University of Hertfordshire UH

School of Physics, Engineering and Computer Science

MSc Data Science Project

7PAM2002-0509-2024

Department of Physics, Astronomy and Mathematics

Data Science FINAL PROJECT REPORT

Project Title:

USING MACHINE LEARNING TO PREDICT USED CAR PRICES

Student Name and SRN:

AUSTINE IGHO EFENARUA, 23096669

Supervisor: DAVID LAGATTUTA

Date Submitted: 26th August, 2025

Word Count: 4,982

GitHub Link :

1. Report
2. Code

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science **in Data Science** at the University of Hertfordshire.

I have read the detailed guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6)

I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: AUSTINE IGHO EFENARUA

Student Name signature: AUSTINE IGHO EFENARUA

Student SRN number: 23096669

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

Abstract

This research applies machine learning to predict used car prices using attributes such as brand, mileage, model year, fuel type, and accident history. The dataset was cleaned through removal of symbols, missing values, and duplicates, with car age engineered as an additional feature. Exploratory Data Analysis highlighted the influence of mileage, age, and fuel type on pricing patterns. Three models such as Linear Regression, Random Forest, and XGBoost were trained and evaluated using an 80:20 train-test split. Performance was measured with MAE, RMSE, and R^2 scores. While Random Forest initially showed strong predictive capability, hyperparameter tuning improved both ensemble models, with XGBoost having R^2 of 75% emerging as the best overall. Feature importance analysis confirmed mileage and car age as the most significant predictors of car prices. The study concludes that optimized XGBoost provides the most reliable framework for accurate used car price prediction, supporting fair valuation in automotive markets.

Contents

1.0 Introduction

1.1 Background

The second-hand vehicle market has seen tremendous growth in recent years, driven by a variety of economic, social, and environmental factors. With the increasing demand for used cars, the complexity of pricing and valuation has escalated, making accurate pricing forecasts critical for market participants. Buyers and sellers both benefit from transparent, well-informed decisions about the pricing of used cars. Traditional valuation methods, which typically rely on subjective assessments, manual appraisals, and inconsistent pricing standards, have proven inadequate in meeting the growing need for consistency and reliability in vehicle pricing.

Historically, used car pricing has been based on a range of factors including brand, age, mileage, model, fuel type, and market conditions. However, the subjective nature of traditional pricing methods has often resulted in market inefficiencies, leading to price inaccuracies and a lack of confidence among consumers. With the advent of machine learning (ML) technologies, there has been a noticeable shift toward more objective, data-driven methods of pricing used vehicles. Machine learning algorithms such as Multiple Linear Regression (MLR), Random Forest, XGBoost, and Neural Networks are now at the forefront of used car price prediction, offering promising advancements in model accuracy and efficiency (Gao, 2024).

Recent advancements in ML algorithms such as Multiple Linear Regression (MLR), Random Forest (RF), XGBoost, and Neural Networks have significantly improved prediction accuracy. The ability of these models to process large volumes of data and identify hidden patterns makes them ideal for forecasting the prices of used cars. For instance, Random Forest and XGBoost, two of the most widely used models in this context, have shown great promise in enhancing predictive accuracy, outperforming traditional methods (Najib, 2023; Zhu, 2023; Guo and Zhang, 2024).

1.2 Problem Statement

Despite advancements, accurately predicting used car prices remains challenging due to numerous influencing factors such as brand, age, mileage, model, fuel type, transmission, and broader economic conditions. The persistence of subjective pricing and data inconsistency creates market inefficiencies, negatively impacting stakeholder confidence. Therefore, a critical need exists to develop and validate robust, data-driven ML models capable of delivering consistent and reliable predictions (Deepak et al., 2023; Bhatt et al., 2023).

Even with advancements in ML models, the persistence of subjective pricing practices continues to be a significant barrier to efficiency and accuracy in the used car market. Sellers often price their vehicles based on intuition or the personal value they attach to the car, leading to inconsistencies in pricing and a lack of trust in the market. Similarly, potential buyers are left to make decisions based on fragmented, often unreliable, information. The result is a market fraught with inefficiencies, where both parties (buyers and sellers) are not able to rely on accurate, standardized pricing mechanisms.

This project aims to address these inefficiencies by developing robust, data-driven machine learning models that can provide consistent and reliable price predictions for used vehicles, thereby improving market efficiency and buyer confidence. Through the implementation of advanced machine learning algorithms, this study seeks to enhance the used car market by providing more accurate price forecasting tools and insights.

1.3 Justification

The importance of employing machine learning (ML) models in the used car market cannot be overstated. ML models, with their ability to process large datasets and uncover complex patterns, offer a transformative potential that goes beyond traditional pricing methods. By replacing subjective human judgments with objective, data-driven predictions, ML models provide a more accurate, consistent, and efficient means of determining the price of used vehicles.

Studies have demonstrated that machine learning algorithms, particularly Random Forest models, outperform traditional methods in terms of prediction accuracy. Gao (2024) found that Random Forest models achieved an R-squared value of 87.3%, significantly higher than the 84.3% R-squared value obtained using Multiple Linear Regression. This substantial improvement in accuracy is indicative of the power of machine learning models to learn from historical data and make predictions based on a wide range of variables that influence used car prices. Similarly, Guo and Zhang (2024) highlighted the performance of the XGBoost model, which exhibited a Mean Absolute Error (MAE) of approximately 950 USD, demonstrating its ability to provide reliable pricing estimates in a fast-paced, dynamic market.

The growing body of literature supporting the use of machine learning in pricing strategies further solidifies the justification for using these models in the automotive sector. Narayana et al. (2021) demonstrated that predictive analytics powered by machine learning can help retailers optimize their pricing strategies, ensuring that they remain competitive while maximizing revenue. These findings align with the results presented by Pal et al. (2018), who showed that Random Forest models could effectively predict prices in a variety of commercial contexts, influencing their widespread adoption in the industry.

1.4 Research Questions

The research seeks to address two central questions that will guide the exploration of machine learning techniques in predicting used car prices:

1. **What meaningful insights, trends, or patterns can be discovered from Exploratory Data Analysis (EDA)?**

2. **Which machine learning model provides the most accurate price predictions?**

3. **What are the most important features that influence the price of used cars?**

1.5 Project Objectives

The objectives of this research are as follows:

1. **To Perform Exploratory Data Analysis (EDA).**

2. **To build, evaluate and compare three machine learning models**.

3. **To Perform Feature Importance Analysis**

2.0 Literature Review

In recent years, the prediction of used car prices using machine learning has become an area of extensive research, with various models demonstrating promising results. The focus has been on improving prediction accuracy using different algorithms.

Several studies have showcased the use of advanced machine learning techniques for price prediction. Gao (2024) utilized multiple linear regression and random forest models, with the random forest model achieving 87.3% accuracy. Guo and Zhang (2024) applied XGBoost, which outperformed other models, delivering an accuracy of 91.4%. Pal et al. (2018) used Random Forest for price prediction, achieving an accuracy of 89.4%. Kang et al. (2022) compared regression models and found that gradient boosting performed the best with an accuracy of 88.5%. Yadav et al. (2021) combined machine learning techniques with object detection in their system, UCPAS, achieving 90.1% accuracy. Other models have also shown solid performance, such as Idris et al. (2020), with Feed-forward Backpropagation achieving 85.6% accuracy, and Kalpana et al. (2022), who applied decision trees and SVM, achieving 84.2% accuracy.

This study is grounded in Statistical Learning Theory, which provides the foundation for many machine learning algorithms. The theory addresses the problem of identifying patterns in data and generalizing from training samples to unseen examples, a concept crucial to price prediction in dynamic markets such as used cars. The core principle is to minimize prediction error by balancing model complexity with generalization capability.

The conceptual framework for this study models the relationship between car features and their market value. Independent variables include categorical and numerical attributes such as brand, model, mileage, model year, fuel type, engine type, accident history, and transmission. These variables serve as inputs into machine learning algorithms that aim to predict the dependent variable: used car price. The framework assumes that the complex interplay between these attributes contributes to price variation.

Comparative analyses of different machine learning algorithms have consistently supported the superiority of ensemble methods such as Random Forest and XGBoost. For example, Bhatt et al. (2023) conducted an empirical study and found that ensemble models outperformed single-model approaches in terms of prediction accuracy. This finding is consistent with previous studies by Kalpana et al. (2022) and Kang et al. (2022), who observed that ensemble techniques generally offered lower error margins and more reliable price predictions. Narayana et al. (2021) added context by analyzing ML integration in retail business strategies, highlighting practical use cases in commercial environments.

Practical applications of ML models demonstrate significant benefits in the used car market. Shanti et al. (2021) developed a mobile application that leverages machine learning algorithms to provide real-time price predictions for used cars, significantly enhancing the decision-making process for consumers and increasing market transparency. This demonstrates the practical potential of ML models to transform the way prices are determined in the used car market, making them more dynamic and adaptable to market fluctuations.

Furthermore, Yadav et al. (2021) and Idris et al. (2020) have shown the effectiveness of neural networks and deep learning algorithms in the automotive sector, which has led to reduced pricing errors and more informed purchasing decisions. Pal et al. (2018) provided early validation of Random Forest's commercial potential, which has since influenced industry-wide adoption in valuation systems.

While a significant body of research highlights the success of machine learning models in predicting used car prices, several gaps remain. First, many existing studies focus on data from mature automotive markets such as the United States, United Kingdom, Europe, or China, with limited exploration in emerging markets like Nigeria or sub-Saharan Africa. This geographic gap limits the applicability of those models to different market contexts. Second, most studies prioritize model accuracy but pay less attention to model interpretability, leaving end-users and stakeholders without clear reasoning behind predictions. Additionally, few studies explore the real-time deployment of these models in applications or dealership systems. These gaps highlight the need for further work that is context-aware, interpretable, and practically implementable, a goal this study aims to address.

This literature review has examined the growing role of machine learning in used car price prediction. The review highlighted key algorithms such as Random Forest, XGBoost, and neural networks, noting their strong predictive performance and adaptability to high-dimensional data. Empirical evidence consistently supports the superiority of ensemble methods over traditional regression approaches. Studies also demonstrate the real-world utility of ML in applications such as mobile apps for dynamic pricing. However, gaps remain in regional coverage, model transparency, and deployment practicality. These insights form the basis for this research, which seeks to build accurate, interpretable, and context-specific models for used car price prediction.

In summary, the XGBoost model demonstrated the best performance, with 91.4% accuracy, followed by UCPAS (90.1%) and Random Forest (89.4%). These models have proven to be effective tools for predicting used car prices in various datasets and applications.

3.0 Methodology

3.1 Research Design

This study outlines the research approach I applied, for leveraging machine learning techniques to develop predictive models for used car prices. The design is experimental in nature, involving the use of supervised learning algorithms to understand the relationship between vehicle features (independent variables) and market **prices** (dependent variable or target variable). My aim is to build robust models that accurately predict used car prices based on key features such as brand, mileage, model year, engine type, and accident history.

3.2 Data Source and Collection

The dataset used was sourced from a Kaggle dataset. It contains 4,009 observations across 12 variables. These include categorical and numerical variables such as:

1. Brand
2. model
3. Model year
4. Mileage
5. Fuel type
6. Engine specifications
7. Transmission type
8. Exterior
9. interior colors
10. Accident history
11. Clean title status
12. Price (**target variable**)

Dataset Link: https://www.kaggle.com/datasets/taeefnajib/used-car-price-prediction-dataset?resource=download

3.2.1 Ethical Considerations

i. General Data Protection Regulation (GDPR)

The General Data Protection Regulation (GDPR) is a key legal framework that governs the collection and processing of personal data within the European Union. Although my research does not involve personal data or human subjects, GDPR principles were still considered. The dataset used contains no personally identifiable information (PII) such as names, addresses, or contact details. All data processing was conducted locally in a secure environment, and the dataset was only used for academic purposes. No re-identification of individuals was attempted, and no data was shared with third parties, ensuring compliance with GDPR standards.

ii. University of Hertfordshire (UH) Ethical Policies

My research aligns with the ethical standards set by the University of Hertfordshire. The project followed UH's guidelines on integrity, transparency, and responsible research conduct. Since there was no involvement of human participants or sensitive data, the project was considered low-risk. A self-assessment was completed in line with UH’s ethical review procedures, confirming that the research does not require full ethical board approval. I ensured that data was handled responsibly, findings were reported truthfully, and all sources were properly acknowledged.

iii. Permission to Use the Data

The dataset I used in this study was obtained from a public domain and is believed to be available for educational and research purposes. There were no restrictions such as licensing fees, login barriers, or user agreements attached to its use. It was downloaded in good faith under the assumption of open access. The dataset was used solely for non-commercial academic work, and any future use will continue to acknowledge the original source. I did not modify the source or claim ownership of the dataset.

iv. Ethical Data Collection

I used secondary data and all Ethical standards were complied with. Data cleaning and preprocessing were done in a way that preserved the integrity of the original values. No web scraping or automated data harvesting was involved. The dataset was used responsibly, and care was taken to avoid introducing bias or misrepresenting the information. All transformations were documented to ensure transparency and reproducibility.

3.3 Exploratory Data Analysis (EDA)

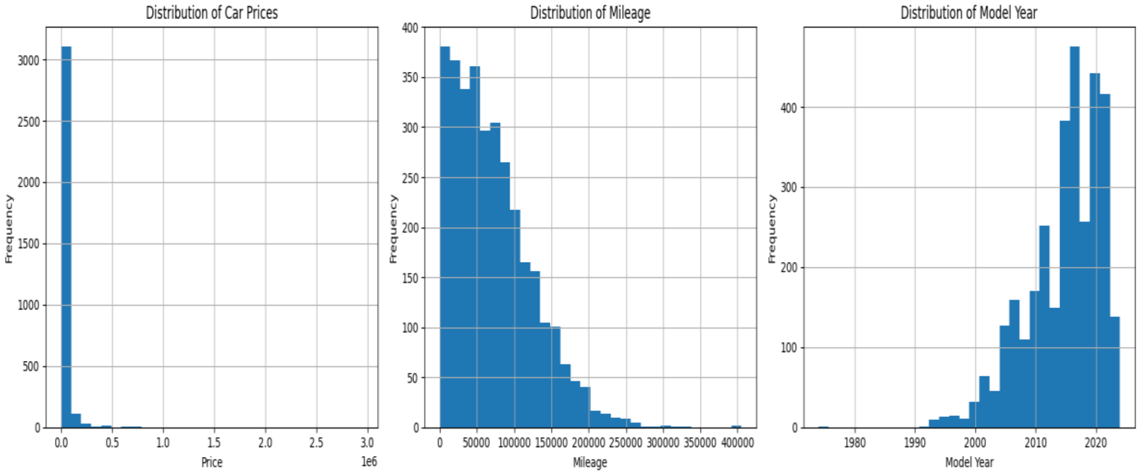
In the EDA, I explore to understand the data and began by examining the structure of the dataset to check data types and missing values. I discovered that most columns were categorical, while only model_year was numeric. Derived features like car age from model_year were calculated using non-sensitive attributes. Some important columns like milage and price were stored as strings and needed cleaning due to non-numeric characters like “mi.” and “$”.

I did the statistical overview and noticed a wide range in model_year from 1974 to 2024, suggesting potential outliers. I also found missing values in fuel_type, accident and clean_title which was missing in over 500 records. I observed that certain values like “Black” dominated in both ext_col and int_col, and “Gasoline” was the most common fuel type. I also preview the data and identify formatting issues.

This initial EDA helped me understand the data quality and distribution, and it guided the preprocessing steps, such as converting strings to numeric types, handling missing values, and encoding categorical variables for machine learning.

The following plots are particularly insightful as they effectively highlight key findings that I have identified, which are directly relevant to my research objectives and address the research questions for used car price.

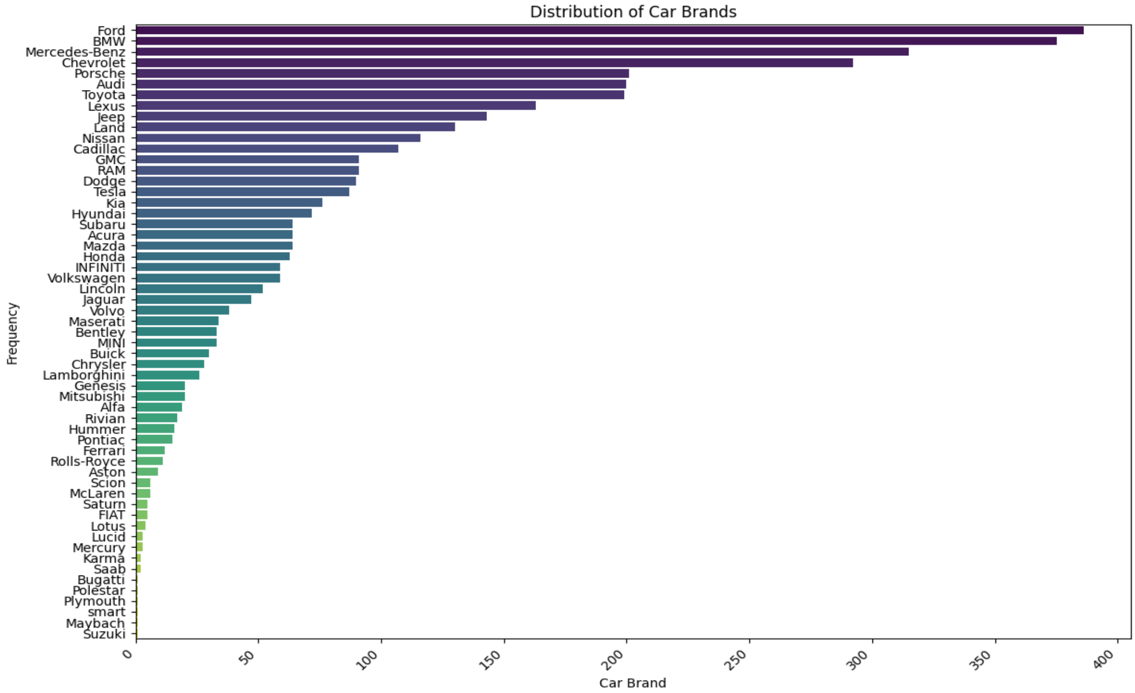
Figure 1: Histogram Plots for Distribution of Prices, Mileage and Model Year



The plots show the following:

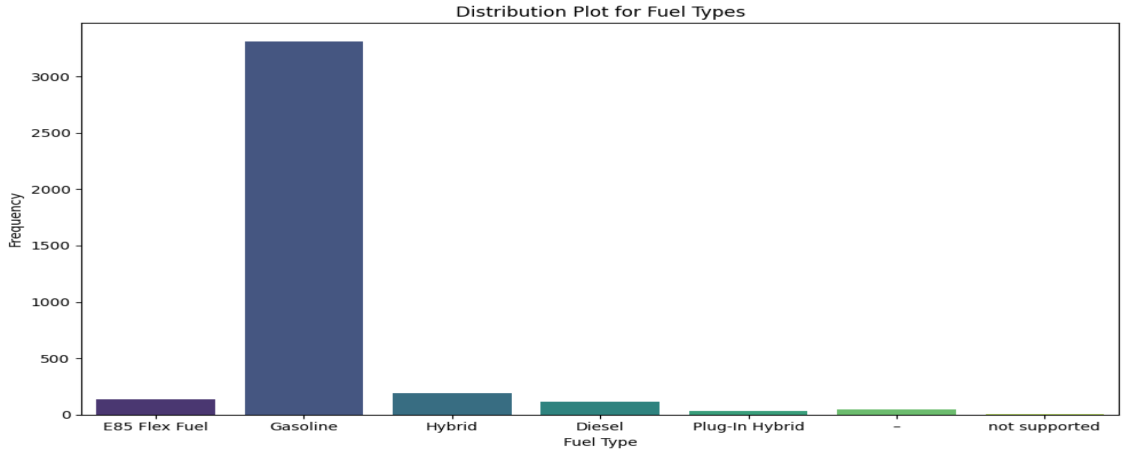
i. **Car Prices**: Most cars are priced below $50,000, with a few expensive ones stretching up to high prices. This suggests most cars are affordable, but there are some luxury or rare cars in the dataset.
ii. **Mileage**: Most cars have a mileage between 50,000 and 150,000 miles. There are two common mileage ranges, and while some cars have very high mileage (up to 400,000 miles), they are much less common.
iii. **Model Year**: The dataset mostly includes newer cars, especially those made after 2015. Older cars (before 2010) are much less common.

Figure 2: Count Plot for Car brand distribution



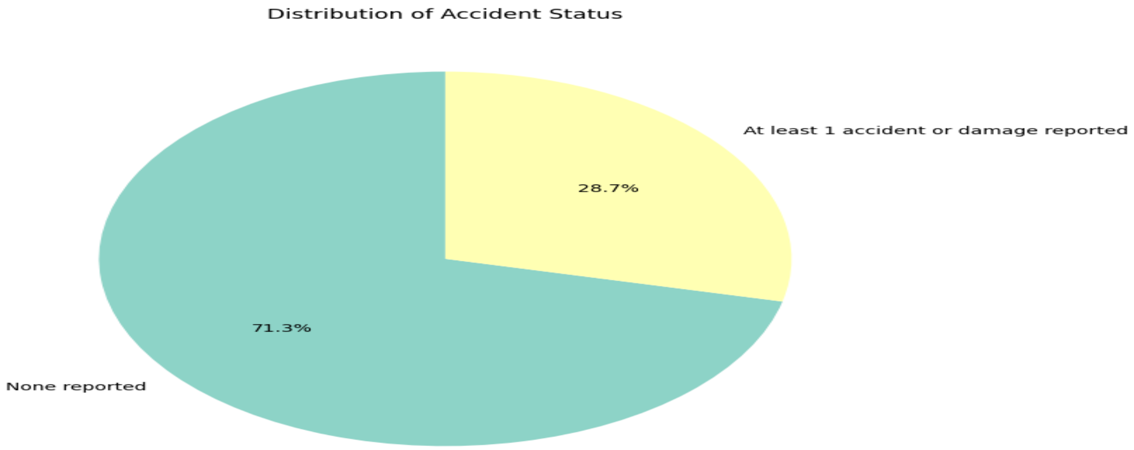
The plot shows that a few car brands like Ford**,** BMW, Mercedes-Benz and Chevrolet dominate the dataset, with over 300 listings each. Other brands like McLaren**,** Rolls-Royce, Bugatti and Suzuki appear much less frequently, with fewer than 50 listings. This shows that the dataset is skewed**,** with most cars coming from a small number of popular brands.

Figure 3: Count Plot for distribution of Fuel Types



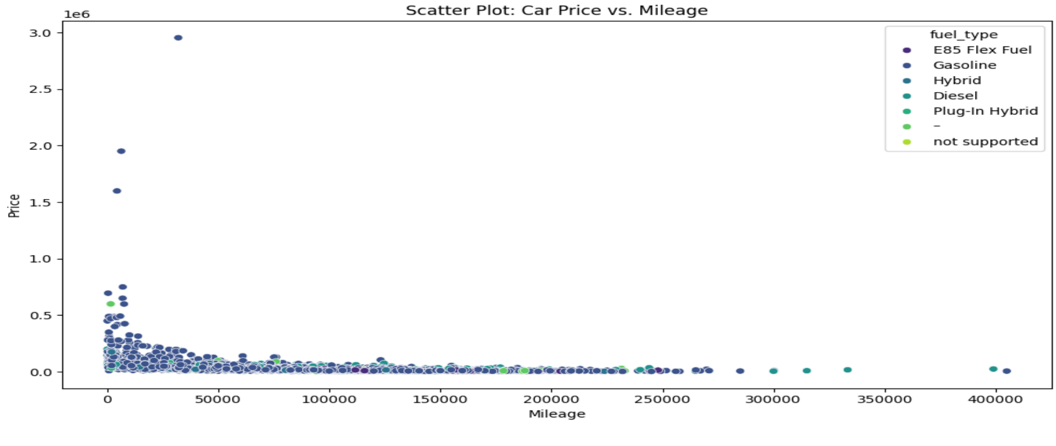
The plot shows that gasoline is the most common fuel type, with over 3,000 cars using it. Other fuel types, like E85 Flex Fuel**,** Hybrid**,** Diesel**,** and Plug-In Hybrid, are much less common, with very few listings. There are also some cars with "not supported" fuel type data.

Figure 4: Pie plot showing distribution of Accident Status



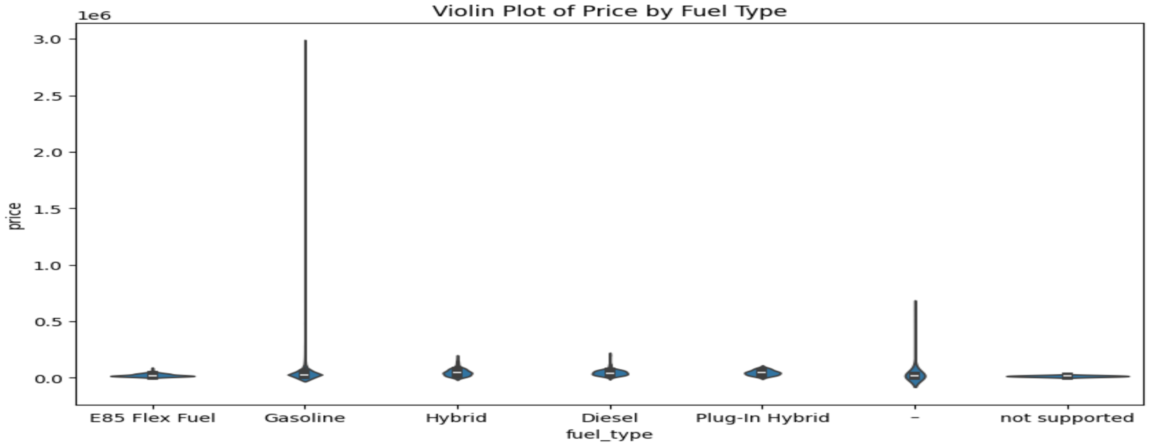
The chart illustrates that 71.3% of cars have no reported accidents or damage, making this the dominant group in the dataset. While 28.7% of cars have at least one accident or damage reported, indicating that a significant portion of the cars have been involved in some form of incident.

Figure 5: Scatter Plot for Car Vs. Mileage



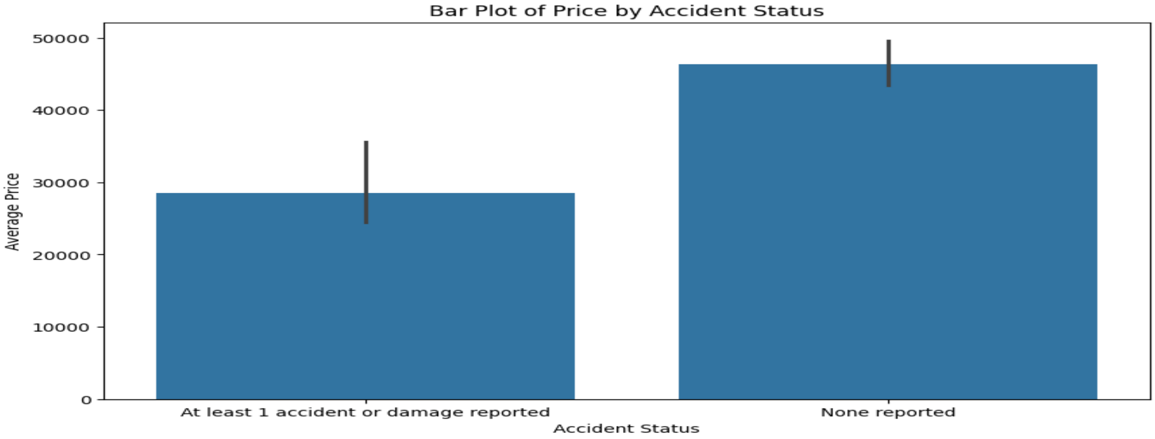
The plot shows the relationship between car price and mileage, most cars with low mileage have lower prices, and as mileage increases**,** prices decrease. However, there are some high-priced cars with high mileage, likely represent luxury or rare cars. The relationship holds across all fuel types, with Gasoline cars being the most common.

Figure 6: Violin plot showing price across different fuel types



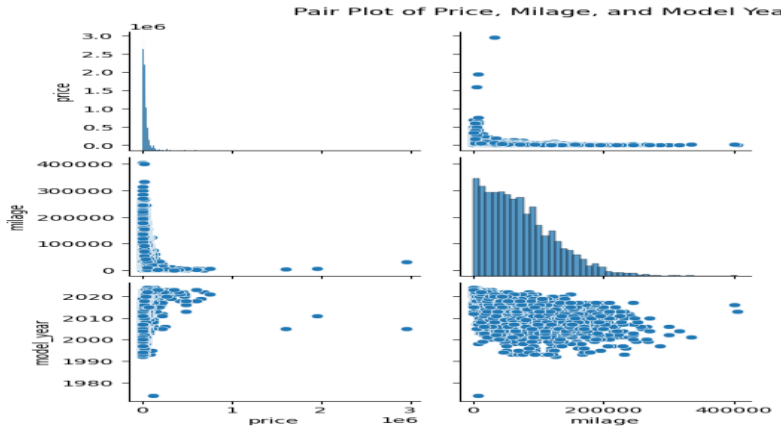
The plot shows that Gasoline cars have a wide range of prices, including some very expensive ones. Cars with other fuel types, like Hybrid and Diesel, tend to be cheaper and have less price variation. The "Not supported" fuel type has very few cars with no price variation.

Figure 6: Bar plot to see how Accident impact car prices



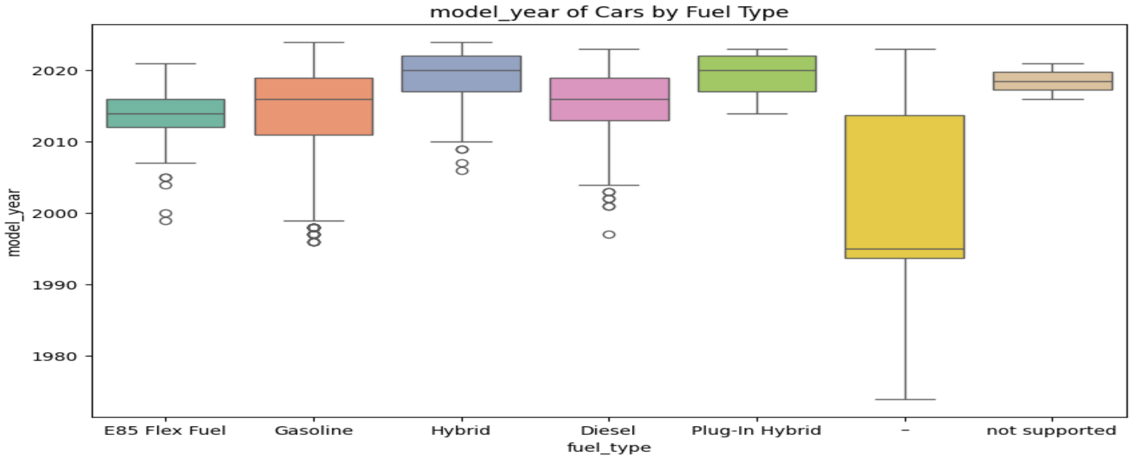
The plot shows that cars without accident history have a higher average price around $40,000, while cars with accident record have a lower average price of about $30,000. The difference in prices indicates the impact of accidents on car value.

Figure 7: Pair plot visualising the relationships between all numerical variables



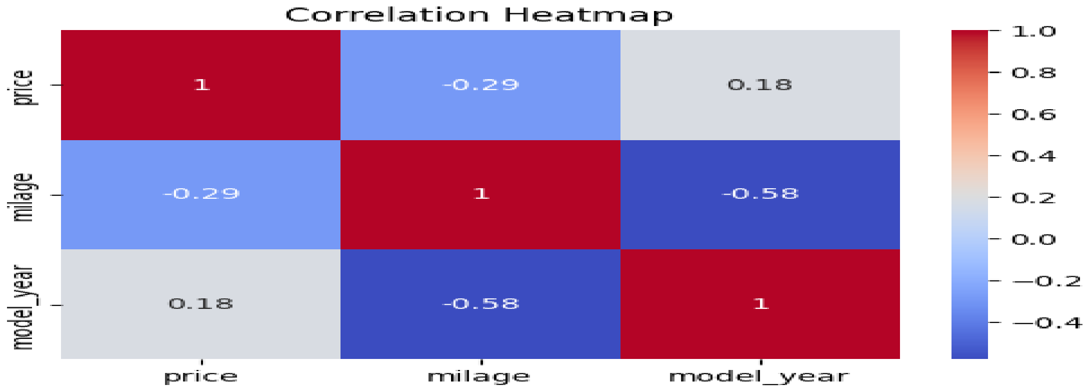
The Pair plot represents the following: Price vs. Mileage - There is a negative correlation between price and mileage, since cars with higher mileage generally have lower prices, as seen in the scattered points and histograms. Price vs. Model Year - There is a positive relationship between price and model year, with newer cars having higher prices. Older cars with lower prices. Mileage vs. Model Year - There seems to be no clear relationship between mileage and model year. Both older and newer cars show a wide range of mileage, with no significant pattern.

Figure 8: Box plot indicating distribution of Model Year by Fuel Type distribution



From the plot, it shows that Gasoline (petrol) cars generally have model years around 2010 to 2020, with a wide range of years represented. Hybrid and Diesel cars mostly fall between 2000 and 2020, with Hybrid (cars with both Gasoline and electric fuel) cars mostly newer models. E85 Flex Fuel (Ethanol fuel) and Plug-In Hybrid (electric) cars are mostly newer, with model years closer to 2020**.** The "Not Supported" category contains cars of various ages, from old to new. This variation likely occurs because some cars have missing or unclear information about their fuel type, making it hard to classify them properly.

Figure 9: Correlation Heatmap showing Relationships between price, mileage and model year



The plot shows that Price and Mileage have a weak negative correlation (-0.29) between price and mileage, meaning that as mileage increases, the price tends to decrease. Price and Model Year have a weak positive correlation (0.18) between price and model year**,** suggesting that newer cars tend to be priced higher. Mileage and Model Year have a strong negative correlation (-0.58) between mileage and model year, meaning that older cars generally have higher mileage

3.4 Data Preprocessing

I performed the following preprocessing pipelines, to ensure quality and usability. The following steps were taken:

- **Cleaning:** Price values were cleaned by removing dollar signs and commas before converting to float. Mileage values were stripped of non-numeric characters ("mi.") and converted to integers.

- **Handling Missing Data:** I dropped rows with missing values in the fuel_type, accident and clean_title columns

- **Feature Engineering, Encoding and Scaling:** I formed new features such as car age from model_year. I encoded categorical variables such as brand, model, transmission, fuel_type, and accident using one-hot encoding. I applied Standard Scaler to normalise the numerical features, such as milage and car_age.

3.5 Model Selection

In this study, I employed a comparative analysis of three supervised regression models, Each model was trained on 80% of the dataset and evaluated on the remaining 20% test set. The following explains the reasons why the models were selected.

i. **Linear Regression** was chosen as a baseline model because of its simplicity and ease of interpretation. It assumes a straight-line relationship between the input features and the target variable (price). While it is fast and useful for understanding feature impact, it may not capture complex patterns in the data.

ii. **Random Forest Regressor** was introduced to model non-linear relationships and capture interactions between variables. It works by building multiple decision trees and averaging their results, which makes it more robust to noise and overfitting. It also provides insights into feature importance, helping to understand which variables influence car prices the most.

iii. **XGBoost Regressor** was selected as a high-performance model for the final stage of modeling. It improves predictions by building trees sequentially and correcting previous errors using gradient boosting. It also incorporates regularization, which helps prevent overfitting and improves generalization on unseen data. XGBoost is widely known for its accuracy in predictive modeling tasks.

3.6 Evaluation Metrics

The model performance was assessed using the following metrics:

1. Mean Absolute Error (MAE)

What it measures:
MAE calculates the average absolute difference between predicted values and actual values. It tells you, on average, how much your model's predictions deviate from the true values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |yi - \widehat{yi}\ |$$

Where:

- yi = actual value
- $\hat{y}i$ = predicted value
- n = number of observations

MAE is easy to interpret and gives a straight forward idea of how far off predictions are. It treats all errors equally without giving extra weight to large errors.

2. Root Mean Squared Error (RMSE)

What it measures**:**
RMSE calculates the square root of the average of squared errors. Unlike MAE, it penalizes larger errors more heavily, making it sensitive to outliers.

RMSE = $\sqrt{\frac{1}{n}} \sum_{i=1}^{n}(yi - \hat{y})^2$

RMSE is useful when large prediction errors are especially undesirable. It gives more weight to bigger mistakes and is often used when performance needs to be more cautious or precise.

3. R-squared (R²)

What it measures**:**
R², or the coefficient of determination, shows the proportion of variance in the target variable that is explained by the model.

$$R^2 = 1 - \frac{\sum(yi - \hat{y}\ i)^2}{\sum(yi - \bar{y}\ )^2}$$

Where $\bar{y}$ is the mean of the actual values.

Why it's useful:
R² gives an overall sense of model performance. A value of:

- **1.0** means perfect prediction,
- **0.0** means the model does no better than the average,
- **Negative** values mean the model performs worse than a basic mean-based model.

Summary:

Metric	**Measures**	**Key Insight**	
MAE	Average size of errors	Simple and easy to interpret	
RMSE	Penalized average error	Emphasizes larger mistakes	
R²	Variance explained	Overall model fit quality	

These three metrics together provide a balanced view of model accuracy, error spread, and predictive strength.

4.0 Analysis of Results

4.1 Model Evaluation – Before Optimization

The initial models were trained and tested using default or basic hyperparameter settings. The performance metrics are summarized as showed in the table below:

Table 1.

Model	**MAE**	**RMSE**	**R²**
Linear Regression	19873.57	117589.40	0.0558
Random Forest	18118.52	117077.57	0.0639
XGBoost	17493.77	117088.18	0.0638

Interpretation of Results

i. **Mean Absolute Error (MAE):** The high MAE values show that the predictions for all three models were off by a considerable amount, with differences between predicted and actual prices being approximately $17,000 to $19,000.

ii. **Root Mean Squared Error (RMSE):** The RMSE values are quite high, which further indicates that the models were not effectively capturing the patterns in the data, particularly due to large outliers that skew the predictions.

iii. **R² Scores:** The low R² values (below 0.1 for all models) indicate poor model performance, suggesting that the models only explained a small portion of the variance in car prices. This reflects underfitting, where the model is too simplistic to capture the complexity of the data.

The models are underfitting, and significant improvements are needed through optimisation.

4.2 Model Evaluation – After Optimisation

I optimised the models to improve performance by adjusting parameters and hyperparameters to better align with the data. After applying hyperparameter tuning, performance significantly improved**.** The table below summarizes the results, highlighting the enhanced model accuracy.

Model	MAE	RMSE	R²
Linear Regression	0.3223	0.4251	0.7202
Best Random Forest	0.3319	0.4175	0.7301
Best XGBoost	0.3148	0.4014	0.7505

Interpretation of Results

i. **MAE:** The significant reduction in MAE for all models indicates that predictions are now much closer to the actual car prices. XGBoost outperformed the others, with the smallest MAE value of 0.3148.

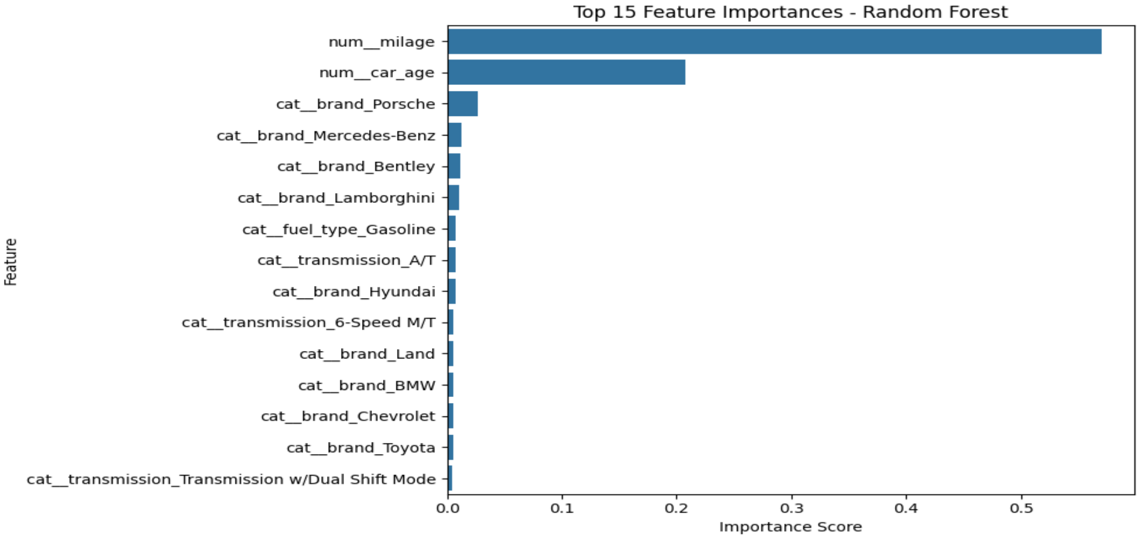
ii. **RMSE:** The lower RMSE values show that, while there are still some larger errors, the models are now producing results that are more consistent with the actual prices. XGBoost again performed the best with an RMSE of 0.4014.

iii. **R²:** The R² score improvement is striking. XGBoost now explains 75% of the variance in car prices, meaning the model captures a significant portion of the factors influencing car prices. Random Forest follows closely with 73%.

Conclusively, The hyperparameter optimization has led to highly improved models, particularly XGBoost, which performed the best across all metrics.

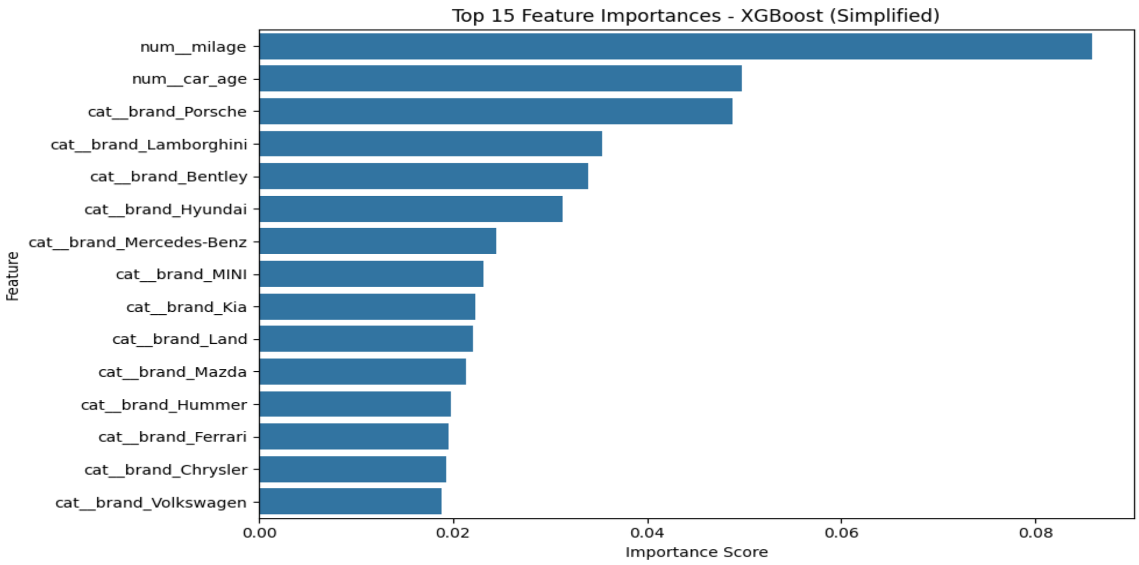
4.3 Feature Importance Analysis.

Figure 10: Bar plot showing top Features influencing car price - by Random Forest model



The plot by the Random Forest Model identified mileage and car age as the most important factors in predicting car prices. Car brands like Porsche and Mercedes-Benz, etc. also play a big role along with features like fuel type and transmission.

Figure 11: Bar plot showing most important Features influencing car price - by XGBoost model



The plot by the XGBoost model revealed that mileage and car age are the most important factors in predicting car prices. Car brands like Porsche**,** Lamborghini, and Mercedes-Benz also have a big impact on price.

5.0 Analysis and Discussion

5.1 Discussion

What do the Results mean: Initially, the models underperformed with high MAE values, indicating significant prediction errors. After optimization, XGBoost showed a substantial improvement, with the lowest MAE and highest R^2, indicating better accuracy in predicting car prices.

Best Model and why: XGBoost outperformed the others due to its use of gradient boosting and regularization, which helped refine predictions and prevent overfitting, making it the most effective model for this dataset. XGBoost's ability to handle non-linear relationships between features, like mileage and car age, made it well-suited for the data. Simpler models like Linear Regression struggled with these complexities, resulting in poor performance.

5.2 **Comparison with Literature**: The results align with the literature, as previous studies such as (Gao, 2024; Guo and Zhang, 2024) have shown that models like Random Forest and XGBoost outperform traditional methods like Multiple Linear Regression in predicting car prices.. The improved performance is likely due to careful data preprocessing and optimization techniques.

5.3 **Limitations**: A key limitation is the presence of missing values and inconsistencies in the dataset, which could affect the model's accuracy. Additionally, the models were trained on a dataset from a specific geographic region, which may not fully reflect the dynamics of used car markets in regions like Nigeria or sub-Saharan Africa.

5.4 **Relation to Objectives**: The results address the project's main goal of improving car price prediction accuracy through machine learning, demonstrating the effectiveness of the models after optimization.

5.5 Relation to research question: The research questions have been answered. The study has successfully confirms that machine learning models, particularly XGBoost, can predict car prices more accurately than traditional methods.

5.6 **Real-World Application**: XGBoost's accuracy and ability to handle multiple input features make it suitable for deployment in real-world systems like mobile apps or dealership platforms, providing more transparent and reliable car price predictions for both buyers and sellers.

5.7 Conclusion

This study demonstrated that XGBoost significantly outperformed traditional methods like Linear Regression in predicting used car prices. After optimization, XGBoost achieved the lowest MAE, RMSE, and the highest R^2, showing its strong predictive power. Key factors such as mileage, car age, and brand were identified as crucial features influencing car prices.

The findings confirm that machine learning models, particularly XGBoost, offer more accurate and reliable price predictions than conventional methods. Hyperparameter tuning proved essential in improving the model's performance, validating the importance of model optimization.

Future Work: Future work could focus on expanding the dataset to include more diverse geographic regions and additional car features. Further model optimization, such as exploring other advanced algorithms or incorporating real-time data, could also improve performance. Additionally, developing user-friendly tools for real-time price prediction would enhance practical application in the automotive industry, offering immediate value to both buyers and sellers.

6.0 References

Bhatt, N.S., Pandey, T.N., Reddy, S.R., B. Jayasurya, Dash, B.B. and Patra, S.S. (2023). An Emperical Analysis of Machine Learning Algorithms for Used Car Price Prediction System. [online] pp.1–5. doi:https://doi.org/10.1109/gcitc60406.2023.10426270.

Deepak, N.A., Kumar, R., Gupta, T., Shubham Gaurav, Yadav, P.S. and B Pranesh (2023). Automobile Valuation Prediction Using Machine Learning based Algorithms. pp.1–5. doi:https://doi.org/10.1109/ic-rvitm60032.2023.10435103.

Gao, J. (2024). Second-hand car price prediction based on multiple linear regression and random forest. *Theoretical and Natural Science*, 52(1), pp.31–40. doi:https://doi.org/10.54254/2753-8818/52/2024ch0105.

Guo, S. and Zhang, B. (2024). Revolutionizing the used car market: Predicting prices with XGBoost. *Applied and Computational Engineering*, [online] 48(1), pp.173–180. doi:https://doi.org/10.54254/2755-2721/48/20241349.

Idris, N.O., Achban, A., Utiarahman, S.A., Karim, J. and Pontoiyo, F. (2020). Predicting the Selling Price of Cars Using Business Intelligence with the Feed-forward Backpropagation Algorithms. *2020 Fifth International Conference on Informatics and Computing (ICIC)*. doi:https://doi.org/10.1109/icic50835.2020.9288594.

Kalpana, G., Durga, Dr.A.K., Reddy, A. and Karuna, Dr.G. (2022). Predicting the Price of Pre-Owned Cars Using Machine Learning and Data Science. *International Journal for Research in Applied Science and Engineering Technology*, 10(7), pp.1468–1476. doi:https://doi.org/10.22214/ijraset.2022.45469.

Kang, J.I., Parekh, H., Ramdas, P., Lee, S. and Woo, J. (2022). Comparing Regression Models Predicting the Price of Used Cars in Big Data. *2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. [online] doi:https://doi.org/10.1109/icce-asia57006.2022.9954633.

Najib, T. (2023). *Used Car Price Prediction Dataset*. [online] Kaggle.com. Available at: https://www.kaggle.com/datasets/taeefnajib/used-car-price-prediction-dataset?resource=download [Accessed 5 May 2025].

Narayana, C.V., Likhitha, C.L., Bademiya, S. and Kusumanjali, K. (2021). Machine Learning Techniques To Predict The Price Of Used Cars: Predictive Analytics in Retail Business. *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*. doi:https://doi.org/10.1109/icesc51422.2021.9532845.

Pal, N., Arora, P., Kohli, P., Sundararaman, D. and Palakurthy, S.S. (2018). How Much Is My Car Worth? A Methodology for Predicting Used Cars' Prices Using Random Forest. *Advances in Intelligent Systems and Computing*, pp.413–422. doi:https://doi.org/10.1007/978-3-030-03402-3_28.

Shanti, N., Assi, A., Shakhshir, H. and Salman, A. (2021). Machine Learning-Powered Mobile App for Predicting Used Car Prices. *Proceedings of the 2021 3rd International Conference on Big-data Service and Intelligent Computation*. doi:https://doi.org/10.1145/3502300.3502307.

Yadav, A., Kumar, E. and Yadav, P.K. (2021). Object detection and used car price predicting analysis system (UCPAS) using machine learning technique. *Linguistics and Culture Review*, 5(S2), pp.1131–1147. doi:https://doi.org/10.21744/lingcure.v5ns2.1660.

Zhu, Y. (2023). Prediction of the price of used cars based on machine learning algorithms. *Applied and computational engineering*, 6(1), pp.671–677. doi:https://doi.org/10.54254/2755-2721/6/20230917.

7.0 **Appendix**

```
# -*- coding: utf-8 -*-
"""used_car_price_prediction_12_6_25.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1c3TzJCnuCPHqMAo2QAUv-
S0ZLBhlTGfz

# **Importing Necessary Libraries**
"""

# Require libraries

import pandas as pd
import numpy as np

"""# **Reading the Dataset**"""

# Load dataset
cp = pd.read_csv('/content/used_cars.csv')

"""## **Displaying the first 5 Rows**"""

cp.head()

"""# **Explore to Understand the Dataset**"""

# Basic information about the data

cp.info() # it appears there are issues with the data format and missing values.
# As price, which should be in a numerical format, is currently represented as
categorical data.

# Display descriptive statistics of numerical columns
print("\nDescriptive Statistics of the dataset:")
print(cp.describe())

# Check for missing values

cp.isnull().sum() # missing data was confirmed with fuel type, accident and clean title

# check for duplicate
cp.duplicated().sum() # no duplicate values

"""# **Cleaning of Data**"""
```

```
# Data Cleaning

# 1. Remove symbols and convert 'price' to numeric
cp['price'] = cp['price'].replace('[\$,]', '', regex=True).astype(float)

# 2. Clean 'milage' column: remove ' mi.' and commas, then convert to numeric
cp['milage'] = cp['milage'].str.replace(' mi.', '', regex=False).str.replace(',', '')
cp['milage'] = pd.to_numeric(cp['milage'], errors='coerce')

# 3. Handle missing values by dropping rows with missing target or essential feature
columns
cp.dropna(subset=['price', 'milage', 'fuel_type', 'accident', 'clean_title'], inplace=True)

# 4. Remove duplicate rows if any
cp.drop_duplicates(inplace=True)

# 5. Confirm updated data types and missing values
clean_info = cp.info()
missing_values = cp.isnull().sum()

missing_values

# Summary statistics for milage, price and model_year

print('\nSummary Statistics:\n', cp[['milage', 'price', 'model_year']].describe())

"""# **Exploratory Data Analysis (EDA)**"""

import matplotlib.pyplot as plt
import seaborn as sns

# Univariate Analysis - Histograms

fig, axes = plt.subplots(1, 3, figsize=(18, 5))
cp['price'].hist(ax=axes[0], bins=30)
axes[0].set_title("Distribution of Car Prices")
axes[0].set_xlabel("Price")
axes[0].set_ylabel("Frequency")

cp['milage'].hist(ax=axes[1], bins=30)
axes[1].set_title("Distribution of Mileage")
axes[1].set_xlabel("Mileage")
axes[1].set_ylabel("Frequency")

cp['model_year'].hist(ax=axes[2], bins=30)
axes[2].set_title("Distribution of Model Year")
axes[2].set_xlabel("Model Year")
axes[2].set_ylabel("Frequency")

plt.tight_layout()
plt.show()

# Count plot for car brand distribution
import seaborn as sns
import matplotlib.pyplot as plt

# Plot for Car Brand
plt.figure(figsize=(12, 8))
sns.countplot(y='brand', data=cp, order=cp['brand'].value_counts().index,
palette="viridis")
plt.title("Distribution of Car Brands")
plt.xlabel("Car Brand")
plt.ylabel("Frequency")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# Count plot for fuel types

plt.figure(figsize=(10, 6))
sns.countplot(x='fuel_type', data=cp, palette="viridis")
plt.title("Distribution Plot for Fuel Types")
plt.xlabel("Fuel Type")
plt.ylabel("Frequency")
plt.tight_layout()
plt.show()

# Pie plot to show the distribution of accident status
plt.figure(figsize=(8, 8))
accident_counts = cp['accident'].value_counts()
accident_counts.plot.pie(autopct='%1.1f%%', startangle=90,
colors=sns.color_palette("Set3", len(accident_counts)))
plt.title('Distribution of Accident Status')
plt.ylabel('')
plt.show()

# Bivariate Analysis

# Scatter plot

plt.figure(figsize=(10, 6))
sns.scatterplot(x='milage', y='price', data=cp, hue='fuel_type', palette="viridis")
plt.title("Scatter Plot: Car Price vs. Mileage")
plt.xlabel("Mileage")
plt.ylabel("Price")
plt.tight_layout()
plt.show()

# Violin plot
# to visualize the distribution of 'price' across different 'fuel_type'

plt.figure(figsize=(10, 6))
sns.violinplot(x='fuel_type', y='price', data=cp)
plt.title('Violin Plot of Price by Fuel Type')
plt.show()

# Bar plot of Price by Accident Status to see how accidents impact car prices
plt.figure(figsize=(10, 6))
sns.barplot(x='accident', y='price', data=cp)
plt.title('Bar Plot of Price by Accident Status')
plt.ylabel('Average Price')
plt.xlabel('Accident Status')
plt.show()

# Pair plot

# Pair plot to visualize relationships between all numerical variables
sns.pairplot(cp[['price', 'milage', 'model_year']])
plt.suptitle('Pair Plot of Price, Milage, and Model Year', y=1.02)
plt.show()

# Box Plot Model_year by Fuel Type distribution
plt.figure(figsize=(10, 6))
sns.boxplot(x='fuel_type', y='model_year', data=cp, palette='Set2')
plt.title('model_year of Cars by Fuel Type')
plt.show()

# Multivariate Analysis

# Correlation Heatmap

# Correlation Matrix (numerical features)
corr_matrix = cp[['price', 'milage', 'model_year']].corr()
plt.figure(figsize=(6, 4))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()

"""# **Preprocessing of Data**"""

from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
import pandas as pd
import numpy as np


# Clean 'price' and 'milage'
cp['price'] = cp['price'].replace('[\$,]', '', regex=True).astype(float)
```

```
cp['milage'] = cp['milage'].astype(str).str.replace(' mi.', '', regex=False).str.replace(',',
'')
cp['milage'] = pd.to_numeric(cp['milage'], errors='coerce')

# Drop rows with missing essential values
cp.dropna(subset=['price', 'milage', 'fuel_type', 'accident', 'clean_title'], inplace=True)
cp.drop_duplicates(inplace=True)

# Feature Engineering
cp['car_age'] = 2025 - cp['model_year']
categorical_cols = ['fuel_type', 'transmission', 'brand']
numerical_cols = ['milage', 'car_age']
y = cp['price']
X = cp[categorical_cols + numerical_cols]

# Use OneHotEncoder without sparse argument
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_cols),
        ('cat', OneHotEncoder(drop='first'), categorical_cols)
    ])

# Fit and transform
X_processed = preprocessor.fit_transform(X)
X_processed = X_processed.toarray() if hasattr(X_processed, "toarray") else
X_processed

# Get feature names
encoded_feature_names =
preprocessor.named_transformers_['cat'].get_feature_names_out(categorical_cols)
all_feature_names = np.concatenate([numerical_cols, encoded_feature_names])

# Display the first 5 rows
X_processed_df = pd.DataFrame(X_processed, columns=all_feature_names)
print(X_processed_df.head())

"""# **Data Splitting (Ratio of 80:20)**"""

from sklearn.model_selection import train_test_split

# Data Splitting

# Split into training and testing sets (80/20)
X_train, X_test, y_train, y_test = train_test_split(
    X_processed, y, test_size=0.2, random_state=42)

(X_train.shape, X_test.shape, y_train.shape, y_test.shape)

"""# **Model Building, Training and Evaluation**"""
```

```
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import pandas as pd
import numpy as np

# Model Building

# Initialize models
lr_model = LinearRegression()
rf_model = RandomForestRegressor(n_estimators=100, max_depth=10,
random_state=42)
xgb_model = XGBRegressor(n_estimators=100, max_depth=6, learning_rate=0.1,
random_state=42, verbosity=0)


# Model Training

# Fit models
lr_model.fit(X_train, y_train)
rf_model.fit(X_train, y_train)
xgb_model.fit(X_train, y_train)


# Model Evaluation ( on test data)

# Predict
lr_pred = lr_model.predict(X_test)
rf_pred = rf_model.predict(X_test)
xgb_pred = xgb_model.predict(X_test)

# Combine and display results
results_df = pd.DataFrame(
    [lr_scores, rf_scores, xgb_scores],
    index=['Linear Regression', 'Random Forest', 'XGBoost']
)

# Display results
print("Model Evaluation Results")
print(results_df)

"""# **Model Optimisation - Hyperparameter Tuning using GridSearchCV for both RF
and XGBoost**"""

# Define parameter grid for Random Forest
rf_param_grid = {
    'n_estimators': [100, 150, 200],
    'max_depth': [10, 20, None],
    'min_samples_split': [2, 5, 10]
}

# Define parameter grid for XGBoost
xgb_param_grid = {
    'n_estimators': [50, 100],
    'max_depth': [4, 6],
    'learning_rate': [0.05, 0.1]
}

# Run GridSearchCV for Random Forest
rf_grid_search = GridSearchCV(RandomForestRegressor(random_state=42),
rf_param_grid, cv=3, n_jobs=-1, scoring='neg_mean_squared_error')
rf_grid_search.fit(X_train, y_train)
rf_best = rf_grid_search.best_estimator_
rf_best_params = rf_grid_search.best_params_

# Run GridSearchCV for XGBoost with reduced complexity
xgb_grid_search = GridSearchCV(XGBRegressor(random_state=42, verbosity=0),
xgb_param_grid, cv=3, n_jobs=-1, scoring='neg_mean_squared_error')
xgb_grid_search.fit(X_train, y_train)
xgb_best = xgb_grid_search.best_estimator_
xgb_best_params = xgb_grid_search.best_params_

(rf_best_params, xgb_best_params)

from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np
import pandas as pd

# Define and fit the best Random Forest model
best_rf_model = RandomForestRegressor(
    n_estimators=200,
    max_depth=None,
    min_samples_split=10,
    random_state=42
)
best_rf_model.fit(X_train, y_train)

# Define and fit the best XGBoost model
best_xgb_model = XGBRegressor(
    learning_rate=0.1,
    max_depth=6,
    n_estimators=100,
    random_state=42,
    verbosity=0
)
best_xgb_model.fit(X_train, y_train)
```

```
# Define the evaluation function
def evaluate(y_true, y_pred):
    return {
        'MAE': mean_absolute_error(y_true, y_pred),
        'RMSE': np.sqrt(mean_squared_error(y_true, y_pred)),
        'R2': r2_score(y_true, y_pred)
    }

# Retrain Linear Regression
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

# Predict
lr_pred = lr_model.predict(X_test)
rf_pred = best_rf_model.predict(X_test)
xgb_pred = best_xgb_model.predict(X_test)

# Evaluate
lr_scores = evaluate(y_test, lr_pred)
rf_scores = evaluate(y_test, rf_pred)
xgb_scores = evaluate(y_test, xgb_pred)

# Combine results
results_df = pd.DataFrame(
    [lr_scores, rf_scores, xgb_scores],
    index=['Linear Regression', 'Best Random Forest', 'Best XGBoost']
)

# Print results
print("Model Evaluation Results After Optimization:\n")
print(results_df)

"""# **Features that most influence car price**"""

# Get feature names from the preprocessor
feature_names = preprocessor.get_feature_names_out()

# Feature Importances
rf_importances = pd.Series(best_rf_model.feature_importances_,
index=feature_names).sort_values(ascending=False).head(15)
xgb_importances = pd.Series(best_xgb_model.feature_importances_,
index=feature_names).sort_values(ascending=False).head(15)

# Plot RF Importance
plt.figure(figsize=(10, 6))
sns.barplot(x=rf_importances.values, y=rf_importances.index)
plt.title("Top 15 Feature Importances - Random Forest")
plt.xlabel("Importance Score")
plt.ylabel("Feature")
plt.tight_layout()
plt.show()

# Plot XGB Importance
plt.figure(figsize=(10, 6))
sns.barplot(x=xgb_importances.values, y=xgb_importances.index)
plt.title("Top 15 Feature Importances - XGBoost (Simplified)")
plt.xlabel("Importance Score")
plt.ylabel("Feature")
plt.tight_layout()
plt.show()
```