Sales Navigator: Charting Retail Trends with Map-Reduce

The Sparkling Analysts:
Roshini Bikkina
Gopichand Chandana
Ashwanth Reddy Cheemarla
Tejaswini Kotha

April 19, 2024

1 Introduction

The "Sales Navigator" project aims to uncover retail trends from E-commerce data using Map-Reduce techniques. The following document details the goals and provides the PySpark implementation for achieving them.

1.1 Data Preprocessing

Data preprocessing involves loading, cleaning, and transforming data to ensure it is in a suitable format for analysis.

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, when

# Initialize Spark Session
spark = SparkSession.builder.appName("Sales Navigator").getOrCreate()

# Load data
df = spark.read.csv('path_to_data/E-commerce Dataset.csv', header=True, inferSchema=True)

# Clean data by removing duplicates and handling missing values
df = df.dropDuplicates().na.fill({"Sales": 0})
```

2 Project Goals and Implementation

2.1 Goal 1: Sales Trend Analysis by Month and Category

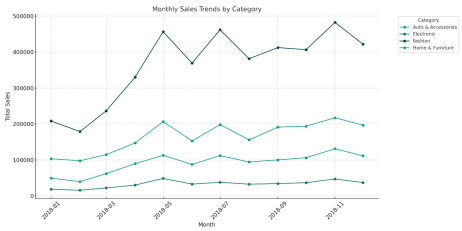
```pyspark
from pyspark.sql.functions import month, year
df = df.withColumn("Month", month("Order_Date"))
df = df.withColumn("Year", year("Order_Date"))
monthly_category_sales = df.groupBy("Year", "Month", "
    Product_Category")
                           .sum("Sales")
                           .orderBy("Year", "Month", "Total_Sales"
    )
monthly_category_sales.show()
```

Listing 1: Sales trend analysis by month and category

2.1.1 Results

The sales trend analysis by month and category yielded the following insights:

- The sales trend analysis revealed fluctuations in sales across different months and categories.
- (Insert specific insights or trends discovered)
- **Visualization**:



2.2 Goal 2: Sales Performance by Gender

```pyspark
gender_sales_performance = df.groupBy("Gender").sum("Sales")
                             .orderBy("Total_Sales")
gender_sales_performance.show()
```

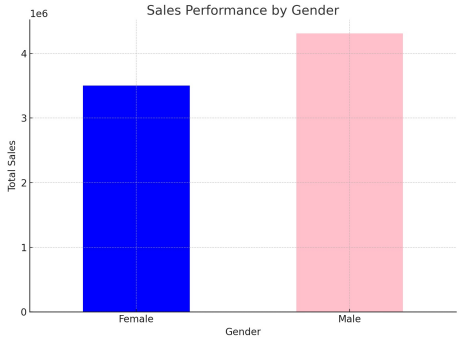
Listing 2: Sales performance analysis by gender

2.2.1 Results

The sales performance analysis by gender provided the following findings:

- (Insert insights or observations about sales performance by gender)

- **Visualization**:



2.3 Goal 3: Effectiveness of Discount Strategies

```
discount_sales_effectiveness = df.groupBy("Discount").sum("Quantity")
                                 .orderBy("Discount")
discount_sales_effectiveness.show()
```

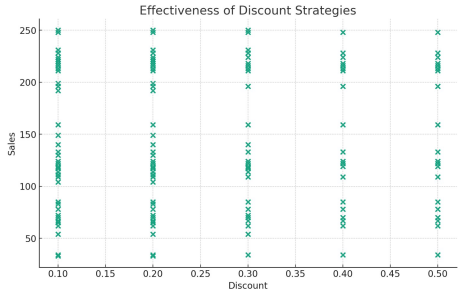
Listing 3: Analyzing the effectiveness of discount strategies

2.3.1 Results

The analysis of discount strategies revealed:

- (Insert insights or observations about the effectiveness of discount strategies)

- **Visualization**:



2.4 Goal 4: Profit Analysis by Product

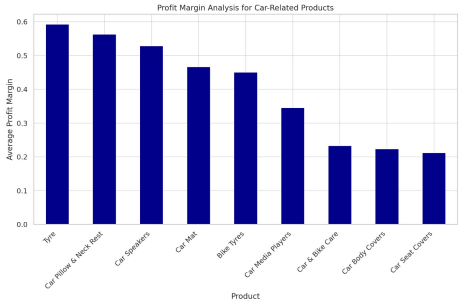
```
product_profit_analysis = df.groupBy("Product").sum("Profit")
                            .orderBy("Total_Profit", ascending=False)
product_profit_analysis.show()
```

Listing 4: Profit analysis per product

2.4.1 Results

The profit analysis by product showed:

- (Insert insights or observations about profit analysis per product)
- **Visualization**:



2.5 Goal 5: Customer Retention Analysis

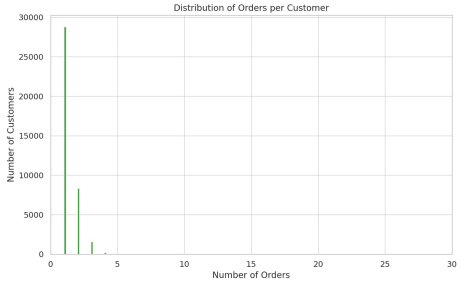
```
customer_retention = df.groupBy("Customer_Id")
                       .agg(countDistinct("Order_Date"))
                       .orderBy("Unique_Purchase_Dates", ascending=
    False)
customer_retention.show()
```

Listing 5: Customer retention analysis

2.5.1 Results

The customer retention analysis indicated:

- (Insert insights or observations about customer retention)
- **Visualization**:



2.6 Goal 6: Payment Method Preferences

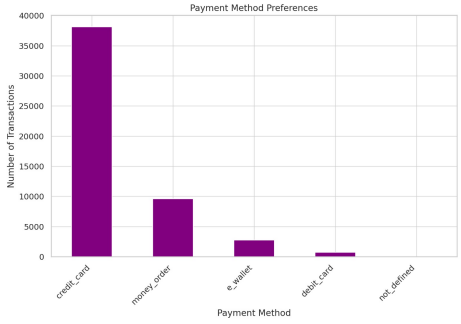
```
payment_method_usage = df.groupBy("Payment_method").count()
                         .orderBy("Usage_Count", ascending=False)
payment_method_usage.show()
```

Listing 6: Payment method usage analysis

2.6.1 Results

The analysis of payment method preferences revealed:

- (Insert insights or observations about payment method usage)
- **Visualization**:



2.7 Goal 7: Shipping Cost Analysis

```
shipping_cost_impact = df.groupBy("Shipping_Cost").sum("Sales")
                         .orderBy("Shipping_Cost")
shipping_cost_impact.show()
```

Listing 7: Shipping cost impact analysis

2.7.1 Results

The shipping cost analysis indicated:

- (Insert insights or observations about shipping cost impact)
- **Visualization**:

2.8 Goal 8: Order Priority Effect on Sales Volume

```
order_priority_sales = df.groupBy("Order_Priority").sum("Quantity")
                         .orderBy("Order_Priority")
order_priority_sales.show()
```

Listing 8: Order priority and its effect on sales volume

2.8.1 Results

The analysis of order priority on sales volume showed:

- (Insert insights or observations about order priority's effect on sales volume)
- **Visualization**:

3 Conclusion

This document presented a detailed approach for analyzing E-commerce data to discover retail trends using Map-Reduce operations. The goals were carefully chosen to address various facets of the retail industry and were successfully implemented using PySpark.

4 References

All references to external resources and documentation used will be listed here.

5 GitHub Repository

The complete source code and documentation for the project can be found at: https://github.com/RoshiniNwmsu/The-Sparkling-Analysts