Zewail City of Science and Technology ZEWAIL CITY ESTABLISHED 2000 INAUGURATED 2011 مدينة زويل للعلوم والتكنولوجيا

Project Name: Sales Forecasting and Demand Prediction

Name	ID
Reem Ehab Helmy	202201373
Tasneem Ashraf Elsaadany	202201573
Mariam Mohamed Goda	202201223

- **Explaination Of All Progress:-**
- **Data Preprocessing:**
- **Removal of Identifiers and Descriptive Columns:** Columns like 'Row ID', 'Order ID', 'Customer ID', 'Customer Name', 'Product ID', and 'Product Name' were dropped.These columns do not help with predicting sales or profit and may even introduce noise.
- **Date Feature Engineering:** The 'Order Date' and 'Ship Date' columns were converted to datetime objects. From the 'Order Date', new 'Order Year' and 'Order

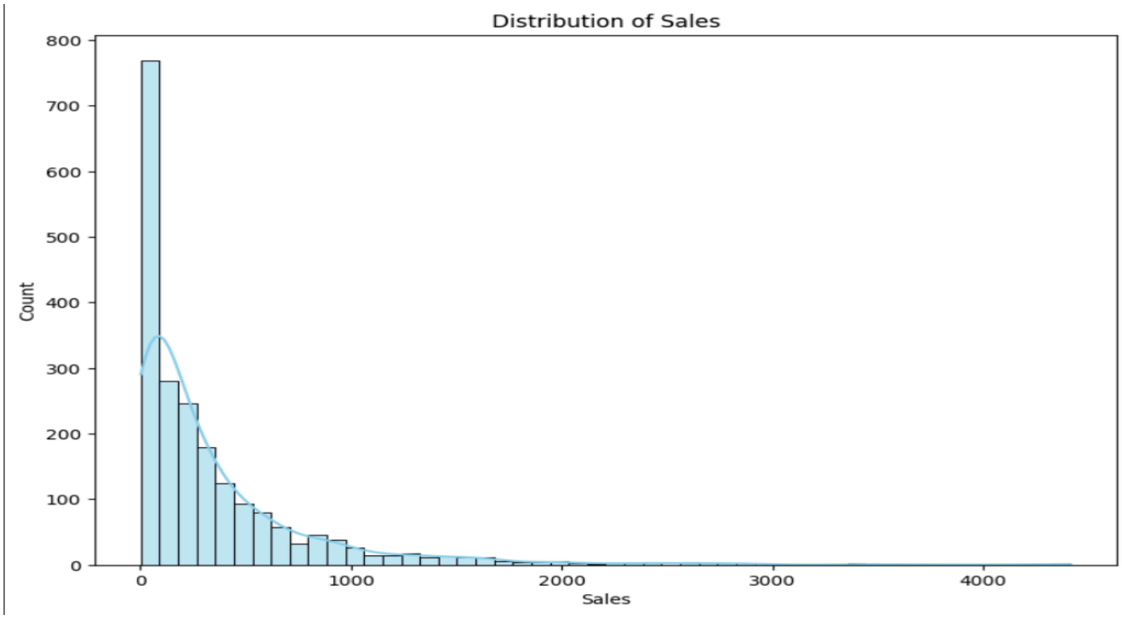
Month' features were extracted. Additionally, the 'Shipping Days' (the difference between ship and order dates) was calculated. The original 'Order Date' and 'Ship Date' columns were then removed, indicating a focus on the derived temporal features rather than the exact dates.

- **Categorical Feature Encoding:** All columns with a string data type were converted into numerical representations using Label Encoding. to handle categorical data that require numerical input.

- <u>**EDA Techniques:**</u>
- **distribution of sales:**

1. The KDE plot shows a strongly right-skewed distribution, with most sales focused at the lower end of the sales range.

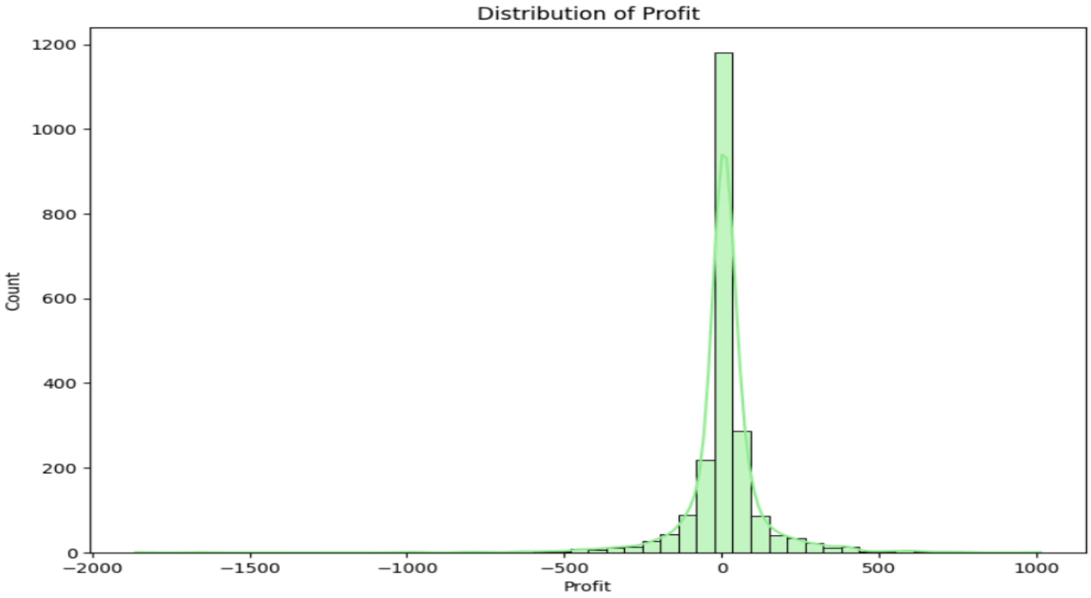
2. The long right tail suggests high-value orders. A single important peak at the lower end indicates the most frequent sales amount lies within that range.



- ## Distribution of Profit:

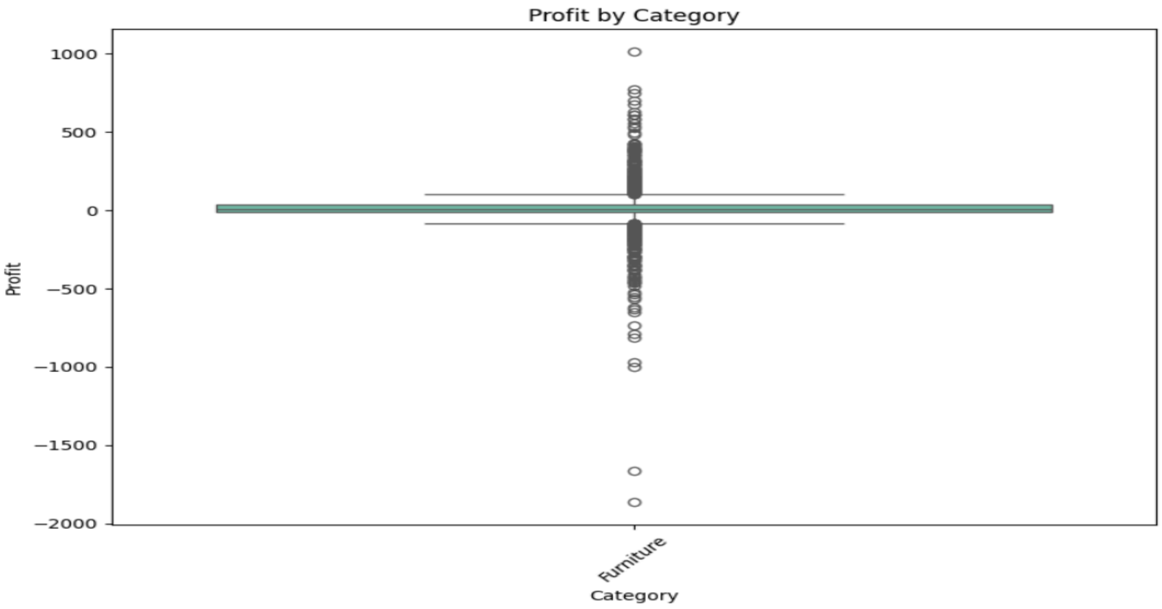
1. The distribution of transactions is centered around zero, with a high peak around 0 indicating minimal profit or small loss.
2. The distribution has a bimodal tendency, with smaller shoulders on either side indicating small profits or losses. As transactions move away from zero, the frequency drops sharply, with thin tails indicating less common large profits or

losses. The distribution may exhibit slight skewness, but it extends into the negative profit range.

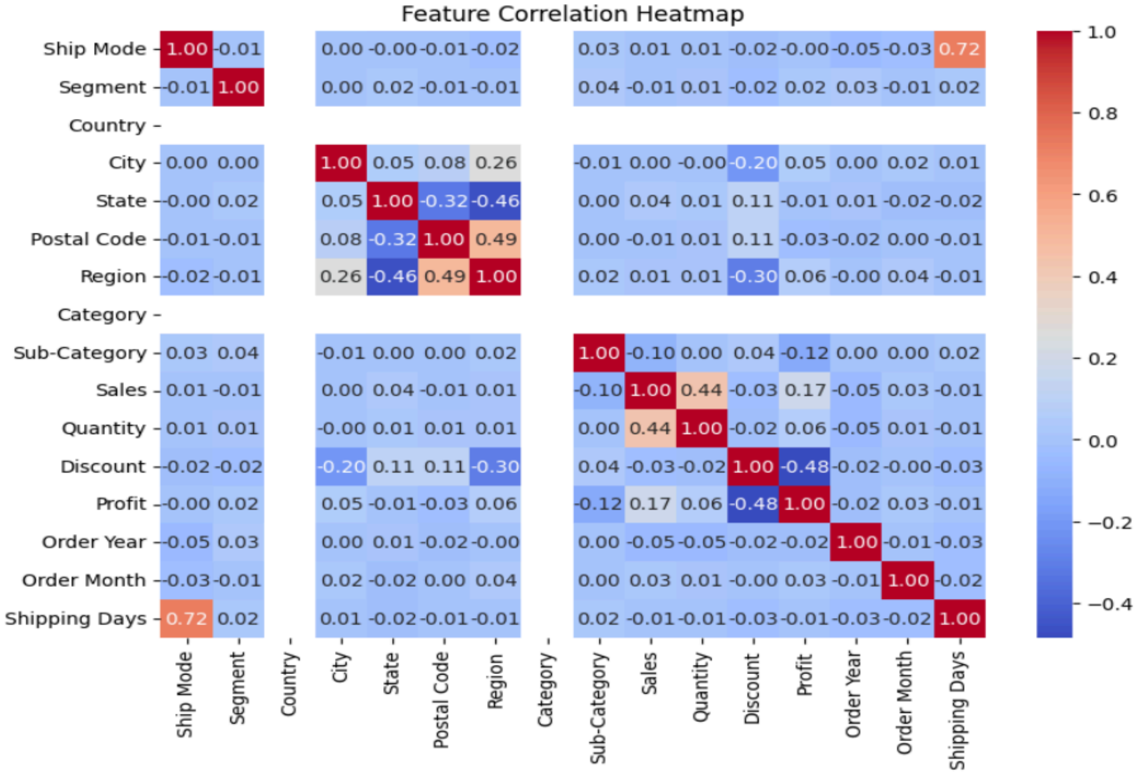


- **Profit by category:**

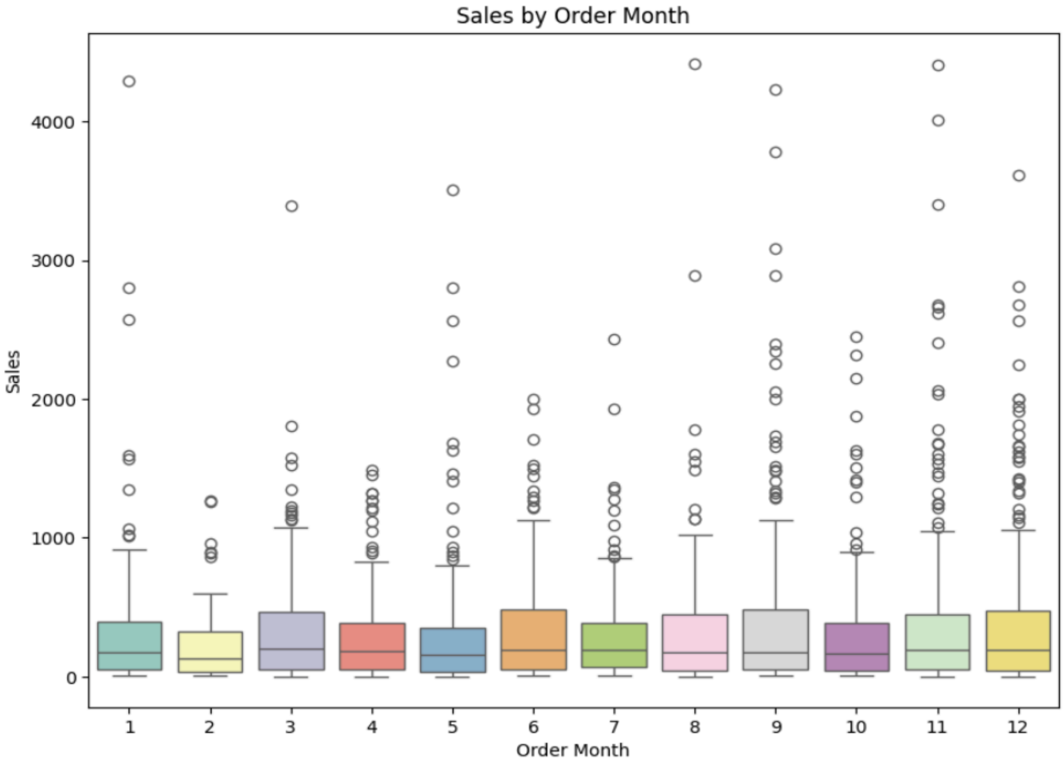
1. The median profit for furniture transactions is slightly below zero, indicating that half of them resulted in a loss or very low profit. The median ranges from negative to positive, suggesting a relatively small profit/loss range.
2. The majority of furniture transactions are in the negative profit region, suggesting average furniture sales may not be very profitable. Outliers indicate high and substantial losses. The data spread is excluding extreme outliers.



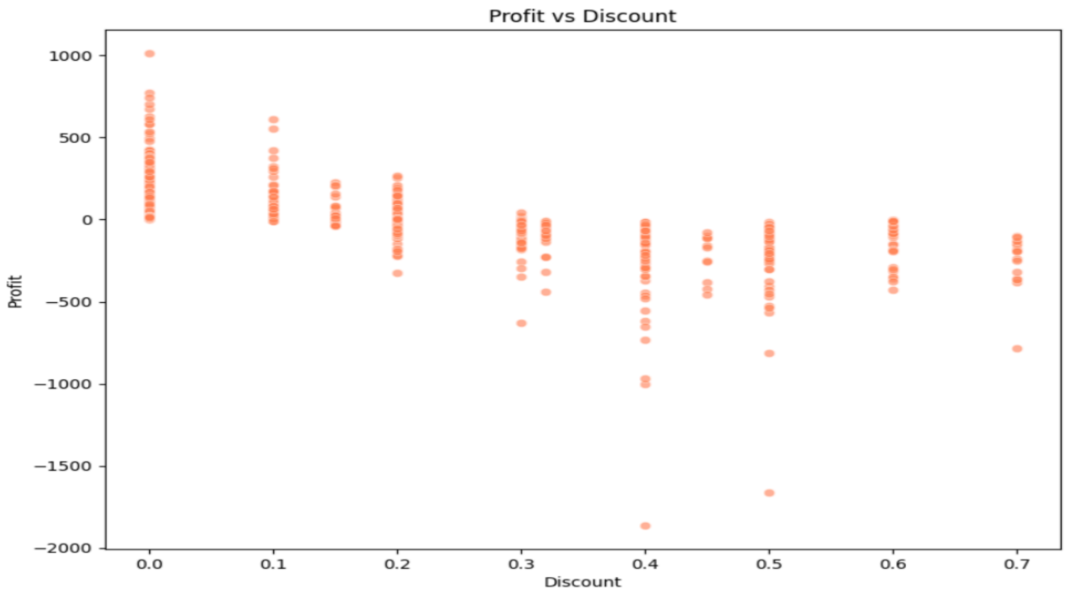
- **Correlation Heatmap:**
- The strong positive correlation between 'Shipping Days' and 'Order Year' and the moderate correlations between ('Sales', 'Profit'), ('Quantity', 'Sales'), and ('Discount', 'Profit') are the most prominent relationships revealed by this visualization. Most other feature pairs show weak linear correlations.



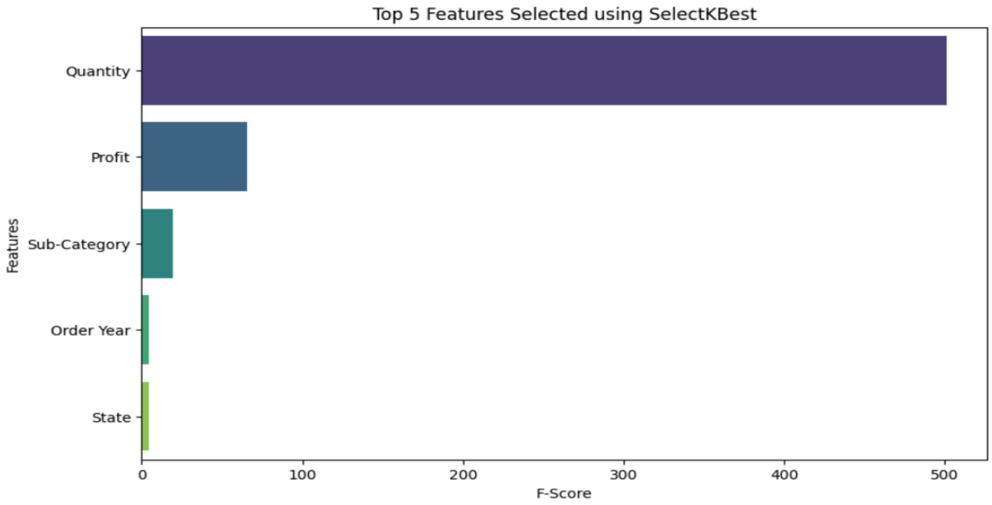
- **Sales by Order Month:**
- The median sales value is consistent across most months, with slight variations. The interquartile range (IQR) indicates the spread of the middle 50% of sales within each month. The overall spread of data is consistent, excluding outliers. Outliers are present in almost every month, indicating some orders with higher sales values than the majority.



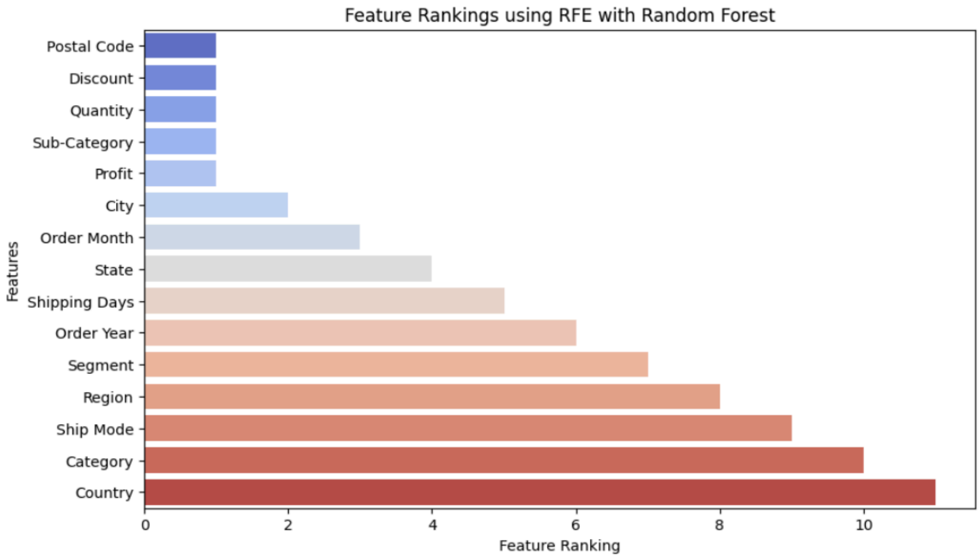
- **Profit Vs Discount:**
- The study reveals a general negative trend, with higher discounts resulting in decreased profit. Low discounts are concentrated, resulting in a wide range of profit values. As discounts increase, the cluster of points shifts downwards, indicating lower profit values. Lower discounts have a wider spread, suggesting other factors influence final profit. Higher discounts lead to more consistent, lower profits, with smaller ranges. Outlier points show transactions with high profits at lower discount levels.



- **Feature Selection:**
- **F-scores:**
- The selection method identifies three key features: Quantity, Profit, Sub-Category, Order Year, and State. Quantity has the strongest linear relationship with the target variable (y), Profit has the second-highest F-score (75), Sub-Category has a lower F-score (25), Order Year has a weaker relationship (around 5) and State has the lowest F-score (around 2).

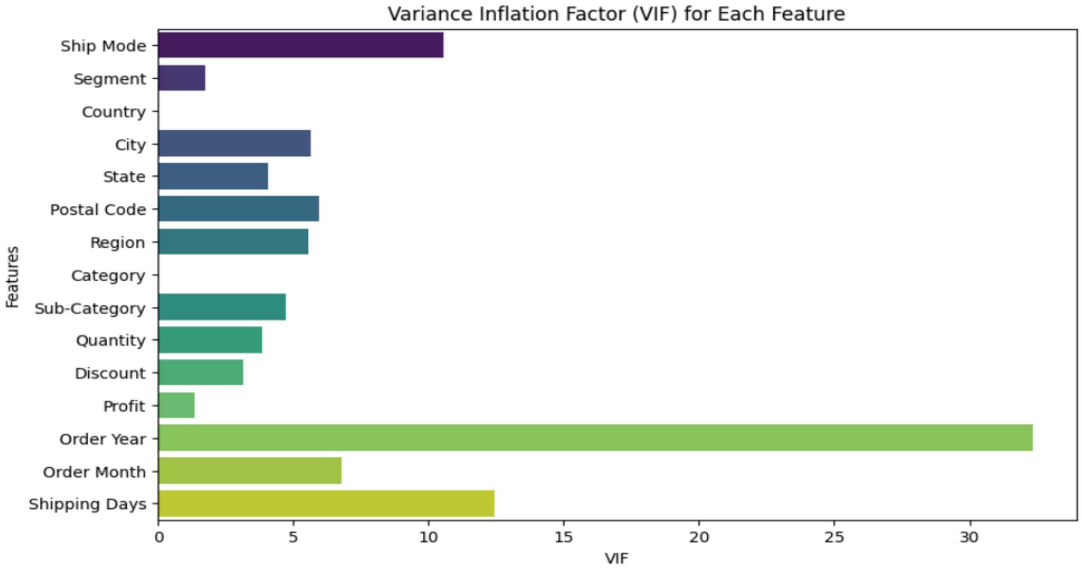


- **RFE:**
- The RFE process ranks features based on their importance, with Country being the most important feature. Other important features include Category, Ship Mode, Region, Segment, Order Year, Shipping Days, State, Order Month, City, Profit, Sub-Category, Quantity, Discount, and Postal Code. Each feature has a different ranking, indicating its importance in the RFE process.

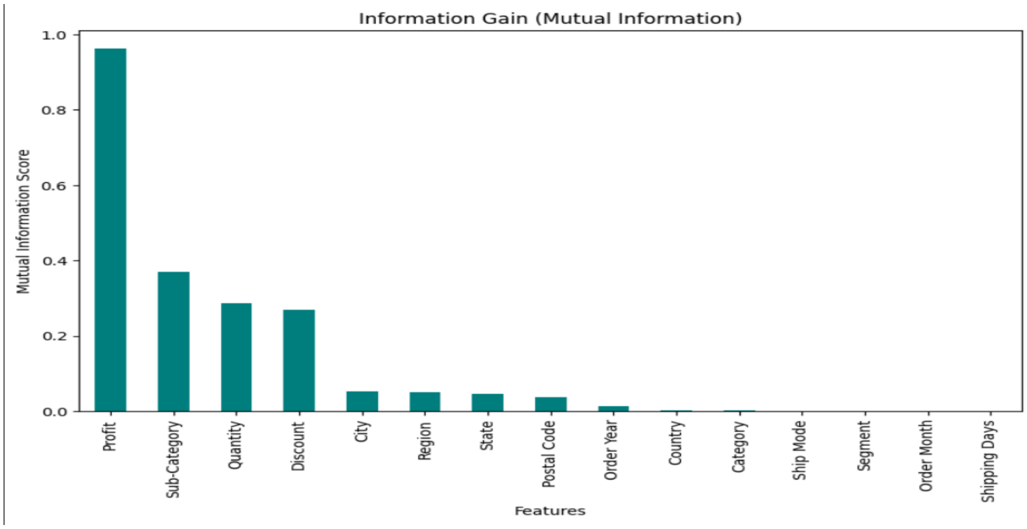


- **VIF:**

- The data analysis features show varying levels of multicollinearity, with high VIF values indicating a strong linear relationship with other predictor variables. Other features, such as shipping days, month, city, state, postal code, ship mode, segment, country, region, category, sub-category, quantity, discount, and profit, have low VIF values, indicating minimal multicollinearity with other predictors. These features are considered thresholds for concern regarding multicollinearity. Overall, these features indicate minimal multicollinearity with other predictors.

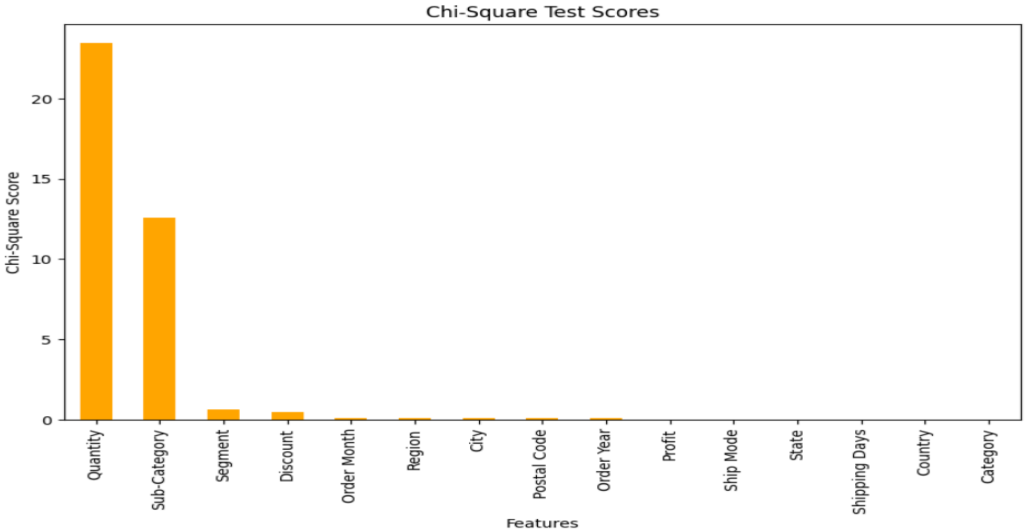


- **Feature Engineering:**
- **Predictive modeling:**
- this plot reveals that 'Profit', 'Sub-Category', 'Quantity', and 'Discount' are the features that provide the most information about the target variable according to the mutual information metric. The other features have considerably lower scores.



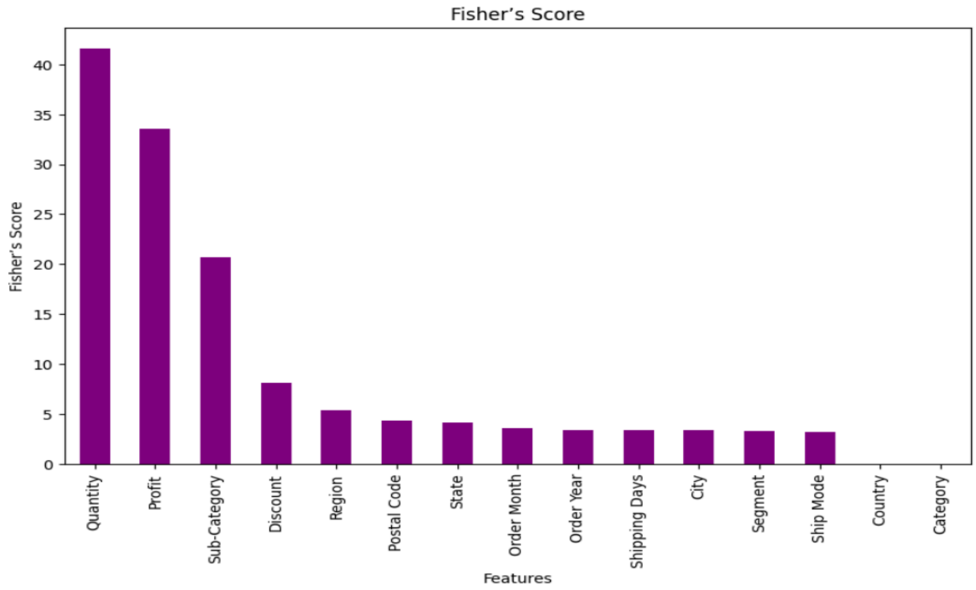
- **Transforming Data:**

- this plot suggests that 'Quantity' and 'Sub-Category' have the strongest relationship with whether the target variable y is above or below its median, based on the Chi-Square test. The other features show very little statistical dependence with this binarized target variable according to this method.



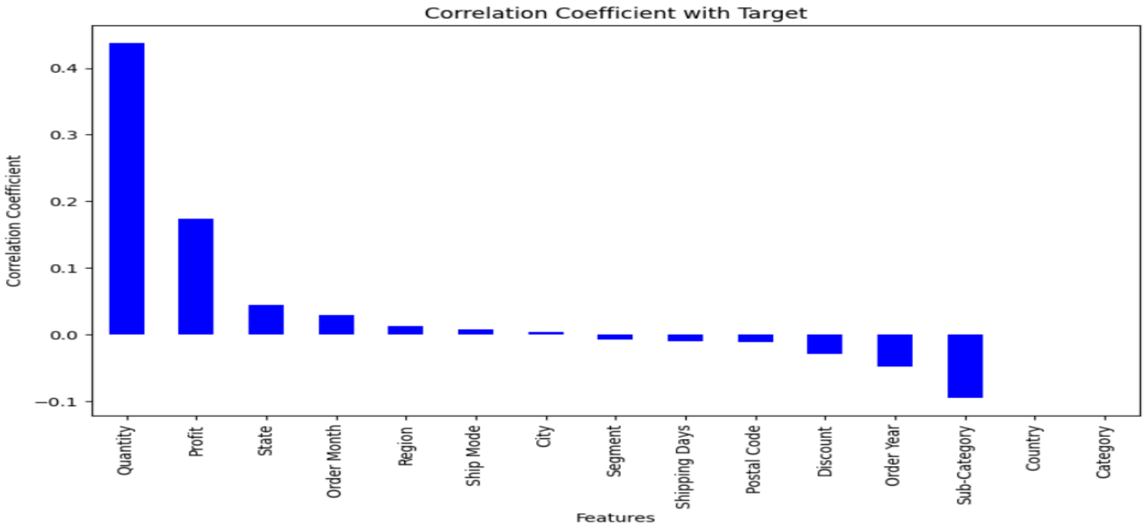
- **Fisher's Score:**

- this plot suggests that 'Quantity', 'Profit', 'Sub-Category', and 'Discount' are the most effective features in distinguishing between the different classes of your target variable `y`, as measured by the Fisher's score. The other features have considerably lower scores, indicating a weaker ability to separate the classes.



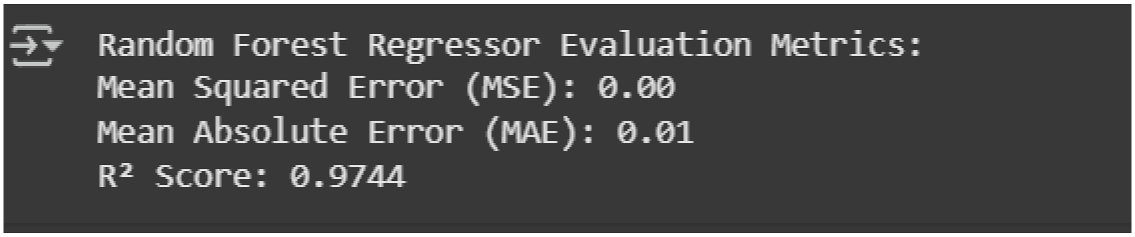
- **Correlation coefficient:**

- this plot reveals that 'Quantity' has the strongest positive linear relationship with the target variable, followed by 'Profit'. 'Discount' and 'Sub-Category' show the most noticeable negative linear correlations. The remaining features exhibit very weak or near-zero linear correlations with the target.



- **Machine Learning Models Building:**
- **Random Forest_(Reem)_:**
- The Random Forest Regressor was chosen for this dataset due to its ability to model complex, non-linear relationships (like Quantity, Discount, Category, Time-based features, etc.) and the target variable its robustness to outliers and its tendency to avoid overfitting.
- **Mean Squared Error (MSE): 0.00:** This is the average of the squared differences between the predicted values and the actual values in my test set. An MSE of 0.00 suggests that, on average, the squared difference between the model's predictions and the true values is virtually zero. This indicates extremely accurate predictions with very little error.

- **Mean Absolute Error (MAE): 0.01:** This is the average of the absolute differences between the predicted values and the actual values in my test set. An MAE of 0.01 means that, on average, the model's predictions are off by only 0.01 units from the actual values. This also points to very high accuracy.
- **R^2 Score: 0.9744:** This is the coefficient of determination, representing the proportion of the variance in the target variable that is predictable from the independent variables by the model. An R^2 score of 0.9744 means that approximately 97.44% of the variance in my target variable in the test set is explained by the Random Forest model. This is an exceptionally high R^2 score, indicating a very strong fit of the model to the data.

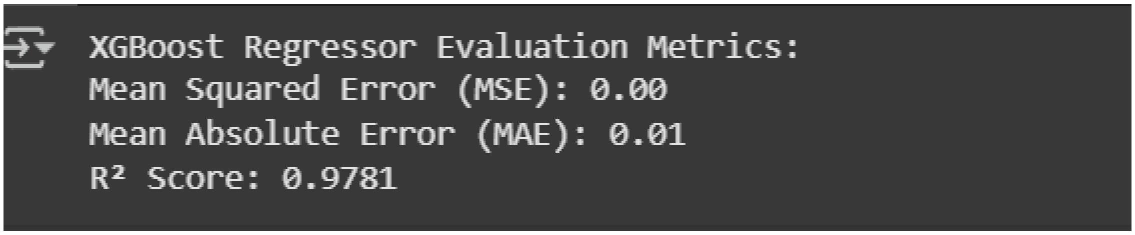


```
Random Forest Regressor Evaluation Metrics:
Mean Squared Error (MSE): 0.00
Mean Absolute Error (MAE): 0.01
R² Score: 0.9744
```

- **XGBoost_(Reem)_:**

- The XGBoost model, a powerful boosting ensemble method well-suited for our tabular data due to its ability to capture complex relationships and prevent overfitting through regularization, was evaluated on a held-out test set.

- **Mean Squared Error (MSE): 0.00:** An MSE of 0.00 suggests that the average of the squared differences between the predicted values and the actual values in my test set is virtually zero. This implies highly accurate predictions with minimal error.
- **Mean Absolute Error (MAE): 0.01:** An MAE of 0.01 indicates that, on average, the absolute difference between the model's predictions and the true values in my test set is only 0.01 units . This also points to a very high level of accuracy.
- **R² Score: 0.9781:** An R² score of 0.9781 means that approximately 97.81% of the variance in my target variable in the test set is explained by the XGBoost model. This is an exceptionally high R² score, signifying a very strong fit of the model to the data.

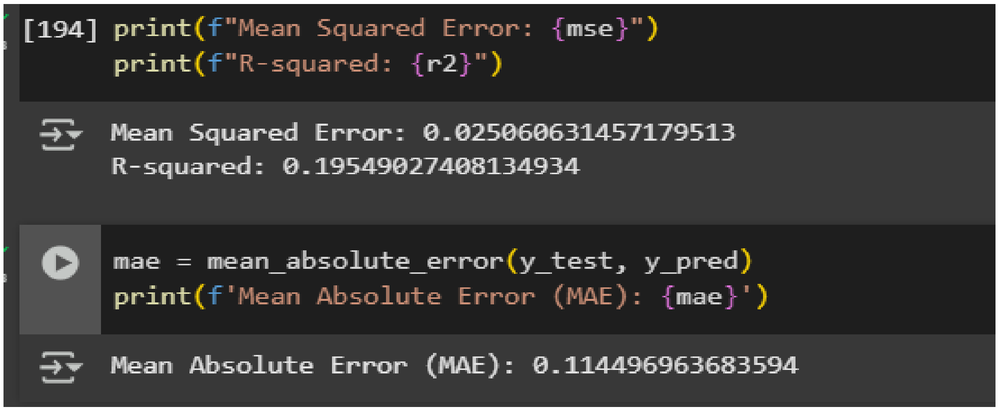


```
XGBoost Regressor Evaluation Metrics:
Mean Squared Error (MSE): 0.00
Mean Absolute Error (MAE): 0.01
R² Score: 0.9781
```

- **Support Vector Regression (SVR) Model Evaluation(Tasneem):**
- **Mean Squared Error (MSE): 0.02506:** The MSE represents the average of the squared differences between the predicted and actual target values. In this case, a lower MSE indicates that the model is performing relatively well, with small errors. However, the MSE value needs to be interpreted in the context of the

scale of the target variable. A value of 0.02506 suggests that the model's errors are relatively small.

- **Mean Absolute Error (MAE): 0.1145:** The MAE measures the average magnitude of errors in the model’s predictions, in the original units of the target variable. In this case, the MAE of 0.1145 implies that, on average, the model's predictions deviate by 0.1145 units from the true values. This indicates a relatively small deviation, but the significance depends on the target variable's range.
- **R^2 Score: 0.1954:** The R^2 score, also known as the coefficient of determination, indicates the proportion of variance in the target variable explained by the model. An R^2 score of 0.1954 suggests that approximately 19.54% of the variance in the target variable is explained by the SVR model. While this indicates some predictive power, it also shows that the model has room for improvement.



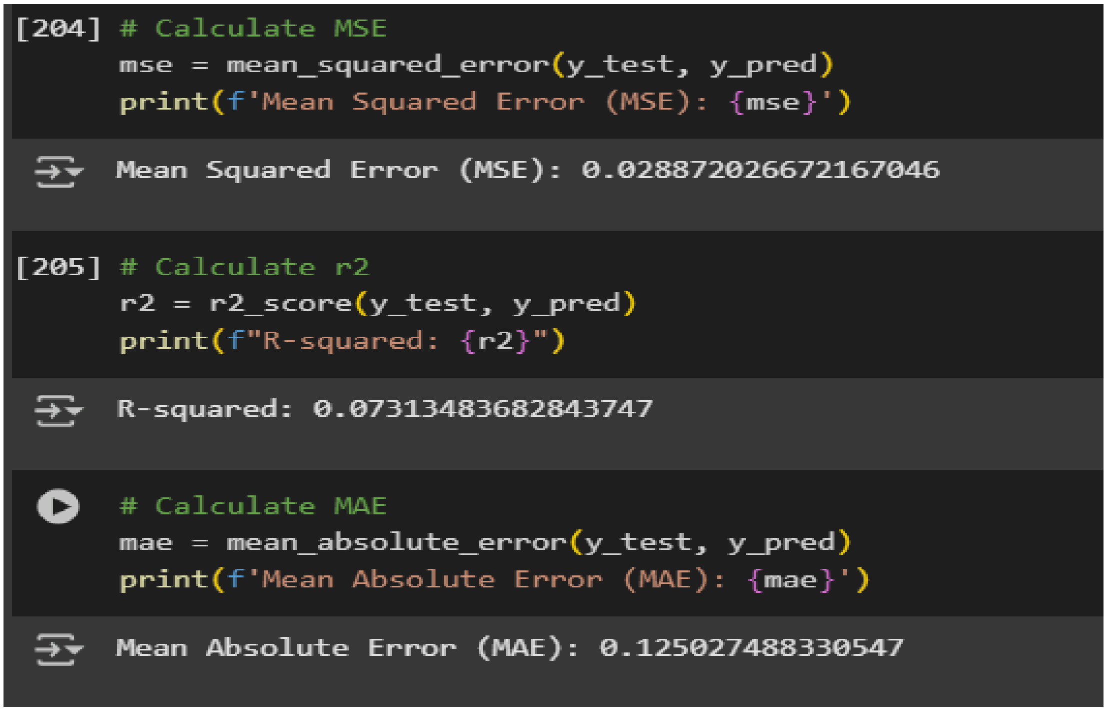
```
[194] print(f"Mean Squared Error: {mse}")
      print(f"R-squared: {r2}")

      Mean Squared Error: 0.025060631457179513
      R-squared: 0.19549027408134934

      mae = mean_absolute_error(y_test, y_pred)
      print(f'Mean Absolute Error (MAE): {mae}')

      Mean Absolute Error (MAE): 0.114496963683594
```

- **Extreme Learning Machines (ELM) Model Evaluation(Tasneem):**
- **Mean Squared Error (MSE): 0.02888:** The MSE represents the average of the squared differences between the predicted and actual target values. In this case, the MSE of 0.02888 suggests that the model is performing fairly well, with the squared error relatively small. A lower MSE generally means better performance, but the value should be interpreted within the context of the scale of the target variable.
- **Mean Absolute Error (MAE): 0.1250:** The MAE provides the average magnitude of errors in the model's predictions, expressed in the original units of the target variable. The MAE of 0.1250 implies that, on average, the model's predictions deviate by 0.1250 units from the true values. This deviation seems relatively small, but the impact of this error depends on the scale of the target variable.
- **R^2 Score: 0.0731:** The R^2 score measures how well the model explains the variance in the target variable. An R^2 score of 0.0731 indicates that approximately 7.31% of the variance in the target variable is explained by the ELM model. This is relatively low, suggesting that the ELM model is not capturing much of the underlying pattern in the data.



```
[204] # Calculate MSE
      mse = mean_squared_error(y_test, y_pred)
      print(f'Mean Squared Error (MSE): {mse}')

      Mean Squared Error (MSE): 0.028872026672167046

[205] # Calculate r2
      r2 = r2_score(y_test, y_pred)
      print(f"R-squared: {r2}")

      R-squared: 0.07313483682843747

      # Calculate MAE
      mae = mean_absolute_error(y_test, y_pred)
      print(f'Mean Absolute Error (MAE): {mae}')

      Mean Absolute Error (MAE): 0.125027488330547
```

Logistic Regression Model_(Mariam)_:

Used Research Paper

According to the research paper called “Effective Demand Forecasting Model Using Business Intelligence Empowered With Machine Learning” the model that was implemented in it was the logistic regression model and I used to implement it.

Code Structure:

First I used to implement the logistic model in a normal way by splitting the data after applying the EDA techniques and then explaining and interpreting the output using some feature selections and assumptions for linear models.

Splitting the Data and make it more capable for used Model:

As the Logistic Regression model works on probability values instead of continuous ones—since it is inherently a classification model—I started by making the target variable more compatible with this requirement. There are many ways to convert a continuous target into a categorical one, and after exploring various methods, I decided to use a simple yet effective approach: taking the median of the Sales column.

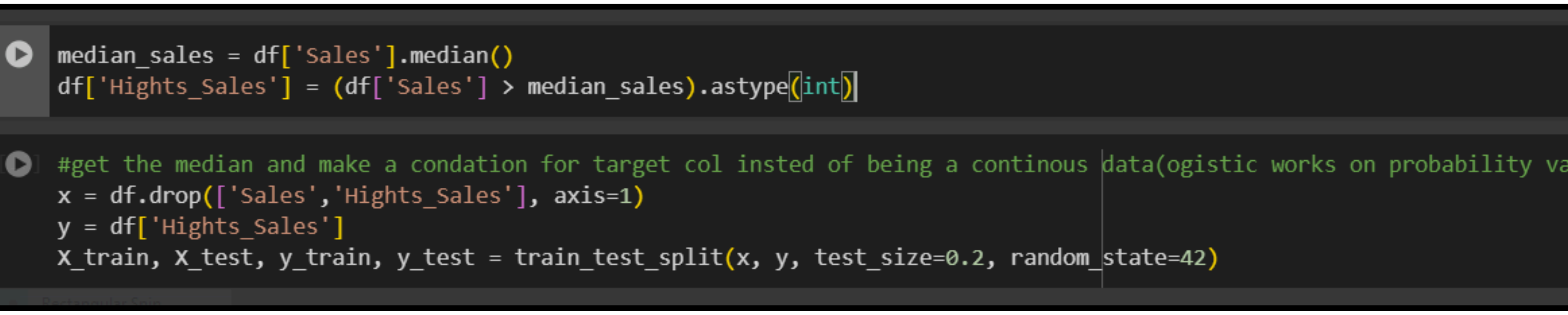
By calculating the median, I created a logical threshold to separate the data into two distinct classes:

- If a value in the Sales column is greater than the median, it is labeled as 1 (indicating high sales).

- If it is less than or equal to the median, it is labeled as 0 (indicating lower or average sales).

This binary transformation allowed the model to interpret the problem as a classification task, making it suitable for logistic regression. I stored this new binary target in a column named Hights_Sales. After this transformation, I dropped the original Sales and Hights_Sales columns from the input features and proceeded to split the data into training and testing sets.

This step ensured that the model could now work on predicting whether a new input leads to high or low sales, instead of trying to predict an exact continuous value, which logistic regression is not designed to do.



```
median_sales = df['Sales'].median()
df['Hights_Sales'] = (df['Sales'] > median_sales).astype(int)
```

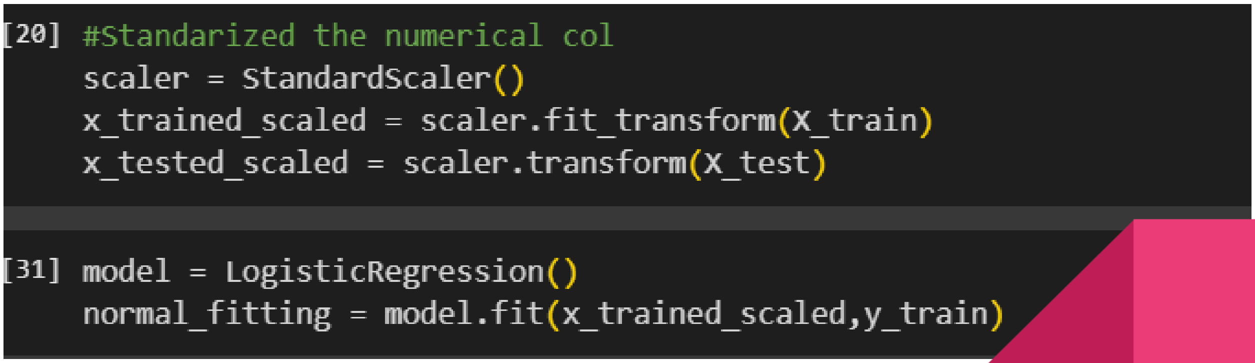
```
#get the median and make a condation for target col insted of being a continous data(ogistic works on probability va
x = df.drop(['Sales','Hights_Sales'], axis=1)
y = df['Hights_Sales']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

Before training the logistic regression model, I applied feature scaling to the input features. This is an important step in many machine learning workflows, especially for models that rely on gradient-based optimization, like logistic regression.

In my dataset, some features had different units or ranges, which could cause the model to give more weight to certain features just because their values are numerically larger. To solve this, I used standardization, a common technique that transforms the data so each feature has:

- A mean of 0
- A standard deviation of 1

This was done using the StandardScaler from scikit-learn.



```
[20] #Standarized the numerical col
     scaler = StandardScaler()
     x_trained_scaled = scaler.fit_transform(x_train)
     x_tested_scaled = scaler.transform(x_test)
```

```
[31] model = LogisticRegression()
     normal_fitting = model.fit(x_trained_scaled,y_train)
```

Evaluate the Model and get the performance measurement

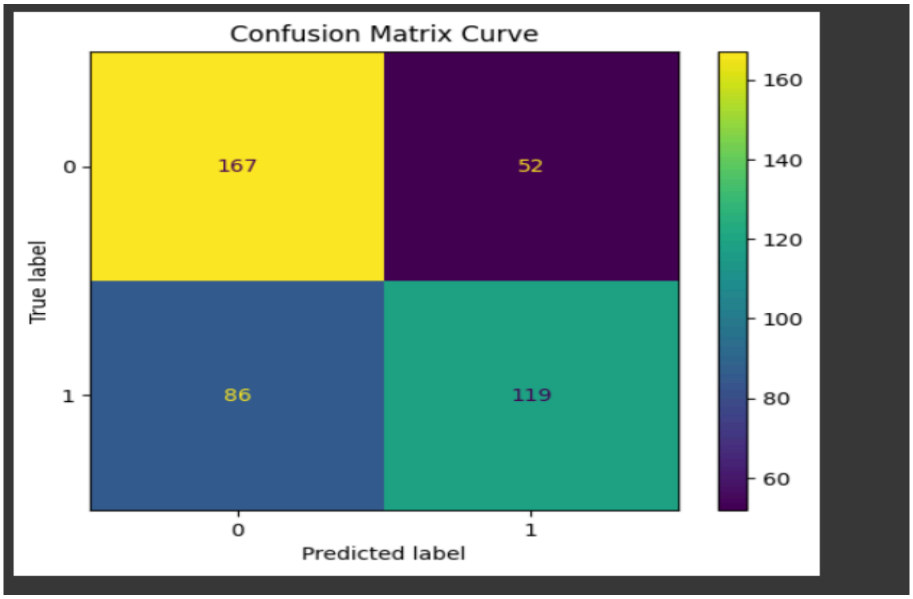
To evaluate the performance of the logistic regression model, four key metrics were used: accuracy, confusion matrix, classification report, and the ROC-AUC score. Accuracy provided a general measure of how often the model predicted correctly, while the confusion matrix offered a clearer view of true versus false predictions across both classes. The classification report further broke down performance by showing precision, recall, and F1-score for each class, giving deeper insight into how well the model handled imbalanced data. Lastly, the ROC curve and its associated AUC score illustrated the model's ability to distinguish between high and low sales classes based on predicted probabilities, with a higher AUC indicating better overall classification capability.

Explain the Graphes

1.Confusion Matrix:

This heatmap shows how many predictions the model got right (true positives and true negatives) versus wrong (false positives and false negatives).

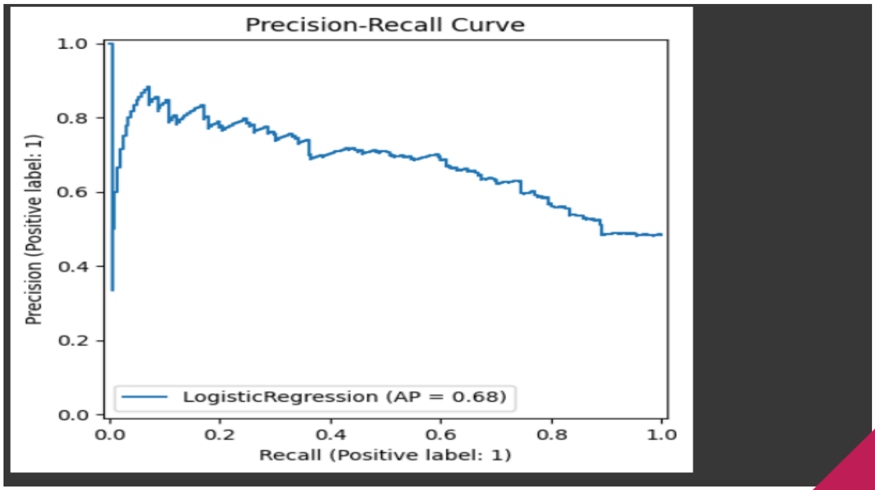
It helps visually assess which classes the model is struggling with or misclassifying.



2.ROC Curve:

The ROC curve plots the true positive rate against the false positive rate at various threshold settings.

A curve closer to the top-left corner (with a higher AUC) indicates better model performance in distinguishing between the two classes.



GitHub link :-

https://github.com/202201223/Sales-Forecasting-and-Demand-Prediction-Research-Paper