In [2]:
```
import pandas as pd
```

In [4]:
```
df = pd.read_csv('auto_mpg.data', header = None, sep = '\s+')

df.columns = ['mpg', 'cylinders', 'displacement', 'horsepower', 'weight', 'acceleration', 'model year', 'origin', 'car na
df.head()
```

Out[4]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130.0	3504.0	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693.0	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150.0	3436.0	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150.0	3433.0	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140.0	3449.0	10.5	70	1	ford torino

In [6]:
```
# Drop unwanted features and prepare Feature matrix
X = df.drop(['mpg','car name', 'model year', 'origin', 'horsepower'], axis = 1)

# Prepare target vector
y = df['mpg'].values

# Dimensions of Feature matrix
print(X.shape)

# Dimensions of Target vector
print(y.shape)

# Display first 5 records in Feature matrix
X.head()

# Display target vector
# print(y)
```

```
(398, 4)
(398,)
```

Out[6]:

	cylinders	displacement	weight	acceleration
0	8	307.0	3504.0	12.0
1	8	350.0	3693.0	11.5
2	8	318.0	3436.0	11.0
3	8	304.0	3433.0	12.0
4	8	302.0	3449.0	10.5

In [7]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```

Multiple Linear Regression

In [9]:
```
# Split the data into Train and Test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3)

# Instantiate LinearRegression class
lr = LinearRegression()

# Fit/Train the model
lr.fit(X_train, y_train)

# Use the model for prediction
y_pred = lr.predict(X_test)

# Measure the performance of the model
print("R-Squared: ", r2_score(y_test, y_pred))

# Display the parameters of the model
print("Weights: ", lr.coef_)
print("Constant: ", lr.intercept_)
```

```
R-Squared:  0.6635951993802349
Weights:  [-5.04512716e-01  1.19085636e-05 -6.34372554e-03  2.62586611e-01]
Constant:  41.005255793325404
```

Ridge Regression

In [10]:
```
from sklearn.linear_model import Ridge
rr = Ridge(alpha = 100)
rr.fit(X_train, y_train)
y_pred = rr.predict(X_test)

print("R-Squared: ", r2_score(y_test, y_pred))
print("Weights: ", lr.coef_)
print("Constant: ", lr.intercept_)
```

```
R-Squared:  0.6661413226056381
Weights:  [-5.04512716e-01  1.19085636e-05 -6.34372554e-03  2.62586611e-01]
Constant:  41.005255793325404
```

Lasso Regression

In [11]:
```
from sklearn.linear_model import Lasso
lasso = Lasso()
lasso.fit(X_train, y_train)
y_pred = lasso.predict(X_test)

print("R-Squared: ", r2_score(y_test, y_pred))
print("Weights: ", lasso.coef_)
print("Constant: ", lasso.intercept_)
```

```
R-Squared:  0.6716898404315739
Weights:  [-0.         -0.01315825 -0.00599659  0.06124802]
Constant:  42.92316848970836
```

In [ ]:
```

```