**CS 386**

**Machine Learning for Watson**

**Project**

**Evaluation harness and final report**

**Due in stages**

**Part One due Saturday, Feb 22, midnight**

**Part Two due Sunday, March 2, midnight**

**Report due Saturday, March 8, midnight**

**Summary**

In the next class we will talk about evaluating different models /classifiers using just your labeled training data.

The goal of this assignment is to give you a better understanding of how to evaluate the classifiers that you are fitting to the IBM data, and perhaps climb the leaderboard as a result. We have created a skeleton for you to fill in for this project (there will be another skeleton for Part Two). Like the other homework assignments, this project can be done with your team. Note that if you run the code with the same options multiple times, you may get slightly different answers.

The bottom of this document tells you what to turn in – you will be answering some questions and writing some code.

**Instructions for Part Two and the report will be available soon.**

**Instructions for Part One:**

1. Read the section on [cross-validation in sklearn](http://scikit-learn.org/stable/modules/cross_validation.html), stopping when you get to section 3.1.2.4
2. Make sure that all the files in the zip folder for this homework are in the same directory, and in the steps below, run the program within that directory (it looks for the file subset\_train.pkl)
3. I have provided a program, eval\_ibm.py, that runs 5-fold cross-validation and 5-fold stratified cross-validation on a portion of the IBM training data (subset\_train.pkl). It learns decision trees for each fold of the data. It also uses methods from sklearn to create the cross-validation folds. It shows the average precision and recall scores (and standard deviation) for both types of cross-validation. Run this to see what average P/R scores you get with a DecisionTreeClassifier:

% python eval\_ibm

* **What precision and recall did you get for cross-validation?**
* **What precision and recall did you get for stratified cross-validation?**

1. Next, you will fill in parts of your own evaluation program to recreate and add to the results you just got, and also allow the user to specify which classifier to use to fit the data. First, make a backup copy of eval\_ibm so you can track what you’ve changed.
2. Using the first homework as a model, add the capability to eval\_ibm for the user to optionally specify {–clf <name of classifier>} when they run it from the command line. Add both KNeighborsClassifierwith K=5 (--clf KNN) and naive\_bayes.GaussianNB (--clf NaiveBayes) as options. Now run the program two more times with each of these options.

* **What precision and recall did you get for cross-validation for KNN?**
* **What precision and recall did you get for stratified cross-validation for KNN?**
* **What precision and recall did you get for cross-validation for NaiveBayes?**
* **What precision and recall did you get for stratified cross-validation for NaiveBayes?**

1. The other additions to eval\_ibm will be done in Part Two.
2. Using whatever approach your team devises for trying to improve your score, including anything you learned in running the above experiments, make another contest submission on the IBM contest site.
   * **What did you do to try to improve your score?**
   * **What score did you get?**

**What To Turn In**

Turn in two files on Canvas (again, only one member of each team needs to turn in these files):

1. answers.txt, the answers to all questions asked above (in bold font). Provide the number of the question followed by the answer to the question
2. eval\_ibm.py, your modified version of the code