1. Data/Domain Understanding and Exploration

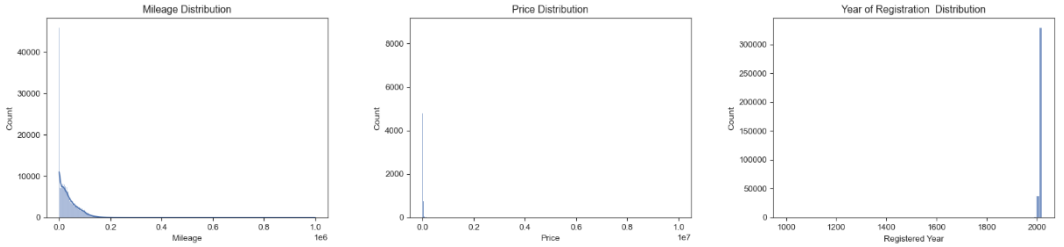
1.1 Meaning and Type of Features; Analysis of Distributions

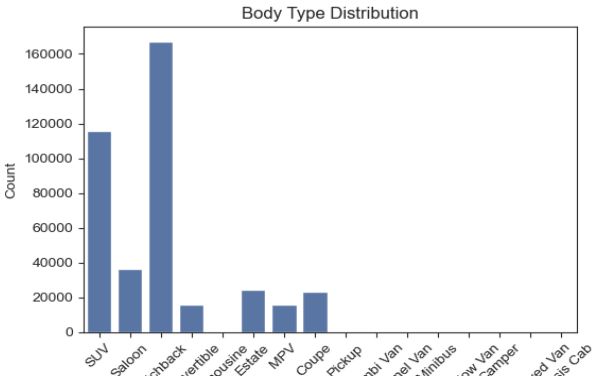
There are total **402,005** records available in the dataset and **12** features which describe the vehicles numerically and categorically. Some of the numerical variables are; **mileage**, **year_of_registration**, and price while categorical variables include; **standard_make**, **vehicle_condition**, and **fuel_type**. The majority of the entries in most of the columns are full with no missing values, but some vital variables like mileage and **year_of_registration** have some significant number of missing values, **127** and **33,311** respectively. This missing value could potentially reduce the accuracy of the models esp. where features like the **year_of_registration**, upon which an estimate of the vehicle age is based on, is absent.

This table presents all features observed in the dataset and their description, type and the panda data type. It separates the numerical and categorical features which gives a good idea about the structure of the dataset we are going to work on. **Mileage**, **year_of_registration**, and price are measures that are quantifiable, while **standard_make**, **vehicle_condition**, and **fuel_type** are qualitative variables. It also displays the data types corresponding to each feature which helps in making it clearer on what type of data is comprised and is useful preparation for further analysis and modelling.

Feature	Description	Type	Data Type (pandas)
public_reference	Unique identifier for each vehicle.	Categorical	int64
mileage	Distance traveled by the vehicle.	Numerical	float64
reg_code	Regional registration code for the vehicle.	Categorical	object
standard_colour	Color of the vehicle.	Categorical	object
standard_make	Manufacturer or brand of the vehicle.	Categorical	object
standard_model	Specific model of the vehicle.	Categorical	object
vehicle_condition	Condition of the vehicle, e.g., NEW or USED.	Categorical	object
year_of_registration	Year when the vehicle was first registered.	Numerical	float64
price	Selling price of the vehicle (target variable).	Numerical	int64
body_type	Type of vehicle body, e.g., SUV, Saloon.	Categorical	object
crossover_car_and_van	Whether the vehicle is a crossover between a car and a van.	Binary	bool
fuel_type	Type of fuel used by the vehicle.	Categorical	object

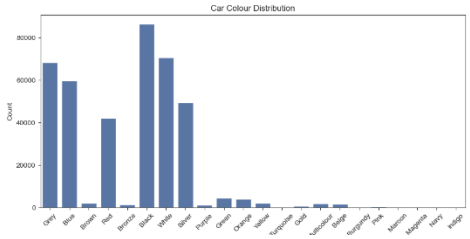
Histograms of the **Mileage**, **Price**, and **Year of Registration** give an overview of the dataset's characteristics and perhaps, some quality concerns. The shape of the Mileage distribution is similar to that of the Age distribution: while the majority of vehicles has low mileage, a few cars have very high mileage, as evidenced by the long tail extending to **1,000,000**. This implies that there could be a few large values that are either wrongly estimated or are from extremes which should also be dealt with during preprocessing. The Price distribution looks like the previous one, where the majority of cars costs less than **£20,000**, but there are some cars with extremely high price, up to **£10,000,000**. Presumably, these high prices reflect the cost of the rather exotic or expensive models, but may stem from errors in data input. Finally, the Year of Registration distribution indicates that most cars were registered between **2000** and **2020** and the specific peak for cars registered in the early **1900s**. It may be assumed that this difference is due to data entry mistakes, which have to be corrected. These distributions raise the need for data preprocessing especially detecting outliers and cleaning the data for better results during analysis and modelling.

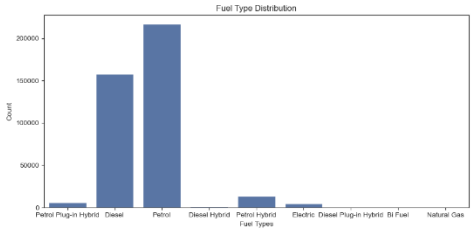




It is notable that the histograms of **Body Type**, **Car Colour**, and **Fuel Type** give an idea of the characteristics of vehicles in the dataset. The **Body Type** distribution reveals that **SUV** and **Saloon** are the most popular cars in the market, with **Hatchback** being the third most popular cars. Other types of body include **Coupe**, **Estate** and **MPV**, which are not very popular, while categories like **Pickup** and **Van** are present in a tiny proportion of the dataset. This imply that the market is dominated by larger vehicle such as **SUVs** and **saloons**. In the **Car Colour** distribution, it can be seen that **White**, **Black** and **Silver** are three most common colours with other colours such as **Purple**, **Orange** and **Yellow** having a smaller number of data due to societal trends of car colour selection. The **Fuel Type** distribution shows that **Petrol** vehicles take the largest portion of the data then by the **Diesel**, whereas the **hybrid** and **electric** vehicles are less common in the data set. This demonstrates that traditional petrol and diesel cars are dominating the used car market while the frequency of alternative fuel vehicles

is rare. These distributions will be useful in identifying market preferences and will be instrumental in identifying the impact of common attributes on the price as opposed to the relatively rare ones.

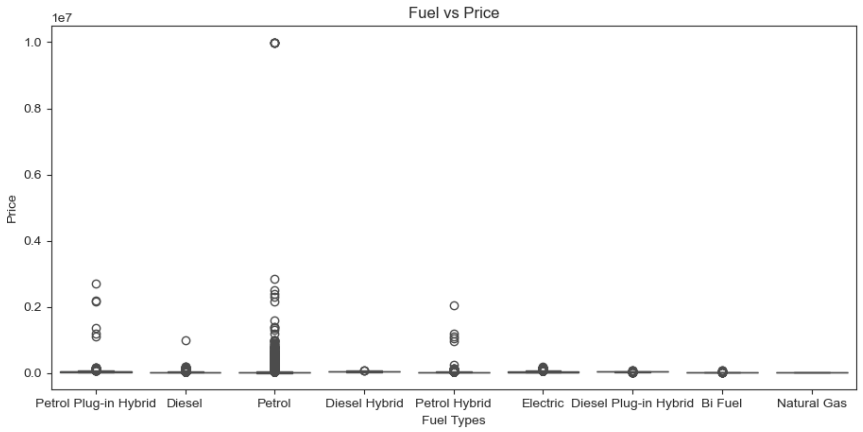


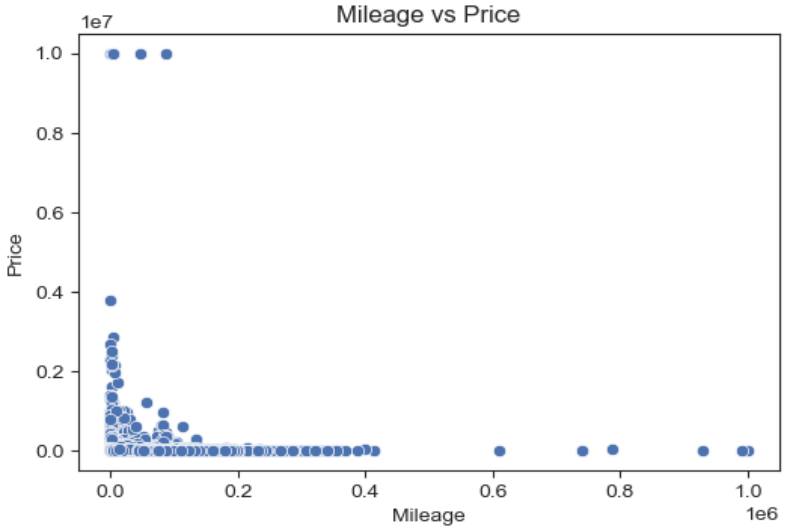


1.2 Analysis of Predictive Power of Features

The correlation matrix shows relationships between **mileage**, **year_of_registration**, and **price**. **Mileage** has a weak negative correlation with **price** (-0.16), suggesting higher mileage slightly lowers car price. **Year_of_registration** has a small positive correlation with **price** (0.1), indicating newer cars tend to have a slightly higher price. **Mileage** and **year_of_registration** are moderately negatively correlated (-0.38), meaning older cars typically have higher mileage. The heatmap visualizes these relationships, with red indicating stronger correlations and blue showing weaker ones.

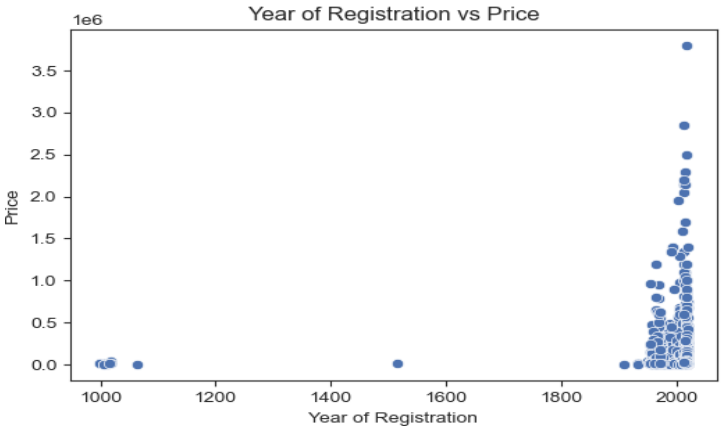
Looking at the **Fuel vs Price plot**, we note that the dataset is greatly dominated by **Petrol** and **Diesel** vehicles, most of which are relatively cheaper. However, there are one or two exceptions especially within the **Petrol Plug-in Hybrid** where the cars are relatively expensive. That means, **hybrid** and alternative fuel vehicles are less frequently sold but they are costlier based on their technological features or market trend. Now, **Electric** and **Diesel Hybrid** vehicles are less frequent in the dataset and while the average price is lower, there are still expensive models, meaning that specialised fuel types lead to increased car prices.





The **Mileage vs Price** plot shows a clear trend: as expected, higher priced cars are associated with cars with low mileage, where the average price for car with mileage below **100000** is lower. Such negative relationship indicate that buyers are more willing to incur a higher price premium for cars with lesser usage. However, there are few of them where high mileage car has almost same price as that of a low mileage car. These could be cars of certain brands that belong to the luxury or low population base, where other variables determine the price. This help to explain why car pricing is not always well defined by the distance covered but other factors come into play.

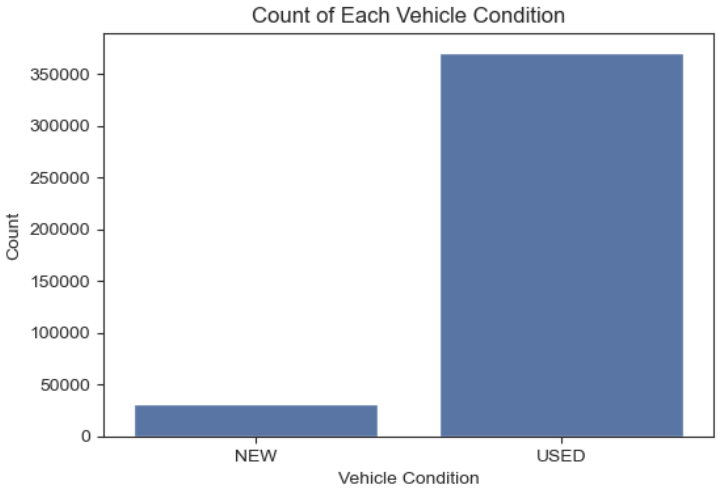
These results are further evidenced by the scatter plot determining **Year of Registration vs Price**, where we can observe that cars registered between year **2000-2020** have relatively higher prices, due to the higher utility associated with newer cars. However, the presence of some observation with very low values of registration years, for instance, cars from the **1000s**, may be attributed to data entry errors. These erroneous values could have skewed the analysis in case they were not cleaned. The distribution also proves that newer cars have more value; nonetheless, these outliers indicate that data quality concerns should be resolved to obtain the precise price prediction based on the year of registration.

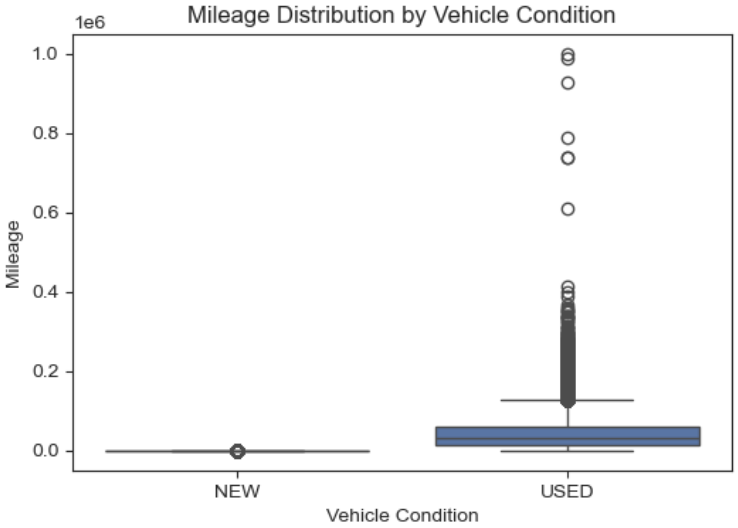


1.3 Data Processing for Data Exploration and Visualisation

In this part of the analysis, we look at some of the variables in the dataset and explore them more deeply by means of graphical displays. This will enable us to obtain other potential attributes of **vehicles** including the **Condition**, **Mileage**, **Price**, **type of body**, and **type of fuel**, which are important in the predictive modelling stage.

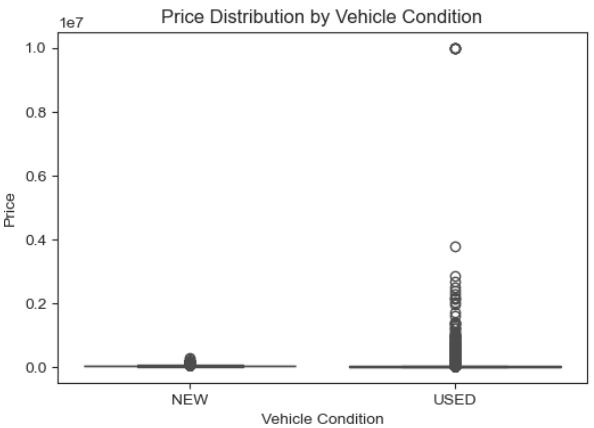
The **vehicle condition** graph reveals that the dataset is dominated by **USED** vehicles without a doubt. Many of the cars listed are used, which goes a long way towards characterising the dataset. New car and used car ratios can distort the models, especially when it comes to the price by features like mileage or car age. Awareness of this distribution guarantees that the model can be trained appropriately, countering bias from the high ratio of used cars.





The **mileage distribution** for different **vehicle condition** reveals that new cars usually have very little travelling distance in comparison to the used cars where most of the cars show higher travelling distance. This visualisation enables one to notice that there are some extreme values in the data with regards to the mileage which may require further investigations during data cleaning phase. It also helps in seeing how mileage overlays with the general state of a car, which is an important factor for the price prediction models.

The distribution of **price** of **vehicles** based on their condition reveals that **used cars** have a higher variability in price than **new cars** with some extremely expensive cars. New car models, on the other hand, have very low price point in the dataset, this could be due to lack of consistency or the fact that there are few new car models in the market. The variation of price by condition assists the model to identify patterns in each category leading to better price prediction.



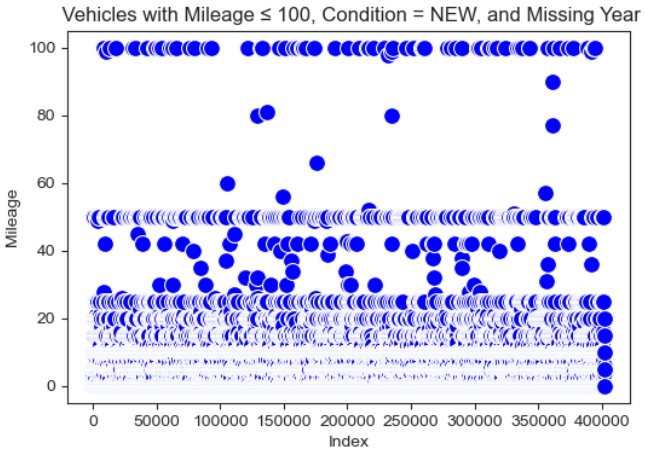
From the table, the number of times each feature has missing values in the dataset is indicated. The **mileage** column alone has **127** missing values, which are not very many compared to the size of the data set. The **reg_code** variable also has a lot of missing values, **31,857** to be specific, this means that it needs to be handled in some way or the other. The other features such as **standard_colour**, **year_of_registration** and **fuel_type** also contain missing values while price and **vehicle_condition** have no missing values. These missing values should also be treated by data cleaning procedures because they may distort the accuracy of the model. Using missing data appropriately makes a point that the machine learning models are trained using data which does not contain missing information.

```
public_reference          0
mileage                 127
reg_code              31857
standard_colour        5378
standard_make             0
standard_model            0
vehicle_condition         0
year_of_registration  33311
price                     0
body_type               837
crossover_car_and_van     0
fuel_type               601
dtype: int64
```

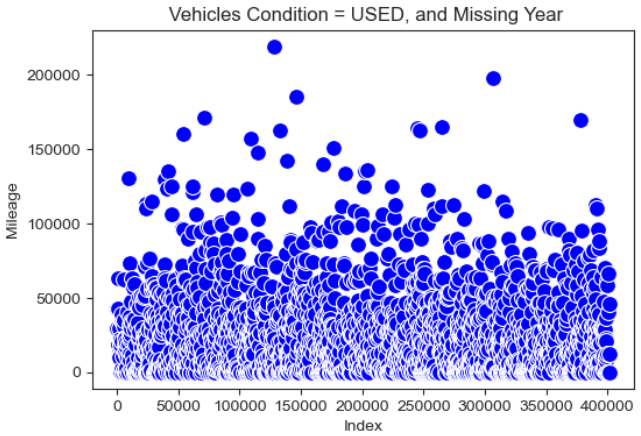
2. Data Processing for Machine Learning

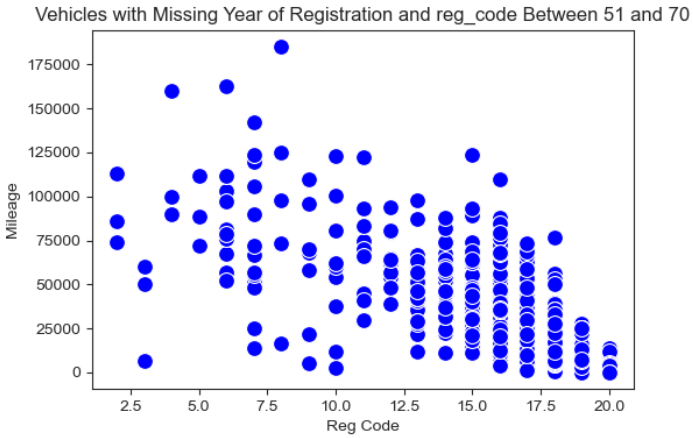
2.1 Dealing with Missing Values, Outliers, and Noise

In this step, the dataset was queried by the condition that the **'Mileage '≤ 100' AND 'condition = NEW' AND 'Year of registration missing'**. As a result of analysing the missing data, the missing year of registration values were replaced by since it is the year with the highest value in the dataset. This helps to fill the missing values in as coherent and credible a fashion as possible, for subsequent objective analysis.

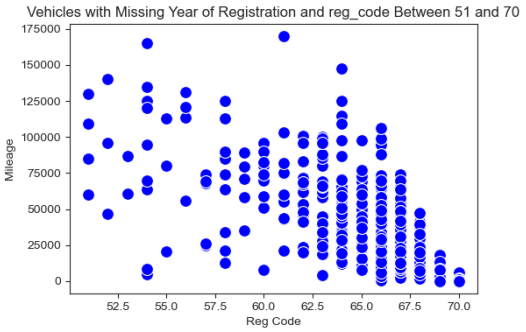


In the **first plot**, we select vehicles with **condition = USED** and a missing **year of registration** and look at the distribution of the mileage for these vehicles in order to look for patterns. On this basis, in the **second plot**, we look at **vehicles with a missing year of registration** and **reg_code** values ranging from 2 to 20. In order to fill the missing **year of registration**, the **reg_code** is matched to the **UK Vehicle registration plates** since the **reg_code** has a relationship with the **year of registration** of the vehicle. This enables us to correctly estimate the missing values based on this developed relationship. The following observations can be made after dealing with the missing values in the **year of registration**: All together **1,255** entries still have no registered year.

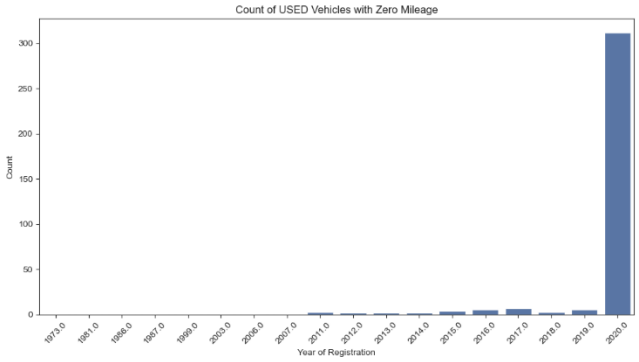


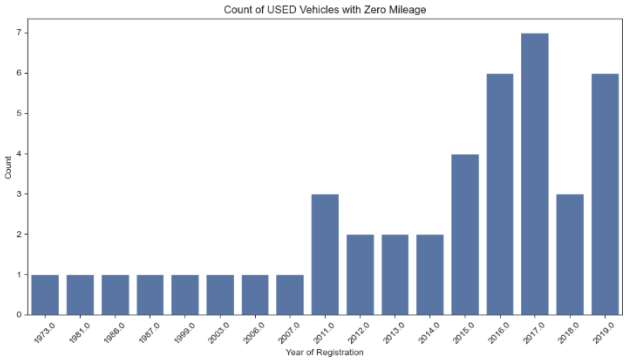


In below graph, we will consider the vehicles that did not have the year of registration as the parameter **reg_code** comes in between **51 and 70**. These records include **UK car registration** data, where we use the **reg_code** to complete the missing year of registration values through the UK registration plates. After this process, only **378** entries have missing year of registration. For these remaining missing values, we impute them with the **median(2017)** year of registration, and our dataset is now ready for further analysis.



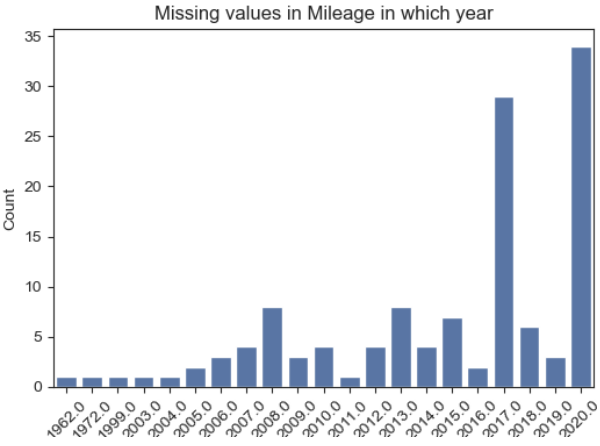
In the **first image**, we are checking the **mileage** column where **mileage = 0** and vehicle **condition = USED**. We also see that more than **300** entries are from the year **2020**, which is strange because these are listed as USED cars but have no mileage assigned to them. After analysing these findings, the authors agreed to make a modification to the vehicle condition to **NEW** for these 2020 vehicles because it is logical for new vehicles to have zero mileage. In the **second image**, after making the adjustments, all the entries left where **mileage = 0** and vehicle **condition = USED** are belong to year 2019 and before. But when we analyse it again, we come to a conclusion that it is not possible for vehicles used from **1973 to 2019** to have 0 mileage. This means that there could be more data errors that we will need to deal with in the next steps, some of the entries in the dataset may be unrealistic and will therefore be deleted or corrected.

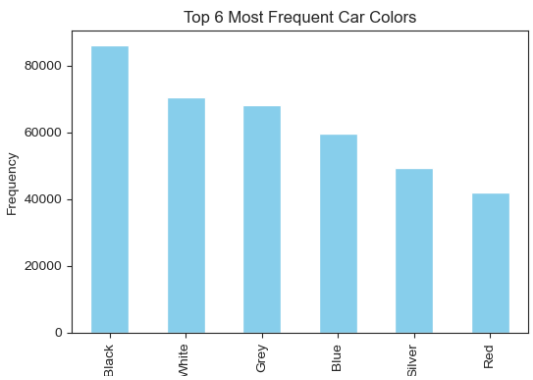




Next, we concentrate on the entries for which the **mileage = 0**, and the year of registration is between **1973 and 2019** as such entries do not hold any sense. To address this, we noticed that the same dataset has other records with actual mileage values for years **1973-2019**. In order to compute for mean, we got the mean value **44,164**. We then used this value to impute the missing mileage values so that the dataset is less inconsistent and more realistic without compromising the data.

In the **provided histogram**, we analyse the distribution of the **missing mileage values** for various years. From the chart, it can be noted that majority of the missing **mileage** values are for the relatively current years especially the year **2019 and 2020**. Upon analysing this information, we then estimated the **median** mileage for the vehicles that had missing values of the variable, which was **30,086**. This median value was then applied in order to fill in the missing **mileage** values making the data set a little more uniform and the missing values are replaced with a moderate value from the set.

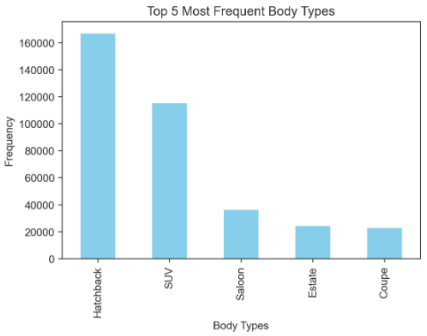




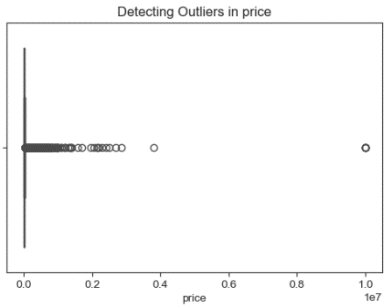
For the missing **color values**, we will first impute them with one of the **Top 6 Most Frequent Car Colors**: **Black**, **White**, **Grey**, **Blue**, **Silver**, **Red**, following the frequency of these colours in the given dataset. All the other colours that have not been grouped among the above mentioned 6 will be grouped under "**Other**". This method proves useful in making the colour data more consistent while at the same time working well for the less frequent colours.

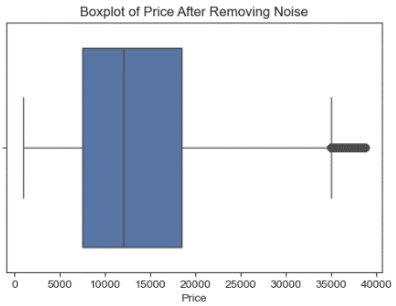
For the missing values of the **fuel types**, we first drop the entries Natural Gas and Bi Fuel since they are very rare and do not play a large role in this context. Finally, we add two new columns, **petrol_hybrid_and_plugin**, which merges the **Petrol Hybrid** and **Petrol Plug-in Hybrid** fuel types. Likewise, we lump **Diesel Hybrid** and **Diesel Plug-in Hybrid** into a new column that we call as **diesel_hybrid_and_plugin**. After these adjustments, we fill in the missing fuel type values by using the **mode** (the most often occurring value) for the corresponding rows. This approach also makes the fuel type data more manageable and cleaner, although it retains the flavour of the most frequently occurring fuels in the data set.

In the case of the **body type** column, we also went through the same process as the colour column. We found out the **Top 5 Most Frequent Body Types**, and then use **mode**, which is the most frequently occurring value, to fill in the missing values for those variables. This makes it possible to fill the missing body type values with a sensible and consistent value derived from the datasets' most popular values.

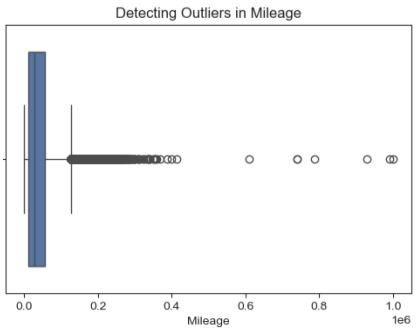


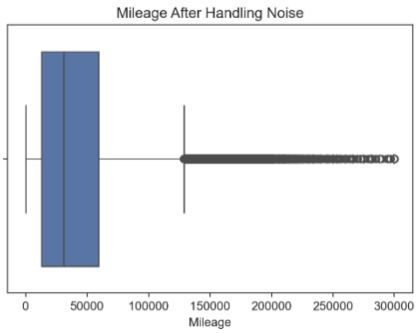
From the **first image**, it is possible to see the outliers in the **prices** data analysis by means of the box plot. These outliers are values which may influence the results and the model's error, and therefore should be dealt with properly. For this we employed the **Interquartile Range (IQR)** method to deal with these outliers as shown below. As seen from the **second image**, after applying the **IQR** method, the outliers are filtered out and the price data is much more uniform and within the right range for modelling the data set. The adjustments made to the data has improved its distribution to a more common one, which will make our analysis more accurate.

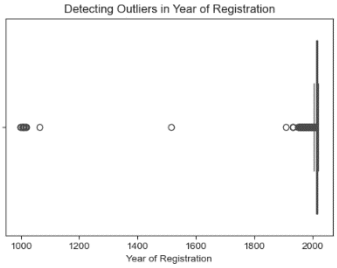


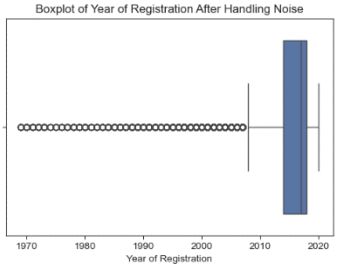


In the **first image**, we use a boxplot to determine the extreme values in the data by observing those that are out of the box. To address this, we philtre out the noise by setting the constraint on minimum mileage of **0** and maximum of **300,000**. The **second picture** illustrates the outcome after addressing the noise and here the **mileage** values are within this range. This adjustment is done in order to filter out outrageous values that are not quite reliable within assessment and modeling processes.





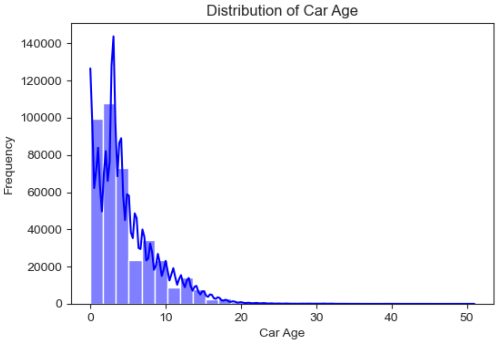


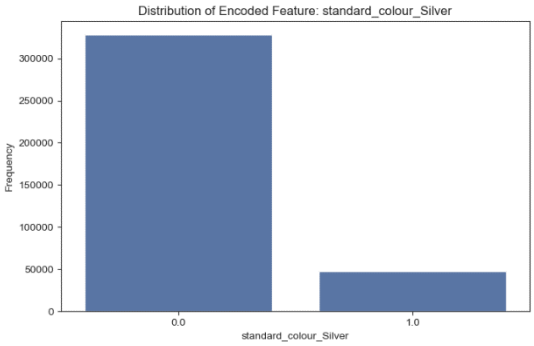


In the **first image**, we identified the outliers in the **year of registration**, and as expected, we find such values as **1000**, **1500** and so on. To philtre out this noise, we consider cars from **1969 to 2020**, modern and vintage cars. The second picture depicts the data after noise was addressed and the distribution of the year of registration is within a reasonable range, from **1970 to 2020**.

2.2 Feature Engineering, Data Transformations, Feature Selection

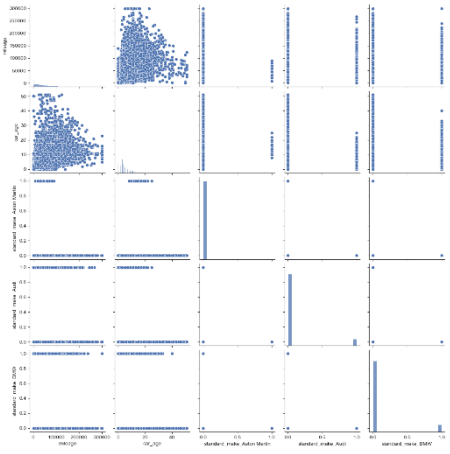
In this step, we also added another new column known as where the registration year for each car was subtracted with the current year, which is 2020. The histogram represents car age and it shows that majority of the cars are relatively young, with the age of 0 to 5 years and the number of cars reduce greatly as the age of the car increases.





After creating the **Car Age** column, we dropped the following columns: **public_reference**, **reg_code**, **year_of_registration**, **and crossover_car_and_van** were excluded from the analysis as they are no longer relevant.We also followed up with the encoding of the dataset, and then converted categorical features into numerical features. For Example: The **plot** represents the **standard_colour_Silver** with the majority of vehicles represented as 0 (not silver) and a few as 1 (silver). This encoding helps bring the data into a form that it can be modelled.

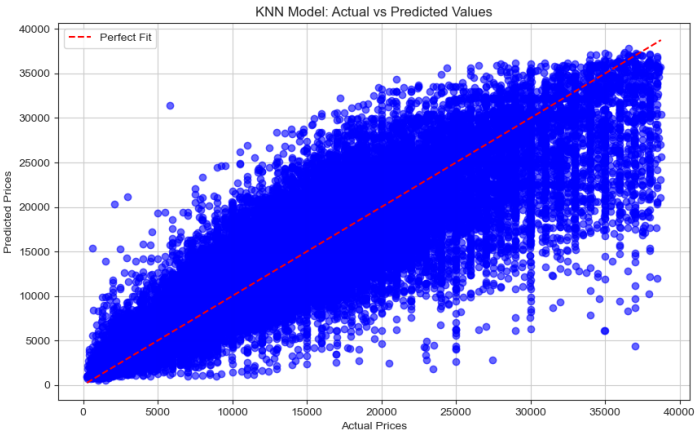
After having encoded the categorical features we decided to select a **40% sample** of data and split it into training and testing set. We then proceeded to normalise the features using the **standard scaling** method as this was seen to enhance performance of many models. After that, we did **feature selection** to select the features that are most important in predicting the target variable. The **plot represents the five features** that were chosen out of the total number of features in the dataset to minimise the dimensionality and increase efficiency of the model.

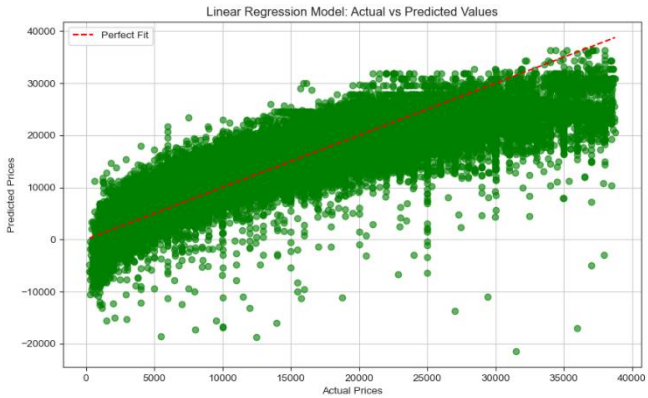


3. Model Building

3.1 Algorithm Selection, Model Instantiation and Configuration

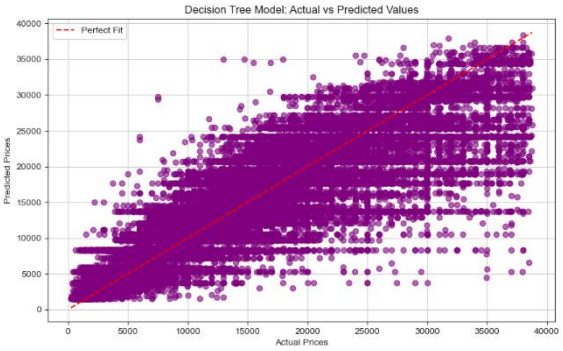
While constructing **KNN** model, the first data preprocessing step, which was performed, was **standard scaling** of the dataset. After scaling, we then applied the KNN model on the training data. The model received an **R^2** value of **0.77** and **RMSE** of **4012** this means that the model fits the data fairly well. The **plot shows** the **actual and predicted prices**; the **red dashed line** represents the "ideal" line. The farther away the points are from this line, the worse the model predictions are compared to actual price. When comparing the actual values to the predicted values from the model, the scatter plot shows that the model is good at representing the data but there are some anomalies.



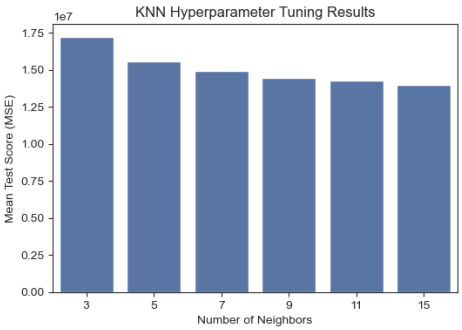


Once the **Linear Regression** model has been built and fit to the dataset, we attained an **R^2** of **0.71** and an **RMSE** of **4429**; so, it is a decent fit but there is still much that could be done as compared to the KNN model. This **chart** compares the **actual prices and the predicted prices** on the same graph with a reference line of the perfect fit in. By observing the scatter plot, the author was able to determine that the model was able to predict the general behaviour of the data though the model was not able to predict the behaviour of the data perfectly especially at higher prices. This has a hint at the model that can be tuned again or better features could be included in the model.

When creating the **Decision Tree** model and training it with the parameter **max_depth=10**, we received the following evaluation characteristics: **R²** of **0.75**; **RMSE** of **4112**. The **graph** shows the **actual and predicted price** and the shows the ideal line of fit. The model establishes the overall movement of the data values and its predictions are relatively close to the actual values with some discrepancies at higher prices. The Decision Tree model is found to be suitable and has a higher R² value than the previous models.

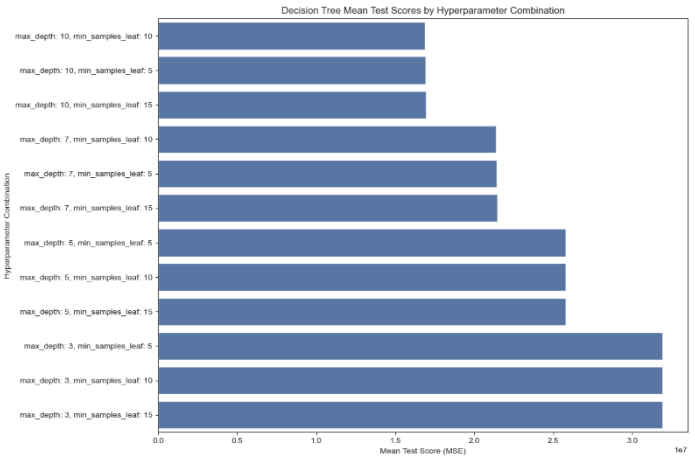


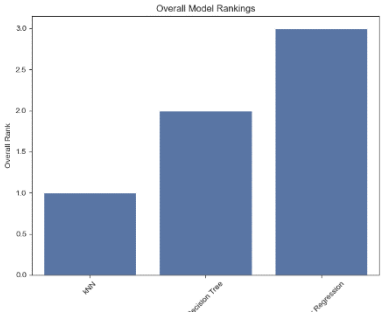
3.2 Grid Search, and Model Ranking and Selection



After performing **Grid Search** for the **KNN model**, we tested multiple values for the **number of neighbors: 3**, **5**, **7**, **9**, **11**, and **15**. The **Grid Search results** showed that setting **n_neighbors** to **15** offered the highest results with **R²** of **0.79** and **RMSE** of **3801**. The **graph** provided also points out the **mean test score (MSE)** for the case of each setting of hyperparameters and identifies those 15 neighbours offered the best prediction of price. This hyperparameter will be used to select the final model to employ on the system.

It is also important to note that while we cannot pass **hyperparameters** in **Linear Regression** as in the case of other models because Linear Regression is a simple model that does not depend on the tuning of other parameters it depends on the input variables. For the **Decision Tree model**, we used the **Grid Search** with **hyperparameters**: **max_depth [3, 5, 7, 10]** and **min_samples_leaf [5, 10, 15]** During the Grid Search, the best hyperparameters were set **to max_depth = 10** and **min_samples_leaf = 10**, the **RMSE** was **4107** and the **R²** was **0.75**. In the **following graph**, we did Grid Search on the Decision Tree model with two hyperparameters; max_depth and min_samples_leaf. The values which gave the minimum MSE were **max_depth 10** and **min_samples_leaf 10**, The model had an R² of 0.75 and the RMSE of 4112. This configuration offers the least average error and hence the best bias-variance trade-off for the Decision Tree model.





We did model ranking where **KNN** scored the highest compared to Decision Tree and Linear Regression as shown in the graph above. From the evaluation criteria, KNN was chosen as t he model most suitable for the task. From the model selection output we find the best KNN with the hyperparameter **n_neighbors = 15** with an **R²** of **0.79**, and **RMSE** of **3801**.

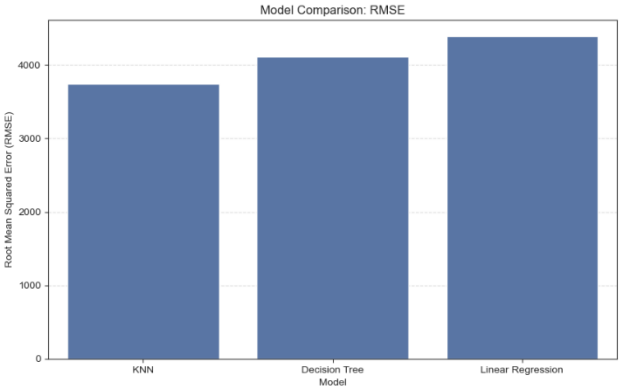
4. Model Evaluation and Analysis

4.1 Coarse-Grained Evaluation/Analysis

The **Coarse-Grained Evaluation** also presented **KNN** model as the best performer with the **RMSE** of **3,801**and the highest **R²** score of **0.79** placing the model at the highest ranking. This also shows that the KNN from using the **K=5 nearest neighbours** was proving to be the best model of accurately predicting the car prices hence able to capture the relationships in the given data set. The model that follows in second place is the Decision Tree model, with a slightly **RMSE (4,107.03),** and **R² (0.75)** demonstrating acceptable performance, but a slightly higher error from **KNN**. Last but not the least, the Linear Regression model came at last position with the **RMSE 4,429.36** and the lowest value of **R² 0.71**. This means that Linear Regression failed to work in decoding the data variance and other non-linear relationships within the set pattern, thus, a severely limited ability to predict. In total, KNN model was the most accurate and was followed by Decision Tree and then Linear Regression with the varying level of accuracy.

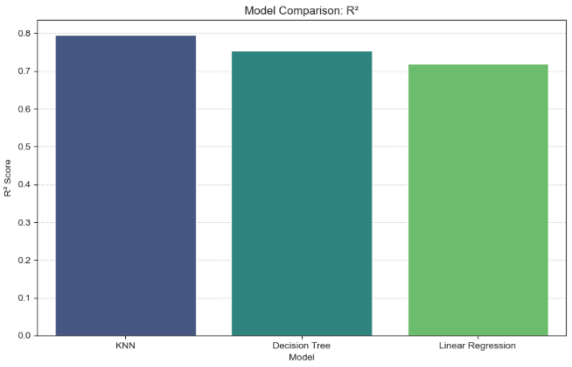
Coarse-Grained Evaluation Table:

	Model	MSE	RMSE	R^2	Overall Rank
0	KNN	1.444860e+07	3801.131446	0.789379	1.0
1	Decision Tree	1.686769e+07	4107.028990	0.754115	2.0
2	Linear Regression	1.961922e+07	4429.359051	0.714005	3.0

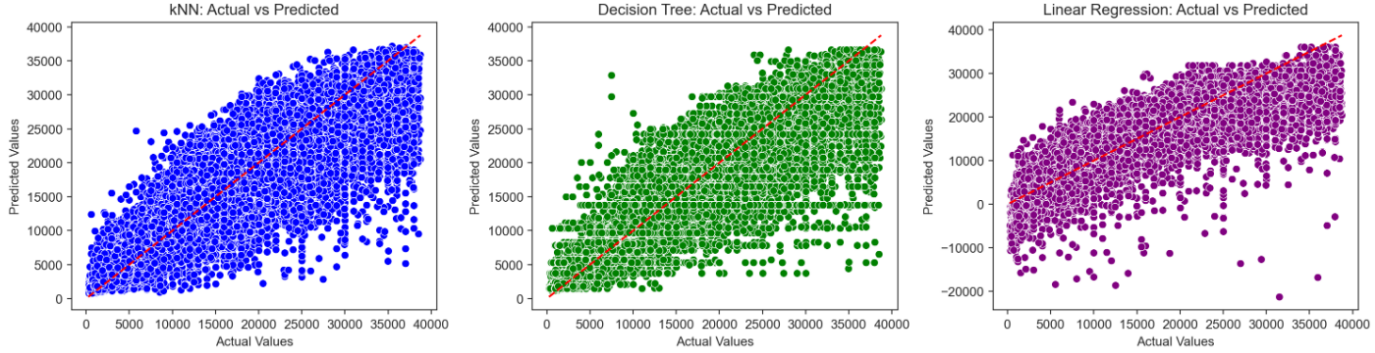


After performing the **RMSE** comparison, the **KNN** model achieved the **lowest RMSE** of approximately **3,801**, indicating that it had the least error in predictions among all the models tested. The **Decision Tree** model followed with an **RMSE** of about **4,107**, which was slightly higher than KNN, suggesting that it produced a bit more error in its predictions. Linear Regression had the highest **RMSE** of around **4,429**, indicating that it struggled more with the accuracy of its predictions compared to both KNN and Decision Tree models. Overall, KNN was the best performing model in terms of RMSE, followed by Decision Tree and Linear Regression.

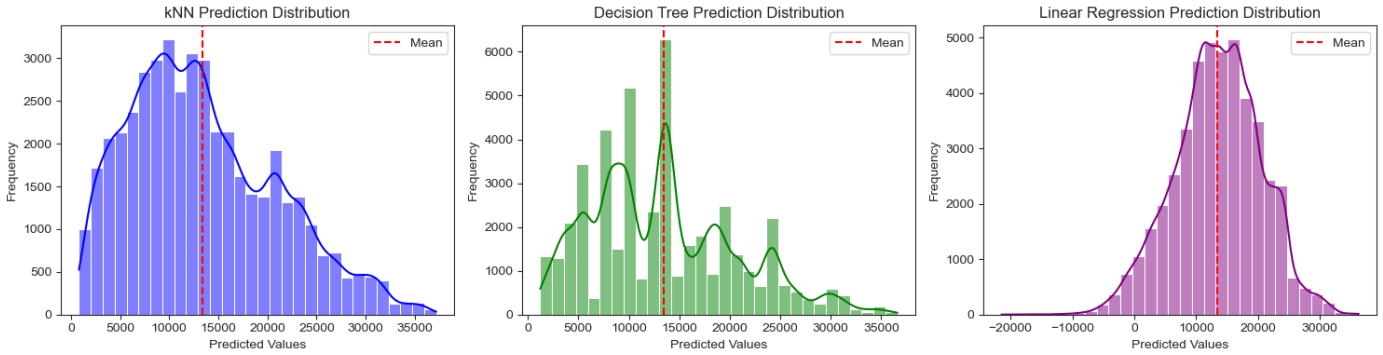
After comparing the models using the **R²** score, we see that **KNN** performed the best with the highest **R²** score, close to **0.79**. This indicates that KNN was able to explain most of the variance in the data. The Decision Tree model followed closely with an **R²** score of around **0.75**, suggesting that it was also a strong performer, though slightly less accurate than **KNN**. Linear Regression, while still reasonable, had the lowest R^2 score among the three models, which was closer to **0.71**, indicating that it explained the least amount of variance in the data.



After comparing the **R²** scores for the models, we visualized the actual vs predicted values for each model: **KNN**, **Decision Tree**, and **Linear Regression**, as seen in the scatter plots. In these plots, each model's predicted values are compared against the actual values, with a **dashed red line** representing a **perfect fit** (where predicted values equal the actual values). From these plots, we can observe that KNN and Decision Tree models produce predictions that are relatively closer to the actual values compared to Decision Tree. These visual comparisons help in evaluating the models' performance.



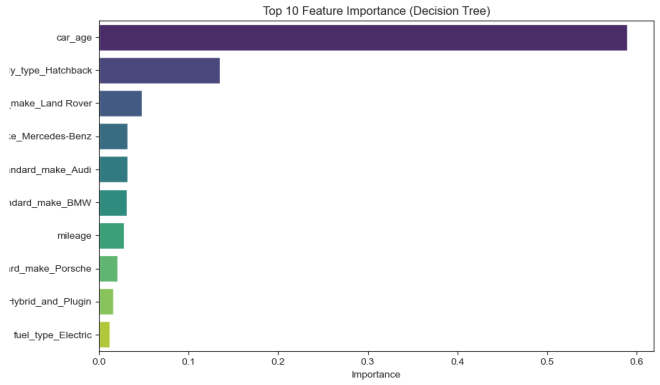
The actual and the predicted values and their distributions of all three models were plotted by us. The spread of the predictions using **KNN model** is relatively flat and slightly skewed towards lower values, so the model seems to make more or less evenly distributed predictions over the range of price. The **Decision Tree** model has more focus on a particular value which means that the model makes predictions mainly in a certain price range and relatively fewer extreme values. The prediction distribution of the **Linear Regression** model is positively skewed, which indicates that the model provides more prediction toward the lower end of the price scale with the long tail toward high end. Such differences in the prediction distribution reveal the divergent propensities of each model. KNN is able to give more predictions but the price is slightly lower than the actual price while Decision Tree gives only limited number of predictions which could be partially biassed towards the given price intervals. Indeed, Linear Regression which has a skewed distribution appears to overestimate the lower prices and underestimate the higher ones. These considerations aid in determining which of the models is more suitable in instances of different kinds of data or for particular applications of business intelligent, by considering the distribution of the predicted values and overall performance indexes.

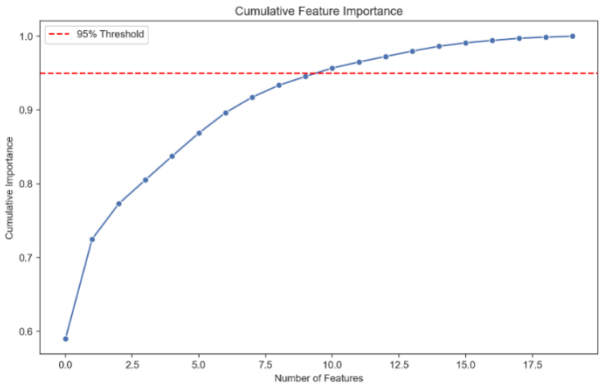


4.2. Feature Importance

In this section, feature importance analysis was done using **Decision Tree** model. As it is seen from the results, the most important feature is **car age** with the importance score of **0.589**, which indicates that car age plays a great role in the prediction of car prices. After that, the **body type**, that is, **Hatchback**, is ranked significantly with an importance score of **0.135** of contributing to the price. Also, the car make also has an important influence on car prices, which are also influenced by brands like **Land Rover**, **audi**, **Mercedes Benz** and **BMW**. This is well illustrated in the bar chart as comes out most significant, followed by the body type and the different brands of cars. This analysis supports the observations that the car's age and brand are important considerations when setting the car's price as witnessed in the markets.

	Feature	Importance
1	car_age	0.589102
16	body_type_Hatchback	0.135354
10	standard_make_Land Rover	0.048133
3	standard_make_Audi	0.032871
11	standard_make_Mercedes-Benz	0.032718
4	standard_make_BMW	0.031638
0	mileage	0.028301
12	standard_make_Porsche	0.021003
19	fuel_type_Petro_Hybrid_and_Plugin	0.018222
18	fuel_type_Electric	0.011902
9	standard_make_Jaguar	0.011393
14	standard_make_Volvo	0.008207
13	standard_make_Volkswagen	0.008048
17	body_type_SUV	0.007485
8	standard_make_Dacia	0.006567
5	standard_make_Bentley	0.004642
2	standard_make_Aston Martin	0.003228
15	vehicle_condition_USED	0.001187
7	standard_make_Carbodies	0.000000
6	standard_make_Buick	0.000000

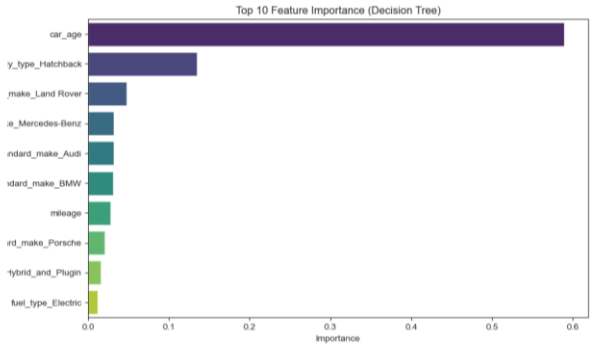




In this step, the **cumulative feature** importance analysis of the **Decision Tree** model was carried out. The graph illustrates how the cumulative importance of features grows when we add more features in terms of their importance. The red broken line points out the **95%** worth line, which shows the total cumulative importance of the specific element of the product. Analysing the graph, it becomes clear that after a specific number of features, the cumulative importance does not increase much, and it is quite stagnant. This implies that only a few features produce most of the important information regarding the target variable hence designing a less complex model.

Now, we were able to concentrate on **Linear Regression**. This can be seen in the **table** where the features with the highest coefficients were car brands such as **Aston Martin**, **Bentley**, **Porsche** and **Maserati**. These brands exerted a high influence on the predicted prices as shown by their coefficient measures. The **second picture** illustrates the feature importance in **Linear Regression** more graphically. The graph orders the features based on the absolute values of the coefficients where the intercept "**standard_make_Aston Martin**" has the highest coefficient and other premium car makers like Bentley, Porsche, and Maserati. This implies that these variables make the most contribution to the model in the prediction of the car prices because luxury car brands are usually expensively priced.

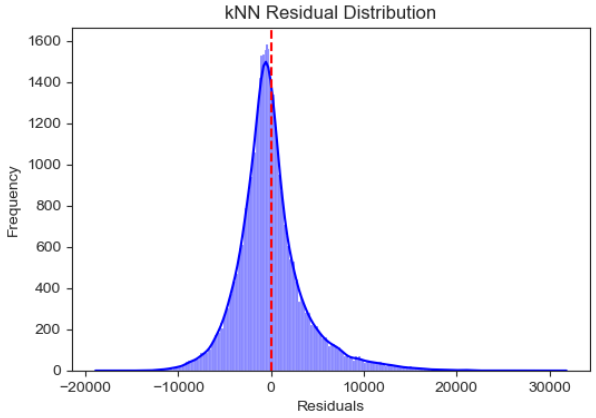
	Feature	Importance
1	car_age	0.589102
16	body_type_Hatchback	0.135354
10	standard_make_Land Rover	0.048133
3	standard_make_Audi	0.032871
11	standard_make_Mercedes-Benz	0.032718
4	standard_make_BMW	0.031638
0	mileage	0.028301
12	standard_make_Porsche	0.021003
19	fuel_type_Petro_Hybrid_and_Plugin	0.018222
18	fuel_type_Electric	0.011902
9	standard_make_Jaguar	0.011393
14	standard_make_Volvo	0.000207
13	standard_make_Volkswagen	0.000048
17	body_type_SUV	0.007485
8	standard_make_Dacia	0.006567
5	standard_make_Bentley	0.004642
2	standard_make_Aston Martin	0.003228
15	vehicle_condition_USED	0.001187
7	standard_make_Carbodies	0.000000
6	standard_make_Buick	0.000000



4.3. Fine-Grained Evaluation

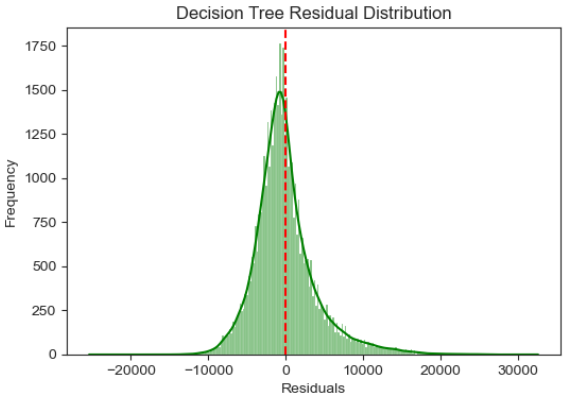
During the **Fine-Grained Evaluation** phase, we analysed the results of each model (**kNN**, **Decision Tree**, and **Linear Regression**) with the actual values. The table shows the residuals for each of the models in question, the difference between the actual value and the predicted value. This means that the values of the actual values exceeded the predicted values in the cases where residuals have negative signs, whereas the predicted values were lower than the actual ones in the cases where residuals have positive signs. The examination of residuals provides information on how accurately each of the models predicted, thus providing information on how well each of the models did in individual predictions. Such an analysis enables us to establish the advantages and the disadvantages of the various models in estimating the car prices.

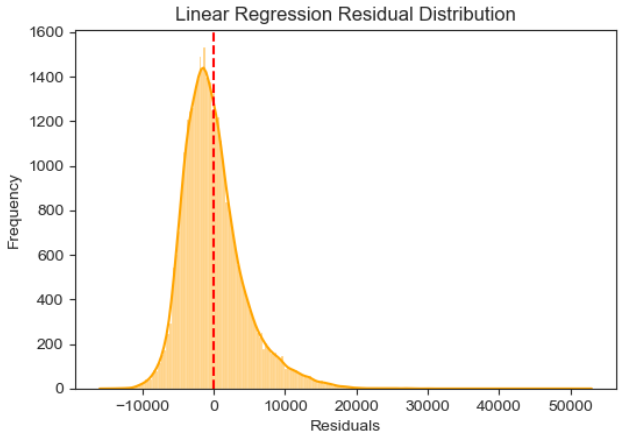
	Actual Value	kNN Predicted	kNN Residual	Decision Tree Predicted	Decision Tree Residual	Linear Regression Predicted	Linear Regression Residual
249564	15700	15775.400000	-75.400000	18985.056002	-3285.056002	15317.098227	382.901773
362841	17700	19218.666667	-1518.666667	24187.781729	-6487.781729	18990.527432	-1290.527432
92443	6495	15891.733333	-9396.733333	15288.090909	-8793.090909	12934.996752	-6439.996752
109649	13000	16704.200000	-3704.200000	15920.276224	-2920.276224	17780.145477	-4780.145477
248961	9963	8023.266667	1939.733333	8285.195611	1677.804389	12246.764215	-2283.764215
...	...	...	...	...	...	...	...
362923	2995	3555.200000	-560.200000	11150.686275	-8155.686275	-15251.765465	18246.765465
165510	20688	16104.200000	4583.800000	17677.110196	3010.889804	13671.315071	7016.684929
317325	22991	20851.400000	2139.600000	17677.110196	5313.889804	19278.208929	3712.791071
143410	9988	16586.666667	-6598.666667	20457.600000	-10469.600000	17219.919646	-7231.919646
79117	16899	25297.666667	-8398.666667	26496.333333	-9597.333333	21467.916899	-4568.916899



Thus, after **analysing the residuals**, we proceeded to plot the of the **KNN** model from the graph. To support this observation the plot of the residuals is presented, which indicates that the frequency of the residuals is rather high and has a peak at zero. This suggests that while the KNN model gives fairly good approximate of the actual values, there are a number of outlying or large residual values above and below the line of best fit. The red dashed line indicates the mean of the residuals and this shows how the predictions are on average. It shows that the majority of the points are highly concentrated around zero which means that the model is good most of the time but may fail with some predictions.

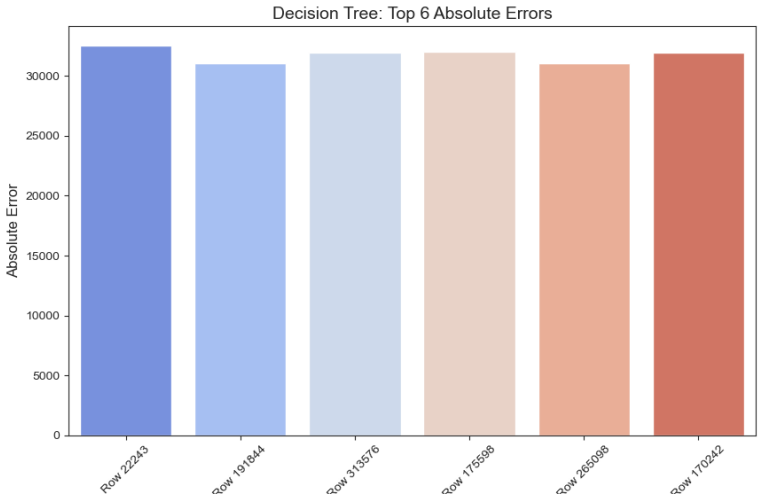
Upon inspection of the residuals, the **residual distribution** for the **Decision Tree** model was plotted and is illustrated in the graph. As in the case of the KNN Residuals we have a picture that is centred around zero which means the Decision Tree model is making fairly accurate predictions. However, the distribution of the data points is wider than that of KNN and it has more outlying points and larger residuals. The red dashed line is at zero, and the peak around this line demonstrates that the Decision Tree model is good on the average, but has more dispersion.

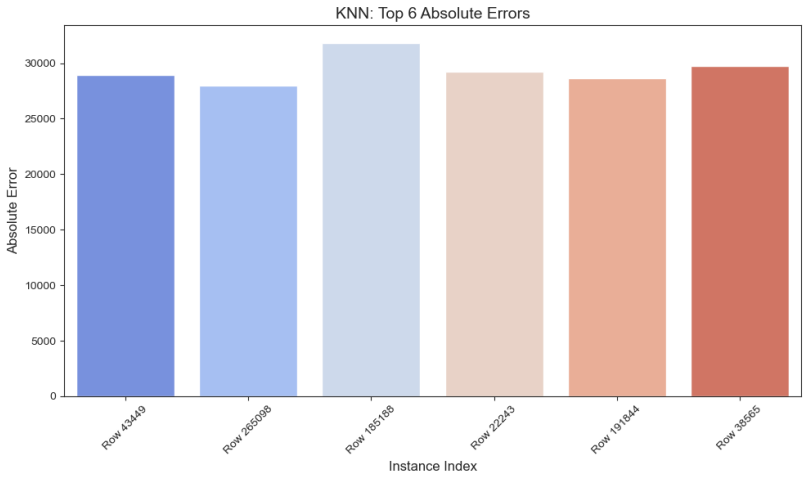




After visualising the **residual distribution** for **Linear Regression** we also see the same pattern like previous models having a peak around **zero** and **higher** variation. The Linear Regression model residuals displayed in the graph mean that most of the forecasted values are near the actual ones. However, the spread is even wider, particularly for greater residuals, indicating that the Linear Regression model makes more critical mistakes in some forecasts. The red dashed line represents the mean and as you can see the distribution is still around zero which means that the model is somewhat accurate but with much variation.

This **bar chart** shows the **absolute errors** of the **first six** important features used by the **Decision Tree model**. The vertical bars above the numbers indicate the rows in which the largest errors were made. These errors are especially important, with values of approximately , which means that the Decision Tree model did not work well on these instances. The graph helps to visually pin-point these high-error occurrences and adds understanding to which values the model was most off on.





This **graph** is showing the **six highest absolute errors** from the **KNN model**. The bar to the right represents the instance index where the prediction of the model is very far from the actual value. There are **4349** such instances with error size of roughly ; there are other similarly-sized high-error instances. This visualisation gives information about the circumstances in which the KNN model gave a low accuracy and can help in evaluating the model's predictions.

This **graph shows** the correlation between the actual and predicted value of the **Linear Regression** model and the **six largest absolute errors** out of all observations. It displays the rows where the model performance was worst in terms of the car prices' predictions. The largest absolute error was observed at row **152528** which exceeds **50 000**. Other cases with large prediction errors are also described. From these errors one can look for some forms of temporal trends or other forms of abnormalities within the data that may be accountable for such enormous discrepancies and possible ways to rectify the model for better precision.