Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

SCHOOL OF COMPUTER SCIENCE AND STATISTICS

INDIFIFYING RELAPSE OF RARE DISEASES USING SYNTHETIC DATA

ANGKIRAT SINGH SANDHU

AUGUST 19, 2022

SUPERVISED BY: DR. JAMES NG

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MSC IN COMPUTER SCIENCE (DATA SCIENCE)

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Signed: Angkirat Singh Sandhu Date: August 19, 2022

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Signed: Angkirat Singh Sandhu Date: August 19, 2022

Abstract

Machine learning and predictive analysis have become an integral part of society. It has sped up the transformation of various fields, especially Medical Science. The main objective of this report is to identify the relapse of antineutrophil cytoplasmic antibody-associated vasculitis (AAV) based on different biomarkers. We are using machine learning algorithms to determine the relapse early to create a personalized treatment plan for the patient. Every case is unique, leading to a non-standard treatment process resulting in some missing biomarkers. These missing biomarkers cause the machine learning models to often fails due to it being a statistical formula. One of the most common methods of resolving the missing values problem is the Multivariate Imputation by Chained Equations, more commonly referred to as the MICE package in R programming language. Another hindrance to using Machine Learning is that Rare diseases such as ANCA-Associated Vasculitis have a meagre data count. Many restrictions are due to federal use rules like HIPAA and GDPR, adding multiple more restrictions to the available data and making the process of the analysis complex. To circumvent the issue of insufficient data, we are using statistical methods to synthesize new data points with similar functionality as the original Data. A constant evaluation using the pairwise correlation comparison and log cluster ensures the integrity of the new synthetic data. Sallow machine learning algorithms like Decision Tree and Random Forest are trained on the newly synthesized data and then tested on the original data along with the added imputed data points. Cross Validation and Random Search help identify the parameters for creating an unbiased model with the best performance.

Acknowledgements

I sincerely thank and acknowledge my supervisor, Dr James Ng (Assistant Professor, Statistics at Trinity College Dublin). I could not have completed the journey of undertaking such a project involving statistical analysis in the healthcare sector. Your push and guidance encouraged me to research and understand the project more deeply.

I also appreciated the support of Dr Arthur White(Assistant Professor, Statistics at Trinity College Dublin). I thank Arthur for his valuable suggestions during the review call that helped me improve the thesis. Having completed a module in Advanced Statistics taught by Arthur helped me brush in the concepts that formed the basis for this thesis.

I also would like to thank my Parents, Mr Amrith Pal Singh Sandhu and Mrs Gurkanwal Sandhu (Gurgaon, Haryana, India) my brother, Mr Arjan Singh Sandhu, for encouraging and supporting me throughout my master's degree. Finally, I would like to thank my role model and elder sister Dr Harleen Arora Gill, who helped me and guided me through the course and helped with the medical aspect of the thesis.

My motivation for doing this thesis was based on my interest in data science and how it impacts the healthcare industry. Ever since I was a child, I have been obsessed with computers and understanding how they work. Having had my first internship in Philips Healthcare as a Data Scientist, I have become interested in helping out patients and saving a life with the help of Data Science.

Contents

List of Figures

List of Tables

1 | Introduction

This report aims to create a comprehensive method to analyse rare disease data and create predictive models to identify the complications from this autoimmune condition risk of infection. This report uses the ANCA data procured from the Rare kidney Deseases (RKD) registry database. Since ANCA specific vasculitis is a rare occurrence, the number of patients for which we have the record is limited, hence developing machine learning algorithms is challenging.

The report helps create an analytical method to overcome these issues by creating synthetic data to increase the ability to train machine learning models for these rare diseases and conditions to identify them early, identify their complications timely and create a personalised treatment plan for the patients.

Various neighbour-based approaches like SMOTE, B-SMOTE and ANS create synthetic data, which can help analyse patterns and perform detailed statistical analysis. The synthetic data generated aims to resemble the original data patterns for artificially providing the data that is unavailable. Another advantage of using synthetic data is maintaining the patient's anonymity and securing their rights under the federal regulations while studying and analysing the patterns of the data. Synthetic data evaluations help to ensure that the data has the same principal properties as the original dataset to ensure the integrity of the data.

1.1 Vasculitis in adults

The vasculitis are distinctly described as inflammation causing blood cells within blood vessels with reactive damage to the mural structures. Blood vessels (including arteries and veins) carry oxygenated and deoxygenated blood to various body parts to drive aerobic cellular respiration. This autoimmune disease process of attacking body's own healthy cells can affect one or multiple body systems including nervous system, gastrointestinal system, genitourinary, eyes, skin [26]. In vasculits, the integrity of vessel wall leads to abnormal bleeding and compromises the inner lumen of the vessel. This process leads to tissue dysfunction and eventually tissue death. In general, the affected vessels may differ in size, types and location which may manifest in different symptomology the condition presents with. The mechanism of why vasculitis manifest in certain individuals is unknown [10].

Vasculitis is a serious disease that requires acute recognition and therapy [21]. The symptomology involves around the affected organs. Hence this can lead to focal or multi-system symptoms and signs. Diagnosing vasculitis is challenging due to a wide variety of focal or multi-system symptoms that it can present with. Medical practitioners make a diagnosis with careful integration of patient's history, examination and investigations. Various biomarkers and clinical presentation must be considered to identify the relapse of vasculitis timely. The suppression of Vasculitic symptoms and achieving remission is essential to avoid organ dysfunction. The treatment of Vasculitis involves suppression of inflammation and/or suppression of autoimmunity. Using ML models would help in the early identification to start timely treatment of the patients with active Vasculitis.

1.2 Report Structure

The thesis project report is structured to have the literature review next, followed by the Methodology and result and a Conclusion and future works to wrap up. The methodology encompasses how the data is standardised and preprocessed post which we use Synthetic data generation algorithms like SMOTE, B-SMOTE and ANS. We use the synthetic data to train multiple machine learning algorithms. The most important thing here is to create a simple, unbiased and high-precision machine learning algorithm trained on the synthetic data.

2 Literature Review

2.1 Technology in Medical Science

There has been a significant surge in technological advancement in the healthcare sector to detect and identify infections or irregularities in the Human body. These advancements are both in hardware and software devices working in tandem to identify the medical risk in a patient. That may not be possible for a human doctor due to the complex relations between various biomarkers, behavioural patterns and other features are taken into account by a machine learning algorithm [15]. Sometimes it is just the speed of detecting an infection that helps save a person's life. An example would be a sensor that helps detect any bacterial infection in a wound, leading to severe implications like loss of limb or death if not treated early to contain the spread 7591093. But by far, the most significant advancement in medical science has been brought by using Artificial Intelligence to predict or identify irregularities in the human body [22] [3].

2.2 Machine Learning for Medical Science Dataset

A major problem in detecting diseases is identifying the issue and its root cause; given the symptoms and biomarkers, the same symptoms could be due to many health conditions. Ensemble models seem to be an excellent fit for such needs as they are excellent at filtering out the noise from the data improving the precision by leaps and bounds [28]. The current method of detecting ANCA vasculitis involves long medi-

cal tests involving a wait time and resources incurring a cost on the patient [5]. Even for Covid 19, the ML models helped identify patients infected with the Virus saving a lot of money and resources spent on acquiring testing kits, which were hard to come across given that Covid 19 was declared a Global pandemic [25]. Simple Ensemble based models are a standard method of prediction of ANCA-associated glomerulonephritis [2]. One major issue with ensemble models or any other machine learning algorithm is that they heavily depend on data for complete unbiased training [20].

2.3 Data Synthesis for Medical Science Dataset

When dealing with rare conditions such as ANCA Vasculitis or other rare diseases, there often is the issue of insufficient data, which could cause problems while training machine learning models. This issue also arises when the test is more expensive or confidential, like an X-Ray, Ultrasound or other medical tests. Generative Adversarial Networks (GAN) are a common technique used to create synthetic data to train a model more efficiently [1], [23], [11]. Another commonly applied method is SMOTE algorithm [24] primarily used to settle the imbalance in the data by creating new data points. Synthetic data is instrumental in the scenario where personal data is involved. Since the information is generated using computer algorithms, it does not belong to any specific patient, allowing more accessible access to perform a more comprehensive analysis with greater freedom [6].

3 | Dataset Description

Since the thesis work is an extension of a previous report, the dataset used was the cleaned dataset which was already engineered and processed. The dataset has the following properties in it:

1. The data contains data from 69 patients.; The data records data for 39 features
2. Of the 39 columns, we use 12 features (including the target feature)
3. The target column is converted into a binary column with Long-Term Remission Off-Therapy (LTROT) as 0 (-ve class) and Relapse as 1 (+ve class)
4. 9 columns of the 11 feature columns are numeric, and the remaining 2 are binary columns.
5. 47 records of the original 69 have one or more missing values in them.

```
> summary(data)
 Weight..KG.          CRP          Neutrophil.count.x10.9.L Lymphocyte.count.x10.9.L Neutrophil...Lymphocyte.ratio Eosinophil.Count.x10.9.L Platelet.count.x10.9.L    IgG.g.dL
Min.   :-1.7935   Min.   :-0.3491   Min.   :-1.7249          Min.   :-1.72094         Min.   :-0.75533              Min.   :-1.0129          Min.   :-2.22555       Min.   :-2.08282
1st Qu.:-0.7666   1st Qu.:-0.3071   1st Qu.:-0.6440          1st Qu.:-0.88806         1st Qu.:-0.55342              1st Qu.:-0.5040          1st Qu.:-0.60779       1st Qu.:-0.45011
Median :-0.1196   Median :-0.2616   Median :-0.1603          Median : 0.05761         Median :-0.28168              Median :-0.2496          Median :-0.05334       Median :-0.06579
Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000          Mean   : 0.00000         Mean   : 0.00000              Mean   : 0.0000          Mean   : 0.00000       Mean   : 0.00000
3rd Qu.: 0.5412   3rd Qu.:-0.1576   3rd Qu.: 0.4191          3rd Qu.: 0.58684         3rd Qu.: 0.06236              3rd Qu.: 0.1066          3rd Qu.: 0.60364       3rd Qu.: 0.44450
Max.   : 2.7766   Max.   : 5.2218   Max.   : 3.8650          Max.   : 2.58228         Max.   : 4.64637              Max.   : 4.8388          Max.   : 2.48343       Max.   : 3.22899
NA's   :15        NA's   :17        NA's   :5                NA's   :6                NA's   :6                     NA's   :16               NA's   :11             NA's   :27
 Anti.PR3.MPO.Level  Uninalysis.Protein Uninalysis.Blood Stratification
Min.   :-0.533348   Min.   :0.00       Min.   :0.0      Min.   :0.0000
1st Qu.:-0.533348   1st Qu.:0.00       1st Qu.:0.0      1st Qu.:0.0000
Median :-0.473917   Median :0.00       Median :0.0      Median :1.0000
Mean   : 0.000000   Mean   :0.75       Mean   :0.5      Mean   :0.5797
3rd Qu.:-0.005074   3rd Qu.:1.00       3rd Qu.:1.0      3rd Qu.:1.0000
Max.   : 3.362671   Max.   :3.00       Max.   :3.0      Max.   :1.0000
NA's   :10          NA's   :21         NA's   :21
```

Figure 3.1: Data Summary

The Figure 3.1 gives an overall summary of the complete data imported into the R programming environment. It provides an overview of the entire dataset with min, max, the mean and standard deviation for the column and the number of missing values in each column (all features have missing values in them).

4 | Methodology

The report aimed to create an unbiased machine learning algorithm which could identify high-risk patients basis the different biomarkers. Two programming languages, R and Python, are used to code methodology. R programming language was used to perform data filtration, cleaning and preprocessing, while python was used for ML model creation, selection and testing.

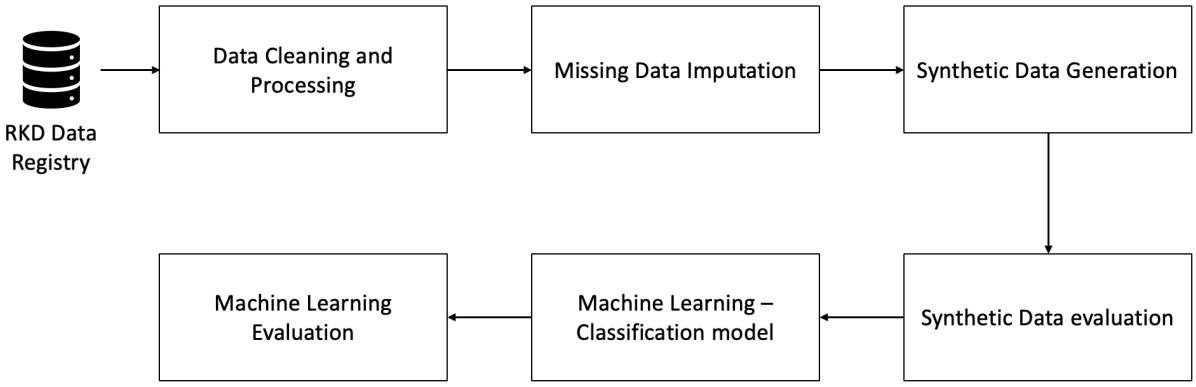


Figure 4.1: Data Pipeline for predicting ANCA Data Prediction

The Figure 4.1 gives a broad overview of the data pipeline and the various actions taken to process or pre-process the data to get an unbiased model with a high precision rate.

4.1 Data Filteration and Preprocessing

Before applying any ML operation, we need to process the features to create an unbiased model with reliable predictions:

1. Data Normalization or/and Standardisation: It is the process of scaling data.

While normalisation is the process of scaling using minimum and maximum values, standardisation is the process of scaling using mean, and standard deviation.

2. Imputation of missing values: Sometimes, the data for patients is not complete. In such cases, imputations are performed packages like Multivariate Imputation by Chained Equation (MICE) can help overcome the issue.

3. Transformation of records: Once we have all the data, we may need to transform the columns by expanding or reducing the feature set. Principal Component Analysis (PCA) is a widespread method to reduce the number of numeric columns in the dataset.

The original dataset consists of 69 patients (rows) and 39 biomarkers (columns). Of these, we move forward with the analysis with only 12 biomarkers which are the feature columns for the study. The columns consist of numeric values, all with varying units making some column values much more significant than others. Large data values with unbiased scales could lead to biased ML learning models. The standard scaling method ensures that all the values in the dataset columns are between 1 and 0, bringing the data to an equitable scale.

4.2 Missing Value Imputation

The Figure 4.2 shows that all 11 feature columns (9 numeric columns and 2 binary columns) have missing values. The missing values can be caused due to various factors like unrecorded data or an error in recorded data. Most machine learning models fail to encounter missing values in the dataset since these models are based on mathematical formulas. Some of the algorithms like KNN and Naive Bayes do work with missing values leading to creating a model with incorrect results, which could be biased hence getting a model with low precision.

Missing data could essentially classify data into 4 categories:

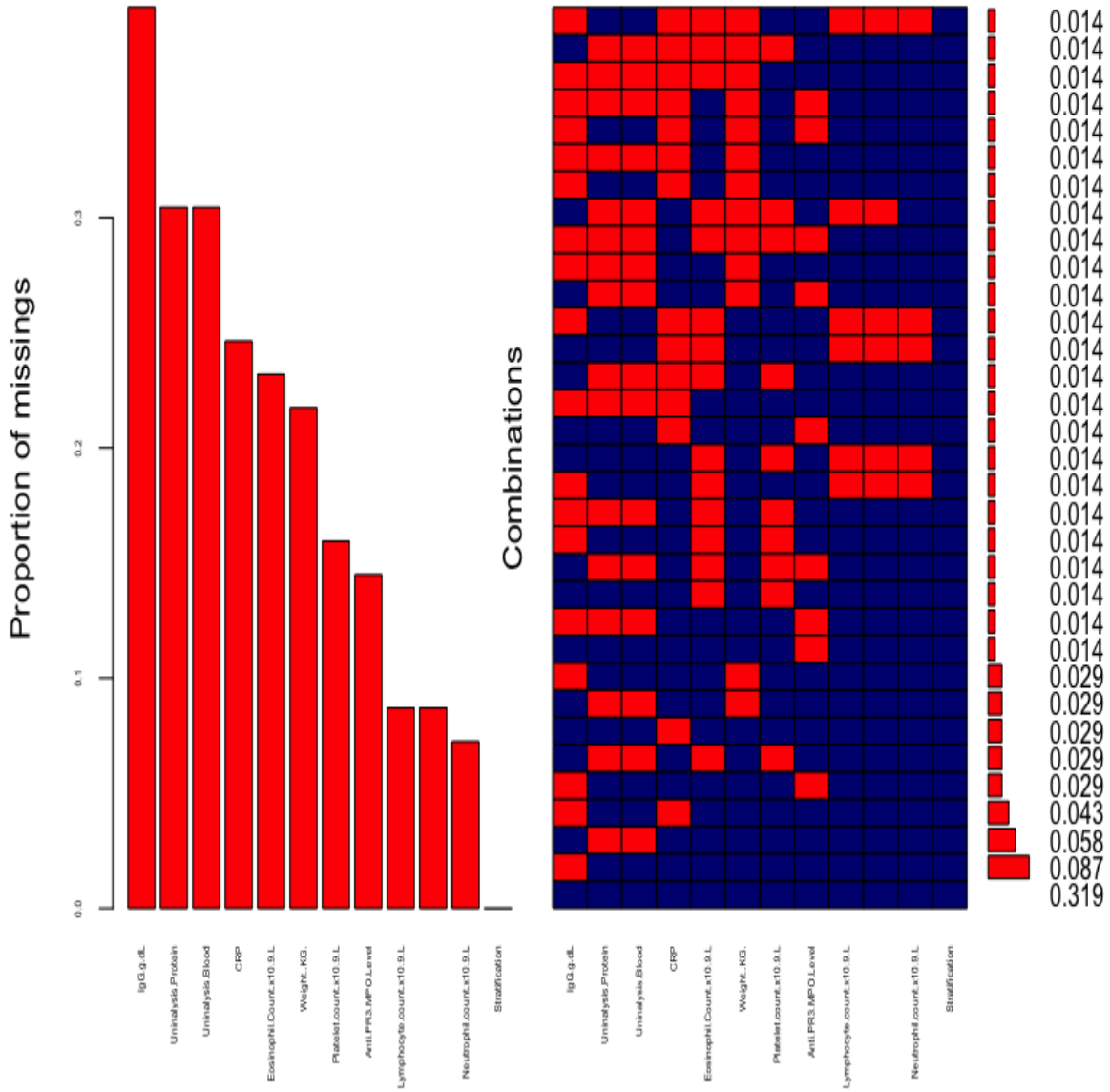


Figure 4.2: Missing Value Summary

- Missingness Completely At Random (MCAR): here, the probability of missingness is the same for all units. Thus, it implies no relationship between the data present and the missing data, making it extremely hard to implement.
- Missingness at Random (MAR): the probability of missingness in these cases is the same within groups of the observed data.
- Missingness that depends on unobserved Predictors: Here, the data not missingness is not random but is dependent on variables that are not available in the dataset.

- Missingness that depends on the missing value: Here, the probability of the missing data depends on the variable.

The missing values are imputed using the Multivariate Imputation by Chained Equations, commonly referred to as the MICE package in the R programming language. The function takes four significant parametric inputs:

1. Data: The input data with the missing values columns in it.
2. Method: The method used to identify the value for the actual value imputed in the missing places.
3. m: Iteration count for multiple imputations.
4. maxit: the number of internal imputations for each iteration.

For the ANCA data, which has missing values for both the numeric and non-numeric column (binary column), the Predictive Mean Matching (PMM) method is used to create 10 imputation datasets with 50 iterations in each imputation.

4.3 Synthetic Data Generation

We often encounter a shortage of data, leading to an under-fitted machine learning model with low precision. Machine learning models rely on a training mechanism which learns and creates a mathematical formula based on observed patterns in the data. To overcome this issue, we need to generate new synthetic data points. The smotefamily package of R programming is one of the most commonly used package to create new data points. The package heavily depends on the Neighbour based approach like K-Nearest Neighbour (KNN).

3 commonly used algorithms for generating synthetic data:

1. **Synthetic Minority Oversampling Technique (SMOTE)**: This is one of the most common approaches to creating synthetic data. The algorithm makes new data points between a randomly selected minority class and its neighbours. The prin-

cipal parametric input to the algorithm is the value of K specifying the number of neighbours to be chosen and the duplicate size determining the duplication factor by which the minority class needs to be increased. The Figure 4.3 illustrates how SMOTE algorithm works whic his explained in Algorithm 1

Algorithm 1 Synthetic Minority Oversampling Technique (SMOTE) Algorithm

Require: $k > 1$
Require: $duplicationSize > 1$
number of new points =size of minority class
$counter = 0$
while $counter \leq$ number of new points **do**
$counter = counter + 1$
Select a point p at random
Select a neighbour point np from one of the k nearest neighbour points
create new random point between $n\&np$
end while

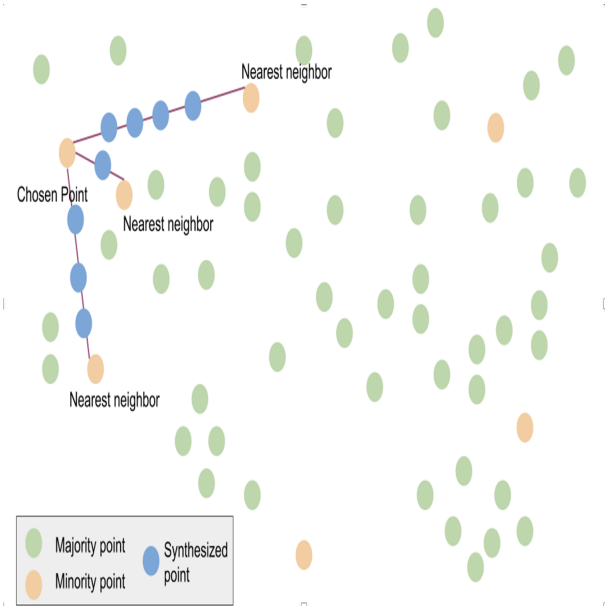


Figure 4.3: Synthetic Minority Oversampling Technique (SMOTE) [4]

2. Borderline SMOTE (B-SMOTE): This algorithm is an extension of the previously discussed SMOTE algorithm. The data points selection, unlike SMOTE, is selected only on the border region of the minority class, unlike the stochastic method of choice performed in SMOTE. The input parameters for the B-SMOTE algorithm are same as that in SMOTE. Figure 4.4 illustrates how the B-SMOTE algorithm works the same way as the SMOTE algorithms, only adding the new

points on the edge of the minority class. The data density using the B-SMOTE algorithm generally increases around the edge of the minority class.

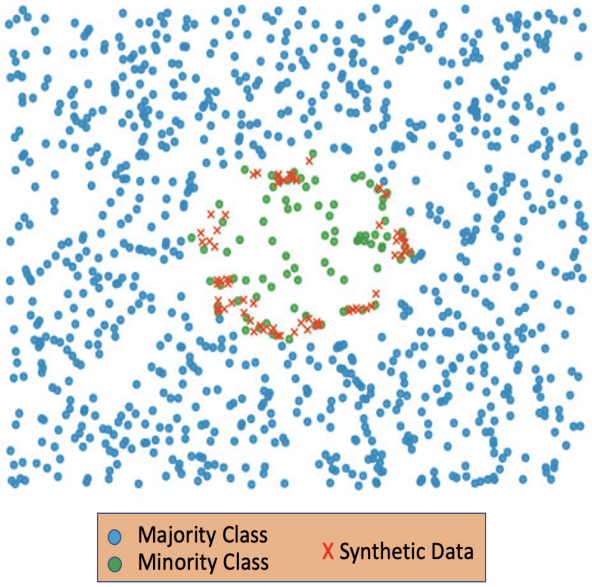


Figure 4.4: Borderline SMOTE (B-SMOTE) [13]

3. Adaptive Neighbor Synthetic (ANS): This algorithm is a density-based approach. The algorithm self-adjusts the k neighbour parameters. The new data points are placed where the data density is low, making the data selection process less stochastic than the SMOTE algorithm, which puts the points at a position between two randomly selected data points. Since the ANS algorithm self adjust the K value in the algorithm the only input parameter is the duplication size. Figure 4.5 shows that the density of the minority class is a criterion for deciding where to position the newly created synthetic data. The position of the new point e in the Figure is such that it can fill the void of missing minority data between the x1 and x2 points. The algorithms hence ensure that the density of the data is more uniform with no gaps between the data.

4.4 Synthetic Data Evaluation

Following are the evaluation metrics used to evaluate the data integrity:

- **Pairwise Correlation Difference (PCD)** [7]: it compares the correlation matrix

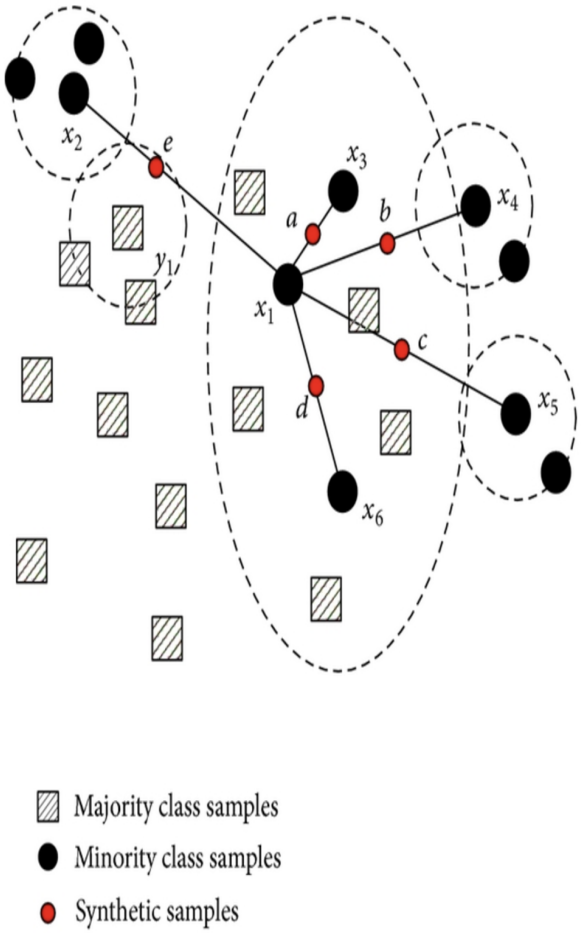


Figure 4.5: Adaptive Neighbor Synthetic (ANS) [8]

of the original value and the matrix formed using the new synthetic data. This metric intends to see how well the inter-column dependencies are maintained in the newly synthesised data. The representation is a matrix created using the Equation 1 where X_R is the real data, and X_S is the synthetic data.

$$PCD(X_R, X_S) = \| \ Corr(X_R) - Corr(X_S) \ \| \tag{1}$$

- **Log Cluster Evaluation** U_c [27]: This metric evaluates the underlying latent structure of the data by using Cluster analysis. Clusters are created on the complete dataset (real + synthetic data). The Elbow method helps determine the optimal number of Clusters symbolised as G for the given dataset. The Equation 2 helps to get the value of U_c, n_j represents the total number of points in the jth cluster. The lower the value of U_c, the better the data spread. If the value is too high, it means the synthetic data is overshadowing the original data, and a NA value means that there are clusters in the data that do not have either real or synthetic data.

$$\begin{aligned} U_c &= \log(\frac{1}{G}\sum_{j=1}^{G}[\frac{n_j^R}{n_j} - c]^2) \\ c &= \frac{n^R}{(n^R + n^S)} \end{aligned} \tag{2}$$

4.5 Predictive Analysis

Predictive analysis is creating a statistical model that learns from patterns observed in the data. There are 2 principal categories to classify all the ML algorithms:

- **Supervised Learning**: Here, there is an available target column, and the training objective is to understand the patterns in the data to predict the target variable. Supervised learning algorithms involve Classification and Regression algorithms.
- **Unsupervised learning**: unlike supervised learning, here we do not have an explicit target variable, and the learning process involves finding patterns in the data. The unsupervised Learning method consists of clustering algorithms.

Since we need to classify the patient records as LROT or Relapse, a known target, Thus making supervised classification algorithms the main focus.

For the Current dataset, we have mainly implemented two algorithms:

1. **Decision Tree** [16]: It is a decision-making tool based on a binary tree-like model of decision-making and the resultant consequences. The algorithm uses the Gini index entropy method expressed in the equation 3 to identify the data split. The j value in the equation is the number of classes in the ith node, while P represents the ratio of classes of the ith node. The decision tree method is advantageous as it is a non-parametric learning algorithm which creates a model which is easy to visualize and understand. The disadvantage of using a decision tree is that it is susceptible to noise in the data.

$$Gini = 1 - (\sum_{i=-1}^{j} (p(i|t))^2) \tag{3}$$

2. **Random Forest** [19] It is a decision tree-based ensemble learning algorithm based

on out-of-bag error. An ensemble learning model uses multiple classifiers instead of a single classifier algorithm to have an improved prediction result. The out-of-bag error involves having subsets of data created randomly with replacement to train the individual classifier. This way, the classifiers are exposed to different aspects of the data making the overall model less biased. The baseline classifier in the random forest model is the decision tree algorithm. The model's performance could be adjusted by changing the number of trees in the model, which is defined by the n-estimator parameter. Another significant difference between random forest and decision tree is the bagging and boosting algorithms that bring a certain amount of stochasticity to the model. The stochasticity Ensuring that two random forest models trained on the same dataset could lead to different prediction results on the same test data.

4.5.1 Model Selection

One of the essential parts of Machine Learning is finding the most optimal Hyperparameter values for the algorithm for which the model will make the most optimal predictions. For this purpose, we have used RandomSearch **??** and K-Fold Cross-validation [18].

RandomSearch involves a user identifying the adjustable hyperparameters and inputting a range of values for each one of those hyperparameters. The RandomSearch algorithm then creates an ML model using a randomly created hyperparameter set (values are randomly selected from the inputted value set). The model is then stored before a new hyperparameter set is selected, and the process repeats n times (in our case, 10 but is an adjustable parameter of RandomSearch). The best model is selected by comparing all the models' mean cross-validation scores. Algorithm 2 explains how cross-validation works. This model selection method ensures that the model is not biased, achieved by cross-validation due to the multiple pieces of training on each subset. It also allows for the full exploration of the hyperparameter sample space. RandomSearch helps explore the sample space with a limited number of models sav-

ing time and computational space.

Algorithm 2 K-fold Cross-Validation

Require: $k > 1$
Require: $Data = Input\_Dataset$
Require: $ML\_Algorithm$
  $dataset_list = krandomsplitsofData$
  $accuracy\_list = []; model\_list = []$
  $counter = 1$
  while $counter \leq$ k **do**
    $model = Train(ML\_Algorithm, (Data - dataset_list[k]))$
    $accuracy = Test(ML\_Algorithm, dataset_list[k])$
    $accuracy\_list.append(accuracy)$
    $model\_list.append(model)$
    $counter = counter + 1$
  end while
  return $accuracy\_list$

4.6 Prediction Evaluation

Following are the classification evaluation metrics used to compare the model performance:

1. **Confusion Matrix** [14]: It is a square matrix summary of the prediction result. The number of rows and columns equals the number of target classes. In the case of the binary target, the matrix is as shown in Figure 4.6: The confusion matrix

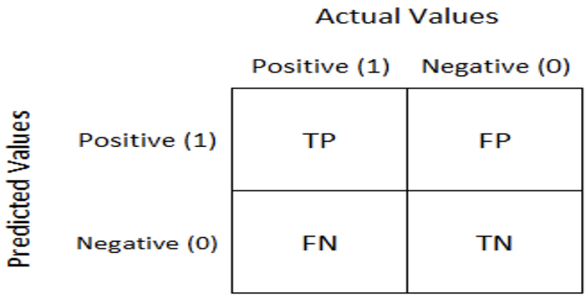


Figure 4.6: Confusion Matrix [17]

   metrics are as follows:

   - True Positive (TP): The correctly predicted +ve class.
   - True Negative (TN): The correctly predicted -ve class.

- False Positive (FP): The wrongly predicted +ve class.
- False Negative (FN): The wrongly predicted -ve class.

2. **Precision** [9]: It helps measure how close the measurement of the same items are to each other. $\frac{TP}{(TP+FP)}$
3. **Recall** [9]: It helps measure how close the measurement of different items are to each other. $\frac{TP}{(TP+FN)}$
4. **F1 Score** [9]: It is the harmonic mean of the precision and recall. The metric helps compare the individual values of precision and recall and help find the model with best value for both. The maximum F1 score possible is 1, this is when both the precision and recall is 1 indicating a perfect model. The minimum score for F1 is 0 this happens when either the value of precision or recall is 0 indicating a biased model. $F1 = 2\frac{precision.recall}{(precision+recall)} = \frac{tp}{tp+0.5(fp+fn)}$

5 Result

5.1 Synthetic Data Evaluation

The synthetic data evaluation methods help evaluate the quality of the newly created data and ensure that it has similar properties as the original dataset. We are using 2 ways to ensure the quality of synthetic data evaluation the Pairwise Correlation Difference test and Log Cluster.

5.1.1 SMOTE

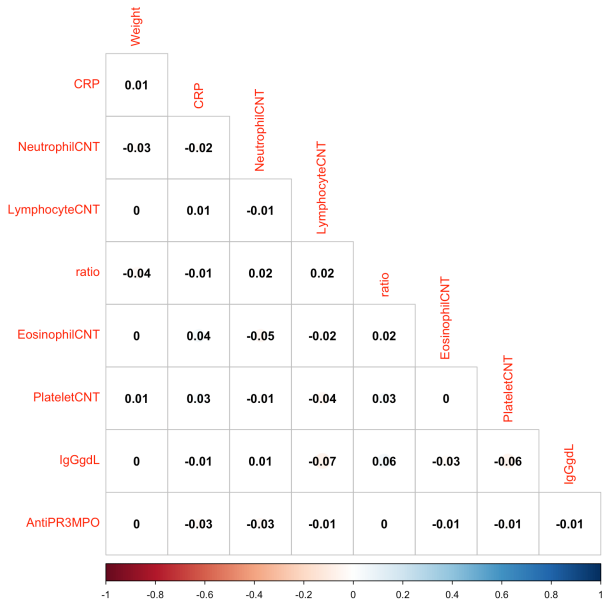


Figure 5.1: Data Pipeline for predicting ANCA Data Prediction

The Figure 5.1 shows the PCD matrix for the original data and the newly created Synthetic data created using the SMOTE algorithm. We see minimal values in the matrix,

denoting no significant shift in the correlation matrix. The overall mean absolute difference for the matrix is 0.019.

The Log cluster value for the SMOTE data is -8.98 (10 clusters were selected A.1), which is a good metric. The algorithm has generated 3740 new rows from the original 455 uniquely imputed rows, including 22 not null original data points and 433 imputed values obtained by performing 10 imputations on the 47 rows with missing data.

5.1.2 BSMOTE

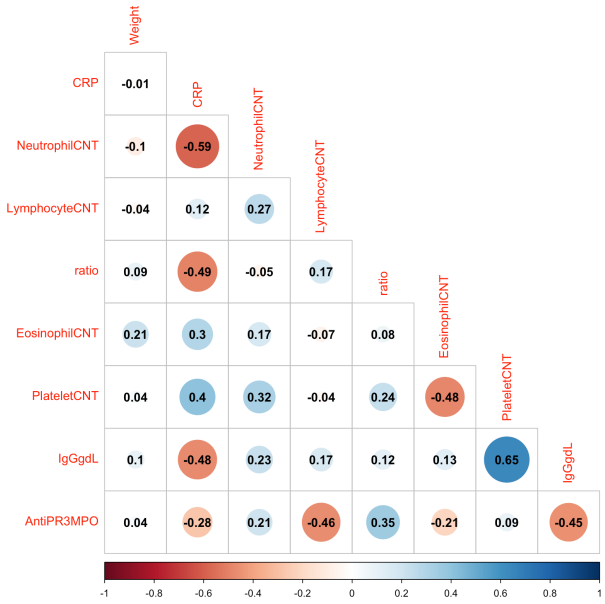


Figure 5.2: Data Pipeline for predicting ANCA Data Prediction

The Figure 5.2 shows the PCD matrix for the original and B-SMOTE synthetic data. The values in the matrix are comparatively much higher than what we see in the case of SMOTE algorithm, with an absolute average coming to an approx of 0.203, which is significantly higher (approximately 10 times more) than the absolute mean value observed in the case of SMOTE algorithm.

Since the B-SMOTE algorithm only creates the synthetic data points close to the borders leaving a few clusters void of synthetic data, leaving some Clusters with no synthetic data in them. The Log cluster value, in this case, is -3.47 (10 clusters were

selected A.2), which is lower than the SMOTE algorithms indicating better cluster integrity than that of the SMOTE algorithm. The algorithm generates around 194 unique synthetic data points from the original 455 data points.

5.1.3 ANS

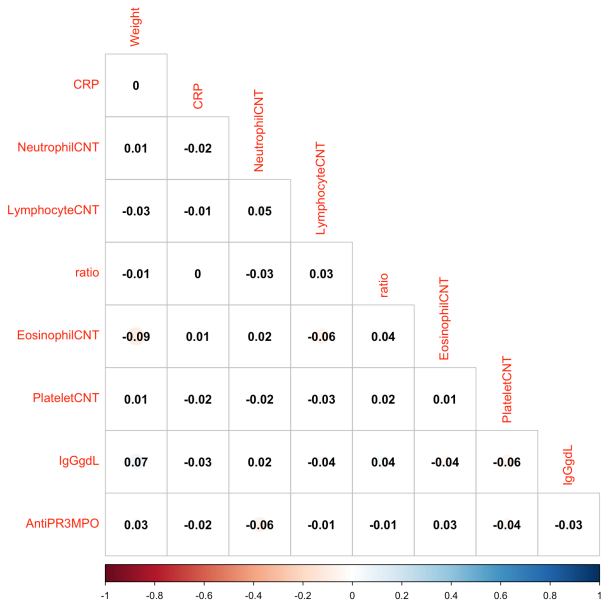


Figure 5.3: Data Pipeline for predicting ANCA Data Prediction

The Figure shows the PCD matrix for the original and ANS synthetic data. The values in the matrix are comparable to the PCD matrix of the SMOTE algorithm with a few inflexion points. The absolute mean value of the matrix is 0.026, which is only a little higher than the SMOTE absolute mean value.

The log cluster value for the ANS algorithm is -7.51 (10 clusters were selected A.3), which is lower than that observed for the SMOTE algorithm. Indicate that the cluster integrity for The ANS is more than that of SMOTE Algorithm; the difference is negligible. The total number of unique synthetic data points obtained using the ANS algorithm is 3635 over the original 455 data points.

5.2 Machine Learning

RMF and DT models are used to predict the Relapse of the Vasculitis in the patient. The evaluation of the models is thus a significant part of the workflow as it helps us identify the best prediction method for the data set while ensuring no bias. Selection of the best RMF and DT modes is done using the Random Search algorithm. (Results of the Random search are in Appendix A.1, A.2, A.3, A.4, A.5, A.6)

5.2.1 SMOTE

Column Name	Decision Tree	Random Forest
AntiPR3MPOLevel	0.232	0.210
NeutrophilLymphocyteratio	0.079	0.128
Neutrophilcountx109L	0.197	0.124
EosinophilCountx109L	0.036	0.120
UninalysisProtein	0.097	0.079
Lymphocytecountx109L	0.170	0.079
CRP	0.059	0.063
UninalysisBlood	0.060	0.056
IgGgdL	0.016	0.055
Plateletcountx109L	0.045	0.047
WeightKG	0.008	0.038

Table 5.1: SMOTE Feature Importance; Sorted Descending on Random Forest Feature importance

The Figure 5.4 is a 5-length imputed decision tree (the entire DT of max depth 15 is available in the Appendix A.4). Parallelly the Table 5.1 outlines the feature importance based on the Gini index for features analysed in the Decision tree and the average Gini index obtained by the multiple trees constituting the RandomForest model. The majority of the columns in the table show that the order of the features is not affected when ordered on the Random Forest values or the Decision tree values. The only exception is the Eosinophil Count, which goes up to the 4th position in the Random forest case while at the 9th position when ordered for the Gini index.

The Figure 5.5 shows the Correlation matrix and the DT and RMF model classification report. The correlation matrix shows that the number of incorrect predictions is much

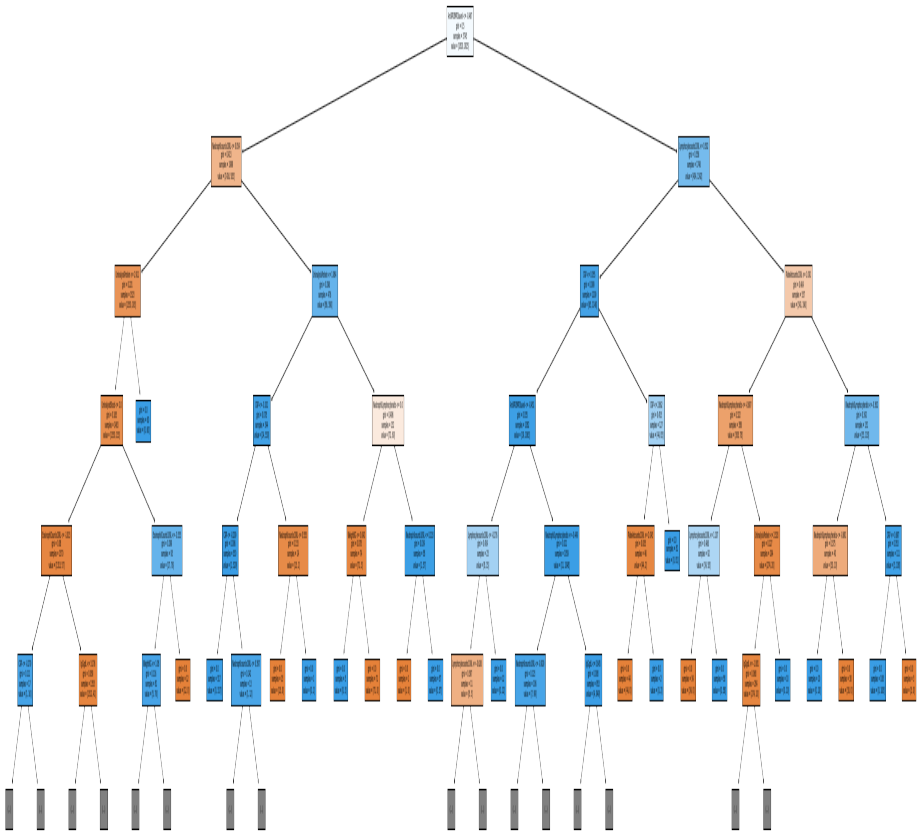


Figure 5.4: SMOTE Decision tree; imputed for depth 5

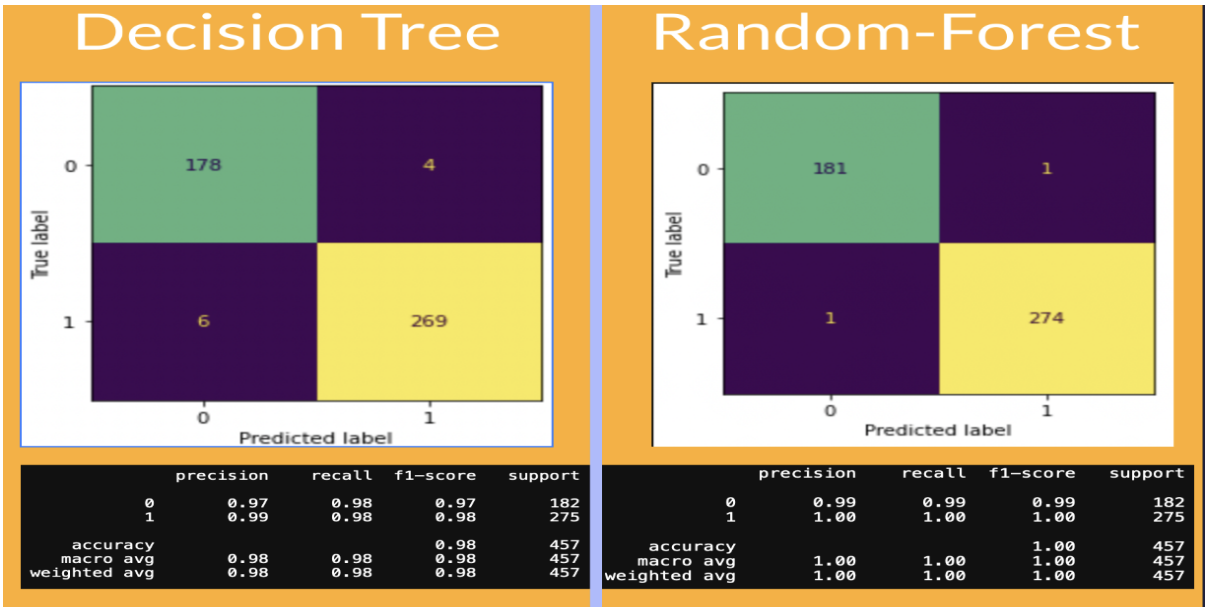


Figure 5.5: SMOTE Confution Matrix and Classification Report

less for the RMF model (2) compared to the DT (10). It leads to a better F1 score and accuracy (0.98 for DT and 1 for RMF) for the RMF model for both the positive and Negative predictions. Indicating means the model is overall unbiased and has excellent prediction capability.

5.2.2 BSMOTE

Column Name	Decision Tree	Random Forest
CRP	0.306	0.214
Neutrophilcountx109L	0.046	0.142
AntiPR3MPOLevel	0.045	0.107
UninalysisProtein	0.273	0.091
IgGgdL	0.133	0.085
NeutrophilLymphocyteratio	0.067	0.085
WeightKG	0.036	0.073
Plateletcountx109L	0.094	0.063
Lymphocytecountx109L	0.000	0.060
EosinophilCountx109L	0.000	0.046
UninalysisBlood	0.000	0.034

Table 5.2: B-SMOTE Feature Importance; Sorted Descending on Random Forest Feature importance

The Figure 5.6 is a 5-length imputed decision tree (the entire DT of max depth 7 is available in the Appendix A.5). Parallelly the Table 5.2 outlines the feature importance

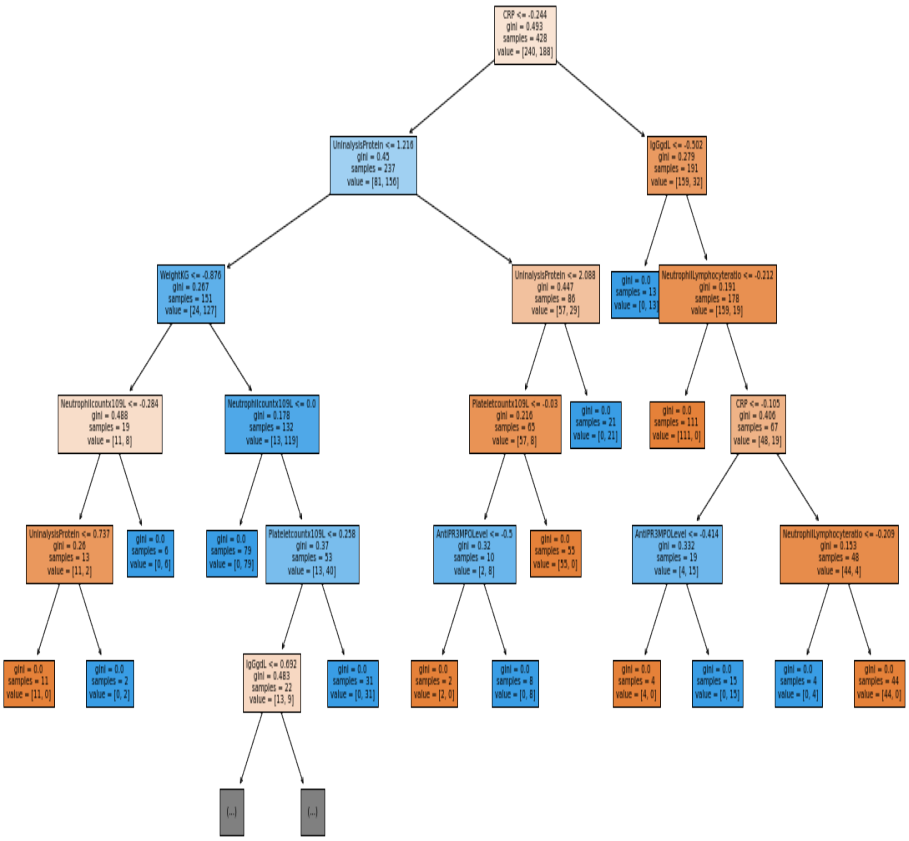
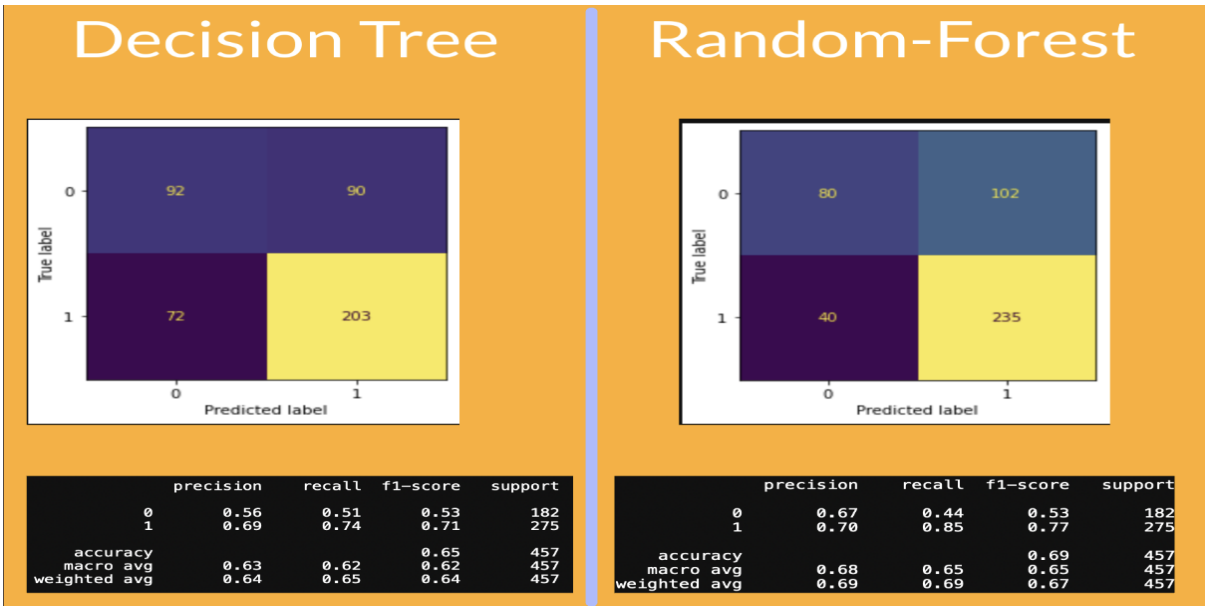


Figure 5.6: B-SMOTE Decision tree; imputed for depth

based on the Gini index for features analysed in the Decision tree and the average Gini index obtained by the multiple trees constituting the RandomForest model. Similar to the SMOTE algorithm, we see that the order of feature importance does not change much for both the DT and RMF models. The distinguishing feature, though, is the 0 value we see for the last three most insignificant features of the DT model, essentially ignoring them completely.



	precision	recall	f1-score	support
0	0.56	0.51	0.53	182
1	0.69	0.74	0.71	275
accuracy			0.65	457
macro avg	0.63	0.62	0.62	457
weighted avg	0.64	0.65	0.64	457

	precision	recall	f1-score	support
0	0.67	0.44	0.53	182
1	0.70	0.85	0.77	275
accuracy			0.69	457
macro avg	0.68	0.65	0.65	457
weighted avg	0.69	0.69	0.67	457

Figure 5.7: B-SMOTE Confution Matrix and Classification Report

The Figure 5.7 shows the confusion matrix and the classification report for the DT and RMF model trained on the B-SMOTE synthetic data. The confusion matrix shows that even though the RMF model has 20 more points correctly predicted, the model has a bias for -ve prediction. The classification report shows a significantly higher precision and recall rate for the LROT prediction than the Relapse value, further reinforcing the bias highlighted by the confusion matrix.

5.2.3 ANS

The Figure 5.8 is a 5-length imputed decision tree (the entire DT of max depth 7 is available in the Appendix A.6). Parallelly the Table 5.3 outlines the feature importance based on the Gini index for features analysed in the Decision tree and the average Gini index obtained by the multiple trees constituting the RandomForest model. Unlike the

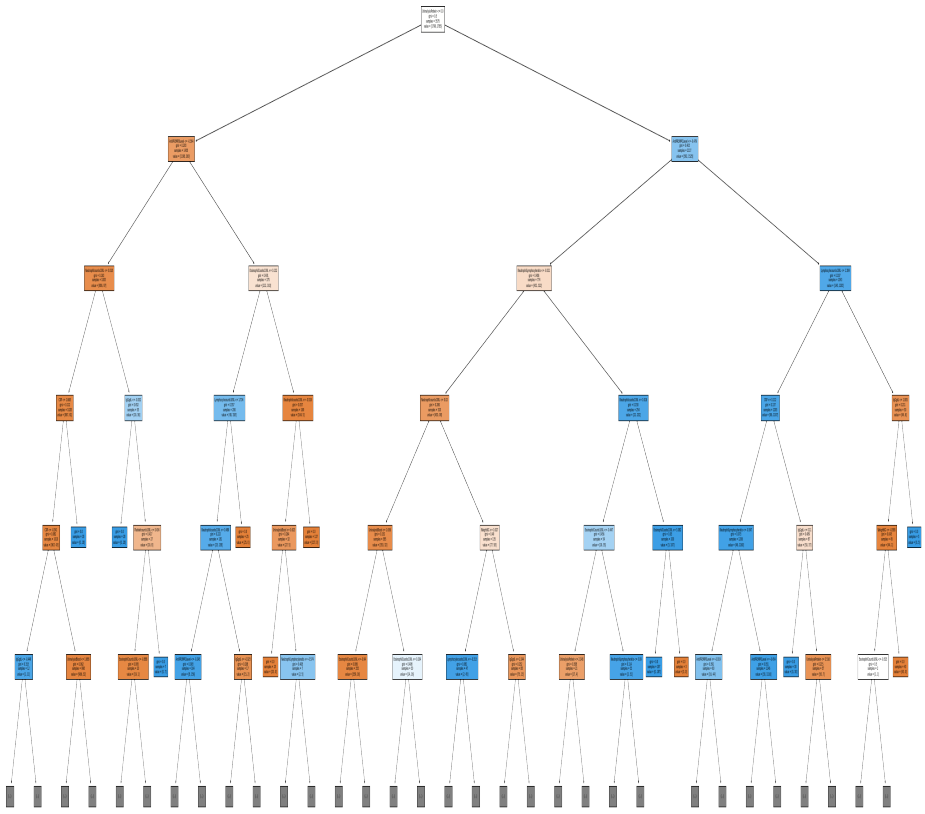


Figure 5.8: ANS Decision tree; imputed for depth

Column Name	Decision Tree	Random Forest
CRP	0.063	0.214
Neutrophilcountx109L	0.060	0.142
AntiPR3MPOLevel	0.174	0.107
UninalysisProtein	0.293	0.091
IgGgdL	0.049	0.085
NeutrophilLymphocyteratio	0.118	0.085
WeightKG	0.031	0.073
Plateletcountx109L	0.015	0.063
Lymphocytecountx109L	0.066	0.060
EosinophilCountx109L	0.102	0.046
UninalysisBlood	0.029	0.034

Table 5.3: ANS Feature Importance; Sorted Descending on Random Forest Feature importance

SMOTE and B-SMOTE algorithms, we do see a significant change in the order of the feature importance for the RMF and DT models. The fact that the ANS model makes the data spread more expansive than the SMOTE model causes a shift in the subsets of the data, which affects the overall Gini index average leading to the change in the order of the feature significance.

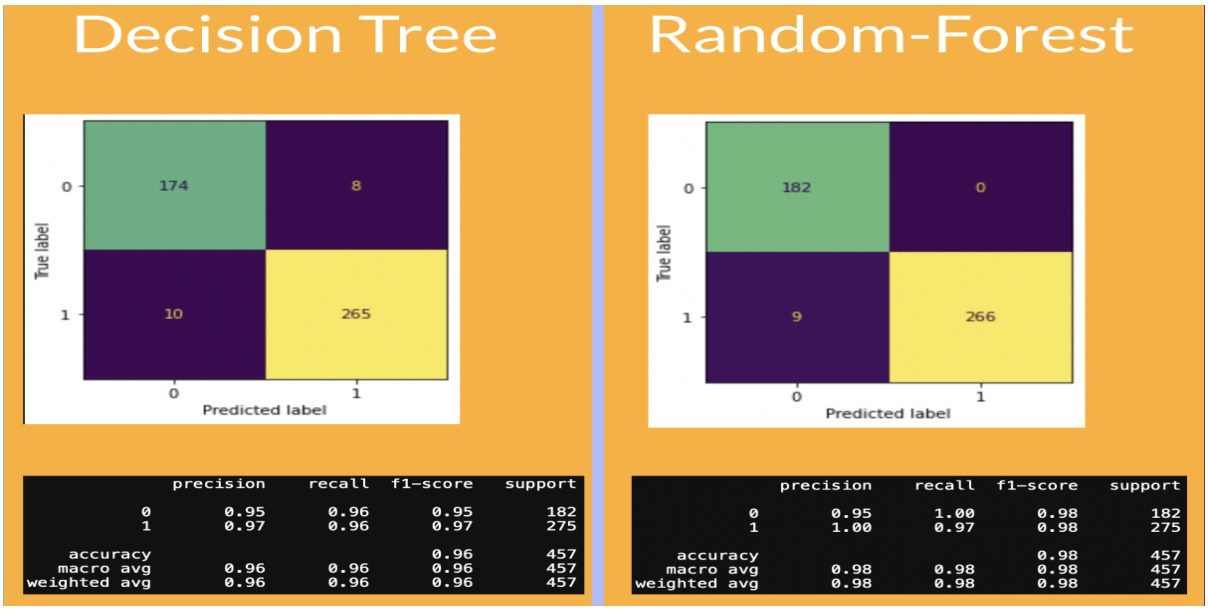


Figure 5.9: ANS Confution Matrix and Classification Report

The Figure 5.9 shows the confusion matrix and the classification report for the DT and RMF model trained on the ANS synthetic data, which helps infer that the RMF model, in this case, performs better than the DT model. Unlike the BSMOTE, both models

have a comparable Precision and recall rate, concluding that the models are unbiased. The overall performance of both models is much better than the models trained on the B-SMOTE synthetic data but still falls short when compared to the SMOTE models by a very slim margin.

5.3 Result Summary

Column Name	**Decision Tree**	**Random Forest**	
Parameter	**SMOTE**	**B-SMOTE**	**ANS**
Synthetic Data Row Count	3740	194	3635
PCD (absolute mean)	0.019	0.203	0.026
Log Cluster	-8.98	-3.47	-7.51
Best ML Algorithm	Random Forest	Random Forest	Random Forest
Best ML Accuracy	100%	69%	98%

Table 5.4: Summary of all the important metrics

The table 5.4 summarises the above result. From the table, we can infer that given the low value of the PCD absolute mean and low log cluster value (taking the row count into consideration) coupled with great prediction result, we can conclude the SMOTE algorithm is the best performing model.

6 | Discussion

There are various rare diseases or disorders which, if left undiagnosed or unattended, could be fatal. One such rarely occurring autoimmune condition is the ANCA vasculitis which causes inflammation of blood vessels stiffening the flow of blood to various parts of the body. Leaving these conditions as are could result in tissue damage affecting one or many body organs such as the eyes, skin, lungs, gut, kidneys or nervous system. Even after performing immunosuppressive therapies, the risk of the patient's mortality reduces by 2.3%, and a parallel risk of relapse in approximately 50% of the cases persists [12]. Another issue is diagnosing rare conditions since they share symptoms with various other diseases and, in some cases, may involve expensive tests or diagnostic procedures like CT-scan or MRI scan. Thus having a reliable predictive model that looks at the various biomarkers in the data to detect a relapse would be beneficial. This thesis aims to generate synthetic data around this relapse data so that we can more optimally analyse and create predictive ML models with low bias and high precision.

One of the preliminary asks of any predictive algorithm is reliable data in sufficient quantity to train the model so that it is not under-fitted. When we consider rare diseases such as ANCA vasculitis, we often see that the most significant problem is the shortage of data. The available data has a lot of missing values in them and are highly restrictive. Thus, synthetic data generation is a convenient solution to the problem since it helps overcome the problem of insufficient data and allows more exhaustive analysis without risking confidential data. This task can be achieved using the R

programming language library "smotefamily", which bundles multiple synthetic data generation algorithms like SMOTE, B-SMOTE and ANS. The main application of such algorithms is to balance out imbalanced datasets enabling better model training with low bias.

Synthetic data is only valid when it has similar properties to the original data. In this thesis report, we evaluated the synthetic data using 2 metrics, PCD and Log cluster. PCD ensures that the newly generated synthetic data has the same inter-column dependency properties as the original data. At the same time, the Log cluster technique help evaluates the spread, providing the maintenance of the underlying structure of the data. Since the data count generated by all the 3 algorithms is different, it is essential to consider that while evaluating these methods. From the Results section, it is clear that the SMOTE algorithm is an ideal choice, given it generates 3740 new and unique data points while keeping a very low absolute PCD mean of 0.019, ensuring the properties of the original data do not change. The high log cluster value could be attributed to the newly generated points being almost 8 times more than that initially imputed dataset.

Simple supervised classification Machine learning algorithms could suffice the task of predicting the relapse with high precision when trained on the newly created synthetic dataset. Overall the ensemble nature of the Random forest gave it an edge over the decision tree model by a slim margin in accuracy, making it the ideal choice for predicting the relapse chance in a patient.

7 | In The End

7.1 Future work

This thesis report has only scratched the surface of using synthetic data generation algorithms to create hyper-realistic data for rare medical conditions such as ANCA vasculitis. More evaluation of the synthetic data [7] needs to be done to ensure no new patterns are made. Work also needs to be done to identify methods for including non-numeric columns in the data generation process, as they also play an essential role in statistical analysis and machine learning.

Another future work task would include testing SMOTE-based algorithms for data generations on other rare conditions. As we know, each data has its properties and behaviour. It is essential to test if the algorithms will run smoothly on a different dataset and maintain the same characteristics as the original data. The actual test of this method would be on a validation dataset that was separate from the data from which the synthetic data set was created. This could be done by using live patient records as testing data to identify any missed scenario or condition that could affect the model's outcome.

Like any research, the final step would always be a way to use the proposed method in the real world. This method will open possibilities to make high-precision machine learning models using a limited dataset for rare diseases. Since the technique still needs medical professionals to work in tandem to provide the patient with the necessary treatment, it is crucial that we perfrom explainable analysis and create models

with easy-to-understand outcomes and not just a black box.

7.2 Conclusion

The main objective of this report was to identify a method for creating synthetic data given a dataset to train an unbiased and high-precision machine learning model. The thesis used the ANCA Vasculitis data from the RKD registry database. SMOTE, B-SMOTE and ANS were the three synthetic data generation algorithms used. On evaluation, we saw that the SMOTE algorithm created sufficient hyperrealistic data to train a machine learning model. While there was not much difference in the accuracy of both the decision tree model and the Random forest model, the trend was consistent throughout all the synthetic datasets. Constructing the ML models solely on the synthetic gives the freedom to use the data in a much broader manner since it is computer generated and does not belong to any individual, thus breaching no privacy clauses.

Bibliography

[1] Mrinal Kanti Baowaly, Chao-Lin Liu, and Kuan-Ta Chen. Realistic data synthesis using enhanced generative adversarial networks. In *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 289–292, June 2019.

[2] Silke R. Brix, Mercedes Noriega, Pierre Tennstedt, Eik Vettorazzi, Martin Busch, Martin Nitschke, Wolfram J. Jabs, Fedai Özcan, Ralph Wendt, Martin Hausberg, Lorenz Sellin, Ulf Panzer, Tobias B. Huber, Rüdiger Waldherr, Helmut Hopfer, Rolf A.K. Stahl, and Thorsten Wiech. Development and validation of a renal risk score in anca-associated glomerulonephritis. *Kidney International*, 94(6):1177–1188, 2018.

[3] Alison Callahan and Nigam H. Shah. Chapter 19 - machine learning in healthcare. In Aziz Sheikh, Kathrin M. Cresswell, Adam Wright, and David W. Bates, editors, *Key Advances in Clinical Informatics*, pages 279–291. Academic Press, 2017.

[4] Vivek Vinushanth Christopher. Abstract view of the functioning of smote by author. `https://towardsdatascience.com/handling-imbalanced-data-using-geometric-smote-770b49d5c7b5`. Accessed: 2020-08-20.

[5] Elena Csernok and Frank Moosig. Current and emerging techniques for anca detection in vasculitis. *Nature Reviews Rheumatology*, 10(8):494–501, Aug 2014.

[6] Irina Deeva, Petr D. Andriushchenko, Anna V. Kalyuzhnaya, and Alexander V. Boukhanovsky. Bayesian networks-based personal data synthesis. In *Proceedings*

of the 6th EAI International Conference on Smart Objects and Technologies for Social Good, GoodTechs '20, page 6–11, New York, NY, USA, 2020. Association for Computing Machinery.

[7] Andre Goncalves, Priyadip Ray, Braden Soper, Jennifer Stevens, Linda Coyle, and Ana Paula Sales. Generation and evaluation of synthetic patient data. *BMC Medical Research Methodology*, 20(1):108, May 2020.

[8] Feng Hu and Hang Li. A novel boundary oversampling algorithm based on neighborhood rough set model: Nrsboundary-smote. *Mathematical Problems in Engineering*, 2013, 01 2013.

[9] Yuanpeng J. Huang, Robert Powers, and Gaetano T. Montelione. Protein nmr recall, precision, and f-measure scores (rpf scores): structure quality assessment measures based on information retrieval statistics. *Journal of the American Chemical Society*, 127(6):1665–1674, Feb 2005.

[10] G G Hunder, W P Arend, D A Bloch, L H Calabrese, A S Fauci, J F Fries, R Y Leavitt, J T Lie, R W Lightfoot, Jr, and A T Masi. The american college of rheumatology 1990 criteria for the classification of vasculitis. introduction. *Arthritis Rheum.*, 33(8):1065–1067, August 1990.

[11] Yifan Jiang, Han Chen, Murray Loew, and Hanseok Ko. Covid-19 ct image synthesis with a conditional generative adversarial network. *IEEE Journal of Biomedical and Health Informatics*, 25(2):441–452, Feb 2021.

[12] Michael J Kemna, Jan Damoiseaux, Jos Austen, Bjorn Winkens, Jim Peters, Pieter van Paassen, and Jan Willem Cohen Tervaert. ANCA as a predictor of relapse: useful in patients with renal involvement but not in patients with nonrenal disease. *J. Am. Soc. Nephrol.*, 26(3):537–542, March 2015.

[13] Yang Liu, Xiang Li, Xianbang Chen, and Huaqiang Li. High-performance machine learning for large-scale data classification considering class imbalance. *Scientific Programming*, 2020, 05 2020.

[14] Nadav David Marom, Lior Rokach, and Armin Shmilovici. Using the confusion matrix for improving ensemble classifiers. In *2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel*, pages 000555–000559, Nov 2010.

[15] Puneet Mathur. *Key Technological advancements in Healthcare*, pages 13–35. Apress, Berkeley, CA, 2019.

[16] Anthony J. Myles, Robert N. Feudale, Yang Liu, Nathaniel A. Woody, and Steven D. Brown. An introduction to decision tree modeling. *Journal of Chemometrics*, 18(6):275–285, 2004.

[17] Sarang Narkhede. Confusion matrix [image 2] (image courtesy: My photoshopped collection). `https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62`. Accessed: 2022-08-20.

[18] Payam Refaeilzadeh, Lei Tang, and Huan Liu. *Cross-Validation*, pages 1–7. Springer New York, New York, NY, 2016.

[19] Steven J. Rigatti. Random Forest. *Journal of Insurance Medicine*, 47(1):31–39, 01 2017.

[20] Barry Ryan. Using molecular biomarkers to identify anca-associated vasculitis patients at risk of relapse. Year: 2020/21.

[21] Philip Seo and John H Stone. The antineutrophil cytoplasmic antibody-associated vasculitides. *Am J Med*, 117(1):39–50, July 2004.

[22] K. Shailaja, B. Seetharamulu, and M. A. Jabbar. Machine learning in healthcare: A review. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 910–914, March 2018.

[23] Liyan Sun, Jiexiang Wang, Yue Huang, Xinghao Ding, Hayit Greenspan, and John Paisley. An adversarial learning approach to medical image synthesis for lesion detection. *IEEE Journal of Biomedical and Health Informatics*, 24(8):2303–2314, Aug 2020.

[24] Juanjuan Wang, Mantao Xu, Hui Wang, and Jiwu Zhang. Classification of imbalanced data by using the smote algorithm and locally linear embedding. In *2006 8th international Conference on Signal Processing*, volume 3, Nov 2006.

[25] Leo Wang, Haiying Shen, Kyle Enfield, and Karen Rheuban. Covid-19 infection detection using machine learning. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 4780–4789, Dec 2021.

[26] Richard A Watts, Ravi Suppiah, Peter A Merkel, and Raashid Luqmani. Systemic vasculitis–is it time to reclassify? *Rheumatology (Oxford)*, 50(4):643–645, April 2011.

[27] Mi-Ja Woo, Jerome P. Reiter, Anna Oganian, and Alan F. Karr. Global measures of data utility for microdata masked for disclosure limitation. *Journal of Privacy and Confidentiality*, 1(1), Apr. 2009.

[28] Ye Yuan, Tianhong Quan, Youyi Song, Jitian Guan, Teng Zhou, and Renhua Wu. Noise-immune extreme ensemble learning for early diagnosis of neuropsychiatric systemic lupus erythematosus. *IEEE Journal of Biomedical and Health Informatics*, 26(7):3495–3506, July 2022.

A Appendix

A.1 Elbow Curve for Log Cluster

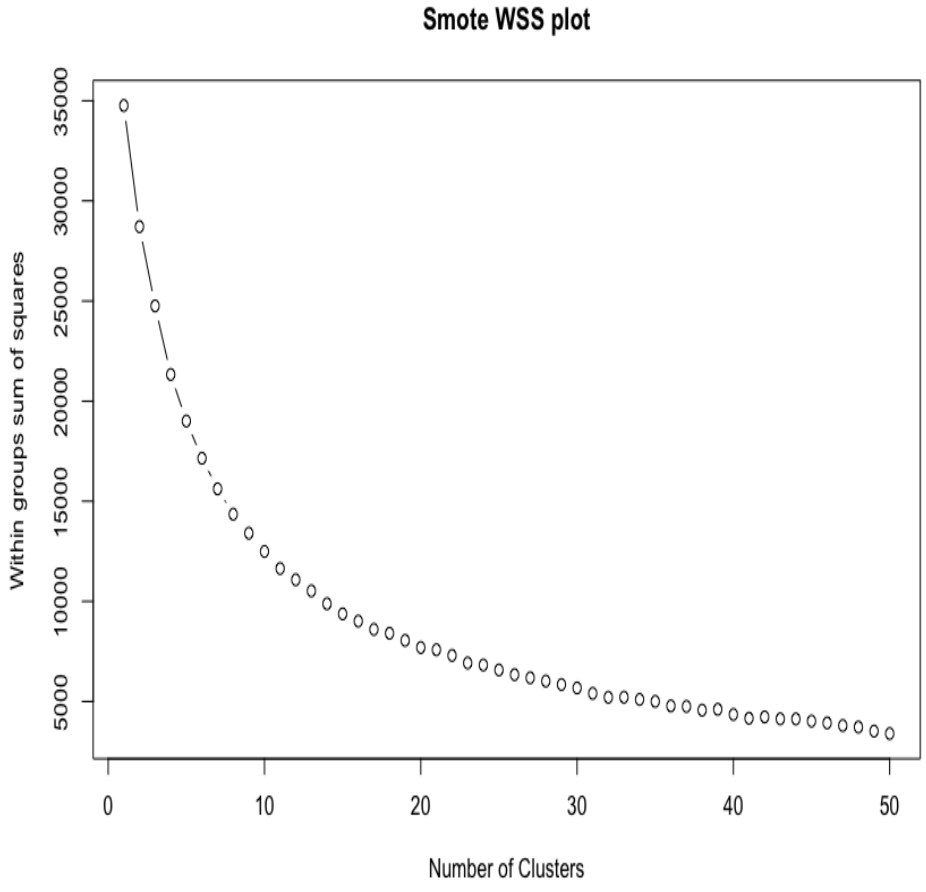


Figure A.1: Cluster count identification for Log Cluster - SMOTE

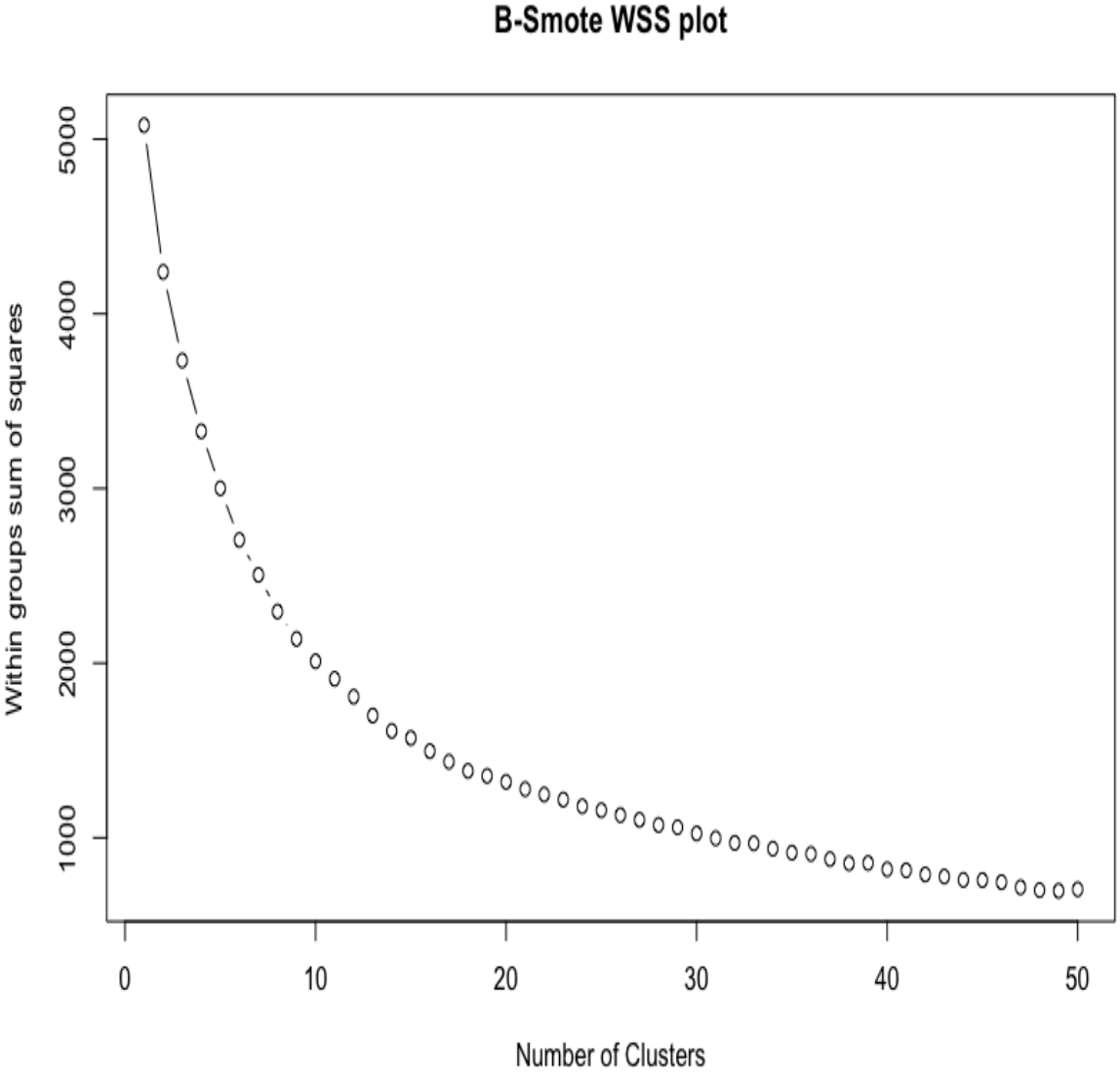


Figure A.2: Cluster count identification for Log Cluster - B-SMOTE

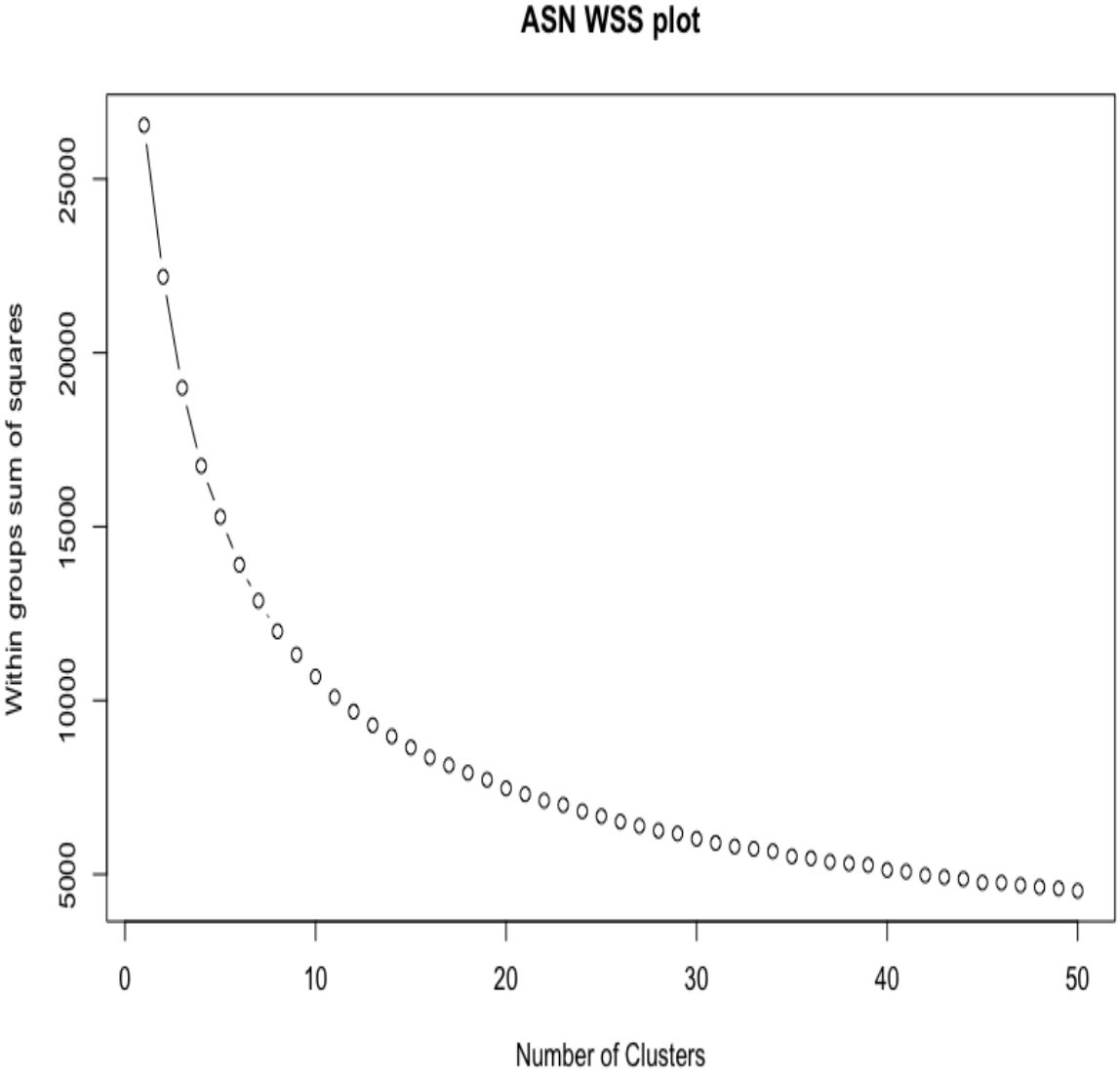


Figure A.3: Cluster count identification for Log Cluster - ANS

A.2 RandomSearch Result

params	mean_test_score	rank_test_score
"{'max_features': None, 'criterion': 'gini'}"	0.9516083916083916	1
"{'max_features': 'sqrt', 'criterion': 'gini'}"	0.9401398601398601	6
"{'max_features': 'log2', 'criterion': 'gini'}"	0.9365034965034965	8
"{'max_features': None, 'criterion': 'entropy'}"	0.9448951048951049	4
"{'max_features': 'sqrt', 'criterion': 'entropy'}"	0.9387412587412587	7
"{'max_features': 'log2', 'criterion': 'entropy'}"	0.9507692307692308	2
"{'max_features': None, 'criterion': 'log_loss'}"	0.9457342657342658	3
"{'max_features': 'sqrt', 'criterion': 'log_loss'}"	0.9359440559440559	9
"{'max_features': 'log2', 'criterion': 'log_loss'}"	0.944055944055944	5

Table A.1: SMOTE DT RandomSearch Result

params	mean_test_score	rank_test_score
"{'n_estimators': 180, 'criterion': 'gini'}"	0.9773426573426572	7
"{'n_estimators': 428, 'criterion': 'entropy'}"	0.975944055944056	10
"{'n_estimators': 265, 'criterion': 'gini'}"	0.9773426573426572	7
"{'n_estimators': 304, 'criterion': 'gini'}"	0.9793006993006994	1
"{'n_estimators': 150, 'criterion': 'gini'}"	0.9784615384615385	4
"{'n_estimators': 111, 'criterion': 'gini'}"	0.979020979020979	3
"{'n_estimators': 480, 'criterion': 'gini'}"	0.9779020979020979	6
"{'n_estimators': 498, 'criterion': 'gini'}"	0.9784615384615385	4
"{'n_estimators': 342, 'criterion': 'entropy'}"	0.9762237762237762	9
"{'n_estimators': 467, 'criterion': 'gini'}"	0.9793006993006992	2

Table A.2: SMOTE RMF RandomSearch Result

params	mean_test_score	rank_test_score
"{'max_features': None, 'criterion': 'gini'}"	0.9516083916083916	1
"{'max_features': 'sqrt', 'criterion': 'gini'}"	0.9401398601398601	6
"{'max_features': 'log2', 'criterion': 'gini'}"	0.9365034965034965	8
"{'max_features': None, 'criterion': 'entropy'}"	0.9448951048951049	4
"{'max_features': 'sqrt', 'criterion': 'entropy'}"	0.9387412587412587	7
"{'max_features': 'log2', 'criterion': 'entropy'}"	0.9507692307692308	2
"{'max_features': None, 'criterion': 'log_loss'}"	0.9457342657342658	3
"{'max_features': 'sqrt', 'criterion': 'log_loss'}"	0.9359440559440559	9
"{'max_features': 'log2', 'criterion': 'log_loss'}"	0.944055944055944	5

Table A.3: B-SMOTE DT RandomSearch Result

params	mean_test_score	rank_test_score
"{'n_estimators': 180, 'criterion': 'gini'}"	0.9773426573426572	7
"{'n_estimators': 428, 'criterion': 'entropy'}"	0.975944055944056	10
"{'n_estimators': 265, 'criterion': 'gini'}"	0.9773426573426572	7
"{'n_estimators': 304, 'criterion': 'gini'}"	0.9793006993006994	1
"{'n_estimators': 150, 'criterion': 'gini'}"	0.9784615384615385	4
"{'n_estimators': 111, 'criterion': 'gini'}"	0.979020979020979	3
"{'n_estimators': 480, 'criterion': 'gini'}"	0.9779020979020979	6
"{'n_estimators': 498, 'criterion': 'gini'}"	0.9784615384615385	4
"{'n_estimators': 342, 'criterion': 'entropy'}"	0.9762237762237762	9
"{'n_estimators': 467, 'criterion': 'gini'}"	0.9793006993006992	2

Table A.4: B-SMOTE RMF RandomSearch Result

params	mean_test_score	rank_test_score
"{'max_features': None, 'criterion': 'gini'}"	0.9516083916083916	1
"{'max_features': 'sqrt', 'criterion': 'gini'}"	0.9401398601398601	6
"{'max_features': 'log2', 'criterion': 'gini'}"	0.9365034965034965	8
"{'max_features': None, 'criterion': 'entropy'}"	0.9448951048951049	4
"{'max_features': 'sqrt', 'criterion': 'entropy'}"	0.9387412587412587	7
"{'max_features': 'log2', 'criterion': 'entropy'}"	0.9507692307692308	2
"{'max_features': None, 'criterion': 'log_loss'}"	0.9457342657342658	3
"{'max_features': 'sqrt', 'criterion': 'log_loss'}"	0.9359440559440559	9
"{'max_features': 'log2', 'criterion': 'log_loss'}"	0.944055944055944	5

Table A.5: ANS DT RandomSearch Result

params	mean_test_score	rank_test_score
"{'n_estimators': 180, 'criterion': 'gini'}"	0.9773426573426572	7
"{'n_estimators': 428, 'criterion': 'entropy'}"	0.975944055944056	10
"{'n_estimators': 265, 'criterion': 'gini'}"	0.9773426573426572	7
"{'n_estimators': 304, 'criterion': 'gini'}"	0.9793006993006994	1
"{'n_estimators': 150, 'criterion': 'gini'}"	0.9784615384615385	4
"{'n_estimators': 111, 'criterion': 'gini'}"	0.979020979020979	3
"{'n_estimators': 480, 'criterion': 'gini'}"	0.9779020979020979	6
"{'n_estimators': 498, 'criterion': 'gini'}"	0.9784615384615385	4
"{'n_estimators': 342, 'criterion': 'entropy'}"	0.9762237762237762	9
"{'n_estimators': 467, 'criterion': 'gini'}"	0.9793006993006992	2

Table A.6: ANS RMF RandomSearch Result

A.3 Full Decion Trees

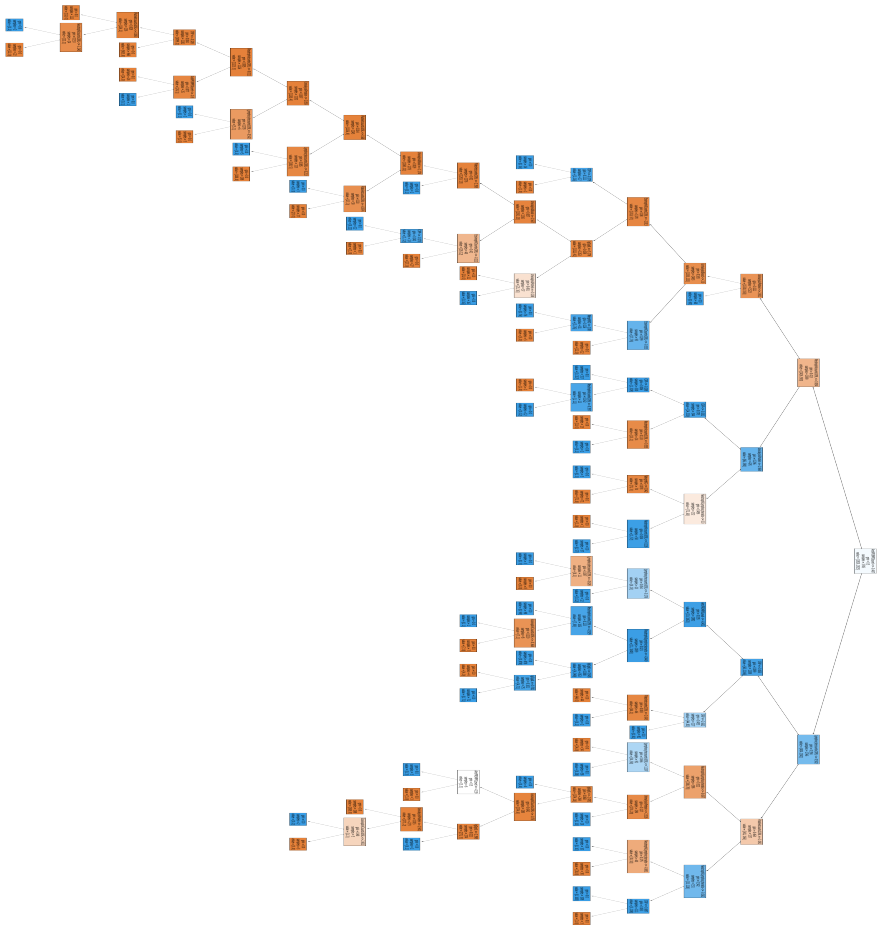


Figure A.4: Complete SMOTE Decition TREE

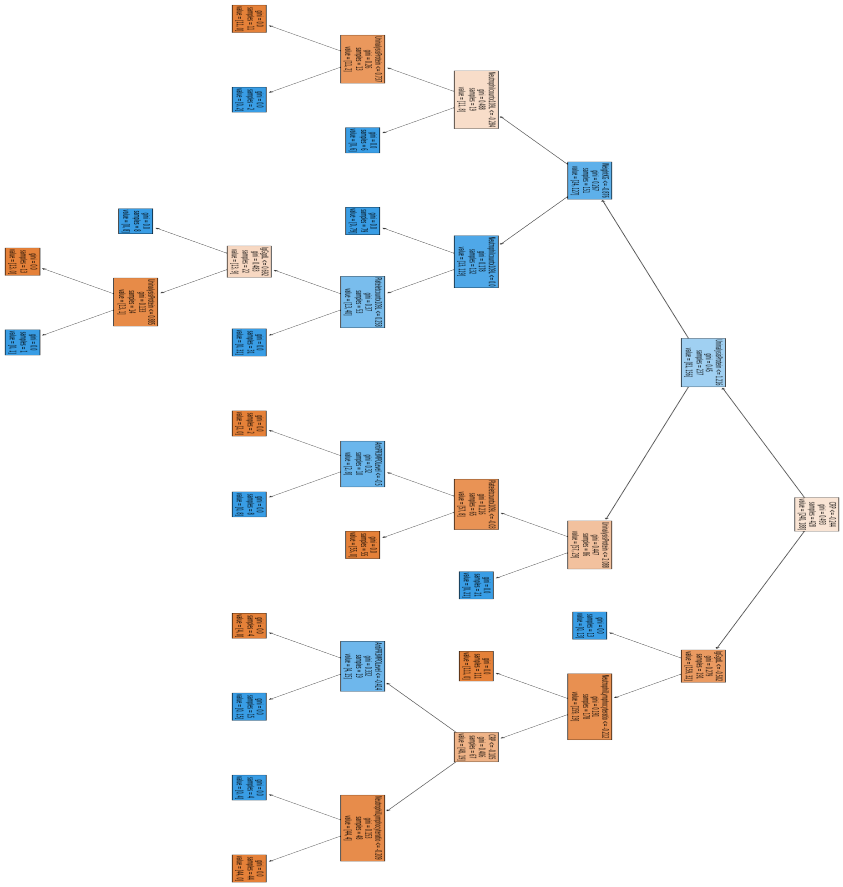


Figure A.5: Complete B-SMOTE Decition TREE

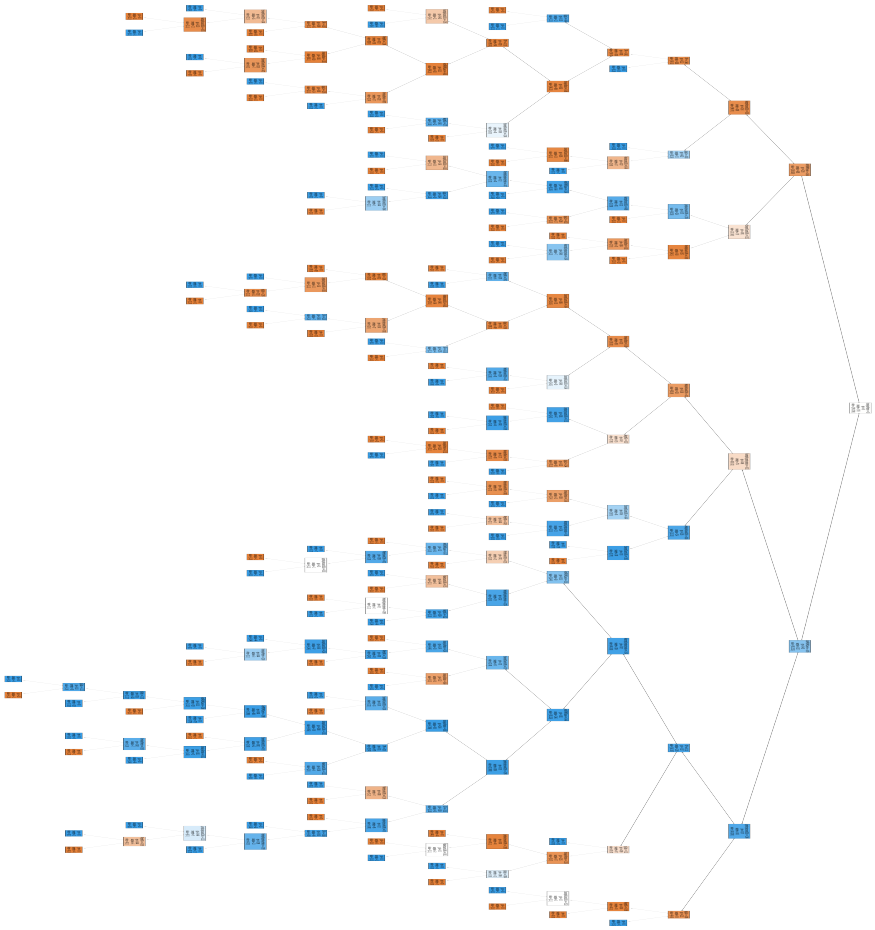


Figure A.6: Complete ANS Decition TREE