REPORT ON

CAR PRICE PREDICTION USING REGRESSION MODELS

Group Number: 01

|  |  |
| --- | --- |
| Team Members | Roll Number |
| Aluwala Sanjeev Kumar | AM.EN.U4AIE22103 |
| Dadida Mahesh | AM.EN.U4AIE22115 |
| Kasa Amareswar Reddy | AM.EN.U4AIE22125 |
| Padarthi Khushwant | AM.EN.U4AIE22138 |

**Abstract**

The following report considers the prediction of car prices by applying machine learning techniques over three different datasets: used cars, new cars available at factory outlets, and new and used cars produced between the years 2010-2020. Data preprocessing was applied to clean and standardize the different datasets for ideal model performance in this paper. Four kinds of models have been used: Linear Regression, K-Nearest Neighbors, Random Forest, and finally, Support Vector Regression. Accuracy and predictive power were used as the base of evaluation for the models. Our results show that although each model has its own interesting strengths, the Random Forest model turned out to be consistently at the top among the others in nearly all datasets on the most accurate and reliable prediction tasks. These results underline the importance of appropriate models and preprocessing technique selection in car price prediction tasks.

**Introduction**

In the ever-changing automotive industry, accurate car price prediction is very important for manufacturers, dealers, and buyers. The problem is the fact that these prices must consider a great number of factors due to being affected by the car's condition, make, and model as well as the market trends. By making use of machine learning techniques, the project has a goal of creating car price predictive models that would be strong. We employed four different machine learning models, such as Linear Regression, Support Vector Regression (SVR), K-Nearest Neighbors (KNN), and Random Forest, to make a wide range of car price factor analysis. These models underwent three sets of training and testing sessions on the following: the manufacturing data of new cars, the data of used cars, and the data of cars out of production between 2010 and 2016.

An accurate car price calculation is extremely important because of several reasons. For manufacturers, it really helps them identify the direction of their operations, make production forecasts, and set reasonable prices for new models. Dealers, in turn, improve through inventory optimization, pricing strategies, and sales. Buyers, on the other hand, are enlightened about price fairness, which in turn facilitates them in making buying decisions backed by information thus preventing them from being overcharged.

**Significance of Car Price Prediction**

For car manufacturers, the ability of the price to be predicted accurately means the decisions on the launch of a new model, changes in the production volume, and marketing strategies can be taken with certainty. Knowing what the market needs is one of the advantages that it offers since they saw that at that time, production and consumer demand were matched. Adopting predictive models makes it possible for the dealers to maximize their stock and raise recovery prices, by that maintain whole customer satisfaction. If the consumers can make accurate predictions of prices, it would result in them being able to take the best decisions when purchasing or selling cars, which also leads to the sale of fair transactions.

**Machine Learning Models Employed**

Our goal was accomplished by the employ of four machine learning models:

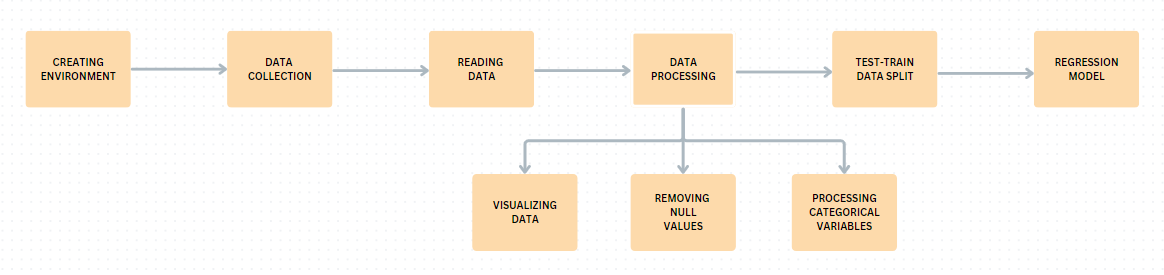
* Linear Regression: This is a basic approach excellent in the statistical modeling field, linear regression is a statistical method to determine the relationship between a dependent variable (car price) and independent variables (car features). This is the most primitive model for our project.
* Support Vector Regression (SVR): Support Vector Machines extension, SVR is an effective model in high-dimensional spaces and models with complex structure. The best-fitting hyperplane is found, which allows for the margin of tolerance (epsilon).
* K-Nearest Neighbors (KNN): This KNN is a non-parametric method that uses the "k" most similar instances in the training data and predicts the prices. KNN is most effective in capturing non-linear relationships among car features and prices.
* Random Forest: The ensemble learner constructs different decision trees during the training phase and then it gives the mean prediction of the individual trees. Random Forest is recognized for being robust which means it can deal very well with large volume data with many dimensions. Thereby it shows less over-fitting.

**Datasets Used**

Our predictive models were trained and tested on three distinct datasets:

* Manufacturing Data of New Cars: This dataset lists everything to do with the production of a new vehicle, including the year, car, and engine specifications, as well as the quality of the product, and the introduction of the product along with the pricing for the cars. The newly manufactured cars dataset is an intermediate for price of new cars.
* Data of Used Cars: This data set covers various attributes of used cars, including age, mileage, condition, service history, and previous ownership details. The used car market is a multipart one, whereby past depreciation together with reduction sustains prices.
* Data of Cars Manufactured Between 2010 and 2016: This dataset is a historical one giving the information about the cars that are of the year neither new nor old. This is a database of middle-range cars now and then that shows what the price influencers are.

**Methodology Overview**

Fig-1: Workflow of Study

* Data Collection and Preprocessing: We gathered and preprocessed the datasets, handling missing values, encoding categorical variables, and scaling numerical features to ensure data quality and readiness for analysis.
* Feature Selection and Engineering: We identified the most relevant features influencing car prices and engineered new features to enhance model performance, ensuring the models captured key aspects of the data.
* Model Training and Evaluation: The datasets were split into training and testing sets. We trained the models using the training data and evaluated their performance on the testing data using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R²) score.
* Hyperparameter Tuning: We optimized the hyperparameters of each model to improve predictive accuracy, fine-tuning them to achieve the best performance.
* Result Interpretation and Insights: We interpreted the results, drew insights from the models' predictions, and discussed their implications for the automotive market, providing valuable information for stakeholders.

**Implementation**

Prior to dataset prediction, all the datasets are pre-processed, which means that all null and missing points are filled, the dataset has the plotted graph of outliers to be removed from and, at the end, a correlation heat map is visualized, which helps in knowing the connection between the features, and non-related features are dropped since their existence when applying regression models does not cause changes to the model's performance. We used three datasets which are collected from the Kaggle.

**Dataset 1 (Manufacturing Data of New Cars):**

**Data Exploration and Visualization:**

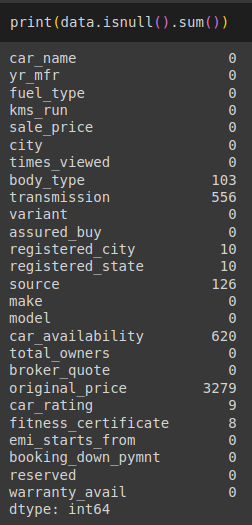
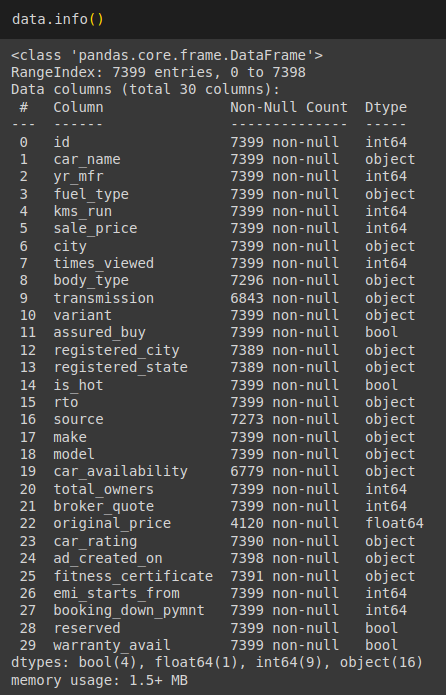
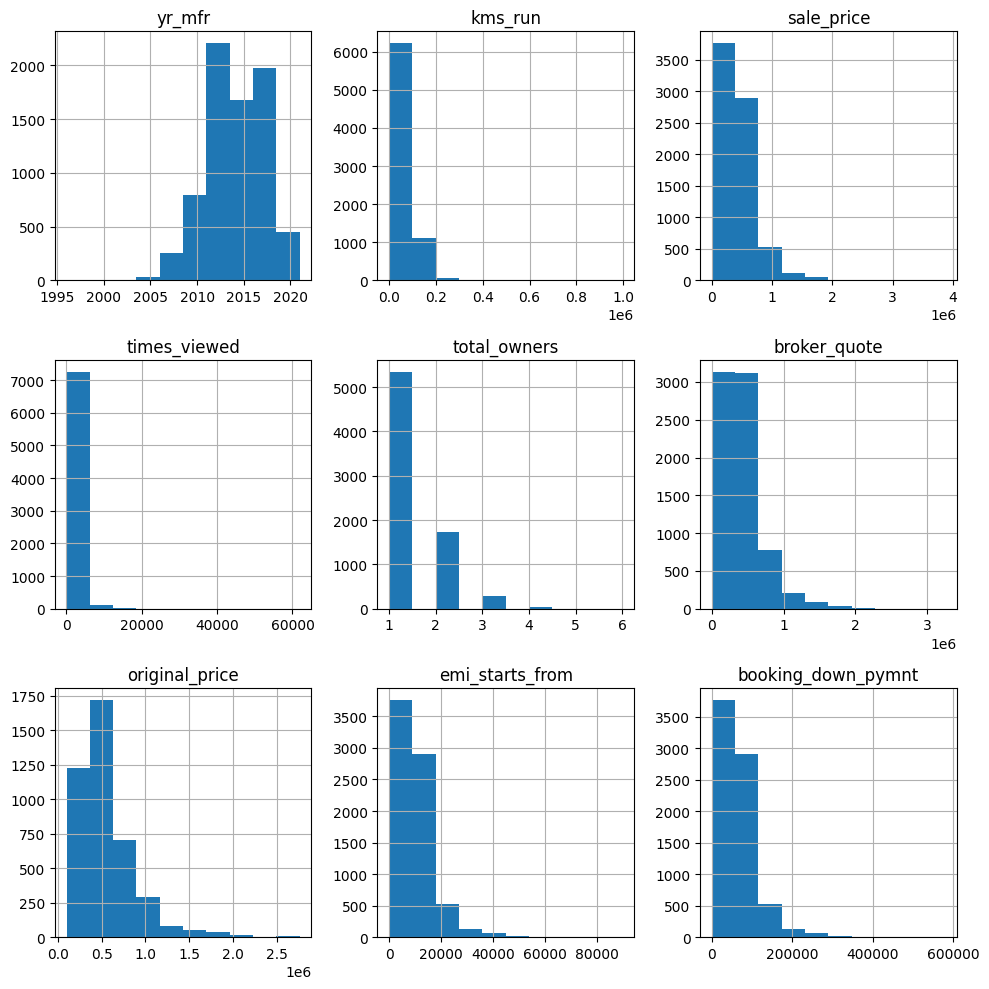


Fig-2: Understanding the Data

* Shape of Dats set: 7399 records with 30 attributes
* Number of Numerical Attributes: 10
* Number of Boolean Attributes: 4
* Number of Categorical Attributes: 16
* Number of Columns with Missing Data: 7

Initially we deleted a few of the features from the data because of their irrelevance for the train of model so, we removed id, is\_hot, rto and ad\_created\_on columns.

**Histogram of the numerical features** Fig-3 Histogram of the numerical features

Most of the data have left skewness which means that common variable measures like kilometers ran sale price, broker quote, and the number of times viewed are all largely affected by one factor which is lower values. From this, it follows on that the selling market has transitioned to the latest version of cars which are either new, which have few miles/those that are cheaper, as the number of those with high mileage, high prices, or high view counts has decreased.

**Data Pre-Processing:**

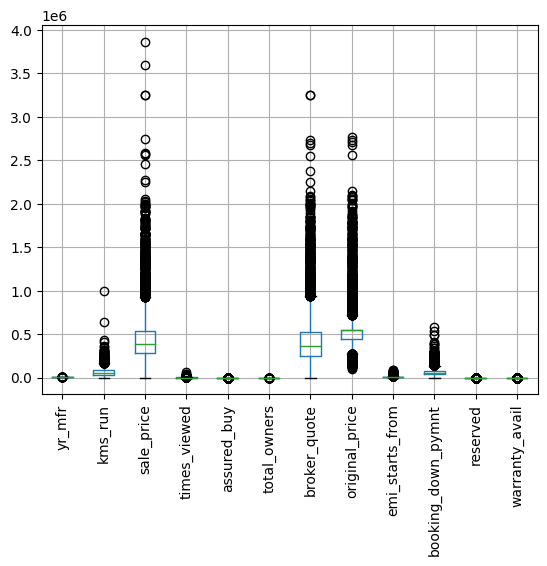
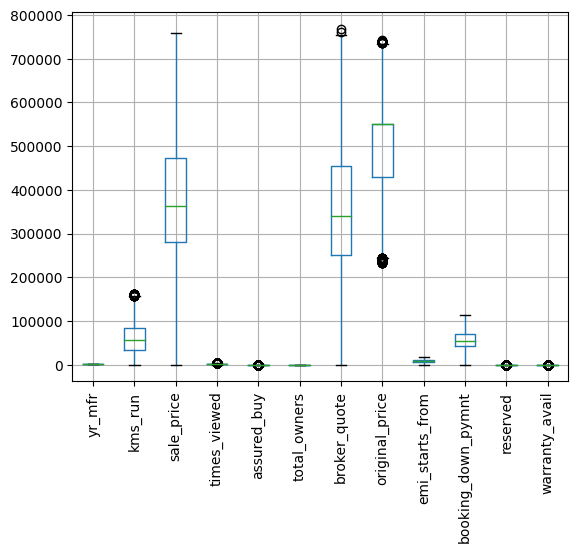
**Handling Missing Data**

Two different imputation methods were selected for use in the first data set. The selection was determined by the type of feature—categorical or numerical. For categorical features, we used the Simple Imputer with the strategy being set to "most frequent." We used this method for 'body\_type', 'transmission', 'registered\_city', 'registered\_state', 'source', 'car\_availability', 'car\_rating', 'fitness\_certificate' features. Using this method, which is essentially a mode of imputation, we replace the missing data with the most repeated value in each categorical column. This approach entails that the imputed values are in accordance with the existing data distribution, and the data's integrity is thus preserved.

On the side of numeric, we resorted to the method where the arithmetic mean of data was computed and used for imputation. This action takes care of the unfilled areas in the specified columns where it uses the average value of the data in these columns. This is used for original price feature. Through the mean approach, we obtain a realistic guess for the missing data, which nevertheless does not change the overall distribution of the quantitative parameters. This unbiased method is crucial in keeping the statistical parameters of the dataset unchanged, thus ensuring the following analysis and model training are based on accessible and credible data.

**Removing Outliers**

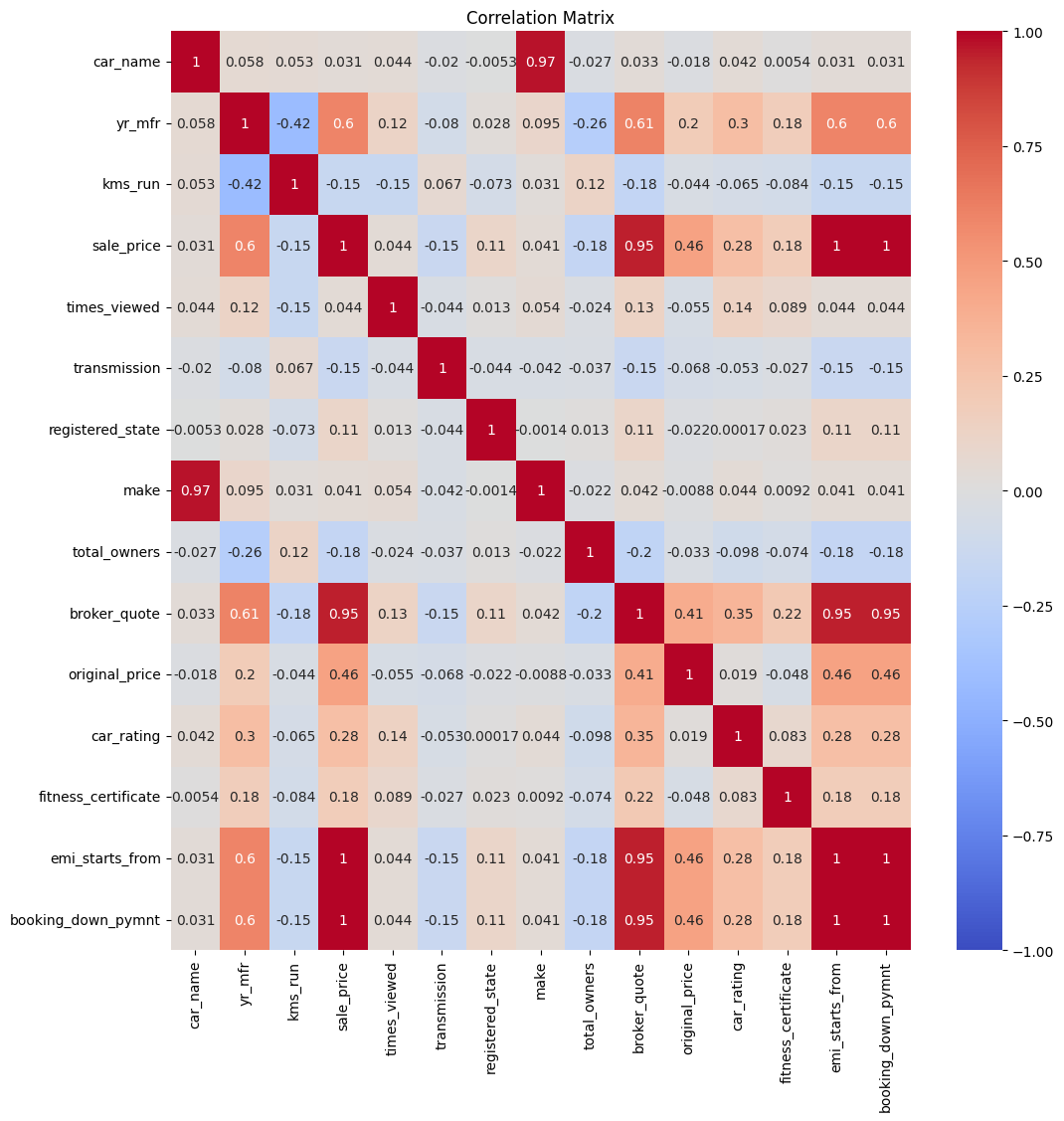
Most outliers in our dataset were found in the sale\_price, broker\_quote, original\_price, and booking\_down\_payment columns. To handle these, we used the Interquartile Range (IQR) method. The IQR is calculated as the range between the first quartile (Q1) and the third quartile (Q3). Outliers are identified as data points that fall below the lower bound (Q1 - 1.5 × IQR) or above the upper bound (Q3 + 1.5 × IQR). By removing data points outside these bounds, we effectively minimized the impact of extreme values, ensuring that our dataset remained reliable and that our predictive models were not skewed by these anomalies.

    
 Before Removing Outliers After Removing Outliers

**Encoding Categorical Data**

Initially we dropped some of the features which had so many of unique values because after removing the outlier's records were keep down to around 6000 in that these two features were having nearly 200 to 600 unique values if we encoded them the data would become unusual. We employed encoding techniques to handle categorical variables based on their unique features. For features with a limited number of categories, such as fuel\_type, body\_type, source, and car\_availability, we used OneHotEncoder to create binary columns, effectively capturing the presence or absence of each category without imposing any ordinal relationship. We used Label Encoder for features like transmission, car\_rating, fitness\_certificate, make, registered\_state, and car\_name because these features have numerous unique categories that are not suitable for one-hot encoding due to the high dimensionality it would create. Label encoding is more memory efficient and helps in maintaining the ordinal relationships where applicable. For example, car\_rating and fitness\_certificate have inherent ordinal relationships that are preserved through label encoding.

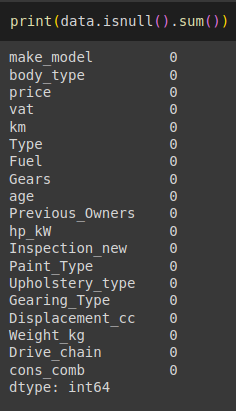
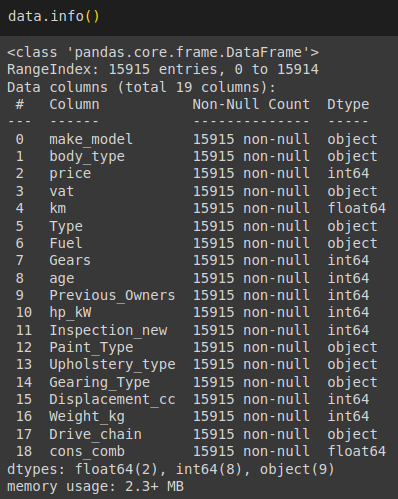
Next, we removed some of the features by examining the features correlation through heat map, so we removed the features which were having very low co-relation like below 0.01.



So, we removed the features times\_viewed, registered\_city, registered\_state, city which were having very less relation with the sale\_price target feature. Now, the preprocessing has completed then we have the data into X(features) and y (target feature) then we have split the data into train and test using the train\_test\_split in 8:2 ratio. Then we standardized the features by removing the mean and scaling to unit variance, this ensures that the features have a mean of 0 and a standard deviation of 1.

**Dataset 2 (Data of Used Cars):**

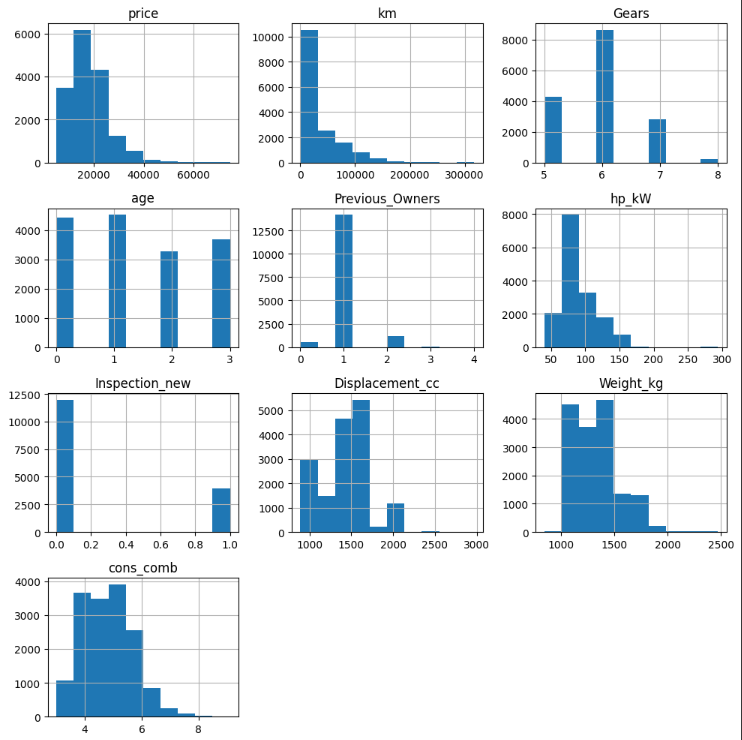
**Data Exploration and Visualization:**

 Figure-No: Understanding the Data

* Shape of Dats set: 15915 records with 19 attributes
* Number of Numerical Attributes: 10
* Number of Boolean Attributes: 0
* Number of Categorical Attributes: 9
* Number of Columns with Missing Data: 0

Prior to the data preprocessing only “id” column is dropped as it has no use and may miss lead the prediction.

**Histogram of the numerical features**



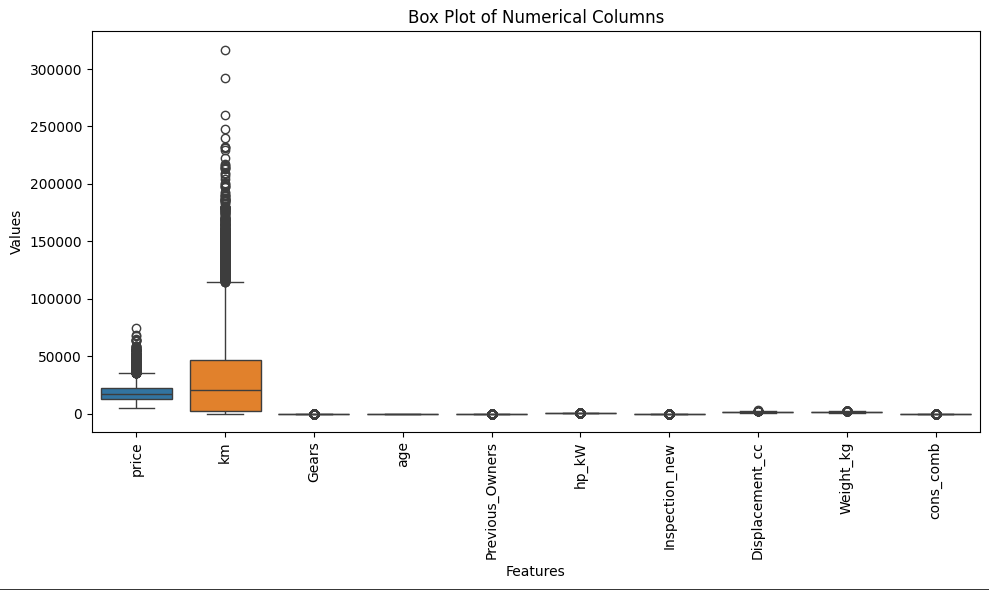
Some attributes of the data have the right skewness which means that common variable measures like price, km, hp\_kW, Previous\_Owners and Displacement\_cc are all largely affected by one factor which is lower values. From this, it follows on that the selling market has transitioned to the latest version of cars which are either new, which have few miles/those that are cheaper, as the number of those with high mileage, high prices, or high view counts has decreased, attributes like Weight\_kg, cons\_comb, Gears are normally distributed which means most of the data is recorded in at the center, age attribute is distributed evenly throughout all the datapoints in that attribute and inspection\_new attribute is concentrated only at ends.

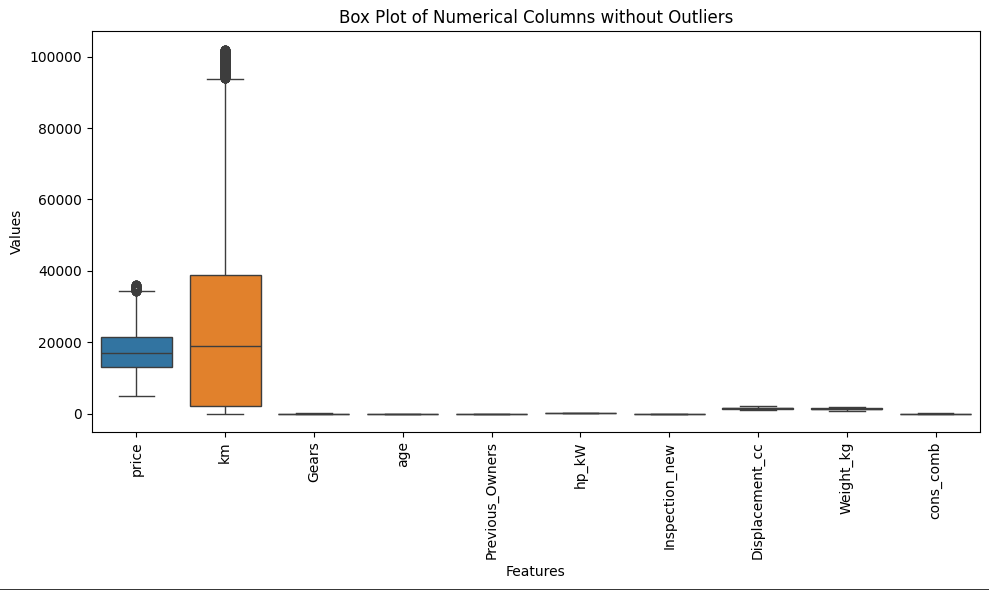
**Handling Missing Data**

As there are no missing values in the 2nd data set, there is no need to use any imputation methods.

**Removing Outliers**

Most outliers in our 2nd dataset were found only in the km column. To handle these, we used the Interquartile Range (IQR) method. The IQR is calculated as the range between the first quartile (Q1) and the third quartile (Q3). Outliers are identified as data points that fall below the lower bound (Q1 - 1.5 × IQR) or above the upper bound (Q3 + 1.5 × IQR). By removing data points outside these bounds, we effectively minimized the impact of extreme values, ensuring that our dataset remained reliable and that our predictive models were not skewed by these anomalies.

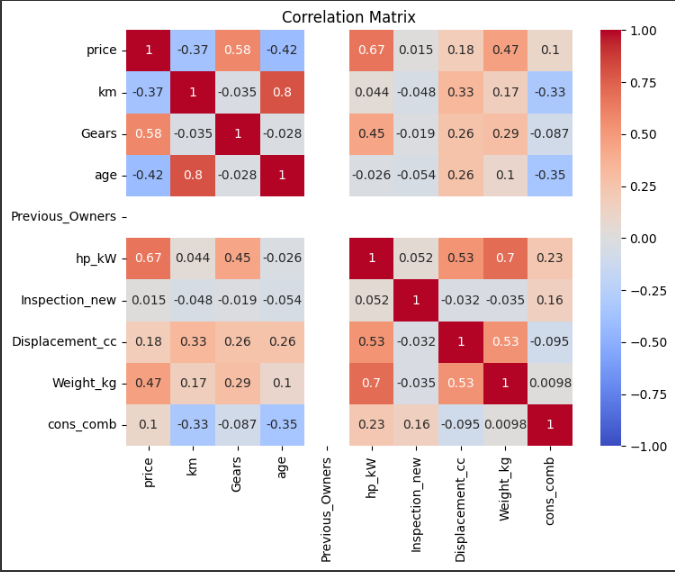




**Encoding Categorical Data**

After removing outliers around 3000 records have been removed from the dataset. And prior to that we have dropped some of the unimportant features from the data, so dropping furthermore features will cause reduced accuracy, model instability and may increase bias. So instead of removing all the categorical features if we Encode them, we can escape from the above problems. We used encoding techniques like Label Encoding, Ordinal Encoding and One hot Encoding to convert them into numerical values. We used ordinal encoding where the ranking of the values in that feature will have some use when predicting the price of the car like the make\_model and body\_type. One hot encoding is applied for Gears feature as it has less unique values and if we had used other it may cause unnecessary complexity or misleading of the prediction. Label encoding is used where there are a greater number of unique values in feature to use One hot encoding and not able rank them to use Ordinal encoding. Label encoding is used in features like vat, type, Fuel, Paint\_type, Upholstry\_type and Drive\_chain.

Next, we tried to remove some of the features by examining the features co-relation through heat map, so we removed the features which were having very low co-relation like below 0.01.

We did not drop any features as they are evenly related to one of the features if not another.

Now, the preprocessing has completed then we have the data into X(features) and y (target feature) then we have split the data into train and test using the train\_test\_split in 8:2 ratio. Then we standardized the features by removing the mean and scaling to unit variance, this ensures that the features have a mean of 0 and a standard deviation of 1.

**Dataset 3 (Data of Cars Manufactured Between 2010 and 2016):**

**Data Exploration and Visualization:**

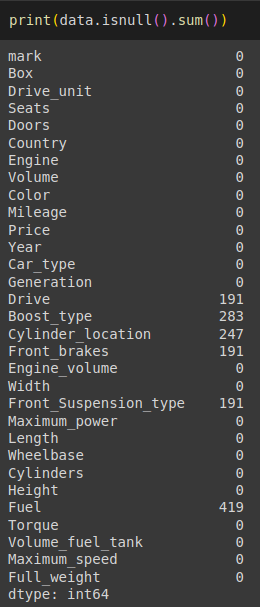
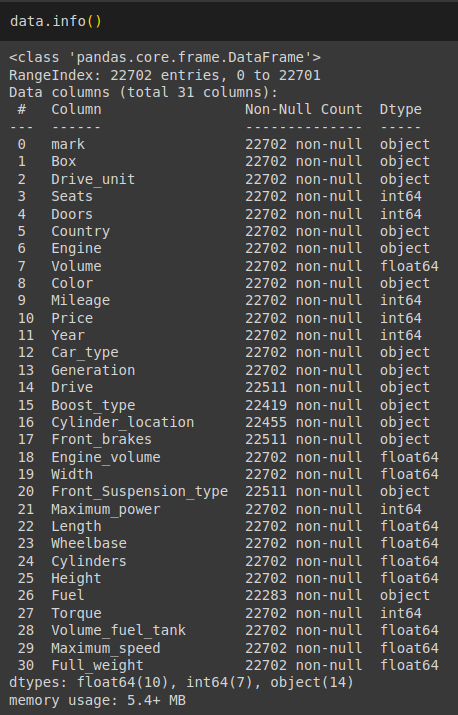
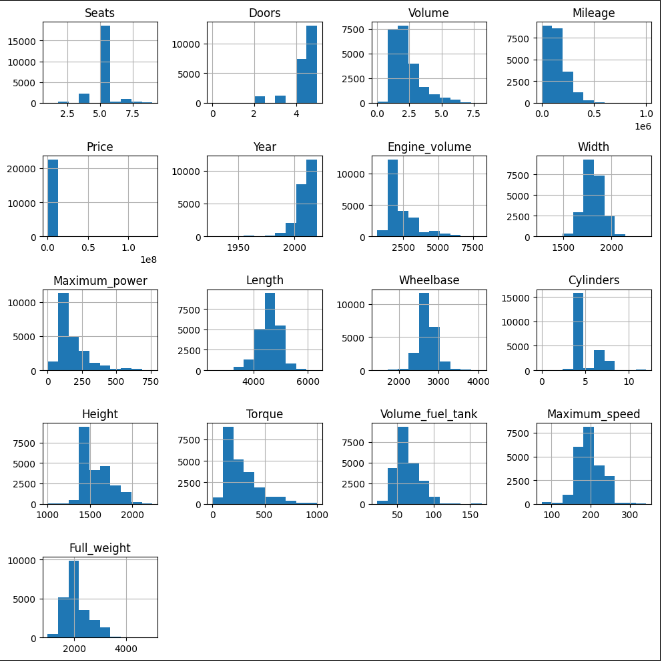


Figure-No: Understanding the Data

* Shape of Dats set: 22702 records with 31 attributes
* Number of Numerical Attributes: 17
* Number of Boolean Attributes: 0
* Number of Categorical Attributes: 14
* Number of Columns with Missing Data: 6

Prior to the data preprocessing only “id” column is dropped as it has no use and may miss lead the prediction.

**Histogram of the numerical features**



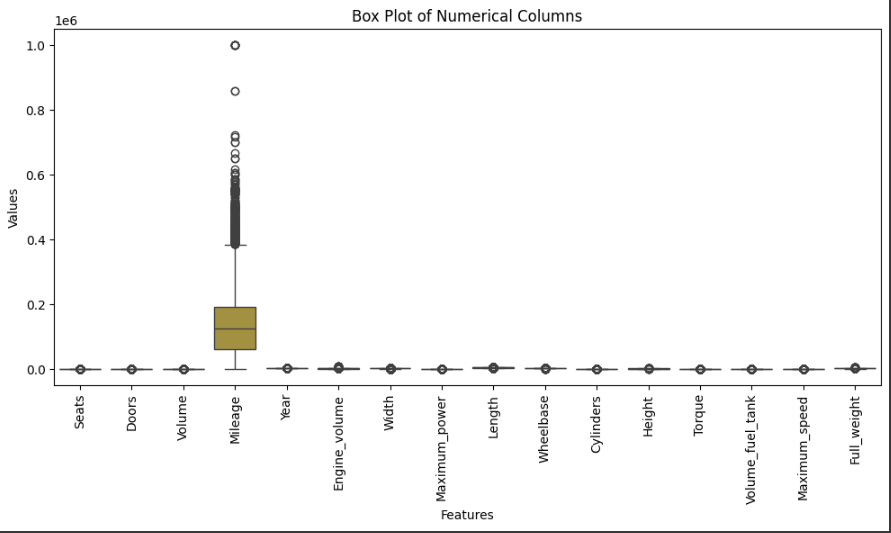
From the above plot we can observe that Length, Max\_speed, Full\_Weight, Wheelbase, Cylinders, Width and Volume\_fuel\_tank is normally distributed. Height, Torque, Max\_power, Engine\_volume and Cylinders are left skewed. And the year of manufacture is only right skewed.

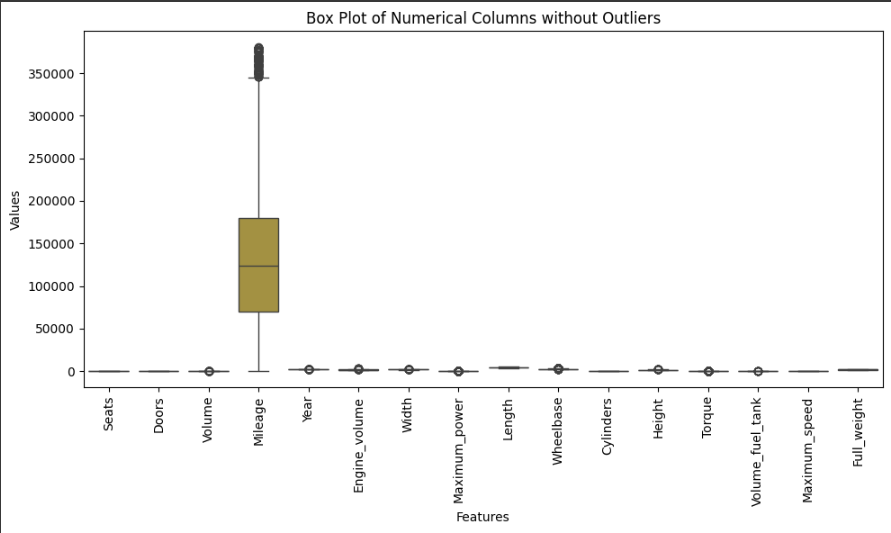
**Handling Missing Data**

As all the missing values in the 3rd data set are from Categorical attributes. The imputation method used here is Mode imputation which replaces the missing values with the most common Categorical values in that attribute. This approach entails that the imputed values are in accordance with the existing data distribution, and the data’s integrity is thus preserved.

**Removing Outliers**

Most outliers in our 2nd dataset were found only in the Mileage column, most far way points from each column are removed. To handle these, we used the Interquartile Range (IQR) method. The IQR is calculated as the range between the first quartile (Q1) and the third quartile (Q3). Outliers are identified as data points that fall below the lower bound (Q1 - 1.5 × IQR) or above the upper bound (Q3 + 1.5 × IQR). By removing data points outside these bounds, we effectively minimized the impact of extreme values, ensuring that our dataset remained reliable and that our predictive models were not skewed by these anomalies.



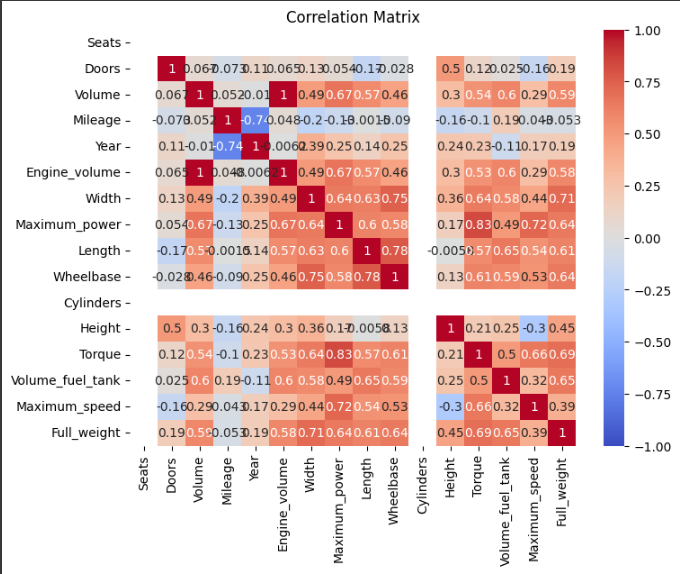


**Encoding Categorical Data**

Instead of removing all the categorical features if we Encode them, we can reduce model instability and bias. We used encoding techniques like Label Encoding, Ordinal Encoding and One hot Encoding to convert them into numerical values. We used ordinal encoding where the ranking of the values in that feature will have some use when predicting the price of the car like the front\_brakes and Car\_type. One hot encoding is applied for Cylinder\_location and Engine feature as it has less unique values and if we had used other it may cause unnecessary complexity or misleading of the prediction. Label encoding is used where there are a greater number of unique values in feature to use One hot encoding and not able rank them to use Ordinal encoding. Label encoding is used in features like Drive, Boost\_type, Country, Fuel, Drive\_unit, Front\_Suspension\_type, mark, Box and Color.

We had dropped feature Generation as it has too many unique values which may also cause bias and instability of Regression Models.

Next, we tried to remove some of the features by examining the features co-relation through heat map, so we removed the features which were having very low co-relation like below 0.01.

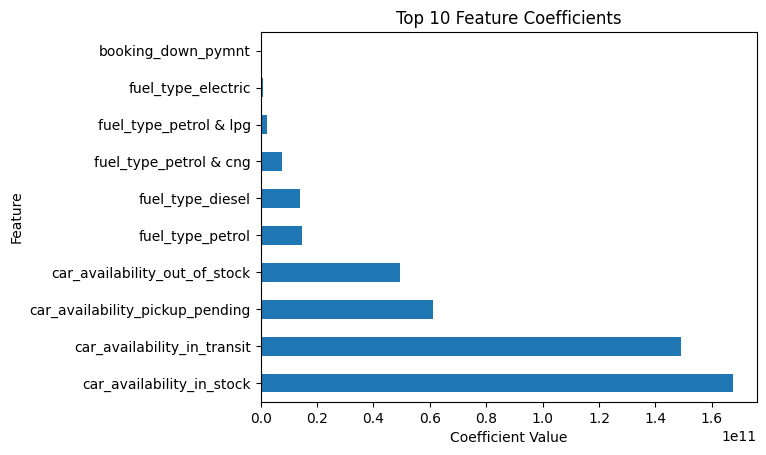


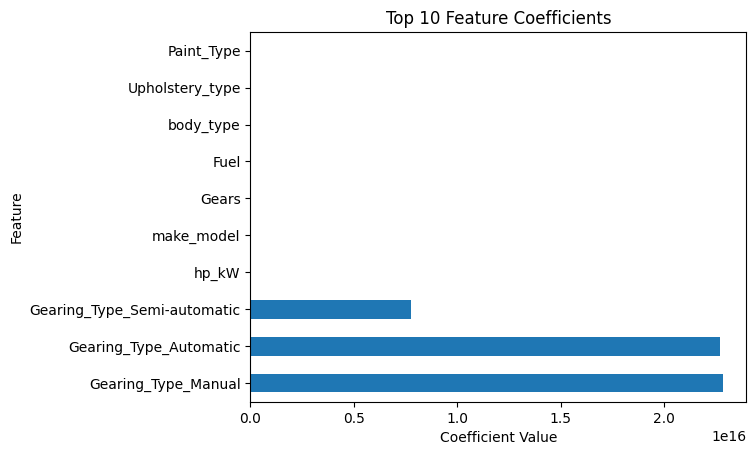
Here even though Mileage is not co-related with nay of the features we are not dropping it as if we consider generally based on the mileage a car gives also affects the price of the car.

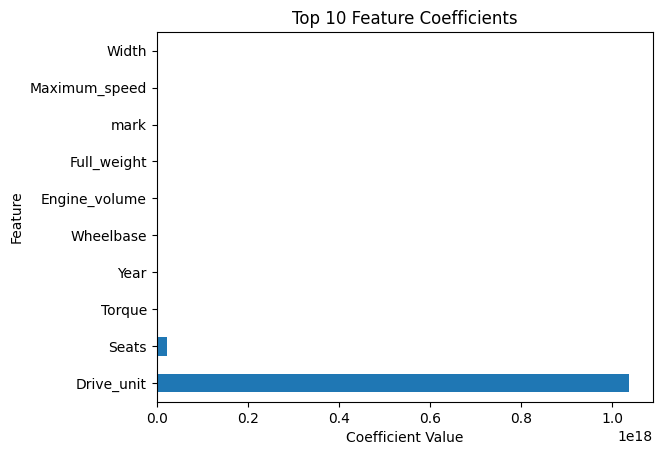
As the preprocessing was completed then we divided the data into X(features) and y (target feature) then we split the data into train and test using the train\_test\_split in 8:2 ratio. Then we standardized the features by removing the mean and scaling to unit variance, this ensures that the features have a mean of 0 and a standard deviation of 1.

**Results after Model Training and Testing**

**1. Linear Regression:** We trained and tested this model on three datasets the results were:

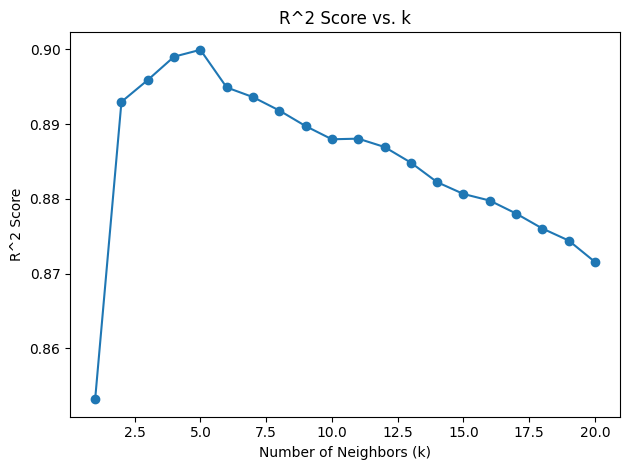
Dataset 1: R2 score was 0.99, the top 10 features with the largest coefficients were;

Dataset 2: R2 score was 0.77, the top 10 features with the largest coefficients were;

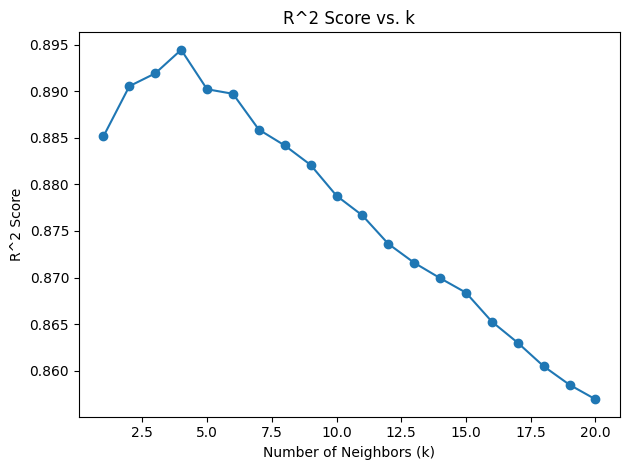
Dataset 3: R2 score was 0.73, the top 10 features with the largest coefficients were;

2. **K Nearest Neighbors (KNN)**: For KNN initially we chose 5 nearest Neighbours then replaced it with the ‘k’ value which had more R^2 score by visualizing the plot of k vs R2 score.

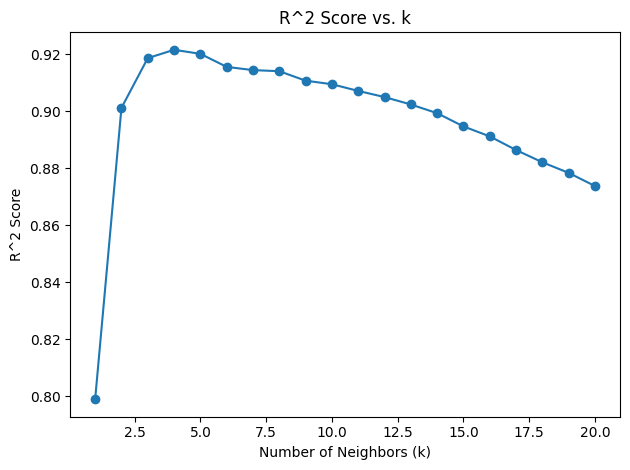
Dataset 1: R2 score was 0.89 for this dataset 5 was the optimal k value



Dataset 2: R2 score was 0.88 for this data 4 was optimal k value



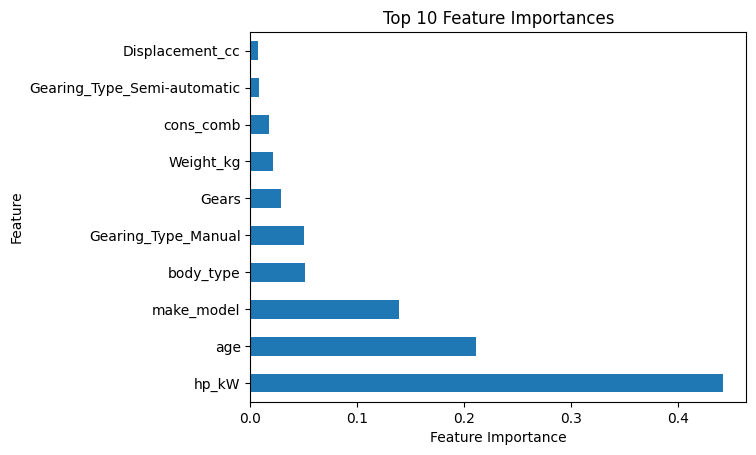
Dataset 3: R2 score was 0.91 for this data 4 was optimal k value



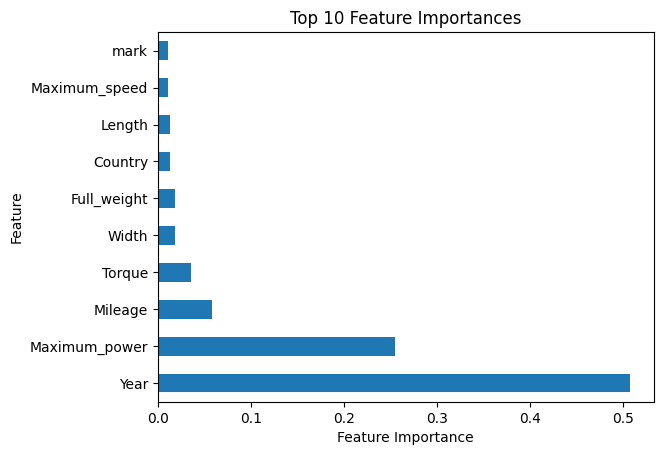
3. **Random Forest:** This model outperformed the three dataset this recorded the height R2 value among the four models, for all the datasets n\_estimators were set to 100.

Dataset 1: R2 score was 0.99, booing\_down\_payment and emi\_starts\_from where the two features which showed importance more.

Dataset 2: R2 score was 0.93, the features which showed importance were;



Dataset 3: R2 score was 0.95, the features which showed importance were;



4. Support Vector Regression (SVR): For the default conditions of this model the prediction was very poor to outcome this we performed hyper parameter tuning by changing the C, epsilon and kernels. So, this were output:

Dataset 1: Initial R2 score was 0.17, then after parameter tuning for the C value of 10.0, epsilon value 0.5 and kernel with linear the R2 score was improved to 0.936547.

Dataset 2: For this dataset we didn’t use any hyper parameter tuning because it has a descent R2 score of 0.72 so we didn’t perform tuning to avoid overfitting.

Dataset 3: Like dataset 1 model was unable to perform well on this dataset so we performed hyper parameter tuning and attained R2 score of 0.609139 for C value of 100, epsilon value of 0.5 and linear kernel.