NAME:
COLLABORATOR(S): *You are not allowed to have any collaborators.*

CS480 – Project Phase 2
Assigned on: Wednesday, 10/29/2014
Due: Sunday, 11/9/2014, 11:59pm
Please submit your solutions through black board assignment page.

In this phase, unlike phase 1, you are not provided the probability of being Good. Rather, your agent needs to learn a function/classifier that can predict the probability of a product being Good, given the product's features. Your agent is provided 12 dataset groups, each group having three datasets:

- a training dataset that has products along with their features (X_train), and whether each product is in Good or Bad condition (y_train),
- a validation dataset again that has products along with their features (X_val), and whether each product is in Good or Bad condition (y_val),
- and a test dataset where only products and features (X_test) are given and your agent needs to predict the probabilities for being Good or Bad.

In this phase, you will use the scikit-learn library's classifiers.

You are provided three files.

- agents.py
    - This file has the base class Agent, a baseline agent RatioAgent, and an example agent, called the NaiveBayesAgent. Make sure that you get the this version of the agents file, which is a bit different from Phase 1.
- simulate_agents_phase2.py
    - This file simulates various agents.
- datasets.zip
    - This file has 12 groups of datasets, where each one is denoted as datasetN, where N ranges from 1 to 12. Each datasetN has 5 files; X_train, y_train, X_val, y_val, and X_test.

Your task is to create an agent called Agent_<hawk_username> where <hawk_username> is your Hawk username. Specifically:

1. Create an agent_<hawk_username>.py file.
2. In that file, define an Agent_<hawk_username> class, which inherits the base Agent class.
3. Implement the fit_a_classifier method. This method should fit the best classifier, given the X_train, y_train, X_val, and y_val. You should fit one of scikit-learn's classifiers. Make sure that the classifier is enabled to predict probabilities. I will walk over the details in class.

4. Your agent should not override the constructor (__init__ method). We'll run your code with the default constructor of the Agent class.
5. In the simulate_agents_phase2.py file, add your agent to the list of other agents.
6. Run the simulation and compare your agent's performance to the performance of other agents for each of the dataset groups.

Submit 14 files:

1. agent_<hawk_username>.py file.
2. A short report in .txt format. Name this file as report.txt. The report should have a brief description of your strategy to find the best classifier, given the training and validation sets.
3. 12 simulation files in .txt format. Name each of the files as simulation_datasetN.txt. This is the output from the simulate_agents_phase2.py file after you add your agent to the list. Submit a simulation file per dataset group.