**ST4061 – Computer Intensive Statistical Analytics II**

2022-23

Continuous Assessment 2

**Question 1**

Figure 1 below shows the output of a random forest model fit to a sample of data points. This dataset comprises of 5 variables: Length, Width, Leaf, Curve and Age (Young; Intermediate; Mature).



Figure 1 – Diagram for Question 1(a).

(1) Indicate which of the 5 variables is the response variable Y. Briefly justify your answer.

(2) Quote the number of nodes that would be required in each layer of the architecture of a fully connected, single-hidden-layer, feed-forward neural network with 5 hidden neurons, for that model to be applicable to this dataset. Briefly justify your answer.

(3) How many model coefficients would need to be estimated in total, in order to fit the neural network described in (2) to this dataset? Briefly justify your answer.

**Answers to Question 1**

|  |  |
| --- | --- |
| **Question** | **Your answer** |
| **1** | The Age variable is the response variable. The categories for this variable (Young, Intermediate, Mature) appear in the terminal nodes (leaves) of the tree indicating that this is the variable being predicted. |
| **2** | Input layer: 4 nodes  (Assuming all variables listed are included in the model)  The input layer has a node for each predictor variable in the model. In this case, we have four predictor variables: Length, Width, Leaf and Curve  Hidden layer: 5 nodes  This was given in the question. The number of nodes in the hidden layers can be selected by the user.  Output layer: 3 nodes  As this is a multi-class classification problem there must be a node for each possible class. Each node will output the probability that the input belongs to that specific class. This results in a probability distribution of the classes. In this case, the classes being predicted are Young, Intermediate and Mature. |
| **3** | Weights: 4\*5 + 5\*3 = 35  Weights are required to connect all of the neurons in subsequent layers. For the first two layers, 20 weights are required and for the last two layers, 15 weights are required.  Biases: 5 + 3= 8  A bias term is added to all neurons after the input layer. Therefore 5 are needed for the hidden layer and 3 are needed for the output layer.  Total coefficients to be estimated: 43  This is the total number of parameters to be estimated during training. |

**Question 2**

You are required to use the **caret** package for the entire question.

Load the following datasets into your R session:

X = read.csv(file="Q2\_X.csv", header=TRUE)

Y = read.csv(file="Q2\_Y.csv", header=FALSE, stringsAsFactors=TRUE)[,1]

X.valid = read.csv(file="Q2\_Xvalid.csv", header=TRUE)

Y.valid = read.csv(file="Q2\_Yvalid.csv", header=FALSE,

stringsAsFactors=TRUE)[,1]

The eight measurements in **X** and **X.valid** correspond to percentages by weight of oxides contained in a sample of glass fragments. These glass fragments were classed into six types. The aim here is to build a methodology for classification of glass fragments on the basis of concentration of these eight oxides (Na: sodium, Mg: manganese, Al: aluminium, Si: silicon, K: potassium, Ca: calcium, Ba: barium, Fe: iron).

**Note: do** **not** apply scaling to the set of predictors or the response variable **Y**.

1. Train a kNN classifier on the data (**X**,**Y**) via single 5-fold cross-validation using the appropriate function from the **caret** package with default settings (other than those specified here for cross-validation). Run set.seed(6041) before generating your output. Quote:
   1. the optimal value of *k* found via CV;
   2. average test-set prediction accuracy for this classifier.
2. Train a random forest with recursive feature elimination via single 5-fold cross-validation as in (a) on the (**X**,**Y**) dataset. Run set.seed(6041) before generating your output. Quote:
   1. The set of optimal predictors;
   2. the cross-validated accuracy corresponding to the best predictor subset found for the random forest.
3. Generate predictions for the validation set by applying the random forest trained in (2) on **X.valid**. Quote the validation set prediction accuracy for this classifier.
4. Train a single hidden-layer neural network using method **nnet** on the (**X**,**Y**) dataset via single 5-fold cross-validation using the appropriate function from the **caret** package. Run set.seed(6041) before generating your output. Generate predictions for the test data **X.valid** from that fit. Quote:
   1. the size of the input layer for the optimal architecture for this classifier;
   2. the number of neurons in the hidden layer for the optimal architecture for this classifier;
   3. the validation set prediction accuracy for this classifier.

**Answers to Question 2**

|  |  |
| --- | --- |
| **Question** | **Your answer** |
| **1(a)** | k = 5 |
| **1(b)** | Training accuracy = 0.6764037 |
| **2(a)** | Mg, Al, Ca, Ba, Na, K |
| **2(b)** | Training accuracy = 0.7443998 |
| **3** | Validation accuracy = 0.7 |
| **4(a)** | Size of input layer = 8 neurons (8 predictor variables) |
| **4(b)** | Size of hidden layer = 5 neurons |
| **4(c)** | Validation accuracy = 0.62 |

**R code for Question 2:**

library(caret)

X = read.csv(file="C:\\Users\\zachb\\OneDrive\\Documents\\Machine Learning and Math Modelling\\Semester 2\\ST4061 - Statistical Methods for Machine Learning II\\CA2\\Q2\_X.csv", header=TRUE)

Y = read.csv(file="C:\\Users\\zachb\\OneDrive\\Documents\\Machine Learning and Math Modelling\\Semester 2\\ST4061 - Statistical Methods for Machine Learning II\\CA2\\Q2\_Y.csv", header=FALSE, stringsAsFactors=TRUE)[,1]

X.valid = read.csv(file="C:\\Users\\zachb\\OneDrive\\Documents\\Machine Learning and Math Modelling\\Semester 2\\ST4061 - Statistical Methods for Machine Learning II\\CA2\\Q2\_Xvalid.csv", header=TRUE)

Y.valid = read.csv(file="C:\\Users\\zachb\\OneDrive\\Documents\\Machine Learning and Math Modelling\\Semester 2\\ST4061 - Statistical Methods for Machine Learning II\\CA2\\Q2\_Yvalid.csv", header=FALSE, stringsAsFactors=TRUE)[,1]

#View the data

head(X)

dim(X)

head(Y)

length(Y)

#X has 8 predictors and 164 records

#There are six levels for response variable Y

length(Y.valid)

#Validation set has 50 entries

#(1)

set.seed(6041)

#Set 5 fold CV for train function

trC = trainControl(method="cv", number=5)

#train knn models with 5 fold CV to find optimal k value

knn.o = train(X, Y, method='knn', trControl=trC)

#(a)

#Best model:

knn.o$bestTune

#K=5

#(b)

#Accuracy on training data

knn.o$results$Accuracy[1]

#0.6764037

knn.o$finalModel

#Predict validation data

knn.p <- predict(knn.o, X.valid)

#Create table

(tb = table(Y.valid,knn.p))

#Find accuracy on test data

(acc = sum(diag(tb)) / sum(tb))

#0.66

#(2)

set.seed(6041)

subsets <- c(1:8)#Look at sizes 1 to 8

#Set rfeControl with CV and random forest

rfeC <- rfeControl(functions = rfFuncs, method = "cv",number = 5)

#Run rfe

rf.rfe <- rfe(X,Y,sizes = subsets,rfeControl = rfeC)

#View results

rf.rfe

#Plot accuracy for each number of variables

plot(rf.rfe, type=c("g", "o"), colour = "r",ylim=c(0,1))

#Clearly the model with 6 variables performs the best.

#(a)

#View predictors

rf.rfe$optVariables

#"Mg" "Al" "Ca" "Ba" "Na" "K"

#(b)

#View CV accuracy

max\_acc\_idx <- which.max(rf.rfe$results$Accuracy)

rf.rfe$results$Accuracy[max\_acc\_idx]

#Accuracy = 0.7443998

#(3)

#Predict validation data

rf.p = predict(rf.rfe$fit, X.valid )

#Create table

(tb = table(Y.valid,rf.p))

#Find accuracy on test data

(acc = sum(diag(tb)) / sum(tb))

#Accuracy = 0.7

#(4)

set.seed(6041)

#Set 5 fold CV for train function

trC.nnet = trainControl(method="cv", number=5)

#train neural network models with 5 fold CV to find optimal hidden layer size

nnet.o = train(X, Y, method='nnet', trControl=trC.nnet)

nnet.o

#(a)

#8 predictors used => 8 predictors in the input layer

#(b)

#Find best hidden layer size

nnet.o$bestTune$size

#Hidden layer size = 5

nnet.p = predict(nnet.o, X.valid )

#Create table

(tb = table(Y.valid,nnet.p))

#Find accuracy on test data

(acc = sum(diag(tb)) / sum(tb))

#Accuracy = 0.62

**Question 3**

Consider the problem of performing Principle Components Analysis on the R dataset **mtcars**.

1. Should one perform a PCA on the scaled or unscaled data? Briefly justify your answer
2. Remove variable **mpg** from the dataset, and perform PCA on the rest of the data (without any data splitting). Compare the first eigenvector from each analysis, and briefly comment on the values contained in these vectors in terms of the role of each feature. What causes the difference, and why?
3. Based on the PCA of the scaled data, indicate which 5 variables seem to capture most of the information. Briefly justify your answer.
4. Fit a linear regression model using these 5 variables as predictors X, and **mpg** as the dependent variable Y. Are any of these variables significant predictors? Briefly explain why they are or are not.

**Answers to Question 3**

|  |  |
| --- | --- |
| **Question** | **Your answer** |
| **1** | PCA should be performed on scaled data. PCA finds the direction of maximum variance in data. If the data is left unscaled, the variables that have large scales will have a larger effect on the principle components. Scaling the variables puts them all on the same scale, giving them zero mean and unit variance, and allowing them to be compared. |
| **2** | Eigenvectors for unscaled data:  cyl: 0.012042615, disp: 0.900235270, hp: 0.435074057, drat: -0.002661394, wt: 0.006242550, qsec: -0.006676533, vs: -0.002731293, am: -0.001963245, gear: -0.002606103, carb: 0.005767541  Eigenvectors for scaled data:  cyl: 0.4029711, disp: 0.3959243, hp: 0.3543255, drat: -0.3155948,  wt: 0.3668004, qsec: -0.2198982, vs: -0.3333571, am: -0.2474991,  gear: -0.2214375, carb: 0.2267080  Different variables are considered important in each of the results. When PCA was performed on the unscaled data the disp and hp features dominate the analysis with values of 0.900 and 0.435 respectively. This is because these variables are measured on much larger scales than the others and therefore have higher variance.  After the data is scaled, each feature will have equal weighting and variance leading to a more balanced analysis. In this example, each variable contributes more evenly to the first PC after scaling has been done. |
| **3** | From the histogram of variance explained we can see that the first principle component accounts for the majority of the variance explained (57.6%) from the PCA on the scaled data. From this first PC the five variables with the highest absolute loading are cyl, disp, wt, hp and vs. These were chosen as the most important variables. |
| **4** | wt was the only variable found to be significant with a p-value of 0.00101. wt has a coefficient of -3.88719 meaning it has a negative relationship with mpg.  All other variables (cyl, disp, hp and vs) were not statistically significant as they had p-value greater than 0.05. This shows that while PCA may be useful for identifying variables that capture the most variation in the data, it does not guarantee that these variables will be significant predictors for regression. |

**R code for Question 3:**

#(2)

# Load dataset

data(mtcars)

#View dataset

head(mtcars)

dim(mtcars)

# Remove mpg from the dataset

X <- mtcars[, -1]

Y <- mtcars[, 1]

# Perform PCA on the unscaled data

pca\_unscaled <- prcomp(X, scale = FALSE)

# Perform PCA on the scaled data

pca\_scaled <- prcomp(X, scale = TRUE)

#Get eigenvectors

R\_unscaled = pca\_unscaled$rotation

#Get eigenvectors

R\_scaled = pca\_scaled$rotation

#Print first eigenvectors for each case

R\_unscaled[,1]

#cyl disp hp drat wt qsec vs am gear carb

#0.012042615 0.900235270 0.435074057 -0.002661394 0.006242550 -0.006676533 -0.002731293 -0.001963245 -0.002606103 0.005767541

#names(sort(abs(R\_unscaled[,1]), decreasing = TRUE)[1:8])

R\_scaled[,1]

#cyl disp hp drat wt qsec vs am gear carb

#0.4029711 0.3959243 0.3543255 -0.3155948 0.3668004 -0.2198982 -0.3333571 -0.2474991 -0.2214375 0.2267080

#names(sort(abs(R\_scaled[,1]), decreasing = TRUE)[1:8])

#(3)

# View the results

summary(pca\_scaled)

plot(pca\_scaled)

#The first PC accounts for over 57.6% of the total variance explained.

R\_scaled[,1]

#View the names of the five most important variables for the first PC

rownames(R\_scaled)[order(abs(R\_scaled)[,1],decreasing = TRUE)][1:5]

#"cyl" "disp" "wt" "hp" "vs"

#The highest five loadings on PC1 are from cyl, disp, wt, hp and vs

#(4)

#Fit regression model

lm.o = lm(mpg ~ cyl+disp+wt+hp+vs, data = mtcars)

summary(lm.o)

#wt is a significant predictor. The others are not.