Superstore Sale Analysis

Abstract

The SuperStore Spark application is a robust data analytics tool designed to analyze and derive insights from sales data stored in a MySQL database. Leveraging Apache Spark, the application integrates seamlessly with the database to perform a wide array of analyses, including null checks, category distribution, sales and profit/loss assessments, discount strategy evaluations, duplication checks, and various summaries such as yearly and monthly sales. The tool also offers functionalities for item details retrieval, total sales revenue calculation, day-wise profit/loss analysis, identification of top-selling items, and examination of discount counts and peak sales hours. By providing a comprehensive set of analyses, the SuperStore application equips decision-makers with valuable information for optimizing business strategies, enhancing sales performance, and maximizing revenue potential.

MOTIVATION OF PROJECT

The motivation behind developing the SuperStore Spark application stems from the increasing importance of data-driven decision-making in the retail sector. In today's competitive business landscape, understanding and leveraging data are critical for optimizing sales strategies, enhancing operational efficiency, and staying ahead of market trends. The SuperStore project addresses the need for a versatile and scalable analytics tool specifically tailored for sales data analysis. By utilizing Apache Spark and integrating seamlessly with a MySQL database, the application empowers business stakeholders to gain actionable insights into various facets of their operations, ranging from null value detection and category distribution to detailed sales performance assessments. The project's motivation lies in providing a user-friendly and efficient solution that enables businesses to make informed decisions, improve overall performance, and stay agile in the dynamic retail environment.

INTRODUCTION

In the contemporary landscape of retail, data-driven insights play a pivotal role in guiding strategic decisions and optimizing operational efficiency. Recognizing the growing significance of analytics in the retail sector, the SuperStore Spark application has been developed as a powerful tool for comprehensive analysis of sales data. Leveraging the capabilities of Apache Spark and seamless integration with a MySQL database, this application addresses the critical need for a versatile platform that enables businesses to extract actionable insights from their sales datasets. With functionalities ranging from null value checks, category distribution analysis, and sales performance assessments to detailed evaluations of discount strategies and peak sales hours, SuperStore empowers decision-makers with the tools necessary to make informed choices, improve overall business performance, and navigate the dynamic challenges of the retail industry. This introduction sets the stage for understanding the pivotal role SuperStore plays in aiding businesses to harness the power of data for strategic advantage and sustained growth.

Database Setup and Data Loading Report:

The provided SQL script outlines the creation of a MySQL database named vegetable_db along with the creation of four tables (item_table, sales_table, whole_sale, and loss). Additionally, the script loads data into these tables from CSV files. Here is a breakdown of the process:

Database and Tables Creation:

- The script begins by displaying existing databases and creating a new one named vegetable_db.
- The focus then shifts to vegetable_db, where the script shows the existing tables or an empty set if no tables are present.
- The script includes commented-out lines for dropping existing tables, providing flexibility for maintenance.

Table Definitions:

- **Table item_table:** Defines attributes for items, including Item_Code, Item_Name, Category_Code, and Category_Name.
- **Table sales_table:** Defines attributes for sales transactions, including sale_id, Date, Time, Item_Code, Quantity_Sold, Unit_Selling_Price, Sale_or_Return, and Discount. It establishes a foreign key relationship with item_table.
- **Table whole_sale:** Defines attributes for wholesale transactions, including Date, Item_Code, and Wholesale_Price. It establishes a foreign key relationship with item_table.
- **Table loss:** Defines attributes for loss rates, including Item_Code, Item_Name, and Loss_Rate.

Table Data Loading:

- The script uses the LOAD DATA INFILE command to load data from CSV files into respective tables.
- CSV files include Item_category.csv for item_table, whole_sale.csv for whole_sale, loss_rate.csv for loss, and sales.csv for sales_table.
- Each LOAD DATA INFILE command specifies the file path, field and line terminators, and the number of lines to be ignored (headers).

Integration with Database

The application integrates with a MySQL database named "vegetable_db" using the JDBC connector. The database connection details, such as URL, username (root), and password (1234567890), are specified in the code. The data is loaded into Spark DataFrames using the get_table method, which leverages Spark's JDBC data source.

Analysis and Methodology

1. Null Checks:

Method: checkNulls

Purpose: Identifies and prints the number of null values in a specified column (Item_Code in this case).

2. Category Count:

Method: categoryCount

Purpose: Joins the sales and items DataFrames on the Item_Code column and calculates the count of each category.

3. Total Sales and Profit/Loss Yearly:

Method: totalSalesAndProfitLossYearly

Purpose: Computes the total quantity sold and profit/loss on a yearly basis for a specified item code.

4. Evaluate Discount Effectiveness:

Method: evaluateDiscountEffectiveness

Purpose: Calculates the average quantity sold for different discount levels, aiding in evaluating the effectiveness of discount strategies.

5. Duplication Checks:

Method: checkDuplicates

Purpose: Identifies and prints the presence of duplicate values in the specified column (Item_Code).

6. Total Yearly Sales:

Method: totalYearlySales

Purpose: Calculates the total quantity and revenue on a yearly basis.

7. Total Sales and Profit/Loss Monthly:

Method: totalSalesAndProfitLossMonthly

Purpose: Computes the total quantity sold and profit/loss on a monthly basis for a specific item code.

8. Count 'Sale or Return':

Method: countSaleOrReturn

Purpose: Counts the occurrences of 'Sale' or 'Return' in the Sale_or_Return column.

9. Get Item Details:

Method: item_details

Purpose: Retrieves and prints details (Item_Name, Category_Code, Category_Name) of a specific item code.

10. Total Sales Revenue:

Method: totalSalesRevenue

Purpose: Calculates the total revenue by multiplying the quantity sold with the unit selling price and orders the result in descending order.

11. Add a 'Day' Column to dfSales:

Method: addDayColumn

Purpose: Adds a new column ('Day') to the sales DataFrame, indicating the day of the week based on the sales date.

12. Total Profit/Loss Day Wise:

Method: totalProfitLossDayWise

Purpose: Computes the total profit/loss on a day-wise basis by joining sales and wholesale DataFrames.

13. Top 10 'Item Name':

Method: topItems

Purpose: Identifies and prints the top 10 selling items based on the total quantity sold.

14. Count 'Discount':

Method: countDiscount

Purpose: Counts the occurrences of different discount levels.

15. Sales Time Analysis (Peak Sales Hours):

Method: peakSalesHours

Purpose: Identifies and prints the peak sales hours by grouping sales data based on the hour of the day.

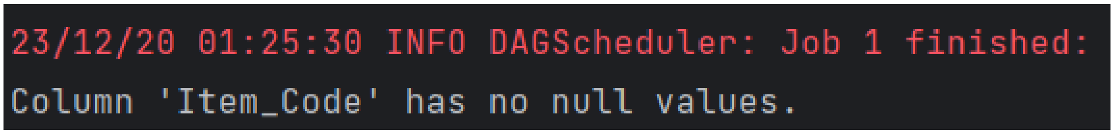
Main Method Execution:

The main method demonstrates the application of the defined methods on specific DataFrames (df_items, df_sales, df_wholesale, df_lossrate) retrieved from the MySQL database.

Each method call is accompanied by a print statement providing a description of the analysis being performed.

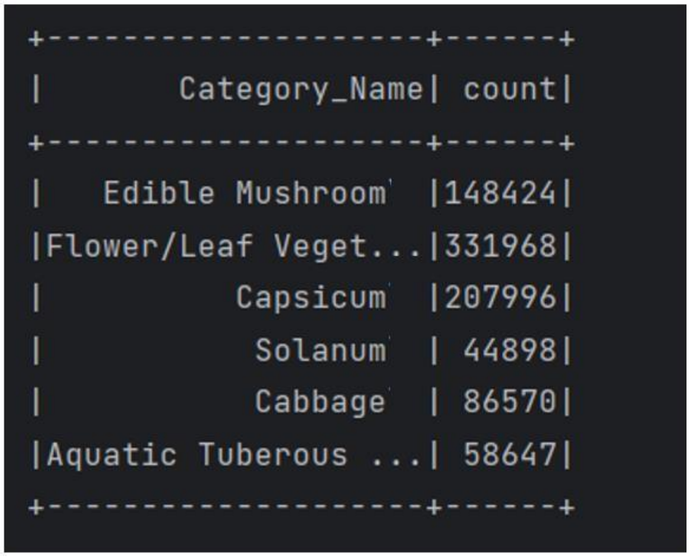
Results:

1. Null Checks:



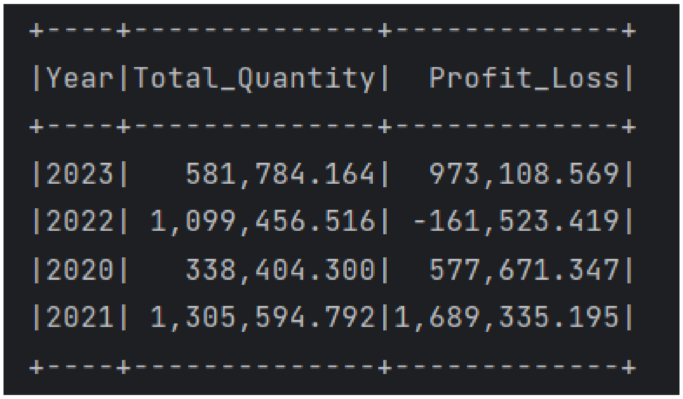
```
23/12/20 01:25:30 INFO DAGScheduler: Job 1 finished:
Column 'Item_Code' has no null values.
```

2. Category Count:



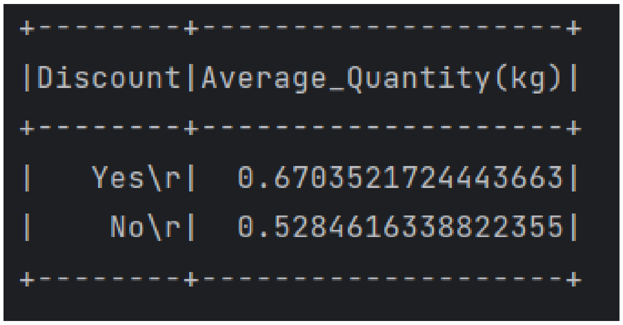
```
+--------------------+------+
|       Category_Name| count|
+--------------------+------+
|   Edible Mushroom  |148424|
|Flower/Leaf Veget...|331968|
|          Capsicum  |207996|
|           Solanum  | 44898|
|           Cabbage  | 86570|
|Aquatic Tuberous ...| 58647|
+--------------------+------+
```

3. Total Sales and Profit/Loss Yearly:



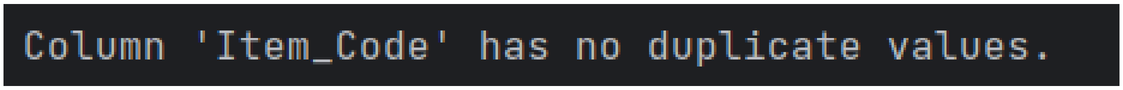
```
+----+--------------+-------------+
|Year|Total_Quantity|  Profit_Loss|
+----+--------------+-------------+
|2023|   581,784.164|  973,108.569|
|2022| 1,099,456.516| -161,523.419|
|2020|   338,404.300|  577,671.347|
|2021| 1,305,594.792|1,689,335.195|
+----+--------------+-------------+
```

4. Evaluate Discount Effectiveness:



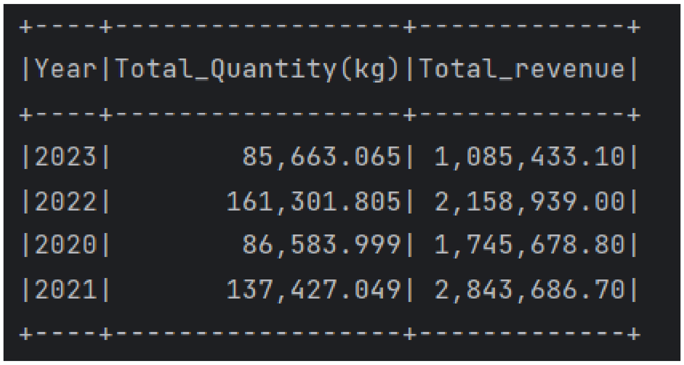
```
+--------+--------------------+
|Discount|Average_Quantity(kg)|
+--------+--------------------+
|   Yes\r|  0.6703521724443663|
|    No\r|  0.5284616338822355|
+--------+--------------------+
```

5. Duplication Checks:



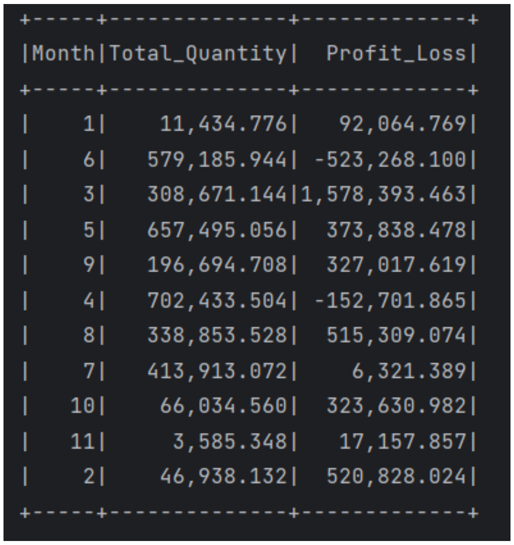
```
Column 'Item_Code' has no duplicate values.
```

6. Total Yearly Sales:



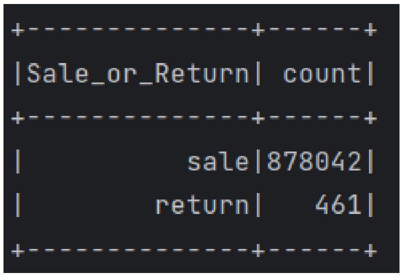
```
+----+-------------------+-------------+
|Year|Total_Quantity(kg)|Total_revenue|
+----+-------------------+-------------+
|2023|         85,663.065| 1,085,433.10|
|2022|        161,301.805| 2,158,939.00|
|2020|         86,583.999| 1,745,678.80|
|2021|        137,427.049| 2,843,686.70|
+----+-------------------+-------------+
```

7. Total Sales and Profit/Loss Monthly:



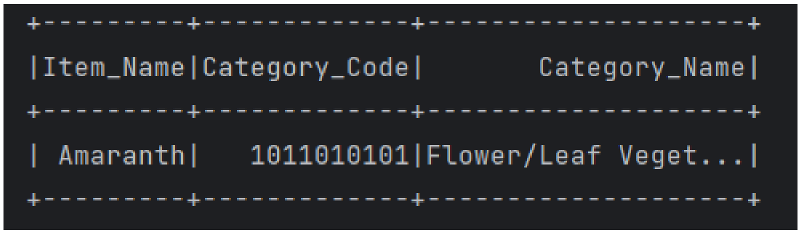
```
+-----+--------------+-------------+
|Month|Total_Quantity|  Profit_Loss|
+-----+--------------+-------------+
|    1|    11,434.776|   92,064.769|
|    6|   579,185.944| -523,268.100|
|    3|   308,671.144|1,578,393.463|
|    5|   657,495.056|  373,838.478|
|    9|   196,694.708|  327,017.619|
|    4|   702,433.504| -152,701.865|
|    8|   338,853.528|  515,309.074|
|    7|   413,913.072|    6,321.389|
|   10|    66,034.560|  323,630.982|
|   11|     3,585.348|   17,157.857|
|    2|    46,938.132|  520,828.024|
+-----+--------------+-------------+
```

8. Count 'Sale or Return':



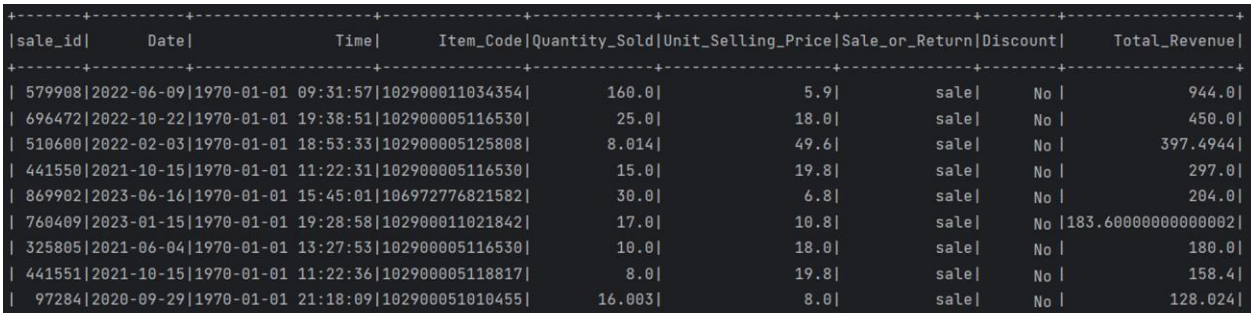
```
+--------------+------+
|Sale_or_Return| count|
+--------------+------+
|          sale|878042|
|        return|   461|
+--------------+------+
```

9. Get Item Details:



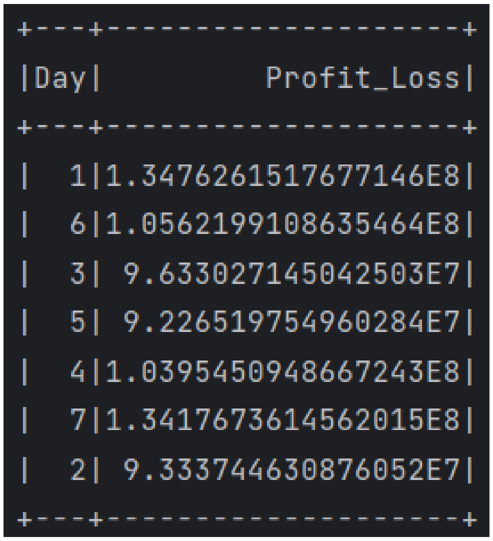
```
+---------+-------------+--------------------+
|Item_Name|Category_Code|       Category_Name|
+---------+-------------+--------------------+
| Amaranth|   1011010101|Flower/Leaf Veget...|
+---------+-------------+--------------------+
```

10. Total Sales Revenue:



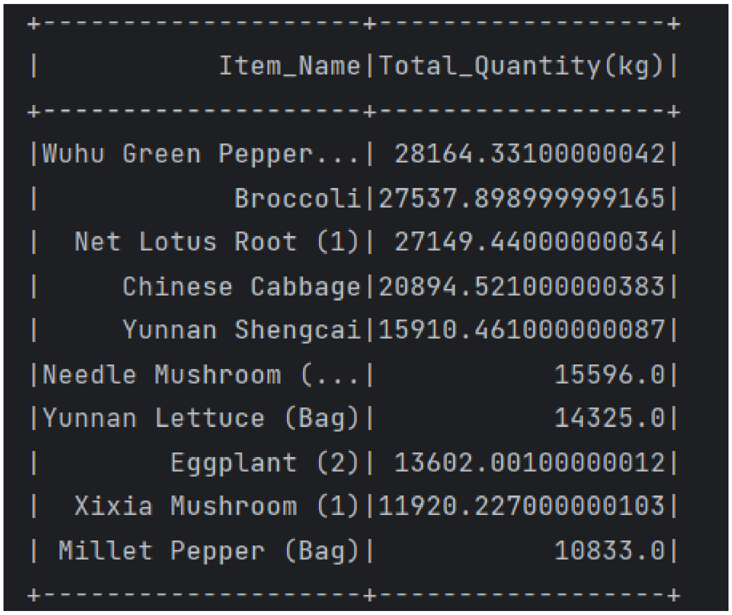
```
+-------+----------+-------------------+---------------+-------------+------------------+--------------+--------+------------------+
|sale_id|      Date|               Time|      Item_Code|Quantity_Sold|Unit_Selling_Price|Sale_or_Return|Discount|     Total_Revenue|
+-------+----------+-------------------+---------------+-------------+------------------+--------------+--------+------------------+
| 579908|2022-06-09|1970-01-01 09:31:57|102900011034354|        160.0|               5.9|          sale|     No |             944.0|
| 696472|2022-10-22|1970-01-01 19:38:51|102900005116530|         25.0|              18.0|          sale|     No |             450.0|
| 510600|2022-02-03|1970-01-01 18:53:33|102900005125808|        8.014|              49.6|          sale|     No |          397.4944|
| 441550|2021-10-15|1970-01-01 11:22:31|102900005116530|         15.0|              19.8|          sale|     No |             297.0|
| 869902|2023-06-16|1970-01-01 15:45:01|106972776821582|         30.0|               6.8|          sale|     No |             204.0|
| 760409|2023-01-15|1970-01-01 19:28:58|102900011021842|         17.0|              10.8|          sale|     No |183.60000000000002|
| 325805|2021-06-04|1970-01-01 13:27:53|102900005116530|         10.0|              18.0|          sale|     No |             180.0|
| 441551|2021-10-15|1970-01-01 11:22:36|102900005118817|          8.0|              19.8|          sale|     No |             158.4|
|  97284|2020-09-29|1970-01-01 21:18:09|102900051010455|       16.003|               8.0|          sale|     No |           128.024|
```

12. Total Profit/Loss Day Wise:



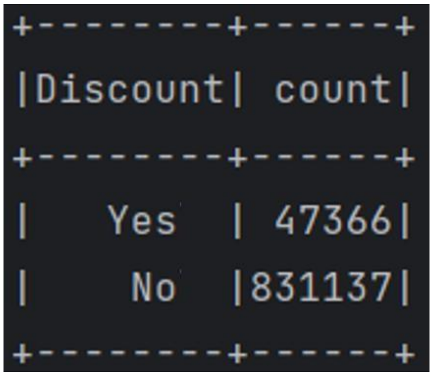
```
+---+--------------------+
|Day|         Profit_Loss|
+---+--------------------+
|  1|1.3476261517677146E8|
|  6|1.0562199108635464E8|
|  3| 9.633027145042503E7|
|  5| 9.226519754960284E7|
|  4|1.0395450948667243E8|
|  7|1.3417673614562015E8|
|  2| 9.333744630876052E7|
+---+--------------------+
```

13. Top 10 'Item Name':



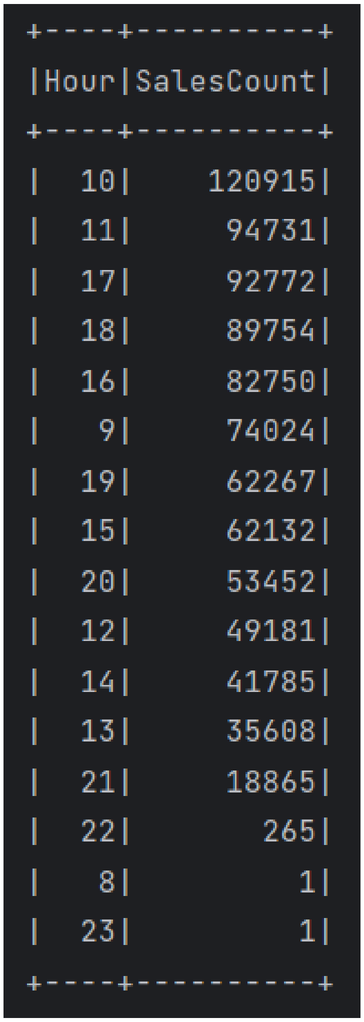
```
+--------------------+------------------+
|           Item_Name|Total_Quantity(kg)|
+--------------------+------------------+
|Wuhu Green Pepper...| 28164.33100000042|
|            Broccoli|27537.898999999165|
|  Net Lotus Root (1)| 27149.44000000034|
|     Chinese Cabbage|20894.521000000383|
|     Yunnan Shengcai|15910.461000000087|
|Needle Mushroom (...|           15596.0|
|Yunnan Lettuce (Bag)|           14325.0|
|        Eggplant (2)| 13602.00100000012|
|  Xixia Mushroom (1)|11920.227000000103|
| Millet Pepper (Bag)|           10833.0|
+--------------------+------------------+
```

14. Count 'Discount':



```
+--------+------+
|Discount| count|
+--------+------+
|   Yes  | 47366|
|    No  |831137|
+--------+------+
```

15. Sales Time Analysis (Peak Sales Hours):



```
+----+----------+
|Hour|SalesCount|
+----+----------+
|  10|    120915|
|  11|     94731|
|  17|     92772|
|  18|     89754|
|  16|     82750|
|   9|     74024|
|  19|     62267|
|  15|     62132|
|  20|     53452|
|  12|     49181|
|  14|     41785|
|  13|     35608|
|  21|     18865|
|  22|       265|
|   8|         1|
|  23|         1|
+----+----------+
```

Conclusion

The SuperStore Spark application provides a comprehensive set of analyses on the sales data from the MySQL database. It allows users to gain insights into various aspects of the business, including null values, category distribution, sales performance, discount effectiveness, duplication checks, yearly and monthly summaries, item details, total sales revenue, day-wise profit/loss, top-selling items, discount counts, and peak sales hours.

This application can be a valuable tool for decision-makers in understanding the sales trends, optimizing discount strategies, and identifying key items and time periods for maximizing revenue. Additionally, the ability to integrate with a MySQL database ensures that the analysis is performed on the most up-to-date data.

In conclusion, the SuperStore Spark application is a versatile and powerful tool for analyzing and gaining insights from sales data, contributing to informed decision-making within the business.