# Task 3: Critical Reflection

## Introduction

This report provides a critical review of the processes and strategies I used during the data analysis task. It explains why I chose the data analysis methodologies I used, the obstacles I faced while applying them, how the approaches performed in relation to my analytical question, and lessons I've learned for future projects.

The dataset chosen for this exercise is aimed at car price prediction. The dataset consists of up to 25 features that are to be investigated to determine whether they influence the price variable. The dataset can be downloaded from Kaggle via this [link](https://www.kaggle.com/datasets/hellbuoy/car-price-prediction). The data comprises of both numerical and categorical features which are to be investigated to uncover their relationships with each another and the target variable ‘price’.

The data analysis section seeks to answer the following analytical questions: ***Which variables are significant in predicting the price of a car? And how well do those variables describe the price?***

## Data Analysis

Data analysis involves all the steps undertaken to gain a better understanding of the data. It involves data wrangling, transformation, visualization and modelling to discover useful insights that can influence decisions. The first step in the data analysis process was importing all the necessary libraries that would facilitate the analysis. This was followed by displaying a section of the dataset so as to get a glimpse of the kind of data I was dealing with. The describe() function was used to provide statistical summaries of the numerical variables. This revealed that the features had different distributions, calling for standardization in later steps.

## Data Pre-processing

Data pre-processing entails the data transformation steps carried out to make the data understandable and usable. It facilitated easier data handling and enhanced data visualization. The data pre-processing step involved first dropping of the ‘car\_ID’ feature. Being a unique number used to identify the vehicles, I decided that it would not be relevant in the prediction. This was then followed by separation of numerical variables from categorical variables. This was necessary because they need to undergo different pre-processing techniques.

Numerical variables required standardization while categorical features needed to be transformed into a numerical format that can be understood by machine learning models. I used the sci-kit-learn StandardScaler class to transform the numerical variables. Most of the features in the dataset demonstrated a uniform distribution. The Standard Scaler was thus ideal for feature scaling as it assumes the data exhibits normal distribution and independently centres and scales each feature around 0, with a standard deviation of 1. Categorical feature encoding was done in later steps to allow for data pre-processing and data visualization.

### Feature Engineering

Feature engineering is the process of modifying features through addition, deletion, combination and mutation. It is aimed at improving the machine learning model training accuracy and efficiency, leading to better models.

Further analysis of the categorical columns reveals that the ‘carName’ feature contains information about the brand and model information of the vehicles. Through feature engineering, this column was broken down into two new columns, ‘Brand’ and ‘Model’, and then discarded. This transformation resulted in missing values for the model columns. The rows with missing information were dropped because they were very few, i.e, only 2 rows. A number of brands were also misspelled or written in different formats. It was therefore necessary to streamline the names so as to have the same representation for each unique brand. The ‘Model’ feature was diverse with more than 50 unique categories. I decided to drop this column and use only the brand feature so as to reduce the dimensions of the final dataset.

### Data Visualization

Exploratory Data Analysis involved both univariate and bivariate analysis to uncover the distribution of variables their interrelations. This step begins with a visualization of how the ‘Brand’ feature is distributed across the dataset and how it relates to the target variable ‘price’. Toyota is the most popular brand, followed by nissan and mazda. The least common brands are renault and mercury. The most expensive brands are jaguar, buick and Porsche while the cheapest brands are chevrolet and dodge. The popularity of a car seems to have no relation to its price. For instance, the chevrolet is one of the cheapest cars, but it is also one of the least popular cars and the jaguar is the most expensive and it is also one of the less popular brands.

Distribution of categorical variables were further analysed using pie chart subplots. Most of the features had one category dominating the distribution. For instance, gas is more popular that diesel, most engines were located at the front and most cars have 4 cylinders. The final plot on categorical variables were box plots which portrayed the distribution of each of the features relative to the ‘price’ variable. While most of the categorical variables influenced the price of the car, the ‘doornumber’ feature did not show much disparity in prices. This feature was later dropped as it was not going to be significant in the prediction of the price variable.

Visualization of how the numerical variables relate to price revealed the following:

* The symboling variable is categorized into 5 distinct groups and can be considered a categorical variable.
* Most features have a linear relationship with price e.g. carlenght, carwidth, enginesize and horsepower which were directly proportional and citympg and highwaympg which were inversely proportional to price. The other variables have a non-linear relationship.
* The compression ratio variable is clustered into two groups. Further investigation reveals that it in grouped according to the fuel type.
* The price variable follows a normal distribution.

### Feature Correlation

The dataset has multiple highly correlated fearures such as:

1. wheelbase has high positive correlation with carlength,carwidth and curbweight
2. carlength has high postive correlation with curbweight
3. carlength has negative correlation with highwaympg
4. carwidth has high postive correlation with curbweight and engine size
5. enginesize has high positive correlation with horsepower
6. curbweight has high positive correlation with engine size and horse power, negative correlation with highwaympg
7. horsepower has negative correlation with citympg and highwaympg
8. citympg and highwaympg are highly correlated

Highly correlated features contribute less in predicting the output but increase the computational cost. In order to escape the curse of dimensionality, I set a threshold of 0.90, and dropped features with a correlation beyond the set threshold.

### Recursive Feature Elimination

Recursive Feature Elimination (RFE), is a feature selection procedure that minimizes the complexity of a model by focusing on the most important features while eliminating the less important ones. The selection method eliminates less important features one by one until the optimum number is reached to ensure top performance. This technique was used to select only the top 10 features from the numerical variables. As such, the following variables were dropped: ‘peakrpm’, ‘symboling’, ‘citympg’ and ‘boreratio’.

### Categorical Feature Encoding

The categorical features were later transformed using the OneHotEncoder into numerical representations and then concatenated with the numerical features to create the final dataset. OneHotEncoding maps a column of categorical features into binary vectors unique to each available category. Therefore, for each row, 1 is assigned for the feature representing the row’s category and the rest of the features marked 0.

### Modelling

After that, the dataset was separated into training and test sets, followed by machine learning modelling. The training set is used to train the machine learning model, whereas the test set is a subset of the data used to see how well the model can predict the target variable. The data is divided into two groups: 80% training and 20% test.

### Model Comparison

The algorithms selected were regression-based models such as the linear regression model, the random forest model and the extreme gradient boosting model. The evaluation criteria applied was the r2 score and the mean percentage error.

The best performing model was the Extreme Gradient Boosting (XGB) model. It has the highest r2 score and the lowest mean percentage error. The XGBoost model performs well because it is an ensemble model that combines the estimates from a set of simpler and weaker models to get a single prediction. Aside from the base learners, XGBoost has its own objective function that minimizes the difference between the actual values and predicted values.

## Challenges faced

1. The highly correlated nature of the dataset features. This made it difficult to set a threshold without risking losing important data points.
2. A large number of features complicated the feature selection process, imposing the need to implement dimensionality reduction techniques.

## Lessons learned for future works

1. It is important to first understand the data by looking at the number of features, number of entries, data types of features and the nature of the target variable.
2. Utilize built-in statistical libraries to provide summary statistics and visualize the information to provide a clearer picture of the data.
3. Gauge feature importance by looking at how they relate to the target variable and drop highly correlated features. It is also possible to apply feature selection techniques to choose the features that influence the model the most
4. One of the strategies that can be used to improve model performance is bagging different models to boost the performance to the overall model.