A close-up of a form

AI-generated content may be incorrect.

GitHub Repository: https://github.com/mohammadjavadi8804/B142-Data-Integration.git

Video Demonstration: https://youtu.be/y9IRvyxOrsA

Dataset link: https://www.kaggle.com/datasets/wspirat/germany-used-cars-dataset-2023

1-Introduction…………..…………..………….…………..………..……….…………..…………..………….

2-System Design…………..…………..……….…….…………..…………...………….…………..………….

3-Implementation…………..……………..…………..…………..………..…………..…………..…………..

4-challenges and solutions…………………..…………..…………..…..…………..…………..…………..

5-Results and insights…………..……………..…………..………………..…………..…………..…………..

6-Conclusion and future work………………..…………..……………..…………..…………..…………..

7-References…………..…………………………..…………..……………....…………..…………..…………..

1-Introductions:

What is my analysis about?

In this project, I designed and implemented a big data pipeline using Apache Spark to process and analyze a large scale dataset of used cars in Germany. Based on what we have discussed in the classroom there is no need to have a dataset that has over 1 million and over that. Based on it, my dataset from Kaggle contains 200,000 car listings with features such as brand, model, fuel type, gearbox, price, horsepower, and mileage.

Based on the simplicity of Databrick, I am trying to choose the second method for my project. Following this, Using Spark’s SQL distributed computing capabilities, I developed an ETL workflow. In addition, such a workflow ingests raw CSV files, cleans and transforms the data and at the end, we output the results in optimized Parquet format. Key transformations included handling missing values, converting data types, and creating new columns(for example, price per horsepower) to enrich the dataset.

Likewise, I chose the combined data from multiple tables from Kaggle. I hope this project showcases the practical use of big data tools for data engineering and analytics tasks and it would be beneficial for the automotive market.

2-System Design:

2.1.Overview of the Design:

First and foremost, let’s give an overview of the design. Based on what I said, my project processes a large CSV dataset (200k+ rows) of used cars in Germany. This data is treated as coming from one or more structured data sources, similar to a relational database. The system is designed to read, clean, transform, and analyze the data using Apache Spark. The key and most important parts are:

Input raw data (CSV format).

Spark DataFrames for processing.

Cleaned and transformed data saved as Parquet.

Analytics performed data saved as Parquet.

Analytics performed using Spark SQL.

A diagram of a spark

AI-generated content may be incorrect.

2.2.ERD(Entity relationship diagram) explanation:

As I working with one CSV file, I can simulate An ERD by logically splitting my dataset into two tables:

A diagram of a brand

AI-generated content may be incorrect.

1-CarInfo table:

Car\_id(pk), brand\_id(FK), model, year, price, mileage, fuel\_type, transmission, hp.

2-BrandDetails table:

Brand\_id(pk), brand\_name, country

3-Implementation:

In this project, I developed an ETL pipeline using Apache spark( second accepted method) to process and analyze a large scale dataset of used cars in Germany. Following this, the pipeline consists of several key stages including reading the raw data, cleaning and transforming it, performing analytics using Spark SQL, and finally saving the transformed data in Parquet format. Based on what my teacher expect us to do, below are the main steps of the implementation with brief descriptions and code references.

3.1.Reading the data:

The first step of my pipeline is reading the raw CSV file using Spark’s read function:

A screenshot of a computer

AI-generated content may be incorrect.

Explaination: This block reads the orginal dataset,set header to True, and helps us to infer schema automatically. It loads the data into a Spark DataFrame for transformation.

3.2. Data Cleaning and preprossing:

A screenshot of a computer

AI-generated content may be incorrect.

Explaination: Here I removed rows with missing values in key columns like price, brand, and year, and also renamed column names for better readability. Data types were standardized ( for example: converting price to integer)

3.3 Data Transformation:

A screenshot of a computer

AI-generated content may be incorrect.

Explaination:I created new calculated columns such as price\_per\_hp, and filtered out outliers( for example, cars priced below 500 euro and above 100,000). This made the data more useful for analysis.

3.4 Saving as Parquet:

A screen shot of a computer

AI-generated content may be incorrect.

Explaination: Cleaned and transformed data is saved in Parquet format, which is faster to read and ideal for large scale processing and future querying.

3.5. Analytics with Spark SQL:

A screenshot of a computer

AI-generated content may be incorrect.

Explaination:This query shows the average price per brand. I registered the DataFrame as a temporary table and used Spark SQL for aggregation and filtering.

3.6. Visualization preparation:

A graph of different colored bars

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

Explanation: I used basic visualization to show price trends by brand and transmission type.

4-challenges and solutions:

First and foremost, it was my first experience using Apache Spark in Databricks and all the thing was new and weird for me. Following this, while implementing the ELT pipeline using Apache Spark in Databricks, I encountered several challenges in the real world complexity of building a big data system. We are supposed to use the community edition of Databricks and it also has some limitations. I did not want to create a new cluster each time because they expired every 2 hours and it is funny I think. Likewise, one of the main challenges was setting up the development environment in Databricks. In addition, understanding how to navigate the platform, upload data, and configure the workspace correctly took time, because it was my first time working with Databricks. Another difficulty was related to understanding the dataset structure. For example, column names like price\_in\_euro and power\_ps were not initially intuitive and I faced errors in data transformations and column references. Another key challenge was during the data cleaning and transformation stage. I initially referred to non-existent columns such as price or hp, which led to unresolved column errors. To resolve this, I carefully reviewed the schema using printSchema() and adjusted the column names in the code accordingly. Creating the derived column price\_per\_hp also required handling division by zero and ensuring missing or invalid data was filtered out before the transformation.

In the end, I would say, saving data in Parquet format and executing Spark SQL queries came with a learning curve. Understanding the path system in DBFS and writing queries against temporary views was a new experience for me. However, with repeated testing and reviewing Spark documentation, I was able to overcome these issues and complete the pipeline successfully.

5-Results and insights:

So, after building and running the ELT pipeline in Apache Spark(I mean second acceptable method), I generated some analytics results from the cleaned Germany Used Cars dataset. Following this, using Spark SQL queries on the transformed data, I was able to extract key insights into pricing, brand, popularity, fuel types, and horsepower distribution.

I think, one of the main results showed that premium brands like BMW and Audi have significantly higher average prices than other brands and this confirms their luxury market position. The most common fuel types were petrol and diesel, but there was also a small presence of hybrid and electric vehicles, which highlights the market’s gradual shift toward sustainability. Because in Germany we have many restrictions in terms of air pollution and especially in car emissions. In addition, manual transmissions were more common than automatic ones, which reflects driving culture in Germany and I think it is somehow related to the price of the cars as well.

I also calculated a new metric which is: price per horsepower(price\_per\_hp). Following this, it revealed that some lower cost brands offer better value in terms of performance per euro.

These findings and insights can be useful for both buyers and market analysis. Likewise, all these results were generated within Spark and displayed using .show() for fast and scalable memory processing.

The last thing would be, that my cleaned output can easily be exported as a CSV or Parquet file and further visualized in tools like Jupyter Notebook, Power BI, or Google Colab for deeper exploration and dashboard creation.

6-Conclusion and future work:

In conclusion, my project demonstrated how Apache Spark can be used to efficiently process and analyze a large scale dataset which I chose from Kaggle. Following this, in this case, over 200,000 records of used cars in Germany. I successfully built an ETL pipeline that handled data ingestion, cleaning, transformation, and analysis. The results revealed useful insights into pricing trends, brand value, and performance metrics within the German automotive market.

The lessons in the class also helped me develop practical skills in data engineering using Databricks and Spark SQL.

If I had more time, I mean six months instead of a few weeks, I would like to extend the project by integrating real time data sources such as live car listings via APLs. I also would like to connect the pipeline to dashboard tools like Power BI or Tableau to create interactive reports.

In addition, I would like to explore machine learning techniques to build a price prediction model, helping users estimate a fair price based on a car’s brand, year, mileage, and horsepower. I think making these additions would significantly increase the project’s value in a real world business context or industry.

7-References:

1. Apache Spark (n.d.) Apache Spark – Unified Analytics Engine for big Data. Available at: <https://spark.apache.org> ( Accessed: 18 june 2025).
2. Databricks (n.d.) Databricks: The Data and AI compaby. Available at: <https://databricks.com/> (Accessed: 18 June 2025).
3. Kaggle (n.d.) Used Cars Dataset: <https://www.kaggle.com/datasets/wspirat/germany-used-cars-dataset-2023>
4. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S. and Stoica, I. (2010) ‘Spark: Cluster computing with working sets’, Proceedings of the 2nd USENIX conference on hot topics in cloud computing ( HotCloud’10), Boston, MA, USA.
5. Parquet (n.d.) Apache Parquet Documentation. Available at: <https://parquet.apache.org>( Accessed: 18 June 2025).
6. Microsoft (n.d.) Power BI. Available at: <https://powerbi.microsoft.com/(Accessed>: 18 june 2025)
7. Zaharia,M.,Das, T.,lI,h.,Hunter, T.,Shenker, S. and Stoica,I.,2016.Apache Spark: A unified engine for big data processing.Communications of the ACM,59(11).pp.56-65.
8. Apache Software Foundation,2024.Apache Spark documentation.Available at: <https://spark.apache.org/docs/latest> Accessed 18 Jun 2025.