BUS 641 Test 3

This test will assess the R skills and statistical concepts you have covered in Modules 7, 8, and 9. You will continue to use the same data here that you used in Tests 1 and 2.

Instructions:

You have been provided with an Excel workbook containing data to help you complete the following questions. You must create an R script showing the code that you write to read the indicated sheet into a dataframe along with any code you need to write to answer the questions.

Type or paste your answers from R into this document. When you generate output or a graph in R to answer a question, paste a snip of the output or graph into this document at that question.

You must submit both your R script and this document with the question answers typed/pasted into it.

Your grade will depend in part on your ability to write the necessary code in R to answer the questions. I will try to run your code to reproduce the results you paste into this document. If your code will not run, you will not get credit for what you have entered here. In your code, please add comments to indicate which question you are answering in that section of code.

1. Use the **TechSales_Reps** sheet of the provided Excel workbook, BigDataFiles.xlsx. Read the data into a dataframe in R. The code for this step must appear in your script. See the data dictionary to help you understand the data and the meaning of each variable. This can be found in Appendix A of your textbook. Look for the description of this dataset titled: Data 7: Tech Sales Reps Data. Also read more about this case in section 2.6 of your textbook so you can understand why the variable NPS (Net Promoter Score) is important. The Sales_Rep variable is simply an identifier/index, so you do not need to include it in the following steps.

2. Reduce the dimensionality of the data by converting **numerical** (quantitative) variables such as age, years with the company, number of certifications, etc into a smaller set of principal components that retain at least 85% of the information in the original data. Then use the identified principal components as predictor variables for building models for predicting the net promoter score (NPS). Apply any transformations that are necessary so that your data is appropriate for this machine learning technique.

a. Paste the R summary of your PCA results and PC weights here:

PCA results

```
> print(pca_summary)
Importance of components:
                          PC1    PC2    PC3    PC4    PC5     PC6     PC7
Standard deviation     1.4658 1.0453 1.0142 1.0019 0.9795 0.65206 0.58465
Proportion of Variance 0.3069 0.1561 0.1469 0.1434 0.1371 0.06074 0.04883
Cumulative Proportion  0.3069 0.4630 0.6100 0.7534 0.8904 0.95117 1.00000
```

PC weights

```
> print(pca_result$rotation[, 1:num_components])
                    PC1        PC2        PC3         PC4         PC5
Age           0.15271058  0.7146264 -0.1199836 -0.02642862 -0.59989968
Female       -0.07442073 -0.4871693 -0.6243128 -0.02770475 -0.57148133
Years         0.15208643  0.2685269 -0.6668554  0.40949251  0.47687756
Certificates  0.45109120 -0.3440266  0.2805752  0.39672158 -0.18051240
Feedback      0.29284098 -0.0665229 -0.1946446 -0.81528817  0.20324531
Salary        0.58637086  0.1546236  0.1317556 -0.06519438 -0.07780198
NPS           0.56119592 -0.1821549 -0.1310453  0.06720670  0.07855400
```

b. Now, for the principal components that account for at least 85% of the covariance of the predictors, use their principal component scores to predict NPS. Use least squares regression to perform PCR (principal components regression). Paste the R summary of your regression model output here:

```
> print(pcr_summary)

Call:
lm(formula = NPS ~ ., data = pc_scores_df)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2213 -0.7533 -0.0174  0.7344  4.9227

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.278445   0.007424  845.74   <2e-16 ***
PC1          1.213908   0.005065  239.68   <2e-16 ***
PC2         -0.394015   0.007102  -55.48   <2e-16 ***
PC3         -0.283461   0.007320  -38.72   <2e-16 ***
PC4          0.145373   0.007410   19.62   <2e-16 ***
PC5          0.169918   0.007579   22.42   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.101 on 21984 degrees of freedom
Multiple R-squared:  0.7411,      Adjusted R-squared:  0.741
F-statistic: 1.258e+04 on 5 and 21984 DF,  p-value: < 2.2e-16
```

c. What percentage of the variation in NPS is explained by the model with the principal components as predictors? 74.11%

d. Based on the value you identified in part ~~f~~ C and your answer to part ~~d~~ b, are PCA and PCR helpful in this instance? Is it better to use the principal components or just the original predictors to predict NPS? Explain your answers to these questions, stating any pros and/or cons that help you make your point.

   i. Yes, PCA and PCR are helpful in this instance
   ii. The use of the principal components is better to use to predict the NPS
   iii. Pros and Cons:
      1. Pros of Using Principal Components:
         a. Principal components are uncorrelated, which stabilizes the regression coefficients
         b. Simplification of the model through dimensionality reduction.
      2. Cons of Using Principal Components:
         a. Reduced interpretability of the results makes it harder to interpret the influence of individual variables.
         b. Although a significant portion of the variance is retained, some information may still be lost, which might be important for prediction accuracy in certain contexts.

3. Using all other variables as predictors, predict whether each sales professional will receive a high (>7) net promoter score (NPS) or not. Note that you may need to perform data transformation to meet the requirements of the analytics technique.

    a. Use the KNN technique to accomplish this. In R, partition the data sets into 60% training and 40% validation and implement 10-fold cross-validation. Use the statement set.seed(1) to specify the random seed prior to data partitioning *and* prior to cross-validation (applied in the kNN algorithm). When searching for the optimal value of *k*, search within possible *k* values from 1 to 25. In R use the createDataPartition, trainControl, expand.grid, and train functions as demonstrated in textbook examples. Paste a snip your kNN output here:

```
> print(knn_model)
k-Nearest Neighbors

13194 samples
   10 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 11875, 11875, 11875, 11874, 11874, 11875, ...
Resampling results across tuning parameters:

  k   RMSE       Rsquared   MAE
   1  0.3957822  0.4087832  0.1568518
   2  0.3554976  0.4581056  0.1790206
   3  0.3486724  0.4615855  0.1971421
   4  0.3480776  0.4570175  0.2111280
   5  0.3499938  0.4487247  0.2220151
   6  0.3525247  0.4399610  0.2311627
   7  0.3549327  0.4324264  0.2377865
   8  0.3578071  0.4234391  0.2441299
   9  0.3599521  0.4167539  0.2496261
  10  0.3623168  0.4093221  0.2545112
  11  0.3643976  0.4028741  0.2589553
  12  0.3663137  0.3970347  0.2627078
  13  0.3681283  0.3913641  0.2659573
  14  0.3702533  0.3844950  0.2693954
  15  0.3716342  0.3800408  0.2724650
  16  0.3733830  0.3744138  0.2752520
  17  0.3753048  0.3679902  0.2781791
  18  0.3765966  0.3635921  0.2805374
  19  0.3773269  0.3610652  0.2823812
  20  0.3780004  0.3589122  0.2840706
  21  0.3788231  0.3562480  0.2856927
  22  0.3797288  0.3531233  0.2873857
  23  0.3807292  0.3499432  0.2890324
  24  0.3818843  0.3459884  0.2907210
  25  0.3826946  0.3432648  0.2921609

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was k = 4.
>
```

b. List the optimal value of k, the associated accuracy, and the error (misclassification) rate for this value of k:

K=__4___
Accuracy=__0.950727449078565____
Misclassification rate= 0.0492725509214355

c. Generate the confusion matrix information (use the knnClass and confusionMatrix functions, as demonstrated in textbook examples) to assess performance of the kNN optimal k model on the validation set, and paste the result here:

```
> print(conf_matrix)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4138  167
         1   87  763

               Accuracy : 0.9507
                 95% CI : (0.9445, 0.9565)
    No Information Rate : 0.8196
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8276

 Mcnemar's Test P-Value : 7.163e-07

            Sensitivity : 0.9794
            Specificity : 0.8204
         Pos Pred Value : 0.9612
         Neg Pred Value : 0.8976
             Prevalence : 0.8196
         Detection Rate : 0.8027
   Detection Prevalence : 0.8351
      Balanced Accuracy : 0.8999

       'Positive' Class : 0
```

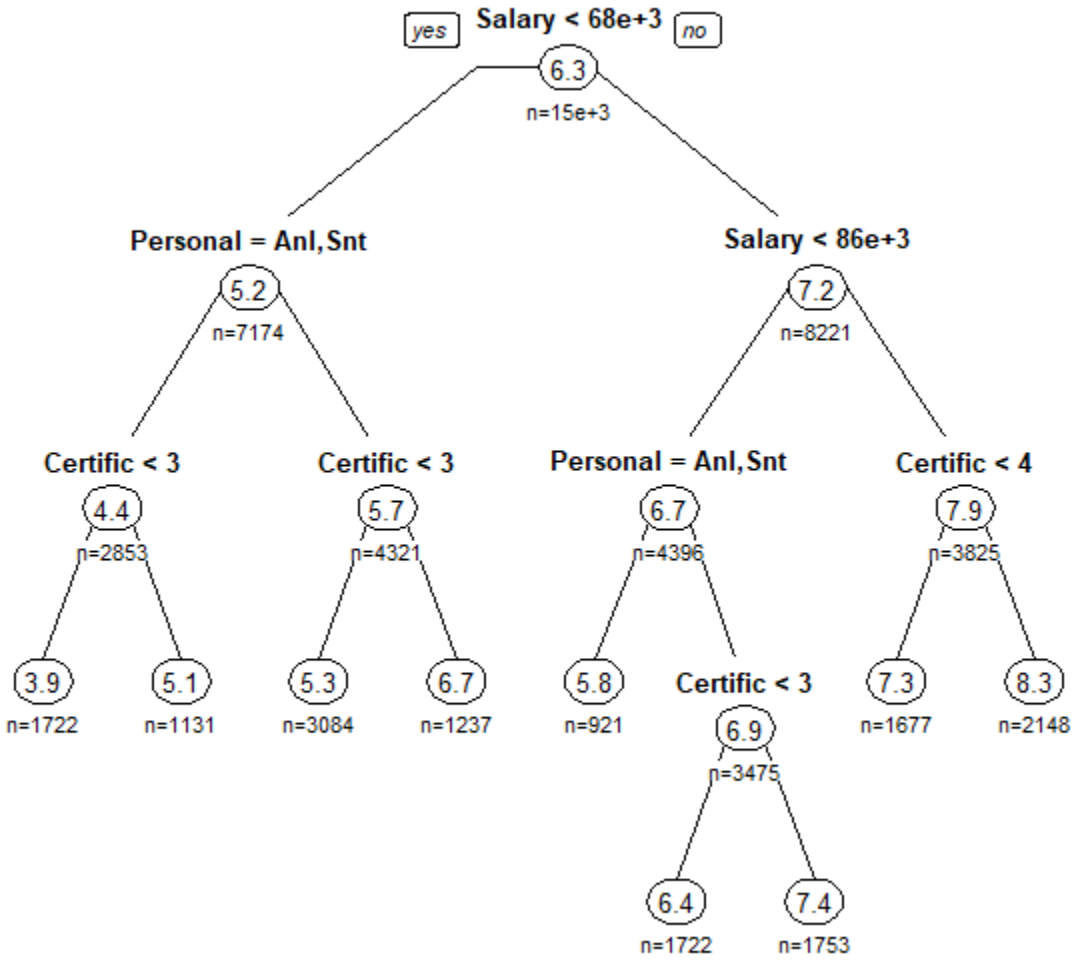
d. What is the performance of the naïve rule on the validation set? How does the kNN model performance compare to the naïve rule? Explain what this tells us.

- Naïve rule accuracy= 0.683037744429286
  - The kNN model accuracy of 0.950727449078565 (i.e., 95.07%) significantly outperforms the naïve rule accuracy of 0.683037744429286 (i.e., 68.30%). This indicates that the kNN model is effectively utilizing the features of the data to make more accurate predictions. The substantial improvement over the naïve rule demonstrates the benefit of using machine learning techniques for this classification task.

e. Based on the results of this kNN analysis, can you tell which predictors appear to be most important for predicting high NPS? Explain your answer.
    - kNN model's effective performance suggests that the predictors collectively contribute to accurate high NPS prediction. Since kNN uses the distance between data points, all scaled predictors contributed to the classification. The high accuracy (95.07%) indicates that the chosen predictors are relevant and useful for the task.
    - Although kNN does not provide a direct way to determine predictor importance, domain knowledge can offer insights into which features are likely to be significant. For instance, features like age, years with the company, number of certifications, feedback, and salary are expected to influence the NPS based on their roles in employee performance and satisfaction.

4. Develop a decision tree model to predict net promoter score (NPS), using all other variables as predictors. Note: You may need to perform data transformation to meet the requirements of the analytics technique.

    a. In R, partition data sets into 70% training and 30% validation. Use the statement set.seed(1) to specify the random seed of 1 for both data partitioning and cross-validation. Create a regression tree (use the **rpart** function to build a default regression tree, as demonstrated in textbook examples). Paste your resulting tree here:



b. How many leaf nodes/splits are in the default regression tree?
    i. Number of splits: 8
    ii. Number of leaf nodes: 9
c. List the 3 most important predictors for predicting NPS based on the default tree?
    i. Certificates
    ii. Salary
    iii. Personality

d. Report the validation set RMSE and MAPE for the default tree.
RMSE= 1.691873
MAPE= 26.20339

e. Use the **rpart** function to build a full-grown regression tree as demonstrated in textbook examples. Do not print the tree. What is the lowest cross-validation error produced by this (you can find this in the Cp table produced by running rpart)? How many splits are in the minimum error tree? What is the Cp value that is associated with the lowest cross-validation error?
    - Minimum cross-validation error= 0.5160309
    - Nsplits= 83
    - Cp= 0.000449567

f. Use the Cp table of the full tree to find the simplest tree (fewest splits) that has cross-validation error within one standard error of the cross-validation error of the minimum error tree. List the cross-validation error, Cp value, and number of splits for this simplest tree:
    - Minimum cross-validation error= 0.5210317
    - Nsplits= 62
    - Cp= 0.0006663948

g. Use the prune function to prune the full tree to the best pruned (simplest) tree you identified in part f. List the 3 most important predictors for predicting NPS based on the default tree:
    - Salary
    - Certificates
    - Personality

h. Report the validation set RMSE and MAPE for the best pruned tree you identified in f.
RMSE= 1.532653
MAPE= 23.14292

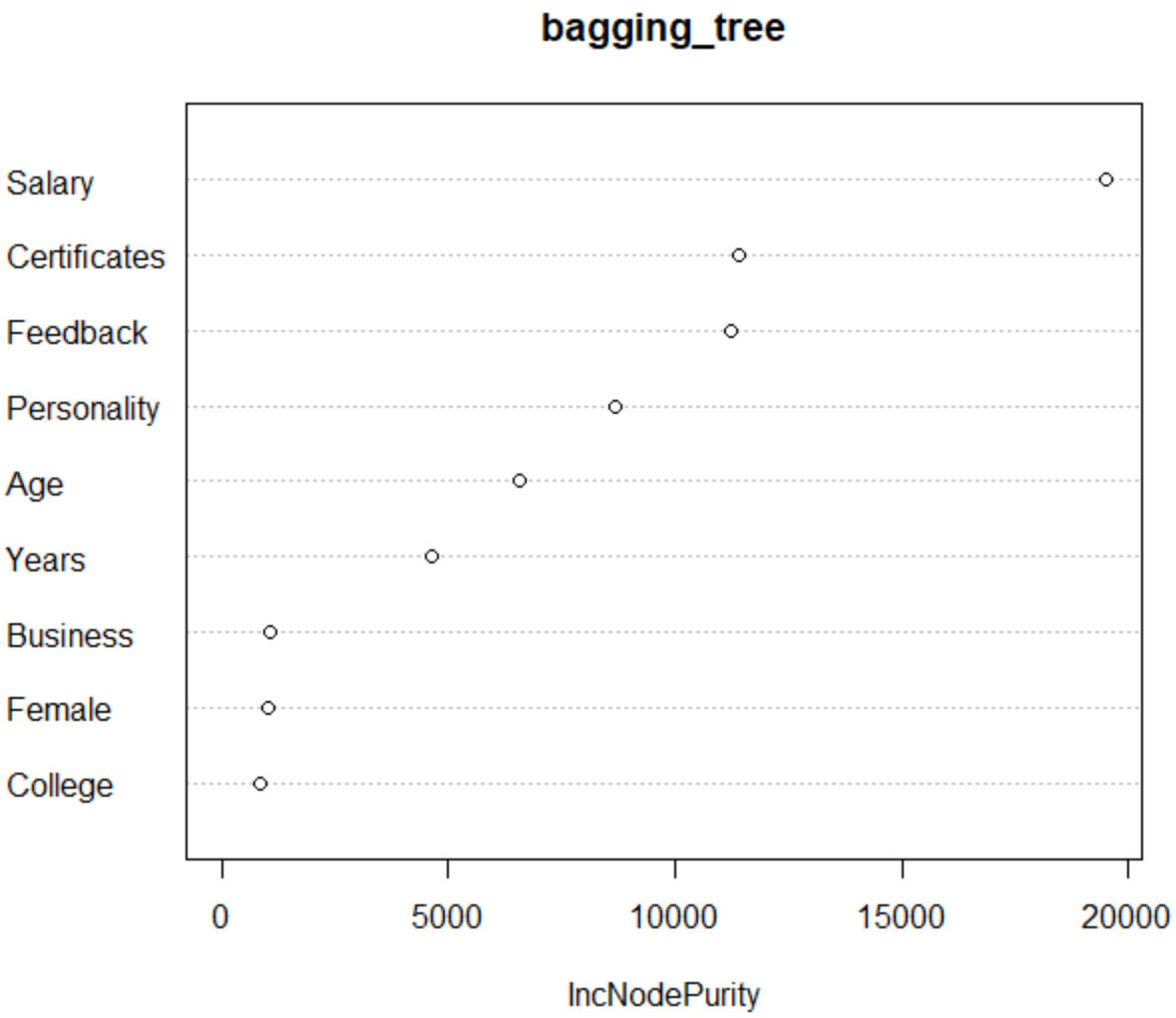
i. Complete the following table to help you compare the trees you created:

Tree	# Splits	Validation Set RMSE	Validation Set MAPE	Top 3 Important Predictors
Default	8	1.691873	26.20339	Certificates, Salary, Personality
Best Pruned	54	1.532653	23.14292	Salary, Certificates, Personality

j. Use the above table to compare and contrast the performance of the default tree with the best pruned tree. Is the increase in tree complexity worth the improvement in validation set accuracy? Consider the scale and context of NPS values you are trying to predict when looking at the RMSE values. Explain your answer.

- The best pruned tree is more accurate, with lower RMSE and MAPE compared to the default tree. However, the best pruned tree is much more complex, with 54 splits compared to the default tree's 8 splits. The pruned tree is more accurate but also more complex and harder to interpret. The slight improvement in accuracy may not justify the added complexity for practical use. The default tree, although simpler, is still reasonably accurate.

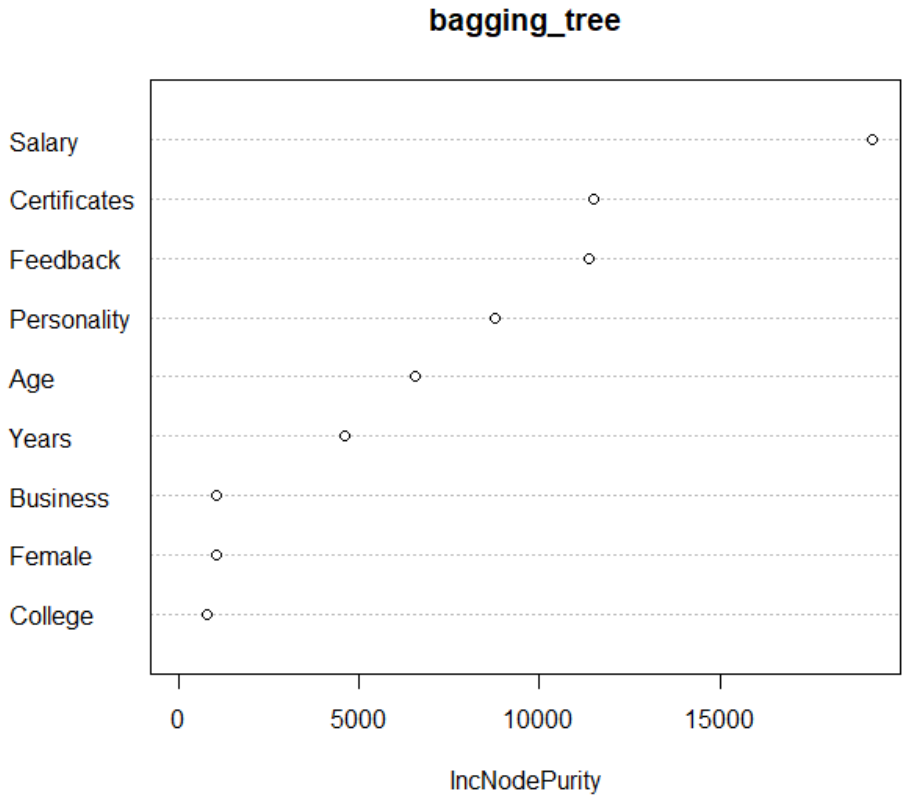
k. Use the randomForest function (as demonstrated in textbook examples) in R to construct an ensemble regression tree model using the **bagging** strategy. Create 100 weak learners when constructing the ensemble model. Construct a variance importance plot (use varImpPlot function) and paste your plot here:



l. Estimate the validation set RMSE and MAPE of the tree produced in part k.

RMSE= 1.074842
MAPE= 15.16483

m. Use the randomForest function (as demonstrated in textbook examples) in R to construct an ensemble regression tree model using the **random forest** strategy. Create 100 weak learners when constructing the ensemble model. Specify mtry=3. Construct a variance importance plot (use varImpPlot function) and paste your plot here:



n. Estimate the validation set RMSE and MAPE of the tree produced in part m.

RMSE= 1.075753

MAPE= 15.19228

o. Compare the results in parts k, l, m, and n with the table in part i.

- RMSE Comparison:
  - The bagging tree (k) and the random forest (mtry=3) (l) both have lower RMSE values (1.074842 and 1.075753 respectively) compared to the default (1.691873) and best pruned trees (1.532653). This indicates that both the bagging and random forest models are better at predicting the NPS on the validation set.
- MAPE Comparison:
  - Similarly, the MAPE values for the bagging tree (k) and random forest (mtry=3) (l) are lower (15.16483 and 15.19228 respectively) compared to the default (26.20339) and best pruned trees (23.14292). This further confirms that the ensemble methods (bagging and random forest) provide more accurate predictions.
- Top 3 Important Predictors:
  - The important predictors for the bagging tree and random forest models are slightly different from those in the default and best pruned trees. For the ensemble methods, "Feedback" appears as one of the top 3 predictors instead of "Personality."
- Conclusion
  - The ensemble methods (bagging and random forest) outperform the single decision tree models (default and best pruned) in terms of both RMSE and MAPE on the validation set. Additionally, while "Salary" and "Certificates" remain consistently important predictors across all models, "Feedback" emerges as a key predictor in the ensemble methods. This analysis suggests that ensemble methods should be preferred for this dataset to achieve better prediction accuracy.