FINAL PROJECT REPORT

Forecasting Online Retail Business

GROUP-1

EXECUTIVE SUMMARY

Many small online retailers and new entrants to the online retail sector are keen to practice data mining and consumer-centric marketing in their businesses, yet technically lack the necessary knowledge and expertise to do so. For the past 10 years, we have witnessed a steady and strong increase in online retail sales. According to Interactive Media, this remarkable increase in online sales indicates that the way consumers shop for and use financial services has fundamentally changed.

Compared with traditional shopping in retail stores, online shopping has some unique characteristics and each customer's shopping process, and activities can be tracked instantly and accurately each customer's order is usually associated with a delivery and billing address, and each customer has an online store account with essential contact and payment information. These desirable, special online shopping characteristics have enabled online retailers to treat each customer as an individual with personalized understanding of each customer and to build upon customer-centric business intelligence. More than 60% of the business's total sales in recent years have come from its most valuable customers, while only 4% of those sales came from its least valuable customers. It is crucial to learn more about the products that each of these consumer groups has purchased, the products that have been bought together most frequently, and the order in which the products have been bought. By investigating the connections between different consumer groups and the things they have purchased, the firm may better understand its customers. The link can be investigated at the product/item level as well as the product category level. Many of the company's customers were institutional customers who bought large quantities of a given commodity every transaction. The firm will benefit from looking at what things, and what kinds of products, they typically buy during certain times of the year (seasons). It will also be interesting to examine if there are any disparities in the shopping habits of different consumer kinds, such as organizational and individual customers. In the long run, it is very beneficial for the business to keep track of the diversity of the most diverse client group and to forecast which consumer may join the most or least lucrative group. Finding useful predictors or indicators for such forecasts is quite valuable.

PROBLEM DESCRIPTION

The online retail company faces several challenges that need to be addressed in order to improve its performance and increase its profitability. Some of the main problems faced by the company include inaccurate demand forecasting, stockouts, excess inventory, inefficient supply chain management, and poor customer experience. These problems result in lost sales, increased costs, and decreased customer satisfaction, which ultimately affect the company's bottom line.

The business must adopt a data-driven strategy that makes use of machine learning algorithms to classify and predict clusters based on quantity, customer ID, country, product type, color type, revenue, predicting the quantity column, and precisely forecasting the quantity demand of the products in order to address these issues. The online retail company can increase its performance and accomplish its goals by resolving these difficulties. In this project, some of the machine learning models will be created, tested, and validated using historical sales data. The model's performance will then be assessed based on how well it increases sales (quantity).

BRIEF DESCRIPTION OF THE DATA

This is a transnational data set that contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retailer. The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers.

Table 1: Variables in the customer transaction dataset (541909 instances)

Variable Name	Data type	*Description; typical values and meanings*
InvoiceNo	Object	Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
StockCode	Object	Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
Description	Object	Product (item) name. Nominal.
Quantity	Int64	The quantities of each product (item) per transaction. Numeric.
InvoiceDate	Object	Invoice Date and time. Numeric, the day and time when each transaction was generated.
UnitPrice	Float64	Unit price. Numeric, Product price per unit in sterling.
CustomerId	Float64	Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
Country	Object	Country name. Nominal, the name of the country where each customer resides.

PREPROCESSING OF THE DATA

Preprocessing of data refers to the steps taken to transform raw data into a format that is suitable for analysis. The goal of preprocessing is to improve the quality of the data by removing any irrelevant or redundant information, dealing with missing values, transforming categorical variables into numerical ones, and handling outliers. The preprocessing steps may include cleaning the data by removing errors or inconsistencies, such as duplicates or missing values. This can involve techniques such as imputation, dropping missing values, or filling in missing values using statistical methods. Another important aspect of preprocessing is featuring engineering, which involves creating new features or transforming existing ones to improve the performance of the analysis. This can include scaling, normalization, or encoding categorical variables. Finally, preprocessing may involve splitting the data into training, validation, and testing sets, to ensure that the model is able to generalize well to new, unseen data.

Preprocessing is a crucial step in data analysis and machine learning, as it helps to ensure that the data is of high quality and the model is able to learn meaningful patterns from the data.

Checking for actual and unique values in each attribute and for null values. Dropping null values as imputation is not possible in the "Description" and "Customer ID" columns. Creating new features from the "Description" column by performing POS tagging to extract product type and color type information. Adding new columns for product type and color type and dropping the "Description" and "Invoice Date" columns. Creating a new feature "Revenue" by multiplying the "Unit Price" and "Quantity" columns. Encoding categorical features using label encoding and changing the data type of attributes as needed. Performing train-test split of the new data frame X. Using K-prototype clustering to cluster similar items based on the new attribute "cluster number". Checking the optimal number of clusters and generating the centroids for those clusters. Generating a new attribute "cluster number" and adding it to the data frame. Adding the "Invoice Date" column back to the dataframe and performing feature engineering on it. Classifying the test data into a number of clusters. Treating cluster numbers as the target variable to match the records from the validation and testing sets with the clusters from the training set. Splitting the merged data frame into training and validation sets for classification. Using Linear SVC for classification, which gave the best result among other machine learning algorithms. Validating the model on the test data and evaluating its performance.

Support Vector Machine (SVC)

Linear SVC (Support Vector Machine) is a variant of the SVC algorithm that is specifically designed for linearly separable data. In contrast to the standard SVC, which uses a non-linear kernel function to transform the data into a higher-dimensional space, the linear SVC uses a linear kernel function to learn a hyperplane that separates the classes. The goal of linear SVC is to find the optimal hyperplane that separates the data with the maximum margin between the two classes. It works by identifying the support vectors, which are the data points closest to the decision boundary and using them to define the hyperplane. The algorithm then minimizes the margin between the classes, subject to the constraint that the data points are correctly classified.

Linear SVC has several advantages over other linear classification algorithms, such as logistic regression, in terms of performance and interpretability. It can handle high-dimensional data and is robust to outliers. In addition, the learned hyperplane can be easily interpreted in terms of the importance of each feature. To use linear SVC in practice, the data must be preprocessed, and hyperparameters such as the regularization parameter and the maximum number of iterations must be tuned. The performance of the model should be evaluated using metrics such as accuracy, precision, recall, and F1 score.

- In our model the performance evaluation of the SVC is 88.8%.

CONCLUSION

In this project, we applied K-prototype clustering to identify distinct clusters based on both categorical and numerical features. We used linear SVC, KNN, and logistic regression to classify the clusters, achieving an accuracy score of 0.97 in the classification report of logistic regression. We also explored Random Forest Regression and Gradient Boosting Regression to predict the target variable (revenue) and found that Gradient Boosting Regression had the best performance.

Holt-Winters approach-based Exponential Smoothing method is an effective way for predicting sales data. 80% of the dataset was used to train the model, while the remaining 20% was used to test it. Both the MAE and RMSE assessment scores were relatively low, showing that the model is capable of producing reliable predictions. Overall, the Exponential Smoothing method can be a powerful tool for time series forecasting, especially when applied to online retail sales data. While our results were promising, further research could be done to explore other models and feature selection techniques to further improve performance.

REFERENCES

- https://archive.ics.uci.edu/ml/datasets/online+retail
- https://www.researchgate.net/publication/263376827_The_evolution_of_direct_data_and_digital_marketing
- https://dl.acm.org/doi/abs/10.1145/2640087.2644161
- https://www.sciencedirect.com/science/article/pii/S0957417414008008?via%3Dihub

APPENDIX

Table 1: Variables in the customer transaction dataset (541909 instances)

Variable Name	Data type	*Description; typical values and meanings*
InvoiceNo	Object	Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
StockCode	Object	Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
Description	Object	Product (item) name. Nominal.
Quantity	Int64	The quantities of each product (item) per transaction. Numeric.
InvoiceDate	Object	Invoice Date and time. Numeric, the day and time when each transaction was generated.
UnitPrice	Float64	Unit price. Numeric, Product price per unit in sterling.
CustomerId	Float64	Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
Country	Object	Country name. Nominal, the name of the country where each customer resides.

Figure-1: Variables in the customer transaction data

Label encoding of categorical features

```
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()

for col in ["InvoiceNo", "StockCode", "CustomerID","Country", "Product Type","Colour_type"]:
  X[col] = label_encoder.fit_transform(X[col])
```

```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50000 entries, 0 to 79177
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   InvoiceNo     50000 non-null  int64
 1   StockCode     50000 non-null  int64
 2   Quantity      50000 non-null  int64
 3   UnitPrice     50000 non-null  float64
 4   CustomerID    50000 non-null  int64
 5   Country       50000 non-null  int64
 6   Product Type  50000 non-null  int64
 7   Colour_type   50000 non-null  int64
 8   Revenue       50000 non-null  float64
dtypes: float64(2), int64(7)
memory usage: 3.8 MB
```

Figure-2: Label Encoding

```
import matplotlib.pyplot as plt
from kmodes.kprototypes import KPrototypes

cost = []
for num_clusters in list(range(2,5)):
    kproto = KPrototypes(n_clusters=num_clusters, init='Cao')
    kproto.fit_predict(train, categorical=[4,5,6,7])
    cost.append(kproto.cost_)
    labels=kproto.labels_
plt.plot(cost)
```

```
[<matplotlib.lines.Line2D at 0x7fb9dc9c62e0>]
```

Figure-3: K-prototype clustering

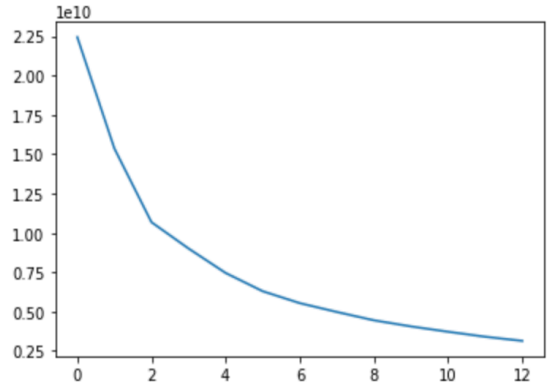


Figure-4: Elbow curve for the k-prototype clustering

Classification of test data into number of clusters

Cluster numbers were treated as a target variable as the objective

was to match the records from the validation and testing sets with the clusters from the training set. ¶

Splitting of merdedDF dataframe into traina nd validation

```
from sklearn.model_selection import train_test_split
train_, val_= train_test_split(mergedDf, train_size = 0.8, random_state = 0)
```

```
train_y=train_["Cluster number"]
train_x=train_.drop(['Cluster number'],axis=1,inplace=False)

val_y=val_["Cluster number"]
val_x=val_.drop(['Cluster number'],axis=1,inplace=False)
```

Figure-5: Classification of the clustering

```
LinearSVC()
```

```
pred_y = model1.predict(val_x)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(val_y,pred_y)
```

```
0.888125
```

Figure-6: Accuracy of the Linear SVC Algorithm

```
LogisticRegression()
```

```
predictions = lr.predict(val_x)
```

```
accuracy = accuracy_score(val_y, predictions)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.999875
```

Figure-7: Accuracy of Logistic Regression Algorithm

```
# Print the classification report
print(classification_report(val_y, lrc))
```

```
              precision    recall  f1-score   support

           0       0.93      1.00      0.97      3254
           1       0.99      0.93      0.96      2418
           2       0.99      0.96      0.98      2328

    accuracy                           0.97      8000
   macro avg       0.97      0.96      0.97      8000
weighted avg       0.97      0.97      0.97      8000
```

Figure-8: Classification report of the Logistic Regression

```
KNeighborsClassifier(n_neighbors=4)
```

```
# Make predictions on the test data
y_pred = knn.predict(val_x)
```

```
# Calculate the accuracy of the model
accuracy = accuracy_score(val_y, pred_y)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.888125
```

Figure-9: Accuracy of KNN Algorithm

```python
from sklearn.metrics import mean_squared_error, r2_score

# Calculate mean squared error and R-squared for the predictions
mse = mean_squared_error(val_y, y_pred)
r2 = r2_score(val_y, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

```
Mean Squared Error: 96.37127009820104
R-squared: 0.7546162164816421
```

Figure-10: Mean squared error and R-squared of the Random Forest Regressor

```python
from sklearn.metrics import mean_squared_error, r2_score

mse = mean_squared_error(val_y, preds)
r2 = r2_score(val_y, preds)

print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

```
Mean Squared Error: 44.61663521662306
R-squared: 0.8863956161814895
```

Figure-11: Mean squared error and R-squared of the Gradient Boosting Regressor

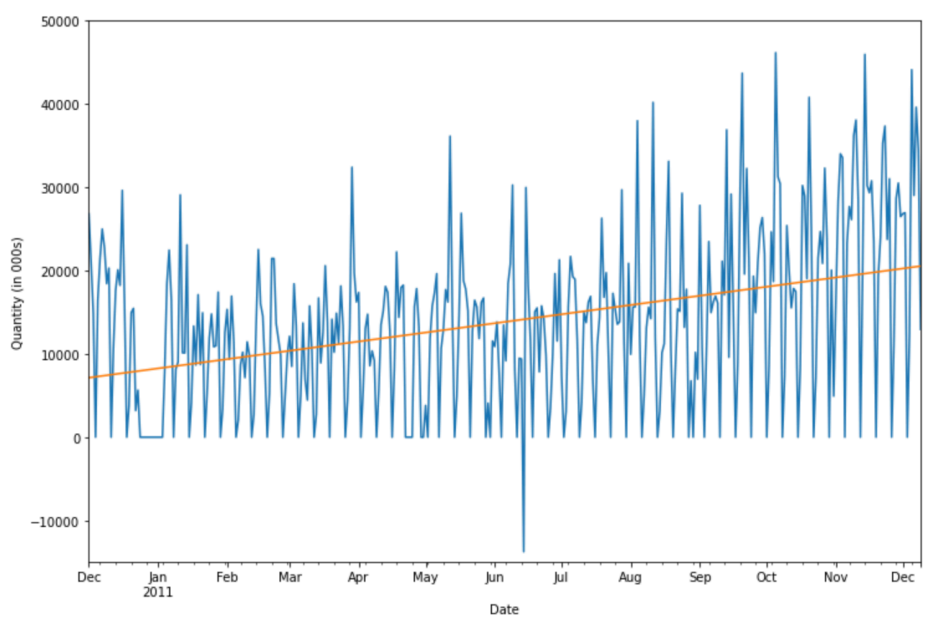


Figure-12: Fitting Trend

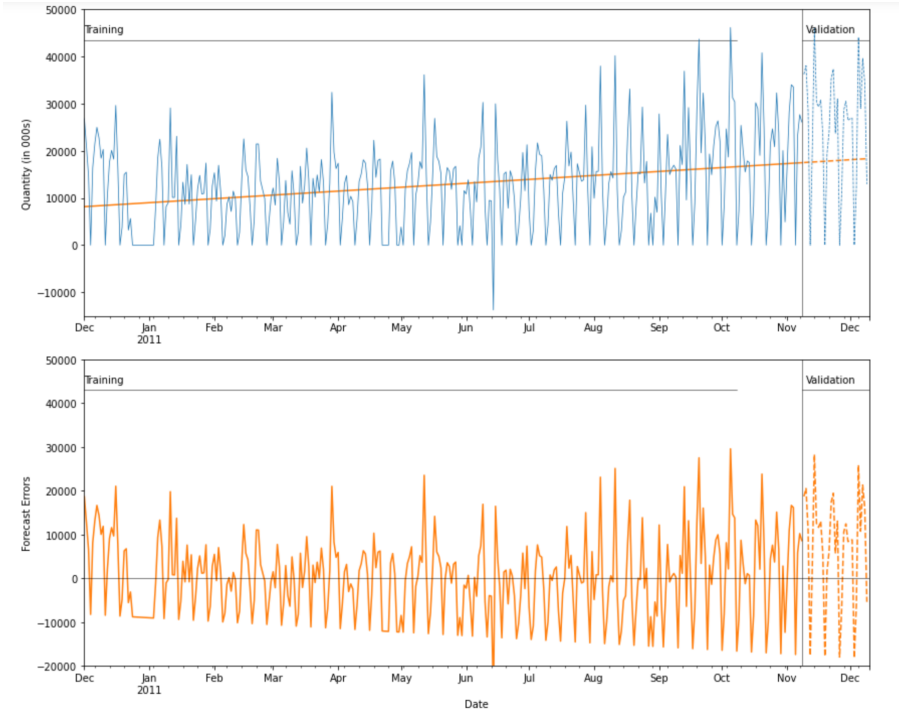


Figure-13: Linear Regression

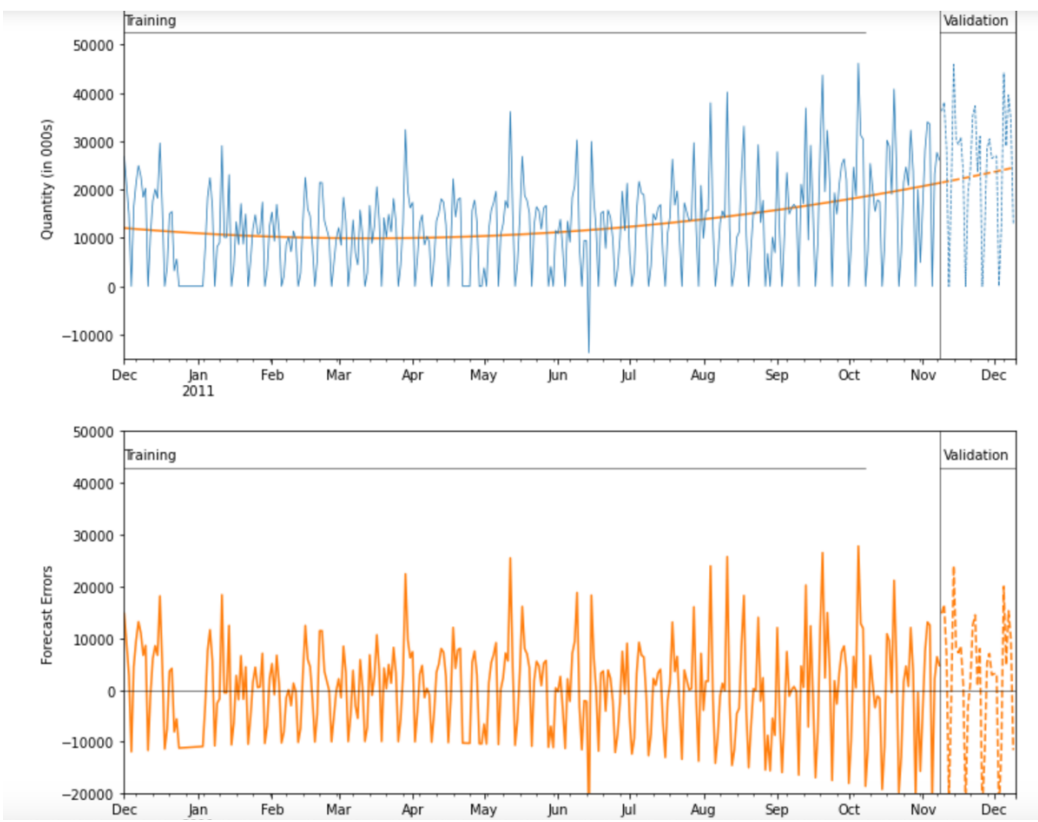
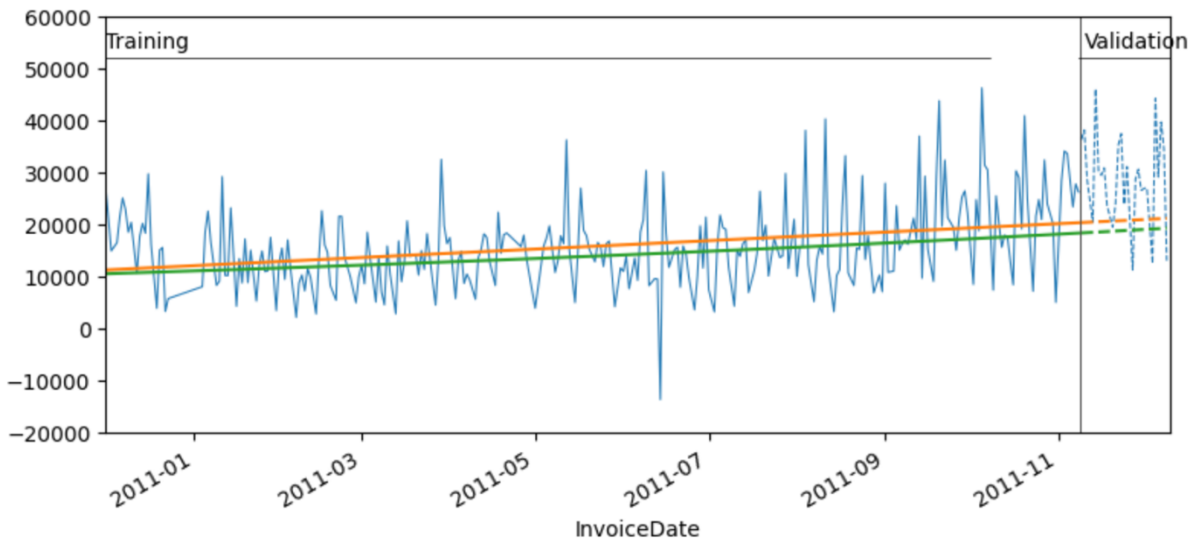


Figure-14: Polynomial Regression




```
print(retail_lm.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Quantity   R-squared:                       0.079
Model:                            OLS   Adj. R-squared:                  0.076
Method:                 Least Squares   F-statistic:                     29.31
Date:                Thu, 27 Apr 2023   Prob (F-statistic):           1.17e-07
Time:                        05:10:08   Log-Likelihood:                -3617.7
No. Observations:                 343   AIC:                             7239.
Df Residuals:                     341   BIC:                             7247.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   8134.9360    999.996      8.135      0.000    6167.998    1.01e+04
trend         27.2769      5.039      5.413      0.000      17.366      37.188
==============================================================================
Omnibus:                        4.430   Durbin-Watson:                   1.452
Prob(Omnibus):                  0.109   Jarque-Bera (JB):                4.243
Skew:                           0.269   Prob(JB):                        0.120
Kurtosis:                       3.084   Cond. No.                         398.
==============================================================================
```

Figure-15: OLS Regression Results

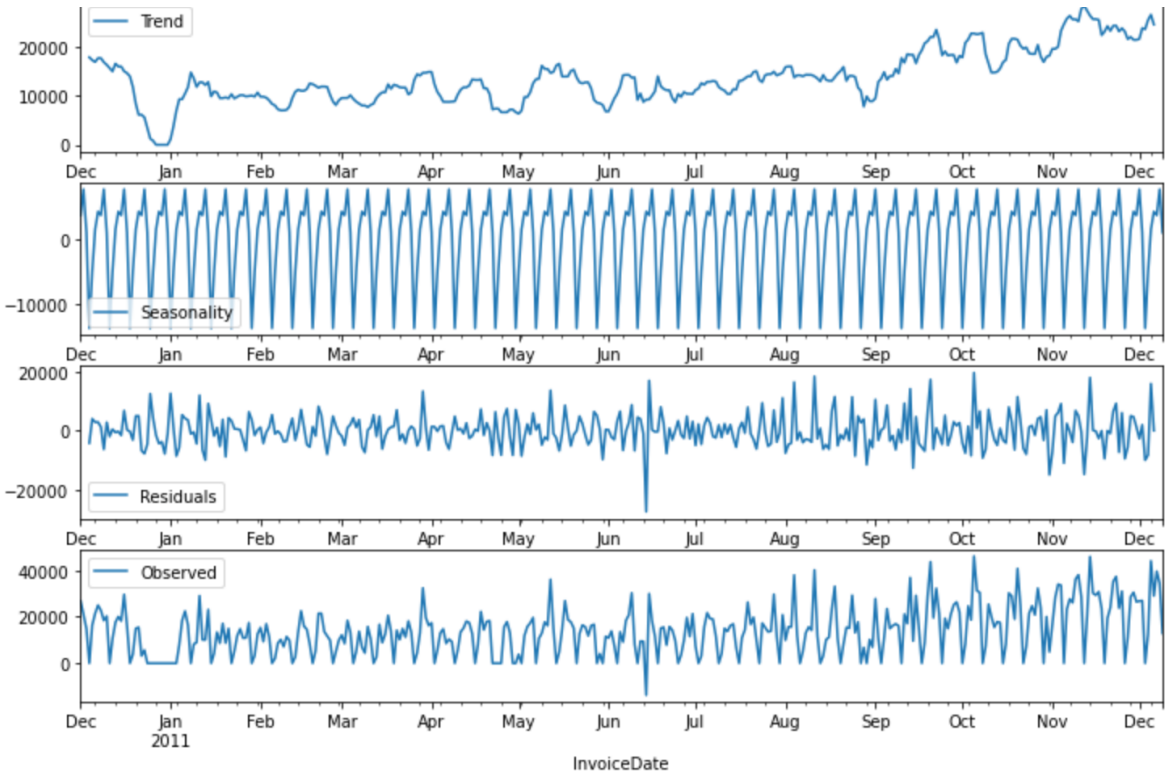
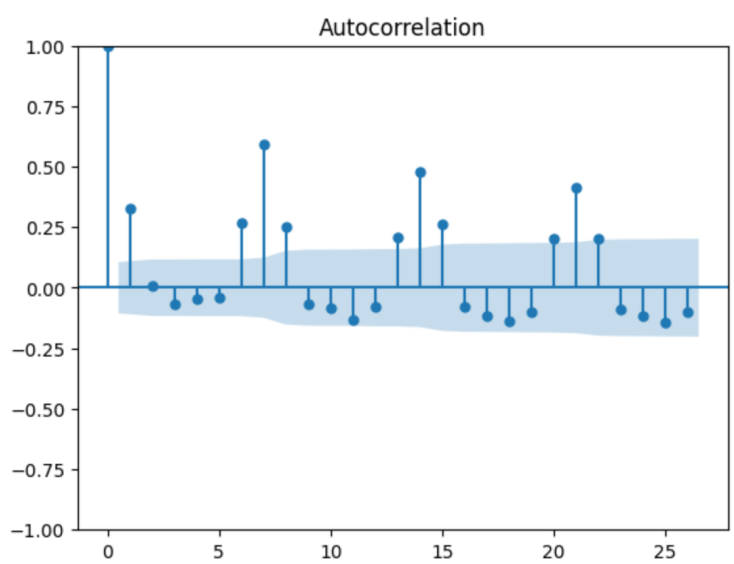


Figure-16: Seasonal Decomposition Plot




```
[ 1.          0.32539252  0.00770729 -0.07022728 -0.04751163 -0.03808945
  0.26543371  0.59379559  0.25339882 -0.06887479 -0.08494012 -0.13144002
 -0.08120279  0.21032656  0.48088654  0.26342894 -0.08025096 -0.11916777
 -0.13719967 -0.09793068  0.20332829  0.41460082  0.20068396 -0.08846814
 -0.11588134 -0.14573862]
```

Figure-17: AutoCorrelation function (ACF)

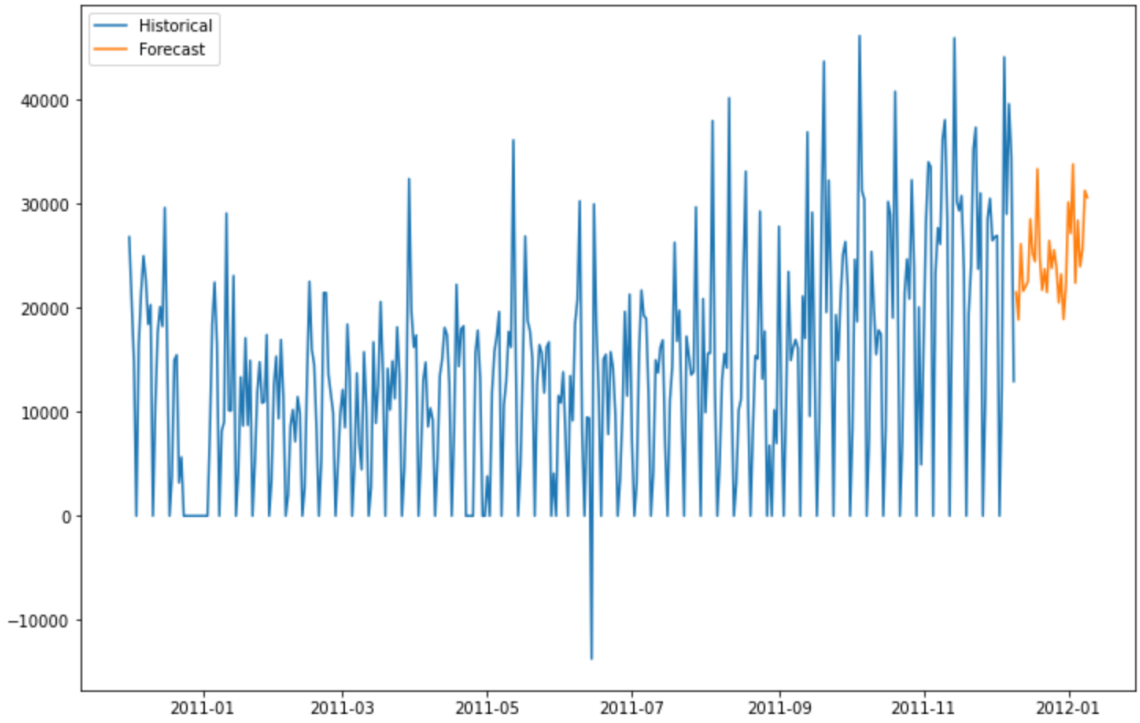


Figure-18: SARIMAX model

```
MAE: 6887.654143844463
RMSE: 8653.700743557236
```

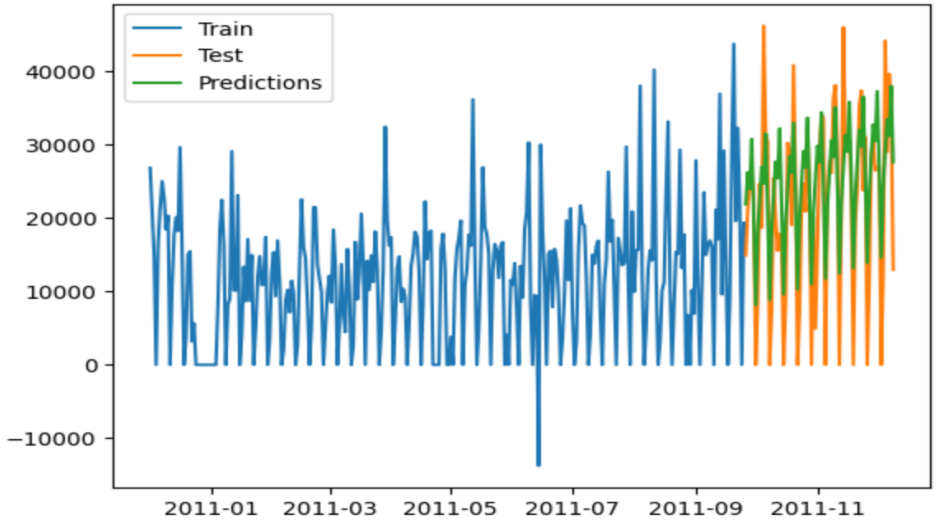


Figure-19: Exponential Smoothing