Assignment 4

Lavinia Wang #1473704

Due Date: Monday, March 5th, by midnight

Total number of points: 50 points

Problem 1 (10 points):

A. (2 points) Which of the following statements are true? Briefly explain your answer.

1. Training a k-nearest-neighbors classifier takes more computational time than applying it.

False. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In application, a test point is classified by assigning the label which are most frequent among the k training samples nearest to that query point – hence higher computation.

2. The more training examples, the more accurate the prediction of a k-nearest-neighbors.

True. From learning curve, we know that accuracy increases when sample size grows.

3. k-nearest-neighbors cannot be used for regression.

False. We can also use k-NN for regression problems. In this case the prediction can be based on the mean or the median of the k-most similar instances.

4. A k-nearest-neighbors is sensitive to outliers.

True. The larger the distance to the k-NN, the lower the local density, the more likely the query point is an outlier.

B. (4 points) Would the following binary classifiers be able to correctly separate the training data (circles vs. triangles) given in Figure 1? Briefly explain your answer and show the decision boundary for each one of the two classifiers:

1. Decision tree classifier
2. 3-nearest neighbor classifier with the Euclidean distance

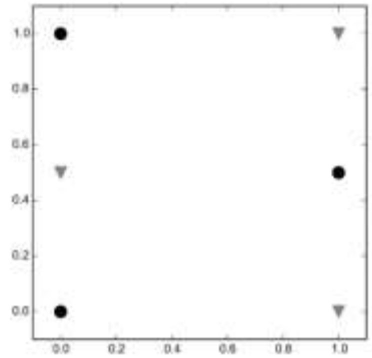

Figure 1: Training data

DT: Yes. Partition the space with lines orthogonal to the axes in such a way that every sample ends up in a different region.

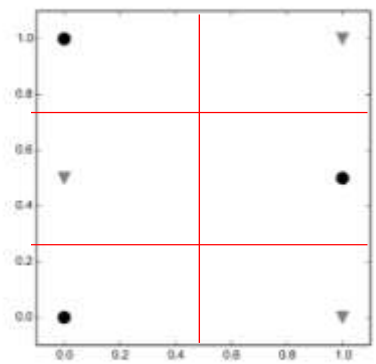

Figure 2: Decision Tree

3-NN: No. 3 nearest neighbors of any point in our training set are 1 of the same class and 2 of the opposite class, therefore 3-NN will be systematically wrong.

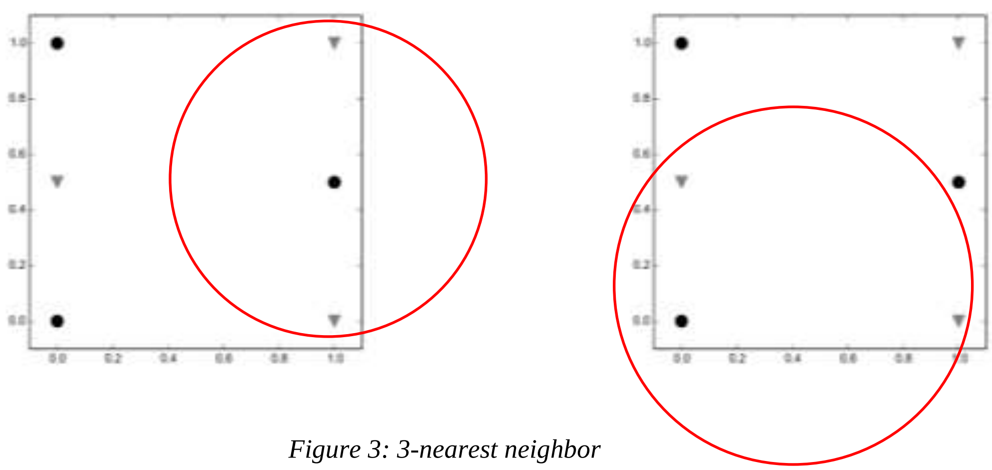

Figure 3: 3-nearest neighbor

C. (4 points) Figure 2 presents the performance of several algorithms applied to the problem of classifying molecules in two classes: those that inhibit Human Respiratory Syncytial Virus (HRSV), and those that do not. HRSV is the most frequent cause of respiratory tract infections in small children, with a worldwide estimated prevalence of about 34 million cases per year among children under 5 years of age.

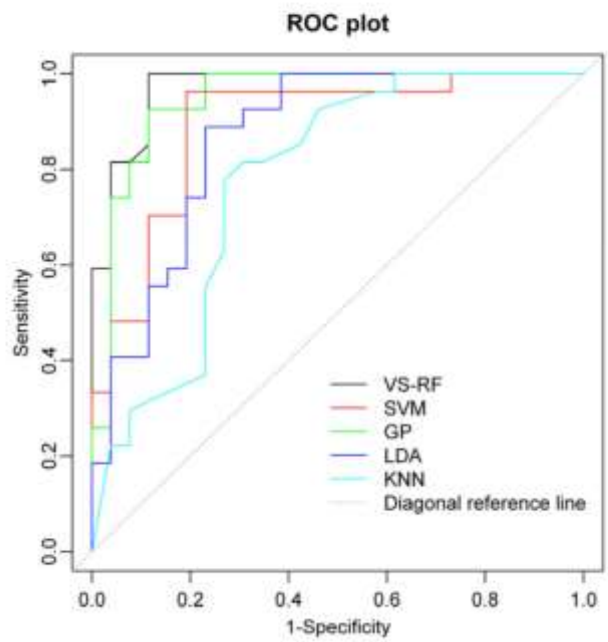

Figure 2: : ROC curves for several algorithms classifying molecules according to their action on HRSV, computed on a test set. Sensitivity = True Positive Rate. Specificity = 1 - False Positive Rate. VS-RF: Random Forest. SVM: Support Vector Machine. GP: Gaussian Process. LDA: Linear Discriminant Analysis. kNN: k-Nearest Neighbors. Source: M. Hao, Y. Li, Y. Wang, and S. Zhang, Int. J. Mol. Sci. 2011, 12(2), 1259-1280.

1. Which method gives the best performance? Explain your answer.

Random Forest. The best method should have the largest AUC(area under curve). From figure 2 it is obvious that VS-RF denoted in black line is the one closest to point (0.0, 1.0) and has largest area.

2. The goal of this study is to develop an algorithm that can be used to suggest, among a large collection of several millions of molecules, those that should be experimentally tested for activity against HRSV. Compounds that are active against HSRV are good leads from which to develop new medical treatments against infections caused by this virus. In this context, is it preferable to have a high sensitivity or a high specificity? Which part of the ROC curve is the most interesting?

We want a low false positive rate (meaning those who actually has no disease but diagnosed has a disease), i.e. high specificity. We’re interested in the left part of the curve: given a fixed specificity, what is the highest sensitivity we can get.

3. In this study, the authors have represented the molecules based on 777 descriptors. Those descriptors include the number of oxygen atoms, the molecular weights, the number of rotatable bonds, or the estimated solubility of the molecule. They have fewer samples (216) than descriptors. What is the danger here? How would you solve this issue?

With such small sample size, there might be overfitting in the model. In order to avoid this problem, if time and other condition permits, I would collect more sample for training.

Problem 2 (20 points):

Download the letter recognition data from: **http://archive.ics.uci.edu/ml/datasets/Letter+Recognition**

The objective is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15. Below is the attribute information, but more information on the data and how it was used for data mining research can be found in the paper:

P. W. Frey and D. J. Slate. "Letter Recognition Using Holland-style Adaptive Classifiers". (Machine Learning Vol 6 #2 March 91)

Attribute Information:

1. lettr capital letter (26 values from A to Z)
2. x-box horizontal position of box (integer)
3. y-box vertical position of box (integer)
4. width width of box (integer)
5. high height of box (integer)
6. onpix total # on pixels (integer)
7. x-bar mean x of on pixels in box (integer)
8. y-bar mean y of on pixels in box (integer)
9. x2bar mean x variance (integer)
10. y2bar mean y variance (integer)
11. xybar mean x y correlation (integer)
12. x2ybr mean of x * x * y (integer)
13. xy2br mean of x * y * y (integer)
14. x-ege mean edge count left to right (integer)
15. xegvy correlation of x-ege with y (integer)
16. y-ege mean edge count bottom to top (integer)
17. yegvx correlation of y-ege with x (integer)

Create a classification model for letter recognition using decision trees as a classification method with a holdout partitioning technique for splitting the data into training versus testing.

a. Changing the values for the depth, number of cases per parent and number of cases per leaf produces different tree configurations with different accuracies for training and testing. Choose at least five different configurations and report the accuracy for training and testing for each one of them. Which configuration will you choose as the best model? Explain your answer.

Hold-out Partitioning (66%)			
	Training	Testing	Complexity
N_P=115	64.10%	62.70%	75
N_C=57			
N_P=100	65.10%	64.00%	76

Hold-out Partitioning (66%)			
	Training	Testing	Complexity
N_P=24	79.60%	75.70%	256
N_C=12			
N_P=23	80.00%	75.90%	248

N_C=50			
N_P=50 N_C=25	71.40%	69.70%	137
N_P=40 N_C=20	74.90%	72.00%	162
N_P=35 N_C=17	76.10%	73.00%	182
N_P=30 N_C=15	76.80%	72.90%	223
N_P=25 N_C=12	78.70%	73.90%	249
N_P=20 N_C=10	82.40%	78.90%	304
N_P=15 N_C=7	84.60%	79.40%	383
N_P=10 N_C=5	87.80%	80.70%	503

N_C=12			
N_P=22 N_C=11	80.30%	76.50%	258
N_P=21 N_C=11	80.70%	77.10%	275
N_P=20 N_C=10	81.20%	77.20%	311
N_P=19 N_C=10	81.10%	77.40%	282
N_P=18 N_C=9	82.00%	77.10%	301
N_P=17 N_C=9	82.90%	77.90%	313
N_P=16 N_C=8	83.30%	77.80%	348
N_P=15 N_C=7	84.60%	79.80%	385

Table 1: Decision tree model for letter data accuracy

The number of training data is 20000 * 66% = 13,200. The initial number of parent node is square root of 13,200, which is approximately 115. Then I decreased number of both parent and child node until I reach (N_P=10, N_C=5). At this configuration, it is obvious that overfitting occurred. So I went back to test parent node number in range [15, 25] and found out that accuracy of my training data would improve as I decrease the number of parent node while accuracy of my test data was stable around 78%. After comparing configuration N_P=21, N_C=11with complexity 275 and N_P=20, N_C=10 with complexity 304, with both depth=20, I think the former is a better model because of its less complexity.

b. For the best tree configuration, report the misclassification matrix and interpret it. In your opinion, is accuracy a good way to interpret the performance of the model? If not, suggest other measures.

Classification

Sample		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Percent Correct
		Predicted																										
Training	A	445	0	5	0	2	0	7	1	0	2	4	13	4	3	4	0	3	0	16	6	0	0	0	0	5	3	85.1%
	B	0	420	0	2	3	4	12	9	3	0	9	0	0	0	5	10	0	16	3	0	2	0	1	2	2	3	83.0%
	C	0	2	348	1	6	11	18	1	2	0	7	7	0	0	14	0	8	1	9	4	1	0	0	4	10	1	76.5%
	D	0	22	0	423	0	16	3	13	0	1	0	0	0	9	18	7	1	5	8	2	1	0	1	5	0	3	78.6%
	E	0	0	3	4	375	4	19	0	5	1	10	8	0	0	11	0	6	1	33	4	4	0	0	16	4	3	73.4%
	F	0	4	0	1	0	410	5	5	2	11	1	5	0	1	3	20	0	0	1	12	1	4	1	8	2	1	82.3%
	G	1	4	5	9	10	2	386	6	0	2	3	3	4	1	22	1	15	2	16	2	0	0	1	10	2	3	75.7%
	H	3	13	0	28	3	18	13	291	0	8	27	3	2	3	13	3	2	10	10	1	3	9	6	12	0	2	60.2%
	I	0	4	3	2	1	9	4	0	442	15	2	2	0	0	2	9	0	1	10	0	0	0	2	2	1	0	86.5%
	J	0	2	0	1	3	3	4	8	22	416	5	7	0	0	3	7	1	0	2	2	0	0	0	3	4	1	84.2%
	K	0	12	1	0	4	11	3	1	0	1	415	2	6	1	4	0	2	6	13	1	0	0	6	16	0	0	82.2%
	L	4	3	5	0	11	0	2	1	2	6	5	443	0	0	1	1	6	5	15	0	0	0	5	4	0	5	84.5%
	M	11	0	0	3	0	8	6	3	0	0	0	4	441	19	4	0	3	6	1	4	3	0	11	1	2	1	83.1%
	N	1	3	0	15	1	4	2	7	0	3	10	2	2	437	2	8	1	0	0	0	7	1	9	2	0	0	84.5%
	O	0	5	0	16	2	3	12	2	4	4	3	1	0	4	391	10	14	10	3	4	2	0	3	12	2	0	77.1%
	P	0	2	2	2	0	22	8	3	4	4	0	1	0	8	4	429	0	1	0	7	1	4	1	2	5	0	84.1%
	Q	0	8	0	3	22	0	17	3	2	6	5	0	0	0	26	0	380	8	5	4	2	3	7	6	2	2	74.4%
	R	6	13	2	7	2	6	4	4	5	12	22	2	0	7	5	1	1	372	17	1	2	0	4	5	0	0	74.4%
	S	0	13	0	3	12	13	12	9	5	6	7	0	0	3	4	5	4	4	341	9	4	2	0	14	1	9	71.0%
	T	1	4	0	2	2	14	2	0	1	2	1	8	1	0	3	4	0	1	1	419	9	4	3	7	26	0	81.4%
	U	1	1	0	0	1	0	7	2	0	0	7	0	7	18	14	0	1	0	1	1	489	0	3	1	4	0	87.6%
	V	0	3	0	0	1	2	15	0	0	2	4	0	1	4	1	7	1	1	0	1	4	448	5	0	14	0	87.2%
	W	2	2	0	1	0	1	8	1	0	0	1	0	8	15	2	6	0	0	0	0	12	3	416	0	2	0	86.7%
	X	1	15	3	2	3	15	2	3	0	4	23	9	0	0	0	2	1	3	6	2	0	0	0	436	2	2	81.6%
	Y	3	0	0	3	0	1	0	3	0	5	0	0	0	8	2	2	8	0	0	15	6	16	0	5	446	2	85.0%
	Z	0	9	0	3	4	3	7	1	0	6	0	1	0	0	4	2	9	1	5	0	0	0	1	15	0	412	85.3%
	Overall Percentage	3.6%	4.3%	2.9%	4.0%	3.5%	4.4%	4.4%	2.9%	3.8%	3.9%	4.3%	3.9%	3.6%	4.1%	4.3%	4.0%	3.5%	3.4%	3.9%	3.8%	4.2%	3.7%	3.7%	4.4%	4.1%	3.4%	80.7%
Test	A	215	1	0	0	0	0	5	0	0	0	6	11	3	1	1	2	6	1	6	6	0	0	0	0	1	1	80.8%
	B	0	205	0	3	1	4	16	1	0	2	4	0	0	0	2	6	0	13	0	0	2	0	0	1	0	0	78.8%
	C	0	1	213	0	3	3	13	0	5	0	3	7	0	0	11	0	1	1	9	1	3	0	0	0	4	3	75.8%
	D	0	9	0	211	0	7	0	2	0	1	0	0	0	2	10	1	0	4	12	1	1	0	0	4	0	2	79.0%
	E	1	1	1	5	179	0	12	0	4	0	6	7	0	1	5	0	3	0	17	1	1	0	0	8	4	1	69.6%
	F	0	3	0	2	0	219	3	6	1	4	0	3	1	2	3	12	0	0	0	5	0	0	0	5	6	2	79.1%
	G	0	5	4	5	5	0	187	4	0	1	4	2	2	2	17	0	7	5	6	2	0	0	0	5	0	0	71.1%
	H	0	8	0	16	1	4	9	143	1	5	11	0	1	1	6	2	0	9	10	4	5	8	3	1	1	2	57.0%
	I	0	5	2	3	0	7	1	0	196	8	1	1	0	0	6	4	1	1	4	1	0	0	1	2	0	0	80.3%
	J	0	2	0	1	1	2	2	4	4	201	2	3	1	1	2	2	3	2	5	2	3	0	0	8	1	1	79.4%
	K	1	5	2	1	2	5	2	0	0	0	169	1	2	0	6	0	1	8	8	0	0	5	0	16	0	0	72.2%
	L	3	0	4	0	8	0	0	0	0	2	4	193	0	0	0	2	2	1	7	0	0	0	1	6	1	3	81.4%
	M	8	1	0	2	0	4	7	0	0	0	3	2	216	6	0	0	0	3	1	1	1	0	3	3	0	0	82.8%
	N	2	2	0	5	0	3	2	3	0	0	6	0	0	226	3	4	0	0	0	0	4	0	4	0	2	0	85.0%
	O	0	6	0	5	0	1	17	3	0	0	0	0	0	2	183	6	6	4	2	1	3	0	2	5	0	0	74.4%
	P	0	1	1	6	1	18	5	2	2	4	0	2	0	7	2	227	0	1	1	7	0	1	3	1	1	0	77.5%
	Q	0	7	2	3	13	1	7	2	1	2	5	1	0	2	25	1	174	4	3	2	0	2	2	5	8	0	64.0%
	R	2	7	0	8	1	1	5	1	5	1	13	3	0	8	4	1	4	185	4	0	0	0	1	3	1	0	71.7%
	S	0	3	0	1	5	9	5	5	3	2	3	1	0	1	2	2	6	1	201	4	2	0	1	8	1	2	75.0%
	T	0	0	0	2	1	6	0	0	0	1	7	4	0	0	2	2	0	0	0	225	9	3	0	0	19	0	80.1%
	U	0	0	3	1	0	1	4	4	0	0	0	0	9	14	9	0	1	0	0	3	199	0	1	0	6	0	78.0%
	V	0	0	0	1	0	3	9	0	0	0	1	0	1	1	0	4	7	0	0	0	9	204	8	0	2	0	81.6%
	W	0	1	0	2	0	0	6	0	0	0	1	0	15	13	6	4	0	0	0	0	3	3	215	0	3	0	79.0%
	X	1	8	3	1	1	8	0	1	0	1	19	0	0	0	0	0	0	1	6	1	0	0	0	200	1	1	79.1%
	Y	4	1	0	1	0	0	1	0	0	0	0	0	0	4	0	1	4	0	3	7	1	4	0	1	227	2	87.0%
	Z	0	1	1	1	6	0	4	1	2	4	0	1	0	0	2	1	2	0	3	1	0	0	0	8	1	212	84.5%
	Overall Percentage	3.5%	4.2%	3.5%	4.2%	3.4%	4.5%	4.8%	2.7%	3.3%	3.5%	4.0%	3.6%	3.7%	4.3%	4.5%	4.2%	3.4%	3.6%	4.5%	4.1%	3.6%	3.4%	3.6%	4.3%	4.3%	3.4%	77.1%

Growing Method: CRT
Dependent Variable: capital letter

Table 2: Letter data misclassification matrix

Confusion Matrix

		Predicted																									
Sample		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Total
Training	A	445	0	5	0	2	0	7	1	0	2	4	13	4	3	4	0	3	0	16	6	0	0	0	0	5	3	523
	B	0	420	0	2	3	4	12	9	3	0	9	0	0	0	5	10	0	16	3	0	2	0	1	2	2	3	506
	C	0	2	348	1	6	11	18	1	2	0	7	7	0	0	14	0	8	1	9	4	1	0	0	4	10	1	455
	D	0	22	0	423	0	16	3	13	0	1	0	0	0	9	18	7	1	5	8	2	1	0	1	5	0	3	538
	E	0	0	3	4	375	4	19	0	5	1	10	8	0	0	11	0	6	1	33	4	4	0	0	16	4	3	511
	F	0	4	0	1	0	410	5	5	2	11	1	5	0	1	3	20	0	0	1	12	1	4	1	8	2	1	498
	G	1	4	5	9	10	2	386	6	0	2	3	3	4	1	22	1	15	2	16	2	0	0	1	10	2	3	510
	H	3	13	0	28	3	18	13	291	0	8	27	3	2	3	13	3	2	10	10	1	3	9	6	12	0	2	483
	I	0	4	3	2	1	9	4	0	442	15	2	2	0	0	2	9	0	1	10	0	0	0	2	2	1	0	511
	J	0	2	0	1	3	3	4	8	22	416	5	7	0	0	3	7	1	0	2	2	0	0	0	3	4	1	494
	K	0	12	1	0	4	11	3	1	0	1	415	2	6	1	4	0	2	6	13	1	0	0	6	16	0	0	505
	L	4	3	5	0	11	0	2	1	2	6	5	443	0	0	1	1	6	5	15	0	0	0	5	4	0	5	524
	M	11	0	0	3	0	8	6	3	0	0	0	4	441	19	4	0	3	6	1	4	3	0	11	1	2	1	531
	N	1	3	0	15	1	4	2	7	0	3	10	2	2	437	2	8	1	0	0	0	7	1	9	2	0	0	517
	O	0	5	0	16	2	3	12	2	4	4	3	1	0	4	391	10	14	10	3	4	2	0	3	12	2	0	507
	P	0	2	2	2	0	22	8	3	4	4	0	1	0	8	4	429	0	1	0	7	1	4	1	2	5	0	510
	Q	0	8	0	3	22	0	17	3	2	6	5	0	0	0	26	0	380	8	5	4	2	3	7	6	2	2	511
	R	6	13	2	7	2	6	4	4	5	12	22	2	0	7	5	1	1	372	17	1	2	0	4	5	0	0	500
	S	0	13	0	3	12	13	12	9	5	6	7	0	0	3	4	5	4	4	341	9	4	2	0	14	1	9	480
	T	1	4	0	2	2	14	2	0	1	2	1	8	1	0	3	4	0	1	1	419	9	4	3	7	26	0	515
	U	1	1	0	0	1	0	7	2	0	0	7	0	7	18	14	0	1	0	1	1	489	0	3	1	4	0	558
	V	0	3	0	0	1	2	15	0	0	2	4	0	1	4	1	7	1	1	0	1	4	448	5	0	14	0	514
	W	2	2	0	1	0	1	8	1	0	0	1	0	8	15	2	6	0	0	0	0	12	3	416	0	2	0	480
	X	1	15	3	2	3	15	2	3	0	4	23	9	0	0	0	2	1	3	6	2	0	0	0	436	2	2	534
	Y	3	0	0	3	0	1	0	3	0	5	0	0	0	8	2	2	8	0	0	15	6	16	0	5	446	2	525
	Z	0	9	0	3	4	3	7	1	0	6	0	1	0	0	4	2	9	1	5	0	0	0	1	15	0	412	483
	Total	479	564	377	531	468	580	578	377	499	517	571	521	476	541	562	534	467	454	516	501	553	494	486	588	536	453	13223
	Recall	85.1%	83.0%	76.5%	78.6%	73.4%	82.3%	75.7%	60.2%	86.5%	84.2%	82.2%	84.5%	83.1%	84.5%	77.1%	84.1%	74.4%	74.4%	71.0%	81.4%	87.6%	87.2%	86.7%	81.6%	85.0%	85.3%	**80.6%**
	Precision	92.9%	74.5%	92.3%	79.7%	80.1%	70.7%	66.8%	77.2%	88.6%	80.5%	72.7%	85.0%	92.6%	80.8%	69.6%	80.3%	81.4%	81.9%	66.1%	83.6%	88.4%	90.7%	85.6%	74.1%	83.2%	90.9%	**81.2%**
	Specificity	99.7%	98.6%	99.7%	99.0%	99.1%	98.4%	98.2%	99.2%	99.4%	99.0%	98.5%	99.2%	99.7%	99.0%	98.4%	99.0%	99.2%	99.2%	98.3%	99.2%	99.4%	99.6%	99.3%	98.5%	99.1%	99.6%	**99.1%**
	F1 score	88.8%	78.5%	83.7%	79.1%	76.6%	76.1%	71.0%	67.7%	87.5%	82.3%	77.1%	84.8%	87.6%	82.6%	73.2%	82.2%	77.7%	78.0%	68.5%	82.5%	88.0%	88.9%	86.1%	77.7%	84.1%	88.0%	**80.7%**
Test	A	215	1	0	0	0	0	5	0	0	0	6	11	3	1	1	2	6	1	6	6	0	0	0	0	1	1	266
	B	0	205	0	3	1	4	16	1	0	2	4	0	0	0	2	6	0	13	0	0	2	0	0	1	0	0	260
	C	0	1	213	0	3	3	13	0	5	0	3	7	0	0	11	0	1	1	9	1	3	0	0	0	4	3	281
	D	0	9	0	211	0	7	0	2	0	1	0	0	0	2	10	1	0	4	12	1	1	0	0	4	0	2	267
	E	1	1	1	5	179	0	12	0	4	0	6	7	0	1	5	0	3	0	17	1	1	0	0	8	4	1	257
	F	0	3	0	2	0	219	3	6	1	4	0	3	1	2	3	12	0	0	0	5	0	0	0	5	6	2	277
	G	0	5	4	5	5	0	187	4	0	1	4	2	2	2	17	0	7	5	6	2	0	0	0	5	0	0	263
	H	0	8	0	16	1	4	9	143	1	5	11	0	1	1	6	2	0	9	10	4	5	8	3	1	1	2	251
	I	0	5	2	3	0	7	1	0	196	8	1	1	0	0	6	4	1	1	4	1	0	0	1	2	0	0	244
	J	0	2	0	1	1	2	2	4	4	201	2	3	1	1	2	2	3	2	5	2	3	0	0	8	1	1	253
	K	1	5	2	1	2	5	2	0	0	0	169	1	2	0	6	0	1	8	8	0	0	5	0	16	0	0	234
	L	3	0	4	0	8	0	0	0	0	2	4	193	0	0	0	2	2	1	7	0	0	0	1	6	1	3	237
	M	8	1	0	2	0	4	7	0	0	0	3	2	216	6	0	0	0	3	1	1	1	0	3	3	0	0	261
	N	2	2	0	5	0	3	2	3	0	0	6	0	0	226	3	4	0	0	0	0	4	0	4	0	2	0	266
	O	0	6	0	5	0	1	17	3	0	0	0	0	0	2	183	6	6	4	2	1	3	0	2	5	0	0	246
	P	0	1	1	6	1	18	5	2	2	4	0	2	0	7	2	227	0	1	1	7	0	1	3	1	1	0	293
	Q	0	7	2	3	13	1	7	2	1	2	5	1	0	2	25	1	174	4	3	2	0	2	2	5	8	0	272
	R	2	7	0	8	1	1	5	1	5	1	13	3	0	8	4	1	4	185	4	0	0	0	1	3	1	0	258
	S	0	3	0	1	5	9	5	5	3	2	3	1	0	1	2	2	6	1	201	4	2	0	1	8	1	2	268
	T	0	0	0	2	1	6	0	0	0	1	7	4	0	0	2	2	0	0	0	225	9	3	0	0	19	0	281
	U	0	0	3	1	0	1	4	4	0	0	0	0	9	14	9	0	1	0	0	3	199	0	1	0	6	0	255
	V	0	0	0	1	0	3	9	0	0	0	1	0	1	1	0	4	7	0	0	0	9	204	8	0	2	0	250
	W	0	1	0	2	0	0	6	0	0	0	1	0	15	13	6	4	0	0	0	0	3	3	215	0	3	0	272
	X	1	8	3	1	1	8	0	1	0	1	19	0	0	0	0	0	0	1	6	1	0	0	0	200	1	1	253
	Y	4	1	0	1	0	0	1	0	0	0	0	0	0	4	0	1	4	0	3	7	1	4	0	1	227	2	261
	Z	0	1	1	1	6	0	4	1	2	4	0	1	0	0	2	1	2	0	3	1	0	0	0	8	1	212	251
	Total	237	283	236	286	228	306	322	182	224	239	268	242	251	294	307	284	228	244	308	275	246	230	245	290	290	232	6777
	Recall	80.8%	78.8%	75.8%	79.0%	69.6%	79.1%	71.1%	57.0%	80.3%	79.4%	72.2%	81.4%	82.8%	85.0%	74.4%	77.5%	64.0%	71.7%	75.0%	80.1%	78.0%	81.6%	79.0%	79.1%	87.0%	84.5%	**77.1%**
	Precision	90.7%	72.4%	90.3%	73.8%	78.5%	71.6%	58.1%	78.6%	87.5%	84.1%	63.1%	79.8%	86.1%	76.9%	59.6%	79.9%	76.3%	75.8%	65.3%	81.8%	80.9%	88.7%	87.8%	69.0%	78.3%	91.4%	**77.9%**
	Specificity	99.6%	98.5%	99.5%	98.5%	99.0%	98.3%	97.4%	99.2%	99.4%	99.2%	98.1%	99.0%	99.3%	98.7%	97.6%	98.9%	98.9%	98.8%	97.9%	99.0%	99.1%	99.5%	99.4%	98.2%	98.8%	99.6%	**98.8%**
	F1 score	85.5%	75.5%	82.4%	76.3%	73.8%	75.1%	63.9%	66.1%	83.8%	81.7%	67.3%	80.6%	84.4%	80.7%	66.2%	78.7%	69.6%	73.7%	69.8%	80.9%	79.4%	85.0%	83.2%	73.7%	82.4%	87.8%	**77.2%**

Table 3: Letter data DT confusion matrix

The final decision tree was built with a depth of 20; Minimum cases in parent node of 21; Minimum cases in Child node of 11with a growing method of CRT (CRT growing method attempts to maximize within-node homogeneity) and the default impurity index of GINI.

My decision tree model has an accuracy of 80.7% for training data and 77.1% accuracy for testing data. Accuracy is a great measure but only when we have symmetric datasets where values of false positive and false negatives are almost same. By looking at table 3, we can see for this letter recognition dataset, we have quite different false positive and false negative values, which generated different precision and recall for each class. Therefore, we have to look at other parameters to evaluate the performance of the model.

Such as ROC curve, specificity, precision, sensitivity/recall, f1 score and combine these measures with accuracy to give an overall model performance.

c. What are the most important three attributes for recognizing the letters?

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
mean edge count bottom to top	.264	100.0%
correlation of x-ege with y	.259	98.2%
mean x variance	.253	96.0%
mean y of on pixels in box	.243	92.1%
mean x y correlation	.238	90.3%
mean edge count left to right	.237	89.8%
mean y variance	.224	85.0%
mean of x * y * y	.215	81.7%
mean x of on pixels in box	.203	76.9%
mean of x * x * y	.187	71.0%
correlation of y-ege with x	.186	70.6%
total # on pixels	.094	35.6%
width of box	.086	32.7%
horizontal position of box	.077	29.3%
vertical position of box	.069	26.1%
height of box	.066	24.9%

Growing Method: CRT

Dependent Variable: capital letter

Table 3: Letter data independent variable importance

The most important three attributes are: y_ege(mean edge count bottom to top), xegvy(correlation of x-ege with y) and x2bar (mean x variance).

Problem 3 (20points):

On the same data from Problem 1, apply a K-nearest neighbor classifier to classify the data. Report the following:

1. If you are doing any data transformation, explain the transformation and why it is needed.

Statistics

		horizontal position of box	vertical position of box	width of box	height of box	total # on pixels	mean x of on pixels in box	mean y of on pixels in box	mean x variance	mean y variance	mean x y correlation	mean of x * x * y	mean of x * y * y	mean edge count left to right	correlation of x-ege with y	mean edge count bottom to top	correlation of y-ege with x
N	Valid	20000	20000	20000	20000	20000	20000	20000	20000	20000	20000	20000	20000	20000	20000	20000	20000
	Missing	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mean		4.02	7.04	5.12	5.37	3.51	6.90	7.50	4.63	5.18	8.28	6.45	7.93	3.05	8.34	3.69	7.80
Median		4.00	7.00	5.00	6.00	3.00	7.00	7.00	4.00	5.00	8.00	6.00	8.00	3.00	8.00	3.00	8.00
Mode		4	9	5	6	2	7	7	3	5	7	6	8	3	8	4	8
Std. Deviation		1.913	3.305	2.015	2.261	2.190	2.026	2.325	2.700	2.381	2.488	2.631	2.081	2.333	1.547	2.567	1.617
Variance		3.660	10.920	4.059	5.114	4.798	4.105	5.407	7.290	5.668	6.193	6.923	4.329	5.441	2.392	6.590	2.616
Range		15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
Minimum		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maximum		15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15

Table 4: Letter data frequency table

All the attributes in the dataset have the same range, i.e.15, meaning they have the same scale. So it is not necessary to normalize the data on this dataset. But in general, if the attributes do not have the same scale, we must normalize the data before doing any classification.

2. Report the misclassification matrix and the appropriate performance metrics for different values of K (K=1, 3, 5, and 7).

(Misclassification matrix is too large to insert. See attached excel file.)

	K=1		K=3		K=5		K=7	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Accuracy	86.9%	87.0%	85.4%	85.0%	86.0%	86.2%	85.5%	86.6%
Recall	86.9%	87.0%	85.2%	85.0%	85.9%	86.3%	85.4%	86.6%
Precision	86.9%	87.1%	86.5%	86.0%	86.6%	86.6%	86.0%	86.9%
Specificity	99.4%	99.4%	99.3%	99.3%	99.4%	99.4%	99.3%	99.4%
F measure	86.9%	87.0%	85.5%	85.1%	86.1%	86.3%	85.6%	86.6%

Table 5: Letter data K-NN performance metrics table

3. Interpret the results and also compare them with the ones obtained by using the decision trees.

When K=1, KNN has the best accuracy, which is 86.9% for training and 87%for testing. As the number of K increased, accuracy didn't improve much and was between 85% and 86%. Compared with decision trees, KNN has much better accuracy, recall, precision and f measure and slightly better specificity. If we calculate P for decision tree model and KNN(k=1):

$P = \frac{|E_1 - E_2|}{\sqrt{q(1-q)\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}} = \frac{|0.229-0.13|}{\sqrt{\frac{0.229+0.13}{2}\left(1-\frac{0.229+0.13}{2}\right)\left(\frac{1}{6777}+\frac{1}{6715}\right)}}$ =14.98 *Where* $E_1 = 1 - 0.771 = 0.229$, $E_2 = 1 - 0.87 = 0.13$, $n_1 = 6777$ *and* $n_2 = 6715$.

So we are 95% confident that the difference in the test set performance of decision tree and KNN is significant. However, even though KNN has a better accuracy, we do not know which attribute(s) would recognize letters. But decision tree would give us a clear rule for classification and is better to interpret.