Copyright

by

Anuprabha Arputharaj

2019

# RULE EXTRACTION FOR INFREQUENT CLASS

by

Anuprabha Arputharaj, B.E

### THESIS

Presented to the Faculty of

The University of Houston-Clear Lake

In Partial Fulfillment

Of the Requirements

For the Degree

# MASTER OF SCIENCE

in Software Engineering

### THE UNIVERSITY OF HOUSTON-CLEAR LAKE

MAY, 2019

# RULE EXTRACTION FOR INFREQUENT CLASS

by

Anuprabha Arputharaj

## APPROVED BY

Soma Datta, PhD, Chair

Kewei Sha, PhD, Committee Member

Khondker Hasan, PhD, Committee Member

# RECEIVED/APPROVED BY THE COLLEGE OF SCIENCE AND ENGINEERING:

Said Bettayeb, PhD, Associate Dean

Ju H. Kim, PhD, Dean

# Dedication

То

The Field of Software Engineering and My family

#### Acknowledgements

Firstly, I would like to express my very profound gratitude to the Almighty God and those who all directly and indirectly guided and supported me throughout my thesis work. I will always be grateful for their selfless love and help.

I would like to thank my advisor and my committee chair Dr. Soma Datta for her extensive professional and personal guidance throughout my thesis work. From day one her motivation and support have guided me in the right direction towards completing my thesis successfully. Her deepest knowledge of this research field inspired me to work more on this paper. Besides, I express my sincere gratitude to my thesis committee, Dr. Kewei Sha and Dr. Khondker Hasan for their invaluable suggestions and encouragement.

Nobody has been more important to me in the pursuit of this thesis work than the members of my family. I express my immense love to my Grandparents, Appa (Dad) Arputharaj, Amma (Mom) Sukirtha and my brother Vasanth. It is only their love and prayers which guided me throughout my research work. The reach I have made is completely impossible without them. My greatest thanks and love to all my friends back in India, Saran, Aarathi, Niveda, Mythili and Gopal for their immense love and encouragement in all my tough times.

Finally, I would like to thank my friends and roommates in the US who made me feel at home. I extend my greatest gratitude to Parthipan, Khaviya, Hari Kumar, Vignesh, Sapnah, Ramanjit, Sasidhar and the members of Clear Lake Baptist Church for showering me with their love and support continuously.

#### ABSTRACT

#### RULE EXTRATION FOR INFREQUENT CLASS

Anuprabha Arputharaj University of Houston-Clear Lake, 2019

#### Thesis Chair: Soma Datta, PhD

The thesis narrates the classification rules that are developed in the infrequent class to make decisions about their future actions. Rules are the most expressive and most human-readable representation for any kind of hypotheses made in the prediction world. Dealing with the imbalanced datasets it is always portrayed that the standard classifier algorithms are always biased towards the Majority class which finally gives more rules for the majority class when compared to the infrequent class. That is because the conventional algorithms loss functions attempt to optimize quantities such as error rate and not taking the data distribution into consideration. The importance of the infrequent class will be picturized clearly only in the form of the rules that are developed from them. The thesis emphasizes the use of Undersampling technique which is one of the naïve methods used to balance the data and apply the clustering algorithm which clusters the attributes of the similar features and categorize them according to their distance as Euclidean distance and Manhattan distance. The clusters that are generated from the Euclidean distance contributes to the majority class and the Manhattan distance contributes to the minority class. This helps in

increasing the minority count of the dataset when compared to the original dataset. Creating a new dataset from them are applied to the conventional classification algorithm to obtain more rules for the minority class which helps in further predictions. The proposed algorithm generates more readable and understandable rules with increased coverage for the minority class when compared to the previously published works.

List of Tables	X
List of Figures	xi
CHAPTER I: INTRODUCTION	1
CHAPTER II: MOTIVATION AND CONTEXT	7
What is an imbalanced class?	7
What is Random Undersampling?	7
What is K-Means Clustering?	8
Why distance measures?	8
What are the Performance Metrics?	9
What is rule extraction?	9
Limitations of Sampling and Clustering algorithms	9
CHAPTER III: RELATED WORK	11
CHAPTER IV: METHODOLOGY	15
Preprocessing the original dataset	
Modeling nearest Neighbor	16
Handling the missing values	16
Undersampling using WEKA software	17
Clustering using K-Means Clustering algorithm	
K-Means algorithm	18
Elbow method	19
Distance Measures	21
Euclidean Distance	21
Manhattan Distance	22
Combining the clusters from Euclidean and Manhattan distances	24
Applying Decision Tree algorithm	25
Performance Metrics	
CHAPTER V: EXPERIMENTAL AND RESULTS	29
Adult dataset	29
CHAPTER VI: CONCLUSION AND FUTURE WORK	46
Conclusion	46
Future Work	47
Contribution to Research Community	47

REFERENCES	
APPENDIX A: OTHER EXPERIMENTAL RESULTS	51
Adult Dataset	51
Breast Cancer Dataset	
Balance Dataset	
Car Dataset	
House Vote Dataset	
Mushroom Dataset	
Student Retention Dataset	

# LIST OF TABLES

Table 1.1 Confusion matrix for two-class classification	3
Table 4.1 Experimental setup for KValid algorithm	
Table 4.2 Comparison of performance metrics	
Table 5.1 Original class distribution	
Table 5.2 After applying the nearest neighbor algorithm	
Table 5.3 Class distribution settings	
Table 5.4 After applying the undersampling technique (Spread Subsample)	
Table 5.5 After applying the K-Means algorithm to the balanced dataset	
Table 5.6 New dataset after a combination of clusters	
Table 5.7 Decision tree for the clustered dataset	
Table 5.8 Experimental setup for Rule classification algorithms	
Table 5.9 Displaying the output from several rule classifiers	
Table 5.10 Displaying the dataset's characteristics	
Table 5.11 The proposed methodology for all the datasets	
Table 5.12 Clustering and Decision tree algorithm for all the dataset	41
Table 5.13 Apriori Settings	
Table 5.14 Decision Tree and Association mining for all the datasets	
Table 5.15 Displaying Top Rules	
Table A.1 Original dataset applied with the c45 algorithm	72
Table A.2 Original dataset applied with EM clustering	73
Table A.3 Original dataset applied with K-means algorithm	74
Table A.4 Original dataset applied with Make Density cluster	75
Table A.5 Comparison of the several algorithms	76

# LIST OF FIGURES

Figure 1.1. Class Distribution of Imbalanced Class	2
Figure 4.1. Overall workflow	15
Figure 4.2. Undersampling the majority class	
Figure 4.3. Elbow curve	21
Figure 4.4. Euclidean Distance	22
Figure 4.5. Manhattan distance	23
Figure 4.6. Combining the clusters	
Figure 5.1. Comparison of the rule extraction	44
Figure A.1. Original Class Distribution	51
Figure A.2. After Applying Nearest Neighbor Algorithm	51
Figure A.3. After Applying Undersampling technique (Spread Sub Sample)	52
Figure A.4. After applying K-Means algorithm to the balanced dataset	52
Figure A.5. New Dataset after the combination of clusters	53
Figure A.6. Decision tree for the clustered dataset	53
Figure A.7. Original Class Distribution	54
Figure A.8. After applying the nearest neighbor algorithm	54
Figure A.9. After applying the undersampling technique (Spread Sub Sample)	55
Figure A.10. After applying K-means algorithm to the balanced dataset	55
Figure A.11. New dataset after a combination of clusters	56
Figure A.12. Decision tree for the clustered dataset	56
Figure A.13. Original class distribution	57
Figure A.14. After applying the nearest neighbor algorithm	57
Figure A.15. After applying Undersampling technique (Spread Sub Sample)	
Figure A.16. After applying the K-means algorithm to a balanced dataset	
Figure A.17. New dataset after a combination of clusters	59
Figure A.18. Decision tree for the clustered dataset	59
Figure A.19. Original class distribution	60
Figure A.20. After applying the nearest neighbor algorithm	60
Figure A.21. After applying the undersampling technique (Spread Sub Sample)	61

Figure A.22. After applying K-means algorithm to a balanced dataset	61
Figure A.23. New dataset after a combination of clusters	62
Figure A.24. Decision tree for the clustered dataset	62
Figure A.25. Original class Distribution	63
Figure A.26. After applying the nearest neighbor algorithm	63
Figure A.27. After applying the undersampling technique (Spread Sub Sample)	64
Figure A.28. After applying K-means algorithm to a balanced dataset	64
Figure A.29. New dataset after a combination of clusters	65
Figure A.30. Decision tree for the clustered dataset	65
Figure A.31. Original class distribution	66
Figure A.32. After applying the nearest neighbor algorithm	66
Figure A.33. After applying the undersampling technique (Spread Sub Sample)	67
Figure A.34. After applying K-means algorithm to the balanced dataset	67
Figure A.35. New dataset after combining the clusters	68
Figure A.36. Decision tree for the clustered dataset	68
Figure A.37. Original class distribution	69
Figure A.38. After applying the nearest neighbor algorithm	69
Figure A.39. After applying the undersampling technique (Spread Sub Sample)	70
Figure A.40. After applying K-means algorithm to a balanced dataset	70
Figure A.41. New dataset after a combination of clusters	71
Figure A.42. Decision tree for the clustered dataset	71

#### CHAPTER I:

#### INTRODUCTION

Most real-world classification problems display some level of class imbalance, which is when each class does not make up an equal portion of the dataset. In this distribution of dataset, the total number of negative samples is called the majority class since it contains a majority of the instances and the total number of positive samples is called the majority class as it contains a smaller number of instances. The class imbalance problem is one of the important problems for classification studies in data mining because this class imbalance targets real-life applications like credit card transactions, medical diagnoses in the identification of rare diseases, detection of oil spills, financial industry, e-mail foldering, anomalies, electricity pilferage, etc. The classification problem for imbalanced data is interesting and challenging to researchers because most standard data mining methods are biased and inaccurate towards the imbalanced class [2]. This means that the standard learning model suffers from the accuracy paradox which causes poor classification of the majority class. The main reasons for the poor performance of the existing classification algorithms on imbalanced datasets are 1. The conventional algorithms are accuracy driven and they tend to minimize the overall error to which the majority class contributes very little, 2. They tend to assume that there is an equal distribution of data for all the classes. An imbalanced problem in the data could interfere with the detection process and lead to misclassifying the problem which involves real-life application datasets [2].

Researchers are addressing this data imbalance as a major problem because the prediction rate of the minority class is very low when compared to the majority class mainly in disease diagnosis and fraudulent detections [3] [4].



Figure 1.1. Class Distribution of Imbalanced Class

The imbalanced class problem can be explained with the help of the most popular imbalanced dataset which involves fraudulent and non-fraudulent transactions. The fraud observations constitute just 0.1% of the entire dataset, representing a typical case of the imbalanced class. After applying the classification algorithm, the model shows a very high accuracy since it contains 95% of the transactions which are non-fraudulent, and the model predicts all non-fraudulent transactions as accurate. However, because fraudulent only accounts for 0.1% and predictive fraud accounts for 5% of the total observations, there is no evidence that the model has higher accuracy since the 5% of the observations are not taken into account since they are considered as noise. In this case, the cost of the false negative is usually much larger than the false positive, yet the machine learning classification algorithms penalize both with similar weight. The importance of the imbalanced class is neglected here due to the minimal number of observations. This leads to a minimal number of rules from the target class, which are unable to predict how the fraudulent transactions are made. This classification algorithm performance is measured by the confusion matrix which contains information about the actual and predicted classes.

Table 1.1Confusion matrix for two-class classification

Actual	Predicted		
	Positive class	Negative class	
Positive class	True positive (TP)	False negative (FN)	
Negative class	False positive (FP)	True negative (TN)	

Accuracy of a model= (TP + TN) / (TP + FN + FP + TN)

Researchers have generally addressed two kinds of solutions for data classifications dealing with imbalanced problems: solving by data level by sampling and solving by algorithm level by using sophisticated design classification approaches. The research on algorithm level improves the classification algorithm mainly according to the characteristics of the imbalanced datasets. This approach can be executed two ways; they are the costsensitive approach and the recognition-based approach. This primarily means setting different weights for different classes, changing the probability density distribution, and adjusting the classification boundaries. A problem with using the algorithmic approach is that most of the machine learning algorithms penalize false positive and false negative values equally. So, modifying the algorithm itself will boost the performance of the majority class. Most of the tree-based ensemble techniques are efficient for this approach

[11]. This approach works by combining predictions from multiple models. These models are broadly classified into two categories; they are the bagging-based trees and boosting based trees. The research on data level deals with the instances involved in the majority and majority class data and performs some data preprocessing techniques in which the model is trained. This approach deals with re-sampling the dataset which would mitigate the effect caused by class imbalance. This approach is mainly classified into two categories; they are the oversampling and undersampling techniques. This data level approach has gained popular acceptance among researchers as it is more flexible [4], [10], [14], [15]. Further, the rules that are generated from either the data level approach or the algorithm level approach gives us a better understanding of the target class.

The proposed methodology involves the combination of both the data level and algorithm level approach to solving the imbalanced class problem. Since the standard classification algorithms in the data mining technique are biased towards the majority class, this is an important area where the issue must be fixed. A simple way to fix the imbalanced datasets is simply to balance them using the sampling level approach which is the undersampling technique. The undersampling technique is generally divided into two types; they are the Prototype generation and Prototype selection [17]. The Prototype generation is a technique which will reduce the number of samples in the target class and the remaining samples will be generated and are not selected from the original dataset. The Prototype selection is a technique which will select samples from the original dataset. The proposed methodology comes under the Prototype selection technique which is the undersampling technique. This technique selects samples from the majority class. The next step is the clustering technique which helps in finding the structure of the data. This unsupervised learning method divides the data objects into designated clusters based only on the information present in the dataset. The clustering method involved in this thesis is the K-Means clustering technique. This method classifies the given dataset into several clusters defined by the value of "k" in which each object belongs to the cluster with the nearest mean. The next step is the core part of the work which is the implementation of different distance measures. The distance metrics use a distance function to help the algorithms recognize similarities between the data instances. Instead of the default distance measure that is employed in the clustering technique, the proposed methodology involves two types of distance measures based on their coverage of instances in the feature space,

which is Euclidean distance and Manhattan distance, this distance measure is subjective and is dependent on the domain and the application of the dataset.

This study provides a comparative study among the popular data level approaches and algorithm level approaches which gives us a clear and better understanding about the characteristics of the dataset and which type of approach is suitable for which type of dataset.

The proposed methodology has shown some improvement in the prediction rate of the majority class via a rule-based classification algorithm which is the Decision tree (C45). Several datasets from various applications are analyzed in this method. Sampling and clustering algorithms are applied to it to gather the results. To measure the performance of the algorithms, metrics play a key role in the infrequent class. The widely used metric to measure the performance of the algorithm is the accuracy metric. But in case of class imbalance accuracy performs in a biased fashion towards the majority class and hence appropriate metrics like precision, recall, F1 measure, ROC and AUC curves are chosen for the study. Here precision describes the number of positive predictions that were correct, and recall describes the coverage of actual positive samples and F1 measure describes the harmonic mean of precision and recall. The ROC (Receiver Operating Characteristics) is the probability curve and this curve is plotted with True Positive Rate (TPR) against the False Positive Rate (FPR). The AUC (Area under Curve) represents the degree or measure of separability. Higher the AUC curve, better is the model in distinguishing between the classes. And finally, the rules that are generated from the classifiers gives us information about the attributes that contributes to the minority class. The rules obtained here are extracted from the Decision tree. This is because decision trees are build based upon the entropy and the information gain which is beneficial while considering imbalanced classes. The proposed algorithm uses rule-based classification as they are suggestive, easy to

generate and easy to interpret. However, these advantages are hypothetical from the proposed work. Certainly, the thesis work focuses on the importance of the distance measures as it defines how the similarity of elements is calculated and how it will influence the shape of the clusters. The work aims in evaluating the effects of the two different distance measures on seven different datasets which are taken from the UCI repository. The datasets are divided into three categories: Categorical, Numerical and Mixed datasets.

# CHAPTER II: MOTIVATION AND CONTEXT

The motivation and context of the thesis is explained using various data mining and machine learning contexts such as the "Imbalanced class", which explains the necessity of considering the majority class in a dataset; Supervised learning algorithms, which helps in finding the relationships of the input and effectively produce precise output; Unsupervised learning algorithm, which helps in the exploratory analysis of the whole dataset; Distance Measures, which aids in finding the similarity among the data instances and ultimately extracting rules for the infrequent class, helping in future predictions from the entire dataset which plays a crucial role in the thesis.

#### What is an imbalanced class?

A dataset is said to be an imbalanced dataset when the classification categories (target classes) are not approximately equally represented. And the class which has a comparatively smaller number of instances than the others are said to be an imbalanced class. This topic has gained the attention from researchers [2], [3] because this problem affects real-world applications such as Medical diagnosis; fraud detection in areas like credit cards, phone calls, insurance, etc.; network intrusion detection, pollution detection fields like biomedical and bioinformatics and fields like remote sensing which includes land mines, underwater mines, etc. The standard classifiers assume that the training samples are equally distributed among the classes and they behave in a biased fashion towards the majority class [2]. Hence certain ensemble algorithms are intended to be applied in the minority class which evaluates their performance.

#### What is Random Undersampling?

The Random Undersampling method randomly chooses a set of majority class instances and removes these samples to adjust the balance of the original dataset. This helps

in keeping the minority class instances intact and further increases the sensitivity of the model. The core idea of the thesis work is to extract more rules for the infrequent class and hence this type of sampling is chosen so that the entire infrequent class remains as a whole throughout the entire process instead of losing some data from the infrequent class. Undersampling works better in terms of time and memory complexity [6].

#### What is K-Means Clustering?

When the number of examples representing positive classes differs from the number of examples representing a negative class, clustering is a highly useful technique to overcome the challenge faced by the imbalanced class as it finds the hidden relationship between each one to group the instances into clusters [15]. Generally, this Unsupervised learning method aims to divide the data objects into groups so that the objects in the same group are like one another and different from objects in other groups of clusters. The thesis work involves the K-Means clustering algorithm which is the most widely used clustering algorithm in many cluster ensemble studies.

#### Why distance measures?

Distance or Similarity measure in the data mining context is the distance with dimensions which represents the features in the dataset. If the distance is small, it will be a high degree of similarity and when the distance is large it will be a low level of similarity. Of the several distance metrics used in machine learning algorithms, the default distance used by the algorithms is the Euclidean distance. Since Euclidean distance is highly sensitive to sparse data and it covers only a spherical domain space [18], an alternative distance measure is to be introduced to handle the infrequent class. The alternative distance is the Manhattan distance, which in turn can handle higher dimensions of data and it has more coverage of the data points in the data space when compared to Euclidean distance [18].

#### What are the Performance Metrics?

The final step after building a model for a classification model is to look at the accuracy of the model and validate the correct number of predictions from all the predictions made. Though we have a model which we assume to produce robust predictions, it is also crucial to know whether the model is good enough to solve our problem. In order to make this decision, accuracy alone is typically not enough to make the decision and this metric could be misleading in the case of imbalanced classes [19]. The other performance metrics which contribute to the imbalanced classes are the precision, recall, F-Measure, and ROC.

#### What is rule extraction?

The rule extraction is a procedure which is meant to find frequent patterns, correlation, and associations among the attributes in the dataset. The rule extraction can be done directly by using Sequential covering algorithms or indirectly by using data mining methods like Decision tree building or Association rule mining. Of the several methods, this thesis emphasizes Decision tree algorithms because they help in finding the attributes that return the highest information gain. In Decision trees, the nodes are aligned such that the entropy decreases with further splitting downwards. This means that more appropriate splitting gives a definite decision which is highly needed for the minority class [11].

#### Limitations of Sampling and Clustering algorithms

Balancing the imbalanced dataset is the most common approach of handling the class imbalance efficiently which increases the performance of the dataset. Yet the sampling algorithm has a limitation of information loss and involvement of noisy instances which degrade the performance. The proposed methodology handles this limitation by involving a data cleaning process. Further, the K-Means clustering technique has the limitation of choosing the K-Value which is mandatory for dividing the clusters. The proposed methodology handles this situation by deciding the cluster size with the help of the Elbow method. Despite all the limitations the proposed methodology performs well for the infrequent class which is the target of the thesis work.

#### CHAPTER III:

#### RELATED WORK

Most of the real-world applications are suffered by the class imbalance and they are solved using Supervised and Unsupervised learning algorithms which influenced us to develop the existing methodology to generate better results.

Neelam Rout et al. [1] analyzed the performances of various algorithms that handle the imbalanced datasets. They discussed the pros and cons of the widely used Data level approaches, Algorithm level approaches, and Ensemble and Hybrid methods. They performed the experiments with the datasets from KEEL repository.

Usha Rani et al. [2] performed the Sampling algorithm which is the Synthetic majority oversampling technique on the imbalanced breast cancer dataset. To get rid of redundant and unnecessary attributes Principal Component Analysis (PCA) is applied. After preprocessing the experiments are conducted with 5 classifiers: -KNN (K nearest neighbor), SVM (Support Vector Machines), Logistic Regression, C45, and Random Forest.

Donghui et al. [3] explored the effectiveness of using cost-sensitive learning methods to classify the unknown cases in imbalanced bad debts datasets and compares with the results of other methods: Oversampling and Undersampling. In addition, it also analyzes the function of the semi-supervised learning method in different circumstances. This showed an improvement in good classification accuracy rates.

Jaya Lakshmi et al. [4] evaluated the effectiveness of various combinations of Undersampling, SMOTE, Cost-Sensitive learning, Ensemble techniques like Bagging, AdaBoost and Random Forest classification algorithms. The performance is compared by considering the precision, recall, F-Measure and area under ROC curve accuracy measures. They concluded that the combination of SMOTE and Bagging with Random Forest classification algorithm gave the best AUROC.

Ignacio et al. [5] presented a novel supervised classification approach for Induction motor faults based on Adaptive boosting algorithm with an optimized sampling technique which is the SMOTE algorithm and the training data is applied on the K-Fold cross-validation with different values of K. The results obtained from the training data is applied with the AdaBoost algorithm. The performance of the algorithm is measured by Sensitivity, Precision, and Recall.

Jia Song et al. [6] proposed a bi-directional sampling based on clustering for the imbalanced data classification. This algorithm combines the SMOTE oversampling algorithm and Undersampling algorithm based on K-Means to solve within-class imbalance problem and between class imbalance problems. This method makes the training dataset balance both between class and within the class. Over-fitting by random oversampling and important samples deleted by undersampling are avoided.

Tince et al. [7] proposed the SMOTE-Simple Genetic Algorithm which determines the sampling rate of each instance in order to obtain unequal amounts of synthetic instances. The tests are performed and compared by measuring using G-measure and F-measure.

Anantaporn et al. [8] proposed a technique which is a hybrid sampling approach which is the combination of well-known oversampling algorithm called SMOTE and the undersampling technique by removing the ambiguous instances from the majority class instances. The algorithm is divided into three parts which involve: grouping, sampling, and gathering. And finally, the trained data is applied with Decision tree algorithm and Naïve Bayes model in order to calculate the F-Measure and the accuracy of the model. Yoga et al. [9] explained about the imbalanced class on multiclass Education Data Mining dataset which is handled by the mechanism of the combination of SMOTE and OSS (One Sided Selection) which provided a balancing mechanism for the dataset's distribution, which showed the classification results enhancement in terms of classification performance. Here the working principle of OSS is the same as that of the Undersampling technique which divides the samples as Noise, Borderline, Redundant and Safety.

Datta and Mengel [10] proposed an Elastic Multi-Stage Decision Methodology to create rules for the infrequent class. The proposed methodology is divided into three parts: Clustering (which made a study on two important clustering techniques: EM and K-means), Minimizing the depth of the Decision tree and Association mining. In this technique, the rules obtained from the decision tree are generated after each split which covers a higher accuracy range. Pruning the decision trees allowed to contribute more accurate rules for the infrequent class. Further, they extended their research [11] and proposed Adaptable Multi-Phase rules over the infrequent class which involves two techniques: Decision trees and Association mining. This ensemble learning is used in an adaptive manner so that they expand and contract to accommodate the characteristics of the dataset.

Azadeh et al. [12] proposed a new confabulation-inspired association rule mining for rare and infrequent item sets. This approach uses a cogency inspired measure for generating rules. For the rule comparison, the measure used here is the Classification error rate. This algorithm can produce a higher performance for mining association rules from rare items, particularly when the rare items are important.

Astha et al. [13] proposed a hybrid sampling method, SCUT: Multiclass Imbalanced data classification using SMOTE and Cluster-based undersampling. This approach oversamples majority class examples through the generation of synthetic examples and employs cluster analysis in order to undersample the majority classes. Mishra [14] in his paper compared the results of two sampling techniques: SMOTE and Random Undersampling with and without proper validation on a randomly generated imbalanced dataset, with Random Forest and XG Boost as the underlying classifiers.

Santhosh Kumar et al. [15] proposed a Subset K-Means approach for handling imbalanced distributed data. The proposed algorithm consists of a random subset generation technique implemented by defining a number of subsets depending upon the unique properties of the dataset.

Monica et al. [17] in their paper made a preliminary study on the Prototype selection in imbalanced data for dissimilarity representation. This paper conducted a study to investigate the effects of several prototype selection schemes when the datasets are imbalanced and also their benefits when the class imbalance is handled by resampling the dataset.

Bora et al. [18] in their paper conducted an experimental study in MATLAB regarding the effect of distance measures on the performance of K-Means clustering algorithm. This paper involved different types of distance measures and evaluated their performances based on the datasets.

Chawla [19] in his paper, Data mining for imbalanced class: An overview discussed the sampling techniques that are used for balancing the datasets and certain performance measures that are appropriate for mining the imbalanced datasets.

Zhang et al. [20] in their paper, Efficient missing data imputation for supervised learning discussed about the quality of the supervised learning algorithms which are affected by the missing values and proposed an imputation algorithm to handle them.

#### CHAPTER IV:

#### METHODOLOGY

The overall workflow of the thesis is explained in figure 4.1. Initially, the imbalanced dataset is sent into the data cleaning process which then leads to the balancing of the dataset. After undersampling the dataset, the clustering algorithm is applied which divides the data set into many clusters. The divided clusters are applied with the Euclidean and Manhattan distance. A Decision tree classification algorithm is applied to the final clusters which help in extracting rules for the majority class.



Figure 4.1. Overall workflow

#### **Preprocessing the original dataset**

The datasets handled in this thesis belong to the imbalanced class. Since the classes must be balanced, some of the instances must be removed. If the data instances are removed randomly, that might affect the overall performance of the dataset in the upcoming processes. Hence, the instances must be removed carefully. Initially, data instances are divided into three categories. They are safe instances, borderline instances, and noisy instances. Safe instances are those that help in explaining the target class. These instances are located very close to the target class. Borderline instances are the ones which are located either very close to the decision boundary between majority and minority classes or located in the area surrounding class boundaries where classes overlap. The noisy instances are the ones that belong to one class located deep inside the region of the other class. These instances do. These instances also act as anomalies in certain datasets. Among these instances, the noisy instances must be removed from the dataset as they would degrade the performance. Missing values in the dataset will degrade the performance of the dataset and they have to be handled.

#### **Modeling nearest Neighbor**

This approach will analyze the class distribution in the K-nearest approach which fixes the concern by the number of nearest instances. This method helps to get rid of the outliers which are located farthest away. Since the outliers don't explain well about the target class, they are eliminated. In this way, only the safe and borderline instances are considered for the majority class.

#### Handling the missing values

The concept of missing values is important in handling a dataset because if the missing values are not handled properly then we may end up drawing an inaccurate

inference about the dataset [20]. There are three main reasons for missing values in a dataset. They are as follows:

CASE 1: Missing At Random (MAR): Here the missing values are not randomly distributed but are distributed within one or more sub-samples.

CASE 2: Missing Completely At Random (MCAR): This exists when the missing values are randomly distributed across the datasets.

CASE 3: Missing Not At Random (MNAR): This exists when the missing values depend on the hypothetical value and when the missing value is dependent on some other variable's value.

Generally, there are two methods of handling the missing values: Deleting the values and performing Data Imputation. By analyzing the characteristics of the dataset that are involved in this thesis, the datasets fall under the categories of Case 1 and Case 2. Removing the data with missing values, depending upon their occurrences, is the safest method for the datasets since it falls under the first two cases [20].

#### Undersampling using WEKA software

Random Undersampling is implemented in the WEKA software tool which is performed via a Spread subsample filter. This non-heuristic method randomly undersamples the majority class based on the spread frequency which is user-defined between the rarest and the most common classes. The number of instances is now reduced which helps in the feasibility of learning. Figure 4.2 explains about the undersampling in the software where some of the important samples from the majority class are taken and balanced with the minority class.



Figure 4.2. Undersampling the majority class

### **Clustering using K-Means Clustering algorithm**

This Unsupervised learning method allows us to create clusters which refer to the collection of data instances aggregated together because of certain similarities. Predefining the number of clusters (K) to be created, the algorithm divides the dataset accordingly. At this stage, two different types of distance measures are used to divide the clusters. The distances are the Euclidean distance and the Manhattan distance.

#### K-Means algorithm

This K-Means algorithm analyzes the natural groups of the data instances based on similarities. The algorithm locates the centroid of the groups and then evaluates the distance between each point from the centroid of the cluster. There are two main steps in this algorithm: The Data Assignment step and the Centroid Update step [16]. The steps involved in this algorithm are as follows:

• Determine the K value for each dataset after undersampling.

- Identifying the cluster centroids (mean point) for each cluster.
- Computing the distance from each instance and allot instances to the cluster where the distance from the centroid is minimum.
- After re-allocating the instances, the centroid of the newly formed clusters is determined.

#### **Elbow method**

Determining the number of clusters is highly crucial in the exploratory analysis of the algorithm. The K-Value decides the number of clusters to be created. The number of clusters must be decided carefully as they could influence the performance of the algorithm. There are several ways in determining the K-Value. Since the thesis works on the impact of distance measure in several algorithms, we end up using the Elbow method for determining the number of clusters. To use the Elbow method in WEKA, we use KValid algorithm which is a simple clustering package. It uses the simple K-Means algorithm as a backend to cluster the instances and tells which the best K-Value is and plots the graph.

The KValid algorithm is a package which must be installed in WEKA software and certain experimental setup has to be done in order to change the default values. This KValid is a simple clustering evaluation package for WEKA. It uses simple K-Means algorithm as a backend to cluster the instances and evaluates the clusterer using some algorithms, currently Silhouette-Index and Elbow method. It also validates simple K-Means algorithm. Since KValid is a cluster algorithm we can see that in cluster menu in WEKA. The experimental setup for the KValid algorithm is shown in table 4.1 which helps in plotting the graph which determines the number of clusters to be used for every dataset.

Table 4.1Experimental setup for KValid algorithm

Options	Default Values	Experimental Values	What They Mean
Cascade	False	True	Iterative algorithm which
			produces values for more
			successively densely
			spaced instances
Debug	False	True	Displays the output
Distance Function	Euclidean	Euclidean	Used to find similar data
			objects
doNotCheckCapabilities	False	True	Returns their capabilities
			in regard to their datasets
Seed	10	10	Seed for random data
			shuffling
Initialization method	Random	K-Means ++	Guarantees centroid
			initialization for KValid
			algorithm
minimumK	3	2	Sets the minimum value
			for the clusters
maximumK	10	10	Sets the maximum value
			for the clusters
show Graph	False	True	Displays the elbow curve
			graph
validation Method	Silhouette Index	Elbow method	Validates the simple
			kmeans algorithm and
			determines the best value
			of K

The average within cluster distance to the centroid as a function of "K" value is plotted and the "elbow point" is where the rate of distance decreases sharply and determines the number of clusters, which is shown in figure 4.3.

Weka Explorer	-	a x
Preprocess Classify Cluster Associate Select a	ttributes Visualize CPython Scripting	
Clusterer		
Choose KValid -init 0 -N 5 -A "weka.core.Euclidean	Distance -R first-last" -I 500 -validation 1 -cascade -minK 3 -maxK 10 -show-graph -S 10	
Cluster mode	Clusterer output	
Use training set Uses so clusters evaluation Nomi income  Generation Ignore attributes	native-country United-States U	
		<b>_</b>
Status		
ок	Log	×0 ×0

Figure 4.3. Elbow curve

#### **Distance Measures**

Deciding the distance measures is a critical step in clustering algorithms. Because choosing the right distance measure for the given datasets is the biggest challenge. The distance measures determine how the similarity of two elements (x, y) is calculated and it will influence the shape of the clusters. This is because some of the instances will be close to one another in a particular distance and they can also lie farther away according to other distance. Thus, choosing the right distance measure is purely based upon the nature and application of the dataset. Since the thesis work involves the imbalanced dataset distance measures has to be chosen appropriately to handle the imbalanced class.

#### **Euclidean Distance**

Euclidean Distance is the most common distance. This distance is most commonly used for dense or continuous datasets.

#### Algorithm:

Let  $X = \{x1, x2, x3, \dots, x_n\}$  be the set of instances in the data space.

Let  $V = \{v_1, v_2, v_3, \dots, v_n\}$  be the set of centers for calculating the distance.

1. Choose any value for 'c' cluster centers randomly.

2. Calculate the distance between each instance and cluster centers using the following Euclidean distance:

$$Dist_{XY} = \sqrt{\sum_{k=1}^{m} (X_{ik} - X_{jk})^2}$$
(1)

3. Each instance is assigned to the cluster center whose distance from the cluster center is a minimum of all the cluster centers.

4. New cluster center is calculated using:

$$V_i = \left(\frac{1}{C_i}\right) \sum_{i=1}^{C_i} x_i \tag{2}$$

Where 'ci' denotes the number of data points in the ith cluster.

- 5. The distance between each data point and new obtained cluster centers is recalculated.
- 6. If no data point was reassigned then stop, otherwise repeat steps from 3 to 5[18].



Figure 4.4. Euclidean Distance

### **Manhattan Distance**

Manhattan distance is the distance which can handle highly imbalanced class and categorical dataset. Since Euclidean distance suffers from the problem of "Curse of

dimensionality", they lead to overfitting. This problem is overcome by the Manhattan distance.

#### Algorithm:

Let  $X = \{x1, x2, x3, \dots, x_n\}$  be the set of instances in the data space.

Let  $V = \{v1, v2, v3, \dots, v_n\}$  be the set of centers for calculating the distance.

1. Choose any value for 'c' cluster centers randomly.

2. Calculate the distance between each instance and cluster centers using the following Manhattan distance:

$$Dist_{XY} = |X_{ik} - X_{jk}| \tag{3}$$

3. Each instance is assigned to the cluster center whose distance from the cluster center is a minimum of all the cluster centers.

4. New cluster center is calculated using:

$$V_i = \left(\frac{1}{C_i}\right) \sum_{1}^{c_i} x_i \tag{4}$$

Where 'ci' denotes the number of data points in ith cluster.

5. The distance between each data point and new obtained cluster centers is recalculated.

6. If no data point was reassigned then stop, otherwise repeat steps from 3 to 5[18].



*Figure 4.5. Manhattan distance* 

#### Combining the clusters from Euclidean and Manhattan distances

Clusters are generated from both the Euclidean distance and Manhattan distance. At this stage, customization is made in choosing the clusters among the distances to handle the imbalanced class and to derive more rules for them. Of the clusters that are generated from Euclidean distance, only the clusters that belong to the majority class are chosen. Similarly, of the clusters that are generated from the Manhattan distance, only the clusters that belong to the minority class are chosen. This is because of the coverage that is handled by both of the distances [18]. The combination of clusters from the Euclidean and Manhattan distance is explained in Figure 4.6.



Figure 4.6. Combining the clusters
#### **Applying Decision Tree algorithm**

The new dataset is formed by combining the majority class clusters and minority class clusters from the Euclidean and Manhattan distance. At this stage, if we take notice at the distribution of the instances in the minority class, we can see a difference in the weight of the instances. This is because some of the majority class instances which contribute to the distribution of minority class is present in the infrequent class. The increase in a number of instances in the minority class will lead to an increase in rules which are extracted from them.

In order to obtain rules from the minority class, initially rule-based classifiers like CART, PART, and RIPPER are used but Decision trees yielded better results when compared to others [11]. The performance measures in the decision trees are evaluated in terms of coverage of the infrequent class rules, average accuracy, precision, recall and ROC of the classifiers.

All the rules that are created using the Decision tree algorithm contain some duplicates. The performance of the model will be lowered if it contains duplicates and hence it must be removed in order to obtain better results. The obtained rules from the C45 algorithm are initially removed from duplicates and the accuracy and the coverage of each and every rule is represented.

- Rule Accuracy= Support/ Total number of rules in infrequent class
- Rule Coverage= Support/ Total number of infrequent instances in the whole dataset

#### **Performance Metrics**

In imbalanced class learning, the performance metrics used for the model selection would play a vital role. However, while working in these imbalanced domain datasets, accuracy is not an appropriate measure to evaluate model performance. This is because while handling imbalanced datasets, this performance metric suffers from the Accuracy Paradox whose results will be misleading [19]. There are several performance metrics available to handle the imbalanced classes and the widely used metrics are discussed here.

1. Precision: It means that the percentage of the results which are relevant

Precision= True Positive

True Positive + False Positive

2. Recall: It means that the percentage of the total relevant results that are correctly classified by the algorithm

Recall= True Positive

True Positive + False Negative

F-Measure: It combines precision and recall relative to a specific positive class
 F-Measure= 2 \* (Precision \* Recall)

(Precision + Recall)

4. ROC curve: This curve gives a comparison between two operating characteristics, they are the True Positive Rate (TPR) and False Positive Rate (FPR)

The above metrics are used in this thesis and there are some other metrics used to handle the imbalanced class, but they are not used in this work. They are as follows:

- 1. Kohonen's kappa: Using this metric to measure the performance will not necessarily increase how the model fits the data.
- 2. G-Measure: This metric is the geometric mean of precision and recall.

- Jaccard Index: This metric is actually used to measure the similarity between the classes.
- 4. Log-Loss: This metric involves accuracy as its measure, and it incorporates the idea of probabilistic confidence.
- 5. The Kolmogorov- Smirnov: This measure is not used in this thesis work because it measures the differences based on the distribution of instances in each class.
- Matthews Correlation Coefficient (MCC): This measure takes every cell in the confusion matrix, but it ended up in poor results for the given datasets.
   MCC = TP \* TN − FP \* FN / √ (TP +FP) \* (TP + FN) \* (TN + FP) \* (TN + FN)

Table 4.2 shows the results obtained from all the datasets after applying the decision tree algorithm with all the performance metrics used in this work and it shows that the ROC outperforms other metrics. Since accuracy is biased towards the majority class, the performance metrics has to be chosen carefully to understand the performance of the model. For this reason, several metrics are usually considered, which permits the polyhedral characteristics of the classification performance to be viewed from different points of views. The widely used metrics to evaluate the performance of the imbalanced classes are the Precision, Recall, ROC and Fmeasure. A major issue in the classification of class imbalanced datasets involves in the determination of the most suitable performance metrics.

Table 4.2Comparison of performance metrics.

Dataset	Accuracy	ROC	Fmeasure	Precision	Recall
Adult	86.73	0.907	0.709	0.811	0.630
Balance	76.34	0.650	0.000	0.000	0.000
Breast Cancer	84.40	0.719	0.553	0.756	0.436
Car	96.07	0.916	0.832	0.838	0.826
Mushroom	99.93	1.000	0.999	0.999	0.999
Retention	81.76	0.855	0.743	0.717	0.771
House Vote	97.40	0.986	0.973	0.965	0.965

#### CHAPTER V:

#### EXPERIMENTAL AND RESULTS

Here, the results that are obtained by using the undersampling techniques and clustering techniques and the rules extracted using the methodology as explained in earlier chapters are evaluated and analyzed.

#### Adult dataset

Table 5. 1 shows the class distribution of all the original datasets obtained from the UCI repository. This original dataset will contain some outliers and missing values which are handled in upcoming techniques. In these datasets, the distribution range of the datasets is uneven and hence they come under the imbalanced category of datasets. The datasets which are taken for analysis belongs to different categories. The Adult dataset and retention dataset have mixed characteristics, whereas the Balance, Breast cancer, Car, Mushroom and House vote have categorical characteristics.

Dataset Name	Original dataset		
	Majority	Minority	
Adult	37155	11687	
Balance	576	49	
Breast Cancer	201	85	
Car	1210	518	
Mushroom	4208	3916	
House Vote	267	168	
Retention	6402	2838	

Table 5.1 Original class distribution

Table 5. 2 shows the application of the preprocessing technique which is the nearest neighbor algorithm on the majority class to get rid of outliers and missing values and it is

implemented in WEKA using the Rseslib package. This package works on both numerical and nominal attributes and it implements fast neighbors searching algorithm which makes the classifier work for large data sets. Presence of outliers and missing values will affect the performance of the classifiers and hence they to be handled before applying any classification algorithm on the dataset. The class distribution obtained after applying this algorithm is shown briefly in Table 5.2

Dataset Name	Preprocessed Dataset			
	Majority	Minority		
Adult	34014	11208		
Balance	576	49		
Breast Cancer	139	63		
Car	1210	518		
Mushroom	3488	2156		
House Vote	244	153		
Retention	6189	2523		

After applying the nearest neighbor algorithm

Table 5.2

Table 5.3 shows the settings for the undersampling technique which is performed using Spread Sub Sample technique in WEKA. This package comes under the Supervised filter under the instances. This package produces a random subsample of a dataset. The original dataset must fit entirely in memory. This filter allows us to specify the maximum "spread" between the rarest and the most common classes. This filter when used in the batch mode, subsequent batches are not resampled. Several default settings in the filter has to be modified in order to balance the dataset using the undersampling technique. Several default values and experimental values for this filter is given in table 5.3

Table 5.3Class distribution settings

Options	Default Values	Experimental Values	What They Mean
adjust Weights	False	True	Adjusts instances weights To maintain total weight per class
debug	False	False	Displays the output
distributionSpread	0.0	1.0	Sets the value for distribution spread
doNotCheckCapabilities	False	True	Returns their capabilities in regard to their datasets
maxCount	0.0	0.0	Sets the value for maximum count
randomSeed	1	1	Suitable for displaying the output in gui

Table 5. 4 shows the application of the undersampling technique which makes the dataset balanced. After changing the default settings, mainly, changing the distribution spread from 0.0 to 1.0, the filter will perform the random undersampling technique and it derives the important instances from the majority class and balances them with the minority class. The adjust weights option will adjust the weights which are unevenly spread and makes them balanced so that the total weights per class is maintained. Also, individual instance weighting is not preserved. Now the balanced dataset is applied with the forthcoming techniques.

Dataset Name	Balanced dataset	
	Majority	Minority
Adult	11208	11208
Balance	49	49
Breast Cancer	63	63
Car	518	518
Mushroom	2156	2156
House Vote	153	153
Retention	2523	2523

Table 5.4After applying the undersampling technique (Spread Subsample)

Table 5. 5 shows the application of the K-Means algorithm which separates the whole dataset into clusters. Instead of the default distance measure, which is the Euclidean distance, the thesis focuses on the alternative distance measure. The alternative distance is the Manhattan distance. By changing the default distance measure in the settings of the K-Means algorithm, in the clustering section in WEKA, this experiment can be done. The number of clusters are decided based on the Elbow curve and different clusters are produced. Both the Euclidean and Manhattan distance will produce majority and minority clusters and they are divided separately and are shown in table 5.5

Dataset name	Clustered dataset			
	Euclidean Majority cluster	Euclidean Minority cluster	Manhattan Majority cluster	Manhattan Minority cluster
Adult	8288	15086	7750	15624
Balance	32	17	24	25
Breast Cancer	57	46	52	58
Car	222	162	153	198
Mushroom	2034	1989	2027	2009
House Vote	97	56	45	108
Retention	2316	1879	2267	2348

Table 5.5After applying the K-Means algorithm to the balanced dataset

Table 5. 6 shows the class distribution of the dataset after combining the clusters from Euclidean and Manhattan distance. The above table shows the types of clusters which are produced from the K-Means algorithm. From the table 5.5, it is evident that the Euclidean distance has a greater number of majority class clusters and the Manhattan distance has a greater number of minority class clusters. The difference in the distribution of clusters is mainly due to the coverage of different distances. Instead, of working in the default clustered dataset, we produce a new dataset, which combines the majority class cluster from Euclidean distance and minority class cluster from the Manhattan distance. Since our ultimate aim is to produce a greater number of rules for the minority class, increase in the number of instances will automatically increase the number of rules.

Dataset name	Combination of clusters		
	Majority	Minority	
Adult	8288	15624	
Balance	32	25	
Breast Cancer	57	58	
Car	222	198	
Mushroom	2034	2009	
House Vote	97	108	
Retention	2316	2348	

Table 5.6New dataset after a combination of clusters

Table 5. 7 shows the number of rules obtained from the decision tree algorithm to the combined new dataset. This decision tree algorithm is the implementation of ID3. It divides the decision tree based on the information gain. This information gain decides, in each tree node, which variable fits better in terms of target variable prediction. With the help of greedy approach, the decision tree builds trees which helps in rule extraction. The rules that are shown in Table 5.7 are the ones which are obtained from the pruned tree, which are smaller and less complex. These rules are created from each path from the root to the leaf node. These leaf nodes hold the class prediction, forming the rule consequent.

Dataset name	Decision tree			
	Majority	Minority		
Adult	1174	478		
Balance	32	12		
Breast Cancer	52	47		
Car	15	49		
Mushroom	27	16		
House Vote	39	32		
Retention	181	137		

Table 5.7Decision tree for the clustered dataset

Although decision tree algorithm is chosen because of its entropy and information gain, several other rule classification algorithms like PART, JRIP and Random Tree algorithms are taken into account to make a comparative study among all the algorithms. The experimental setup for the rule classification algorithms is shown in table 5.8. In order to derive rules from the decision tree certain default values has to be changed which helps in displaying the pruned decision tree with rules. By making this experimental setup we develop decision trees which gives us useful rules. Some of the classification rules found to be undesirable for the users, resulting in scalability and efficiency problem. They can be handled by pruning the trees.

Options **Default Values** Experimental What They Values Mean False True Displays the Debug output doNotCheckCapabilities False True Returns their capabilities in regard to their datasets Seed 1 1 Seed for random data shuffling Unpruned False False Returns the value of unpruned tree 3 3 Numfolds Sets number of folds for reduced error pruning Usepruning True True Returns a pruned tree

Table 5.8Experimental setup for Rule classification algorithms

Table 5.9 shows the output obtained from several rule classifiers from the classification algorithm which shows the number of rules obtained from each algorithm and their respective coverage. The ultimate aim of the thesis is to extract more rules for the minority class. There are several rule classifiers available in WEKA to extract rules. In order to find which algorithm performs best we performed the rule generation with all the rule classifiers like PART, C45, RANDOM TREE and JRIP. We have separated the overall rule generation into the majority and minority class. After separating, the coverage of the rules is obtained which helps in determining which classifier performs best. From Table 5.9, it is evident that the Random Tree classifier produces a greater number of rules, but it has very low coverage and hence we ended up in using C45 algorithm which produces more rules with better coverage.

Table 5.9

Displaying the output from several rule classifiers

Dataset Name	Overall Class Rule Generation Minority Class Rule Generation					
	Classifier	Number Of Rules	Classifier	Number Of Rules	Coverage	
Adult dataset	Part	1002	Part	445	0.4441	
	C45	1174	C45	478	0.4789	
	Random Tree	26639	Random Tree	4327	0.16243	
	JRip	12	JRip	11	0.9166	
	Part	27	Part	11	0.407	
Balance dataset	C45	56	C45	12	0.503	
	Random Tree	246	Random Tree	122	0.495	
	JRip	10	JRip	-	-	
	Part	26	Part	14	0.5384	
Breast Cancer	C45	59	C45	19	0.6387	
	Random Tree	220	Random Tree	47	0.2136	
	JRip	4	JRip	3	0.75	
Car	Part	37	Part	31	0.550	
	C45	89	C45	49	0.837	
	Random Tree	199	Random Tree	81	0.407	
	JRip	18	JRip	16	0.888	
Mushroom	Part	26	Part	16	0.615	
	C45	100	C45	56	0.933	
	Random Tree	216	Random Tree	144	0.666	
	JRip	14	JRip	13	0.723	
Retention	Part	345	Part	165	0.478	
	C45	180	C45	52	0.857	
	Random Tree	9566	Random Tree	1472	0.153	
	JRip	7	JRip	6	0.288	
House Vote	Part	7	Part	3	0.428	
	C45	9	C45	32	0.666	
	Random Tree	109	Random Tree	47	0.293	
	JRip	3	JRip	2	0.222	

Table 5.10 shows the dataset's characteristics with the help of metrics like Accuracy, Precision, Recall, and F-Measure. Although the goal of the thesis is to extract more rules for the minority class, the next important part of the thesis is to measure the performance of the classifier. The classifier's performance is evaluated in terms of Performance metrics. For making a comparative study we have evaluated the performance metrics for all the four classifiers which are discussed in Table 5.9. From the table it is evident that the C45 algorithm outperforms other classifiers in terms of all the performance metrics. The baseline accuracy shown in Table 5.10 is obtained from the ZeroR algorithm which is the standard algorithm for obtaining the accuracy of the model.

Table 5.10

Displaying the dataset's characteristics

Dataset	Classification		Accuracy	Fmeasure	Precision	Recall
	Method					
	Baseline	Classifier				
	Accuracy					
	%					
Adult	75.92	Part	86.73	0.866	0.852	0.880
		C45	88.98	0.890	0.868	0.912
		Random	99.88	0.999	0.999	0.999
		Tree				
		JRip	82.89	0.829	0.807	0.852
Balance	46.08	Part	76.34	0.659	0.739	0.610
		C45	80.64	0.767	0.759	0.797
		Random	100	1.000	1.000	1.000
		Tree				
		JRip	66.66	0.504	0.545	0.469
Breast	70.28	Part	84.39	0.830	0.822	0.838
Cancer						
		C45	87.61	0.862	0.877	0.899
		Random	97.70	0.975	0.980	0.970
		Tree				
		JRip	69.26	0.613	0.716	0.535
Car	70.02	Part	96.45	0.961	0.960	0.964
		C45	97.19	0.978	0.984	0.972
		Random	100	1.000	1.000	1.000
		Tree				
		JRip	90.75	0.934	0.887	0.988
Mushroom	51.79	Part	99.93	0.999	0.999	0.999
		C45	99.97	1.000	1.000	0.999
		Random	100	1.000	1.000	1.000
		Tree				
		JRip	99.87	0.998	0.997	1.000
Retention	69.29	Part	86.41	0.817	0.812	0.822
		C45	87.98	0.880	0.886	0.874
		Random	99.29	0.993	0.996	0.990
		Tree				
		JRip	80.95	0.810	0.800	0.820
House Vote	61.23	Part	97.40	0.973	0.982	0.965
		C45	97.98	0.979	0.982	0.977
		Random	99.71	0.997	1.000	0.994
		Tree				
		JRip	97.11	0.970	0.982	0.959

Table 5.11 shows the number of rules that are obtained from the proposed algorithm which are obtained from sampling, clustering and applying decision tree. The original dataset is first applied with the undersampling technique and the balanced dataset is further applied with the K-Means clustering algorithm with two different distances: Euclidean and Manhattan and finally decision tree algorithm is applied to the new dataset after combining the clusters from the two distances. The table 5.11 shows the different metrics that are applied to the minority rules in order to evaluate the performance of the proposed methodology. The average accuracy and coverage of the proposed work is obtained with the help of the overall rules and infrequent rules and the range where the number of rules falls under each category is shown in table 5.11.

Table 5.11

The	proposed	methodo	logy for	all the	datasets
1110	proposed	memouou	$\mathcal{O}_{\mathcal{S}}$ y $\mathcal{O}_{\mathcal{I}}$		unuscis

Metrics	Adult	Balance	Breast Cancer	Car	Housevote	Mushroom	Retention
Total number of rules	506	45	62	65	71	27	345
Total number of infrequent rules	137	12	47	49	32	16	52
Average Accuracy	0.9155	0.1538	0.8186	0.8449	0.8670	0.7777	0.8309
Range:90- 100	86	-	5	10	16	14	12
Range:80- 89	32	-	2	-	3	-	4
Range:70- 79	5	-	2	5	-	-	5
Range:60- 69	-	-	3	-	-	-	-
Range:50- 59	-	-	-	32	5	-	7
Coverage%	47.90	26.89	22	51.12	19	57.68	63.25

Table 5.12 shows the number of rules obtained from the previously published works which involve decision tree and clustering algorithm. This methodology uses the K-Means clustering algorithm to obtain clusters and finally decision tree algorithm is applied on it to extract rules for the minority class. The table 5.12 shows the different metrics that are applied to the minority rules in order to evaluate the performance of the proposed methodology. The average accuracy and coverage of the proposed work is obtained with the help of the overall rules and infrequent rules and the range where the number of rules falls under each category is shown in table 5.12. By comparing Table 5.12 with Table 5.11, it is evident that the proposed methodology outperforms this method in case of Adult, Balance, Breast Cancer, Car, Housevote and Mushroom datasets.

Table 5.12Clustering and Decision tree algorithm for all the dataset

Metrics	Adult	Balance	Breast Cancer	Car	Housevote	Mushroom	Retention
Total number of rules	731	69	30	131	-	-	158
Total number of infrequent rules	26	-	17	-	-	-	16
Average Accuracy	0.940	-	0.768	-	-	-	0.8
Range:90-100	15	-	-	-	-	-	10
Range:80-89	11	-	13	-	-	-	6
Range:70-79	-	-	4	-	-	-	-
Range:60-69	-	-	-	-	-	-	-
Range:50-59							
Coverage%	22.08	-	0.720	-	-	-	69.00

The algorithm which was previously published involves association mining algorithm which has to be applied with certain settings which are explained in table 5.13. In order to make a comparison with the previously published works on rule extraction, we perform association mining algorithms with several settings. The default values have to be changed to certain experimental values which helps in extracting more rules from the Apriori algorithm. By making these changes the rules are generated by leaving the rules which does not contribute more to the model's performance. Certain range is set which helps in extracting rules which has higher accuracy which is specified in the minMetric options in Table 5.13.

Table 5.13 *Apriori Settings* 

Options	Default Values	Experimental Values	What They Mean
Car	False	True	Generates rules with the class attribute
Delta	0.05	0.1	Iteratively decrease support by this factor
minMetric	0.9	0.85	It will consider rules with accuracy of 0.85 or higher
numRules	10	100	Number of rules to generate
outputItemSets	False	True	Item sets are shown in output
UpperBoundMinsupport	1.0	1.0	The highest value of minimum support, the process starts and iteratively decreases until lower boundary

Table 5.14 shows the number of rules obtained from the previously published works which involve decision tree and association mining algorithm. After making the changes from default value to experimental value, a greater number of rules are extracted for the datasets which follows a decision tree and association mining algorithm. The table 5.14 shows the different metrics that are applied to the minority rules in order to evaluate the performance of the proposed methodology. The average accuracy and coverage of the proposed work is obtained with the help of the overall rules and infrequent rules and the range where the number of rules falls under each category is shown in table 5.14. From Table 5.14, it is evident that the proposed methodology outperforms Table 5.14 in case of Adult, Balance, Car, Mushroom and Retention datasets.

Table 5.14

Metrics	Adult	Balance	Breast Cancer	Car	Housevote	Mushroom	Retention
Total number of rules	696	33	202	-	171	-	496
Total number of infrequent rules	125	-	165	-	100	-	200
Average Accuracy	0.735	-	0.766	-	0.921	-	0.710
Range:90-100	31	-	19	-	100	-	100
Range:80-89	13	-	117	-	-	-	6
Range:70-79	16	-	44	-	-	-	94
Range:60-69	-	-	-	-	-	-	-
Range:50-59							
Coverage%	42.70	-	24.79	-	29.23	-	52.73

Decision Tree and Association mining for all the datasets

Figure 5.1 shows the number of extracted from the proposed methodology. This also gives us a comparison of the number of rules that are extracted using the previous methodology.



Figure 5.1. Comparison of the rule extraction

Table 5.15 shows the top rules from the proposed methodology which has the highest accuracy and highest coverage from the obtained rules. There are several metrics which is used to evaluate the performance of the rules extracted. Of them, few of the metrics like support, accuracy and coverage are used to evaluate the performance. The formula for calculating the support, accuracy and coverage of the rules are shown in Chapter IV under the applying the decision tree algorithm section. Of the several rules that are obtained from the proposed algorithm, only the top two rules from each dataset is shown in Table 5.15 as they have the maximum accuracy and coverage.

Table 5.15 *Displaying Top Rules* 

Rule	Dataset	Support	Accu-	Coverage
Marital status=Married-civ-spouse AND educational- num>=9 AND capital gain>=4386 AND age>=36: Class=>50K	Adult	1490	0.9289	0.12749
Marital status = Married civ spouse AND capital gain>5060 AND age<=60: Class=>50k	Adult	1575	0.9974	0.13476
Right wgt=1 AND Left wgt=1: Class=B	Balance	5	0.2000	0.1020
Right wgt=4 AND Left dis=4: Class=B	Balance	5	0.200	0.1020
Inv nodes= 0-2 AND tumor size= 20-24 AND menopause= premeno AND irradiant = yes: Class= recurrence events	Breast Cancer	2	1.000	0.023529
Inv nodes= 0-2 AND tumor size= 20-24 AND menopause= It40: Class=recurrence events	Breast Cancer	4	1.000	0.04705
Safety=med AND person=more AND maint=med AND buying=med: Class=acc	Car	11	0.9166	0.0286
Safety=med AND person=4 AND maint=low AND buying=low AND lug-boot=small: Class=acc	Car	4	1.000	0.0104
Bruises= f AND gill spacing = c AND ring type=f: Class= P	Mush- room	1296	1.000	0.3309
Bruises= f AND gill spacing = c AND ring type=e AND stalk-surface above ring=k: Class= P	Mush- room	896	1.000	0.2288
Adoption of the budget resolution=n AND el-Salvador- aid=y AND immigration=n AND physician-fee- freeze=y AND synfuels-corporation-cutback=n AND export-administration-act-south-africa=n: republican	House vote	21	1.000	0.125
Adoption of the budget resolution=n AND el-Salvador- aid=y AND immigration=y AND education- spending=y AND synfuels-corporation cutback=y AND handicapped-infants=n: republican	House vote	7	0.875	0.0416
Grant_y = N AND loan_y = N AND Ethnic = WH AND fall attempt range = F AND Parent-char = P AND Age numeric <= 20 AND percentile range = Next15 AND GPA-char = Medium: class=D	Student Retention	9	0.9	0.0031
Grant_y = N AND loan_y = N AND Age numeric > 19 AND Spring attempt range = L: class=D	Student Retention	114	0.9344	0.0507

# CHAPTER VI: CONCLUSION AND FUTURE WORK

#### Conclusion

This thesis intends to generate more rules for the minority class. Generating more rules gives us more accurate predictions for the minority class. These rules will explain more about how fraudulent cases occur, what are the ways that could lead to certain diseases and many other rare class problems which are a huge threat. The thesis work focuses more on the rule generation because rules are suggestive, easy to generate and easy to interpret. Since the proposed methodology uses Decision trees to generate rules, the attributes which contribute more to the majority class become evident, as the decision trees are generated based on the information gain. The proposed methodology follows a preprocessing technique which cleans the dataset without any anomalies. Following that, the dataset is applied with an undersampling algorithm which samples the majority instances and the majority instances are kept intact. The balanced dataset is applied with cluster analysis as they would find the hidden relationships between each other to group a set of instances into clusters. The core idea of the thesis relies on the two different distance measures: Euclidean and Manhattan. These distances are involved in creating the clusters due to their measure of coverage. After combining the clusters from their respective distances, classification algorithm (C45) is applied to the training data which builds a decision tree and generates rules respectively. The proposed methodology was able to generate more rules for the Adult, Balance, Car and Mushroom datasets. These datasets belong to the categorical characteristics. The standard K-Means algorithm isn't directly applicable to categorical data as the sample space for the categorical data is discrete and doesn't have a natural origin and hence the alternative distance measure. Manhattan

distance, is used to handle the dataset [18]. This helps us in the future prediction of the majority class.

#### **Future Work**

The number of datasets included in this study is minimal. Hence, a greater number of datasets have to be included in order to evaluate the performance of the proposed methodology.

Some of the datasets involved in this study didn't produce a greater number of rules for the majority class yet they produced rules which have better accuracy and coverage. Hence, customization must be done in the proposed methodology in order to generate more rules.

Datasets of different characteristics must be involved in order to learn about the structure of the dataset and apply the proposed methodology.

#### **Contribution to Research Community**

Despite numerous algorithms and re-sampling methods being used in the last few decades to handle imbalanced classes, there is no consistent winning strategy for all kinds of datasets. This requires a special attention that needs to be paid to the data in the datasets. Mining these kinds of datasets can only be improved by the algorithms which are tailored to the data characteristics; henceforth, it is important to do an exploratory analysis on the datasets which is performed using the K-Means algorithm. By observing the characteristics of the datasets, it is evident that the default Euclidean distance alone will not contribute in handling the missing datasets. Instead, the Manhattan distance performed better.

#### REFERENCES

[1] Neelam Rout, "Handling Imbalanced Datasets-A survey", Proceedings of the 18<sup>th</sup> International Proceedings on Advances in Soft Computing, Intelligent systems and applications, Advances in Intelligent systems and computing 628 doi.org/10.1007/978-981-10-5272-9\_39

[2] Vaishali Ganganwar, "An overview of classification algorithms for imbalanced datasets", *International Journal of Emerging Technology and Advanced Engineering*, Volume 2, April 2012, ISSN 2250-2459, Issue 4.

[3] K. Usha Rani, G. Nagaramadevi, D. Lavanya, "Performance of synthetic majority oversampling technique on Imbalanced Breast cancer data", 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2016, pp. 1623-1627.

[4] D. Shi, J. Guan and J. Zurada, "Cost-Sensitive Learning for Imbalanced Bad Debt Datasets in Healthcare Industry," *2015 Asia-Pacific Conference on Computer Aided System Engineering*, Quito, 2015, pp. 30-35.

[5] T. J. Lakshmi and C. S. R. Prasad, "A study on classifying imbalanced datasets," 2014
First International Conference on Networks & Soft Computing (ICNSC2014), Guntur,
2014, pp. 141-145. Doi: 10.1109/CNSC.2014.6906652

[6] I. Martin-Diaz, D. Morinigo-Sotelo, O. Duque-Perez, and R. de J. Romero-Troncoso, "Early Fault Detection in Induction Motors Using AdaBoost with Imbalanced Small Data and Optimized Sampling," in *IEEE Transactions on Industry Applications*, vol. 53, no. 3, pp. 3066-3075, May-June 2017. [7] J. Song, X. Huang, S. Qin and Q. Song, "A bi-directional sampling based on K-means method for imbalance text classification," *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, Okayama, 2016, pp. 1-5.

[8] T. E. Tallo and A. Musdholifah, "The Implementation of Genetic Algorithm in Smote (Synthetic Majority Oversampling Technique) for Handling Imbalanced Dataset Problem," 2018 4th International Conference on Science and Technology (ICST), Yogyakarta, 2018, pp. 1-4.

[9] A. Hanskunatai, "A New Hybrid Sampling Approach for Classification of Imbalanced Datasets," 2018 3rd International Conference on Computer and Communication Systems (ICCCS), Nagoya, 2018, pp. 67-71.

[10] Y. Pristyanto, I. Pratama and A. F. Nugraha, "Data level approach for imbalanced class handling on educational data mining multiclass classification," *2018 International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, 2018, pp. 310-314.

[11] S. Datta and S. Mengel, "Elastic Multi-Stage Decision Rules for Infrequent Class,"
 2016 3rd International Conference on Soft Computing & Machine Intelligence (ISCMI),
 Dubai, 2016, pp. 110-114.doi: 10.1109/ISCMI.2016.20

[12] S. Datta and S. Mengel," Adaptable multi-phase rules over infrequent class", *Proceedings of Soft Computing*, 2018, 22:6067

[13] A. Soltani and M. Akbarzadeh-T., "Confabulation-Inspired Association Rule Mining for Rare and Frequent Itemsets," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2053-2064, Nov. 2014. [14] A. Agrawal, H. L. Viktor and E. Paquet, "SCUT: Multi-class imbalanced data classification using SMOTE and cluster-based undersampling," 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), Lisbon, 2015, pp. 226-234.

[15] S. Mishra," Handling Imbalanced data: SMOTE vs Under-Sampling", 2017 International Research Journal of Engineering and Technology, pp.317-320.

[16] C.N.S Kumar, K.N Rao, A. Govardhan, N. Sandhya, "Subset K-Means Approach for Handling Imbalanced Distribution data", In: Satapathy S., Govardhan A., Raju K., Mandal (eds), *Emerging ICT for Bridging the Future-Proceedings of the 49<sup>th</sup> Annual Convention of the Computer Society of India, CSI volume 2, Advances in Intelligent Systems and Computing, vol 338,2015, Springer, Cham* 

[17] M. Milan-Giraldo, V. Garcia, J.S. Sanchez, "Prototype Selection in Imbalanced data for Dissimilarity Representation- A preliminary study", ICPRAM 2012- Proceedings of 1<sup>st</sup> International Conference on Pattern Recognition Applications and Methods, 2012, pp.242-247.

[18] D.J. Bora, A.K. Gupta, "Effect of Distance Measures on the performance of the K-Means algorithm: An experimental study in MATLAB". *International Journal of Computer science and Information technologies, vol 5(2), 2014, 2501-2506.* 

[19] N.V. Chawla, "Data Mining for Imbalanced datasets: An overview", In Maimon O., Rokach L. (eds), Data Mining and Knowledge Discovery Handbook, Springer, Boston, MA, 2009.

 [20] S. Zhang, X. Wu and M. Zhu, "Efficient missing data imputation for supervised learning", 9<sup>th</sup> IEEE International Conference on Cognitive Informatics (ICCI), Beijing, 2010, pp. 672-679

### APPENDIX A:

### OTHER EXPERIMENTAL RESULTS

	- D >
Preprocess Classity Cluster Associate Select attributes visualize CPytrion Scripting	
Open file Open URL Open DB Ger	enerate Undo Edit Save
Filter	
Choose None	Apply Stop
Current relation	Selected attribute
Relation: adult dataset (1) Attributes: 15 Instances: 48842 Sum of weights: 48842	Name: income Type: Nominal Missing: 0(0%) Distinct: 2 Unique: 0(0%)
Attributes	No. Label Count Weight
All None Invert Pattern	1 ≪E0K 37155 37155 0 2 ≻50K 11687 11687.0
1         ape           1         ape           2         word           3         adoction           5         adoction           6         marti-status           7         occupation           8         retilineship	Class: income (Nom)
9 race 9 race 10 gender 11 capita-gain 12 capita-loss 13 hours-per-week 14 native-country 15 income	37185
Remove	11627
Status	
ОК	Log 🛷

#### **Adult Dataset**

Figure A.1. Original Class Distribution

Preprocess Classify Cluster Associate Select attributes Visualize CPython Scripting				
Open file Open URL Open DB Gen	erate Ur	1do E	idit Si	ive
ilter				
Choose None			Ap	ply Stop
urrent relation	Selected attribute			
Relation: adult test Attributes: 15 Instances: 48842 Sum of weights: 48842	Name: income Missing: 0 (0%)	Distinct: 2	Type: Nominal Unique: 0 (0%)	
ttributes	No. Label	Count	Weight	
All None Invert Pattern	1 <=50K 2 >50K	37155 11687	37155.0 11687.0	
No.         Name           1         ape           2         workclass           3         hhwgt           4         education           5         education-tum           6         martial-status				
7       occupation         8       relationship         9       race         10       gender         11       capital-gain         12       capital-Joss         13       hours-per-week         14       native-country         15       income	Class: income (Nom)			Visualize /
Ramova		j	1687	
atus				-

Figure A.2. After Applying Nearest Neighbor Algorithm

😮 Weka Explorer	- 0 ×
Preprocess Classify Cluster Associate Select attributes Visualize CPython Scripting	
Open file Open URL Open DB Get	inerate Undo Edit Save
Filter	
Choose SpreadSubsample - M 1.0 - X 0.0 - S 1	Apply Stop
Current relation	Selected attribute
Relation: adult test-weka filters supervised instance.SpreadSubsample-M0.0-X0 Attributes: 15 Instances: 23374 Sum of weights: 23374	Name: income         Type: Nominal           Missing: 0 (0%)         Distinct: 2         Unique: 0 (0%)
Attributes	No. Label Count Weight
All None Invert Pattern	1 <=50K 11687 11687.0 2 >50K 11687 11687.0
No.         Name           1         age           2         workdass           3         hnwd           4         eductions-turm           5         eductions-turm           6         outrals-table           7         occupation           8         relationships           9         race           10         gender           11         capital-sain           12         capital-sain           13         hours-per-werk           14         native-country           15         income	Class: income (Nom) Visualize All
OK .	Log 🛷 X

*Figure A.3. After Applying Undersampling technique (Spread Sub Sample)* 



Figure A.4. After applying K-Means algorithm to the balanced dataset

	Associate Select attributes '	Visualize CPython Scripting			
Open file	Open URL	Open DB	Generate	Undo	dit Save
iter					
Choose SpreadSubsample -M	1.0 -X 0.0 -S 1				Apply Stop
urrent relation			Selected attribute		
Relation: adult test Instances: 23912		Attributes: Sum of weights:	15 Name: class 23912 Missing: 0 (0%)	Distinct 2	Type: Nominal Unique: 0 (0%)
ttributes			No. Label	Count	Weight
All	None	Invert Pattern	1 <=50K 2 >50K	122/1 11641	12271.0
2 workclass 3 fnlwgt 4 education 5 educational-num 6 Marital status 7 Occupation			Class: class (Nom)		Visualize
8 realionship 9 race 10 sex 11 capital-gain 12 capital-gain 13 hours per-week 14 halve county 15 clips	Remove				1641

Figure A.5. New Dataset after the combination of clusters



Figure A.6. Decision tree for the clustered dataset

Preprocess     Classify     Cluster     Associate     Select attributes       Open IIe     Open URL     Open DB     Generate     Undo       Filer     Choose     Name     Name     Itable       Current relation     Select attributes     Name     Select attributes       Attributes     Sum of weights: 286     Distinct: 2     Name       All     None     Invert     Pattern       No.     Name     Name     201	Edit	Save
Open IIe         Open URL         Open DB         Generate         Undo           Filer         Choose None         Selected attribute         Name         Selected attribute           Current relation         Attributes:10         Sum of weights: 286         Name         Selected attribute           Attributes         Sum of weights: 286         Name         Courrence-events         201           All         None         Invert         Pattern         85	Edit	Save
Selected attributes           Choose None           Current relation           Relation: breast (1) Instances: 286           Attributes:           Attributes           All           None           No.	2 Type: Nomina 2 Unique: 0 (%) nt Weipht 201,0 85,0	Apply Stop
Choose         None           Current relation         Selected attribute           Relation: breast (1) Instances: 286         Attributes: 10 Sum of weights: 286         Name: Class Missing: 0 (0%)         Distinct: 2 No.           All         None         Invent         Pattern           No.         Name         85	2 Type: Nomina 2 Unique: 0 (%) nt Weiph 201,9 85,0	Apply Stop
Current relation         Selected attribute           Relation: breast (1) Instances: 286         Attributes: 10 Sum of weights: 286         Name: Class Missing: 0 (0%)         Distinct: 2 Distinct: 2 No.           Attributes         Attributes         Name: Class No.         Label         Court           Attributes         Pattern         201         2         recurrence-events         201           No.         Name         Pattern         0         1         0         0	2 Type: Nomina Unique: 0 (0%) nt Weight 2010 85.0	1
Relation: breast (1) Instances: 286         Attributes: 10 Sum of weights: 286         Name: class Missing: 0 (0%)         Disting: 2 Disting: 2           Attributes         No.         None         Increase         Court           All         None         Pattern         201           No.         Name         85	Type: Nomina           2         Unique: 0 (0%)           int         Weight           201.0         85.0	al
Attributes         No.         Label         Could           All         None         Invert         Pattern         21         221	nt Weight 201.0 85.0	
All None Invert Pattern	201.0 85.0	
1         dass           2         age           3         menopase           4         tumoristas           5         ink-nodes           6         node-caps		
7     deg-malig       8     breast       9     breast-quad       10     irradiat         Remove         Status	85	Visualize All

**Breast Cancer Dataset** 

Figure A.7. Original Class Distribution

Weka Explorer     Prannonesse     Classify     Chuster     Associate     Selent attributes     Visualize     Protono Scription	- a ×
Open Ile         Open URL         Open DB         Gene	rrate
Choose None	Apply Stop
Current relation	Selected attribute
Relation: breast test Attributes: 10 Instances: 286 Sum of weights: 286	Name: class Type: Nominal Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)
All     None     Invert     Pattern       No.     Name     1     dass       2     age     3     menopuse       4     Unor-size     5     invoides       5     invoides     6     00de-caps       7     broath     9     broash-puad       10     irradiat     10     irradiat	1         no-recurrence-wents         208         208.0           2         recurrence-events         78         78.0             Class: class (Nom) <ul> <li>Visualize A</li> <li>209</li> </ul> 209         74
Remove	
ок	Log 🛷

Figure A.8. After applying the nearest neighbor algorithm

Wekk Explorer           Preprocess         Classify         Cluster         Associate         Select attributes         Visualize         CPython Scripting	-
Open file Open URL Open DB Gen	erate
ilter Choose SpreadSubsample -M 1.0 -X 0.0 -S 1	Apply Stop
current relation	Selected attribute
Relation: breast test-weka filters.supervised.instance.SpreadSubsample-M1.0-X0 Attributes: 10 Instances: 156 Sum of weights: 156	Name: dass Type: Nominal Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)
Attributes	No. Label Count Weight
All     None     Invert     Patern       No.     Name       1     class       2     age       3     menopause       4     bimosize       5     onderease       6     onderease	1 no-recurrence-events 78 78.0 2 recurrence-events 78 78.0
7 Geomaig 8 Forest 9 Freast-quad 10 Frradiat	Class class (Nom) Visualize #
Status	
ок	Log 🦽

Figure A.9. After applying the undersampling technique (Spread Sub Sample)

Weka Explo	orer Y	Y	r	Y	Y	Ŷ			- 0 ×
Preprocess	Classify	Cluster	Associate	Select attribu	ites Visualize	CPython Sc	cripting		
Clusterer									
Choose	SimpleK	Means -init	0 -max-ca	G Weka Cluste	erer Visualize: 16:	02:54 - Simple	KMeans (breast te	test-weka.filters.supervised.instance.SpreadSubsample D X 10	
Cluster mode				X: Instance_n	umber (Num)		*	Y: class (Nom)	
🔾 Use tra	ining set			Colour: Cluste	er (Nom)		*	Select Instance	
🔾 Supplie	ed test set		Set	Reset	Clear	Open	Save	Jitter	
O Percen	tage split		%	Plot: breast tes	t-weka.filters.s	upervised.ins	tance. Spread St	Subsample-M1.0-X0.0-S1_clustered	
(Nom)	s to cluster	rs evaluatio	n	۲J			• • × × × •		
Store c	lusters for v	visualizatior	1	c		*			
			_	ro			0.0		
	Ign	iore attribute	9S	er ne				· · · · · · · · · · · · · · · · · · ·	
s	tart		Stop	ec eu					
Result list (rij	ht-click fo	or options)		er					
16:02:54 -	SimpleKMe	eans		en					
				te s					_
				e v v ×	× × ×	,0 0 × ,	•	8 ¥	
				e n o was	× *** **		• •		
				0		<u> </u>	77.5	155	
				Class colour					
				class colodf					
							e	cluster0 cluster1	
L					•				7.
Status									
ок									

Figure A.10. After applying K-means algorithm to the balanced dataset

Weka Explorer     Preprocess Classify Cluster Associate Select attributes Visualize CPython Scripting	- a ×
Open file Open URL Open DB Gene	rate Undo Edit Save
ilter	
Choose SpreadSubsample -M 1.0 -X 0.0 -S 1	Apply Stop
Current relation	Selected attribute
Relation: breast test Attributes: 10 Instances: 218 Sum of weights: 218	Name: Class Type: Nominal Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)
Attributes	No. Label Count Weight
	1 no-recurrence-events 119 119.0 2 recurrence-events 99 99.0
1     Class       2     ago       3     menopause       4     tumorstan       5     min-rotes       6     node-caps       7     deg-malign       8     broast       9     broast-quad       10     irradiant	Class: Class (Nom) Visualize /
Status	
OK	Log 🛷

Figure A.11. New dataset after a combination of clusters



Figure A.12. Decision tree for the clustered dataset

### **Balance Dataset**

Weka Explo	orer											22 <del>-</del> 22	
Preprocess	Classify	Cluster	Associate	Select attributes	Visualize CPython S	Bcripting							
0	pen file		Ope	n URL	Open DB		Gen	ierate		Undo	Edit	Sav	Ð
Filter													
Choose	SpreadSu	bsample	-M 1.0 -X 0.0 -	S1								Appl	y Stop
Current relat	ion							Selected a	ttribute				
Relation: Instances:	Balance da 625	ataset				Attri Sum of we	ibutes: 5 eights: 625	Name Missing	Class Nam 0 (0%)	ne Distinct 3	Type: Unique:	Nominal 0 (0%)	
Attributes								No.	Label	Cour	nt We	eight	
								1	в	49	49	1.0	
	All		None		Invert	Patter	m	2	2 R	288	28	8.0	
2 ( 3 ( 4 ( 5 (	Left Dist Right We Right Dist	ance eight stance					-	Class: Cla	ss Name (No	om)		•	Visualize All
				Remove				4					
Status										80			
ок												Log	×0

Figure A.13. Original class distribution

Prennocess Classify Cluster	Associate   Select attributes	Visualize CPvthon Scripting				
Open file	Open URL	Open DB	Generate		Undo	dit. Save
llar		( <u> </u>				
Choose SpreadSubsample	M 1.0 -X 0.0 -S 1					Apply Stop
urrent relation			Selec	ted attribute		
Relation: Balance test Instances: 625		Attr Sum of w	ibutes: 5 Mi reights: 625 Mi	lame: Class Name ssing: 0 (0%)	Distinct 3	Type: Nominal Unique: 0 (0%)
tributes			No	Label	Count	Weight
	Nono	Imont Ratio		1 B 2 R	49 288	49.0 288.0
No. Name 1 Class Name 2 Left Weight						
4 Right Weight 5 Right Distance				(Class Name (Nom)		• Meunite
	Remove		4	, crass realine (roun)	20	200

Figure A.14. After applying the nearest neighbor algorithm

Preprocess Classify Cluster Associate Select attributes Visualize CPython Scripting	
Open file Open URL Open DB Ger	anerate Undo Edit Save
ter	
Choose SpreadSubsample -M 1.0 -X 0.0 -S 1	Apply St
Irrent relation	Selected attribute
Relation: Balance test-weka filters.supervised.instance.SpreadSubsample-M1.0-X Attributes: 5 Instances: 147 Sum of weights: 147	Name:         Class Name         Type:         Nominal           Missing:         0 (0%)         Distinct:         3         Unique:         0 (%)
tributes	No. Label Count Weight
All None Invert Pattern	1 B 49 480 2 R 49 480 3 L 49 480
No.         Name           1         Class Name           2         Left Weight           3         Left Distance           4         Right Weight           5         Petro Distance	
	Class: Class Name (Nom) Visual
Remove	

*Figure A.15. After applying Undersampling technique (Spread Sub Sample)* 



Figure A.16. After applying the K-means algorithm to a balanced dataset

Preprocess Classify Cluster Associate Select attributes Visualize	CPython Scripting			
Open file Open URL C	open DB Gener	rate Unc	to E	dit Save
ter				
Choose SpreadSubsample -M 1.0 -X 0.0 -S 1				Apply
Irrent relation		Selected attribute		
Relation: balance test Instances: 186	Attributes: 5 Sum of weights: 186	Name: Class name Missing: 0 (0%)	Distinct 3	Type: Nominal Unique: 0 (0%)
ttributes		No. Label	Count	Weight
		1 B 2 R	64 62	64.0 62.0
No. Name 1 Class name	Pattern	3 L	60	60.0
No.         Name           1         Class nome           2         Left Wat           3         Left Dis           4         Right Wgt           5         Right Dis	Pattern	3 L	60	60.0
No. Name 1 Class name 2 deft Wort 3 deft Dis 4 Right Wort 5 Right Dis	Pattern	3 L Class: Class name (Nom)	60	60.0 • Visua
No. Name           No.         Name           1         Glass name           2         Left Wat           3         Spirit Wat           5         Right Dis	Patern	Class: Class name (Nom)	60 	60.0 • Visua

Figure A.17. New dataset after a combination of clusters



Figure A.18. Decision tree for the clustered dataset

## **Car Dataset**

Preprocess Classify Cluster Associate Select attributes Visualize CPvthon Scripting			
Open file Open URL Open DB Gene	erate	Undo	Edit Save
Choose SpreadSubsample - M 1.0 - X 0.0 - 8 1			Apply Stop
urrent relation	Selected attribute		
Relation: car dataset Attributes: 7 Instances: 1728 Sum of weights: 1728	Name: class Missing: 0 (0%)	Distinct 4	Type: Nominal Unique: 0 (0%)
ttributes	No. Label	Count	Weight
All         None         Invert         Pattern           No.         Name	1 acc 2 good 3 unacc 4 vgood	394 69 1209 66	344.0 89.0 1208.0 86.0
6 ☐ safety 7	Class: class (Nom)		▼ Visualize
Ramove	314	99	1299
tatus			
ок			Log 🧬

Figure A.19. Original class distribution

Preprocess Classify Cluster Associate Select attributes Visualize CPython Scripting			
Open file Open URL Open DB Gener	rate Ur	do E	dit Save
ter			
Choose SpreadSubsample -M 1.0 -X 0.0 -S 1			Apply Stor
Irrent relation	Selected attribute		
Relation: cartest         Attributes: 7           Instances: 1728         Sum of weights: 1728	Name: class Missing: 0 (0%)	Distinct: 4	Type: Nominal Unique: 0 (0%)
tributes	No. Label	Count	Weight
All None Invert Pattern No. Name 1 boying 2 maint 3 doors 4 persons 5 lug_boot	1 acc 2 good 3 unacc 4 vgood	384 69 1209 66	344.0 199.0 1209.0 66.0
6 safety 7 ctass	Class: class (Nom)		Visualize
Remove	281	1399 9	8

Figure A.20. After applying the nearest neighbor algorithm
Weka Explorer Preprocess Classify Cluster Associate Select attributes Visualize CPython Scripting			- D :
Open file Open URL Open DB Gene	rate	Indo E	dit Save
ter			
Choose SpreadSubsample -M 1.0 -X 0.0 -S 1			Apply Stop
irrent relation	Selected attribute		
Relation: cartest-weka.filters.supervised.instance.SpreadSubsample-M1.0-X0.0-S1 Attributes; 7 Instances: 264 Sum of weights: 264	Name: class Missing: 0 (0%)	Distinct 4	Type: Nominal Unique: 0 (0%)
tributes	No. Label	Count	Weight
All None Invert Pattern	1 acc 2 good 3 unacc 4 vgood	66 66 66 66	66.0 66.0 66.0
1 _ buying 2 _ maint 3 _ dours 4 _ personal 4 _ personal 5 _ safety 5 _ safety 5 _ safety 5 _ safety	Class: class (Nom)		Vieualize
Remove	89	99 99	
latus			
OV.			Log

*Figure A.21. After applying the undersampling technique (Spread Sub Sample)* 



Figure A.22. After applying K-means algorithm to a balanced dataset

Selected attribute           No.         Label           1         unacc           2.8         acc           3.900d         4.900d           Class: Class (Nom)         Class (Nom)	Distinct 4 Count 94 83 92 89	Save           Apply         Stop           Type: Nominal Unique: 0 (0%)         Stop           Weight         94.0           83.0         92.0           88.0         92.0
Selected attribute Name: Class Missing: 0 (0%) No. Label 1 unacc 2 acc 3 vgood 4 i good Class: Class (Nem)	Distinct 4 Count 94 83 92 89	Type: Nominal Unique: 0 (0%) Weight 94.0 83.0 92.0 88.0
Selected attribute Nmc Class Missing: 0(%) No. Label 1 unacc 2 acc 3 vgood 4 good Class: Class (Nom)	Distinct 4 Count 94 83 82 92 88	Apply         Stop           Type:         Nominal           Unque:         0 (0%)           Weight         94.0           83.0         92.0           88.0         92.0
Selected attribute           Neme Class           Nissing: 0(%)           No.         Label           1         unacc           2         acc           3         vgood           4         good	Distinct 4 Count 93 83 92 89	Type: Nominal Unique: 0 (0%) 94.0 83.0 92.0 88.0
No. Label 9.0 Label 1 unacc 2 acc 3 ygod 4 good Class: Class (Nom)	Distinct 4 Count 94 83 92 98	Type: Normeal Unique: 0 (0%) Weight 94.0 83.0 92.0 88.0
No. Label 1 unacc 2 acc 3 vgood 4 good Class: Class (Nom)	Count 94 83 92 89	Weight 940 830 920 880
1 unacc 2 acc 3 vgood 4 good Class: Class (Nom)	94 83 92 88	940 830 920 880
Class: Class (Nom)		
Class: Class (Nom)		
94	83	99

Figure A.23. New dataset after a combination of clusters



Figure A.24. Decision tree for the clustered dataset

## **House Vote Dataset**

Preprocess Classify Cluster Associate Select	attributes Visualize CF	Python Scripting						
Open file Open URL	Oper	DB	Gene	erate	Un	ot	Edit	Save
Iter						25 C		
Choose SpreadSubsample -M 1.0 -X 0.0 -S 1								Apply Sto
urrent relation				Selected at	tribute			
Relation: house vote Instances: 435		Attri Sum of we	butes: 17 eights: 435	Name: Missing:	Class_2ame 0 (0%)	Distinct 2	Type: Nomin Unique: 0 (0%)	al
ttributes				No.	Label	Count	Weight	
All None	Invert	Patter	m	1	republican democrat	168 267	168.0 267.0	
Construction     C								
7 religious-groups-in-schools 8 and-satellitet-schan 9 aid-to-nicaraguan-contras 10 mx-missile 11 immigration 12 syntules-corporation-cutback 13 education-spending 14 superfund-right-sue				Class: Clas	is_2ame (Nom)		287	Visualiz
16 crime 16 du/y-free-exports 17 export-administration-act-south-africa	move							

Figure A.25. Original class Distribution

Open Ne     Open URL     Open DB     Generate     Undo     Edit     Save       itler     Choose     SpreadSubsample-M1.0.×00-81     Apply     Store       Current relation     Apply     Selected attribute     Name     Type: Nominal       Relation: housevole fielt     Sum of weights: 435     Sum of weights: 435     Name     Type: Nominal       Main     None     Invert     Pattern     No     Label     Count     Weight       1     frage.gening     168.0     207.0     207.0     168.0     207.0       1     frage.gening     168.0     207.0     207.0     168.0       2     balance     168.0     207.0     207.0     168.0       3     water-project-cost-sharing     207.0     207.0     168.0       3     water-project-cost-sharing     168.0     207.0     170.0       4     addition-distribution     Count     Weight     168.0       5     physician-feefreeze     160.0     168.0     207.0       3     water-project-cost-sharing     20.00     20.00     160.0       4     addition-distration-act-south-africa     170.0     170.0       12     spreture instructure     170.0     170.0       3 <td< th=""><th>Preproces</th><th>s Classify</th><th>Cluster</th><th>Associate</th><th>Select attributes</th><th>Visualize CPython Scripting</th><th>]</th><th></th><th></th><th></th><th></th><th></th><th></th></td<>	Preproces	s Classify	Cluster	Associate	Select attributes	Visualize CPython Scripting	]						
Apply     State       Choose     SpreadSubsample-M1.0.×0.0.6.81     Apply     State       Carrent relation     Relation. housevole fielt     Sum of weights: 475     Sum of weights: 435       Name     Imply     Distinct:     Unge: Case     Unge: Case     Unge: Case       All     None     Invent     Pattern       1     Class     20m of weight: 435     Unge: Case     Unge: Case       2     Aname     Pattern     No     Label     Count     Weight: 1       1     Crease 20me     188     188.0     2     democrat     207.0       3     water-projet Cases haring     3     3     188     188.0       3     water-projet Cases haring     3     188     188       4     about cases     188     188     188       5     perfinite haring     188     188	с	pen file		Ope	n URL	Open DB	Gen	erate		Undo	Edit	Sa	we
Choose     SpreadSubsample-M1.0.×0.0-81       Current relation       Relation: housevole fielt     Attitudes: 17       Relation: housevole fielt     Sum of weight: 435       All     None     Invert       Pattern       No.     Invert       Pattern       Poil     Destand: 2       All     None       Invert     Pattern       Invert     Pattern <td>ilter</td> <td></td>	ilter												
Current relation     Selected attribute       Relation: housevide fast     Attributes: 17       Instances: 435     Sum of weight: 435       Name     Invert       All     None       Invert     Pattern       Image: Selected attribute     Use (0%)       Distanct: 2     Ungetty       Image: Selected attribute     Use (0%)       Image: Selected attribute     Image: Selected attribute       Image: Selected attribute     Image: S	Choose	SpreadSub	sample ·	M 1.0 -X 0.0 -	31							Ap	ply Stop
Relation: Nousevole field     Attitudies: 17.3       Instances: 435     Sum of weights: 435       Main     None       Image: Control instances: 435     Sum of weights: 435       Main     None       Image: Control instances: 435     Sum of weights: 435       Main     None       Image: Control instances: 435     Sum of weights: 435       Main     None       Image: Control instances: 435     Sum of weights: 435       Image: Control instance     Image: Control instance       Image: Control instance     Image: Control ins	Current rela	tion						Selected at	ttribute				
No.     Name       1     None       1     Class.       2     handlcapped-inants       3     water-project-cost-baring       4     adoption-dhe-sudget-solution       5     physican-tefrezze       6     de-bardoc-rade       7     eligious-groups-in-sthools       8     deriverse       9     patentine-station-act-south-africa       11     previne-station-act-south-africa	Relation Instances	n: housevote t s: 435	est			Sum o	Attributes: 17 f weights: 435	Name Missing	Class_2ame 0 (0%)	Distinct: 2		Type: Nominal Unique: 0 (0%)	
No.     Name       No.     Name       2     handicapped Atalia       3     hashador-add       7     religious-frome-two-subject solution       8     and solution-atalianing       0     add-micraspondin-three-subject solution       1     republican       18     add-micraspondin-three-subject solution       0     add-micraspondin-three-subject solution       10     mm:missile       11     appertunct ontras       11     republican       12     appertunct ontras       13     add-micraspondin-three       14     appertunct ontras       15     oth/refere exponts       17     expont-administration-act-south-athica       16     ddt/refere exponts       17     expont-administration-act-south-athica	Attributes							No.	Label	Count		Weight	
All     None     invert     Pattern       No.     Name     Invert     Pattern       1     Class 2 ane     267     267.0       2     Indicaped - finition     200     267.0       3     Water-project-code-flaming     267     267.0       4     addition-dise-baddition     267     267.0       5     addit-diseped - finition     267     267.0       6     addition-dise-baddition     267     267.0       7     religious-droup-baddition     267     267.0       8     addition-dise-baddition     267     267.0       10     intrimingation     9     364.0     36.0       11     immigration     36.0     36.0     36.0       12     synthelis-coprotation-cutback     36.0     36.0       13     eduction-spanding     36.0     36.0       14     superfinishinghtw-size     36.0       15     exprint-administation-act-south-africa     37       110     intriministation-act-south-africa     36.0       120     synthelic-coprotation-cutback     36.0       131     exprint-administation-act-south-africa     37       141     superfinishinghtw-size     36.0       15     exprint-administation-act-s								1	republican	168		168.0	
No.     Name       1     Ctass_Qame       2     handlicapped-intants       3     water-project-cost-haring       4     adopt-of-the-budget-solution       5     physician-the-freeze       6     et-shando-raid       7     et-glocus_provash-solution       9     et-advado-raid       10     memissite       11     immigration       12     synthele-coprotation-cuback       13     education-spanding       14     superfund-ightNe-sue       15     cirine       16     experturb-soluth-addica       17     experturb-soluth-addica		All		None		Invert Pa	ittern	2	democrat	267		267.0	
1     Citas 2009       2     Definition       3     vater-crojet-cost-sharing       4     addro-diffe-todgete-finition       5     physican-ter-secate       6     et-shardbard       7     et-gloues-groups-in-sthoots       8     and-sheffe-todgete-solution       9     add-in-ficate solution       10     add-in-ficate solution       11     immigration       12     synthele-solution-add-       13     education-spending       14     appertunctified       15     crime       16     dity-free-exports       17     export-administration-add-south-africa	No.	Name											
2     haddcaped-infants       3     wade-royded-cost-baring       4     adoption-diff-budgetesolution       5     physican-feefreeze       6     etsalwado-raid       7     etglous-groups-in-schools       8     anti-stellite-levbacket-baring       10     mc-missile       11     immigration       12     synthetic-copration-cutback       13     eduction-spanding       14     immigration       15     physice       16     opperation-cutback       17     export-administration-act-south-africa	1	Class 2a	me	-									
3     vate-project-cost-blaring       4     adoption-d/Re-tudget solution       5     physican-tefrezz       6     destandariad       7     eligious-provision-cuback       10     mornissite       11     immigration       12     synthele-coprotation-cuback       13     education-spacefing       14     superfund-influt-sue       15     crime       16     dity-free-exports       17     export-administration-act-south-africa	2	handicap	ped-infai	nts									
adoption-differe-budgetresolution       5       6       4       adoption-differe       8       add-addition-add       10       monitorial       11       immigration       12       13       14       15       16       17       expondention-addreack       13       14       15       16       17       expondention-addreack       16       17       expond-administration-act-south-africa	3	water-pro	ject-cost-	sharing									
S     Physican-Refreze       S     Physican-Refreze <t< td=""><td>4</td><td>adoption-</td><td>of-the-bu</td><td>dget-resolutio</td><td>n</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>	4	adoption-	of-the-bu	dget-resolutio	n								
b     B-3 shido-ado       c     B-3 shido-ado       c     ant-satistic escontas       c     ant-satistic es	5	physician	n-fee-free	ze									
Imploye-by outputs - index controls       Imploye-by outputs - index controls </td <td>0</td> <td>el-salvado</td> <td>or-aid</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>10</td> <td></td>	0	el-salvado	or-aid									10	
a distributive source		religious-	groups-ir	1-SCHOOLS				Class: Clas	ss_2ame (Nor	1)			Visualize A
10     ms-msale       11     minigaton       12     synthels-corporation-cuback       13     education-spending       14     superfunds-fifthe-sue       15     education-spending       16     dayhife=exports       17     expertunds-fifthe-sue       18     education-spending       19     dayhife=exports       11     expertunds-fifthe-sue       12     fifthe-superinds-fifthe-sue       13     education-act-south-africa	0	aid-to-nic	araquan.	contras									
11     Immigration       12     synthes-corporation-cuback       13     education-spending       14     spending       15     otime       16     education-sect-south-africa	10	mx-missil	le	-							267		
12     synthelis-coprotion-cuback       13     ducation-spanding       14     superfund righthe-sue       15     crime       16     crime       19     duf-free-exports       10     crime       11     expertund righthe-sue       12     expertund righthe-sue       13     duf-free-exports       14     expertund righthe-sue       15     crime       16     crime       17     export-administration-act-south-africa       18     crime       19     tabus	11	immigrat	ion										
13     education-specificity       14     upperfind-fpHt0-sue       15     orine       15     orine       16     eduf-free-topris       17     export-administration-act-south-africa	12	synfuels-	corporatio	on-cutback									
16     uperfund-right-sue       15     utorine       16     utorine       17     export-administration-act-south-atrica	13	education	-spendin	10									
Isource and the second	14	superfund	d-right-to-	sue									
16 dtyfree-exports 17 export-administration-act-south-aftica Remove Remove Returns	15	Crime						168					
17 _ export-administration-act-south-attica	16	duty-free-	exports										
Latus	17	export-ad	ministrati	ion-act-south-	amca								
Status					Remove								
Status	tatua												
	status												

Figure A.26. After applying the nearest neighbor algorithm

Open file Open URL Open DB Ge	enerate Undo Edit Save
ter	
Choose SpreadSubsample -M 1.0 -X 0.0 -S 1	Apply Stop
irrent relation	Selected attribute
Relation: housevote test-weka filters supervised instance. SpreadSubsample-M1.0 Attributes: 17 Instances: 336 Sum of weights: 336	Name: Class_2ame         Type: Nominal           Missing: 0 (0%)         Distinct: 2         Unique: 0 (0%)
tributes	No. Label Count Weight
All None Invert Pattern	1 republican 168 168.0 2 democrat 168 168.0
No. Name	
2 handicapped-infants	
3 water-project-cost-sharing 4 adoption-of-the-budget-resolution	
5 physician-fee-freeze	
7 🗌 religious-groups-in-schools	Class: Class_2ame (Nom) Visualize
8 anti-satellite-test-ban 9 aid-to-nicaraouan-contras	
10 mx-missile	168
12 syntuels-corporation-cutback	
13 🔄 education-spending	
14 superfund-right-to-sue	
14 superfund-right-to-sue 15 crime	
14superfund-right-b-sue 15crime 16duty-free-exports 17export-administration-ad-south-africa	
14superfund-sight-bosue 15crime 16duty-free-exports 17export-administration-act-south-africa	
14superfund-rightAb-sue 15rime 16dutyfree-reports 17export-administration-act-south-strica Remove	
14     superfund-rightNo-sue       15     crime       16     duty-free-suports       17     export-administration-act-south-africa         Remove	

Figure A.27. After applying the undersampling technique (Spread Sub Sample)



Figure A.28. After applying K-means algorithm to a balanced dataset

Open file Open URL Open DB Ger	ierale Undo Edit Save
ler	
Choose SpreadSubsample -M 1.0 -X 0.0 -S 1	Apply
rrent relation	Selected attribute
Relation: housevote test Attributes: 17 Instances: 535 Sum of weights: 535	Name: Class_2ame Type: Nominal Missing: 1 (0%) Distinct: 2 Unique: 0 (0%)
ributes	No. Label Count Weight
All None Invert Pattern	1 republican 331 331.0 2 democrat 203 203.0
3 water-project-cost-sharing 4 adoption-ofthe-budget-solution 5 physician-fe-freeze 6 el-salador-ind 7 etilgious-groupe-in-schools 8 anti-salitife-sets-ban 9 aid-6-nicaraguar-contras 10 mc-missile 11 immigration 12 synthe-scriptration-cutback 13 education-spending 14 superfund english-cue 15 crime	Class: Class_2ame (Nom) Visu
17  export-auministration-act-south-amca	

Figure A.29. New dataset after a combination of clusters

Tree View	07.50 <b>V</b>	
1: adoption-offhe-budget-resolution = n =		
2:el-salvador-ald	71: aid-to-nicaraguan-contras	
= y = n	= n = y	
$= n \qquad = y \qquad = n \qquad = y = n$		
4: physician-feeze 53: religio. 60. 73: symbols-cu. 80: superior $x^{(1)} = x^{(2)} = n$ $= x^{(2)} = x^{(2)} =$	fund-right 96: physician-fee-freeze 113: el-salvador-aid y = n = y = n = y = n	
5:symLuf5-comporation-cutb	ph, 106 super. 114: physician-fee-f. 127: immigration n = y = n = n = y	
8 expont-administratio. 15∵. 2. 32′w., 42′. 47′. 67′. 82′. 63′. 17′. 82′. 85′. 191′. = y = n = y = n = n y = n = y = n y = n y = n y = n y = y =	98:s         1         115: water-project⊧co.         122: .         133: physician-fee-freezener.           a         a         b         a         b	)e
7. religi 14 33. export. 38 43 44 49, 58 59. de 64, 51 58 677 78 53 54 rep 85 53	104:** 104: republi *** 11, 1. 134: religious-grc 137: crime en =ney =yen =ney =ney =ney =ney =yen	
P g: d. 17: 27 25: hand 34 35: 37: 38: democra 50 51: democrat (2.01/0.01) 87: 88: dem	101: water- 110 ** 118: 12 124 12:130 131 *** 136 141 : democrat (34	4.75/0.
= ney = ney = ney	= h= y = h= y = h= y	
18. export ad 21 · repui 26 27 · republican (4.02/2) - e y - y n	10, 103 : democrat (; 119 : 120 : republican (1.54/0) 139 140 : democrat (5.44	\$/0.02)

Figure A.30. Decision tree for the clustered dataset

## **Mushroom Dataset**

Treprocess Classily Cluster Associate Select attribute	s Visualize CPython Scripting			
Open file Open URL	Open DB	Generate	Undo	Edit Save
Iter				
Choose SpreadSubsample -M 1.0 -X 0.0 -S 1				Apply Sto
arrent relation		Selected attribute		
Relation: Mushroom Instances: 8124	Attributes: 23 Sum of weights: 8124	Name: class Missing: 0 (0%)	Distinct 2	Type: Nominal Unique: 0 (0%)
ttributes		No. Label	Count	Weight
		1 p	3916	3916.0
All None	Invert Pattern	2 8	4200	4206.0
No.         Mailie           1         dashape           2         dashape           3         dashape           4         bashape           5         odor           6         odor           7         oli-attachment           8         oli-stachment           9         oli-stachment           11         falli-shape           12         statik-roat           13         falli-surface-ablow-ring           14         statik-surface-ablow-ring		Class class (Nom)		Visualiz
16 stail-color-above-ring 16 stail-color-below-ring 17 veil-type 18 veil-color		<u> </u>		
16 stalk-color-below-ring 17 vell-tpe 18 vell-color 18 vell-color Remove				

Figure A.31. Original class distribution

Preprocess Classify Cluster Associate Select attributes Visualize CPython Scripting				
Open file Open URL Open DB Gene	rate	Indo E	idit Sa	ive
Filter				
Choose SpreadSubsample -M 1.0 -X 0.0 -S 1			Ap	ply Stop
Current relation	Selected attribute			
Relation: Mushroom test Attributes: 23 Instances: 8124 Sum of weights: 8124	Name: class Missing: 0 (0%)	Distinct 2	Type: Nominal Unique: 0 (0%)	
Attributes	No. Label	Count	Weight	
All         None         Invert         Pattern           No.         Name         1         closs         1           1         closs         4         cap-state         4           3         cap-state         4         cap-state         4           5         bruises         6         odor         4	2 6	4208	4208.0	
7       jil-stachment         8       jil-spacing         9       jil-size         10       jil-color         11       stall-shape         12       stall-surgece-show-ring         13       stall-color-above-ring         14       stall-color-above-ring         15       stall-color-above-ring         16       stall-color-above-ring         17       veli-color-above-ring         18       veli-color-above-ring         17       veli-type	Class: class (Nom)		100	Visualize A
Remove				
Status				

Figure A.32. After applying the nearest neighbor algorithm

🕽 Weka Explo	rer	Y	Ŷ	Ŷ	Ŷ	Y							- '	o x
Preprocess	Classif	y Cluste	r Associate	Select attribute	s Visualize	CPython Scrip	ting							
Op	en file		Ope	en URL		Open DB		Senerate		Undo ]	Edit	][	Save	
Filter														
Choose	SpreadS	ubsample	-M 1.0 -X 0.0 -	81									Apply	Stop
Current relation	on							Selected at	tribute					
Relation: Instances:	Mushroor 7832	m test-wei	a.filters.super	vised.instance.Sp	preadSubsamp	ole-M1 Sun	Attributes: 23 n of weights: 7832	Name: Missing:	class 0 (0%)	Distinct: 2		Type: Nominal Unique: 0 (0%)		
Attributes								No.	Label	Count		Weight		
	411		None	10	Invert		Pattern	1	p e	3916 3916		3916.0 3916.0		
100 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 16 16 16 16 16 16 16 16 16	cap-shi cap-shi cap-col bruises odor gill-atta gill-spa gill-spa gill-sca stalk-so stalk-so stalk-so stalk-so stalk-so	ape face or chment cing or nape ot urface-abou plor-above plor-below	ve-ring w-ring ring					Class: class	s (Nom)		3916			/isualize Al
Status OK	veil-col	7		Remove									Log	<b>~~</b> ``

*Figure A.33. After applying the undersampling technique (Spread Sub Sample)* 



Figure A.34. After applying K-means algorithm to the balanced dataset

Preprocess Classify Cluster Associate Select attributes Visualize CPython Scripting			
Open file Open URL Open DB Genera	iteU	ndo E	Edit Save
ter			
Choose SpreadSubsample - M 1.0 - X 0.0 - S 1			Apply Stop
irrent relation	Selected attribute		
Relation: mushroom test Attributes: 23 Instances: 7832 Sum of weights: 7832	Name: class Missing: 0 (0%)	Distinct 2	Type: Nominal Unique: 0 (0%)
tributes	No. Label	Count	Weight
	1 p 2 e	3142 4690	3142.0 4690.0
No.         Name           6         gli-Bitachment         *           7         gli-Boardo         *           8         gli-Bitachment         *           9         gli-Color         *           10         stalk-shape         *			
12     sain-sainta-selow-ring       13     sain-sainta-selow-ring       14     sain-sainta-selow-ring       15     sain-sainta-selow-ring       16     sainta-selow-ring       17     sainta-selow-ring       18     sign-sainta-selow-ring       19     sainta-selow-ring       10     sainta-selow-ring       11     sing-sainta-selow-ring       12     space-selow-ring       13     sing-sainta-selow-ring       14     sainta-selow-ring       15     sing-sainta-selow-ring       16     sainta-selow-ring       17     sing-sainta-selow-ring       18     sainta-selow-ring       19     sing-sainta-selow-ring       19     sing-sainta-selow-ring       19     sing-sainta-selow-ring       19     sing-sainta-selow-ring       10     sing-sainta-selow-ring       11     sing-sainta-selow-ring       12     sabinta-       13     sainta-selow-ring-selow-ring       14     sainta-selow-ring-selow-	3142		50 
atus			
atus			

Figure A.35. New dataset after combining the clusters



Figure A.36. Decision tree for the clustered dataset

## **Student Retention Dataset**

Preprocess Classily Cluster Associa	ate Select attributes	Visualize	CPython Scripting							
Open file	Open URL	0	pen DB	Gene	erate		Undo	Edit	s	ave
ilter										
Choose SpreadSubsample -M 1.0 -X 0	.0-81								A	oply Stop
urrent relation					Selected a	ttribute				
Relation: Student retention Instances: 9240			Attr Sum of w	ibutes: 17 eights: 9240	Name Missing	Class 0 (0%)	Distinct: 2		Type: Nominal Unique: 0 (0%)	
ttributes					No.	Label	Count		Weight	
All Nor		Invert	Patt	ern	1	R D	6402 2838		6402.0 2838.0	
AGE     GENDER     GENDER     Genderentile range     Genderentin     Genderentile range     Genderentile rang					Class: Clas	ss (Nom)				Visualize
9 GRANT_Y 10 LOAN_Y 11 Maj_dng 12 SAT_RANGE1 13 Major_CHANO_FAL_Spring 14 Greek_Life 15 FAL_ATIM_RANGE 16 Spring_ATIM_RANGE 17 Class		_	_		6402			2018		
	Remove									

Figure A.37. Original class distribution

Open file.         Open QRL         Open DB.         Generate.         Undo         Edt.         Saw           iter         Iter         Iter         Apply         Apply         Apply           iterative         Iterative         Iterative         Apply         Apply           iterative			1	CPython Scripting	utes Visualize	Select attribut	Associate	uster A	Clust	lassify	s Cl	cess	oces	ces	ess	ess	SS	s	s	:[	s	s	;[	CI	Cla	as	ssi	si	si	si	si	si	si	si	si	ss	ss	ss	ss	ss	ss	sif	fy		ľ	ľ	ľ	0	Clu	lus	ste	er )	A	Ass	500	ciate	Se	elect	ct attri	ibute	es	Visu	ualize	elo	CPyth	ion S	cripti	ing																							
ter Choose SpreadSubsample -41 10 - X 00 - S 1 Trent relation Relation relation test instances: 224 Attribute: 17 Attribute: 37 Attribute: 37 Attribute: 37 Attribute: 37 Attribute: 47 Attribute: 47 Attribu	Undo Edit		Gene	pen DB		URL	Open			le	pen fi	Op	0	0	Op	Op	Ope	Ope	per	pen	pen	)per	pen	en fi	file	e																								-	)	C				Ope	n UR	RL						Ope	en DE	B				Ger	nerate.				U	ndo					Edit.							Save			
Choose         SpreadSubsample - M1 0 - X 0 0 - 81         Apply           rment relation         Attributes: 77         The Nome of Weights: 9240         Name (Lass in the Nome of Weights: 9240)           Inflammes: 9240         Attributes: 77         The Nome of Weights: 9240         Nome (Lass in the Nome of Weights: 9240)           No.         Name (Lass in the Nome of Weights: 9240)         Nome (Lass in the Nome of Weights: 9240)         Nome (Lass in the Nome of Weights: 9240)           No.         Name (Lass in the Nome of Weights: 9240)         Nome (Lass in the Nome of Weights: 9240)         Nome (Lass in the Nome of Weights: 9240)           No.         Name (Lass in the Nome of Weights: 9240)         Nome (Lass in the Nome of Weights: 9240)         Nome (Lass in the Nome of Weights: 9240)           No.         Name (Lass in the Nome of Weights: 9240)         Nome (Lass in the Nome of Weights: 9240)         Nome (Lass in the Nome of Weights: 9240)           No.         Second (Lass in the Nome of Weights: 9240)         Nome of Weights: 9240)         Nome of Weight in the Nome of																										_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_			_	_	_	_	_																		_																		
Internet relation     Selected attribute       Relation retention test     Attributes: 77       Infrances: 2240     Sum of velophis: 5240       Infrances: 2240     Infrances: 2240       Infrances: 2240     Infrances: 2240       All     None     Invert       Pattern     Pattern       No.     Invert       2     Add       3     GRNCERE       3     GRNCERE       3     Summer, Y.N       0     GRNT Y       10     LOAN,Y       11     Mar, Chang       12     Sat, TANGEE       13     Mar, Chang       14     Fremove						31	1.0 -X 0.0 -S	nple -M 1.	sampl	eadSubs	Spr	ose	ose	ose	se	e			s	S	Sr	s	S	Spr	rea	ead	nd!	dS	dS	dS	dS	dS	dS	dS	d	ad;	ad	ad	ad	ad	d	IS	Su	ıb	bs	os	os	sa	an	np	ole	e - M	M 1.	1.0	-X	0.0 -8	S 1					_			_													_		_		_	_	_	_			Apply		St	to
Relations:     Attitudes:     7:       indications:     Sum of weights:     State:     Unique:     0(%)       indication:     Pattern     None     Indication:     None     Indication:       indication:     Indication:     Pattern     Indication:     None     Indication:       indication:     Indication:     Pattern     Indication:     Indication:     Indication:       indication:     Indication:     Indication:     Indication:     Indication:     Indication:       indication:     Indication:     Indication:     Indication:     Indication:     Indication:       indication:     Indication:     Indication:     Indication:     Indication:     Indication:	ttribute	ected attribute									tion	relati	relat	relat	elati	latio	atio	ation	tion	tion	tion	tion	ion	n																													_								_										Sel	lected	attr	ibute			 														
Image: Control of the second	: Class Type: Nomina : 0 (0%) Distinct: 2 Unique: 0 (0%)	Name: Class Missing: 0 (0%)	ttributes: 17 weights: 9240	Attri Sum of w					st	ntion tes 0	: rete : 924	ation: nces:	ation	ation	tion: ces:	on: es:	in: r is: 9	n: n s: 9	n: re 1: 92	ret 92	: ret : 92	n: re a: 9:	ret 92	rete 924	ent 40	ntic )	ior	on	on	on	on	on	on	on	or	ior	io	io	io	io	or	n	te	es	s	st	st	st																		5	Sum	Attrib of we	outes: 17 lights: 92	17 9240		Nam Missin	ie: ( ig: (	Class ) (0%)			Dis	tinct :	2				,	Typ	pe: N ue: C	Vomir 0 (0%)	nal /)				
All       None       Invert       Pattern         All       None       Invert       Pattern         I       ETHNIC       233       2838.0         2       D       2838       2838.0         1       ETHNIC       2406       2838.0         3       GENDER       6       6         4       percentia_range       5       8         5       Pattern       Class: Class (Nom)       *         10       DONALY       1       10         11       Marc, CHANG, FALL, Spring       1       14         12       SAT, XANGE       1       1         13       Mair, CHANG, FALL, Spring       1       1         14       Greaker, Life       1       1       1         13       Mair, CHANG, FALL, Spring       1       1       1       1         14       Greaker, Life       1	Label Count Weight	No. Label										es	es	s	s																																																									No.		Label				Cou	nt						Weig	ght					
No.     Name       1     ETHNIC       2     AGE       3     GENDER       4     percentia_mange       5     Percentia_mange       6     Semmer_XN       9     GRANTY       10     LOAN Y       11     Main Chay       12     SAT, RANGE1       13     Main CHANG FALL Spring       14     Greek, Life       15     FALL_ATUP_RANGE       16     Songa, AtuB_RANGE       17     Glass	R 6402 6402.0 2 D 2838 2838.0	1 R 2 D	attern	Path	Invert		None				All							4	۵	4	4	4	4																										ĥ	ſ					No	one			h			Inve	art			_		Patte	10				1 2	R D				640 283	2 3						640 283	2.0 8.0					
1     ETHNC       2     AGE       3     GENDER       4     perent Chas       6     Petter       6     Petter       7     GRANT_Y       9     GRANT_Y       10     LOAN_Y       11     Majc.chay       12     SAT_RANGE1       13     Major.CHANG.PALL.Spring       14     Greek.Life       15     FALL_ATUP_RANGE       16     Spring ATUP_RANGE       17     Class					inten		Hone			ame	N				-	1	1			1	1			N	Var	m	ne	пе	ne	ne	e	e	e	ne	пе	ne	me	ne	me	ne	ne	e	_		_	_	_	_		0	-	_		_							_					_		Tutter																							
2 → AGE 3 → GENDER 4 → protriktig range 5 → Prent Char 9 → Rent Char 10 → Char 11 → Rent Char 12 → Right Renk CE 13 → Rent Char 13 → Rent Char 14 → Greek Line 15 → Full_ATUP_RANCE 15 → Rent Char 16 → Rent Char 17 → Class										THNIC	_ E	1,	1	1	10	1								E	TH	H	IN	NI	NI	NI	NI	NI	NI	NI	N	IN	IN	IN	IN	IN	N	110	С																																																
4     percentie_range       5     Petter.Char       6     Petter.Char       7     CPA_CHAR       8     Summer_Y_N       9     CR4NT_Y       10     LONALY       11     Sore-CHAR SAUCE1       12     SAT/PANGE1       13     Major_CHANG SAUL_Spring       14     Create-Life       15     Soring, TAM_PANGE       16     Series										GE ENDER		2	2	2	21	2		Н	Н			Н		AC G	GE	EN	I VD	ID	D	D	DI	DI	DI	D	ID							DE	EF	R	R	2	2																																												
s   Parent Ohar   Class: Class (Nom)								nge	_rang	ercentile	pe	4	4	4	4	4		ğ	ğ	Į,	Ö,	ğ	Į,	pe	ere	rce	er	er	en	en	en	en	en	en	er	cer	ce	ce	ce	ce	e	n	til	le	e_	e	e	1	rar	ing	je																																								
7     GPA_CHAR       8     Summery, N       9     GRANT, Y       10     LOAN, Y       11     May, CHANG, FALL_Spring       12     SAT, RANGE1       13     Main, CHANG, FALL_Spring       14     Greek, Line       15     Syng, ATMP, RANGE       17     Class									har	arent_Ch ELL Y		6	6	6	6	5	2	Н	Н		H	Н	H	P	an	rei	Ent	Ent	I	I	nt	nt	nt	I	E	en L	en	en	er	en	E	nt	Y	C	Ch	h	h	ha	ar																																										
8 Summer_V.N 9 CoxHVT Y 10 LOAN,Y 12 SALSROE1 13 Major_CHWACFALL_Spring 14 Oreke_Line 15 Spring_ATAM_RANGE 17 Class Remove	ss (Nom)	ss: Class (Nom)							AR	PA_CHA	G	7 [	7	7	7	7	Ē	Ĭ	ŏ	į,		ğ	Į,	G	SPA	A	1	0	0	0	0	0	0	0	0	U	4	1	A_	1	J	C	2	1/	A	AF	AF	R	2																						Cla	ass: Cla	ass	(Nom)														-	Vis	uali	lize
10     LOALY       11     Mary robg       12     SAT_FANGE1       13     Marc (ANNG, FALL, Spring)       14     Greek, Life       15     FALL, TATP_FANGE       16     Spring, ATMP_RANGE       17     Class	-							4	Y_N	ummer_ RANT Y		9	9	9	9	9		Н	H	H	H	Н	۲	G	sun GR/	RAI	۹۳ ۸۷	N.	IM N	IM N	m N	m N	m N	IM N	nm N	nn AN	nn AN	nn AN	nr AN	nn AN	N	m VT	iei T	ĥ	Y	Y	Y	Y	-1	N																																									
11 Jag_ong 12 SARANGE1 13 Jajor_CHANG_FALL_Spring 14 Greek_Life 15 FALL_ATHP_RANGE 17 Cosss Remove		6402								Y_NAC		10	10	10	10	0	Ē	ğ	ğ	į,		ğ	į,	L	OA	AN	N	N	N_	N_	N_	N_	N_	N_	N	N	N	N	N	N	N	1	Y	ī																												6402						_													
13 Major CHWS PALLSpring 14 Greek-Life 15 FALL_ATUP_RANCE 17 Class Remove								1	IGE1	ajr_cng		12	12	12	12	2		Н	Н	2	8	Н	H	S	Jaji SAT	IJC_ T		_C	_c R	_c R	R	R	R	_c R		-	H	-	-	-	F	R	A	g N	NC	IC	IC	IGI	E	1																																									
14 Greek_Life 15 FALLTHP_RANGE 16 Spring_ATMP_RANGE 17 Class Remove Remove							Spring	G_FALL_	ANG_	ajor_CH	🔲 M:	13 [	13	13	13	3	3							M	laj	ijoi	or_	or_	r_	r_	r_	r_	r_	r_	or_	or.	or,	or,	or	or,	or_	4	c	H	Η/	HA	HA	A	N	G_	F)	ALI	1.3	Sp	orin	ng																																			
16 Gpring_ATMP_RANGE								DANCE		reek_Life	G	14	14	14	14	4		Н	H		-	Н	4	G	Gre	ee	ek,	∋k,	ek.	ek.	K_	K_	K_	ek.	∋k,	ek	ek	ek	eł	ek	∋k	k		.if	ife	fe	fe	le III			AN		E																																						
17 Class Remove							E	RANGE	TMP_R	pring_AT	S	16	16	16	16	6	5	ŏ	ŭ	ī,		ŏ	ő.	S	Spri	rin	ng	ng	ng	ng	ig	ig	ig	ng	ng	ing	ing	ing	in	ing	ng	g.	1	A	T	TI	TI	ТМ	MP	-	RA	ANC	IGE	E																																					
Remove	2838									lass		17	17	17	17	7								C	Cla	85	35	s	s	s	s	s	s	s	s	ss	ss	ss	ss	ss	s	8																																							2838	_	_	_	_	_	_	_	_	_	-
Remove																																																																																											
						Remove																																																				Re	emov	ve																															
atus																																																																							1																				

Figure A.38. After applying the nearest neighbor algorithm

Preprocess Classify Cluster Associate Select attributes Visualize CPython Scripting				
Open file Open URL Open DB Gene	erate	Undo	Edit 5	Save
ter				
Choose SpreadSubsample -M 1.0 -X 0.0 -S 1			A	stop
rrent relation	Selected attribute			
Relation: retention test-weka filters supervised instance.SpreadSubsample-M1.0 Attributes: 17 Instances: 5676 Sum of weights: 5676	Name: Class Missing: 0 (0%)	Distinct: 2	Type: Nominal Unique: 0 (0%)	
tributes	No. Label	Count	Weight	
All Next Delta	1 R 2 D	2838 2838	2838.0 2838.0	
3 GENDER 4 percentile_ange 5 Parent_Char 6 PELLY 7 GPA_CHAR 8 Summer_VN 9 GRANT_Y 10 LOAN_Y	Class: Class (Nom)		208	Visualize A
11				
Remove				
atus				

Figure A.39. After applying the undersampling technique (Spread Sub Sample)

Preprocess Classify Cluster Ass	ociate Select attributes Visualize CPython Scripting	
usterer		
Choose SimpleKMeans -init 0 -ma	ok Ca 🥥 Weka Clusterer Visualize: 18:56:26 - SimpleKMeans (retention test-weka.filters.supervised.instance.SpreadSubsampl – 🗆 🗙 10	
uster mode	X: Instance_number (Num)	
<ul> <li>Use training set</li> </ul>	Colour: Cluster (Nom)	
O Supplied test set Set	Reset Clear Open Save	
O Percentage split	Mark retention first works Ellers evenemined instance Encode Cubesmalle M4.0 V0.0 E4, shustered	
<ul> <li>Classes to clusters evaluation</li> </ul>	Pior Tetenuon test-weka.metris.supervised.instance.spread subsample-wito-xo.or.st_clustered	
(Nom) Class		
Store clusters for visualization		
Ignore attributes	× <u>39</u>	
1		
Start	Stor	
sult list (right-click for options)		
16:02:54 - SimpleKMeans		
16:28:02 - SimpleKMeans	h h h	
16:28:37 - SimpleKMeans		
16:50:59 - SimpleKMeans	141)741,882954657755	
17:27:03 - SimpleKMeans		
17:48:13 - SimpleKMeans	2837.5 5575 <b>8</b>	
18:56:26 - SimpleKMeans		
	Class colour	
	cluster0 cluster1	
100		

Figure A.40. After applying K-means algorithm to a balanced dataset

Preprocess Classify Cluster Associate Select attributes Visualize CPython Scripting			
Open file Open URL Open DB Gen	erate Undo		dit Save
iter			
Choose SpreadSubsample -M 1.0 -X 0.0 -S 1			Apply Stop
urrent relation	Selected attribute		
Relation: retention test Attributes: 17 Instances: 6510 Sum of weights: 6510	Name: class Missing: 0 (0%)	Distinct: 2	Type: Nominal Unique: 0 (0%)
ttributes	No. Label	Count	Weight
	1 R 2 D	3286 3224	3286.0 3224.0
No. Name			
1 ☐ Etnic 2 ☐ Apenumeric 3 ☐ gender 4 ☐ percentile range 5 ☐ Parent-char 6 ☐ Pelliyar			
/         GrAvchar           8         SummerN           9         Granty           10         Inclusion           110         Inclusion           12         SATrange_1           13         major change tail spring           14         greek life           15         frail attempt range           16         Spring attempt range           17         Class	Class: class (Nom)		Visualize.
Remove			

Figure A.41. New dataset after a combination of clusters



Figure A.42. Decision tree for the clustered dataset

Table A.1Original dataset applied with the c45 algorithm

Metrics	Adult	Balance	Breast Cancer	Car	Housevote	Mushroom	Retention
Total number of rules	719	33	10	135	6	24	135
Total number of infrequent rules	146	-	4	121	3	9	35
Average Accuracy	0.9017	-	0.806	0.8670	0.9451	1.000	0.7653
A. Range:90- 100	63	-	1	63	2	9	14
A. Range:80- 89	38	-	1	-	1	-	7
A. Range:70- 79	39	-	1	25	-	-	7
A. Range:60- 69	4	-	1	1	-	-	3
A. Range:50- 59	1	-	-	13	-	-	3
maximum coverage	0.000964	-	0.094	0.0308	0.8673	0.5515	0.82910
minimum coverage	0.000164	-	0.011	0.0077	0.0242	0.0040	0.00035
median coverage	0.000657	-	0.070	0.0212	0.1388	0.0490	0.00422
90-100	3	-	1	1		6	6
80-89	-	-	-	-		1	-
70-79	7	-	1	103		-	-
60-69	-	-	-	-		-	2
50-59	1	-	-	-		-	3

Table A.2Original dataset applied with EM clustering

Metrics	Adult	Balance	Breast Cancer	Car	Housevote	Mushroom	Retention
Total number of rules	524	-	5	-	4	2	32
Total number of infrequent rules	131	-	2	-	2	1	17
Average Accuracy	0.8923	-	0.766	-	0.87565	0.6933	0.6879
Range:90- 100	34	-	-	-	1	-	-
Range:80-89	32	-	-	-	1	-	2
Range:70-79	27	-	1	-	-	-	5
Range:60-69	3	-	-	-	-	1	4
Range:50-59	1	-	-	-	-	-	-
maximum coverage	0.0823	-	0.0470	-	0.1230	0.0653	0.0169
minimum coverage	0.00017	-	0.0117	-	0.0059	0.0653	0.0003
median coverage	0.00265	-	-	-	-	0.0653	0.0007
90-100	9	-	-	-	-	-	1
80-89	2	-	-	-	-	-	-
70-79	1	-	-	-	-	-	-
60-69	5	-	-	-	-	1	1
50-59	5	-	-	-	-	-	-

Table A.3			
Original dataset	applied with	K-means	algorithm

Metrics	Adult	Balance	Breast	Car	Housevote	Mushroom	Retention
			Cancer				
Total number	-	-	-	-	6	24	169
of rules							
Total number	-	-	-	-	3	9	59
of infrequent							
rules							
Average	-		-	-	0.9483	1.000	0.8766
Accuracy							
Range:90-	-	-	-	-	2	9	14
100							
Range:80-89	-	-	-	-	1	-	10
Range:70-79	-	-	-	-	-	-	7
Range:60-69	-	-	-	-	-	-	1
Range:50-59	-	-	-	-	-	-	-
maximum	-	-	-	-	0.8673	0.5515	0.82910
coverage							
minimum	-	-	-	-	0.0131	0.00408	0.00035
coverage							
median	-	-	-	-	0.1345	0.0813	0.00664
coverage							
90-100	-	-	-	-	-	-	3
80-89	-	-	-	-	1	-	-
70-79	-	-	-	-	-	-	-
60-69	-	-	-	-	-	1	2
50-59	-	-	-	-	-	-	-

Table A.4	
Original dataset applied with Mak	ke Density cluster

Metrics	Adult	Balance	Breast Cancer	Car	Housevote	Mushroom	Retention
Total number of rules	267	-	-	-	3	18	11
Total number of infrequent rules	50	-	-	-	1	7	4
Average Accuracy	0.0926		-	-	0.9769	0.9362	0.9438
Range:90- 100	-	-	-	-	1	7	3
Range:80- 89	-	-	-	-	-	-	1
Range:70- 79	-	-	-	-	-	-	-
Range:60- 69	-	-	-	-	-	-	-
Range:50- 59	-	-	-	-	-	-	-
maximum coverage	0.00265	-	-	-	0.0305	0.3309	0.0574
minimum coverage	0.00017	-	-	-	0.0305	0.0015	0.0003
median coverage	0.000513	-	-	-	0.0305	0.0490	0.0017
90-100	-	-	-	-	-	-	-
80-89	-	-	-	-	-	-	-
70-79	-	-	-	-	-	-	-
60-69	-	-	-	-	-	-	-
50-59	-	-	-	-	-	-	-

Table A.5Comparison of the several algorithms.

Dataset	Metrics	Undersampling	Oversampling	Bagging	Boosting
Mushroom	F1	0.984	1.000	1.000	0.962
	ROC	0.995	1.000	1.000	0.995
	Precision	1.000	1.000	1.000	0.947
	Recall	0.969	1.000	1.000	0.978
Balance	F1	0.930	0.918	?	?
	ROC	0.991	0.982	0.682	0.863
	Precision	0.912	0.918	0.000	?
	Re call	0.949	0.918	?	0.000
Car	F1	0.952	0.962	0.963	?
	ROC	0.997	0.992	0.995	0.897
	Precision	0.966	0.962	0.963	?
	Recall	0.969	0.962	0.962	0.700
Retention	F1	0.997	0.842	0.867	0.807
	ROC	0.997	0.864	0.927	0.829
	Precision	0.997	0.846	0.866	0.807
	Recall	1.000	0.860	0.868	0.808
Breast	F1	0.816	0.813	0.788	0.741
Cancer					
	ROC	0.873	0.797	0.884	0.751
	Precision	0.798	0.820	0.823	0.741
	Recall	0.835	0.844	0.811	0.755
Adult	F1	0.997	0.855	0.829	0.902
	ROC	1.000	0.931	0.871	0.956
	Precision	0.999	0.856	0.834	0.903
	Recall	0.995	0.855	0.842	0.905
House Vote	F1	0.979	0.917	0.957	0.954
	ROC	0.977	0.979	0.956	0.992
	Precision	0.982	0.917	0.951	0.954
	Recall	0.977	0.917	0.956	0.954