Name: Sarah Huang

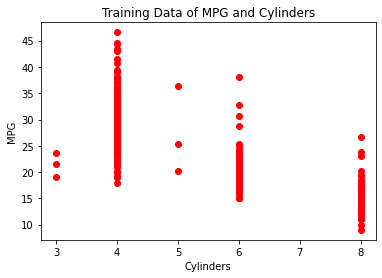
Date: 10/27/2023

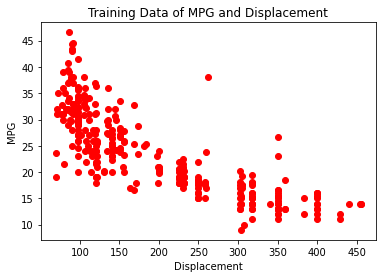
# Lab 4: Linear Regression and Linear SVMs

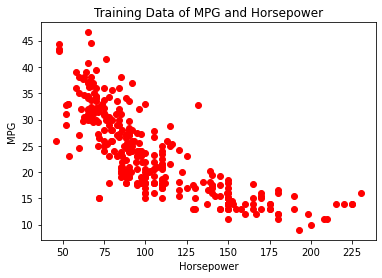
The fixed random state I use in my calculations is **42**.

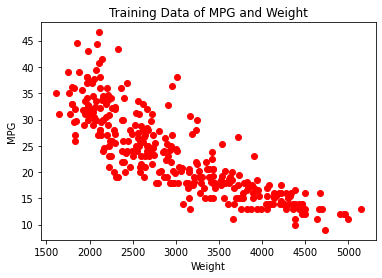
**Question 1.1 (60 points – 12 points for each feature): For each of the five features (cylinders, displacement, horsepower, weight, and acceleration), do the following:**

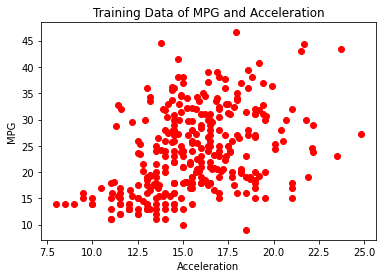
* Create an X matrix where each row is the current feature of interest.
* Do the train test split as indicated above. Create a plot that shows the current feature on the x-axis and the MPG on the y-axis for the training data.











* In the report, note whether you expect linear regression will perform well using that feature to predict the MPG value and why you believe it will perform well or not perform well.

I believe linear regression will perform better on data points that form a line like displacement, horsepower, and weight.

* Use sklearn’s LinearRegression() to fit to the training data.
* Create a prediction vector based on the training data, and calculate and print the mean-squared error and the R2 score using sklearn on the training set.

Cylinders

* Mean squared error on TRAIN: 24.655233534565706
* R2 score on TRAIN: 0.6092657121665905

Displacement

* Mean squared error on TRAIN: 21.538297693471062
* R2 score on TRAIN: 0.6586626770902894

Horsepower

* Mean squared error on TRAIN: 24.47516827368345
* R2 score on TRAIN: 0.6121193728863614

Weight

* Mean squared error on TRAIN: 19.04510619658238
* R2 score on TRAIN: 0.6981745885310532

Acceleration

* Mean squared error on TRAIN: 52.386427206105246
* R2 score on TRAIN: 0.16978383928843055
* Create a prediction vector based on the testing data, and calculate and print the mean-squared error and the R2 score using sklearn on the testing set.

Cylinders

* Mean squared error on TEST: 21.813981353532167
* R2 score on TEST: 0.5726149669760379

Displacement

* Mean squared error on TEST: 21.2274842896114
* R2 score on TEST: 0.5841057656051312

Horsepower

* Mean squared error on TEST: 22.153237123863413
* R2 score on TEST: 0.5659681822256185

Weight

* Mean squared error on TEST: 17.693388269545686
* R2 score on TEST: 0.6533466675646016

Acceleration

* Mean squared error on TEST: 40.073906749162774
* R2 score on TEST: 0.21486189605562878
* Print the coefficient and intercept parameters from the sklearn model for that feature.

Cylinders

* Coefficient: [-3.65215459]
* Intercept: 43.62203602314662

Displacement

* Coefficient: [-0.06222756]
* Intercept: 35.765941772604975

Horsepower

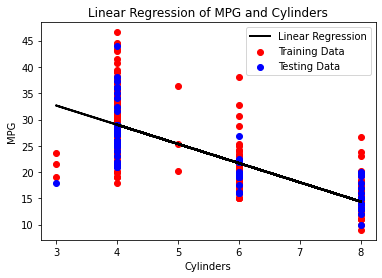
* Coefficient: [-0.16259724]
* Intercept: 40.606097600118346

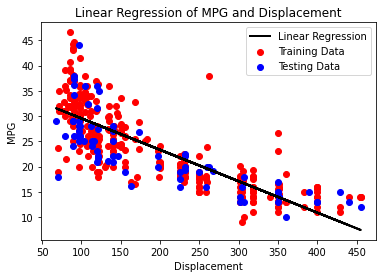
Weight

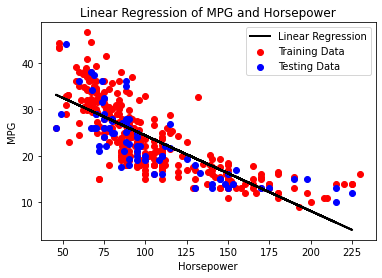
* Coefficient: [-0.00790361]
* Intercept: 47.200526427552106

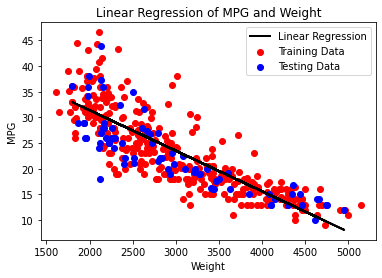
Acceleration

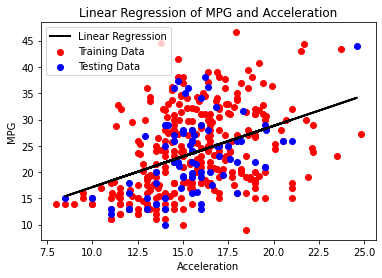
* Coefficient: [1.16342078]
* Intercept: 5.5150444023134995
* Plot the line created from that linear regression along with points for the training data (as one color) and the testing data (as another color). Include a legend to denote which are training and which are testing. Note that the x-axis should be the feature you’re predicting on and the y-axis should be MPG.











**In the report, note which of the features you believe is best for predicting MPG and why you selected that feature.**

I believe weight is the best feature for predicting MPG because it has the highest R2 score and lowest mean-squared error. Additionally, the data points most resemble a line which makes a better linear regression.

**Question 1.2 (10 points): Combining all of the data, do the following:**

* Create an X matrix that includes all five features.
* Do the train test split as indicated above.
* Use sklearn’s LinearRegression() to fit to the training data.
* Create a prediction vector based on the training data, and calculate and print the mean-squared error and the R2 score using sklearn.

Mean squared error on TRAIN: 17.87980377929501

R2 score on TRAIN: 0.7166422136497115

* Create a prediction vector based on the testing data, and calculate and print the mean-squared error and the R2 score using sklearn.

Mean squared error on TEST: 17.939171976631265

R2 score on TEST: 0.6485312110889081

**Does using all of the data improve performance over using each of the features individually?**

Yes, because the mean-squared error and R2 score of the combined data is around the same as just the weight’s mean-squared error and R2 score. Using all of the data gives better results than the rest of the features individually.

**Question 2.1 (25 points): Finding hyperparameters:**

* Create a validation set and subtraining set from the training set (a single validation set, rather than KFolds) with 0.125 of the training set serving as the validation set.
* Fit the LinearSVC classifier on the training set and evaluate on the validation set for all combinations of max iter=[1000, 10000, 100000, 1000000] and C=[0.01, 0.1, 1, 10, 100, 1000].
* Print the validation accuracy for each combination and show a heatmap comparing of the validation accuracies across the combinations (include a colorbar).

C: 0.01 max\_iter: 1000 Validation Accuracy: 0.20833333333333334

C: 0.1 max\_iter: 1000 Validation Accuracy: 0.20833333333333334

C: 1 max\_iter: 1000 Validation Accuracy: 0.20833333333333334

C: 10 max\_iter: 1000 Validation Accuracy: 0.20833333333333334

C: 100 max\_iter: 1000 Validation Accuracy: 0.20833333333333334

C: 1000 max\_iter: 1000 Validation Accuracy: 0.20833333333333334

C: 0.01 max\_iter: 10000 Validation Accuracy: 0.6666666666666666

C: 0.1 max\_iter: 10000 Validation Accuracy: 0.625

C: 1 max\_iter: 10000 Validation Accuracy: 0.5

C: 10 max\_iter: 10000 Validation Accuracy: 0.5

C: 100 max\_iter: 10000 Validation Accuracy: 0.5

C: 1000 max\_iter: 10000 Validation Accuracy: 0.5

C: 0.01 max\_iter: 100000 Validation Accuracy: 0.7083333333333334

C: 0.1 max\_iter: 100000 Validation Accuracy: 0.875

C: 1 max\_iter: 100000 Validation Accuracy: 0.5833333333333334

C: 10 max\_iter: 100000 Validation Accuracy: 0.7083333333333334

C: 100 max\_iter: 100000 Validation Accuracy: 0.5833333333333334

C: 1000 max\_iter: 100000 Validation Accuracy: 0.7916666666666666

C: 0.01 max\_iter: 1000000 Validation Accuracy: 0.9166666666666666

C: 0.1 max\_iter: 1000000 Validation Accuracy: 0.7083333333333334

C: 1 max\_iter: 1000000 Validation Accuracy: 0.6666666666666666

C: 10 max\_iter: 1000000 Validation Accuracy: 0.9166666666666666

C: 100 max\_iter: 1000000 Validation Accuracy: 0.8333333333333334

C: 1000 max\_iter: 1000000 Validation Accuracy: 0.9583333333333334



* Note the best parameter combination and discuss why those parameter values might be the best for this dataset (1-3 sentences).

The best parameter combination is C=1000 and max\_iter=1000000 because it has the highest validation accuracy score. It just happens that this data set needs to classify all training examples and run the most iterations to get the best results.

**Question 2.2 (5 points): Use the maximum iterations and C values you found to perform**

**best to inform how you create your LinearSVC() and fit on the whole training set. Compute and print the training and testing accuracy for the Stars dataset.**

Training Accuracy: 0.9166666666666666

Testing Accuracy: 0.8333333333333334