Mohammadreza Tabatabaei 24831904

1.Data Understanding and Exploration

Introduction:

The given report is developed to predict the car price on different features based on its mileage, the year in which it has been registered, fuel, standard of the make of the car, condition, the type of its body and its color. Linear Regression, k-Nearest Neighbors, and Decision Tree were employed, and models were performed for their skill. There has been the most advanced application of the preprocessing on the dataset: Power Transformer and Simple Imputer that handled data to scale after missing value handling.

1.1. Data Understanding

This dataset offers anonymized auto sales advertisements that detail the vehicle's attributes and pricing. Below is a dataset summary:

Total Rows: 402,005

Total Columns:12

Target Variable: Price

- **Numerical Features:** mileage, year_of_registration.
- **Categorical Features:** fuel_type, standard_make, body_type, vehicle_condition, standard_colour.

Descriptive Statistics of Numerical Features

	mileage	year_of_registration	price
count	401878.0	368694.0	402005.0
mean	37743.59565589557	2015.006205688186	17341.965798932848
std	34831.724017746776	7.962667440560224	46437.46095064824
min	0.0	999.0	120.0
25%	10481.0	2013.0	7495.0
50%	28629.5	2016.0	12600.0
75%	56875.75	2018.0	20000.0
max	999999.0	2020.0	9999999.0

Table 1: *Descriptive Statistics of Numerical Features*

Distribution analysis

I analyzed numerical features such as Year, Mileage, and Price to understand the distributions in the data. Here are some important findings from these analyses:

• The majority of the cars are inexpensive, but a few high-end models stand out due to the positive skew in the price distribution.

• The distribution of mileage among all the cars shows most drive a car fairly with very few really well-worn cars sticking out, whereas the year distribution implies the majority of cars have fallen under newer models.

The figures below illustrate the above observations:

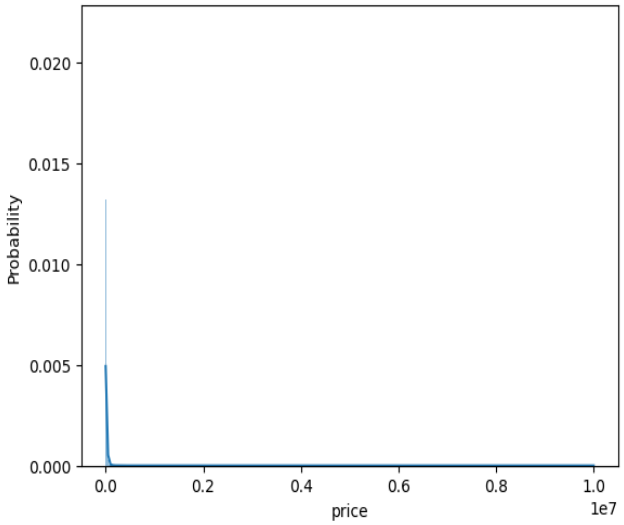

Figure 1: *Histogram of the price of cars without filters*

- The histogram is somewhat misleading because most of the cars are at the low end, but few cars have extremely high prices.
- Outliers are produced by these high prices, which could skew analysis and predictive model performance. **I will deal with the outliers in section 2-1 and show the filtered histogram in that section.**

Violin Plot Price

To visualize this distribution in further depth, a violin plot that incorporates density and outliers can be made:

Unfiltered Violin Plot:

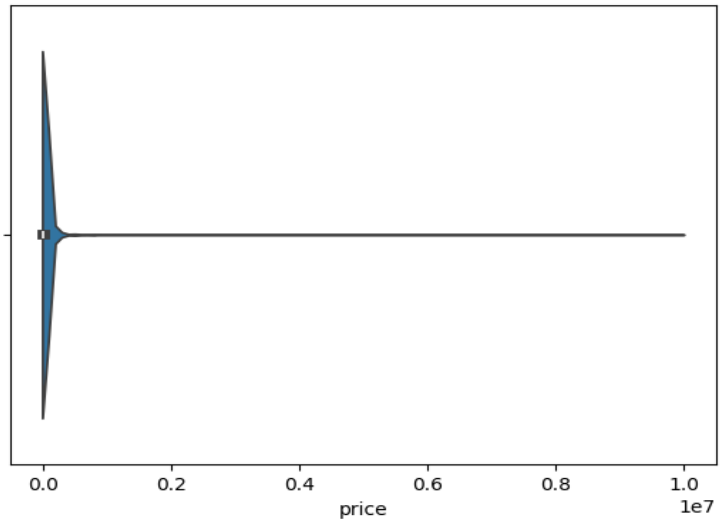

Figure 2: *Violin plot of car prices without filter*

- This plot demonstrates how the price distribution is skewed to the right.

I will deal with the outliers in section 2-1 and show the filtered violin plot in that section.

Numerical Features' Distribution:
Mileage: The distribution is pulled toward lower high-mileage cars, in that most of the automobiles have mileage under 100,000.

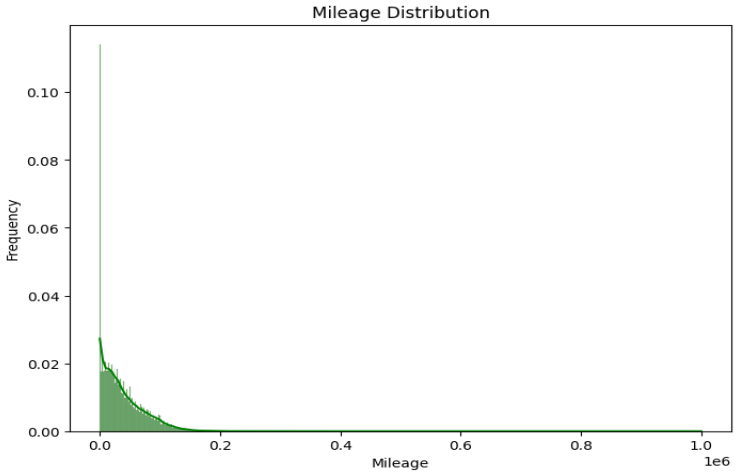

Figure 3: *Mileage Distribution*

Year: The majority of cars are relatively new, with the last five to ten years accounting for the majority of them.

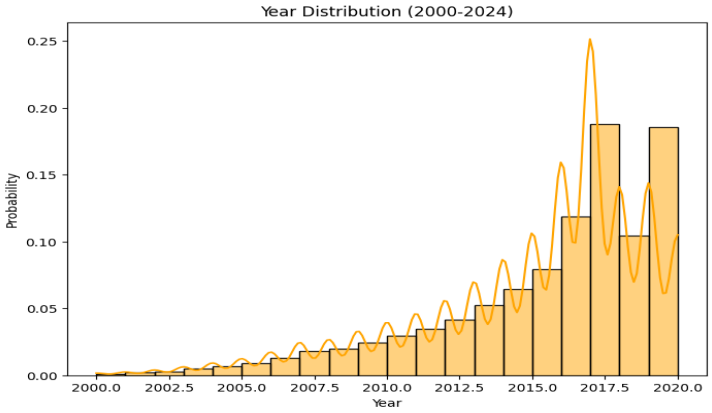

Figure4: *Year Distribution (2000-2024)*

.....

1.2. Analysis of Predictive Power of Features

1.2.1. Examining the relationships between features and price

Visualizations were used to compare price against key numerical and categorical attributes to identify features that influence car price.

To analyze the relationship between mileage and price, the following scatter plot was used:

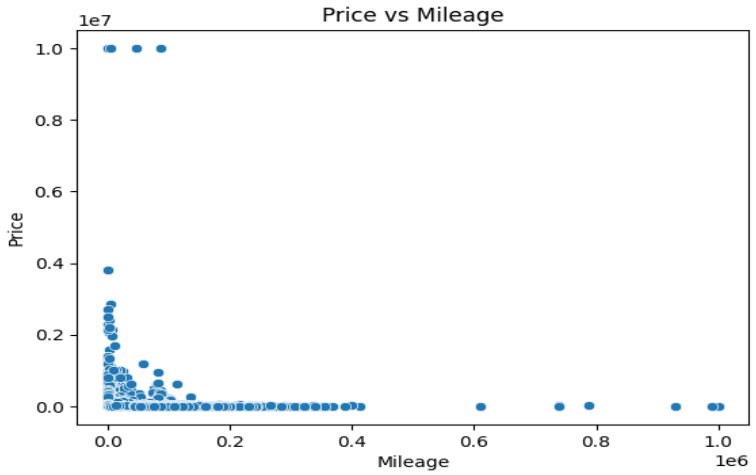

Figure 5: *Scatter Plot of Price vs Mileage*

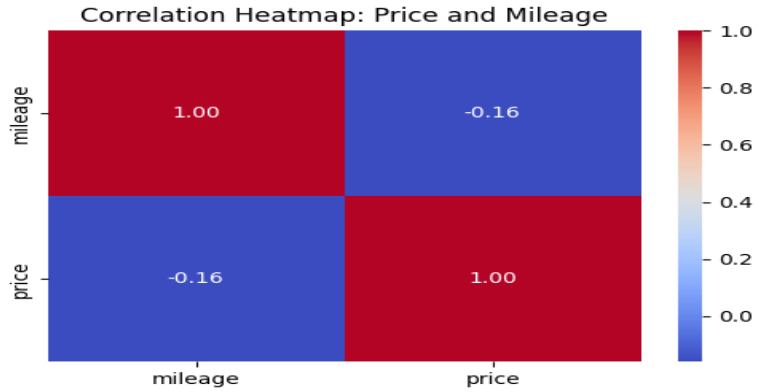

Figure 6: *Correlation Heatmap of Price vs Mileage*

Not surprisingly, there is a negative correlation between Mileage and Price: higher mileage normally translates into lower price, as dictated by the usual depreciation pattern.

1.2.2. Price vs Year

A scatter plot was used to see how year (manufacturing year) affects price:

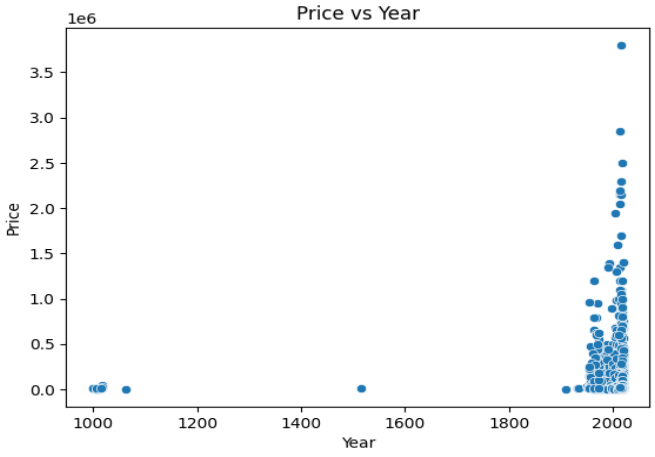

Figure 7: *Scatter Plot of Price vs Year of Registration*

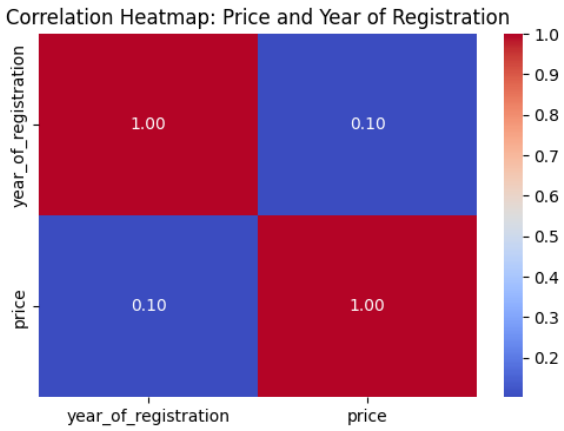

Figure 8: *Correlation Heatmap of Price vs Year of Registration*

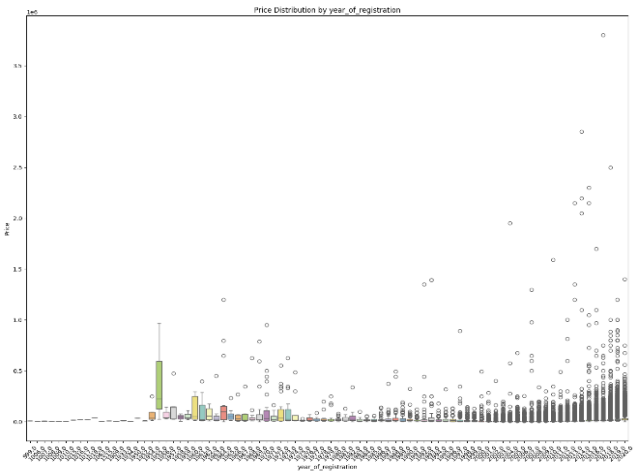

Figure 9: *Price Distribution by Year of Registration*

It is showed that newer cars tend to be more expensive. We can see a positive relationship between year of manufacture and price, which is common in the used car market.

The Relationship Between Categorical Variables and Price

In this section, the relationship of price-the target variable-with the categorical attributes is explored. A summary of how multiple categories influence the prices of cars can be seen from the following visualizations.

Note that the graphs below, as will be described in Section 2.1 (Handling Missing Values and Encoding Categorical Characteristics) are based on the categorical characteristics after encoding.

1.2.6. Price vs Standard Color (Target Encoded)

The scatter plot displays the color of the car versus the price, showing the distribution and dispersion of prices for given colors. It emphasizes some outliers and price trends according to color. Complementary is the bar plot showing for each color category its average price. It shows more precisely which colors tend to have higher averages, such as black or white, while other colors may be in the lower price ranges. This encoding process has taken on the task of assigning numerical values to colors based on their respective average prices. For example: - 'Black' is encoded as 1.2 - 'White' is encoded as 1.1 - 'Red' is encoded as 0.9

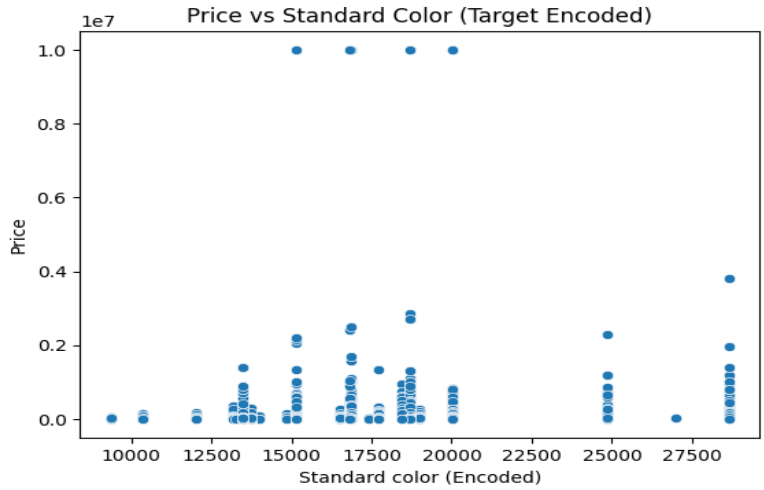

Figure 10: *Scatter Plot of Price vs Standard Color (Target encoding)*

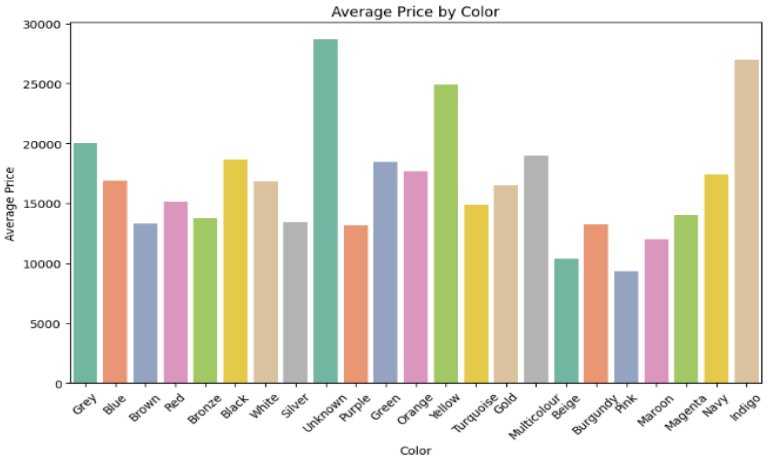

Figure 11: *Bar Plot of Average price vs Standard Color (Target encoding)*

1.2.4. Price vs standard_make (Target Encoded)

1.2.4.1. Target Encoding for standard make (brands)

Target Encoding of the car brand feature standard_make was done by calculating the mean price for each brand. This encoded categorical feature then became a numerical one with its relationship to price intact. The scatter plot shows the correlation between the encoded car brands and their respective prices.

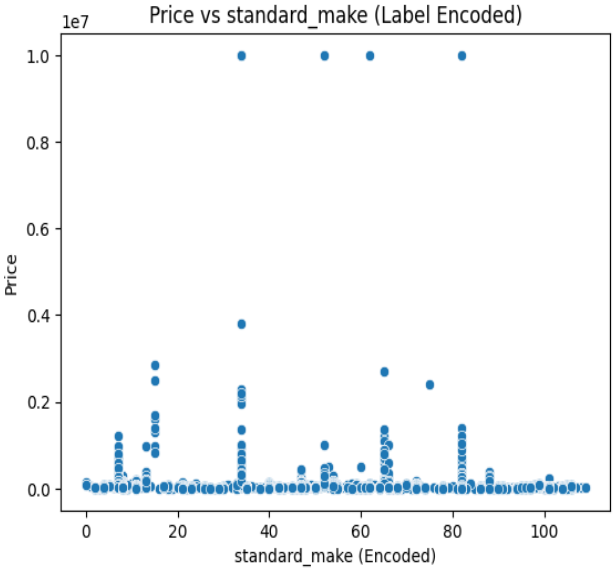

Figure 12: *Scatter Plot of Price vs Standard make (Target encoding)*

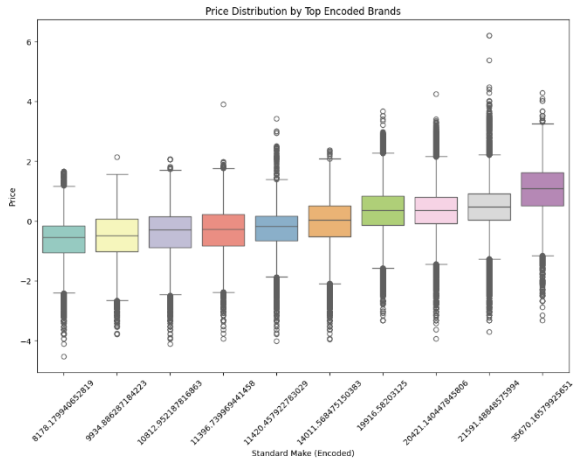

Figure13: *Box Plot of Price vs Standard Make (Target encoding)*

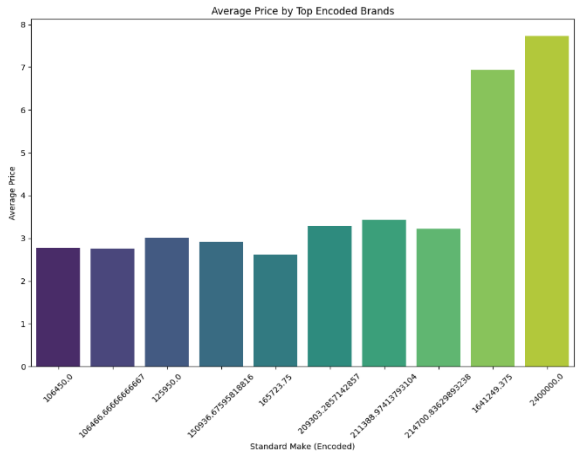

Figure 14: *Bar Plot of Price vs Standard Make (Target encoding)*

1.2.3. Price vs Fuel Type

Fuel Type is a categorical variable, so we first encoded it to numerical values using Label Encoding. Then we created a scatter plot to examine the relationship between price and the coded fuel type.

Label Encoding for Fuel Type

By using label encoding, the feature of fuel type, which originally contained categorical values such as "Bi Fuel", "Diesel" and "Electric", was converted to numeric values. The following is the mapping of Fuel Type categories to numerical values:

'Bi Fuel': 0 ,'Diesel': 1, 'Diesel Hybrid': 2 and etc.

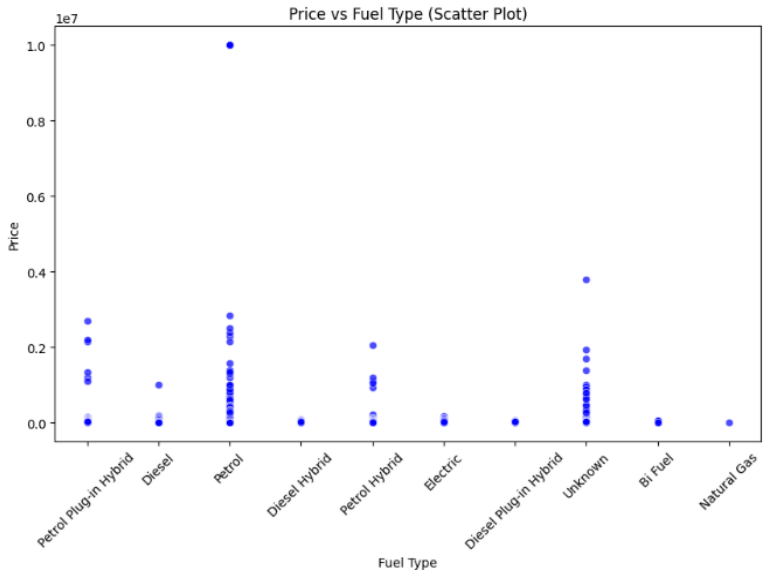

Figure 15: *Scatter Plot of Price vs Fuel Type (Label Encoded)*

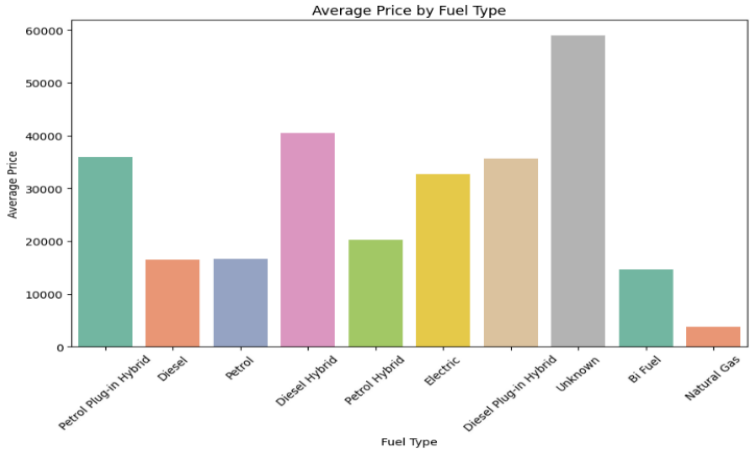

Fig 16: *Average Price by Fuel Type (lable Encoding)*

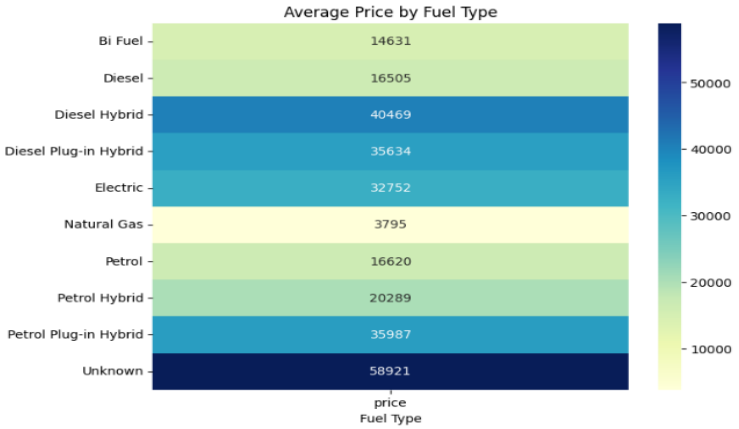

Fig 17: *Average Price by Fuel Type (lable Encoding)*

1.2.5. Price vs Vehicle Condition

Afte using label encoding, the feature of Vehicle Condition, which originally contained categorical values was converted to numeric values. The following is the mapping of Vehicle Condition categories to numerical values:

'New' :0, 'Used':1

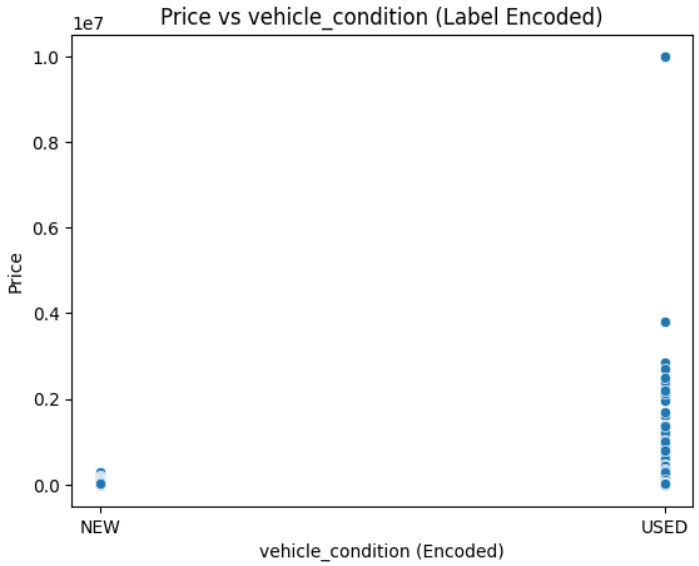

Figure 18: *Scatter Plot of Price vs Vehicle condition (Label Encoded)*

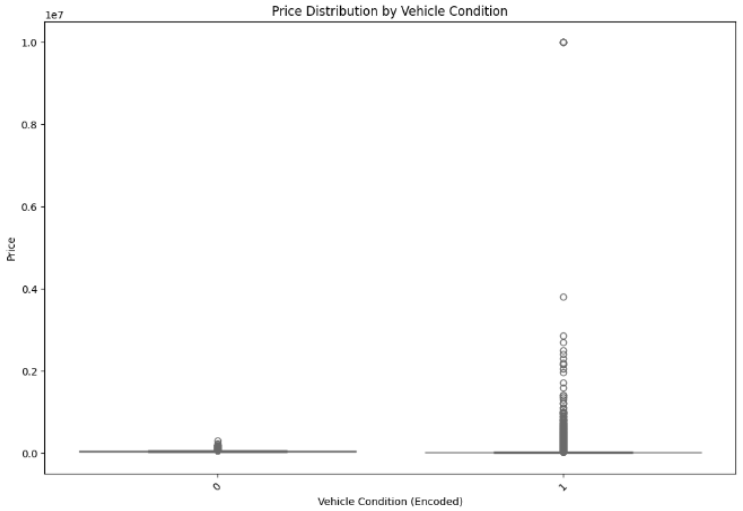

Figure 19: *Box Plot of Price vs Vehicle condition (Label Encoded)*

1.2.6. Price vs Body Type

Afte using label encoding, the feature of Body Type, which originally contained categorical values such as "saloon", "SUV" and " Hatchback", was converted to numeric values. The following is the mapping of Body Type categories to numerical values:

'SUV':13, 'Saloon':14, 'Hatchback':7, etc.

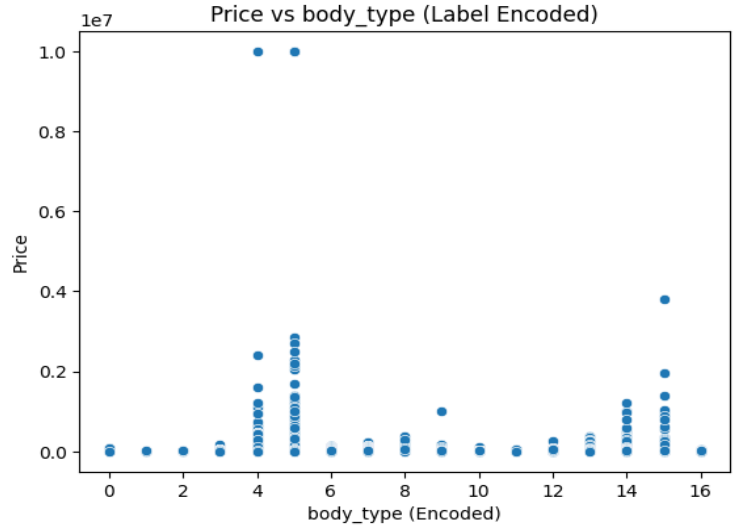

Figure 20: *Scatter Plot of Price vs Body Type(Label Encoded)*

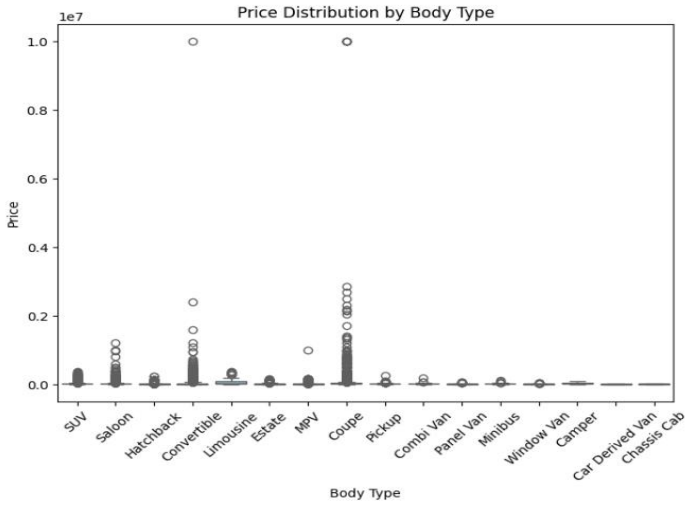

Figure 21: *Box Plot of Price vs Body Type(Label Encoded)*

.....

1.3. Data Processing for Data Exploration and Visualization

This stage, according to the assessment outline, covers the preparation of the dataset for exploratory data analysis and visualization. The key objective is basic data processing to prepare the data for efficient visualization. To ensure that the dataset is prepared for understandable and informative visualizations, this section focuses on making short-term changes and creating features, while Section 2.1 completes comprehensive and long-term preparation.

New features were created that would help in research and give a better understanding of the data.

1.3.1. Price Categories: This is picturing logical price bins of the column of prices, divided into three categories: "Low," "Medium," and "High." It helps in better categorization of vehicles and analysis of price distribution.

The bar chart below shows the distribution of cars in three price ranges: Low, Medium, and High. As can be seen, most cars in the dataset fall within the low-price range.

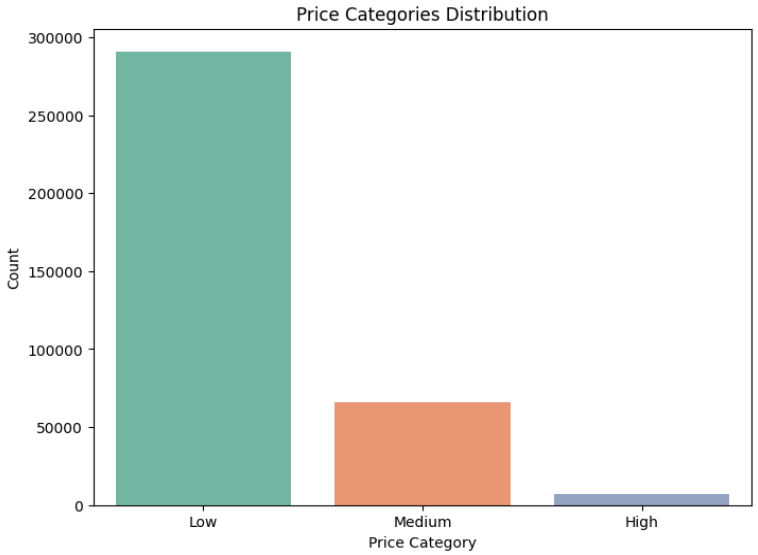

Fig 22. *Price Categories Distribution*

1.3.2. Price Distribution By Year Group: The box plot below shows the price discrepancies between cars from different periods and how automobile costs change within different year groups. Based on the box plot, the average price of cars registered between 2011 and 2020 is the highest.

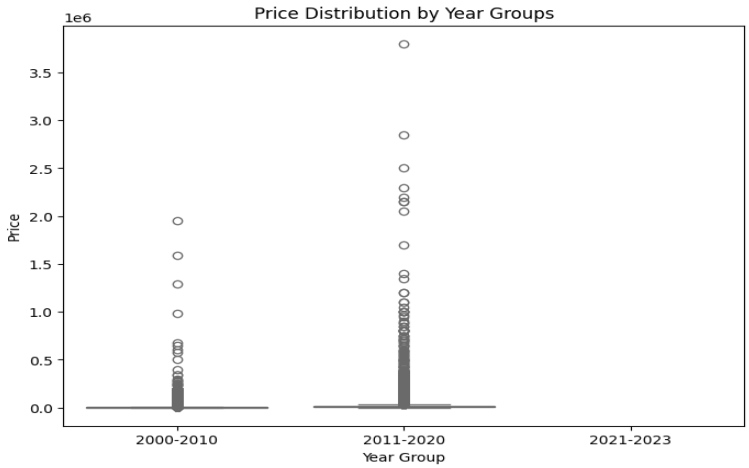

Fig 23. *Price Distribution by Year Groups*

1.3.3. The Number of Null Values in Each Column: The following bar chart provides a view into which columns may need special attention because it depicts the count of null values across each column. As seen in the bar chart, among all the columns, the year_of_registration column has the maximum number of missing values. This again puts forth that this feature needs to handle null values even further.

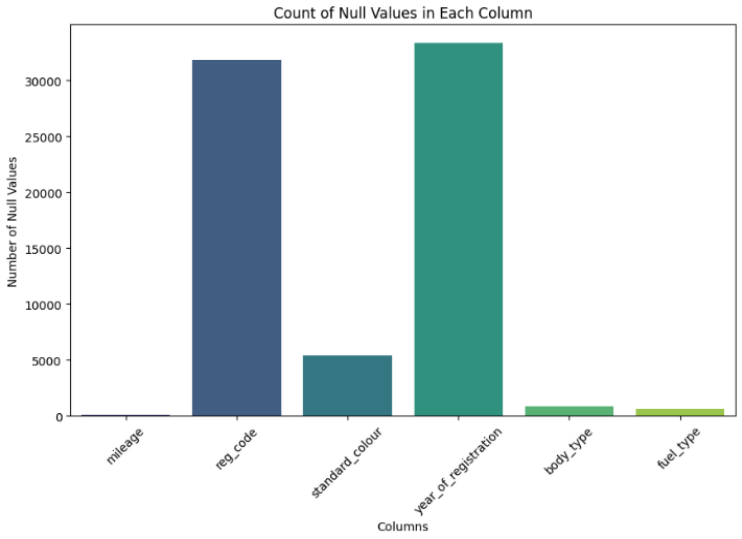

Fig 24. *The Number of Null Values in Each Column*

1.3.4. There are two types of cars in the dataset: used and new. I designed the bar chart below based on this classification. The bar chart shows that the number of used cars is much higher than new cars.

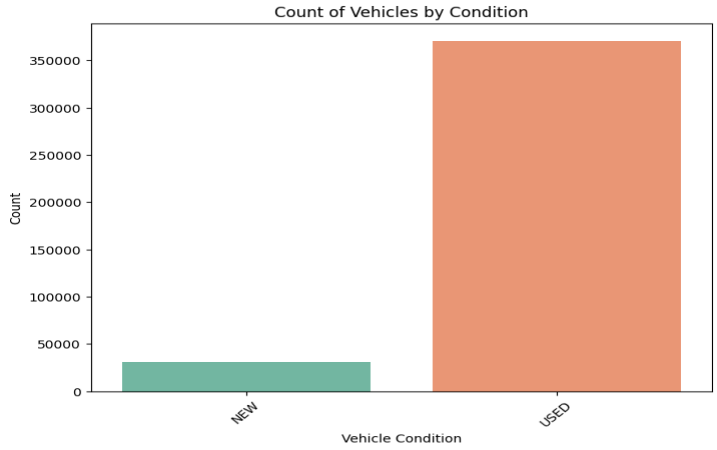

Figure 25: *Bar Chart: Vehicle Condition Counts*

1.3.5. Distribution of Prices by Body Type:

Car prices are displayed using a box plot according to body type (e.g., sedan, SUV, hatchback). The median price of SUVs is typically greater than that of sedans and hatchbacks.

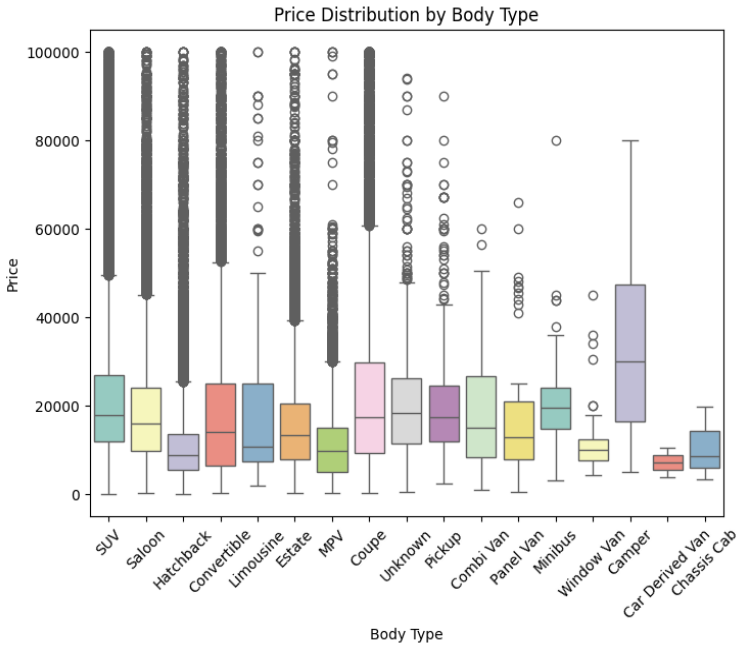

Figure 26. *Box Plot: Price vs Body Type*

.....

2. Data Processing for Data Exploration and Visualization

2.1. Data Cleaning for Better Analysis

2.1.1. Managing Missing Values

Techniques Used:

Numerical Columns: To avoid data skewness, missing values were imputed with the median of that variable. By taking the median, we make sure that it is resistant to extreme outliers and that this value remains representative because the integrity of the distribution function is not affected by it. Missing values in categorical columns were replaced by "Unknown." I replaced with 'Unknown' for protecting our data, avoiding bias, and allowing the model to treat missing values as a separate category during training.

Tools Used: I used **SimpleImputer** for numerical columns to make sure that there is consistency across the dataset.

2.1.2. Categorical encoding:

1- Car Brands, Car Colors: I used **target encoding** to color column and brand column, mapping the average car price for each color and each brand to its respective category.

2-**Fuel Type, Body Type, and Vehicle Condition:** These columns were encoded using **Label Encoding** to convert them into numerical values.

Please note that all the plots of these features after encoding were shown in section 1-2 because that section was about Analysis of Predictive Power of Features. As a result, I put the plots related to these features there. You can see them now.

.....

2.2. Feature Engineering, Data Transformations, Feature Selection

2.2.1. Feature Transformation

Power Transformation: I applied power transformation to normalize the skewed distributions of numerical features such as mileage, year of registration, and price. This step improved the stability of the models and ensured a better fit during training.

2.1.2. A reasonable threshold was used to exclude outliers from the Price column. To ensure that the data represents real car costs, vehicles costing over $100,000 were not included.

The following histogram of the transformed price data showed a more normal-like distribution.

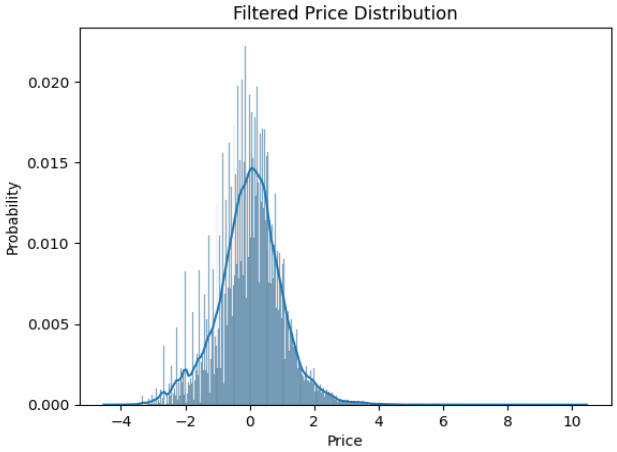

Figure 27. *Histogram of transformed Car Prices*

2.2.3. filtered Violin Plot: The violin plot of the transformed prices gives the distribution of car prices after applying the power transformer.

Figure 28. *transformed car violin price (0-100000)*

2.2.4. Transformed Price vs Mileage

The scatter plot below shows the relationship between car mileage and their transformed prices, highlighting trends and potential outliers.

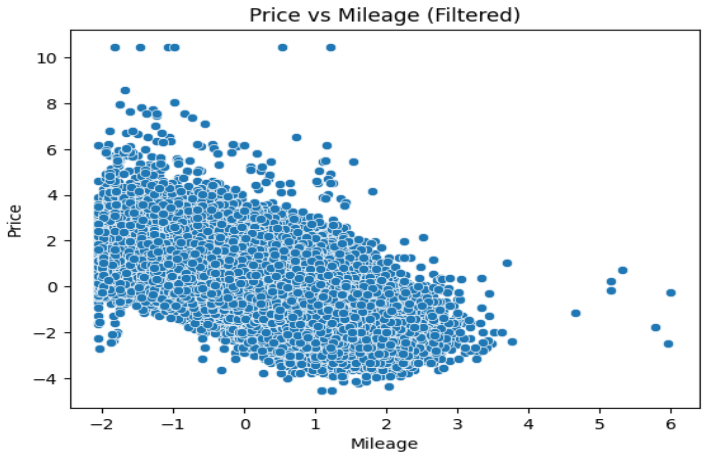

Fig 29. *Scatter plot of transformed Price vs Mileage*

2.1.6.Scaling Features

Feature scaling was used to enhance machine learning models' performance, especially distance-based models like k-NN. The numerical features (Mileage, Year, and the encoded category features) were scaled using a StandardScaler to have a mean of 0 and a standard deviation of 1.

2.2.5. Selection of Features and Targets

The features listed below were chosen to train the model:

- **Mileage**
- **Year**
- **Fuel Type (Encoded)**
- **Standard Make (Encoded)**
- **Color (Encoded)**
- **Body Type (Encoded)**
- **Vehicle Condition (Encoded)**

For regression modeling, the price goal variable was kept.

2.2.6. Split Train-Test

To assess the models' performance, the dataset was divided into training and test sets:

• Training Set: Machine learning models are trained using this set.
• Test Set: This is utilized for the last assessment.

The models were trained using 80% of the data. 20% served as the test set for assessing the model. This keeps an independent dataset for performance testing while guaranteeing that the models are trained on enough data.

Fig 30. *Train and Test Split Visualization*

.....

3. Model Building

3.1. Algorithm Selection

The following three algorithms were chosen for this project to predict the prices of cars based on the features of Mileage, Year, and Fuel Type:

1. **Linear Regression**: The most straightforward of models, assuming a direct connection between the features and Price which is the target variable.
2. **k-Nearest Neighbors (k-NN)**: The model which predicts the price based on the data points closest to a query point can capture non-linearities.
3. **Decision Tree**: A model which dichotomizes the data into discrete segments based on feature values and can capture interaction effects complex in nature.

I chose these three algorithms because of their simplicity and due to different approaches to modeling data, which allows making a wide comparison performance.

Cross-Validation

I evaluated the baseline performance of our models using K-fold Cross-validation before the optimization process. The method works by dividing the data into k subsets, or folds, training the model on k-1 of the folds, and testing the model on the remaining fold. This is repeated k times, computing average performance.

Results from Cross-Validation:

- **Linear Regression**: Cross-Validated R^2: 0.6842
- **k-Nearest Neighbors**: Cross-Validated R^2: 0.8713
- **Decision Tree**: Cross-Validated R^2: 0.8234

Based on these results **k-Nearest Neighbors** did best during cross-validation.

3.2. Hyperparameter Tuning

In regard to model performance improvements, the following hyperparameters have been optimized using Grid Search:

- **Linear Regression**: fit_intercept: True
- **k-Nearest Neighbors (k-NN)**:
- n_neighbors: 7
- weights: uniform
- **Decision Tree**:
- max_depth: None
- min_samples_split: 10

Rank	n-neighbors	weights	Mean Test Score	Std Test Score
1	7	uniform	0.8848	0.0015
2	9	uniform	0.8838	0.0014
3	5	uniform	0.8836	0.0015

Rank	Max Depth	Min Samples Split	Mean Test Score	Std Test Score
8	None	10	0.8617	0.0010
9	None	5	0.8430	0.0017

Table2,3. *Summary of Top Hyperparameter Configurations and Their Rankings*

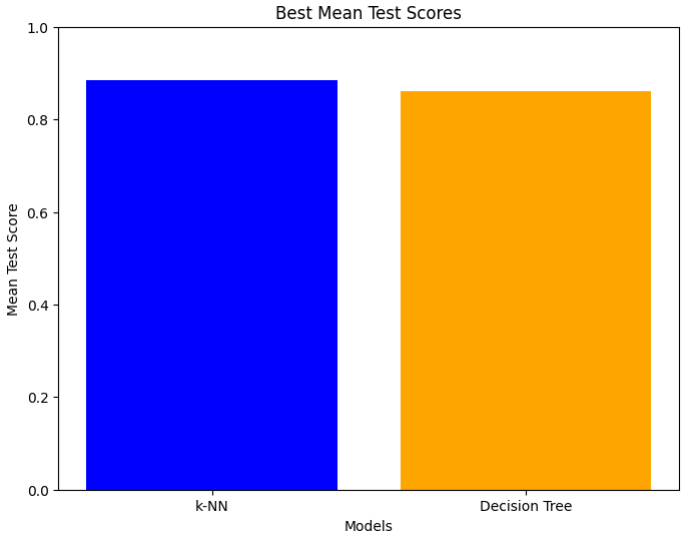

Fig31. *Visualization of Test Scores of the Top-ranked Models*

.....

4. Model Evaluation and Analysis

Performance evaluation for each model was based on three key metrics:

- **Mean Squared Error (MSE)**: It stands for Mean Squared Error, which calculates the average of

the squared differences between actual and predicted. Lower MSE means better performance.

- **Mean Absolute Error (MAE)**: It shows the average absolute difference between predicted values and actual values.
- **R^2 Score**: The proportion of variation in the target variable is described by the model. The closer the value to 1, the better the fit.

The above-mentioned metrics were used for the final comparison of the models.

4.1.1. Model Performance Overview

The table below shows the performance of the model after optimization.

Model	MAE	MSE	R^2 Score
Linear Regression	0.38	0.30	0.6919
k-NN	0.23	0.11	0.8875
Decision Tree	0.25	0.14	0.8618

Table4. *performance of the model after optimization*

4.1.2. Evaluation

1. Linear Regression: Among the three models, Linear Regression had the poorest performance among them with the highest MSE and lowest R2 score.

2. k-Nearest Neighbors (k-NN): Highest R2 score and the lowest MSE among the three performed the best.

3.Decision Tree: Outperformed linear regression on all accounts with lower MSE and MAE.

4.1.3. Model Performance Visualization

To give a better idea of each model's performance, I show the Actual vs. Predicted figures. These figures show how closely the expected values match the starting values.

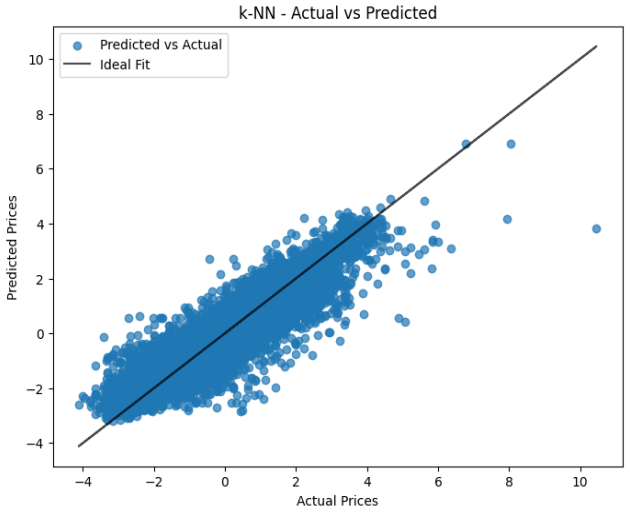

Figure 32. *k-NN - Actual vs Predicted*

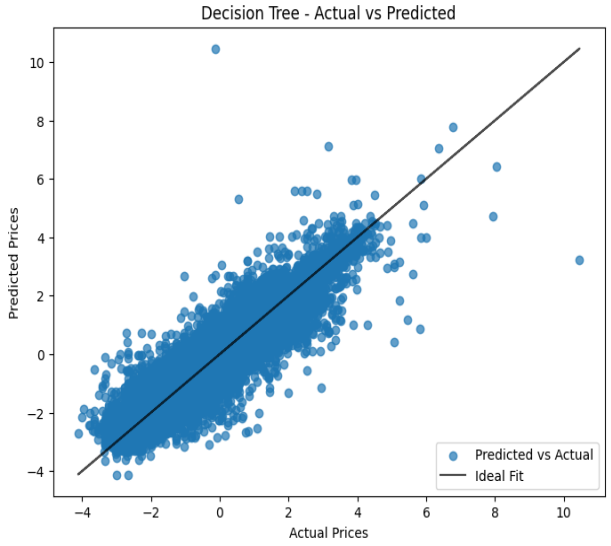

Figure 33. *Decision Tree - Actual vs Predicted*

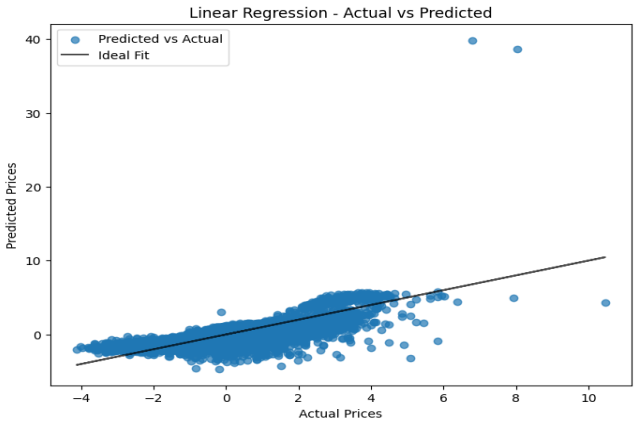

Figure 34. *Linear Regression - Actual vs Predicted*

4.1.4. Visualization of Model Performance

To understand the performance of the different models and compare them, three bar charts were created: one for the

Mean Squared Error, one for the Mean Absolute Error, and one for the R2 scores. These charts very well present the variation in accuracy and inaccuracy between the models.

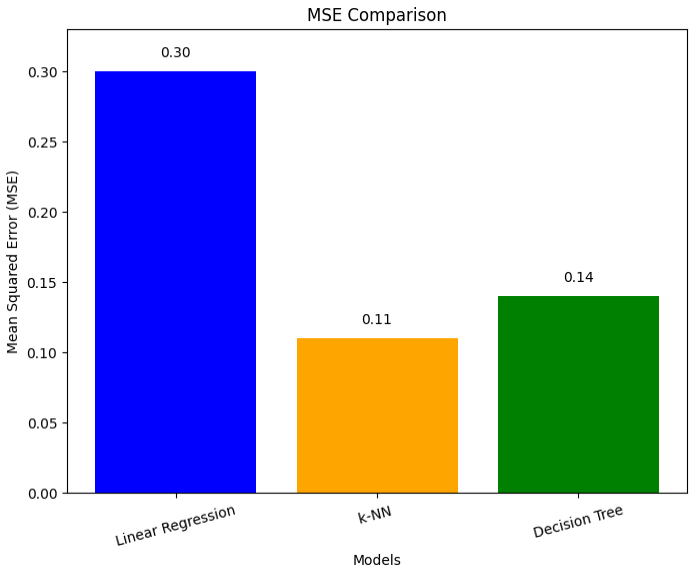

Figure 35. *MSE Comparison*

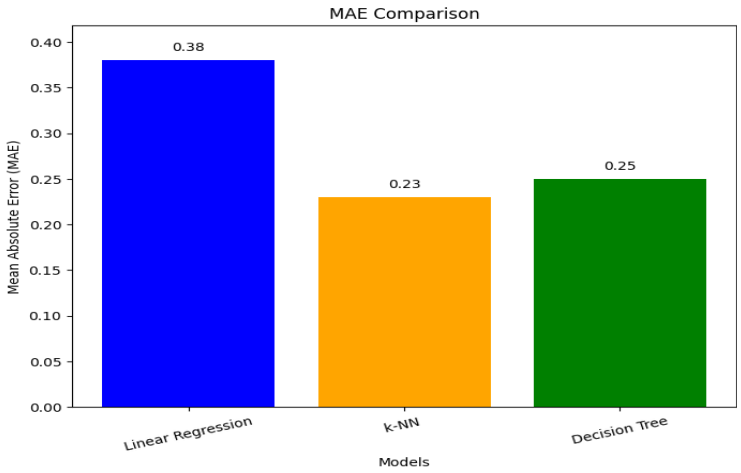

Figure 36. *MAE Comparison*

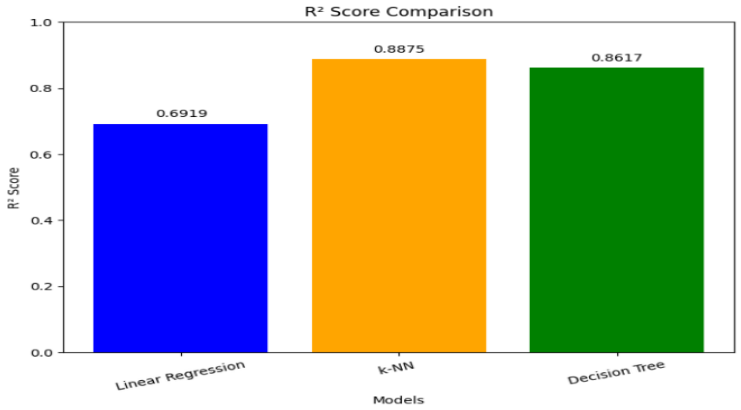

Figure 37. *R^2 Score Comparison*

4.5. Conclusion

Based on the evidence, the **k-Nearest Neighbors (k-NN)** model is undoubtedly the best in the prediction of automobile prices, as it has a strong R^2 score and low error metrics, which is evidence that it effe ctively captures the associations between features and target variables.

.....

4.2. Feature Importance

Feature importance analysis, in turn, helps the data scientist find those features which most influence the model's predictions. The knowledge of the importance of one attribute in comparison to others allows us to focus on the most important ones and tune our model accordingly.

4.2.1. Feature Importance in the Decision Tree Model

The importance of each feature was identified after the Decision Tree model was trained using the best hyperparameters (max_depth=none, min_samples_split=10), and the results are compiled below:

Year_of_registration: It is the most important feature with an importance value of 0.450944.

Standard Make (Encoded): The brand of the cars is the second most important feature with an importance value of 0.320109. This feature is clearly important because luxury brands like Mercedes or BMW are considerably more expensive.

Mileage: This feature of cars is the third most important feature with an importance value of 0.152393. Since high mileage cars usually have low pricing, it suggests that mileage plays a significant role in determining car prices.

Body type (Encoded), Fuel Type (Encoded) ,Color (Encoded) and Vehicle Condition (Encoded) features are in the next stages. Vehicle Condition (Encoded) is the least important feature, which has an importance score of 0.001941

4.2.2. Visualization

The following bar chart was made to show how important each feature is:

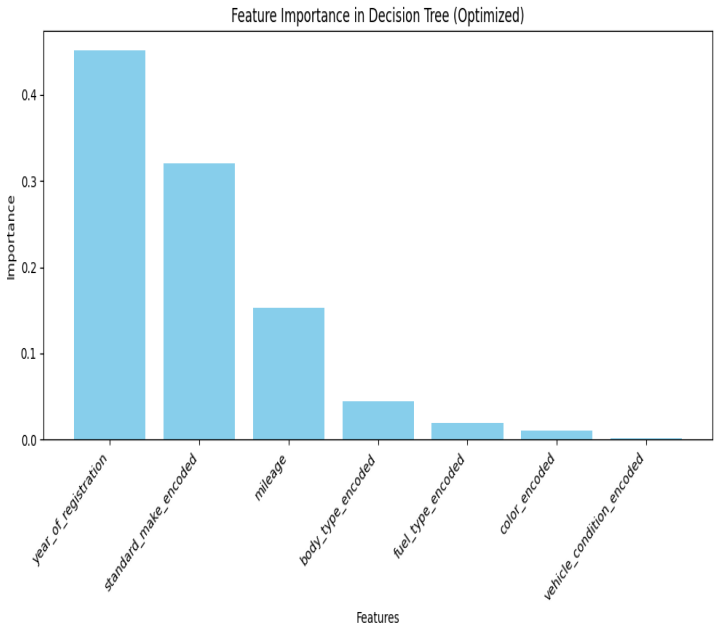

Figure 38. *Features importance in Decision Trees*

.....

4.3. Fine-Grained Evaluation

I have also done the fine-grained analysis, which is more about assessing the model on a more instance-level of prediction error. The absolute error for each prediction was computed and marked as cases with the largest error.

4.3.1. Instance-level errors

The absolute errors computed for every sample were defined as the difference between the actual and expected prices.

We then ranked all the instances by the error value and found the samples with the smallest prediction errors. Following is a summary of the cases with ten smallest errors:

Actual	Predicted	Absolute Error
0.348344	0.348344	0
0.233962	0.233962	0
0.233962	0.233962	0
0.233962	0.233962	0
0.233962	0.233962	0
0.233962	0.233962	0
0.233962	0.233962	0
0.233962	0.233962	0
0.186364	0.186364	0
0.233962	0.233962	0

Table6. *a summary of the cases with ten smallest errors*

4.3.2. The distribution of errors

A scatter plot was created to visualize the relationship between actual and expected price. The red dashed line, which reflects the ideal place all the points should fall, shows perfect predictions. Very far from this line means large prediction errors. With the help of the scatter plot, finding the patterns and outliers where the model does a bad job is pretty easy.

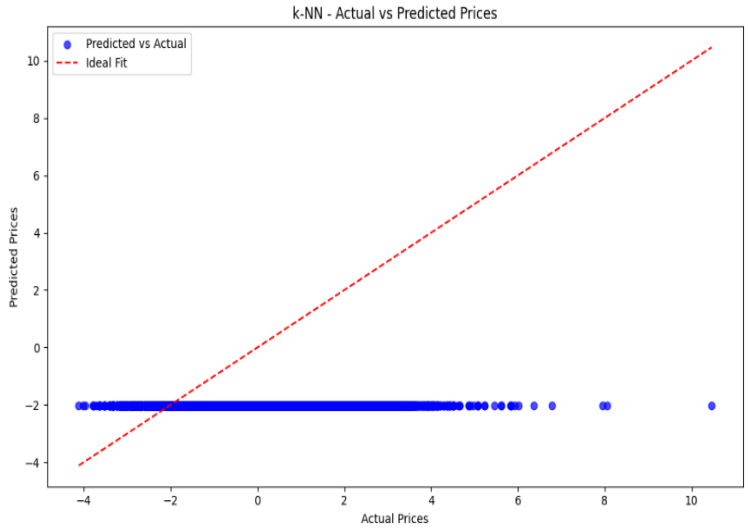

Figure39. *Actual vs Predicted Prices*

4.3.3. Top Errors

For the identification of specific samples where the model's predictions were far off, the top ten cases with the largest errors were analyzed. The table below shows five of them. This may indicate certain trends, such as outliers or cases where specific features or inconsistent data make it hard for the model to generalize.

Actual	Predicted	Absolute Error
10.461789	0.219486	10.242303
8.047257	-1.347944	9.395201
6.372815	-1.347944	7.720759
7.945294	0.522094	7.423200
6.784861	0.246467	6.538394

Table7. *a summary of the cases with five top errors*

4.3.4. Conclusion

We can understand the strengths and weaknesses of the m-odels by evaluating it at the instance level. This fine-grained research helps in pointing out areas for development: feature engineering, better data preparation, or more sophisticated models to handle intricate patterns.