PROJECT 3 REPORT

(Arun Jiju Joseph Id:1002140653)

Introduction:

We are using the same data as for the previous homework. Using the df2 table created from last homework where it has already cleaned, dropped some of the variables and also created the dummy variables.

Part 1: Data from our lives:

Described a situation or problem for which a variable selection/feature reduction would be appropriate. The example which I used was based on Predicting Product Defects in Manufacturing, the production process involves various factors such as temperature, pressure, machine settings, and raw material quality. The manufacturer can implement a machine learning model to predict whether a product will have defects based on these production-related features. Methods like LASSO (Least Absolute Shrinkage and Selection Operator), or correlation analysis can help identify and prioritize the most relevant features for defect prediction. This process aids in building a more focused and efficient model, improving the accuracy of defect predictions, and potentially providing insights into the root causes of defects in the manufacturing process.

Data manipulation/Exploratory Data Analysis:

The main objective of this step is to clean the data in order to get it ready for analysis. The dataset we're using for Part two is taken from the 1985 Auto Imports Dataset. This data must be processed because it is raw before the analysis can be done. The null values, duplicate entries, etc. in the data may affect the final results. We clean and organise the data in order to make it understandable so that we can draw any inferences from it. Data cleaning is the process of removing unnecessary or undesirable data from a dataset. Data wrangling is the process of getting this clean data into a format that can be read and used for analysis.

We are using the Auto Imports Dataset, which is a CSV file. We'll outline the procedure we'll use to clean this in the manner that follows. The first step is to read the csv file and import all the relevant libraries, such as NumPy, pandas, and sklearn. We will examine the variable data types. The method is demonstrated in the following:

```
data_types = df.dtypes
print(data_types)
```

```
fuel_type        object
body             object
wheel_base      float64
length          float64
width           float64
heights         float64
curb_weight       int64
engine_type      object
cylinders        object
engine_size       int64
bore             object
stroke           object
comprassion     float64
horse_power      object
peak_rpm         object
city_mpg          int64
highway_mpg       int64
price             int64
dtype: object
```

The three main data types are float64, int64, and object. There are 18 different types of this variable data. Next task is to Replace '?' with None and to Change the variables: bore, stroke, horse_power, peak_rpm to float64.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 201 entries, 0 to 200
Data columns (total 18 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   fuel_type     201 non-null    object
 1   body          201 non-null    object
 2   wheel_base    201 non-null    float64
 3   length        201 non-null    float64
 4   width         201 non-null    float64
 5   heights       201 non-null    float64
 6   curb_weight   201 non-null    int64
 7   engine_type   201 non-null    object
 8   cylinders     201 non-null    object
 9   engine_size   201 non-null    int64
 10  bore          197 non-null    float64
 11  stroke        197 non-null    float64
 12  compression   201 non-null    float64
 13  horse_power   199 non-null    float64
 14  peak_rpm      199 non-null    float64
 15  city_mpg      201 non-null    int64
 16  highway_mpg   201 non-null    int64
 17  price         201 non-null    int64
dtypes: float64(9), int64(5), object(4)
memory usage: 28.4+ KB
```

The datatypes of the variables bore, stroke, horse_power, peak_rpm is changed to float64. The columns that have just null values must also be removed, which is done as shown below.

```
df2.isnull().sum()

fuel_type       0
wheel_base      0
length          0
width           0
heights         0
curb_weight     0
engine_size     0
bore            0
stroke          0
compression     0
horse_power     0
peak_rpm        0
city_mpg        0
highway_mpg     0
price           0
dtype: int64
```

The dummy variables for fuel_type within is also removed using get dummies method which is shown below. After executing the code, the dummy variables are removed which the total variables number of variables in the code will be 15.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 195 entries, 0 to 200
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   wheel_base      195 non-null    float64
 1   length          195 non-null    float64
 2   width           195 non-null    float64
 3   heights         195 non-null    float64
 4   curb_weight     195 non-null    int64
 5   engine_size     195 non-null    int64
 6   bore            195 non-null    float64
 7   stroke          195 non-null    float64
 8   comprassion     195 non-null    float64
 9   horse_power     195 non-null    float64
 10  peak_rpm        195 non-null    float64
 11  city_mpg        195 non-null    int64
 12  highway_mpg     195 non-null    int64
 13  price           195 non-null    int64
 14  fuel_type_gas   195 non-null    uint8
dtypes: float64(9), int64(5), uint8(1)
memory usage: 23.0 KB
```

Exploratory Data Analysis:

Step 1: Descriptions and features

The data set that we are using has in total 15 variables with float64, int64 and object as their data type. To obtain the mean, maximum, minimum, and other statistical figures for each column, we use the describe () function. We describe the data in order to highlight central tendencies and the distributional structure of the dataset.

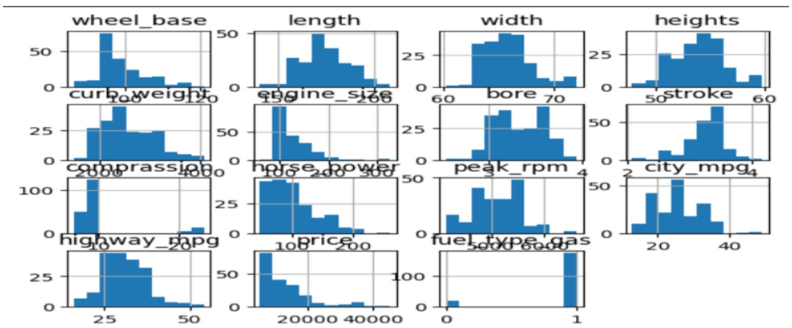
	wheel_base	length	width	heights	curb_weight	engine_size	bore	stroke	comprassion	horse_power	peak_rpm	city_mpg	highway_mpg	
count	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.(
mean	98.896410	174.256923	65.886154	53.861538	2559.000000	127.938462	3.329385	3.250308	10.194974	103.271795	5099.487179	25.374359	30.841026	13248.(
std	6.132038	12.476443	2.132484	2.396778	524.715799	41.433916	0.271866	0.314115	4.062109	37.869730	468.271381	6.401382	6.829315	8056.:
min	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	2.540000	2.070000	7.000000	48.000000	4150.000000	13.000000	16.000000	5118.(
25%	94.500000	166.300000	64.050000	52.000000	2145.000000	98.000000	3.150000	3.110000	8.500000	70.000000	4800.000000	19.500000	25.000000	7756.5
50%	97.000000	173.200000	65.400000	54.100000	2414.000000	120.000000	3.310000	3.290000	9.000000	95.000000	5100.000000	25.000000	30.000000	10245.(
75%	102.400000	184.050000	66.900000	55.650000	2943.500000	145.500000	3.590000	3.410000	9.400000	116.000000	5500.000000	30.000000	35.000000	16509.(
max	120.900000	208.100000	72.000000	59.800000	4066.000000	326.000000	3.940000	4.170000	23.000000	262.000000	6600.000000	49.000000	54.000000	45400.(

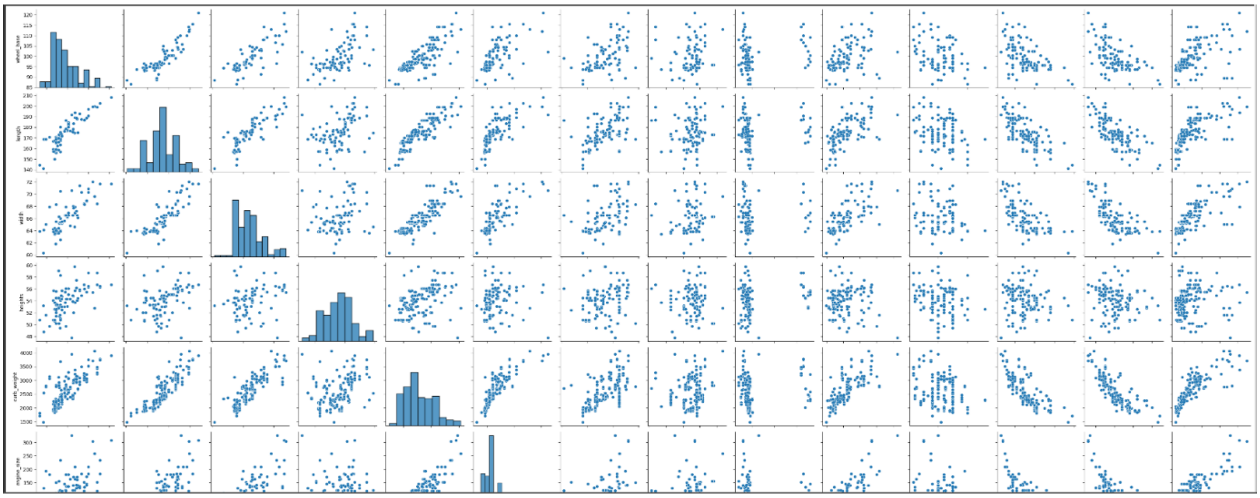
Step 2: Checking Missing value

Looking for null values, duplicate values, and the number of unique values in the columns. The count is used to determine whether the data is balanced.

Step 3: Checking the shape of the data

To determine the shape of the data, we produced historgram and pairplot.



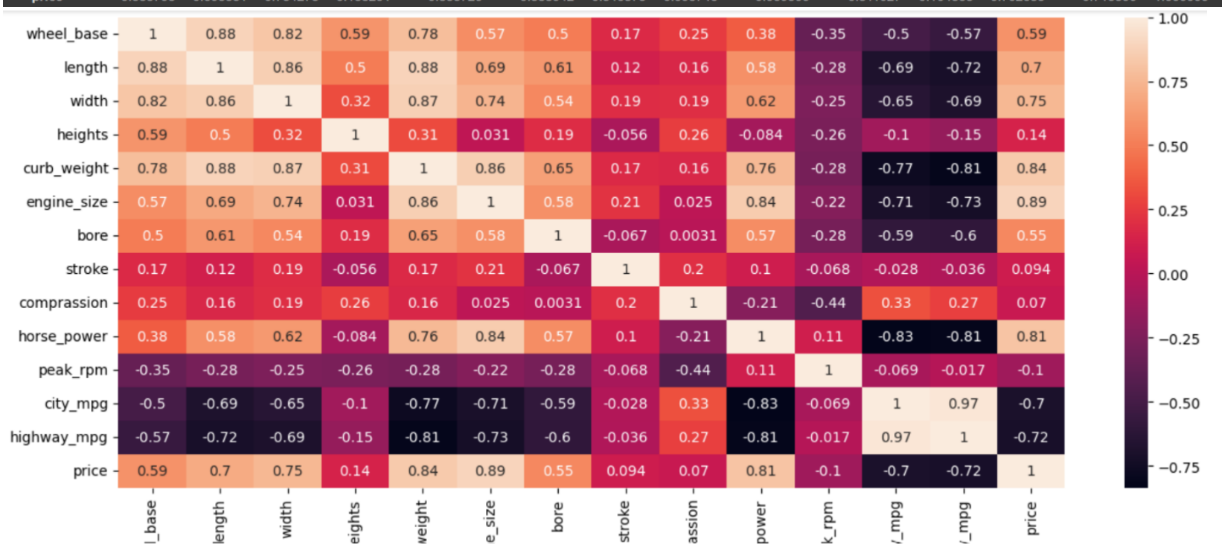


Step 4: Identifying significant correlations

The heatmap, a coloured matrix, displays the correlation between the variables in the data set. All correlations are demonstrated to be positive, and the grid shows how each connection is connected to the others. The variables in the dataset are either positively or directly connected.

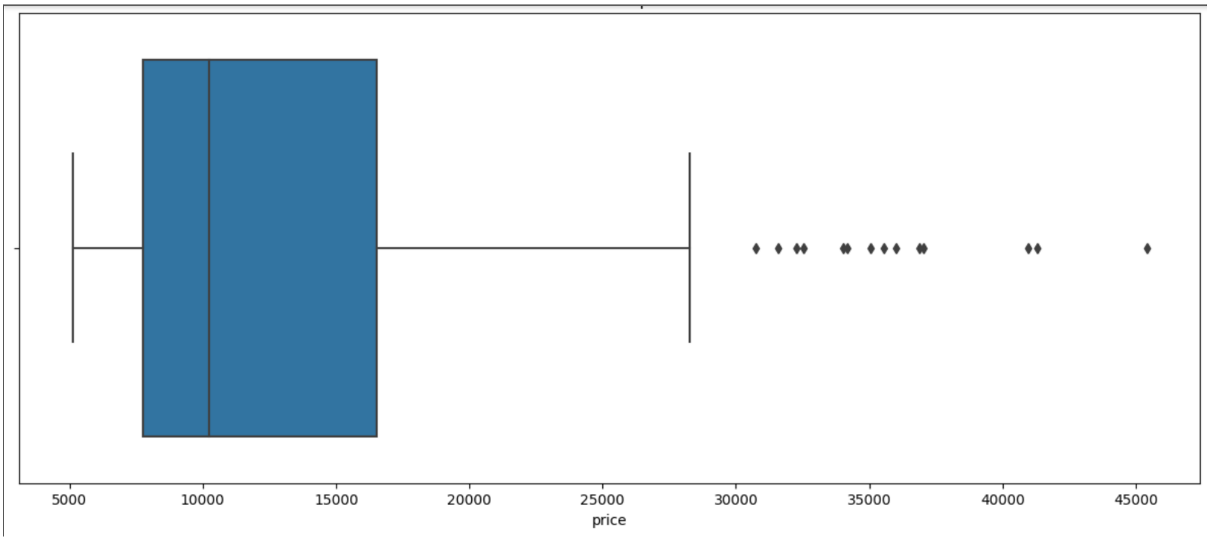
Correlation is used to look at how the variables are related. Based on a scale of -1 to 1, correlation is defined as a negative or indirect connection, +1 as a positive or direct association, and 0 as no correlation. We used Pearson correlation to determine how much the variables were linearly correlated. We created a heatmap to provide as proof of this.

	wheel_base	length	width	heights	curb_weight	engine_size	bore	stroke	compression	horse_power	peak_rpm	city_mpg	highway_mpg	price
wheel_base	1.000000	0.879222	0.819009	0.592500	0.782720	0.569704	0.498228	0.171722	0.247730	0.375541	-0.352331	-0.499126	-0.566355	0.585793
length	0.879222	1.000000	0.858084	0.496218	0.881665	0.687479	0.609437	0.118664	0.160172	0.583813	-0.280986	-0.689660	-0.719324	0.695331
width	0.819009	0.858084	1.000000	0.315834	0.867315	0.740320	0.544311	0.186432	0.190997	0.616779	-0.251627	-0.647099	-0.692220	0.754273
heights	0.592500	0.496218	0.315834	1.000000	0.307732	0.031286	0.189283	-0.055525	0.261160	-0.084412	-0.264078	-0.102367	-0.151188	0.138291
curb_weight	0.782720	0.881665	0.867315	0.307732	1.000000	0.857573	0.645806	0.172785	0.155382	0.760285	-0.278944	-0.772171	-0.812710	0.835729
engine_size	0.569704	0.687479	0.740320	0.031286	0.857573	1.000000	0.583091	0.211989	0.024617	0.842691	-0.219008	-0.710624	-0.732138	0.888942
bore	0.498228	0.609437	0.544311	0.189283	0.645806	0.583091	1.000000	-0.066793	0.003057	0.568527	-0.277662	-0.591950	-0.600040	0.546873
stroke	0.171722	0.118664	0.186432	-0.055525	0.172785	0.211989	-0.066793	1.000000	0.199882	0.100040	-0.068300	-0.027641	-0.036453	0.093746
compression	0.247730	0.160172	0.190997	0.261160	0.155382	0.024617	0.003057	0.199882	1.000000	-0.214401	-0.444582	0.331413	0.267941	0.069500
horse_power	0.375541	0.583813	0.616779	-0.084412	0.760285	0.842691	0.568527	0.100040	-0.214401	1.000000	0.105654	-0.834117	-0.812917	0.811027
peak_rpm	-0.352331	-0.280986	-0.251627	-0.264078	-0.278944	-0.219008	-0.277662	-0.068300	-0.444582	0.105654	1.000000	-0.069493	-0.016950	-0.104333
city_mpg	-0.499126	-0.689660	-0.647099	-0.102367	-0.772171	-0.710624	-0.591950	-0.027641	0.331413	-0.834117	-0.069493	1.000000	0.972350	-0.702685
ighway_mpg	-0.566355	-0.719324	-0.692220	-0.151188	-0.812710	-0.732138	-0.600040	-0.036453	0.267941	-0.812917	-0.016950	0.972350	1.000000	-0.715590
price	0.585793	0.695331	0.754273	0.138291	0.835729	0.888942	0.546873	0.093746	0.069500	0.811027	-0.104333	-0.702685	-0.715590	1.000000

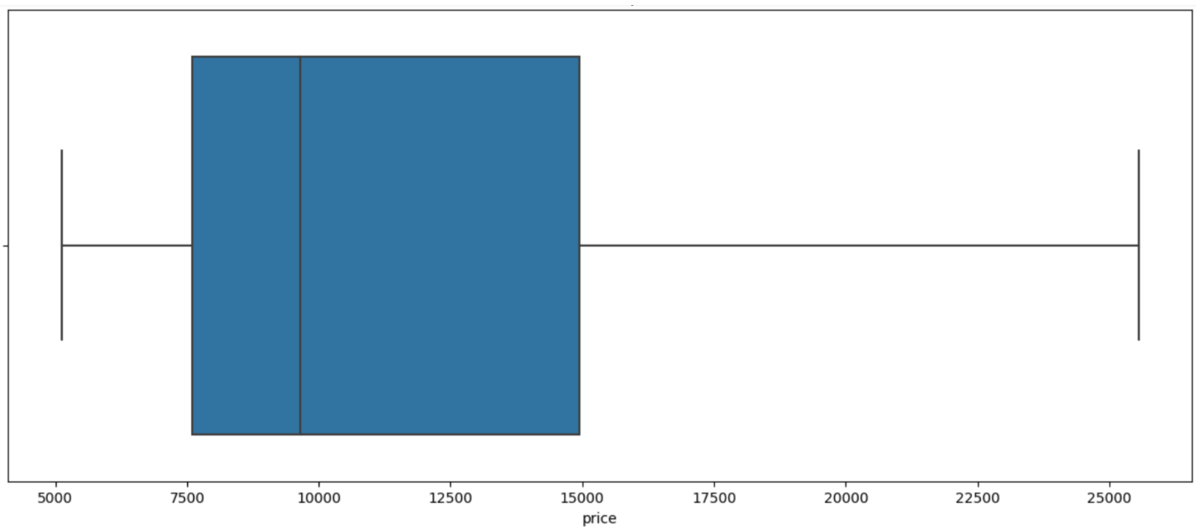


Step 5: Detecting and Handling outliers

Since outliers are outside the box plots, they are easy to identify. Because outliers might make it harder to understand the data, we need to be aware of them. We ultimately plotted a box plot by removing the outliers. We are removing outliers for price variable by plotting a boxplot and the removing from it.



After identifying which rows the outliers are present we will removing it from the data, which we will get as:



Part 2: Variable Selection

2.1. Filtered methods

In filtered methods we will be using ANOVA F-Value for variable selection:

The following output shows Anova values for each feature in the data. From the output, it seems that some variables have low p-values (e.g., 'width', 'curb_weight', 'horse_power', 'city_mpg'), suggesting that these variables are likely to be statistically significant predictors of the dependent variable ('price'). On the other hand, variables with higher p-values may not be statistically significant based on conventional significance levels (e.g., 'engine_size', 'peak_rpm', 'fuel_type_gas').

	sum_sq	df	F	PR(>F)
wheel_base	1.985771e+07	1.0	4.102283	0.044447
length	1.250808e+07	1.0	2.583967	0.109874
width	3.917482e+07	1.0	8.092888	0.005011
heights	7.662985e+04	1.0	0.015830	0.900029
curb_weight	2.993397e+07	1.0	6.183876	0.013893
engine_size	2.197337e+06	1.0	0.453935	0.501421
bore	1.175168e+07	1.0	2.427709	0.121135
stroke	2.578609e+07	1.0	5.326991	0.022247
comprassion	2.272218e+07	1.0	4.694036	0.031711
horse_power	6.326451e+07	1.0	13.069431	0.000399
peak_rpm	6.302263e+04	1.0	0.013019	0.909296
city_mpg	3.411224e+07	1.0	7.047039	0.008722
highway_mpg	1.611827e+07	1.0	3.329775	0.069856
fuel_type_gas	1.234780e+07	1.0	2.550857	0.112159
Residual	7.938662e+08	164.0	NaN	NaN

Individual Feature Significance:

'wheel_base': The F-statistic is 4.10 with a p-value of 0.0444, suggesting that 'wheel_base' is statistically significant in predicting 'price.'

'length': The F-statistic is 2.58 with a p-value of 0.1099, indicating some evidence against the null hypothesis but with less significance.

'width': The F-statistic is 8.09 with a low p-value of 0.0050, indicating strong evidence that 'width' is significant.

'heights': The F-statistic is 0.02 with a high p-value of 0.9000, suggesting that 'heights' is not statistically significant.

'curb_weight': The F-statistic is 6.18 with a p-value of 0.0139, indicating significance.

'engine_size': The F-statistic is 0.45 with a p-value of 0.5014, suggesting that 'engine_size' is not statistically significant.

'bore', 'stroke', 'comprassion', 'peak_rpm', 'fuel_type_gas': These features have varying levels of significance, with p-values above the common threshold of 0.05.

'horse_power': Highly significant with an F-statistic of 13.07 and a very low p-value of 0.0004.

'city_mpg' and 'highway_mpg': Both features are statistically significant, with F-statistics of 7.05 and 3.33, respectively.

The presence of a p-value in 'Residual' suggests that the model, with all features, is statistically significant.

2.2. Wrapper methods

In wrapper methods we will be using Forward selection for variable selection:

The forward selection process likely added these features one at a time, selecting the feature that provided the most significant improvement in model performance at each step. The forward selection process emphasizes the features that contribute the most to improving model performance sequentially.

```
Selected Features (Forward Selection): ['curb_weight', 'horse_power', 'width', 'comprassion', 'stroke']
```

The forward selection process has identified a set of features ('curb_weight,' 'horse_power,' 'width,' 'comprassion,' 'stroke') considered most relevant for predicting car prices. These features collectively capture information related to the car's weight, engine power, dimensions, and engine specifications, all of which are often important factors in determining car prices. The final set of features is considered the most informative for predicting the 'price' variable based on the forward selection criterion.

2.3. Embedded methods

In Embedded methods we will be using Ridge Regression for variable selection:

The Ridge regression model aims to predict the 'price' variable based on a set of independent features while incorporating regularization to prevent overfitting. Here are the coefficients obtained from the Ridge regression model.

```
Ridge Coefficients: [  737.85108926  -607.61442275   971.11393927    -2.32549687
  1295.50926517   957.05259595  -430.763358    -564.87382752
  1940.74118539  1350.98538432   115.37260705 -1609.98864667
   888.0238888   1135.05874975]
```

'const' (Intercept): 737.85

This represents the intercept term in the Ridge regression equation. It is the predicted value of 'price' when all independent variables are zero.

'wheel_base': -607.61

The coefficient indicates the change in the predicted 'price' for a one-unit increase in the 'wheel_base' variable, holding other variables constant.

'length': 971.11

Similarly, a one-unit increase in 'length' is associated with a change of 971.11 units in the predicted 'price,' all else being equal.

'width': -2.33

The coefficient for 'width' suggests a slight decrease in the predicted 'price' for a one-unit increase in 'width.'

'heights': 1295.51

'heights' has a positive coefficient, indicating that an increase in 'heights' is associated with a higher predicted 'price.'

'curb_weight': 957.05

A one-unit increase in 'curb_weight' is associated with a change of 957.05 units in the predicted 'price.'

'engine_size': -430.76

'engine_size' has a negative coefficient, suggesting a decrease in predicted 'price' for a one-unit increase in 'engine_size.'

'bore': -564.87

'bore' has a negative coefficient, indicating a decrease in predicted 'price' for a one-unit increase in 'bore.'

'stroke': 1940.74

'stroke' has a positive coefficient, suggesting an increase in predicted 'price' for a one-unit increase in 'stroke.'

'comprassion': 1350.99

A one-unit increase in 'comprassion' is associated with a change of 1350.99 units in the predicted 'price.'

'horse_power': 115.37

'horse_power' has a positive coefficient, indicating an increase in predicted 'price' for a one-unit increase in 'horse_power.'

'peak_rpm': -1609.99

'peak_rpm' has a negative coefficient, suggesting a decrease in predicted 'price' for a one-unit increase in 'peak_rpm.'

'city_mpg': 888.02

'city_mpg' has a positive coefficient, indicating an increase in predicted 'price' for a one-unit increase in 'city_mpg.'

'highway_mpg': 1135.06

A one-unit increase in 'highway_mpg' is associated with a change of 1135.06 units in the predicted 'price.'

'fuel_type_gas': 0

The coefficient for 'fuel_type_gas' is not included as it represents a binary variable (0 or 1).

Positive coefficients suggest a positive relationship with the predicted 'price,' meaning an increase in the corresponding feature leads to a higher predicted 'price and negative coefficients suggest a negative relationship with the predicted 'price,' meaning an increase in the corresponding feature leads to a lower predicted 'price.'

2.4. Comparing the results of the three methods and also comparing the coefficients to the full linear regression model (model1)

Firstly, a linear regression model has been created using price as the dependant variable and rest of all as independent variable. Then, using the linear regression model created, the coefficients of each variable was found and compared with other methods of the feature selection.

```
ANOVA F-test Results:
        Feature  ANOVA F-statistic       P-value
0     wheel_base         137.146883  8.009792e-24
1         length         219.101058  8.785874e-33
2          width         272.026908  1.270969e-37
3        heights          12.170575  6.125740e-04
4    curb_weight         452.496631  1.210047e-50
5    engine_size         267.988238  2.835681e-37
6           bore          60.734284  5.350204e-13
7         stroke           0.274948  6.006879e-01
8    comprassion           1.783816  1.833963e-01
9    horse_power         215.489248  1.983569e-32
10      peak_rpm           2.037233  1.552504e-01
11      city_mpg         173.588568  4.555872e-28
12   highway_mpg         173.025856  5.256051e-28
13  fuel_type_gas          3.915703  4.938808e-02

Selected Features (Forward Selection):
['curb_weight', 'horse_power', 'width', 'comprassion', 'stroke']

Ridge Regression Model Coefficients:
[  740.87407118  -550.68917354  1000.4031733    -12.09406411
  1348.24800565   564.49602999  -393.50558455  -560.58304663
  1802.33501135  1497.73330715   116.37424168 -1688.15626194
  1074.43359932  1007.93916944]
```

```
Full Linear Regression Model (model1) Coefficients:
const          -46390.036904
wheel_base        159.817469
length            -68.355285
width             555.981338
heights           -13.525249
curb_weight         3.520929
engine_size        13.157204
bore            -1391.978430
stroke          -1631.725723
comprassion       772.398307
horse_power        52.771604
peak_rpm            0.061638
city_mpg         -353.584783
highway_mpg       217.032995
fuel_type_gas    7528.264612
dtype: float64
```

ANOVA F-test Results:

- ANOVA F-test evaluates the overall significance of the linear regression model and the individual features.
- Features such as 'curb_weight,' 'engine_size,' 'length,' and 'width' show high F-statistics and very low p-values, indicating their significance in predicting the 'price' variable.
- 'heights' and 'stroke' have higher p-values, suggesting less significance.

Forward Selection Results:

- The forward selection process identified the following features as the most significant: 'curb_weight,' 'horse_power,' 'width,' 'comprassion,' and 'stroke.'
- These features collectively capture information related to the car's weight, engine power, dimensions, and engine specifications.

Ridge Regression Model Coefficients:

- Ridge regression introduces regularization to prevent overfitting. The coefficients represent the impact of each feature on the predicted 'price' while considering regularization.

- 'curb_weight,' 'horse_power,' 'width,' and 'compression' have noticeable positive coefficients, suggesting positive relationships with the predicted 'price.'
- 'stroke' has a negative coefficient, indicating a negative relationship.

Full Linear Regression Model (model1) Coefficients:

- The full linear regression model includes all features without regularization.
- Some coefficients align with Ridge regression, while others show differences, highlighting the impact of regularization.

Comparison:

- Features selected by forward selection ('curb_weight,' 'horse_power,' 'width,' 'compression,' 'stroke') align with those identified as significant by ANOVA.
- Ridge regression coefficients differ due to the regularization term, with some coefficients being shrunk towards zero. Regularization methods, like Ridge regression, can provide stable models in the presence of multicollinearity.
- Full linear regression model coefficients provide insights into the individual impact of features without regularization.

2.5. Reduce the features with PCA.

PCA is performed on the standardized features, transforms the data into principal components, splits it into training and testing sets, and then fits a linear regression model using the selected principal components.

```
Explained Variance Ratio: [0.47323295 0.21646638 0.09863824 0.06877295 0.04273598]
```

The number of principal components is chosen as 5

Explained Variance Ratio:

1. Principal Component 1 (PC1): 47.32%
2. Principal Component 2 (PC2): 21.65%
3. Principal Component 3 (PC3): 9.86%
4. Principal Component 4 (PC4): 6.88%
5. Principal Component 5 (PC5): 4.27%

Interpretation:

1. **PC1 (47.32%):** This principal component captures the highest proportion of variance in the data. It represents the direction in which the data varies the most.
2. **PC2 (21.65%):** The second principal component explains a substantial portion of the remaining variance after PC1. It is orthogonal to PC1, capturing a different aspect of the data.

3. **PC3 (9.86%):** PC3 captures additional variance not explained by PC1 and PC2. Each subsequent principal component contributes to explaining less variance.
4. **PC4 (6.88%):** The fourth principal component contributes further to the understanding of the data, albeit with a lower proportion of variance.
5. **PC5 (4.27%):** PC5 captures the smallest proportion of variance among the selected components.

Cumulative Explained Variance:

- **PC1 + PC2:** 68.97%
- **PC1 + PC2 + PC3:** 78.83%
- **PC1 + PC2 + PC3 + PC4:** 85.71%
- **PC1 + PC2 + PC3 + PC4 + PC5:** 90.98%

The cumulative explained variance ratio indicates how much of the total variance in the original data is retained by considering multiple principal components.The selected principal components (PC1 to PC5) collectively capture a substantial portion (90.98%) of the variability in the data.