# A Machine Learning Approach to Predict the Price of Used Cars and Introduce a Recommender System

Rohit Baral
*Univerisity of Cnaberra*
u3268702@uni.canberra.edu.au

Sujan Khanal
*Univeristy of Canbarrea*
u3258630@uni.canberra.edu.au

*Abstract*— **The automotive industry, a substantial sector globally, has witnessed significant growth in the used car market due to various socioeconomic factors. The used car market has grown rapidly in recent years as more people seek affordable alternatives to buying new cars. This project tackles the challenge of predicting used car prices accurately, which involves analyzing numerous factors, from mileage, manufacturers to engine type and vehicle condition. By developing a machine learning model, we aim to create a recommender system that helps buyers and sellers make smarter decisions. Our approach includes using popular machine learning algorithms—Linear Regression, XGBoost, Random Forest, and K-nearest Neighbors—to find the best-performing model for price prediction and recommender system to offer personalized suggestions based on user preferences. To evaluate performance, we measure the model's accuracy with Coefficient of Determination and Root Mean Squared Error metrics. This project aims to simplify and enhance the experience of buying and selling used cars, making the process more transparent and informative for everyone involved.**

***Keywords— Car Price Predict, Machine Learning, Recommender System, Predictive Analytics, Linear Regression, XGBoost, Random Forest Regressor***

## I. INTRODUCTION

The automotive industry is one of the largest industries in the world, with an estimated $2.7 trillion of global commercial activity [1]. The United States alone consists of 16 automobile manufacturers with an overall production of over 10.6 million vehicles in 2023 only [2]. Almost every grown individual is in need of a private car, be it for their work travel or other day to day activities. But not every needy person can afford a brand-new car as they desire, so they search for an affordable alternative, that is to opt for pre-owned cars. The used cars market has seen a significant growth in the recent years, which is driven by several important factors such as the financial condition of buyers, depreciation rate of new vehicles and increased durability of cars. Accurate prediction of the price of used cars is not as simple as it may sound, it requires expert knowledge and analysis due to the nature of their dependence on a variety of factors and features such as mileage, make year, model, engine, vehicle type and many more [3]. This project takes in account the need for an intelligent system which interprets large and complex factors and give accurate insights and predictions. This PRML project aims to provide price prediction models and create a recommender system to help guide the individuals looking to sell or purchase cars and deliver them a better insight into the automotive sector. This will be accomplished by utilizing advanced machine learning algorithms like Linear Regression model, XGBoost and Random Forest to better understand the historical preferences of buyers, analyse the features and extract the trends and patterns to provide a reliable recommender system and price prediction.

Additionally, Recommender systems have become a pivotal tool in various industries, helping users navigate large datasets by providing personalized suggestions. In the context of the automotive industry, recommender systems are particularly useful for matching potential buyers with vehicles that meet their preferences, while also predicting accurate market prices. These systems leverage machine learning techniques to analyze both the features of vehicles and user preferences, creating a customized experience for each user.

## II. BACKGROUND

The problem of utilizing machine learning algorithms to estimate automobile prices has been the subject of several studies. The versatility and interpretability of XGBoost, Random Forest, and Linear Regression make them widely used. In their work, Enis Gegic and his group spoke about several regression models that were created to forecast an automobile's price based on its qualities [4]. The data that was taken into consideration is one of the main shortcomings of this study. They used 1105 samples of restricted characteristics from a smaller dataset. There are several samples with minimal attribute values since the data was collected using a web scraper. To estimate automobile prices in Mauritius, Pudaruth [5] used a variety of machine learning methods, including k-nearest neighbors, multiple linear regression analysis, decision trees, and naïve bayes. Since time may significantly affect the car's pricing, the dataset used to build the prediction model was manually gathered from local newspapers in less than a month. Brand, model, cubic capacity, mileage in kilometres, year of manufacture, exterior colour, transmission type, and price were among the characteristics he examined. Nevertheless, the author discovered that decision trees and Naive Bayes could neither predict nor categorize numerical values. Furthermore, a small sample size of dataset instances was unable to provide good classification results (i.e., accuracies below 70%). Noor and Jan [6] used multiple linear regression to create a model for predicting automobile prices. Price, cubic capacity, exterior colour, date of ad posting, number of ad views, power steering, mileage in kilometres, rim type, transmission type, engine type, city, registered city, model, version, make, and model year were among the features that were included in the dataset, which was created over the course of two months. Only engine type, price, model year, and model were taken into consideration by the writers as input features after feature selection. The authors were able to attain a 98% prediction accuracy with the setting that was provided. The authors of the linked study suggested a prediction model based on a single machine learning algorithm.

## III. MOTIVATION

Our initiative is prompted by the motivation to improve the transparency, efficiency, and data-drivenness of the automobile purchasing and selling process. Both sellers and buyers have several difficulties; sellers run the danger of under-pricing or overpricing their cars, which slows down transactions, while buyers sometimes find it difficult to judge if a quoted price is reasonable. By using machine learning to provide precise price forecasts based on past performance and customer preferences, we want to make these decisions easier. Additionally, the gap in individualized suggestions is filled by combining a recommender system with the price prediction model. In addition to predicting the best pricing, this algorithm will recommend vehicles that fit the buyer's particular requirements, including favourite brands, mileage, or amenities. Price prediction and suggestion work together to simplify decision-making for both clients and businesses. By combining customization and predictive analytics, this initiative ultimately seeks to transform the used automobile industry by offering a quicker, more accurate, and more intelligent method of purchasing or selling vehicles. The authors were able to attain a 98% prediction accuracy with the setting that was provided.

## IV. METHODOLOGY

In the project, we developed a model for vehicle price prediction using the approach depicted in Figure 1. We first collected the data and conducted exploratory analysis to get a summarized view of the data, including which columns are involved, how many records there are, what types of data are involved in the dataset, whether any missing or null values are present in the dataset, and many other observations, before starting preprocessing tasks like handling missing data, categorical variables, and feature scaling. Later, feature engineering tasks were performed, including handling outliers, dividing the dataset into train and test halves, and assessing the importance of features in model building. A recommender system has also been presented in the project.

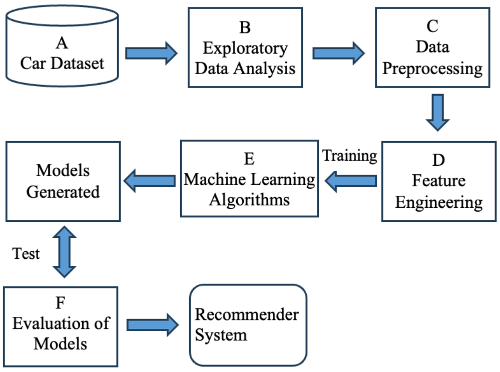


Fig. 1. Flow Chart of the Work.

### A. Dataset Description

This data used in this project is the "vehicle" dataset, taken from the source https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data. The dataset includes several information on used cars which is readily available on Kaggle Website, scrapped and compiled into a single file by Austin Reese. This data includes all relevant information that Craigslist provides on car sales in the United States such as Car price, condition, manufacturer, latitude/longitude and 22 other categories. The dataset is a structured tabular data containing 426,880 entries and a total of 26 columns representing detailed information about used car listings. The data is a mix of numerical (e.g., price, odometer), categorical (e.g., manufacturer, fuel, transmission), and textual attributes (e.g., description). It also includes geospatial data (lat, long) for car location and timestamp information (posting date). While most columns are fully populated, several features such as condition, cylinders, and drive have significant missing values. Some of the key features of the dataset are:

*a) Price:Listed price of the car in USD$.*

*b) Model:Specific car model*

*c) Manufacturer: Car brand (e.g., Ford, Toyota).*

*d) Cylinder: Number of engine cylinders ( 4, 6, 8).*

*e) Odometer: Distance traveled by car in miles.*

*f) Fuel: Type of fuel (e.g., gas, diesel, electric).*

### B. Exploratory Data Analysis

To comprehend the data, collect all relevant information, do preliminary research, identify irregularities and trends, and test the hypothesis using a variety of statistical and visual aids, exploratory data analysis, or EDA, is employed techniques [7]. Results of EDA are as follows.

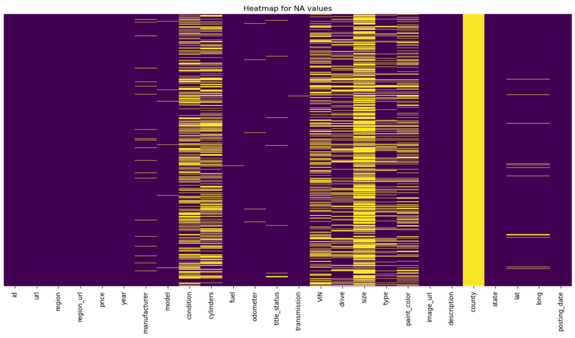


Fig. 2. Heatmap of NA values.

The heatmap visualization clearly shows the distribution of missing values in the dataset, with yellow bars indicating NA values across different features. A few key columns, such as condition, cylinders, VIN, and size, exhibit a significant amount of missing data. Notably, the county column is entirely yellow, confirming that it is completely missing for all observations. Features like drive, paint_color, and type also have a substantial proportion of missing values, which may impact the analysis. On the other hand, essential columns like id, price, url, and state have no missing values, ensuring they are reliable for analysis.

*1) Univariate Analysis*

We created a comprehensive visualization for a single numerical feature using three different types of plots: a Histogram, a QQ Plot, and a Box Plot. This allows us for an easy comparison of the distribution, spread, and presence of outliers in the specified feature.

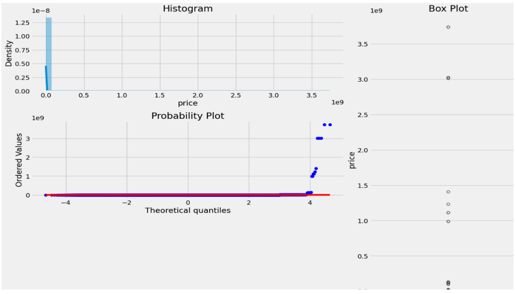


Fig. 3. Visualization of Price(Target Variable).

The histogram shows that most prices are concentrated near zero, with almost no values in the higher range. This distribution is highly right-skewed, as seen by the presence of a few extreme values far from the main cluster.

The probability plot confirms the heavy skewness by showing that most values deviate significantly from the theoretical quantiles. The dots (representing the price values) are not aligned along the straight red line, indicating that the price variable does not follow a normal distribution.

The box plot reveals the presence of extreme outliers in the price variable. There are many points above the upper whisker, indicating abnormally high values. The distribution is compressed towards the lower end, with most values near zero, while a few large values extend far upwards.

This level of skewness and presence of extreme outliers can significantly distort model training and lead to poor performance. Models like linear regression are particularly sensitive to such skewed distributions. Therefore, to make the price variable more normally distributed and to handle these outliers, data transformation techniques like log transformation or capping outliers are required.

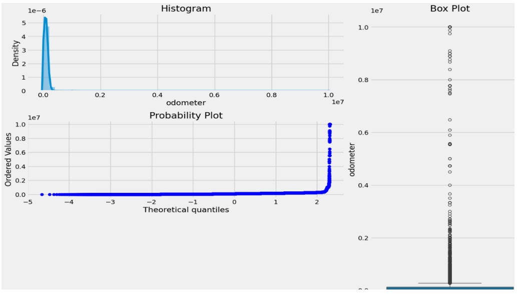


Fig. 4. Visualization of Odometer Distribution.

The histogram shows that most cars have relatively low odometer readings, indicating that most cars in this dataset have not been driven for very high mileage. There is a very sharp peak near zero and then a steep decline, suggesting a heavily right-skewed distribution. The long tail extending to the right highlights the presence of a small number of cars with exceptionally high odometer values.

The probability plot compares the distribution of the odometer variable to a theoretical normal distribution. The extreme curvature and upward shift at the right side indicate a strong deviation from normality. This skewness is a clear sign that the variable has many extreme values, making it far from normally distributed. The dots being closely packed near zero and then suddenly rising further confirm this behaviour.

The box plot visually confirms the heavy skewness and the presence of numerous outliers in the odometer variable. The box itself is squeezed at the bottom, showing that the central data points (25th to 75th percentile) are clustered tightly at low odometer values. The long whisker and the abundance of points outside the upper range show the presence of many cars with exceptionally high mileage compared to the rest of the dataset.

There is a noticeable downward trend, indicating that the car price generally decreases as the car age increases. For cars aged between 0 to 20 years, the prices are relatively stable and clustered around a certain range, showing less variability compared to older cars. After around 20 years, prices start to show a wider spread, suggesting more variation in value based on factors other than age (e.g., condition, brand, vintage appeal. There are several points with extremely high prices, which could represent classic cars or luxury models that retain or increase their value over time.

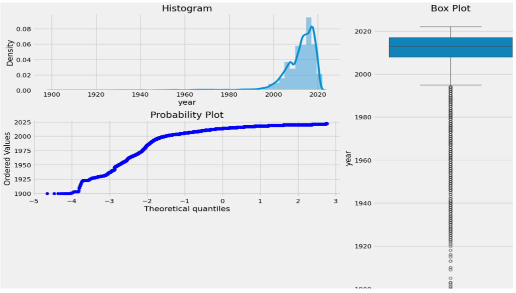


Fig. 5. Visualization of Year Distribution.

Histogram: The histogram reveals that most of the cars in the dataset are from recent years, primarily ranging between 2000 and 2020. There is a steep increase in the density starting around 2000, peaking around 2015, and then dropping sharply as it approaches 2020. This suggests that the dataset is heavily skewed towards newer cars, with very few observations for cars manufactured before 1980.

### 2) Multivariate Analysis

We explored graphically, in a 3D scatter plot, the three most important variables interrelated for our multivariable analysis: odometer, year, and price. This 3D plot is created for analysing how the different variables will interact with each other, which may be difficult to detect in a 2D plot. By mapping odometer to the x-axis, year to the y-axis, and price to the z-axis, we can see how car prices vary with both mileage and the year of manufacture simultaneously.

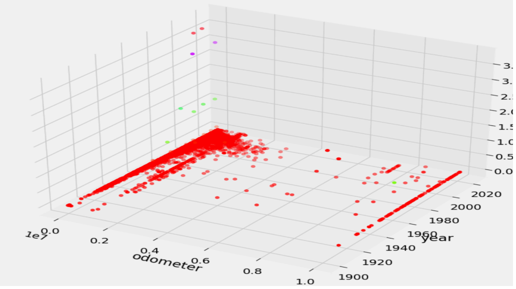


Fig. 6. Visualization of Price(z-axis), Odometer(x-axis) and Year(y-axis).

Most of the cars are clustered at very low-price values, regardless of their year or odometer readings, indicating a high number of inexpensive cars in the dataset. As odometer readings increases, prices of the car generally remain low, indicating a negative correlation between odometer and price. High-mileage cars rarely have high prices, except for a few outliers, potentially representing unique or luxury models. Cars that are manufactured before 1980 show scattered prices, with a few significantly high-priced outliers, indicating that vintage cars may have a higher resale value. Cars produced post-2000 are mostly low-priced, suggesting a more predictable and narrow price range for modern cars. The plot suggests a strong dependency of price on odometer and year.

## C. Data Preprocessing

The Preprocessing process for this project involves transforming the raw data into a clean, structured, and usable format to improve the performance and accuracy of machine learning models. Steps undertaken for the data preprocessing are listed below.

### 1) Filtering

Price: Only the data points within the 10th to 95th percentile of the price range were retained. This helps in removing extreme outliers that could skew the model's predictions, such as unreasonably high or low prices due to data entry errors or rare luxury cars.

Year: Cars manufactured after 2003 were included in the dataset. This filter ensures that very old cars, which may not be representative of the broader market, are excluded from the analysis. The year filter captures cars that are more likely to be relevant for today's market conditions.

Odometer: The odometer readings were filtered to include only values up to the 90th percentile. This reduces the impact of extreme outliers, such as cars with extremely high mileage, which could disproportionately affect model performance. Manual filters are applied to ensure the dataset is representative of typical cars in the market, excluding outliers that could distort model training.

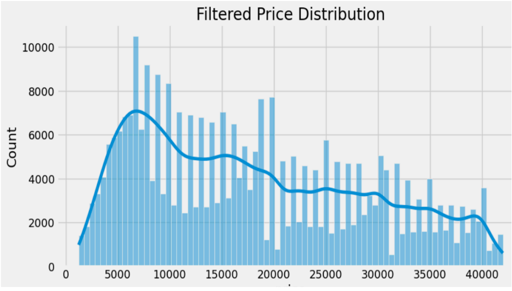


Fig. 7. Visualization of Price distribution after filtering.

### 2) Imputation

This step addresses the problem of missing values, which are common in real-world datasets. The strategy used is:

Most Frequent Value (Mode): For categorical features like Cylinders, Fuel, and Transmission, missing values were imputed with the most frequent value (mode) in each respective category.

Unknown Category: In cases where imputing the mode might lead to bias or isn't suitable, an "Unknown" category is introduced. This allows the model to treat missing data as a separate category, preserving the integrity of the dataset without making assumptions about the missing values.

### 3) Encoding and Scaling

Categorical variables like manufacturer, fuel, and transmission are encoded using techniques such as one-hot encoding. This ensures that these non-numerical features are converted into a format that machine learning algorithms can understand. Encoding helps represent categories as binary vectors or numerical labels, maintaining the relationships between features.

Numeric Features like price, year, and odometer are scaled to ensure that they are on a uniform scale. Scaling is crucial for models such as K-Nearest Neighbors (KNN) and Linear Regression, which can be sensitive to differences in feature scales. Scaling helps improve model performance by preventing features with larger magnitudes from dominating the prediction process.

By encoding and scaling both categorical and numerical features, the dataset becomes suitable for a wide range of machine learning algorithms, ensuring consistent and effective model training.

*4) Standardization:*

Standardization was used for the numerical features to ensure all values are on a similar scale. Since some columns (like mileage or engine size) might have much larger values compared to others, this can confuse the model and give undue importance to certain features. By standardizing, we made sure that all numerical features have the same range, allowing the model to treat them equally.

For categorical features, we used Label Encoding to convert them into numerical format. Unlike One-Hot Encoding, Label Encoding replaces each category with a corresponding number, which is simpler and reduces dimensionality.

### D. Feature Engineering

Feature engineering in data science involves transforming a dataset by adding, removing, combining, or modifying features to enhance machine learning model performance and accuracy. It's the process whereby our system automatically chooses relevant features for our machine learning model, based on the type of problem one is trying to solve. We do this by either the inclusion or exclusion of important features without changing them. This helps in cutting down the noise in our data and reducing the size of our input data.

*1) Feature Transformation*

Initially, we observed that the target variable price was highly skewed in the univariate analysis, which made it challenging to visualize its distribution. To address this issue, we transformed the price variable by creating a new column, price_log, using the logarithmic transformation. This helped in smoothing the distribution of prices and allowed the model to capture the relationships better. Along with this, we performed additional data preparation steps to refine the dataset further. We filtered the year feature and created a new feature called car_age using the formula: car_age = current_year - year. This was done to capture the effect of a car's age on its price more accurately. We also filtered the odometer variable to remove unrealistic values that could distort the predictions. These transformations and filtering steps significantly improved the dataset quality, ensuring that the model could learn meaningful patterns from the data. This comprehensive data preparation strategy was crucial in overcoming the initial issues and enhancing model performance, leading to more reliable and interpretable results.

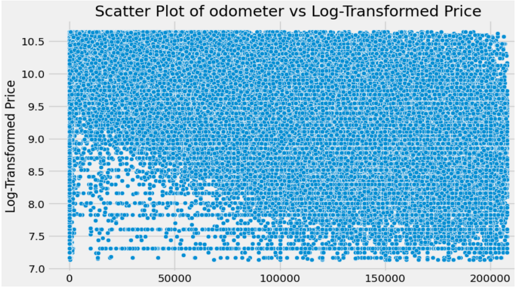
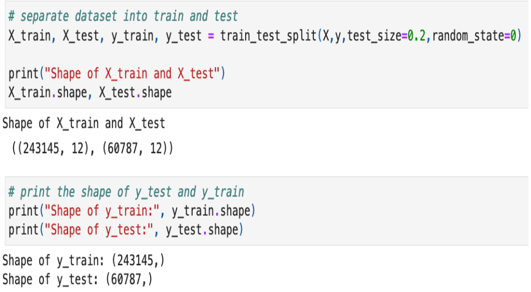


Fig. 8. Scatterplot of odometer vs Price(Log-Transformed).

*2) Splitting the data into Train and Test:*

We split the data into train and test splits such that 80% of data used for training and 20% of data is used for testing the model.



```
# separate dataset into train and test
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=0)

print("Shape of X_train and X_test")
X_train.shape, X_test.shape
```

```
Shape of X_train and X_test

 ((243145, 12), (60787, 12))
```

```
# print the shape of y_test and y_train
print("Shape of y_train:", y_train.shape)
print("Shape of y_test:", y_test.shape)
```

```
Shape of y_train: (243145,)
Shape of y_test: (60787,)
```

Fig. 9. Train Test Split.

*3) Handling Outliers:*

Handling outliers was necessary because extreme values in our data can mess up the model's predictions. Outliers can cause the model to give too much importance to unusual data points, leading to poor performance on normal cases instead of removing the outliers, which could result in loss of useful data, we chose to cap them using the Interquartile Range (IQR) method. By replacing outliers with the calculated upper bounds, we ensured that these values stayed within a reasonable range without completely removing the observations. This way, we retained all rows in our dataset while minimizing the influence of extreme values.

## V. EVALUATION AND RESULTS

### A. Linear Regression

Linear Regression is advantageous because it is computationally efficient and interpretable; each coefficient indicates the expected change in the target variable for a one-unit increase in the corresponding feature, holding all other features constant. This interpretability makes Linear Regression a useful baseline model to understand the relationship between car features and prices. Fitting the Linear Regression model to the dataset, it achieved an $R^2$ score of 0.7499, meaning that approximately 74.99% of the variability in the target variable (price_log) is explained by the features in the model. The Root Mean Squared Error (RMSE) was 0.3591, indicating the average difference between predicted and actual prices in log-transformed term. These findings suggest that while Linear Regression offers valuable insights, its linear constraints may limit its predictive power for this dataset.

### B. *Random Forrest*

Given the limitations observed in Linear Regression, Random Forest was selected as it can capture more complex interactions and reduce the risk of overfitting. Random Forest algorithm is an ensemble-based method that combines multiple decision trees to improve predictive performance. During training, Random Forest builds several individual decision trees using subsets of the data and features, which are randomly selected. This approach leverages bagging (Bootstrap Aggregating), where each tree is trained on a different random sample of the dataset. The final prediction is derived by aggregating the outputs of all trees, typically using majority voting for classification tasks or averaging for regression tasks [8].

With an $R^2$ of 0.886, the Random Forest model significantly improves on Linear Regression's 0.7499, reflecting its capability to capture complex feature interactions. This high $R^2$ score reflects the model's capability to capture complex interactions between variables, which linear models may miss. The Root Mean Squared Error (RMSE) was 0.243, showing the average difference between predicted and actual log-transformed prices. The lower RMSE suggests improved predictive accuracy compared to the Linear Regression model, making Random Forest a more reliable choice for this dataset.

*a) Cross Validation:* To validate the performance and stability of the Random Forest Regressor model, cross-validation (CV) was applied using a 3-fold split. In the baseline Random Forest model, cross-validation was implemented with the $R^2$ metric as the scoring measure. The results yielded a mean $R^2$ score of 0.870 with a standard deviation of 0.002, indicating that the model explained approximately 87% of the variance in the training data on average. This low standard deviation across folds suggests that the model's performance is consistent across different subsets of the data.

*b) Hyperparameter Tuning:* After evaluating the initial Random Forest model, further tuning was conducted to enhance performance, ensuring it could generalize well across different data subsets. The parameters tuned included max_depth, min_samples_leaf, min_samples_split, and n_estimators. The best-performing configuration identified was: max_depth: None (no restriction on depth), min_samples_leaf: 1, min_samples_split: 2, and n_estimators: 200. Using these optimized parameters, the model achieved an $R^2$ score of 0.870 on the training set during cross-validation. When applied to the test set with the optimized configuration, the final model achieved an $R^2$ score of 0.886 and an RMSE of 0.243, confirming that hyperparameter tuning provided a slight improvement in predictive accuracy.

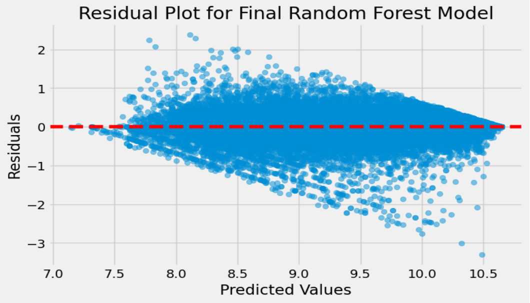


Fig. 10. Scatter plot of Residuals for final Random Forrest Model

The residual plot for the Random Forest model offers insight into prediction accuracy, displaying the distribution of errors for different price levels. The figure illustrates that the residuals are generally centered around zero, suggesting that the Random Forest model captures the underlying trend effectively. However, there is a slight fan shape, with the residual spread widening as predicted values increase. Overall, the plot suggests a strong model fit with minor variations in prediction error across the range.

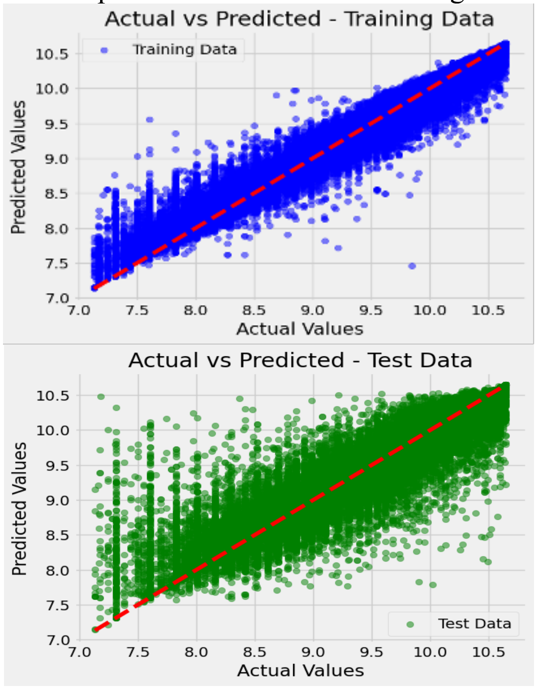


Fig. 11. Actual vs Predicted Values for Training and Test Data

To assess model consistency and accuracy, the following table compares actual versus predicted values for training and test datasets. This table presents the Actual vs. Predicted plots for both the training and test datasets, allowing a side-by-side comparison. The plot on the left illustrates the model's performance on the training data, with predicted values closely aligning with the actual values, indicating strong fit. The plot on the right shows the test data, demonstrating the model's ability to generalize to unseen data, with a similar alignment along the diagonal line. This side-by-side layout highlights the model's consistency across both datasets.

### *C. XGBoost Regressors*

To explore whether a more advanced boosting technique could improve predictive accuracy further, XGBoost was applied to the dataset. This method builds on Random Forest's strengths but includes additional regularization features. XGBoost uses an ensemble approach to build decision trees sequentially, each correcting errors from the previous trees. Its key features—regularization (L1 and L2) to prevent overfitting, shrinkage (learning rate) for gradual model updates, and column subsampling to enhance tree diversity—help make XGBoost more effective compared to traditional boosting methods. Additionally, XGBoost's optimized split-finding algorithm makes it highly efficient for high-dimensional data. These features collectively enable XGBoost to effectively capture both linear and non-linear relationships, making it suitable for regression tasks where accuracy and computational efficiency are critical [3].

After fitting the dataset, the model achieved an $R^2$ score of 0.829 and an RMSE of 0.297. While XGBoost performed well, it did not surpass the Random Forest Regressor in terms of $R^2$, likely due to the specific parameter settings and data structure used. However, XGBoost remains a strong model choice due to its efficiency, handling of non-linear relationships, and capability for tuning.

### *D. K-Nearest Neighbors*

To complement tree-based models, K-Nearest Neighbors (KNN) was evaluated as a non-parametric model that could offer unique insights, especially for localized feature effects. In its default configuration with $k = 5$ k=5, KNN achieved an $R^2$ score of 0.830 and an RMSE of 0.297 on the test data, indicating a reasonable fit to the data without any tuning. However, to better understand the model's stability, cross-validation (CV) was applied using a 3-fold split. The mean $R^2$ score from CV was 0.811 with a standard deviation of 0.002, suggesting that while KNN was relatively stable, there was room for improvement in predictive accuracy.

*a) Hyperparameter Tuning:* To enhance KNN's performance, hyperparameter tuning was conducted using GridSearchCV to systematically search for the optimal settings. The parameter grid included different values for n_neighbors (3, 5, 7, 9, and 11) and two weight options: 'uniform' and 'distance'. These settings allowed the model to adjust the number of neighbors considered for each prediction and the weighting scheme based on distance. The tuning process yielded an improved $R^2$ score of 0.856 on the training data with the best combination of parameters. With these optimized parameters, the KNN model was then evaluated on the test set, where it achieved a final $R^2$ score of 0.872 and an RMSE of 0.257. This improvement in accuracy confirms the effectiveness of tuning, as the optimized KNN model was able to capture more variance in the target variable with reduced error compared to the initial configuration.

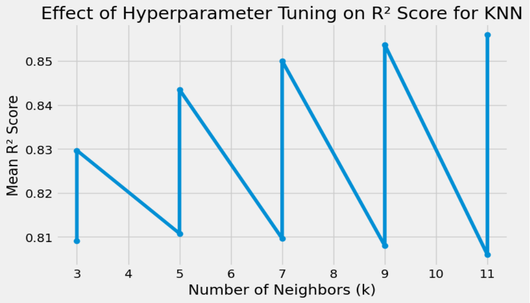


Fig. 12. Line plot to show the effect of Hyperparameter tuning

To optimize the K-Nearest Neighbors (KNN) model, we explored different values of $k$ to understand their impact on predictive accuracy, as shown in the following plot. The figure above shows the relationship between the number of neighbors $k$ and the mean $R^2$ score of the K-Neighbors Regressor (KNN) model during cross-validation. As seen in the plot, the $R^2$ score fluctuates as k varies, with the highest mean $R^2$ achieved at k=11 (approximately 0.856). This suggests that the model achieves optimal predictive accuracy when considering 11 neighbors, while lower and higher values of $k$ k result in reduced performance.

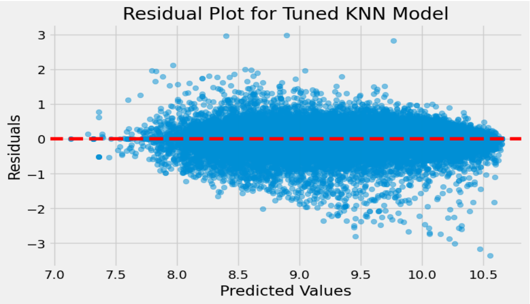


Fig. 13. Scatter plot of Residuals for KNN tuned Model

The figure shows that residuals are centered around zero, suggesting that the model does not exhibit systematic error across the predicted values. This distribution indicates that the model captures the underlying patterns in the data effectively. Additionally, the spread of residuals remains relatively consistent across the range of predicted values, from approximately 7 to 10.5, indicating stable performance across different price levels.

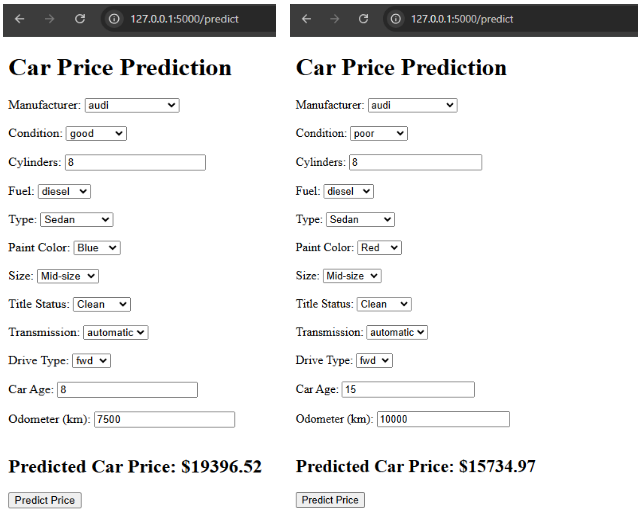
After evaluating and tuning each model, the performance results reveal significant differences in accuracy and error metrics. To consolidate these findings, a summary table is presented below, comparing the initial and final R² scores, RMSE values, and key parameters for each model. Model performance Metrices and results

| Model | Initial $R^2$ (Test) | Initial RMSE (Test) | Cross-Validation (Mean $R^2$) | Cross-Validation (Std Dev) |
|---|---|---|---|---|
| Linear Regression | 0.7499 | 0.3592 | - | - |
| K-Neighbors Regressor | 0.8301 | 0.2961 | 0.811 | 0.002 |
| Random Forest Regressor | 0.8854 | 0.2432 | 0.87 | 0.002 |
| XGBoost Regressor | 0.8288 | 0.2972 | - | - |

| Model | Tuned Parameters | Final $R^2$ (Train) | Final $R^2$ (Test) | Final RMSE (Test) |
|---|---|---|---|---|
| Linear Regression | - | - | 0.7499 | 0.3592 |
| K-Neighbors Regressor | n_neighbors=11,<br>weights='distance' | 0.856 | 0.872 | 0.257 |
| Random Forest Regressor | max_depth=None,<br>min_samples_leaf=1,<br>min_samples_split=2,<br>n_estimators=200 | 0.87 | 0.886 | 0.243 |
| XGBoost Regressor | - | - | 0.829 | 0.297 |

The summary table outlines the performance metrics for the Linear Regression, K-Nearest Neighbors (KNN), and Random Forest models. Each row displays the initial test R² and RMSE values before tuning, followed by cross-validation results (mean R² and standard deviation) to assess model stability. The 'Tuned Parameters' column lists the optimal settings found for each model, and the final R² and RMSE values indicate improvements after hyperparameter tuning. Notably, the Random Forest model achieved the highest test R² score of 0.886 with an RMSE of 0.243, suggesting it is the most accurate model for predicting car prices in this dataset. The KNN model also saw improvements after tuning, reaching a final R² of 0.872, although it performed slightly below the Random Forest model.

To make these predictive models practically accessible to users, a car price recommender system was developed. This system allows users to input various details about a vehicle to receive a real-time price prediction based on the model's analysis. The interface enables users to interact with the prediction model by entering key car attributes, which are then processed to generate an estimated price. The following figure showcases the user interface of the recommender system and demonstrates how different car attributes influence the predicted price.



127.0.0.1:5000/predict

## Car Price Prediction

Manufacturer: audi
Condition: good
Cylinders: 8
Fuel: diesel
Type: Sedan
Paint Color: Blue
Size: Mid-size
Title Status: Clean
Transmission: automatic
Drive Type: fwd
Car Age: 8
Odometer (km): 7500

**Predicted Car Price: $19396.52**

127.0.0.1:5000/predict

## Car Price Prediction

Manufacturer: audi
Condition: poor
Cylinders: 8
Fuel: diesel
Type: Sedan
Paint Color: Red
Size: Mid-size
Title Status: Clean
Transmission: automatic
Drive Type: fwd
Car Age: 15
Odometer (km): 10000

**Predicted Car Price: $15734.97**

Fig. 14. Interface for Car Price Prediction Using the Recommender System

In the car price recommender system, users first enter relevant details about the car, including manufacturer, condition, number of cylinders, fuel type, car type, color, and odometer reading. Once all necessary attributes are selected, the system processes this input and returns a predicted price.

As shown in the figure, two different scenarios were tested. On the left, a car in good condition with a relatively lower odometer reading yielded a higher predicted price which is $19396.52. In contrast, changing the condition to 'poor' and increasing the odometer reading and the age of the car on the right produced a lower predicted price which is $15734.97. This outcome illustrates how the system responds to various car attributes, simulating market conditions where factors like age, mileage, and condition significantly influence the price.

# VI. DISCUSSION

## *A. Key Insights*

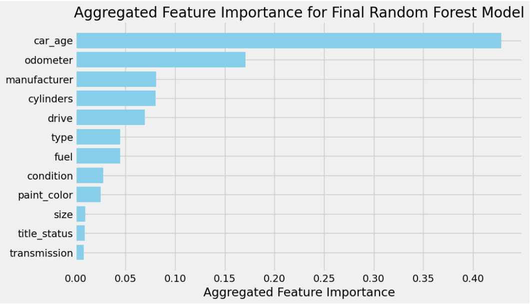


Fig. 15. Aggregated Feature Importances

Through feature importance analysis in the Random Forest model, car age and odometer emerged as primary factors in determining used car prices. These insights reinforce the predictive focus on key attributes that buyers and sellers prioritize. Odometer readings also play a critical role, as higher mileage generally reduces a vehicle's resale value. Other important features include manufacturer, cylinders, and drive type, indicating that brand reputation, engine configuration, and drivetrain characteristics significantly influence the market price. Features like fuel type and condition also contribute to price variations, reflecting buyer preferences for specific fuel efficiency and car quality attributes in the secondary market.

### *B. Limitations and Challenges*

The project encountered several challenges, particularly in handling missing values and tuning model parameters. A substantial portion of the dataset had missing values in key features, which posed a challenge for maintaining data quality. While many reference studies opted to remove entries with missing values, we chose to impute missing values to retain more data. This approach required careful selection of imputation methods to prevent data bias, as improper imputation could distort relationships within the data. Another challenge was the time-intensive process of hyperparameter tuning, especially for the Random Forest and K-Neighbors Regressor models. With multiple parameters to adjust, tuning required significant computational resources, which emphasized the importance of efficiency when working with large datasets and complex algorithms.

### *C. Future Enhancements*

To further improve the model and recommender system, several enhancements could be implemented:

- Refining Feature Selection
- Incorporating More Complex Models
- Expanding the Feature Set
- Enhancing the Recommender System

## VII. CONCLUSION

This study successfully developed a predictive model and recommender system to estimate used car prices, leveraging multiple machine learning algorithms, including K-Neighbors Regressor, Random Forest, and XGBoost. Through an iterative process of model selection, hyperparameter tuning, and validation, Random Forest emerged as the most effective algorithm, achieving the highest accuracy and lowest error rates. Significant features, such as mileage, car age, and manufacturer, were identified as primary influencers on price, aligning with established market trends and providing valuable insights for stakeholders.

## VIII. REFERENCES

[1] E. Jozkowsi, "Global Car & Automobile Manufacturing - Market Size, Industry Analysis, Trends and Forecasts (2024-2029)," March 2024. [Online]. Available: https://www.ibisworld.com/global/market-research-reports/global-car-automobile-manufacturing-industry/#IndustryOverview.

[2] Data, CIEC, "United States Motor Vehicle Production," [Online]. Available: https://www.ceicdata.com/en/indicator/united-states/motor-vehicle-production.

[3] A. AlShared, "Used Cars Price Prediction and Valuation using Data Mining Techniques," 12 2021. [Online]. Available: https://repository.rit.edu/theses/11086/.

[4] B. I. D. K. Z. M. J. K. Enis Gegic, "Car Price Prediction using Machine Learning Techniques," *TEM Journal,* vol. 8, no. 1, pp. 113-118, 2019.

[5] S. Pudaruth, "Predicting the Price of Used Cars using Machine Learning Techniques," *International Journal of Information & Computation Technology,* vol. 4, pp. 753-764, 2014.

[6] S. J. Kanwal Noor, "Vehicle Price Prediction System using Machine Learning Techniques," *International Journal of Computer Applications ,* vol. 167, 2017.

[7] C. L. L. S. B. K. K. Chejarla Venkat Narayana, "Machine Learning Techniques To Predict The Price Of Used Cars: Predictive Analytics in Retail Business," *IEEE Xplore,* 2021.

[8] A. Hassan, "Navigating the Used Car Marketplace," [Online]. Available: https://www.kaggle.com/code/abdalrahmanhassan/navigating-the-used-car-marketplace.

[9] S. J. Kanwal Noor, "Vehicle Price Prediction System using Machine Learning Techniques," *International Journal of Computer Applications,* vol. 167, 2017.

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**