CST 383 - Intro to Data Science

Dr. Glenn Bruns

# Lab: Linear regression, part 1

In this lab we will start performing linear regression in Python.

1. Read the CPU performance data from this csv file into a Pandas DataFrame named 'df'.

<https://raw.githubusercontent.com/grbruns/cst383/master/machine.csv>

You may want to add the 'clock speed' feature, as shown in the hints section below.

1. Look at the information on the following page to learn about what the column names mean. ‘prp’ is the variable that we will try to estimate with linear regression.

<https://archive.ics.uci.edu/ml/datasets/Computer+Hardware>

Note that this data is from way back in 1988 and has little to do with today’s CPUs.

1. Create a matrix of scatter plots using the Seaborn pairplot function. Do some features appear to be strongly correlated?
2. Pick a feature (like ‘mmin’, ‘mmax’, ‘cach’, etc.) and produce a scatterplot with that feature on the x axis and ‘prp’ on the y axis. Does prp seem to change linearly with respect to the feature you chose? Describe the pattern: what is the trend, how focused is the pattern, are there outliers, etc.?
3. Fit a linear model to the data using sklearn.linear\_model.LinearRegression. The model should have ‘prp’ as its target variable, and the feature you chose as its single predictor variable. Assign the model to variable ‘fit’.
4. Plot the predicted relationship on top of your plot of problem 4 above. Compare your plot to the one you would get with Seaborn's regplot (which will find and plot the regression line for you).
5. Create a scatterplot with the actual 'prp' value on the X axis and the predicted 'prp' value on the Y axis. If your predictions were great, what would this plot look like?
6. If you still have time, try again, using different features as your predictors. Try multiple predictors. In each case, plot the actual vs. predicted values, as in the last problem.

## Hints:

1. df = pd.read\_csv("https://raw.githubusercontent.com/grbruns/cst383/master/machine.csv")

df.index = df['vendor']+' '+df['model']

df.drop(['vendor', 'model'], axis=1, inplace=True)

df['cs'] = np.round(1e3/df['myct'], 2) # clock speed in MHz (millions of cycles/sec)

1. -
2. -
3. You could use Seaborn's scatterplot() function for this.
4. -
5. You could use sns.scatterplot() followed by plt.plot() to superimpose a line over the scatterplot.
6. -
7. -