CS480 – Project Phase 3
Assigned on: Monday, 11/21/2015
Due: Sunday, 12/06/2015, 11:59pm

Report Summary

The Phase 3 of the project tells us to implement scikit-learn's three classifier.

Solution:

- Train the classifier's individually using the training data.
- Use the validation data set to find how well the validation data works on the classifier.
- Find the wealth that can be earned using the classifier.
- The classifier with the highest wealth is considered the best classifier for that data set.
- The classifier chosen as the best performer is then returned to the main function fo further evaluation.

class Agent_sguhatha(Agent)

Agent_sguhatha is my agent, which needs to to use the three classifiers internally as three separate agents and then decide on which classifier is performing the best and then return the same to the main train function.

def choose_the_best_classifier(self, X_train, y_train, X_val, y_val):

This is the main function that is called from the Agent class to get the classifier objet from the respective agent.

- First I built a list have the names of the three classifiers.
- Here I call the newly built function recursively to calculate the performance of the individual classifier.
- The return value from the classifiers are stored in a array and then I use it to compare and store the information of the classifier with the maximum wealth.
- Finally we return the object of the classifier of the highest value of wealth.

def best(classifier, X_train, y_train, X_val, y_val,value ,price_trials):

This function is the one newly built. This contains the functionality of all the calculations pertaining to training, evaluating and then calculating the total wealth of the classifier.

- First I take the input from the classifier list and decide as to evaluate based on which classifier.
- Now I pass the training dataset in the classifier to train it.
- As per the training result we mark index based on the product being Excellent or Trash.
- Finally I run the evaluation to find how the classifier performs on the validation data set.

- Internally we also find out the total wealth earned by the classifier and return the wealth.

Notes:

1. To calculate the wealth based on the evaluation done by the classifier I used similar logic which is used my the simulate_agent.py .

2. In the course of running the program I was getting a variety of Deprecation Warnings, to suppress them I used the below to lines of code during execution:

   ```
   import warnings
   warnings.filterwarnings("ignore")
   ```

OUTPUT

SIMULATION RESULTS ON dataset1

Wealth (the larger the better)

Agent_bnb:	$1,775,950.00
Agent_lr:	$1,638,100.00
Agent_svc:	$1,593,950.00
Agent_sguhatha:	$1,775,950.00

Log-loss (the smaller the better)

Agent_bnb:	332.25
Agent_lr:	373.89
Agent_svc:	410.88
Agent_sguhatha:	332.25

0/1 Loss (the smaller the better)

Agent_bnb:	88
Agent_lr:	115
Agent_svc:	151
Agent_sguhatha:	88

Summary:

- As we see from the default constructors the best performing agent is Agent_bnb.
- With respect to my agent's wealth and Agent_bnb It can be found out that both have the same value.
- Conclusion that the classifier object returned from Agent_sguhatha is same as Agent_bnb, which is "BernoulliNB" as expected from the agent.

SIMULATION RESULTS ON dataset2

Wealth (the larger the better)

Agent_bnb:	$1,507,950.00
Agent_lr:	$1,717,100.00
Agent_svc:	$1,539,850.00
Agent_sguhatha:	$1,717,100.00

Log-loss (the smaller the better)

Agent_bnb:	553.13
Agent_lr:	487.71
Agent_svc:	564.30
Agent_sguhatha:	487.71

0/1 Loss (the smaller the better)

Agent_bnb:	250
Agent_lr:	223
Agent_svc:	294
Agent_sguhatha:	223

Summary:

- As we see from the default constructors the best performing agent is Agent_lr.
- With respect to my agent's wealth and Agent_lr It can be found out that both have the same value.
- Conclusion that the classifier object returned from Agent_sguhatha is same as Agent_lr, which is "LogisticRegression" as expected from the agent.

SIMULATION RESULTS ON dataset3

Wealth (the larger the better)

Agent_bnb:	$795,950.00
Agent_lr:	$810,100.00
Agent_svc:	$1,106,800.00
Agent_sguhatha:	$1,107,950.00

Log-loss (the smaller the better)

Agent_bnb:	571.94
Agent_lr:	566.91
Agent_svc:	411.62
Agent_sguhatha:	410.22

0/1 Loss (the smaller the better)

Agent_bnb:	250
Agent_lr:	255
Agent_svc:	160
Agent_sguhatha:	160

Summary:

- As we see from the default constructors the best performing agent is Agent_svc.
- In this case my agent actually over performs the default agent Agent_svc.
- Conclusion that the classifier object returned from Agent_sguhatha is either similar or better than Agent_svc, which is “SVC” as expected from the agent.