Unlocking the Potential of Big Data with Apache Spark

Carolina Almeida (20221855) | Duarte Carvalho (20221900) | Francisco Gomes (20221810) | Maria Henriques (20221952) | Marta Monteiro (20221954)

Introduction

In the era of Big Data, the ability to process and analyze vast amounts of information is a fundamental requirement for businesses across industries. The challenges associated with high-volume, high-velocity, and diverse datasets demand robust, scalable, and efficient analytical tools. Apache Spark, a unified analytics engine, addresses these challenges by enabling powerful and distributed data processing at

This report introduces the capabilities of Apache Spark, focusing on its potential to transform raw data into actionable insights. Leveraging PySpark within the Databricks platform, the project demonstrates Spark's versatility across domains such as machine learning, graph analytics, exploratory data analysis, and data engineering. By adhering to industry best practices, the work emphasizes Spark's role in delivering scalable and efficient solutions for complex data challenges.

The objective is to highlight the practical applications of Spark and its ability to support data-driven decision-making. This serves as a foundation for exploring how modern analytics tools can unlock new opportunities and drive innovation in a data-centric world.

Project Overview

This project showcases the application of scalable data analysis and machine learning techniques using Apache Spark. While the implementation does not involve an actual big data infrastructure, it adheres to principles and best practices designed for processing large datasets distributed across multiple partitions.

For this purpose, we utilized three independent datasets, each designed to highlight a distinct set of functionalities and analytical techniques:

The first dataset [1], focused on car rentals in the United States, provided detailed information about vehicles, customers, and transactions. In this part, the goal was to explore and analyze industry trends, identify patterns, and derive actionable insights into customer behavior and operational performance. This was achieved with DataFrames and Spark SQL for data manipulation, alongside GraphFrames for network analysis. By combining robust data processing with visual storytelling, we gained a deeper understanding of the car rental industry.

The second dataset [2], revolving around credit cards, was constructed by merging two sources: one containing applicant's personal information and another detailing their credit card behaviors. Using this combined dataset, we developed a predictive model aimed at determining loan eligibility based on financial profiles. This part of the project emphasized the use of pipelines for efficient data preparation, as well as the application of Spark's machine learning techniques to create a model capable of minimizing default risks and enabling informed decision-making for financial institutions.

The third dataset [3] contained transactional data from a UK-based e-commerce company, covering a wide range of transactions over a specific period. The functionalities for this dataset were divided into two parts: The first part involved extracting insights about customer behavior, sales patterns, and operational efficiency, using techniques such as RDDs and MapReduce. Time series analysis further revealed patterns like seasonality and trends, which were instrumental in forecasting future sales volumes. Additionally, Spark Streaming was employed to simulate real-time data processing, allowing for the monitoring of sales trends, detection of anomalies, and revenue calculations.

Note that we simulated Spark Streaming by continuously ingesting data from a directory of CSV files. This scenario represented a typical use case where data accumulates over time and requires near real-time processing. Using Spark's capabilities, we processed incoming data, applied transformations, and aggregated results into meaningful metrics through advanced techniques like windowing and watermarking.

The following sections provide a detailed exploration of each task, outlining the methodologies employed, the results obtained, and how Spark's capabilities were leveraged to unlock the potential of large-scale data processing.

Leveraging SparkSQL and GraphFrames [5] for Data Analysis – Unveiling Patterns in Car Rentals

Using SparkSQL allowed us to efficiently run SQL queries on structured data, enabling a seamless process for filtering, summarizing, and discovering trends within the car rental dataset. Through this approach, we were able to extract meaningful insights regarding various aspects of rental activity, car characteristics, and user behaviors.

One key observation was that **California** stands out as the state with the highest number of both reviews and renter trips. In stark contrast, **Montana** presented a unique case, with very few reviews despite a moderate number of trips taken. This disparity suggests that while more trips are being taken, they do not necessarily lead to an increase in customer feedback, possibly due to the nature of the trips or specific consumer behavior in the state.

When analyzing car types, it became clear that **cars** are by far the most popular choice for renters, followed by **SUVs**. In comparison, **minivans**, **trucks**, and **vans** are rented less frequently. This insight can guide car rental businesses in adjusting their inventory to meet demand more effectively.

Fuel types also showed interesting patterns. **Gasoline** vehicles dominate across most car categories, but **electric vehicles** are gaining traction, especially within the **Tesla** brand. **Tesla** stood out as the leading manufacturer, particularly in the car category, while **Jeep** and **Ford** led in the SUV and truck categories, respectively.

Interestingly, the correlation analysis revealed **no strong correlations**, however, a moderate positive correlation was observed between the number of trips taken and the count of reviews, indicating that a higher number of trips could be linked to a greater likelihood of reviews being written.

For future strategies, we could focus marketing efforts on **California**, encourage more **reviews** through incentives, expand the fleet of **electric vehicles** to align with sustainability goals, and investigate the **low activity** in states like **Montana** to address potential issues and improve performance.

We utilized GraphFrames to gain deeper insights into the relationships between car rental data points, focusing on the connections between vehicle owners, manufacturers, models, and their locations. By analyzing the graph structure, we performed several key queries to uncover patterns and understand the dynamics of the network.

In terms of **In-Degree analysis**, we found that **California** had the highest in-degree, with 952 connections, indicating a high level of activity and interactions in this region. Major cities like **Los Angeles**, **Las Vegas**, and **San Francisco** also exhibited high in-degrees, showing their central role within their states. Conversely, more suburban areas, such as **Irmo** or **Elmhurst**, had relatively low in-degrees, suggesting less connectivity in those regions.

For **Out-Degree analysis**, we identified the owner with the highest out-degree (owner ID 1300675), who operates a fleet of vehicles, including models from **Toyota**, **Dodge**, and **Ford**, primarily in **Minneapolis** and **Bloomington**. This high out-degree suggests that this owner has a significant impact on the rental activity in those cities.

The **PageRank algorithm** was also applied to assess the influence of nodes within the network. It revealed that certain vehicles were ranked highly, with **PageRank values** suggesting that these vehicles were particularly influential, possibly due to their location, brand, or renter popularity.

We also performed a **Triangle Count** to assess the clustering of nodes but found that all nodes were part of 0 triangles. This indicates that there are no tightly knit triadic relationships between the entities in the network, suggesting a lack of strong interconnections between vehicle owners, their cars, and locations in a triangular manner.

Moreover, the analysis of **Strongly Connected Components** helped identify clusters of nodes with significant interconnections, providing further insight into regional or brand-related patterns. The **Label Propagation algorithm** was used to group nodes into well-defined clusters, revealing shared characteristics and relationships. The use of the **spring layout** visually emphasized the clusters, allowing us to spot dense areas of interaction and potential outliers (*Annex Figure 1*).

Data Pipelines, Cleaning, Engineering and Spark Machine Learning – Predictive Modelling for Loan Eligibility

In the data preprocessing phase, we began by addressing various data quality issues. We changed the datatypes of columns as needed to ensure consistency across the dataset. We also removed rows with missing values, ensuring that the dataset was complete for analysis. During this step, we found no duplicate entries, ensuring the integrity of our data.

Next, we identified outliers in the numerical data and treated them by replacing extreme values with the median, a robust method to mitigate the impact of outliers on the model. For feature engineering, we transformed the target variable into a binary classification to suit our model's requirements. Upon examination, we discovered that the target variable was **imbalanced**, which led us to consider techniques to handle this issue (*Annex Figure 2*).

To standardize the numerical columns and bring them onto a common scale, we applied the **Standard Scaler** [6] through a **Pipeline**, which streamlined the transformation process. Additionally, we used the **String Indexer** [7] to encode categorical variables into numeric form, making them compatible with machine learning algorithms.

One of the crucial steps in preparing the data for machine learning was combining all features, both categorical and numerical, into a single feature vector. We utilized the **Vector Assembler** to first assemble the categorical and numerical features into separate vectors. Then, we merged these into a unified feature vector that could serve as input for our machine learning model. This ensured that all relevant features were provided to the model in a format that it could efficiently process and learn from.

For modelling, we used the **Random Forest Classifier [8]**. We trained the model using both the original training data and an oversampled version to address the class imbalance. We evaluated both models using a range of metrics, including **Area Under ROC (AUC - ROC), Precision, Recall**, and **F1-Score**.

For the model trained on the original, non-oversampled data, we observed a moderate AUC - ROC of 0.60, indicating that the model had some ability to discriminate between the positive and negative classes but still had room for improvement. The Precision was 0.62, meaning the model was reliable when predicting positive outcomes. However, the Recall was quite low at 0.10, indicating that the model missed a significant number of actual positive cases, a critical concern for imbalanced datasets. The F1-Score was similarly low at 0.18, reflecting the model's struggle to balance Precision and Recall effectively. (*Annex Figure 3*)

For the oversampled model, the results showed improvements, although still not ideal. The metrics indicated that oversampling helped the model learn to identify more instances of the minority class, leading to better performance than the model trained on non-oversampled data. This improvement is a positive sign, but further optimization is needed to enhance recall and overall model performance. (*Annex Figure 4*)

Utilizing RDDs [10] and MapReduce [11] for Efficient Data Processing and Analysis – Trends in E-Commerce

For this analysis, we conducted a series of exploratory data analysis actions to ensure the dataset was properly prepared for the actual analysis. During this process, we identified and removed some inconsistencies, including an unspecified country entry and unusual stock code names.

We also performed some **time series analysis [9]**, leveraging the availability of a date variable. However, we decided against forecasting, as the data covered only a single year, which is insufficient for reliable predictions. (*Annex Figure 5*)

To facilitate large-scale data processing, we converted the data into a **Resilient Distributed Dataset (RDD).** This transformation enabled us to leverage the **MapReduce** model for efficient data processing and extracting valuable insights.

Below are examples of practical applications using these techniques:

- **Revenue Analysis**: MapReduce can calculate average revenue per transaction across countries. Transactions are mapped to key-value pairs (country, revenue), shuffled to group by country, and reduced to compute averages. Resulting in countries with higher revenue, like United Kingdom, and Saudi Arabia with lower revenue.

- **Top-Selling Products**: Product sales data is mapped and shuffled to aggregate quantities by product. The reduce step then identifies the most sold items.
- **Customer Insights**: Transactions are transformed into (customer ID, revenue) pairs, enabling an analysis of customer contributions to total revenue. Reducing these pairs summarizes total spending per customer, supporting targeted campaigns for high-value customers who contribute significantly to sales.
- **Monthly Trends**: Sales data is grouped by month to identify peak revenue periods, facilitating strategic planning and forecasting. This analysis identified November as the highest revenue-generating month, confirming findings from the time series analysis, which identified a significant sales spike during that period.

Real-Time Data Processing with Spark Streaming

As mentioned earlier, we chose to simulate Spark streaming by collecting 100 CSV files, each containing one row, from the E-Commerce dataset. We developed a system that continuously ingests these files, utilizing Spark to read and process them as they arrive. To manage the flow of incoming data, we configured the system to process three files per trigger, controlling the number of files Spark processes in each micro-batch during the streaming session.

We set up a query to process the data every 10 seconds, writing the results in **append mode** and storing the processed data in **memory**. This setup enables fast, interactive queries. As part of our analysis, we performed SQL queries to display quantities and unit prices sorted by invoice date, total sales by stock code, the most recent transactions, and those from the United Kingdom.

Additionally, we experimented with an approach more suitable for long-term storage and larger-scale data handling by outputting the processed data in **Parquet** format, which offers efficient storage and processing capabilities for larger datasets.

Conclusion

This project successfully allowed us to apply the knowledge we gained in class while also integrating innovative ideas to tackle real-world challenges. By leveraging Spark's powerful tools such as DataFrames, SparkSQL, GraphFrames, MLlib, and Spark Streaming, we were able to transform raw data into actionable insights across multiple datasets.

Through the Car Rentals Dataset, we explored industry trends and customer-vehicle relationships; the Credit Card Dataset enabled us to apply predictive modelling for financial decision-making; and the E-Commerce Dataset showcased the potential of Spark Streaming and Map-Reduce for real-time monitoring and forecasting in retail.

While the project did not involve processing actual Big Data, we adhered to best practices for scalable and distributed systems, ensuring that our workflows are applicable to large-scale datasets. The integration of data engineering, machine learning, and graph analytics demonstrated Spark's potential to drive innovation and improve decision-making.

Furthermore, this project gave us the opportunity to learn more about Spark and work in PySpark, a language that was initially unfamiliar to us. We gained hands-on experience in distributed data processing and became proficient with Spark's various frameworks, enhancing our technical skills and confidence in working with large datasets.

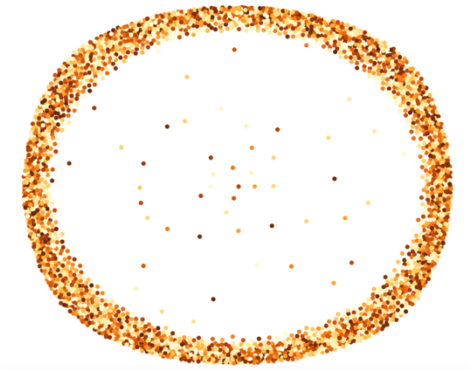
Ultimately, we believe our analysis is well-structured and can serve as a foundation for future strategies, offering valuable insights for businesses and organizations looking to harness the power of Big Data. We were able to meet the objectives of the project with minimal obstacles and are confident that the knowledge gained through this experience will be instrumental in future data-driven endeavors.

References

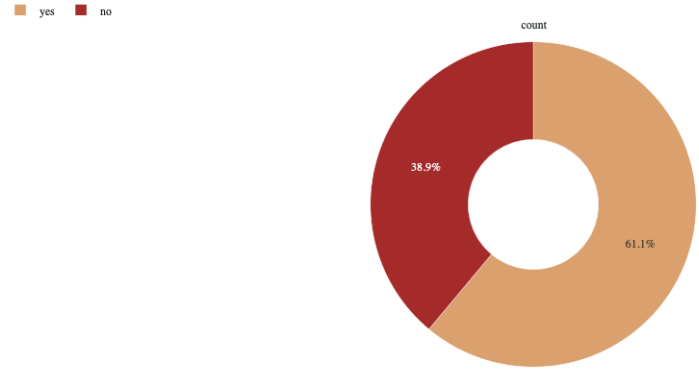
[1] https://www.kaggle.com/datasets/kushleshkumar/cornell-car-rental-dataset/data
[2] https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction
[3] https://www.kaggle.com/datasets/carrie1/ecommerce-data
[4] https://dezimaldata.medium.com/databricks-spark-temporary-views-with-sql-and-python-bd5a908f7092
[5] https://medium.com/data-hackers/introdução-ao-spark-graphx-e-graphframes-9b10089f2e7f
[6] https://scikit-learn.org/1.5/modules/generated/sklearn.preprocessing.StandardScaler.html
[7] https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/
[8] https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[9] https://www.tableau.com/analytics/what-is-time-series-analysis
[10] https://www.databricks.com/glossary/what-is-rdd
[11] https://www.databricks.com/glossary/mapreduce

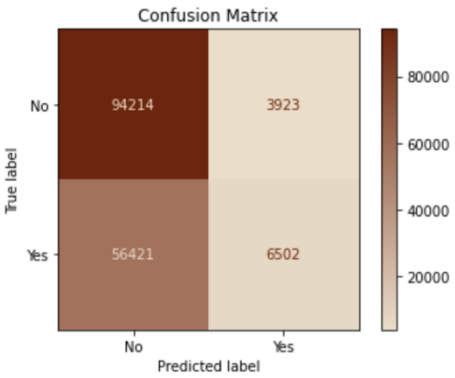
Annex

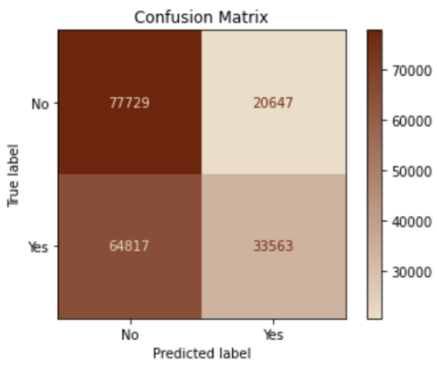


Annex Figure 1 – Graph from Label Propagation

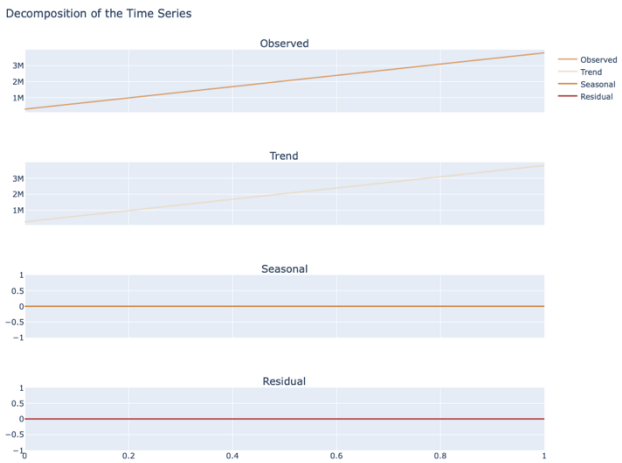


Annex Figure 2 – Target Imbalance





Annex Figures 3 and 4 – Confusion Matrix for Normal Training Data (on the left) and Confusion Matrix for Oversampled Training data (on the right



Annex Figure 5 – Decomposition of the Time Series