**Project Summary**

* **Objective:** Car price prediction
* **Approach:** UtilizingMulti-Linear Regression
* **Target variable:** Price
* **Programming Language:** Python
* **Deployed Python Libraries:** Numpy, Pandas, Matplotlib, Seaborn, Statsmodels, SKlearn
* **TABLE OF CONTENTS**

1. Getting acquainted with the data
2. Exploration results after loading the data in Python
3. Data Cleaning and Null Value Handling
4. Preprocessing
5. Feature Selection
6. Prediction Model with SKLEARN

* **Model\_1** - Dataset Utilized Without Any Scaling
* **Model\_2** – Applying the Logarithm to the Target Variable (Carprice)
* **Model\_3** – Incorporating Scaled Features and Applying the Logarithm to Carprice

1. Result summary
2. **GETTING ACQUAINTED WITH THE DATA**:
3. Upon initial examination of the dataset, numerous observations are marked with the symbol "?"—an indication of missing or unknown values.
4. Notably, certain features present numerical values expressed in words. For instance, the "number of doors" attribute includes entries like "four," and the "drive-wheel" feature has instances represented as "4wd."
5. **EXPLORATION RESULTS AFTER LOADING THE DATA IN PYTHON**:
6. The dataset comprises a total of 206 observations, providing a foundational understanding of the volume of data available for analysis.
7. The dataset contains invalid observations (?), and a strategy to enhance data quality involves eliminating rows with at least one null value, excluding those associated with normalized losses. In the event that the normalized losses feature contains an invalid observation, but there are no other invalid entries in the same row, that specific row will be retained rather than deleted. This decision is driven by the fact that the normalized losses feature has 41 missing values, constituting 20% of the dataset. Removing such a significant portion of the data, given its potential impact on predictive power, is deemed undesirable.
8. **DATA CLEANING AND NULL VALUE HANDLING**
9. Null values across the dataset have been successfully addressed(removed), with the exception of the "Normalized Losses" feature.
10. In cases where null values are exclusively present in the "Normalized Losses" column and not in any other column, the corresponding rows have been retained in the dataset. This targeted approach ensures preservation of valuable information while addressing missing data in a specific feature.
11. Following the removal of null values, our dataset now consists of 193 observations. This constitutes the elimination of 6% of the data, a proportion deemed inconsequential for the model's predictive efficacy.
12. The car brand "Renault" has been entirely removed from the dataset post-null value elimination. This decision was prompted by the absence of peak-rpm values in both of its two observations. Peak-rpm, a potent predictor (identified through optimal feature selection in Python), cannot be imputed arbitrarily. Furthermore, the limited dataset size for Renault (only two observations) renders it insufficient for effective model training.
13. **Normalized losses** – The feature "Normalized Losses" has been excluded due to its substantial 20% missing observations. Given its inability to contribute significantly to model training, its removal was deemed necessary.

* Techniques attempted before exclusion:

1. *Manual Calculation Formula*: The dataset lacked essential information required for the manual calculation of normalized losses.
2. *Replication from Complete Observations*: Efforts to replicate values from cars with complete data and identical observations were unsuccessful. This exhaustive exploration left no viable alternative but to eliminate the feature from the dataset.
3. **PREPROCESSING:**

Preprocessing involves two stages:

**Stage-1**: *Encoding Categorical Values into Numerical Form* - In the initial step, we encode categorical values by assigning dummy variables, transforming them into a numerical format. This process ensures that all values are represented in a standardized numerical form.

**Stage-2**: *Converting Observations to Integer or Float Data Types* - Following the transformation of all observations into numerical values, the subsequent step involves converting them into either integer or float data types. This is imperative as regression models can only be constructed using these specific data types. The uniformity in data types enhances the compatibility of the dataset with regression modeling techniques.

**Stage-1:**

1. Before proceeding with data type changes, it is imperative to confirm the absence of categorical variables in the dataset. Assigning dummy variables is recommended in such instances.
2. Utilizing the "LabelEncoder" function from SKlearn facilitates the assignment of dummy variables.
3. Dummy variables are created in new columns, distinguished by adding the suffix "label" to the column names (e.g., make: make\_labelled).
4. Once dummy variables are incorporated, the original source column can be removed, as it becomes redundant for prediction purposes.

**Stage-2:**

1. All observations are now in numerical form.
2. Feature values are adjusted to either integer or float data types based on their respective values (e.g., length: 168.8 should be float, engine-size: 130 should be integer).
3. With the data types standardized to either integer or float, the dataset is prepared for feature selection.
4. **FEATURE SELECTION:**
5. Identifying features that influence the car price, the target variable, is crucial.
6. Statsmodels package is employed to identify features affecting the target variable, with potential positive or negative impact.
7. The Statsmodels OLS (Ordinary Least Squares) function is instrumental in selecting optimal features for the model.
8. Fitting the dataset into the Statsmodels OLS function yields the model. The R2 value provides a model score, with the initial model featuring all the dataset's features achieving a score of 90.8%.
9. Examining the P-value of each feature helps eliminate any with a P-value less than 0.1 for the regression model.
10. Features with a P-value exactly at 0.1 are cautiously retained without significant impact on model performance.
11. Creating a new model with optimal features (P-value<0.1) results in a model score of 89.9%, indicating that the selected features sufficiently explain the variability in car prices.
12. **Prediction Model with SKLEARN**

* **Model\_1 - Dataset Utilized Without Any Scaling**

1. The dataset maintains its original indexation post the removal of missing values. To ensure accuracy in the train-test split, we reset the index to align with the exact number of rows.
2. With the index reset, the dataset undergoes an 80-20 split, allocating 80% for training and 20% for testing. This approach enhances the model's predictive power by providing ample samples for training.
3. Following the split into train and test sets, the training dataset is fitted into the SKlearn LinearRegression model.
4. *Model\_1 yields the following results*:
   1. Scores on the training set:
      1. R2 (model score): 89% ((indicating an 89% prediction accuracy on the training set)
      2. Mean Absolute Error (MAE): 1791.76
      3. Mean Squared Error (MSE): 5319425.17
   2. Scores on the training set:
      1. R2 (model score): 89%
      2. MAE: 2508.34
      3. Mean Squared Error (MSE): 12561190.91

* **Model\_2 – Applying the Logarithm to the Target Variable (Carprice)**

1. A scatter plot on carprice (target variable) using the Seaborn (SNS) library reveals a non-normal distribution of the car prices.
2. Addressing this, we apply the logarithm method to normalize the distribution, creating a new feature, log\_price.
3. The existing price column is replaced with log\_price in the dataset.
4. *Model\_2, utilizing log\_price, yields the following results*:
   1. Scores on the training set:
      1. R2 (model score): 89.6%
      2. MAE: 0.123
      3. MSE: 0.0246
   2. Scores on the training set:
      1. R2 (model score): 90.4%
      2. MAE: 0.145
      3. MSE: 0.034
5. **NOTE**: Given the logarithmic transformation of carprice, MAE and MSE are in log values.

* **Model\_3 – Incorporating Scaled Features and Applying the Logarithm to Carprice**

1. Some features, such as peak-rpm and curb-weight, have values in the thousands, while others like width and height are in two-digit numbers. This disparity in magnitudes may not accurately represent their respective weights in the regression model. To address this, we can scale all features, ensuring an equalized and consistent scale across the board.
2. Carprice, already in logarithmic form, doesn't require scaling.
3. *Model\_3, built on the scaled dataset and log\_price,* *yields the following results*:
   1. Scores on the training set:
      1. R2 (model score): 89.6%
      2. MAE: 0.123
      3. MSE: 0.0246
   2. Scores on the training set:
      1. R2 (model score): 90.4%
      2. MAE: 0.145
      3. MSE: 0.034
4. **NOTE**: Given the logarithmic transformation of carprice, MAE and MSE are in log values.
5. **RESULT SUMMARY**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Training set | | | Test set | | |
|  | **Model score (R2)** | **MAE** | **MSE** | **Model score (R2)** | **MAE** | **MSE** |
| Model\_1 | 89 | 1719.76 | 5319425.17 | 89 | 2508.34 | 12561190.91 |
| Model\_2 | 89.6 | 0.123 | 0.0246 | 90.4 | 0.145 | 0.034 |
| Model\_3 | 89.6 | 0.123 | 0.0246 | 90.4 | 0.145 | 0.034 |

Upon reviewing the aforementioned summary, we discern that model\_1 achieves nearly 90% accuracy in the prediction model without the application of scaling or normalization. Therefore, we can confidently utilize the first model for predictions.