

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

<b>REPORT SUMMARY</b>	<b>2</b>
<b>CLASS AGENT_SGUHATHA(AGENT)</b>	<b>2</b>
def choose_the_best_classifier(self, X_train, y_train, X_val, y_val):	2
def best(classifier, X_train, y_train, X_val, y_val,value ,price_trials):	2
<b>NOTES:</b>	<b>3</b>
<b>OUTPUT</b>	<b>4</b>
SIMULATION RESULTS ON dataset1	4
SIMULATION RESULTS ON dataset2	5
SIMULATION RESULTS ON dataset3	6

## 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 for further evaluation.

### **class Agent\_sguhatha(Agent)**

Agent\_sguhatha is my agent, which needs 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 object 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 in the `simulate_agent.py`.
2. To resolve the randomness of the wealth amount in SVC classifier I used the fourth parameter as suggested by professor `random_state=0`.

## 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:	\$1,111,000.00
Agent_sguhatha:	\$1,111,000.00

Log-loss (the smaller the better)

Agent_bnb:	571.94
Agent_lr:	566.91
Agent_svc:	408.13
Agent_sguhatha:	408.13

0/1 Loss (the smaller the better)

Agent_bnb:	250
Agent_lr:	255
Agent_svc:	161
Agent_sguhatha:	161

### Summary:

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