Machine Learning Based Cars Price Prediction & Recommendation Project

*** EL KAHLAOUI Youssef**
**** KHAMJANE Aziz**

Digital Transformation and Artificial Intelligence
National School of Applied Sciences of Al Hoceima

23/12/2024

Abstract- The Car Price Prediction Project aims to develop a machine learning application that accurately predicts the prices of used cars based on various features such as manufacturer, model name, engine type, transmission, mileage, price, and age.

This project involves data scraping from the AA Cars website [1], comprehensive data cleaning and preprocessing, exploratory data analysis, and the implementation of multiple machine learning models. The final model, trained on a combined dataset, demonstrates a slight performance improvement over individual datasets. The project also includes an interactive web interface built with Flask, allowing users to input car details and receive price predictions and recommendations for similar cars. This paper presents the methodology, model evaluation, and results, highlighting the effectiveness of the chosen approach in predicting car prices and providing recommendations.

Index Terms- Car Price Prediction, Machine Learning, Recommendation System, Web Application, Data Scraping

I. Introduction

The used car market is a dynamic and complex industry where accurate pricing is crucial for both buyers and sellers. Predicting the price of a used car involves considering various factors such as the car's manufacturer, engine type, transmission, mileage, price, and age. Traditional methods of car price estimation often rely on expert knowledge and manual assessments, which can be subjective and inconsistent. With the advent of machine learning, it is possible to develop models that can predict car prices more accurately and consistently.

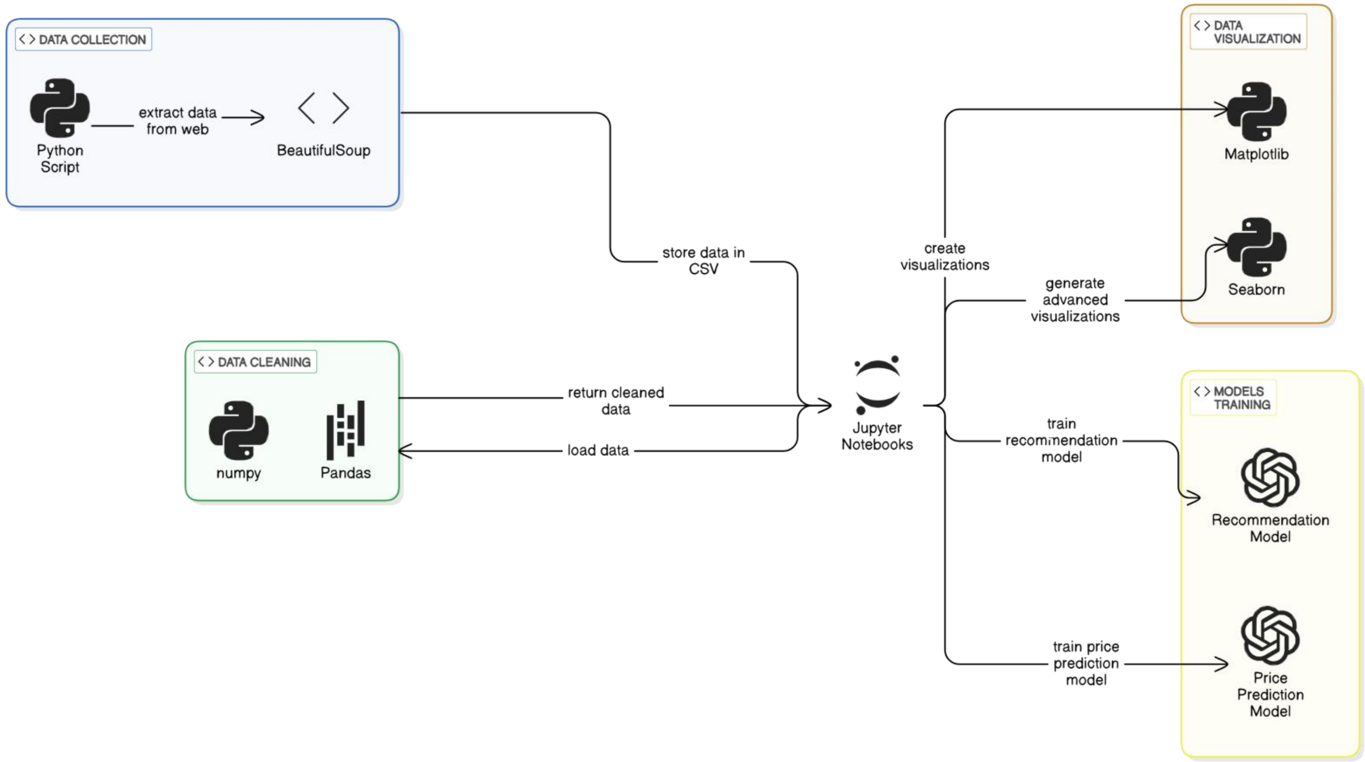
This research paper presents the Car Price Prediction Project, which aims to leverage machine learning techniques to predict the prices of used cars and recommend similar cars based on input features. The project involves several key steps: data scraping from the AA Cars website, data cleaning and preprocessing, exploratory data analysis, and the implementation and evaluation of multiple machine learning models. The final model, trained on a combined dataset, shows a slight performance improvement, indicating the effectiveness of the chosen approach.

The paper is structured as follows: Section 2 describes the architecture of both model and the web application .Section 3 reviews related work in the field of car price prediction and machine learning models. Section 4 describes the methodology, including data collection, preprocessing, feature engineering, model training, and the recommendation system. Section 5 presents the discussion, including model evaluation and comparison as part A. Section 5 part B discusses the limitations of the approach and potential future work. Finally, Section 7 concludes the paper by summarizing the key findings and contributions.

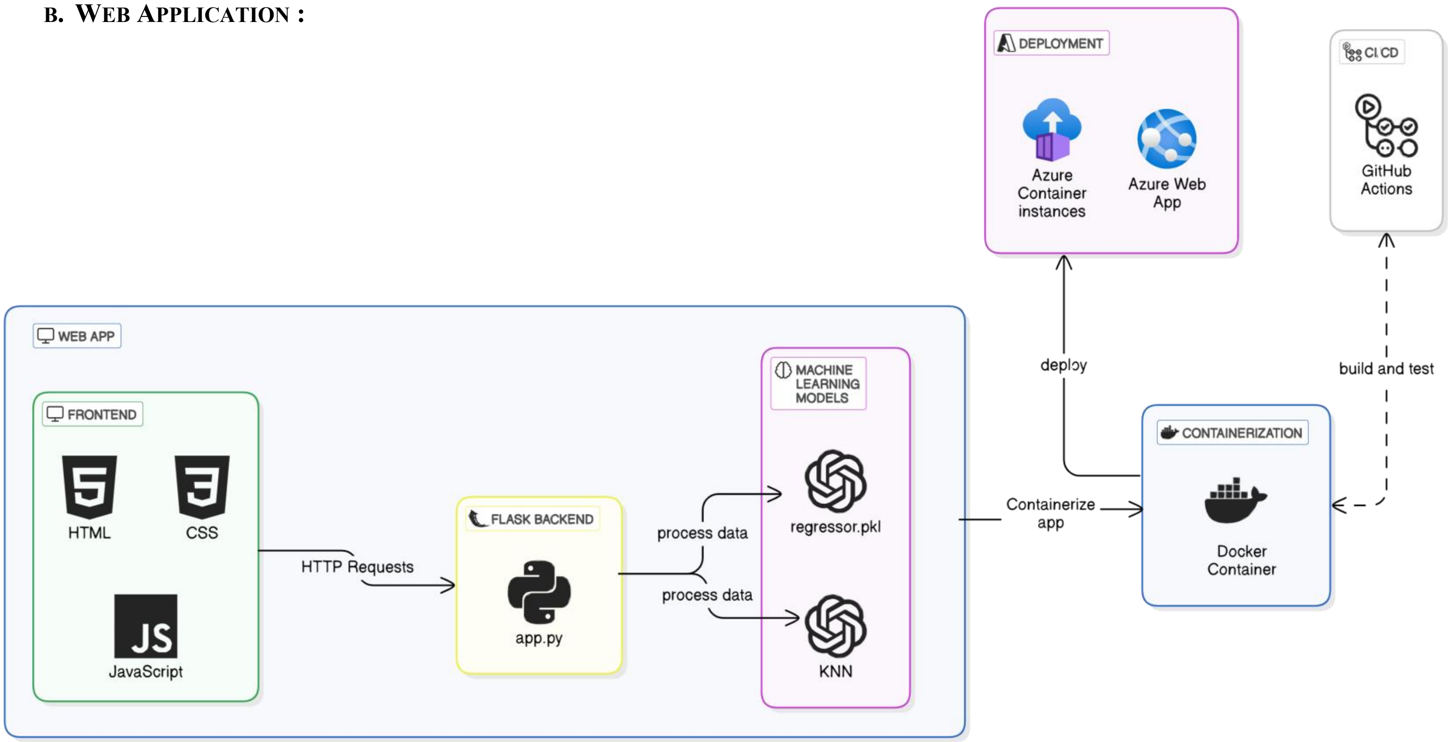
By developing an accurate and reliable car price prediction model and a recommendation system, this project aims to provide valuable insights and tools for both buyers and sellers in the used car market, ultimately contributing to more informed decision-making and fairer pricing.

II. Architecture

A. Models Buildeing :



B. Web Application :



App link : http://mlpro.azurewebsites.net

III. Related Works

Research on estimating the price of used cars is relatively recent and not extensively covered. In her MSc thesis, Listiani [2] demonstrated that a regression model using support vector machines (SVM) can predict the residual price of leased cars more accurately than simple multiple regression or multivariate regression. SVMs are particularly effective in handling high-dimensional data (numerous features used to predict the price) and can avoid both overfitting and underfitting. She employed a genetic algorithm to optimize the SVM parameters efficiently. However, the study did not express the improvement of SVM regression over simple regression in straightforward measures like mean deviation or variance.

In another university thesis, Richardson [4] explored the hypothesis that car manufacturers aim to produce vehicles that do not depreciate quickly. Using multiple regression analysis, he found that hybrid cars (vehicles with both an internal combustion engine and an electric motor) retain their value better than traditional vehicles. This is likely due to increased environmental concerns and higher fuel efficiency. The study also considered other factors such as age, mileage, make, and MPG (miles per gallon). Data for this study was collected from various websites.

Wu et al. [5] utilized a neuro-fuzzy knowledge-based system to predict the price of used cars, considering only three factors: the make of the car, the year of manufacture, and the engine style. The proposed system produced results comparable to simple regression methods. In the USA, car dealers sell hundreds of thousands of cars annually through leasing. Most of these cars are returned at the end of the leasing period and must be resold. Accurately pricing these cars is crucial for economic success. To address this, Du et al. [6] developed the ODAV (Optimal Distribution of Auction Vehicles) system, which not only estimates the best resale price but also advises on the optimal location to sell the car. Given the vast size of the United States, the selling location significantly impacts the price of used cars. A k-nearest neighbor regression model was used for price forecasting. Since its inception in 2003, the system has distributed over two million vehicles.

Gonggi [7] proposed a model based on artificial neural networks to forecast the residual value of private used cars. The study focused on features such as mileage, manufacturer, and estimated useful life. The model was optimized to handle nonlinear relationships, which simple linear regression methods cannot manage. The model proved to be reasonably accurate in predicting the residual value of used cars.

In a study by Sameerchand Pudaruth [8], supervised machine learning techniques were applied to predict the price of used cars in Mauritius. The predictions were based on historical data collected from daily newspapers. Various techniques, including multiple linear regression analysis, k-nearest neighbors, naïve Bayes, and decision trees, were used to make the predictions. The predictions were evaluated and compared to identify the best-performing methods. The study concluded that predicting the price of used cars is a challenging problem that requires sophisticated algorithms for high accuracy. All four methods provided comparable performance.

Inspired by the study conducted by Sameerchand Pudaruth [8], this research explores the application of machine learning techniques to predict the price of used cars. These studies illustrate the diverse approaches and techniques used in predicting the price of used cars, such as support vector machines, multiple regression analysis, neuro-fuzzy systems, k-nearest neighbor regression, and artificial neural networks. Each method has its advantages and limitations, and the choice of method depends on the specific requirements and constraints of the problem.

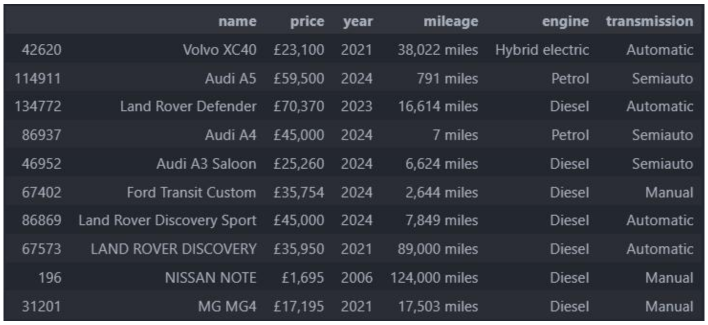
IV. Methodology

The methodology section outlines the comprehensive steps taken to collect, preprocess, analyze, and model the data for predicting used car prices and recommending similar cars. This section is divided into several key parts: data collection, data cleaning and preprocessing, exploratory data analysis, model training and evaluation, and the recommendation system.

A. Data Collection

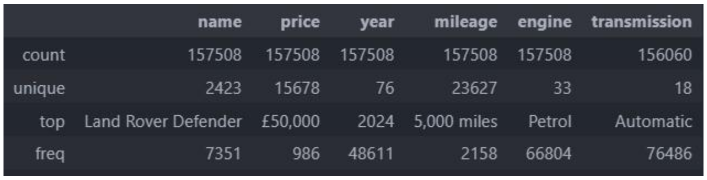
Tools and Libraries: Python, BeautifulSoup, Requests

Data was collected by scraping the AA Cars website using a custom Python script (CarsInfoScraper.py). The script iterates through different price ranges and pages, extracting car details such as **manufacturer**, **model**, **year**, **mileage**, **engine type, transmission type**, and **price**. The scraped data is saved to a CSV file for further processing.
For the moment, it looks like this :



	name	price	year	mileage	engine	transmission
42620	Volvo XC40	£23,100	2021	38,022 miles	Hybrid electric	Automatic
114911	Audi A5	£59,500	2024	791 miles	Petrol	Semiauto
134772	Land Rover Defender	£70,370	2023	16,614 miles	Diesel	Automatic
86937	Audi A4	£45,000	2024	7 miles	Petrol	Semiauto
46952	Audi A3 Saloon	£25,260	2024	6,624 miles	Diesel	Semiauto
67402	Ford Transit Custom	£35,754	2024	2,644 miles	Diesel	Manual
86869	Land Rover Discovery Sport	£45,000	2024	7,849 miles	Diesel	Automatic
67573	LAND ROVER DISCOVERY	£35,950	2021	89,000 miles	Diesel	Automatic
196	NISSAN NOTE	£1,695	2006	124,000 miles	Diesel	Manual
31201	MG MG4	£17,195	2021	17,503 miles	Diesel	Manual

Fig.1: data-set visualisation



	name	price	year	mileage	engine	transmission
count	157508	157508	157508	157508	157508	156060
unique	2423	15678	76	23627	33	18
top	Land Rover Defender	£50,000	2024	5,000 miles	Petrol	Automatic
freq	7351	986	48611	2158	66804	76486

Fig.2:data-set information

Using about 160K rows of cars information, we should clean our data-set and get useful information from it .

B. Data Cleaning and Pteprocessing :

Tools and Libraries: Python, Pandas, Numpy

The raw data was cleaned and preprocessed using a Jupyter notebook (DataCleaner.ipynb). Key steps included:

✓ *Removing Missing Values and Duplicates*: Ensuring data integrity by eliminating incomplete or redundant entries.

✓ *Standardizing Categorical Values*: Converting categorical features (e.g., engine types, transmission types) to a consistent format.

✓ *Currency Conversion*: Converting prices from GBP to MAD by multiplying by 12.8 to make the data relevant for the Moroccan market.

✓ *Feature Engineering*: Creating new features such as car age by subtracting the year of manufacture from the current year, divide name to two columns, one for manufacturer and the other one to spesify the car model which is the name.

✓ Unit Conversion: Converting mileage from miles to kilometers.

✓ Unifying transmission types to 3 categories

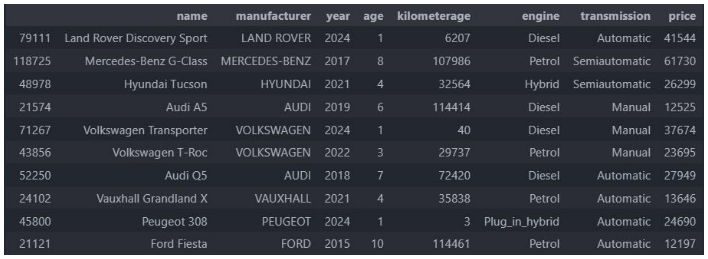
- Automatic
- Sumiautomatic
- Manual

✓ Unifying engines to 5 types of engines

- Diesel
- Petrol
- Hybrid
- Plug in hybrid
- Electric

✓ Remove cars with number of the same model is less than 10 times presented

As final results we got the following data-set with the shape (`65388, 8`) then save it as ready to use data-set on CSV form :



	name	manufacturer	year	age	kilometerage	engine	transmission	price
79111	Land Rover Discovery Sport	LAND ROVER	2024	1	6207	Diesel	Automatic	41544
118725	Mercedes-Benz G-Class	MERCEDES-BENZ	2017	8	107986	Petrol	Semiautomatic	61730
48978	Hyundai Tucson	HYUNDAI	2021	4	32564	Hybrid	Semiautomatic	26299
21574	Audi A5	AUDI	2019	6	114414	Diesel	Manual	12525
71267	Volkswagen Transporter	VOLKSWAGEN	2024	1	40	Diesel	Manual	37674
43856	Volkswagen T-Roc	VOLKSWAGEN	2022	3	29737	Petrol	Manual	23695
52250	Audi Q5	AUDI	2018	7	72420	Diesel	Automatic	27949
24102	Vauxhall Grandland X	VAUXHALL	2021	4	35838	Petrol	Automatic	13646
45800	Peugeot 308	PEUGEOT	2024	1	3	Plug_in_hybrid	Automatic	24690
21121	Ford Fiesta	FORD	2015	10	114461	Petrol	Automatic	12197

Fig.3:data-set after Feature Engineering

C. <u>Exploratory Data Analysis:</u>

Tools and Libraries: Python, Matplotlib, Seaborn

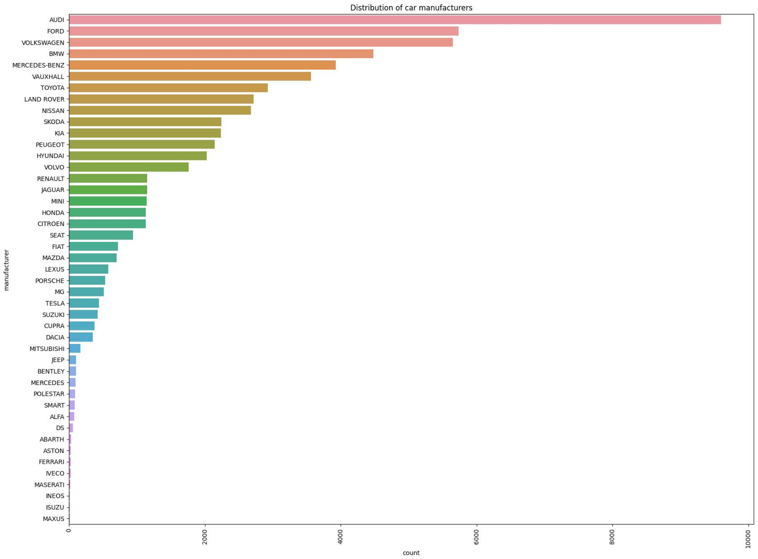
Exploratory data analysis was performed using a Jupyter notebook (DataVisualisation.ipynb). Key visualizations included:

<u>Distribution Analysis</u>**:**

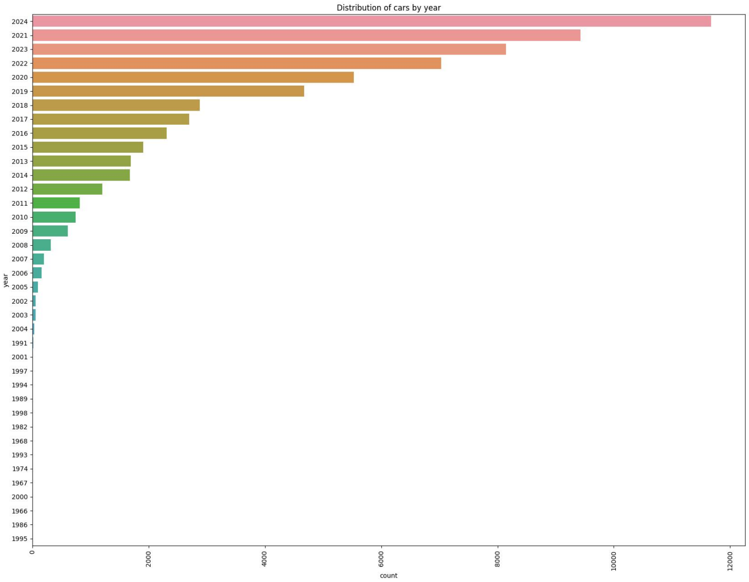
Visualizing the distribution of car prices, mileage, age and count .

(zoom-in please for a clear reading)

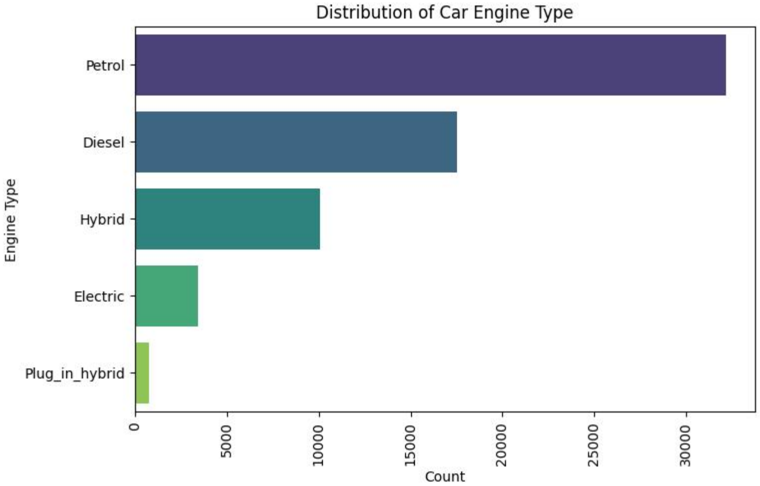
- *Fig.4 :Distribution of car manufacturers:*



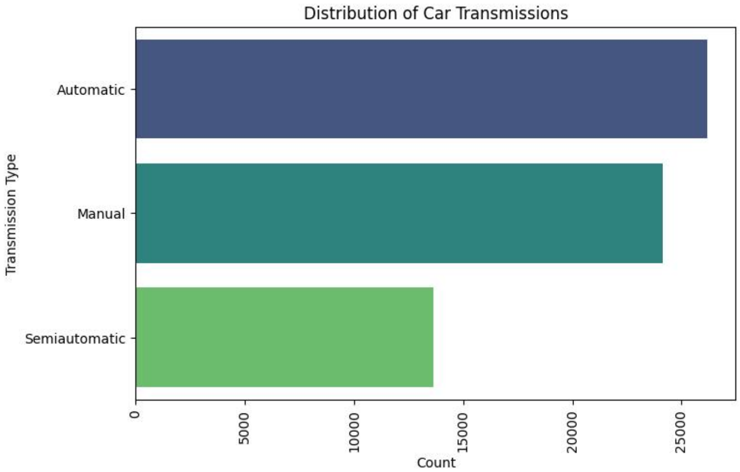
- *Fig.5 :Distribution of car by year :*



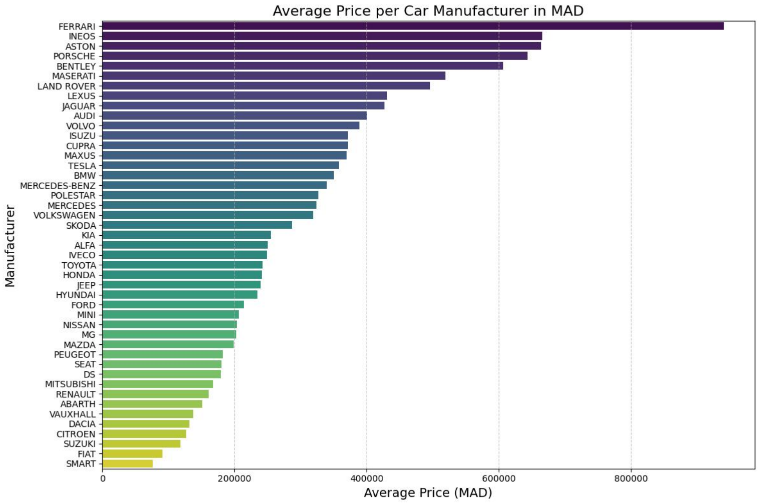
- *Fig.6 :Distribution of Car Engine Type :*



- *Fig.7 :Distrubution of Car Transmission :*



- *Fig.8 :Avrage Price Car Manufacturer :*



Correlation Analysis:

Creating a correlation matrix and heat-map to identify relationships between features, Age, Year,Price and Kilometerage :

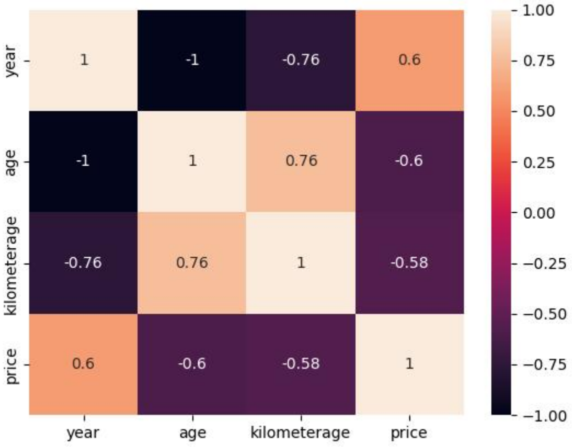
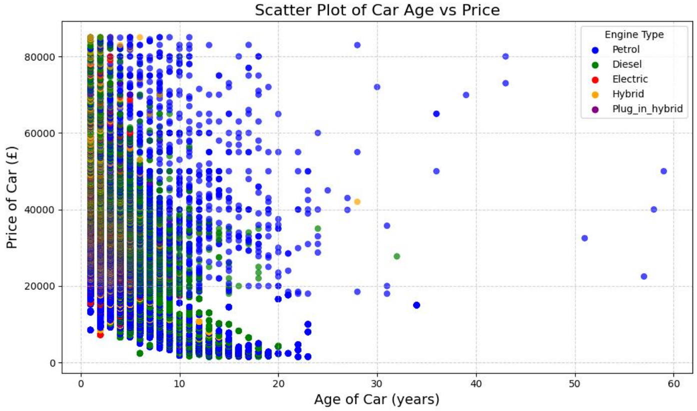


Fig.9: Correlation heatmap

Scatter Plots:

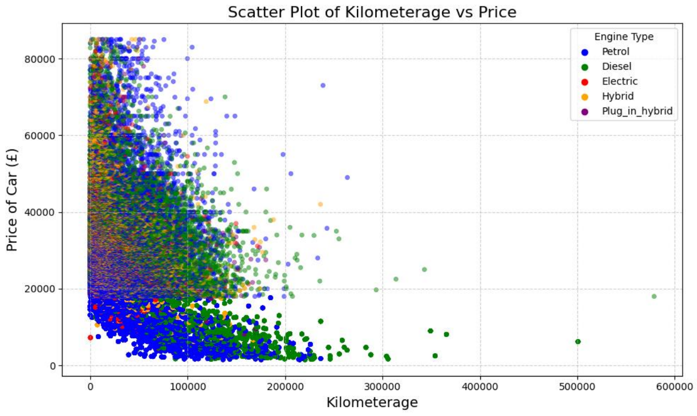
Visualizing the relationship between car age, mileage, and price.

- *Fig.10 :Scatter Plot of Car Age vs Price :*



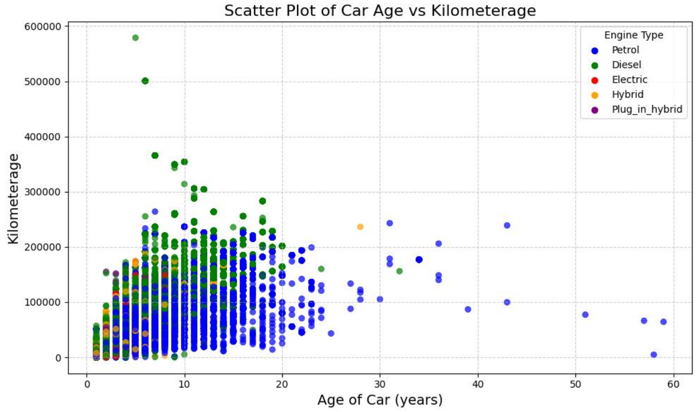
There is negative correlation between the age and prices of cars. As the age increases, the price reduces

- *Fig.11 :Scatter Plot of Car Kilometerage vs Price :*



There is negative correlation between the mileage and prices of the cars. As the number of miles travelled increases, the price reduces

- *Fig.12 :Scatter Plot of Car Kilometerage vs Age:*



There is a positive correlation between the mileage and age of the cars. the number of miles travelled increases with age.

D. MODEL TRAINING AND EVALUATION:

Tools and Libraries: Python, Scikit-learn, XGBoost, CatBoost, LightGBM, RandomForestm, GridSearchCV.

Multiple machine learning models were trained and evaluated using a Jupyter notebook (Models.ipynb). The steps included:

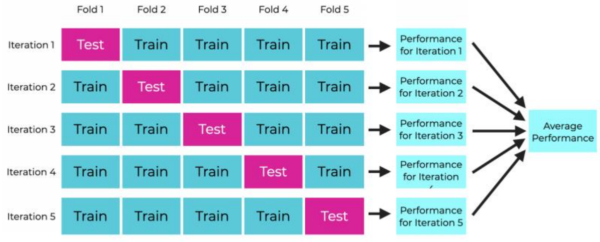
- Data Splitting: Splitting the dataset into training and testing sets to evaluate model performance.

- Feature Encoding: Encoding categorical features using LabelEncoder.

App link : http://mlpro.azurewebsites.net

- Feature Scaling: Scaling numerical features using MinMaxScaler to normalize the data.

In order to robustly assess the performance of regression models and ensure their generalizability, a systematic approach was adopted that involved the use of cross-validation coupled with the implementation of diverse regression algorithms. The dataset, consisting of car listings with various features such as manufacturer, engine type, transmission, mileage, price, and age, was subjected to a meticulous cross-validation procedure. Specifically, a 5-fold cross-validation scheme was employed, where the dataset was partitioned into five subsets, allowing each fold to serve alternately as a training set and a validation set. This methodological choice was motivated by the relatively moderate size of the dataset, aiming to strike a balance between computational efficiency and obtaining reliable estimates of model performance.



	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5		
Iteration 1	Test	Train	Train	Train	Train	Performance for Iteration 1	
Iteration 2	Train	Test	Train	Train	Train	Performance for Iteration 2	
Iteration 3	Train	Train	Test	Train	Train	Performance for Iteration 3	Average Performance
Iteration 4	Train	Train	Train	Test	Train	Performance for Iteration	
Iteration 5	Train	Train	Train	Train	Test	Performance for Iteration 5	

Fig.13: Cross-Validation Method

1. Algorithms :

- **Linear Regression With RidgeExtension:** Ridge regression is a linear regression model with L2 regularization, which helps to prevent overfitting by adding a penalty to the size of the coefficients. The cost function for Ridge regression is:

$$RSS_{ridge}(w,b) = \underbrace{\sum_{i=1}^{n}(y_i - (w_i x_i + b))^2}_{\text{Fit training data well}} + \alpha \overbrace{\underbrace{\sum_{j=1}^{p} w_j^2}_{\text{Keep parameters small}}}^{\text{L2 penalty / Penalty Term / Regularisation Term}}$$

A trade-off between fitting the training data well and keeping parameters small

- **Random Forest:** An ensemble learning method that constructs multiple decision trees and merges them to get a more accurate and stable prediction. Key parameters include the number of trees (n_estimators) and the maximum depth of the trees (max_depth).

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(\overset{\text{Real value (label) known from the training data-set}}{y_i}, \underbrace{\hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)}_{\text{Can be seen as } f(x + \Delta x) \text{ where } x = \hat{y}_i^{(t-1)}}) + \Omega(f_t)$$

- **XGBoost:** An optimized gradient boosting algorithm that uses decision trees. The objective function combines a loss function and a regularization term at iteration t that we need to minimize is the following:

- **CatBoost:** A gradient boosting algorithm that handles categorical features efficiently. It uses ordered boosting to reduce overfitting and improve accuracy.

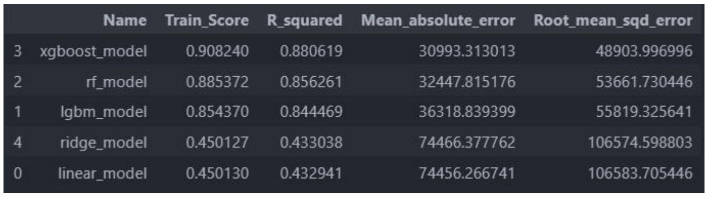
- **LightGBM:** A gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with large datasets.

2. ***Evaluation Metrics :***

- R-squared (R²): Measures the proportion of variance in the dependent variable that is predictable from the independent variables.

- Mean Absolute Error (MAE): The average of the absolute errors between the predicted and actual values.

- Root Mean Squared Error (RMSE): The square root of the average of squared differences between predicted and actual values.



	Name	Train_Score	R_squared	Mean_absolute_error	Root_mean_sqd_error
3	xgboost_model	0.908240	0.880619	30993.313013	48903.996996
2	rf_model	0.885372	0.856261	32447.815176	53661.730446
1	lgbm_model	0.854370	0.844469	36318.839399	55819.325641
4	ridge_model	0.450127	0.433038	74466.377762	106574.598803
0	linear_model	0.450130	0.432941	74456.266741	106583.705446

Fig.14: Models Evaluation

The final model, an XGBoost regressor, was selected based on its superior performance.

3. ***Hyperparameter Tuning :***

GridSearchCV:
Used to find the optimal hyperparameters for each model by performing an exhaustive search over a specified parameter grid. For example, for Ridge Regression, the grid might include different values for the regularization parameter (\lambda). The optimal hyperparameters are selected based on cross-validation performance.

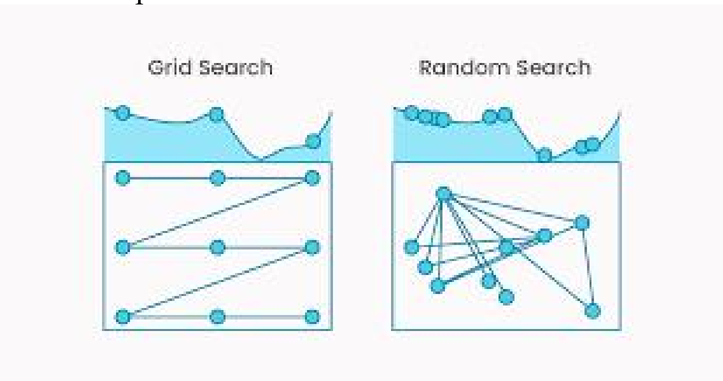
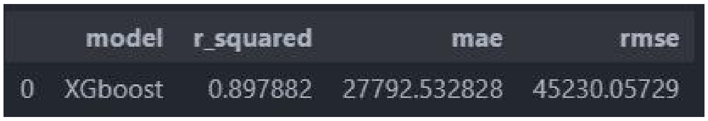


Fig.14 :Grid Search Vs Random Search on two hyperparameters tuning

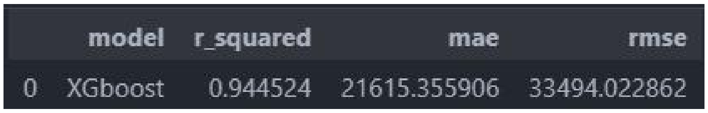
By Tuning our model hyperparameters we got the following results:



	model	r_squared	mae	rmse
0	XGboost	0.897882	27792.532828	45230.05729

Fig.15 :XGBoost Model Evaluation after hyperparameters tuning

The selected model was retrained on the entire dataset (both training and testing sets) to maximize the use of available data and improve the model's predictive power. This final model was then saved as a pickle file (regressorfinal.pkl) for deployment.



	model	r_squared	mae	rmse
0	XGboost	0.944524	21615.355906	33494.022862

Fig.16 :XGBoost Model Evaluation after training on entire dataset

E. Recommendation System:

Tools and Libraries: Python, Scikit-learn

A recommendation system was implemented using the K-Nearest Neighbors (KNN) algorithm to suggest similar cars based on input features, radius and neighbors values. The steps included:

1. *Feature Encoding and Scaling:* Encoding and scaling input features using the same encoders and scalers used during model training.

2. *Training the KNN Model:* Training the KNN model on the entire dataset to find the nearest neighbors.

3. *Generating Recommendations*: Returning a list of similar cars based on the input features.

During the development of the recommendation system, the cosine similarity metric was initially considered. However, it was found to be irrelevant for this project due to the large size of the data-set, which demanded about 60GB of RAM, making it impractical for efficient computation. Therefore, the KNN algorithm was chosen as a more feasible alternative.

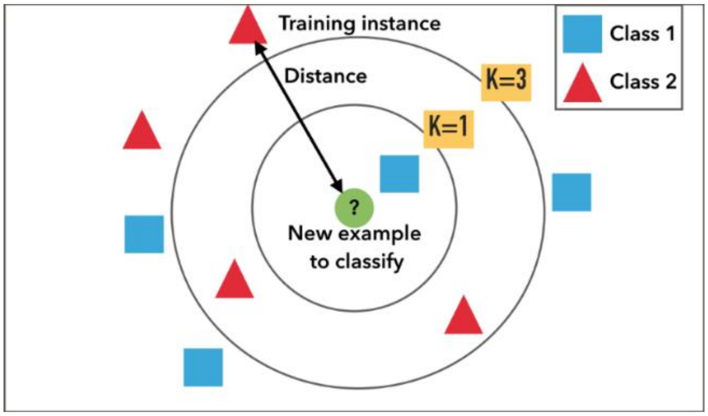


Fig.17 :KNN-based Recommendation system classification concept

F. Web Application and Deployment:

Tools and Libraries: Python, Flask, Docker, Azure Web App, GitHub Actions.

The web application was developed to provide an interactive interface for users to input car details and receive price predictions and recommendations for similar cars. The application was built using Flask, a lightweight web framework for Python, and deployed using Docker and Azure Web App. Continuous integration and deployment were managed using GitHub Actions to automate the build and deployment process.

Steps Involved in Web Application Development and Deployment:

1. *Web Application Development:*

Frontend: The frontend of the web application was built using HTML, CSS, and JavaScript. This provided a user-friendly interface for users to input car details such as manufacturer, model, year, mileage, engine type, and transmission type.
Backend: The backend was developed using Flask. Flask handles HTTP requests, processes input data, and communicates with the machine learning models to generate predictions and recommendations. The backend also serves the HTML templates and static files required for the frontend.

2. *Containerization*:

Docker: The entire application, including the frontend, backend, and machine learning models, was containerized using Docker. Docker ensures that the application can be easily deployed and run in any environment by packaging all dependencies and configurations into a single container.

3. *Continuous Integration and Deployment:*

GitHub Actions: GitHub Actions was used to automate the build and deployment process. The workflow included steps to build the Docker image, run tests, and deploy the application to Azure Web App. This ensured that any changes to the codebase were automatically tested and deployed, maintaining the application's reliability and availability.

4. *Deployment:*

Azure Web App: The Docker container was deployed to Azure Web App for hosting. Azure Web App provides a scalable and secure environment for running web applications. The deployment process ensured that the application was accessible online and could handle user requests efficiently.
here we enlist the proven steps to publish the research paper in a journal.

V. LIMITATIONS :

1. DATA QUALITY AND AVAILABILITY:

- The accuracy of predictions depends on the quality of data scraped from the AA Cars website, which may contain inaccuracies or inconsistencies.
- The dataset may not represent the entire used car market, leading to biased predictions.

2. GEOGRAPHICAL AND CURRENCY LIMITATIONS :

- The data is from the UK market (AA Cars website), but the project targets Moroccan users. Prices were converted from GBP to MAD by multiplying by 12.8, but regional factors affecting car prices in Morocco may not be captured.
- Differences in market dynamics and economic conditions between the UK and Morocco could impact prediction accuracy.

3. FEATURE LIMITATIONS :

- Important factors like car condition, service history, and market demand were not included due to data unavailability.
- External factors such as economic conditions and seasonal trends were not considered.

4. MODEL COMPLEXITY AND INTERPRETABILITY :

- Complex models like XGBoost and CatBoost are less interpretable than simpler models, making it challenging to explain predictions.
- The KNN-based recommendation system may not scale well with large datasets.

5. HYPERPARAMETER TUNING :

- Hyperparameter tuning using GridSearchCV is computationally expensive and time-consuming. More advanced techniques could yield better results.
- Optimal hyperparameters may not be universally applicable, limiting generalizability.

6. DEPLOYMENT AND MAINTENANCE:

- Deployment on Azure Web App introduces challenges related to scalability, security, and maintenance.
- Regular updates to the model and data are necessary to maintain accuracy and relevance.

7. USER INTERACTION AND EXPERIENCE :

- The web interface may not cater to all user needs. Enhancing the user experience with additional features could improve satisfaction.
- The recommendation system may not always align with user preferences. Incorporating user feedback could enhance its relevance.

By acknowledging these limitations, we can identify areas for improvement and future work to enhance the Car Price Prediction Project's accuracy, robustness, and usability.

VI. Conclusion

The Car Price Prediction Project successfully demonstrates the application of machine learning techniques to predict the prices of used cars and recommend similar cars based on input features. By leveraging data scraped from the AA Cars website, comprehensive data cleaning and preprocessing, and the implementation of multiple machine learning models, the project achieved a slight performance improvement with the final model trained on a combined dataset.

The project highlights the effectiveness of advanced machine learning models such as XGBoost and CatBoost in handling high-dimensional data and providing accurate predictions. Additionally, the K-Nearest Neighbors (KNN) algorithm proved useful in developing a recommendation system that suggests similar cars based on user input.

Despite the promising results, several limitations were identified, including data quality and availability, geographical and currency differences, feature limitations, model complexity, and deployment challenges. Addressing these limitations in future work could further enhance the accuracy, robustness, and usability of the prediction and recommendation systems.

The deployment of the web application using Flask and Azure Web App ensures accessibility and provides a user-friendly interface for users to input car details and receive predictions. However, ongoing maintenance and updates are necessary to keep the application relevant and accurate.

In conclusion, this project provides valuable insights and tools for both buyers and sellers in the used car market, contributing to more informed decision-making and fairer pricing. Future work should focus on addressing the identified limitations, incorporating additional features, and exploring more sophisticated algorithms to further improve the system's performance and user experience.

App link : http://mlpro.azurewebsites.net

References

[1] NATIONAL TRANSPORT AUTHORITY. 2014. Accessible from : [link]

[2] MOTORS MEGA. 2014. accessible from : [link]

[3] LISTIANI, M., 2009. Support Vector Regression Analysis for Price Prediction in a Car Leasing Application. Thesis (MSc). Hamburg University of Technology.

[4] RICHARDSON, M., 2009. Determinants of Used Car Resale Value. Thesis (BSc). The Colorado College.

[5] WU, J. D., HSU, C. C. AND CHEN, H. C., 2009. An expert system of price forecasting for used cars using adaptive neuro-fuzzy inference. Expert Systems with Applications. Vol. 36, Issue 4, pp. 7809-7817.

[6] DU, J., XIE, L. AND SCHROEDER S., 2009. Practice Prize Paper - PIN Optimal Distribution of Auction Vehicles System: Applying Price Forecasting, Elasticity Estimation and Genetic Algorithms to Used-Vehicle Distribution. Marketing Science, Vol. 28, Issue 4, pp. 637-644

[7] GONGGI, S., 2011. New model for residual value prediction of used cars based on BP neural network and non-linear curve fit. In: Proceedings of the 3rd IEEE International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Vol 2. pp. 682-685, IEEE Computer Society, Washington DC, USA.

[8] International Journal of Information & Computation Technology. ISSN 0974-2239 Volume 4, Number 7 (2014), pp. 753-764, accessibel from : [link]

Members

***EL KAHLAOUI Youssef** : Author name, engineering student , Digital Transformation and Artificial Intelligence option, accessible from : youssef.elakahlaoui@etu.uae.ac.ma

****KHAMJANE Aziz** :– Supervisor name, P.hd, Assistant professor, Abdelmalek Essaadi University
Verified email at uae.ac.ma
accessible from: aziz.khamajane@uae.ac.ma

To Accesses Web Application :

SCAN ME