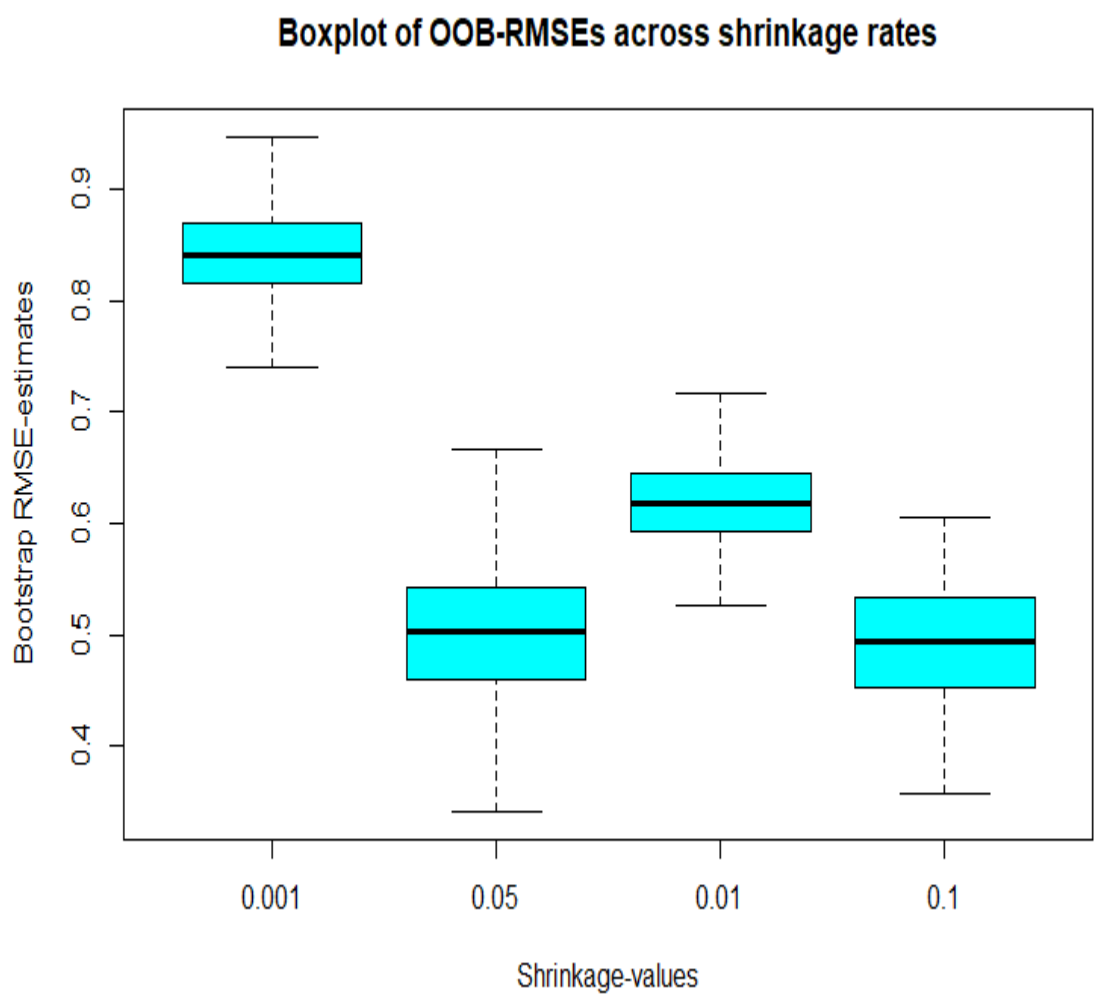
ST4061 – Statistical Methods for Machine Learning II
ST6041 – Machine Learning and Statistical Analytics II

2023-24
Continuous Assessment 2

Answers to Question 1

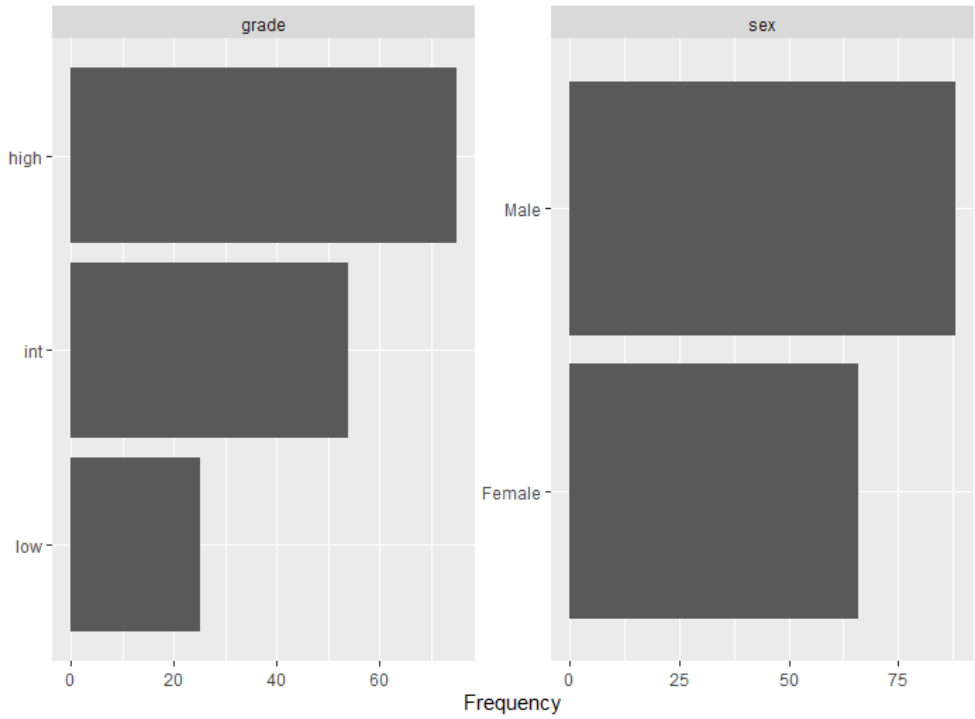
Question	Your answer
1	Mean OOB RMSE for 0.001 shrinkage rate – 0.8414 Mean OOB RMSE for 0.05 shrinkage rate – 0.5053 Mean OOB RMSE for 0.01 shrinkage rate – 0.6195 Mean OOB RMSE for 0.1 shrinkage rate – 0.4933
2	

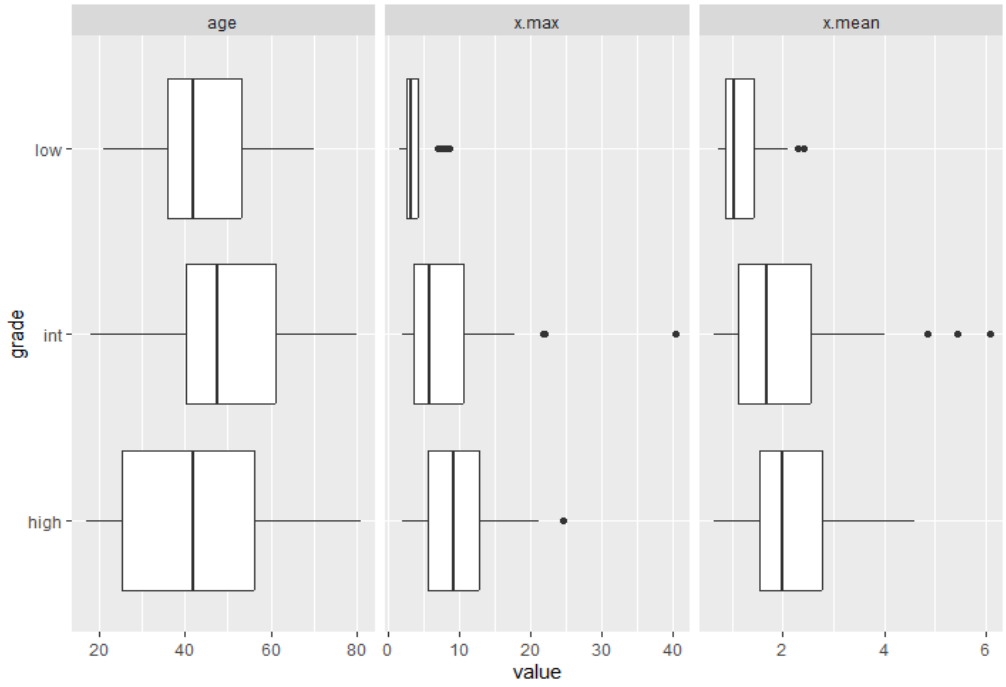
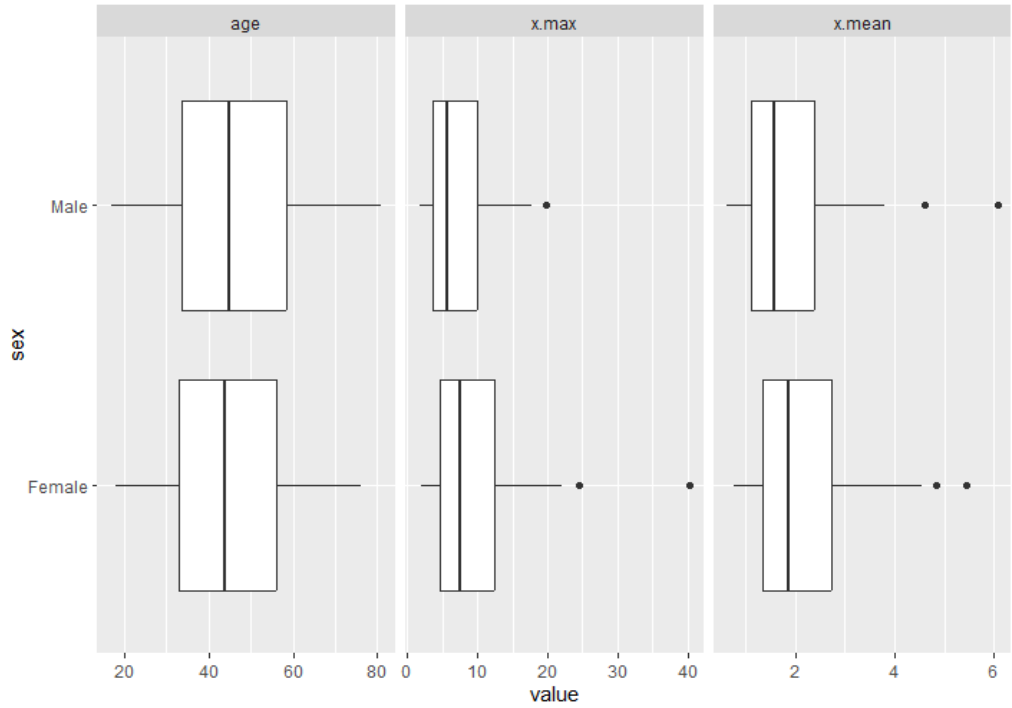
3	Shrinkage parameter recommended for training this model (with 100 trees) is 0.1. The mean OOB RMSE for models with shrinkage parameters 0.05 and 0.1 is very similar (0.5053 & 0.4933 respectively), however the spread of the OOB RMSEs is lower for the model with 0.1 as shrinkage parameter compared to that of 0.05 indicating better consistency. Since, the number of trees is not too high over here, a higher shrinkage rate (0.1) has resulted in a better training making it the final choice.

R code for Question 1

```
## Question 1
rm(list=ls())
require(gbm)
require(ISLR)
df = na.omit(Hitters)
df$Salary = log(df$Salary)
rates = c(0.001, 0.05, 0.01, 0.1)
L = length(rates)
set.seed(4061)

n = nrow(df)
B = 100

OOB_RMSEs = matrix(NA, nrow = B, ncol = L)
colnames(OOB_RMSEs) = rates
for (i in 1:L) {
  for (j in 1:B){
    idxs = sample(1:n, n, replace=TRUE)
    X_train = df[idxs,]
    X_test = df[-idxs,]
    Y_test = df[-idxs,]$Salary
    gbm_tree = gbm(Salary~., data=X_train, distribution = 'gaussian', shrinkage = rates[i])
    test_preds = predict(gbm_tree, X_test)
    OOB_RMSEs[j, i] = sqrt(mean((test_preds - Y_test)^2))
  }
}
head(OOB_RMSEs)

## Question 1(1)
OOB_RMSEs_mean = apply(OOB_RMSEs, 2, mean)
round(OOB_RMSEs_mean, 4)

## Question 1(2)
par(mfrow=c(1,1))
boxplot(OOB_RMSEs, main="Boxplot of OOB-RMSEs across shrinkage
rates",xlab="Shrinkage-values",ylab="Bootstrap RMSE-estimates", col = 'cyan')
```

Answers to Question 2

Question	Your answer
1	"Sex" is the only categorical feature here with 2 levels ("Female", "Male"). Grade is also a categorical column, however it's not a feature here but the outcome (labels).
2	 There are more number of Males (approx. 88) compared to the number of females (approx. 65) in the data indicating a gender imbalance in the data present. Also, there is a clear imbalance in the data recorded for the different grade of tumours with category "high" dominating and "low" having the least amount of samples present.

3	 Overall, the values for x.max increase with change from grades from low to int to high respectively. A similar trend is observed in x.mean across grades as well. There is a wide age distribution for "high" grade tumors, whereas for "int" and "low" grade most number of samples are in the age group of 40-60 and 30-50 approximately respectively.
4	

<table>
<tr><td></td><td>The distribution of age for both males and females are very similar to each other with a median age value slightly higher for males. Overall, the values for x.max column is slightly on the higher side for females compared to males with both having wider distribution and a higher median value. For the x.mean column, values are right skewed for both females and males with a wider distribution for females and a higher median value compared to males. Highest x.mean value is for a male (approx.. 6), which is an outlier according to the graph.
However, one important factor to remember here is the number of data points for males and females is NOT the same as the data consists of more males recorded than females indicating a clear gender imbalance.</td></tr>
<tr><td>5</td><td>p-value for Wilcoxon test between Age and Sex – 0.7259
p-value for Wilcoxon test between Age and X.Max – 0.07627
p-value for Wilcoxon test between Age and X.Mean – 0.07691

The Wilcoxon test between Age and Sex yielded a p-value of >0.05 meaning there is no statistically significant difference between distribution of age of males and that of females. (Fail to reject the Null hypothesis)
The Wilcoxon test between Age and X.Max yielded a p-value of >0.05 meaning there is no statistically significant difference between distribution of X.Max of males and that of females. (Fail to reject the Null hypothesis)
The Wilcoxon test between Age and X.Mean yielded a p-value of >0.05 meaning there is no statistically significant difference between distribution of X.Mean of males and that of females. (Fail to reject the Null hypothesis)</td></tr>
<tr><td>6</td><td>Mean of Age after Min-Max Scaling – 0.4465
Mean of X.Mean after Min-Max Scaling – 0.2477
Mean of X.Max after Min-Max Scaling – 0.1655</td></tr>
<tr><td>7</td><td>Overall Error quoted by neuralnet function used - 7.543743 (seed set as 4061)</td></tr>
<tr><td>8</td><td>(i) Overall Accuracy – 0.92208 (92.208%)

(ii) Class-wise Sensitivity for this fit –

<table><tr><th>“high”</th><th>“int”</th><th>“low”</th></tr><tr><td>0.9733
(97.33%)</td><td>0.963
(96.3%)</td><td>0.68
(68%)</td></tr></table></td></tr>
<tr><td>9</td><td>26 columns identified by the correlation filter to remove with 0.95 as cutoff –

['compactness1', 'compactness2', 'sphericity', 'l.major', 'major.axis.length', 'l.minor', 'minor.axis.length', 'l.least', 'least.axis.length', 'RMS', 'mean_HIST', 'sum.avg_GLCM', 'auto.corr_GLCM', 'var_HIST', 'joint.var_GLCM', 'sum.var_GLCM', 'joint.max_GLCM', 'energy_GLCM', 'entropy_GLCM', 'homogeneity_GLCM', 'inv.diff.mom_GLCM', 'diff.entropy_GLCM', 'dissimilarity_GLCM', 'homogeneity.norm_GLCM', 'contrast_GLCM', 'inv.diff.mom.norm_GLCM']</td></tr>
</table>

R code for Question 2

```
## Question 2
require(caret)
require(neuralnet)
require(DataExplorer)
df = read.csv(file="uws.csv", stringsAsFactors=TRUE)
subdf = df[,c("grade","sex","age","x.mean","x.max")]
y = df$grade
x = df
x$grade = NULL

## Question 2(1)
## sex is a categorical feature
str(df)

## Question 2(2)
plot_bar(df)

## Question 2(3)
plot_boxplot(subdf, by = 'grade')

## Question 2(4)
plot_boxplot(subdf, by = 'sex')

## Question 2(5)
age = subdf$age
sex = subdf$sex
x_max = subdf$x.max
x_mean = subdf$x.mean

wilcox.test(age ~ sex, alternative = "two.sided")
wilcox.test(x_max ~ sex, alternative = "two.sided")
wilcox.test(x_mean ~ sex, alternative = "two.sided")

## Question 2(6)
conversion <- function(x){
  # function recoding levels into numerical values
  if(is.factor(x)){
    levels(x)
    return(as.numeric(x))
  } else {
    return(x)
```

```
  }
}
scaling <- function(x){
  # function applying normalization to [0,1] scale
  mins = min(x,na.rm=TRUE)
  maxs = max(x,na.rm=TRUE)
  return((x-mins)/(maxs-mins))
}
df_inter = data.frame(lapply(df,conversion))
df_scaled = data.frame(lapply(df_inter,scaling))
means = apply(df_scaled[, c("age", "x.mean", "x.max")], 2, mean)
round(means, 4)
df_scaled$grade = NULL

## Question 2(7)
set.seed(4061)
mod = neuralnet(y~., data = df_scaled, hidden=c(5), linear.output = FALSE)
(error = mod$result.matrix["error",])

## Question 2(8)
col_names = colnames(mod$response)
final_preds = as.factor(col_names[max.col(predict(mod, df_scaled))])
cf_mat = caret::confusionMatrix(final_preds, y)
(overall_accuracy = cf_mat$overall[1])
specificity_class = cbind(cf_mat$byClass[1],cf_mat$byClass[2],cf_mat$byClass[3])
colnames(specificity_class) = c("high","int","low")
round(specificity_class, 4)

## Question 2(9)
x$sex = as.numeric(x$sex)
correlation_matrix <- cor(x)
cols = colnames(x)
pairs = c()
for (i in cols) {
  for (j in cols) {
    if (i!=j & abs(correlation_matrix[i,j]) > 0.95) {
      pairs = c(pairs, c(i,j))
    }
  }
}
cat("Columns to remove : ", unique(pairs))
unique(pairs)
```