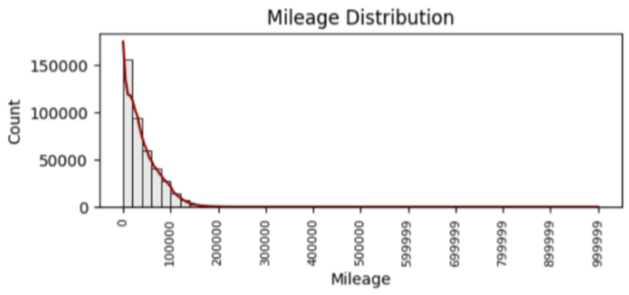
1. Data/Domain Understanding and Exploration

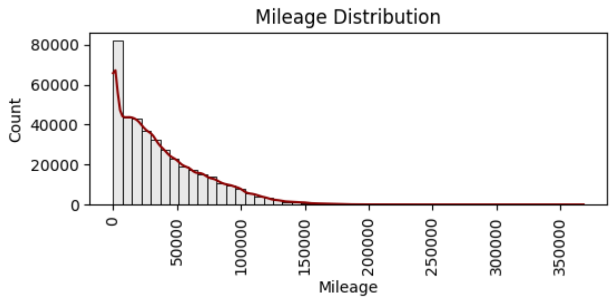
1.1. Meaning and Type of Features; Analysis of Distribution (3-4)

1.1.1. mileage

Mileage is a numerical feature which represents the distance the vehicle travelled in a year inferred from its negative correlation with price and year_of_registration. The values of this feature contain decimal numbers in the range of 0.0 to 999999.0 with 127 null values. Mileage distribution is right-skewed and the peak indicates the lower values of mileage, and the long tail indicates the higher values of mileage (Plot1.). It can be interpreted that the majority of vehicles in this dataset have lower amounts of mileage. So, most vehicles are lightly used and are in good condition. The long tail and the extremely high amounts of mileage values indicate the presence of outliers or probably old or heavily used vehicles which are less frequent. In the data preprocessing procedures, the missing values will be filled by the median value of mileage in each group that is made after applying the grouping method and is based on standard_make and standard_model. Outliers will be managed by capping method, which replaces high outliers with the maximum value (upper bound) allowed (Plot2.).



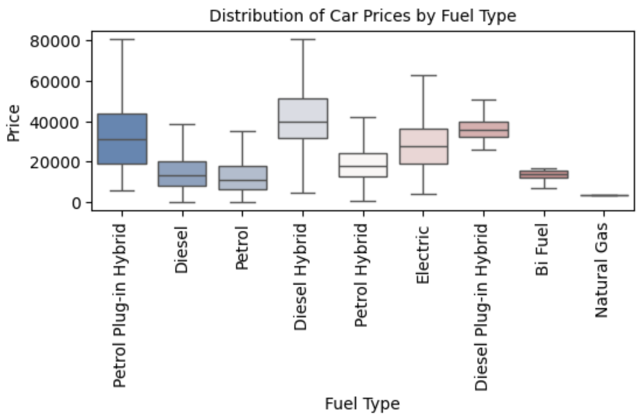
Plot1. Distribution of Mileage Before Data Pre-Processing



Plot2. Distribution of Mileage After Data Pre-Processing

1.1.2. Fuel Type

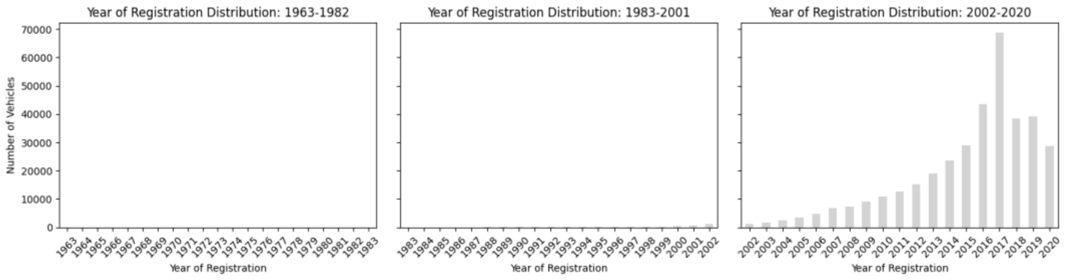
This categorical feature represents the fuel type of vehicles in 9 groups. The majority of vehicles (almost 54% of the records) consume petrol but we also have one record that displays Natural Gas consumption. There are 601 missing values in this column which will be filled based on the vehicles' standard_make and standard_model by grouping method. Plot3. illustrates the distribution of car prices across different fuel types. It seems that vehicles with Petrol Plug-in Hybrid and Diesel Hybrid fuel types have the widest price ranges according to the length of their boxes in the plot. Regarding the plot, we can confirm that the vehicle with Diesel Hybrid, Diesel Plug-in Hybrid and Petrol Plug-in Hybrid fuel type are more expensive than others.



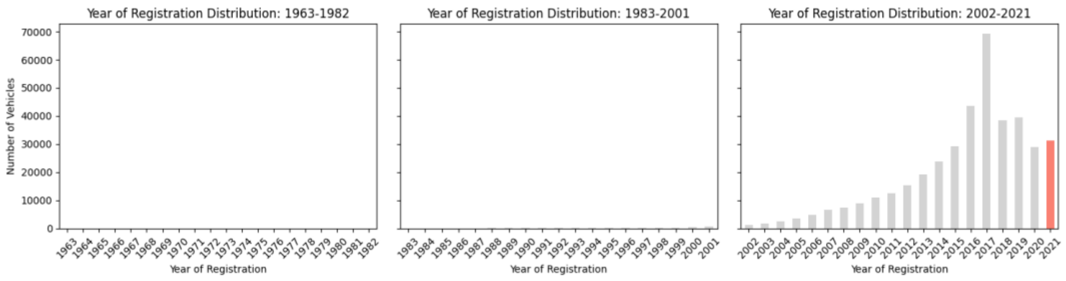
Plot3. Distribution of Car Prices Across Fuel Type

1.1.3. Year of Registration

This numerical feature shouldn't be mistaken with the car's age because a car might be sold, re-register and eventually get an updated year_of_registration and reg_code. The Dtype attribute for this feature is float64 which indicates the values that are entered as decimal numbers. There are 33,311 null values for this feature in the original data set. The values of this feature are in a range of 999. to 2020. Plot4. displays the distribution of registration year of vehicles before data preprocessing, we can observe that the majority of records are in the time frame of 2002-2020, and especially in the year 2015-2019 which might be the market demand or technological advancements results. We also have 31,277 records that we want to assign year_of_registration of 2021 to them because of the value of their vehicle_condition which is NEW. We can observe that these records have a relatively large proportion of the data and are shown in Plot5., which displays the year_of_registration distribution after the data processing procedures.



Plot4. Distribution of Registration Year before Data Preprocessing

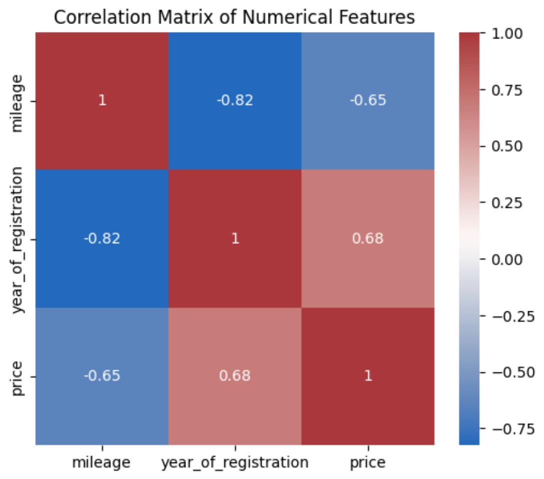


Plot5. Distribution of Registration Year after Data Preprocessing

1.2. Analysis of Predictive Power of Features (2-3)

1.2.1. Correlation of Numerical Features

The correlation matrix (Plot6.), illustrates the relationships between numerical features in this dataset.
As we can observe, there is a moderate negative correlation (-0.65) between features **mileage** and price which indicates that mileage has an important role in predicting the target value, price which is as expected in the real-world market. Vehicles with high amounts of mileage tend to have lower prices because the mileage represents the amount that the given vehicle is used. Between **year_of_registration** and price, there is a moderate strong correlation (0.68). Although, we know that year_of_registration is not necessarily representative of the vehicle's age, but we still can assume that vehicles registered in recent years (probably newer ones) generally have higher prices due to their less usage and the modern versions of equipment. These correlations highlight the significant role of mileage and year_of_registration as predictive features for price.

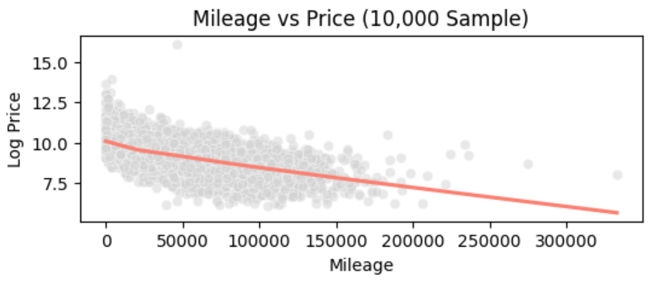


	mileage	year_of_registration	price
mileage	1	-0.82	-0.65
year_of_registration	-0.82	1	0.68
price	-0.65	0.68	1

Plot6. Correlation Matrix of Numerical Features

1.2.2. Mileage vs. Price

This scatter plot (Plot7.) displays how the target feature price, varies across different mileage values in a selected sample of 10,000 records in the dataset. The log transformation is applied to the y-axis of the plot to provide us with a clearer view of the price variable. The regression line highlights the negative relationship of these features with each other. As mileage increases, we can see the decline in the log- transformed price, which means vehicles with high amounts of mileage tend to have lower prices. Also, price reduction is more noticeable at lower amounts of mileage due to the steeper slope that we can observe in the regression line. It means that in vehicles with very high amounts of mileage, the price is still decreasing but at a smoother rate, indicating mileage has less effect on the price at higher values.

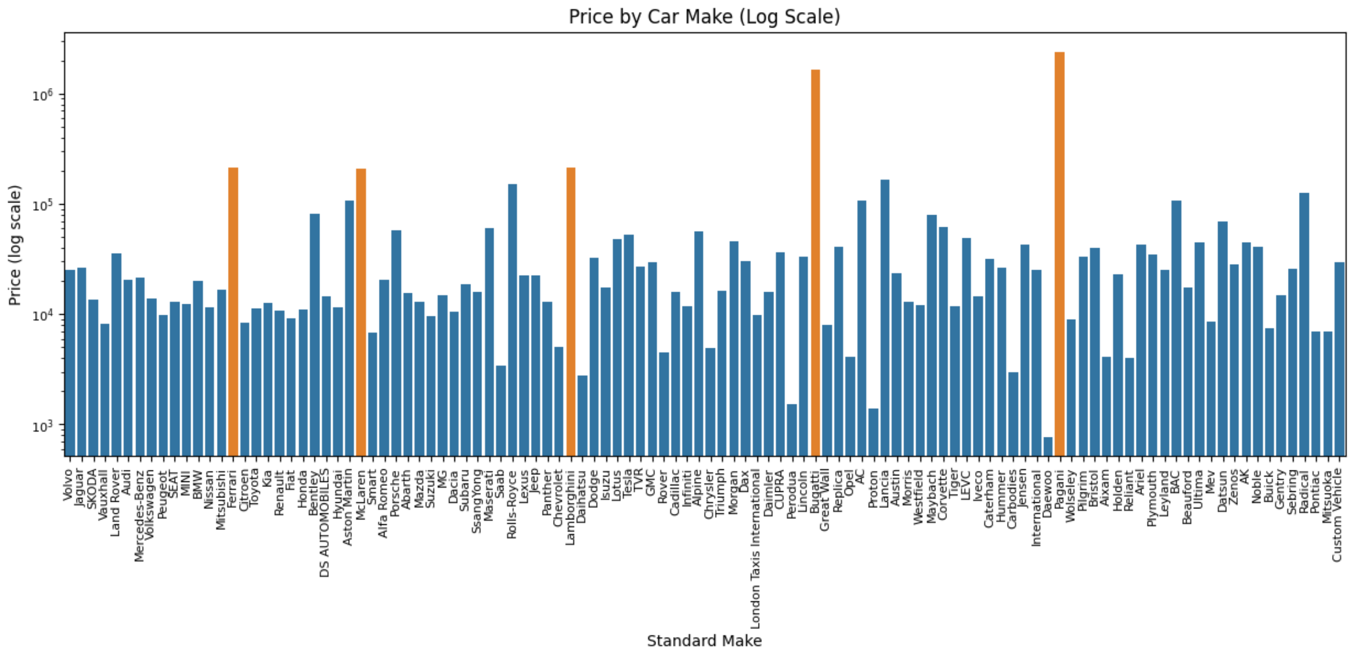


Plot7. Distribution of Price across Mileage

1.2.3. Standard Maker vs. Price

This bar plot (Plot8.) illustrates the price distribution across different car companies. The log transformation is applied to y-axis of the plot to provide us a clearer view of the price variable. The clear distinctions in average price levels across different manufacturers allow us to leverage each manufacturer's price patterns and that makes standard_make a good feature to consider for predictive modelling purposes.

This plot also indicates the dominance of high-priced and luxury brands including Pagani, Bugatti, Lamborghini, Ferrari, and McLaren which are represented by orange colour. Although Pagani has the highest observed price among all manufacturers, this observation isn't entirely reliable because we only have one record of this company and this value may not represent the true average price of this company so, its price average is biased due to insufficient records in comparison other companies with a larger number of records in the dataset which provide a more reliable representation of their price distribution.

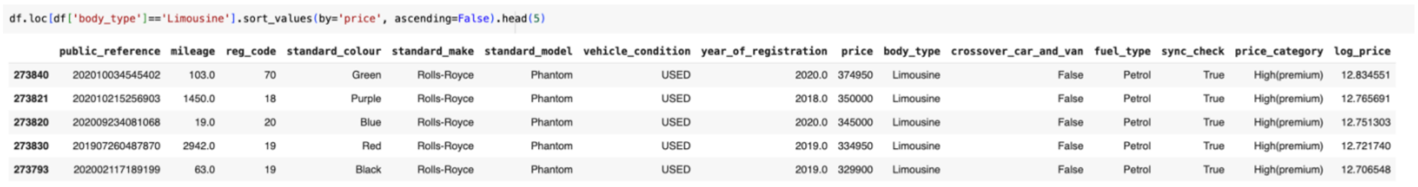


Plot8. Average Price of Manufacturers

1.3. Data Processing for Data Exploration and Visualisation (2-3)

1.3.1. Data Filtering for Data Exploration

In several parts of this project, data filtering was used to narrow down the records that we wanted to focus on as a group of data with given conditions to help us have a thorough comprehension of the whole dataset in data exploring stage. One of the examples of implementing this strategy is shown below (Code Snippet1.). In this part, after grouping the average price of vehicles based on the body_type category, we got Limousines as the most expensive vehicles in the output. In the next stage, a specific condition such as Limousine for the body_type is set and the records are sorted based on their average prices. While it is expected that one of the high-priced coupes, convertibles or campers has the highest price, a Green Phantom Rolls-Royce limousine with price of £374,950 has this title(Fig2.).



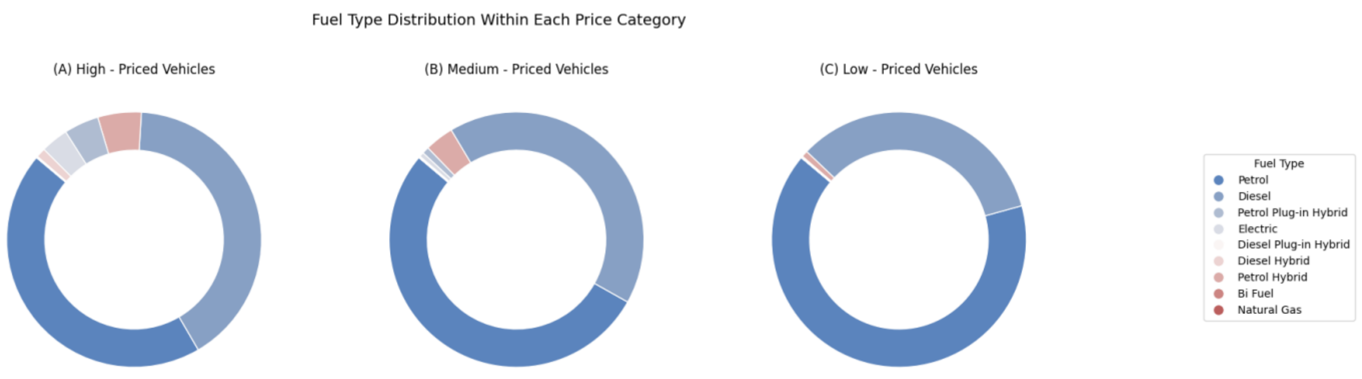
```
df.loc[df['body_type']=='Limousine'].sort_values(by='price', ascending=False).head(5)
```

```
        public_reference  mileage  reg_code  standard_colour  standard_make  standard_model  vehicle_condition  year_of_registration   price  body_type  crossover_car_and_van  fuel_type  sync_check  price_category  log_price
273840  202010034545402     103.0        70            Green    Rolls-Royce         Phantom               USED                2020.0  374950  Limousine                  False     Petrol        True  High(premium)   12.834551
273821  202010215256903    1450.0        18           Purple    Rolls-Royce         Phantom               USED                2018.0  350000  Limousine                  False     Petrol        True  High(premium)   12.765691
273820  202009234081068      19.0        20             Blue    Rolls-Royce         Phantom               USED                2020.0  345000  Limousine                  False     Petrol        True  High(premium)   12.751303
273830  201907260487870    2942.0        19              Red    Rolls-Royce         Phantom               USED                2019.0  334950  Limousine                  False     Petrol        True  High(premium)   12.721740
273793  202002117189199      63.0        19            Black    Rolls-Royce         Phantom               USED                2019.0  329900  Limousine                  False     Petrol        True  High(premium)   12.706548
```

Code Snippet1. An example of Data Processing to Simplify Data Exploration Procedures

Figure2. Green Phantom Rolls-Royce limousine [1]

1.3.2. Apply Binning Method on Price for Data Exploration and Visualisation
Applying binning method on the target variable helped us in both Data Exploration and Visualisation. In this process, the continuous values of price feature are divided into three distinct categories of Low, Medium, and High priced based on their quartile distributions.This Transformation in data visualization provides a clearer perspective on which price category the data points under our analysis fall into Plot9. is an example of how applying binning method help us to make a clear and intuitive visualisation of fuel type distribution within each price range.
In data exploration, it enables us to measure the association of categorical features with the categorized price by measuring the Chi-square method, facilitating a more thorough comprehension of the relationships between these features (Output4.).



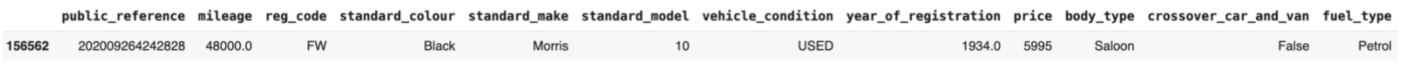
Plot.9. An Example of Data Processing to Simplify Data Visualisation

2. Data Processing for Machine Learning

2.1 Dealing with Missing Values, Outliers, and Noise (2-3)
In this segment, three procedures in data pre-processing which are used to address issues such as missing values and outliers are described. While similar approaches are applied to other features, we focus on year_of_registration and reg_code to have a thorough comprehension of the changes made and maintain consistency.

2.1.1. Handling Missing Values and Dealing with Incorrect Values: Filling Values Based on Other Features
The reg_code and year_of_registration features are filled based on the defined dictionary (year_to_reg_codes) for years 1963-2001. Three decisions are made in this stage based on the condition:
a) If both reg_code and year_of_registration have null values, and “NEW” for the vehicle_condition, we assume the year 2021 which is the year after the last year is recorded in the dataset.
b) If the reg_code value is valid, year_of_registration will be filled based on the relevant reg_code for that record.
c) If the reg_code value is invalid, we look into the year_of_registration feature and by mapping, we assign the relevant value of year_of_registration to the reg_code.

2.1.2. Dealing with Outliers: Dropping the Records
One way to deal with the outliers is by removing them from the original dataset. Although some of these records are valid (Output1.) (Fig.1.), their removal simplifies the process of defining the mentioned dictionary and improves the accuracy of statistical measures by reducing noise and ensuring consistency across the dataset.
There are 44 observed records from years before 1963 which are decided to be removed from the dataset. Removing this amount of data (0.01%), won’t significantly affect the insights derived from the dataset.



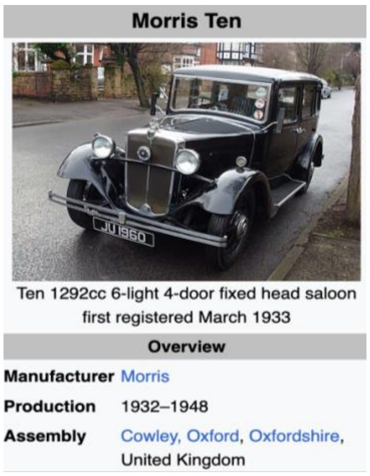
	public_reference	mileage	reg_code	standard_colour	standard_make	standard_model	vehicle_condition	year_of_registration	price	body_type	crossover_car_and_van	fuel_type
156562	202009264242828	48000.0	FW	Black	Morris	10	USED	1934.0	5995	Saloon	False	Petrol

Output1. Example of a Valid Outlier

2.1.3. Handling Missing Values and Dealing with Incorrect Values: Grouping and Imputation Values
By using the dictionary and applying the mapping process, the issue in records that have incorrect or null values of year_of_registration and reg_code is solved. However, this approach falls short when one feature contains a null value while the other has an incorrect value and this is because the dictionary is defined based on only the valid values for both of these features. To resolve this, an imputation function (fill_missing_years) was introduced to fill the year_of_registration based on the standard_make and standard_model features. The logic behind the decision to use standard_make and standard_model features for grouping is their association with year_of_registration. The year_of_registration is linked to those given features because certain vehicle models are produced in specific periods of time. By grouping the dataset by standard_make and standard_model, and filling the year_of_registration with the observed mode in each group (Output.2.), and reapplying the dictionary, there will be no more records with null values in reg_code and year_of_registration.

```
grouped_modes = df.groupby(['standard_make', 'standard_model'])['year_of_registration'].agg(lambda x: x.mode()[0] if not x.mode().empty else None).reset_index()
grouped_modes.columns = ['standard_make', 'standard_model', 'year_of_registration_mode']
grouped_modes
```

Code Snippet2. Data Grouping and Mode Imputation



Morris Ten

Ten 1292cc 6-light 4-door fixed head saloon first registered March 1933

Overview

Manufacturer	Morris
Production	1932–1948
Assembly	Cowley, Oxford, Oxfordshire, United Kingdom

Fig1. Example of a Valid Outlier [2]

```
      standard_make  standard_model  year_of_registration_mode
0                AC           Cobra                     2017.0
1                AK           Cobra                     2011.0
2            Abarth      124 Spider                     2018.0
3            Abarth             500                     2009.0
4            Abarth            500C                     2012.0
...             ...             ...                        ...
1195      Westfield              Se                     2017.0
1196      Westfield             Sei                     1992.0
1197      Westfield           Sport                     1997.0
1198       Wolseley           6/110                     1964.0
1199          Zenos             E10                     2018.0
1200 rows × 3 columns
```

Output2. Groups of Data with Relevant Registration Year

The same grouping procedure based on standard_make and standard_model is applied to other features including mileage, standard_colour, body_type, and fuel_type due to the strong association of these features with the vehicle model. Unlike year_of_registration, these features still have null values after grouping. This is likely due to some records not fitting into the defined groups. To solve that issue, a global mode for categorical features such as standard_colour, body_type, and fuel_type and a global median for numerical features like mileage is defined to be applied in case of having null values after grouping procedures. This ensures that all missing values are handled appropriately after the grouping process.

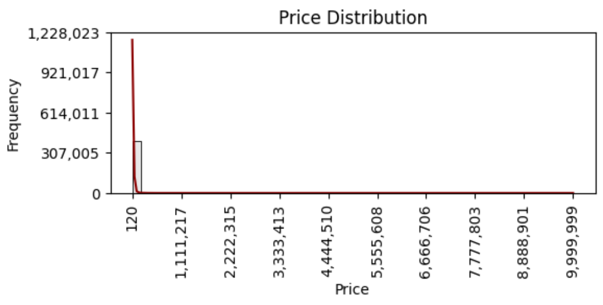
2.2. Feature Engineering, Data Transformation, Feature Selection (2-3)

2.2.1. Feature Engineering: Apply Binning Method on Price

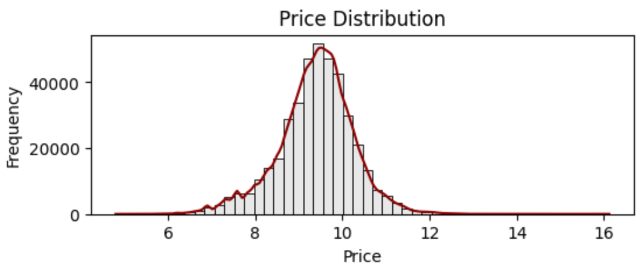
One of the approaches to analyse the association of vehicle prices with other features is through the binning method which divides the continuous numerical values of price into 3 distinct categories: Low, Medium, and High(premium) price. The price_category column is added to the original dataset through this implementation. These categories are defined based on the price distribution percentiles as follows:
1) Low Price Vehicles: Records below the 25th percentile
2) Medium Price Vehicles: Records between 25th and 75th percentiles
3) High Price Vehicles: Records above the 75th percentile
The binning method was applied using this logic because of the significant differences between high-priced vehicles and other records across other features. These kinds of records will cause issues, including bias in the model training process which affects the accuracy and reliability of the predictions in the future so, it is important to treat high-priced vehicles differently.

2.2.2. Data Transformation: Applying Log Transform Method on Mileage

Log transformation is applied on the price feature in this dataset and the transformed values are creating a new column named log_price. By applying this method, we can improve the price distribution normality and reshape it into a more symmetric bell-shaped curve. So, we can have a clearer visualization, valid accuracy and better model performance in later procedures. The plots below illustrate the price distribution before and after applying the log transformation. We can observe positive skewness distribution in Plot10. because of outliers' presence and in Plot11., the range of outliers is reduced and made the curve more balanced.



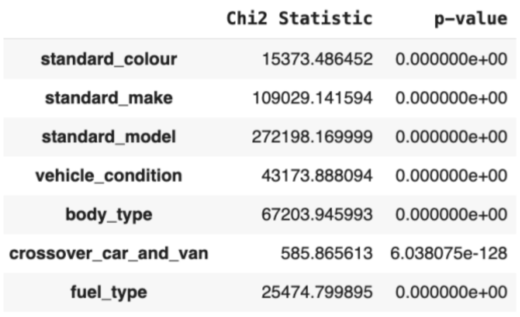
Plot10. Price Distribution



Plot11. Log Transformation Effect on Price Distribution

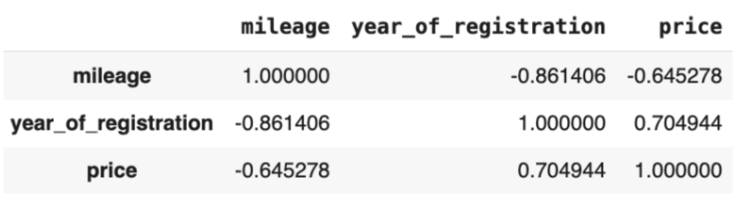
2.2.3. Feature Selection

For this part, the association of features in the dataset is evaluated with the target feature price to ensure which features are actually important for us to be considered in future procedures such as model training.
The association of categorical features are examined with feature price_category and for this purpose, the Chi-squared test is applied to compute the Chi-square statistic and p-value for the hypotheses test of independence of observed frequencies [3]. The assumption for H0, is that there’s no association among the selected features. By having a look to the result of this function (Output3.) and the measured p-values, we can confirm that all of those features have significant impacts on our target price. Also, based on the Chi-squared scores in this output, we can find out that the standard_model has the highest association with the price.
The association of numerical features is examined with feature price and to do so, the correlation function by using Spearman method is applied. Unlike Pearson correlation, Spearman correlation can make us capable of capturing non-linear relationships among the features. This function result displays that both mileage and year_of_registration features have a strong negative correlation with each other (Output4.). As a result, these given features are also should be kept for future procedures.



	Chi2 Statistic	p-value
standard_colour	15373.486452	0.000000e+00
standard_make	109029.141594	0.000000e+00
standard_model	272198.169999	0.000000e+00
vehicle_condition	43173.888094	0.000000e+00
body_type	67203.945993	0.000000e+00
crossover_car_and_van	585.865613	6.038075e-128
fuel_type	25474.799895	0.000000e+00

Output3. Association of Categorical Features with Target Feature of price_category



	mileage	year_of_registration	price
mileage	1.000000	-0.861406	-0.645278
year_of_registration	-0.861406	1.000000	0.704944
price	-0.645278	0.704944	1.000000

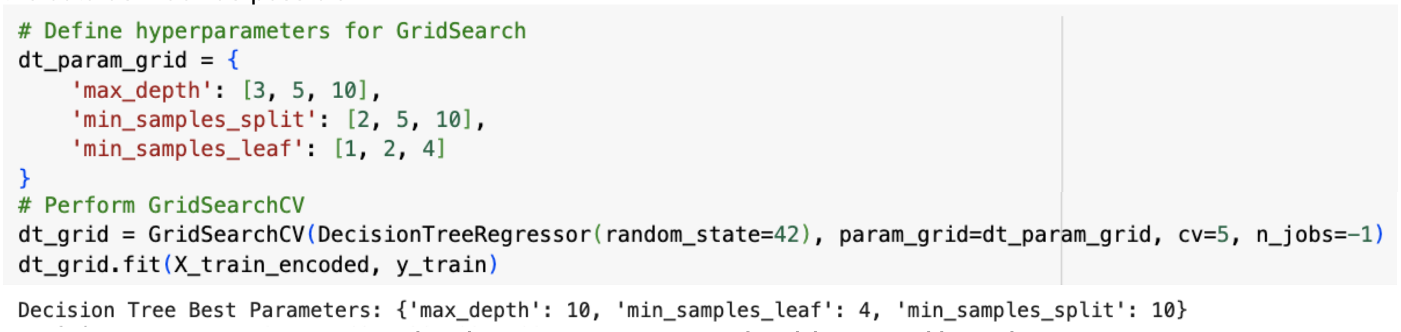
Output4. Association of Numerical Features with Target Feature of Price

3. Model Building

In this section, the process of **implementing** three ML predictive models and steps of **configuring** each model, including encoding, scaling, and performing hyper parameter tuning by using **grid search** is described. Finally, these models were **ranked** and **selected** based on considering different parameters such as validation metrics and time consumption. In this section, log transformation is applied on mileage and price features of each model.
Since, these models can’t handle the features categorical values, these values should be converted into a numerical format by using encoding approaches. OneHotEncoder is applied on the two distinct categories in vehicle_condition and crossover_car_and_van features by converting them into two columns with binary values and removing one of those columns. In this case we keep vehicle_condition_NEW and crossover_car_and_van_True columns. Applying TargetEncoder on other categorical features is a better choice due the high number of categories, these features contain. Applying other encoding approaches like label encoding will lead to adding a large number of columns to the dataset which is not the most optimised way due to the high dimensionality related problems that will be caused. Target encoding measures the price variable mean for each category and then replace the category with the corresponding mean value. To prevent target leakage and ensure that no information from the test set influences the training process, the dataset is split into training and test subsets. The target encoder is fitted only on the training set and then for transformation, it is applied to the test set.

A) Decision Tree

This model uses grid search with 5-fold cross-validation to tune hyper parameters such as tree depth, number of samples that are required to split an internal node, and number of samples that are required to be at a leaf node. Decision tree model identifies the optimised configuration for predicting the target variable. Among all the values that we considered for the parameters to build this model, the following model with these values for each parameter was selected as the best Decision Tree model, regarding our aim to avoid overfitting while capturing the information from the data as much as possible.



```
# Define hyperparameters for GridSearch
dt_param_grid = {
    'max_depth': [3, 5, 10],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
# Perform GridSearchCV
dt_grid = GridSearchCV(DecisionTreeRegressor(random_state=42), param_grid=dt_param_grid, cv=5, n_jobs=-1)
dt_grid.fit(X_train_encoded, y_train)

Decision Tree Best Parameters: {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 10}
```

Code Snippet3. Hyper Parameters of Decision Tree Grid Search

B) Linear Regression
This model explores the linear relationship between the features and price variable. Due to the model sensitivity to the magnitude of the features, before the process of implementing and training the model, feature scaling is applied on the encoded dataset by using StandardScaler. In the linear regression model we don't have the process of hyper parameter tuning by applying grid search but predictions will be made on the test set, after the model is trained and cross-validation with 5 folds is performed on the dataset.

```
cv_r2_scores = cross_val_score(lr, X_train_scaled, y_train, cv=5, n_jobs=-1)
cv_mse_scores = cross_val_score(lr, X_train_scaled, y_train, cv=5, n_jobs=-1)
```

Code Snippet4. Cross-Validation of Linear Regression

C) K- Nearest Neighbour
This model is a distance-based model, so before the model implementation and training process, feature scaling is applied to the encoded dataset by using StandardScaler. By using grid search with 5-fold cross-validation, hyper parameters for assigning the number of closest neighbours for each sample, distance metric, and adding weights to the data points are tuned. We can observe that among all the values that we considered for these parameters to build this model, the best values are displayed below.

```
# Define hyperparameters for GridSearch
knn_param_grid = {
    'n_neighbors': [3, 5, 10],
    'weights': ['uniform', 'distance'],
    'p': [1, 2]  # Manhattan (1) or Euclidean (2) distance
}
# Perform GridSearchCV
knn_grid = GridSearchCV(KNeighborsRegressor(), param_grid=knn_param_grid, cv=5, scoring='r2', n_jobs=-1)
knn_grid.fit(X_train_scaled, y_train)
```

```
KNN Best Parameters: {'n_neighbors': 10, 'p': 1, 'weights': 'distance'}
```

Code Snippet 5. Hyper Parameters of K-Nearest Neighbours Grid Search

4. Model Evaluation and Analysis

In this section, we evaluate the performance of these three models. The models' prediction outcomes and the time consumption (in seconds) are stored in separate datasets to simplify the models comparison with each other.

4.1. Coarse-Grained Evaluation/ Analysis (3)

We used different evaluation metrics such as Mean Square Error (MSE) and R^2, to compare the performance of the implemented prediction models. Additionally, in this section, the performance of all implemented models both before applying the log transformation and after applying it on price and mileage features is displayed.
The results gained are as follows:

	Model	MSE	RMSE	R^2	Time Consumption
0	Decision Tree Before	2.530268e+08	15906.816147	0.548497	76.028746
1	Linear Regression Before	2.843447e+08	16862.523882	0.492613	2.620239
2	KNN Before	2.232380e+08	14941.150160	0.601652	655.111527

Table1. Models' Performance before Data Transformation

	Model	MSE	RMSE	R^2	Time Consumption
0	Decision Tree After	0.063700	0.252389	0.912981	75.883015
1	Linear Regression After	0.136816	0.369886	0.813101	1.713258
2	KNN After	0.046862	0.216476	0.935984	475.159436

Table2. Models' Performance after Data Transformation

The reason behind the high amount of MSE that is observed in the implemented models before the data transformation procedures, is because of the high values of these features themselves. Therefore, by applying this method, as mentioned before, the impact of outliers and skewness in these features is reduced and we can observe model's ability to capture the patterns in the data while reducing the impact of outliers.

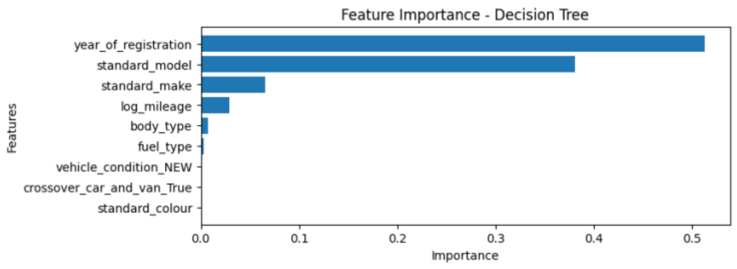
The results highlight the improvements across almost all of the models after the data transformation. The **Decision Tree**'s performance changed slightly after the transformation which illustrates that this model is less sensitive to the skewness and outliers in the features and was already capturing the patterns in the dataset effectively. It still can be a good option due to its interpretability and moderate computational costs.

Although **Linear Regression** had the lowest time consumption among the models, its low scores indicate that it was not the most efficient model to display the relationship between the features and the target variable price. If the simplicity and speed are more important factors than the accuracy of the prediction for us, this model will be preferred.
The time consumption for **K-Nearest Neighbours** remained higher than other models due to the computational cost of this distance-based algorithm and if our main priority is high accuracy, the computational cost of this model can be waived.
Overall, in this case, KNN had the best performance in comparison to other models due to being successful in capturing the complex and non-linear relationships while it was computationally inefficient.

4.2. Feature Importance (2-4)

In this section, we use this technique and assign scores to input features to our predictive models that indicate the relative importance of each feature [3]. After fitting each model on the dataset, the importance scores are summarized for each input feature and by drawing a bar chart, we can have a better comprehension of the relative importance of the features.

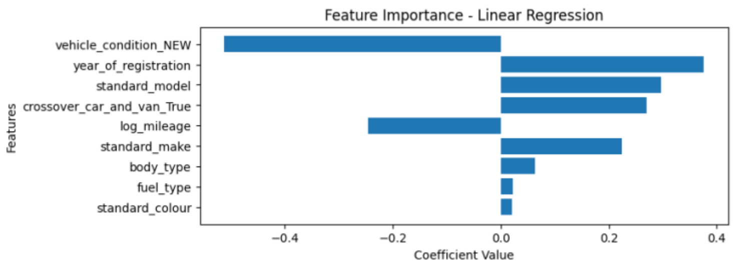
In **Decision Tree** algorithms, importance scores are offered based on the reduction in the criterion used to select split points like Gini or entropy. After fitting the model, the model provides a feature_importance_ property that can be accessed to retrieve the relative importance scores for each input feature [4]. Plot12. Displays that year_of_registration and standard_model were the most important features in this model and had a significant effect on the price prediction.



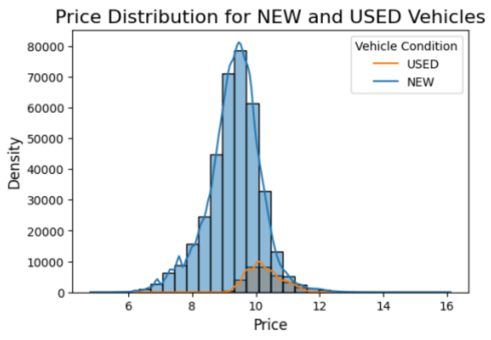
Plot12. Feature Importance in Decision Tree Model

In this process, the **Linear Regression** model finds a set of coefficients to use in the weighted sum of the input values in order to make a prediction and these coefficients can be used directly as feature importance. We apply the coef_ function which reports the coefficient value for each feature [4]. We expect that used vehicles to have lower prices, but despite our expectation, we can observe that vehicle_condition_NEW has negative coefficient in the linear regression model equation.

We can have multiple assumptions regarding this matter:
- Due to the presence of luxury cars with. Higher prices in the USED vehicle category, the prices of NEW vehicles do not appear significantly higher. As a result, being a new vehicle does not necessarily lead to a higher price and eventually a positive coefficient in the model equation. But, we can deny this assumption by looking at Plot14. which displays that, overall, new vehicles are more concentrated in the higher price ranges.
- Another assumption can be due to the imbalanced data in the dataset; having significantly more used vehicles than new ones will lead to assigning more weight to the used vehicle patterns and skewing the coefficient for vehicle_condition_NEW. This assumption might be more reliable based on the observed data patterns.
log_mileage also has a significant negative coefficient displaying that higher mileage will lead to price reduction, meaning that if the usage of vehicles is high and has a higher amount of mileage as a result, the price will be reduced.

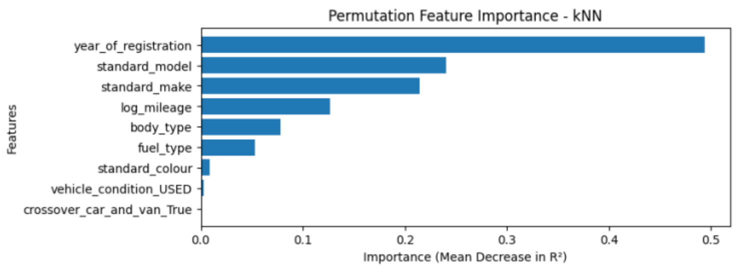


Plot13. Feature Importance in Linear Regression



Plot14. Price Distribution over Vehicle Condition

K-Nearest Neighbours is a nonparametric machine learning algorithm, meaning that unlike the Linear Regression algorithms that we can estimate the coefficients of the line equation, does not assume anything about the form of the mapping function other than patterns that are close [4]. Therefore, for this model, permutation_importance function is used which is a technique for calculating relative importance scores that is independent of the model used [4]. This plot also indicates that the year_of_registration effectively influences on the vehicle price.

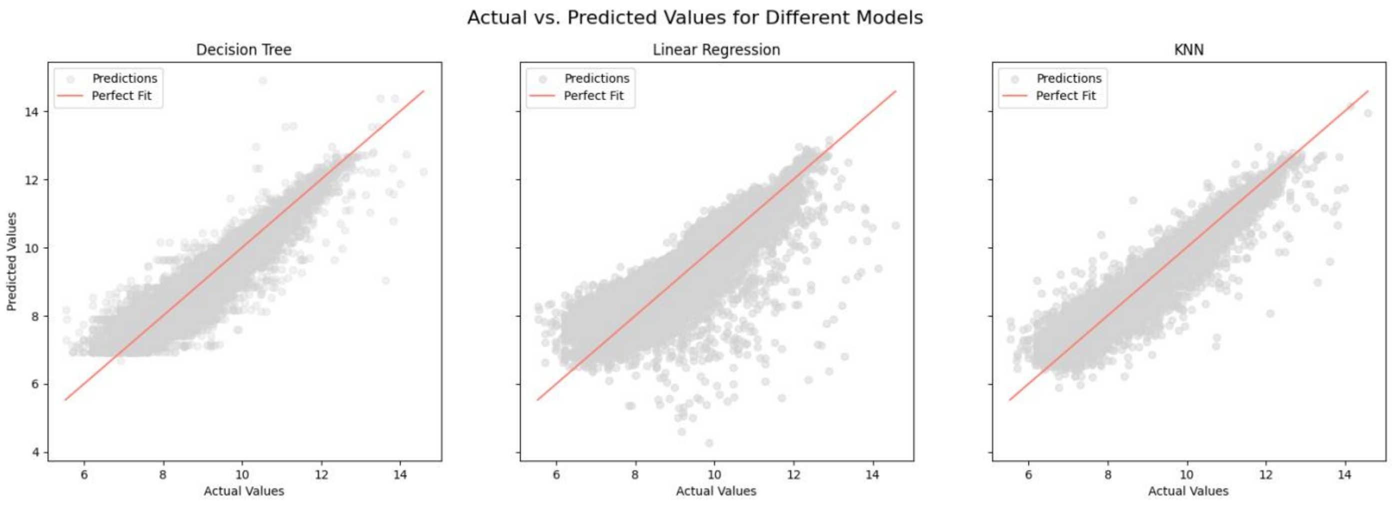


Plot15. Feature Importance in K-Nearest Neighbours

4.3. Fine-Grained Evaluation (2-4)

4.3.1. Visualising the Residuals Scatter Plot: In this section, a plot comparing the actual and predicted values for each implemented model is presented to illustrate their predictive performance at the instance level. Across all models we can see that almost most of the predicted data points are close to the perfect fit line, indicating that all the models performed reasonably.

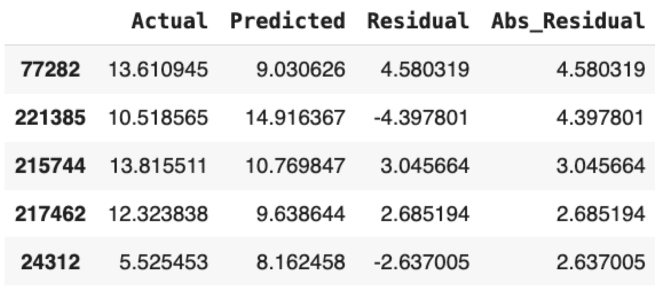
Decision Tree shows a wider spread at the edges, suggesting its effectiveness is reduced at those regions. Linear Regression model struggles in capturing non-linear relationships and as a result, we can see sparser predicted data points around the perfect fit line. However, **K-nearest neighbour**, demonstrates better adaptation to the data patterns. Overall, it can be concluded that the Decision Tree and KNN models, perform better at capturing data variability compared to Linear Regression, although each of these models has their own limitations.



Plot 16. Comparing Actual and Predicted Values through Different Models

4.3.2. Residual Analysis: In this section, for each model the instances with high absolute residual amounts are displayed. The large amounts of prediction errors might be caused due to factors such as outliers, noise, and the model's inability to train the data well or handle complex relationships within the data.

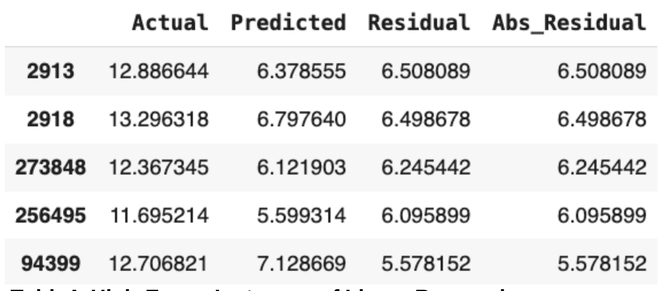
Although Decision Trees are good at capturing non-linear relationships, they show inefficiency in capturing the variability in certain instances and their reliance on strict splits can lead to inaccuracies. The possibility of overfitting might also lead to observe these large residuals.



	Actual	Predicted	Residual	Abs_Residual
77282	13.610945	9.030626	4.580319	4.580319
221385	10.518565	14.916367	-4.397801	4.397801
215744	13.815511	10.769847	3.045664	3.045664
217462	12.323838	9.638644	2.685194	2.685194
24312	5.525453	8.162458	-2.637005	2.637005

Table3. High-Errors Instances of Decision Tree

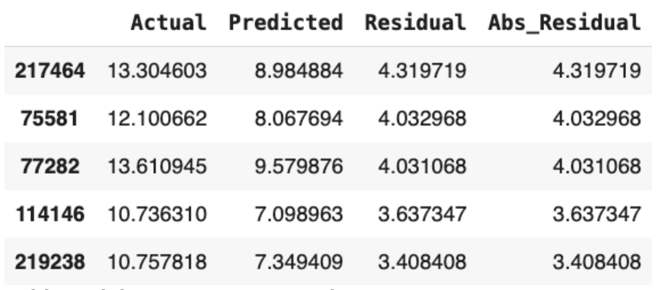
Linear Regression model is limited by the assumption of linear relationships between the features and the target variable. The Large residual values observed in this table suggest that this assumption is not valid and the relationships between these data points might be non-linear. Among all models, linear regression shows the largest residuals (up to 6.51) highlighting its inefficiency in capturing complex patterns and poor performance.



	Actual	Predicted	Residual	Abs_Residual
2913	12.886644	6.378555	6.508089	6.508089
2918	13.296318	6.797640	6.498678	6.498678
273848	12.367345	6.121903	6.245442	6.245442
256495	11.695214	5.599314	6.095899	6.095899
94399	12.706821	7.128669	5.578152	5.578152

Table4. High-Errors Instances of Linear Regression

KNN relies on finding nearby data points where features have similar target values. The high residual values observed in this table suggest that the KNN model struggles with instances that are located in regions where data points show irregular patterns. However, this model shows the lowest residuals (4.32) in comparison to other models, suggesting that this model generally adapts well to data variability.



	Actual	Predicted	Residual	Abs_Residual
217464	13.304603	8.984884	4.319719	4.319719
75581	12.100662	8.067694	4.032968	4.032968
77282	13.610945	9.579876	4.031068	4.031068
114146	10.736310	7.098963	3.637347	3.637347
219238	10.757818	7.349409	3.408408	3.408408

Table5. High-Errors Instances of KNN

References

[1] WrapStyle, “Rolls Royce Phantom Green Chrome Wrap,” WrapStyle, 2023. [Online]. Available: https://www.wrapstyle.com/gallery/58-rolls- royce-phantom-green-chrome-wrap.

[2] Wikipedia contributors, “Morris Ten,” Wikipedia, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Morris_Ten.

[3] J. Brownlee, "Calculate Feature Importance With Python," *Machine Learning Mastery*, [Online]. Available: https://machinelearningmastery.com/calculate-feature-importance-with-python/.

[4] J. Brownlee, "Parametric and Nonparametric Machine Learning Algorithms," *Machine Learning Mastery*, [Online]. Available: https://machinelearningmastery.com/parametric-and-nonparametric-machine-learning-algorithms/.

Chatgpt and Grammarly tools are used in this text for text revising and code optimisation.