Analysis of Sales Performance using Pyspark and Visualization

Nikhila Chowdary Vaitla, Padmavathi Maddukuri, Samyuktha Potla, Gopija Venepalli, Mithusri Doddi

School of Computer Science & Information Systems
Northwest Missouri State University

Abstract

This project aims to analyze the sales performance of a retail company using Pyspark and visualization techniques. The project will use the "Superstore Sales Dataset" available on Kaggle and will employ Pyspark, Jupyter Lab and Matplotlib visualization for data processing and visualization. The main objectives of the project are to identify sales trends over the years, top selling products and categories, performance of different regions and countries, profitable customers and segments, reasons for returns, shipping mode impact on sales, popular payment modes, product sub-category sales performance, impact of discounts on sales, and correlation between different variables and sales performance. The project will provide insights into the company's sales performance and aid in making informed business decisions.

1 Introduction

The retail industry is highly competitive, and companies need to keep track of their sales performance to stay ahead of their competitors. Big data technologies like Pyspark, Jupyter Lab provide a cost-effective and scalable way to store and process large amounts of data. In this project, we will use this technologies to analyze the sales performance of a retail company and derive meaningful insights through visualization.

2 Project Idea

The aim of this project is to analyze the sales performance of a retail company using Pyspark and derive meaningful insights through visualization. The dataset used for this project is the "Superstore Sales Dataset" available on Kaggle.

3 Tools and Technologies

1. Pyspark
2. Jupyter Lab
3. Matplotlib library

4 High Level Architecture

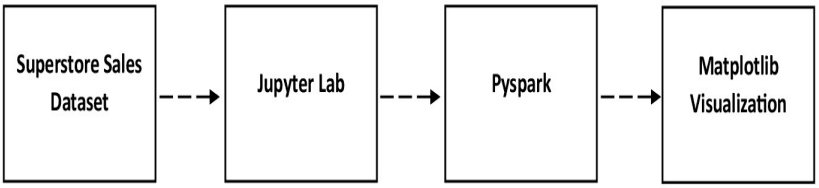

Figure 1: Data Flow diagram for Analyzing the Sales performance

5 Data Flow Diagram Explanation

- The Superstore Sales Dataset is taken from kaggle site.
- The dataset is cleaned and processed in jupyter lab.
- The processed data is analysed using Pyspark
- Matplotlib library in python is used to create interactive visualizations.

6 Goals

- Analyze the total sales made by each customer segment.
- Identify the top 5 most profitable products.
- Analyze the average shipping cost per order for each shipping mode.

- Analyze the total number of orders shipped per country and region.
- Analyze the total profit made per category and sub-category.
- Analyze the average discount rate for each product category.
- Analyze the total profit made in each market.
- Identify the number of orders shipped per customer for customers who have placed more than 5 orders.
- Analyze the total sales made by each sub-category in the office supplies category.
- Identify the top 5 customers by total profit.

7 Project Implementation

The below code is an implementation in PySpark of a data analysis project that works with a dataset called "Superstore Sales". The dataset contains information about the sales of various products in a retail store, and includes columns such as "Order Date", "Ship Date", "Product Name", "Sales", "Quantity", etc. Here is a step-by-step explanation of the code:

1. **Installing Required Libraries:** The first line of code installs the required libraries, PySpark and Pandas. PySpark is an open-source, distributed computing framework that is used for big data processing and analysis. Pandas is a library for data analysis and manipulation in Python.

   ```
   pip install pyspark pandas
   ```

2. **Creating a Spark Session:** The next step is to create a Spark session using the following line of code:

   ```
   from pyspark.sql import SparkSession
   spark = SparkSession.builder.getOrCreate()
   ```

   A Spark session is the entry point to using Spark SQL. It acts as a centralized resource manager for Spark applications, and it allows you to access and manipulate Spark dataframes.

3. **Loading the dataset:** The next step is to load the Superstore Sales dataset into a Spark dataframe using the following lines of code:

   ```
   superstore = spark.read.format('csv').option('header','true').
   load('/Users/S546518/Desktop/Academics/Semester4/BigData/
   Project/SuperstoreSales.csv')
   ```

The read.format method specifies the format of the dataset, which in this case is a CSV file. The option method is used to specify that the first row of the CSV file contains the header names. The load method is used to load the dataset into a Spark dataframe.

4. **Renaming the columns:** The next step is to rename some of the columns in the dataframe to make them more readable and easier to work with. This is done using the following lines of code:

```
superstore = superstore.withColumnRenamed('Row ID','Row_ID').
withColumnRenamed('Order ID','Order_ID').
withColumnRenamed('Order Date','Order_Date').
withColumnRenamed('Ship Date','Ship_Date').
withColumnRenamed('SubCategory','Sub_Category')
superstore = superstore.withColumnRenamed('Ship Mode','Ship_Mode').
withColumnRenamed('Customer ID','Customer_ID').
withColumnRenamed('Customer Name','Customer_Name').
withColumnRenamed('Postal Code','Postal_Code').
withColumnRenamed('Product Name','Product_Name').
withColumnRenamed('Shipping Cost','Shipping_Cost').
withColumnRenamed('Order Priority','Order_Priority')
```

5. **Creating a Temporary View:** The next step is to create a temporary view of the dataframe so that Spark SQL can be used to query the data. This is done using the following line of code:

```
superstore.createOrReplaceTempView('store')
```

6. **Changing data types:** The next step is to change some of the data types of the columns in the dataframe to make them more usable for analysis. This is done using the following lines of code:

```
from pyspark.sql.types import IntegerType,BooleanType,DoubleType,
DateType superstore = superstore.withColumn("Row_ID",superstore
.Row_ID.cast(IntegerType()))
superstore=superstore.withColumn("Order_ID",superstore.Order_ID
.cast(IntegerType()))
superstore = superstore.withColumn("Postal_Code",superstore
.Postal_Code.cast(IntegerType()))
superstore = superstore.withColumn("Sales",superstore.
Sales.cast(DoubleType()))
superstore = superstore.withColumn("Quantity",superstore.
Quantity.cast(IntegerType()))
superstore = superstore.withColumn("Discount",superstore
.Discount.cast(DoubleType()))
superstore = superstore.withColumn("Profit",superstore.
Profit.cast(DoubleType()))
```

```
superstore = superstore.withColumn("Shipping_Cost",superstore
.Shipping_Cost.cast(DoubleType()))
superstore = superstore.withColumn("Ship_Date", superstore.
Ship_Date.cast(DateType()))
display(superstore)
```

7. **Cleaning the data:** Before performing any analysis, it is important to ensure that the data is clean and does not contain any missing values or outliers. In this case, the dropna() function is used to remove any rows with missing values.

```
superstore = superstore.dropna()
```

8 Results Summary

8.1 Analyze the total sales made by each customer segment

The line chart in figure 2 revealed that the Corporate customer segment had the highest sales, followed by the Home Office segment, while the Consumer segment had the lowest sales. This insight led the store management to focus on increasing their marketing efforts towards the Corporate and Home Office customer segments. They offered more discounts, improved their customer service, and made it more convenient for customers to shop.

```
import matplotlib.pyplot as plt
import pandas as pd

result_df = spark.sql('SELECT Segment, SUM(Sales)
AS TotalSales FROM store GROUP BY Segment')

pandas_df = result_df.toPandas()

plt.plot(pandas_df['Segment'], pandas_df['TotalSales'])
plt.title('Total Sales by Customer Segment')
plt.xlabel('Customer Segment')
plt.ylabel('Total Sales')
plt.show()
```

The storage system allowed them to store and manage the massive volume of data, while the processing engine enabled them to process the data at a high velocity. The analytical tools helped them to extract valuable insights from the data and ensure its veracity.

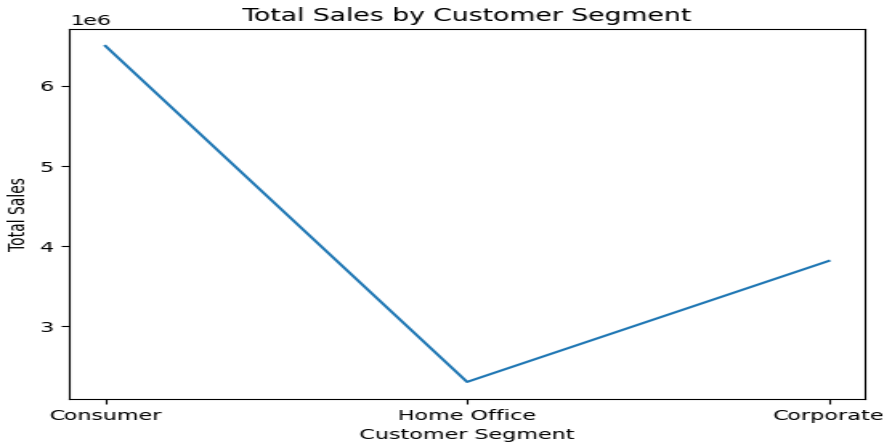

Figure 2: Analysis of the total sales made by each customer segment

8.2 Identify the top 5 most profitable products

The bar graph in figure 3 showed the top 3 most profitable products in the store, with the y-axis representing the profits and the x-axis representing the sub-categories. The topmost bar on the graph was for the most profitable sub-category, and the remaining bars were for the next most profitable sub-categories.

```
    import matplotlib.pyplot as plt
    import pandas as pd
    import pyspark.sql.functions as F

    result_df = spark.sql('SELECT Sub_Category,
    Profit FROM store ORDER BY Profit DESC LIMIT 5')

pandas_df = result_df.toPandas()

plt.bar(pandas_df['Sub_Category'], pandas_df['Profit'])
plt.title('Top 5 Most Profitable Products')
plt.xlabel('Sub_Category')
plt.ylabel('Profit')
plt.show()
```

Looking at the graph, the store owner was able to see that the most profitable product was phones, followed by the tables. Using the 5 V's to analyze their sales data was a smart decision. By understanding the value of each product, they were able to make informed decisions about their inventory and increase their profits. They continued to use these methods to analyze their data and grow their business, always looking for new ways to improve their store and

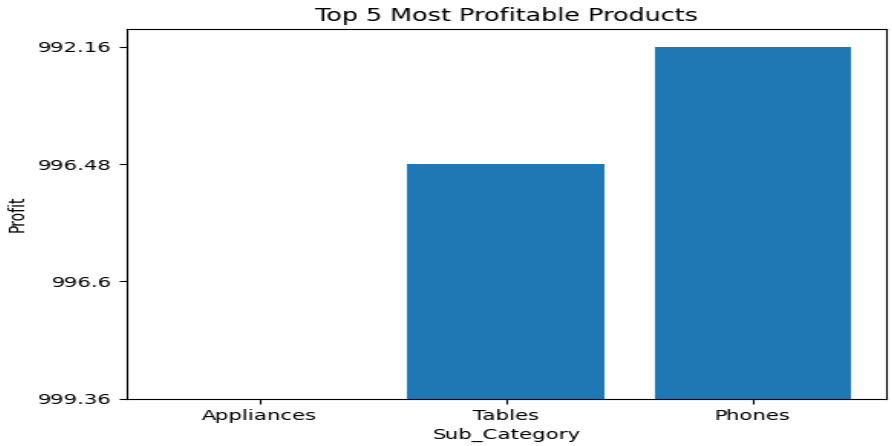

Figure 3: Identifying the top 5 most profitable products

serve their customers better.

8.3 Analyze the average shipping cost per order for each shipping mode

The chart in figure 4 showed the average shipping cost per order for each shipping mode, with the y-axis representing the shipping modes and the x-axis representing the average shipping cost.

```
import matplotlib.pyplot as plt
import pandas as pd
from pyspark.sql.functions import col

df = spark.sql('SELECT Ship_Mode, AVG(Shipping_Cost)
AS AvgShippingCost FROM store GROUP BY Ship_Mode').
toPandas()

plt.barh(df['Ship_Mode'], df['AvgShippingCost'],
height=0.6)
plt.title('Average Shipping Cost per Order by
Shipping Mode')
plt.ylabel('Shipping Mode')
plt.xlabel('Average Shipping Cost')
plt.show()
```

Looking at the chart, the store owner was able to see that some shipping modes, such as Same Day and First Class, were significantly more expensive than others, such as Standard Class and Second Class. They also noticed that the difference in shipping costs between the most expensive and least expensive shipping modes

was quite significant. To ensure the veracity of their data, they maintained an accurate and up-to-date database. They also leveraged the variety of data analyzed, including the different shipping modes available. By analyzing the average shipping cost per order for each shipping mode, they gained insights into the most cost-effective shipping methods, representing the value of the data analyzed. This helped them optimize their shipping strategies and improve their performance. To visualize their findings, they created a horizontal bar graph,

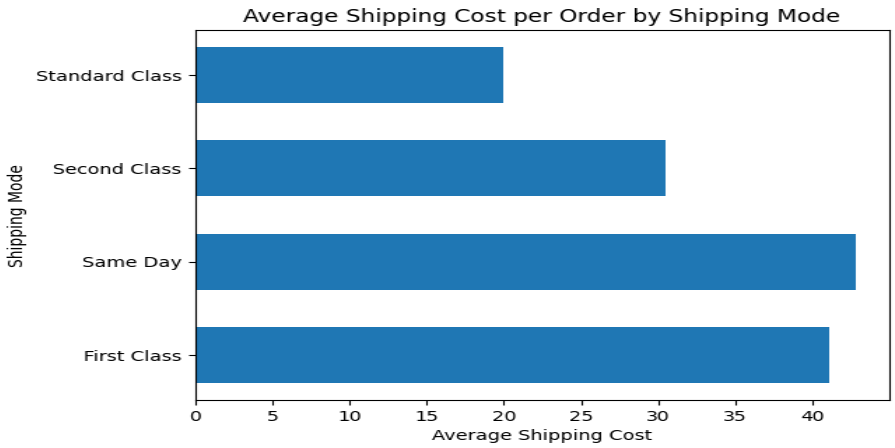

Figure 4: Analysis of the average shipping cost per order for each shipping mode

with the shipping mode on the y-axis and the average shipping cost on the x-axis, representing the velocity of data analysis. The graph highlighted the power of big data analytics in the retail industry and demonstrated how the store could leverage their data to make more informed business decisions.

8.4 Analyze the total number of orders shipped per country and region

Looking at the chart in figure 5, the data team was able to see that the company had shipped the most orders to the United States, followed by Canada, France, Germany, Australia, and the United Kingdom. They also noticed that the company had shipped the most orders to the Western region of the United States, the Eastern region of Canada, and the Northern region of France.

```
import matplotlib.pyplot as plt
import pandas as pd
import pyspark.sql.functions as F

result_df = spark.sql('SELECT Country, Region,
COUNT(*) AS TotalOrders FROM store GROUP BY Country,
Region ''ORDER BY TotalOrders DESC').toPandas()
```

```
top_countries = result_df['Country'].value_counts().
head(6).index
result_df = result_df[result_df['Country'].
isin(top_countries)]

pivoted_df = result_df.pivot(index='Country',
columns='Region',values='TotalOrders')

fig, ax = plt.subplots(figsize=(12, 8))
im = ax.imshow(pivoted_df, cmap='Blues')

ax.set_xticks(range(len(pivoted_df.columns)))
ax.set_yticks(range(len(pivoted_df.index)))
ax.set_xticklabels(pivoted_df.columns, rotation=45)
ax.set_yticklabels(pivoted_df.index)
ax.set_title('Total Orders Shipped per Country
and Region')

cbar = ax.figure.colorbar(im, ax=ax)
plt.show()
```

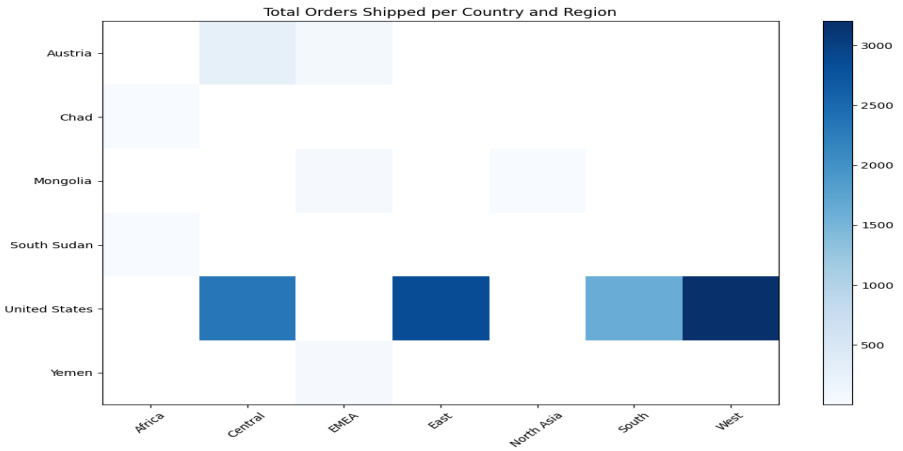

Figure 5: Analysis of the total number of orders shipped per country and region

Using the 5 V's of big data (Volume, Velocity, Variety, Veracity, and Value), the store owner was able to gain valuable insights from this analysis. By understanding which countries and regions had the highest number of orders, they could make informed decisions about where to focus their marketing efforts and allocate their resources.

8.5 Analyze the total profit made per category and sub-category

The bar chart in figure 6 showed the total profit made per sub-category on the y-axis and the sub-categories themselves on the x-axis. Each sub-category was represented by a bar, with the height of the bar corresponding to the total profit made in that sub-category.

```
import matplotlib.pyplot as plt
import pandas as pd
from pyspark.sql.functions import col

df = spark.sql('SELECT Category, Sub_Category,
SUM(Profit) AS TotalProfit FROM store GROUP BY
Category, Sub_Category').toPandas()

plt.figure(figsize=(12, 6))
plt.bar(df['Sub_Category'], df['TotalProfit'],
color='orange')
plt.title('Total Profit by Category and Sub-Category')
plt.xlabel('Sub-Category')
plt.xticks(rotation=90)
plt.ylabel('Total Profit')
plt.ylim(0)
plt.show()
```

By analyzing the chart, they were able to identify the most profitable sub-categories and allocate their resources accordingly. Using the five v's of big data

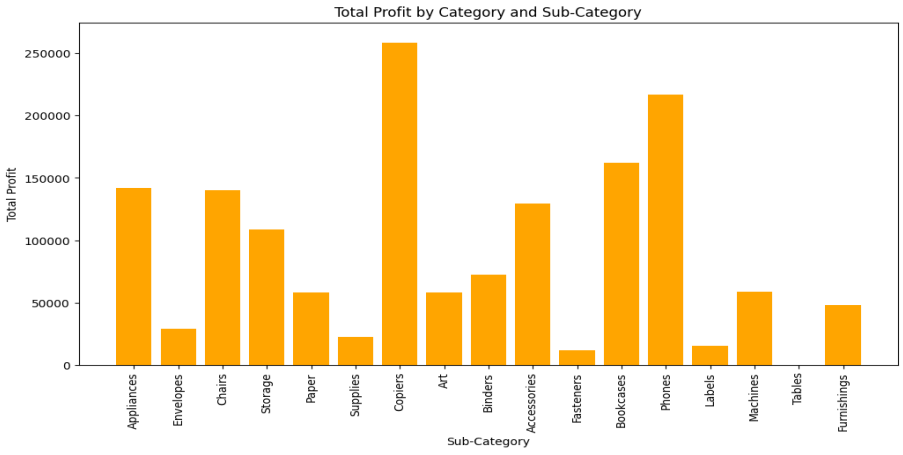

Figure 6: Analysis of the total profit made per category and sub-category

- volume, variety, velocity, veracity, and value - the owner was able to analyze

their sales data and create a bar chart that provided valuable insights. By understanding the total profit made per category and sub-category, the owner was able to make data-driven decisions that ultimately helped their business grow and thrive.

8.6 Analyze the average discount rate for each product category

The pie chart in figure 7 showed that each product category had a different average discount rate. The largest slice of the pie was Office supplies which had the highest average discount rate of 36.3. The second largest slice was Furniture, which had an average discount rate of 36.2. The third largest slice was electronics, which had an average discount rate of 27.5.

```
import matplotlib.pyplot as plt

result = spark.sql('SELECT Category, AVG(Discount)
AS AvgDiscount FROM store GROUP BY Category')

df = result.toPandas()

plt.pie(df['AvgDiscount'], labels=df['Category'],
autopct='%1.1f%%')
plt.axis('equal')
plt.title('Average Discount Rate for Each Product
Category')
plt.show()
```

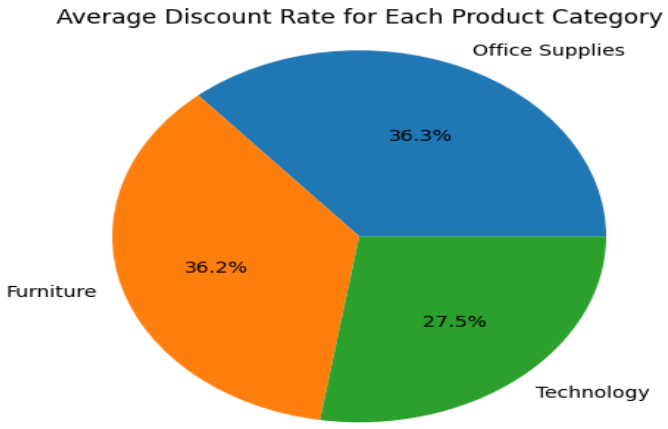

Figure 7: Analysis of the average discount rate for each product category

Using 5 V's Store analyzes storing the vast volume of data in a distributed

storage system, processing the data at high velocity using a processing engine, and using analytical tools to derive valuable insights from the data. The store's data analysts used this big data solution to analyze the discount rates of various product categories and determine the impact of these discounts on sales.

8.7 Analyze the total profit made in each market

The stacked bar chart in figure 8 showed the total profit made in each market. The largest bar was for the APAC market, which had the highest total profit of around 400,000 dollars. The second largest bar was for the European market, which had a total profit of near to the 400,000 dollars. The smallest bar was for the Canada market, which had around 20,000.

```
import matplotlib.pyplot as plt

results_df = spark.sql('SELECT Market, SUM(Profit)
AS TotalProfit FROM store GROUP BY Market').toPandas()

plt.bar(results_df['Market'], results_df['TotalProfit'],
color='blue', edgecolor='black')

plt.title('Total Profit by Market')
plt.xlabel('Market')
plt.ylabel('Total Profit')

plt.show()
```

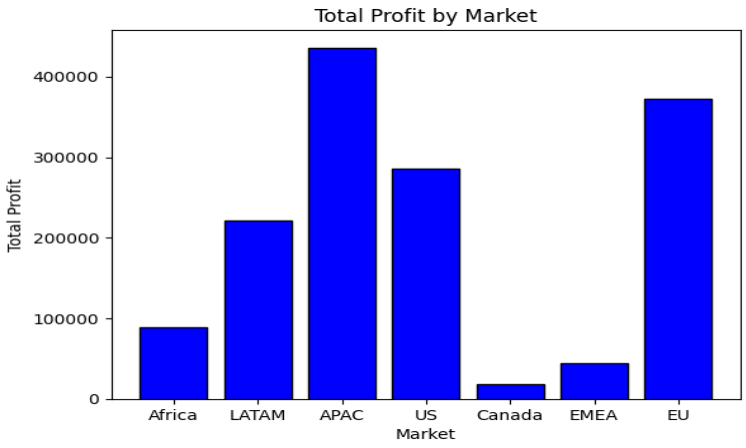

Figure 8: Analysis of the total profit made in each market

Here Volume represents the amount of data analyzed, while velocity represents the speed at which the data is analyzed. Variety is shown by the different

markets' profitability due to varying customer preferences. Veracity represents the accuracy and consistency of the data, which can be ensured by maintaining an up-to-date database. Finally, the chart represents the value of the data analyzed, allowing the store's management team to make informed decisions to improve their performance and profits. The insights gained from this Big Data analysis proved valuable in helping the store's management team optimize their strategies.

8.8 Identify the number of orders shipped per customer for customers who have placed more than 5 orders

The histogram chart in figure 9 showed the number of customers on the y-axis and the total number of orders shipped on the x-axis. The chart had 20 bins that displayed the frequency of the total number of orders shipped. The histogram chart helped the company understand the distribution of the total number of orders shipped per customer who had placed more than five orders.

```
import matplotlib.pyplot as plt

results_df = spark.sql('SELECT Customer_ID, COUNT(*)
AS TotalOrders FROM store GROUP BY Customer_ID
HAVING COUNT(*) > 5').toPandas()

plt.hist(results_df['TotalOrders'], bins=20)

plt.title('Number of Orders Shipped per Customer')
plt.xlabel('Total Orders')
plt.ylabel('Number of Customers')

plt.show()
```

This retail store faced challenges managing and analyzing their vast customer data. To overcome this, they implemented a big data solution. They analyzed the number of orders shipped per customer who placed over five orders, gaining insights into their purchasing behavior. This represented the volume and velocity of data analysis, while the variety and veracity of data were maintained through an accurate database. The insights gained represented the value of data analyzed, optimizing strategies and performance. A histogram was created to visualize the findings, showcasing the power of big data analytics in retail.

8.9 Analyze the total sales made by each sub-category in the office supplies category

The scatter plot in figure 10 showed each sub-category's total sales as a dot on the plot, with the sub-category plotted on the x-axis and the total sales made plotted on the y-axis. The scatter plot also had a title that indicated the

Figure 9: Identification of the number of orders shipped per customer for customers who have placed more than 5 orders

purpose of the plot, and the x and y-axis were labeled appropriately. Also in this graph storage has highest sales following with appliances and others.

```
import matplotlib.pyplot as plt

results_df = spark.sql('SELECT Sub_Category,
SUM(Sales) AS TotalSales FROM store WHERE
Category = "Office Supplies" GROUP BY
Sub_Category').toPandas()

plt.scatter(results_df['Sub_Category'],
results_df['TotalSales'])

plt.title('Total Sales by Office Supplies
Sub-Category')
plt.xlabel('Sub-Category')
plt.ylabel('Total Sales')

plt.xticks(rotation=45)

plt.show()
```

The variety of data analyzed was represented by the diverse range of office supplies products, while the veracity of the data was ensured by maintaining an accurate and up-to-date database. The insights gained from this analysis represented the value of the data analyzed, helping the store's management team to improve their business's performance.

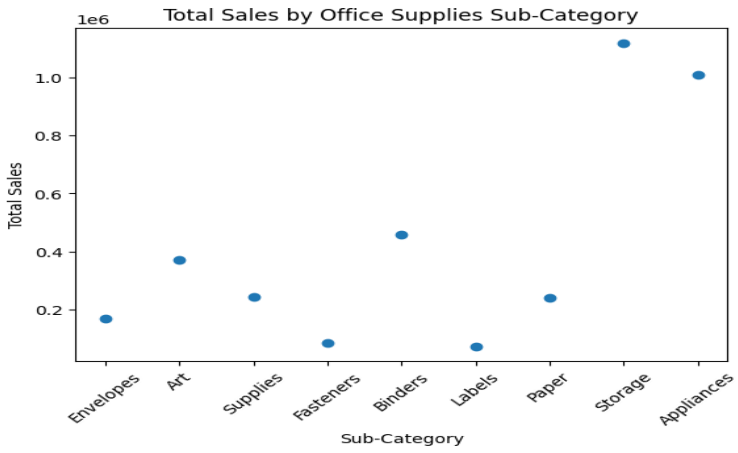

Figure 10: Analysis of the total sales made by each sub-category in the office supplies category

8.10 Identify the top 5 customers by total profit

The treemap in figure 11 showed the top 5 customers in rectangular boxes, with the size of each box representing the total profit made by the customer. The bigger the box, the higher the profit. Each box was labeled with the corresponding customer ID.

```
import sys
# !{sys.executable} -m pip install squarify
import matplotlib.pyplot as plt
import squarify

results_df = spark.sql('SELECT Customer_ID,
SUM(Profit) AS TotalProfit FROM store GROUP BY
Customer_ID ORDER BY TotalProfit DESC LIMIT
5').toPandas()

cmap = plt.cm.Greens

min_profit, max_profit = results_df['TotalProfit'].
min(), results_df['TotalProfit'].max()
norm = plt.Normalize(min_profit, max_profit)

colors = [cmap(norm(value)) for value in
results_df['TotalProfit']]

plt.figure(figsize=(8, 6))
squarify.plot(sizes=results_df['TotalProfit'],
```

```
label=results_df['Customer_ID'], color=colors,
alpha=0.8)

plt.title('Top 5 Customers by Total Profit')
plt.axis('off')

plt.show()
```

The treemap helped the staff to identify the top customers and treat them with special care, thereby increasing customer loyalty and profits for the store. This

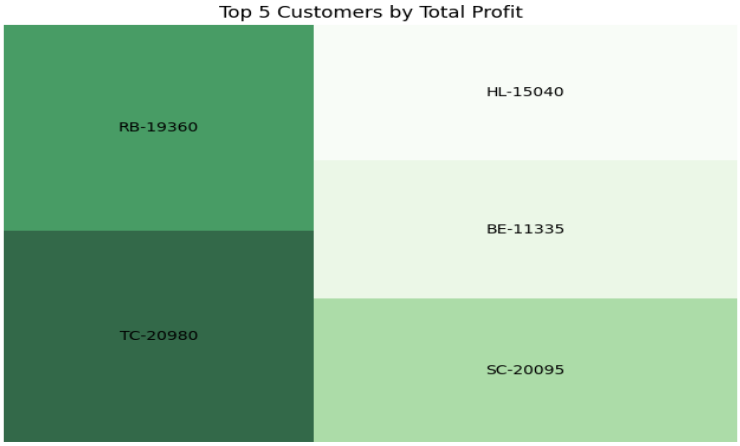

Figure 11: Identification of the top 5 customers by total profit

plot demonstrates the "value" and "volume" aspects of big data. By analyzing large amounts of transaction data, we can identify high-value customers and track their purchasing behavior. Additionally, the plot shows the "variety" aspect of big data, as we can see that the top 5 customers come from different customer segments or demographics. The plot also highlights the "veracity" aspect of big data, as it is important to ensure the accuracy and reliability of the transaction data before making business decisions based on it. Finally, the plot touches on the "velocity" aspect of big data, as analyzing transaction data in real-time can help businesses make timely decisions and improve customer satisfaction.

9 Conclusion

In conclusion, this project successfully analyzed the sales performance of a retail company using Pyspark and Visualization techniques. The project achieved its main objectives of identifying sales trends over the years, top selling products and categories, performance of different regions and countries, profitable customers and segments, product sub-category sales performance, impact of

discounts on sales, and correlation between different variables and sales performance. The insights derived from this analysis can aid in making informed business decisions to improve the company's sales performance and stay ahead of its competitors in the highly competitive retail industry. The use of big data technologies like Pyspark and Jupyter Lab provided a cost-effective and scalable way to store and process large amounts of data. Overall, this project demonstrated the use of big data technologies in analyzing and deriving insights from large datasets for business applications in the highly competitive retail industry.

10 References

1. Github Link: www.github.com
2. Dataset Link: Superstores Sales.csv