06/04/2018, Monday

IS 580 KNOWLEDGE, DISCOVERY AND MINING PROJECT

Ahmet Kuzubaslı (1674431), Emre Dogan (2093656), Tayfun Eylen (1626183)

Proposed to: Tugba Taskaya Temizel

1. ABSTRACT

Background and Objective: Early childhood caries (ECC) is a potentially severe disease affecting children all over the world. In the paper "Using association rule mining to identify risk factors for early childhood caries", Association rule mining method was discussed to observe several parameters affecting ECC rate. But some classification and clustering models could also be used to extract more information from the same dataset and make successful predictions.
Methods: ECC data was collected in a cross-sectional analytical study of the 10% sample of preschool children in the South Bac˘ka area (Vojvodina, Serbia). Association rules were extracted from the data by association rule mining. Risk factors were extracted from the highly ranked association rules. For classification, 5 separate models were created as Naive Bayesian, KNN, SVM, Random Forest and ANN. At the end, these models were compared from different perspectives.
Conclusion: The discovered risk factors are mostly confirmed by the literature and the testing results of classification and clustering models are consistent with the rules concluded.

2. DATASET AND INTRODUCTION

ECC Dataset provides the early childhood caries status with some attributes which could affect the existence of this disease. When the dataset is investigated, it is clear that these attributes could provide a relation with the existence of the disease.
In this report, first the descriptive analysis of dataset will be investigated to have an opinion of the attribute values. If necessary, some preprocessing methods, i.e. normalization, feature selection, factorization, will be applied to get better results from the models. Secondly, the association rule mining model in the paper and 5 other classification models are developed to have a predictive model.
Lastly, 3 different type of clustering models are created to see the cluster distribution of the dataset.

3. DESCRIPTIVE ANALYSIS

First, the dataset is investigated with summary() command to find out its range, quartiles, median and mean information. Each attribute's summary is examined and concluded with some facts;

- There is only one non-numeric attribute which indicates "CITY" of the EEC evidence.

- The numerical attributes are splitted in two ways, nominal and ordinal attributes.
- 3 attributes have their maximum value at “999” value. These values are meaningless and correspond to ‘NA’ data type. These ‘NA’ data should be considered as a problem and handled by missing value imputation methods.

Besides the summary of attributes, descriptive statistics of attributes given below are examined and can be found in the .Rmd file given in Appendix.

- Histograms
- Geometric Mean
- Range
- Interquantile Range
- Variance & Coefficient of Variance
- Correlation
- Box Plots

From these statistics and plots, the dataset is overviewed and now, the models can be constructed.

4. CLASSIFICATION MODELS

Before constructing any model, the ‘NA(999)’ values should be replaced. For this purpose, One Hot Encoding is applied.
One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. An attribute with n different type of values can be represented in one hot encoding method as n different unit vector. For example, an attribute with 3 different values can be represented with the vectors given below.

[[1,0,0,0],
[0,1,0,1],
[0,0,1,0]]

The attributes having ‘999’ is not a numerical problem as they are converted to unit vector and defined as a new attribute.
Also, normalization is applied to the dataset to get better results in model accuracy.
Also, it is noticeable that the dataset includes many nominal attributes and they should be converted to their own type by *ordered()* function.

- **Association Rule Mining (implemented on the paper)**

To achieve association rules, Apriori function is used to get related rules for *rhs=c("ECC=2") and rhs=c("ECC=1")* cases. For *rhs=c("ECC=2")* case, a larger support and confidence parameter is selected as the number of (ECC=2) data sample is much greater.

Justification for Model Parameters:

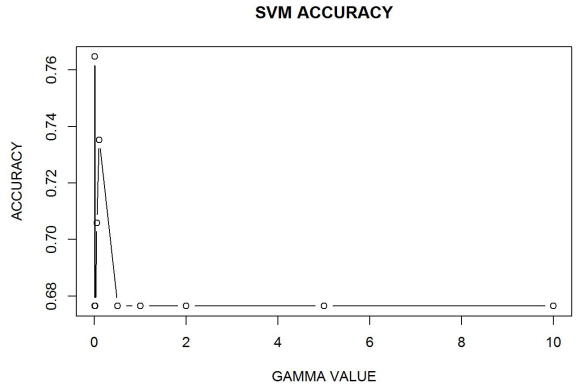
- "rules1" is the rules obtained from the data set having ECC value of 1 at the right hand side with support value 0.1 and confidence value 0.4.

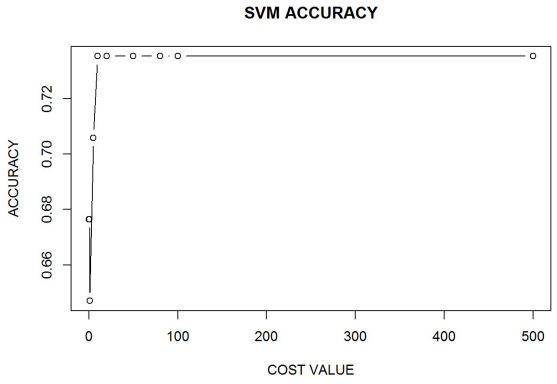
- "rules2" is the rules obtained from the data set having ECC value of 2 at the right hand side with support value 0.3 and confidence value 0.8.

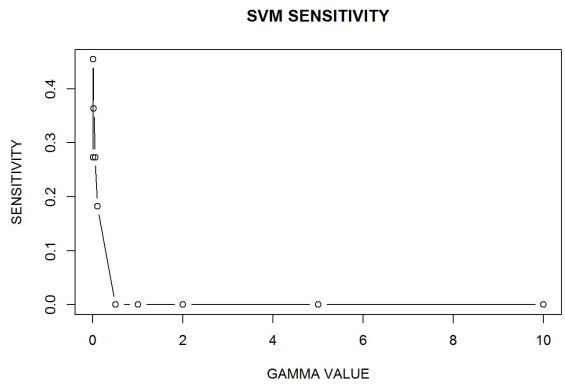
- ECC dataset is unbalanced. As a result, when both ECC values are kept and rules are generated, dense part, which have ECC value of 2, dominates all obtained rules. When support and confidence values are kept low to obtain rules for both ECC values, Number of rules becomes a very large number. With above parameters, ~250 rules are generated for ECC=2 and ~120 rules are generated for ECC=1.

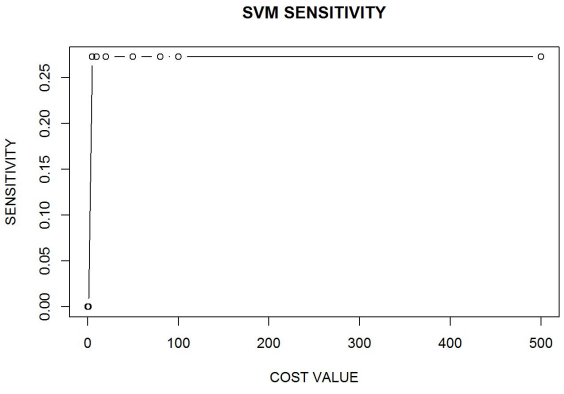
- **Support Vector Machine (SVM)**

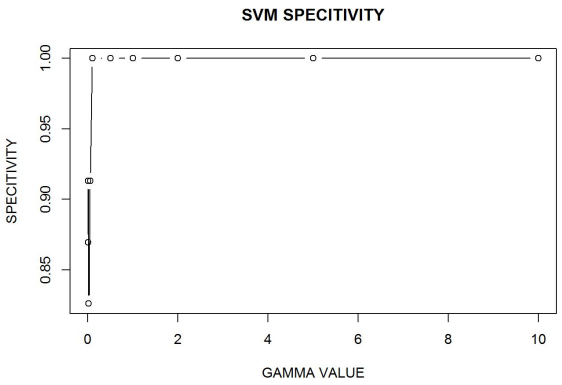
Support Vector Machine is the 2nd classification model to be applied to the dataset.
By changing gamma and cost parameters, the optimal model is chosen. While doing this, accuracy, sensitivity and specitivity of each model is compared with changing cost and gamma parameter. The related plots are given below.

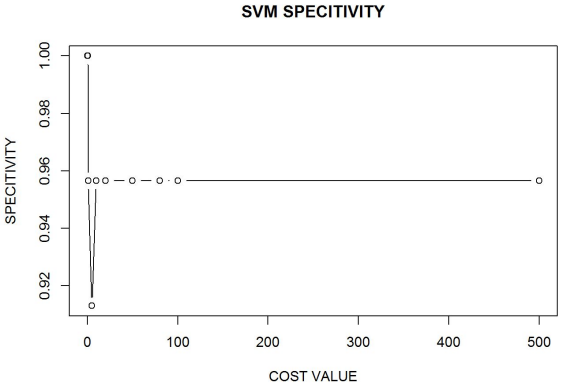












With these plots, the best gamma value is found 0.01 and best cost value is found 5. The model is evaluated with the help of confusion matrix which can be seen below.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1  5  2
##          2 17 44
##
##                Accuracy : 0.7206
##                  95% CI : (0.5985, 0.8227)
##     No Information Rate : 0.6765
##     P-Value [Acc > NIR] : 0.261543
##
##                   Kappa : 0.2236
##  Mcnemar's Test P-Value : 0.001319
##
##             Sensitivity : 0.22727
##             Specificity : 0.95652
##          Pos Pred Value : 0.71429
##          Neg Pred Value : 0.72131
##              Prevalence : 0.32353
##          Detection Rate : 0.07353
##    Detection Prevalence : 0.10294
##       Balanced Accuracy : 0.59190
##
##        'Positive' Class : 1
##
```

Justification for Model Parameters:

- For the SVM model, Cost is how much we penalize the SVM for data points within the margin. If we decrease the cost, the error rate would increase where the margin gets larger. Gamma defines how far the influence of single training example reaches.

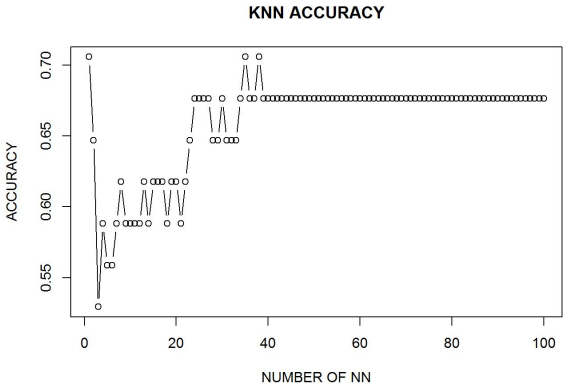
- If the value of Gamma is high, then our decision boundary will depend on points close to the decision boundary and nearer points carry more weights than far away points due to which our decision boundary becomes more wiggly.

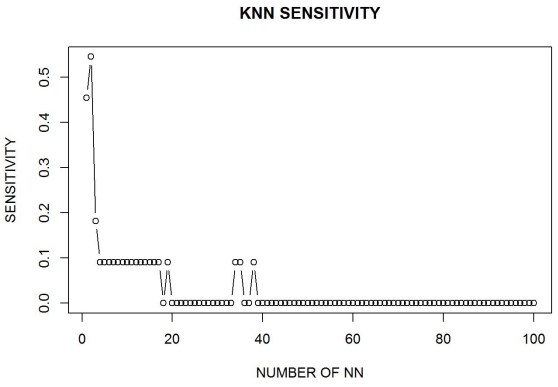
- If the value of Gamma is low, then far away points carry more weights than nearer points and thus our decision boundary becomes more like a straight line.

- The value of gamma and C should not be very high because it leads to the overfitting or it shouldn't be very small (underfitting). Thus we need to choose the optimal value of C and Gamma in order to get a good fit. In our case, different costs and Gamma values were tried an adjusted for the best performance.

- **KNN**

Another classification method applied to the dataset is KNN. With changing k parameter, the results were compared. For doing this, accuracy and sensitivity values with changing k parameter are plotted. These plots can be seen below,





Although the best accuracy is found in k=1, this cannot be chosen because this case leads directly to overfitting issue. So the best case in KNN is with the k=32.
Again, the confusion matrix is investigated as it can be seen below.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1  8  7
##          2 14 39
##
##                Accuracy : 0.6912
##                  95% CI : (0.5674, 0.7976)
##     No Information Rate : 0.6765
##     P-Value [Acc > NIR] : 0.4545
##
##                   Kappa : 0.2306
##  Mcnemar's Test P-Value : 0.1904
##
##             Sensitivity : 0.3636
##             Specificity : 0.8478
##          Pos Pred Value : 0.5333
##          Neg Pred Value : 0.7358
##              Prevalence : 0.3235
##          Detection Rate : 0.1176
##    Detection Prevalence : 0.2206
##       Balanced Accuracy : 0.6057
##
##        'Positive' Class : 1
##
```

Justification for Model Parameters:

- For the KNN model, the most and only important parameter is the 'k value'. it looks through the training data and finds the k training examples that are closest to the new example. It then assigns the most common class label (among those k training examples) to the test example.

- When the data is directly fed to the model, we observed that k=1 gives the best results within all k values. Normally, k=1 might show the appearance of overfitting. But in our case, it does not. As our class labels are nominal and have small number of types, 1-NN does not directly show overfitting.

- Also, we tried this model for the ordered (nominal) dataset. The optimal k value is not 1 but equal to 39 in this case. But the accuracy result did not change surprisingly.

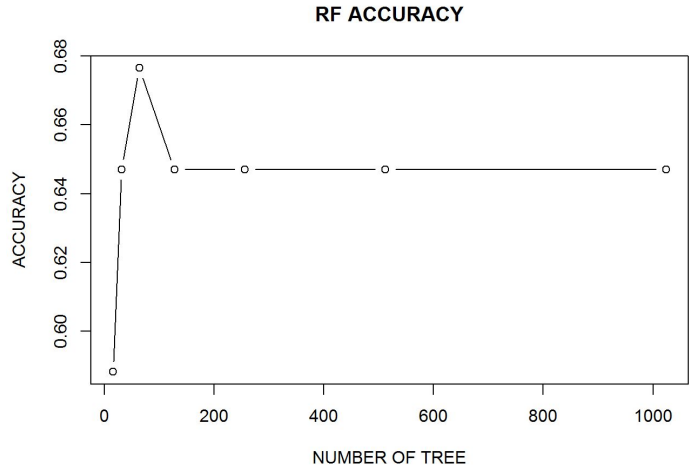
- **Naive Bayesian**

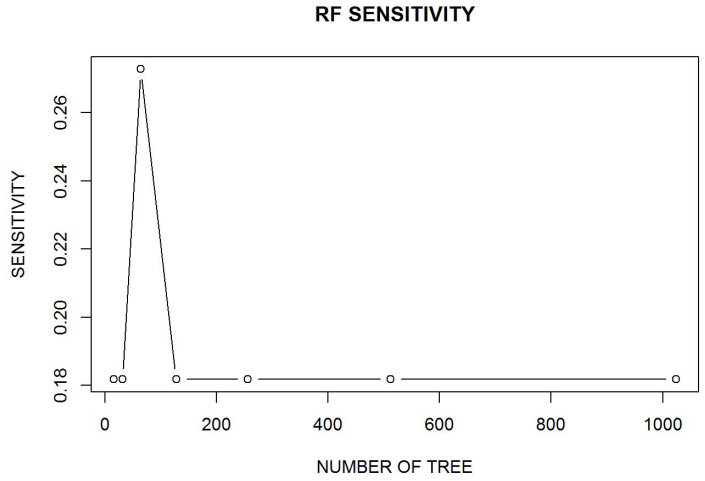
The Naive Bayes classification algorithm is a probabilistic classifier. It is based on probability models that incorporate strong independence assumptions.
The independence assumptions often do not have an impact on reality. Therefore, they are considered as naive.
It is clear that there is no parameter for this model and so, no justification.
The result of the confusion matrix can be seen below.

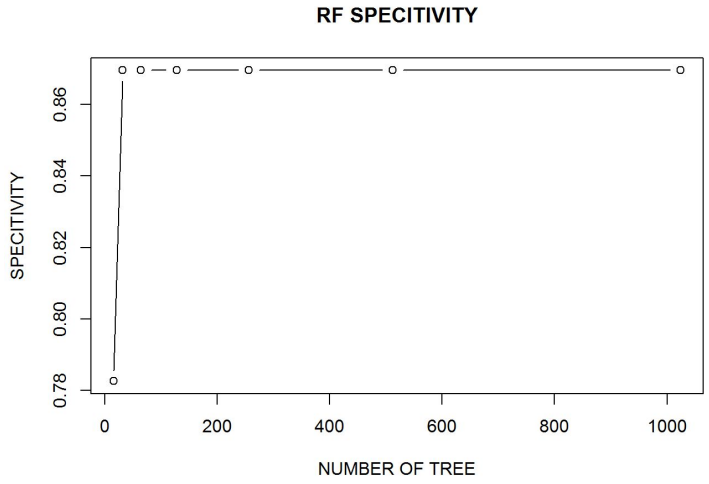
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 15 28
##          2  7 18
##
##                Accuracy : 0.4853
##                  95% CI : (0.3622, 0.6097)
##     No Information Rate : 0.6765
##     P-Value [Acc > NIR] : 0.9996453
##
##                   Kappa : 0.0585
##  Mcnemar's Test P-Value : 0.0007232
##
##             Sensitivity : 0.6818
##             Specificity : 0.3913
##          Pos Pred Value : 0.3488
##          Neg Pred Value : 0.7200
##              Prevalence : 0.3235
##          Detection Rate : 0.2206
##    Detection Prevalence : 0.6324
##       Balanced Accuracy : 0.5366
##
##        'Positive' Class : 1
##
```

- **Random Forest Model**

The forth applied model is the random forest model. It is based on the idea of decision trees and show a higher success rate than the previous models. For Random Forest, nTree parameter which denotes the “number of trees” is the only significant parameter. The effect of this parameter can be observed from the graphs given below.







From these plots, It can be seen that the optimal value of nTree is 64 where it shows the best accuracy and sensitivity character.

After choosing the necessary parameter, we evaluated the model from the confusion matrix given below.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1  4  2
##          2 18 44
##
##                Accuracy : 0.7059
##                  95% CI : (0.5829, 0.8102)
##     No Information Rate : 0.6765
##     P-Value [Acc > NIR] : 0.3537252
##
##                   Kappa : 0.1707
##  Mcnemar's Test P-Value : 0.0007962
##
##             Sensitivity : 0.18182
##             Specificity : 0.95652
##          Pos Pred Value : 0.66667
##          Neg Pred Value : 0.70968
##              Prevalence : 0.32353
##          Detection Rate : 0.05882
##    Detection Prevalence : 0.08824
##       Balanced Accuracy : 0.56917
##
##        'Positive' Class : 1
##
```

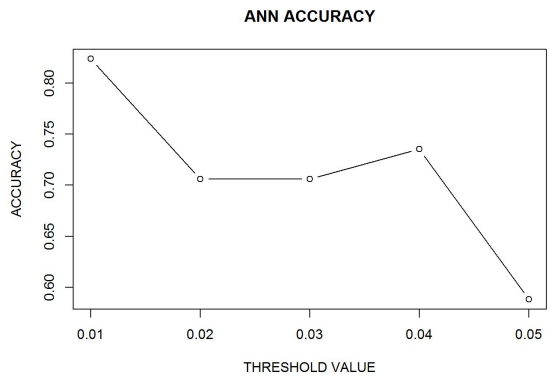
Justification for Model Parameters:

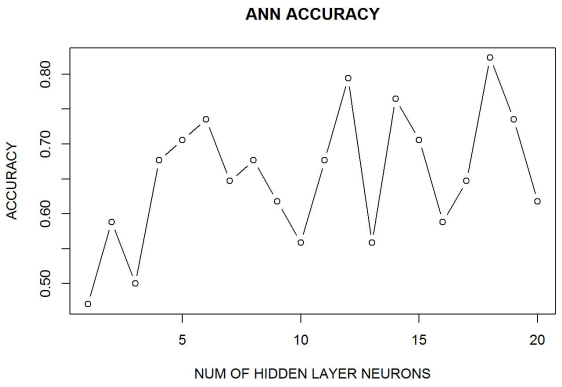
- The most important parameter for this model is the number of trees. This parameter is tried for different values and with the performance comparison, it is justified.

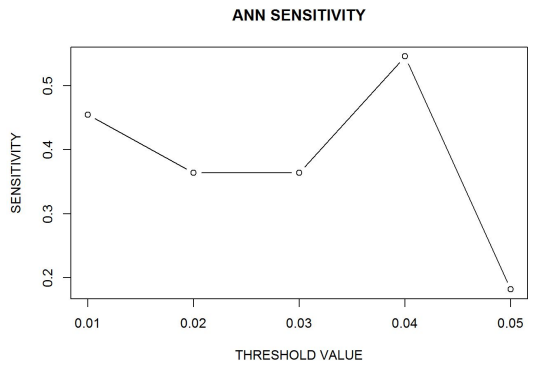
- Notice that When the data is not considered as nominal for the necessary attributes and given to the model directly, the number of tree parameter is equal to 64. But when we preprocess the data to specify its type, this parameter becomes 2048. There is a trade-off situation where increasing 'number of tree' parameter gives better accuracy but wastes more space in the memory.

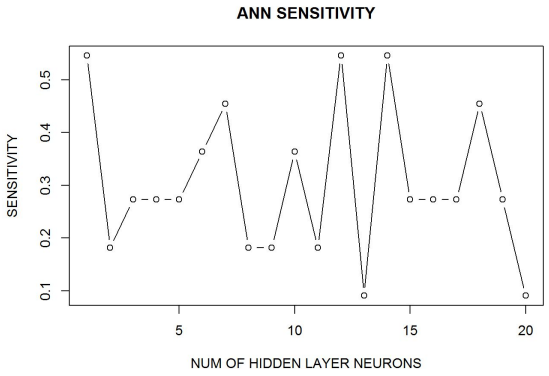
- **Artificial Neural Network (ANN):**

Threshold value and Number of Hidden Layer Neurons are two important parameters affecting the performance of ANN. These parameters are evaluated from the graphs given below and according to them, the necessary justifications are made.









After justification, number of hidden layer neurons are set to 18 and threshold value is set to 0.01 to get the best accuracy and sensitivity performances.

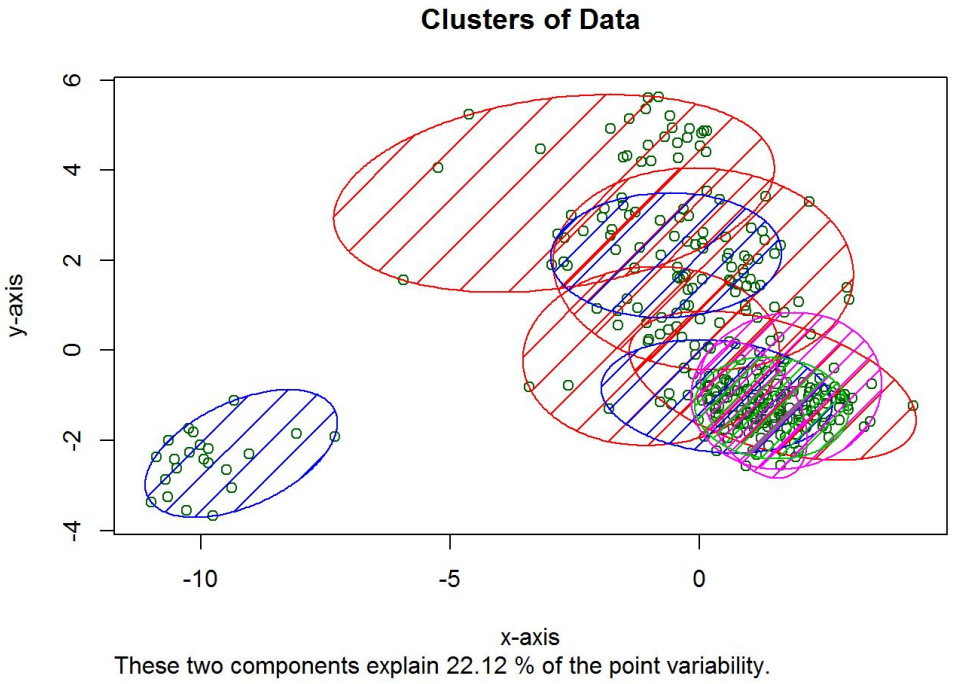
After setting these parameters, the performance is observed with the help of confusion matrix which is given below.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 12  7
##          1 10 39
##
##                Accuracy : 0.75
##                  95% CI : (0.6301776, 0.8471195)
##     No Information Rate : 0.6764706
##     P-Value [Acc > NIR] : 0.1203633
##
##                   Kappa : 0.4077869
##  Mcnemar's Test P-Value : 0.6276258
##
##             Sensitivity : 0.5454545
##             Specificity : 0.8478261
##          Pos Pred Value : 0.6315789
##          Neg Pred Value : 0.7959184
##              Prevalence : 0.3235294
##          Detection Rate : 0.1764706
##    Detection Prevalence : 0.2794118
##       Balanced Accuracy : 0.6966403
##
##        'Positive' Class : 0
##
```

5. CLUSTERING MODELS

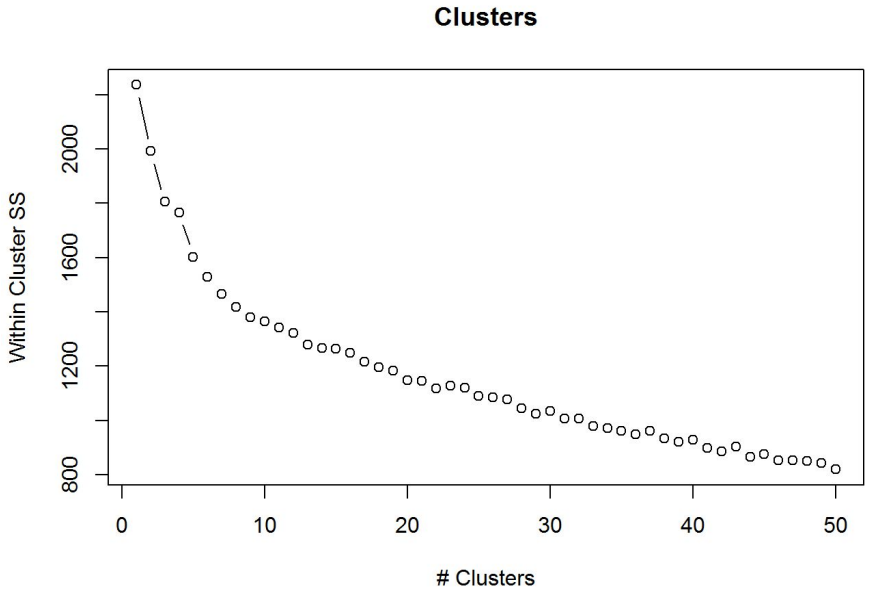
- **K-Means**

To find the optimal k value, the elbow method is used.



Justification of the Model:

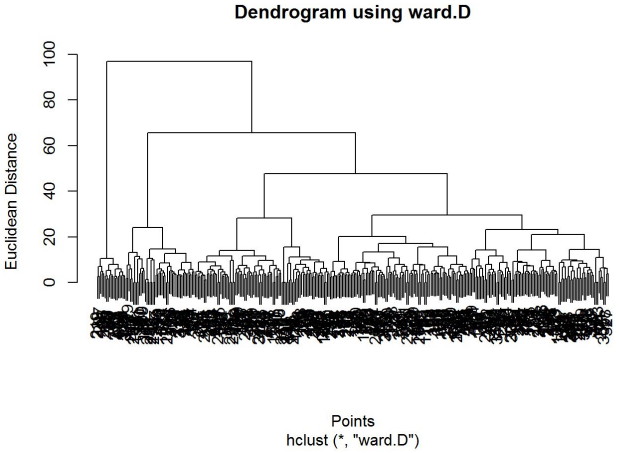
Initial configuration is fixed. We will run k-means for k = 1:10. vi. Plot error vs k to find optimal number of clusters by using the elbow method.

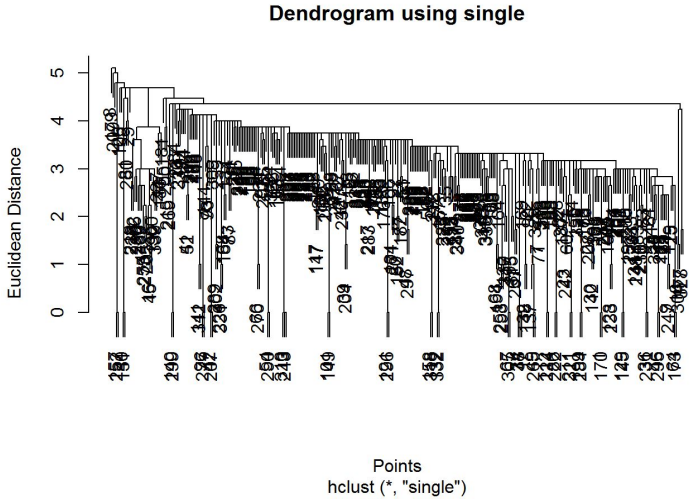


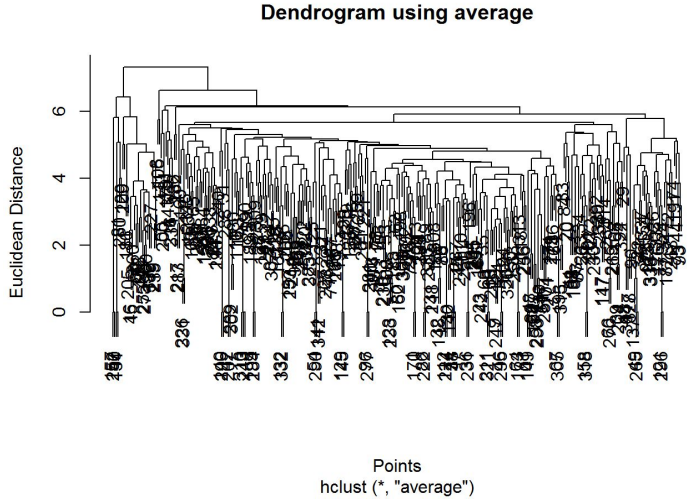
It can be seen that ideal k value is around 2 & 3. This is expected as we know that the data has 2 clusters.

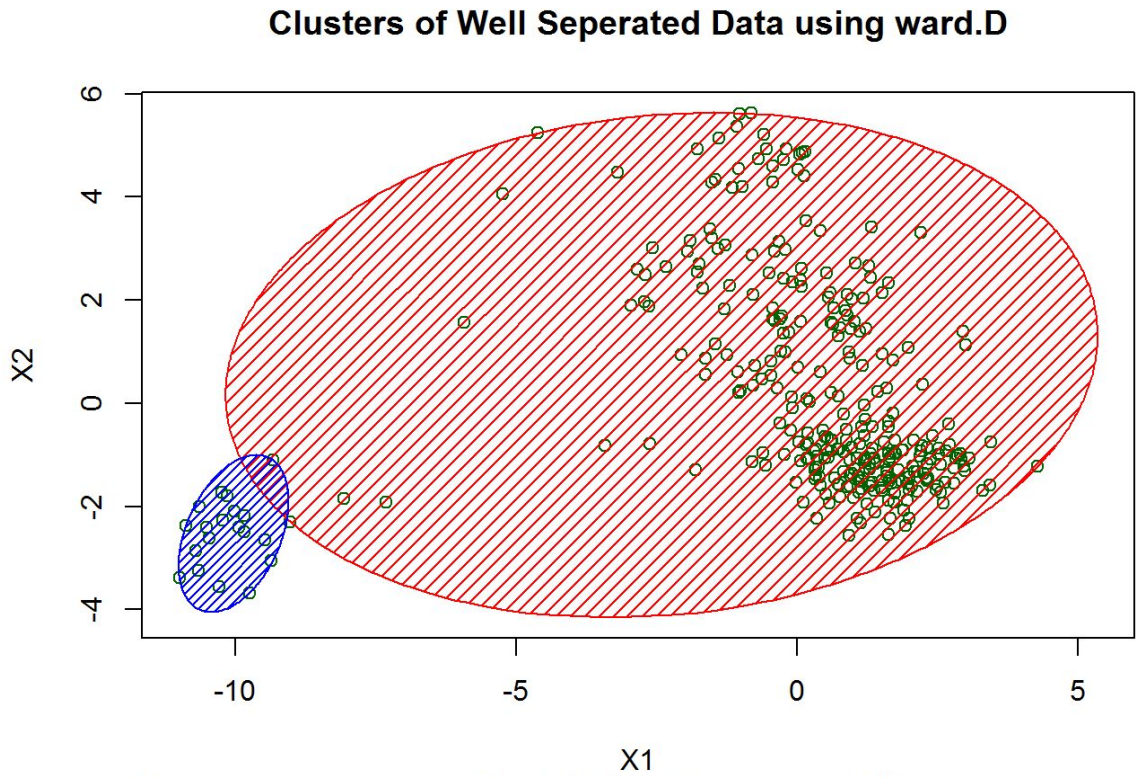
- **Hierarchical Clustering**

In this section, we also apply hierarchical clustering. In order to understand with linkages work best for the well separated data, we plot their dendrogram in a for loop. As seen from the dendrograms, the best separation is obtained when warD is used.


Dendrogram using ward.D
Euclidean Distance
0
20
40
60
80
100
Points
hclust (*, "ward.D")


Dendrogram using single
Euclidean Distance
0
1
2
3
4
5
Points
hclust (*, "single")


Dendrogram using average
Euclidean Distance
0
2
4
6
Points
hclust (*, "average")

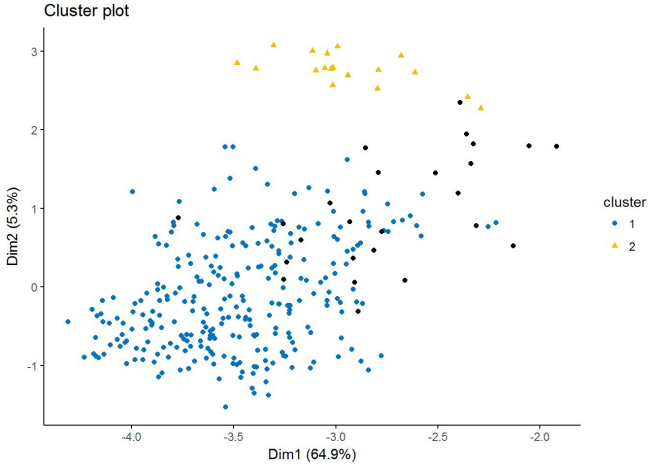


Justification for H-clustering Parameters:

For H-clustering parameters, we first plot the dendogram of the clusters. On this dendogram, we see the separation distance (length) of the linkages. Then, we find the cluster numbers by cutting the tree at maximum length point.as Fitting hierarchical clustering to the mall dataset with k = 5 (found using dendrogram)

- **DB-SCAN Clustering**

This model could not be successful at all in our dataset as it can be seen below.



6. COMPARISON FOR CLASSIFICATION MODELS

For this dataset, We applied Associative Rule Mining and 5 different classification models. When the parameters of all models have been set, the following accuracy results were achieved.

Naive Bayesian: 0.48
SVM: 0.72
KNN: 0.69
Random Forest: 0.75
ANN (Artificial Neural Networks): 0.82

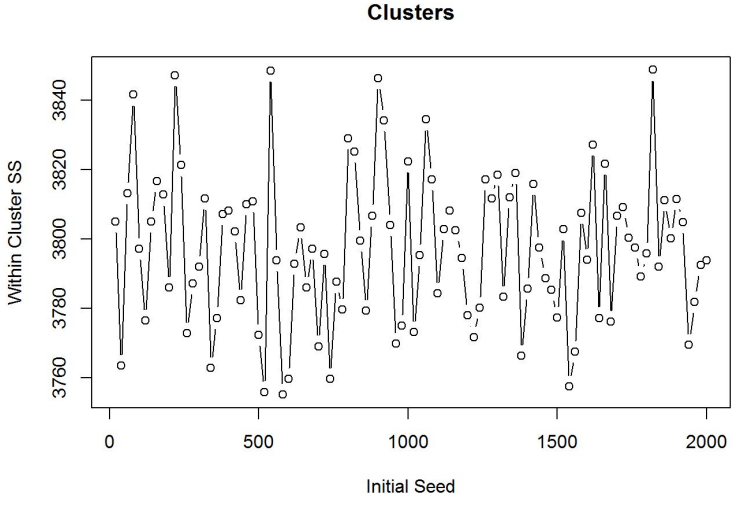
From these models, Random Forest and ANN are the models giving the best accuracy results. From these two, ANN is harder to implement whereas Random Forest is a much easier model than the ANN. The problem with Random Forest is that in some cases, the number of trees may get larger and this leads to memory issues.

SVM and KNN gives mid-level results. They are also easy to implement and adjust parameters.

Naive Bayesian cannot handle this dataset. This is clearly a failure as it even could not catch 50% accuracy rate.

7. COMPARISON FOR CLUSTERING MODELS

Now, we compare our clustering models using wcss analysis. wcss is a vector of within-cluster sum of squares, one component per cluster. To do this, we begin with an empty wcss vectors and we calculate and sum within ss values of clusters by running the model with 100 different initial configurations.. We can view the sum of within cluster sum of squares error and look at indices with minimum error.



In the above analysis, we created kmeans models with different k values (from k=2 to k=10) and initialize them from different initialization points by manipulating the random seed. Then, we sum wcss for each time and compare them against to find insensitivity to initialization point.

8. CONCLUSION

In this project, we focused on the "Using association rule mining to identify risk factors for early childhood caries" paper. In the paper itself, association rule mining is used to extract information about the several attributes affecting the early childhood caries disease.
First, we found these rules from the dataset by using Apriori function in R. Then, 5 different classification models were examined by applying necessary preprocessing methods. Among these models, Random Forest and ANN gave us the best prediction results. But, both of these models are complicated to implement and have a high computational complexity . KNN and SVM models resulted with a little worse results than ANN &RF. But still, they may be a good option as their implementation is not complex at all. Lastly, the Naive Bayesian classification gave us the worst results and cannot be used as a predictive model for this dataset. For all models, the dataset affects the performance in a bad way. As explained in the descriptive analysis, the dataset has nominal and ordinal attributes and it makes harder for models to predict the result successfully.
We also developed 3 different clustering models and had the chance to observe differences between them.

9. REFERENCES

- *The ECC paper*
- *stackoverflow.com*
- *r-bloggers.com*
- *analyticsvidhya.com*