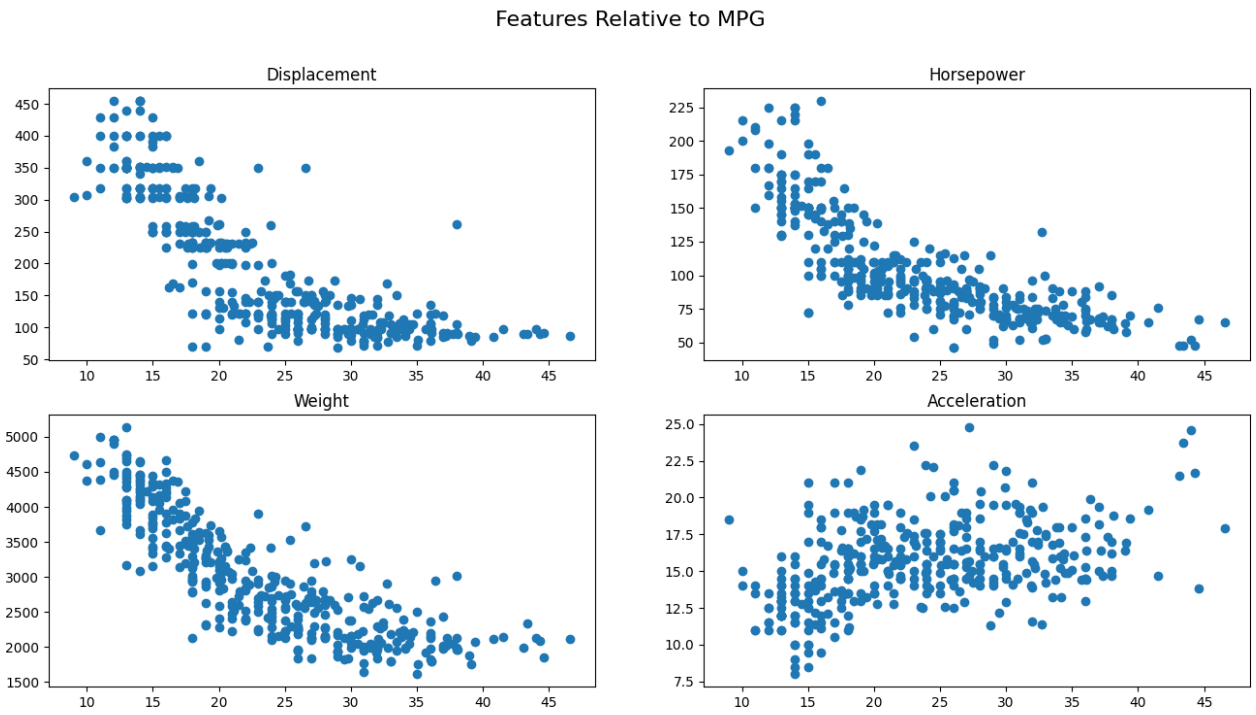
Christopher Gjersing
CS 422 1002 | Fall 2021
21 October 2021

Assignment 3: Predicting MPG with OLS and GD Linear Regression

The dataset I used for this assignment comes from the University of California, Irvine Machine Learning Repository. The name of the dataset is "Auto MPG" and lists information about cars. The attributes provided were MPG, cylinders, displacement, horsepower, weight, acceleration, model year, origin, and car name. Not all of these attributes are continuous real-valued, so I trimmed the dataset down to only MPG, displacement, horsepower, weight, and acceleration. Using these attributes, I decided to see if the linear regression algorithms could predict the MPG of a car when provided with the other 4 attributes.

Some data had question marks indicating a missing attribute and I removed those examples from the data, leaving me with 392 instances. Using pandas and numpy I removed the unused attributes (cylinders, model year, etc) and separated the X and y vectors into numpy arrays. I then added the bias vector column to the input. I use the LinearRegression() and SGDRegressor() models from the scikit library for this assignment. Both of these algorithms can estimate the bias for you, so I ran the tests with fit_intercept being estimated by the model as well as with the bias parameter vector. The difference is minimal, likely attributed to the differences that occur between regressions, but I wanted to test it and decided to include it. The following graph shows the inputs relative to the MPG output (x-axis).

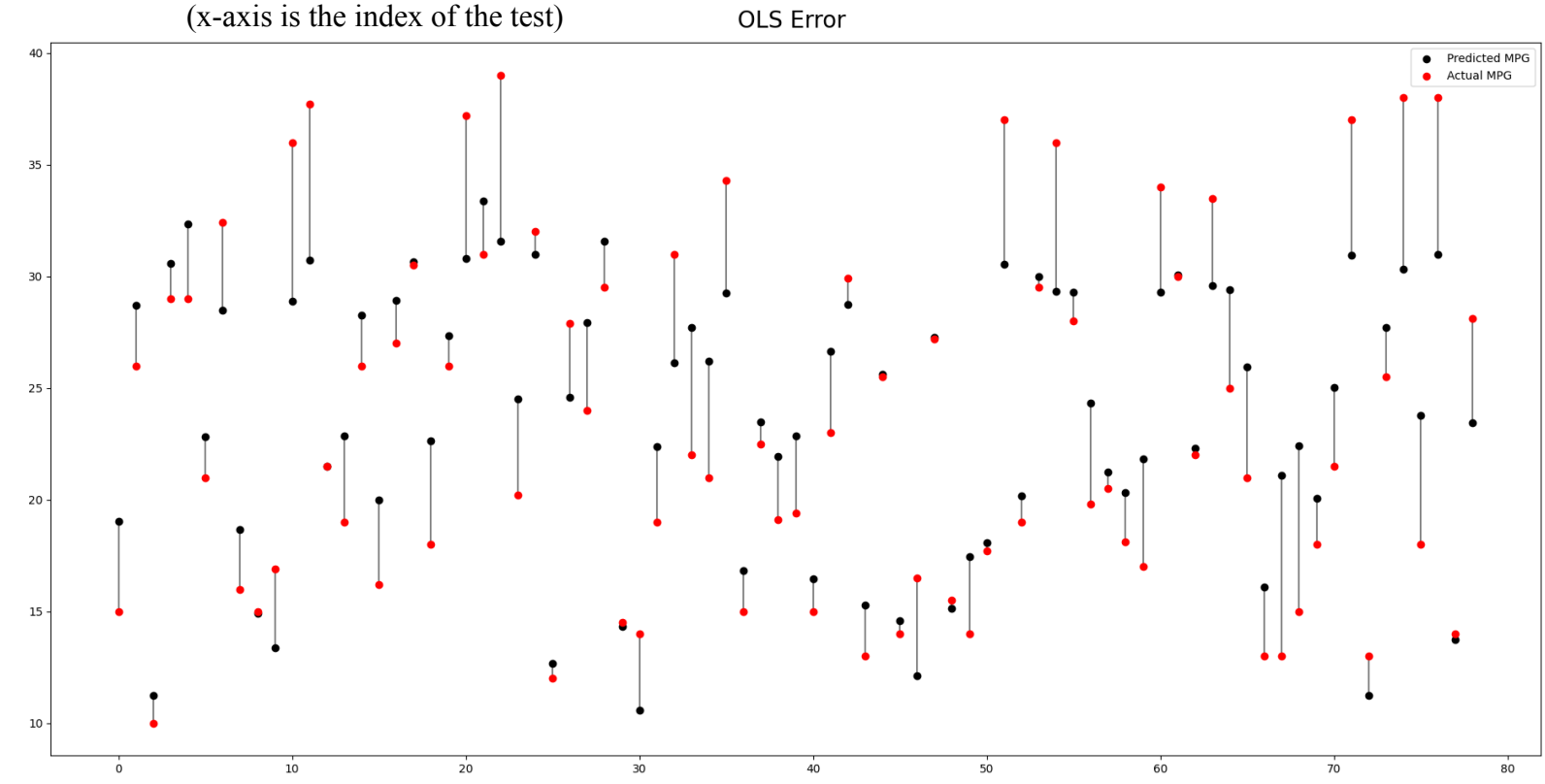


OLS Algorithm

Fit_intercept = true | No bias vector (Bias provided from model)	Fit_intercept = false | Bias vector (Bias is 5th coefficient value)
Bias (Intercept): 46.136221723954314	Bias (Intercept): 43.5711762
Coefficients (w): [-0.00866564 -0.04765974 -0.00486358 -0.10340465]	Coefficients (w): [-0.00327284 -0.04174672 -.00552811 0.08112463]
Training MAE: 3.394470529703743	Training MAE: 3.304547445395422
Training MSE: 19.08542888498837	Training MSE: 18.99619523993138
Training R2: 0.6887625612526174	Training R2: 0.6903223432438201
Test MAE: 2.5175595114339506	Test MAE: 2.953230263596278
Test MSE: 12.875002201342847	Test MSE: 13.331713762906903
Test R2: 0.7800523893401352	Test R2: 0.7704288834896648

The following graph shows the Predicted MPG relative to the Actual MPG for the OLS test.

(x-axis is the index of the test)



Gradient Descent Algorithm

For gradient descent I used SGDRegressor from scikit with the ordinary least squares fit loss function. Initially, regardless of used learning rate, I had very poor results with ridiculous values and r2 scores until I used the StandardScaler to remove the mean and scale the data to unit variance. This additional preprocessing greatly improved the algorithm results, making them similar to my OLS findings. With only 392 instances of data, with features ranging from double digit to quadruple digit values, my data was too small and too varied to be used without scaling. The documentation for SGDRegressor states “Stochastic Gradient Descent is sensitive to feature scaling, so it is highly recommended to scale your data.”

Learning Rate: **0.01**
Conv. Iterations: 1246 (min: 400 | max: 5000)
Bias (Intercept): 23.47684609
Coefficients (w): [-0.91069536 -0.98901462 -4.74898994 0.15684471]
Training MAE: 3.2849969477843963
Training MSE: 18.57403613265497
Training R2: 0.6952449496893545
Test MAE: 3.069731903332755
Test MSE: 15.035255737369045
Test R2: 0.7478824360163525

Learning Rate: **0.005**
Conv. Iterations: 1196 (min: 400 | max: 5000)
Bias (Intercept): 23.50957305
Coefficients (w): [-0.23909815 -1.68650597 -4.89034613 -0.21845068]
Training MAE: 3.274518486515746
Training MSE: 18.29526387631361
Training R2: 0.7060449479785977
Test MAE: 3.0813188223761974
Test MSE: 16.159646355659604
Test R2: 0.7056732072745042

I’m not a car expert or mechanic, but based on the feature graphs I’d assume that displacement, horsepower, weight, and acceleration have only some influence over the MPG of a car. Using OLS and GD linear regression solutions, it seems that machine learning can help us comprehend that association. The algorithms give decent predictions to the MPG of the car, with most estimates being within 5 miles per gallon. For future assignments and machine learning implementations, I will probably try to find a data set with more instances that has a better correlation between the input features and the output.