Optimizing Car Purchases with Extreme Gradient Boosting: A Machine Learning Approach to Affordable Dream Cars

Mohamad Yazan Adi

2016544

Project Dissertation

Swansea University
Prifysgol Abertawe

Department of Computer Science

8th February 2022

Declaration

Statement 1

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed .... Yazan Adi (2016544)

Date 08/02/2023..................................... Yazan Adi (2016544)

Statement 2

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by citations giving explicit references. A bibliography is appended.

Signed .... Yazan Adi (2016544)

Date 08/02/2023..................................... Yazan Adi (2016544)

Statement 3

The University's ethical procedures have been followed and, where appropriate, ethical approval has been granted.

Signed .... Yazan Adi (2016544)

Date 08/02/2023..................................... Yazan Adi (2016544)

Abstract

Buying and selling used cars is a prevalent practice worldwide, with sellers and buyers relying on market trends and advertisements to determine car prices. Despite the existence of blue books offering pricing estimates, real market prices vary due to demand and supply dynamics. This study aims to automate this process by leveraging advanced machine learning techniques and an up-to-date dataset collected from an active commercial website. We developed semi-automated rule-based scripts for data cleaning and preparation, enabling us to apply the eXtreme Gradient Boosting (XGBoost) algorithm for predicting car prices.

Our methodology included an extensive exploratory data analysis, data visualization, correlation analysis, and feature engineering to identify the most influential factors affecting car prices, such as "Horsepower," "Cylinder Volume," "Car Body Type," and "Transmission." We employed K-Fold cross-validation to validate the model's performance, achieving a mean R2 score of 0.837. Hyperparameter tuning and optimization further improved the model's predictive accuracy, resulting in an R2 score of 0.828 for the test set.

Our findings reveal the top three features contributing to the model's predictions: Horsepower (0.242), HP_Cyln_Volume (0.177), and Cylinder Volume (0.121), providing valuable insights for automotive market stakeholders. As a practical application of our research, we developed a mobile application that offers an estimated price based on a car's properties, assisting users in determining a suitable price for their vehicles. In conclusion, our study presents a robust and interpretable machine learning model for predicting car prices, streamlining the car buying and selling process for consumers and industry professionals alike.

Table of Contents

1 Introduction

1.1 Motivation

Over the past few years, it is safe to say that we are living in exciting times where the global automotive market has encountered significant shifts, driven by factors such as economic instability, developing consumer preferences, and emerging technologies. The market has diversified as a result of the increased acceptance of electric vehicles and the growing emphasis on sustainability. As a result, having access to trustworthy pricing systems that can quickly adapt to shifting market conditions is crucial for buyers, sellers, and other stakeholders. Simultaneously, governments worldwide are implementing policies to support environmentally friendly transportation [1]. This fosters a favourable environment for data-driven solutions. Despite the availability of car price determining tools, a more precise, efficient, and solid option is called for, especially one that accounts for the shifting consumer tastes, including those who have disabilities. Although there are car price determining tools available, there is a need for a more precise, efficient, and robust option. This is especially important for addressing shifting consumer preferences, including those with disabilities.

New technologies, in recent years, such as self-driving capabilities, advanced driver-assistance systems (ADAS), and connected car services have added an additional layer of complexity to the automotive market [2]. These innovations have a direct influence towards vehicle pricing, as they introduce new features and capabilities that potential buyers may value in different ways. To remain relevant and compete with an ever-growing industry, car pricing tools must account for these technological advancements and their impact on vehicle worth. The economic volatility that has marked recent years, including fluctuations in currency values, trade policies, and market conditions, further emphasizes the need for a resilient and adaptive car pricing tool. Buyers and sellers require precise, up-to-date information to traverse these uncertainties and make well-informed decisions that minimize financial risks. A dynamic, AI-driven pricing tool can offer the necessary flexibility and responsiveness to accommodate market fluctuations, ensuring users have access to dependable and current pricing data.

Furthermore, deviating attentions towards climate change and the environmental effects of conventional internal combustion engine vehicles have resulted in increased demand for electric and hybrid vehicles. This shift in consumer demand calls for an AI-driven, price determining, tool capable of accurately assessing the value of these zero-emission vehicles, taking into account factors such as battery life, charging infrastructure, and government incentives. Additionally, an inclusive and accessible car pricing tool should cater to the requirements of individuals with disabilities, who might need specific adaptations or accommodations to effectively utilize the tool. Guaranteeing that the solution is user-friendly and compatible with assistive technologies like screen readers or voice recognition software is essential to addressing the varied needs of the population and promoting equal access to valuable market information.

1.2 Significance of The Study

An AI-driven car pricing tool has the potential to revolutionize the automotive industry by addressing the existing challenges and providing numerous benefits. By offering precise and personalized pricing predictions, the tool can bridge the gap between sellers and buyers, empowering them to make informed decisions. It can cater to a vast market segment, including dealerships, individual buyers and sellers, and online platforms. Furthermore, the AI-generated pricing data will equip researchers and analysts with a strong knowledge base required for them to study market trends, consumer behaviour, and the impact of various factors on car prices, providing valuable insights to the automotive industry. The pricing data generated can further be exploited to the benefit of insurance companies such as, “Churchill Car Insurance” or “Admiral”. Leading insurance companies like these can utilize the data produced to refine their risk assessment and pricing models leading to more accurate insurance premiums.

1.3 Aims and Objectives

The primary aim of this paper is to develop a supervised machine learning model and test its accuracy in predicting the retail price of cars using a web scraped dataset. The model takes into account specific factors that influence a car’s value. The objectives of this research are as follows:

1- Collect and pre-process a dataset acquired through a manually written python script which web scrapes relevant car details from an automobile website. However, ensure the data is suitable for training and testing the machine learning model.

2- Develop a model that accurately predicts the retail price of cars based on the given independent variables:

   - **Year**: The year in which the car has been registered with the road authority.
   - **Mileage**: The number of miles the car has been driven.
   - **Cylinder Volume**: The volume of each cylinder in the engine.
   - **Horsepower**: The power output an engine produces.
   - **Transmission**: The transmission of the car, either manual or automatic.
   - **Fuel Type**: The type of fuel the car uses, for the dataset used, we have petrol, diesel, electric or hybrid.
   - **Owners**: The number of unique owners the car has been registered with.

3- Evaluate the performance of the developed machine learning model, using suitable metrics, by comparing its predictions with the actual prices in the dataset.

4- Examine the results, pinpoint potential areas of improvement, and explore the implications of the model’s precision for the stakeholders in the automotive sector.

1.4 Methodology Overview

To achieve these objectives, an Extreme Gradient Boosting Machine (XGBM) model will be developed to analyse and exploit large datasets. The XGBM model is well-suited for this project due to its ability to handle large datasets, address the complexities of various features, and provide accurate price predictions [3]. The XGBM model combines the strengths of multiple weak learners to improve the overall performance and reduce potential overfitting, making it ideal for this task. The data for the XGBM model will be obtained through harnessing the power of web scraping from the website pistonheads.com [4], which offers a vast number of listings, making it an ideal source for training the model. Prior to feeding the data into the model, necessary pre-processing steps will be undertaken, such as cleaning the data, handling missing values, and encoding categorical variables. By following these procedures, we ensure the model is fed cleansed data that facilitates its training and testing parameters guaranteeing reliable results of price prediction. In order to keep the AI-driven car pricing tool relevant and up-to-date in the continuously fluctuating market, web scraping will be employed to update the large datasets in real time, ensuring the tool remains a valuable long-term resource for users. The performance of the extreme gradient boosting machine model will be assessed using appropriate evaluation metrics to ensure the accuracy, reliability, and scalability of the AI-driven car pricing tool, guaranteeing its longevity and competitiveness in the ever-growing industry. By incorporating this methodology, the study will address the challenges and gaps in the current car pricing landscape, resulting in a more efficient, accurate, and user-friendly tool that can adapt to the dynamic market conditions and cater to a wide range of users, including those with disabilities.

1.5 Choice of Language

In the realm of machine learning, Python's unrivalled simplicity and readability make it accessible to developers with diverse skill levels [5]. The language's welcoming nature encourages swift prototyping and efficient iteration throughout the development process. When constructing a machine learning model dependent on scraped data from numerous vehicle listings, having a language that permits effortless experimentation and speedy adjustments is crucial. Python's streamlined syntax and semantics allow developers to concentrate on the AI system's logic and structure, creating a setting that fosters innovation and productivity.

Secondly, Python is renowned for having a vast ecosystem of modules that are specifically made for activities including web scraping, data manipulation, and machine learning. Strong functionality for extracting and processing vehicle listing data from websites is provided by libraries like Beautiful Soup, Selenium, and Scrapy. Python's data manipulation and analysis packages, such pandas and NumPy, offer strong capabilities for cleaning, preparing, and converting the data into a machine learning-compatible format once the data has been acquired. Additionally, the Gradient Boosting Machine (GBM) algorithm's implementation is streamlined by Python's diverse machine learning library collection, such as scikit-learn and LightGBM, making it easier to incorporate the model into the AI system.

Moreover, Python's adaptability as a general-purpose language enables the smooth integration of various stages within the AI system's pipeline. This project demands coordination between web scraping, data pre-processing, and machine learning components, and Python excels in managing these diverse tasks within a unified framework. The capacity to utilize a single language throughout all development stages minimizes context-switching and fosters collaboration among team members, ultimately enhancing the development process's efficiency and the AI system's effectiveness.

1.6 Overview

The remainder of this paper outlines the related work, in section 2. In section 3, describes the methodology used for conducting the research. Finally section 4 concludes the findings and achieved results from the study.

2 Related Work

Surprisingly, the application of machine learning algorithms to predict car prices has been found to outperform traditional methods in terms of accuracy and efficiency. With the availability of large datasets and the continuous advancements of machine learning techniques, it is expected that this trend will continue to grow in the coming years. Çelik and UO Osmanoglu presented a study in 2019 titled "*Prediction of The Prices of Second-Hand Cars*" [6]. They used car data collected from an auction website, 5041 second-hand vehicles, to create a model. Çelik and Osmanoglu have ensured the noisy dataset they had initially has been cleansed and pre-processed to guarantee their model produced optimal results. Their primary method was linear regression, and they divided the data into different ratios, using a 70-30% and an 80-20% split for training and testing respectively. Their highest R-square result was 89.1%, which is fairly impressive. To evaluate the model's performance, they utilized the R-square metric, as it appeared to provide a better indication than prediction accuracy since it could incorporate more information about the model. However, their study only considered three factors: price, model, and production year. As a result, there is potential for enhancement by taking into account additional factors such as mileage, horsepower, transmission and petrol type.

In a research titled "*Predicting the Price of Used Cars using Machine Learning Techniques*", Sameer and Pudaruth investigate the application of supervised machine learning techniques to predict the price of used cars in Mauritius [7]. The predictions are based on historical data collected from daily newspapers. The paper explores the use of multiple linear regression analysis, k-nearest neighbours, naïve Bayes, and decision trees for making predictions. The study aims to find the most efficient method for predicting the price of used cars. Since their study conducted cars in Mauritius, they only considered the most popular makes in that area, Nissan and Toyota. Firstly, for multiple linear regression, the lack of the mileage data for the majority of the cars did not allow them to forecast the price, hence it was from the analysis. Furthermore, using Pearson correlation coefficient [8], they concluded that the year of manufacture has the highest correlation with the price of the car. They attempted enhance this relation by removing outliers, and by using the logarithmic value of price instead of the actual values. The results showed that while logarithmic regression provided slightly better results than simple regression, linear regression alone on one variable is not sufficient to predict an accurate price of used cars. The authors also found the regression coefficient to be much higher for Nissan cars (0.917) as compared to Toyota cars (0.803). Secondly, for k-nearest neighbours, Sameer and Pudaruth only consider three parameters (make, year, cylinder volume) and normalise the data to prevent large values from overshadowing the smaller values. To evaluate the performance of kNN, they have split the dataset into Toyota and Nissan cars. Their results showed that the performance was significantly better for Nissan cars than for Toyota cars, implying that prices generated for Nissan cars are more consistent against Toyota cars. The study shows that with cross-validation of 10 folds and k = 1, the mean absolute error (Rs) for Toyota cars is 45,189 whilst Nissan's is 27,258. The best value of k for Toyota is 1, while for Nissan cars, the best value of k is 5, achieving a mean absolute error of only 25,741. In addition, the third machine learning algorithm explored was decision trees. For the J48 algorithm (Java implementation of the C4.5 decision tree algorithm), it identified the year of the manufacture as the most decisive feature, which is consistent with the evidence found earlier using regression analysis. The Random Forest algorithm performed well on the whole training set, but its performance on the testing set was comparable to that of J48. Naïve Bayes, the final explored algorithm, is simple to implement and often yields accuracy comparable to more complex algorithms, such as the one used in this paper (Extreme Gradient Boosting Machine).

The researchers' paper uses the same attributes as that of decision trees and found that Naïve Bayes performed better when using the original data rather than normalising the attributes' values. The results show that Naïve Bayes has comparable performance to decision trees, with slightly better performance when using the original data. All in all, their study suggests that no single technique provides a highly accurate price prediction, and all four algorithms have comparable performance. The authors have recommended more sophisticated algorithms to be investigated in the future for better predictions.

The research paper "*Prediction of Prices Car Price: Prediction with Machine Learning*" (2022) [9] explores the used car market in India, highlighting the growth of online portals that facilitate the exchange of information on used car prices. The authors, Sachin and Damandeep, aim to develop a statistical model using Machine Learning algorithms such as Linear Regression and Random Forest to predict the value of a used car based on a given set of features. The performance of these algorithms is compared to determine the most accurate one. The authors used a dataset from Kaggle.com, which contained data on used cars listed on CarDekho.com, a leading Indian car search portal. The dataset was already cleaned and contained various features such as car name, year, price, current price, kilometres driven, fuel type, seller type, transmission, and number of previous owners. The dataset was divided into a conventional 80/20 split for training and testing. Sachin and Damandeep constructed a Random Forest classifier and evaluated it, resulting in the following performance metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). The results were 2.30, 0.90, and 1.52 for MSE, MAE, and RMSE rounded to two decimal places respectively. The study demonstrates the potential of using machine learning algorithms, particularly Random Forest classifier, for predicting the value of used cars in the Indian market. By providing accurate predictions, this model can help both buyers and sellers make informed decisions regarding used car transactions. The study provides promising results, however, there is room for improvement. Firstly, the authors could explore other machine learning algorithms to determine if better performance can be achieved. Additionally, they could experiment with feature engineering and selection techniques to enhance the predictive power of their model. Lastly, the authors could test their model on different datasets to validate its generalizability across various used car markets.

Published in 2022, a paper titled "*Used Cars Price Prediction in DKI Jakarta Using Extreme Gradient Boosting and Bayesian Optimization Algorithm*" was conducted by three researchers focuses on predicting the price of used cars based numerous features including but not limited to, brand, model, and transmission [10]. They initiate the paper with an introduction to the used car market's growth and the importance of determining the right price for used cars. The authors note that during the pandemic, people preferred used cars over new cars, leading to a 20% increase in sales compared to the previous year. They mention the wide variations in used car prices on different websites, which may result in buyers overpaying for cars. The authors opted to use the XGBoost and Bayesian Optimisation algorithms to complete the task. The article commences with an overview of the growth of the used car market and the significance of ascertaining the appropriate price for second hand vehicles. They further highlight that, amid the covid pandemic, consumers preferred pre-owned vehicles over factory-sealed ones, causing a 20% incline in sales compared to the year prior. The authors also discuss the considerable discrepancies in used car prices across various websites which could lead to purchasers overspending on cars. Furthermore, they have discussed in their methodology their choice of XGBoost and Bayesian Optimisation algorithms. XGBoost was selected as a result of its lowest RMSE on a literature review and its success in data mining competitions. As for Bayesian Optimisation, an automatic hyperparameter tuning algorithm [11], it was chosen to optimize the model, avoid overfitting, and reduce the error rate. In their data preparation stage, the authors address data cleaning, distribution of target variables, handling of outliers, and feature engineering. They handle missing values in the dataset, drop irrelevant columns, and filter data for the DKI Jakarta area. They also perform scaling on the price column using the logarithmic method to reduce the range of values that are too far apart. The results of the study show that the machine learning model is not overfitting, and the model can predict the price of used cars well based on its features. The authors conclude that their model is good enough to predict the price of used cars in DKI Jakarta. The Root Mean Squared Error (RMSE) metric shows that the model is not overfitting, with values of 0.12129 on the validation dataset and 0.13471 on the testing dataset. The R Squared metric indicates that the model can predict the car price well based on its features, with values of 0.98271 on the validation dataset and 0.97870 on the testing dataset, which is remarkable. Regarding the implications of this study, they are significant for potential buyers of pre-owned vehicles, as it delivers a means for a reasonable price prediction of a used car based on its features, preventing them from overpaying. The authors' research also contributes to the emerging field of machine learning and its applications in various domains, particularly the automotive industry. The paper shows promising results overall, however, there are a couple of aspects which can be improved. Firstly, the study could be expanded to help people in other regions or parts of the world in making informed decisions before performing a car transaction. Secondly, additional features such as a car’s maintenance history or number of owners could be included in the model parameters to improve prediction accuracy. Equally important, the authors could potentially investigate other algorithms such as linear regression for bench marking. Lastly, the study could be extended to include real-time data from various sources, such as car dealerships or online marketplaces, to ensure that predictions are up-to-date and accurate.

The research paper "*Competitive Analysis of the Top Gradient Boosting Machine Learning Algorithms*" published in 2022 aims to explore the performance of four popular gradient boosting algorithms - XGBoost, CatBoost, LightGBM, and SnapBoost - by comparing their accuracy and training times across four diverse datasets (numeric, categorical, image, and temporal) [12]. The boosting algorithms were selected due to their effectiveness in addressing various machine learning tasks, offering faster training times and higher accuracy compared to other techniques, while also providing users with the flexibility to choose and fine-tune a wide range of hyperparameters to suit their requirement. The paper discusses two approaches for hyperparameter tuning: manual tuning and automated tuning using Bayesian Hyper-Parameter optimisation (HPO) techniques, such as the open-source library 'HyperOpt' [15] . A key focus of the paper is the comparison of these four algorithms in terms of their accuracy and training time across different datasets with and without HyperOpt. The following observations were made:

1. Numeric Dataset: SnapBoost achieved the highest accuracy of 99% without HPO and 98% with HPO. LightGBM and XGBoost showed improvements of around 2-3% in accuracy with HPO

2. Categorical Dataset: CatBoost outperformed other algorithms with an accuracy of 99.5% without HPO and showed a 2% improvement with HPO. SnapBook was the runner up with 98% accuracy without HPO and improved by 1% with HPO.

3. Temporal Dataset: LightGBM achieved the highest accuracy of 54.45% without HPO, followed closely by CatBoost with 54.12%. With, HPO, SnapBoost showed the most significant improvement with a staggering 27% increase in accuracy, while others showed improvements of around 13-16%.

4. Image Dataset: CatBoost was the clear winner with an accuracy of 61% without HPO. SnapBoost followed with a 50% accuracy. With HPO, XGBOOST showed an impressive 13% improvement. While others showed only marginal changes.

The experimental results indicate that SnapBoost and XGBoost consistently performed well in terms of accuracy and training time across all datasets, and the use of hyper-parameter optimisation can further improve their accuracy. CatBoost excels in handling categorical data and outperforms other algorithms in image data; however, it is the slowest algorithm in three out of four cases. LightGBM is fast in terms of training on all datasets except image data, but its accuracy is inconsistent. The use of HyperOpt for hyperparameter optimisation significantly improves the accuracies of all algorithms, reducing overfitting in some cases and helping in making better fits to the data in others. This demonstrates the value of automated hyperparameter tuning techniques in enhancing the performance of gradient boosting algorithms. Based on the findings of this study, there several improvements that can be considered. Further exploration of the impact of hyper-parameter tuning on the performance of these algorithms, possibly using other optimisation techniques. Other optimisers that could be exploited are the traditional random search, or grid search, an exhaustive search over a manually specified subset of hyperparameters. However, grid search can be computationally expensive but might lead to better performance by exploring more combinations. Another improvement that could be made is the investigation of the algorithms' performance on a larger dataset or more complex tasks to better understand the scalability and applicability in real-world problems. Furthermore, evaluate the performance of the algorithms on different hardware configurations and computing platforms to gain insights into their compatibility and performance under variating conditions. Finally, development of hybrid or ensemble models which combine the strengths of each algorithm. This may allow the achievement of a better overall performance and robustness.

3 Methodology

In this section of the research, I outline a comprehensive approach to predicting car prices using a machine learning technique, Extreme Gradient Boosting. As depicted in Figure 1, the proposed architecture illustrates the various stages involved in the research process, beginning with data collection through web scraping of the website pistonheads.com [4] which boasts a huge number of listings. This stage entails extracting pertinent information such as car details, features, and pricing, followed by storing the raw data in a suitable format like CSV. Subsequently, data pre-processing and cleaning are conducted to handle missing values, remove duplicate records and outliers, correct data entry errors and inconsistencies, normalize and scale numerical data, encode categorical data, and perform feature engineering.

3.1 Data Collection and Webscraping

Web scraping is an automated technique for the extraction of unstructured data from a webpage. It is a powerful method for researchers, analysts, and organizations to acquire vast quantities of data from online sources efficiently, enabling them to glean valuable insights, trends, and patterns [15]. The web scraping methodology entailed several stages, including initiating the browser, locating elements on the webpage, extracting data, interacting with buttons, scrolling through the page, managing cookies and cache, and saving the data to a CSV file. To execute browser automation and engage with web elements, the Selenium library was employed, while the Python CSV library was utilized to write the extracted data to a CSV file.

The Selenium library's `WebDriver` was employed to launch the Chrome browser and maximize the window, ensuring proper loading and rendering of all elements on the webpage. The WebDriverWait class was used to wait until specific elements, such as the cookie consent button or car listings, were available before proceeding with the scraping process. A series of custom functions were implemented to systematically extract the required data from the PistonHeads website. These functions enabled the location and extraction of relevant elements and specifications from the website. Additionally, certain functions facilitated the scrolling and loading of more car listings, as well as the interaction with the "View more" button, when available, to load further listings. To avoid being detected and blocked by the website during the web scraping process, the use of a VPN was recommended, and it was essential to periodically clear cookies and cache. Furthermore, a function was implemented to limit the number of pages scraped in each session, reducing the risk of being blocked by the website and minimizing waiting time during testing.

The process began with the initialization of necessary Python libraries and the Selenium WebDriver, which navigated to the website and accepted cookies. The WebDriver then executed a loop to scroll through and load car advertisements on the search results page, clearing cookies and cache periodically for optimal performance.

Upon extracting the car title, specifications, and price for each advertisement, the specifications were parsed to obtain details such as year, car body type, mileage, cylinder volume, horsepower, transmission, petrol type, and the number of owners. Prior to recording the extracted data in a CSV file, the script performed data validation to ensure the absence of empty or missing values. Simultaneously, duplicate entries were checked and avoided by monitoring car titles and mileages, thereby maintaining data uniqueness.

Finally, the validated and unique data was written to a CSV file for subsequent analysis and modelling in this research. The WebDriver was closed upon completion of the web scraping process, and the total execution time was displayed. This methodology facilitated the collection of a clean and accurate dataset, providing a robust foundation for the ensuing analysis and machine learning tasks in this study. The flowchart below, in Figure 2, illustrates the order in which the scraping process adapted. The car features extracted from the website are 10 in total which are: title of the listing, year, car body type, mileage, cylinder volume, horsepower, transmission, petrol type, owners, and price.

3.1.1 Challenges and Limitations

Despite the utility of web scraping for gathering data from the PistonHeads website, the process entailed several challenges and limitations:

1. Website access restrictions: The use of a VPN was necessary to avoid being blocked by the website during the scraping process. Additionally, the script had to be adapted to limit the number of pages scraped in each session.

2. CAPTCHA challenges: Websites often employ CAPTCHA challenges to protect against automated scripts and bots. Although no specific CAPTCHA challenges were encountered during the web scraping process in this project, it is crucial to remain vigilant to such potential obstacles.

3. Inconsistent website formatting: Occasionally, the formatting of the website's content might be inconsistent, complicating the consistent extraction of data. To address this

issue, the script incorporated a function that checked if three or more values in the extracted row were empty or None, skipping the row if any such values were encountered.

4. Lengthy waiting times: Due to the large volume of data being scraped, the script required a significant amount of time to complete, taking approximately 14 hours for the entire dataset. By limiting the number of pages scraped during testing, waiting times could be reduced.

In conclusion, despite these challenges and limitations, the web scraping process enabled the successful acquisition of a comprehensive dataset on car listings from the pistonheads.com website. This data was then employed to analyse and derive valuable insights for the study.

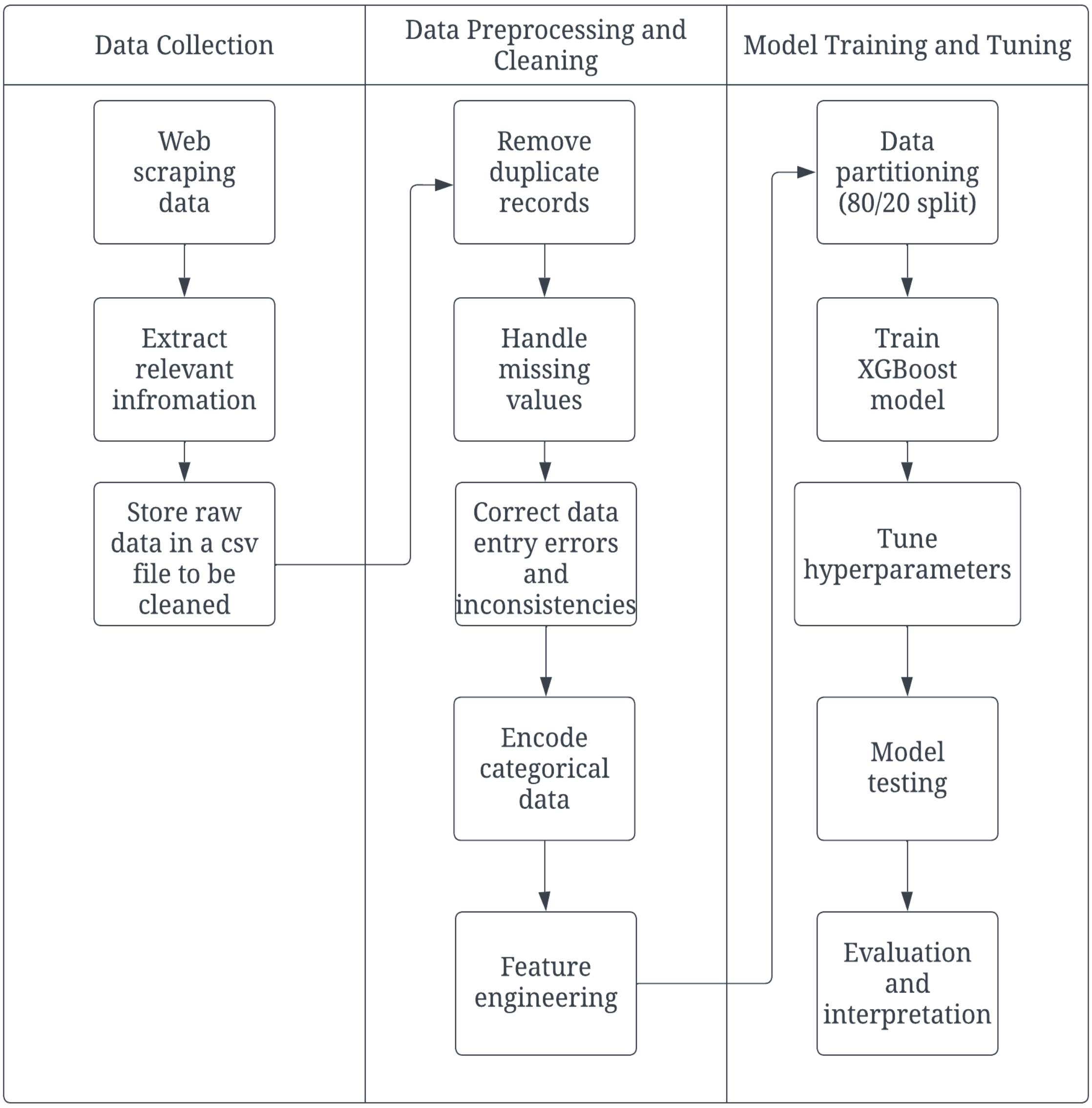

Data Collection
Data Preprocessing and Cleaning
Model Training and Tuning
Web scraping data
Extract relevant infromation
Store raw data in a csv file to be cleaned
Remove duplicate records
Handle missing values
Correct data entry errors and inconsistencies
Encode categorical data
Feature engineering
Data partitioning (80/20 split)
Train XGBoost model
Tune hyperparameters
Model testing
Evaluation and interpretation

Figure 1: Approach to Model Construction

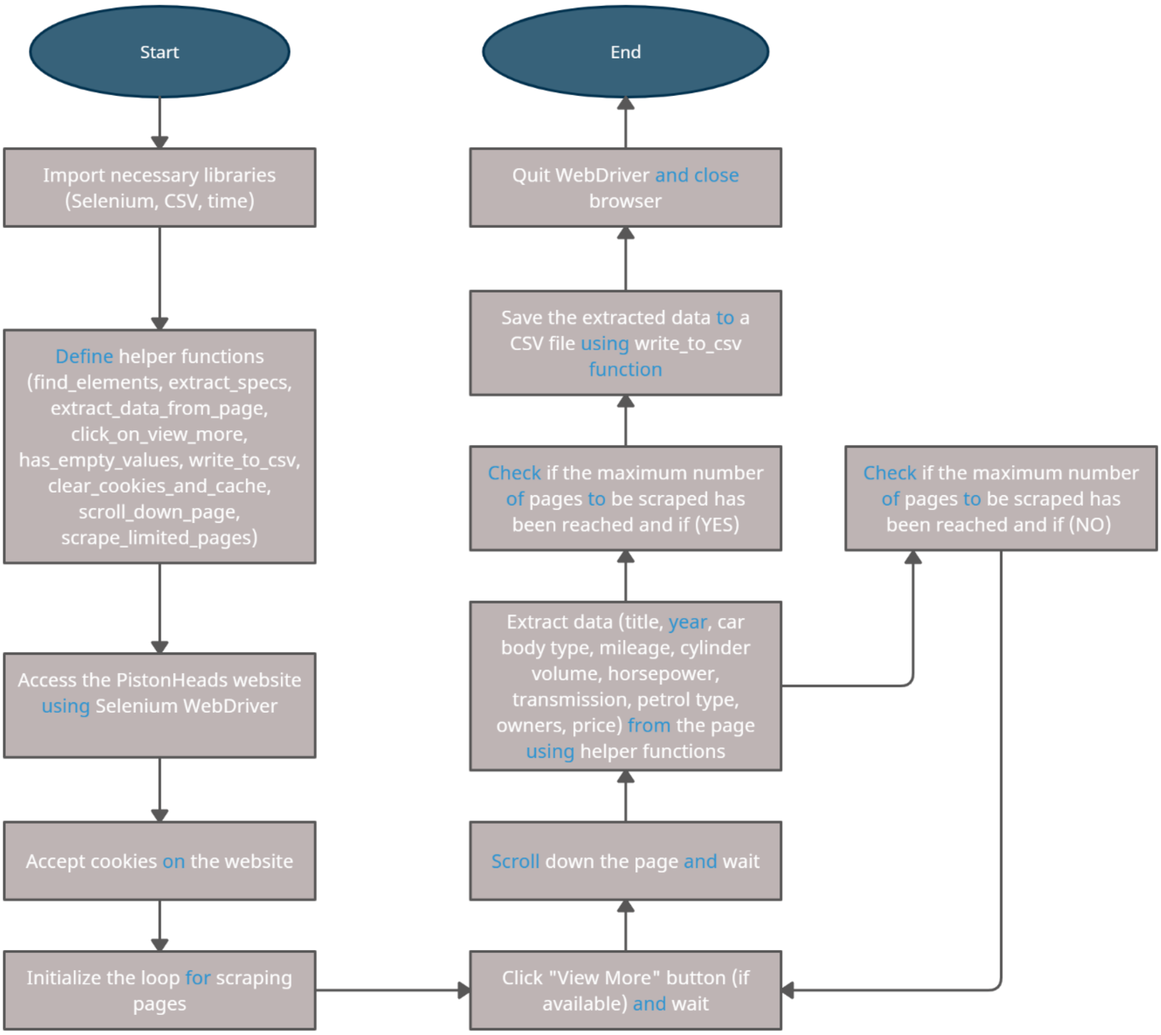

Start
Import necessary libraries (Selenium, CSV, time)
Define helper functions (find_elements, extract_specs, extract_data_from_page, click_on_view_more, has_empty_values, write_to_csv, clear_cookies_and_cache, scroll_down_page, scrape_limited_pages)
Access the PistonHeads website using Selenium WebDriver
Accept cookies on the website
Initialize the loop for scraping pages
Click "View More" button (if available) and wait
Scroll down the page and wait
Extract data (title, year, car body type, mileage, cylinder volume, horsepower, transmission, petrol type, owners, price) from the page using helper functions
Check if the maximum number of pages to be scraped has been reached and if (YES)
Check if the maximum number of pages to be scraped has been reached and if (NO)
Save the extracted data to a CSV file using write_to_csv function
Quit WebDriver and close browser
End

Figure 2: Webscraping Algorithm Implemented

3.1.2 Data Cleaning and Preprocessing

3.1.2.1 Cleaning

The dataset scraped was unclean and not ready for use; hence, prior to feeding it into the XGBoost model, several preprocessing steps were undertaken to ensure the dataset's suitability for training and testing. Table 1 shows how the data looks like before cleaning has been. Firstly, irrelevant or redundant variables were removed, and inconsistencies or errors in the data were addressed. Rows in the dataset containing more than 3 empty values were removed as they were deemed insignificant and affected the model’s performance. Secondly, missing values were filled using appropriate imputation methods such as mean, median, or mode. For the "*Car Body Type*" attribute, mode imputation was employed to handle missing values due to the categorical nature of the data. The mode imputation method replaces missing values with the most frequent category within the attribute. In the context of car body types, it is reasonable to assume that the most common type would be representative of the missing values. This method helps maintain the overall distribution of the data while introducing minimal bias, which is crucial for the accuracy and reliability of the downstream analysis and modeling.

Since the skewness value for the "*Mileage*" is 1.30, indicating a positively skewed distribution (right-skewed), the median was used for imputing missing values. The median is less sensitive to extreme values and outliers than the mean; in a skewed distribution, the median typically provides a better representation of the central tendency of the data. For the same reason, the median was used for the "*Cylinder Volume*" attribute, which had a positive skewing value of 2.28. Furthermore, mean imputation was applied to the "*Horsepower*" variable, as it only had a skewing value of 0.48, implying that the data is approximately symmetric. Although the mean is more influenced by outliers than the median [16], this was not an issue, as outliers were addressed, which will be elaborated upon in a section below.

Regarding the "*Transmission*" and "*Petrol Type*" attributes, the mode was used, as both features are categorical. Since the most common transmission and petrol type are likely to be representative estimates for the missing values, this method contributes to a more accurate and robust analysis and modeling process. To better handle these categorical features, "*Car Body Type*" and "*Petrol Type*" were target encoded due to their multiple classes. Target encoding allows for a more compact representation of these categorical variables while preserving the relationship between the categories and the target variable. In contrast, the "*Transmission*" attribute was label encoded, as it only contained three classes: manual, auto, or other. Label encoding assigns an integer value to each category, which is suitable for a small number of distinct categories.

Lastly, for both "*Owners*" and "*price*," the median method was imputed, as they had skewing values of 1.45 and a whopping 6.11, respectively. Table 1 below illustrates the numerical attributes, the skewing value, and the imputation method used.

Attribute/Variable	Skew Value	Imputation Method
Mileage	1.29	Median
Cylinder Volume	0.78	Median
Horsepower	0.46	Mean
Owners	1.45	Median
price	6.11	Median

Table 1: Representing Imputation Methods of Numerical Variables

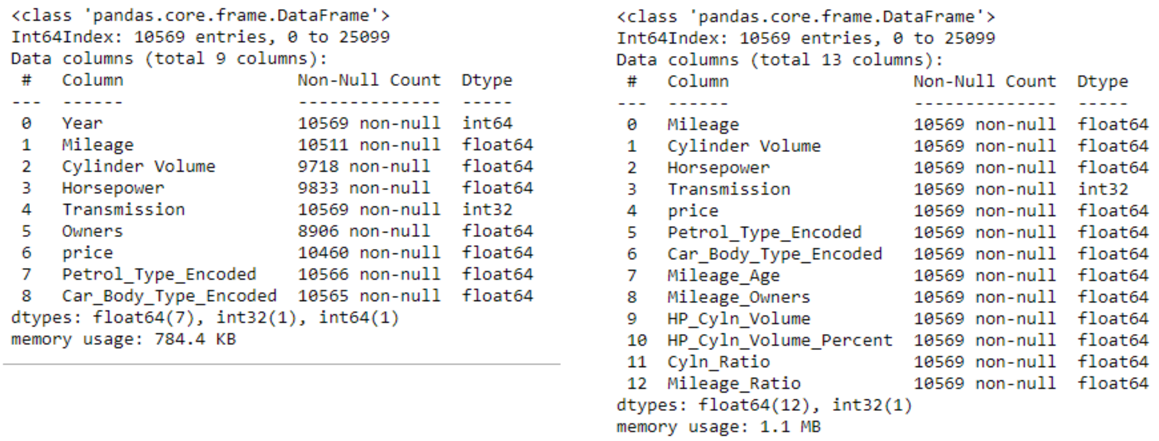
3.1.2.2 Feature Extraction and Selection

The feature extraction and selection phase aimed to identify the most relevant and influential attributes and create new features to improve the predictive performance of the XGBoost model. The initial dataset contained the following attributes: title, Year, Car Body Type, Mileage, Cylinder Volume, Horsepower, Transmission, Petrol Type, Owners, and price. The "title" attribute was dropped, as it was irrelevant to the model.

To create new features, the following transformations were applied:

- "*Age*" was computed as the difference between the current year and the car's manufacturing year.
- "*Mileage_Age*" was derived by multiplying the "*Mileage*" and "*Age*" variables, as both were negatively correlated with the target variable.
- "*Mileage_Owners*" combined the "*Mileage*" and "*Owners*" attributes, which were both negatively related to the car price.
- "*HP_Cyln_Volume*" integrated the "*Horsepower*" and "*Cylinder Volume*" attributes, as both were positively related to the price.
- "*HP_Cyln_Volume_Percent*" represented the ratio of "*Horsepower*" to "*Cylinder Volume*."
- "*Cyln_Ratio*" was calculated by normalizing the "*Cylinder Volume*" attribute using min-max normalization.
- "*Mileage_Ratio*" was calculated by normalizing the "*Mileage*" attribute using min-max normalization.

After creating these new features, any missing values resulting from the transformations were addressed. For "*Cyln_Ratio*" and "*HP_Cyln_Volume_Percent*," missing values were replaced with the median and mean values, respectively. These features were also rounded to four decimal places to ensure a more compact representation. Finally, the original columns "Age," "Year," and "Owners" were dropped from the dataset, as the newly created features better captured their relationships with the target variable. These transformations resulted in a more informative set of features that could enhance the performance of the XGBoost model. Below represents the original and final attributes of the model.



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10569 entries, 0 to 25099
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Year                  10569 non-null  int64  
 1   Mileage               10511 non-null  float64
 2   Cylinder Volume       9718 non-null   float64
 3   Horsepower            9833 non-null   float64
 4   Transmission          10569 non-null  int32  
 5   Owners                8906 non-null   float64
 6   price                 10460 non-null  float64
 7   Petrol_Type_Encoded   10566 non-null  float64
 8   Car_Body_Type_Encoded 10565 non-null  float64
dtypes: float64(7), int32(1), int64(1)
memory usage: 784.4 KB
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10569 entries, 0 to 25099
Data columns (total 13 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Mileage                 10569 non-null  float64
 1   Cylinder Volume         10569 non-null  float64
 2   Horsepower              10569 non-null  float64
 3   Transmission            10569 non-null  int32  
 4   price                   10569 non-null  float64
 5   Petrol_Type_Encoded     10569 non-null  float64
 6   Car_Body_Type_Encoded   10569 non-null  float64
 7   Mileage_Age             10569 non-null  float64
 8   Mileage_Owners          10569 non-null  float64
 9   HP_Cyln_Volume          10569 non-null  float64
 10  HP_Cyln_Volume_Percent  10569 non-null  float64
 11  Cyln_Ratio              10569 non-null  float64
 12  Mileage_Ratio           10569 non-null  float64
dtypes: float64(12), int32(1)
memory usage: 1.1 MB
```

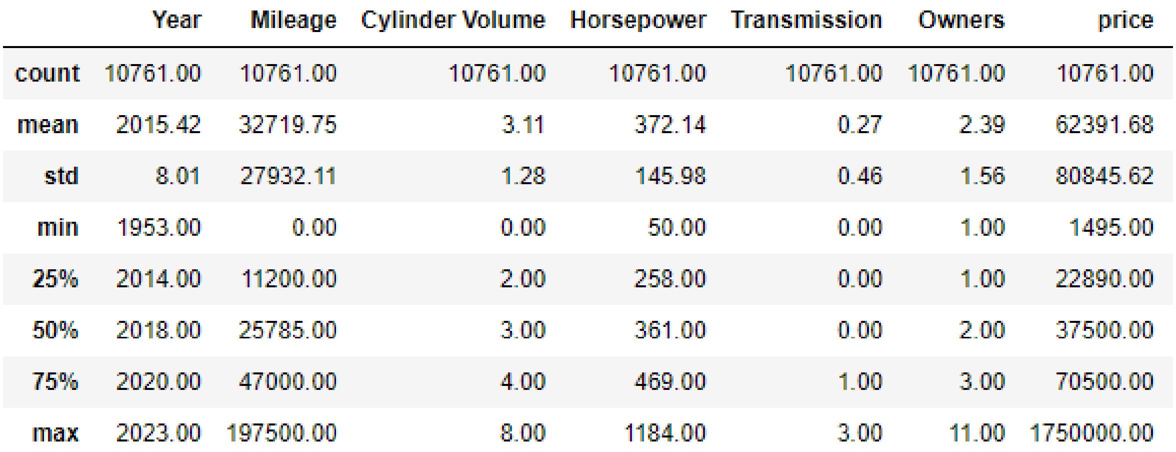
Figure 3:Attributes Before & After Feature Engineering

3.1.3 Visualising and Exploratory the Data

3.1.3.1 Descriptive Statistics

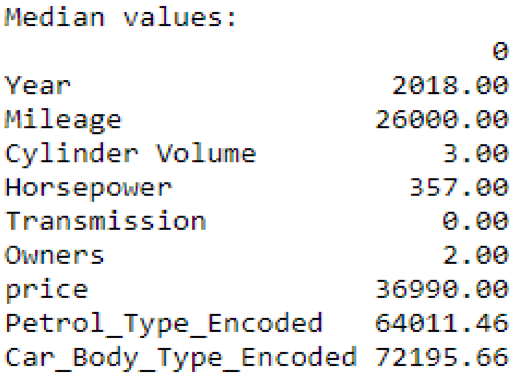
Table 2 and Figure 5 presents the descriptive statistics for the continuous numerical features, including the count, mean, standard deviation, minimum, and maximum values, while Table 3 contains the frequency distribution of the categorical features.

The mean value of "*Mileage*" is 32,720 miles, with a median of 26,000 miles, demonstrating a slight positive skewness in the data. The standard deviation of 27,932 miles indicates considerable variability in the mileages of the cars in the dataset. For "*Cylinder Volume*," the mean and median are 3.11L and 3.0L, respectively, indicating a relatively symmetrical distribution, as further supported by the moderately low skewness value of 0.83. The "Horsepower" variable presents a mean of 372.14 bhp and a median of 357 bhp, with a standard deviation of 145.98 bhp, reflecting moderate variation in the horsepower of the cars.



	Year	Mileage	Cylinder Volume	Horsepower	Transmission	Owners	price
count	10761.00	10761.00	10761.00	10761.00	10761.00	10761.00	10761.00
mean	2015.42	32719.75	3.11	372.14	0.27	2.39	62391.68
std	8.01	27932.11	1.28	145.98	0.46	1.56	80845.62
min	1953.00	0.00	0.00	50.00	0.00	1.00	1495.00
25%	2014.00	11200.00	2.00	258.00	0.00	1.00	22890.00
50%	2018.00	25785.00	3.00	361.00	0.00	2.00	37500.00
75%	2020.00	47000.00	4.00	469.00	1.00	3.00	70500.00
max	2023.00	197500.00	8.00	1184.00	3.00	11.00	1750000.00

Table 2: Dataset Statistics



```
Median values:
                           0
Year                 2018.00
Mileage             26000.00
Cylinder Volume         3.00
Horsepower            357.00
Transmission            0.00
Owners                  2.00
price               36990.00
Petrol_Type_Encoded 64011.46
Car_Body_Type_Encoded 72195.66
```

Figure 5: Median Values of Numerical Data

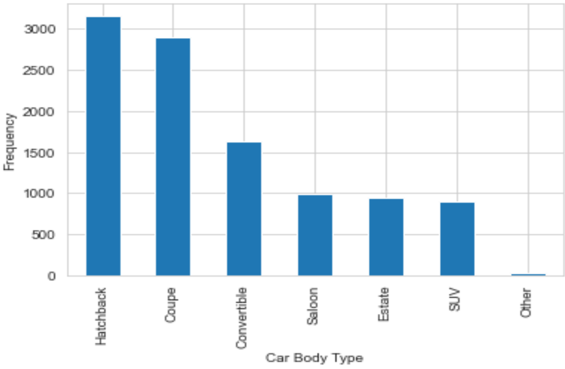


Figure 4: Distribution of Car Body Types

As illustrated in Figure 4: Distribution of Car Body TypesFigure 4, in terms of the categorical features, the most common "*Car Body Type*" is hatchback, representing 32% of the dataset, followed by coupes at 26.4%, and convertibles at 16.2%. The remaining car body types, including saloons, estates, SUVs, and others account for the remaining 25.4%. For

"*Transmission*," automatic transmission is the most prevalent, constituting 72.7% of the dataset, followed by manual at 25.6%, and other at 1.7%. The "Petrol Type" distribution is dominated by petrol vehicles, which represent 71.8% of the dataset, with diesel cars accounting for 7.2%, and other fuel types, such as electric and hybrid, making up the remaining 21%. Frequency distributions are demonstrated below in Figure 6.

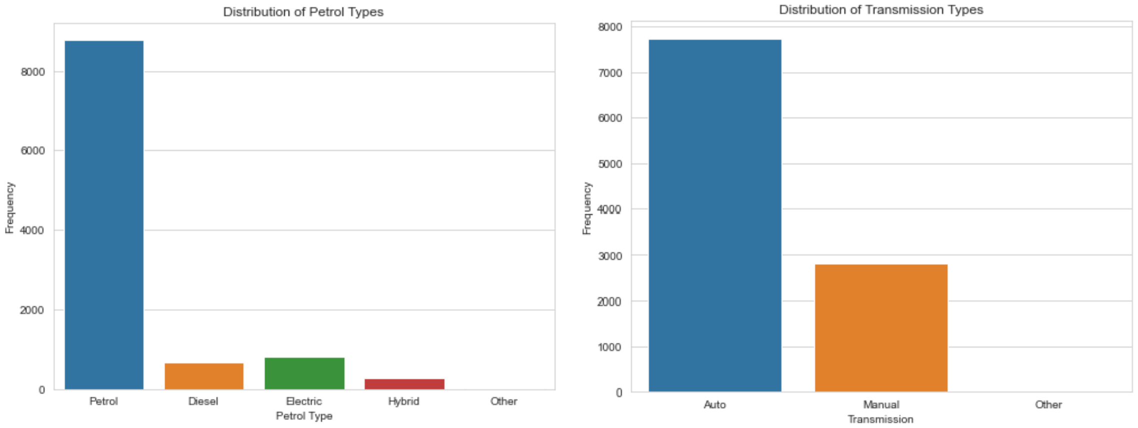
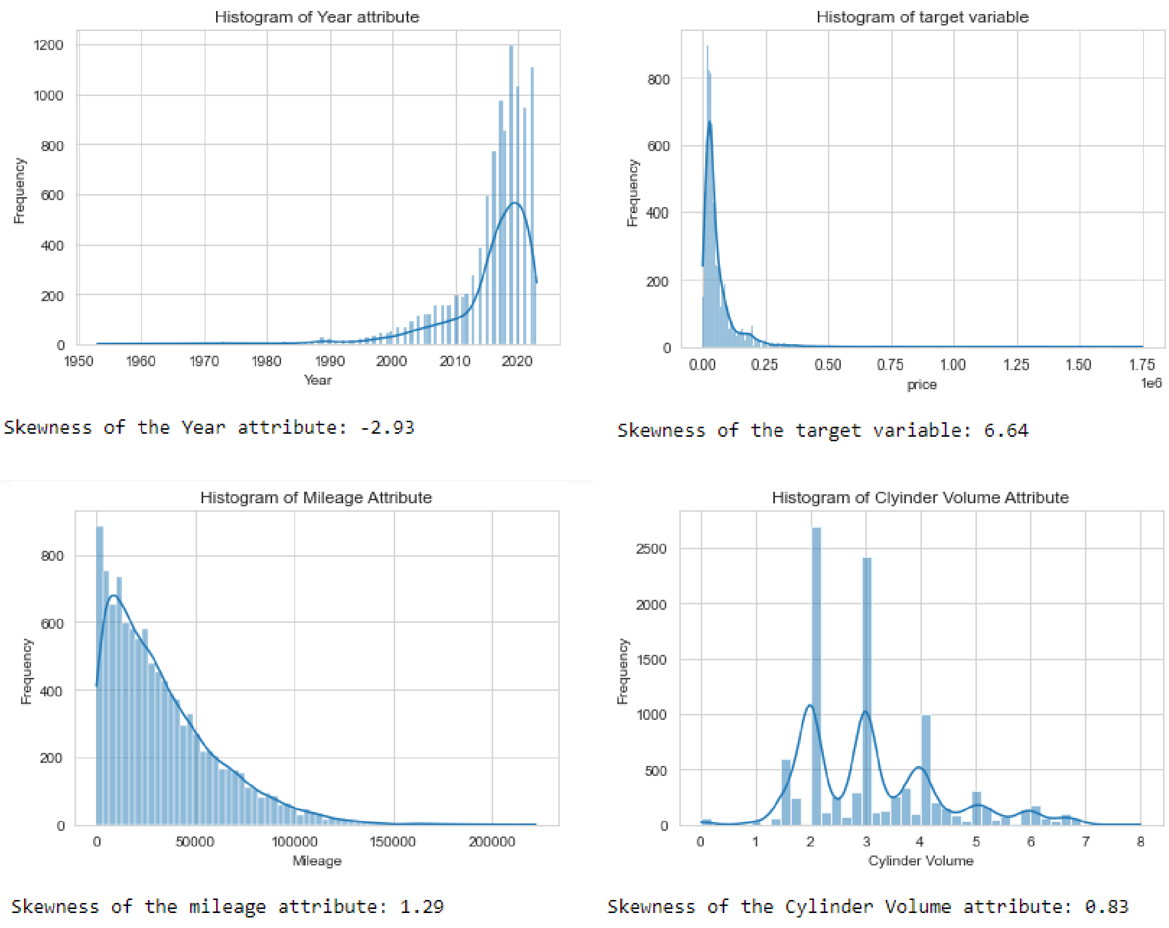


Figure 6: Petrol Type and Transmission Distribution

3.1.3.2 Data Visualization

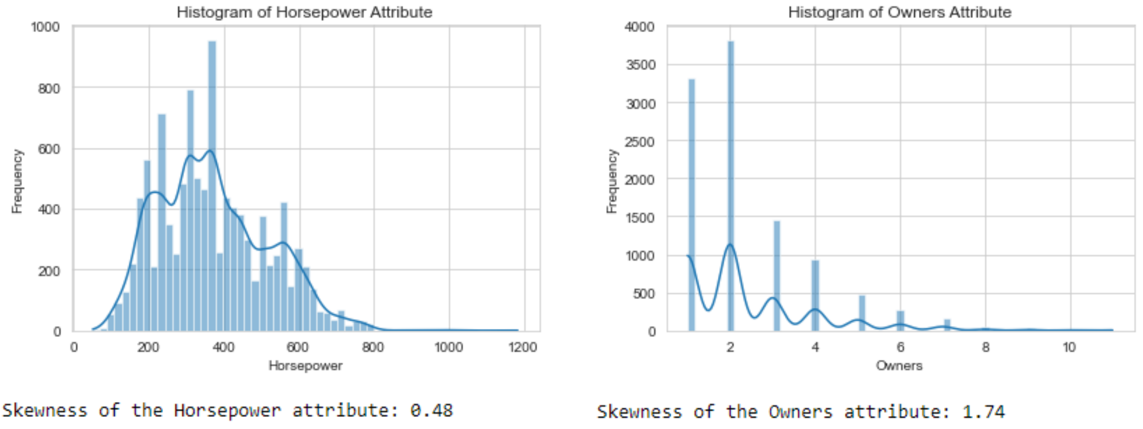
I have conducted data visualisation techniques that have provided me with insights which will aid in the subsequent steps of model development and evaluation. Histograms were created for the following continuous attributes: 'Year,' 'Mileage,' 'Cylinder Volume,' 'Horsepower,' and 'Owners.' Skewness values were calculated for each attribute, revealing that 'Year' has a negative skew (-2.93), while the other attributes have positive skewness. The target variable 'price' has a skewness of 6.64, indicating a highly skewed distribution. Skewness values close to zero are considered normally distributed, while positive or negative values indicate the distribution's asymmetry. The "Horsepower" had the lowest Furthermore, histograms are shown below in Figure 7 and Figure 8 along with the skewing value of each of the numerical variables.

Bar plots were created for the categorical attributes: '*Car_Body_Type_Encoded*,' '*Transmission*,' and '*Petrol_Type_Encoded*.' The bar plots in figure 3 show the distribution of the categories and the mean price for each category. We can see that the mean price for automatic transmission car to be around £71,000 whilst manual cars' mean is around £37,200.



```
Skewness of the Year attribute: -2.93
Skewness of the target variable: 6.64
Skewness of the mileage attribute: 1.29
Skewness of the Cylinder Volume attribute: 0.83
```

Figure 7: Histograms of Numerical Variables and their Skew Values



```
Skewness of the Horsepower attribute: 0.48
Skewness of the Owners attribute: 1.74
```

Figure 8: Histograms of Numerical Variables and their Skew Values

3.1.3.3 Correlation Analysis

In this section, I conducted a correlation analysis to evaluate the relationships between various variables and the target variable, price. I first examined the distributions of the continuous variables: Year, Mileage, Cylinder Volume, Horsepower, and Owners, through histograms and skewness values.

To analyse the relationships between categorical variables and the target variable, I visualized the distribution of Car Body Type, Transmission, and Petrol Type using bar plots. Additionally, I have applied logarithmic transformation to the Mileage column and plotted it against the price to observe its distribution as it was clustered heavily on the bottom left corner of the scatter plot. Figure 9 shows the mileage before applying log transformation and after.

I have then calculated the Pearson correlation coefficients and p-values for the continuous variables against the target variable. The results shown in figure tell us that "*Year*" has a very weak positive correlation (-0.05) with the target variable, while "*Cylinder Volume*" and "*Horsepower*" have relatively strong positive correlations with the price, 0.43 and 0.52 respectively. On the other hand, the "*Mileage*" variable has a relatively strong negative relation against the price, with a value Pearson correlation value of -0.36. Finally, the "*Owners*" attribute was the weakest of all, with a negative relation value of -0.12.

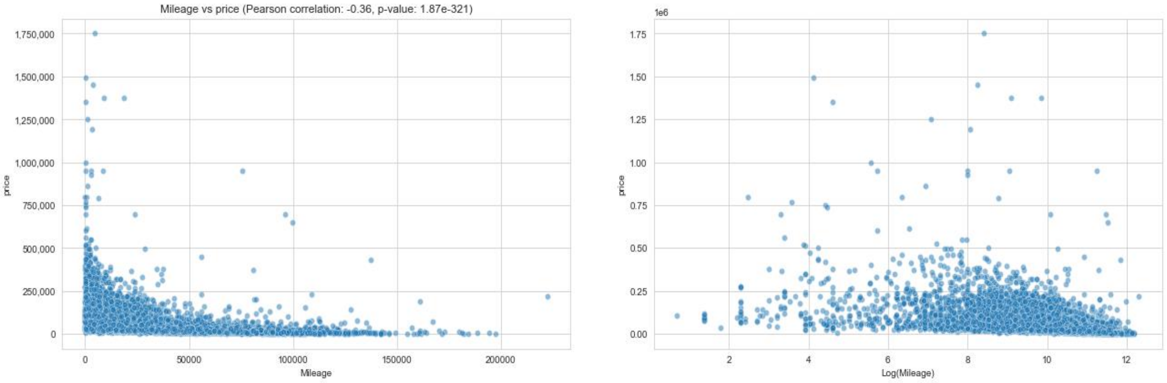


Figure 9: Mileage before and after transformation

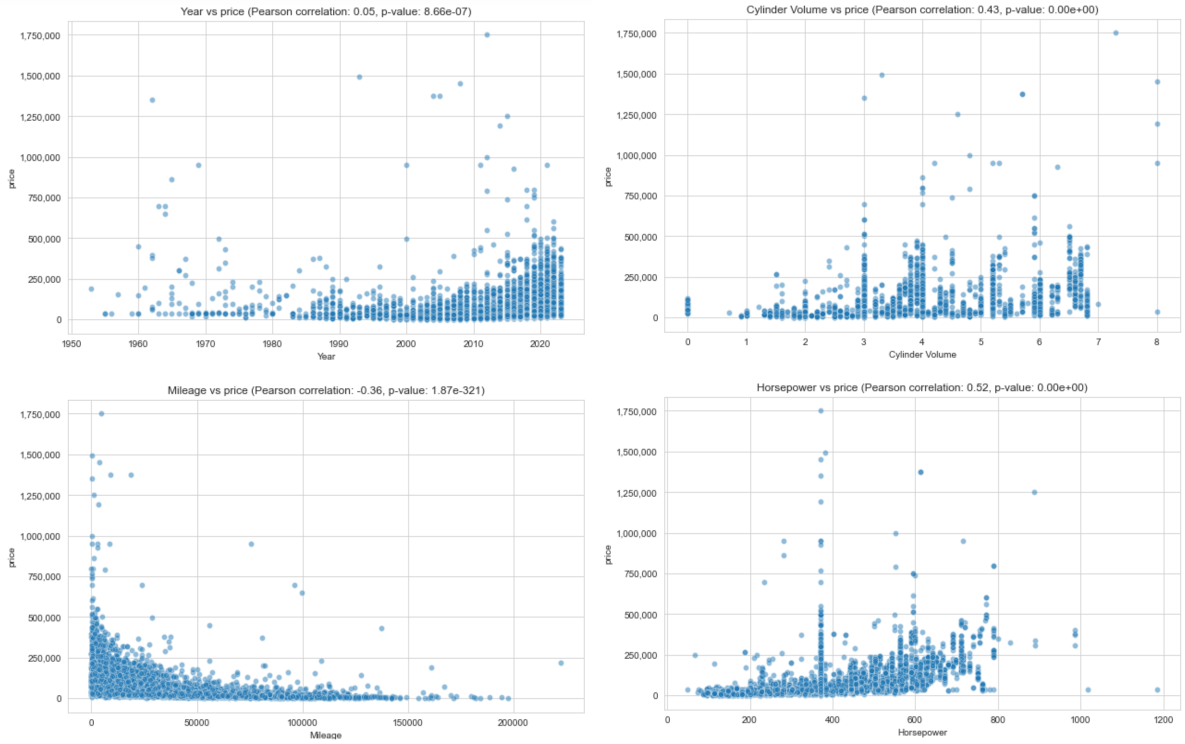


Figure 10: Scatter plots and Pearson correlation (A)

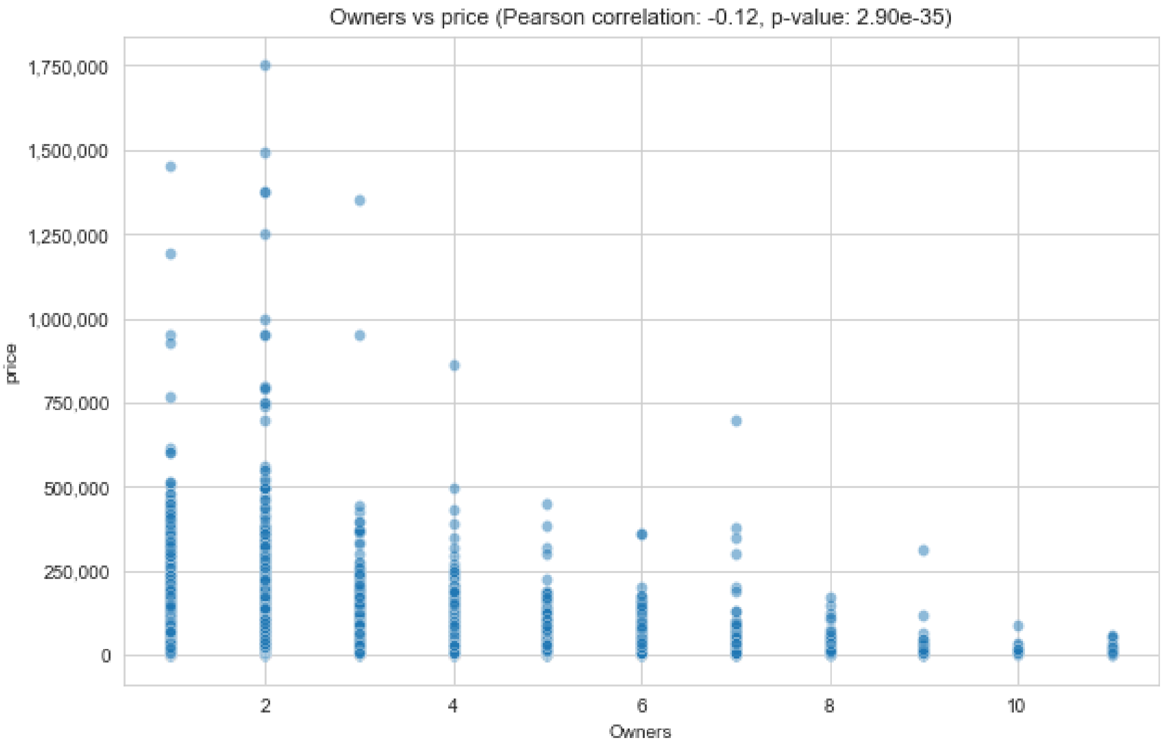


Figure 11: (B)

I have also visualized the relationships between the target-encoded categorical variables (Car Body Type and Petrol Type) and the target variable using box plots. The correlations were calculated and found to be weak. Car Body Type had the highest positive correlation with a value of 0.26, while Petrol Type was found to have a weak positive correlation of only 0.12. Lastly, surprisingly, Transmission had a negative weak correlation against the target variable with a person value of -0.15.

I created new interaction features (e.g., Mileage_Age, Mileage_Owners, HP_Cyln_Volume) by combining pairs of existing variables that were correlated with the target variable. Scatter plots were created to visualize these relationships, and Pearson correlation coefficients and p-values were calculated. The interaction features demonstrated varying degrees of correlation with the target variable, with HP_Cyln_Volume exhibiting the strongest positive correlation.

To further understand the relationships between variables and the target variable 'price', a correlation matrix was calculated for the dataset, and a heatmap was generated to visualize the correlations between all the variables. The heatmap in Figure 12 provides an overview of the linear relationships between pairs of variables, with the strength and direction of the correlations represented by the colour and the annotated correlation coefficient.

The Mileage_Age variable demonstrates a negative correlation with the price, with a correlation coefficient of -0.23. This indicates that as the ratio of mileage to the age of the car increases, the price of the car tends to decrease, albeit with a relatively weak relationship. Mileage_Owners also exhibits a negative correlation with the price, with a coefficient of -0.30. This suggests that as the ratio of mileage to the number of owners increases, the price of the car is likely to decrease, indicating a somewhat stronger relationship compared to Mileage_Age. On the other hand, the HP_Cyln_Volume variable, which represents the product of horsepower and cylinder volume, displays a positive correlation with the price, with a correlation coefficient of 0.60. This implies that cars with higher combined horsepower and cylinder volume values are likely to have higher prices, indicating a moderately strong relationship. The HP_Cyln_Volume_Percent variable, calculated as the ratio of horsepower to cylinder volume, demonstrates a weak positive correlation with the price, with a correlation coefficient of 0.18. This suggests that cars with higher ratios of horsepower to cylinder volume may have slightly higher prices. The Cyln_Ratio variable, representing the normalized cylinder volume, exhibits a moderately strong positive correlation with the price, with a coefficient of 0.48. This indicates that cars with higher values of normalized cylinder volume are likely to have higher prices. Lastly, the Mileage_Ratio variable shows a negative correlation with the price, with a correlation coefficient of -0.44. This indicates that as the ratio of the current mileage to the average mileage for the car's age increases, the price of the car tends to decrease, suggesting a moderately strong relationship.

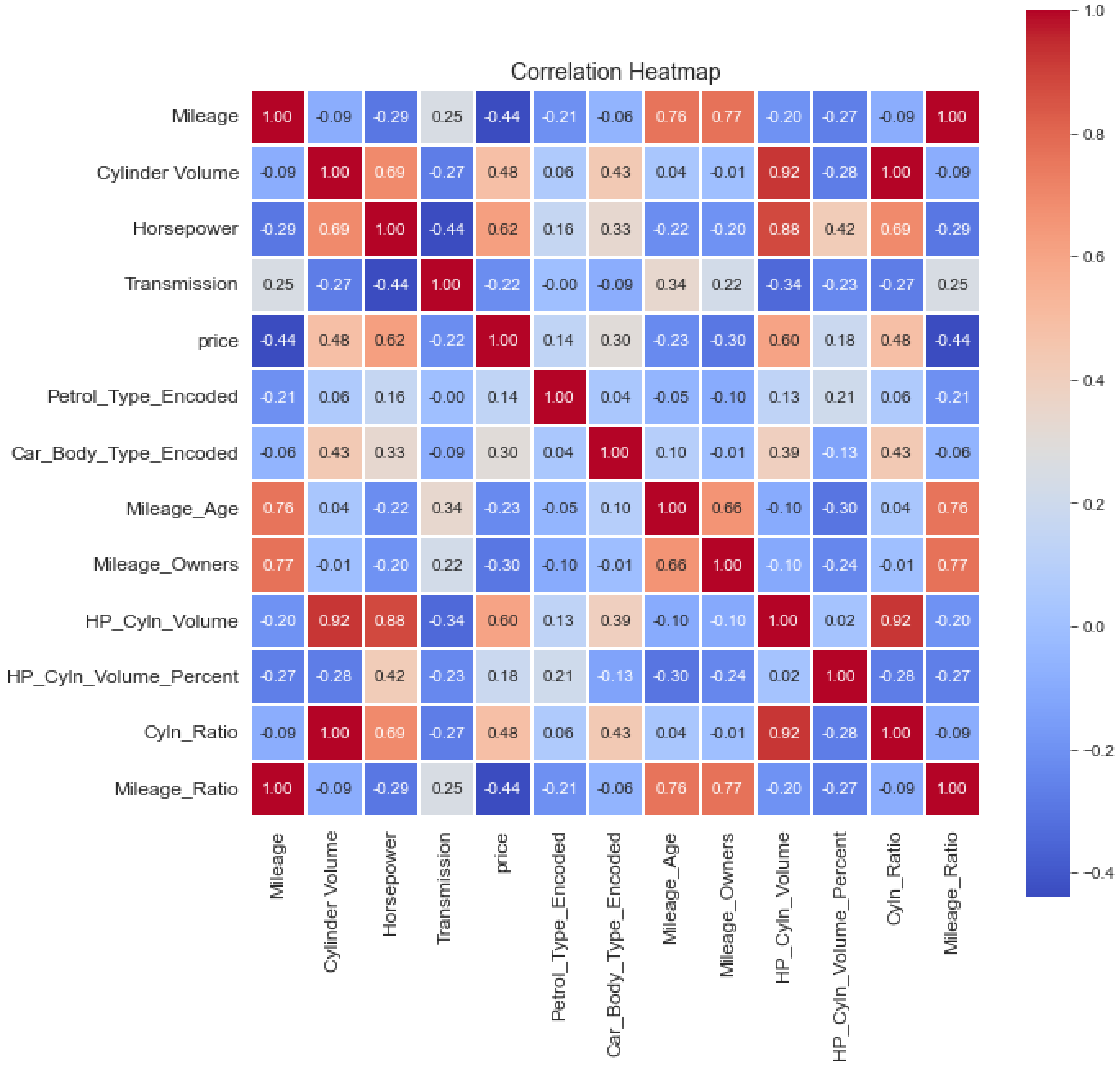


Figure 12: Correlation Heatmap

3.1.3.4 Feature Importance

In addition to correlation analysis, feature importance was also investigated to understand the impact of each feature on the target variable 'price'. The XGBoost model, known for its strong predictive power, was utilized to estimate the importance of each feature. The importance scores were calculated based on the number of times a feature was used to split the data across all trees in the ensemble, and the improvement in the model's performance as a result of these splits.

The feature importance scores were extracted from the trained XGBoost model, and the scores were mapped to the corresponding feature names. The features were then sorted by their importance scores in descending order as shown below in Figure 13.

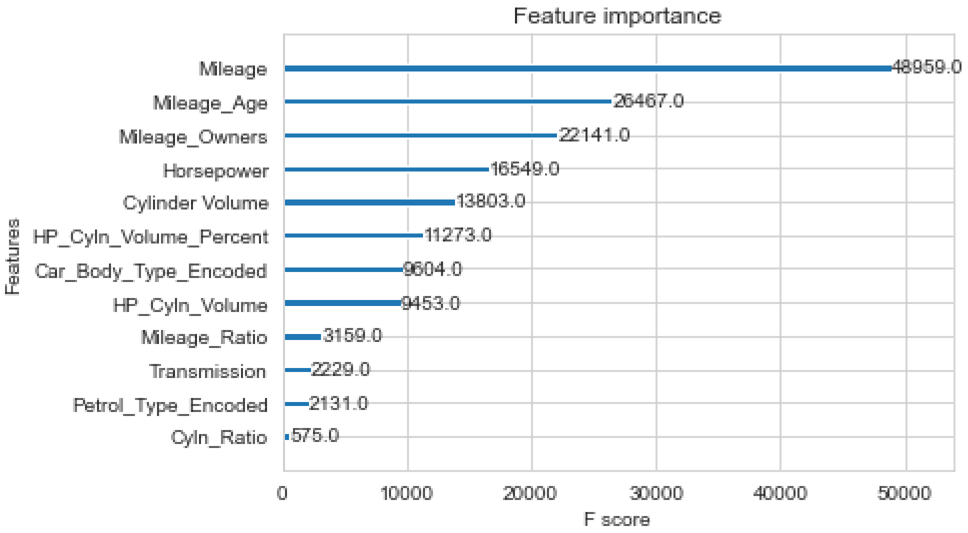


Figure 13: Feature Importance Scores

3.1.4 Model Development and Evaluation

3.1.4.1 Selection of Machine Learning Algorithm

The selection of an appropriate machine learning algorithm is a critical step in building an effective predictive model. In this study, I chose the eXtreme Gradient Boosting (XGBoost) algorithm for building the regression model. XGBoost is an advanced implementation of gradient boosting machines, designed to be more efficient and deliver better performance than traditional gradient boosting techniques. The decision to use XGBoost was based on several factors:

- Robustness: XGBoost has been proven to deliver excellent performance across various datasets and problems. It consistently ranks among the top algorithms in machine learning competitions, demonstrating its robustness and ability to handle diverse data types and distributions.

- Speed and efficiency: XGBoost is designed for parallel processing and can utilize multiple CPU cores, making it faster and more efficient than other tree-based models. This is particularly important when dealing with large datasets, as it allows for quick model training and evaluation.

- Handling missing values: XGBoost has built-in support for handling missing values, which is an essential feature when working with real-world datasets that often contain missing or incomplete information.

- Regularization: XGBoost incorporates L1 and L2 regularization techniques that help prevent overfitting and improve model generalization. This is particularly important when working with high-dimensional datasets, as it can help reduce the complexity of the model and improve its predictive accuracy.

- Flexibility: XGBoost allows for easy customization of various hyperparameters, enabling us to fine-tune the model to our specific dataset and problem. This flexibility ensures that the model can be tailored to achieve the best possible performance.

- Interpretability: XGBoost provides feature importance scores, which can help in understanding the impact of individual features on the target variable. This is essential for gaining insights into the data and making informed decisions.

3.1.4.2 Model Training and Cross-Validation

The dataset was first divided into input features (X) and target variable (y), where 'price' represents the target variable. The data was then split into training and test sets in an 80/20 ratio, using a random state of 42 to ensure reproducibility of results. I employed the eXtreme Gradient Boosting (XGBoost) regressor for training the model, with the objective function set to 'reg:squarederror', along with initial hyperparameters such as max_depth, learning_rate, n_estimators, subsample, reg_alpha, and reg_lambda.

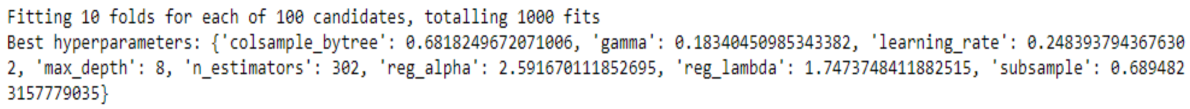
To validate the model's performance, I used K-Fold cross-validation with ten splits, shuffling the data before each fold. This technique ensures the model's generalization across different subsets of the dataset. The mean absolute error (MAE) and R2 score were calculated for each fold, resulting in a mean MAE of 10181.63 and a mean R2 score of 0.830. The standard deviation for MAE and R2 scores were 866.10 and 0.025, respectively, indicating a relatively low variance in model performance across different folds.

Upon fitting the model on the training set, the R2 scores for training and testing sets were 0.996 and 0.83, respectively. The difference between these scores suggests that the model may be overfitting the training data, which was addressed by hyperparameter tuning and model optimization.

3.1.4.3 Hyperparameter Tuning and Model Optimization

To optimize the model and mitigate the overfitting issue, I performed hyperparameter tuning using a randomized search with cross-validation. The search space for hyperparameters included various combinations of learning_rate, max_depth, n_estimators, subsample, colsample_bytree, gamma, reg_alpha, and reg_lambda, which are crucial for controlling the model's complexity and generalization.

The randomized search was conducted with 100 iterations, using K-Fold cross-validation (as described earlier) and scoring based on the negative mean absolute error. The best hyperparameters obtained from the search are shown in Figure 14: Best Hyperparameters



```
Fitting 10 folds for each of 100 candidates, totalling 1000 fits
Best hyperparameters: {'colsample_bytree': 0.6818249672071006, 'gamma': 0.18340450985343382, 'learning_rate': 0.248393794367630
2, 'max_depth': 8, 'n_estimators': 302, 'reg_alpha': 2.591670111852695, 'reg_lambda': 1.7473748411882515, 'subsample': 0.689482
3157779035}
```

Figure 14: Best Hyperparameters

I then trained the XGBoost model using these optimal hyperparameters and calculated the root mean squared error (RMSE) for both training and testing sets. The RMSE for training and testing sets were 2912.59 and 26778.81, respectively, which indicates a reduction in overfitting compared to the initial model.

The model's performance was further evaluated using mean absolute error (MAE), mean squared error (MSE), and R-squared score for the test set. The results were MAE=11697.84, MSE=717104709.33, and R2=0.826, suggesting a satisfactory predictive ability of the model.

Lastly, I analysed the feature importance to understand the impact of each feature on the model's predictions. The top three features contributing to the model were Horsepower (0.279), HP_Cyln_Volume (0.058), and Cylinder Volume (0.056). These findings can provide valuable insights into the most influential factors affecting car prices, which can be useful for decision-makers and stakeholders in the automotive industry.

4 Conclusion

In conclusion, the research project presented in this thesis has developed a robust predictive model for car prices using XGBoost. This model exhibits a satisfactory performance, offering valuable insights into the relationship between various features and car prices. Reflecting on the project, there are several points to consider regarding how the project could have run better, risks that occurred, and possible improvements within the given timeframe or with additional resources.

Hindsight is always valuable, and looking back at the project, there are a few aspects that could have been improved. Firstly, more thorough data cleaning and preprocessing could have been carried out to ensure the highest quality data for model training. Secondly, a more comprehensive feature selection process, including the utilization of domain knowledge, would have been beneficial for refining the model's inputs. Finally, experimenting with different models and hyperparameters in parallel could have provided a more extensive comparison of model performances, leading to the identification of the most suitable approach.

In terms of risks, the limited dataset size and the presence of outliers had a potential impact on model performance. Smaller datasets are more prone to overfitting, while outliers can skew model predictions. Mitigating these risks involved removing outliers and applying cross-validation techniques to ensure model robustness.

Considering the three-month timeframe, certain improvements could have been made. These include conducting more extensive feature engineering and selection, performing a more thorough hyperparameter optimization process, and incorporating additional validation metrics to assess model performance from different perspectives. With the availability of further resources, including time, equipment, and access to more extensive datasets, the following improvements could be pursued:

- Expanding the dataset to include a broader range of car makes and models, increasing the model's generalizability and applicability.

- Incorporating additional features such as car color, trim level, or additional vehicle specifications that could provide further insights into car prices.

- Utilizing more advanced feature selection and engineering techniques, such as principal component analysis (PCA) or recursive feature elimination (RFE), to refine the model's inputs further.

- Experimenting with other machine learning algorithms, such as deep learning or ensemble models, to assess their suitability and performance relative to the XGBoost model.

- Investigating the potential benefits of transfer learning and pre-trained models for predicting car prices.

In summary, this study has made valuable progress in creating a predictive model for car pricing. The discoveries and insights gathered from this work have practical implications for the automotive sector, as well as enhancing the wider understanding of factors that affect car pricing. By examining the lessons learned and pinpointing areas of enhancement, this project has laid groundwork for future research in the field of machine learning.

5 Bibliography

[1] “Supporting green transportation with transport impact assessment: Its deficiency in Chinese cities - ScienceDirect.” https://www.sciencedirect.com/science/article/pii/S1361920919302627.

[2] A. Tigadi, R. Gujanatti, and A. Gonchi, “ADVANCED DRIVER ASSISTANCE SYSTEMS,” vol. 4, no. 3, 2016.

[3] Z. Peng, Q. Huang, and Y. Han, “Model Research on Forecast of Second-Hand House Price in Chengdu Based on XGboost Algorithm,” in *2019 IEEE 11th International Conference on Advanced Infocomm Technology (ICAIT)*, Oct. 2019, pp. 168–172. doi: 10.1109/ICAIT.2019.8935894.

[4] “PistonHeads UK | Cars for Sale | Car News | Motoring Forum.” https://www.pistonheads.com/ (accessed Apr. 16, 2023).

[5] A. Nagpal and G. Gabrani, “Python for Data Analytics, Scientific and Technical Applications,” in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, Feb. 2019, pp. 140–145. doi: 10.1109/AICAI.2019.8701341.

[6] Ö. Çelik and U. Ö. Osmanoğlu, “İkinci El Araba Fiyatlarının Tahmini,” *European Journal of Science and Technology*, pp. 77–83, Aug. 2019, doi: 10.31590/ejosat.542884.

[7] S. Pudaruth, “Predicting the Price of Used Cars using Machine Learning Techniques,” *International Journal of Information & Computation Technology*, vol. 4, pp. 753–764, Jan. 2014.

[8] J. Benesty, J. Chen, and Y. Huang, “On the Importance of the Pearson Correlation Coefficient in Noise Reduction,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 4, pp. 757–765, May 2008, doi: 10.1109/TASL.2008.919072.

[9] S. Kumar, D. Kaur, and A. Parvez, “Prediction of Prices Car Price Prediction with Machne Learning,” in *2022 International Conference on Cyber Resilience (ICCR)*, Oct. 2022, pp. 1–4. doi: 10.1109/ICCR56254.2022.9995772.

[10] F. A. Alghifari, R. Andreswari, and E. Sutovo, “USED CARS PRICE PREDICTION IN DKI JAKARTA USING EXTREME GRADIENT BOOSTING AND BAYESIAN OPTIMIZATION ALGORITHM,” in *2022 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, Nov. 2022, pp. 01–05. doi: 10.1109/ICADEIS56544.2022.10037301.

[11] P. I. Frazier, “A Tutorial on Bayesian Optimization.” arXiv, Jul. 08, 2018. Available: http://arxiv.org/abs/1807.02811

[12] S. R, S. S. Ayachit, V. Patil, and A. Singh, “Competitive Analysis of the Top Gradient Boosting Machine Learning Algorithms,” in *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Dec. 2020, pp. 191–196. doi: 10.1109/ICACCCN51052.2020.9362840.

[13] “hyperopt/hyperopt at master · hyperopt/hyperopt,” *GitHub*. https://github.com/hyperopt/hyperopt.

[14] “Hyperopt Documentation.” https://hyperopt.github.io/hyperopt/.

[15] A. S. Bale, N. Ghorpade, R. S, S. Kamalesh, R. R, and R. B. S, “Web Scraping Approaches and their Performance on Modern Websites,” in *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Aug. 2022, pp. 956–959. doi: 10.1109/ICESC54411.2022.9885689.

[16] K. Zhang and M. Luo, “Outlier-robust extreme learning machine for regression problems,” *Neurocomputing*, vol. 151, pp. 1519–1527, Mar. 2015, doi: 10.1016/j.neucom.2014.09.022.