CUSTOMER DATA MANAGEMENT AND ANALYSIS

DEPI

Our Team

- Aya Elsheshtawy
- Shahd Ahmed Abdelsalam
- Fatma Nageh
- Soha Saad
- Sama Mohamed Elqasaby
- Alaa Osama Mohamed

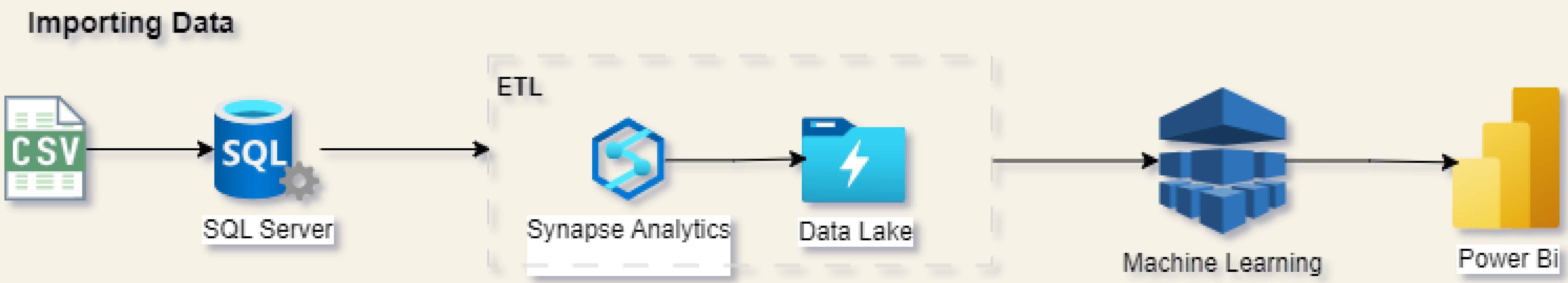
Agenda

- Project Objective
- Data Management
- Data Warehousing
- Machine Learning
- Visualization

Project Objective

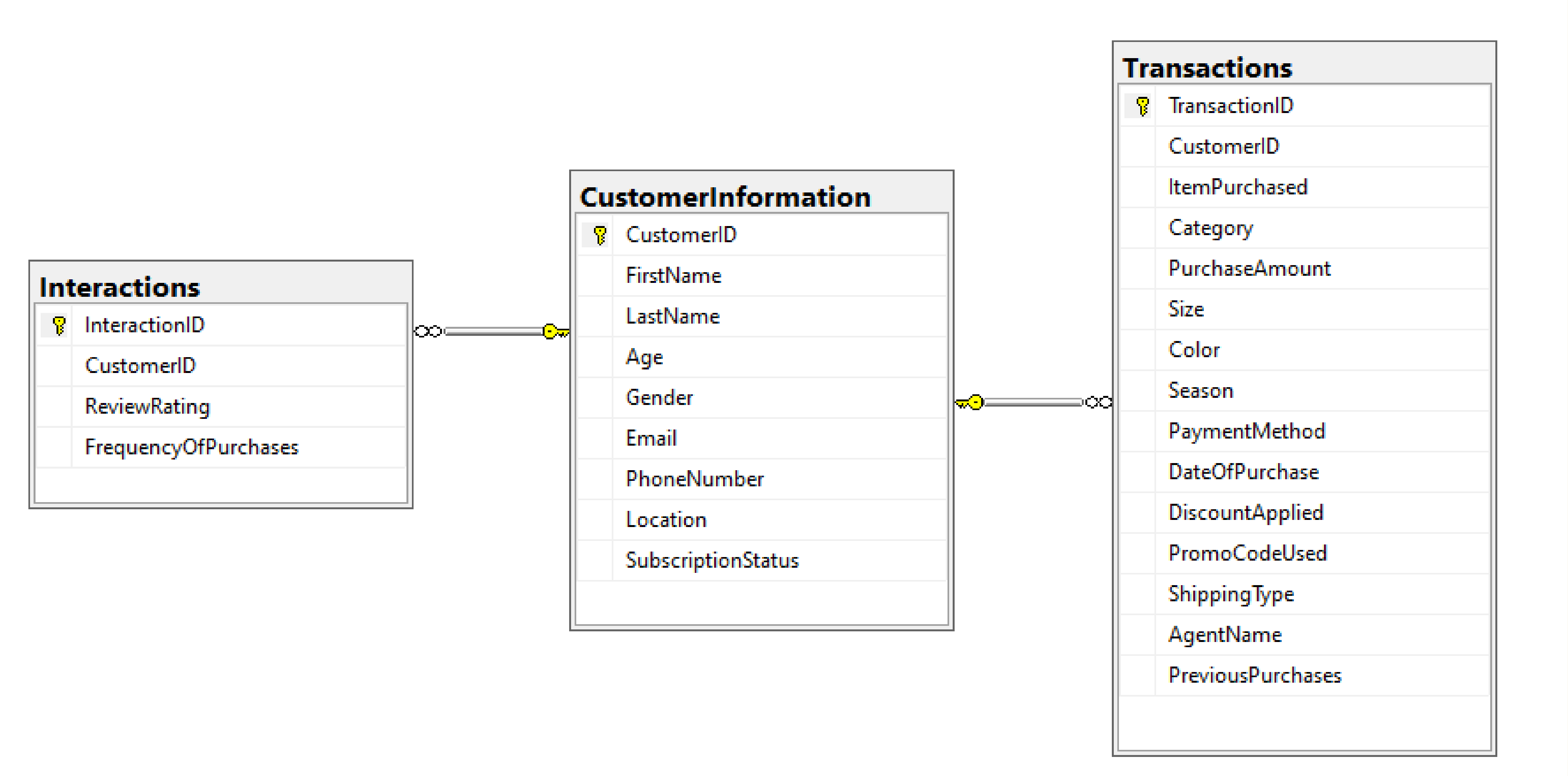
Design and implement a scalable data management solution that integrates customer data, enabling detailed analysis and predictive modeling to support informed business decisions.

This project helps businesses manage and analyze customer data more effectively, leading to better customer insights and smarter business strategies



Data Management

To develop a robust SQL database schema that efficiently organizes customer, transaction, and interaction data, enabling effective data extraction and analysis for improved customer insights and strategic decision-making



Interactions

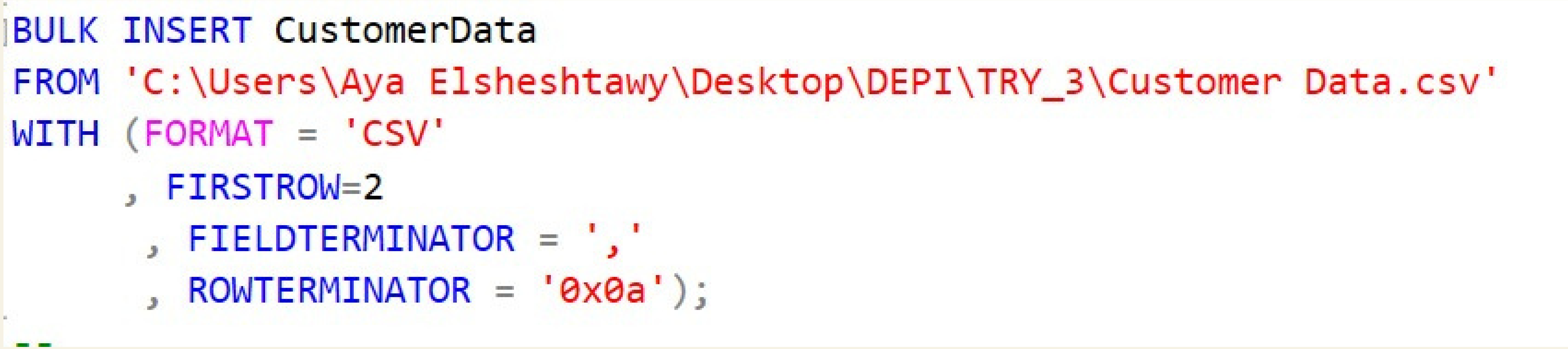
- InteractionID
- CustomerID
- ReviewRating
- FrequencyOfPurchases

CustomerInformation

- CustomerID
- FirstName
- LastName
- Age
- Gender
- Email
- PhoneNumber
- Location
- SubscriptionStatus

Transactions

- TransactionID
- CustomerID
- ItemPurchased
- Category
- PurchaseAmount
- Size
- Color
- Season
- PaymentMethod
- DateOfPurchase
- DiscountApplied
- PromoCodeUsed
- ShippingType
- AgentName
- PreviousPurchases



```sql
BULK INSERT CustomerData
FROM 'C:\Users\Aya Elsheshtawy\Desktop\DEPI\TRY_3\Customer Data.csv'
WITH (FORMAT = 'CSV'
     , FIRSTROW=2
      , FIELDTERMINATOR = ','
      , ROWTERMINATOR = '0x0a');
```

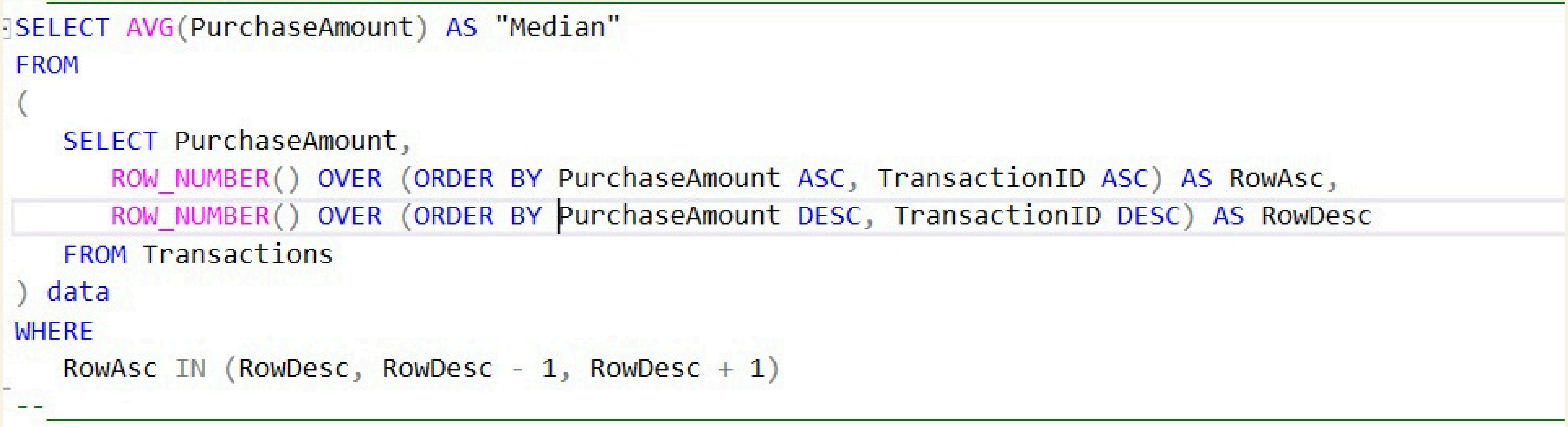
BULK INSERT command is used to efficiently load large amounts of data from a file into a SQL Server table.

Creating Tables:

1)CustomerInformation:
Includes essential customer details like first and last names, age, gender, contact information, location, and subscription status.
The CustomerID is the primary key, ensuring unique identification for each customer.

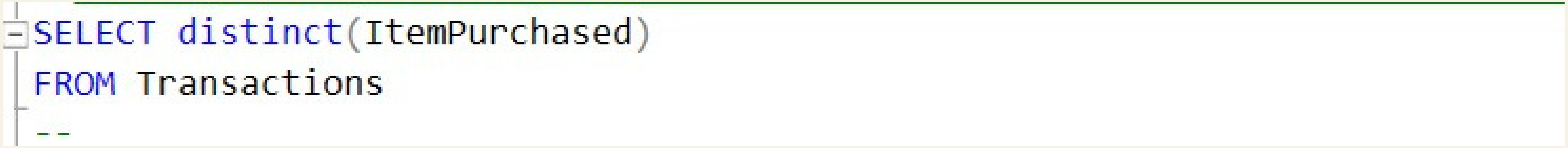
2)Transactions:
Records individual purchase events with detailed information about the item purchased, category, purchase amount, size, color, season, payment method, date of purchase, discounts, promo codes, shipping type, agent name, and previous purchases.
The TransactionID is the primary key, ensuring a unique identifier for each transaction.
The CustomerID foreign key establishes the relationship between transactions and customers.

3) Interactions:
Tracks customer interactions, including review ratings and purchase frequency.
The InteractionID is the primary key, ensuring a unique identifier for each interaction.
The CustomerID foreign key establishes the relationship between interactions and customers.



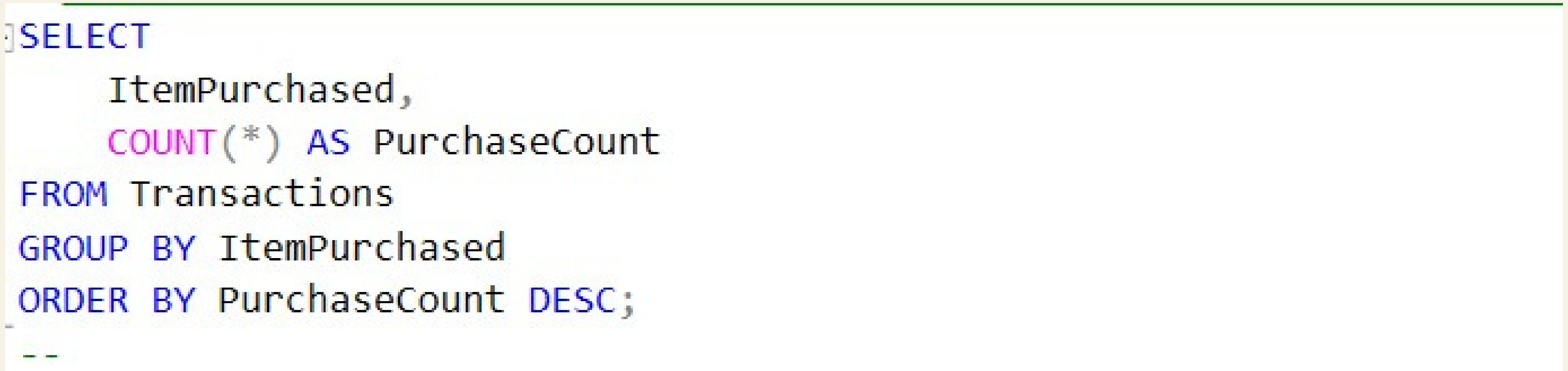
```
SELECT AVG(PurchaseAmount) AS "Median"
FROM
(
   SELECT PurchaseAmount,
      ROW_NUMBER() OVER (ORDER BY PurchaseAmount ASC, TransactionID ASC) AS RowAsc,
      ROW_NUMBER() OVER (ORDER BY PurchaseAmount DESC, TransactionID DESC) AS RowDesc
   FROM Transactions
) data
WHERE
   RowAsc IN (RowDesc, RowDesc - 1, RowDesc + 1)
--
```

Query to Calculate the median purchase amount.
This query uses a common technique to calculate the median by finding the middle value or the average of the two middle values in a sorted dataset.



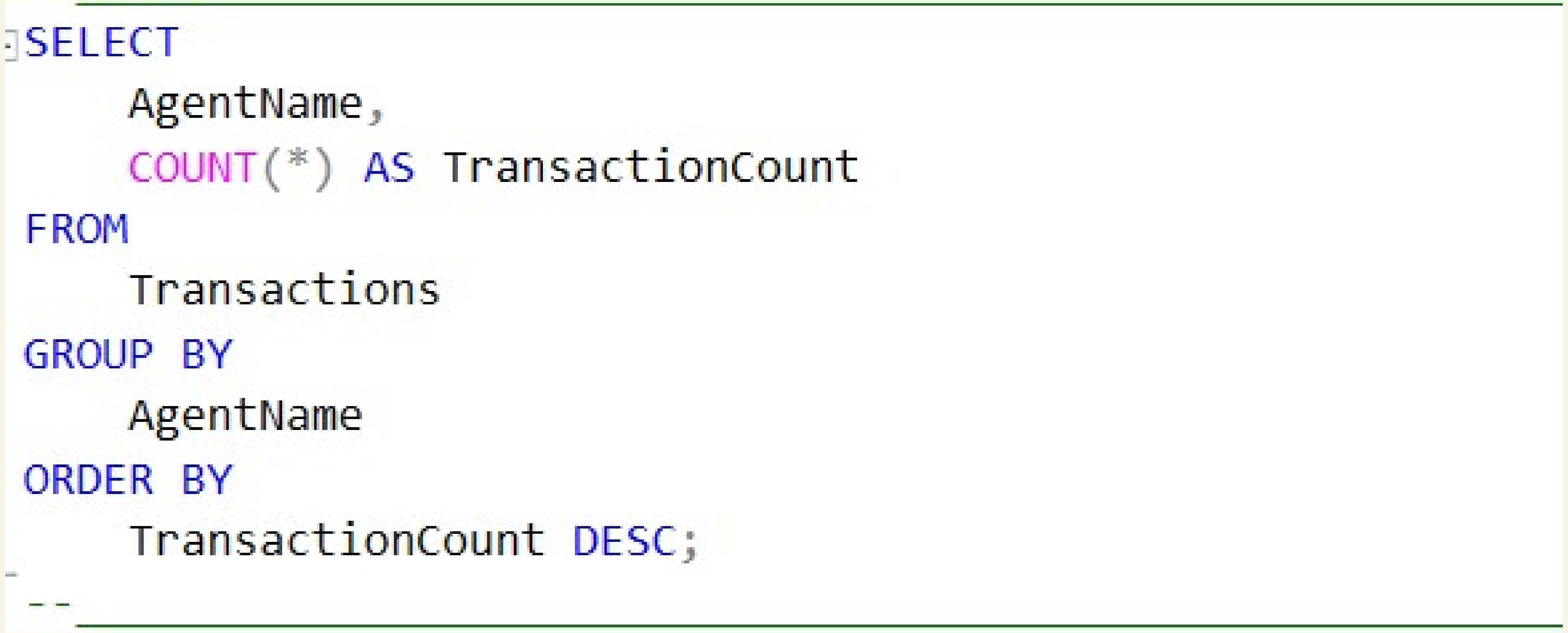
```sql
SELECT distinct(ItemPurchased)
FROM Transactions
--
```

Lists all unique items purchased.
This query provides a list of products offered or sold.



```sql
SELECT
    ItemPurchased,
    COUNT(*) AS PurchaseCount
FROM Transactions
GROUP BY ItemPurchased
ORDER BY PurchaseCount DESC;
--
```

Counts the number of times each item has been purchased.
This query identifies the most popular items based on purchase frequency.

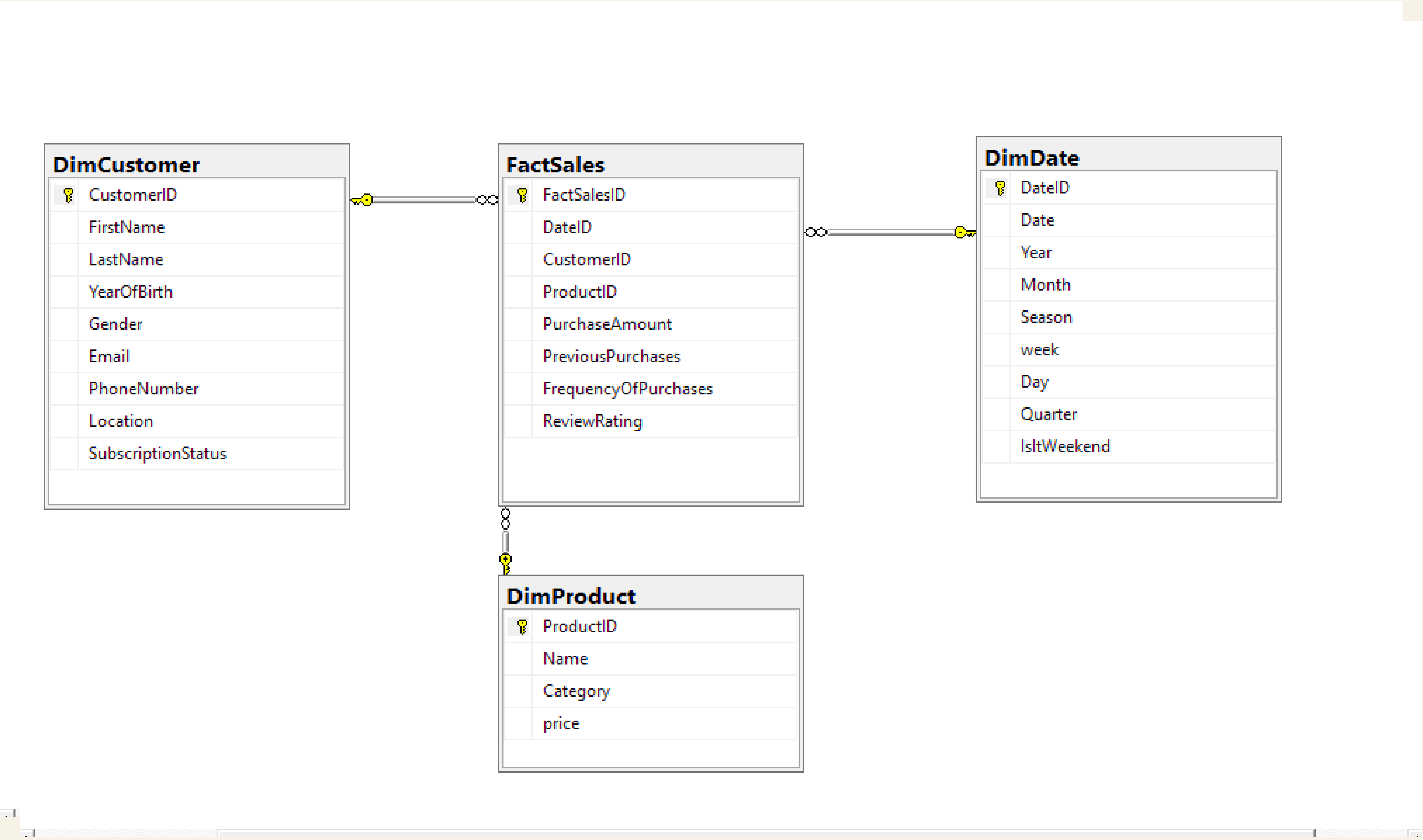


```sql
SELECT
    AgentName,
    COUNT(*) AS TransactionCount
FROM
    Transactions
GROUP BY
    AgentName
ORDER BY
    TransactionCount DESC;
--
```

Counts the number of transactions handled by each agent.
This query evaluates the performance or workload of different agents.

Data Warehousing

To create a comprehensive data warehouse that organizes customer, transaction, and interaction data for enhanced reporting and analysis using **Azure Synapse** and **Data Lake Gen2**.



DimCustomer
CustomerID
FirstName
LastName
YearOfBirth
Gender
Email
PhoneNumber
Location
SubscriptionStatus

FactSales
FactSalesID
DateID
CustomerID
ProductID
PurchaseAmount
PreviousPurchases
FrequencyOfPurchases
ReviewRating

DimDate
DateID
Date
Year
Month
Season
week
Day
Quarter
IsItWeekend

DimProduct
ProductID
Name
Category
price

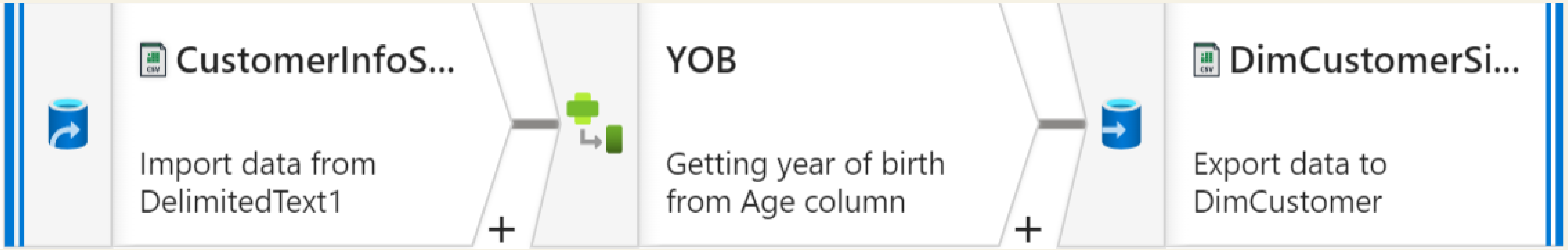
Logical Model

The **star schema** is ideal for its simplicity, fast query performance, and seamless integration with BI tools, making it perfect for efficient data analysis and reporting.

- **Date Dim:** is an essential table in a data model that allows us to analyze performance more effectively across different time periods.
- **Customer Dim:** Describe the information about the Customers and their Subscription Status.
- **Product Dim:** describes each product for analysis.
- **FactSales:** Records sales transactions, linking customers, products, and dates to provide insights into sales performance and trends.

Dimensional Model Overview :

Customer Dimension (DimCustomer):
Transformation : Changed the Age column to **YearOfBirth**.
Purpose : This makes it easier to track customer demographics over time and understand how their behavior changes.

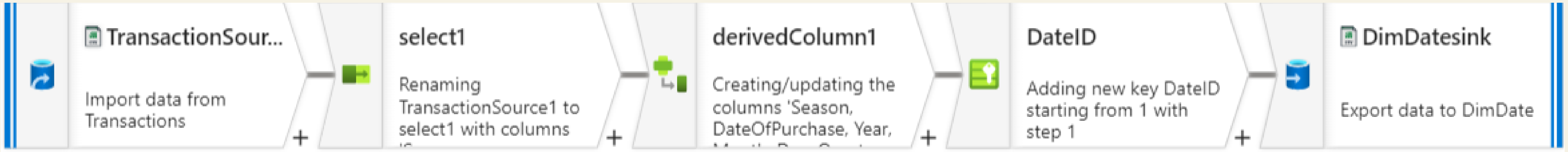


CustomerInfoS...
Import data from DelimitedText1

YOB
Getting year of birth from Age column

DimCustomerSi...
Export data to DimCustomer

Date Dimension (DimDate):
Transformation : Broke down **DateOfPurchase** into **Year**, **Month**, **Day**, **Quarter**, **Week**, and **IsItWeekend**.
Purpose : This helps us analyze sales trends at different time levels, like monthly growth or seasonal changes.



TransactionSour...
Import data from Transactions

select1
Renaming TransactionSource1 to select1 with columns

derivedColumn1
Creating/updating the columns 'Season, DateOfPurchase, Year,

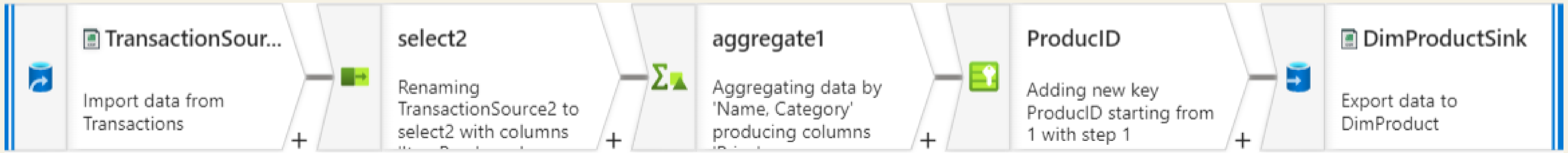
DateID
Adding new key DateID starting from 1 with step 1

DimDatesink
Export data to DimDate

Product Dimension (DimProduct) :

Transformation : Collected **Product** and **Category** data from the Transaction table.

Purpose : Centralizing this information allows us to easily assess how different products perform and make better business decisions.



TransactionSour...
Import data from Transactions

select2
Renaming TransactionSource2 to select2 with columns

aggregate1
Aggregating data by 'Name, Category' producing columns

ProducID
Adding new key ProducID starting from 1 with step 1

DimProductSink
Export data to DimProduct

Fact Table (FactSales) :

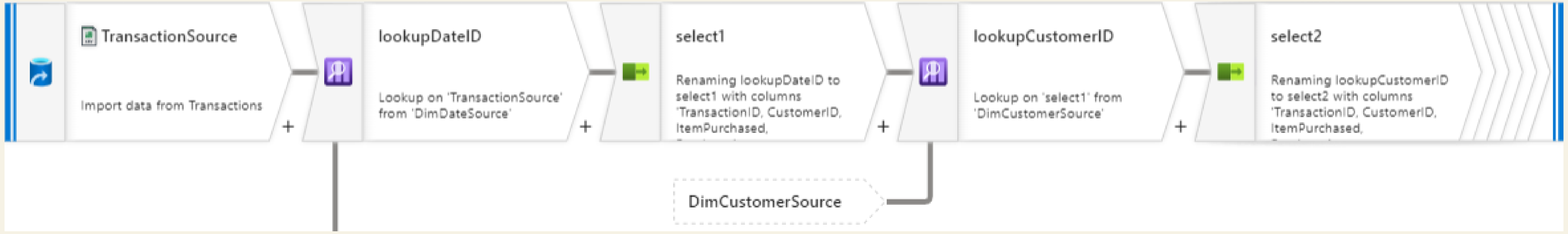
Linked the **DimCustomer**, **DimDate**, and **DimProduct** tables using surrogate keys to form a solid foundation for our data model.

Key Measures :

AmountOfPurchase and **PreviousPurchase**: Important for understanding spending habits.

FrequencyOfPurchase: Shows how often customers buy, indicating loyalty.

ReviewRating: Links sales data to customer feedback for a complete view of their experience.



TransactionSource
Import data from Transactions

lookupDateID
Lookup on 'TransactionSource' from 'DimDateSource'

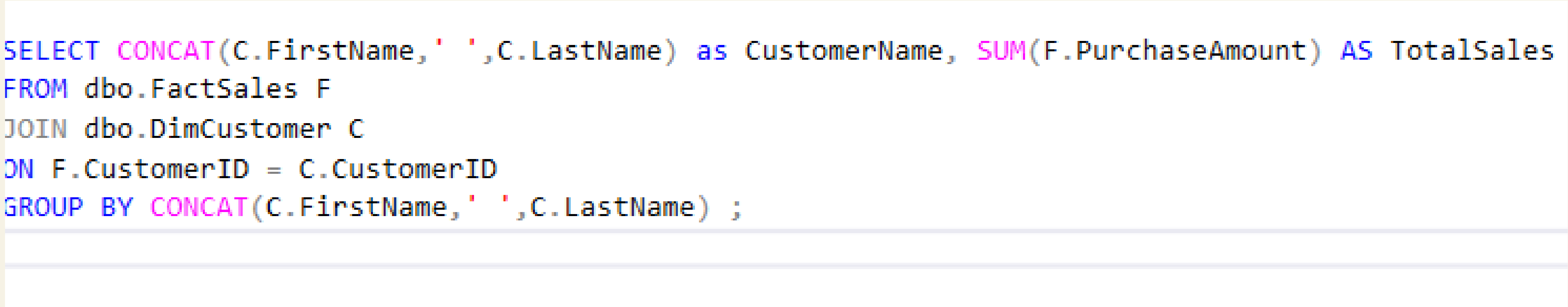
select1
Renaming lookupDateID to select1 with columns 'TransactionID, CustomerID, ItemPurchased,

lookupCustomerID
Lookup on 'select1' from 'DimCustomerSource'

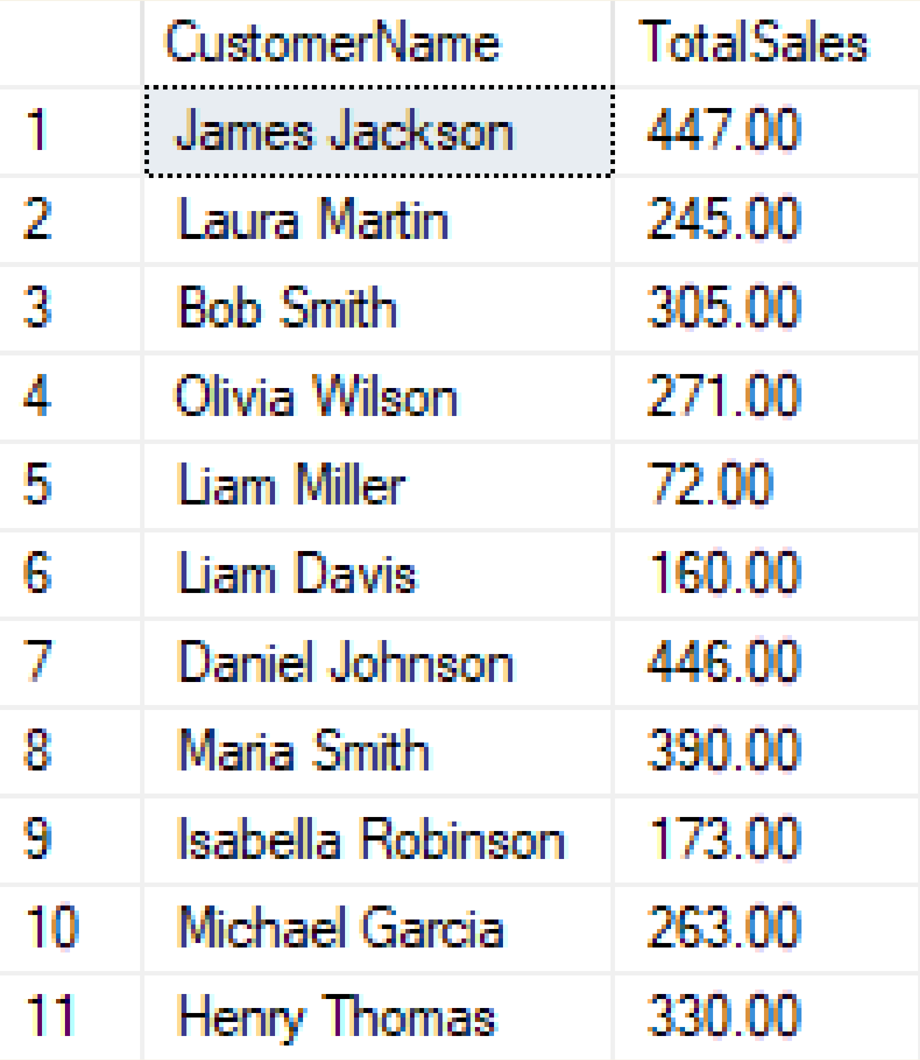
select2
Renaming lookupCustomerID to select2 with columns 'TransactionID, CustomerID, ItemPurchased,

DimCustomerSource

Total Sales by Customer



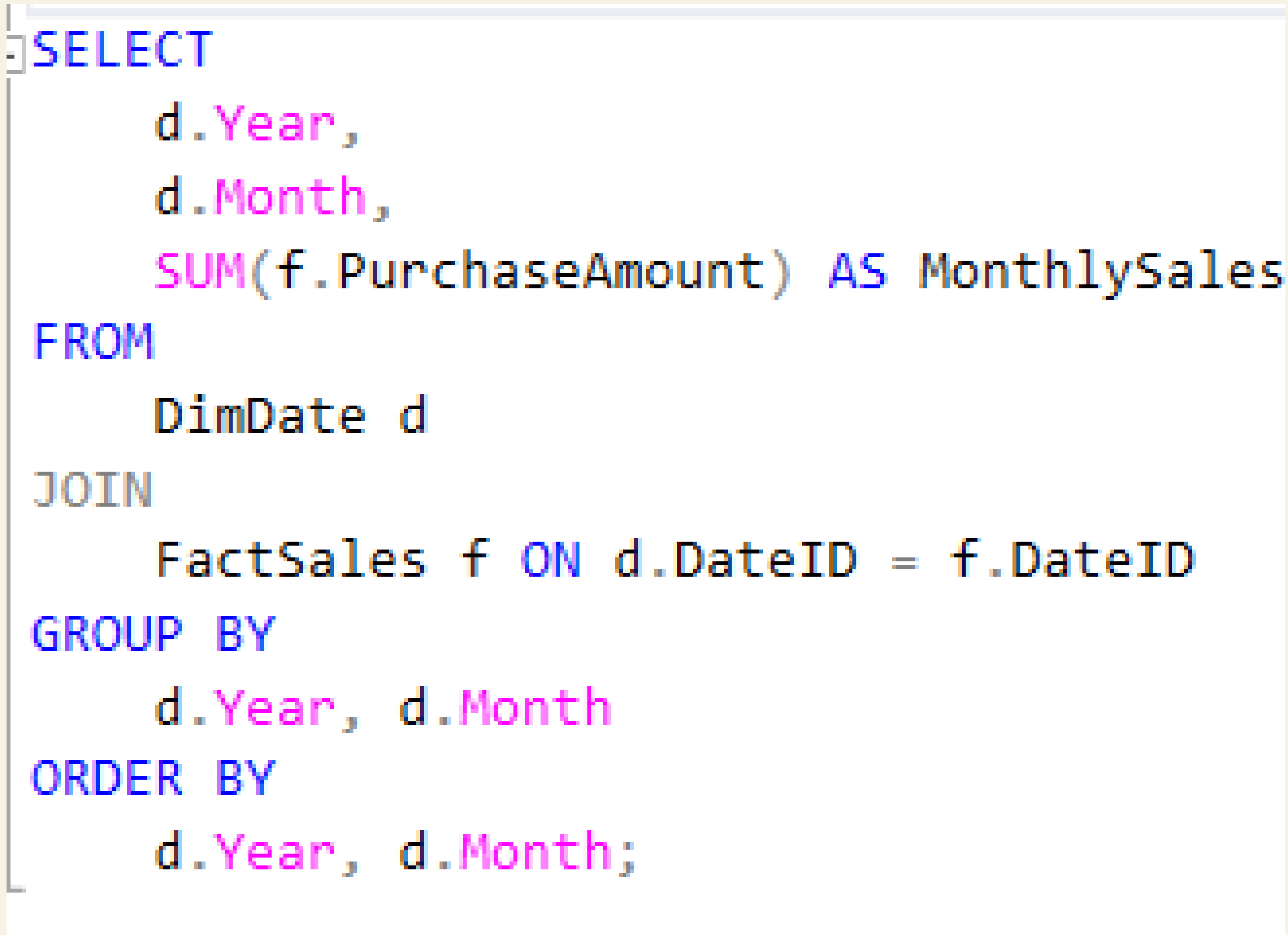
```sql
SELECT CONCAT(C.FirstName,' ',C.LastName) as CustomerName, SUM(F.PurchaseAmount) AS TotalSales
FROM dbo.FactSales F
JOIN dbo.DimCustomer C
ON F.CustomerID = C.CustomerID
GROUP BY CONCAT(C.FirstName,' ',C.LastName) ;
```



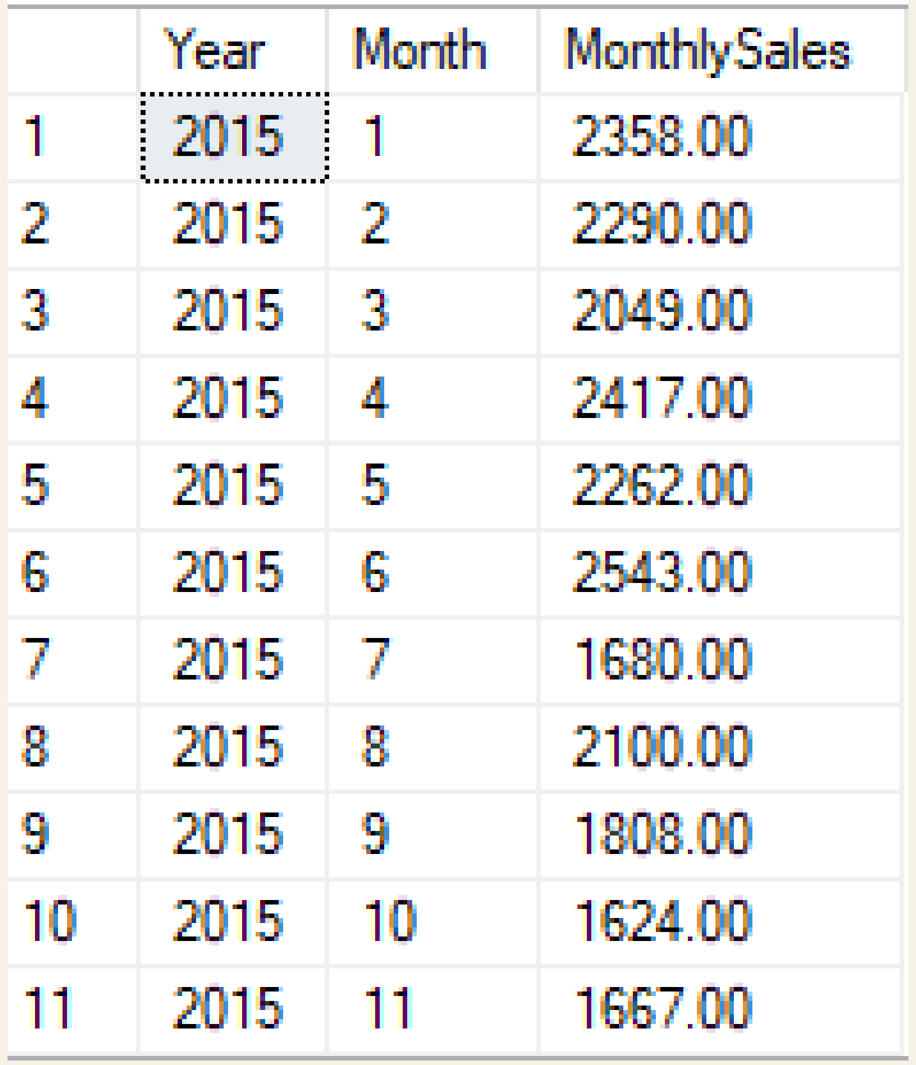
	CustomerName	TotalSales
1	James Jackson	447.00
2	Laura Martin	245.00
3	Bob Smith	305.00
4	Olivia Wilson	271.00
5	Liam Miller	72.00
6	Liam Davis	160.00
7	Daniel Johnson	446.00
8	Maria Smith	390.00
9	Isabella Robinson	173.00
10	Michael Garcia	263.00
11	Henry Thomas	330.00

This query calculates the total sales per customer, offering insights into individual customer contributions. It helps businesses identify high-value customers, enabling targeted marketing and data-driven decision-making.

Sales Trend Over Time



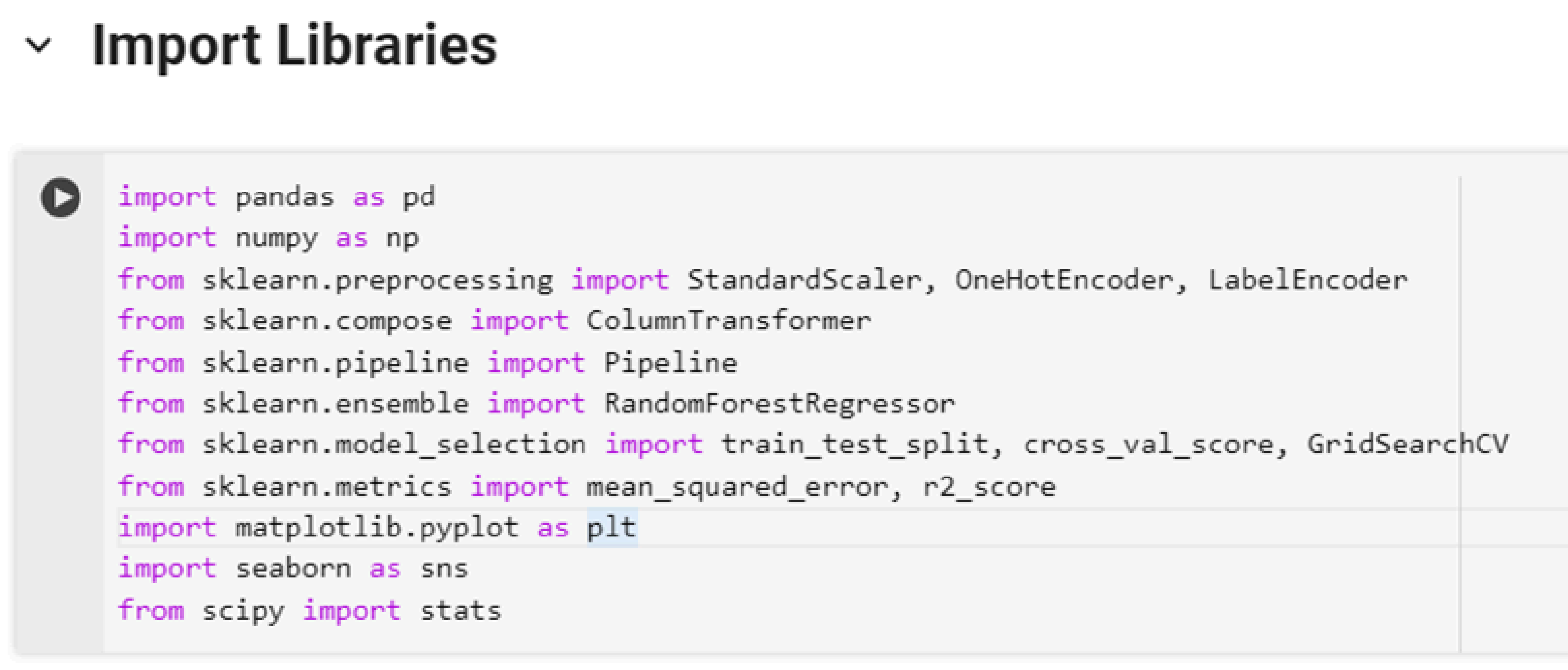
```
SELECT
    d.Year,
    d.Month,
    SUM(f.PurchaseAmount) AS MonthlySales
FROM
    DimDate d
JOIN
    FactSales f ON d.DateID = f.DateID
GROUP BY
    d.Year, d.Month
ORDER BY
    d.Year, d.Month;
```



	Year	Month	MonthlySales
1	2015	1	2358.00
2	2015	2	2290.00
3	2015	3	2049.00
4	2015	4	2417.00
5	2015	5	2262.00
6	2015	6	2543.00
7	2015	7	1680.00
8	2015	8	2100.00
9	2015	9	1808.00
10	2015	10	1624.00
11	2015	11	1667.00

his query calculates the monthly sales totals by summing purchase amounts for each month and year. It helps businesses analyze sales trends over time, identify seasonal patterns, and make informed decisions to optimize sales strategies.

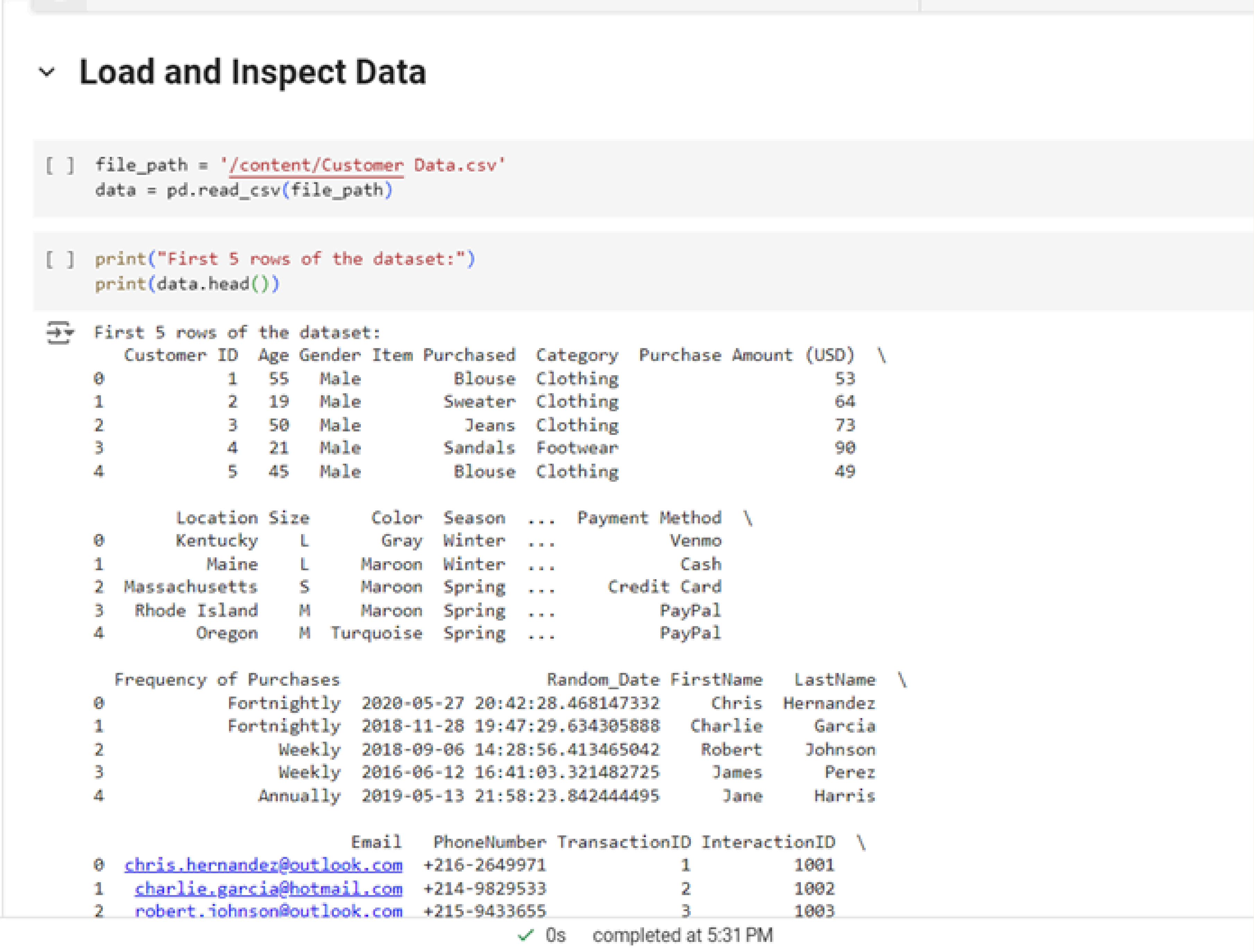
Machine Learning



Import Libraries

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler, OneHotEncoder, LabelEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

In this project, we focus on predicting customer purchase amounts based on various features using machine learning. We begin by importing necessary libraries, such as pandas for data handling, sklearn for modeling, and matplotlib for visualizations."



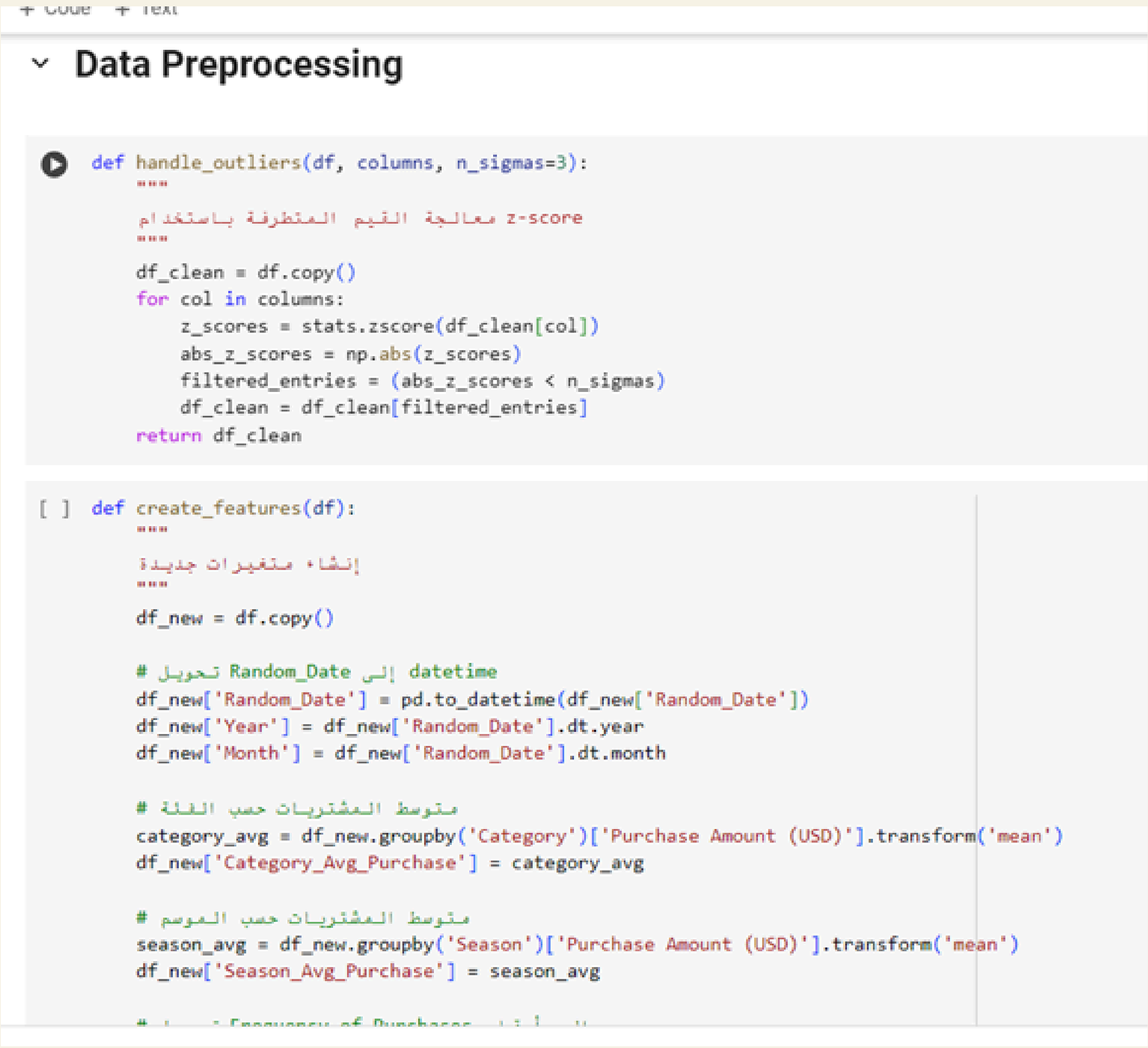
Load and Inspect Data

```
[ ] file_path = '/content/Customer Data.csv'
    data = pd.read_csv(file_path)
```

```
[ ] print("First 5 rows of the dataset:")
    print(data.head())
```

```
First 5 rows of the dataset:
   Customer ID  Age Gender Item Purchased  Category  Purchase Amount (USD)  \
0            1   55   Male         Blouse  Clothing                     53
1            2   19   Male        Sweater  Clothing                     64
2            3   50   Male          Jeans  Clothing                     73
3            4   21   Male        Sandals  Footwear                     90
4            5   45   Male         Blouse  Clothing                     49

        Location Size      Color  Season  ...  Payment Method  \
0       Kentucky    L       Gray  Winter  ...           Venmo
1          Maine    L     Maroon  Winter  ...            Cash
2  Massachusetts    S     Maroon  Spring  ...     Credit Card
3   Rhode Island    M     Maroon  Spring  ...          PayPal
4         Oregon    M  Turquoise  Spring  ...          PayPal

  Frequency of Purchases                   Random_Date FirstName   LastName  \
0            Fortnightly 2020-05-27 20:42:28.468147332     Chris  Hernandez
1            Fortnightly 2018-11-28 19:47:29.634305888   Charlie     Garcia
2                 Weekly 2018-09-06 14:28:56.413465042    Robert    Johnson
3                 Weekly 2016-06-12 16:41:03.321482725     James      Perez
4               Annually 2019-05-13 21:58:23.842444495      Jane     Harris

                        Email  PhoneNumber TransactionID InteractionID  \
0  chris.hernandez@outlook.com +216-2649971             1          1001
1   charlie.garcia@hotmail.com +214-9829533             2          1002
2   robert.johnson@outlook.com +215-9433655             3          1003
```

The dataset contains various customer attributes like Age, Category, and Purchase Amount. We load the data and inspect the first five rows to understand its structure."



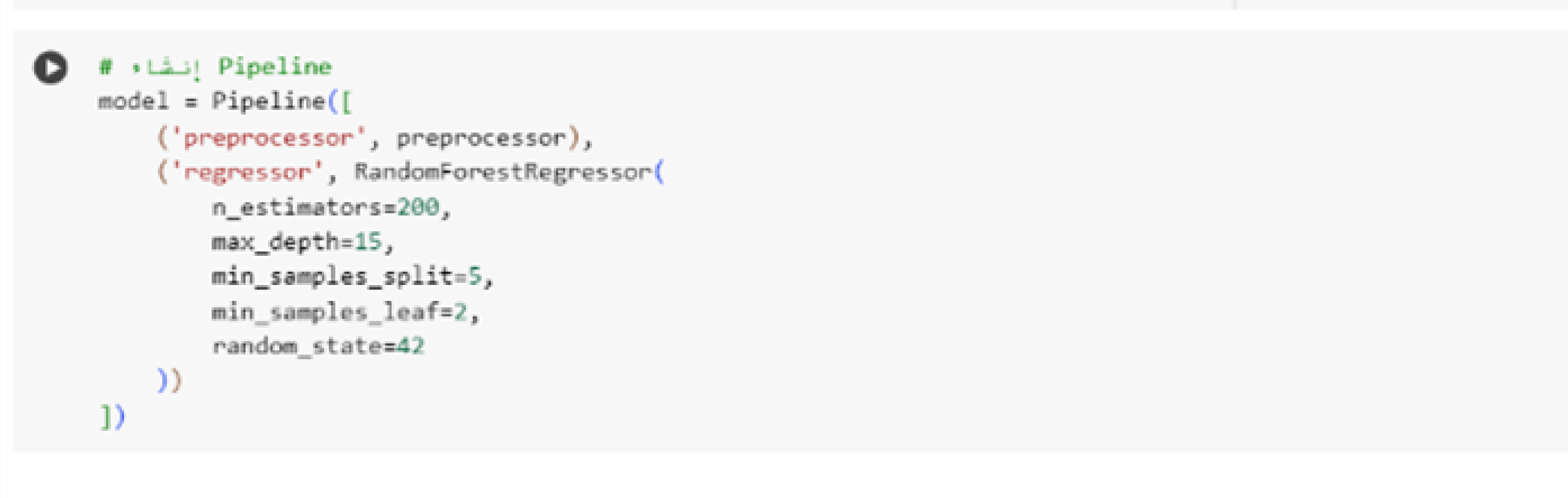
Data Preprocessing

```python
def handle_outliers(df, columns, n_sigmas=3):
    """
    معالجة القيم المتطرفة باستخدام z-score
    """
    df_clean = df.copy()
    for col in columns:
        z_scores = stats.zscore(df_clean[col])
        abs_z_scores = np.abs(z_scores)
        filtered_entries = (abs_z_scores < n_sigmas)
        df_clean = df_clean[filtered_entries]
    return df_clean
```

```python
def create_features(df):
    """
    إنشاء متغيرات جديدة
    """
    df_new = df.copy()

    # تحويل Random_Date إلى datetime
    df_new['Random_Date'] = pd.to_datetime(df_new['Random_Date'])
    df_new['Year'] = df_new['Random_Date'].dt.year
    df_new['Month'] = df_new['Random_Date'].dt.month

    # متوسط المشتريات حسب الفئة
    category_avg = df_new.groupby('Category')['Purchase Amount (USD)'].transform('mean')
    df_new['Category_Avg_Purchase'] = category_avg

    # متوسط المشتريات حسب الموسم
    season_avg = df_new.groupby('Season')['Purchase Amount (USD)'].transform('mean')
    df_new['Season_Avg_Purchase'] = season_avg

    # Frequency of Purchases
```

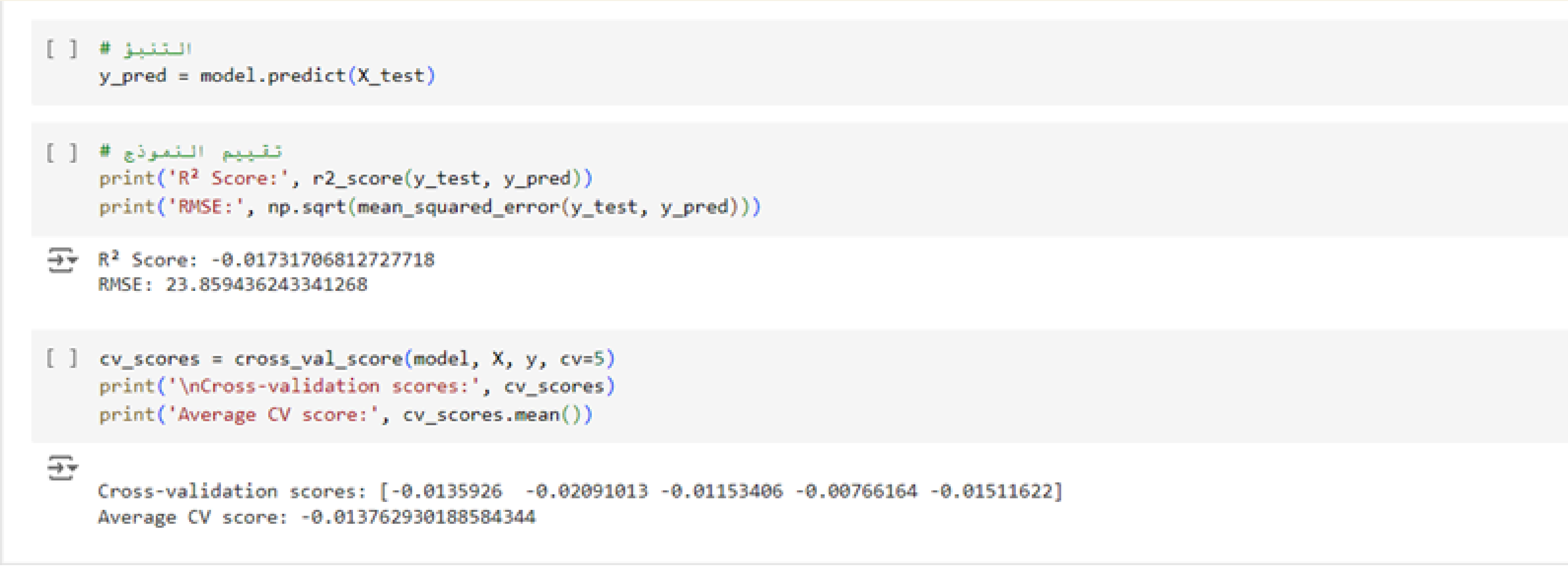
1. We performed data preprocessing by handling outliers and creating new features such as 'Year' and 'Month' from the random date column.

2. "We engineered new features, like the average purchase amount by category and season, and converted purchase frequency to numeric values."



```python
# إنشاء Pipeline
model = Pipeline([
    ('preprocessor', preprocessor),
    ('regressor', RandomForestRegressor(
        n_estimators=200,
        max_depth=15,
        min_samples_split=5,
        min_samples_leaf=2,
        random_state=42
    ))
])
```

We use a Random Forest Regressor to predict the purchase amount. The model pipeline includes preprocessing steps such as scaling and encoding, followed by the regression model.”



```
[ ] # التنبؤ
    y_pred = model.predict(X_test)
```

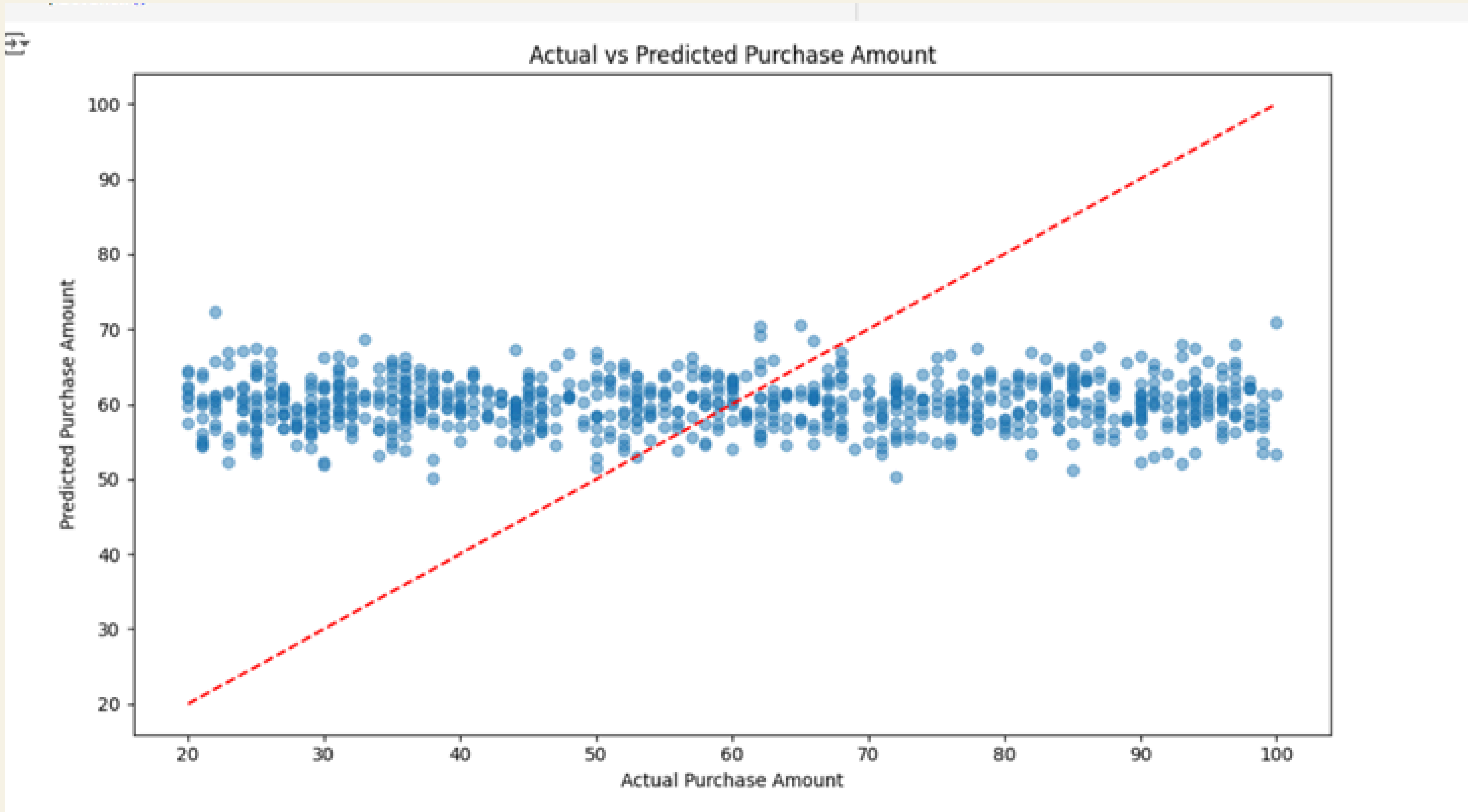
```
[ ] # تقييم النموذج
    print('R² Score:', r2_score(y_test, y_pred))
    print('RMSE:', np.sqrt(mean_squared_error(y_test, y_pred)))
```

```
R² Score: -0.01731706812727718
RMSE: 23.859436243341268
```

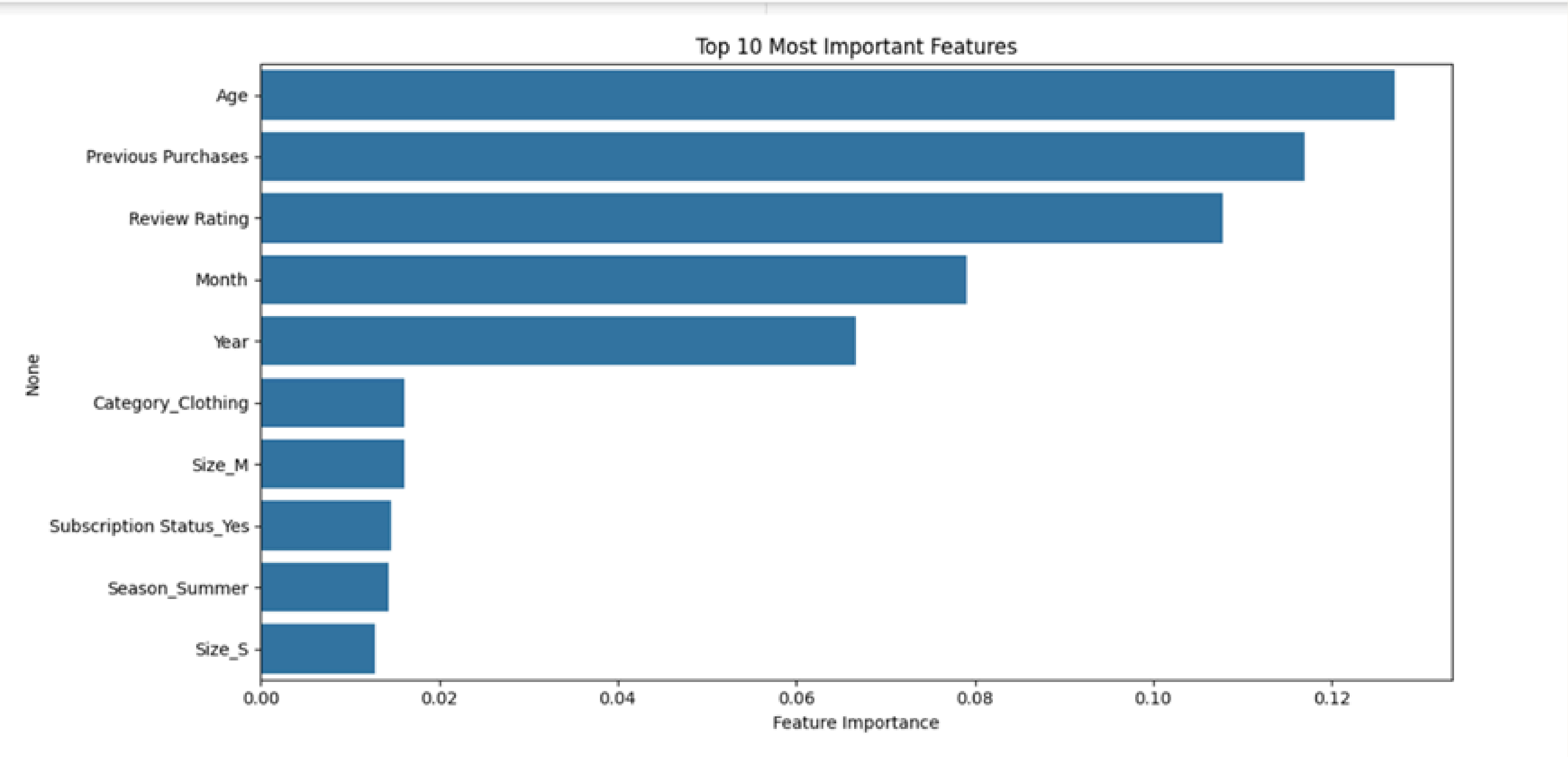
```
[ ] cv_scores = cross_val_score(model, X, y, cv=5)
    print('\nCross-validation scores:', cv_scores)
    print('Average CV score:', cv_scores.mean())
```

```

Cross-validation scores: [-0.0135926  -0.02091013 -0.01153406 -0.00766164 -0.01511622]
Average CV score: -0.013762930188584344
```

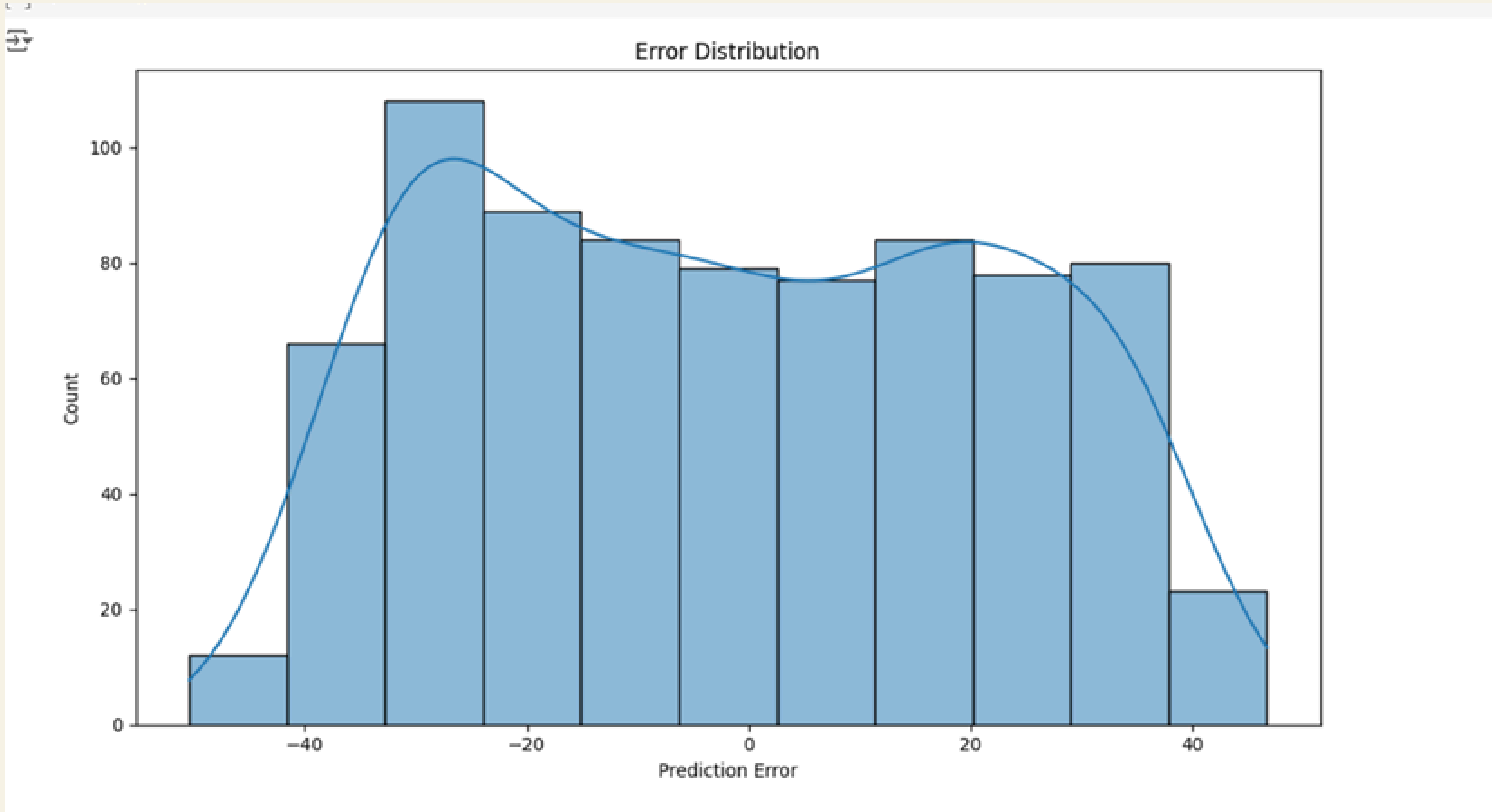
After training the model, we evaluate its performance using the R^2 score and RMSE. These metrics tell us how well the model fits the data."



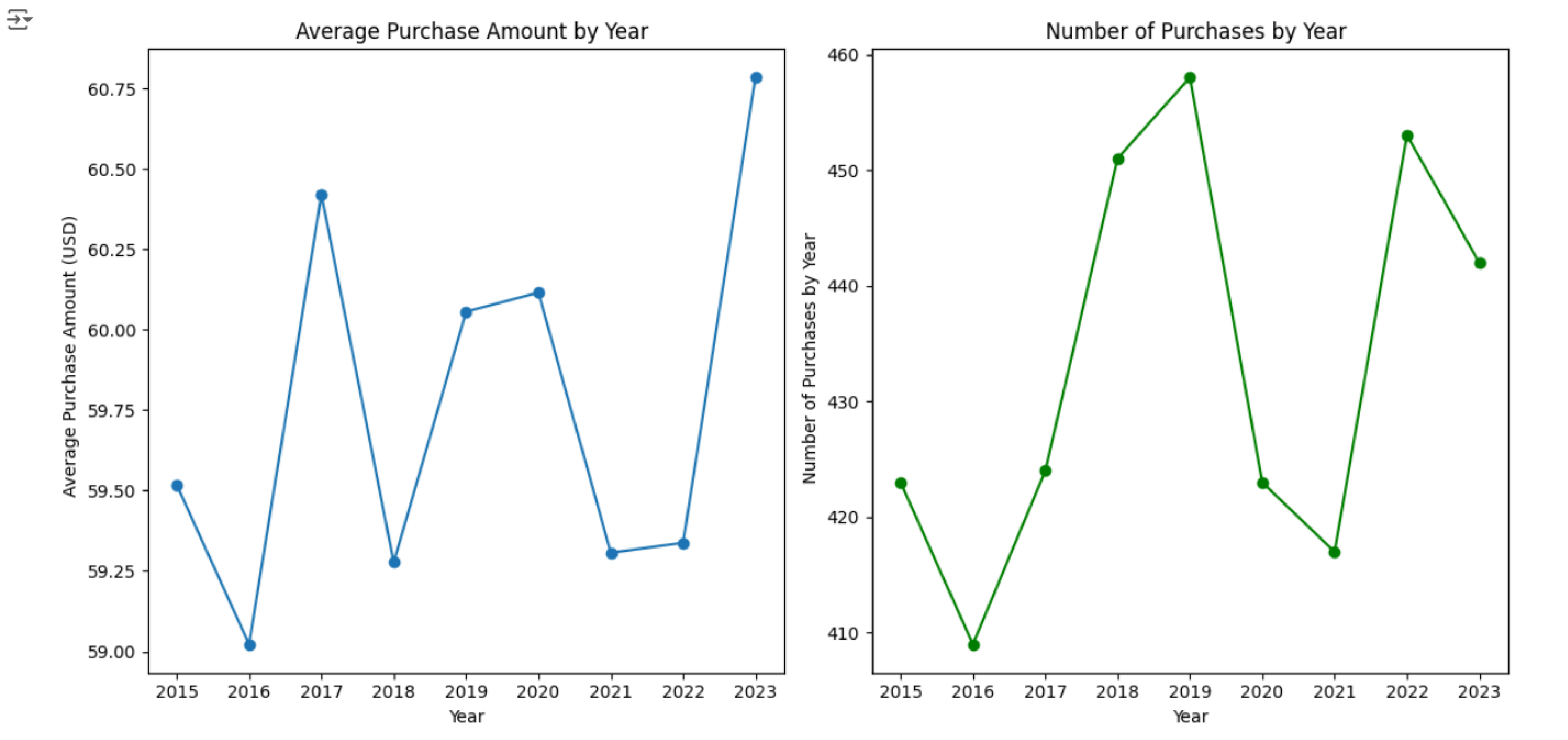
This plot shows the comparison between the actual purchase amounts and the predicted amounts made by the model. Ideally, the points should lie along the red dashed line, which represents perfect predictions. The more closely the points align with the line, the better the model's performance."



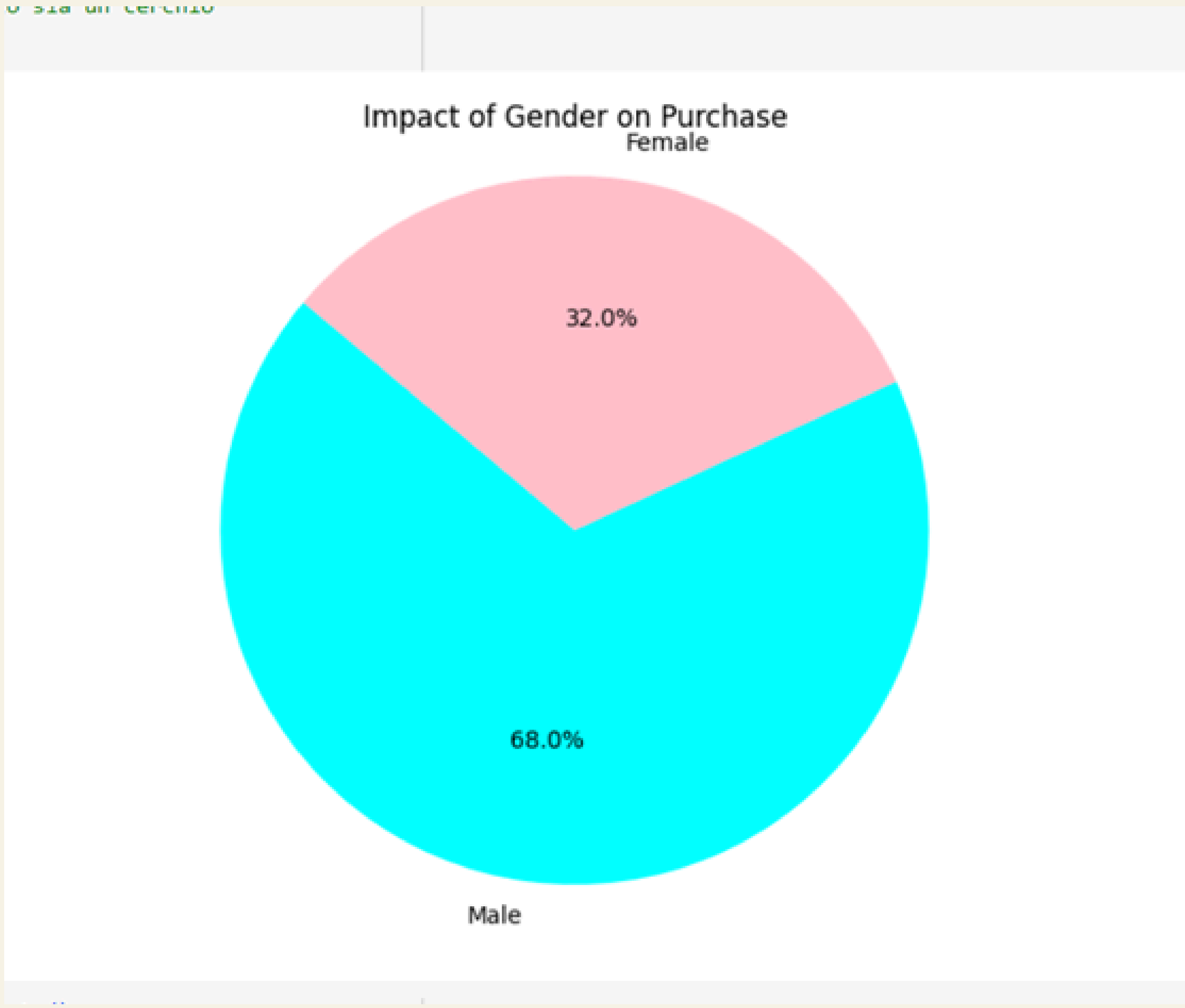
This bar plot highlights the top 10 most important features influencing the purchase amount. Features like Age, Previous Purchases, and Review Rating have a strong impact on the predictions, as shown by their high importance scores."



"This plot visualizes the distribution of prediction errors. Ideally, the errors should be centered around zero, indicating that the model is not biased in its predictions. A normal distribution of errors is a good sign of model reliability."

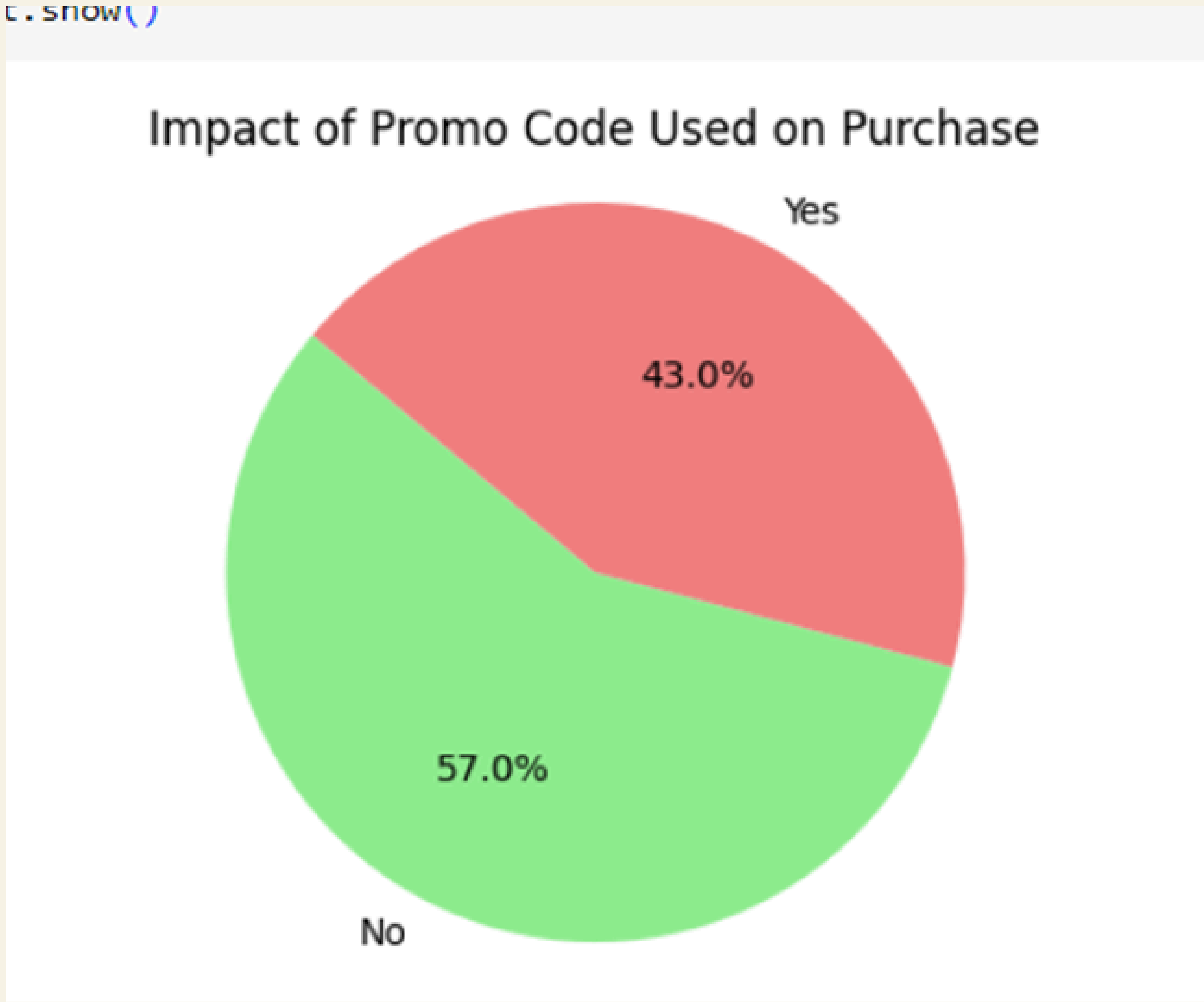


This line plot shows the trend of the average purchase amount over the years. It helps us understand how customer behavior and purchasing patterns have changed over time."



```
o sia un cerchio
```

his pie chart shows the distribution of purchases between different genders. It provides insight into how gender might affect customer purchasing behavior."

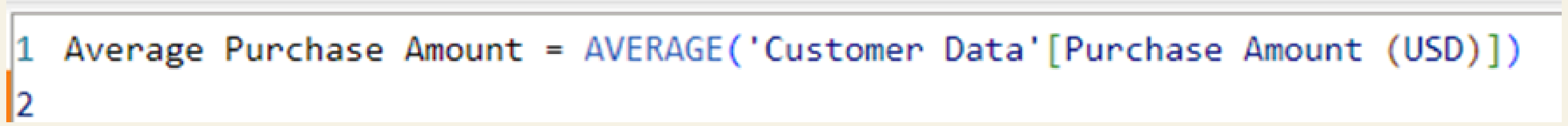


```
t.show()
```

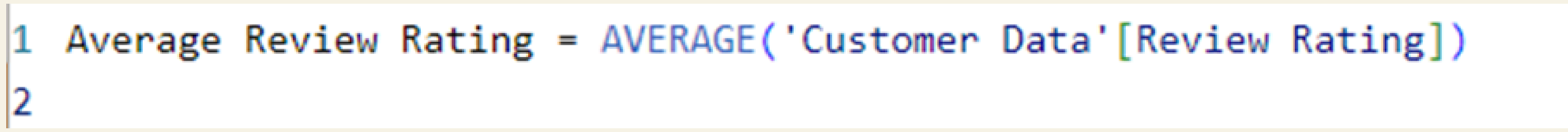
This pie chart illustrates the percentage of customers who used promo codes during their purchase. It helps understand how promotional offers influence buying behavior."

Visualization

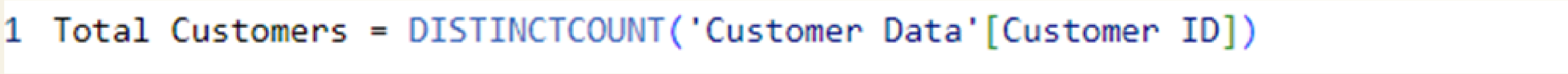
Calculate measures in DAX



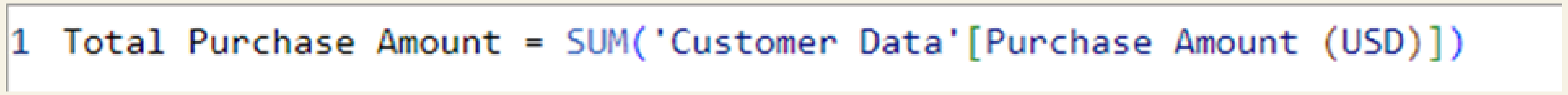
```
Average Purchase Amount = AVERAGE('Customer Data'[Purchase Amount (USD)])

```



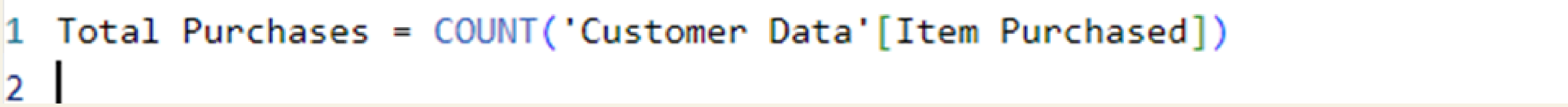
```
Average Review Rating = AVERAGE('Customer Data'[Review Rating])

```



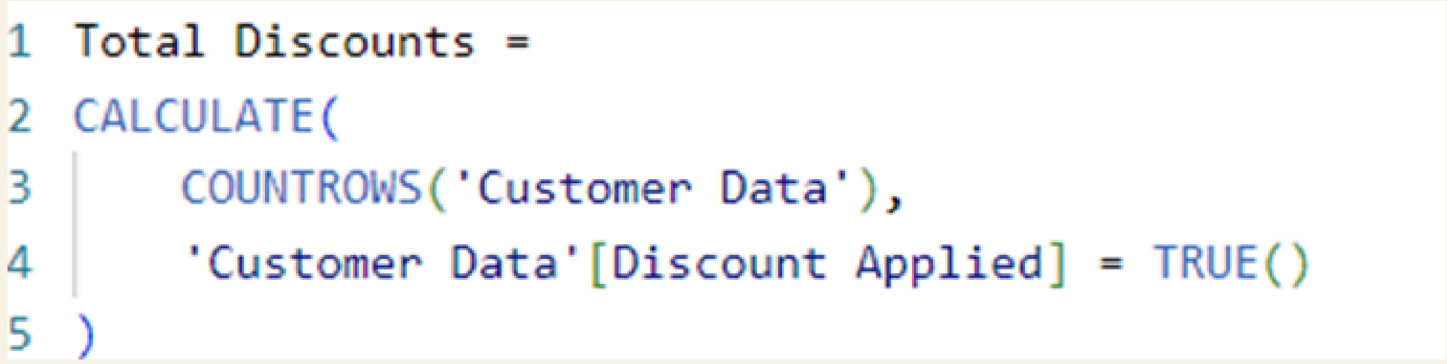
```
Total Customers = DISTINCTCOUNT('Customer Data'[Customer ID])
```



```
Total Purchase Amount = SUM('Customer Data'[Purchase Amount (USD)])
```

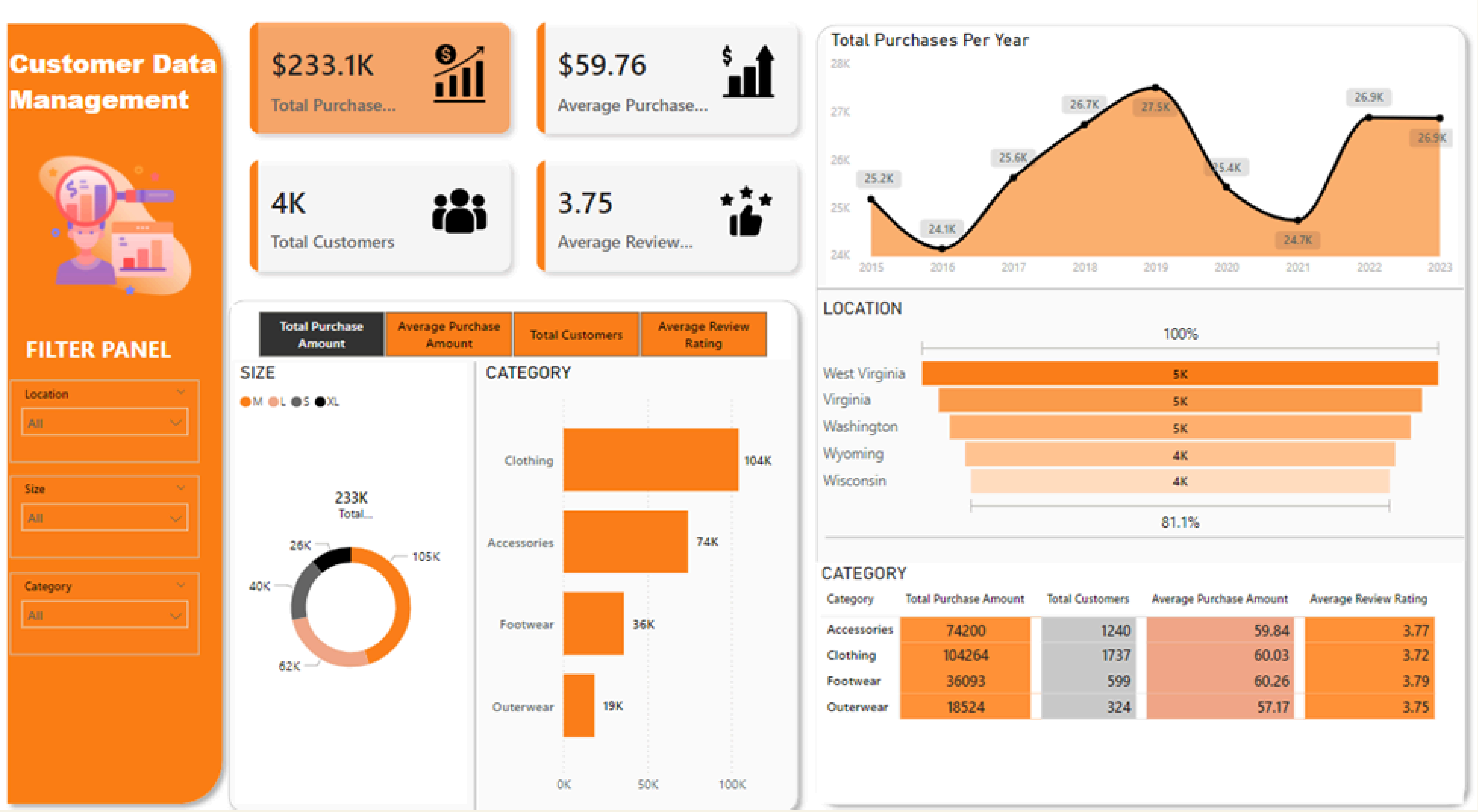


```
Total Purchases = COUNT('Customer Data'[Item Purchased])

```



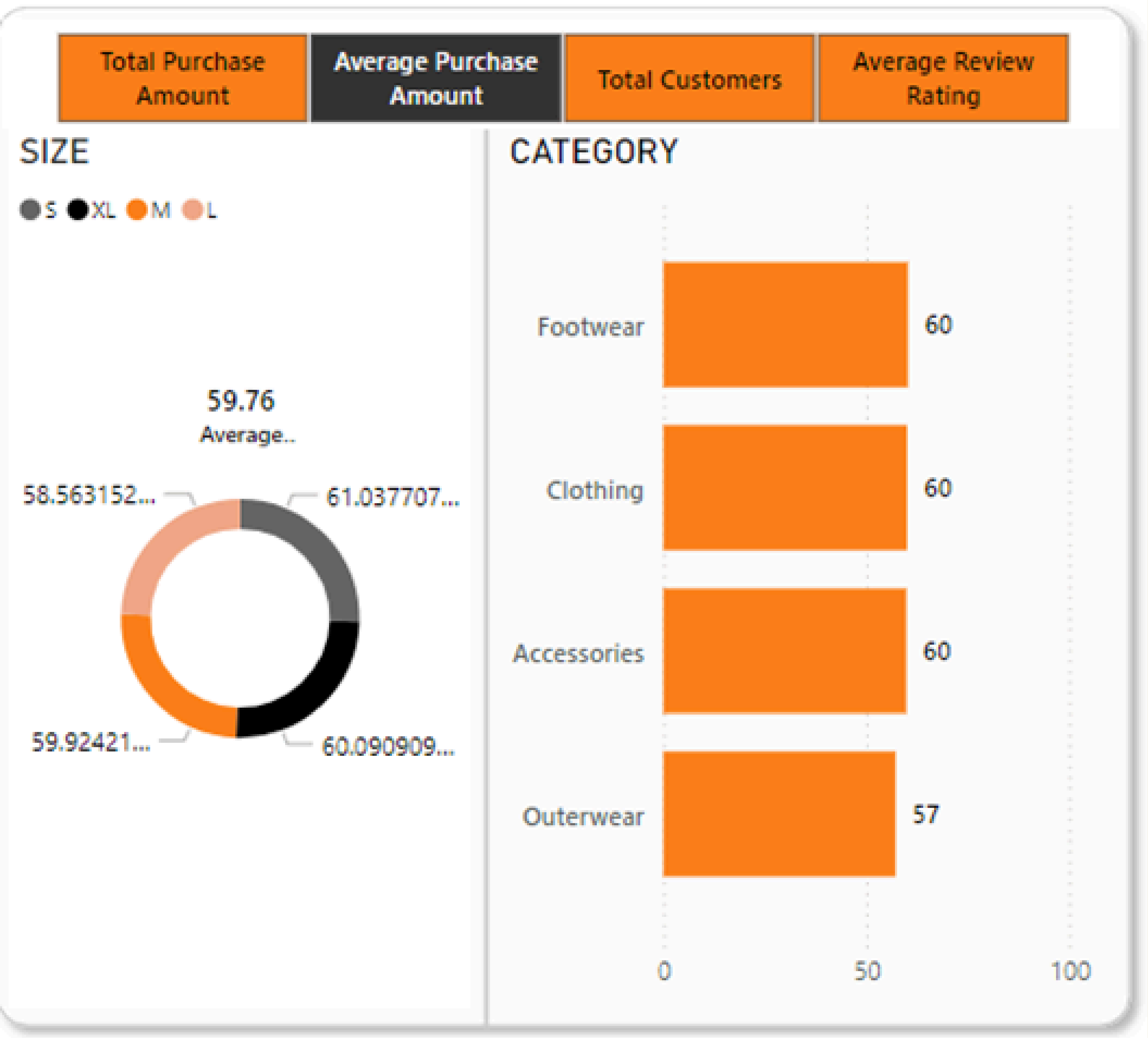
```
Total Discounts =
CALCULATE(
    COUNTROWS('Customer Data'),
    'Customer Data'[Discount Applied] = TRUE()
)
```

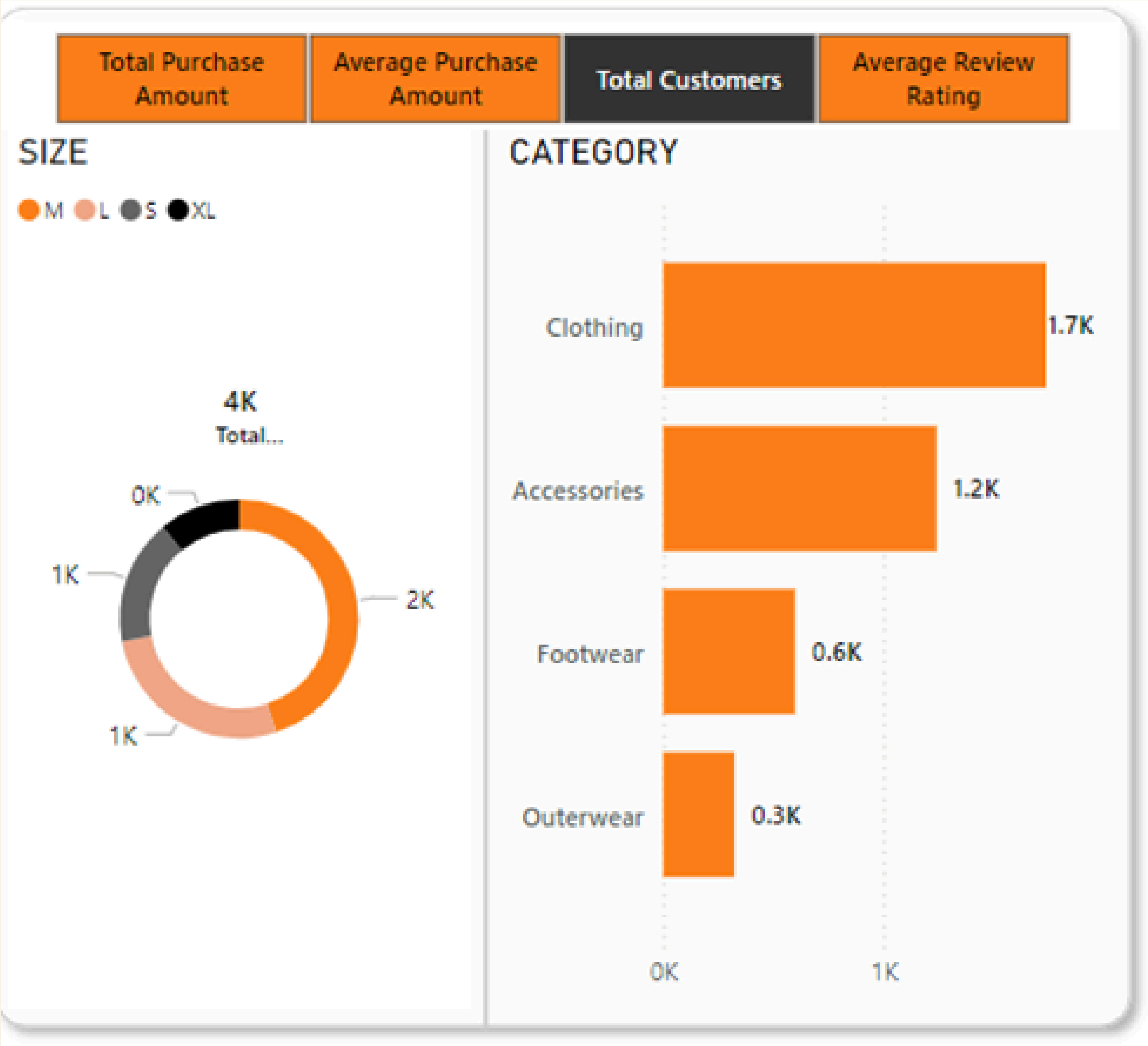
Interactive Dashboard

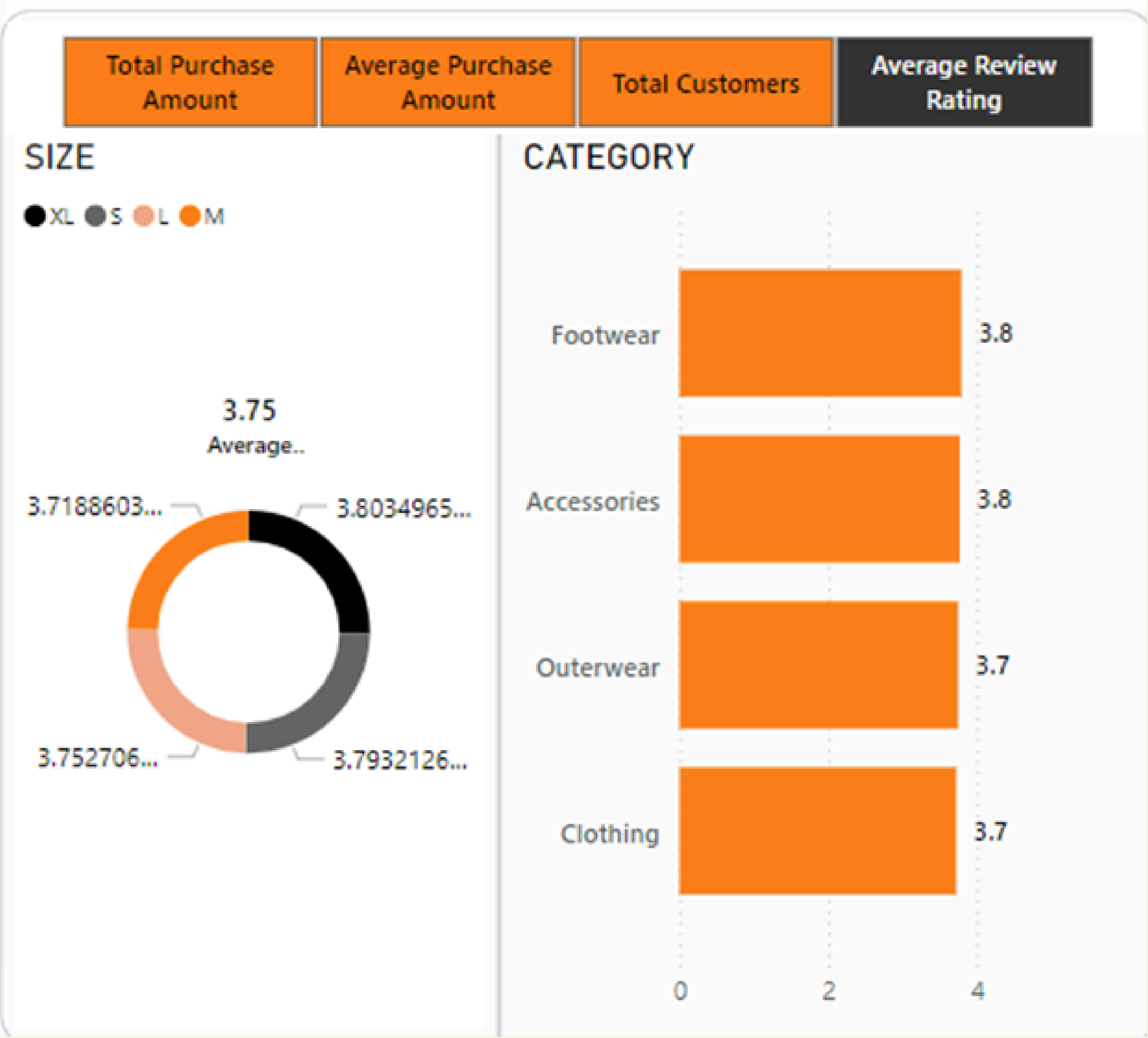


Category	Total Purchase Amount	Total Customers	Average Purchase Amount	Average Review Rating
Accessories	74200	1240	59.84	3.77
Clothing	104264	1737	60.03	3.72
Footwear	36093	599	60.26	3.79
Outerwear	18524	324	57.17	3.75

- Significant increase in purchases between 2015-2023, with peak activity in 2018 and 2022.
- States with the highest purchases include West Virginia, Virginia, and Washington.
- **Size Breakdown :** Medium and Large sizes dominate customer purchases.
- **Top Categories :** Clothing , Accessories
- **Customer Satisfaction :** Accessories have the highest average review rating , while Outerwear has the lowest


Total Purchase Amount
Average Purchase Amount
Total Customers
Average Review Rating
SIZE
● S ● XL ● M ● L
59.76
Average..
58.563152...
61.037707...
59.92421...
60.090909...
CATEGORY
Footwear
60
Clothing
60
Accessories
60
Outerwear
57
0
50
100


Total Purchase Amount
Average Purchase Amount
Total Customers
Average Review Rating
SIZE
● M ● L ● S ● XL
4K
Total...
0K
1K
2K
1K
CATEGORY
Clothing
1.7K
Accessories
1.2K
Footwear
0.6K
Outerwear
0.3K
0K
1K


Total Purchase Amount
Average Purchase Amount
Total Customers
Average Review Rating
SIZE
● XL ● S ● L ● M
3.75
Average..
3.7188603...
3.8034965...
3.752706...
3.7932126...
CATEGORY
Footwear
3.8
Accessories
3.8
Outerwear
3.7
Clothing
3.7
0
2
4

THANK YOU