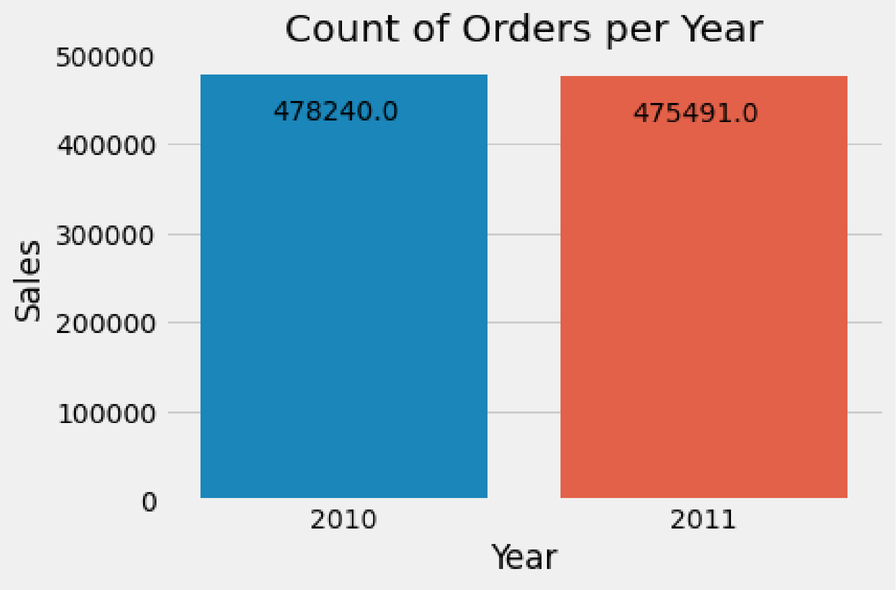


Figure 21: Daily sales

Since this data is for an online store people would be busy on Saturday and make their orders on Sunday and that is reflected in the noticeable decrease in sales on Saturdays.

Next look at sales and customer count for year 2010 compared with year 2011


Sum Sales per Year
1e6
7
6
5
4
3
2
1
0
2010
2011
Year


Count of Orders per Year
500000
400000
300000
200000
100000
0
Sales
478240.0
475491.0
2010
2011
Year

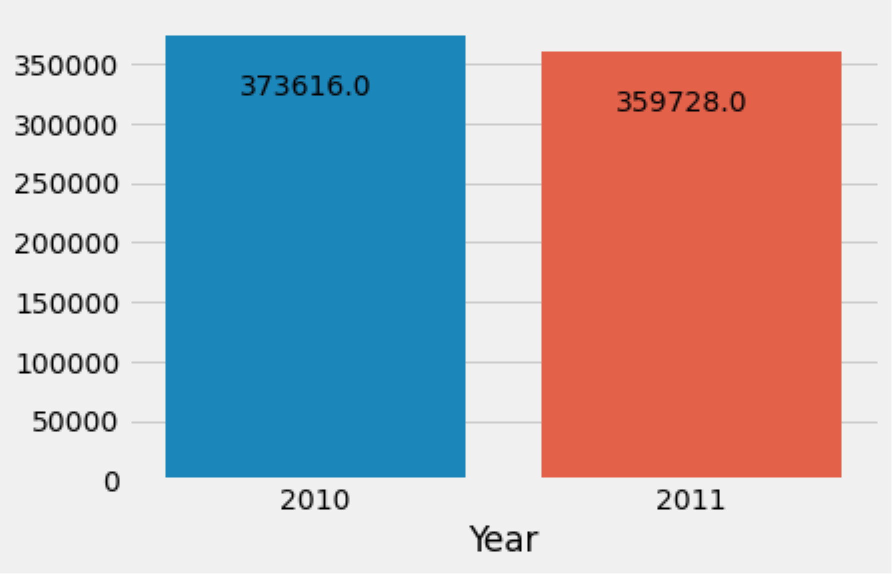


Figure 22: compare 2010 and 2011 in sales, number of orders and customer count.

There is a slight decrease in the number of orders and customers from 2010 to 2011, which the company needs to find out why.

Sales Prediction:

The first thing to do is explore the response variable sales to see if its is normal and if there are outliers. Many machine learning models, like linear regression, are easily impacted by the outliers in the training data. Knearest Neighbor regressor an random forest regressor are more robust towards outliers but can still be affected. So this is the boxplot of sales:

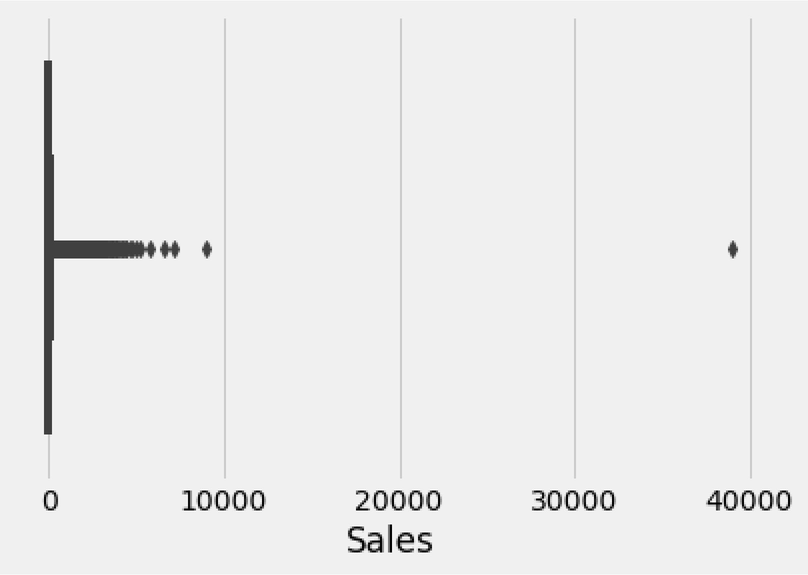


Figure 23: Sales Boxplot

It is very evident that there are many outliers in sales. The range goes up to 40,000 pounds. To solve this problem, only 99% of the data is kept and this is the distribution

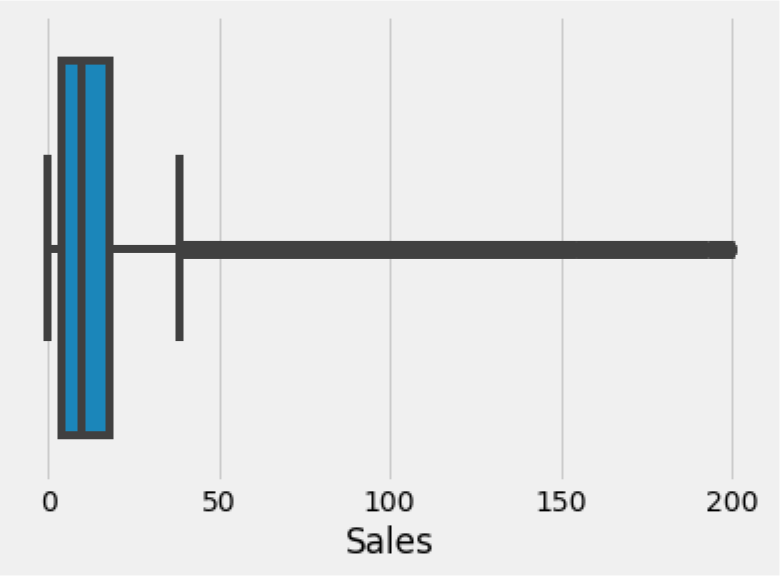


Figure 24: 99% of sales

There are still outliers but it is looking much better. Now the distribution of sales

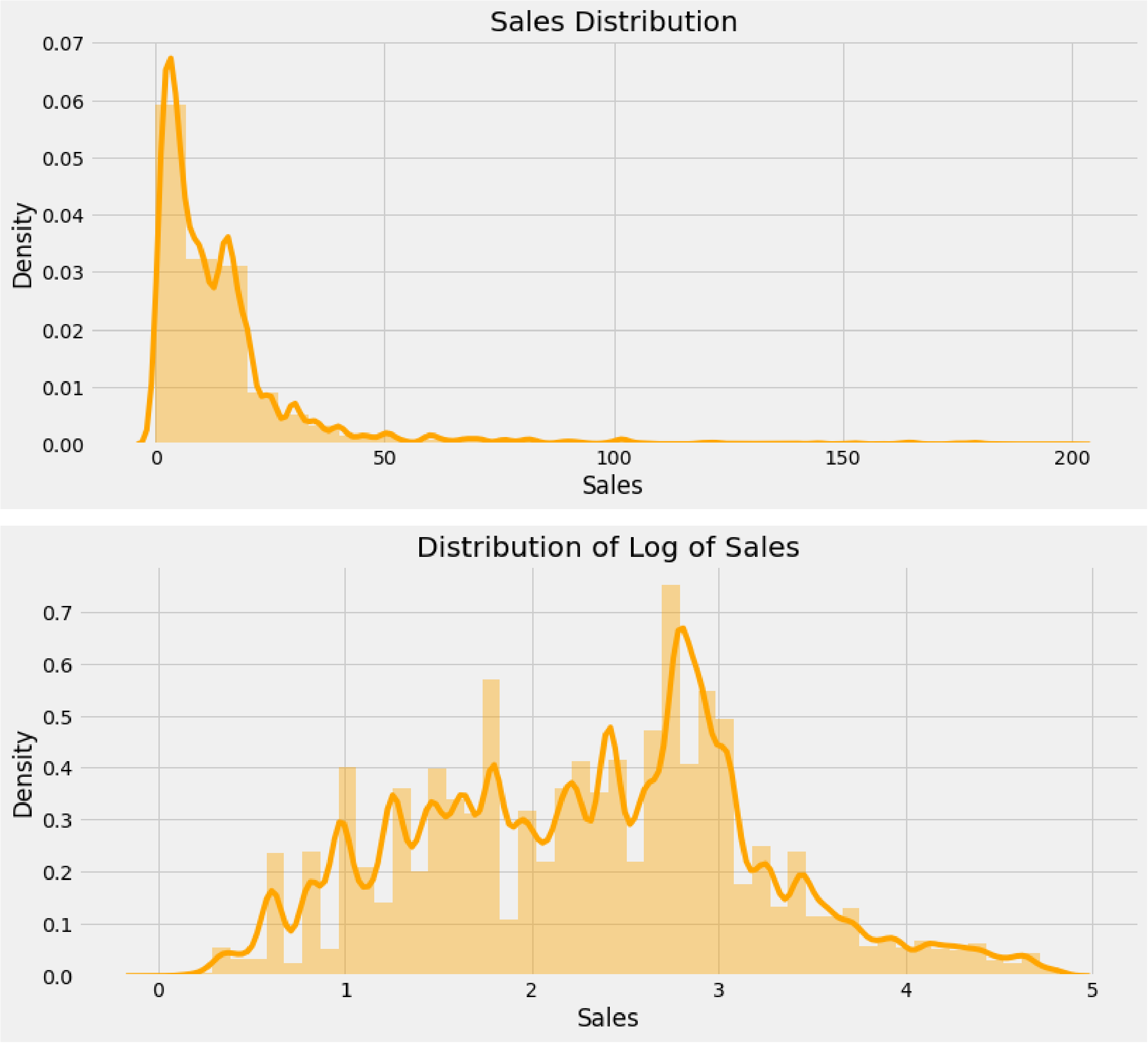


Figure 25: sales and log1p of sales distribution

Taking log1p of sales gives a more normal looking distribution.

Logistic Regression:

Since sales depend only on time, Customer Id, stock code, invoice quantity and price were all dropped and what was left was year, month, week, day quarter, week_day, and day_of_year. Then the data standardized using StandardScaler, then was split into 80% training and 20% test sets. The response variable is log1p of sales which was also split into 80% training and 20% test.

The metrics used are R_squared, mean squared error and mean absolute error, also root mean squared error. R2 is between 0 and 1, but it can be negative. If it is close to 0 then the model does not explain the variance of y, and if it is negative then it does not explain y at all and it is worse than having all predictions be the mean.

For Logistic regression these were the metrics:

r_square: 0.00224

MSE: ', 0.89

RMSE: 0.94,

MAE: 0.77

These were the residuals:

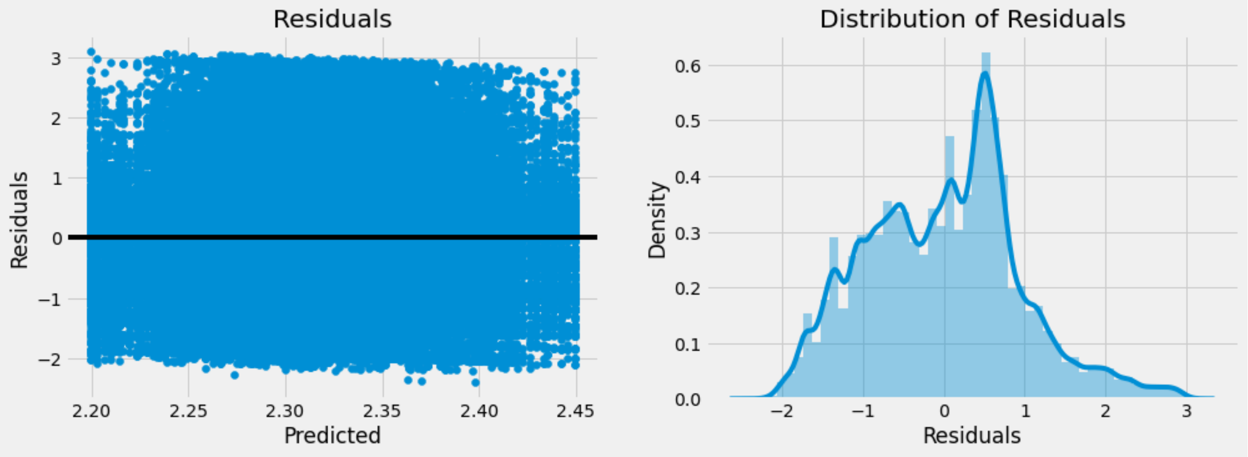


Figure 26: Linear Regression residuals

I tried to run knearest neighbors and random forest regressors with all the data but it took way too, so I ran them with data from year 2011.

Knearest Neighbors:

Parameters to be tuned are number of neighbors, from 1 to 20 neighbors and weight of points whether closer points get more weight or not. Using RandomizedSearchCV, these were the results:

weights: uniform, n_neighbors: 18 that gave the best scores, and they were

r_square: -0.01924

MSE: 0.93

RMSE: 0.96

MAE: 0.78

These are the residuals:

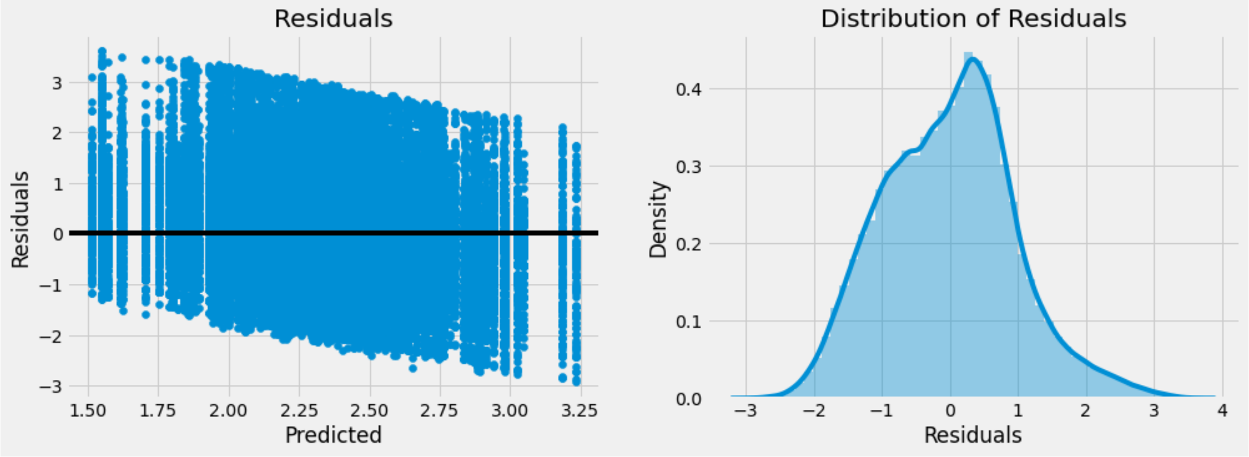


Figure 27: Knearest Neighbor Residuals

Random Forest Regressor:

Using RandomizedSearchCV these were the parameters that needed to be tuned:

- Number of trees in random forest or n_estimators = [200,400,800]
- Number of features to consider at every split or max_features = ['auto', 'sqrt']
- Maximum number of levels in tree or max_depth = [int(x) for x in np.linspace(5, 30, num = 6)]
- Minimum number of samples required to split a node or min_samples_split = [2,5,10, 15, 100]
- Minimum number of samples required at each leaf node or min_samples_leaf = [1, 2, 5, 10]

These were the metrics:

r_square: 0.0412

MSE: 0.88

RMSE: 0.94

MAE: 0.76

These were the residuals:

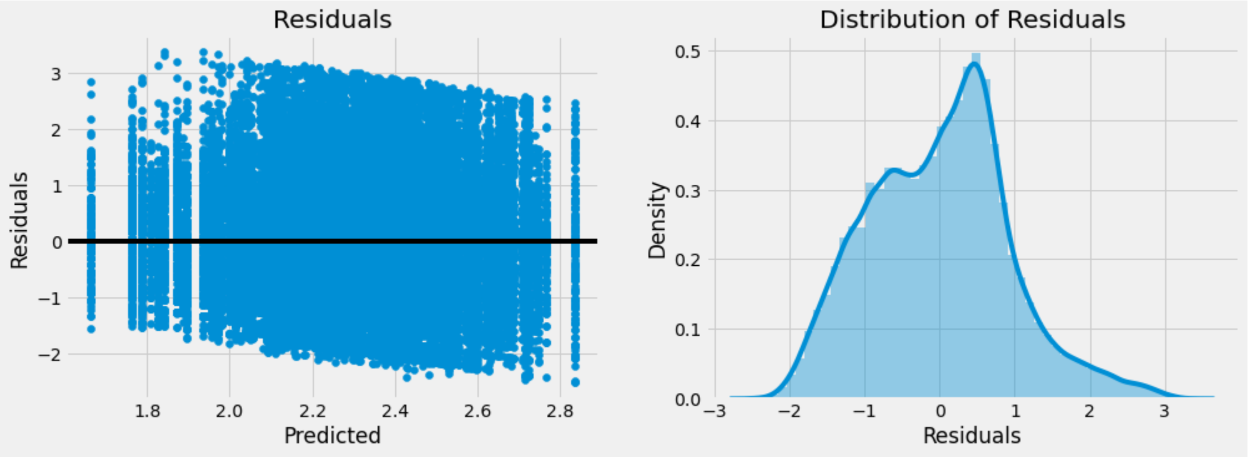


Figure 28: RandomForestRegressor residuals

Residual plot should be consistent with random errors with no pattern. If there is a pattern in the errors, this means that one error can be used to predict another. Looking at the residual plots, there is a clear pattern for the knearest neighbor and random forest models, which indicates that the models are failing to explain some relations within the data. The linear regression gave the best residuals with no patterns, however the residuals were not normally distributed which means that it is an inadequate model and the errors the model makes are not consistent across variables and observations.

ARIMA Model:

ARIMA stands for Autoregressive Integrated Moving Average. It is a class of models that explain a given time series based on the series' own past values. It is the most commonly used method for time-series forecasting however it can be inaccurate under conditions like financial crises.

Autoregression means a model that uses the dependent relationship between an observation and some number of lagged observations. Integrated means subtracting an observation from an observation at the previous time step in order to make the time series stationary, and Moving Average meaning a model that uses the dependency between an observation and a residual error from a moving average applied to lagged observations. In other words, ARIMA makes use of lagged moving averages to smooth time series data. It assumes that the future will resemble the past.

There are three parameters that need to be tuned:

- p is the number of lags of Y to be used as predictors
- d is the minimum number of differencing needed to make the series stationary. A d =0 means the series is stationary
- q refers to the number of lagged forecast errors that should go into the Model.

In order to use the ARIMA model the time series needs to be stationary. There are two tests to run:

1. Rolling mean: The window of one week is rolled across the data and an average is taken which is compared to the original data with the rolled data. Using visualization, if the original data and rolled data look similar, it means that the mean and variation do not change over time and that they are constant, so no transformation is necessary
2. Dicky_Fuller_test: it tests the null hypothesis that the data is not stationary and the alternative hypothesis is that the data is stationary. If the p-value is less than the critical value, 0.05 I will reject the null hypothesis and say that data is stationary.

This is the graph for rolling mean:

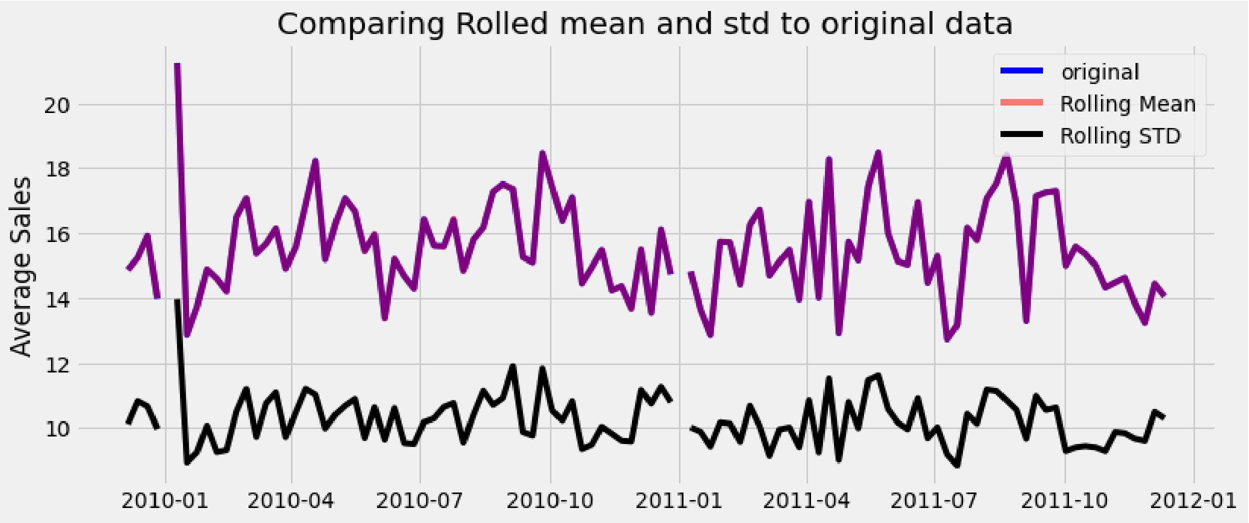


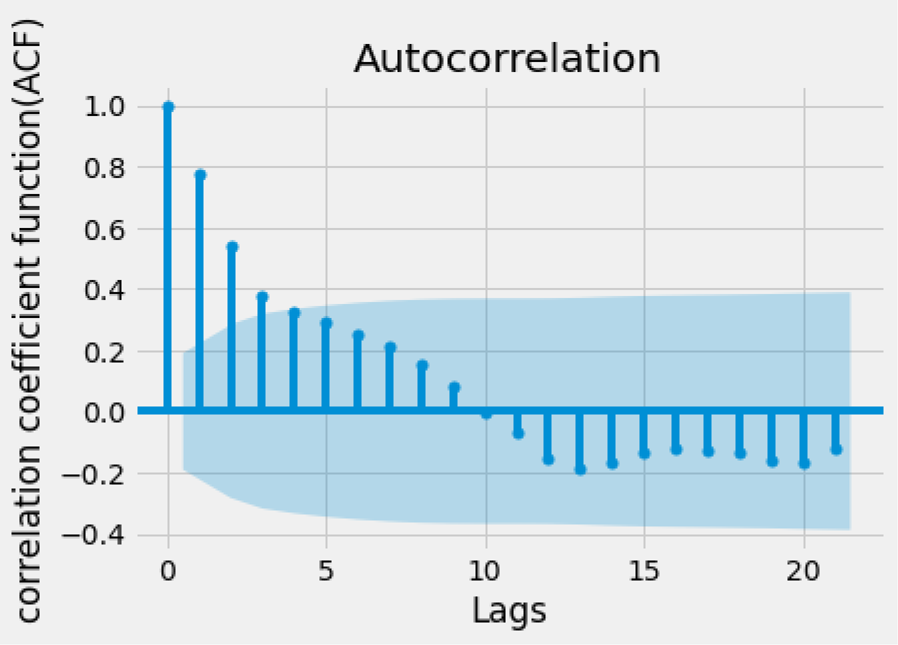
Figure 29: Rolling Mean

The plot of the original and rolled data are identical and they fall on top of each other, meaning the mean is constant over time and the data is stationary.

The Dicky Fuller test results were significant with p_value of 0.014, which means the null hypothesis can be rejected and the series is stationary.

Parameter Tuning:

First I ran the model with preliminary values that were gotten from autocorrelation and partial autocorrelation graphs:



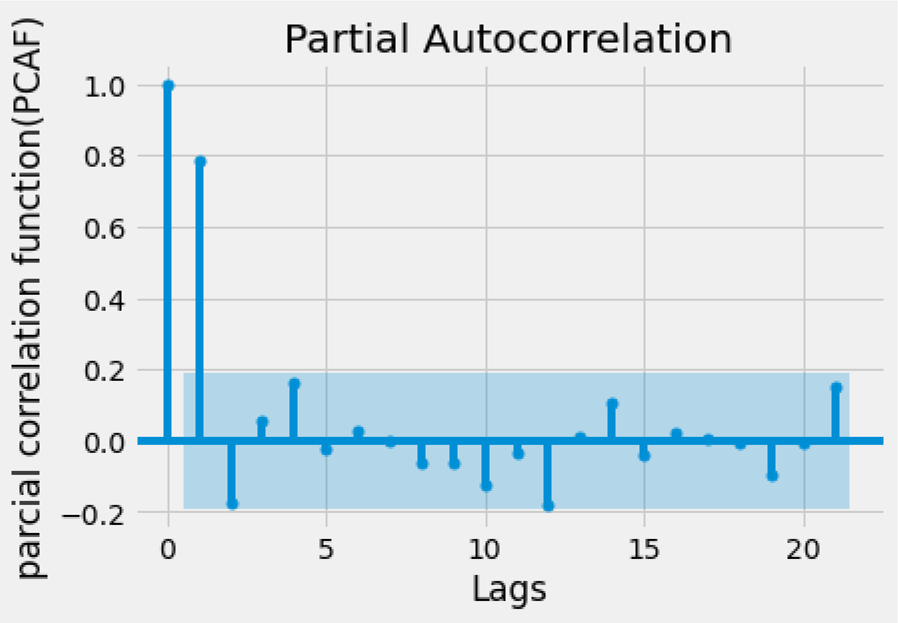


Figure 30: Autocorrelation and partial autocorrelation

- The Autocorrelation function (ACF) plot shows how a given time series is correlated with itself. It refers to how correlated a time series is with its past values.This is the plot used to see the correlation between the points, up to and including the lag unit. From the ACF graph the time series is significantly positively correlated with its past values at 1,2 and 3 lags.
- Partial correlation is a conditional correlation. It is a summary of the relationship between an observation with observations at prior time steps, here lag 1 is significant
- From the PACF we get the AR or p term, from ACF we get the MA or q term. So the model will be run with p= 1, q=2, and d=0 since the series is stationary.

It is important not to keep too many features which are correlated while modeling as that can create multicollinearity issues. Hence, only the relevant features will be retained. This is the result after running the ARIMA model:

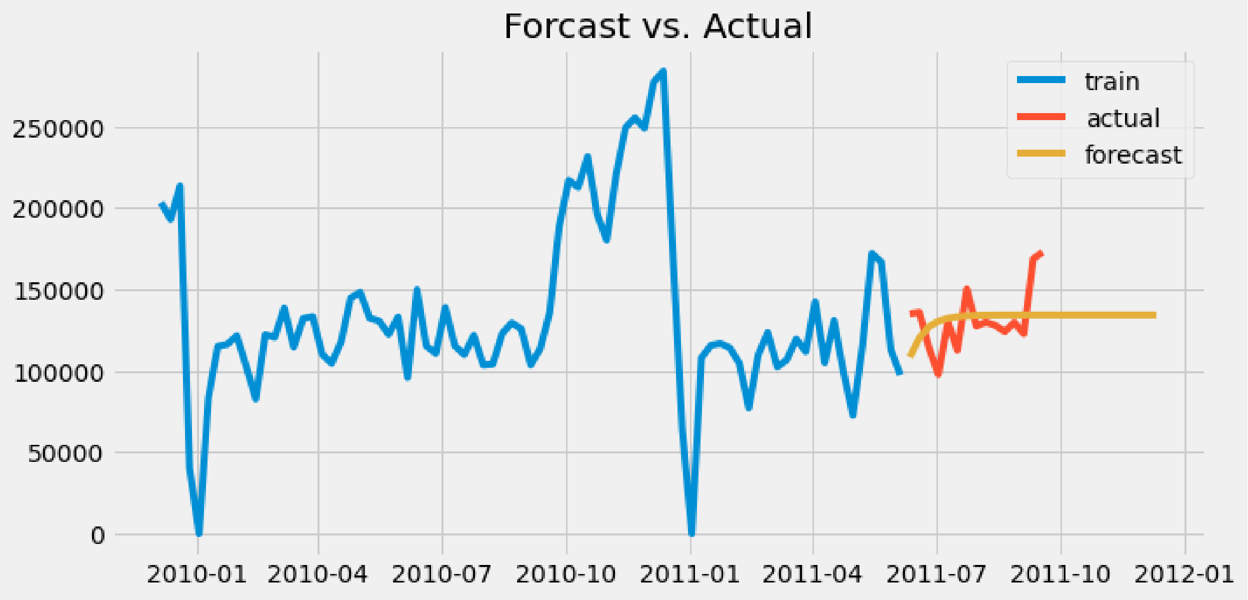


Figure 31: Arima model with p=1,q=2, d=0

Next, a function was created that will take many values of the parameters p,d and q and fit the model and then forecast and measure the root mean squared error so the model with the lowest error can be obtained. The best model had the values p=10, d=2, q=0. This is the forecasts

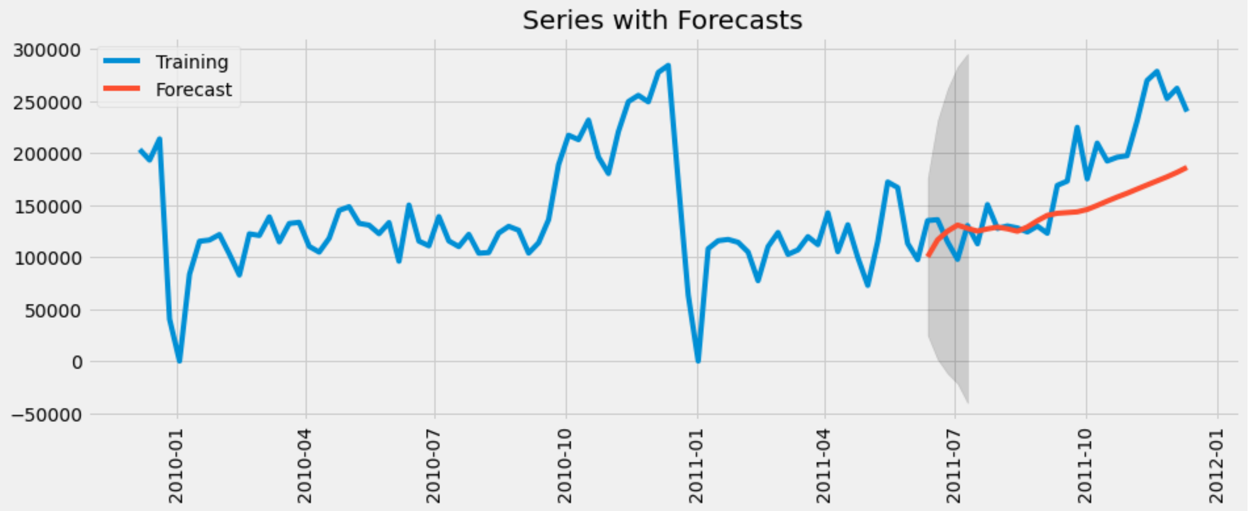


Figure 32: Sales forecasts

The sales are trending up like the sales. These are the residuals:

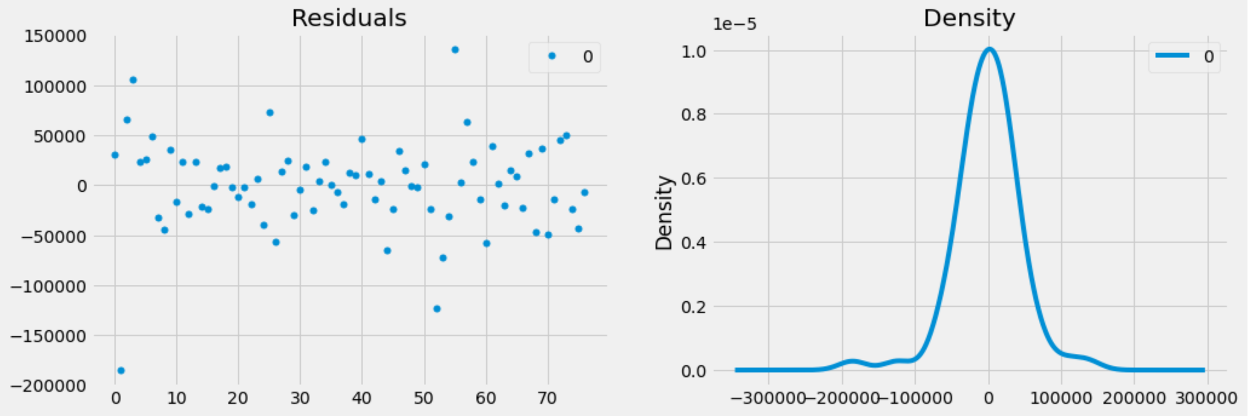


Figure 33: Residuals

The residuals are random and normally distributed.

Since the best model gave me a d value of 2 instead of 0 for stationary series, I ran another ARIMA model with the data that is differenced once, and then ran it through the iterative function to find optimum values of the parameters. Then, the differencing on the forecasts were reversed and plotted and this is the result:

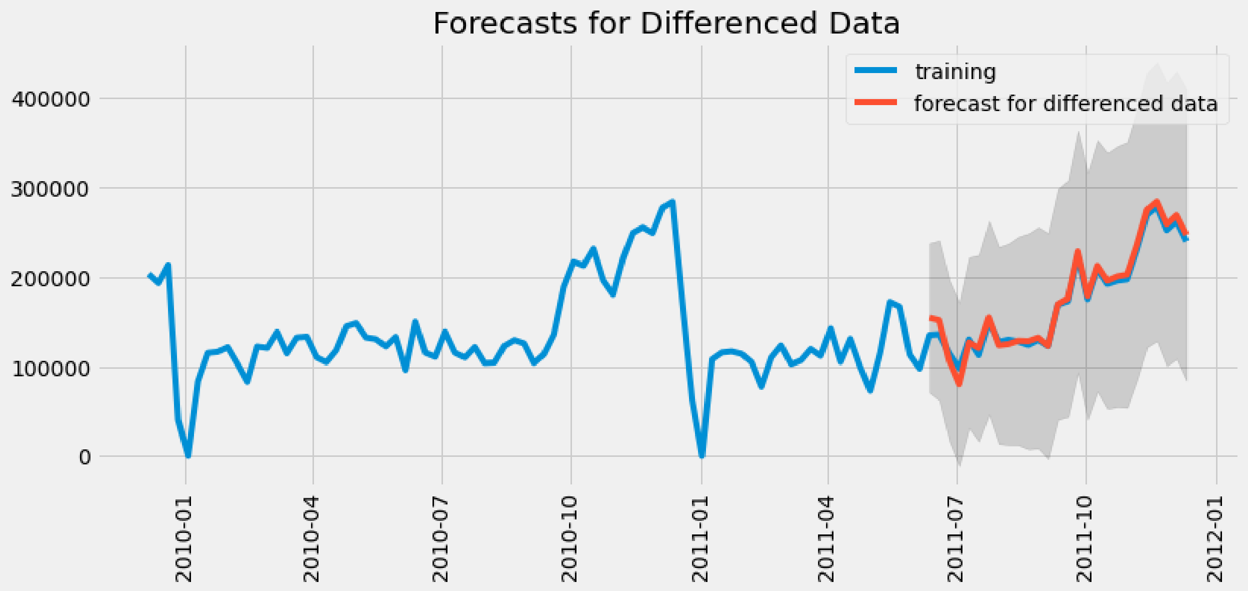


figure 34: forecasting for differenced data with p=4,d=1,q=0 and

Forecasts for the differenced model look much closer to actual values but they fail to predict when there is a sudden drop or rise, as is apparent when the red part goes down too much or up too much, but this model has a lower error and its predictions are closer to the actual values so it is a better model.