

Solution: To compare the classifiers, wealth has been used as performance parameter as the wealth measure always trumps the logloss, 0/1 loss, and other performance measures. Classifier for a particular data set has been selected basis on the fact how they have performed for wealth for the validation data set after training from the training data.

The one which helps to maximize the wealth will be used for that particular set of test data.

Implementation detail: The implementation details can be explained as follows

1. Consider there is an agent for each of the three classifiers.
2. Assign 0 wealth to each of these agents. `agent_wealths= {}` is the array which holds the wealth of each agent.
3. Train each of classifier with the training data i.e. features of all the products and their quality.

To train the classifiers,

a. Firstly, object of each classifier class has been created.

b. Then their train method has been called for training with the help of their objects.

c. Lastly in the for loop, `predict_proba` method of each has been called with the help of their object.

Which takes a vector of feature as input to find the probability of quality of a product being Excellent

or Trash having these feature vector.

4. Run each of the classifier on the validation set after training and calculate the wealth of each agent.

5. Compare the wealth of classifiers.

6. The classifiers which helps it's agent to gain maximum wealth will be the best one for that data set.

7. Send the new object of the classifier to the tra

in method.

8. In the train method, classifier is run on the training data set.

Result: The agent Agent_bsharma, since uses the best classifier for a particular data set after being validated, hence is supposed to behave the best for testing data. Which is evident from the output. In the output, Agent_bsharma have the same performance as the bestline agent per each datasheet.

SIMULATION RESULTS ON dataset1

Wealth (the larger the better)

Agent_bnb:	\$1,775,950.00
Agent_lr:	\$1,638,100.00
Agent_svc:	\$1,596,800.00
Agent_bsharma:	\$1,775,950.00

Log-loss (the smaller the better)

Agent_bnb:	332.25
Agent_lr:	373.89
Agent_svc:	408.75
Agent_bsharma:	332.25

0/1 Loss (the smaller the better)

Agent_bnb:	88
Agent_lr:	115
Agent_svc:	147
Agent_bsharma:	88

SIMULATION RESULTS ON dataset2

Wealth (the larger the better)

Agent_bnb:	\$1,507,950.00
Agent_lr:	\$1,717,100.00
Agent_svc:	\$1,540,800.00
Agent_bsharma:	\$1,717,100.00

Log-loss (the smaller the better)

Agent_bnb:	553.13
Agent_lr:	487.71
Agent_svc:	564.25
Agent_bsharma:	487.71

0/1 Loss (the smaller the better)

Agent_bnb:	250
Agent_lr:	223
Agent_svc:	293
Agent_bsharma:	223

SIMULATION RESULTS ON dataset3

Wealth (the larger the better)

Agent_bnb:	\$795,950.00
Agent_lr:	\$810,100.00
Agent_svc:	\$1,102,800.00
Agent_bsharma:	\$1,102,800.00

Log-loss (the smaller the better)

Agent_bnb:	571.94
Agent_lr:	566.91
Agent_svc:	413.10
Agent_bsharma:	413.10

0/1 Loss (the smaller the better)

Agent_bnb:	250
Agent_lr:	255
Agent_svc:	165

Agent_bsharma: 165