VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY
INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DATA MINING

FINAL REPORT

Course by Prof. Nguyen Thi Thanh Sang

MSc. Nguyen Quang Phu

TOPIC:
STROKE DETECTION

BY GROUP C – MEMBER LIST

1.	Nguyễn Hoàng Hồng Ân	ITDSIU22151	Team Leader
2.	Phạm Nguyễn Quỳnh Anh	ITDSIU22130	Team Member
3.	Đoàn Võ Thảo My	ITDSIU22138	Team Member
4.	Nguyễn Phúc Minh Quân	ITDSIU22163	Team Member
5.	Châu An Phú	ITDSIU22158	Team Member

Table of Contents

Chapter 1: Introduction

1.1 Abstract

A stroke is a medical condition that happens when the oxygen flow to the brain is interrupted, causing the brain cells lacking oxygen and nutrients and starting to die. If not treated during the early stage, a stroke may leave the patients with life-long consequences from minor symptoms such as impaired cognitive and language abilities. According to the World Health Organization stroke is the second most leading cause of death globally, responsible for approximately 11% of total deaths.

However, traditional stroke diagnosing techniques such as MRI or CT scan can be time-consuming, expensive as well as requiring high expertise. As a result, the increasing need for efficient and accurate stroke diagnosis, coupled with the continuous advancements in ML, is driving the development of these innovative tools. As ML technology matures, it has the potential to revolutionize stroke diagnosis, leading to improved patient care and ultimately saving lives.

1.2 Project Overview

This project extracts knowledge by analyzing *the stroke prediction dataset* [1] to gain understanding of the relationship between the likelihood of having a stroke and the patient's demographic, medical history and genetic information.

The data mining process is divided into data gathering, data preprocessing, data modeling and evaluation.

In the first step, the data is collected from *the stroke prediction dataset* as a csv file which is a tabular data containing rows and columns, each row represents a patient, each column contains different attributes such as demographic and health conditions. The raw data may contain missing value, duplicated value or redundant features; therefore, it is handled during the data preprocessing stage. The refined data is then split into the training set and testing set. The training set is used to train classification models such as J48, Random Forest, Naive Bayesian, the models then make predictions on the testing set. The result is evaluated for the by various metrics including accuracy, recall, f1-score.

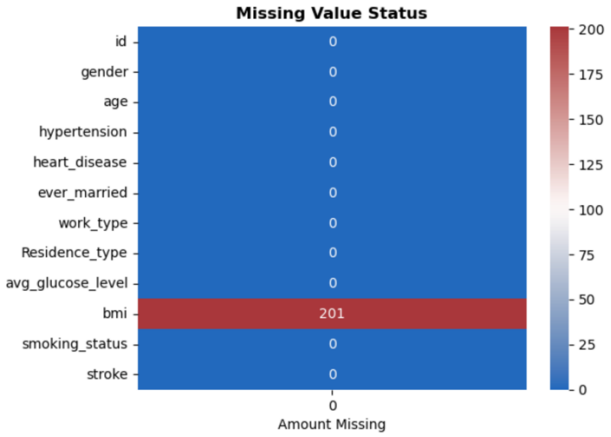
1.3 Goal

Through the data mining process, the trained classification models are expected to provide fast and reliable preliminary diagnosis, helping to reduce the time and workload for the doctors. Furthermore, based on the fitted classification rules, the models can provide insights into the pattern of stroke, accelerating the doctor's process in developing prevention and treatment for the patients.

Chapter 2: Project Analysis

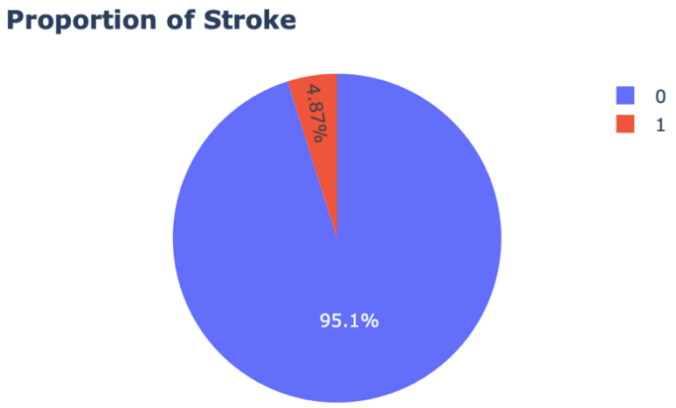
2.1 Data processing – Exploratory Data analysis

- Handle missing values in BMI column (N/A values)

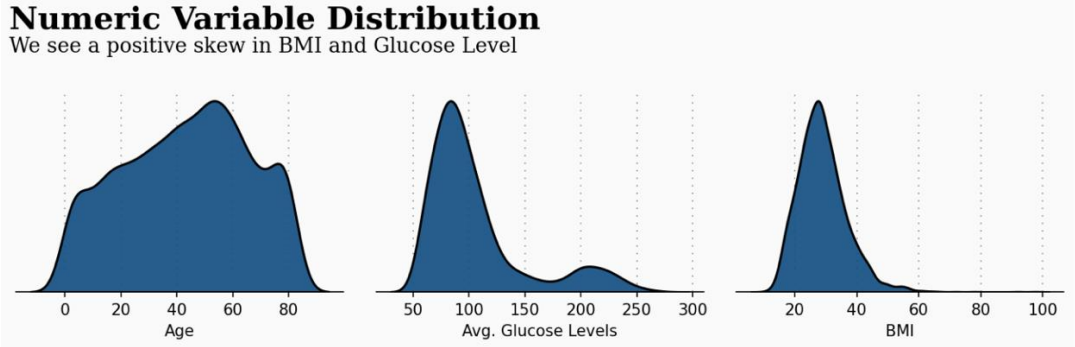


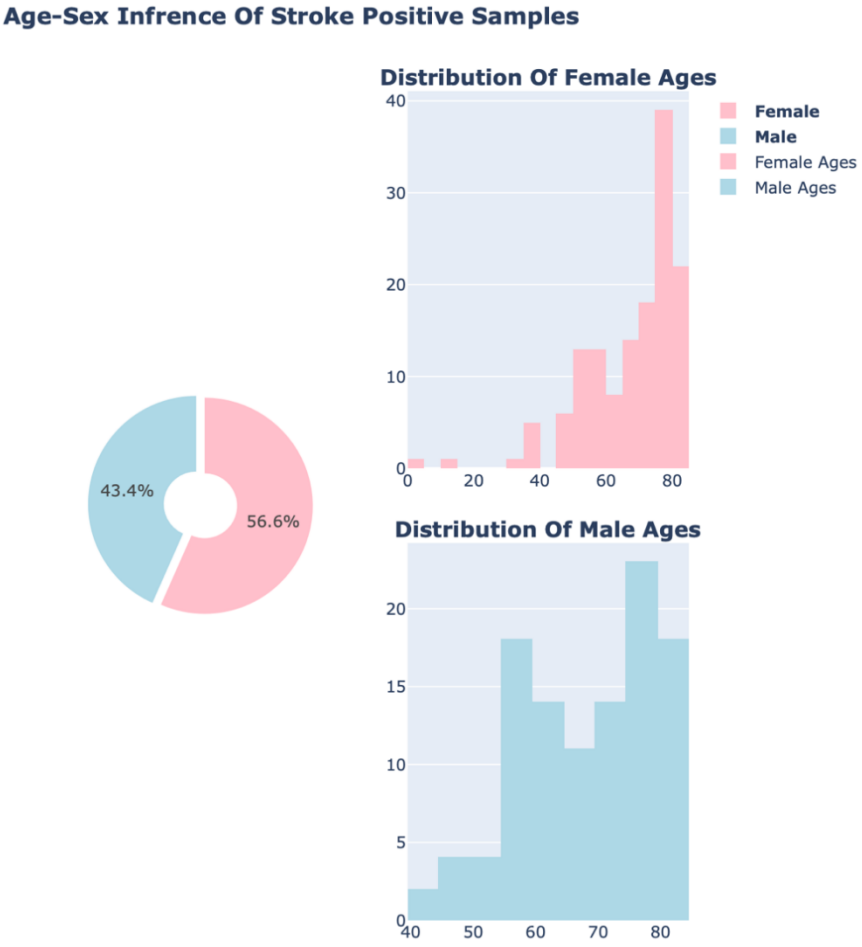
⇨ The dataset had **201 samples** with **absent** BMI value; we impute it with the median.

```
import statistics

def replace_with_median(df):

    return df.copy().fillna({'bmi': df['bmi'].median()})

df_new = replace_with_median(df.copy())

print(df_new.head(3))
```

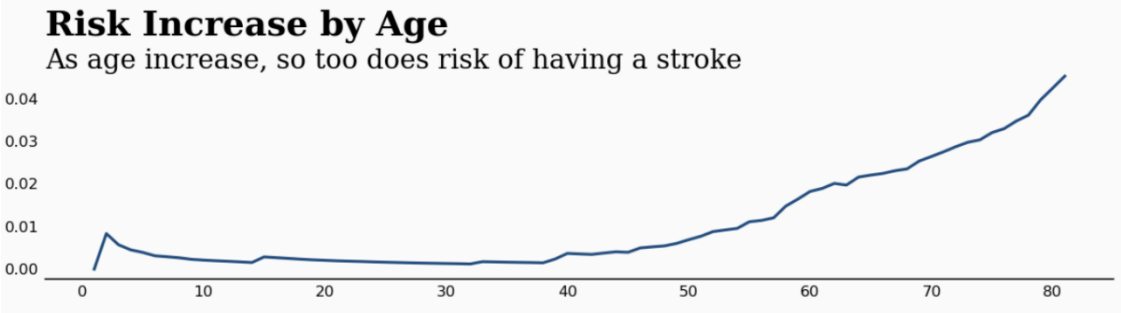


⇨ We are dealing with an **imbalanced** dataset.

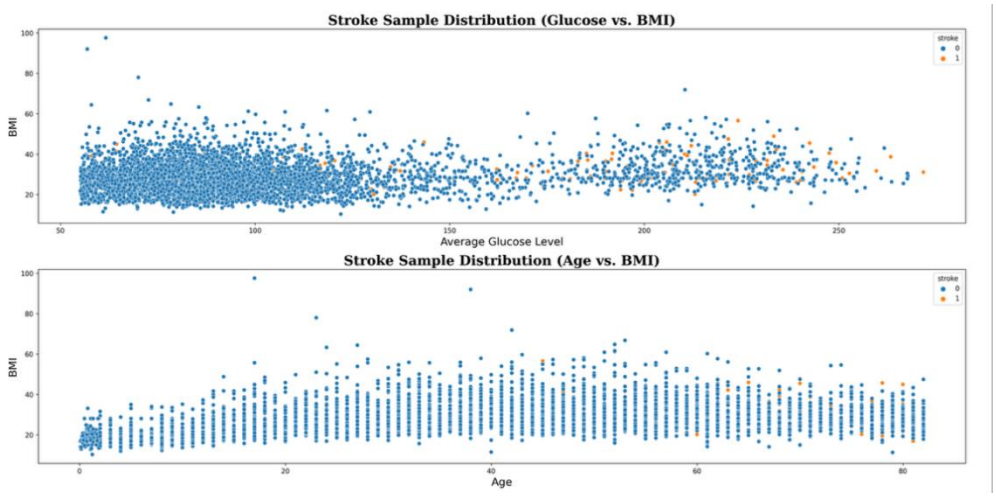




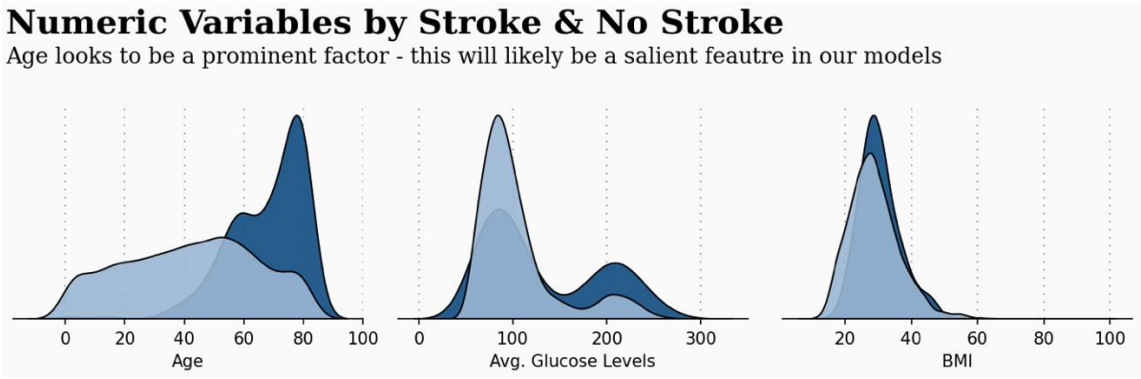
⇨ When looking at the inner distribution of our different attributes among stroke positive samples, we see that female, although appearing more than males in our dataset, also surpass the males in the stroke sample space, the second point to be noted is that males are more prone to strokes in their **early 50/60** where the median of the women stroke age is around **75-79.**



⇨ The older you get, the more at risk you get. However, you may have noticed the low-risk values on the y-axis. This is because the **dataset is highly imbalanced**. Only 249 strokes are in our dataset which totals 5000 - around 1 in 20.



⇨ In both scatterplots the individuals who had a stroke are in the BMI value region **under 60** and in **high glucose levels** as well as old age.

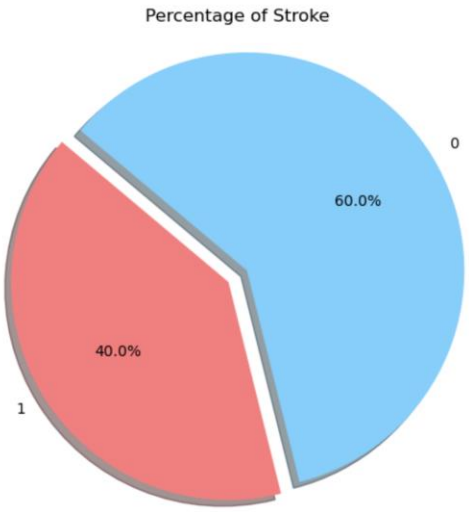


⇨ Based on the above plots, it seems clear that **Age is a big factor** in stroke patients - the older you get the more at risk you are.

- SMOTE (Oversampling): SMOTE is an oversampling technique designed to increase the number of instances in the minority class by creating synthetic examples rather than simply duplicating existing ones. This approach helps to prevent overfitting that can occur with simple duplication.

- Undersampling: Undersampling is another technique used to handle class imbalance by reducing the number of instances in the majority class. This approach aims to balance the class distribution by removing some of the majority class instances.

  Given that our dataset does not meet the necessary conditions for effective training due to its imbalanced nature ,in this case, class 1 (stroke) has 249 instances while class 0 (non-stroke) has 4861 instances. We will implement both the SMOTE (Synthetic Minority Over-sampling Technique) and undersampling methods. This combined approach will help balance the class distribution, thereby enhancing the performance and reliability of our predictive models.



For the Apriori algorithm, which only accepts categorical attributes

```
@attribute id numeric
@attribute gender {Male,Female,Other}
@attribute age numeric
@attribute hypertension {1, 0}
@attribute heart_disease {1, 0}
@attribute ever_married {Yes,No}
@attribute work_type {Private,Self-employed,Govt_job,children,Never_worked}
@attribute Residence_type {Urban,Rural}
@attribute avg_glucose_level numeric
@attribute bmi numeric
@attribute smoking_status {'formerly smoked','never smoked',smokes,Unknown}
@attribute stroke {0, 1}
```

we will preprocess the dataset following these steps:

1. Remove all unnecessary numeric attributes, such as "id".

```
@attribute gender {Male,Female,Other}
@attribute age numeric
@attribute hypertension {1, 0}
@attribute heart_disease {1, 0}
@attribute ever_married {Yes,No}
@attribute work_type {Private,Self-employed,Govt_job,children,Never_worked}
@attribute Residence_type {Urban,Rural}
@attribute avg_glucose_level numeric
@attribute bmi numeric
@attribute smoking_status {'formerly smoked','never smoked',smokes,Unknown}
@attribute stroke {0, 1}
```

2. Transform the remaining numeric attributes into three categorical bins like “age”, “avg_glucose_level” , “bmi” which is important.

```
@attribute gender {Male,Female,Other}
@attribute age {low,medium,high}
@attribute hypertension {1, 0}
@attribute heart_disease {1, 0}
@attribute ever_married {Yes,No}
@attribute work_type {Private,Self-employed,Govt_job,children,Never_worked}
@attribute Residence_type {Urban,Rural}
@attribute avg_glucose_level {low,medium,high}
@attribute bmi {low,medium,high}
@attribute smoking_status {'formerly smoked','never smoked',smokes,Unknown}
@attribute stroke {0, 1}
```

3. Exclude the class attribute "Stroke" from the dataset.

```
@attribute gender {Male,Female,Other}
@attribute age {low,medium,high}
@attribute hypertension {1, 0}
@attribute heart_disease {1, 0}
@attribute ever_married {Yes,No}
@attribute work_type {Private,Self-employed,Govt_job,children,Never_worked}
@attribute Residence_type {Urban,Rural}
@attribute avg_glucose_level {low,medium,high}
@attribute bmi {low,medium,high}
@attribute smoking_status {'formerly smoked','never smoked',smokes,Unknown}
```

2.2 Classification Algorithms

2.2.1 J48 – Decision tree

The J48 algorithm is an implementation of the C4.5 decision tree algorithm, which is used for classification tasks.

In the context of predicting strokes, J48 builds a decision tree based on the attributes in the dataset, such as age, gender, hypertension, and lifestyle factors like smoking status and BMI. The algorithm splits the data at each node based on the attribute that provides the highest information gain, thus creating a tree that can be used to make predictions.

The final model consists of a series of if-else rules derived from the tree, making it interpretable and straightforward to understand.
For this dataset, J48 helps in identifying patterns and relationships between the attributes and the likelihood of a stroke, providing a visual and logical representation of the decision-making process.

2.2.2 Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the class that is the mode of the classes of the individual trees.

In predicting strokes, Random Forest takes advantage of the diversity of multiple trees to improve prediction accuracy and handle overfitting. Each tree in the forest is built on a random subset of the data and considers a random subset of features when splitting nodes, ensuring a wide exploration of the feature space.

This approach is particularly effective for the stroke dataset as it captures the complex interactions between various attributes such as average glucose level, BMI, and work type, and combines the insights from numerous trees to make robust predictions about stroke risk.

2.2.3 Naive Bayes

Naive Bayes is a probabilistic classification technique based on Bayes' theorem, assuming strong independence between the features.

When applied to stroke prediction, Naive Bayes calculates the posterior probability of a stroke given the input features by combining the prior probability of a stroke with the likelihood of observing the given features in stroke and non-stroke cases. Despite its simplicity and the assumption of feature independence, Naive Bayes performs surprisingly well in many practical situations.

For this dataset, it leverages the distribution of continuous variables like age and average glucose level, and categorical variables like gender and smoking status, to compute probabilities and classify individuals as at risk of stroke or not. Its efficiency and effectiveness make it a valuable model for quick, initial analysis of stroke risk.

2.2.4 OneR

The OneR algorithm is a simple yet powerful classification method that generates a single rule for each feature and selects the rule with the lowest error rate. In the context of predicting strokes, OneR evaluates each attribute in the dataset, such as age, gender, hypertension, and lifestyle factors, and creates a rule based on the attribute that best separates the stroke and non-stroke cases.

For instance, OneR might create a rule such as "if age > 50, then predict stroke," if age is the most informative attribute. Despite its simplicity, OneR often produces surprisingly accurate models and serves as a useful baseline for more complex models. It is particularly valuable for understanding the primary attribute influencing stroke risk in the dataset, offering a clear and easily interpretable model.

2.2.5 SimpleLogistic

SimpleLogistic is a classifier that builds a logistic regression model using a simple iterative method. Logistic regression is a statistical model that predicts the probability of a binary outcome, such as the presence or absence of stroke, based on one or more predictor variables.

In predicting strokes, SimpleLogistic uses attributes like age, BMI, average glucose level, and lifestyle factors to estimate the likelihood of a stroke. The algorithm fits a logistic function to the data, providing coefficients for each attribute that describe its influence on stroke risk. The resulting model is interpretable, as the coefficients indicate the direction and magnitude of the relationship between each attribute and the outcome.

SimpleLogistic is effective for this dataset because it can handle both continuous and categorical variables, and it provides insights into the relative importance of different risk factors for stroke. Its straightforward probabilistic framework allows for clear communication of the risk associated with each attribute.

2.3 Sequence Mining Method

Sequence mining is a data mining technique used to discover sequential patterns or relationships in data. One prominent application of sequence mining is in analyzing sequences of events or transactions over time. One common algorithm used for sequence mining is the Apriori algorithm.

Moreover, sequence mining can also uncover rare but significant patterns that may not be apparent through traditional statistical analysis. These patterns can provide novel insights into the complex dynamics of stroke development and help refine predictive models for better accuracy.

Overall, sequence mining offers a powerful tool for unraveling the temporal dynamics of stroke risk factors and facilitating more targeted interventions for stroke prevention.

2.4 Model Evaluation

2.4.1 Random Forest

```
Training data using RandomForest
Evaluation results:

Correctly Classified Instances         166               83      %
Incorrectly Classified Instances        34               17      %
Kappa statistic                          0.6488
Mean absolute error                      0.2639
Root mean squared error                  0.3515
Relative absolute error                 54.9707 %
Root relative squared error             71.7552 %
Total Number of Instances              200

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.842     0.188     0.871       0.842   0.856      0.649    0.902     0.944     0
               0.813     0.158     0.774       0.813   0.793      0.649    0.902     0.825     1
Weighted Avg.  0.830     0.176     0.832       0.830   0.831      0.649    0.902     0.897

===Overall Confusion Matrix===

   a   b   <-- classified as
 101  19 |   a = 0
  15  65 |   b = 1

10-fold cross validation evaluation results:

Correctly Classified Instances         647               80.875  %
Incorrectly Classified Instances       153               19.125  %
Kappa statistic                          0.6063
Mean absolute error                      0.2907
Root mean squared error                  0.3773
Relative absolute error                 60.5474 %
Root relative squared error             77.011  %
Total Number of Instances              800

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.817     0.203     0.858       0.817   0.837      0.607    0.871     0.915     0
               0.797     0.183     0.743       0.797   0.769      0.607    0.871     0.791     1
Weighted Avg.  0.809     0.195     0.812       0.809   0.810      0.607    0.871     0.865

===Overall Confusion Matrix===

   a   b   <-- classified as
 392  88 |   a = 0
  65 255 |   b = 1
```

Evaluation

- The Random Forest model shows a commendable accuracy level of 83% on the training data and 80.875% with 10-fold cross-validation. These accuracy figures indicate that the model is effective at correctly classifying the majority of instances in both the training and validation phases.
- The Relative Absolute Error (RAE) of 60.5475% with 10-fold cross-validation indicates that, on average, the model's predictions are off by approximately 60.5475% of the mean absolute value of the actual target values.

2.4.2 J48 – Decision tree

```
Training data using J48
Evaluation results:

Correctly Classified Instances         149               74.5    %
Incorrectly Classified Instances        51               25.5    %
Kappa statistic                          0.4654
Mean absolute error                      0.3103
Root mean squared error                  0.4498
Relative absolute error                 64.6356 %
Root relative squared error             91.8141 %
Total Number of Instances              200

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.800    0.338    0.780      0.800    0.790      0.466    0.763     0.794     0
                 0.663    0.200    0.688      0.663    0.675      0.466    0.763     0.629     1
Weighted Avg.    0.745    0.283    0.744      0.745    0.744      0.466    0.763     0.728

===Overall Confusion Matrix===

  a  b   <-- classified as
 96 24 |  a = 0
 27 53 |  b = 1

10-fold cross validation evaluation results:

Correctly Classified Instances         619               77.375  %
Incorrectly Classified Instances       181               22.625  %
Kappa statistic                          0.5323
Mean absolute error                      0.2977
Root mean squared error                  0.4353
Relative absolute error                 62.0238 %
Root relative squared error             88.856  %
Total Number of Instances              800

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.796    0.259    0.822      0.796    0.808      0.533    0.771     0.791     0
                 0.741    0.204    0.707      0.741    0.724      0.533    0.771     0.641     1
Weighted Avg.    0.774    0.237    0.776      0.774    0.775      0.533    0.771     0.731

===Overall Confusion Matrix===

   a   b   <-- classified as
 382  98 |   a = 0
  83 237 |   b = 1
```

\

Evaluation

- The J48 - decision tree model demonstrates reasonable accuracy levels, with 74.5% accuracy on the training data and 77.375% with 10-fold cross-validation. These results indicate the model's capability to correctly classify 77% of instances.
- The Relative Absolute Error (RAE) of 62.0238% with 10-fold cross-validation indicates that, on average, the model's predictions are off by approximately 62.0238% of the mean absolute value of the actual target values.

2.4.3 Naive Bayes

```
Training data using NaiveBayes
Evaluation results:

Correctly Classified Instances         141               70.5    %
Incorrectly Classified Instances        59               29.5    %
Kappa statistic                          0.4088
Mean absolute error                      0.3196
Root mean squared error                  0.4603
Relative absolute error                 66.5795 %
Root relative squared error             93.9504 %
Total Number of Instances              200

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.675    0.250    0.802      0.675    0.733      0.416    0.784     0.872     0
                 0.750    0.325    0.606      0.750    0.670      0.416    0.784     0.623     1
Weighted Avg.    0.705    0.280    0.724      0.705    0.708      0.416    0.784     0.772

===Overall Confusion Matrix===

  a  b   <-- classified as
 81 39 |  a = 0
 20 60 |  b = 1

10-fold cross validation evaluation results:

Correctly Classified Instances         598               74.75   %
Incorrectly Classified Instances       202               25.25   %
Kappa statistic                          0.4836
Mean absolute error                      0.2827
Root mean squared error                  0.4296
Relative absolute error                 58.8912 %
Root relative squared error             87.6981 %
Total Number of Instances              800

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.752    0.259    0.813      0.752    0.781      0.486    0.826     0.887     0
                 0.741    0.248    0.666      0.741    0.701      0.486    0.826     0.719     1
Weighted Avg.    0.748    0.255    0.754      0.748    0.749      0.486    0.826     0.820

===Overall Confusion Matrix===

   a   b   <-- classified as
 361 119 |   a = 0
  83 237 |   b = 1
```

Evaluation

- The Naïve Bayes model demonstrates reasonable performance with an accuracy of 70.5% on the training data and 74.75% with 10-fold cross-validation. These accuracy figures indicate that the model can classify most of instances correctly.
- The Relative Absolute Error (RAE) of 58.8912% with 10-fold cross-validation indicates that, on average, the model's predictions are off by approximately 58.8912% of the mean absolute value of the actual target values.

2.4.4 OneR

```
Training data using OneR
Evaluation results:

Correctly Classified Instances         149               74.5    %
Incorrectly Classified Instances        51               25.5    %
Kappa statistic                          0.4828
Mean absolute error                      0.255
Root mean squared error                  0.505
Relative absolute error                 53.1195 %
Root relative squared error            103.0776 %
Total Number of Instances              200

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.733    0.238    0.822      0.733    0.775      0.487    0.748     0.763     0
                 0.763    0.267    0.656      0.763    0.705      0.487    0.748     0.595     1
Weighted Avg.    0.745    0.249    0.756      0.745    0.747      0.487    0.748     0.696

===Overall Confusion Matrix===

  a  b   <-- classified as
 88 32 |  a = 0
 19 61 |  b = 1

10-fold cross validation evaluation results:

Correctly Classified Instances         597               74.625  %
Incorrectly Classified Instances       203               25.375  %
Kappa statistic                          0.4694
Mean absolute error                      0.2537
Root mean squared error                  0.5037
Relative absolute error                 52.8591 %
Root relative squared error            102.8247 %
Total Number of Instances              800

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.796    0.328    0.784      0.796    0.790      0.469    0.734     0.747     0
                 0.672    0.204    0.687      0.672    0.679      0.469    0.734     0.593     1
Weighted Avg.    0.746    0.279    0.745      0.746    0.746      0.469    0.734     0.685

===Overall Confusion Matrix===

   a   b   <-- classified as
 382  98 |   a = 0
 105 215 |   b = 1
```

Evaluation

- The OneR model, which is a simple rule-based classifier, shows an accuracy of 74.5% on the training data and 74.625% with 10-fold cross-validation. These accuracy levels indicate that the model is effective at correctly classifying a majority of instances.
- The Relative Absolute Error (RAE) of 52.8591% with 10-fold cross-validation indicates that, on average, the model's predictions are off by approximately 52.8591% of the mean absolute value of the actual target values.

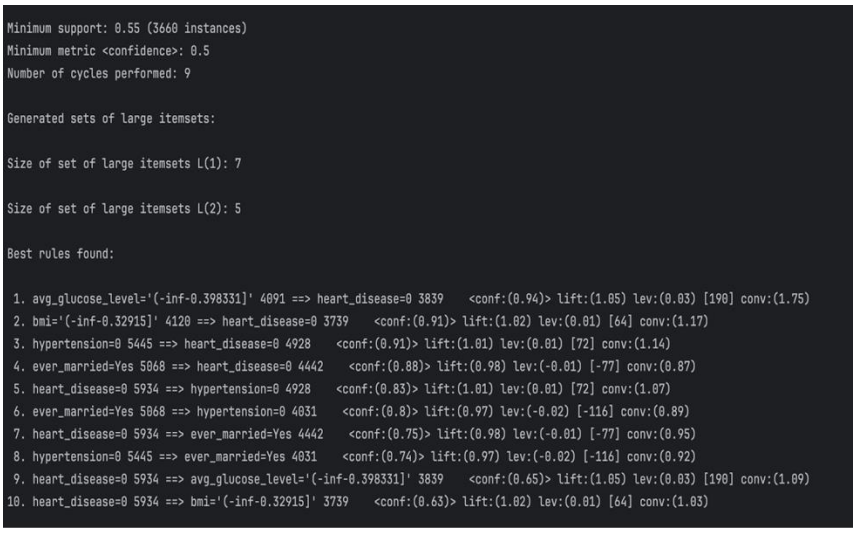
2.4.5 Simple Logistic

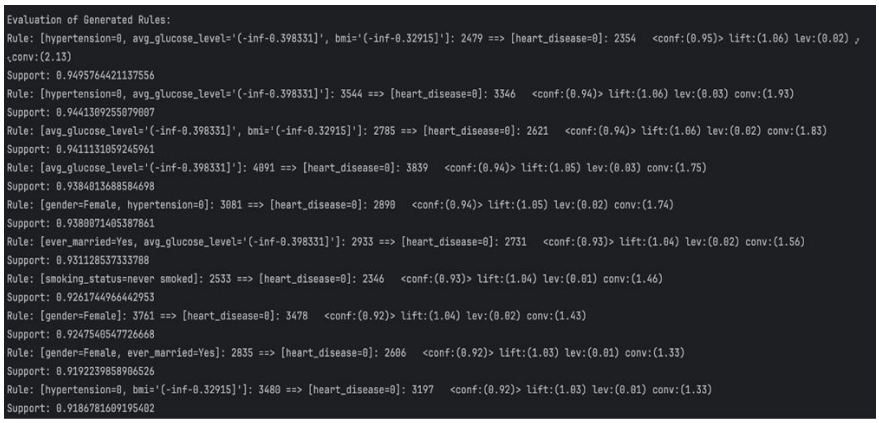
```
Training data using SimpleLogistic
Evaluation results:

Correctly Classified Instances         150               75      %
Incorrectly Classified Instances        50               25      %
Kappa statistic                          0.4748
Mean absolute error                      0.317
Root mean squared error                  0.3988
Relative absolute error                 66.0399 %
Root relative squared error             81.4134 %
Total Number of Instances              200

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.808    0.338    0.782      0.808    0.795      0.475    0.843     0.906     0
                 0.663    0.192    0.697      0.663    0.679      0.475    0.843     0.721     1
Weighted Avg.    0.750    0.279    0.748      0.750    0.749      0.475    0.843     0.832

===Overall Confusion Matrix===

  a  b   <-- classified as
 97 23 |  a = 0
 27 53 |  b = 1

10-fold cross validation evaluation results:

Correctly Classified Instances         613               76.625  %
Incorrectly Classified Instances       187               23.375  %
Kappa statistic                          0.5133
Mean absolute error                      0.317
Root mean squared error                  0.3989
Relative absolute error                 66.0327 %
Root relative squared error             81.4214 %
Total Number of Instances              800

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.804    0.291    0.806      0.804    0.805      0.513    0.839     0.891     0
                 0.709    0.196    0.707      0.709    0.708      0.513    0.839     0.724     1
Weighted Avg.    0.766    0.253    0.766      0.766    0.766      0.513    0.839     0.825

===Overall Confusion Matrix===

   a   b   <-- classified as
 386  94 |   a = 0
  93 227 |   b = 1
```

Evaluation

- The Simple Logistic model demonstrates reasonable accuracy, achieving 75% accuracy on the training data and 76.625% with 10-fold cross-validation. These results indicate the model's capability to correctly classify a majority of instances.
- The Relative Absolute Error (RAE) of 66.0327% with 10-fold cross-validation indicates that, on average, the model's predictions are off by approximately 66.0327% of the mean absolute value of the actual target values.

2.4.6 Apriori





- Rules: Many of the generated rules have high confidence values, indicating strong associations between the antecedent (left-hand side) and consequent (right-hand side) of the rules. For example, rules:
    - "*avg_glucose_level='(-inf,-0.398331]' ==> heart_disease=0 and hypertension=0==> heart_disease=0 and bmi='(-inf-0.32915]' ==> heart_disease=0*" have confidence values around 0.9 or higher.
    - "*ever_married=Yes ==> heart_disease=0 and heart_disease=0 ==> hypertension=0* " have confidence values around 0.88 or higher.
- The support values for the association rules are relatively high, indicating that the antecedent feature combinations occur frequently in the dataset.Example
    - Rule: [hypertension=0, avg_glucose_level='(-inf-0.398331]', bmi='(-inf-0.32915]']
        - Support: 0.9496, Confidence: 0.95, Lift: 1.06
        - Interpretation: Individuals with no hypertension, low average glucose level, and low BMI have a high likelihood (95%) of not having heart disease.

- Rule: [hypertension=0, avg_glucose_level='(-inf-0.398331]']
    - Support: 0.9441, Confidence: 0.94, Lift: 1.06
    - Interpretation: Individuals with no hypertension and low average glucose level have a high likelihood (94%) of not having heart disease.
- Rule: [avg_glucose_level='(-inf-0.398331]', bmi='(-inf-0.32915]']
    - Support: 0.9411, Confidence: 0.94, Lift: 1.06
    - Interpretation: Individuals with low average glucose level and low BMI have a high likelihood (94%) of not having heart disease.

Conclusion

- Association between Health Metrics and Heart Disease: The rules reveal associations between various health metrics such as average glucose level *(avg_glucose_level)*, body mass index *(bmi)*, hypertension status *(hypertension)*, and marital status *(ever_married)* with heart disease *(heart_disease=0)*. These associations can provide valuable insights into the risk factors or indicators of heart disease.
- Gender and Marital Status: Interestingly, rules involving gender (*gender=Female*) and marital status (*ever_married=Yes*) also appear among the top rules. This suggests that there may be gender-specific or marital status-related factors influencing the likelihood of heart disease.

Chapter 3: Conclusion

3.1 Classification

Based on accuracy:

- Random Forest is the top performer, with the highest accuracy both in training and cross-validation.
- Simple Logistic is the next best, providing strong accuracy and good generalization.
- J48 follows, showing decent accuracy, especially in cross-validation.
- Naïve Bayes performs moderately well, with acceptable accuracy levels.
- OneR has the lowest accuracy among the five models, as expected given its simplicity.

⇨ These results suggest that **Random Forest** and **Simple Logistic** are the most reliable models for achieving high accuracy in this classification task.

3.2 Apriori

The Apriori algorithm has effectively identified meaningful associations between various health metrics and demographic factors with heart disease. These rules can aid

healthcare professionals in better understanding the risk factors associated with heart disease and in implementing targeted interventions or preventive measures. Further analysis and validation of these rules with additional data could enhance their utility in clinical practice and public health initiatives.

References

Fedesoriano, stroke-prediction-dataset, Kaggle, **https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/data**

Noureddin Sadawi, WEKA_API, (2015), GitHub repository, https://github.com/nsadawi/WEKA-API