Northeastern University, College of Computer and Information Science

CS 6220 Data Mining
Final Exam

Due Sunday December 8th, 2019 at 11:59pm PST.

Name (print): Yuanjie Yue Signature: Yuanjie Yue

Note:

1. This is an individual exam. Group work is not permitted.
2. This exam contains 10 pages (feel free to expand the number of pages if needed)
3. This exam is open book and open notes but you may not use on-line resources.
4. Write your answers on this document itself for submission.
5. This exam is graded out of 100 points.
6. Do as much as you can; partial credit will be given for partially correct answers.
7. Please notify the instructor and TA (#questions) if any question is unclear.

1: (10 points)

When classifying high-dimensional data, one strategy is to first perform principal component analysis (PCA), and then to perform classification on the dimension-reduced data. Will this strategy help with classification? Why or why not?

Answer 1:

It does help with the classification most of the time.

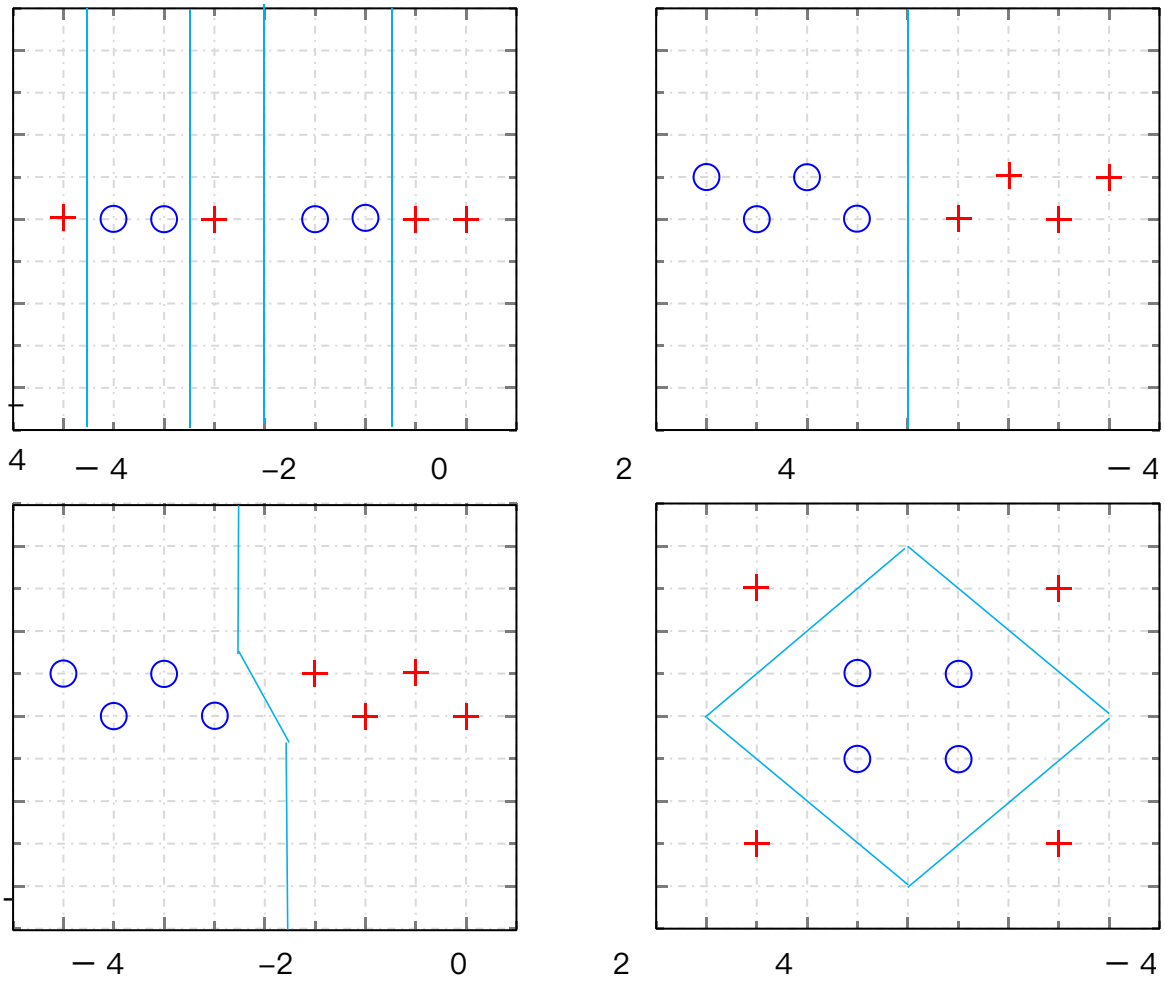
Since when we perform the classification, too many features may degrade the performance, that is why we need a way to reduce the dimensionality of the data.

PCA is good choice, it could help create a new subset features, which are a linear combinations of the original features. These subset features generated are principal components, that capture most of the variance of the original features, thus there will not so much information loss in this process.

However, PCA does have its limitations, that it does not necessarily help separating the data, non-linear structures won't be captured, in which case, we might need other methods to help reduce dimensionality.

2: (10 points)

You use *k* -nearest neighbor (*k* NN) to classify each two-class dataset shown below. You apply each classifier back to the data on which it was trained (or, equivalently, you apply it to new yet identical data). Each trained classifier will generate one or more decision boundaries, which separate classes from one another. Draw the (approximate) decision boundaries produced by the *1-nearest neighbor* classifier on each of the following datasets. Draw a line to denote each decision boundary.



For each dataset, circles and crosses denote different classes of instances.

3: (10 points)

Consider the problem of spam detection. Suppose we train a naive Bayes classifier on a dataset consisting of spam and non-spam emails. Suppose the trained classifier produces the following predictions of the probability of an email being spam for 10 instances. Give the confusion matrix and misclassification rate when we use a prediction threshold of 0.990 (i.e., we classify an instance as spam if and only if the predicted probability is *greater* than 0.990).

Predicted Probability	Actual Label
.001	not spam
.010	not spam
.500	not spam
.600	not spam
.980	not spam
.400	spam
.800	spam
.900	spam
.995	spam
.999	spam

Answer 3:

N = 10	Predicted Positive	Predicted Negative	
Actual Positive	TP = 2	FN = 0	2
Actual Negative	FP = 3	TN = 5	8
	5	5	

Misclassification rate = (FP + FN) / total = (3 + 0) / 10 = 0.3

4: (10 points)

Suppose we want to classify whether a given customer is a good credit risk based on two features: the customer's savings and the customer's income. Given the following training dataset, use the multinomial naïve Bayes classifier to classify the test instance as a "good" or "bad" credit risk based on the features. *Please show your work.*

Training instances:

Savings	Income	Credit Risk
medium	75	good
low	50	bad
high	25	bad
medium	50	good
low	50	good
high	100	good
low	25	bad
medium	75	good

New, unlabeled test instance:

Savings	Income	Credit Risk
low	100	?

Answer 4:

P(good) = P(Saving = low | good) * P(Income = 100 | good)

Credit Risk	Saving = low	
good	Yes	1 / 3
	No	2 / 3
bad	Yes	2 / 3
	No	1 / 3

Credit Risk	Income = 100	
good	Yes	1
	No	0
bad	Yes	0
	No	1

P(good) = 1 / 3 * 1 = 1 / 3, so the credit risk tend to be 'bad'.

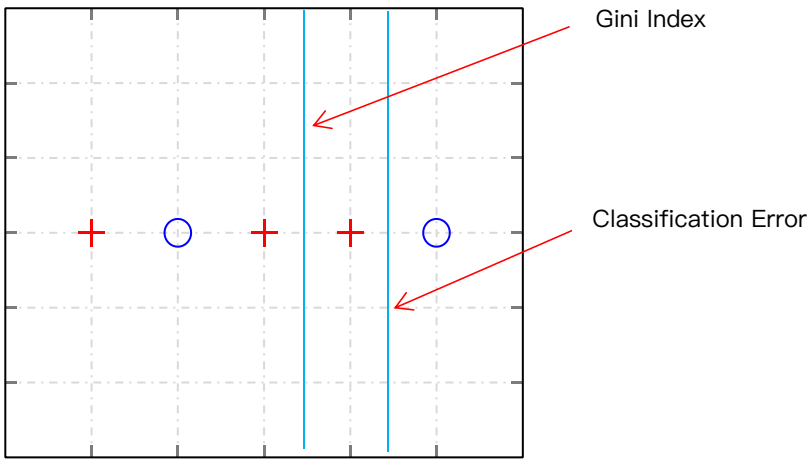
5: (10 points)

Suppose I wish to employ a one-level decision tree (i.e., a decision stump) to classify data. Recall that decision trees split data into subsets, and there are many measures that can be used to determine the best way to split data. I'm interested in two measures of impurity: Gini index and classification error. If we denote the proportion of instances belonging to class k as p_k, where there are K classes ($K = 1, \ldots, n$), then we can formally define these criteria as follows:

$$\text{Gini index} = 1 - \sum_{k=1}^{K} {p_k}^2$$

$$\text{Classification error} = 1 - \max_k p_k.$$

Each measure can be used to decide the best split on the following two-class dataset.



Data to classify (circles and crosses denote different classes of instances).

Help me decide how to classify the above data according to Feature 1 by completing the following tasks. Draw a line to denote each decision boundary:

a) Draw and label a line to indicate the best split as measured by Gini index.

b) Draw and label a line to indicate the best split as measured by classification error.

6: (10 points)

Suppose you built and trained two ensembles, one using bagging (bootstrap aggregating) and one using stacking/blending (stacked generalization), with each ensemble composed of the same three "base" classifiers. You then apply each ensemble to a dataset containing 10 unseen instances. The actual class labels and the labels predicted by each of the base classifiers are illustrated in the following table. In this particular scenario, would one of the two ensemble methods potentially outperform the other? Please explain.

Actual	Predicted		
Label	Classifier 1	Classifier 2	Classifier 3
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
0	1	1	1
0	1	1	1
0	1	1	1
0	1	1	1
0	1	1	1

Answer 6:

As we know that bagging is an ensemble method, in which we create n randomly bootstrap samples from the dataset and build classifiers for each sample, and then output the majority vote of the results of these classifiers.

While for stacking, it has two levels of classifier, in which we first make predictions with all of the base classifiers, then we use a meta classifier to help us reduce the generalization error.

Therefore in this scenario, stacking will have a better performance over bagging. Since in stacking, the meta level classifier is trained on to learn the base classifiers. Thus adding the estimated errors to the output of the base classifiers can improve prediction.

7: (10 points)

Everaldo and Reid sample 2,000 and 200 points from the same population, respectively. They both decide to use a hold-out method by keeping 50% for training and 50% for testing. Everaldo builds a classifier achieving 90% accuracy; Reid also builds a classifier achieving 90% accuracy. Both argue about which classifier's performance is closer to the actual error rate: Reid argues that his classifier is better, while Everaldo argues that his own is better. Who do you think should win this argument and why?

Answer 7:

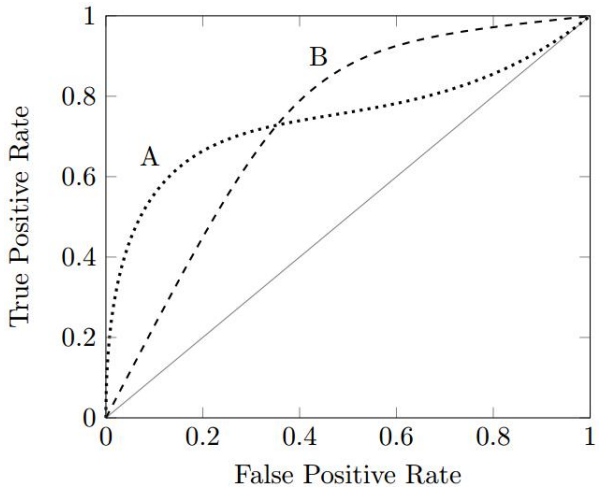
I think Everaldo should win this argument.

It is because he has more instances for both training and testing, therefore his samples tend to cover more about of the information of the data, and his classifier should have a lower error rate than Reid's.

8: (10 points)

HugePharma is going to launch a screening test for a serious disease. Based on a very cheap clinical exam, the test is able to identify potentially ill patients with a certain degree of accuracy. Patients that test positive during the screening are subjected to further inspections with more accurate tests. Accordingly, to ensure that ill patients are not left unidentified, the screening test must prevent *false negative* outcomes as much as possible, even at the cost of additional false positive outcomes.

You are asked to choose between two families of classification models to implement the screening test. Performance of these families are qualitatively described by the following receiver operating characteristic (ROC) curves.



Representative ROC curves for two families of classification models (dashed and dotted lines).

a) Discuss which is, between family A and family B, the best choice for the screening test.

Family B is the best choice.

We know that FNR = 1 - TPR, therefore to minimize FNR, we need to maximize TPR.

For most range of B, it has a greater TPR than A, in the range of 0.3 to 1.

Although B has greater FPR than A in the range of 0 to 0.3, we could take the cost of this additional FPR.

b) Suggest a suitable metric to compare the performance of two different classifiers to be used for the screening test.

Recall should be a suitable metric, since it is related to TP and FN.

Recall = TP / (TP + FN), a classifier with a minimized number of FN, should have a Recall more closer to 1.

9: (10 points)

Is the curse of dimensionality more problematic for a decision tree classifier or a Naive Bayes classifier? Why?

Answer 9:

We know that when the dimensionality increase, the rate of convergence decrease exponentially, that is so called curse of dimensionality.

Decision Tree tend to have more problem when the dimensionality increase, such as over fitting problem. However, dimensionality does not have so much influence on Naive Bayers, since it assumes that each feature is independent from the other, which helps mitigate the curse of dimensionality a little bit.

10: (10 points)

Suppose you were just hired by the world's largest car manufacturer (Toltzwablen) to develop a machine learning model that is capable of predicting what airbags are likely to fail their inspection tests. You were also warned beforehand that, historically, only about 2% of the airbags ever fail these tests. Before actually starting your experiments, you sat down to take note of everything you'll need to successfully complete this task. *What would that list look like?* Describe your entire setup in detail. Make sure to talk about (1) what data would you use to train your model , (2) what type of model you would use, (3) how would you sample/split your data, (4) What metric would you use to evaluate your model and why, (5) how you might deliver your model to the team that will use it, and whatever else you think is relevant.

Answer 10:

(1). Firstly, gather all of the former airbags inspection test data and cleaning the data to make it well formatted.

(2). Secondly, better understand the data, such as checking the distribution of the data and finding out the most important features.

(3). Thirdly, choose one of the classifiers from KNN, Naive Bayers, Decision Tree and Random Forest.

(4). Fourthly, sample the data with one of the cross validation method, such as KFold, ShuffleSplit, StratifiedKFold and StratifiedShuffleSplit. If data is imbalanced, try to leverage SMOTE to better sample the data. For the split ratio, hold out 70% data for training and the rest 30% for testing.

(5). Fifthly, pick up the right metric to evaluate the model, such as Accuracy, ROC, AUC and PR, all of them could be calculated from the confusion matrix.

(6). Finally, export the better trained model out, and create a micro-service with UI interface, in this way, the team could leverage the model to make predictions with fresh new airbag inspection test data.