Module 2 Project: Building the Car of the Future

Kejiang Yao

College of Professional Studies, Northeastern University

Professor: Dr. Mary Donhoffner

Date: 10/04/2022

Introduction

For this week's assignment, we are given the car data (*car.cvs* dataset) with attributes of vehicle such as MPG, Cylinder, Displacement, Horsepower, Weight, etc. To assist the car manufacturer on designing an energy efficient car, we are planning to build a model that would help us determine which attributes may contribute to higher gas mileage so that they can design a more fuel-efficient automobile.

There are three parts needed to be complete in this assignment:

Part 1:

Use proper data cleansing techniques to ensure that you have the highest quality data to model this problem. Detail your process and discuss the decisions you made to clean the data.

Part 2:

Build a linear regression model to accurately predict miles per gallon (MPG) based on the attributes of a vehicle. Discuss the significant attributes and how they can help you build the proper car.

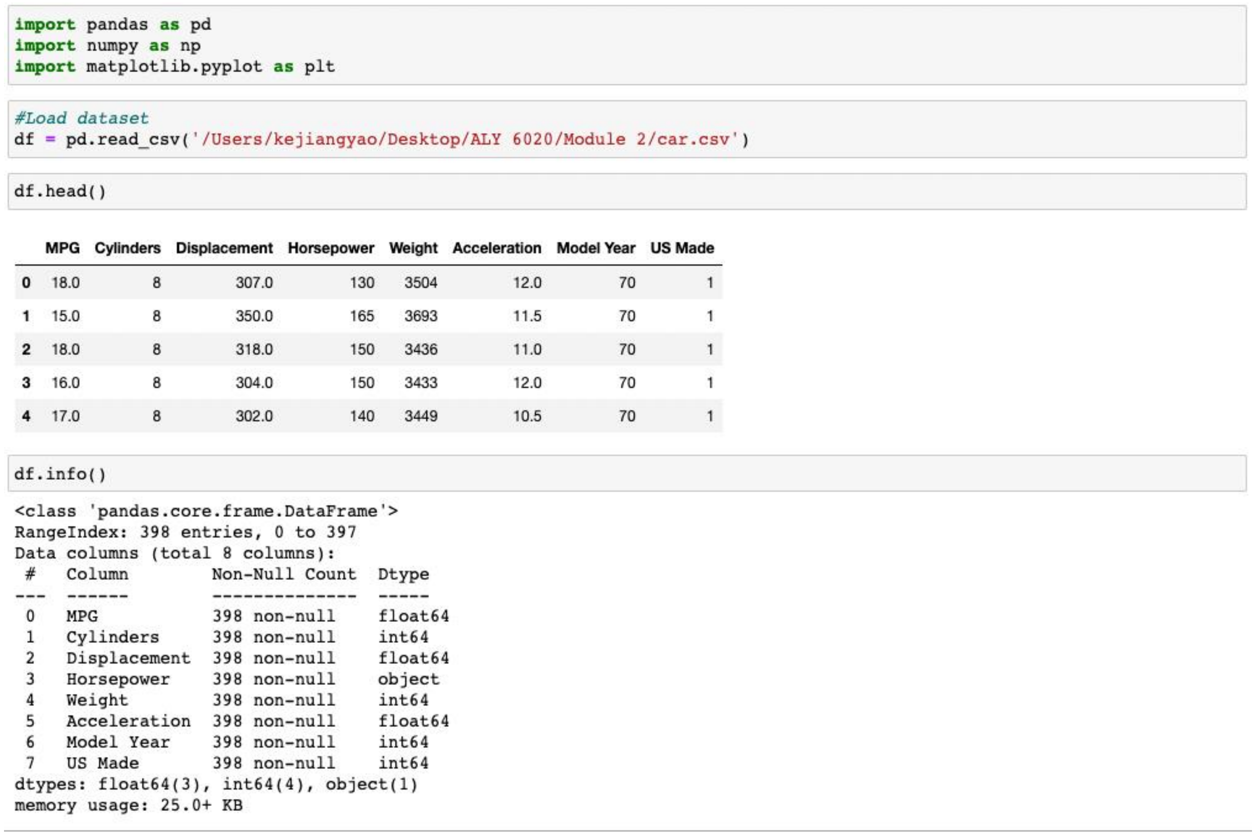
Part 3:

Optimize the model using selection techniques, explain whether the model can achieve the specified goals, and describe which attributes contribute to higher MPG over others.

Both of these three parts will be completed using Python on Jupyter Notebook. Necessary codes and visuals will be attached directly within this report, and the full python script will be submitted separately in another file.

Part 1: Data Cleaning

Firstly, since the ***car.csv*** is a local csv file, we need to load this dataset to the environment for further processing. The screenshot below is the code used for loading data and displaying first five rows of the dataset. There are 8 attributes in this dataset which are MPG, Cylinders, Displacement, Horsepower, Weight, Acceleration, Model Year, and US Made. As we are checking the data type for each attribute, the attribute, Horsepower is unnormal which is Object. All of the attributes in this dataset are numeric. Therefore, we have to convert the data type of this attribute to either float or int64.



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
#Load dataset
df = pd.read_csv('/Users/kejiangyao/Desktop/ALY 6020/Module 2/car.csv')
```

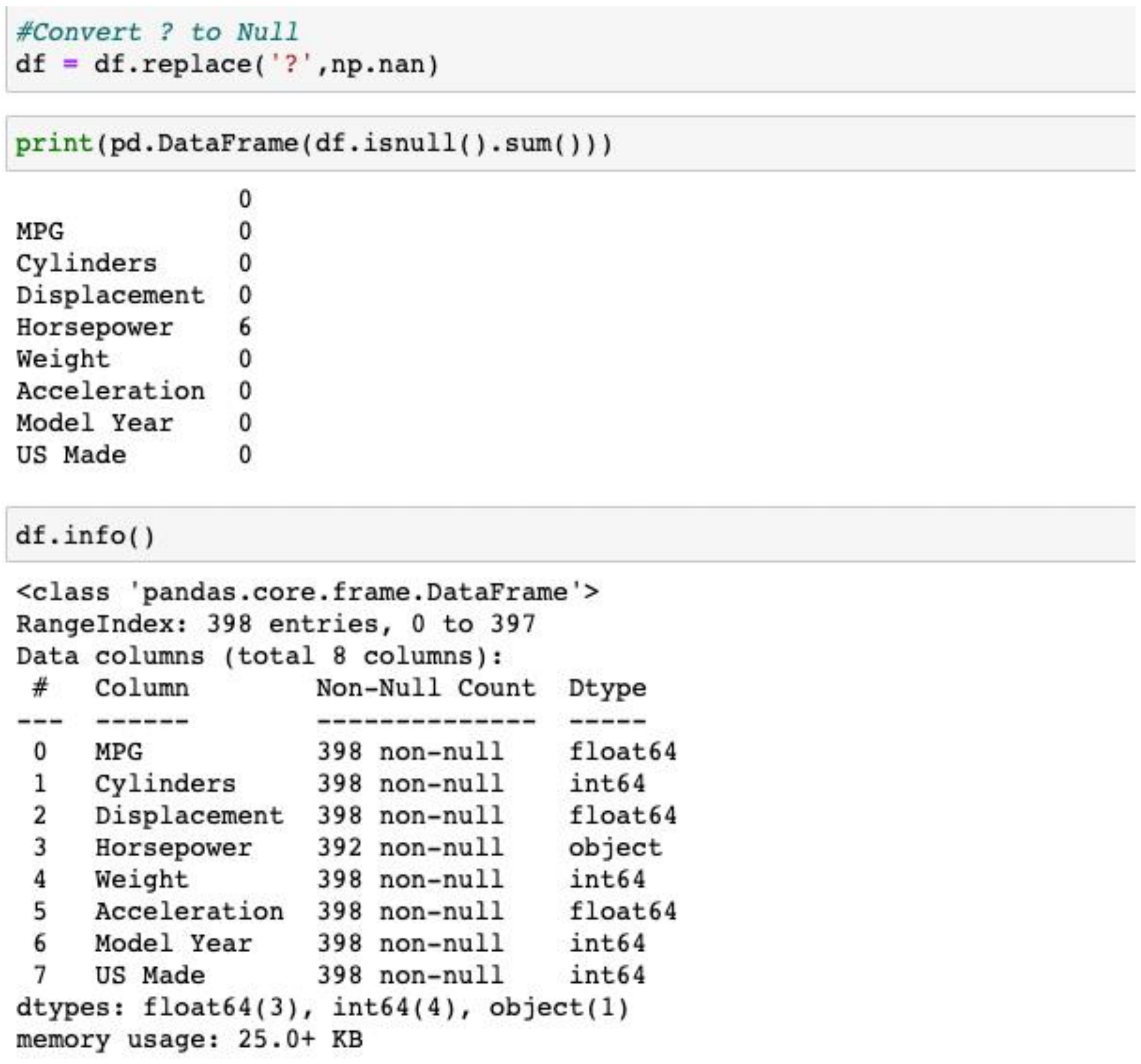
```
df.head()
```

	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model Year	US Made
0	18.0	8	307.0	130	3504	12.0	70	1
1	15.0	8	350.0	165	3693	11.5	70	1
2	18.0	8	318.0	150	3436	11.0	70	1
3	16.0	8	304.0	150	3433	12.0	70	1
4	17.0	8	302.0	140	3449	10.5	70	1

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   MPG           398 non-null    float64
 1   Cylinders     398 non-null    int64
 2   Displacement  398 non-null    float64
 3   Horsepower    398 non-null    object
 4   Weight        398 non-null    int64
 5   Acceleration  398 non-null    float64
 6   Model Year    398 non-null    int64
 7   US Made       398 non-null    int64
dtypes: float64(3), int64(4), object(1)
memory usage: 25.0+ KB
```

As we are further diving into this attribute, there are several null values denoted by ‘**?**’. I converted the ‘**?**’ mark into NaN value and count the number of them. Below is the screenshot of the code and the number of null values in each attribute. There are 6 null values in total which are all from Horsepower.



```
#Convert ? to Null
df = df.replace('?',np.nan)
```

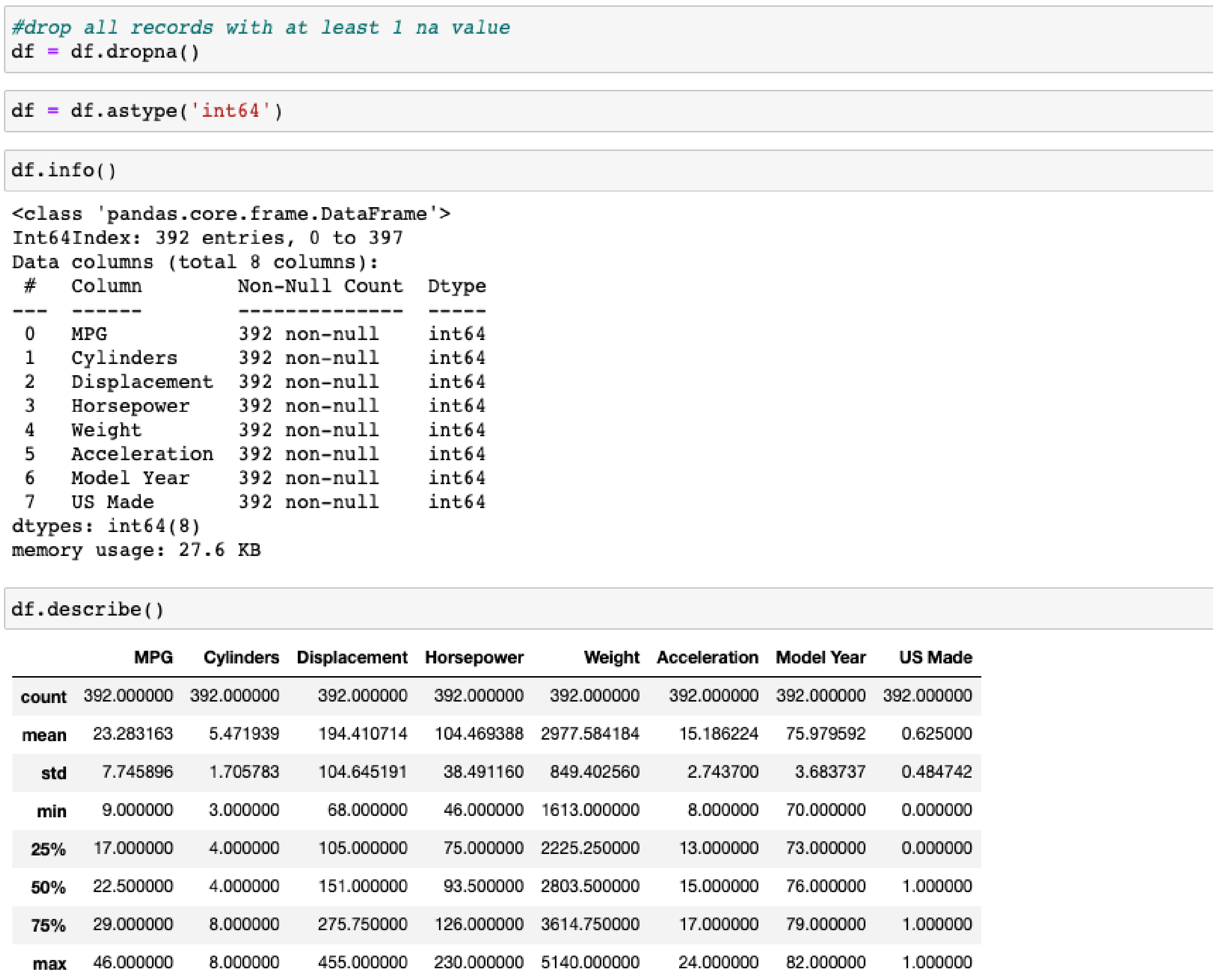
```
print(pd.DataFrame(df.isnull().sum()))
```

```
                0
MPG             0
Cylinders       0
Displacement    0
Horsepower      6
Weight          0
Acceleration    0
Model Year      0
US Made         0
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   MPG           398 non-null    float64
 1   Cylinders     398 non-null    int64
 2   Displacement  398 non-null    float64
 3   Horsepower    392 non-null    object
 4   Weight        398 non-null    int64
 5   Acceleration  398 non-null    float64
 6   Model Year    398 non-null    int64
 7   US Made       398 non-null    int64
dtypes: float64(3), int64(4), object(1)
memory usage: 25.0+ KB
```

Since there are only 6 null values which only account for 1.5% of the dataset, to keep the data intact and unbiased, I directly removed all 6 rows without further manipulations. After removing null values, I changed the data type to int64 and display the summary statistics of all attributes. From the screenshot below, we can see that there are 392 records and 8 columns left. For example, the average MPG of these observations is 23.283163, standard deviation is 7.745896, minimum MPG is 9, and maximum MPG is 46.



```
#drop all records with at least 1 na value
df = df.dropna()
```

```
df = df.astype('int64')
```

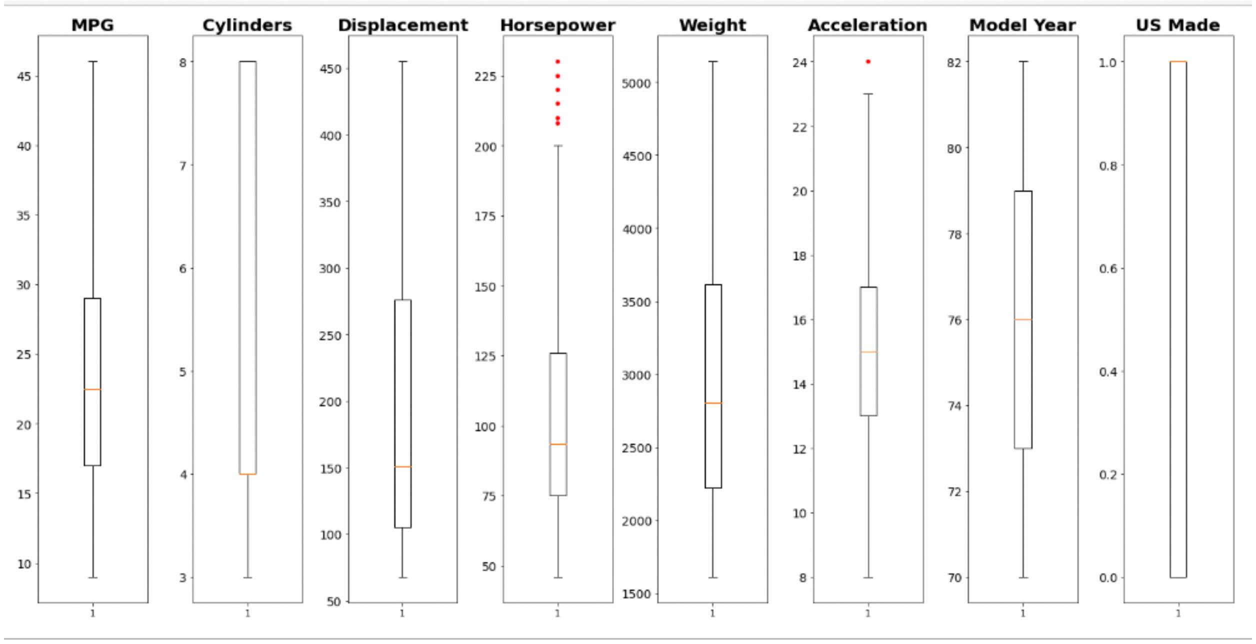
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 392 entries, 0 to 397
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   MPG           392 non-null    int64
 1   Cylinders     392 non-null    int64
 2   Displacement  392 non-null    int64
 3   Horsepower    392 non-null    int64
 4   Weight        392 non-null    int64
 5   Acceleration  392 non-null    int64
 6   Model Year    392 non-null    int64
 7   US Made       392 non-null    int64
dtypes: int64(8)
memory usage: 27.6 KB
```

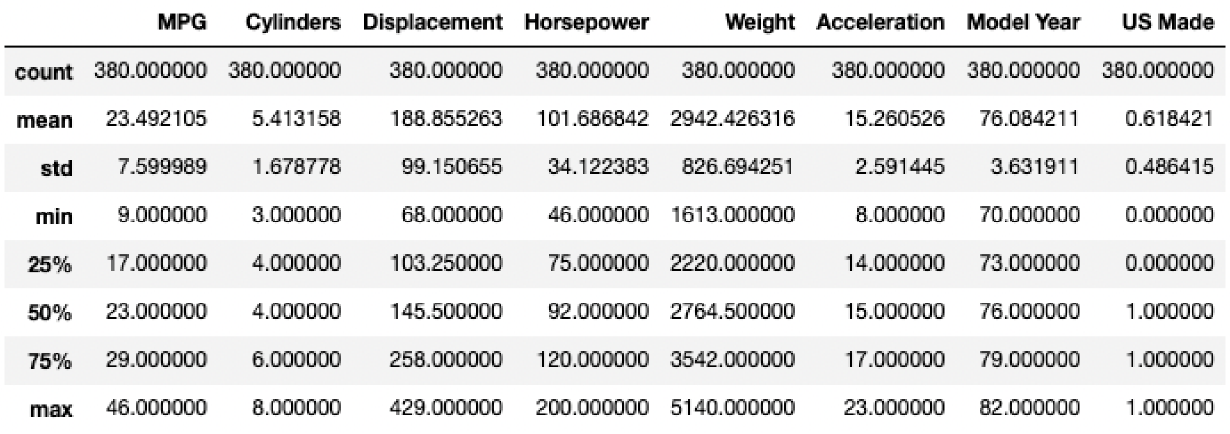
```
df.describe()
```

	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model Year	US Made
count	392.000000	392.000000	392.000000	392.000000	392.000000	392.000000	392.000000	392.000000
mean	23.283163	5.471939	194.410714	104.469388	2977.584184	15.186224	75.979592	0.625000
std	7.745896	1.705783	104.645191	38.491160	849.402560	2.743700	3.683737	0.484742
min	9.000000	3.000000	68.000000	46.000000	1613.000000	8.000000	70.000000	0.000000
25%	17.000000	4.000000	105.000000	75.000000	2225.250000	13.000000	73.000000	0.000000
50%	22.500000	4.000000	151.000000	93.500000	2803.500000	15.000000	76.000000	1.000000
75%	29.000000	8.000000	275.750000	126.000000	3614.750000	17.000000	79.000000	1.000000
max	46.000000	8.000000	455.000000	230.000000	5140.000000	24.000000	82.000000	1.000000

I further visualized the distribution of data points for each attribute. From the boxplots below, we can observe that there are some outliers in ***Horsepower*** and ***Acceleration***. Since the outliers may cause by measurement errors, sampling problem, or natural variation, including them to build the model would affect model's predictive power and could not accurately generalize the trend of the data. Therefore, I decide to remove all outliers from theses two attributes.



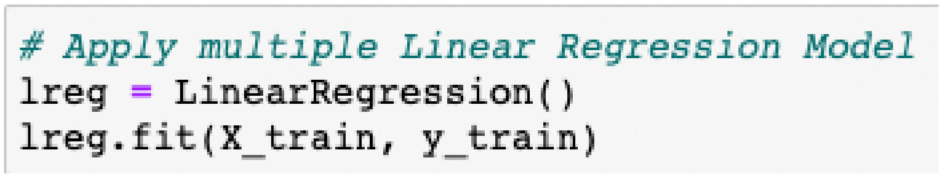
Below is the screenshot of summary statistics after removing all outliers from the dataset.



	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model Year	US Made
count	380.000000	380.000000	380.000000	380.000000	380.000000	380.000000	380.000000	380.000000
mean	23.492105	5.413158	188.855263	101.686842	2942.426316	15.260526	76.084211	0.618421
std	7.599989	1.678778	99.150655	34.122383	826.694251	2.591445	3.631911	0.486415
min	9.000000	3.000000	68.000000	46.000000	1613.000000	8.000000	70.000000	0.000000
25%	17.000000	4.000000	103.250000	75.000000	2220.000000	14.000000	73.000000	0.000000
50%	23.000000	4.000000	145.500000	92.000000	2764.500000	15.000000	76.000000	1.000000
75%	29.000000	6.000000	258.000000	120.000000	3542.000000	17.000000	79.000000	1.000000
max	46.000000	8.000000	429.000000	200.000000	5140.000000	23.000000	82.000000	1.000000

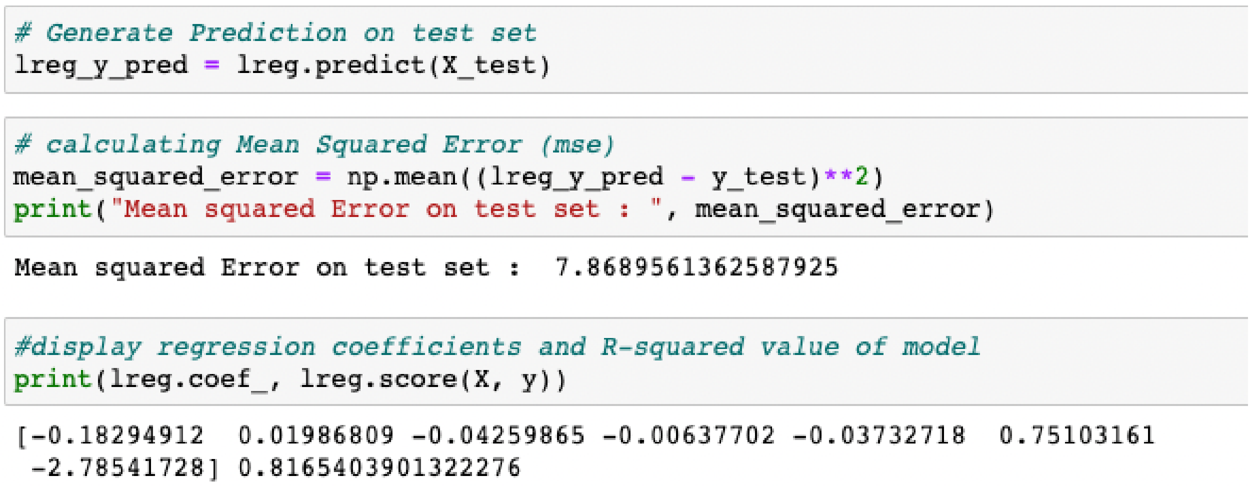
Part 2: Model Building

To build a linear regression model that used to predict the predict miles per gallon (MPG) based on the attributes of a vehicle such as Cylinders, Displacement, Horsepower, Weight, Acceleration, Model Year, and US Made, firstly we need to separate the data into predictor variables (X) and response variable (Y). Then we need to split the dataset into training and testing set by 75% and 25% for later testing of model accuracy. We use the linear regression model from Scikit-learn package and train the model with our training set data.



```
# Apply multiple Linear Regression Model
lreg = LinearRegression()
lreg.fit(X_train, y_train)
```

After training the model, we could use the testing set to predict the MPG. Meanwhile, we also calculated mean square error and r squared of the model to test the accuracy. The r squared value equals to 0.8165 which means using these 7 attributes to build our model could successfully explain around 81.65% of the variation on testing dataset.



```
# Generate Prediction on test set
lreg_y_pred = lreg.predict(X_test)
```

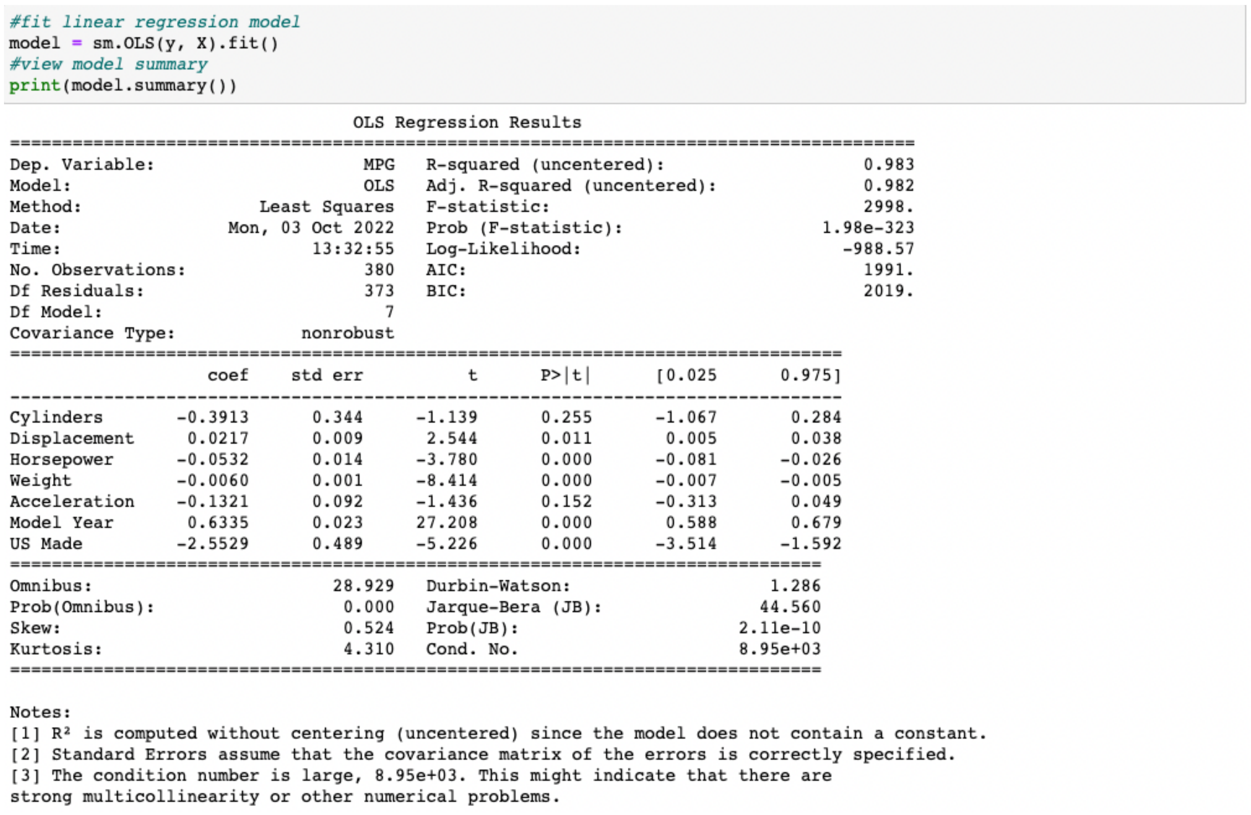
```
# calculating Mean Squared Error (mse)
mean_squared_error = np.mean((lreg_y_pred - y_test)**2)
print("Mean squared Error on test set : ", mean_squared_error)
```

```
Mean squared Error on test set :  7.8689561362587925
```

```
#display regression coefficients and R-squared value of model
print(lreg.coef_, lreg.score(X, y))
```

```
[-0.18294912  0.01986809 -0.04259865 -0.00637702 -0.03732718  0.75103161
 -2.78541728] 0.8165403901322276
```

I also applied the OLS method to build the linear regression model using the OLS function from statsmodels.api package. We fitted the model directly using all data from the dataset without doing train-test split. The screenshot below is the summary table of OLS regression result. From the Adjusted R squared value shown below, we can know that around 98.2% of variation in MPG could be explained by the rest of 7 attributes in this dataset. The other important metric to focus on is the p value for each attribute. If we set the threshold of rejecting null hypothesis is 0.05, there are 5 statistically significant attributes (Displacement, Horsepower, Weight, Model Year, and US Made) from this dataset.



```
#fit linear regression model
model = sm.OLS(y, X).fit()
#view model summary
print(model.summary())
```

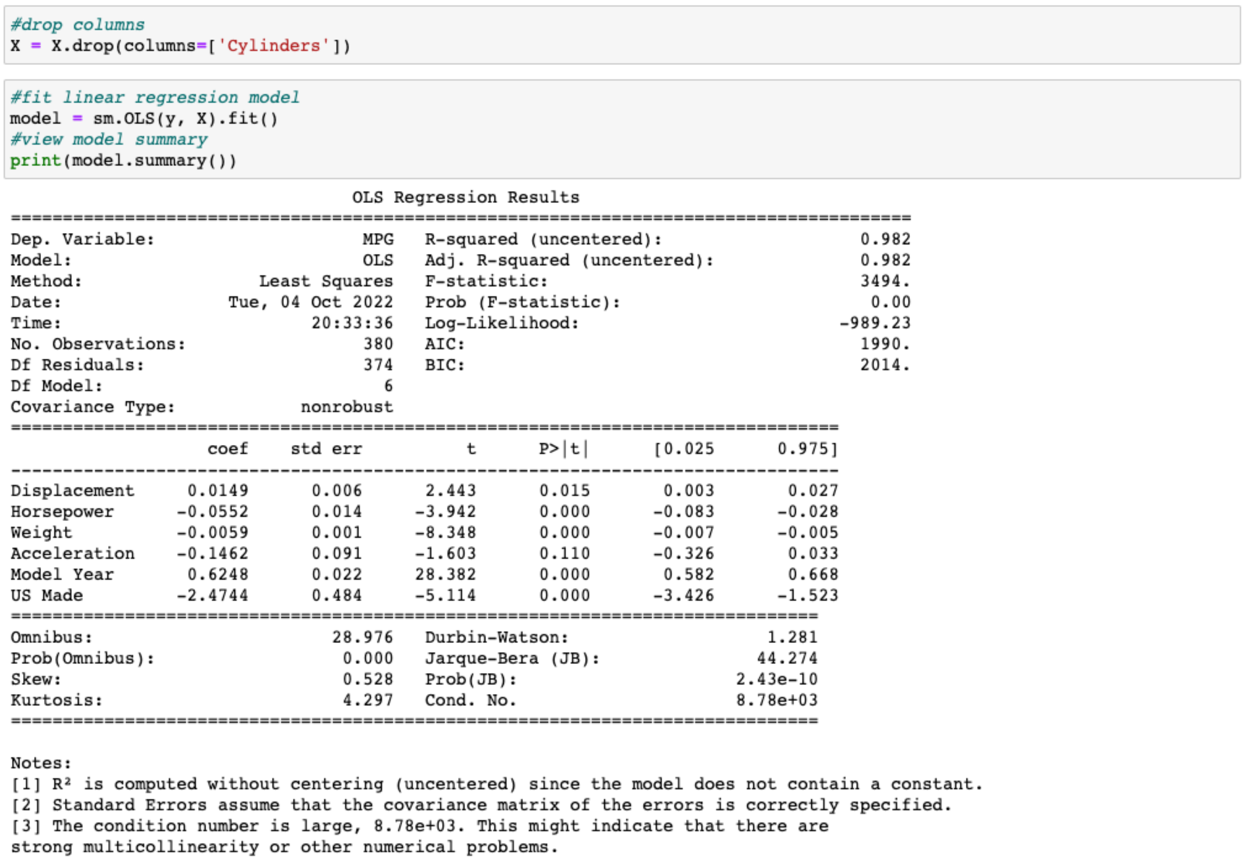
```
                                 OLS Regression Results                                
=======================================================================================
Dep. Variable:                    MPG   R-squared (uncentered):                   0.983
Model:                            OLS   Adj. R-squared (uncentered):              0.982
Method:                 Least Squares   F-statistic:                              2998.
Date:                Mon, 03 Oct 2022   Prob (F-statistic):                   1.98e-323
Time:                        13:32:55   Log-Likelihood:                         -988.57
No. Observations:                 380   AIC:                                      1991.
Df Residuals:                     373   BIC:                                      2019.
Df Model:                           7                                                  
Covariance Type:            nonrobust                                                  
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Cylinders       -0.3913      0.344     -1.139      0.255      -1.067       0.284
Displacement     0.0217      0.009      2.544      0.011       0.005       0.038
Horsepower      -0.0532      0.014     -3.780      0.000      -0.081      -0.026
Weight          -0.0060      0.001     -8.414      0.000      -0.007      -0.005
Acceleration    -0.1321      0.092     -1.436      0.152      -0.313       0.049
Model Year       0.6335      0.023     27.208      0.000       0.588       0.679
US Made         -2.5529      0.489     -5.226      0.000      -3.514      -1.592
==============================================================================
Omnibus:                       28.929   Durbin-Watson:                   1.286
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               44.560
Skew:                           0.524   Prob(JB):                     2.11e-10
Kurtosis:                       4.310   Cond. No.                     8.95e+03
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 8.95e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

These attributes are important if we want to build a fuel-efficient car. For example, the coefficient of US Made is -2.5529. If there is a car switching from US Made to Non-US made, the MPG will increase by 2.5529 unit. In this way, if we want to build a car with high MPG, looking at coefficient of each attribute, we need to maximize the car displacement and ,model year, since the coefficient of these attribute is positive, and minimize the horsepower, weight, and US made which is avoiding making the car in US, since the coefficient of these attribute is negative.

Part 3: Model Optimization

To optimize the model to achieve higher R squared value and F-statistics, we are going to use backward selection technique to remove one variable with the highest p-value each time until all variables' p-value is lower than 0.0005. From the previous OLS result summary table, the attribute cylinder has highest p-value 0.255. Therefore, we are going to remove this attribute and run the regression again to see if there is any improvement to the model.



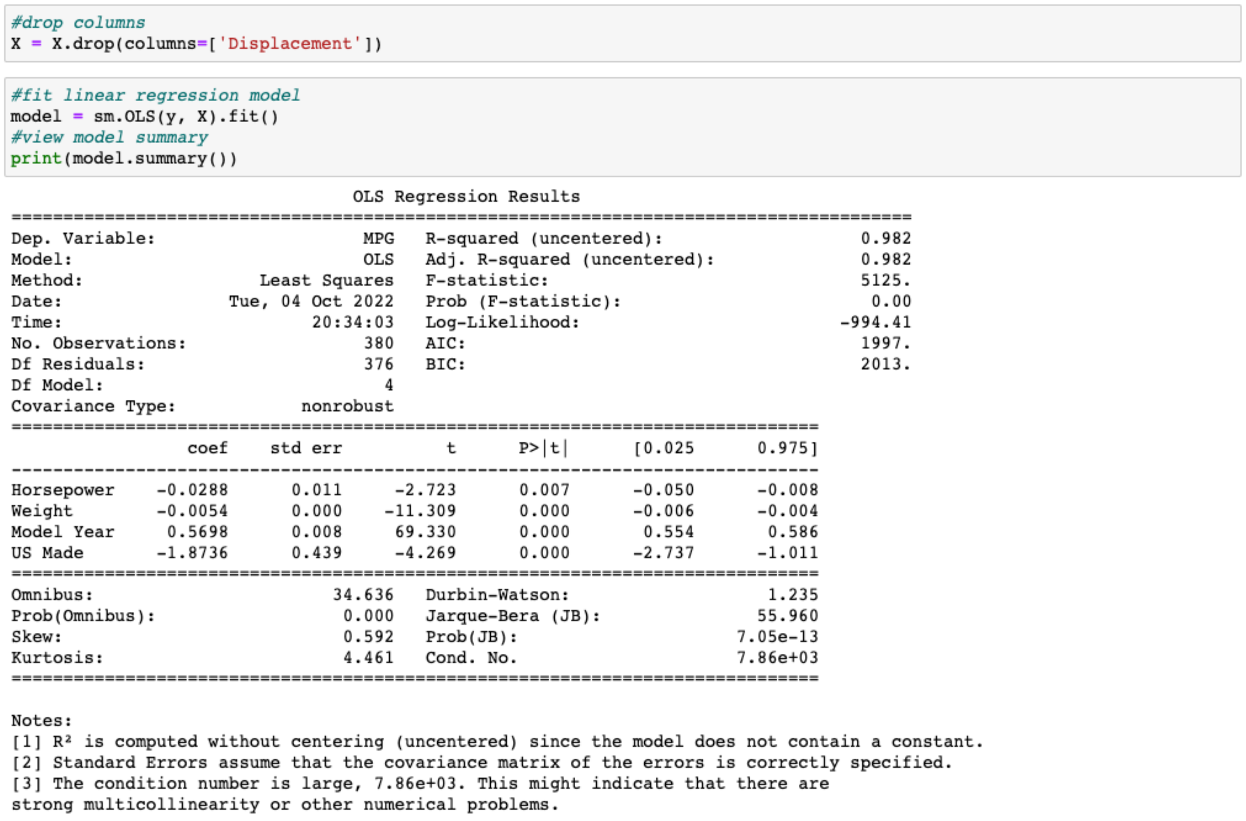
```
#drop columns
X = X.drop(columns=['Cylinders'])
```

```
#fit linear regression model
model = sm.OLS(y, X).fit()
#view model summary
print(model.summary())
```

```
                                 OLS Regression Results                                
=======================================================================================
Dep. Variable:                    MPG   R-squared (uncentered):                   0.982
Model:                            OLS   Adj. R-squared (uncentered):              0.982
Method:                 Least Squares   F-statistic:                              3494.
Date:                Tue, 04 Oct 2022   Prob (F-statistic):                        0.00
Time:                        20:33:36   Log-Likelihood:                         -989.23
No. Observations:                 380   AIC:                                      1990.
Df Residuals:                     374   BIC:                                      2014.
Df Model:                           6                                                  
Covariance Type:            nonrobust                                                  
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Displacement     0.0149      0.006      2.443      0.015       0.003       0.027
Horsepower      -0.0552      0.014     -3.942      0.000      -0.083      -0.028
Weight          -0.0059      0.001     -8.348      0.000      -0.007      -0.005
Acceleration    -0.1462      0.091     -1.603      0.110      -0.326       0.033
Model Year       0.6248      0.022     28.382      0.000       0.582       0.668
US Made         -2.4744      0.484     -5.114      0.000      -3.426      -1.523
==============================================================================
Omnibus:                       28.976   Durbin-Watson:                   1.281
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               44.274
Skew:                           0.528   Prob(JB):                     2.43e-10
Kurtosis:                       4.297   Cond. No.                     8.78e+03
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 8.78e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

As we can see from the screenshot above, after removing the attribute '***Cylinder'***, the adjusted R-squared value does not change, but the F-statistics increased from 2998 to 3494, which means dropping this attribute improve the model's fit.



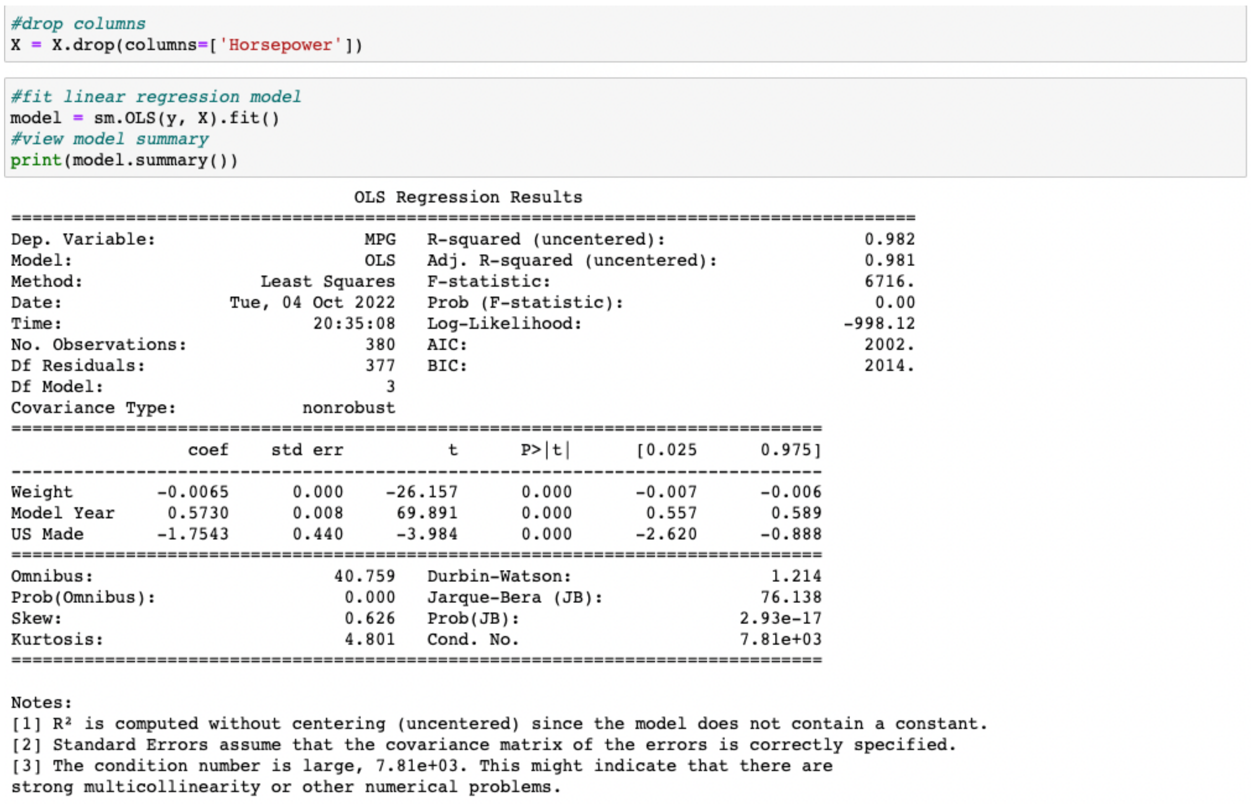
```
#drop columns
X = X.drop(columns=['Displacement'])
```

```
#fit linear regression model
model = sm.OLS(y, X).fit()
#view model summary
print(model.summary())
```

```
                                 OLS Regression Results
=======================================================================================
Dep. Variable:                    MPG   R-squared (uncentered):                   0.982
Model:                            OLS   Adj. R-squared (uncentered):              0.982
Method:                 Least Squares   F-statistic:                              5125.
Date:                Tue, 04 Oct 2022   Prob (F-statistic):                        0.00
Time:                        20:34:03   Log-Likelihood:                         -994.41
No. Observations:                 380   AIC:                                      1997.
Df Residuals:                     376   BIC:                                      2013.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Horsepower    -0.0288      0.011     -2.723      0.007      -0.050      -0.008
Weight        -0.0054      0.000    -11.309      0.000      -0.006      -0.004
Model Year     0.5698      0.008     69.330      0.000       0.554       0.586
US Made       -1.8736      0.439     -4.269      0.000      -2.737      -1.011
==============================================================================
Omnibus:                       34.636   Durbin-Watson:                   1.235
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               55.960
Skew:                           0.592   Prob(JB):                     7.05e-13
Kurtosis:                       4.461   Cond. No.                     7.86e+03
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 7.86e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

We continued to remove the attribute with highest p-value from our regression. This time we removed '***Displacement'***. As the screenshot above shown, after dropping this attribute, the adjusted R-squared value still does not change, but the F-statistics increased from 3494 to 5125, which means dropping this attribute continually improve the model's fit.



```
#drop columns
X = X.drop(columns=['Horsepower'])
```

```
#fit linear regression model
model = sm.OLS(y, X).fit()
#view model summary
print(model.summary())
```

```
                                 OLS Regression Results                                
=======================================================================================
Dep. Variable:                    MPG   R-squared (uncentered):                   0.982
Model:                            OLS   Adj. R-squared (uncentered):              0.981
Method:                 Least Squares   F-statistic:                              6716.
Date:                Tue, 04 Oct 2022   Prob (F-statistic):                        0.00
Time:                        20:35:08   Log-Likelihood:                         -998.12
No. Observations:                 380   AIC:                                      2002.
Df Residuals:                     377   BIC:                                      2014.
Df Model:                           3                                                  
Covariance Type:            nonrobust                                                  
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Weight        -0.0065      0.000    -26.157      0.000      -0.007      -0.006
Model Year     0.5730      0.008     69.891      0.000       0.557       0.589
US Made       -1.7543      0.440     -3.984      0.000      -2.620      -0.888
==============================================================================
Omnibus:                       40.759   Durbin-Watson:                   1.214
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               76.138
Skew:                           0.626   Prob(JB):                     2.93e-17
Kurtosis:                       4.801   Cond. No.                     7.81e+03
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 7.81e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

We continued to drop the highest p-value attribute. This time after we dropped ***'Horsepower'*** , all the attributes now have p-value lower than 0.0005. Therefore, we can stop dropping anymore variables after this. As the screenshot above shown, after dropping this attribute, the adjusted R-squared value dropped from 0.982 to 0.98 which is not desired since lower R-squared value means the model could explain lower variation in response variable (MPG). However, the F-statistics increased from 5125 to 6716, which means dropping this attribute continually improve the model's fit to the overall data.

In the end, we could also conclude that the attributes Weight, Model Year, and US Made have higher contribution to the higher MPG over the other variables.

Conclusion

For this week’s assignment, we used linear regression and OLS method to build the model to assist the car manufacturer on designing an energy efficient car by finding which attributes have more contribution to the higher MPG. Using the backward feature selection method, we found that Weight, Model Year, and US Made have higher contribution to the MPG than the rest of attributes. For model improvement and future optimization, we could split the data set into training set and testing set and comparing other feature selection/regularization techniques such as LASSO, Ridge, and Elastic net to find which is the best method to improve the model.

Reference:

Stepwise Regression Tutorial in Python - Towards Data Science. (2022, January 7). Medium. Retrieved October 4, 2022, from https://towardsdatascience.com/stepwise-regression-tutorial-in-python-ebf7c782c922

McDonald, A. (2022, July 15). *Creating Boxplots Using Matplotlib in Python - Towards Data Science*. Medium. Retrieved October 4, 2022, from https://towardsdatascience.com/creating-boxplots-of-well-log-data-using-matplotlib-in-python-34c3816e73f4

GeeksforGeeks. (2021, May 15). *Implementation of Lasso, Ridge and Elastic Net*. Retrieved October 4, 2022, from https://www.geeksforgeeks.org/implementation-of-lasso-ridge-and-elastic-net/