ECE 562 Machine Learning, Fall 2019
Instructor: Zafer Aydın
Computer Homework 5

## Introduction

Download the template codes for Udacity's Intro to Machine Learning course by running the following command in a terminal

git clone https://github.com/udacity/ud120-projects.git

Sebastian described to us an algorithm for improving a regression, which you will implement in this project. You will work through it in the next few quizzes. To summarize, what you'll do is fit the regression on all training points discard the 10% of points that have the largest errors between the actual y values, and the regression-predicted y values refit on the remaining points.

Submit your scripts as q1.py and/or a text document that includes your answers to the questions below through Canvas.

## Assignment

1. Start by running the starter code (outliers/outlier_removal_regression.py) and visualizing the points. A few outliers should clearly pop out. Deploy a linear regression, where net worth is the target and the feature being used to predict it is a person's age (remember to train on the training data!). Include the figure in your report.

Fit a linear regression model to the train set. The "correct" slope for the main body of data points is 6.25 (we know this because we used this value to generate the data); what slope does your regression have?

2. What is the R2 score you get when using your regression to make predictions with the test data?

3. In outliers/outlier_cleaner.py, you will find the skeleton for a function called outlierCleaner() that you will fill in with a cleaning algorithm. It takes three arguments: predictions is a list of predicted targets that come from your regression, ages is the list of ages in the training set, and net_worths is the actual value of the net worths in the training set. There should be 90 elements in each of these lists (because the training set has 90 points in it). Your job is to return a list called cleaned_data that has only 81 elements in it, which are the 81 training points where the predictions and the actual values (net_worths) have the smallest errors (90 * 0.9 = 81). The format of cleaned_data should be a list of tuples, where each tuple has the form (age, net_worth, error).

Include the new scatter plot figure into your report that shows the data samples as well as the line fitted by linear regression model.

Once this cleaning function is working, you should see the regression result changes. What is the new slope? Is it closer to the "correct" result of 6.25?

4. What's the new R2 score when you use the regression to make predictions on the test set?

5. In the mini-project for the regressions lesson, you used a regression to predict the bonuses for Enron employees. As you saw, even a single outlier can make a big difference on the regression result. There was something we didn't tell you, though, which was that the dataset we had you use in that project had already been cleaned of some significant outliers. Identifying and cleaning away outliers is something you should always think about when looking at a dataset for the first time, and now you'll get some hands-on experience with the Enron data.

You can find the starter code in outliers/enron_outliers.py, which reads in the data (in dictionary form) and converts it into a sklearn-ready numpy array. Since there are two features being extracted from the dictionary ("salary" and "bonus"), the resulting numpy array will be of dimension N x 2, where N is the number of data points and 2 is the number of features. This is perfect input for a scatterplot; we'll use the matplotlib.pyplot module to make that plot. (We've been using pyplot for all the visualizations in this course.) Add these lines to the bottom of the script to make your scatterplot:

```
for point in data:
    salary = point[0]
    bonus = point[1]
    matplotlib.pyplot.scatter( salary, bonus )

matplotlib.pyplot.xlabel("salary")
matplotlib.pyplot.ylabel("bonus")
matplotlib.pyplot.show()
```

Include the figure you obtained to your report.

6. There's one outlier that should pop out to you immediately. Now the question is to identify the source. We found the original data source to be very helpful for this identification; you can find that PDF in final_project/enron61702insiderpay.pdf

What's the name of the dictionary key of this data point? (e.g. if this is Ken Lay, the answer would be "LAY KENNETH L").

Does this outlier seem like a data point that we should include when running machine learning on this dataset? Or should we remove it?

7. A quick way to remove a key-value pair from a dictionary is the following line: dictionary.pop( key, 0 ) Write a line like this (you'll have to modify the dictionary and key names, of course) and remove the outlier before calling featureFormat(). Now rerun the code, so your scatterplot doesn't have this outlier anymore. Are all the outliers gone?

8. We would argue that there's 4 more outliers to investigate; let's look at a couple of them. Two people made bonuses of at least 5 million dollars, and a salary of over 1 million dollars; in other words, they made out like bandits. What are the names associated with those points?