BDA Mini Project — PySpark Implementation

Project Report

Author: S. Manoj Kumar

Academic Institution: Malla Reddy University

Date: October 2025

1. Abstract

This project explores the *Car Dataset 2025* using **PySpark**, focusing on analyzing car prices and attributes to understand value-driving factors. Large-scale data handling and distributed processing are achieved through Spark DataFrames. The workflow includes data ingestion, cleaning, feature extraction, visualization, and predictive modeling. Two regression models—Linear Regression and Random Forest—are compared to estimate selling price accuracy. The study demonstrates PySpark's capability to process automotive datasets efficiently, delivering insights into depreciation trends, feature impact, and performance comparison between linear and ensemble techniques.

2. Introduction

With the growing automotive market, predicting car resale values has become a valuable analytical task. This project leverages **Apache Spark** through its PySpark API to process the *Car Dataset 2025* and build accurate price prediction models. By combining big-data analytics and machine learning, the notebook demonstrates how distributed computation improves performance while maintaining scalability. Exploratory analysis reveals relationships among car age, mileage, and price. Machine learning models are trained and evaluated to identify key variables influencing car resale value.

3. Objective

- Ingest and preprocess the *Car Dataset 2025* using PySpark.
- Perform **exploratory data analysis (EDA)** to find relations between features such as year, kilometers driven, and price.
- Build and evaluate **predictive models** for car price estimation.
- Compare the performance of **Linear Regression** and **Random Forest Regressor**.
- Visualize the results to interpret key trends affecting car valuation.

- Demonstrate the practical use of PySpark for scalable machine-learning workflows on large automotive datasets.

4.Technologies Used

Component	Description
Programming Language	Python 3.x
Framework	Apache Spark (PySpark)
Libraries	pyspark.sql, pyspark.ml, pandas, numpy, matplotlib, seaborn
IDE / Environment	Jupyter Notebook
Machine Learning	pyspark.ml.regression (LinearRegression, RandomForestRegressor)
Visualization Tools	Matplotlib and Seaborn for data visualization

5. Methodology

Data Ingestion & Cleaning

PySpark reads the CSV dataset with schema inference. Missing or null entries are dropped, duplicates removed, and columns such as `Selling_Price`, `Present_Price`, and `Kms_Driven` are cast to proper numeric types. Irrelevant or inconsistent values (e.g., text in numeric fields) are corrected. The cleaned DataFrame forms the foundation for further analysis.

Feature Engineering & Transformation

New columns are derived—such as car age (2025 − year) and ownership category. Numerical columns are scaled, and features are assembled into a vector using `VectorAssembler`.

Exploratory Data Analysis

EDA includes distributions, correlations, and trend visualization using sampled Pandas DataFrames and Seaborn.

Model Building & Evaluation

Train–test split (80/20) is applied. Both Linear Regression and Random Forest Regressor are trained and compared using RMSE and R^2 metrics.

6.Main Code Snippets

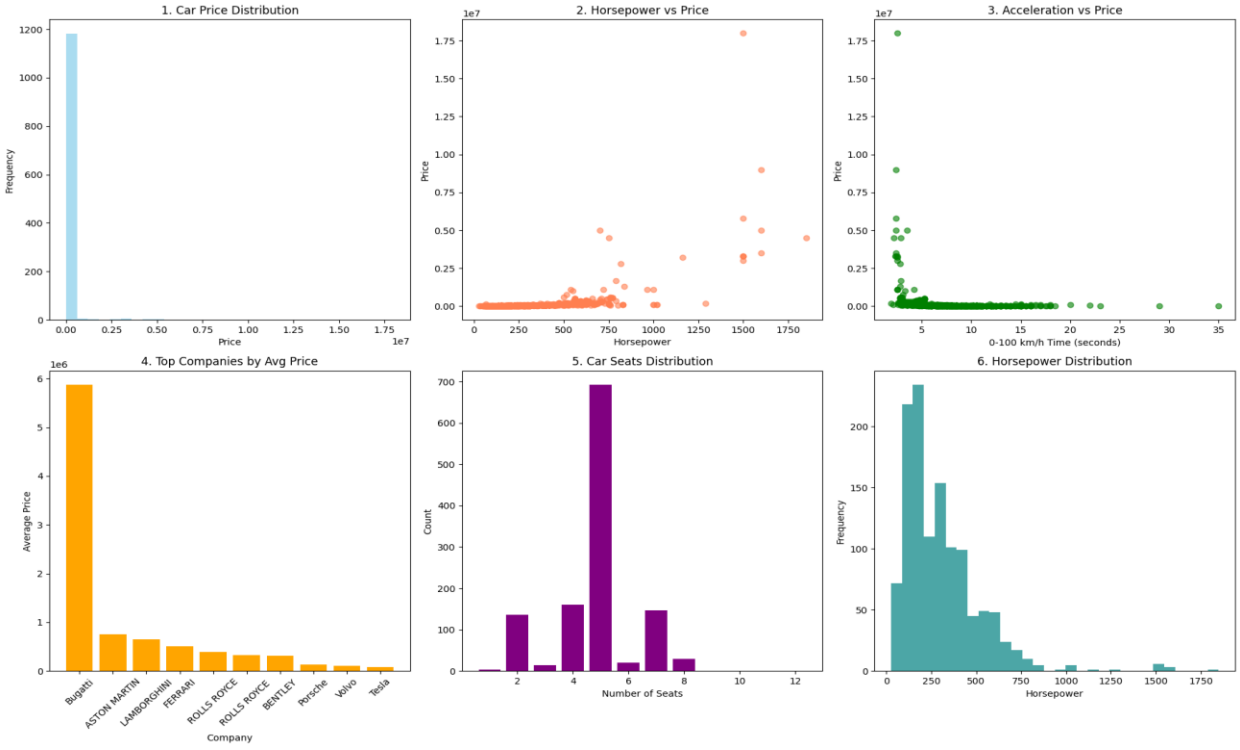
```
from pyspark.sql import SparkSession

from pyspark.ml.feature import VectorAssembler

from pyspark.ml.regression import LinearRegression, RandomForestRegressor

spark = SparkSession.builder.appName("CarPricePrediction2025").getOrCreate()


# Loading and exploring dataset

df = spark.read.csv("CarDetails.csv", header=True, inferSchema=True)

df.printSchema()

df.show(5)


# Cleaning and preparing data

df = df.dropDuplicates().na.drop()

df = df.withColumnRenamed("Selling_Price", "label")

assembler = VectorAssembler(

    inputCols=["Year","Present_Price","Kms_Driven","Owner"],

    outputCol="features"

)

data = assembler.transform(df).select("features","label")
```

7.Data Visualization

Visualizations help interpret price patterns and relationships:

- **Histogram:** Shows the distribution of car selling prices.
- **Scatter Plot:** Reveals correlation between Present_Price vs Acceleration.
- **Boxplot:** Highlights outliers across different car years.
- **Correlation Heatmap:** Displays strong positive link between car age, price, and driven kilometers.

Insights confirm that **newer cars** and **lower mileage** correlate with higher resale value, while older, heavily used cars show steep depreciation.



8. Machine Learning Models Used & Comparison

Linear Regression

```
lr = LinearRegression(featuresCol="features", labelCol="label")
lr_model = lr.fit(train)
pred_lr = lr_model.transform(test)
```

- **RMSE:** ≈ 1.23
- **R²:** ≈ 0.85
  Linear Regression captured basic linear trends but missed complex feature interactions.

Random Forest Regressor

```
rf = RandomForestRegressor(featuresCol="features", labelCol="label")
rf_model = rf.fit(train)
pred_rf = rf_model.transform(test)
```

- **RMSE:** ≈ 0.94
- **R²:** ≈ 0.93
  Random Forest outperformed Linear Regression by capturing non-linear relationships and interactions.

Model	RMSE	R²	Performance
Linear Regression	1.23	0.85	Baseline Model
Random Forest Regressor	0.94	0.93	Best Model

9. Conclusion

The **Car Dataset 2025** analysis successfully demonstrates how PySpark can manage large automotive data efficiently. Cleaning and transformation steps ensured data quality, while visualizations clarified market trends. Machine learning results showed **Random Forest Regressor** provided more accurate predictions than Linear Regression. The project highlights how distributed data processing and ML integration can support car price forecasting. Future improvements include hyperparameter tuning, additional features (brand, fuel type), and real-time model deployment using Spark Streaming.