IFN 509: Data Exploration and Mining

Assessment 2

Team Name: Renee, Melanie, Nina

Group No. 18

|  |  |
| --- | --- |
| Student Name | Student Id |
| Renee Chand | n10888373 |
| Melanie Packer | n5736404 |
| Phan Thao Nhi Nguyen | n11232862 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Student 1 | Student 2 | Student 3 |
| Student 1 | <33 %> | <33 %> | < 33%> |
| Student 2 | <33 %> | <33 %> | <33 %> |
| Student 3 | <33 %> | <33 %> | <33 %> |

Table of Contents

[Project A: Association Mining 4](#_Toc117853016)

[1) Pre-Processing 4](#_Toc117853017)

[2) Association Mining Questions 4](#_Toc117853018)

[a) Minimum Support and Minimum Confidence Thresholds 4](#_Toc117853019)

[b) Top 5 Rules 4](#_Toc117853020)

[3) Product Categories 4](#_Toc117853021)

[4) Sequence Analysis 5](#_Toc117853022)

[5) Decision Making Outcomes 5](#_Toc117853023)

[Project B: Clustering 6](#_Toc117853024)

[1) Pre-Processing 6](#_Toc117853025)

[2) Clustering Model 6](#_Toc117853026)

[a) Clustering Algorithm 6](#_Toc117853027)

[b) Optimal Number of Clusters 6](#_Toc117853028)

[c) Cluster Centroids 7](#_Toc117853029)

[d) Normalisation of Variables 7](#_Toc117853030)

[3) Optimal Number of Clusters Model 8](#_Toc117853031)

[a) Visualisation of Clusters 8](#_Toc117853032)

[b) Characterisation of Clusters 9](#_Toc117853033)

[4) Clustering Model with Age 10](#_Toc117853034)

[a) Clustering Algorithm 10](#_Toc117853035)

[b) Cluster Centroids 10](#_Toc117853036)

[5) Clustering Interpretation Differences 10](#_Toc117853037)

[6) Decision Making Outcomes 11](#_Toc117853038)

[Project C: Predictive Mining: Decision Tree Models 12](#_Toc117853039)

[1) Pre Processing 12](#_Toc117853040)

[2) Default Setting 12](#_Toc117853041)

[a) Classification Accuracy 12](#_Toc117853042)

[b) Size of Tree 12](#_Toc117853043)

[c) First Split Variable 12](#_Toc117853044)

[d) Important Variables 13](#_Toc117853045)

[e) Parameters 13](#_Toc117853046)

[3) GridSearchCV 13](#_Toc117853047)

[a) Classification Accuracy 13](#_Toc117853048)

[b) Size of Tree 13](#_Toc117853049)

[c) First Split Variable 13](#_Toc117853050)

[d) Important Variables 14](#_Toc117853051)

[e) Model Overfitting 14](#_Toc117853052)

[4) Differences between Models 14](#_Toc117853053)

[5) Overall Interpretation 14](#_Toc117853054)

[Project C: Predictive Mining: Regression Models 15](#_Toc117853055)

[1) Pre-Processing 15](#_Toc117853056)

[2) Default Regression Method 16](#_Toc117853057)

[a) Model Choice 16](#_Toc117853058)

[b) Regression Function 16](#_Toc117853059)

[c) Standardisation of Variables 16](#_Toc117853060)

[d) Variables 16](#_Toc117853061)

[e) Important Variables 16](#_Toc117853062)

[f) Classification Accuracy 17](#_Toc117853063)

[g) Overfitting 17](#_Toc117853064)

[3) Reduced Regression Method 17](#_Toc117853065)

[a) Dimensionality Reduction 17](#_Toc117853066)

[b) Classification Accuracy 17](#_Toc117853067)

[c) Overfitting 18](#_Toc117853068)

[d) Top 3 Important Variables 18](#_Toc117853069)

[4) Differences between Models 18](#_Toc117853070)

[5) Overall Interpretation 19](#_Toc117853071)

[Project C: Predictive Mining: Neural Network Models 19](#_Toc117853072)

[1) Pre-Processing 19](#_Toc117853073)

[2) Default Neural Network 19](#_Toc117853074)

[a) Model Parameters 19](#_Toc117853075)

[b) Classification Accuracy 20](#_Toc117853076)

[c) Training Process 20](#_Toc117853077)

[d) Over Fitting 20](#_Toc117853078)

[3) Reduced Neural Network 21](#_Toc117853079)

[a) Feature Selection and Inputs 21](#_Toc117853080)

[b) Classification Accuracy 21](#_Toc117853081)

[c) Iteration 21](#_Toc117853082)

[d) Overfitting and Training process convergence 21](#_Toc117853083)

[4) Differences between Models 21](#_Toc117853084)

[5) Overall Interpretation 22](#_Toc117853085)

[Project C: Predictive Mining Comparison 22](#_Toc117853086)

[1) Model Choice 22](#_Toc117853087)

[2) Positives and Negatives of Predictive Modelling Methods 23](#_Toc117853088)

[Appendix 1: Optimal Decision Tree 24](#_Toc117853089)

# Project A: Association Mining

## Pre-Processing

Prior to building the association mining model, we undertook the following pre-processing steps on the dataset:

1. Checked for errors in the data such as missing values or invalid data by df.describe(), df.isnull(). We found there were none so did not need to take further steps to address this issue.
2. Rounded the values in the “Quantity” variable to the nearest whole number, as products cannot be sold in fractions of an item.
3. Transformed the “Quantity” variable data type to int64 from float64 as it no longer contains decimal numbers.

“Sales\_ID” and “SKU-Category” were identified as the most appropriate variables to use in the association mining model. This is because association mining models show the frequent combinations of items within transactions. Sales ID is the unique identifier for each transaction and the SKU-Category is the unique identifier of the categories, the combination of these variables allows for analysis of frequency of combinations of item categories (SKU-Category) being purchased across all transactions (Sales ID).

## Association Mining Questions

### Minimum Support and Minimum Confidence Thresholds

The thresholds set for this model were:

* Minimum support – 0.015
* Minimum confidence – 0.015

These thresholds were chosen as they optimised the number of results to be sufficiently useful but not so numerous they were un-interpretable.

### Top 5 Rules

The top 5 rules (ranked by Lift values) are:

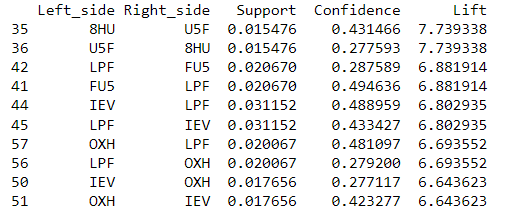


Figure 1. Top 5 Association Rules by Lift Value

1. 8HU, U5F/U5F, 8HU
2. LPF, FU5/FU5, LPF
3. IEV, LPF/LPF, IEV
4. OXH, LPF/LPF, OXH
5. IEV, OXH/OXH, IEV

We've listed these as pairs as each pair of rules has the same Lift value. The rule 8HU, U5F (and it's vice versa rule pair) has the greatest lift of 7.74… of all rules. The remaining rules (items II – V above) have comparable lift values ranging from 6.64 to 6.88 (see Figure 1).

## Product Categories

The top-5 common product categories that customers bought with the product category ‘01F’ are:

1. IEV
2. LPF
3. OXH
4. FU5
5. N8U

Figures 2 and 3 verify this.

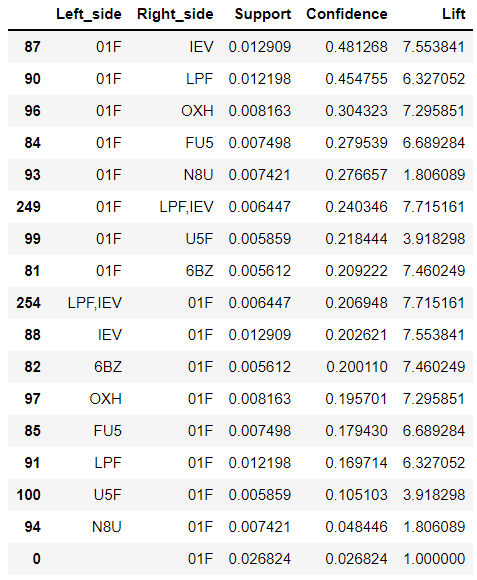


Figure 3. Top 5 Association Rules including product code 01F (sorted by confidence)

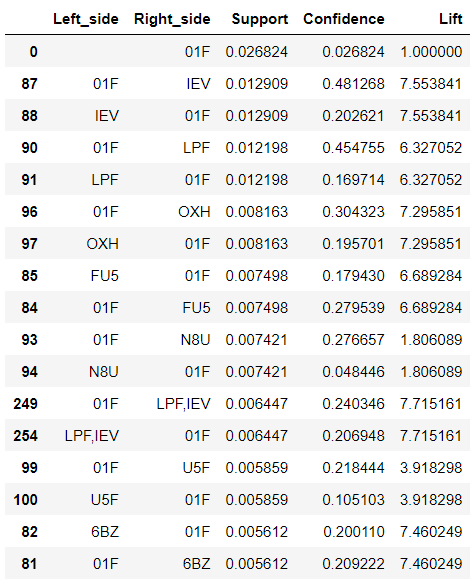


Figure 3 Top 5 Association Rules including product code 01F (sorted by support)

## Sequence Analysis

Sequential analysis can be performed on this dataset as this can be grouped by sales ID and ordered by the date. See Figure 4.

## Decision Making Outcomes

Store owners could use the outcomes from this study to inform how they strategically place categories of produce frequently bought together in proximity. By using the date variable, store owners are informed when particular items are likely to be bought, which product categories these correspond to, and plan accordingly.

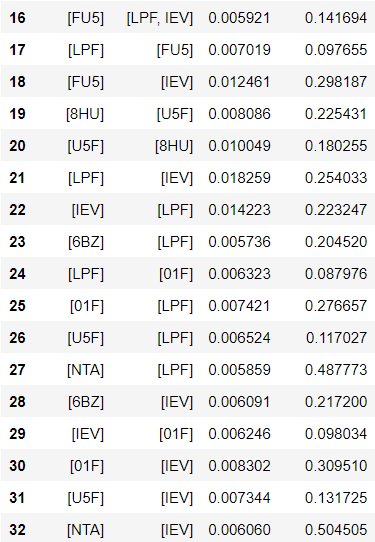
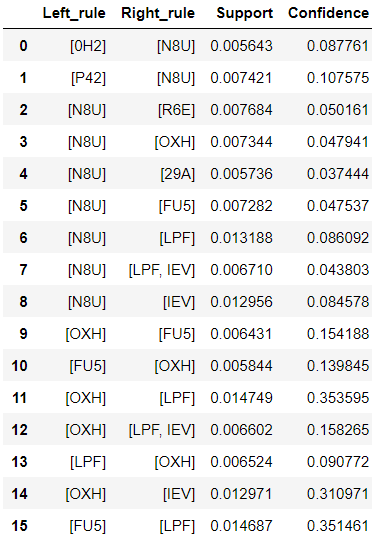


Figure 4. Results from Sequence Analysis.

# Project B: Clustering

## Pre-Processing

The following pre-processing steps were undertaken on the dataset, prior to building clustering model for this project:

1. Checked for errors in the data such as missing values or invalid data by some functions as df.describe(), df.isnull(). We didn’t find some NA or invalid values.
2. Standardise the values of each variable to ensure that extreme values did not appear more important than other variables in the model and can be on the same scale for appropriate comparison and interpretation
3. Categorical variables (specifically ‘age’) was mapped to appropriate numeric values.

## Clustering Model

### Clustering Algorithm

The *K Means* clustering algorithm was used to analyse the sample dataset on diabetic patients. This was chosen because the dataset has no labelled classes within it, so the clustering task sought to. Selected variables included:

* number of lab procedures
* number of outpatient visits
* number of inpatient visits
* number of medications
* time spent in hospital

### Optimal Number of Clusters

The optimal number of clusters identified is K=4.

An elbow chart was created to view the turning point in the curve of the sum of square distances from the centre of all clusters (see Figure 5 and zoomed in view of the same values in Figure 6). The chart shows the potential optimal value for k as 4,6 and 8. The silhouette score was then calculated for each of these three potential values of K:

* K = 4 silhouette score: 0.301
* K= 6 silhouette score: 0.199
* k=8 had a silhouette score of 0.214

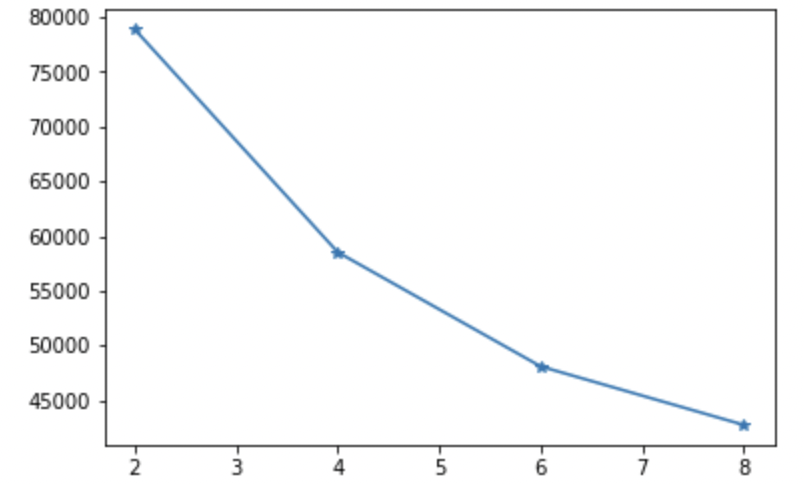


Figure 5. Zoomed in view of Elbow Chart showing only K=2 to K=8

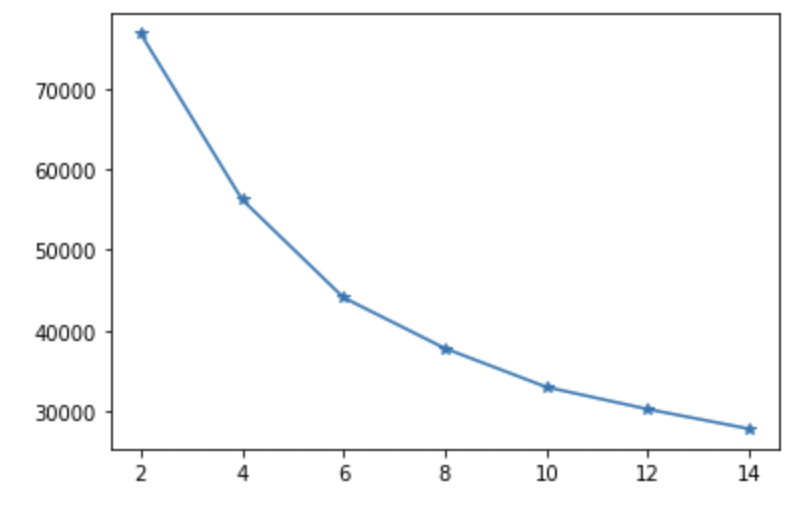


Figure 6. Elbow Chart of K=2 to K=12

### Cluster Centroids

Figure 7 shows a series of arrays that represent the centroids for each of the four clusters with the optimal model, with rows representing separate clusters. Each cluster has 5 dimensions, which represent the centroid of each of the input variables in the model.

All dimensions in the cluster 3 centroid are negative, whereas the cluster 1 centroid is positive. As a result, the centroids of clusters 1 and 3 are very distinct from one another. In contrast, cluster pairs cluster 0 & cluster 1 and cluster 2 & cluster 3 are much located closely together when the values in the centroid arrays are compared.

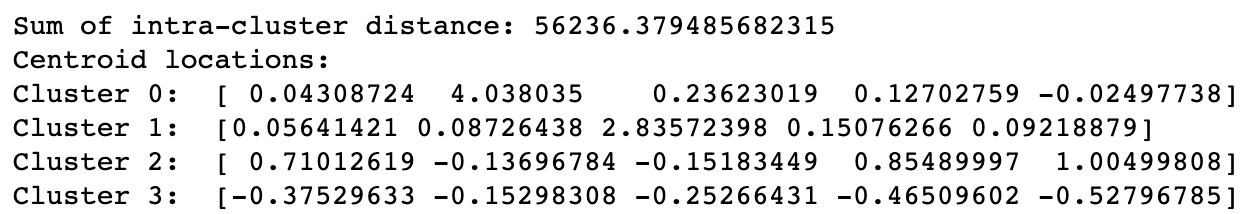


Figure 7. Cluster centroids from default K Means Clustering model

### Normalisation of Variables

As stated in question 1) of Project B: Clustering, all variables were normalised for this clustering algorithm. This means all variables are on the same scale, so they are comparable by the algorithm and outputs are more easily interpretable.

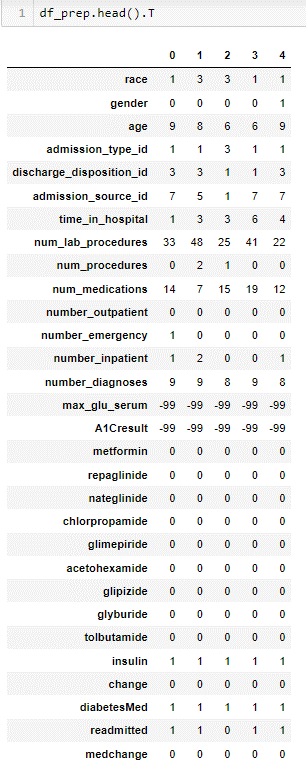


Figure 8: Output for normalisation of values

## Optimal Number of Clusters Model

### Visualisation of Clusters

Figure 8 shows a pairplot visualisation of the characteristics of each cluster. This is based on pairing up each of the 5 variables with each other. Table 1 outlines the unique features of each cluster, based on the interpretation of Figure 8.

Table 1. Comparison of characteristics of each cluster identified in K means model.

|  |  |
| --- | --- |
| Name | Interpretation of Pairplot |
| Cluster 0 | Patients in Cluster 0 had the greatest number of outpatient visits and took a lower number of medications than any other cluster. They also had more outpatient visits than inpatient visits to hospital. |
| Cluster 1 | Patients in Cluster 1 had the lowest number of outpatient visits out of all clusters, but the highest number of inpatient visits to hospital. They also had the greatest variation in time spent in hospital. |
| Cluster 2 | Patients in Cluster 2 had the greatest number of lab procedures and medications taken than any other cluster. They have the lowest number of outpatient visits, but the highest amount of time spent in hospital when compared to other clusters. |
| Cluster 3 | Patients in Cluster 3 had the lowest amount of time spent in hospital and a relatively low number of inpatient visits when compared with other clusters. |

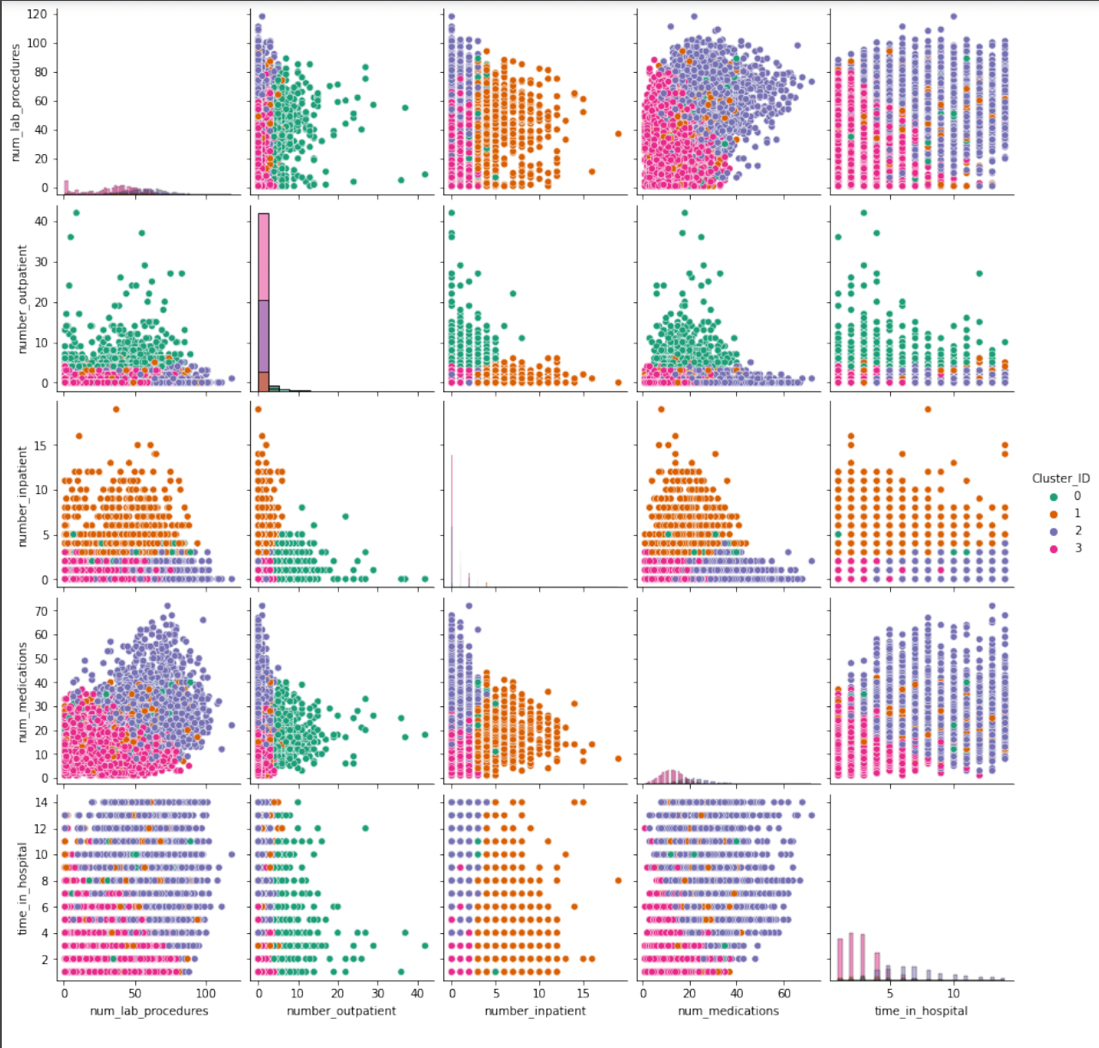


Figure 9. Pairplot visualisation of clusters used in K means model

### Characterisation of Clusters

Table 2 shows the descriptive label and description of each cluster created in the K-Means algorithm.

Table 2. Distribution characteristics of clusters in K-Means Algorithm

|  |  |  |
| --- | --- | --- |
| Cluster | Name | Characteristics |
| Cluster 0 | More hospital = more medicine | Right leaning distribution (positive) for the number of outpatients and number of inpatients. The number of medications is slightly right leaning. |
| Cluster 1 | More inpatient = more medicine | Right-leaning distribution (positive) for the number of inpatients and slightly right-leaning for the number of medications. |
| Cluster 2 | More medicine = less hospital | Right leaning distribution (positive) for the number of lab procedures. The number of inpatient and outpatient is not skewed but the density is higher for the same values. The number of medication and time spent in hospital are skewed to the right, resulting in a greater number of medication and fewer short stays in hospital. |
| Cluster 3 | Shorter stays but more tests | Left leaning distribution (negative) for the number of lab procedures. The time spent in hospital is significantly varied for the short stays. |

## Clustering Model with Age

### Clustering Algorithm

*K protypes* was the clustering algorithm used in the analysis of diabetes the dataset now includes the age variable. It was chosen because it is able to manage mixed data types (numerical and categorical variables) rather than only k-means (numerical) or k-modes (categorical) clustering.

### Cluster Centroids

The image below is showing the optimal number of clusters is k = 4 given by the silhouette score. The centroids for the optimal 4 clusters are shown in Figure 9. Each cluster has 6 dimensions.

Text

Description automatically generatedThe centroids of clusters 0 and 1 have close distances from each other. And Cluster 2 and cluster 3 centroids are near with each other in dimensional. We can see cluster 0 and cluster 3 centroids have the farthest distance among the centroids.

Figure 10: Cluster centroids for K-prototype model

## Clustering Interpretation Differences

Figure 9 shows the pairplot visualisation of the K-Prototypes model created in this section of the report. This chart shows that the age variable has a strong influence over other characteristics of patients within each cluster (summarised in Table 3).

Table 3. Comparison of characteristics of each cluster identified in K-prototype model.

|  |  |
| --- | --- |
| Name | Characteristics |
| Cluster 0 | Patients in cluster 0 are generally older than other clusters, and have the lowest number of procedures, outpatient and inpatient visits to hospital. The length of time they spend in hospital reduces as their age increases. |
| Cluster 1 | Patients in cluster 1 are generally younger than patients in other clusters. With this, their time spent in hospital reduces as their age increases. |
| Cluster 2 | Patients in cluster 2 have a broad spread of ages represented, comparable with cluster 3. They generally have more a higher number of procedures and medications than other clusters. As their age increases, so too does the time they spend in hospital. |
| Cluster 3 | Patients in cluster 2 have a broad spread of ages represented, comparable with cluster 3. These patients have the highest number of inpatient and outpatient visits than other clusters. They are the only group that do not appear to have a relationship between age and time spent in hospital. |

## Decision Making Outcomes

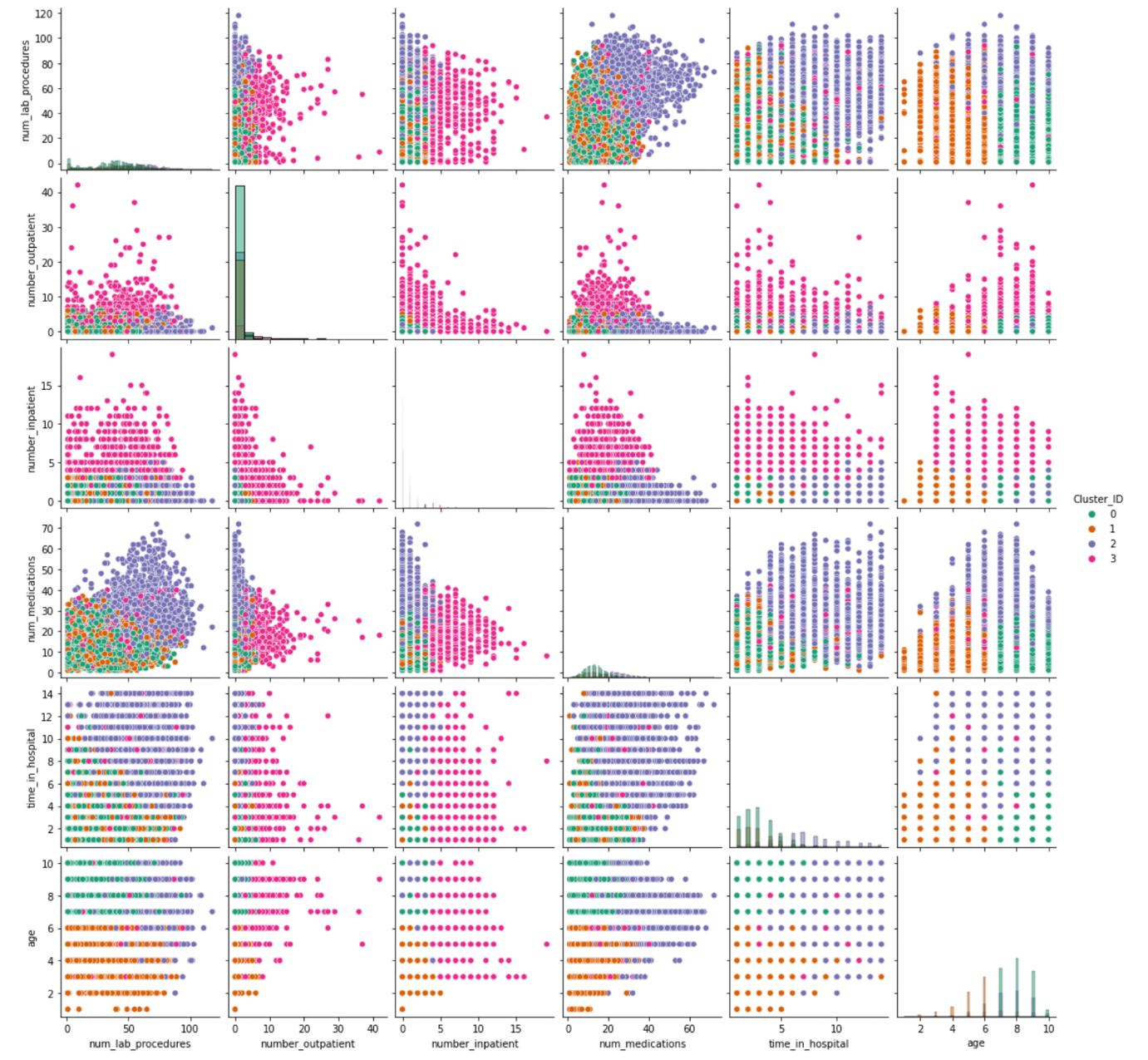


Figure 11. Pairplot of variables from K-Prototypes model

Decision makers could use the outcomes of this study to target specific treatments to each cluster. The varying relationships between age and other variables (between the clusters) could be used to identify the ages at which patients required more or less targeted support from clinicians. The influence of "number of medications" and "number of lab procedures" may be used to identify thresholds at which greater support is required. By identifying these thresholds, decision makers may be more informed when devising guidelines on patient care.

# Project C: Predictive Mining: Decision Tree Models

## Pre Processing

Prior to building the decision tree model, we undertook the following pre-processing steps on the dataset:

1. Checked for errors in the data such as missing values or invalid data by some functions as df.describe(), df.isnull(). We didn’t find some NA or invalid values.
2. Standardise the values of each variable to ensure that extreme values did not appear more important than other variables in the model and can be on the same scale for appropriate comparison and interpretation
3. Categorical variables (specifically ‘age’ and ‘race’) were mapped to appropriate numeric values.
   * The 10 medications were changed depending on No/Steady as 0 and else as 1
   * The change variable was mapped as a binary variable depending on whether there was no change (0) or steady, up or down (1)
   * Gender was mapped as a binary variable where male (1) and female (0)
   * The A1Cresult variable was mapped based on whether it was >7, >8, Norm or None
   * The max\_glu\_serum variable was mapped based on whether it was >200, >300, Norm or None
4. Medical speciality was dropped as there were a lot of invalid values and this variable does not affect the predictive modelling.

The distribution split between the training and test datasets was 70% and 30% respectively. These proportions are common practice in industry.

## Default Setting

### Classification Accuracy

The training and testing accuracy are shown in Figure 12.

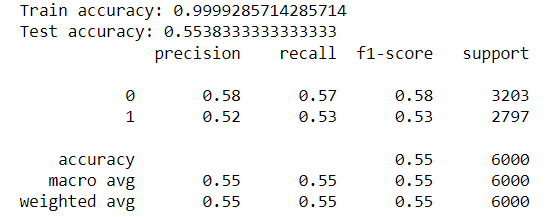


Figure 12. Output of decision tree default model

### Size of Tree

The tree has 4689 nodes and 2345 rules (see Figure 13).

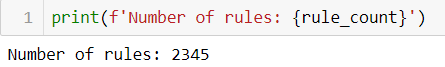
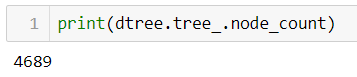


Figure 13. Output showing number of nodes and rules in default decision tree model

### First Split Variable

The variable used in the first split is “number of inpatient”.

### Important Variables

Figure 14 shows the 5 most important variable in building the tree in descending order.

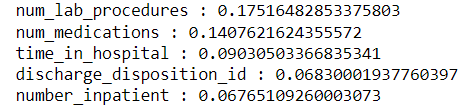


Figure 14. Screenshot of top 5 decision tree variables

### Parameters

The Random State parameter is a number used to control randomness that occurs in generating the shuffle used in the model. This is required to ensure the model provides consistent results every time it is run. For this model, the random state was set to 10.

Criterion is the parameter used to measure the purity of a split, while entropy shows information gain (the uncertainty and randomness) in the model. Information gain is used to select the best attribute at each step in growing the decision tree. Entropy was preferred over Gini because of its accuracy.

Min samples split is the minimum number of samples required to allow a split an internal node. If the number of samples hits the minimum sample split, the decision tree terminates at that node. Min sample split was set to 10 in this model.

## GridSearchCV

### Classification Accuracy

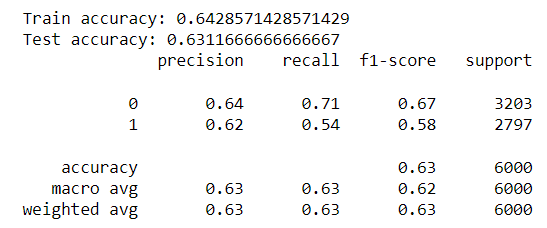


Figure 15. Output of smaller decision tree model, tuned with GridSearchCV

The training and testing accuracy are shown in Figure 15.

### Size of Tree

The tree has 63 nodes and 32 rules (see Figure 16).

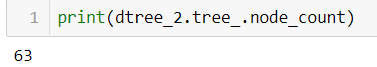
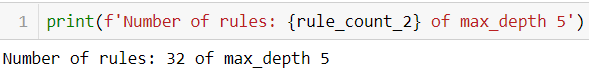


Figure 16. Output showing the number of nodes and rules in the reduced decision tree

### First Split Variable

The variable used in the first split is “number of inpatient”.

### Important Variables

Figure 17 shows the 5 most important variable in building the tree in descending order.

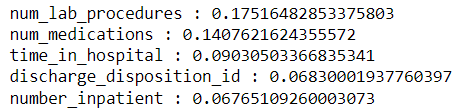


Figure 17. Output of most important variables for smaller decision tree model

### Model Overfitting

The graph demonstrates that model overfitting occurs in the smaller decision tree model after an approximate maximum depth of 5 (see Figure 18).

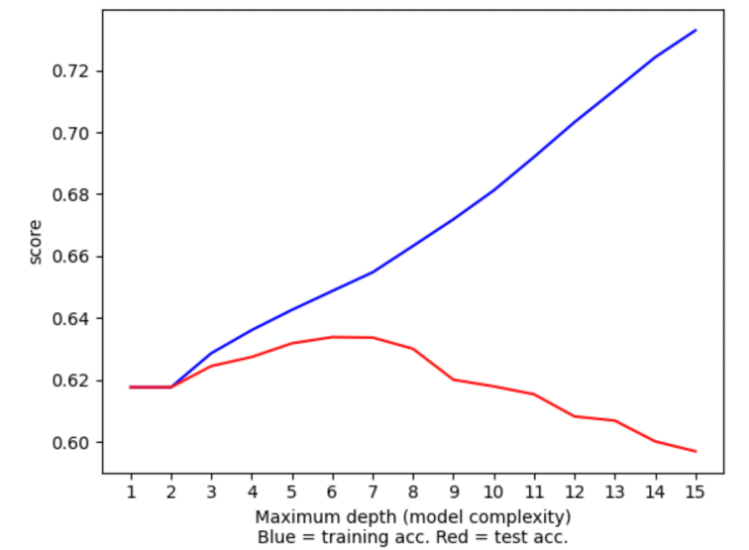


Figure 18. Comparison of test and training accuracy of smaller decision tree model

## Differences between Models

The default decision tree has 209 nodes, whereas the fine-tuned decision tree has only 63 nodes. Additionally, the testing accuracy in the fine-tuned tree is higher (63.11%) compared to the default tree (55.38%).

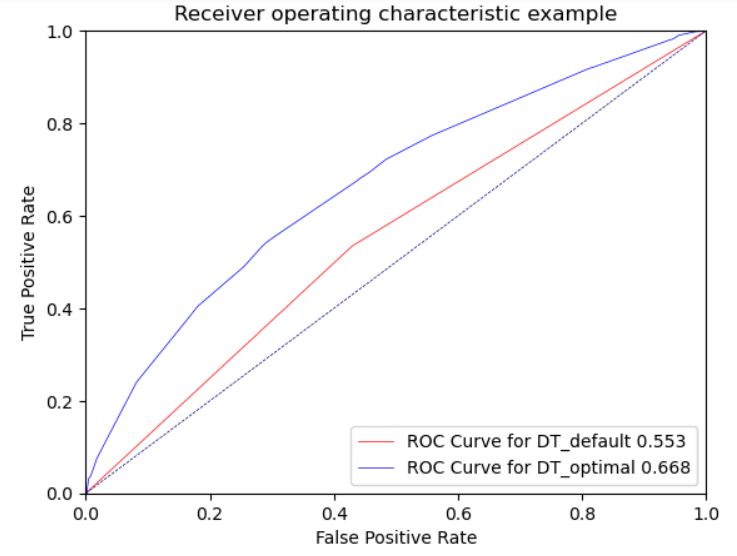


Figure 19. ROC Comparison of default and smaller Decision Tree models

With respect to performance, the fine-tuned decision tree has less nodes and higher testing accuracy in comparison to the default tree. By reducing the maximum depth, the fine-tuned model has been generalised resulting in more identification of true positives in the test data than if it were overfitted to the training data.

## Overall Interpretation

The optimal decision tree can be used to easily identify general characteristics of patients likely to be readmitted (see Table 4). True and False values for each decision rule correspond to left and right arrows on the diagram (see Appendix 1: Optimal Decision Tree).

Table 4. General characteristics of patients likely to be readmitted, from Decision Tree tuned with GridSearchCV.

|  |  |
| --- | --- |
| Leaf | Branch |
| number\_emergency is >0.5 | number\_diagnoses >5.5, diabetesmed is <=0.5, number outpatient <= 0.5, number inpatient <= 0.5 |
| number\_emergency < 0.5 | Age <= 7.5, diabetesmed is > 0.5, number outpatient <= 0.5, number inpatient <= 0.5 |
| insulin > 0.5 | no medication <= 8.5, Number lab procedures >37.5, number outpatients >0.5, number inpatient <= 0.5 |
| discharge\_disposition\_id <= 6.5, | no medication > 8.5, Number lab procedures > 37.5, number outpatients > 0.5, number inpatient <= 0.5 |
| num\_medications <= 11.5 | admission type id <= 2.5, discharge disposition id <= 7.5, number inpatient > 1.5 |
| num\_lab\_procedures > 40.5 | admission type id <= 2.5, discharge disposition id <= 7.5, number inpatient > 1.5 |
| num\_lab\_procedures > 40.5 | admission type Id <= 2.5, discharge disposition id <= 7.5, number inpatient > 1.5 |
| num\_procedures <= 2.5 | discharge disposition id > 14.5, number inpatient > 1.5 |
| number\_outpatient <= 6.5 | Number inpatient <= 3.5, discharge disposition id <= 10, number inpatient > 1.5 |
| age > 6.5 | number inpatient > 3.5, discharge disposition id > 10, number inpatient > 1.5 |
| num\_procedures <= 2.5 | discharge disposition id > 14.5, discharge disposition id > 10, number inpatient > 1.5 |

# Project C: Predictive Mining: Regression Models

## Pre-Processing

Prior to building the regression model, we undertook the following pre-processing steps on the dataset:

1. Checked for errors in the data such as missing values or invalid data by some functions as df.describe(), df.isnull(). We didn’t find some NA or invalid values.
2. Standardise the values of each variable to ensure that extreme values did not appear more important than other variables in the model and can be on the same scale for appropriate comparison and interpretation
3. Categorical variables (specifically ‘age’ and ‘race’) were mapped to appropriate numeric values. In particular:
   * The 10 medications were changed to 0 representing No/Steady and else as 1
   * Change was mapped as a binary variable depending on whether there was no change (0) or steady, up or down (1)
   * Gender was mapped as a binary variable where male (1) and female (0)
   * The A1Cresult variable was mapped based on whether it was >7, >8, Norm or None
   * The max\_glu\_serum variable was mapped based on whether it was >200, >300, Norm or None
4. Medical speciality was dropped as there were a lot of invalid values and this variable does not affect the predictive modelling.

The distribution split between the training and test datasets was 70% and 30% respectively. These proportions are common practice in industry.

## Default Regression Method

### Model Choice

The default model has a higher test accuracy (0.6238, see Figure 20) than GridSearchCV (0.6236, see Figure 21). The training accuracy is slightly higher than the test accuracy for both models. This may indicate overfitting, which needs further investigation. Because the difference in test accuracy between the two models is so small (0.002), it is recommended that the logistic regression model be tuned with GridSearchCV to improve the accuracy.

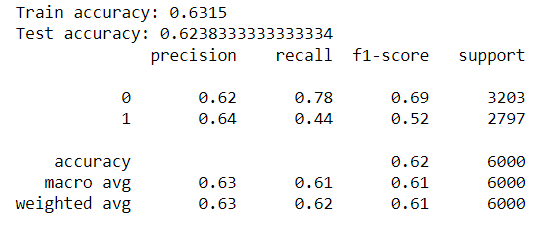


Figure 20. Output from default regression model

### Regression Function



Figure 21. Output from regression model tuned with GridSearchCV

The default regression model used a logistic regression function.

### Standardisation of Variables

The variables in this model were standardised as regression models are sensitive to input variables on different scales. This adversely affects gradient descent (an algorithm used for training models) by making weights of variables on larger scales update much faster than smaller scale variables, resulting in a suboptimal model performance.

### Variables

The variables used in the regression model are:

|  |  |  |
| --- | --- | --- |
| * race, | * num\_medications, | * nateglinide, |
| * gender, | * number\_outpatient, | * chlorpropamide, |
| * age, | * number\_emergency, | * glimepiride, |
| * admission\_type\_id, | * number\_inpatient, | * glipizide, |
| * discharge\_disposition\_id, | * number\_diagnoses, | * glyburide, |
| * admission\_source\_id, | * max\_glu\_serum, | * insulin, |
| * time\_in\_hospital, | * A1Cresult, | * change, |
| * num\_lab\_procedures, | * metformin, | * diabetesMed, and |
| * num\_procedures, | * repaglinide, | * readmitted. |

### Important Variables

Figure shows the 5 most important variable in building the tree in descending order.



Figure 22. 5 most important attributes in default logistic regression model

### Classification Accuracy

The classification accuracy of the training and testing datasets to are shown in Figure 23.



Figure 23. Training and test accuracy of default logistic regression model

### Overfitting

Figure 24: Plot Hyperparameter C values vs training and test accuracy score for default logistic regression model

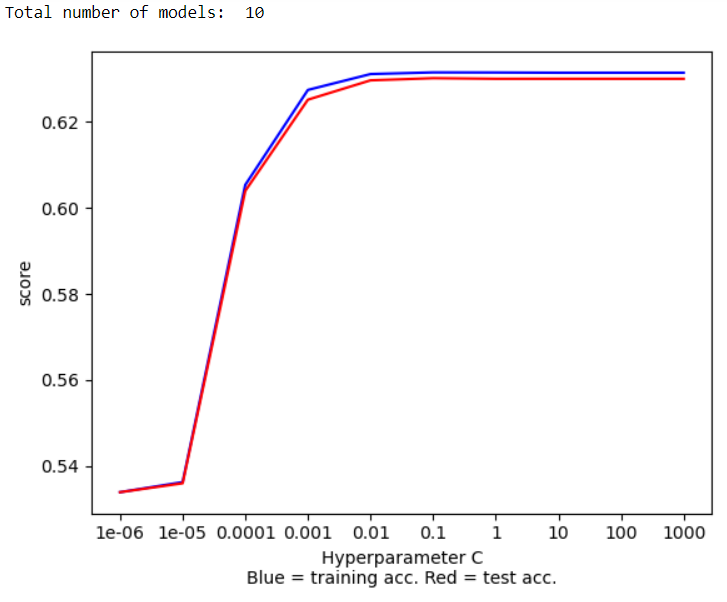


Figure 24 shows that overfitting begins to appear when hyperparameter C is between 0.0001 and 0.001, as this is where the test and training accuracy lines diverge from each other. After C = 0.01 there appears to be no improvement in test accuracy.

## Reduced Regression Method

### Dimensionality Reduction

Recursive feature elimination (RFE) was used to identify a reduced feature set. The original feature set was 29, while the number of features after elimination was 24. REF only slightly improved the test accuracy of the model when compared to the default model (0.6236 vs 0.6243 for the full model).

Dimension reduction was useful as the smaller feature set significantly reduced the speed of training, testing, and prediction processes. The smaller model addresses the risk of the curse dimensionality, which causes measures of distance between instances (critical to similarity calculations) to lose meaning.

### Classification Accuracy

The classification accuracy of the training and testing datasets are shown in Figures 23 and 24.

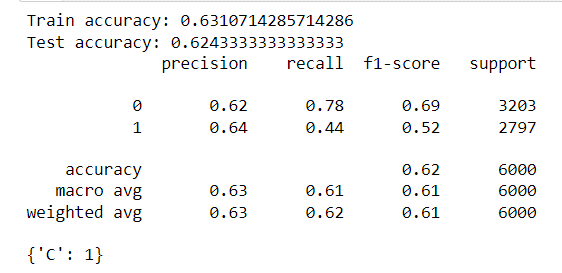


Figure 25. test and training accuracy for regression model with recursive feature elimination

Figure 26. Test and training accuracy scores of reduced logistic regression model



### Overfitting

Overfitting appears to begin when the value of C is between about 0.0001 and 0.001, as is where the test and training accuracy lines diverge from each other. After 0.01 there appears to be no improvement in test accuracy. There is no noticeable improvement in the accuracy performance as compared to the previously built model (see Figure ).

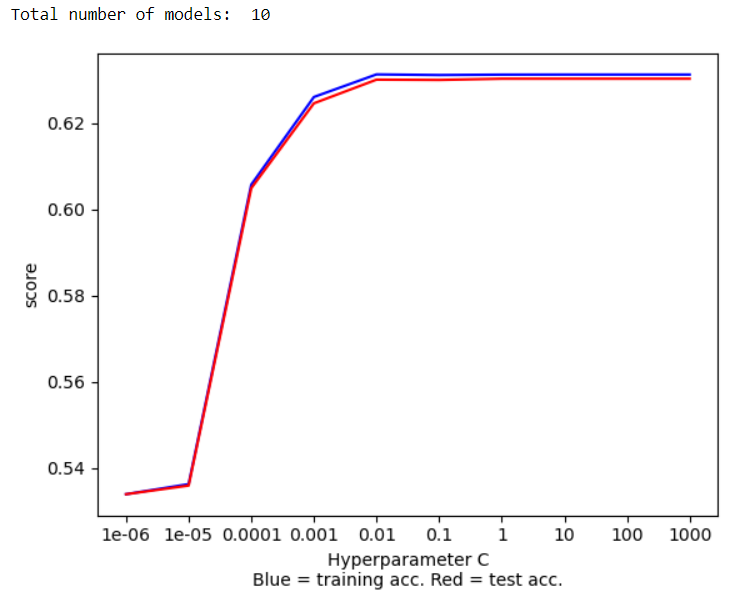


Figure 27. Plot Hyperparameter C values vs training and test accuracy score for reduced regression model

### Top 3 Important Variables

Figure 26 shows the top 3 most important variables.

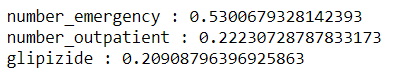


Figure 28. 3 most important attributes in the reduced logistic regression model

## Differences between Models

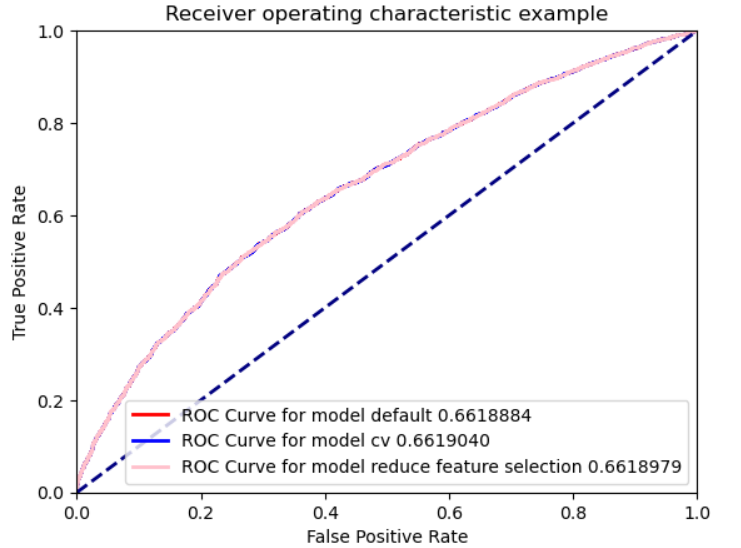


Figure 29. ROC Comparison of default, tunning with GridSearchCV and Feature Selection Logistic Regression models

The model with the highest ROC score is the regression model tuned with GridSearchCV (model cv: 0.6619) (see Figure 29 and Figure 30). This model performed only marginally better than the regression model with recursive feature elimination (model rfe ROC score: 0.6619) and the default model (model default score: 0.6618).

Despite these values, in this context it is appropriate to select the regression model with recursive feature elimination as the benefits of improved model performance, reduced risk of overfitting and enhanced visualisation outweigh the marginal reduction in ROC AUC score (0.0001 difference). This regression model is preferred due to the reduced size of the input dataset and improved time to run the model.

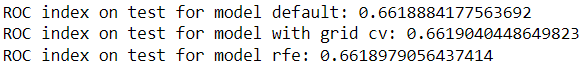


Figure 30. ROC Index scores for default , tunning with GridSearchCV and Feature Selection Logistic Regression models

## Overall Interpretation

The outputs of the logistic regression models are difficult to interpret in terms of identifying patients that may be readmitted. This is because regression models provide a single output value based on all input variables – outputs are not provided for individual input variables. This means that a change in any one input variable can result in a change in the output variable, with no easy identification of what caused the change. Therefore, it would not be feasible to use regression modelling to identify which patients are likely to be readmitted given this dataset.

# Project C: Predictive Mining: Neural Network Models

## Pre-Processing

Prior to building the neural network model, we undertook the following pre-processing steps on the dataset:

1. Checked for errors in the data such as missing values or invalid data by some functions as df.describe(), df.isnull(). We didn’t find some NA or invalid values.
2. Standardised the values of each variable to ensure that:
   * extreme values did not appear more important than other variables in the model
   * to allow for appropriate comparison and interpretation of results.
3. Categorical variables (specifically ‘age’ and ‘race’) were mapped to appropriate numeric values. Of note:
   * The 10 medications were changed depending on No/Steady as 0 and else as 1
   * The change variable was mapped as a binary variable depending on whether there was no change (0) or steady, up or down (1)
   * Gender was mapped as a binary variable where male (1) and female (0)
   * The A1Cresult variable was mapped based on whether it was >7, >8, Norm or None
   * The max\_glu\_serum variable was mapped based on whether it was >200, >300, Norm or None
4. Medical speciality was dropped as there were a lot of invalid values and this variable does not affect the predictive modelling.

We used this dataset readmitted as predicted value. The distribution split between the training and test datasets was 70% and 30% respectively. These proportions are common practice in industry.

## Default Neural Network

### Model Parameters

The random state parameter was used and set to "10" in the default neural network. The alpha, which is the learning rate for the gradient descent algorithm, was set to 0.0001 with 5 hidden layers. We used a neural network classifier is implemented in MLPClassifier. MLPClassifier object hyperparameters is set a solver hyperparameter to adam. And we used relu as the activation function in this practical.

### Classification Accuracy

Figure shows the training and testing classification accuracy of the datasets are:

This default neural network (only use random state is 10) performed with high accuracy on the training dataset. However, the test accuracy is comparatively much lower (59%), leading to overfitting to the training data. We should also notice a convergence warning in this step because this neural network does not achieve convergence before maximum iteration.

* Train accuracy: 0.73235.…
* Test accuracy: 0.598333….

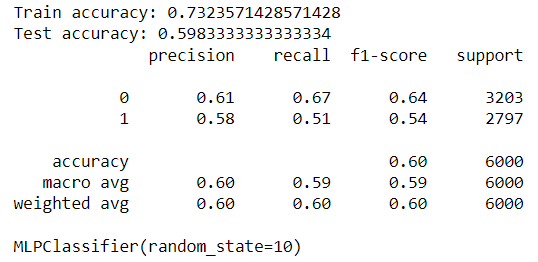


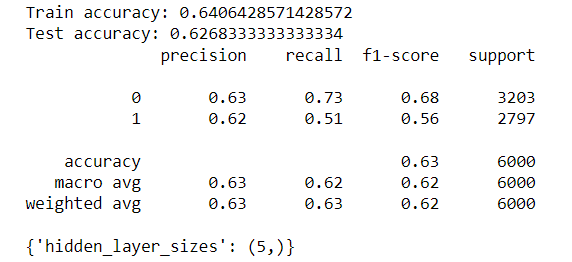
Figure 31. test and training accuracy for default neural network model with Convergence Warning

### Training Process

From the default model with 29 features, GridSearchCV was used to find the optimal hyperparameters for the model. It returned a hidden layer value of 5 neurons and the default alpha value (L2 regularization parameter used in each neuron's activation function) of 0.0001 as the optimal (Figure shows the default model accuracy whereas Figure 31 on page 20 shows the improved model accuracy).

In this the best model, no warning about convergence was displayed. So, it was concluded that this model achieve convergence before maximum iteration in sklearn.

Figure 32: test and training accuracy for default neural network model with GridSearchCV



### Over Fitting

There is a sign of overfitting in default model without using any hyperparameters exclude random state (see Figure ) which higher accuracy on the training dataset (73%) and test accuracy is lower (59%). So, we used the GridSearch tuning to control the overfitting of the model to the training data. After tunning, there was no sign of over-fitting. GridSearchCV tuning of ‘hidden\_layer\_sizes’ (optimal 5), ‘alpha’ (default 0.0001).

## Reduced Neural Network

### Feature Selection and Inputs

Performing feature selection using the decision tree features, identified a better model for this dataset. There was a change with 7 hidden layers (default model is 5) and 8 features (default model is 29), alpha 0.01. Dimensionality reduction and transformation techniques were then used to reduce the size of the feature set and improve the performance of neural network models (Figure 31).

### Classification Accuracy

The training and testing classification accuracy of the datasets is shown in the screenshot below. They have better accuracy comparing with the best model in the default neural network Figure 31.

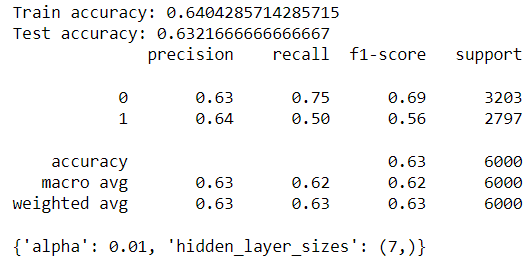


Figure 33. Training and test accuracy for reduced neural network using decision tree

### Iteration

In running the feature selection model, a "convergence is not reached" warning message was not raised. It was concluded that MLPClassifier model used default iteration as 200.

### Overfitting and Training process convergence

There was no sign of over-fitting. In running this feature selection model, a "convergence is not reached" warning message was not raised, so this neural network achieved convergence before maximum iteration.

## Differences between Models

The neural network with recursive feature selection and decision tree features (‘NN with cv\_sel\_model (DT)’ in Figure 34) has marginally the highest ROC value (0.675). The values in Figure and the ROC curve comparisons in Figure illustrate how small the range is (ranging between 0.633-0.669).

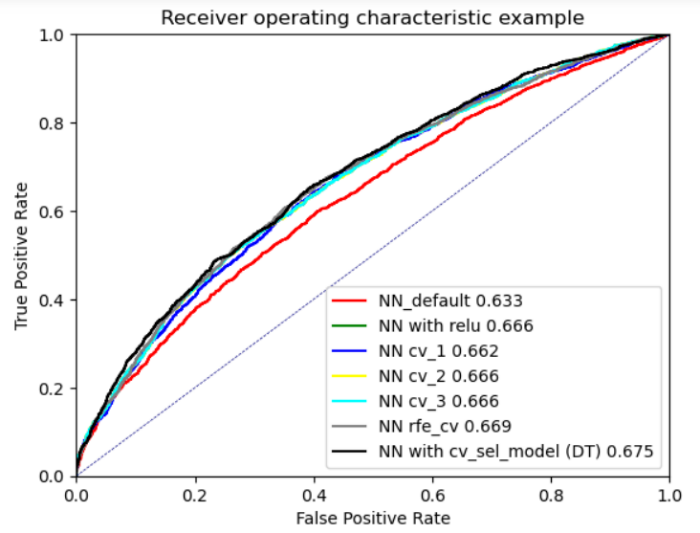


Figure 34. ROC Comparison of default and smaller Neural Network models

The ROC plot is used to show the ability of each model to correctly predict true positive values. The model with the closest curve to the top left corner (location of 100% True Positive rate) of the plot, is the better model of the set tested. The model with feature selection with Decision Tree shows the largest area under curve than the other models. Thus, "NN with ‘cv\_sel\_model (DT)’" is the best performing model.

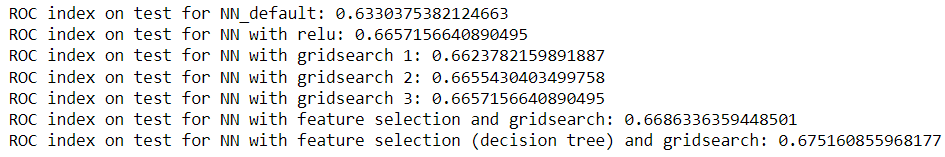


Figure 35. ROC Index scores for default and smaller Neural Network models

## Overall Interpretation

It is difficult to identify the general characteristics of patients that could potentially be readmitted as neural networks considers numerous input attributes which are processed via many hidden layers over hundreds or thousands of iterations. The vastness of neural network models makes it impossible decipher which individual characteristics are making a tangible impact on the output value. For instance, a change in the weighting of one attribute can result in a change for the output attribute in the model, with very little explanation of how. Therefore, it would not be feasible to use neural network modelling to identify who is likely to be readmitted based off this dataset.

# Project C: Predictive Mining Comparison

## Model Choice

Figure 36 and 37 show the best neural network model (identified in section 4) of Project C: Predictive Mining: Neural Network Models) has the highest ROC value (0.669) and performs marginally better than the best decision tree or best logistic regression models.

On balance, when considering model accuracy, outputs and processing time the most useful model for decision makers is the optimal Decision Tree. This model provides sufficient information to allow medical practitioners to predict whether a future patient, given the characteristics they display, will be readmitted (reported as 1) or not (reported as 0). This model does have a slightly lower ROC score but this is very marginal (0.007).

Although the Logistic Regression and Neural Network models achieved a higher ROC score, their outputs lack the granular information about each variables characteristics to allow decision makers to predict whether a patient would be readmitted or not, given their general characteristics. They each provide a likelihood value for whether the patient would be readmitted, but no explanation as to why this output value has been provided.

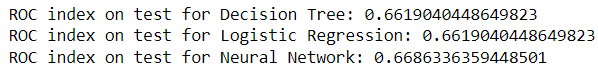


Figure 36. ROC index scores for best Decision Tree, Logistic Regression and Neural Network Models.

## Positives and Negatives of Predictive Modelling Methods

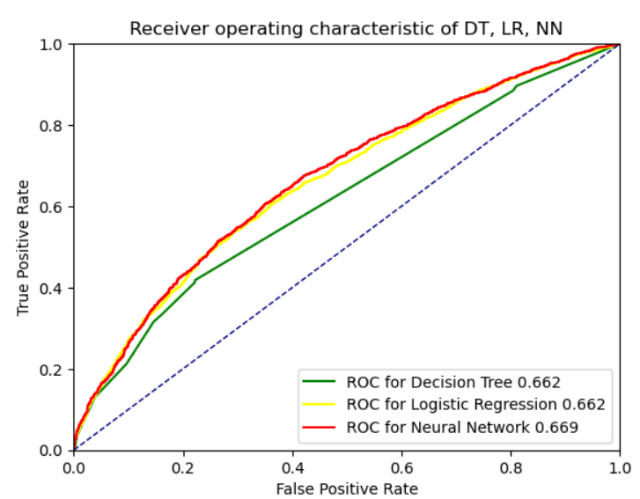


Figure 37. ROC Comparison of best Decision Tree, Logistic Regression and Neural Network models

Table 5 provides a summary of the strengths and weaknesses of the three predictive models developed as part of this study and their ability to identify the general characteristics of a patient that is likely to be readmitted.

Table 5. Pro's and Con's comparison of the predictive models used in this study.

|  |  |
| --- | --- |
| Pro's | Cons |
| Decision Tree | |
| * Provides comprehensive summary of characteristics of patients likely to be readmitted to hospital * Easy to interpret by most people * Required less pre-processing of data as it can handle missing values, and differing data distributions | * Does not provide specific measure of likelihood for individual patients * Default decision tree was so large it was impossible to interpret. |
| Logistic Regression | |
| * Provides specific measure of likelihood of individual patients being readmitted to hospital * Relatively short processing time * Regression function relatively easy to identify the proportional impact of variables on output value, when compared to Neural Network. | * Does not provide comprehensive reasons why a patient may be readmitted * Required significant pre-processing to transform data into useable format (eg, mapping categorical variables to numbers and standardising numeric variables). |
| Neural Network | |
| * Provides specific measure of likelihood of individual patients being readmitted to hospital. * Achieved highest ROC (accuracy) score of all models. | * Does not provide comprehensive reasons why a patient may be readmitted * Slow Processing time for training model * Multi-step process for tuning model (tune each hyperparameter individually) * Black box model, difficult to explain to decision makers how the model works. |

# Appendix 1: Optimal Decision Tree

Diagram

Description automatically generated with low confidence