CST 383 - Intro to Data Science

Dr. Glenn Bruns

# Lab: Linear regression, tuning models

In this lab we will continue to perform linear regression using the CPU data set.

Before starting the lab, you may want to download 'An Introduction to Statistical Learning' by James et al, which is made available by the authors for free as a PDF. It uses R, not Python, but is a great resource for data science:

<https://faculty.marshall.usc.edu/gareth-james/ISL/>

Now let's start the lab. Create a new Python file and write code for the following steps.

1. Read the CPU data:

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

1. Split the data randomly into a training set and a test set, using a 70/30 split (70% training data). Use [train\_test\_split](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html).
2. Use [LinearRegression](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html) to create a linear model to predict performance (feature ‘prp’). Use a couple of predictor variables of your own choice. Create the model using your training set.
3. Compute the MSE of your model on the test data. Do this manually.
4. Repeat steps 2-4, but this time use new randomly-generated test and training sets. How much does the RMSE differ?
5. If you have time, write code that will do steps 2-4 100 times, each time creating different training/test sets. Collect the computed RMSE values, and plot them on a histogram.
6. If you still have time, repeat problem 6, but this time use an 80/20 split.
7. If you still have time, compute MSE using cross validation on the entire data set. Do this many times and plot all the histogram values using a histogram.
8. If you still have time, check out Section 5.1 of 'An Introduction to Statistical Learning'.