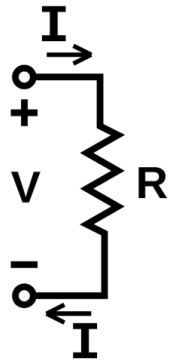
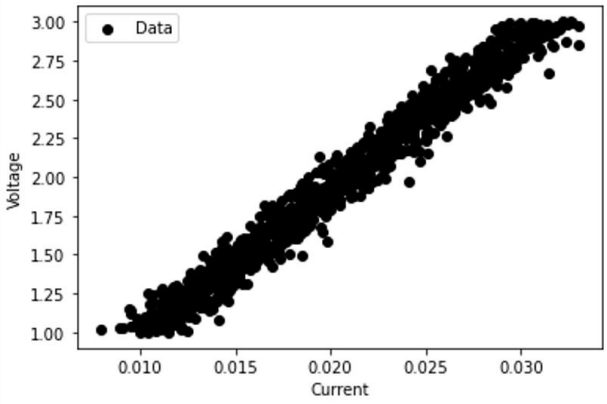
Homework #1: *Linear Regression and Classification*

Assigned: 01/25/2021
Due: 02/08/2021 (11:59 PM on CANVAS)

1. **Ohm's Law with Linear Regression:** In this problem, you will determine the resistance (R) or resistor based on noisy current measurement data. Consider an experiment where the applied voltage (V) in Volts and measured current (I) in Amps are measured across a resistor as shown in the figure below.





It is known from Ohm's law that the relationship between current and voltage in a resistor is given by $V = IR$.

a) Using the data file on CANVAS 'OhmsLaw.csv', estimate the resistance R. Make sure to include your code with your submission. **Hint you can turn off the intercept term on linear regression using with the regression object: linear_model.LinearRegression(fit_intercept=False)*

b) Make a plot of the regression line along with a scatterplot of the original data.

2. **Miles Per Gallon Prediction using Multi-Linear Regression:** In this problem, you will build a linear model to predict the miles per gallon of a vehicle using four features: number of cylinders, weight, 0-60 acceleration, and model year.

a) Compute the standard deviation (SD) in miles per gallon of the training data. *Note* The SD is a measure of the spread of your data around it's average, you can use numpy.std*

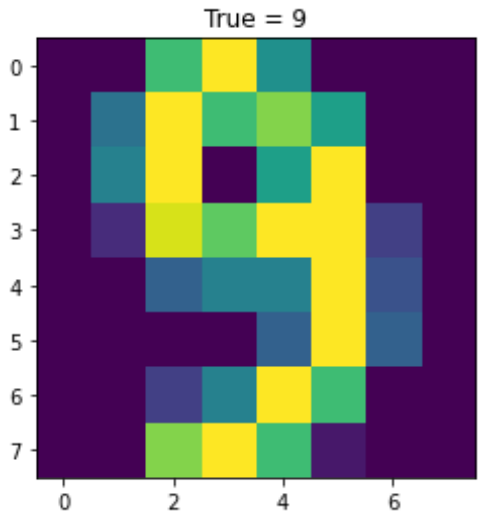
b) Using the data file 'auto_dataset_train.csv' (on CANVAS), create a multi-linear regression model that predicts miles per gallon based on the above mentioned features. State the root mean square error ($RMSE$) of the training data.

c) Compare the $RMSE$ to the SD, what does this imply about the linear model?

d) Using the data file 'auto_dataset_test.csv', test your model on the new data set. State the root mean square error ($RMSE_{test}$) of the model on the test data. How does this compare to the training $RMSE$?

e) Discuss some improvements or changes you would make to this model.

3. **Digit Classification:** For this problem, you will used a Perceptron classifier to detect the number of a handwritten digit between 0 and 9. An example digit (sampled on an 8x8 pixel grid) looks like the following:



The testing data arrays for this problem contain 1257 labeled cases and are given in the Python .npy files as 'X_Train.npy' and 'y_train.npy' (on CANVAS). The X_train array has size 1257x64. Each row of the array corresponds to one image (as the example above), however, the 8x8 grid of pixels has been reordered to 64x1. *Note* you can view the n-th training example using matplotlib.pyplot.imshow(X_train[n,:].reshape([8,8]))*. The array y_train contains the 1257 integers between 0 and 9 which correspond accordingly to each training example.

a) Build a perceptron model using the training data. Show images of the 5^{th}, 111^{th}, 584^{th} and, 1200^{th} training examples along with the true label and predicted label.

b) Determine the accuracy of the model on the training set using the .score method.

c) Test the model from part (a) on test data given in 'X_test.npy' and 'Y_test.npy'. Show images of the 19^{th} , 48^{th}, 336^{th} , and 529^{th} testing examples along with the true label and predicted label.

d) Determine the accuracy of the model on the test set using the .score method. How does this compare to the training data accuracy?

e) Discuss any changes you would make to the model.