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

This document provides a comprehensive and detailed explanation of the implementation steps for the project "Sales Navigator: Charting Retail Trends with Map-Reduce," including discussions on the results achieved from the analysis of the E-commerce Dataset.

2 Implementation Steps

2.1 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})

# Normalize sales data
max_sales = df.agg({"Sales": "max"}).collect()[0][0]
df = df.withColumn("Normalized_Sales", col("Sales") / max_sales)
```

2.2 Map-Reduce Processing

Using PySpark's map-reduce capabilities, we analyze sales trends across different dimensions.

```
from pyspark.sql.functions import sum, avg

# Aggregate sales data to compute total sales per product
total_sales_per_product = df.groupBy("Product").agg(sum("Sales").
    alias("Total_Sales"))

# Compute average sales per category
avg_sales_per_category = df.groupBy("Product_Category").agg(avg("
    Sales").alias("Average_Sales"))
```

2.3 Results Visualization

Utilizing Python libraries to create visualizations that provide insights into the data.

```
import matplotlib.pyplot as plt

# Plot total sales per product
total_sales_data = total_sales_per_product.toPandas()
plt.figure(figsize=(10, 8))
plt.bar(total_sales_data['Product'], total_sales_data['Total_Sales'
    ])
plt.xlabel('Product')
plt.ylabel('Total Sales')
plt.title('Total Sales Per Product')
plt.xticks(rotation=45)
plt.show()
```

3 Results Discussion

3.1 Data Quality and the 5Vs

The implementation ensured high data quality and addressed each of the 5Vs of big data, crucial for robust analysis. Volume was managed with Spark, Velocity through real-time data simulation, Variety by integrating multiple data types, Veracity through data cleansing, and Value by deriving actionable insights.

3.2 Performance Metrics

Performance metrics such as latency, processing time, and resource utilization were carefully monitored and optimized. The project achieved low latency and high efficiency in processing, demonstrating the capabilities of our chosen technologies.

4 Conclusions

The project demonstrated how big data technologies could be leveraged to gain significant insights into retail trends, affecting strategic decisions in marketing and product placement. Further research could explore more advanced machine learning techniques for predictive analytics.

5 Citations

- Apache Spark Documentation
- Python Data Science Handbook by Jake VanderPlas

6 GitHub Repository

Link to our GitHub repository: `https://github.com/RoshiniNwmsu/The-Sparkling-Analysts`