CSD3185 Week 3: Linear Regression

Tasks:

1. Get the file Gamer.csv from the LMS. It contains 1000 cases that you will use to build a regression model. There is a header line that you don't really need except to identify the column that you will predict: **ActionLatency.**

2. Your primary task is to simulate a real-word scenario where a model is created based on labeled data and then applied to NEW UNLABELED data. For the purposes of this lab, you will do this in one step and submit a SINGLE python file. Let's say this file is mypredict.py. You should be able to run the program on the command line to get the output as shown below (the output numbers are illustrative, your model may produce different ones!):

   python3 mypredict.py Gamer.csv NewUnlabeledData.csv

   22.54

   31.65

   51.4

3. A sample NewUnlabeledData.csv is also provided with 3 cases. Note that it contains a header which is almost identical to that of Gamer – just missing the column ActionLatency (which is to be predicted). You can easily create more cases for your testing by simply using lines from Gamer.csv (and removing the ActionLatency entry, of course!)

4. You must use the classes/methods in scikit-learn, which are all installed in your anaconda system. Check out LinearRegression and make use of the fit and predict methods in that class.

5. To test your models, use the score method. Note that the scoring is by the coefficient of determination, which was discussed in the lectures.

6. The lab is worth 3 marks. An attempt (even if with errors) in which you are trying to use the correct methods gets you 1 mark. Building a regression model, and using it to make predictions, without computational errors, gets you another mark. The 3rd mark is for the top 10% students -- those who give the 'best' predictions. This will be assessed by running all programs on a set of new cases for which I know the actual answers, computing

the R^2 value for each submission on these cases, and taking the top 10%.

7. OPTIONAL HINTS: For those students looking to try more advanced things (do this ONLY after you have a basic version submitted!!) you might want to consider the input attributes carefully. Some questions to ask is whether you need all of them to get the best answer. Another issue to consider is whether the assumption of ordinality is fully justified for each. For most it is obvious, but I can tell you that the first one, LeagueIndex, is actually a categorical attribute, even though it is a number. So it is possible you can be better results by treating it as such. A more advanced issue is whether the data could be smoothed (these may require some data transformations!)

ONE SUBMISSION PER student, to the dropbox in the LMS. Submit by 2pm, 1st Feb, Thursday (before the start of week-4 class).