REPORT FOR END OF TERM ASSESSMENT

BY A00288617

1. DECISION TREES

1.1 BUSINESS UNDERSTANDING

The name of this dataset is “Cervical Cancer Behaviour Risk”, downloaded from the UCI machine learning repository. (https://archive.ics.uci.edu/ml/datasets/Cervical+Cancer+Behavior+Risk). This datset contains 19 attributes regarding cervical cancer behaviour. The class label is “ca_cervix”, and it contains 1’s and 0’s. With “1” being positive(with cervical cancer) and “0” being negative(without cervical cancer). We want to try and predict the class of this data set(“ca_cervix”) using decision Trees. Therefore predicting whether the subject has cervical cancer or not.(This Business Understanding is valid for all Analysis carried out(Decision Trees, Knn and Kmeans))

1.2 DATA UNDERSTANDING AND PREPARATION

This data set contains 19 columns, The data set contains 18 attributes, they come from 8 variables which are shown in their column name as the first word. Each column has 71 rows. The total number of records are : 1440.

PREPARATION

For the preparation of the data set we had to shuffle the data and split the data into training data and test data sets. The full data set was split into two, with 50 rows in the training data set and 20 rows in the test data set. The class also had to be set to a factor because it wasn’t a string and wasn’t automatically set as a factor.

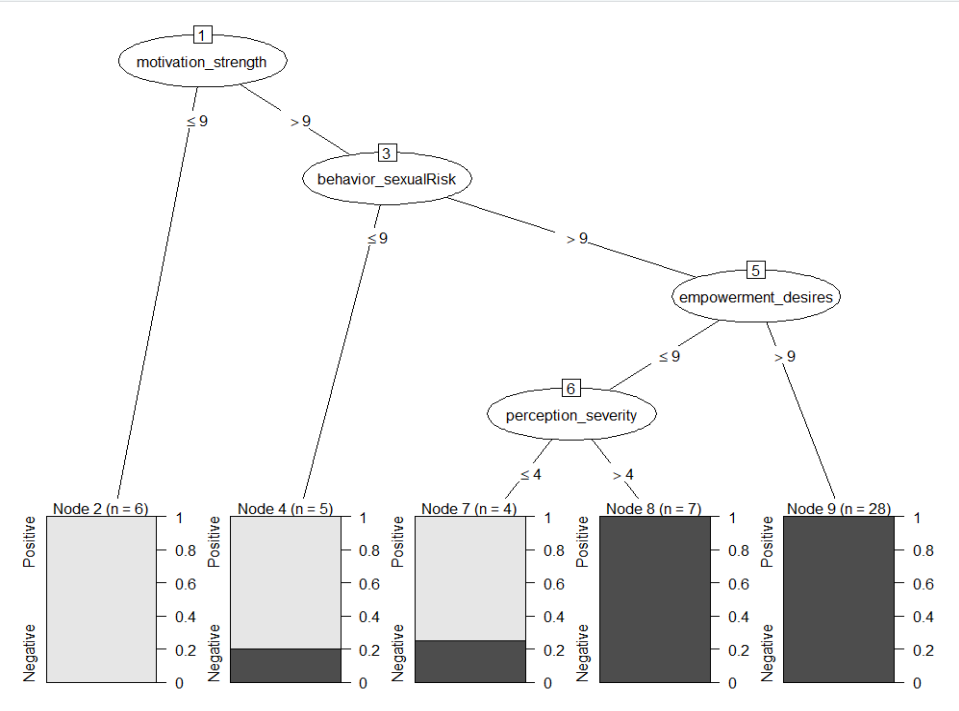
1.3 MODELLING

The model was created using the c50 library. The factor column and the train data are inputted into the function to create the model. Brief details on the model are checked and the model is plotted. Detailed information on the model is also checked using the “summary” function. After the model has been evaluated, we tried to boost the model using the trials parameter. However, even after 100 trails, there was no boost to the model.

Brief details on the Data set

```
> model

Call:
C5.0.formula(formula = ca_cervix ~ ., data = cervix_cancer_train)

Classification Tree
Number of samples: 50
Number of predictors: 19

Tree size: 5

Non-standard options: attempt to group attributes
```

Plotted Decision tree model



Detailed Information on the Data set

```
> summary(model)

Call:
C5.0.formula(formula = ca_cervix ~ ., data = cervix_cancer_train)


C5.0 [Release 2.07 GPL Edition]         Fri Dec 04 18:12:41 2020
-------------------------------

Class specified by attribute `outcome'

Read 50 cases (20 attributes) from undefined.data

Decision tree:

motivation_strength <= 9: Positive (6)
motivation_strength > 9:
:...behavior_sexualRisk <= 9: Positive (5/1)
    behavior_sexualRisk > 9:
    :...empowerment_desires > 9: Negative (28)
        empowerment_desires <= 9:
        :...perception_severity <= 4: Positive (4/1)
            perception_severity > 4: Negative (7)


Evaluation on training data (50 cases):

           Decision Tree
         ----------------
         Size      Errors

            5    2( 4.0%)   <<


          (a)   (b)    <-classified as
         ----  ----
           13          (a): class Positive
            2    35    (b): class Negative


        Attribute usage:

        100.00% motivation_strength
         88.00% behavior_sexualRisk
         78.00% empowerment_desires
         22.00% perception_severity


Time: 0.0 secs
```

Model state after 100 trials

```
  20          5    3( 6.0%)
  21          3    4( 8.0%)
  22          5    6(12.0%)
  23          4    3( 6.0%)
  24          4    6(12.0%)
  25          4    4( 8.0%)
  26          2   15(30.0%)
  27          5    2( 4.0%)
  28          5    3( 6.0%)
  29          3    3( 6.0%)
  30          4   11(22.0%)
  31          3    9(18.0%)
  32          5    1( 2.0%)
  33          3    4( 8.0%)
  34          3   10(20.0%)
  35          5    5(10.0%)
  36          6    2( 4.0%)
boost              0( 0.0%)    <<
```

1.4 EVALUATION

In conclusion, the model is pretty Accurate. This is because when the "predict" function was also used on this model it had a high prediction rate. The model was also predicted on the test data set which further verifies its accuracy. These are the results:

Cross-Table of Actual vs Predicted

```
   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  22

                  | actual default
predicted default |  Positive |  Negative | Row Total |
------------------|-----------|-----------|-----------|
         Positive |         8 |         2 |        10 |
                  |     0.364 |     0.091 |           |
------------------|-----------|-----------|-----------|
         Negative |         0 |        12 |        12 |
                  |     0.000 |     0.545 |           |
------------------|-----------|-----------|-----------|
     Column Total |         8 |        14 |        22 |
------------------|-----------|-----------|-----------|
```

Looking at the cross table, the model has 90 percent accuracy as 20 out of 22 were predicted correctly from the test data. 2 were predicted to be negative instead of positive and none was predicted to be positive instead of negative.

2. KNN

2.1 BUSINESS UNDERSTANDING

The name of this dataset is "Cervical Cancer Behaviour Risk", downloaded from the UCI machine learning repository. (https://archive.ics.uci.edu/ml/datasets/Cervical+Cancer+Behavior+Risk). This datset

contains 19 attributes regarding cervical cancer behaviour. The class label is “ca_cervix”, and it contains 1’s and 0’s. With “1” being positive(with cervical cancer) and “0” being negative(without cervical cancer). We want to try and predict the class of this data set(“ca_cervix”) using decision Trees. Therefore predicting whether the subject has cervical cancer or not.

2.2 DATA UNDERSTANDING AND PREPARATION

DATA UNDERSTANDING

Regarding the data set, the table of the variable to be used as a factor was checked and its class was also checked in order to ensure it was a factor, the table of proportions of the factor was also checked in order to get a feel of the proportions of the data.

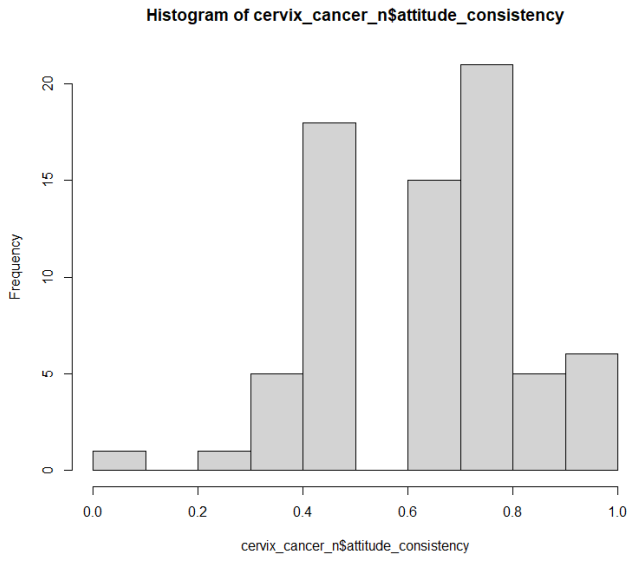
```
> table(cervix_cancer$ca_cervix)

Positive Negative
      21       51
> class(cervix_cancer$ca_cervix)
[1] "factor"
> # Having the Table of Proportions of the factor checked
> prop.table(table(cervix_cancer$ca_cervix))

 Positive  Negative
0.2916667 0.7083333
```

Once this was certified, we began to prepare the data.

PREPARATION

After understanding the data, the data had to be prepared in order to carry out the machine learning on it. After checking the summary of all the numeric values, A normalization function was created. The reason for this was because we wanted the data to be in sync(the numbers should not deviate too far from one another). This was because although the numbers in each column wasn’t too far apart, there were some deviations and just to be sure of the accuracy we normalized the data. After checking if the data had ben normalized, we shuffled it and split the data into training and test data. We also created training and testing labels. The training labels were for the training data and the test labels were for the test data.

Histogram of the class against one of the variables



2.3 MODELLING

The “models” were created using the knn function. Here we set the training data, the testing data and the labels of the training data into the function parameters. We also set the value of k and run the “model”. We then do a table of the “model” on the test labels to somewhat see it accuracy.(Knn does not have a model so the model is the knn training data) To fully see its accuracy, we do the “diag” of the table divided by the “sum” of the table. This results in the accuracy of the model. Thus we tried to determine the best model using different values of k.

2.4 EVALUATION

In order to find the best model we have to change the k values and then find the accuracy of the model by getting the table of the model on(cross referenced with) the test labels. We the use the “sum” and the “diag” function in order to find the accuracy of the model

```
# Testing with different values of K

# k = 1
knn3 <- knn(train = cervix_cancer_knn_train, test = cervix_cancer_knn_test,
                  cl = cacervix_train_label, k=1)

CrossTable(knn3, cacervix_test_label,
           prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE)
cm = table(knn3, cacervix_test_label)
sum(diag(cm))/sum(cm) # Accuracy = 0.9090909

# k = 2
knn4 <- knn(train = cervix_cancer_knn_train, test = cervix_cancer_knn_test,
            cl = cacervix_train_label, k=2)

CrossTable(knn4, cacervix_test_label,
           prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE)
cm = table(knn3, cacervix_test_label)
sum(diag(cm))/sum(cm) # Accuracy = 0.8636364

# k = 3
knn5 <- knn(train = cervix_cancer_knn_train, test = cervix_cancer_knn_test,
            cl = cacervix_train_label, k=3)

CrossTable(knn3, cacervix_test_label,
           prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE)
cm = table(knn5, cacervix_test_label)
sum(diag(cm))/sum(cm) # Accuracy = 0.7727273

# k = 5
knn6 <- knn(train = cervix_cancer_knn_train, test = cervix_cancer_knn_test,
            cl = cacervix_train_label, k=5)

CrossTable(knn3, cacervix_test_label,
           prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE)
cm = table(knn6, cacervix_test_label)
```

After testing with numerous k values, We found out that k = 1 produces the best model with an accuracy of 90 percent

```
             | cacervix_test_label
        knn3 |  Negative | Row Total |
-------------|-----------|-----------|
    Positive |         2 |         2 |
             |     0.091 |           |
-------------|-----------|-----------|
    Negative |        20 |        20 |
             |     0.909 |           |
-------------|-----------|-----------|
Column Total |        22 |        22 |
-------------|-----------|-----------|


> cm = table(knn3, cacervix_test_label)
> sum(diag(cm))/sum(cm) # Accuracy = 0.9090909
[1] 0.9090909
```

3. KMEANS

3.1 BUSINESS UNDERSTANDING

The name of this dataset is “Facebook Metrics Data Set”, downloaded from the UCI machine learning repository. (https://archive.ics.uci.edu/ml/datasets/Facebook+metrics#). This data set is related to posts published on the Facebook page of a renowned cosmetic brand in 2014. The value we are trying to predict from this data set is the “Lifetime Post Total Reach”. This is basically just trying to learn how much attention this post can get.

3.2 DATA UNDERSTANDING AND PREPARATION

Since the data set being used has already been read from the csv, we will then take the data set and put in a new data frame(This is to create a data set we can use for our model). Since we want to predict the classes we will remove the factor column from the new data set. The data would also be scaled using the normalized function created during the Knn analysis.

3.3 MODELLING

A k means model was created using the “kmeans” function. To do that we first had to set the k centres (the class points in the model). We set the points to the first occurrence of each class type in the class column. Then we created the model using the “kmeans” function and set the data and the value of k to 2 which in this case is the number of types of classes in the class column. We then check the model’s clusters, tot.withinss (which is like the accuracy of the model and the most important part of the model) and the models centres.

```
> kmeans_model
K-means clustering with 2 clusters of sizes 28, 44

Cluster means:
  behavior_sexualRisk behavior_eating behavior_personalHygine intention_aggregation intention_commitment attitude_consistency
1           0.9017857       0.8541667              0.5535714             0.7053571            0.8214286            0.6562500
2           0.9943182       0.7916667              0.7500000             0.7585227            0.8131313            0.6420455
  attitude_spontaneity norm_significantPerson norm_fulfillment perception_vulnerability perception_severity motivation_strength
1            0.7797619              0.5446429        0.4166667                0.3690476           0.3392857           0.6726190
2            0.7613636              0.5227273        0.4829545                0.5170455           0.4772727           0.8882576
  motivation_willingness socialSupport_emotionality socialSupport_appreciation socialSupport_instrumental empowerment_knowledge
1              0.2559524                  0.1160714                 0.2008929                  0.3184524             0.2708333
2              0.7500000                  0.6212121                 0.7244318                  0.8030303             0.8560606
  empowerment_abilities empowerment_desires
1             0.1726190           0.2440476
2             0.7518939           0.8371212

Clustering vector:
 [1] 2 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 1 1 1 1 2 1 1
[65] 2 1 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1] 43.30116 66.15417
 (between_SS / total_SS =  25.5 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
> kmeans_model$cluster
 [1] 2 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 1 1 1 1 2 1 1
[65] 2 1 2 2 2 2 2 2
> kmeans_model$tot.withinss
[1] 109.4553
> kmeans_model$centers
  behavior_sexualRisk behavior_eating behavior_personalHygine intention_aggregation intention_commitment attitude_consistency
1           0.9017857       0.8541667              0.5535714             0.7053571            0.8214286            0.6562500
2           0.9943182       0.7916667              0.7500000             0.7585227            0.8131313            0.6420455
  attitude_spontaneity norm_significantPerson norm_fulfillment perception_vulnerability perception_severity motivation_strength
1            0.7797619              0.5446429        0.4166667                0.3690476           0.3392857           0.6726190
2            0.7613636              0.5227273        0.4829545                0.5170455           0.4772727           0.8882576
  motivation_willingness socialSupport_emotionality socialSupport_appreciation socialSupport_instrumental empowerment_knowledge
1              0.2559524                  0.1160714                 0.2008929                  0.3184524             0.2708333
2              0.7500000                  0.6212121                 0.7244318                  0.8030303             0.8560606
  empowerment_abilities empowerment_desires
1             0.1726190           0.2440476
2             0.7518939           0.8371212
```

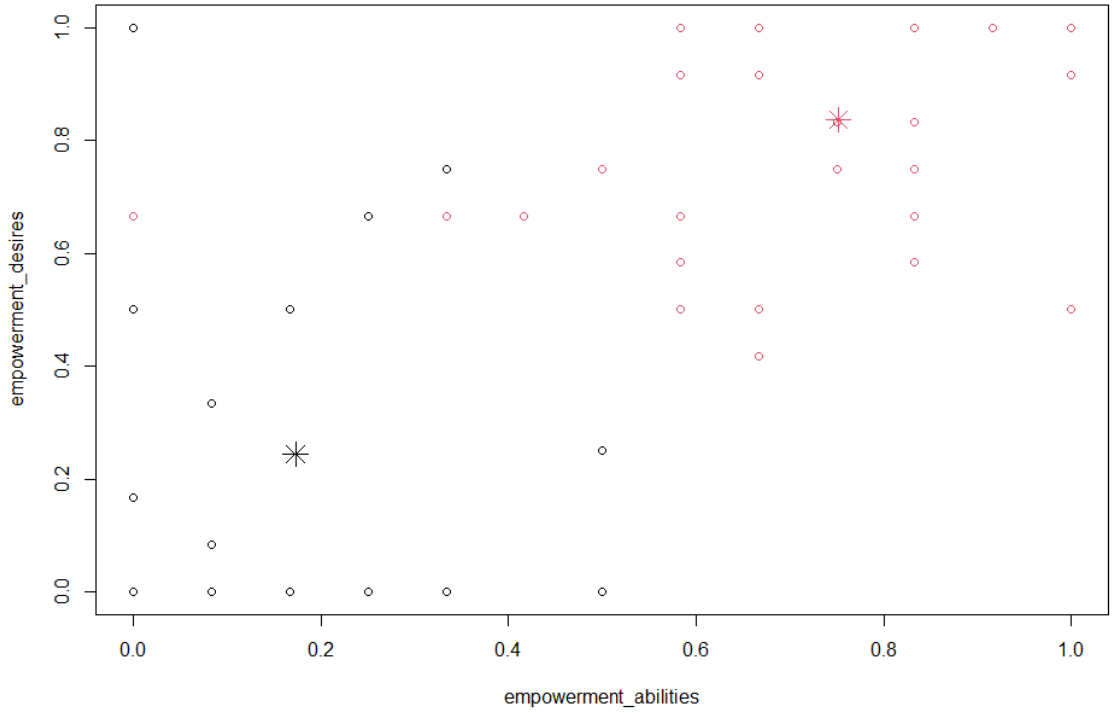
3.4 EVALUATION

The model is evaluated by checking how the clusters created using the model correspond to the actual classes in the dataset

```
> table(cervix_cancer$ca_cervix, kmeans_model$cluster)

    1  2
0 11 40
1 17  4
```

Looking at the table the first class had 40 correct assumptions and 11 wrong ones while the second class had 17 correct assumptions and 4 were wrong. This shows the model is correct to some extent and has about 79 percent accuracy.

A plot of the model using 2 elements from the data set with the k centres shown



The model was also created using the set seed method for the k centres. It however gave a less optimal result.