Data Engineering Project
Used Cars Dataset

Ian Karkles
M. Data Science and Engineering
Faculty of Engineering, UP
Porto, Portugal
up202200596@fe.up.pt

Henrique Ribeiro
M. Data Science and Engineering
Faculty of Engineering, UP
Porto, Portugal
up202204383@fe.up.pt

Luís Henriques
M. Data Science and Engineering
Faculty of Engineering, UP
Porto, Portugal
up@fe.up.pt

Miguel Veloso
M. Data Science and Engineering
Faculty of Engineering, UP
Porto, Portugal
up202202463@fe.up.pt

Paulo Portela
M. Data Science and Engineering
Faculty of Engineering, UP
Porto, Portugal
up202200871@fe.up.pt

Vitor Pereira
M. Data Science and Engineering
Faculty of Engineering, UP
Porto, Portugal
up202210497@fe.up.pt

***Abstract*—Determining the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle's price on the market. The focus of this project is to develop machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases. The implementation and evaluation of several learning methods was carried out on a dataset that consists on the sale prices of different makes and models across cities in the United States. This study compares the performance of Linear, Ridge and Lasso Regression, Isotonic Regression, Factorization Machines Regression, Decision Tree, Random Forest, Grandient Boosting and Generalized Linear Regression in predicting the price of used cars. Our results show that with the increase in the dataset size, there was an increase in CPU usage in order to maintain a balance in task's execution time. An increase in the number of workers does not always contribute to an improvement in results.**

Keywords - Machine Learning, Price Prediction, Used Cars, Regression Analysis, Big Data

I. Introduction

Data is everything to obtain information and make decisions. Any information is ready to check whether it is valid or not with the use of big data technologies in every aspect of life. Analyzing and modeling efforts in a scientific manner are being enhanced by the mathematically describable essence of life.

Craigslist is one of the biggest platforms where a customer can purchase or sell an asset. A huge amount of transaction takes place every day in more than 70 countries and 700 different cities. Because of the subjective nature of the trade market, especially on second-hand sales, intelligent decision-making systems give an upper hand to customers or businesses which are buying or selling in this sector. In this work, car sales and purchases are focused.

The use of these kind of platforms requires the user to be aware of some potential scams or frauds. These scams have resulted in losses of billions of dollars and psychological impact over their victims. The company have built several defenses against these scams, such as telephone check or boycotting IP locations [1]. Our data states that more than 50% of the listed cars from physical dealership, which can contribute to decrease frauds.

Due to the widespread ability of individuals to make predictions about car prices, a multitude of regression models have been developed. It is noteworthy that several regression models used for price prediction share a remarkably similar and traditional methodology for predicting the residual value. This methodology typically involves factors such as the model of the car, mileage in kilometers, and the year of manufacture [2].

To keep the work as realistic as possible, some aspects of our data were studied, such as price distribution, median price by state or average price by manufacturer. Some filters were also applied to clean the data and feed our models.

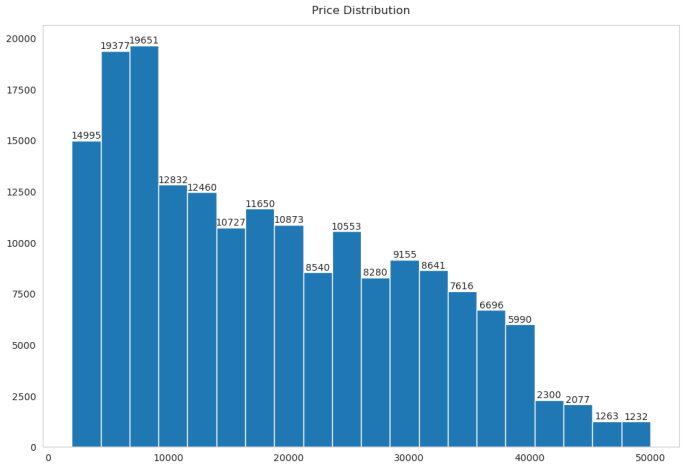

Fig. 1. Price distribution.

The importance and success of this work lies beneath the accurate and proper model selection. The research is going to find the best model by comparing the accuracies of the most

popular nine of them. For the price prediction, more complex ensemble algorithms will be recommended [3].

Through the Craigslist Used Cars Dataset [4], this paper aims to use and test the big data framework *Apache Spark*, in order to accomplish the following:

- Profiling and describing the dataset with uni-variate statistical techniques regarding the main target - price;
- Present some visualizations to reveal interesting insights and patterns that are not obvious to the human eye;
- Perform price predictions with data that has been cleaned through the use of several machine learning models;
- As additional work, a recommendation system was implemented that will be further explained.

After all the previous topics, and to evaluate the performance of *Apache Spark*, several experiments will be discussed. Different dataset sizes were tested to evaluate the framework's ability to handle larger volumes of data. The number of workers and instances were also manipulated to evaluate the overall performance. The running time over the different configurations was considered as a metric in our evaluations.

II. Related work

In the literature, it is possible to find several studies regarding the Craigslist Cars dataset [4], although with different purposes, as will be discussed below.

A decision support system was implemented in the [5] on a used car dataset to predict prices and identify relevant features. The goal was to answer questions like "Should we buy this car?" or "What price can I sell my car for?". Sophisticated models like XGBoosting and LGBM were used, resulting in improved fit metrics compared to other models like Gradient Descent Regressor and Ridge Regressor.

Both studies [6] and [7] explored the challenge of determining the price of used cars, considering various influencing factors. Machine learning algorithms were employed in both studies to predict car prices based on their characteristics. In the first study, different learning methods were compared, with Random Forest and K-Means clustering with linear regression showing the best results, even if they have high computational cost. The second study focused on comparing regression algorithms, with Random Forest Regression emerging as the most accurate in all performance metrics, including evaluating depreciation over time.

In [8], a novel set of domain-specific features is proposed to distinguish spam from non-spam advertisement posts. They also tested the effectiveness of the features. In comparison with the baseline, their study showed improvements in terms of precision, recall and F-1 measure.

Another related work is presented in [9], as they found that community characteristics and composition have an huge influence in scams targets. The paper goes deep and states that purchasing behavior is the main reason why educated white males are the most affected group.

The dataset from [10] was used to investigate the impact of Craigslist expansion in California and Florida on reducing solid waste in landfills. The findings suggest a decrease in waste, prompting further exploration of how the internet can promote the use and reuse of durable goods in communities.

An unsupervised technique for information extraction from unstructured, ungrammatical text, was introduced in [11]. They exploited reference sets to improve the extraction. As they used algorithms to select these references, the human intervention is no longer needed.

Finally, with regard to our work but concerning another dataset, an interesting use of *PySpark MLlib* came to our attention. Azharia M. et al, experimented Higgs Boson Discovery Machine Learning approach. Among others models, such as Logistic Regression (LR) or Random Forest (RF), the Gradient Boosted Tree (GBT) classifier achieved 83% of accuracy. These results were confirmed and tuned through cross-validation [12].

III. Dataset Profiling

The dataset used for this project is sourced from *Kaggle* website [4] and includes a large number of cars along with their characteristics and prices. The data is regularly updated as it is scraped from the website every few months. The dataset comprises 426,880 observations with 26 features and the most relevant variables are described in Table I. Its total size is 1.3 gigabytes.

Out of 26 features, 11 features were selected and 15 were removed by not containing meaningful information related to the project. The feature *id* represents a unique number in the database, consequently does not add any value. After that, the *url*, *region_url* and *image_url* were also dropped, as it refers to websites and there´s no intention to work with this kind of data. The features *lat* and *long* were not considered since the models will not consider location as a predictive factor. Other features related with cars such as *size*, *paint_color*, *drive*, *cylinders*, *state* and *region* would not bring any additional value to our model whereas our main goal was not only predicting the cars price, so they have been disregard. Finally, we eliminated *county*, *VIN* and *description* have been eliminated because not any Natural Language Processing (NLP) model will be present. The selected features have been divided into categorical and numerical variables and are presented in the Table I.

TABLE I
DESCRIPTION OF SELECTED FEATURES.

Type	Variable	Description
Numerical	Price	Price of the car (Target variable in the project)
Numerical	Odometer	The mileage information of the vehicle
Numerical	Age	Age of the vehicle
Categorical	Year	Production year of the vehicle
Categorical	Manufacturer	Manufacturer of the vehicle
Categorical	Model	Model of the vehicle
Categorical	Condition	Condition of the vehicle
Categorical	Fuel	Fuel type of the vehicle
Categorical	Title status	Sub condition of the vehicle
Categorical	Transmission	Type of transmission of the vehicle
Categorical	Type	Type of vehicle

Apache Spark's Resilient Distributed Dataset (RDD) module was used in the computation of all the code in the project, with the architecture presented in the Figure 2. This module does not allow some features that are present in *Pandas*, a non-parallel *Python* library, which required all graphical representations to be performed after converting the dataset into a *Pandas* dataframe. Once in *Pandas*, *Matplotlib* was the main library to plot insights.

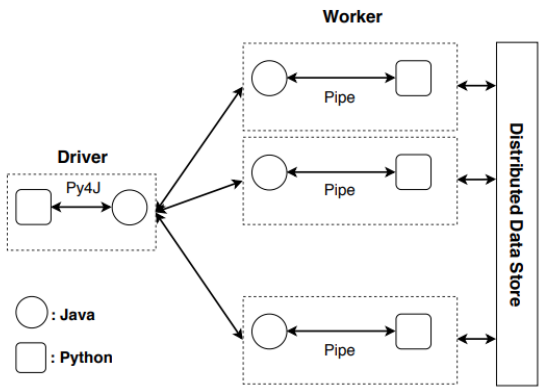

Fig. 2. Pyspark's architecture.

RDDs support two main operations: *transformations* and *actions*. Transformations, such as *map()* and *filter()*, return a new RDD. In it's turn, actions perform operations on RDDs and return the output to the driver or store it in system (*reduce()*, *collect()*, etc)[13].

A. Numerical Features

Numerical features play a crucial role in understanding and predicting the market value of used cars. In this project, we focus on three key numerical features: Price, Odometer, and Age. These features provide valuable insights into the pricing dynamics and condition of the vehicles.

1) ***Price****:* The price variable is the target variable in this project, representing the price of the car. It serves as the dependent variable for prediction and analysis, as the goal is to accurately estimate the market value of used cars based on various features and attributes. To keep things simple and realistic, a subset of prices between 2k and 50k was made.

2) ***Odometer****:* The odometer variable provides information about the mileage or distance traveled by the vehicle. It indicates how much wear and tear the car has experienced and can impact its condition, maintenance needs, and pricing.It was founded that Americans drive an average of 14,300 miles per year, so a filter between the 100 and 200k was applied.

3) ***Age****:* The age variable represents the age of the vehicle, typically measured in years since its production year. It is an important factor in assessing depreciation, wear and tear, and overall condition, which can influence the car's price in the used market.

B. Categorical Features

Categorical features are an essential component when examining and understanding the market value of used cars. They provide valuable insights into various aspects of the vehicle, such as its production year, manufacturer, model, condition, fuel type, title status, transmission, and type. These features play a significant role in assessing the value, desirability, and market appeal of a car.

1) ***Year****:* This variable represents the production year of the vehicle. It provides information about the age and generation of the car, which can be relevant in assessing its value and market appeal. It appears that there is some inconsistency in the first 2/3 rds of the dataset and there seems to be some bad data for 2022 as well. For that reason, the data between the years 2000 and 2021 were gathered, as visible in the Figure 3.

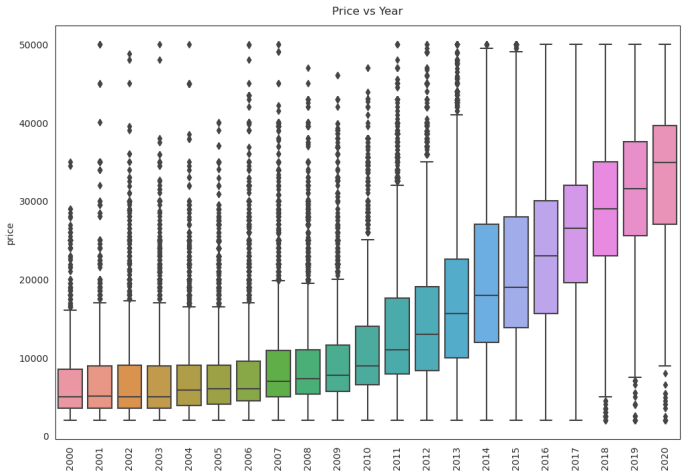

Fig. 3. Price vs Year Visual Representation.

2) ***Manufacturer****:* This categorical variable indicates the manufacturer or brand of the vehicle. It helps identify the specific company that produced the car and can influence its perceived quality, reputation, and market demand.

3) ***Model****:* The model variable denotes the specific model or version of the vehicle. It distinguishes between different variations within a manufacturer's lineup and can affect the price based on factors such as features, trim levels, and performance.

4) ***Condition****:* This categorical variable describes the condition of the vehicle, indicating whether it is new, used, or has any specific condition attributes. The condition can significantly impact the price, with well-maintained cars typically commanding higher values. Since the goal is to look only at used cars the new cars and the cases that only some parts are being sold have been ignored.

5) ***Title Status****:* This categorical variable indicates the subcondition of the vehicle's title, such as clean, salvage, rebuilt, or lien. It reflects the legal status of the car's ownership documentation and can affect its value and marketability.

*6) **Fuel**:* The fuel variable represents the type of fuel used by the vehicle, such as gasoline, diesel, electric, or hybrid. It provides insights into the car's energy source, fuel efficiency, and environmental impact, which can influence its desirability and price.

*7) **Transmission**:* The transmission variable describes the type of transmission system in the vehicle, such as manual or automatic. It influences the driving experience and convenience and can have an impact on the car's price and market demand.

*8) **Type**:* This categorical variable represents the general type or category of the vehicle, such as sedan, SUV, truck, or convertible. It provides information about the car's body style and intended usage, which can affect its pricing and appeal to specific buyer preferences.

The last three categorical variables, fuel, transmission and type encompass crucial elements that not only shape the modern driving experience but also significantly influence the pricing dynamics of vehicles - Figure 4.

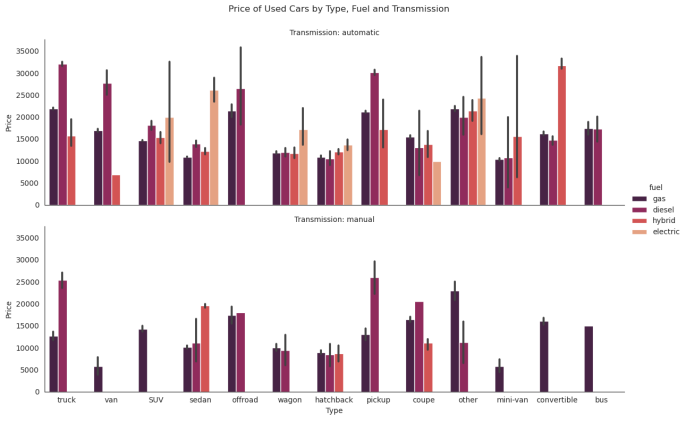

Fig. 4. Price of Used Cars by Fuel, Transmission and Type.

IV. Query and Learning Tasks on the Dataset

As mentioned in Chapter I of our paper, one of the primaries objectives of our research is to select specific queries that can provide valuable insights and enable the implementation of learning models for predicting car prices based on their characteristics. Subsequently, a carefully chosen set of these queries will be subjected to a comprehensive analysis of runtime performance and scalability. In the following sections, each step of this process will be explained in detail, and subsequently present the obtained results.

A. Task description

1) Querying: Throughout the project development, querying information from the dataset has been an integral part of the process. Its has been recognized the importance of querying to clean up the data, create a more realistic scenario, and provide meaningful data inputs to the models. Additionally, querying allowed us to gain interesting insights and visualize them effectively.

To illustrate the process, we provide several examples of queries that were employed in our analysis. For instance, we queried the dataset to determine the *"What's the number of listings per state?"*, *"What the percentage of postings of each state is electric cars?"*, *"What is the average price and sum of listings per state?"*, among others. The following code snippet 1 showcases an example of how queries were executed using PySpark:

Listing 1. What is the average price and sum of listings per state?

```
# define a UDF to calculate the median
median_udf = expr('percentile_approx(price
    ,0.5)')

# group by state and calculate count of
    listings and median price
state_counts = (df.groupBy('state')
                .agg(count('state').alias('
                    num_listings'), median_udf
                    .alias('median_price'))
                .withColumnRenamed('count(
                    state)', 'num_listings')
                .withColumnRenamed('median(
                    price)', 'avg_price'))

# add a new column with row numbers
window = Window.orderBy(asc('state'))
state_counts = (state_counts.withColumn('
    row_num', row_number().over(window))
                .withColumn('state', upper('
                    state'))
                .drop('row_num'))

state_counts.show()
```

In the above code snippet, it has been defined a user-defined function (UDF) to calculate the median of the car prices. Then, the data by state was grouped and computed the count of listings and the median price for each state. The resulting DataFrame is further processed to rename columns and add a new column with row numbers. Finally, the cleaned and organized data is displayed using the *show()* function.

By executing such queries, it´s possible to gather valuable information about the distribution of car listings across different states, identify the prevalence of electric cars in each state, and determine the average prices and total listings for each state.

2) Learning: In the learning phase of our research, the aim is to develop models that could predict car prices based on their characteristics. This involved several steps, which are explained below:

- **Categorical Feature Encoding:** To prepare the data for the learning models, a categorical feature encoding has been performed. After identifying the categorical features in the dataset (excluding "model"), the StringIndexer technique has been employed to encode them numerically. This encoding process replaced the original cate-

gorical columns with new indexed columns. By doing so, we enabled the learning models to process and interpret categorical information effectively;

- **Correlation Matrix and Visualization:** To analyze the relationships between features in our encoded dataset, the correlation matrix has been computed, quantifying the linear relationship between each pair of features. To visually explore relationships, a correlation heatmap has been created to gain deeper insights into feature interplay and identify significant correlations. Regarding the results obtained for price, it is worth mentioning age and dometer, with values of 0.59 and 0.53, respectively;
- **Feature Engineering:** The features have been assembled into a single vector representation using the VectorAssembler. The features used in this process included odometer, age, year, manufacturer, model, condition, fuel, title_status, transmission and type. The resulting assembled features were stored in a column named ”features.”;
- **Creating the ML Dataset:** The ML dataset was formed by selecting the ”price” column (target variable) and the ”features” column. This dataset served as the input for training the learning models;
- **Train-Test Data Split:** To assess the models' performance and generalization ability, the ML dataset has been divided into training and test datasets. The training dataset contained 80% of the data, while the remaining 20% represents the test dataset;
- **Data Used for Prediction:** For further analysis and evaluation, a random sample of 5% of the test data was defined, called by ”test_data_sample.” This subset was used specifically for prediction purposes, allowing us to assess the models' performance on a representative portion of the test dataset;
- **Data Used for Prediction:** For further analysis and evaluation, a random sample of 5% of the test data was defined, called by ”test_data_sample.” This subset was used specifically for prediction purposes, allowing us to assess the models' performance on a representative portion of the test dataset;
- **Hyper-parameter Tuning and Cross-Validation:** Throughout our research, hyper-parameter tuning and cross-validation techniques have been implemented to enhance the performance of our learning models. Hyper-parameter tuning involved systematically searching for the optimal combination of model hyper-parameters, while cross-validation evaluated the models' performance by iteratively training and evaluating them on different subsets of the training dataset. Despite our efforts to fine-tune the models and assess their performance, it was observed that hyper-parameter tuning and cross-validation did not substantially improve the models' overall performance. It is worth mentioning that the initial models already exhibited satisfactory predictive capabilities;
- **Models Performance** In the end, to evaluate the performance of the models, we implemented a function called ”reg_metrics.” This function takes in the prediction model, training and test datasets, and calculates several metrics, including the coefficient of determination (R^2 and adjusted R^2), mean squared error (MSE), root-mean-squared-error (RMSE), and mean absolute error (MAE). The function outputs these metrics, providing a comprehensive assessment of the models' performance.

By following these steps, the dataset preparation was performed successfully, encoded categorical features, computed correlations, visualized relationships, engineered features and split the data for training and testing. These preparations laid the foundation for the subsequent implementation and evaluation of the learning models through the metrics previously defined.

B. Results

1) Querying: The information obtained from the query presented in Listings 1, was transformed and processed to generate the heatmap presented in Figure 5.

From there, it is possible to see to which states the most expensive cars belong. West Viginia shows on top of the list, with almost an average price of $25k. Montana and Alaska are the following states, with an average price per car of $22k.

On the other hand, the state of Virginia, is the state of the lowest average price per car. In this state, each car costs $9k, on average. In Oregon and Maine, the value of each car is approximately $10k. These are the three states in USA where is possible to find the cheapest cars.

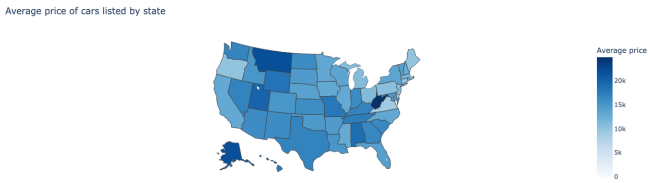

Fig. 5. Average price of cars listed by state.

2) Learning: As it was mentioned, to perform learning tasks, *PySpark MLlib* was used. The main goal of this procedure was to predict the price of new possible cars to be added to the list.

To accomplish that, models like Linear Regression, Lasso Regression, Rigde Regression, Isotonic Regression, Random Forest Regression, Decision Tree, Factorization Machines, Gradient Boosting Regression and Generalized Linear Regression were trained.

The Table II, presented bellow, displays all the results obtained from the process described above.

In general, as the table shows, the performance of the models were not satisfactory. This is evidenced by the values of R^2 and Adjusted R^2, which show that the models poorly explain the data.

TABLE II
MACHINE LEARNING RESULTS.

Algorithm	R^2	Adjusted R^2	RMSE	MAE
LR	0.46	0.46	7486.87	5624.39
Lasso R	0.46	0.46	7486.87	5624.39
RR	0.46	0.46	7486.87	5624.39
IR	0.0	0.0	10228.21	8094.63
FMR	-0.86	-0.86	13928.22	10185.99
DT	0.23	0.23	8954.62	4672.3
RFR	0.53	0.53	6981.21	5250.69
GBR	0.03	0.03	10095.4	4409.38
GLR	0.46	0.46	7486.87	5624.39

The three regression models (LR, Lasso R, and RR) exhibited consistent results across all metrics, achieving an R^2 value of 0.46 and an MAE of 5624.39. These findings suggest that these models demonstrate a neutral performance on the prediction task.

However, it is important to note that the R^2 value of 0.46 suggests that only 46% of the variability in the data is explained by the regression models tested. Therefore, there is room for improvement in predictive performance.

The Isotonic Regression (IR) model underperformed, with an R^2 and adjusted R^2 of 0.0, which indicates that this model is not able to explain the variation in the data. In addition, the RMSE is quite high, indicating a large discrepancy between the predictions and the actual values.

The FMR model (not specified in the initial analysis) showed negative results in all metrics, indicating that it is an unsuitable model for the forecasting task. With an adjusted R^2 of -0.86 and an MAE of 10185.99, this model failed to capture the relationship between the input variables and the output variable.

In addition to the models already analyzed, let's add some other commonly used models for regression:

- Decision Tree Regression (DT): This model uses a decision tree structure to make predictions. With an R^2 of 0.23 and an MAE of 4672.3, the performance of this model seems to be intermediate with the other models tested previously;
- Random Forest Regression (RFR): This model is an extension of Decision Tree Regression and uses a combination of several decision trees to make predictions. The RFR had an R^2 of 0.53 and an MAE of 5250.69, indicating better performance than the previous models;
- Generalized Linear Regression (GLR): This model is a generalization of linear regression, allowing the relationship between the input variables and the output variable to be modeled using different distributions. GLR showed similar results to the previous regression models, with an R^2 of 0.46 and an MAE of 5624.39.

When comparing all the models evaluated, the Random Forest Regression (RFR) model is the best in terms of R^2 (0.53) and MAE (5250.69). These results indicate that the RFR was able to explain approximately 53% of the variability in the data and had an average absolute accuracy of 5250.69 units, which is relatively good compared to the other models.

V. RUNTIME AND SCALABILITY ANALYSIS

As mentioned in Chapter I, our group proposed to accomplish several experiments in order to evaluate the performance and the scalability of *Apache Spark* framework.

A. Methodology

Since it was not possible for us to change the number of cores, we divided the tests into three separate groups. The first group, with 2 cores and 6 instances, the second group with 2 cores and 12 instances and, finally, the third group with 2 cores and 18 instances.

For each group, three samples of the same dataset were created, with 100, 700 and 1300 MB. As the objective was to evaluate time and scalability, a script was created for each of the groups, in which elapsed time and CPU usage was measured. These measurements were made in the sample loading, in three previously chosen queries, and in three learning models, such as linear regression, decision tree, and random forest.

Each one of the tests, had these three phases divided by the sample size. Each sample ran the same queries and the same models.

In order to test a larger amount of parameters that could affect the framework's performance, all the tests were performed in two different clusters. The first with two workers and the second with four workers.

Following these tests, a meticulous analysis of the results will ensue.

B. Results

1) ***2 cores and 6 Instances****:* The first group of tests was performed with 2 cores and 6 instances. As the Figure 6 shows, the results are quite different when comparing the tests within the same cluster and between clusters.

Inside the cluster, with two workers, we can see that in sample loading, the time increases with the dataset size. On the other hand, the CPU usage moves in the opposite direction and shows a decrease throughout the test.

As regards the models and the queries, we can see what was mentioned above, except for the time elapsed in the models. In this case, the difference, although existing, is not so evident.

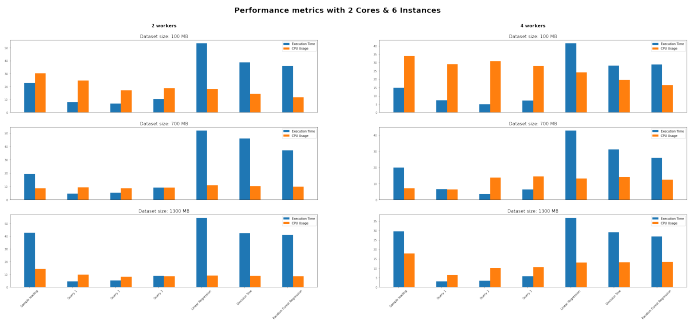

Fig. 6. Performance results with 2 cores and 6 instances.

When comparing the tests in different clusters, the results are quite different. In this specific case, we can observe that with an increase from two to four workers, there is a generalized increase in CPU usage. The dataset with 100 MB was the one that used more CPU among all the tests. The elapsed time presents a slight decrease either in the queries or in the models.

*2) **2 cores and 12 Instances**:* In the second phase of testing, the procedure was maintained to standardise the results, with only the difference that it was intended to be tested at instance level.

As it is possible to see in the Figure 7, the results present differences when compared with the previous test group.

Here, with 2 workers, the results are slightly faster in terms of execution time. CPU usage remained almost unchanged, regardless of the task. The execution time of the models did not show significant improvements as expected.

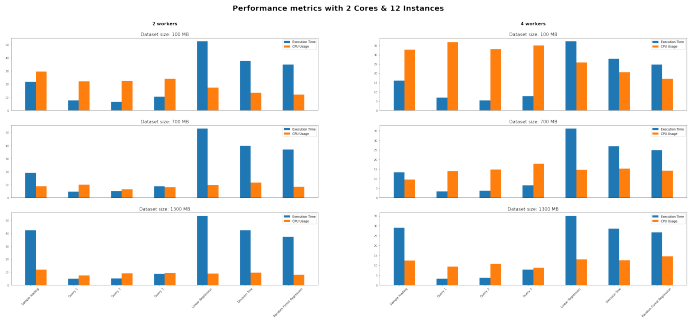

Fig. 7. Performance results with 2 cores and 12 instances.

With the change in the number of workers, from two to four, the previous scenario was repeated again. Although the impact on execution time was not very significant, the same could not be said regarding CPU usage. In this specific parameter, the values increased in almost all tasks, regardless of the dataset size. The execution time of the models is practically unchanged with the increase of instances.

*3) **2 cores and 18 Instances**:* Finally, for the last set of tests we manipulated the number of instances again. In this case, the chosen value was 18 instances.

For the first time, the results at the cluster level with two workers did not present such an evident oscillation as in the previous results.

The execution time remained higher with larger dataset sizes, as expected. Regarding CPU usage, the results once again varied inversely with time, with the 100MB sample presenting the highest CPU usage, as represented in the Figure 8.

Regarding the results obtained with four workers, the trend so far has changed. In the previous results, only the 100MB sample presented a significant increase in CPU usage. In this case, the increase in CPU usage was generalized regardless of the task or dataset size.

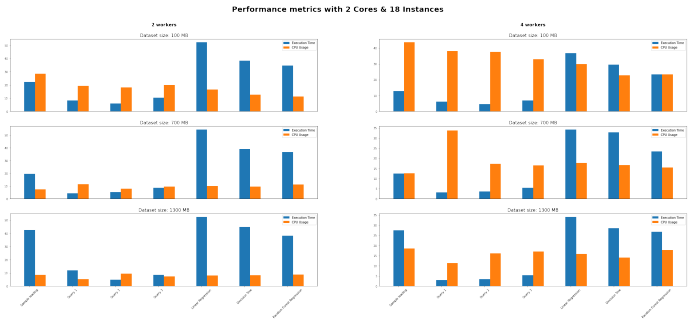

Fig. 8. Performance results with 2 cores and 18 instances.

In terms of execution time, it was in the models that a more accentuated improvement was observed, something that had not happened until this stage.

VI. Recommendation System

As additional work, we created a recommendation system mentioned earlier. This recommendation system consists in creating a dataset by aggregating some columns, which allows grouping the cars according to certain aspects. The columns created are described below:

- Column 'Made', which was brand based, and each brand was associated with the corresponding country.
- Column 'Age', which is calculated by subtracting the current year from the year of manufacture.
- Column 'Mil_Rating' that consists on the average of the number of kilometres by year. If it is above average, we consider it 'above average', if it is below average, we consider it 'below average'.
- Column 'Type_group' where the type of car is evaluated in terms of lust.
- Column 'Car_colour's, with two attributes, dark and light colour. The colour of each car is inserted into each of the types.

After this aggregation, the parameters to insert in the function are the country of manufacture, car type, color, and price range. By inputting these parameters, the function will return a dataframe containing the five cars with the lowest prices within the specified criteria.

VII. Conclusion

A dataset containing 426,880 observations, 26 features and totalling 1.3 gigabytes of data was selected and analysed to evaluate the capabilities of Big Data tools to query and learning as well as evaluate their performance and scalability.

Profiling datasets are a critical aspect of all engineering and data science projects and *Pyspark* provides a huge range of tools to do this. These tools are easy to use and very intuitive. As demonstrated in chapter III, the fact that spark allows the use of *PySpark SQL* makes the tool even more powerful. After the query, the manipulation of the dataframe, as well as the creation of new columns or data is quite simple and practical. The creation of some columns required data aggregation and *Pyspark* is great at doing that.

Regarding machine learning tools, it contains many supervised algorithms for performing regressions. It also contains several algorithms that allow performing classifications, although that was not the objective of this paper.

After training each algorithm, the tool also provides metrics that allow the performance of each model to be evaluated, as shown in Chapter IV.

Finally, the performance and scalability of Apache Spark was also assessed through different tests. Each test allowed the comparison between different instances, dataset sizes and cluster configurations. With the increase in the dataset size, there was an increase in CPU usage in order to maintain a balance in task's execution time. However, the increase in the number of workers did not always contribute towards an improvement in results.

REFERENCES

[1] Hirei, Hassan Mohamed, *Investigating and Validating Scam Triggers: A Case Study of a Craigslist Website*, 2020.

[2] Gegic E. et al, *Car price prediction using machine learning techniques* TEM Journal, 2019.

[3] Kuiper S., *Introduction to Multiple Regression: How Much Is Your Car Worth?*, Journal of Statistics Education, 2008

[4] Kaggle, *"Used Cars Dataset"* [Online] - Available: https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data. [Accessed 25th April 2021]

[5] Demir E., *Big Data Analytics on used car information* , MSc Thesis Instanbul, 2021

[6] Collard M., *A Comparison of Machine Learning Regression Models*, Bsc. Thesis Sweden, 2022

[7] Kumbar K. et al, *Project Report: Predicting Used Car Prices*

[8] Tran et al, *Spam detection in online classified advertisements*, Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality *p. 35-41*, 2011

[9] Garg V. and Nilizadeh S., *Craigslist scams and community composition: Investigating online fraud victimization*, 2013 IEEE Security and Privacy Workshops *p. 123-126*, 2013

[10] Anders Fremstad, *Does Craigslist Reduce Waste? Evidence from California and Florida*, Ecological Economics *p. 135-143*, 2015

[11] Michelson M. and Knoblock C., *Unsupervised information extraction from unstructured, ungrammatical data sources on the world wide web*, International Journal of Document Analysis and Recognition (IJDAR) *p. 211-226*, 2007

[12] Azhari M. et al, *Higgs boson discovery using machine learning methods with pyspark*, Procedia Computer Science *p. 1141-1146* 2020

[13] Le Quoc D. et al, *Sgx-pyspark: Secure distributed data analytics*, The World Wide Web Conference *p. 3564-3563*, 2019