Alex Teboul

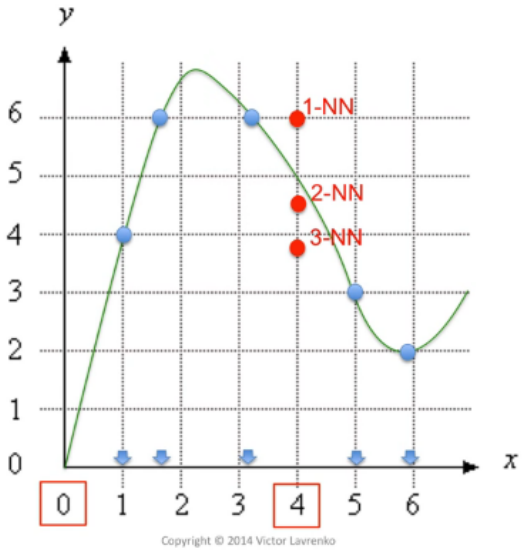
DSC 441

Assignment 4

Problem 1 (10 points):

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

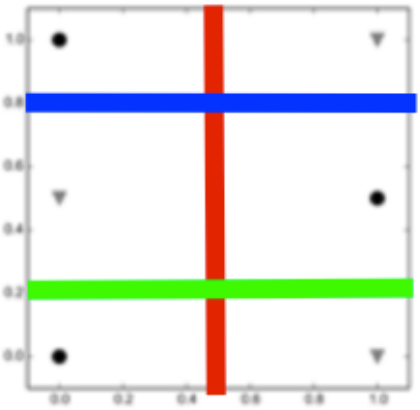
1. **Training a k-nearest-neighbors classifier takes less computational time than testing it.**
    a. True – Yes, KNN is a lazy/instance-based classification algorithm, meaning training consists of storing data samples with their associated features/labels in memory and plotting them in an n-dimensional space based on the n-features. The testing then occurs instance by instance for new data. It does so by plotting the new instances in the n-dimensional space and searching for its k-nearest neighbors using a distance measure from the new point to all training points. This means the testing calculates distances from all training data points to generate a class label. This makes testing significantly more computationally expensive than training.
2. **The more training examples, the more accurate the prediction of a k-nearest-neighbors.**
    a. True – While KNN is not generalizing on training examples to create a model, it still relies on each training instance for the k-nearest calculation. This means that a more fine-grained feature space (one with more training examples), will likely present more cases near the new instance that is to be classified by KNN. In general, more training data is better for classification algorithms as it allows them to make more accurate predictions based on previous examples.
3. **k-nearest-neighbors cannot be used for regression.**
    a. False – KNN can be used for regression, by computing the output value as the average of the k-nearest neighbors values. It isn't always a good idea to do so but it does work in some cases. If you have a couple of points that lie along some continuous value function, KNN can predict/approximate a point along this function. This means it can in some cases do relevant interpolation, but of course extrapolation won't work well because KNN is based on neighboring points. The example below illustrates this point.



4. **A k-nearest-neighbors is sensitive to the number of features.**
    a. True – Even after normalization has set the magnitudes of the different features to the same scale, a plot in a high dimensional space will likely contain a large amount of noise. Specifically, some features will be more discriminative while others will represent noise in the distance calculations of KNN. This makes feature selection important when the number of features is high. Additionally, a high number of features will necessitate a higher number of samples, and higher sample/training numbers further increases the computational expense of KNN. Euclidean Distance performs particularly poorly for higher dimensions, requiring other measures like cosine similarity.

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**
    i. Yes – These 6 points could be partitioned in such a way that each ends up in a difference region of the plot. Specifically, boundaries at 0.5 on the x-axis, 0.2 on the y-axis, and 0.8 on the y-axis could be used to create 6 rules to accurately separate all of the circles and triangles.



2. 3-nearest neighbor classifier with the Euclidean distance

i. No - Each of the points, circles and triangles has 1 same class neighbor and 2 opposite class neighbors, so 3-nearest neighbors cannot work. I believe that new points would also be classified incorrectly with 3-nearest neighbors, except at points where there is an equal distance to a circle and a triangle, with two other points of opposite class. In that case its class could be decided randomly, and thereby getting those points right about 50% of the time. Still a bad method.

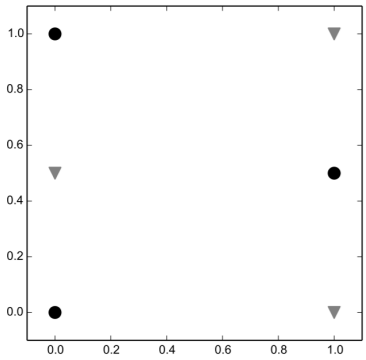


Figure 1: Training data

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.

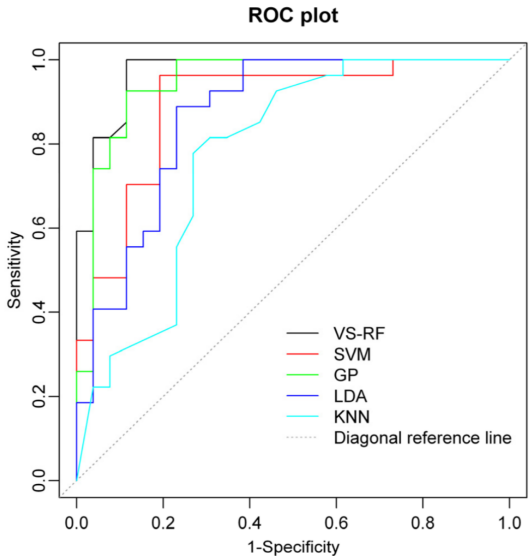


Figure 2: : ROC curves for several algorithms classifying molecules according to their action on HRSV, computed on a test set. Sensitivity = True Positive Rate. Specificity = 1 - False Positive Rate. VS-RF: Random Forest. SVM: Support Vector Machine. GP: Gaussian Process. LDA: Linear Discriminant Analysis. 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.**
    a. Random Forests. That method yields the highest area under the ROC curve (AUC) which indicates the method's ability to distinguish between the HRSV inhibiting molecules and non-inhibiting molecules.

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?**
    a. It is preferable to have a high specificity, implying that a potential HSRV inhibiting molecule that is predicted 'positive' for inhibiting HSRV almost certainly does just that. A high specificity means predicted effective compounds probably are effective, saving on development costs. The most interesting part of the curve is therefore the 'left' part. This is where specificity is high, and ideally a high sensitivity can also be achieved. High sensitivity in this case would indicate that the compounds that are not predicted to be inhibitory are in fact mostly non-inhibitory.

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?**
    a. The danger is clearly overfitting. Dimensionality reduction through feature selection and/or feature extraction is recommended. For example, feature extraction via PCA may be used.

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. (15 points) 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.

a. I would say an np=20, nc=10, 280 rule, 21 deep decision tree is the best model out of the models I tested. This model has the highest accuracy, without appearing to overfit the data. I say it isn't overfitting based on the 81.7% train to 78.8% test accuracies. This doesn't seem to indicate overfitting, though with a depth of 21 layers to the tree, it is possible that this may be occurring slightly. Across multiple tests, the difference between test and train accuracy remained less than 5%, which is reasonable. I also found that the 66%Train to 34%Test split worked just fine, even though the researchers used an 80%|20% split for their models.

66% Train | 34% Test									
max depth	**np**	**nc**	**Training Accuracy**	**Testing Accuracy**	**Complexity**	**#Nodes**	**#Terminal Nodes**	**SPSS Depth**	**Top 5 Important features**
50	400	200	49.9	48.8	rules=30, depth=11	59	30	11	15, 8, 14, 7, 11
50	180	90	61.9	60.9	rules=57, depth=10	113	57	10	14, 13, 12, 15, 10
50	160	80	62.0	60.3	rules=57, depth=11	113	57	11	14, 12, 13, 15, 10
50	140	70	63.1	62.3	rules=63, depth=13	125	63	13	14, 12, 15, 13, 10
50	120	60	63.6	63.3	rules=74, depth=15	147	74	15	14, 12, 11, 10, 15
50	100	50	65.7	64.9	rules=82, depth=15	163	82	15	14, 12, 11, 15, 10
50	20	10	81.7	78.8	rules=280, depth=21	559	280	21	14, 12, 13, 10, 11
80% Train | 20% Test (Recommended by researchers – 16000 train, 4000 test)									

max depth	np	nc	Training Accuracy	Testing Accuracy	Complexity	#Nodes	#Terminal Nodes	SPSS Depth	Top 5 Important features
50	400	200	54.1	54.5	rules=37, depth=9	73	37	9	14, 12, 15, 13, 10
50	180	90	62.2	62.3	rules=64, depth=15	127	64	15	14, 15, 12, 10, 7
50	160	80	63.5	62.2	rules=67, depth=14	133	67	14	14, 12, 13, 15, 10
50	140	70	64.8	62.7	rules=70, depth=12	139	70	12	14, 12, 13, 10, 7
50	120	60	65.1	64.3	rules=84, depth=13	167	84	13	14, 12, 15, 10, 7
50	100	50	68.1	66.6	rules=101, depth=16	201	101	16	14, 12, 11, 10, 15
50	20	10	83.4	79.0	rules=348, depth=21	695	348	21	14, 12, 11, 15, 10

b. (4 points) 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.

a. Accuracy is a reasonable way to interpret the performance of the model because the target variable classes (letters) are nearly balanced. In the frequency table below, it is shown that each letter appears roughly the same number of times as other letters. Sensitivity and specificity are also worthy measures of model performance. Unlike in say a medical diagnosis model, there isn't a great need for high specificity, but it would still be ideal to have.

b. I submitted the misclassification matrix in the form of a pdf document as well because it didn't fit on this page. From the misclassification matrix however it is clear that the accuracy for each letter is approximately the same, which is good. That means the model does not favor particular letters over others.

Risk

Sample	Estimate	Std. Error
Training	.182	.003

Test	.226	.005

Growing Method: CRT
Dependent Variable: Letter

Letter

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	A	789	3.9	3.9	3.9
	B	766	3.8	3.8	7.8
	C	736	3.7	3.7	11.5
	D	805	4.0	4.0	15.5
	E	768	3.8	3.8	19.3
	F	775	3.9	3.9	23.2
	G	773	3.9	3.9	27.1
	H	734	3.7	3.7	30.7
	I	755	3.8	3.8	34.5
	J	747	3.7	3.7	38.2
	K	739	3.7	3.7	41.9
	L	761	3.8	3.8	45.7
	M	792	4.0	4.0	49.7
	N	783	3.9	3.9	53.6
	O	753	3.8	3.8	57.4
	P	803	4.0	4.0	61.4
	Q	783	3.9	3.9	65.3
	R	758	3.8	3.8	69.1
	S	748	3.7	3.7	72.8
	T	796	4.0	4.0	76.8
	U	813	4.1	4.1	80.9
	V	764	3.8	3.8	84.7

	W	752	3.8	3.8	88.5
	X	787	3.9	3.9	92.4
	Y	786	3.9	3.9	96.3
	Z	734	3.7	3.7	100.0
	Total	20000	100.0	100.0	

c. (1 point) What are the most important three attributes for recognizing the letters?

a. The three most important attributes for recognizing the letters are V14. x-ege mean edge count left to right (integer), and V12. x2ybr mean of x * x * y (integer), V13. xy2br mean of x * y * y (integer).

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.250	100.0%
V12	.247	98.9%
V13	.230	92.1%

Growing Method: CRT
Dependent Variable: Letter

Problem 3 (20points):

On the same data from Problem 2, apply a K-nearest neighbor classifier to classify the data. Report the following:

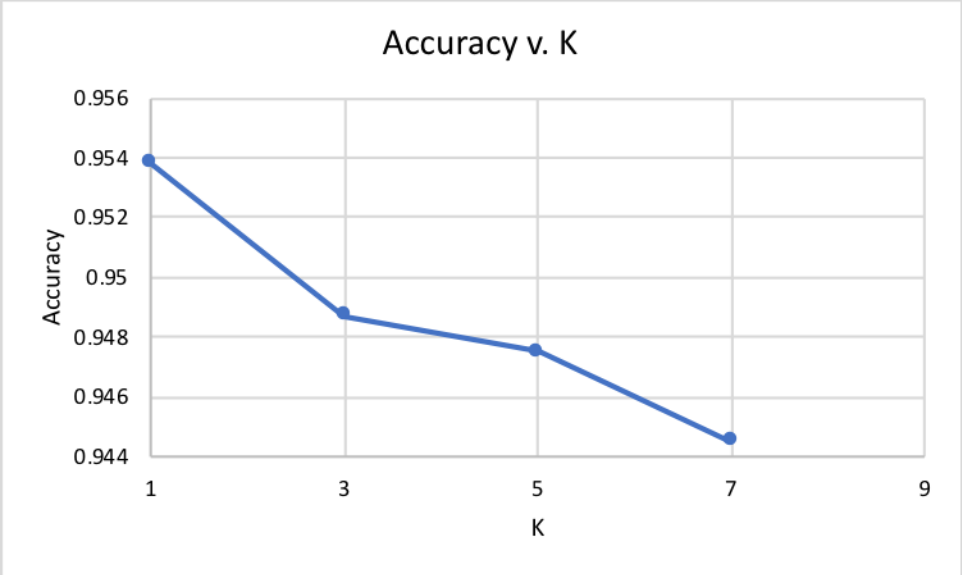
1. **(2 points) If you are doing any data transformation, explain the transformation and why it is needed.**

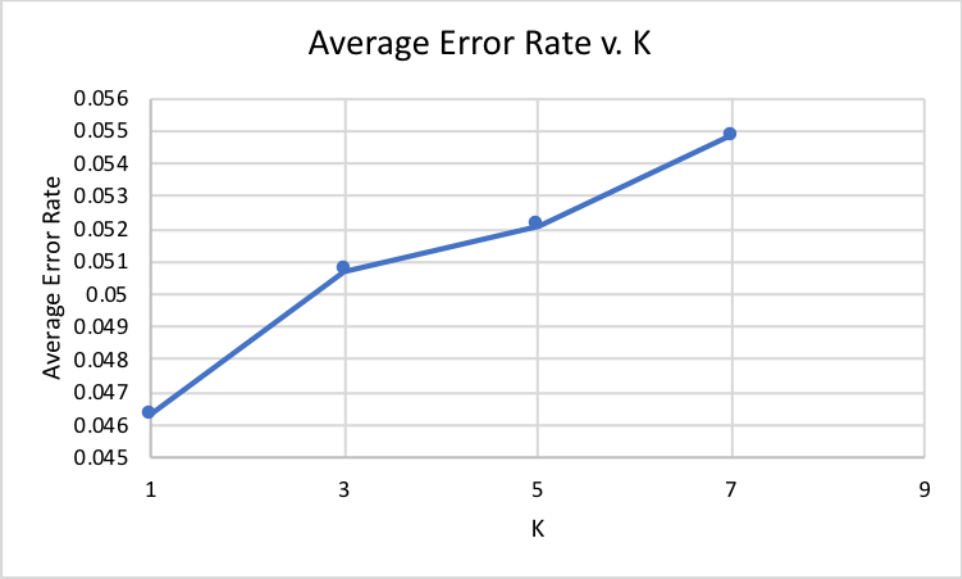
a. Yes, the data should be normalized in order to perform the K-nearest neighbor classification. KNN is sensitive to variations in the scales of the features because it uses a distance measure between points to predict test case's class. I chose to normalize the data using min-max normalization on the scale [0,1].

2. **(16 points) Report the misclassification matrix and the appropriate performance metrics for different values of K (K=1, 3, 5, and 7).**

    a. I report here the accuracy and average error rate of each K-NN model calculated from the misclassification matrix in Excel. I will submit the excel document as well because the misclassifaction matrix does not fit in this document. For K=1 there is a chance that it is overfitting, so the slightly lower accuracy of K=3 is acceptable as the best model. K=1 would not be as robust in terms of dealing with outliers or noise in the data. So for K=3, accuracy is 94.87% and the error rate is approximately 0.05. This is a high degree of accuracy, especially compared to the decision tree model.

K	Accuracy	Average Error Rate
1	0.9538	0.046322793
3	0.9487	0.050713788
5	0.9475	0.052093817
7	0.9445	0.054839704





3. **(2 points) Interpret the results and also compare them with the ones obtained by using the decision trees.**
    a. K-NN performed much better at classifying the letters than the decision trees did. The best decision tree achieved a test accuracy of 78.8%, while all K-NN models tested achieved above a 94% accuracy. The 'best' K-NN model I decided on was the K=3 model, which achieved 94.87% accuracy and an average error rate of only approximately 0.05.

Problem 2 Appendix

66/34 splits

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter
	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17
	Validation	Split Sample

	Maximum Tree Depth	50
	Minimum Cases in Parent Node	140
	Minimum Cases in Child Node	70
Results	Independent Variables Included	V12, V8, V11, V7, V10, V14, V13, V9, V15, V5, V2, V3, V16, V6, V17, V4
	Number of Nodes	125
	Number of Terminal Nodes	63
	Depth	13

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.194	100.0%
V12	.174	89.6%
V15	.165	85.3%
V13	.163	84.3%
V10	.159	82.0%
V11	.149	76.9%
V7	.147	75.8%
V9	.145	74.9%
V16	.139	71.7%
V8	.102	52.5%

V17	.101	51.9%
V2	.040	20.4%
V6	.039	20.2%
V4	.038	19.6%
V3	.025	12.9%
V5	.025	12.8%

Growing Method: CRT
Dependent Variable: Letter

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter
	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17
	Validation	Split Sample
	Maximum Tree Depth	50
	Minimum Cases in Parent Node	120
	Minimum Cases in Child Node	60
Results	Independent Variables Included	V12, V8, V11, V7, V2, V4, V10, V14, V13, V15, V9, V3, V5, V6, V16, V17
	Number of Nodes	147
	Number of Terminal Nodes	74
	Depth	15

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.247	100.0%
V12	.186	75.2%
V11	.171	69.1%
V10	.157	63.7%
V15	.154	62.4%
V16	.152	61.4%
V7	.139	56.0%
V13	.130	52.6%
V9	.127	51.4%
V8	.125	50.4%
V17	.085	34.3%
V6	.054	22.0%
V4	.047	18.9%
V2	.046	18.5%
V5	.027	10.9%
V3	.024	9.8%

Growing Method: CRT
Dependent Variable: Letter

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter

	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17
	Validation	Split Sample
	Maximum Tree Depth	50
	Minimum Cases in Parent Node	100
	Minimum Cases in Child Node	50
Results	Independent Variables Included	V12, V8, V11, V7, V10, V14, V13, V15, V9, V5, V2, V3, V16, V17, V6, V4
	Number of Nodes	163
	Number of Terminal Nodes	82
	Depth	15

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.252	100.0%
V12	.195	77.3%
V11	.192	76.1%
V15	.183	72.5%
V10	.165	65.3%
V9	.160	63.5%
V16	.158	62.8%

V7	.149	59.2%
V13	.147	58.3%
V8	.125	49.4%
V17	.107	42.3%
V6	.055	21.9%
V4	.053	21.1%
V2	.050	19.9%
V3	.027	10.6%
V5	.026	10.4%

Growing Method: CRT
Dependent Variable: Letter

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter
	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17
	Validation	Split Sample
	Maximum Tree Depth	50
	Minimum Cases in Parent Node	160
	Minimum Cases in Child Node	80
Results	Independent Variables Included	V12, V8, V11, V7, V10, V14, V13, V15, V9, V5, V3, V2, V17, V6, V16, V4
	Number of Nodes	113

	Number of Terminal Nodes	57
	Depth	11

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.206	100.0%
V12	.187	90.6%
V13	.173	84.0%
V15	.172	83.6%
V10	.155	75.3%
V7	.142	69.0%
V11	.141	68.3%
V16	.140	67.7%
V9	.130	63.3%
V8	.098	47.8%
V17	.094	45.4%
V6	.037	17.9%
V2	.036	17.6%
V4	.034	16.7%
V5	.021	10.3%
V3	.018	8.7%

Growing Method: CRT
Dependent Variable: Letter

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter
	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17
	Validation	Split Sample
	Maximum Tree Depth	50
	Minimum Cases in Parent Node	180
	Minimum Cases in Child Node	90
Results	Independent Variables Included	V12, V8, V11, V7, V10, V14, V13, V15, V9, V3, V5, V6, V17, V2, V4, V16
	Number of Nodes	113
	Number of Terminal Nodes	57
	Depth	10

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.196	100.0%
V13	.181	92.1%
V12	.169	86.0%
V15	.159	81.3%
V10	.154	78.7%
V7	.150	76.2%
V11	.146	74.4%
V16	.144	73.7%
V9	.137	69.6%
V8	.107	54.7%
V17	.093	47.3%
V6	.037	18.8%
V2	.034	17.3%
V4	.034	17.1%
V5	.025	12.8%
V3	.022	11.0%

Growing Method: CRT
Dependent Variable: Letter

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter
	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17

	Validation	Split Sample
	Maximum Tree Depth	50
	Minimum Cases in Parent Node	400
	Minimum Cases in Child Node	200
Results	Independent Variables Included	V12, V8, V11, V7, V10, V14, V13, V15, V9, V5, V3, V6, V2, V17, V16, V4
	Number of Nodes	59
	Number of Terminal Nodes	30
	Depth	11

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V15	.127	100.0%
V8	.124	97.9%
V14	.122	95.7%
V7	.114	90.1%
V11	.099	78.1%
V13	.099	77.6%
V10	.097	76.8%
V12	.087	68.8%
V9	.078	61.3%

V16	.073	57.3%
V17	.045	35.1%
V4	.032	25.2%
V6	.026	20.1%
V2	.020	15.4%
V5	.013	10.0%
V3	.013	9.9%

Growing Method: CRT
Dependent Variable: Letter

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter
	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17
	Validation	Split Sample
	Maximum Tree Depth	50
	Minimum Cases in Parent Node	20
	Minimum Cases in Child Node	10
Results	Independent Variables Included	V12, V8, V11, V7, V10, V14, V13, V15, V9, V5, V3, V2, V6, V17, V4, V16
	Number of Nodes	559
	Number of Terminal Nodes	280
	Depth	21

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.250	100.0%
V12	.247	98.9%
V13	.230	92.1%
V10	.226	90.6%
V11	.225	90.2%
V15	.217	87.1%
V16	.217	86.8%
V9	.209	83.9%
V7	.204	81.9%
V17	.174	69.8%
V8	.158	63.5%
V6	.080	32.1%
V2	.077	30.8%
V4	.069	27.6%
V3	.060	24.2%
V5	.058	23.2%

Growing Method: CRT
Dependent Variable: Letter

80/20 splits

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter
	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17
	Validation	Split Sample
	Maximum Tree Depth	50
	Minimum Cases in Parent Node	400
	Minimum Cases in Child Node	200
Results	Independent Variables Included	V12, V8, V11, V7, V2, V4, V10, V14, V13, V15, V9, V3, V5, V6, V17, V16
	Number of Nodes	73
	Number of Terminal Nodes	37
	Depth	9

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.167	100.0%
V12	.155	93.1%
V15	.136	81.4%
V13	.120	71.7%
V10	.118	70.4%

V7	.114	68.1%
V11	.112	67.3%
V16	.097	58.3%
V9	.092	55.1%
V8	.086	51.3%
V17	.071	42.3%
V2	.029	17.6%
V6	.027	16.4%
V4	.027	16.4%
V5	.017	10.1%
V3	.014	8.2%

Growing Method: CRT
Dependent Variable: Letter

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter
	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17
	Validation	Split Sample
	Maximum Tree Depth	50
	Minimum Cases in Parent Node	180
	Minimum Cases in Child Node	90
Results	Independent Variables Included	V12, V8, V11, V7, V10, V14, V13, V15, V9, V5, V3, V2, V6, V17, V16, V4

	Number of Nodes	127
	Number of Terminal Nodes	64
	Depth	15

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.234	100.0%
V15	.191	81.8%
V12	.189	80.8%
V10	.163	69.6%
V7	.149	63.7%
V11	.149	63.5%
V13	.136	58.0%
V16	.132	56.5%
V9	.131	56.1%
V8	.131	56.0%
V17	.090	38.4%
V4	.049	21.0%
V6	.048	20.4%
V2	.042	18.1%
V3	.024	10.3%
V5	.024	10.2%

Growing Method: CRT
Dependent Variable: Letter

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter
	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17
	Validation	Split Sample
	Maximum Tree Depth	50
	Minimum Cases in Parent Node	160
	Minimum Cases in Child Node	80
Results	Independent Variables Included	V12, V8, V11, V7, V2, V4, V10, V14, V13, V15, V9, V5, V3, V16, V6, V17
	Number of Nodes	133
	Number of Terminal Nodes	67
	Depth	14

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.196	100.0%
V8	.157	79.9%

V11	.146	74.6%
V10	.145	74.2%
V15	.143	73.1%
V16	.138	70.3%
V7	.137	70.0%
V9	.122	62.3%
V13	.119	60.5%
V12	.107	54.3%
V17	.072	36.5%
V4	.044	22.5%
V6	.040	20.1%
V2	.037	19.1%
V3	.024	12.5%
V5	.023	11.8%

Growing Method: CRT
Dependent Variable: Letter

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter
	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17
	Validation	Split Sample
	Maximum Tree Depth	50
	Minimum Cases in Parent Node	140

	Minimum Cases in Child Node	70
Results	Independent Variables Included	V12, V8, V11, V7, V10, V14, V13, V15, V9, V5, V3, V16, V2, V17, V6, V4
	Number of Nodes	139
	Number of Terminal Nodes	70
	Depth	12

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.199	100.0%
V12	.176	88.4%
V13	.176	88.1%
V10	.170	85.0%
V7	.166	83.1%
V16	.163	81.5%
V15	.162	81.5%
V11	.159	79.9%
V9	.137	68.5%
V8	.116	58.0%
V17	.112	56.2%
V6	.044	22.2%
V2	.040	20.0%
V4	.037	18.6%

V5	.028	13.9%
V3	.024	11.9%

Growing Method: CRT
Dependent Variable: Letter

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter
	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17
	Validation	Split Sample
	Maximum Tree Depth	50
	Minimum Cases in Parent Node	120
	Minimum Cases in Child Node	60
Results	Independent Variables Included	V12, V8, V11, V7, V4, V2, V10, V14, V13, V15, V9, V5, V3, V6, V17, V16
	Number of Nodes	167
	Number of Terminal Nodes	84
	Depth	13

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.253	100.0%
V12	.185	73.1%
V15	.177	70.0%
V10	.174	68.8%
V7	.161	63.6%
V11	.157	62.3%
V13	.151	59.7%
V16	.145	57.5%
V9	.135	53.4%
V8	.127	50.5%
V17	.087	34.6%
V4	.056	22.1%
V2	.051	20.2%
V6	.051	20.0%
V3	.031	12.2%
V5	.030	11.9%

Growing Method: CRT
Dependent Variable: Letter

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter

	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17
	Validation	Split Sample
	Maximum Tree Depth	50
	Minimum Cases in Parent Node	100
	Minimum Cases in Child Node	50
Results	Independent Variables Included	V12, V8, V11, V7, V10, V14, V13, V15, V9, V5, V3, V2, V6, V16, V17, V4
	Number of Nodes	201
	Number of Terminal Nodes	101
	Depth	16

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.264	100.0%
V12	.205	77.6%
V11	.195	74.1%
V10	.185	70.1%
V15	.180	68.3%
V16	.174	66.2%

V9	.155	59.0%
V7	.153	58.1%
V13	.146	55.5%
V8	.137	52.2%
V17	.114	43.4%
V6	.064	24.2%
V2	.063	24.0%
V4	.063	23.8%
V3	.040	15.1%
V5	.033	12.7%

Growing Method: CRT
Dependent Variable: Letter

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Letter
	Independent Variables	V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17
	Validation	Split Sample
	Maximum Tree Depth	50
	Minimum Cases in Parent Node	20
	Minimum Cases in Child Node	10

Results	Independent Variables Included	V12, V8, V11, V7, V10, V14, V13, V15, V9, V5, V3, V2, V16, V6, V17, V4
	Number of Nodes	695
	Number of Terminal Nodes	348
	Depth	21

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
V14	.329	100.0%
V12	.257	78.1%
V11	.255	77.5%
V15	.234	71.1%
V10	.233	70.9%
V16	.222	67.4%
V7	.211	64.2%
V9	.210	63.7%
V13	.195	59.1%
V8	.172	52.3%
V17	.164	49.7%
V6	.098	29.7%
V2	.091	27.8%
V4	.088	26.8%
V3	.070	21.4%

V5	.062	18.8%

Growing Method: CRT
Dependent Variable: Letter