MTH 522: Advanced Mathematical Statistics

Predicting MPG with Multiple Variables Using Multiple Regression

By: Dhyey Doshi

02/19/2023

Issues

Using the predictor variables "displacement," "weight," "horsepower," and "acceleration," we design a multiple regression model in this report to predict a vehicle's miles per gallon (MPG). Multiple regression refers to the process of attempting to predict Something by combining two or more variables. We discuss examining the variable that is most useful for predicting MPG as a baseline variable to base the remainder of the model on. The key variables for predicting MPG are covered in this report. When creating a multiple regression model, it's possible that only some predictor variables will be helpful and others won't.

Findings

After receiving the results, we can draw the reasonable conclusion that the model works well, with an r-squared of 0.61, or around 61%, indicating that the model accurately describes the data. Additionally, we draw the conclusion that the residuals exhibit almost normal distribution, indicating that our fitted values are relatively correct and the predictions exhibit minimal error. The residuals do not follow a perfect normal distribution, and the variables in the model can only precisely represent roughly 61% of the miles per gallon data. This should be taken into consideration when planning to utilize the model.

Discussion

We find solutions to our problem, which enabled us to reach our conclusion, by using statsmodel in the python programming language. One of our first problems is identifying the most useful variable from which to base the prediction, and these methods help us realize that factors like weight are crucial in predicting miles per gallon for a car. Coupled with these straightforward summary statistics, we were able to properly add variables to the multiple regression model, enabling us to create a model that performs admirably and accounts for around 61% of the data related to miles per gallon.

Appendix A: Methods

The automobile data utilized in this report has five variables. Four factors—displacement, weight, horsepower, and acceleration—are predictors, and one is the target value—miles per gallon, or MPG—that we want to predict. The first step is to choose a useful predictor variable, one that best aids in predicting miles per gallon. To achieve this, we compute a basic linear model for each predictor variable, i.e., displacement predicts MPG, followed by weight predicts MPG, giving us a total of four simple models. We also take model summary to look at r^2 value for each model and identify best variable for best model. After building the multiple regression model with forward selection process we must determine its accuracy, to do this check the if the residuals/errors follow a normal distribution. We have used quantile plot for the residual points on the line.

Appendix B: Results

Each Model Summary Charts

Displacement

Dep. Variable:	mpg	**R-squared:**	0.623
Model:	OLS	**Adj. R-squared:**	0.622
Method:	Least Squares	**F-statistic:**	656.5
Date:	Sun, 19 Feb 2023	**Prob (F-statistic):**	3.69e-86
Time:	21:20:04	**Log-Likelihood:**	-1183.1
No. Observations:	399	**AIC:**	2370.
Df Residuals:	397	**BIC:**	2378.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	34.9557	0.491	71.183	0.000	33.990	35.921
displacement	-0.0587	0.002	-25.623	0.000	-0.063	-0.054

Omnibus:	26.466	**Durbin-Watson:**	2.024
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	31.706
Skew:	0.577	**Prob(JB):**	1.30e-07
Kurtosis:	3.760	**Cond. No.**	447.

#Horsepower

Dep. Variable:	mpg	**R-squared:**	0.557
Model:	OLS	**Adj. R-squared:**	0.556
Method:	Least Squares	**F-statistic:**	498.8
Date:	Sun, 19 Feb 2023	**Prob (F-statistic):**	3.74e-72
Time:	21:22:20	**Log-Likelihood:**	-1215.4
No. Observations:	399	**AIC:**	2435.
Df Residuals:	397	**BIC:**	2443.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	38.9850	0.721	54.034	0.000	37.567	40.403
horsepower	-0.1472	0.007	-22.334	0.000	-0.160	-0.134

Omnibus:	16.637	**Durbin-Watson:**	1.950
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	17.736
Skew:	0.514	**Prob(JB):**	0.000141
Kurtosis:	3.104	**Cond. No.**	309.

#Weight

#Acceleration

Dep. Variable:	mpg	**R-squared:**	0.658
Model:	OLS	**Adj. R-squared:**	0.657
Method:	Least Squares	**F-statistic:**	764.1
Date:	Sun, 19 Feb 2023	**Prob (F-statistic):**	1.50e-94
Time:	21:23:02	**Log-Likelihood:**	-1163.7
No. Observations:	399	**AIC:**	2331.
Df Residuals:	397	**BIC:**	2339.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	46.0649	0.832	55.358	0.000	44.429	47.701
weight	-0.0076	0.000	-27.642	0.000	-0.008	-0.007

Omnibus:	23.726	**Durbin-Watson:**	2.138
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	26.981
Skew:	0.562	**Prob(JB):**	1.38e-06
Kurtosis:	3.599	**Cond. No.**	1.13e+04

Dep. Variable:	mpg	**R-squared:**	0.122
Model:	OLS	**Adj. R-squared:**	0.120
Method:	Least Squares	**F-statistic:**	55.25
Date:	Sun, 19 Feb 2023	**Prob (F-statistic):**	6.57e-13
Time:	21:23:29	**Log-Likelihood:**	-1351.8
No. Observations:	399	**AIC:**	2708.
Df Residuals:	397	**BIC:**	2716.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	9.3004	1.999	4.653	0.000	5.371	13.230
acceleration	0.9286	0.125	7.433	0.000	0.683	1.174

Omnibus:	19.834	**Durbin-Watson:**	2.014
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	21.934
Skew:	0.568	**Prob(JB):**	1.73e-05
Kurtosis:	2.834	**Cond. No.**	89.3

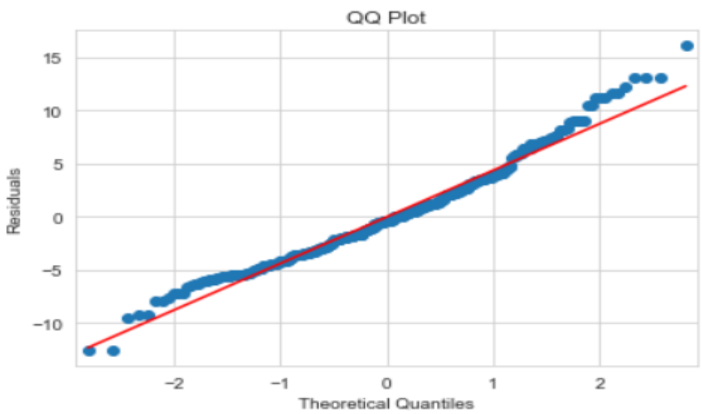
#Multiple Regression Summary

Dep. Variable:	mpg	**R-squared:**	0.671
Model:	OLS	**Adj. R-squared:**	0.670
Method:	Least Squares	**F-statistic:**	404.2
Date:	Sun, 19 Feb 2023	**Prob (F-statistic):**	2.27e-96
Time:	21:42:17	**Log-Likelihood:**	-1155.9
No. Observations:	399	**AIC:**	2318.
Df Residuals:	396	**BIC:**	2330.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	42.8992	1.141	37.592	0.000	40.656	45.143
displacement	-0.0213	0.005	-3.974	0.000	-0.032	-0.011
weight	-0.0051	0.001	-7.604	0.000	-0.006	-0.004

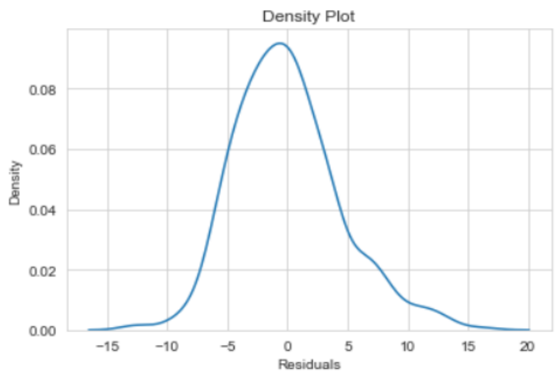
Omnibus:	26.639	**Durbin-Watson:**	2.107
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	31.398
Skew:	0.590	**Prob(JB):**	1.52e-07
Kurtosis:	3.704	**Cond. No.**	1.58e+04

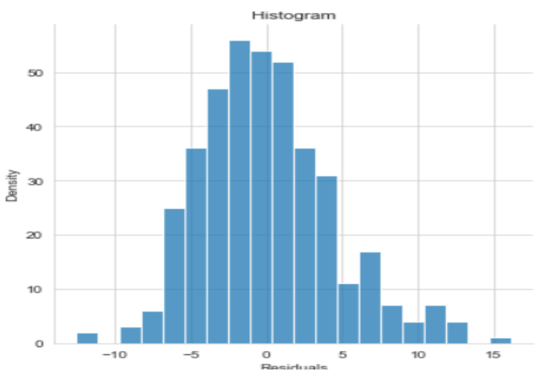
Below a quantile plot can be found of the multiple regression models residuals, this is testing whether the residuals follow a normal distribution.



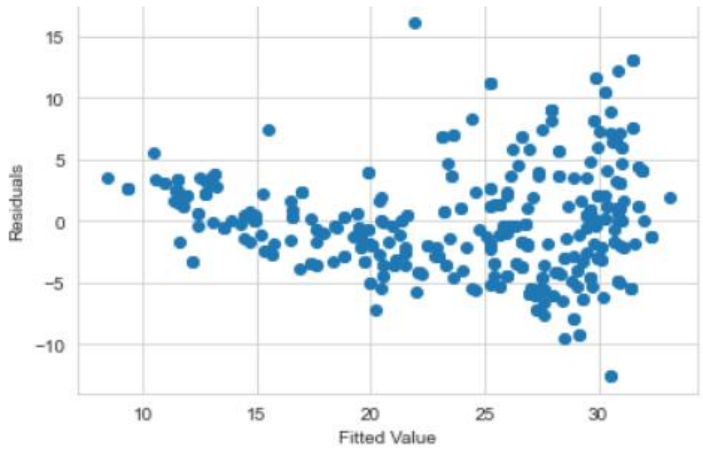
#Density Plot

#Histogram





#Test of heteroscedasticity



Appendix C: Code

```
import numpy as np
import pandas as pd
import scipy.stats as sts
import statsmodels.api as sm
from statsmodels.formula.api import ols
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_excel('E:\\Study\\MTH\\auto_data_doshi_dhyey.xlsx')

df.describe()
```

```
displacement = df['displacement']
horsepower = df['horsepower']
weight = df['weight']
acceleration = df['acceleration']
mpg = df['mpg']


#Displacement

displacement = sm.add_constant(displacement)
displacementModel = sm.OLS(mpg, displacement).fit()
displacementModel.summary()


#Horsepower

horsepower = sm.add_constant(horsepower)
horsepowerModel = sm.OLS(mpg, horsepower).fit()
horsepowerModel.summary()


#Weight

weight = sm.add_constant(weight)
weightModel = sm.OLS(mpg, weight).fit()
weightModel.summary()


#Acceleration

acceleration = sm.add_constant(acceleration)
accelerationModel = sm.OLS(mpg, acceleration).fit()
accelerationModel.summary()


print('Displacement:', np.sum(np.square(displacementModel.resid)))
print('Horsepower:', np.sum(np.square(horsepowerModel.resid)))
print('Weight:', np.sum(np.square(weightModel.resid)))
print('Acceleration:', np.sum(np.square(accelerationModel.resid)))


X = df[['displacement', 'weight']]
y = df[['mpg']]
X = sm.add_constant(X)
multiRegModel = sm.OLS(y, X).fit()
multiRegResiduals = multiRegModel.resid
multiRegModel.summary()
```

```
sm.qqplot(multiRegResiduals, line='s')
plt.title('QQ Plot')
plt.ylabel('Residuals')
plt.show()

sns.kdeplot(multiRegResiduals)
plt.ylabel('Density')
plt.xlabel('Residuals')
plt.title('Density Plot')
plt.show()

sns.displot(multiRegResiduals)
plt.ylabel('Density')
plt.xlabel('Residuals')
plt.title('Histogram')
plt.show()

plt.scatter(multiRegModel.fittedvalues, multiRegResiduals)
plt.xlabel("Fitted Value")
plt.ylabel("Residuals")
plt.show()
```