Data Mining

Homework #7 (The last)

주의: 이 과제는 기말 프로젝트 성적의 50%에 해당함. 1번과 2번은 필수이고, 3번 문제는 도전 과제. 과제 발표 당일에 질문 #2 와 #3 부분에 대해서 발표.

Submisssion due: Dec. 10(Sunday)

Fill out you codes in the Jupyter Notebook (2 files) included !!!

Question #1: **Learning a classifier for the "Imbalanced Iris" Data Set – Part II**

Try a Decision Tree based approach with 10-fold cross-validation.

1. Discretize the Iris data set into three bins. Then use the DecisionTreeClassifier with a 10-fold stratified cross validation and compute the accuracy. Afterwards plot the decision tree.
2. Remove the discretization and adjust the max_depth parameter of DecisionTreeClassifier to increase the accuracy.
   Does the accuracy change? Compare the complexity of the two models.

Question #2: **Who should get a bank credit?**

German Credit data set from UCI data set library describes the customers of a bank with respect to whether they should get a bank credit (대출과 같은 은행신용) or not.
https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

1. Plot ROC curves for k-NN (different k values), Decision Tree and Naïve Bayes classification (you can use the given **avg_roc** function). Which classification approach looks most promising to you?
2. For the two most promising classification approaches, compute the accuracy and confusion matrix in a 10-fold cross-validation setup (use **cross_val_predict** function). Which level of accuracy do you reach?
3. What do the precision and recall values for the class "bad" customer tell you? Try to improve the situation by **increasing the number of "bad" customers in the training set** (in the cross-validation!). How do precision and recall change if you apply this procedure?
4. To **model a use-case specific evaluation**, compute the cost of all misclassifications. Set up your cost matrix by assuming that you will lose 1 unit if you refuse a credit to a good customer, but that you lose 100 units if you give a bad customer a credit. (That is, confusion matrix= ((0,100) (1,0)) Re-run the experiments from step 1, step 2 and step 3 and evaluate the results.

Question #3: 도전!!! (**Parameter Tuning**) Who should get a bank credit?

1. What were the default parameters of the Decision Tree algorithm used in Question #2?
2. Now try to find a more appropriate configuration for the Decision Tree classifier. Use the ***GridSearchCV*** from scikit-learn. Try the following parameters of the Decision Tree:
    - criterion: ['gini', 'entropy']
    - 'max_depth': [1, 2, 3, 4, 5, None] (What does None mean?)
    - 'min_samples_split': [2,3,4,5]

    You should come up with 48 (2 x 6 x 4) combinations.
    What is the best configuration for the data set and the classification approach?
3. What is the cost of misclassification for this configuration?
4. How does the optimal decision tree differ from the one you have learned in Question 2.4

Submit: Jupyter Notebook runned with comments.