Name: Subhasree Vadukoot Student Number: 3014289

Course	MSCBD-DMAT
Stage / Year	1
Module	Data Mining Algorithms & Techniques
Semester	2
Assignment	Assignment 1
Date of Title Issue	05th March 2020
Assignment Deadline	23rd March 2020
Assignment Submission	Upload to Moodle
Assignment Weighting	15% of module

Objective of the Assignment

To successfully apply a set of data mining skills imparted through lectures and lab session to a previously unseen dataset using Weka to achieve knowledge discovery and producing a written technical paper format report.

Deliverables

A single zip called FirstName_LastName_StudentNumber._ass1.zip to be uploaded to Moodle containing the following files

- This file edited to contain the results of your investigation. Each of the **NUMBERED** headings should be expanded to satisfy the requirements of the section.
- A set of supporting files including but not limited to the following, which should be clearly referenced from your documentation.
    - dataset.arff
    - trainigSet.arff
    - testingSet.arff
    - j48tree.arff
    - associationrules.arff
    - kmeans.arff
    - dbscan.arff

Choosing Your Dataset

1. Your dataset should concern a real-world problem that lends itself to easy understanding by your classmates.
2. It should ideally have >1000 tuples/rows/instances.
3. It should ideally have >=6 attributes
4. It should have attributes which can serve as labels so that the accuracy of your data analysis can be determined.
5. If you cannot find one dataset which is suitable for use with all techniques, then you may choose 2. Please clearly indicate which dataset was used in which case and introduce this dataset

* Please refer to additional materials section in moodle for datasets links.

* Please post to the student discussion forum “Assignment 1 - Dataset Selection” clearly indicating which set you are using so that other students do not select the same dataset.

Part 1 – Classification

1. Description of your dataset and findings – 10%

- **Title**: J48 Classification and Apriori Association Rule Mining with *Diabetes 130-US hospitals for years 1999-2008* Data Set.
- **Objective**:

  1. To predict diabetes readmission risks through classification techniques and association rule mining after understanding the effects of diabetes in patients through literature review and detailed study of the Diabetes 130-US hospitals for years 1999-2008 Data Set and its attributes.

  2. To apply various feature selection methods in Weka to identify the best features and to vary the hyperparameters to find and evaluate the models with better performance and accuracy after training the model and testing on the trained model.

  3. To implement and evaluate ensemble methods such as Bagging, Boosting in Weka Experimenter to understand the effect of these ensemble methods on the model with J48 algorithm.

  4. To discover patterns by generating association rules from the dataset using Association Rule Mining, Apriori Algorithms and to evaluate the rules from their confidence, support and lift.
- **Data description:**

The dataset represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes (UCI Machine Learning Repository: Diabetes 130-US hospitals for years 1999-2008 Data Set, n.d.). The original dataset contained 100,000 instances which was then randomly resampled to produce a dataset with 10176 instances using Weka filter for the purpose of this assignment, the full details of the resampling are given in the pre-processing section.

1. The problem domain:

World Health Organization defines diabetes as "a chronic, metabolic disease characterized by elevated levels of blood glucose (or blood sugar), which leads over time to serious damage to the heart, blood vessels, eyes, kidneys, and nerves." (WHO,2020). According to Harvard Medical School, there are two types of diabetes: type 1 diabetes (juvenile-onset diabetes) and type 2 diabetes (adult-onset diabetes). Type 1 diabetes is an autoimmune disease that occurs to children and teenagers, with no cure. On the other hand, adults with obesity or lack of physical activity are more vulnerable to type 2 diabetes (Harvard. 2020).

422 million people worldwide are suffering from diabetes, and the number of people with Type 2 Diabetes has been increasing sharply over the years. Rozalina McCoy, M.D., an internal medicine physician and endocrinologist at Mayo Clinic, says that adults with diabetes have more chances of hospitalization and unplanned readmission. Hospital readmission is defined as "an episode when a patient who had been discharged from a hospital is admitted again within a specified time interval". Centers for Medicare and Medicaid Services (CMS) introduced Hospital readmission rates, as part of the Patient Protection and Affordable Care Act (ACA) in the year of 2010 (Wikipedia, 2020). A new research published in the Journal of General Internal Medicine, states that severe dysglycemia (uncontrolled hyperglycemia - high blood sugar, or hypoglycemia - low blood sugar) causes unplanned readmission in diabetes patients (ScienceDaily, 2017). Readmission cause extreme financial burden on the diabetes patients and decrease the reputation of the hospitals. Predicting readmissions can help in strategy making, enhanced diagnosis, pattern prediction and thereby help the patients and hospitals alike.

2. Source of the data

The dataset was downloaded from UCI Machine Learning Repository. The data was submitted on behalf of the Virginia Commonwealth University Center for Clinical and

Translational Research, a recipient of the NIH CTSA grant UL1 TR00058 and a recipient of the CERNER data by John Clore, Krzysztof J. Cios , Jon DeShazo.
The Health Insurance Portability and Accountability Act of 1996 states that data should be deidentified (CDC, 2020) This data is a de-identified abstract of the Health Facts database (Cerner Corporation, Kansas City, MO) (UCI Machine Learning Repository: Diabetes 130-US hospitals for years 1999-2008 Data Set, n.d.). Health Facts database (Cerner Corporation, Kansas City, MO), is a national data warehouse that collects and keeps records of hospitals in the United States. Cerner Health Facts provides information and data on clinical, economic, process, functional, and satisfaction which are characterised as five health outcomes (Cerner Health Facts | SC CTSI, 2020). Health Facts is a voluntary program offered to organizations which use the Cerner Electronic Health Record System. The data was collected from participating institutions' electronic medical records and also has details of encounter (emergency, outpatient, and inpatient), medical specialty, demographics (age, sex, and race), diagnoses and in-hospital procedures, medications, number of lab procedures (Strack et al., 2014).

3. Agencies working with the data

Diabetes mellitus is described as the epidemic of the century in a paper published in the World Journal of Diabetes (Kharroubi, 2015). This clearly demonstrates the nature and the rate at which the diseases is affecting citizens worldwide. Thus, there are several agencies, interested in the diabetes data. The data was collected by CERNER Corporation and was made available to the public and other institutions after de-identification
The dataset has become very popular ever since and is used by several agencies and also, studies utilising this data are published in several acclaimed and internationally recognised journals, some of them are as follows:

3.1. BioMed Research International

A research article named *Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records* by Beata Strack, Jonathan P. DeShazo, Chris Gennings published on BioMed Research International used the CERNER Health Facts Diabetes Data for their study (Strack et al., 2014).

3.2. Procedia- Elseiver

The 5th International Symposium on Emerging Information, Communication and Networks (EICN-2018)-Predicting Hospital Readmission among Diabetics using Deep Learning used the dataset to predict the readmission rate using Deep Learning and Neural Networks (Hammoudeh, Al-Naymat, Ghannam and Obied, 2018).

4. Intended use of the data

The data was collected and prepared to understand the factors/attributes that are associated with the readmission and several other possibilities that pertain to diabetes patients. Care and treatment of diabetes patients during hospitalization is an extremely important factor that can affect their health in the future after discharge from the hospital. According to a study funded by National Institutes of Health, U.S. Department of Health & Human Services, the investigators identified more than 11,000 young people under the age of 20 who were diagnosed with type 1 diabetes and 2,800 young people aged 10 to 19 with type 2 diabetes. The researchers analysed differences between different genders also. For type 1 diabetes, the rate rose more among males (2.2% annual increase) than females (1.4%). For type 2 diabetes, the rate rose more among females (6.2%) than males (3.7%). There were significant differences in the percentage increase among different ethnic groups (National Institute of Health, 2017).
This study reveals the importance of factors such as age, gender, race etc in understanding the effect of diabetes and their chances of being readmitted. Thus, data about patients being admitted to different hospitals in the U.S with diabetes and related diseases is extremely important. Understanding the reason behind hospital readmissions improves the patient outcomes and quality of care, with lowered health expenses for patients

coming from various backgrounds. New interventions with more data mining and analytics can improve outcomes for patients, resulting in less readmissions. (CDC, 2020) The data can be used by researchers to identify patterns, to understand the use of a drug in the treatment of the diabetes and to understand the effect of demographics in a larger scale (Cerner Health Facts | SC CTSI, 2020)

5. The attribute types of the data

The detailed description of attributes is taken from Table 1 Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore, "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records," BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014.

Feature	Type	Description and values
Encounter ID	Numeric	Unique identifier of an encounter
Patient number	Numeric	Unique identifier of a patient
Race	Nominal	Values: Caucasian, Asian, African American, Hispanic, and other
Gender	Nominal	Values: male, female, and unknown/invalid
Age	Nominal	Age of the patients
Weight	Nominal	Weight in pounds.
Admission type	Nominal	Integer identifier corresponding to 8 distinct values, for example, emergency, urgent, elective, newborn, and not available 1 Emergency 2 Urgent 3 Elective 4 Newborn 5 Not Available 6 NULL 7 Trauma Center 8 Not Mapped
Discharge disposition	Nominal	Integer identifier corresponding to 24 distinct values, for example, discharged to home, expired, and not available. The mapping is available in IDS_mapping.csv. (Some ids may not be present in the data after resampling)
Admission source	Nominal	Integer identifier corresponding to 14 distinct values, for example, physician referral, emergency room, and transfer from a hospital. The mapping is available in IDS_mapping.csv. (Some ids may not be present in the data after resampling)
Time in hospital	Numeric	Integer number of days between admission and discharge
Payer code	Nominal	Integer identifier corresponding to 16 distinct values, for example, Blue Cross/Blue Shield, Medicare, and self-pay
Medical specialty	Nominal	Integer identifier of a specialty of the admitting physician, corresponding to 51 distinct values, for example, cardiology, internal medicine, family/general practice, and surgeon
Number of lab procedures	Numeric	Number of lab tests performed during the encounte

Number of procedures	Numeric	Number of procedures (other than lab tests) performed during the encounter
Number of medications	Numeric	Number of distinct generic names administered during the encounter
Number of outpatient visits	Numeric	Number of outpatient visits of the patient in the year preceding the encounter
Number of emergency visits	Numeric	Number of emergency visits of the patient in the year preceding the encounter
Number of inpatient visits	Numeric	Number of inpatient visits of the patient in the year preceding the encounter
Diagnosis 1	Nominal	The primary diagnosis (coded as first three digits of ICD9);
Diagnosis 2	Nominal	Secondary diagnosis (coded as first three digits of ICD9);
Diagnosis 3	Nominal	Additional secondary diagnosis (coded as first three digits of ICD9);
Number of diagnoses	Numeric	Number of diagnoses entered to the system
Glucose serum test result	Nominal	Indicates the range of the result or if the test was not taken. Values: “>200,” “>300,” “normal,” and “none” if not measured
A1c test result	Nominal	Indicates the range of the result or if the test was not taken. Values: “>8” if the result was greater than 8%, “>7” if the result was greater than 7% but less than 8%, “normal” if the result was less than 7%, and “none” if not measured.
Change of medications	Nominal	Indicates if there was a change in diabetic medications (either dosage or generic name). Values: “change” and “no change”
Diabetes medications	Nominal	Indicates if there was any diabetic medication prescribed. Values: “yes” and “no”.
24 features for medications	Nominal	For the generic names: metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, sitagliptin, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, and metformin-pioglitazone, the feature indicates whether the drug was prescribed or there was a change in the dosage. Values: “up” if the dosage was increased during the encounter, “down” if the dosage was decreased, “steady” if the dosage did not change, and “no” if the drug was not prescribed
Readmitted	Nominal	Days to inpatient readmission. Values: “<30” if the patient was readmitted in less than 30 days, “>30” if the patient was readmitted in more than 30 days, and “No” for no record of readmission.

The original dataset as obtained from UCI Machine Learning Repository contained 101766 instances.

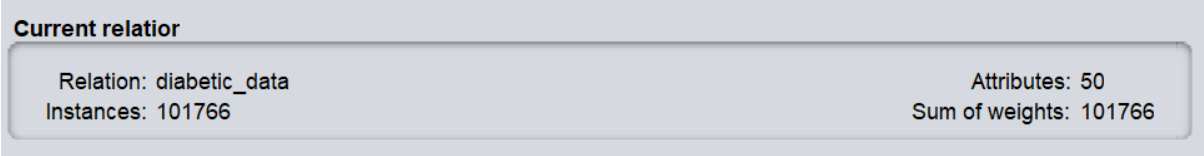
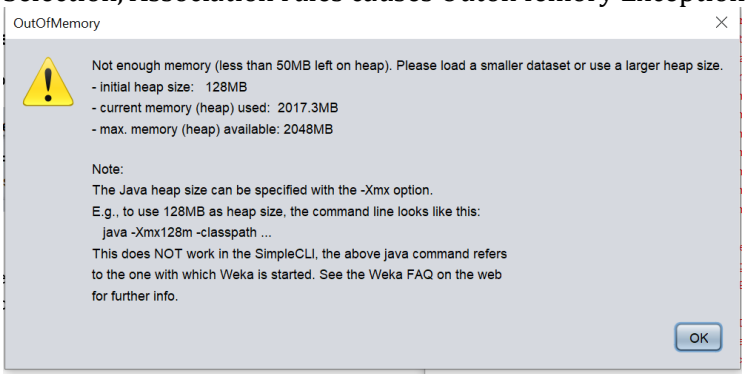


Figure 1 Original data- summary

Processing of large data with 101766 instances and 50 attributes, particularly Feature Selection, Association rules causes OutofMemory Exception in Weka.



This exception is caused even after the heap size was increased. Thus, the data is resampled to reduce the size of the data, for the purpose of this assignment. Resampling is done randomly and since this is a de-identified dataset, resampling doesn't affect the quality of the data. The details of resampling are:

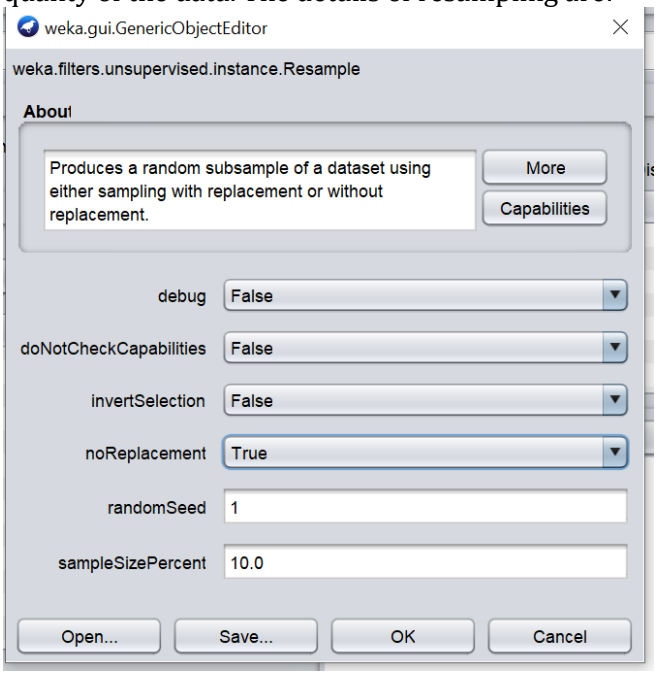


Figure 2 Resample Generic Object Editor

The data is resampled and number of instances are reduced to 10176 instances. The resampling stage ensures efficient and quick processing without much effect on the performance of the algorithm. noReplacement option is set to true, since for machine learning we don't want duplicate data and invertSelection is also set to false as this is a random resampling.

After the resampling, the dataset summary

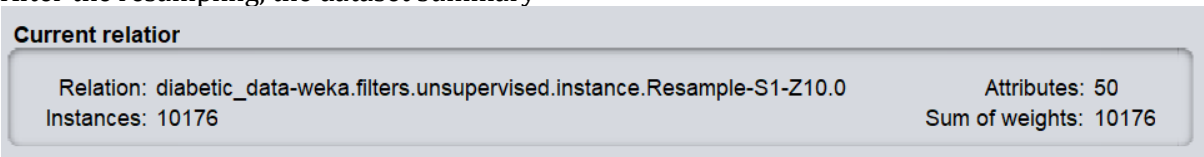


Figure 3 Resampled dataset

For the purpose of classification, this dataset will be used. It is saved as diabetic_dataset.arff.

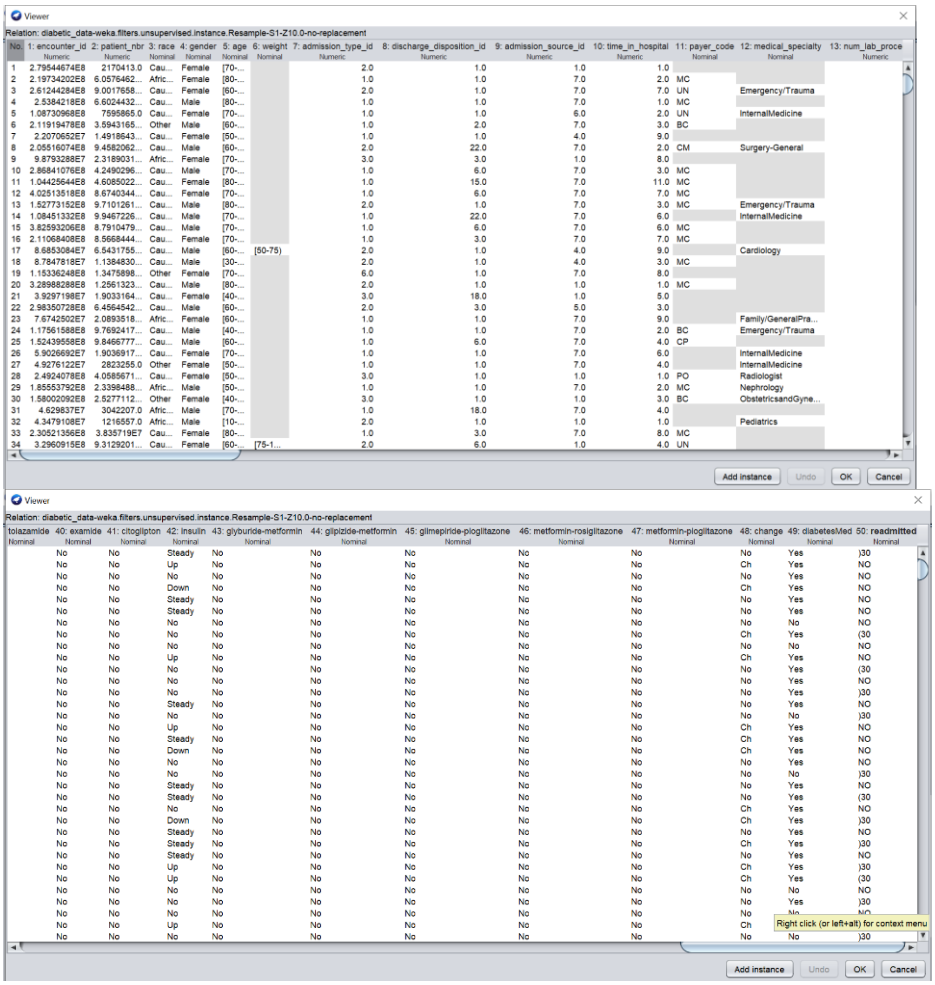


Figure 4 Dataset View

Screenshots of data summary and graphs through Weka

The dataset is opened in Weka Explorer. The data summary and visualizations of each attribute is obtained using Explorer. Some screenshots of the important attributes, univariate attribute description, their summaries, number of missing values, unique values and their visualizations are given below. Since there are 50 attributes, not all of their summaries in Weka are presented below as most of the attributes especially some of the medications are not of use for the objective of this assignment. All visualizations are performed according to class readmitted, since this class will be used in both clustering and classification.

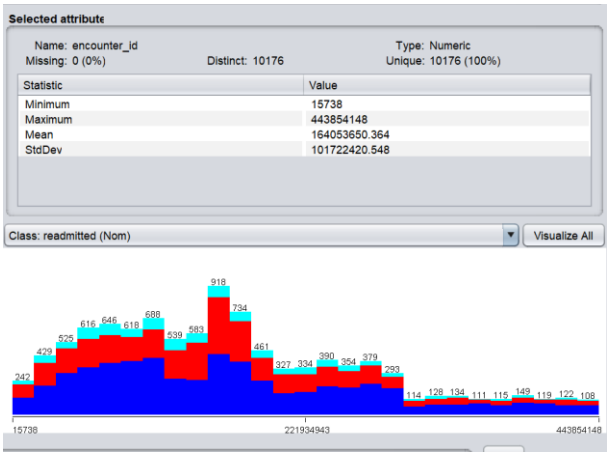


Figure 5 encounter_id

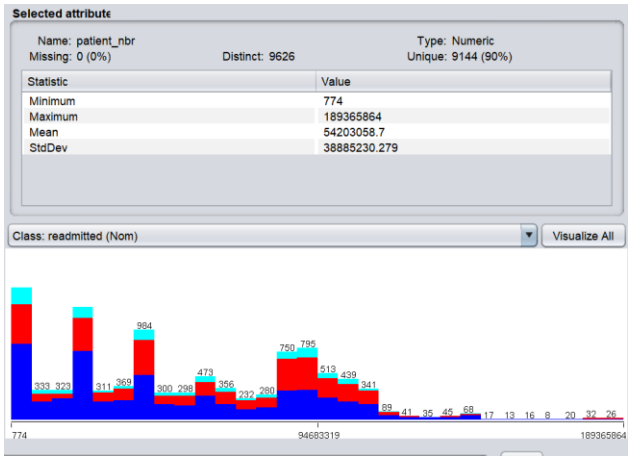


Figure 6 patient_nbr

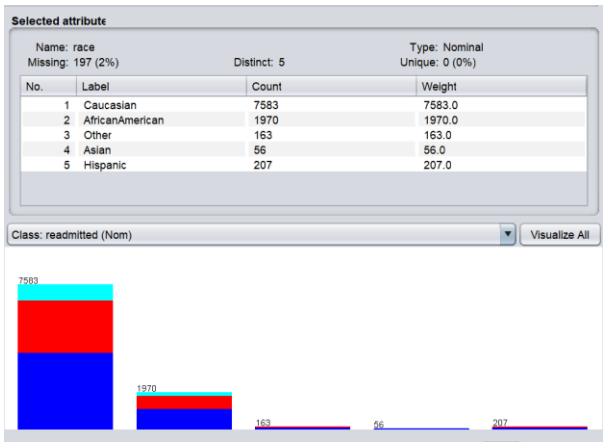


Figure 7 race

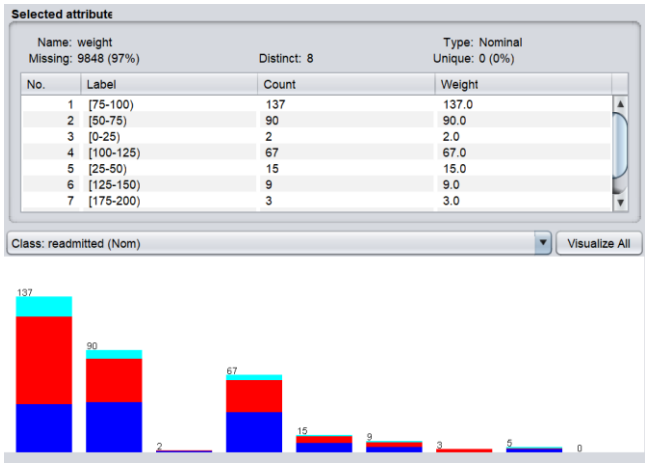


Figure 8 age

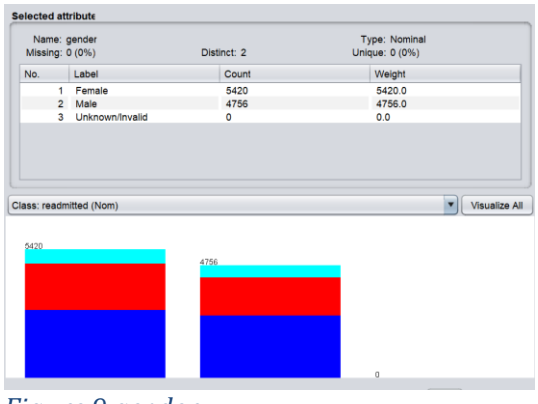


Figure 9 gender

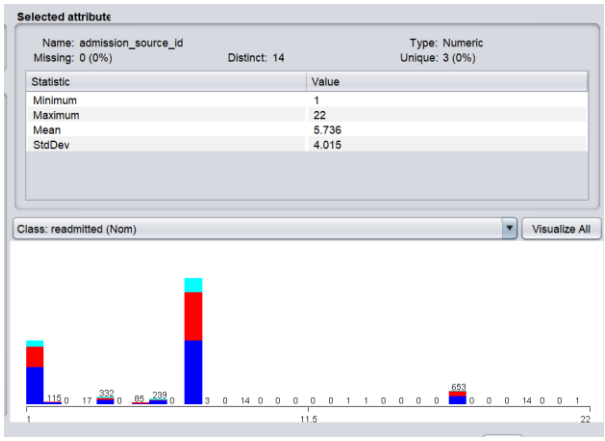


Figure 10 admission_ source_ id

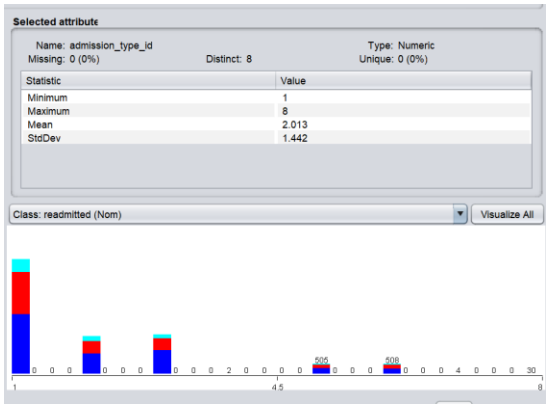


Figure 11 admission_type_id

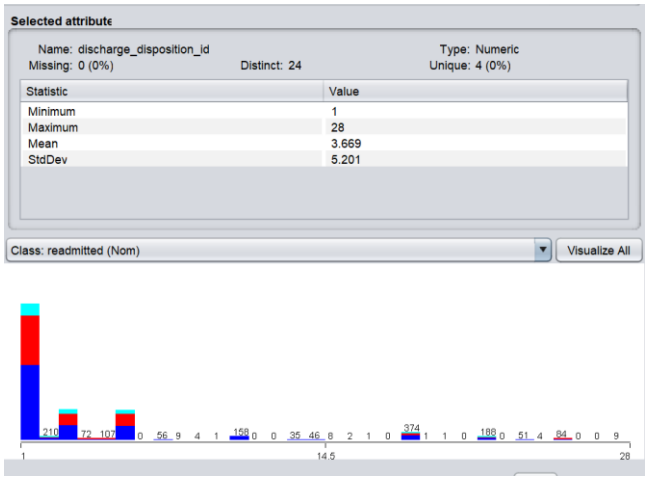


Figure 12 discharge_disposition_id

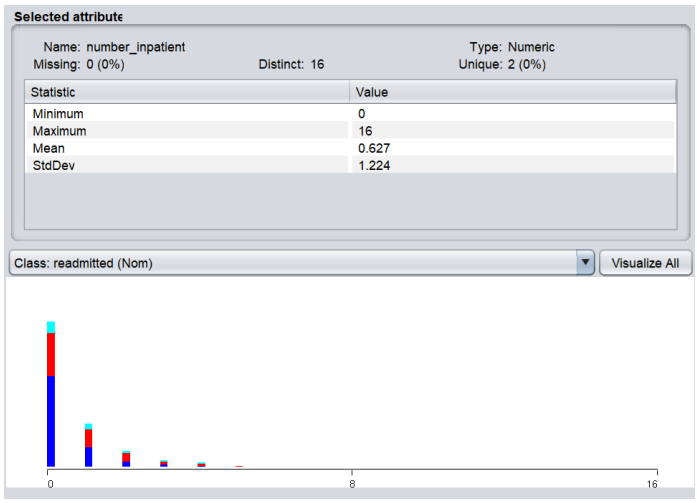


Figure 13 number_inpatient

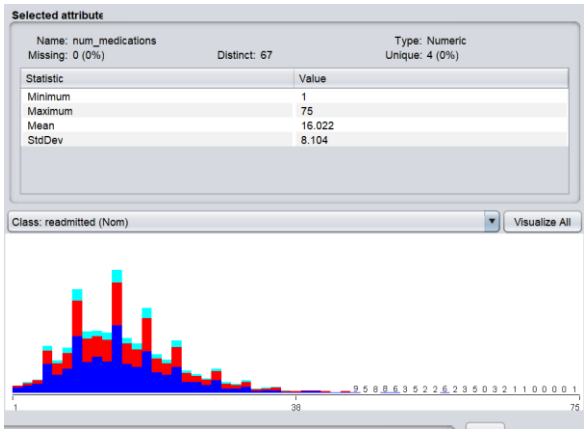


Figure 14 num_medications

Figure 15 num_procedures

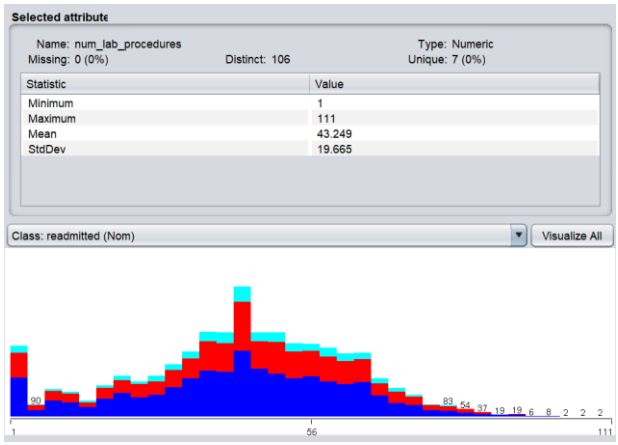


Figure 16 num_lab_procedures

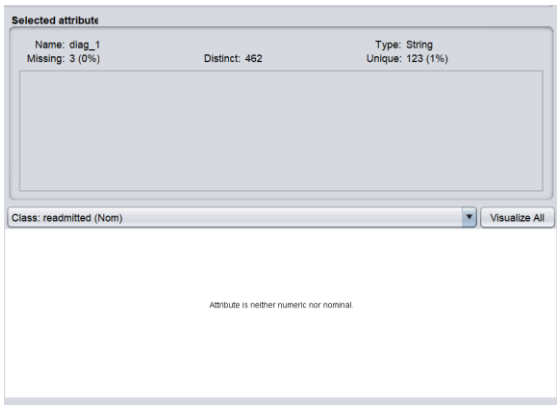


Figure 17 diag_1

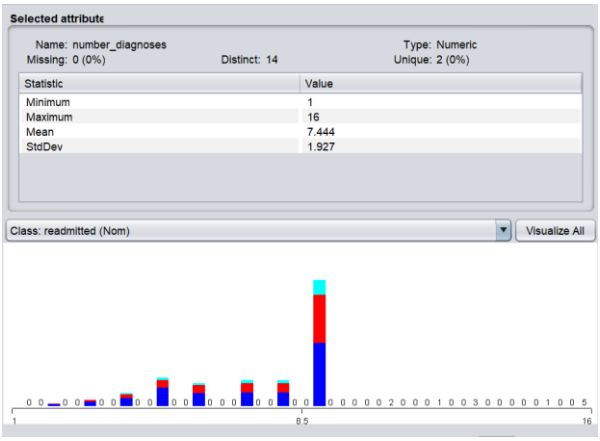


Figure 18 number_diagnoses

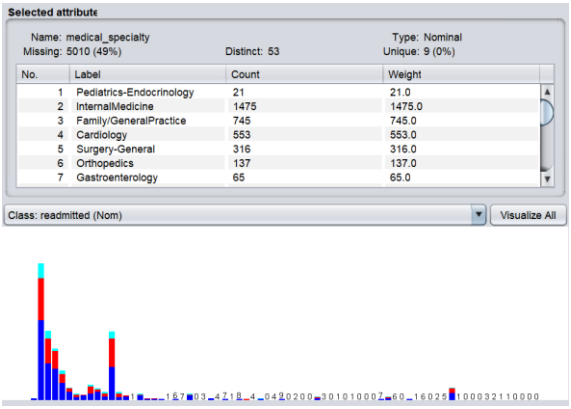


Figure 19 medical_speciality

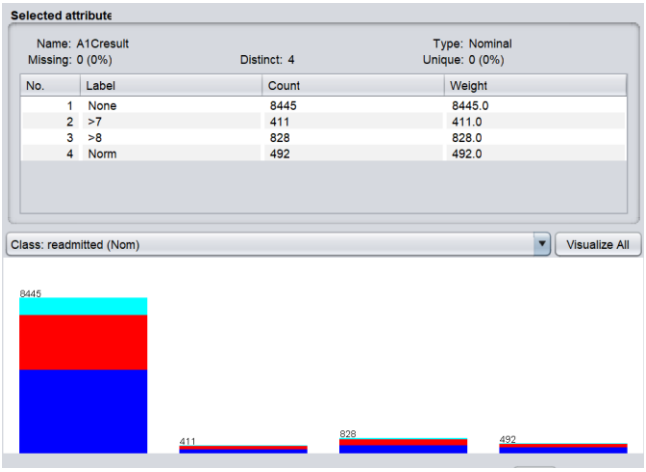


Figure 20 A1cresult

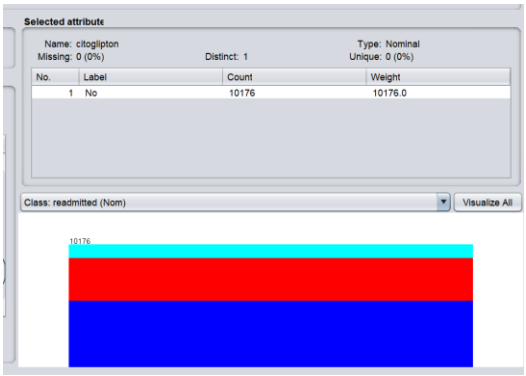


Figure 21 citoglipton

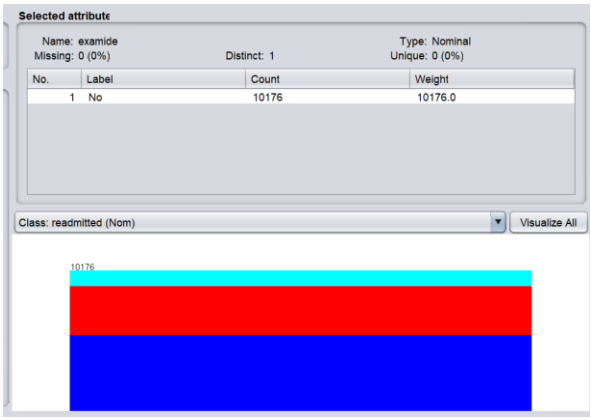


Figure 22 examide

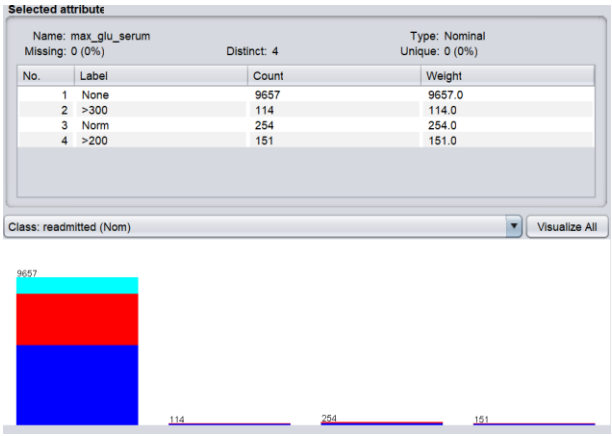


Figure 23 max_glu_serum

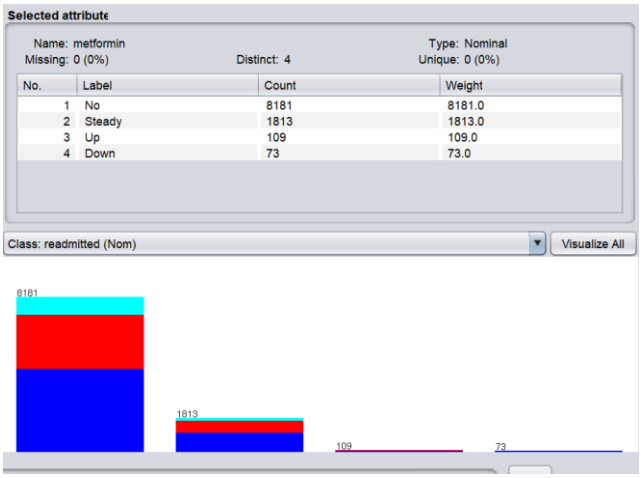


Figure 24 metformin

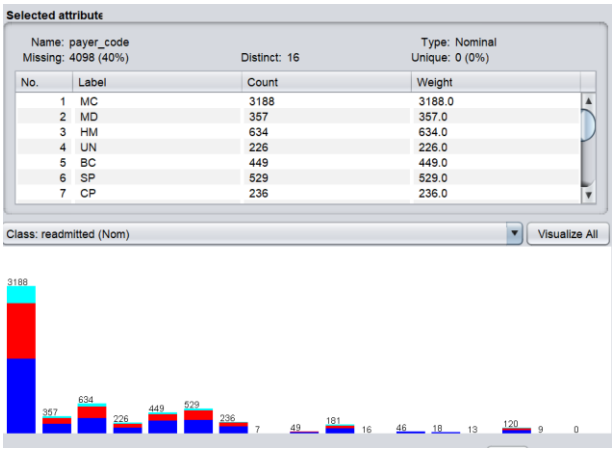


Figure 25 payer_code

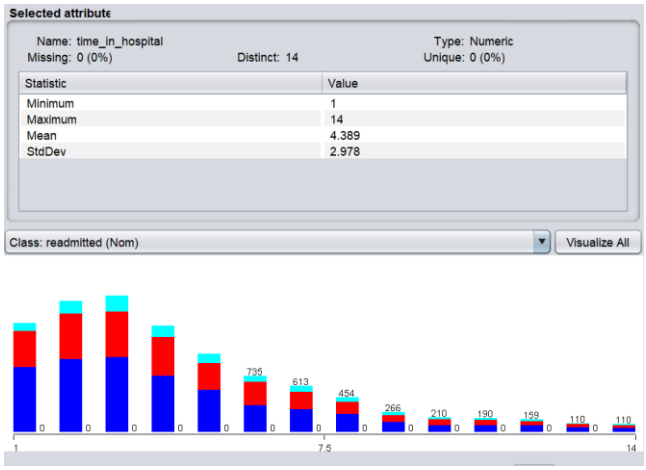


Figure 26 time_in_hospital

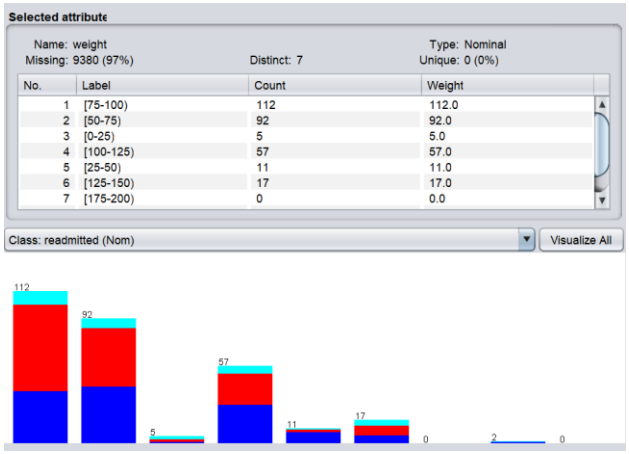


Figure 27 weight

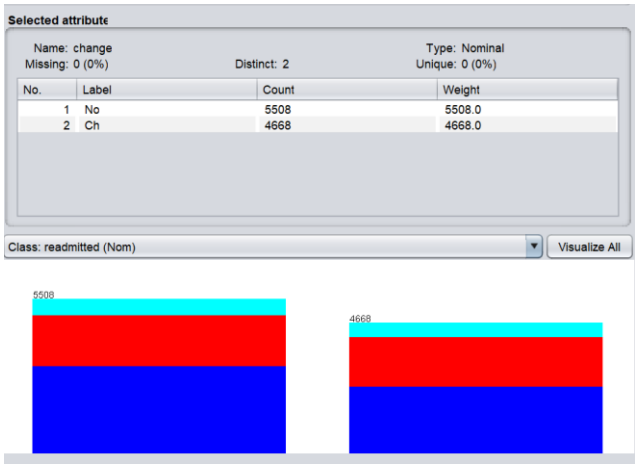


Figure 28 change

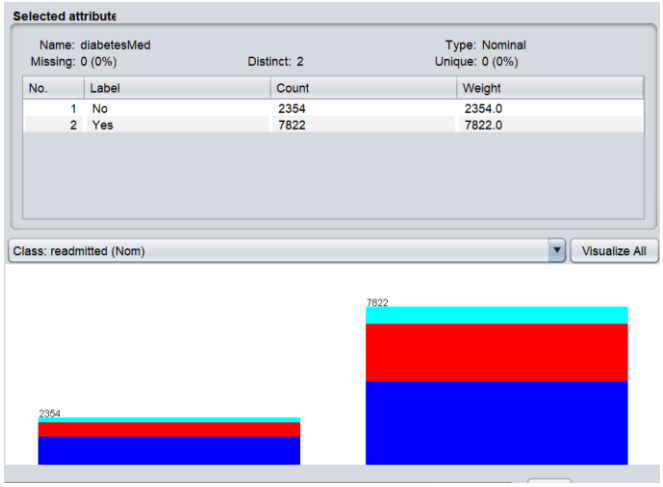


Figure 29 diabetesMed

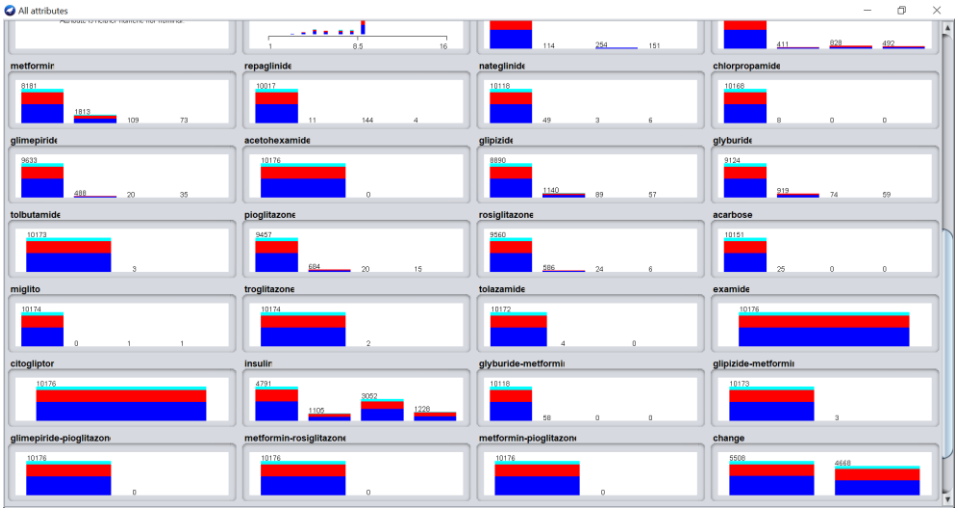


Figure 30 All visualizations

Readmitted class

This is the class we want to predict using classifers.

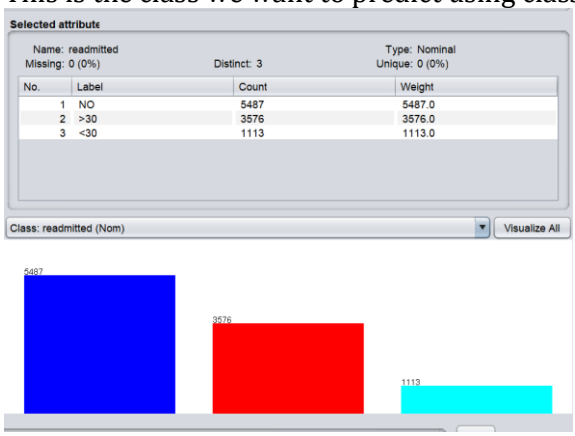


Figure 31 readmitted

Observations from visualizing the attributes

1. There are a mix of numeric, nominal and string variables
2. encounter_id, patient_nbr have all unique values
3. weight has 97% missing values, payer_code has 40% missing values, medical_speciality has 50% missing values
4. examide and citoglipton have just one value for all instances. i.e, nominal value No
5. All medications except insulin have 70% or more of nominal value No which means these medicines were taken by or prescribed to only very few patients.
6. Readmitted have values: "<30" if the patient was readmitted in less than 30 days, ">30" if the patient was readmitted in more than 30 days, and "No" for no record of readmission.

- **Summary of Findings**:

The objective of this data mining process is to predict the risk of readmission of diabetes patients into the 130 hospitals in the U.S. Effectiveness of any machine learning algorithm relies on the efficient preprocessing techniques. Preprocessing techniques such as feature encoding, feature selection, replacement of missing values, conversion of datatypes, outliers, normalization, discretization, class imbalance etc. were analysed for effectiveness on this dataset.

1. **Preprocessing**
   - The readmission feature is converted to 1 for >30 and <30 and 0 for 'NO'.
   - 'encounder_id' and patient_nbr' contains all 10176 unique values, thus they are not useful for classification. 'weight' feature contains 97% missing values. It is thus not useful for data mining and is removed from the dataset.

'payer_code' has 40% missing values, 'medical_speciality' has 50% missing values. This is also removed after applying feature selection algorithms. 'examide' and 'citoglipton' contains 'No' for all instances and hence removed. Medications other than insulin, nateglinide is removed because their effect on readmission is not significant. Feature selection techniques such as ClassifierAttributeEval and CorrelationAttributeEval are applied to select the features for the J48 classification algorithm. 18 features are selected. The selected features are gender, age, admission_type_id, time_in_hospital, num_lab_procedures, num_procedures, num_medications, number_outpatient, number_emergency, number_inpatient, max_gluserum, A1Cresult, metformin, nateglinide, insulin, change, diabetesmed, readmitted.

- Normalization was not performed since the most of the attributes in the dataset are distinct ids or nominal values.
- Age is already discretized into categories from [0-10] to [10-100].
- Data Type Conversion is performed using NumericToNominal and StringToNominal to convert all the 18 attributes to Nominal values.
- There are no outliers in the dataset.
- Applying SMOTE filter increased the performance of classification algorithm J48 in this dataset dramatically. This is because the SMOTE filter reduces class imbalance. 6 nearest neighbours were used.
- All missing values are replaced with modes and means accordingly using ReplaceMissingValues Filter. Preprocessed file is saved as 'dataset.arff' .

2. Splitting

- The dataset is split into training and test dataset using 9:1 ratio and Resample filter. These files are stored in 'trainigSet.arff' and 'testingSet.arff'.

3. Classification -J48

- Three different experiments were carried out by supplying the test set and by varying the hyperparameters to see the performance of each model, the detailed description of each model and its results are included in Experiments section below.
- In first experiment, confidence factor 0.25 and minNumObj is set as 2, with pruned tree and 68.729% of instances were correctly classified by this model. Since this is a complex dataset in healthcare, predicting patients with or without readmission risks with 68% accuracy is a good choice. Pruned tree increased the accuracy. The model used a pruned tree because pruning mostly reduces the complexity of the final classifier, reduce overfitting to increase predictive accuracy(Wikipedia Decision Tree Pruning, 2020). From, the confusion matrix it is understood that the model is more good at identifying the patients with risk of readmission (b=1) – correctly classified 830 out of 948 and only incorrectly classified it as 0 only 118 instances out of 948, whereas in identifying patients with no risk of readmission (a=0), it only classified 192 correctly and 347 were classified incorrectly. From the visualization of the tree, it is evident that the tree gained more information from the number_inpatient attribute and further branching was made on this attribute. **number_inpatient** is the **best feature** on which the data is further split.

- In second experiment, the parameters are varied to tune the model and it increased the model accuracy by 1.0087 % to reach 69.7377%. However Root Mean Squared Error(RMSE) increased to 0.46 from 0.45.The parameters that had a effect on this are explained in detail in the Experiments section. From, the confusion matrix it is understood that the model is still not that good at identifying the patients with no risk of

readmission (a=0) as compared to patients with risk of readmission (b=1)– correctly classified 223 out of 539 and only 316 instances out of 539 incorrectly classified it as 1, whereas in identifying patients with risk of readmission (b=1), it classified 814 instances correctly and 134 were classified incorrectly.From the visualization of the tree, it is evident that the tree began from the **discharge_disposition attribute** which is an attribute that corresponds to different ids such as discharged to home. The tree has become quite large due to binary splitting on nominal variables. However, this parameter increased the number of correctly classified instances.

- **Highest accuracy achieved with J48 is 70.0754 in third experiment**. This is achieved by using a pruned tree with minimum number of instances per leaf(minNumObj) set to 3. It is observed that pruning mostly reduces the complexity of the final classifier, reduce overfitting to increase predictive accuracy (Wikipedia Decision Tree Pruning, 2015). confidenceFactor is set to 0.25 for pruning (smaller values incur more pruning). binarySplits is set to true to use binary splits on nominal attributes when building the trees, this however increased the tree size. SubTreeRaising operation is also set to False. Increasing minNumObj reduces the tree size and improved the accuracy. reducedErrorPruning is also set to True, in order for improved pruning. In-detail findings from two other experiments are included in the respective sections of this document.
- Tree size increased when binarySplts was set to True, however, this also increased the number of correctly classified instances, it performs binary splitting on nominal variables.
- In Experimenter, Using AttributeSelectedClassifier an accuracy of **95.56%** was achieved with Train-Test Order Preserved with 9:1 ratio in dataset.arff**.** This model used BestFirst search and Ranker evaluator. AdaBoostM1 was the least peforming ensemble method here and accuracy of 86.68%. Bagging achieved an accuracy of 91.79%. These ensemble methods aims to achieve low bias and variance, and was almost successful.
- The result buffers and models of all the J48 experiments are stored in 'J48 RESULT BUFFERS AND MODELS' with appropriate and self-explanatory names.
- 'Experiment1J48tree.arff','Experiment2J48tree.arff', Experiment3J48tree.arff' are the files that contain J48 visualization with predicted class.

4. Association Rule Mining

- Apriori Algorithm with different parameters was used to perform Association Rule Mining. A total of three experiments were carried out on dataset. arff.
- In the first experiment, The lowerBoundMinSupport=0.1 and upperBoundMinSupport=1.0 are set. outputItemSets are set as True to show all frequent itemsets. Rules generated are ranked by metricType (default Confidence). Only rules with score higher than minMetric (default 0.9 for Confidence) are considered and delivered as the output. The algorithm started with MinSupport as 100% and stopped at 80% after running 4 times. Most patients with number of emergency visits are 0 and with Glucose Serum test is not conducted, have nateglinide not administered to them with 0.99 confidence. max_glu_serum=None

happened with conjunction to number_emergency=0 for 8651 instances, that is the support or coverage.

- In the second experiment, With treatZeroAsMissing set to True, another 10 rules have been generated. This attribute helped to identify more rules, otherwise the algorithm was focused on the 0 values of the nominal attributes. One of the rules show 100% of Males with change of medicine and a risk of readmission are prescribed diabetes Medications 2135 instances. That is the support. Confidence is 1 (2135/2135). This is obviously logical since change of medication indicates a prescribed medication. All rules have 100% confidence
- In experiment 3, With lift metric type and minMetric as 1.1, some other rules are generated, a rule show that Males who are prescribed diabetes Medications have a change of medication 62% of times. Lift is 1.32, leverage is 0.05 and conviction Is 1.39
- **The rules are mainly focused on 'gender', 'diabetesMed', 'max_glu_serum','num_emergency' and 'change'. Most of the rules generated have high confidence and support.**
- The result buffers of association rule experiments are stored in 'APRIORI RESULT BUFFERS'.
- The results are stored as text files in 'Apriori_Experiment1_associationrule.txt', 'Apriori_Experiment2_associationrule.txt', 'Apriori_Experiment3_associationrule.txt'

2. Preprocessing – 10%

Efficient data mining relies on effective preprocessing as the data may not always be directly useful for the specific objectives. Preprocessing is time consuming but highly rewarding, as it enhances the performance of classifiers.

1. Feature Encoding

Readmitted have values: "<30" if the patient was readmitted in less than 30 days, ">30" if the patient was readmitted in more than 30 days, and "No" for no record of readmission. Since the objective of this classification is to predict the readmission, it is useful to convert the readmission feature to 1 for >30 and <30 and 0 for 'NO'. This step was performed in Excel before all other preprocessing steps.

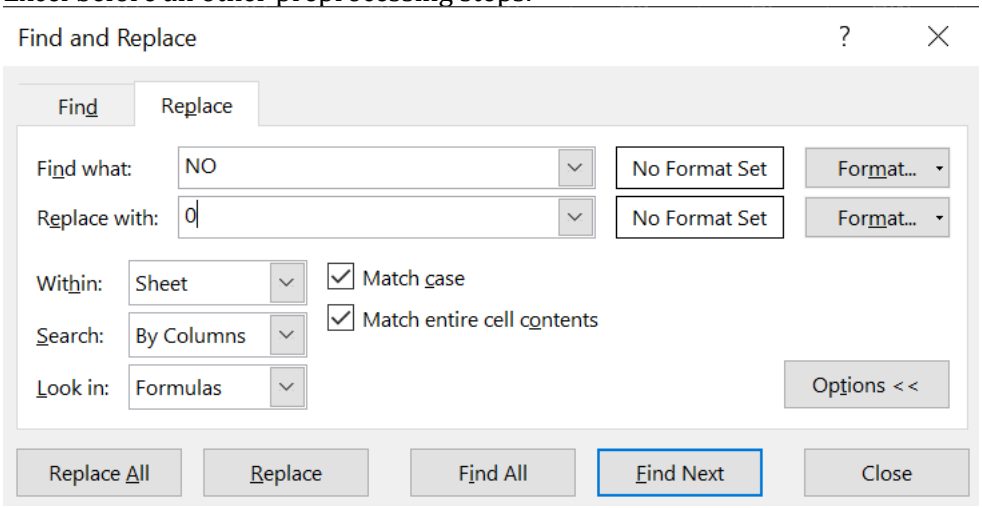


Figure 32 Feature Encoding

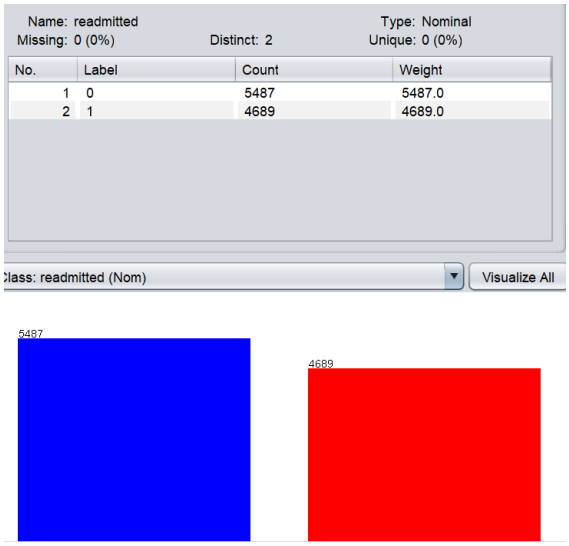


Figure 33 readmitted after encoding

- 0 - No risk of readmission
- 1 -Risk of readmission within or after 30 days

2. Selecting or filtering the attributes

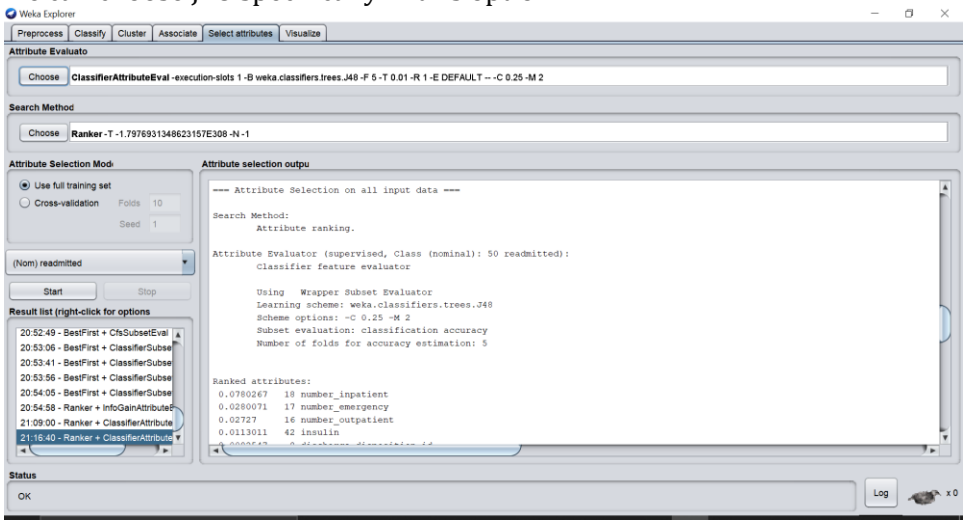
Feature selection is an extremely important step in any data mining process. It not only reduces the dimensions, but also improves the performance of the algorithms to a great extent. The dataset contains 50 attributes which is a big number, thus removing or selecting certain features based on their importance and correlation, increases the ease for processing, improves accuracy and overall provides better results. To select the predictor variables for readmitted class, it is essential to remove the features that are obviously futile for the task.

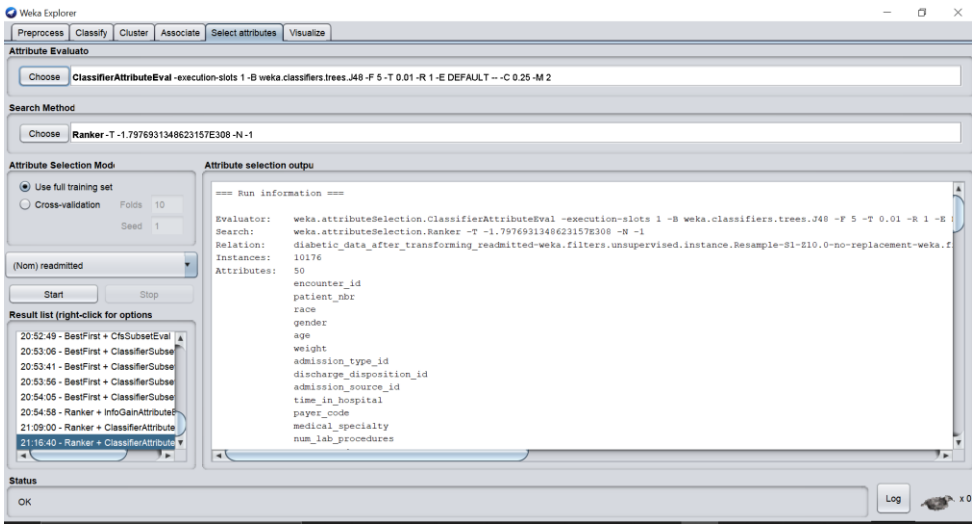
1. 'weight' feature contains 97% missing values. It is thus not useful for data mining and is removed from the dataset. payer_code has 40% missing values, medical_speciality has 50% missing values. This is also removed after applying feature selection algorithms.
2. 'examide' and 'citoglipton' contains 'No' for all instances. They have just 1 distinct and unique value, and hence there will only be little to no effect of these features on the chances of readmission. Hence, these features are removed.
3. Medications other than insulin, nateglinide can be removed because their effect on readmission is not significant.

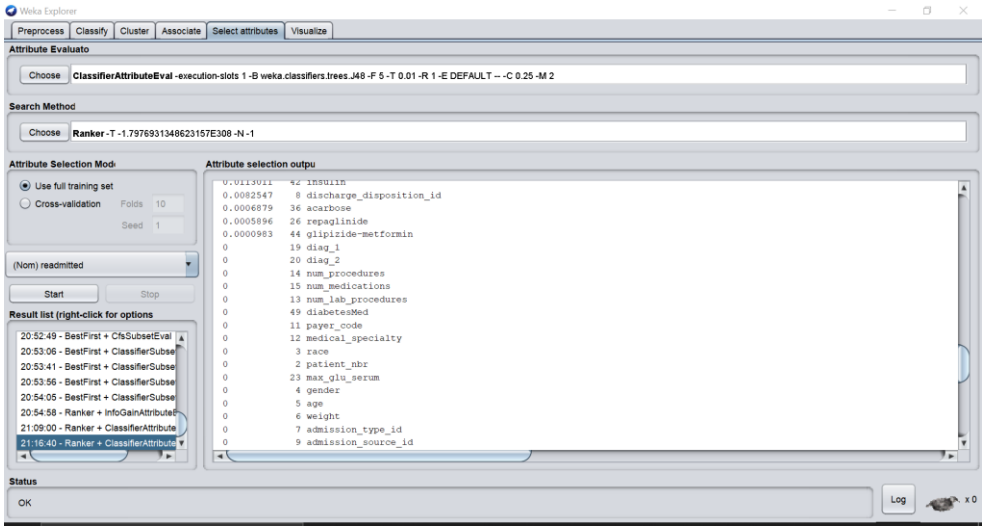
Various feature selection methods are also used to select the best features.

ClassifierAttributeEval with Ranker

It evaluates the worth of an attribute by using a user-specified classifier. This is very useful because we are interested in doing J48 for the purpose of this assignment. Hence we can choose J48 specifically in this option.







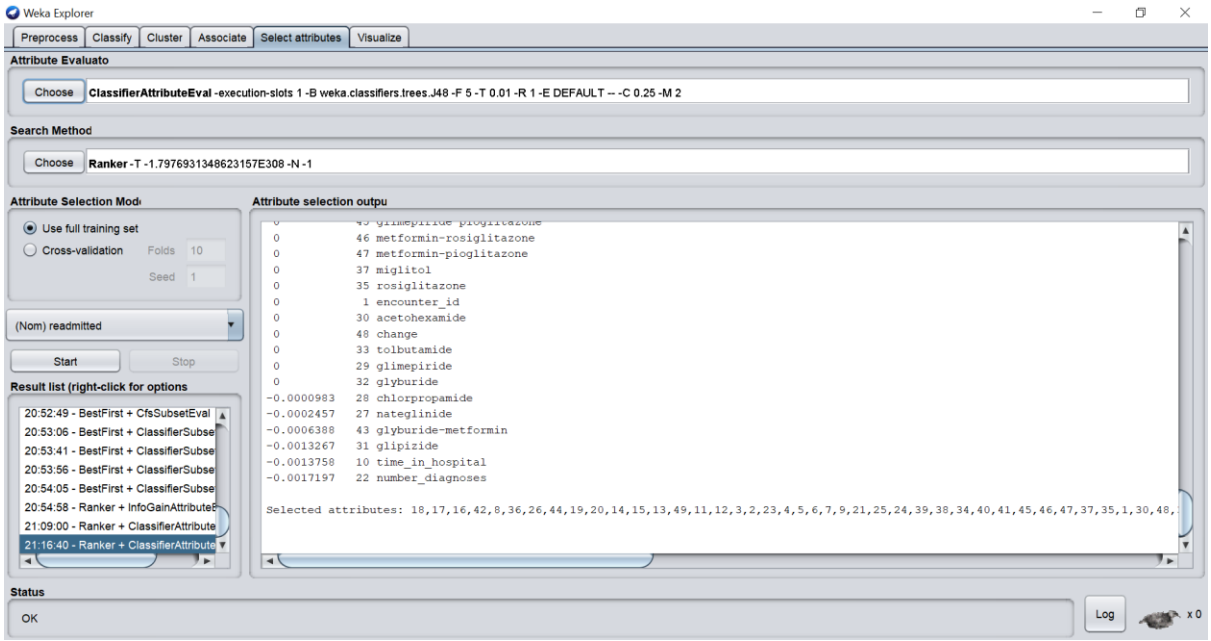


Figure 34 ClassifierAttributeEval with Ranker

CorrelationAttributeEval with Ranker

It evaluates the worth of an attribute by measuring the correlation (Pearson's) between it and the class. Correlation is a useful statistic measure, especially in this dataset with 50 attributes. Ranker is very useful because it gives the rank of the features. Features can be selected depending upon this information.

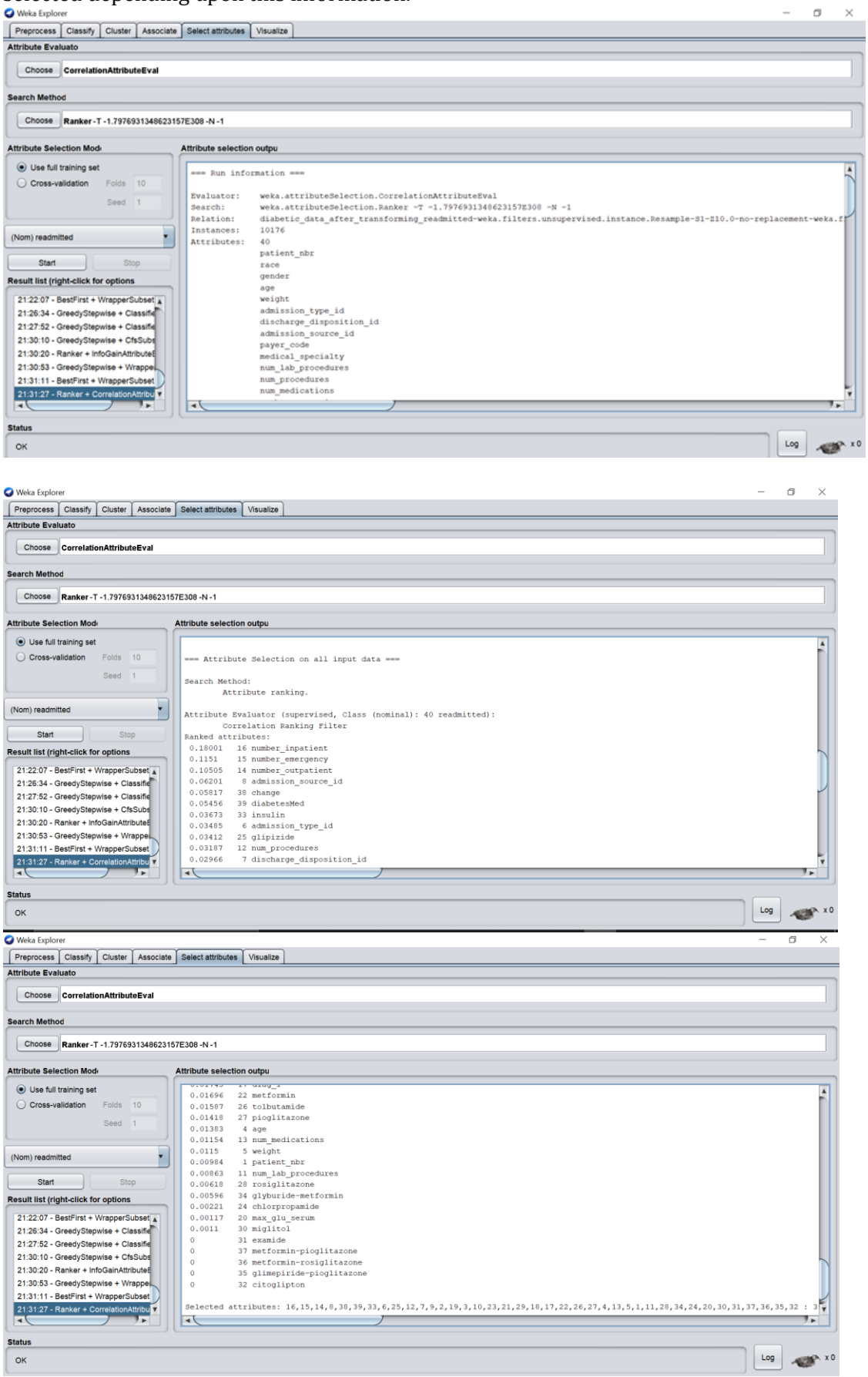


Figure 35 CorrelationAttributeEval with Ranker

3. Visualize

Weka's Visualize tab is also used to identify the attributes that are useful for classification.

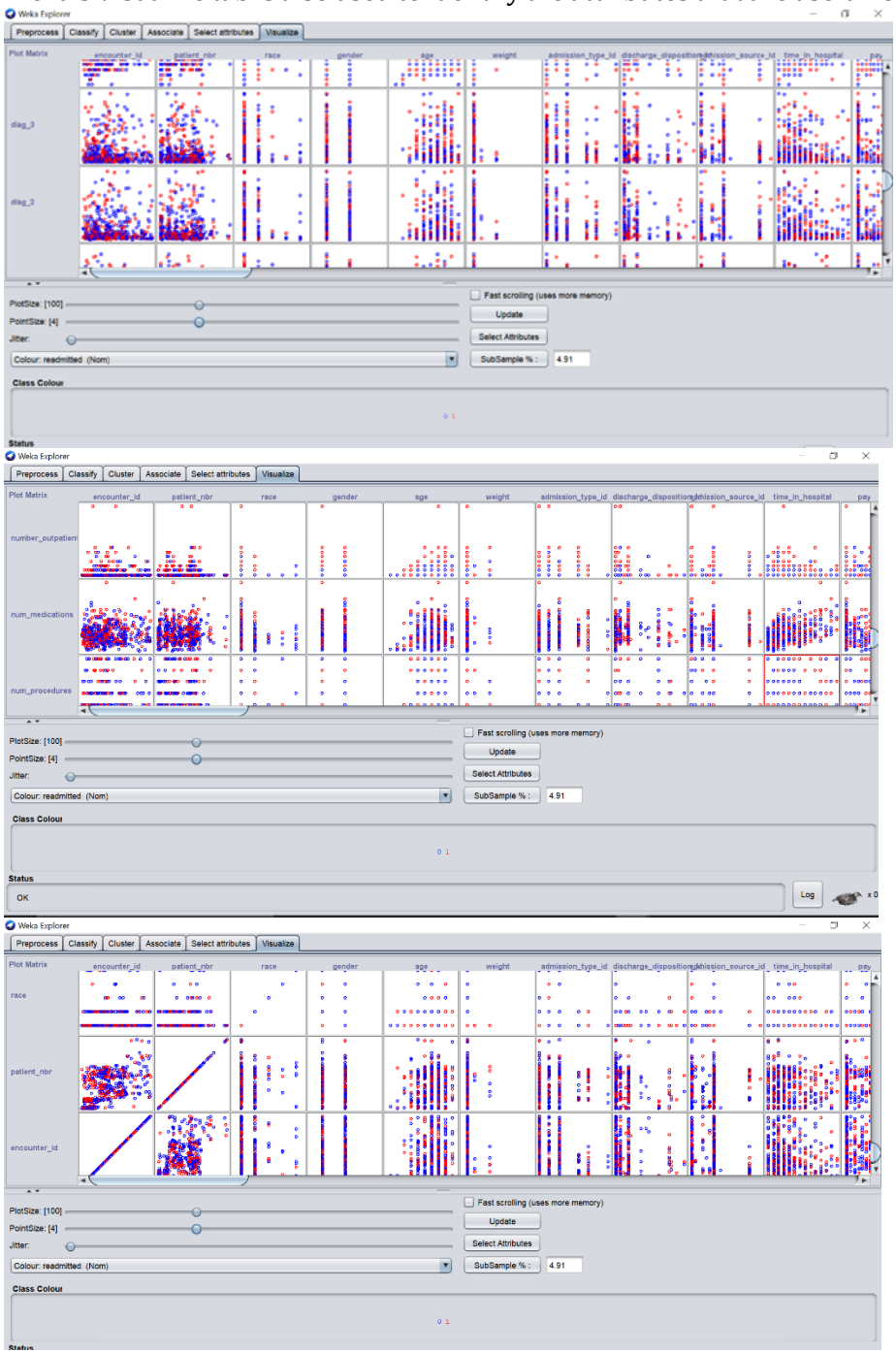


Figure 36 Visualize tab

After feature selection and visualization, the features selected depending upon the number of missing values, unique values, spread of attributes, algorithms like CorrelationAttributeEval and ClassifierAttributeEval are:

1. gender
2. age
3. admission_type_id
4. time_in_hospital
5. num_lab_procedures
6. num_procedures
7. num_medications
8. number_outpatient
9. number_emergency
10. number_inpatient
11. max_gluserum

12. A1Cresult
13. metformin
14. nateglinide
15. insulin
16. change
17. diabetesmed
18. readmitted

4. Normalization

Normalization can be performed using Weka's normalize filter. This ensures that numerical values are scaled by removing the different min and max values and scaling them down to a standard range which makes certain machine learning algorithms work better (Dr. Abubakr Siddig- Datasets,EDA and altering data structure lecture , 2020).

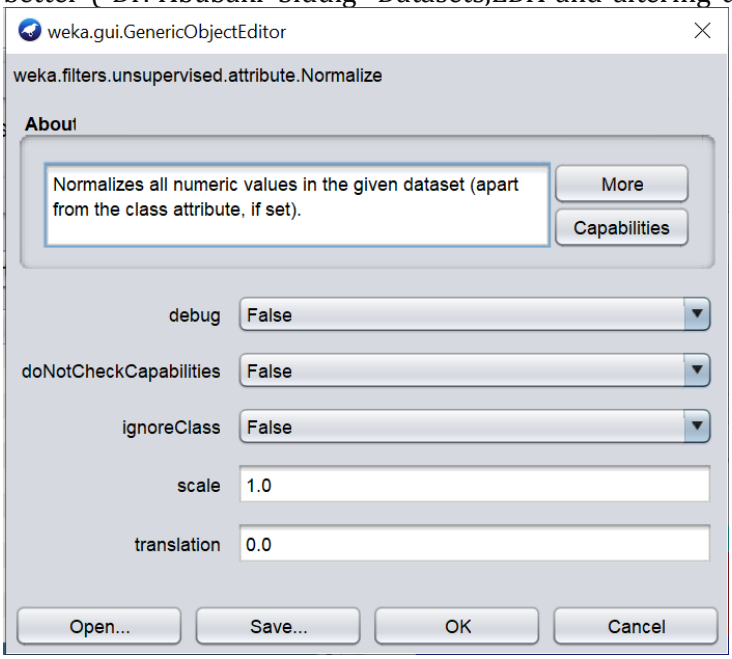


Figure 37 Normalization

However, for this dataset, most of the selected features are ids or nominal values. Applying normalization was not beneficial when tested using 10-fold crossvalidation. Hence the preprocessed data is not normalized.

5. Data type conversion

Data type conversion is necessary for certain algorithms like J48. J48 requires all attributes to be nominal. Hence conversion is performed using NumericToNominal, StringToNominal filters.

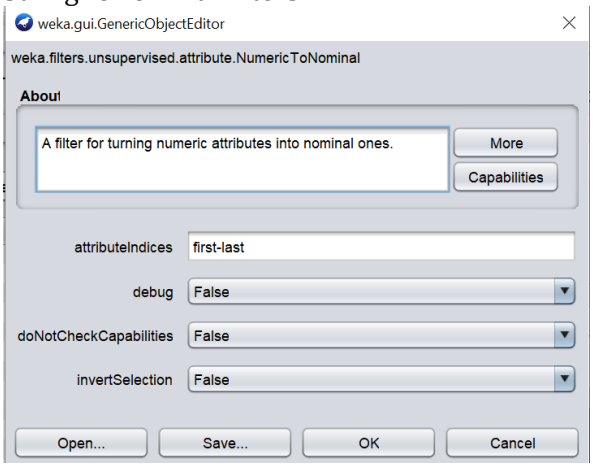


Figure 38 NumericToNominal Filter

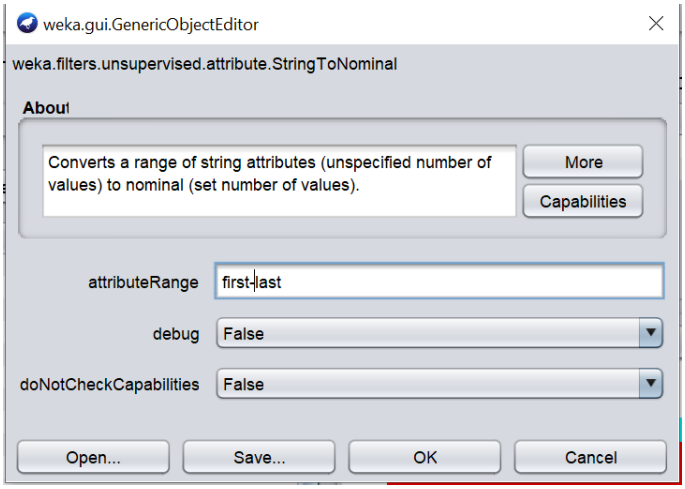


Figure 39 StringToNominal Filter

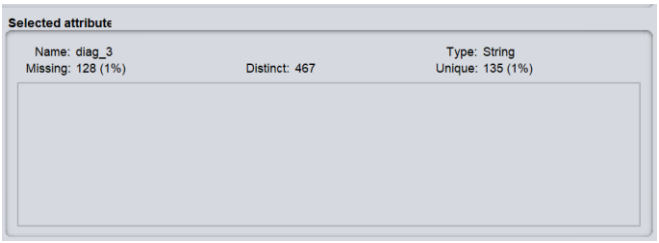


Figure 40 diag_1 before conversion

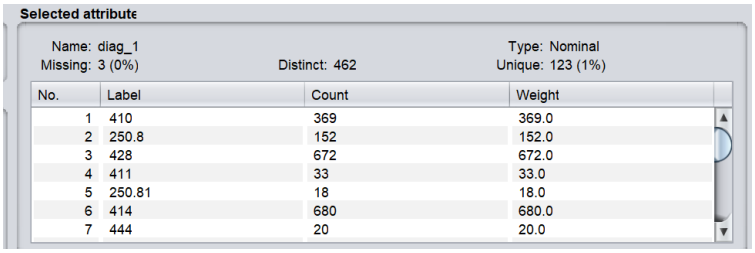


Figure 41 diag_1 after conversion

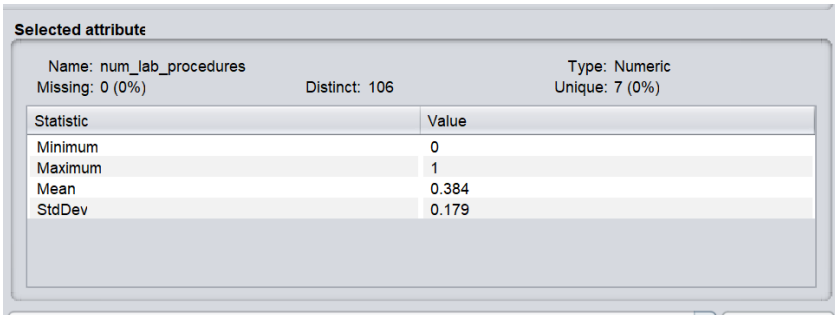


Figure 42 num_lab_procedures before conversion

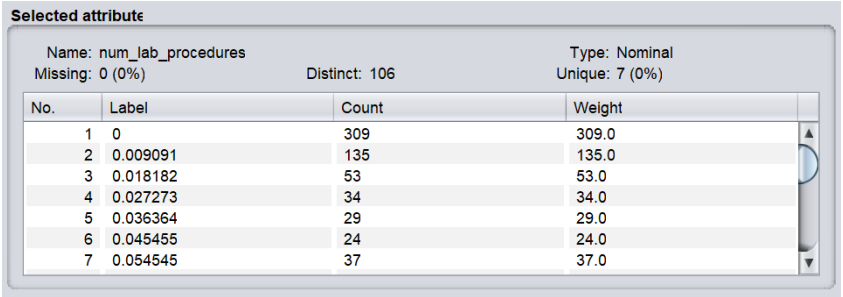


Figure 43 num_lab_procedures after conversion

There are 3 string attributes diag_1,diag_2, diag_3 which will be converted to nominal. attributeRange is mentiones as first-last for both these filters because Weka automatically identifies numeric or string and then convert them to nominal.

6. Discretization (Binning)

Binning may improve the accuracy of predictive models by reducing the non-linearity or noise and it is also useful for certain classifiers. After binning, outliers, missing or invalid values can be easily identified. This can be performed using Weka's Discretize filter. However, for this dataset age is a useful feature to be binned, and that is already done in the dataset. Hence the discretize filter need not be applied.

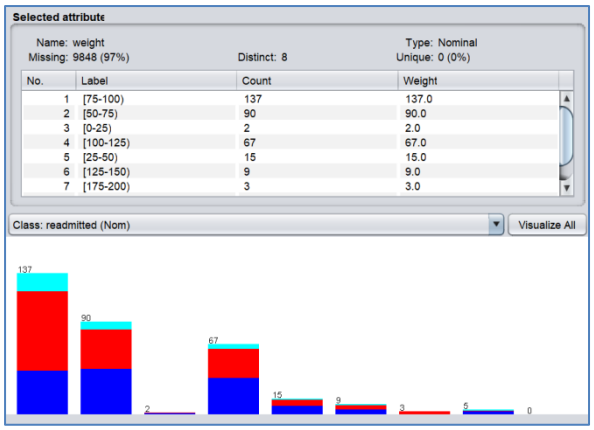


Figure 44 categorised age

7. Missing Values

Data can be missing due to a variety of reasons- hesitance of the respondents to provide complete information, malfunctioning of equipments, errors when entering the data in to the database, sudden changes etc etc (Dr. Abubakr Siddig- Datasets,EDA and altering data structure lecture , 2020). A small amount of missing value is almost unavoidable in large datasets. However, a significant percentage of missing values can be problematic. 'weight' feature contains 97% missing values. It is thus not useful for data mining and is removed from the dataset. The other features and the percentage of missing values are: 'payer_code' - 40%, 'medical_speciality'- 49%, 'race'- 2%, 'diag_3' - 1%, 'diag_1'- 1 missing value. 'payer_code' is not useful for predicting the readmitted class since it is a unique value and hence it is also removed. Other missing values of various attributes are removed by replacing with modes and means using ReplaceMissingValues Filter.

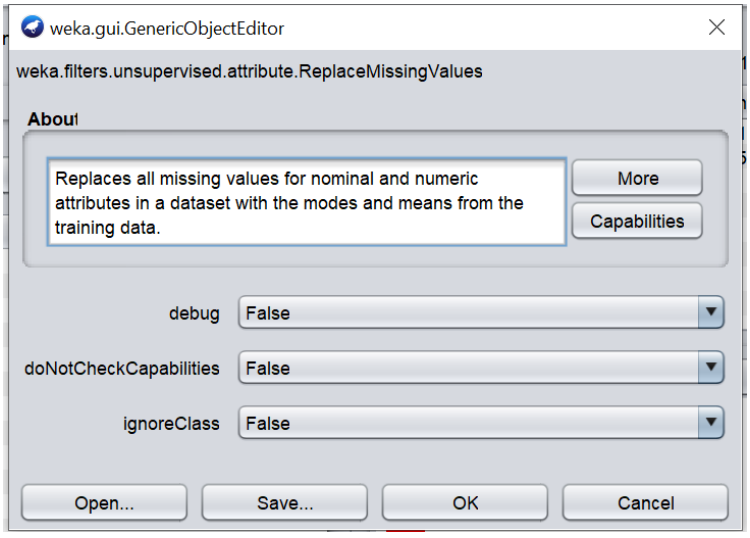


Figure 45 ReplaceMissingValues Filter

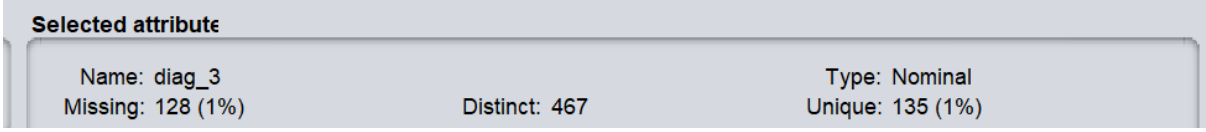


Figure 46 Before and after applying ReplaceMissingValues Filter

8. Outliers

Outlier is a data point that differs significantly from other observations which can be due to error in measurements or exceptional cases (Wikipedia Outlier, 2020). Outliers and extreme values in the dataset are detected using Weka's unsupervised attribute InterQuartileRange Filter. It gives us the middle spread of the data. This filter skips the class values. It creates two new features Outlier and ExtremeValue with two distinct values 'No' and 'Yes' for all instances.

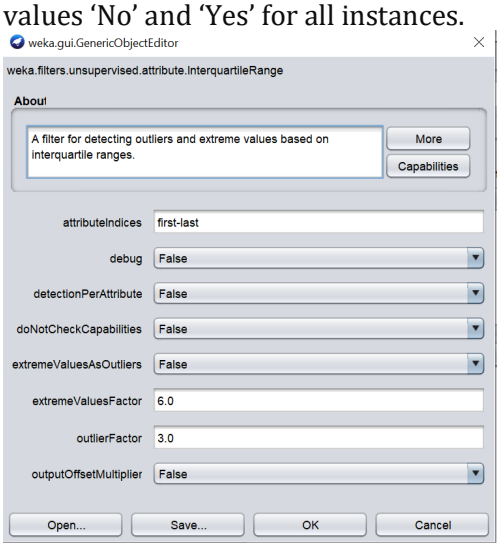


Figure 47 InterQuartileRange Filter

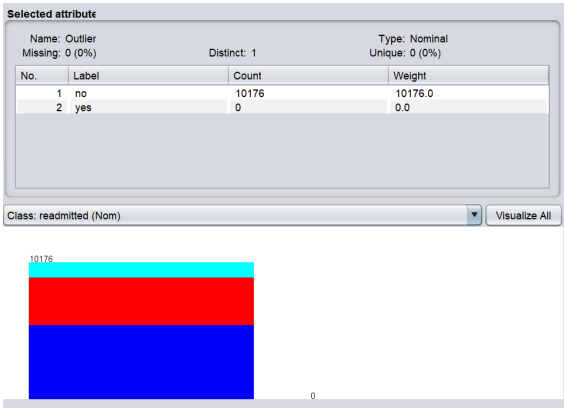


Figure 48 Outlier

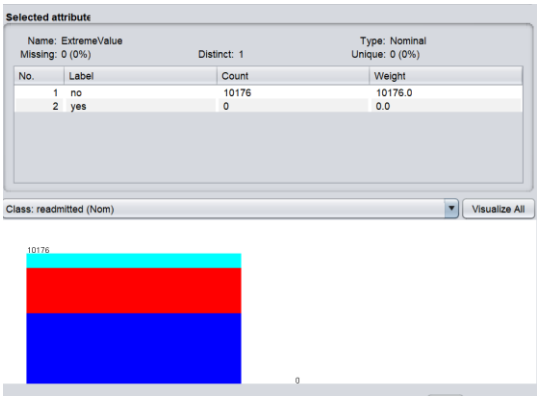


Figure 49 ExtremeValue

This dataset do not contain any outliers or extreme values.

9. Class Imbalance handling with SMOTE(Synthetic Minority Oversampling Technique)

Class imbalance is a problem where "the total number of a class of data (positive) is far less than the total number of another class of data (negative)". Machine learning algorithms tend to work better with roughly equal classes (Chioka, 2013). Weka's SMOTE filter can be used to balance the classes. SMOTE filter was installed using Weka's Tools Option with PackageManager. 6 Nearest Neighbours were used. Applying SMOTE increased the performance of J48 dramatically in this dataset. An increase in accuracy of 5% was observed after class balancing. SMOTE performs oversampling of the examples in the minority class (Brownlee, 2020) After applying SMOTE, there is 14865 instances.

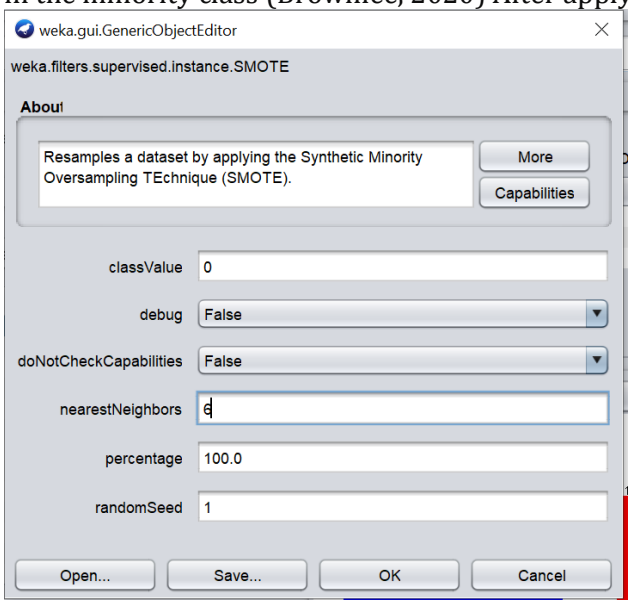


Figure 50 SMOTE filter with 6 nearest neighbours

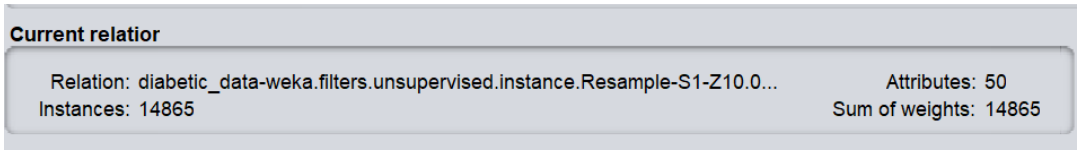


Figure 51 After SMOTE

10. Reordering of Attributes

Weka's Reorder filter can be used to reorder the attributes. However, in this dataset the class attribute is already set as the last attribute and there is no need to rearrange the order of the attributes. Hence this will not be performed.

All of the above steps and their effects on the data is checked during the preprocessing stage. Not all of these methods are applied on the data as some of them are not effective on this dataset as mentioned in each of the cases above. The preprocessed data is stored in 'dataset.arff'.

3. Divide your dataset into training and test set – 5%

The dataset is divided into training and test dataset. It was also initially divided into crossvalidation set for evaluation. However, only training and test data sets are submitted. Training data is the sample of data that is used to fit the learning model and test data is defined as "The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset" (Medium, 2017).

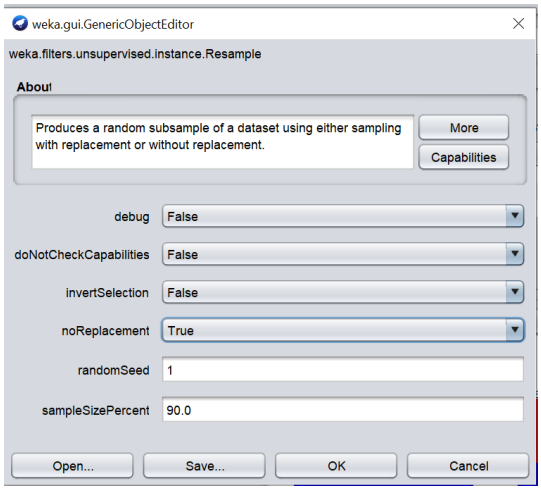


Figure 52 Creating training data

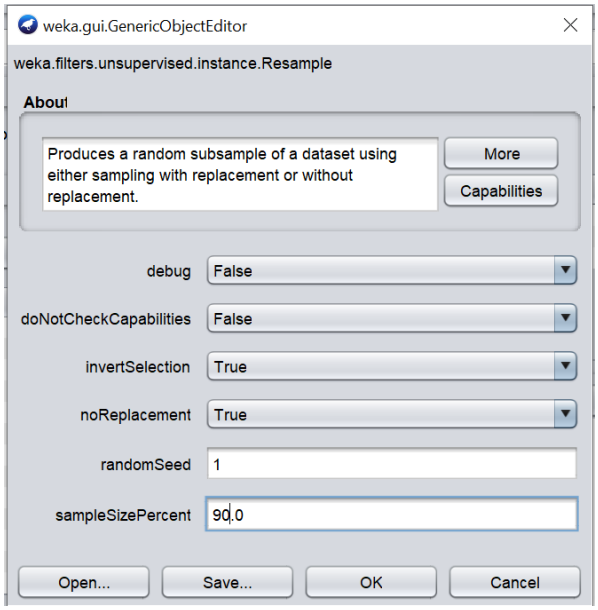


Figure 53 Creating test dataset

While creating test dataset, previous operation is undone and invertSelection is set to True. This is because we want our dataset to be properly split between training and test data.
These files are saved in trainingSet.arff and testingSet.arff

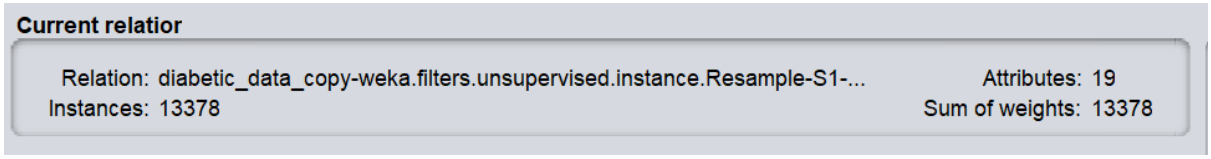


Figure 54 Training data

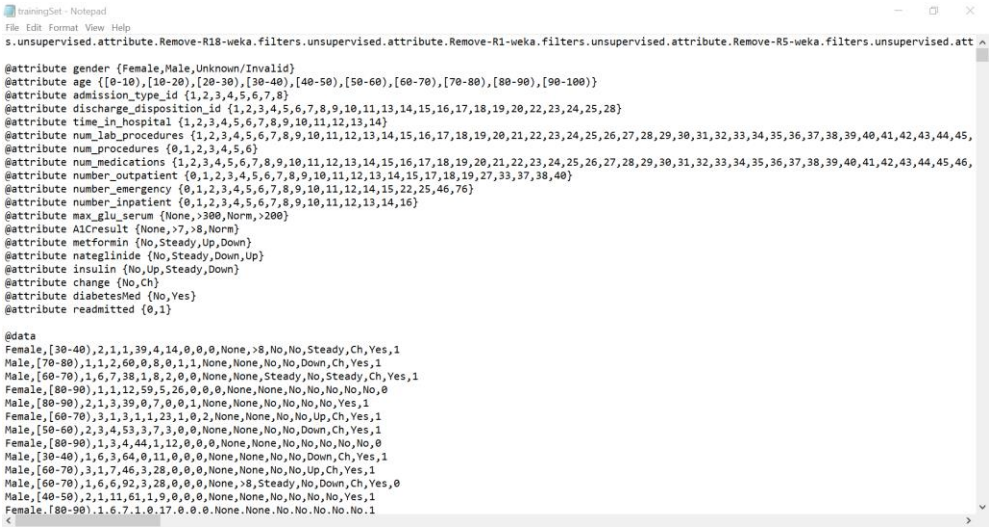


Figure 55 Training data file

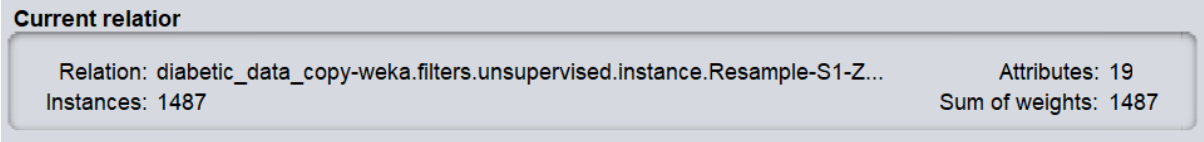


Figure 56 Test data

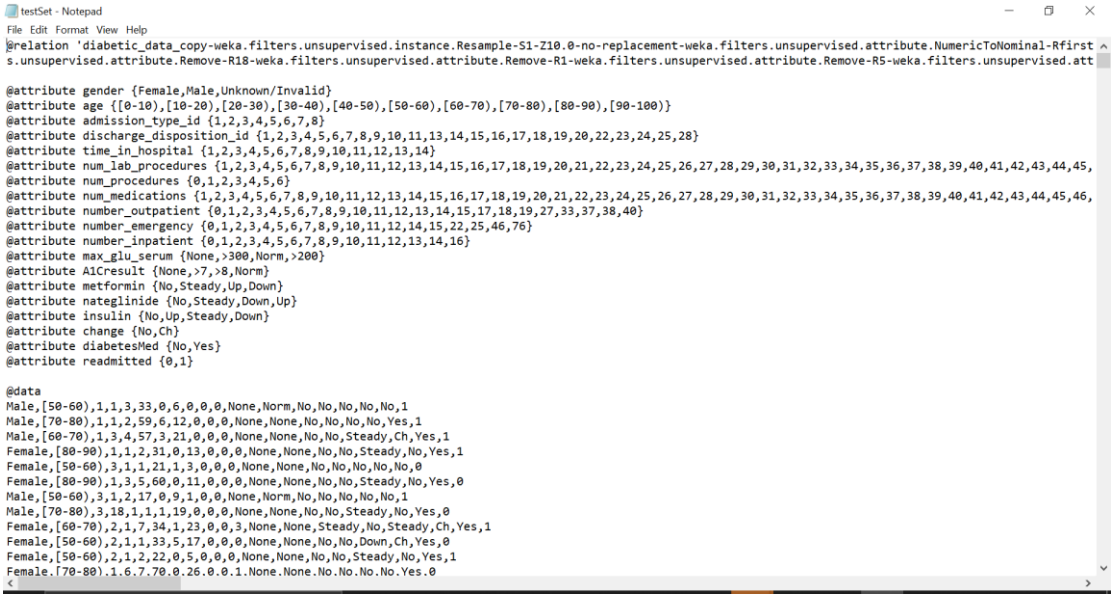


Figure 57 Test data file

Experiments

For each of the following classification techniques

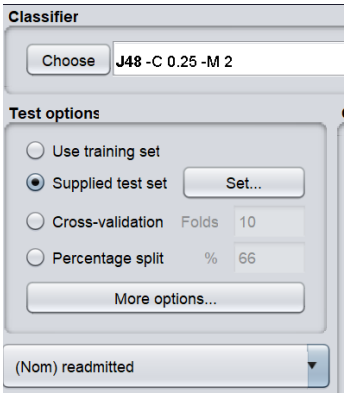
1. Train your model using trainingSet.arff
2. Test your model using testingSet.arff
3. Write a few paragraphs analyzing the results. Be sure to vary parameters at least 3 times in each case. Support this analysis with screenshots of the following
    a. The model or a visualization of the model
    b. The results of the model
    c. Any additional output of the model including but not limited to
        i. Rules
        ii. Confidence Values
        iii. Confusion Matrixes
        iv. Etc.
    d. Simple references to the notes or URL links to online resources complete with a sentence or two of explanation.

3.1. Classification: J48 Tree – 10%

For the classification task, J48 decision tree is used. Decision trees are versatile and they can fit into complex datasets with a divide and conquer approach. (Dr. Abubakr Siddig- Decision Trees- lecture , 2020). J48 provides the best performance, the diagonal elements will be evenly distributed.

Experiments using training and test data

Weka Explorer is used to test the trained model on a supplied test set. This is a very useful feature. Parameters are varied to see the effect in performance. All experiments are performed with J48 algorithm. Parameters will restrict the freedom of the tree thus preventing overfitting (Dr. Abubakr Siddig- Decision Trees-2 lecture , 2020)



The parameters that are significant when performing J48 (As per the Weka definitions) are:

1. minNumObj - The minimum number of instances per leaf. Increasing this parameter will decrease the treesize as we will see below.
2. unpruned- Whether pruning is performed. Both pruned and unpruned options will be tested.
3. confidenceFactor - The confidence factor used for pruning (smaller values incur more pruning).
4. subtreeRaising - Whether to consider the subtree raising operation when pruning.

All of the experiments below are performed using supplied test set –'testingSet.arff'.

3.1.1. Experiment 1

For the first experiment, the parameters chosen are:

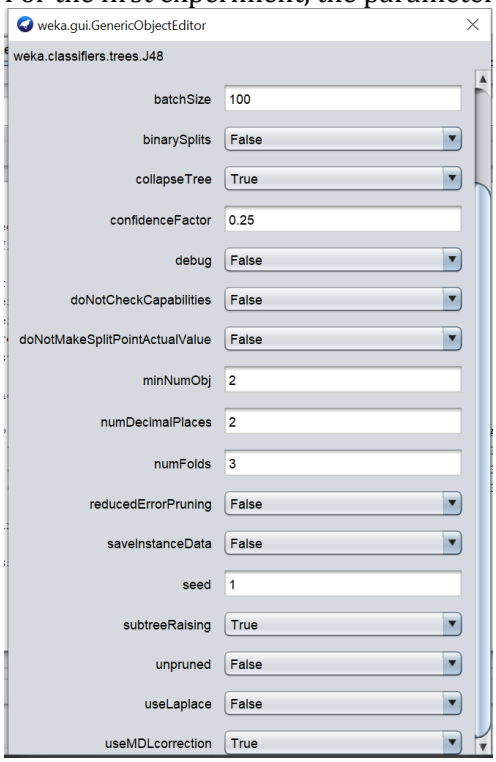


Figure 58 Experiment 1- J48 - parameters

Leaving parameters unstrained will fit the tree very closely and mostly overfit the data (Dr. Abubakr Siddig- Decision Trees-2 lecture, 2020). However, the parameters are kept set in their default options initially, to see the performance of the algorithm with these options.

Model

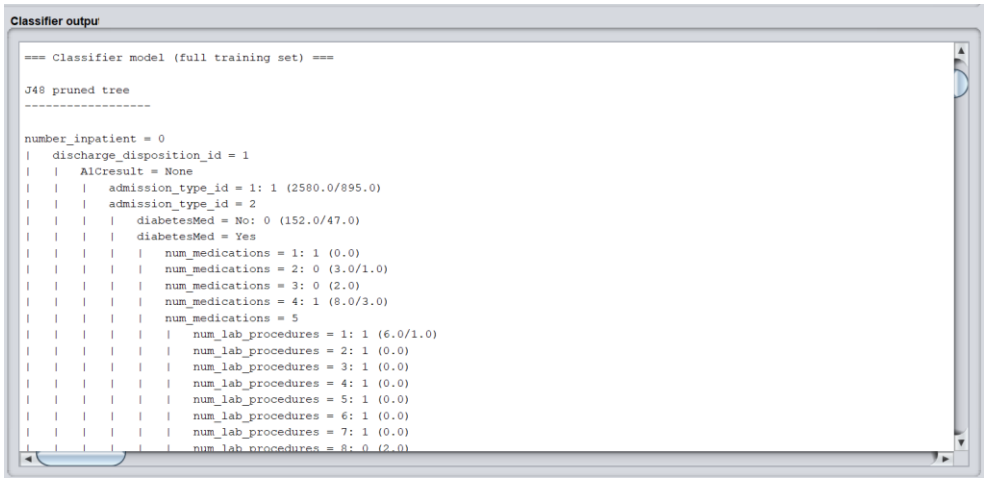


Figure 59 Experiment 1- J48 model beginning

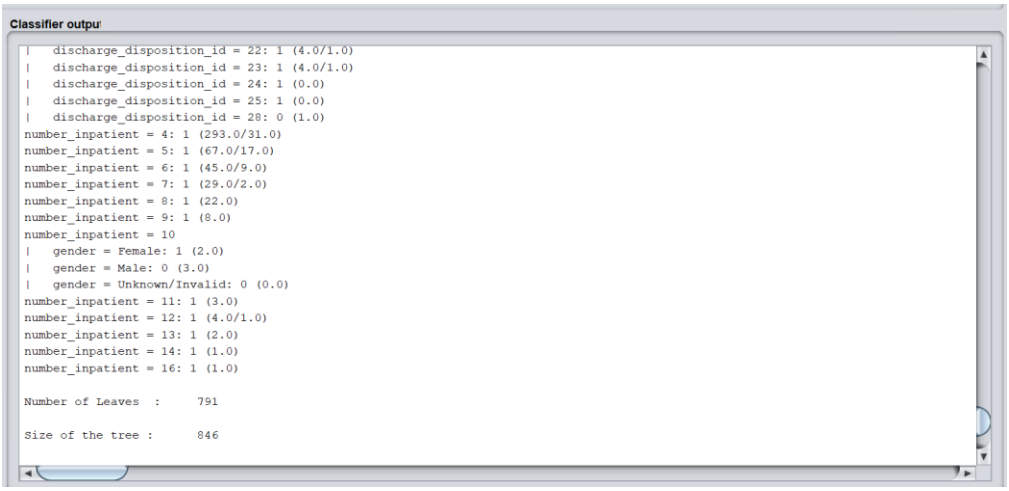


Figure 60 Experiment 1- J48 model end

Results with confidence values, confusion matrix, tree, rules.

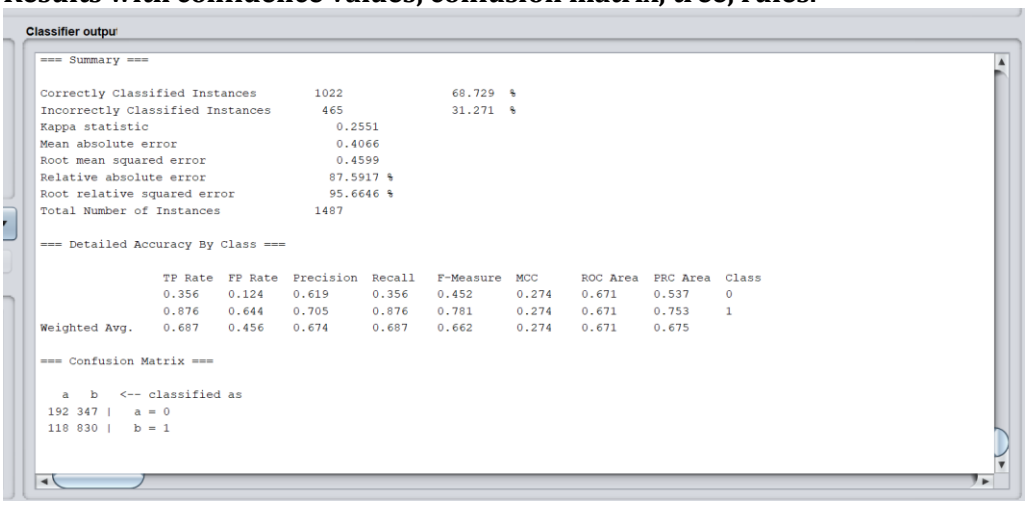


Figure 61 Experiment 1 -J48 -Result

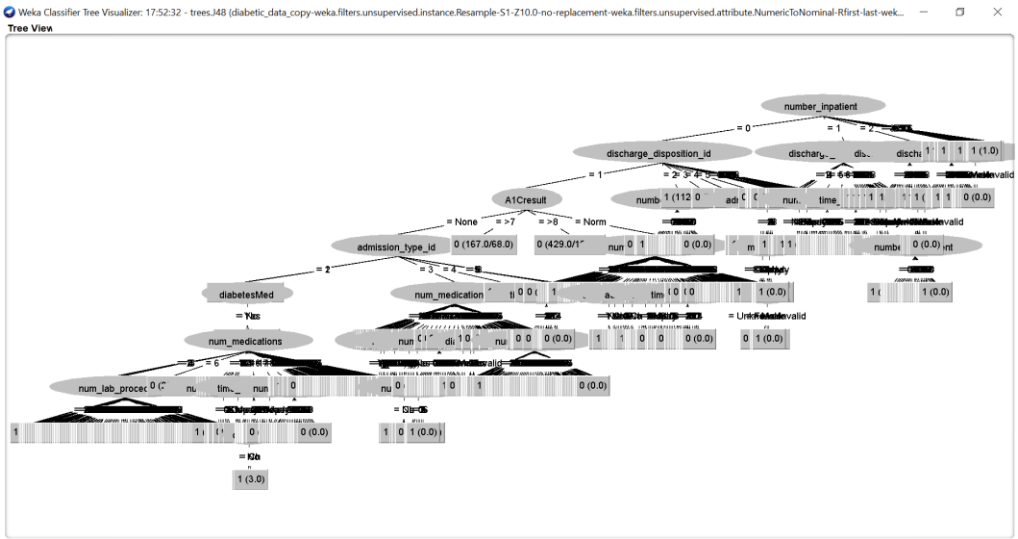


Figure 62 Experiment 1- J48- Tree

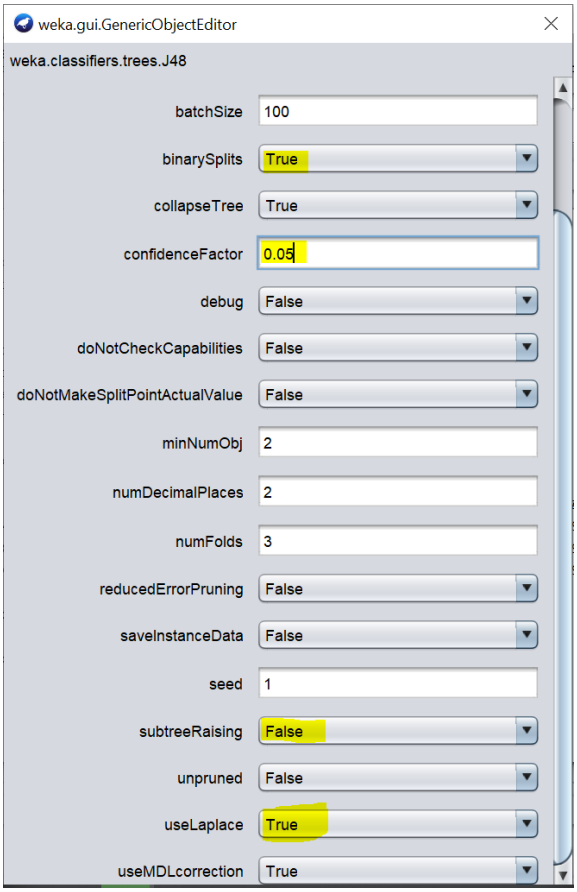
Findings

1. Using confidence factor 0.25 and minNumObj as 2, with pruned tree - 68.729% of instances were correctly classified by this model.
2. Pruned tree increased the accuracy. The model used a pruned tree because pruning mostly reduces the complexity of the final classifier, reduce overfitting to increase predictive accuracy. (Wikipedia Decision Tree Pruning, 2020)

3. From, the confusion matrix it is understood that the model is more good at identifying the patients with risk of readmission (b=1) – correctly classified 830 out of 948 and only incorrectly classified it as 0 only 118 instances out of 948, whereas in identifying patients with no risk of readmission (a=0), it only classified 192 correctly and 347 were classified incorrectly.
4. From the visualization of the tree, it is evident that the tree gained more information from the number_inpatient attribute and further branching was made on this attribute. This also seems logical as the variable is the number of inpatient visits of the patient in the year preceding the encounter. **number_inpatient** is the **best feature** on which the data is further split.

3.1.2. Experiment 2

The parameters are set to the following as shown below. The changes from previous experiment are highlighted. subtreeRaising is set to false , asking not to perform the subtree raising operation when pruning. confidenceFactor is set to 0.05 for pruning (smaller values incur more pruning).



Model

```
discharge_disposition_id = 11: 0 (146.0)
discharge_disposition_id != 11
|   discharge_disposition_id = 14: 0 (46.0/4.0)
|   discharge_disposition_id != 14
|   |   age = [0-10): 0 (11.0)
|   |   age != [0-10)
|   |   |   A1Cresult = None
|   |   |   |   number_inpatient = 8: 1 (22.0)
|   |   |   |   number_inpatient != 8
|   |   |   |   |   number_emergency = 7: 1 (19.0)
|   |   |   |   |   number_emergency != 7
|   |   |   |   |   |   discharge_disposition_id = 15: 1 (17.0)
|   |   |   |   |   |   discharge_disposition_id != 15
|   |   |   |   |   |   |   discharge_disposition_id = 13
|   |   |   |   |   |   |   |   num_lab_procedures = 59: 1 (3.0)
|   |   |   |   |   |   |   |   num_lab_procedures != 59: 0 (28.0/5.0)
|   |   |   |   |   |   |   discharge_disposition_id != 13
|   |   |   |   |   |   |   |   time_in_hospital = 13
|   |   |   |   |   |   |   |   |   num_medications = 17: 1 (5.0)
|   |   |   |   |   |   |   |   |   num_medications != 17
|   |   |   |   |   |   |   |   |   |   num_medications = 22: 1 (2.0)
|   |   |   |   |   |   |   |   |   |   num_medications != 22
|   |   |   |   |   |   |   |   |   |   |   num_medications = 25: 1 (2.0)
|   |   |   |   |   |   |   |   |   |   |   num_medications != 25
```

Figure 63 Experiment 2- Model (Small part)

Results with confidence values, confusion matrix, tree, rules.

```
Classifier output

=== Summary ===

Correctly Classified Instances        1037               69.7377 %
Incorrectly Classified Instances       450               30.2623 %
Kappa statistic                          0.2938
Mean absolute error                      0.4071
Root mean squared error                  0.4622
Relative absolute error                 87.6928 %
Root relative squared error             96.1289 %
Total Number of Instances             1487

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.414    0.141     0.625      0.414     0.498     0.307    0.639     0.524     0
                 0.859    0.586     0.720      0.859     0.783     0.307    0.639     0.718     1
Weighted Avg.    0.697    0.425     0.686      0.697     0.680     0.307    0.639     0.647

=== Confusion Matrix ===

   a   b   <-- classified as
 223 316 |   a = 0
 134 814 |   b = 1
```

Figure 64 Experiment 2- Result- J48

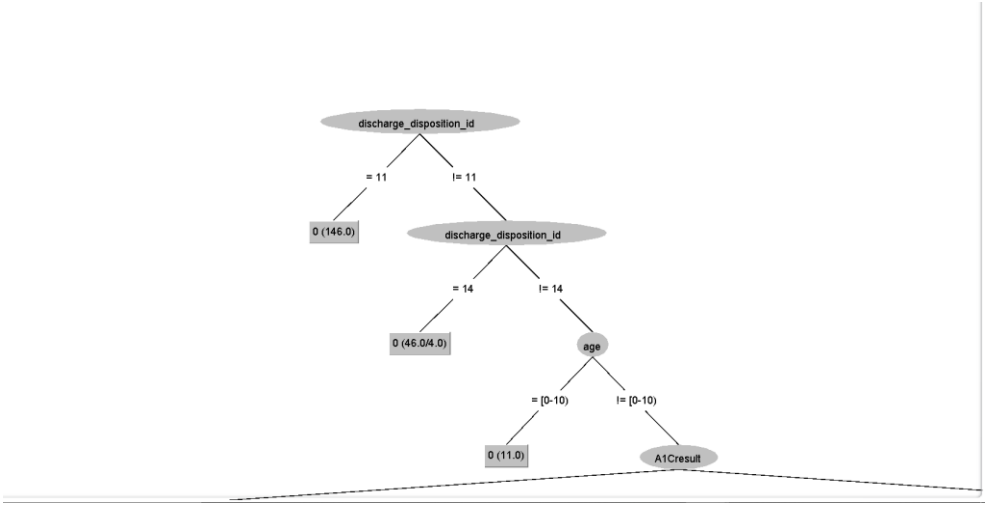


Figure 65 Experiment2- J48- Root of the tree

Figure 66 Experiment 2 -J48- Full tree after binarySplit

Due to limitations on space, the entire model and tree visualization screenshot is not included.

Findings

1. In this experiment, the parameters are varied to tune the model and it increased the model accuracy by 1.0087 % to reach 69.7377%. However Root Mean Squared Error(RMSE) increased to 0.46 from 0.45.
2. The parameters that had a effect on this are:
    - unpruned is set as False, because pruned decision trees prevent the overfitting of the training data. (Dr. Abubakr Siddig- Decision Trees-2 lecture , 2020). binarySplits is set to true to use binary splits on nominal attributes when building the trees.
    - collapseTree is set to tree to remove parts that do not reduce training error.
    - subtreeRaising is set to false , asking not to perform the subtree raising operation when pruning
    - confidenceFactor is set to 0.05 for pruning (smaller values incur more pruning).
    - useLaplace is also set to true to smooth count at leaves based on Laplace. But this parameter had minimal effect on the performance.
3. From, the confusion matrix it is understood that the model is still not that good at identifying the patients with no risk of readmission (a=0) as compared to patients with risk of readmission (b=1)– correctly classified 223 out of 539 and only 316 instances out of 539 incorrectly classified it as 1, whereas in identifying patients with risk of readmission (b=1), it classified 814 instances correctly and 134 were classified incorrectly.
4. From the visualization of the tree, it is evident that the tree began from the discharge_disposition attribute which is an attribute that corresponds to different ids such as discharged to home. The tree has become quite large due to binary splitting on nominal variables. However, this parameter increased the number of correctly classified instances.

3.1.3. Experiment 3

In experiment 3, minNumObj is increased to 3 as shown below and confidence factor is set to 0.25 to avoid overfitting.

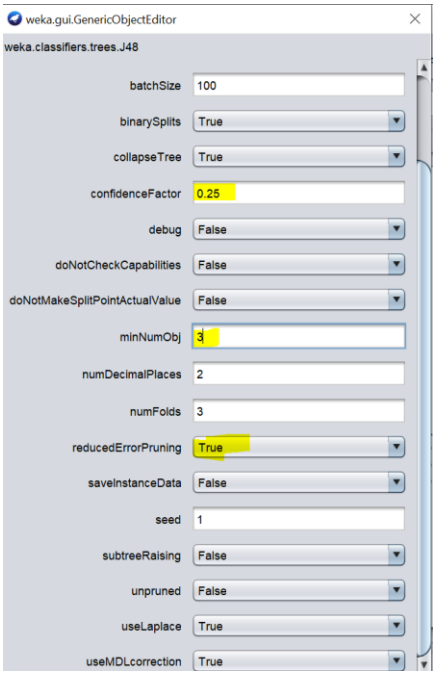


Figure 67 Experiment 3-J48

Model

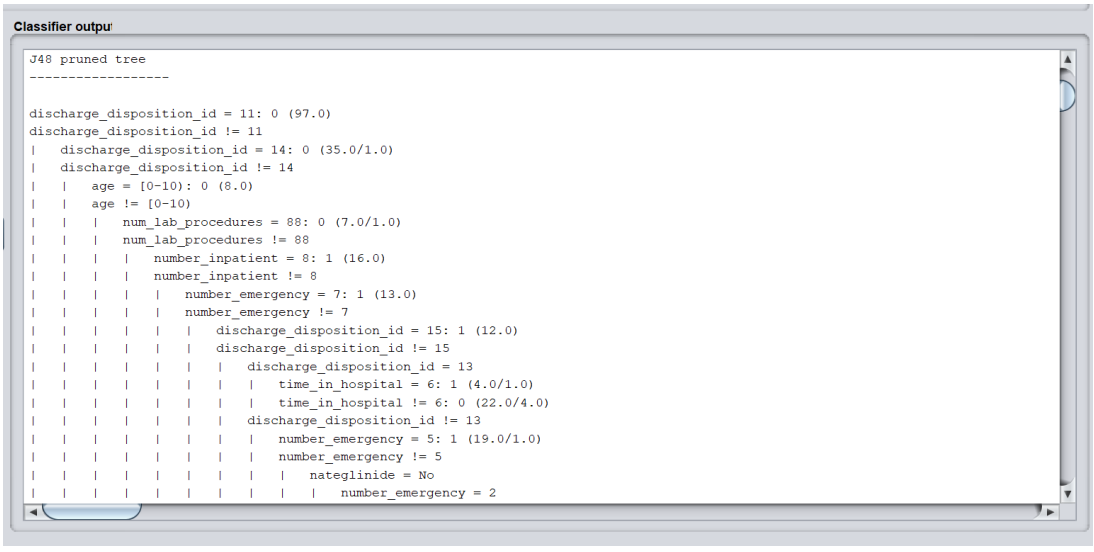


Figure 68 Experiment 3- J48-Model

Results with confidence values, confusion matrix, tree, rules.

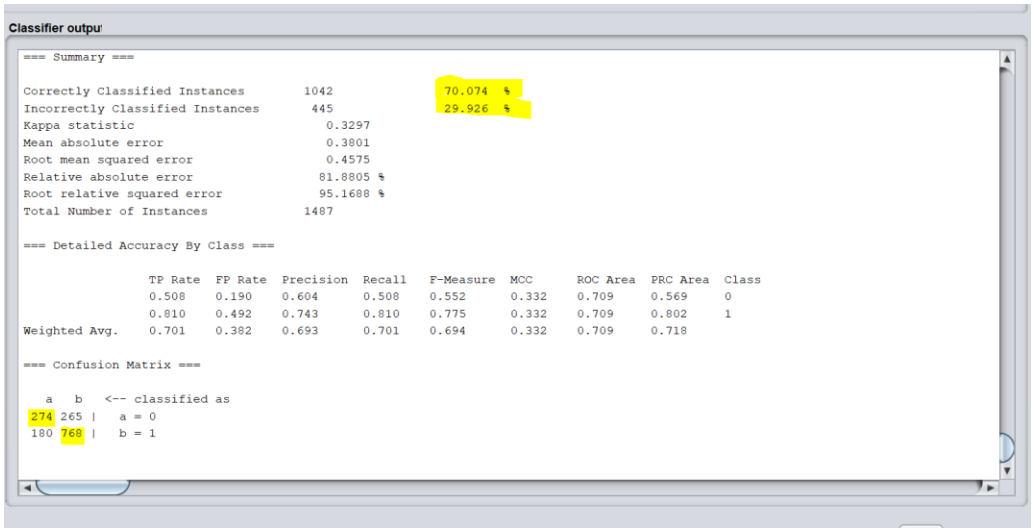


Figure 69 Experiment 3-Result

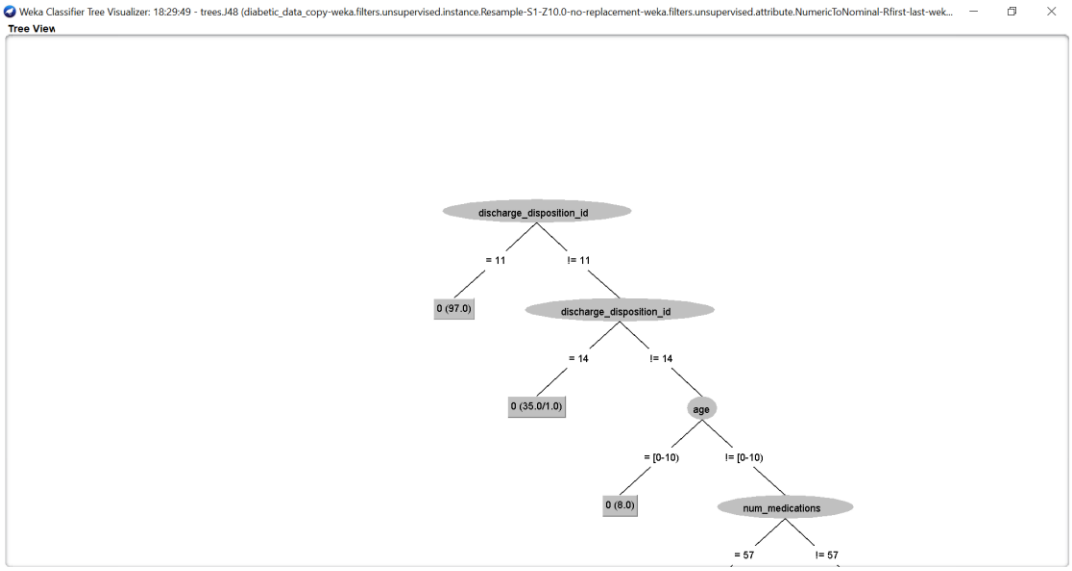


Figure 70 Experiment 3- J48 -Tree Root

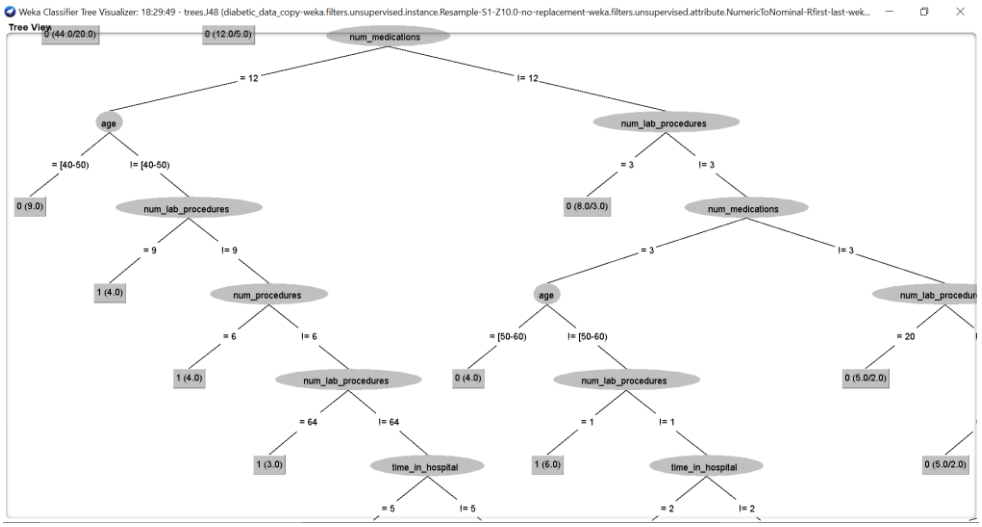


Figure 71 Experiment 3- J48 auto scaled

Figure 72 Experiment 3- J48 Full tree after binarySplit

Findings

1. In this experiment, the minNumObj , i.e minimum number of instances per leaf is increased to 3. reducedErrorPruning is also set to True which means reduced-error pruning is used instead of C.4.5 pruning. and it increased the model accuracy from 69.7377%. to **70.074 %.** This is mainly because increasing minimum number of instances optimally will help generalize the model better (Dr. Abubakr Siddig- Decision Trees-2 lecture , 2020) Root mean squared error decreased from 0.4622 to 0.4575. Relative absolute error decreased from 87.51% to 81.8805%. Weighted average of precision increased from 0.686 to 0.693 and recall increased from 0.697 to 0.701.
2. From, the confusion matrix it is understood that the model is now getting better at identifying the patients with no risk of readmission (a=0) as well as patients with risk of readmission (b=1)– correctly classified 274 out of 539 and only 265 instances out of 539 incorrectly classified it as 1, whereas in identifying patients with risk of readmission (b=1), it only classified 768 instances correctly and 180 were classified incorrectly(6 more from the previous experiment).
3. The tree has become quite large due to binary splitting on nominal variables. However, this parameter increased the number of correctly classified instances.
4. This is the highest accuracy achieved by varying J48 hyperparameters

3.1.4. Experimenter- Bagging, Boosting and Stacking

Now, Weka experimenter is used to rank algorithms such as AdaboostM1, Bagging and Stacking. This helps us understand and infer the J48 performance. These techniques are used to alter variance and bias so that model can perform better. However, if the data is already having low bias and variance this result in overfitting.

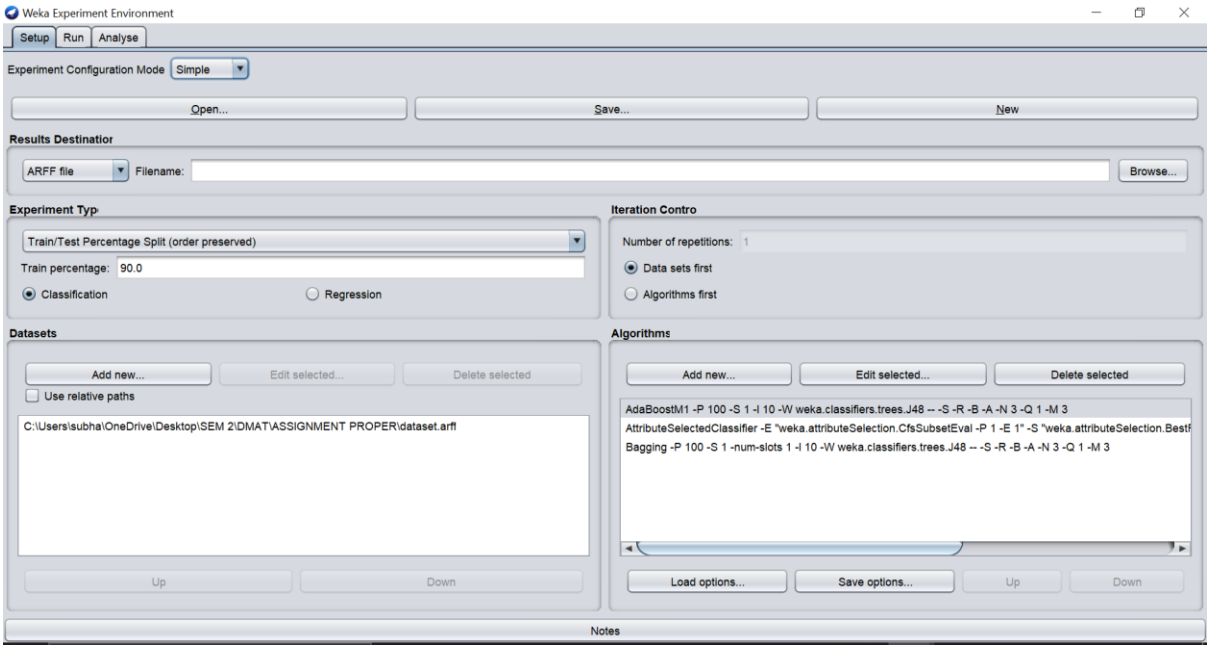


Figure 73 Experimenter- J48

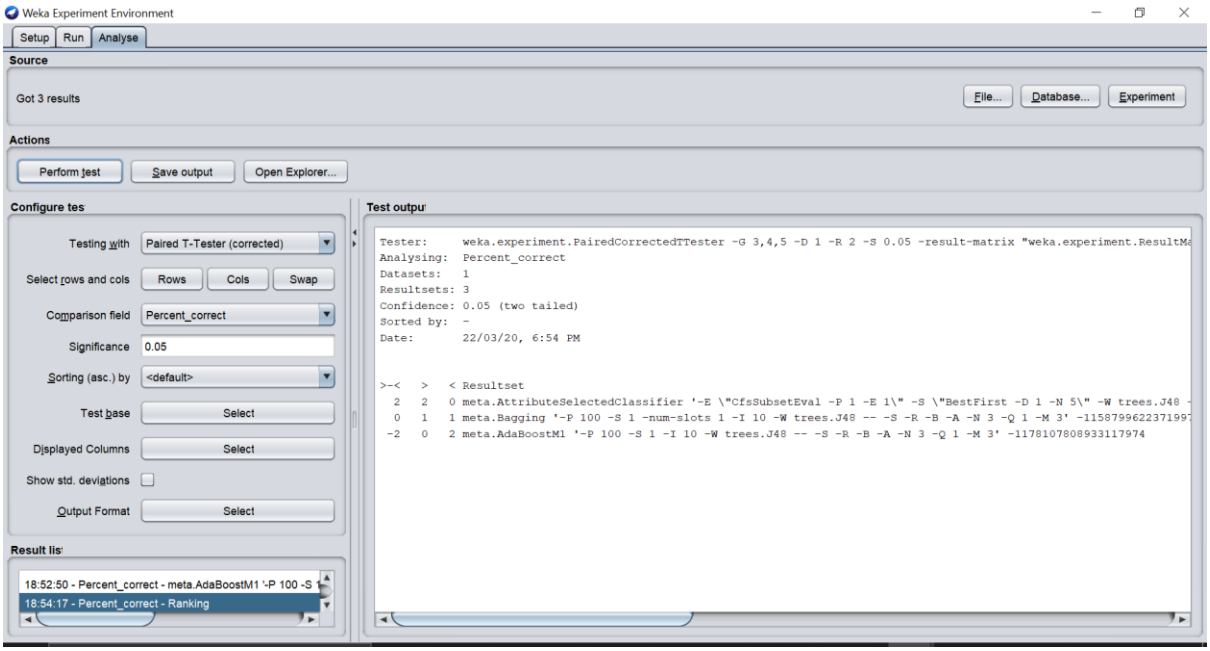


Figure 74 Experimenter -J48 - Ranker

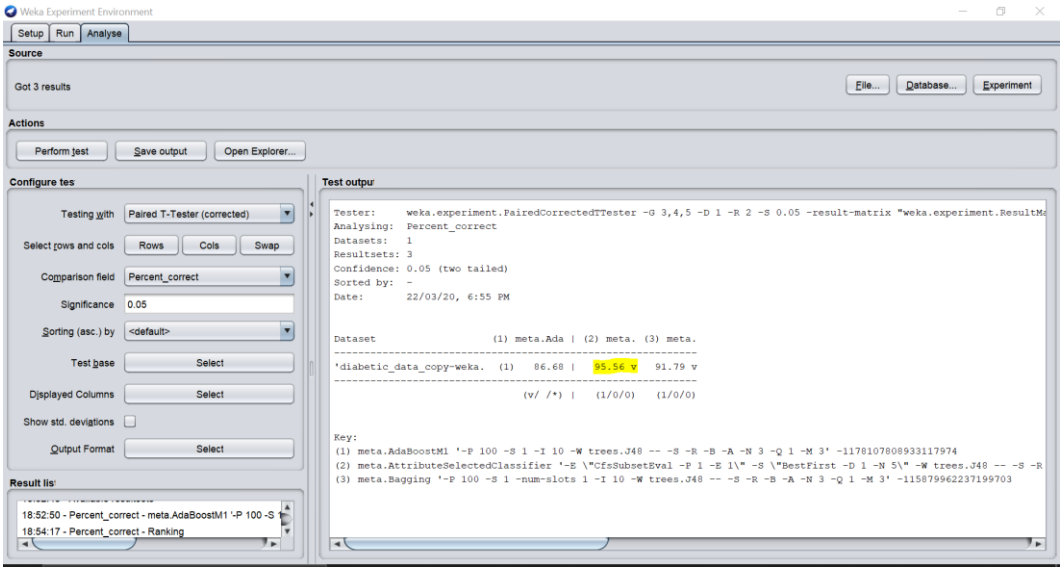


Figure 75 Experimenter - J48

Findings

1. In Experimenter, three algorithms were experimented
    - AttributeSelectedClassifier
    - AdaBoostM1
    - Bagging
2. AttributeSelectedClassifier performed better with an accuracy of **95.56% .** The ranking clearly shows the best performance which shows that AdaBoostM1 was beaten two times.
3. AdaBoostM1 is an ensemble method that also improved the performance compared to previous results. It achieved an accuracy of 86.50%. It creates a strong classifier from a number of weak classifiers (Brownlee, 2016).
4. Bagging is another ensemble method which achieved accuracy of 91.79%. In Bagging individual decision trees are grown deep and we have low bias
5. AttributeSelectedClassifier reduces the dimensions by attribute selection and it achieves best performance using CfsSubsetEval evaluator and BestFirst search. This selected the best features to perform J48 with 0.25 confidence factor. 0.1 confidence factor reduced the accuracy in this case. Thus, optimal values were chosen.

The model achieved an accuracy of 95.56% with AttributeSelectedClassifier in Experimenter with J48(minNumObj=3, binarySplit=True, confidenceFactor=0.25, subTreeRaising=False) and an accuracy of 70.074 with the test data with J48 (minNumObj=3, binarySplit=True, confidenceFactor=0.25, subTreeRaising=False,unpruned=False)

3.2. Classification: Association Rules – 10%

Association rule mining is used to find frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories (Dr. Abubakr Siddig- Association lecture , 2020)

Some important parameters in Weka for Association Rule algorithm Apriori that we will be considering are:

1. metricType - Set the type of metric by which to rank rules. There are four types- Confidence, Lift, Leverage and Conviction. Confidence is a conditional probability that a transaction having Y also contains Z to find all the rules Y ➔ Z. (Dr. Abubakr Siddig- Association lecture , 2020)
2. lowerBoundMinSupport - Lower bound for minimum support.
3. upperBoundMinSupport - Upper bound for minimum support. Start iteratively decreasing minimum support from this value.
4. Support is the probability that a transaction contains { Y ➔ Z } (Dr. Abubakr Siddig- Association lecture , 2020)
5. numRules - Number of rules to find.
6. outputItemSets - If enabled the itemsets are output as well.

Three different experiments are performed for Association Rule Mining by varying the parameters using Apriori.

3.2.1. Experiment 1

In the first experiment, we will use all parameters with default option as shown below. It starts with a minimum support of 100% of the data items and decreases this in steps of 5% (when delta =0.05) until there are at least 10 rules with the required minimum confidence of 0.9 or until the support has reached a lower bound of 10%, whichever occurs first (Data Mining: Practical Machine Learning Tools and Techniques, 2020)

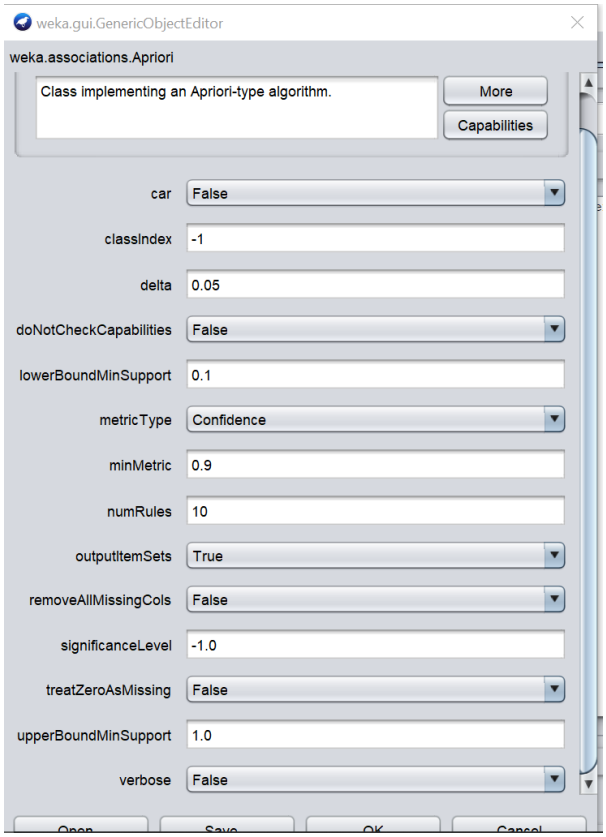


Figure 76 Association Rule Experiment 1

This outputs itemsets and 10 rules with minimum confidence of 0.9

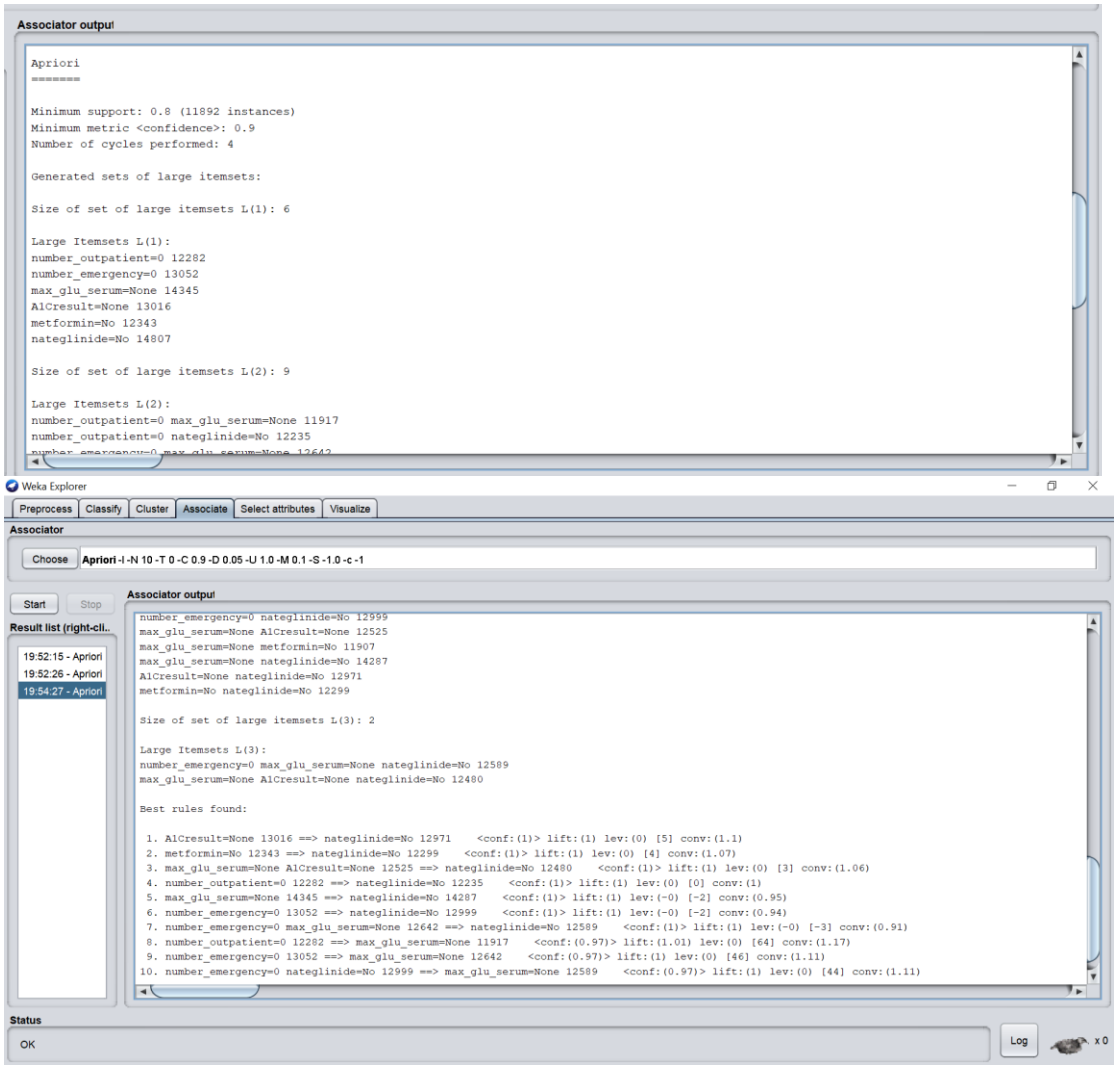


Figure 77 Association rule- Experiment 1 – Item sets and Best Rules

The lowerBoundMinSupport=0.1 and upperBoundMinSupport=1.0 are set. outputItemSets are set as True to show all frequent itemsets. Rules generated are ranked by metricType (default Confidence). Only rules with score higher than minMetric (default 0.9 for Confidence) are considered and delivered as the output.

Findings

1. The algorithm started with MinSupport as 100% and stopped at 80% after running 4 times.
2. The algorithm produced 10 best rules and displayed frequent itemsets in 3 categories L(1), L(2),L(3)
    - Size of set of large itemsets L(1): 6
    - Size of set of large itemsets L(2): 9
    - Size of set of large itemsets L(3): 2
3. From the rules, it is evident that
    3.1. number_emergency=0 max_glu_serum=None 8651 ==> nateglinide=No 8598 <conf:(0.99)> lift:(1) lev:(-0) [-3] conv:(0.91)
        - Most patients with number of emergency visits are 0 and with Glucose Serum test is not conducted, have nateglinide not administered to them with 0.99 confidence. max_glu_serum=None happened with conjunction to number_emergency=0 for 8651 instances, that is the support or coverage.

    3.2. number_outpatient=0 8511 ==> max_glu_serum=None 8146
        - Most patients with number of outpatients visits 0, haven't conducted Glucose Serum Test. This is an almost obvious rule.

3.2.2. Experiment 2

The highlighted parameters are the changes made from the previous experiment.

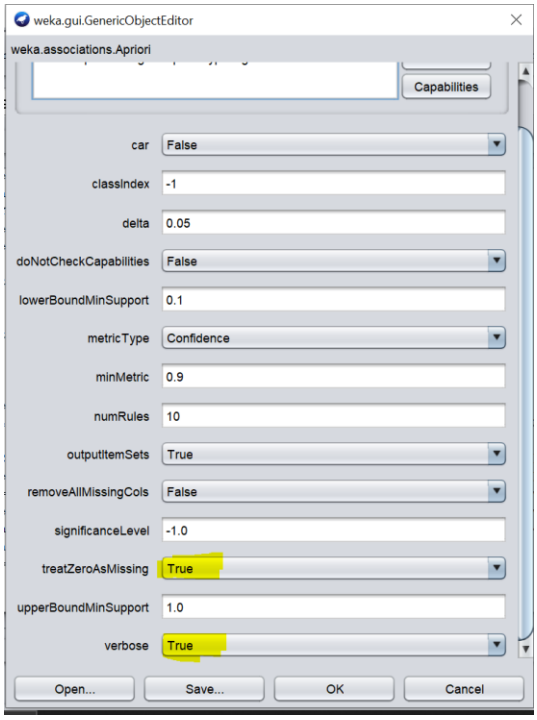


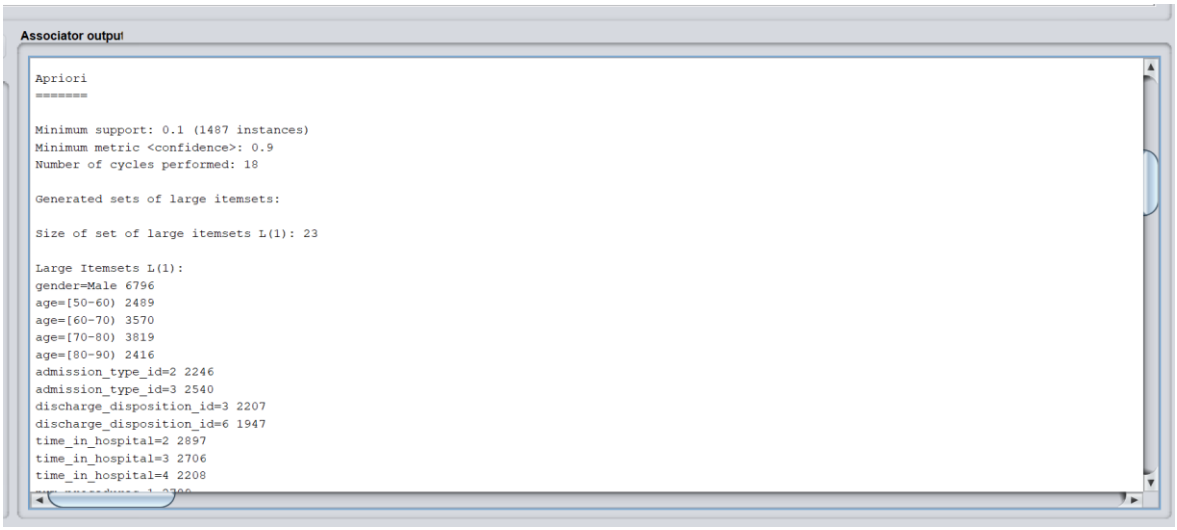
Figure 78 Experiment1- Apriori- Parameters

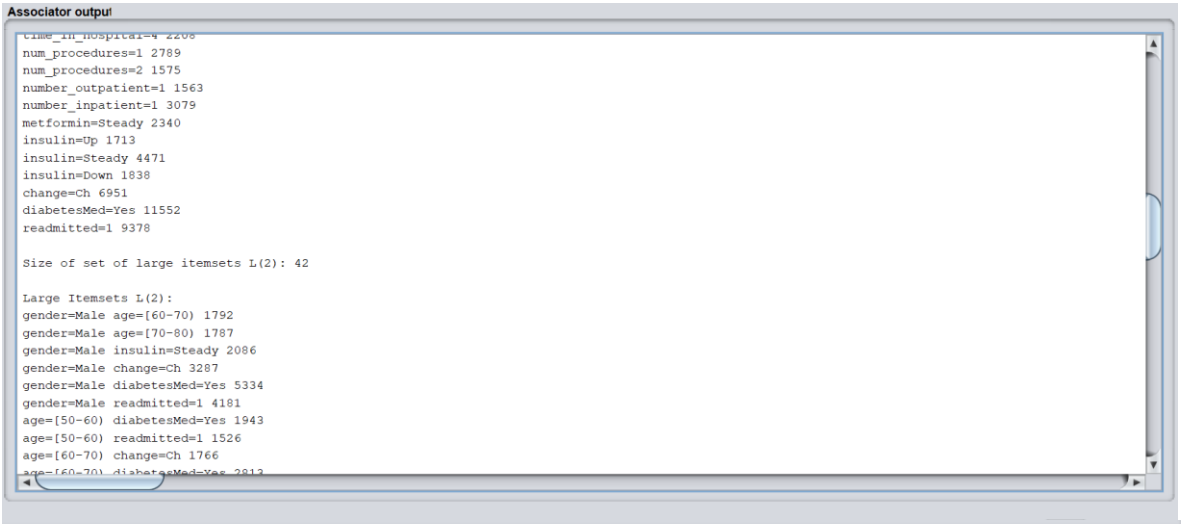
The two parameters that are set to True are:

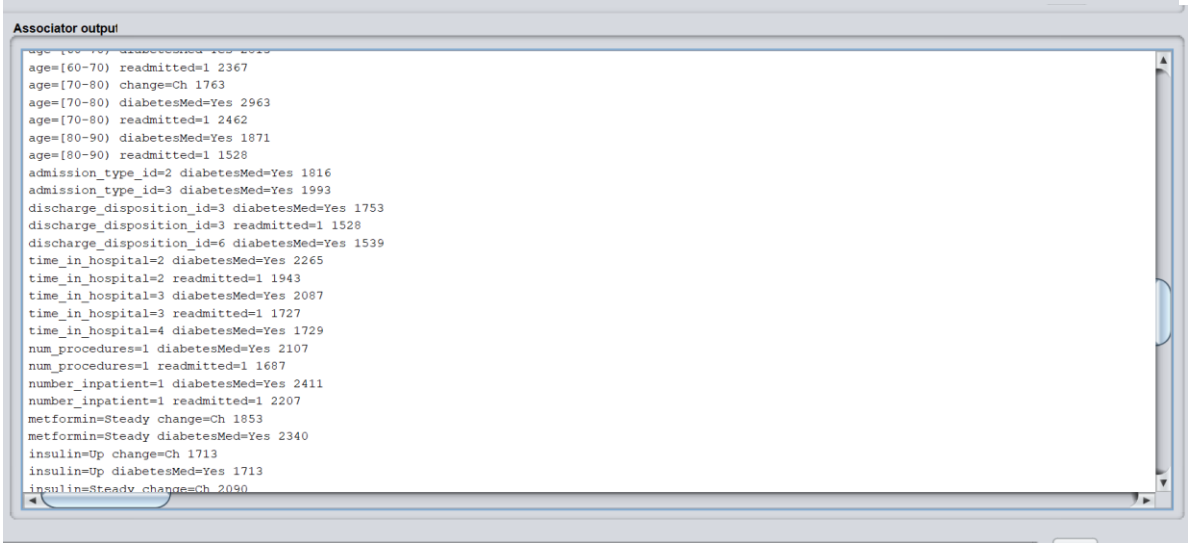
1. treatZeroAsMissing – the first values of nominal (i.e, 0) will be treated as zero. This is set to true here, to uncover more patterns as compared to the previous experiment. Many attributes and rules were not generated in the first experiment.

2. verbose- the will be run in verbose mode, to get extra information during iteration (Nabble, 2012)

Itemsets and Rules







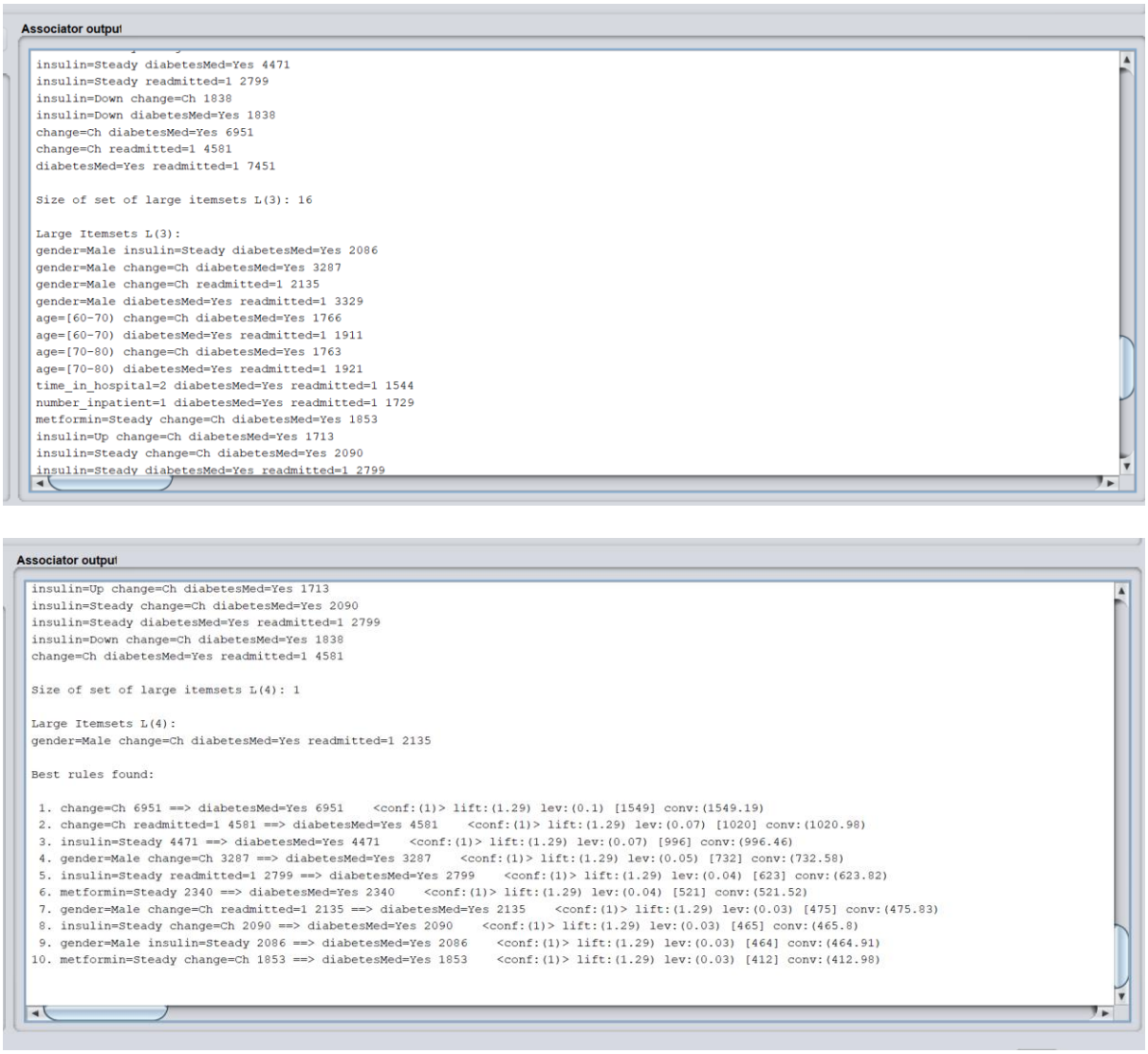


Figure 79 Apriori- Experiment 2- Rules and Itemsets

Findings

1. With treatZeroAsMissing set to True, another 10 rules have been generated. This attribute helped to identify more rules, otherwise the algorithm was focused on the 0 values of the nominal attributes. The algorithm performed 18 cycles, much greater than experiment 1. Confidence cutoff is 1 here.
2. Three itemsets are generated.
    - Size of set of large itemsets L(1): 23
    - Size of set of large itemsets L(2): 42
    - Size of set of large itemsets L(3): 16
3. Some rules that are generated are:
    3.1. gender=Male change=Ch readmitted=1 2135 ==> diabetesMed=Yes 2135 <conf:(1)> lift:(1.29) lev:(0.03) [475] conv:(475.83)
        - 100% of Males with change of medicine and a risk of readmission are prescribed diabetes Medications 2135 instances. Confidence is 1 (2135/2135). This is obviously logical since change of medication indicates a prescribed medication.
    3.2. insulin=Steady readmitted=1 2799 ==> diabetesMed=Yes 2799 <conf:(1)> lift:(1.29) lev:(0.04) [623] conv:(623.82)

- All patients with Steady insulin and risk of readmission are prescribed diabetes Medications with 100% confidence and support is 2799.

4. All of the rules here have 100% confidence because of two attributes 'diabetesMed' and 'change'. When there is a change of medication , it is obvious that diabetes medicine is prescribed.

3.2.3. Experiment 3

In this Experiment metricType used is Lift with minMetric set to 1.1 (default). The lift of a rule is defined as the ratio of the observed support to that expected if X and Y were independent (Wikipedia Association Rule Mining, 2020). The minimum metric is set to 1.1.

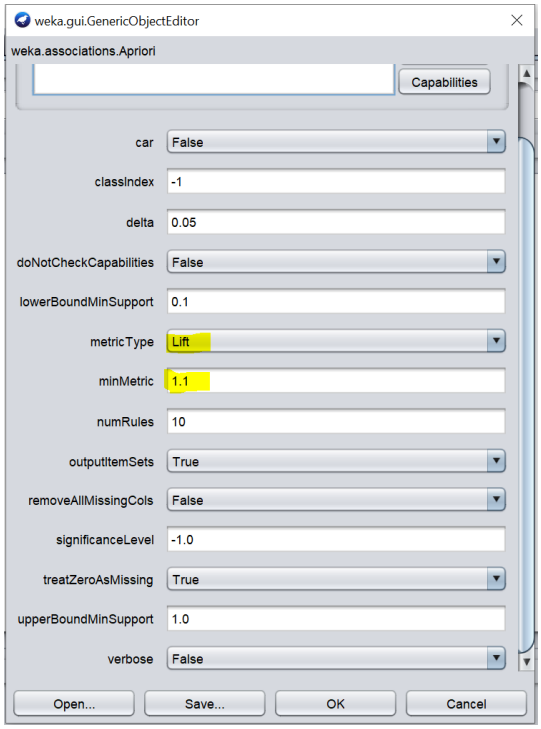
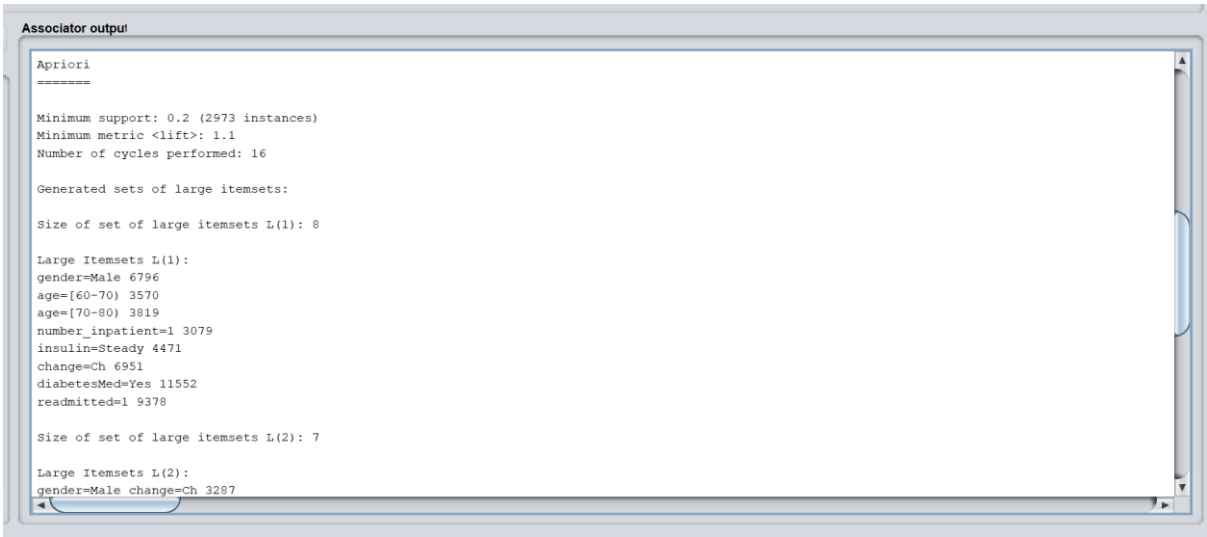


Figure 80 Apriori Experiment 3 -Lift

Rules and ItemSets



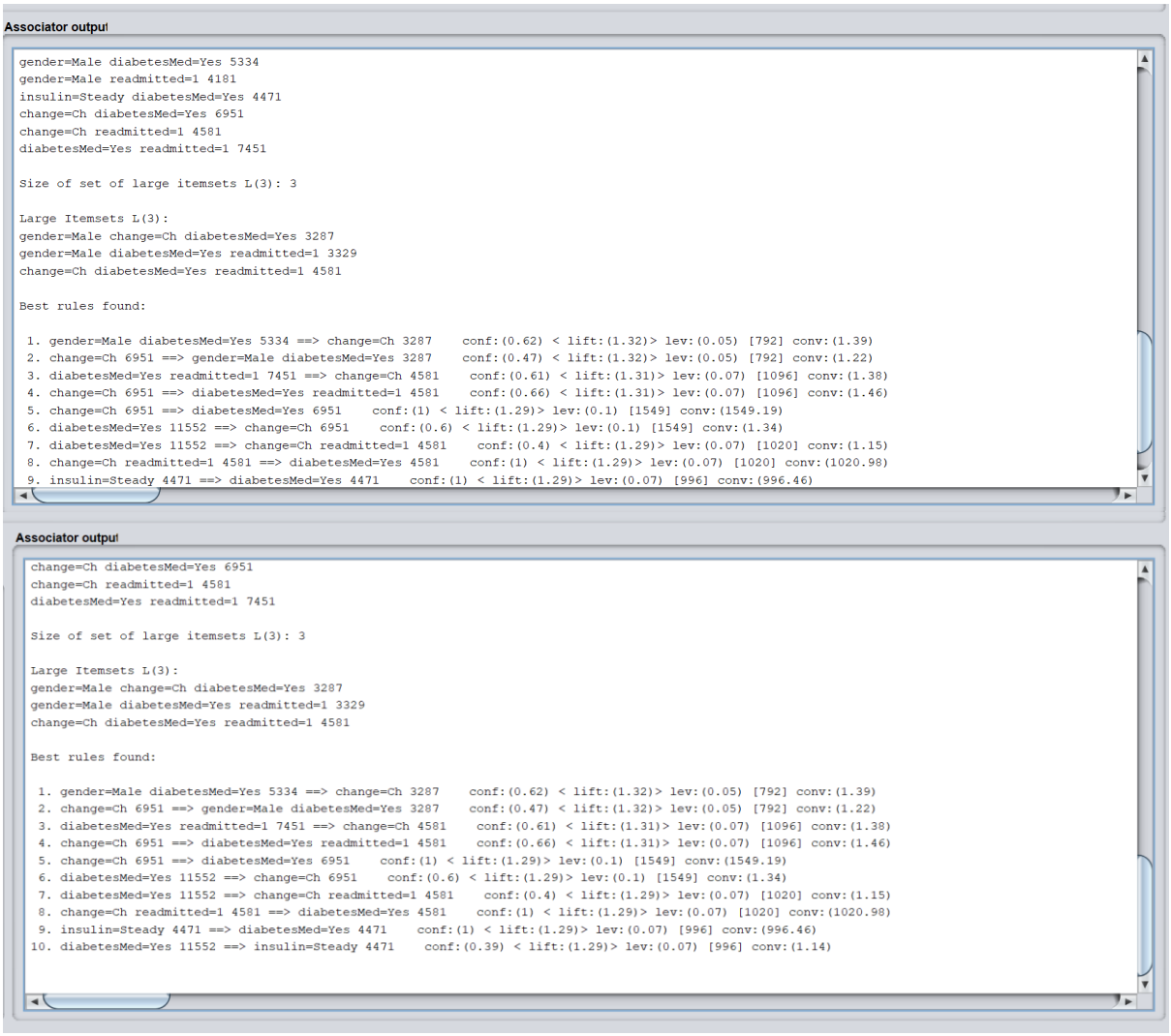


Figure 81 Apriori- Experiment 3- Result

Findings

1. With lift metric type and minMetric as 1.1, some other rules are generated. The algorithm stopped after 16 cycles.
2. There are three itemsets
    - Size of set of large itemsets L(1): 8
    - Size of set of large itemsets L(2): 7
    - Size of set of large itemsets L(3): 3
3. Some of those rules are:
    3.1. gender=Male diabetesMed=Yes 5334 ==> change=Ch 3287 conf:(0.62) < lift:(1.32)> lev:(0.05) [792] conv:(1.39)
        - Males who are prescribed diabetes Medications have a change of medication 62% of times. Lift is 1.32, leverage is 0.05 and conviction Is 1.39
    3.2. Change=ch 6951 ==> gender =Male diabetesMed= Yes 3287 conf:(0.47) < lift:(1.32)> lev:(0.05) [792] conv:(1.22)
        - When there is a change of medication, with 47% of confidence the gender is Male and there is a change in diabetes Medication
4. The rules are mainly focused on 'gender', 'diabetesMed' and 'change'.

Part 2 - Clustering

1. Description of your dataset and findings – 10%

- **Title**: K-Means and DBSCAN Clustering with *Diabetes 130-US hospitals for years 1999-2008* Dataset
- **Data description:** The dataset used is *Diabetes 130-US hospitals for years 1999-2008 Dataset* after resampling which is described in Part 1.
- **Objective**:
  1. To partition the data to good quality clusters from *Diabetes 130-US hospitals for years 1999-2008 Dataset* using K-Means Clustering with predefined number of clusters.
  2. To create good quality clusters from *Diabetes 130-US hospitals for years 1999-2008 Dataset* using DBSCAN Clustering with different and optimal values for epsilon and minPoints and different distance functions.
- **Summary of Findings**
  1. **Preprocessing**
     - Preprocessing steps used in clustering is not the same as the steps used in the classification problem. Hence preprocessing steps are performed again on the original resampled dataset diabetic_dataset_resampled. arff
     - Feature encoding of readmission attribute is performed, feature selection and 25 attributes are selected. encounder_id' and patient_nbr' contains all 10176 unique values, thus they are not useful for classification. 'weight' feature contains 97% missing values. It is thus not useful for data mining and is removed from the dataset. 'payer_code' has 40% missing values which is also removed since it is not a useful feature in terms of information gain. 'examide' and 'citoglipton' contains 'No' for all instances and hence removed. Medications other than 'insulin', 'metformin' is removed, since more than 75% of the patients were not administered these medications.
     - The dataset contains three string attributes 'diag1', 'diag2' and 'diag3'. This is converted to Nominal Using StringToNominal Filter.
     - It is useful to convert attributes 'admission_type_id','admission_source_id', 'discharge_disposition_id' to nominal attributes because these are IDs of certain events with associated mapping and we don't want them to be treated as numeric during clustering.
     - Missing Values are removed using ReplaceMissingValues filter similar to what is done in Part A.
     - Normalization was not performed, as in Weka, when choosing the distance function for clustering, we can set don'tNormalize to false. Thus normalization during preprocessing is not necessary. Standardization and discretization is also not performed. Age which is useful to be binned, is already discretized in the dataset.
     - Preprocessed file is saved as dataset_clustering. arff
  2. **K-Means Clustering**
     - Three experiments were performed using K-Means clustering on the dataset dataset_clustering. arff in Weka Explorer.Classes to Cluster Evaluation was used in all three of the experiments, to evaluate the performance of each cluster.
     - In experiment 1, with Euclidean distance measure, seed set to 10 and numClusters=2, 2 clusters are generated. Random initial points was used. Sum of Within cluster sum of squared errors is 1396.82. Classes

To Cluster evaluation is used (cs.ccsu.edu, nd), and it is identified that only 3969 classes with no risk of readmission is assigned to cluster 0 and 1518 instances is assigned to cluster 1. However, 3577 instances of patients with risk of readmission is assigned to cluster 0 and 1112 is assigned to cluster 0. 49.9312% of instances were incorrectly clustered. Cluster centroids are the mean vectors for each cluster. In the final cluster centroids, some of the factors that are evident are:

- Cluster 0- It is a Caucasion female aged 70-80 in the Internal Medicine speciality prescribed with diabetes medications, but no change of medication with 7.3366 number of diagnoses.
- Cluster 1- It is a Caucasion male aged 70-80 in the Internal Medicine speciality prescribed with diabetes medications, but no change of medication with 7.75 number of diagnoses.

From the visualization of cluster with colour as readmitted, it is clearly evident that cluster 1 is dominated by patients with no risk of readmission (1518 instances) and cluster 0 is also dominated by patients with no risk of readmission. (3969 instances), however this is slightly difficult to identify from the cluster visualization due to roughly same distribution. As shown by the final centroid and also in the visualization it can be identified that cluster 0 is dominated by female and cluster 1 is dominated by males.

- In experiment 2, few parameters are varied for developing good quality clusters. A different initialization algorithm called canopy clustering is used. It is an unsupervised pre-clustering algorithm that can be used as pre-processing step for K-Means (Wikipedia, Canopy Clustering Algorithm, 2020). In Weka, this pre-clustering algorithm is available. Minimum canopy density is set to 2 and canopy periodic pruning rate is set to default 10000. Using K-Means Algorithm with Euclidean distance measure, seed set to 10 and numClusters=2, 2 clusters are generated. It slightly improved the performance with radius T2 radius: 2.128 and T1 radius: 2.660. Sum of Within cluster sum of squared errors is reduced to 1230.151. This is due to the preclustering algorithm Canopy. It is identified that only 3979 classes with no risk of readmission is assigned to cluster 0 and 1508 instances is assigned to cluster 1. However, 3594 instances of patients with risk of readmission is assigned to cluster 0 and 1095 is assigned to cluster 0. 49.8624% of instances were incorrectly clustered. In the final cluster centroids, some of the factors that are evident are:
  - Cluster 0- It is a Caucasion female aged 70-80 in the Internal Medicine speciality prescribed with diabetes medications, but no change of medication with 7.3366 number of diagnoses.
  - Cluster 1- It is a Caucasion male aged 70-80 in the Internal Medicine speciality prescribed with diabetes medications, but no change of medication with 7.75 number of diagnoses.

  From the visualization of cluster with colour as readmitted, it is clearly evident that cluster 0 is dominated by patients with risk of readmission (3979 instances) and cluster 1 is dominated by patients with no risk of readmission. (1508 instances).
- In experiment 3, Manhattan distance function with seed set to 10 and all other parameters set to their default option will be used. This improved the classes to cluster evaluation results. Incorrectly clustered instances reduced to 46.079(4689 instances), a drop in approximately 3%. This can be due to the good performance of Manhattan with attributes of different kind. A good quality will have high intra-class similarity and low inter-class similarity. 2828 instances of patients with no risk of readmission (0) is assigned to cluster 0, 2659 to cluster 1 and 2030 instances of patients with risk of readmission (1) is assigned to cluster 0 and 2659 to cluster 1. However, sum

of within cluster distances is 6768.77 in this experiment, which is not good compared to Experiments 1 and 2. Centroids shows the characteristics of each cluster. Final centroids shows that:

- Cluster 0- Caucasion Female aged 60-70 in Internal Medicine with 35 lab procedures with 12 medications and without a change in medication etc.
- Cluster 1- Caucasion Female aged 70-80 in Internal Medicine with 54 lab procedures with 17 medications and with a change in medication etc.

Readmission attribute is visualized on both the clusters. The distribution of classes in cluster 1 seems to be roughly equal as we have seen in classes to cluster evaluation. Few clusters seem to be a bit far away from their group, while majority are clustered tightly. In the other visualization with colour as readmitted we can clearly see that cluster 0 consists of more patients with no risk of readmission (0) – 2828 and cluster 1 consists of equal instances of patients with risk of readmission and no risk of readmission(1) – 2659 instances .

- Weka's unsupervised filter 'AddCluster' is used with DBSCAN epsilon 2.2 and minPoints=10, to visualize all of the attributes and their distribution in each of the 2 clusters.
- The result buffers and models of all K-Means Experiments is stored in the folder 'K-MEANS RESULT BUFFERS AND MODELS'.
- All the cluster visualizations are stored as 'K-Means-Experiment1-Cluster.arff', 'K-Means-Experiment2-Cluster.arff', 'K-Means-Experiment3-Cluster.arff'

3. DBSCAN Clustering

- DBSCAN is a density based clustering algorithm which is extremely sensitive to epsilon and minPoints. In the first experiment, DBSCAN algorithm with epsilon =0.9 and minPoints= 6 generated only 1 cluster. If p is a core point, then it forms a cluster together with all points (core or non-core) that are reachable from it, and If a point is density-reachable from some point of the cluster, it is part of the cluster as well. So, in this case all of the points are grouped together to one cluster. There is no NOISE point, 100% of instances are assigned to cluster 0. 46.079% of instances are incorrectly classified with classes to cluster evaluation.
- In experiment 2, IBk LinearNN search was performed to identify option k nearest neighbours for estimating epsilon. Choosing epsilon 9 and minPoints 40 with Manhattan distance didn't make a difference in this experiment. There is no NOISE point, 100% of instances are assigned to cluster 0. 46.079% of instances are incorrectly classified with classes to cluster evaluation
- In experiment 3, **Epsilon is set to 2.2** which is the optimum epsilon that has to be used. This epsilon was chosen after going through several journals which is mentioned in the references section and minPoints is set to 10. Euclidean distance measure is set to first-last for the algorithm to choose to deal with the attributes accordingly. 2 clusters are produced. There are 1198 unclustered instances. There are few NOISE points. 8968 instances are assigned to cluster 0 and 10 instances are assigned to cluster 1. From the classes to cluster evaluation, it is evident that patients with no risk of readmission(0) are assigned to cluster 0 4822 instances and to cluster 1, 6 instances. Patients with risk of readmission(1) are assigned to cluster 0 4146 instances and to cluster 1, 4 instances. Thus, 40.8019% of instances are incorrectly clustered. Given the complexity of the dataset, this is not a poor clustering. From the visualization, we can see that in cluster 1 there are few points, and it is mostly dominated by patients with no risk of readmission. This is the best performance that was achieved.
- All of the experiments are explained in detail in respective sections. From both K-Means and DBSCAN, it can be understood that most of our data behaves similarly. There are low density groups that gets combined with large density ones when epsilon is varied. Even when 2 clusters are

produced using K-Means, there was tendency for the clusters to behave similarly when it comes to attributes like 'race', 'age', 'gender' etc. DBSCAN produced 2 clusters with epsilon 2.2 and minPoints 10, however, most of the points were assigned to cluster 0.

- The result buffers and models of all K-Means Experiments is stored in the folder 'DBSCAN RESULT BUFFERS AND MODELS'.
- All the cluster visualizations are stored as 'DBSCAN-Experiment1-Cluster.arff', 'DBSCAN-Experiment2-Cluster.arff', 'DBSCAN - Experiment3-Cluster.arff'

2. Preprocessing – 10%

The preprocessing steps used in Clustering slightly differ from the ones used in Classification because the objective of clustering is different from classification. The following preprocessing techniques and their appropriateness was verified. The dataset that is used for clustering is diabetes_resampled.arff with 10176 instances.

1. Feature Encoding

Readmitted have values: "<30" if the patient was readmitted in less than 30 days, ">30" if the patient was readmitted in more than 30 days, and "No" for no record of readmission. Since the objective of this classification is to predict the readmission, it is useful to convert the readmission feature to 1 for >30 and <30 and 0 for 'NO'. This step was performed in Excel before all other preprocessing steps. This step is performed in the original dataset for Part 1.

- 0 - No risk of readmission
- 1 -Risk of readmission within or after 30 days

2. Feature Selection

Some attributes are removed from the dataset for better clustering. 'encounder_id' and patient_nbr' contains all 10176 unique values, thus they are not useful for classification. 'weight' feature contains 97% missing values. It is thus not useful for data mining and is removed from the dataset. 'payer_code' has 40% missing values which is also removed since it is not a useful feature in terms of information gain. 'examide' and 'citoglipton' contains 'No' for all instances and hence removed. Medications other than 'insulin', 'metformin' is removed, since more than 75% of the patients were not administered these medications. 25 attributes are selected for further preprocessing.

3. Outliers

Outlier is a data point that differs significantly from other observations which can be due to error in measurements or exceptional cases (Wikipedia Outlier, 2020). Outliers and extreme values in the dataset are detected using Weka's unsupervised attribute InterQuartileRange Filter. It gives us the middle spread of the data. This filter skips the class values. It creates two new features Outlier and ExtremeValue with two distinct values 'No' and 'Yes' for all instances.

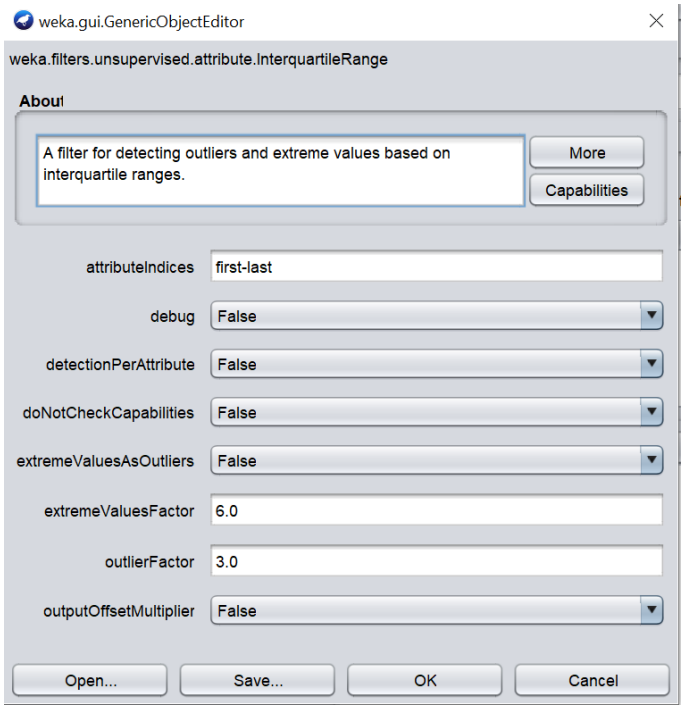


Figure 82 InterQuartileRange Filter

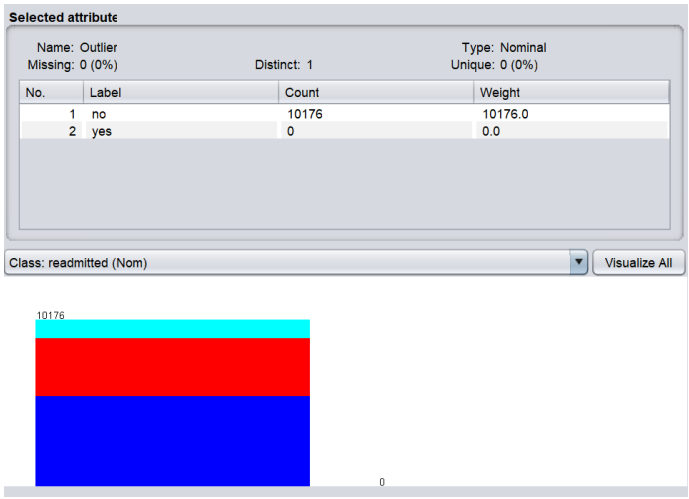


Figure 83 Outlier

4. Normalization

Normalization can be performed using Weka's normalize filter. This ensures that numerical values are scaled by removing the different min and max values and scaling them down to a standard range which makes certain machine learning algorithms work better etc (Dr. Abubakr Siddig- Datasets,EDA and altering data structure lecture , 2020). For clustering, normalization is a useful procedure since k-means algorithms work with numerical data. This improves k-Means distance computation because each dimension will be weighted equally (EduPristine, 2020). However, our dataset contains numeric attributes which are IDs which need not be normalized and then few other numerical attributes like 'time_in_hospital' , 'num_lab_procedures', 'num_procedures' etc. However, Normalization didn't improve or decrease clustering performance. EuclideanDistance and ManhattanDistance uses normalization by default. Hence, it is not applied.

5. Standardization

Standardization is the process of multiplying each variable by a constant. This is useful to cluster analysis and self-organizing maps, as these algorithms implicitly weight variables according to their range (Data Preparation For Cluster-Based Segmentation – Displayr, 2020).However, just like normalization because of the nature of the dataset which contains numeric attributes which are IDs which need not be normalized and then few other numerical attributes like 'time_in_hospital' , 'num_lab_procedures',

'num_procedures' etc standardization didn't improve or worsen clustering sum of squared error. Hence it is not applied.

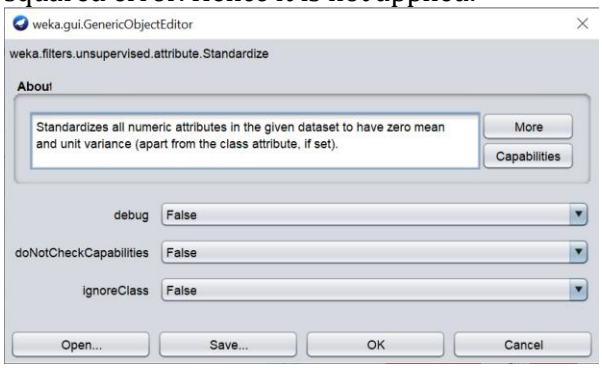


Figure 84 Standardize filter in Weka

6. Data Type Conversion

Clustering algorithms work with inte-rval-scaled variables, binary variables, nominal, ordinal, ratio and variables of mixed type (Dr. Abubakr Siddig- Clustering lecture , 2020). Clustering algorithms like KMeans don't work with string attributes. The dataset contains three string attributes 'diag1', 'diag2' and 'diag3'. This is converted to Nominal Using StringToNominal Filter.

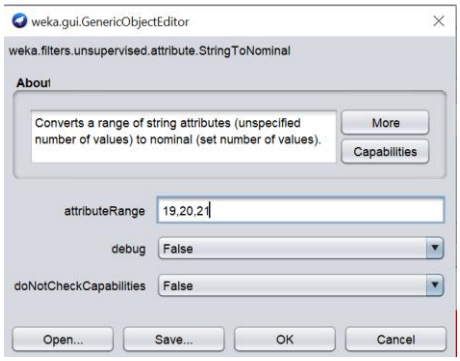


Figure 85 StringToNominal

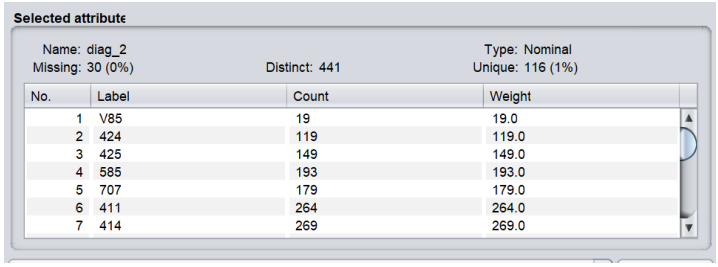


Figure 86 After StringToNominal Conversion

It is useful to convert attributes 'admission_type_id','admission_source_id', 'discharge_disposition_id' to nominal attributes because these are IDs of certain events with associated mapping and we don't want them to be treated as numeric during clustering. Nominal variables are not converted because K-Means can automatically deal with it, even though distance base computation is performed on numerical variables.

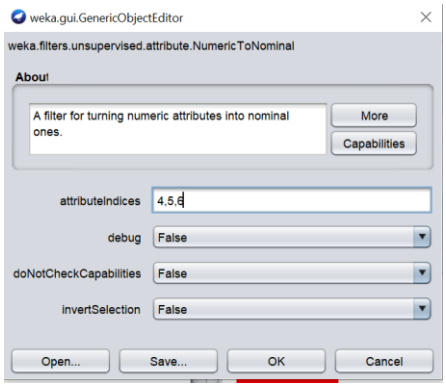


Figure 87 Numeric to Nominal

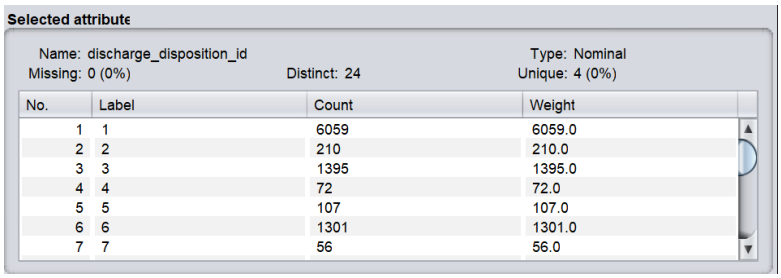


Figure 88 After NumericToNominal Conversion

7. Discretization (Binning)

Binning may improve the accuracy of predictive models by reducing the non-linearity or noise and it is also useful for certain classifiers. After binning, outliers, missing or invalid values can be easily identified. This can be performed using Weka's Discretize filter. However, for this dataset age is a useful feature to be binned, and that is already done in the dataset. Hence the discretize filter need not be applied.

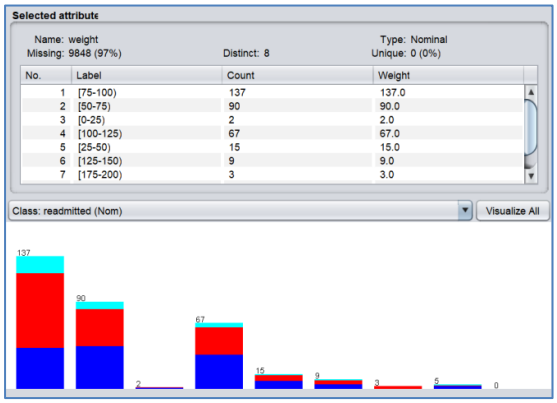


Figure 89 categorised age

8. Missing Values

Data can be missing due to a variety of reasons- hesitance of the respondents to provide complete information, malfunctioning of equipments, errors when entering the data in to the database, sudden changes etc (Dr. Abubakr Siddig- Datasets,EDA and altering data structure lecture , 2020). A small amount of missing value is almost unavoidable in large datasets. However, a significant percentage of missing values can be problematic. Missing Values are removed using ReplaceMissingValues filter similar to what is done in Part A.

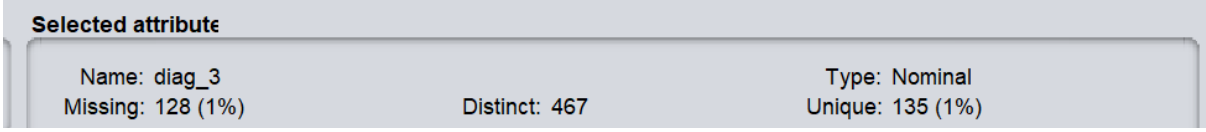


Figure 90 Before and after applying ReplaceMissingValues Filter

This file is saved as **dataset_clustering.arff.**

Experiments

For each of the following 2 clustering techniques

1. Use dataset.arff as input. If adaptions are necessary clearly indicate them.
2. Write one or two paragraph analyzing the results of the clustering. Be sure to vary parameters at least 3 times in each case. Support this analysis with screenshots of the following
    a. The clusters and/or a visualization of the clusters
    b. The results of the clusters
    c. Any additional output of the clustering process
    d. Simple references to the notes or URL links to online resources complete with a sentence or two of explanation.
    e. Evaluate the clusters using the "classes to clusters evaluation". A worked example may be found here http://www.cs.ccsu.edu/~markov/ccsu_courses/datamining-ex3.html

For all of the clustering experiments, dataset_clustering.arff is used as input.

2.1. Clustering: K-Means – 10%

K-Means is a relatively efficient algorithm that often terminates at local optimum and handles numerical data. For categorical data, it replaces means of clusters with modes. Number of clusters need to be specified earlier. For a mixture of categorical and numerical data, k-prototype method is used. (Dr. Abubakr Siddig- Clustering lecture , 2020)

2.1.1. Experiment- 1

For the first experiment, default settings of SimpleK-Means will be used.

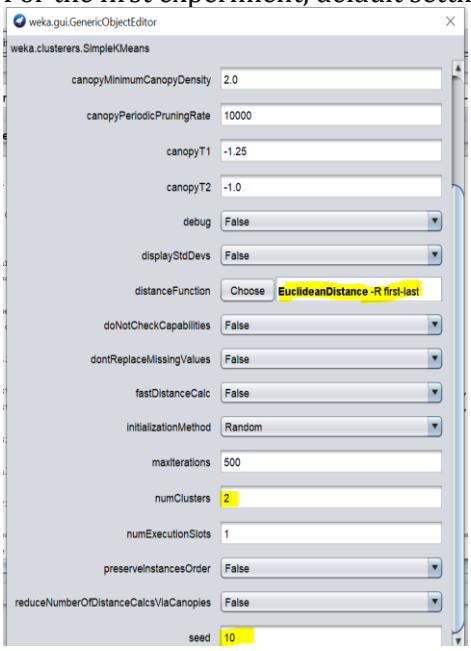


Figure 91 K-Means Experiment 1

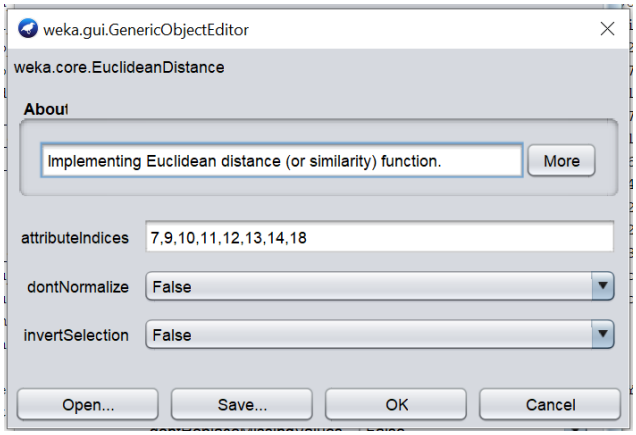
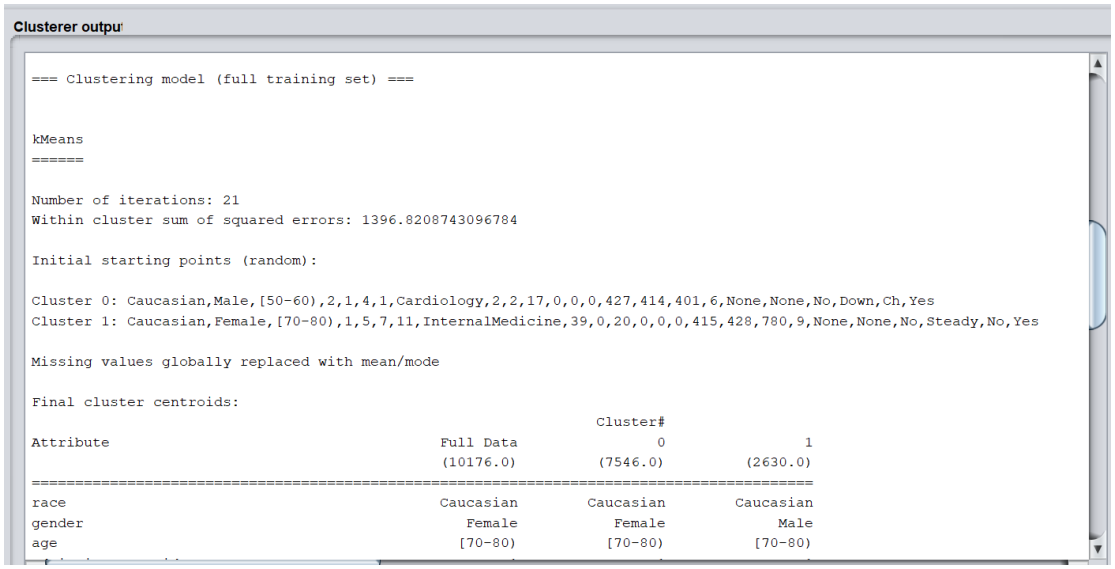
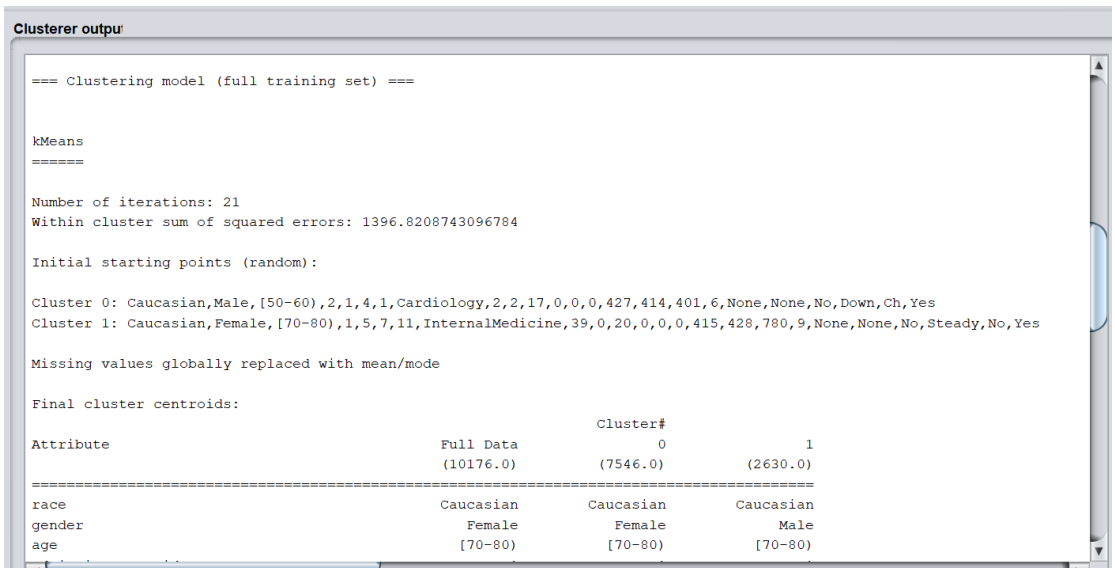


Figure 92 Euclidean distance for numerical variables

The distance function used is Euclidean distance is straight-line distance between two points in Euclidean space. Euclidean distance here is used for the numerical attributes with indices as shown above. Without this step, sum of squared errors was drastically high (72000) compared to 1360 obtained now, because K-Means distance computation requires numerical attributes. dontNormalize is set to False because we need these numerical attributes to be scaled, since that is not done beforehand. Seed is set to 10 and numClusters is set to 2 for generating 2 clusters.

Result





Clusterer output

admission_type_id	1	1	1
discharge_disposition_id	1	1	1
admission_source_id	7	7	7
time_in_hospital	4.3888	3.8832	5.8392
medical_specialty	InternalMedicine	InternalMedicine	InternalMedicine
num_lab_procedures	43.2494	41.9206	47.062
num_procedures	1.3343	0.4779	3.7916
num_medications	16.0216	14.2167	21.2004
number_outpatient	0.3673	0.3757	0.3433
number_emergency	0.1977	0.2179	0.1399
number_inpatient	0.6274	0.6664	0.5152
diag_1	414	428	414
diag_2	276	276	411
diag_3	250	250	250
number_diagnoses	7.4439	7.3366	7.7517
max_glu_serum	None	None	None
A1Cresult	None	None	None
metformin	No	No	No
insulin	No	No	No
change	No	No	No
diabetesMed	Yes	Yes	Yes

Time taken to build model (full training data) : 0.12 seconds

=== Model and evaluation on training set ===

Clustered Instances

0 7546 (74%)
1 2630 (26%)

Figure 93 K-Means - Experiment 1- Result

```
Class attribute: readmitted
Classes to Clusters:

    0    1  <-- assigned to cluster
 3969 1518 | 0
 3577 1112 | 1

Cluster 0 <-- 1
Cluster 1 <-- 0

Incorrectly clustered instances :       5081.0    49.9312 %
```

Figure 94 Classes to Cluster Evaluation

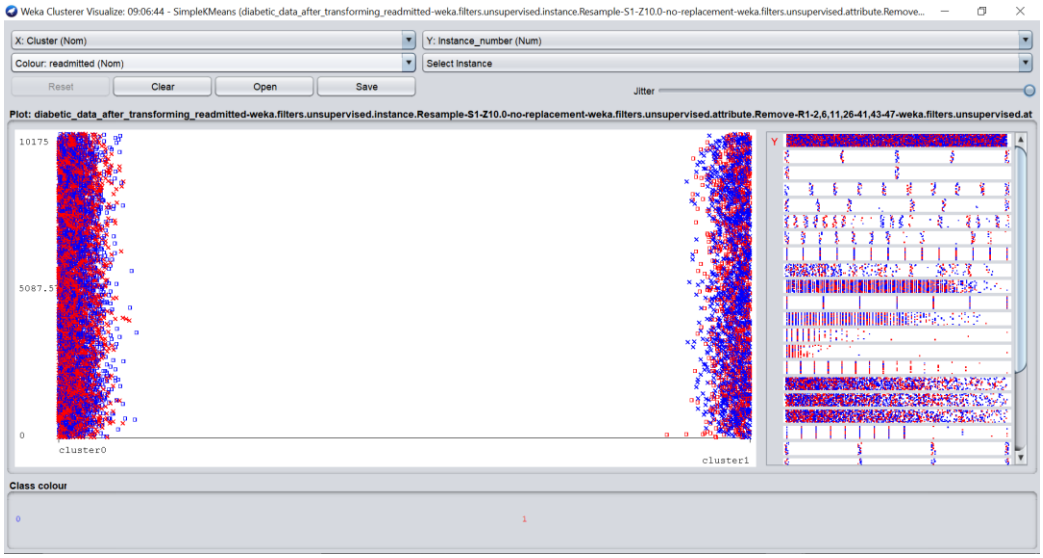


Figure 95 Clusters with colour as readmitted

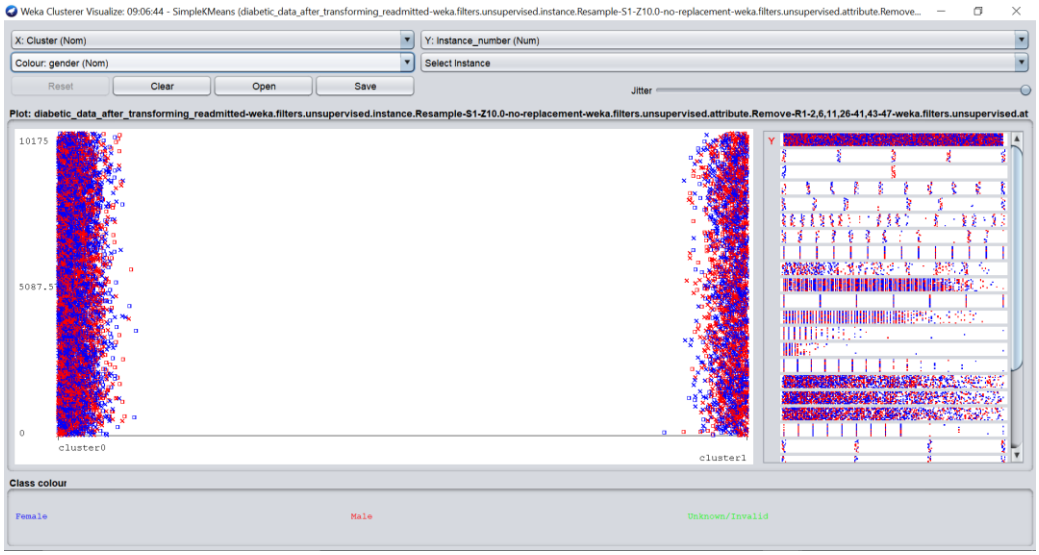


Figure 96 Clusters with colour as gender

Findings

1. Using K-Means Algorithm with Euclidean distance measure, seed set to 10 and numClusters=2, 2 clusters are generated. Random initial points was used.
   Cluster 0: Caucasian, Male, [50-60), 2, 1, 4, 1, Cardiology, 2, 2, 17, 0, 0, 0, 427, 414, 401, 6, None, None, No, Down, Ch, Yes
   Cluster 1: Caucasian, Female,[70-80), 1, 5, 7, 11, InternalMedicine, 39, 0, 20, 0, 0, 0, 415, 428, 780, 9, None, None, No, Steady, No, Yes
2. Sum of Within cluster sum of squared errors: 1396.82. This error have to be reduced in further experiments. A good clustering method will produce high quality clusters with high intra-class similarity(Dr. Abubakr Siddig- Clustering lecture , 2020).
3. Classes To Cluster evaluation is used (cs.ccsu.edu, nd), and it is identified that only 3969 classes with no risk of readmission is assigned to cluster 0 and 1518 instances is assigned to cluster 1. However, 3577 instances of patients with risk of readmission is assigned to cluster 0 and 1112 is assigned to cluster 0. 49.9312% of instances were incorrectly clustered. Thus this has to be improved.
4. Cluster centroids are the mean vectors for each cluster. In the final cluster centroids, some of the factors that are evident are:
   - Cluster 0- It is a Caucasion female aged 70-80 in the Internal Medicine speciality prescribed with diabetes medications, but no change of medication with 7.3366 number of diagnoses.

- Cluster 1- It is a Caucasion male aged 70-80 in the Internal Medicine speciality prescribed with diabetes medications, but no change of medication with 7.75 number of diagnoses.

5. From the visualization of cluster with colour as readmitted, it is clearly evident that cluster 1 is dominated by patients with no risk of readmission (1518 instances) and cluster 0 is also dominated by patients with no risk of readmission. (3969 instances), however this is slightly difficult to identify from the cluster visualization due to roughly same distribution.
6. As shown by the final centroid and also in the visualization it can be identified that cluster 0 is dominated by female and cluster 1 is dominated by males.

2.1.2. Experiment 2

In experiment 2, few parameters are varied for developing good quality clusters. A different initialization algorithm called canopy clustering is used. It is an unsupervised pre-clustering algorithm that can be used as pre-processing step for K-Means (Wikipedia, Canopy Clustering Algorithm, 2020). In Weka, this pre-clustering algorithm is available. Minimum canopy density is set to 2 and canopy periodic pruning rate is set to default 10000.

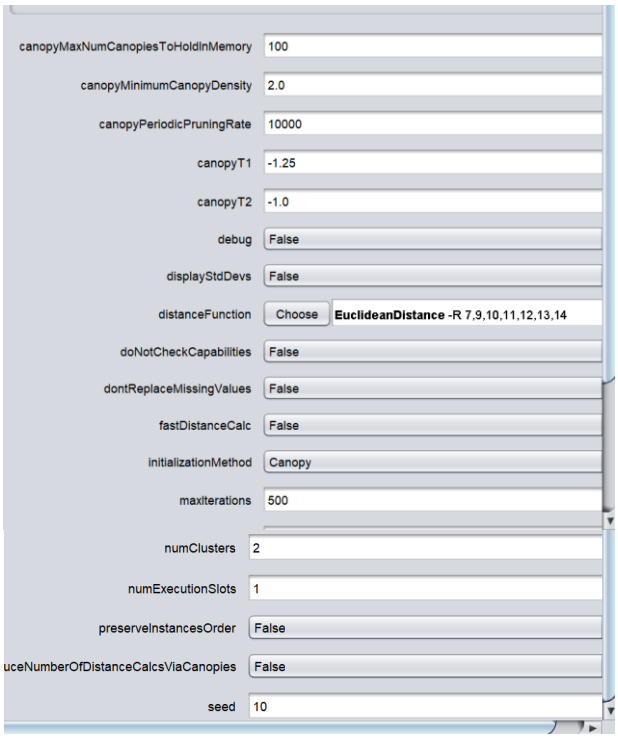


Figure 97 K- Means with canopy initialization method

Results

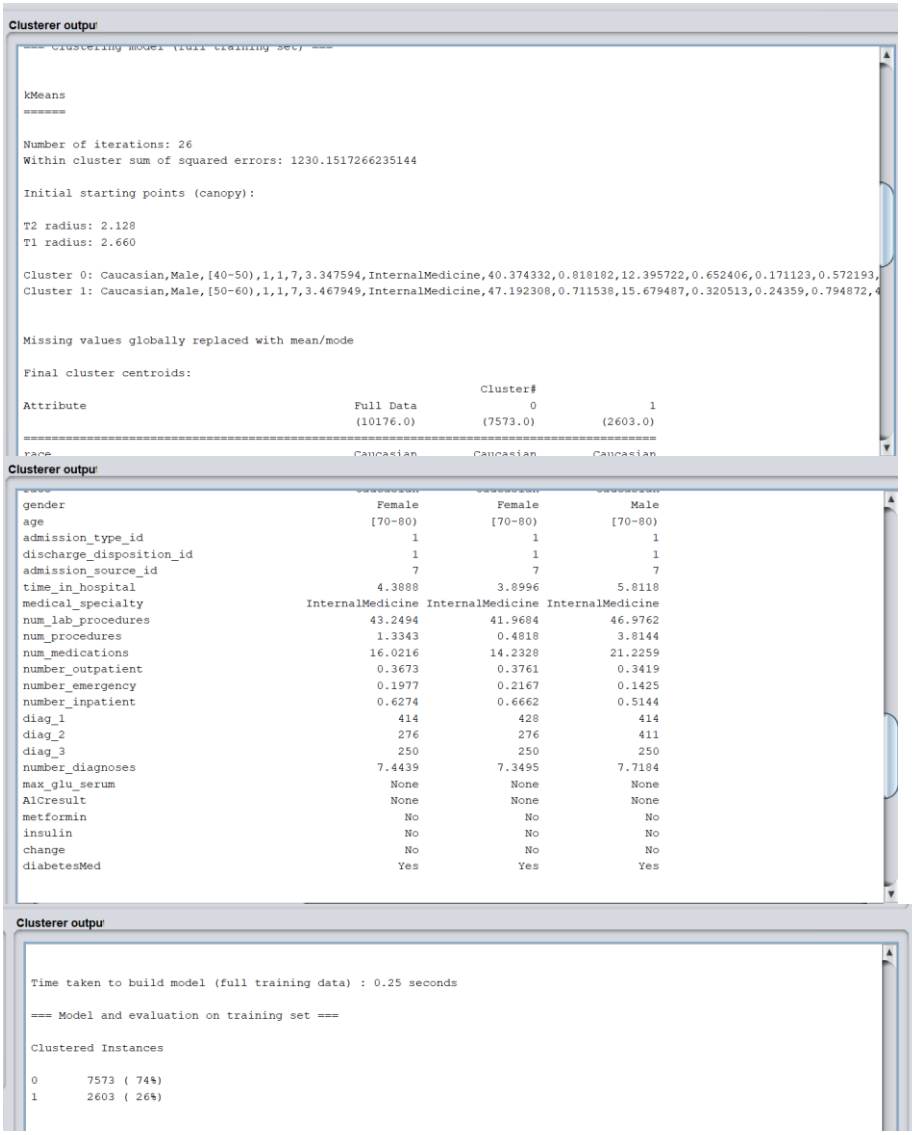


Figure 98 K-Means- Experiment 2- Result

```
Class attribute: readmitted
Classes to Clusters:

    0    1  <-- assigned to cluster
 3979 1508 | 0
 3594 1095 | 1

Cluster 0 <-- 1
Cluster 1 <-- 0

Incorrectly clustered instances :       5074.0    49.8624 %
```

Figure 99 K-Means- Experiment 2- Classes to Clusters

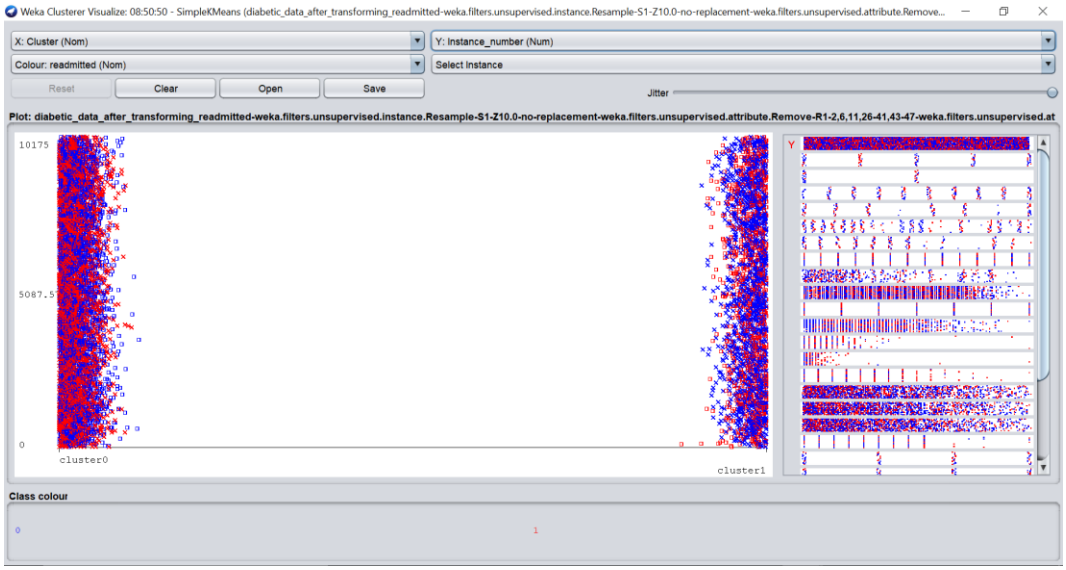


Figure 100 Cluster with colour as readmitted

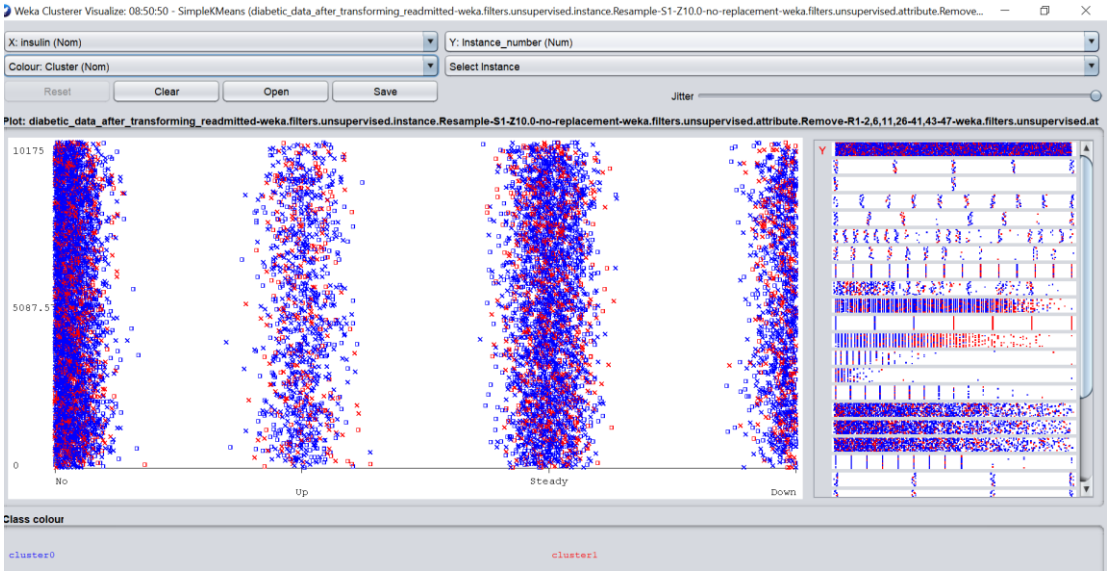


Figure 101 Insulin clusters

Findings

1. Using K-Means Algorithm with Euclidean distance measure, seed set to 10 and numClusters=2, 2 clusters are generated. Canopy initial method is used and it slightly improved the performance with radius:
   T2 radius: 2.128
   T1 radius: 2.660
   Cluster 0: Caucasian,Male,[40-50), 1, 1, 7, 3.347594, InternalMedicine, 40.374332, 0.818182, 12.395722 , 0.652406, 0.171123, 0.572193, 428,427,401, 7.695187,None,None,No,No,No,No,{187} <0,1>
   Cluster 1: Caucasian,Male,[50-60), 1, 1, 7, 3.467949, InternalMedicine, 47.192308, 0.711538, 15.679487, 0.320513, 0.24359, 0.794872, 428,427,276,7.865385,None,None,No,Down,Ch,Yes,{156} <0,1>
2. Sum of Within cluster sum of squared errors is reduced to 1230.151. This is due to the preclustering algorithm Canopy.
3. Classes To Cluster evaluation is used (cs.ccsu.edu, nd), and it is identified that only 3979 classes with no risk of readmission is assigned to cluster 0 and 1508 instances is assigned to cluster 1. However, 3594 instances of patients with risk of readmission is assigned to cluster 0 and 1095 is assigned to cluster 0. 49.8624% of instances were incorrectly clustered.
4. In the final cluster centroids, some of the factors that are evident are:

- Cluster 0- It is a Caucasion female aged 70-80 in the Internal Medicine speciality prescribed with diabetes medications, but no change of medication with 7.3366 number of diagnoses.
- Cluster 1- It is a Caucasion male aged 70-80 in the Internal Medicine speciality prescribed with diabetes medications, but no change of medication with 7.75 number of diagnoses.

5. From the visualization of cluster with colour as readmitted, it is clearly evident that cluster 0 is dominated by patients with risk of readmission (3979 instances) and cluster 1 is dominated by patients with no risk of readmission. (1508 instances).
6. From the visualization, there are very few instances with insulin Up where as most of instances are either no for insulin or have steady levels. This is the case for both the clusters.

2.1.3. Experiment 3

In the previous experiments, Euclidean distance was used as a similarity function. Manhattan distance is another similarity function that can be used with K-Means in Weka. Manhattan distance is calculated using:

d(i, j)=|x i1 -x j1 |+|x i2 -x j2 |+...+|x ip -x jp | where i = (xi1, xi2, …, xip) and j = (xj1, xj2, …, xjp) are two p-dimensional data objects, and q is a positive integer. (Dr. Abubakr Siddig-Clustering lecture , 2020).

In this experiment, Manhattan distance function with seed set to 10 and all other parameters set to their default option will be used.

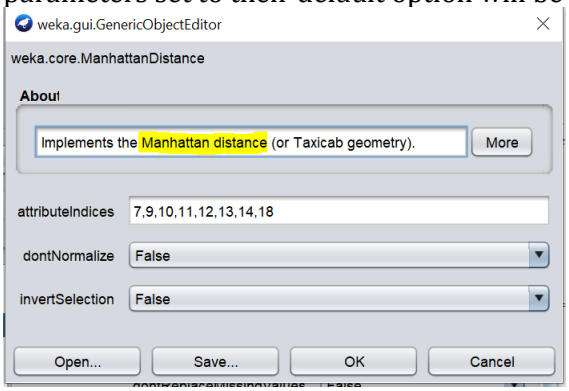


Figure 102 Manhattan distance

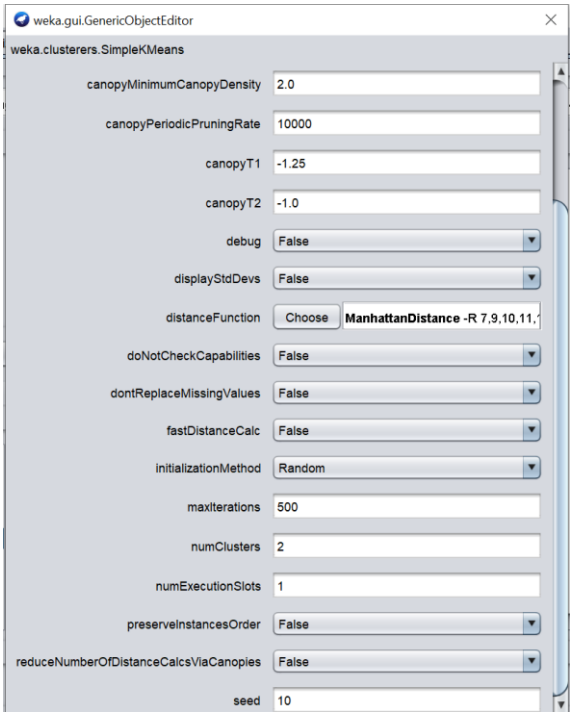
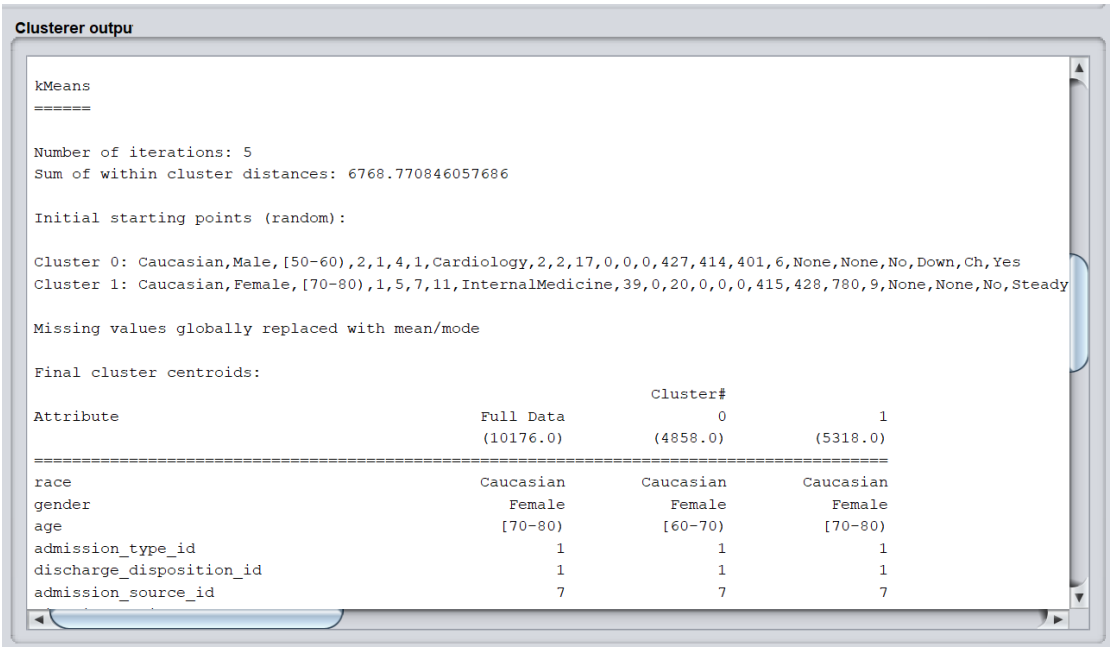


Figure 103 K-Means Experiment 3 parameters

Results



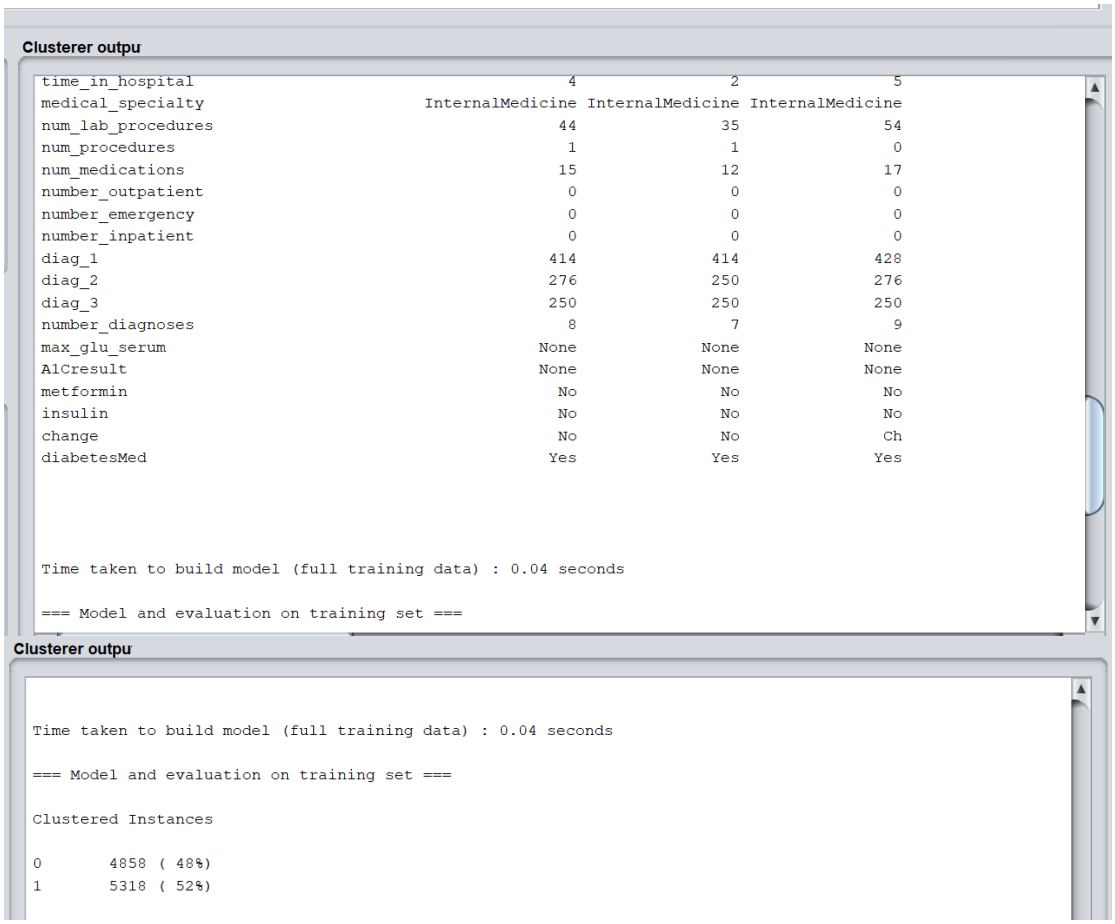


Figure 104 K-Means Experiment 3 Result

```
Class attribute: readmitted
Classes to Clusters:

    0    1  <-- assigned to cluster
 2828 2659 | 0
 2030 2659 | 1

Cluster 0 <-- 0
Cluster 1 <-- 1

Incorrectly clustered instances :       4689.0    46.079  %
```

Figure 105 K- Means Experiment 3 Classes to Cluster Evaluation

In order to see standard deviation, displayStdDevs is set to True which gives the following results.

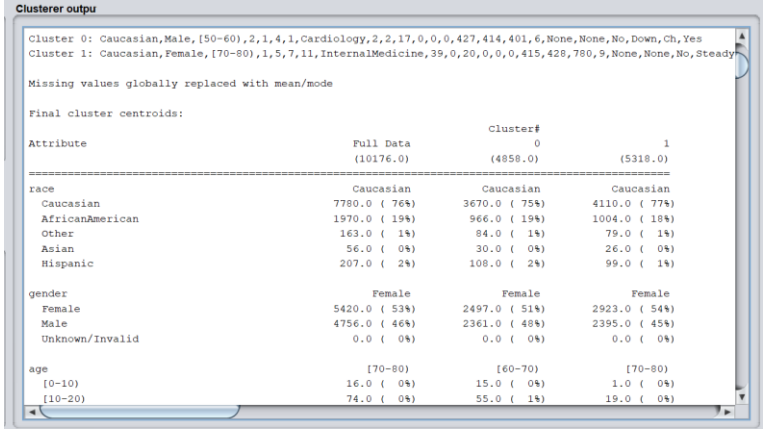


Figure 106 K-Means results with standard deviation

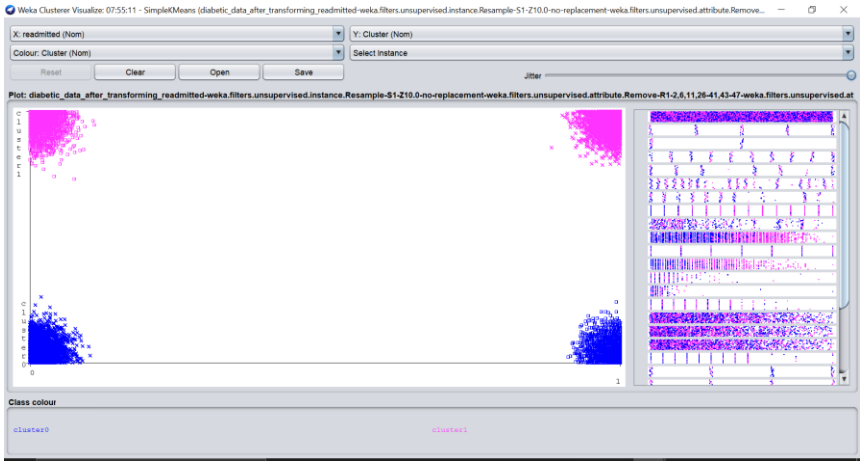


Figure 107 Cluster visualizations with readmission on X axis and cluster on Y axis

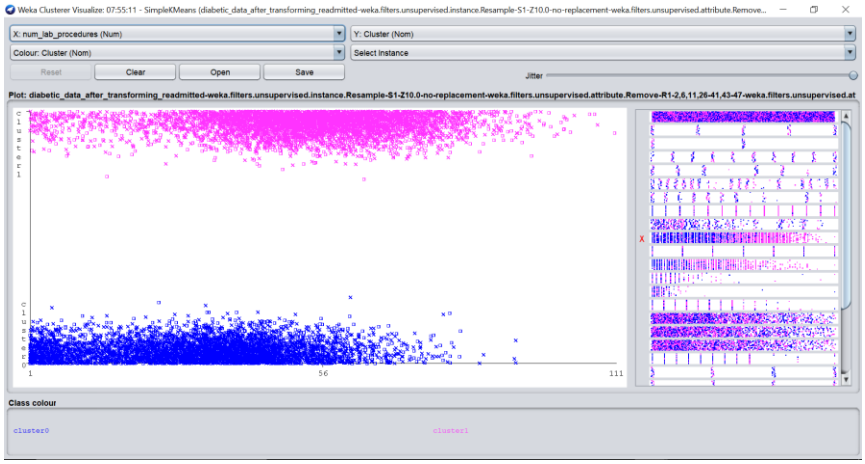


Figure 108 Cluster: num_lab_procedures vs Cluster

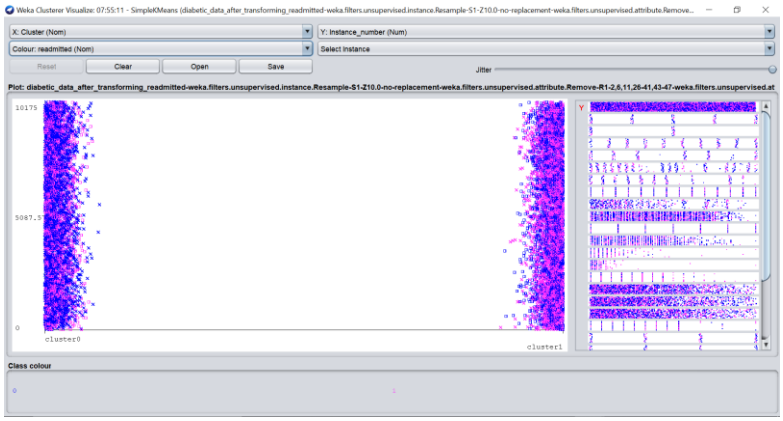


Figure 109 Cluster with colour as readmission

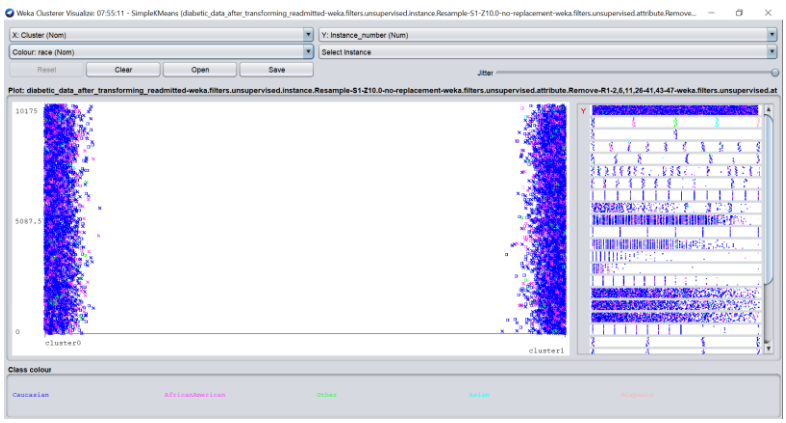


Figure 110 Cluster with race as colour

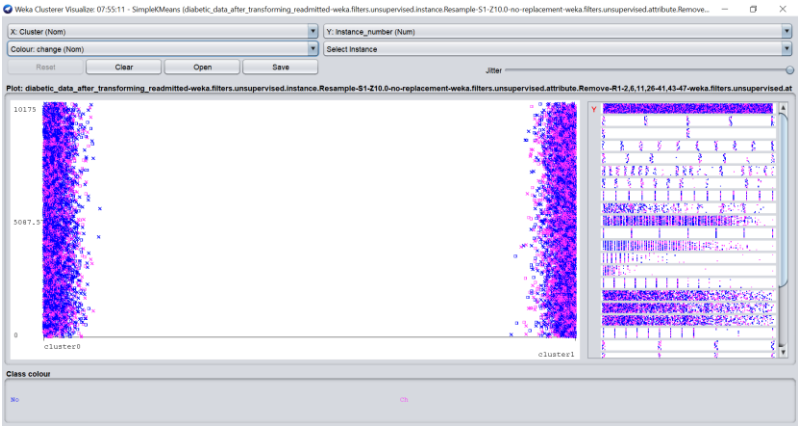


Figure 111 Cluster with change as colour

Findings

1. Manhattan distance function is used as a similarity function with seed set to 10. There were 5 iterations
   Initial starting points were randomly selected:
   Cluster 0: Caucasian,Male,[50-60), 2, 1, 4, 1, Cardiology, 2, 2, 17, 0 ,0 ,0, 427, 414 ,401, 6,None,None,No,Down,Ch,Yes
   Cluster1: Caucasian,Female,[70-80),1,5,7,11,InternalMedicine,39,0,20,0,0,0, 415,428,780,9,None,None,No,Steady,No,Yes
2. This improved the classes to cluster evaluation results. Incorrectly clustered instances reduced to 46.079(4689 instances), a drop in approximately 3%. This can be due to the good performance of Manhattan with attributes of different kind. A good quality will have high intra-class similarity and low inter-class similarity. 2828 instances of patients with no risk of readmission (0) is assigned to cluster 0, 2659 to cluster 1 and 2030 instances of patients with risk of readmission (1) is assigned to cluster 0 and 2659 to cluster 1.
3. However, sum of within cluster distances is 6768.77 in this experiment, which is not good compared to Experiments 1 and 2.
4. Centroids shows the characteristics of each cluster. Final centroids shows that:
   - Cluster 0- Caucasion Female aged 60-70 in Internal Medicine with 35 lab procedures with 12 medications and without a change in medication etc.
   - Cluster 1- Caucasion Female aged 70-80 in Internal Medicine with 54 lab procedures with 17 medications and with a change in medication etc.
5. Readmission attribute is visualized on both the clusters. The distribution of classes in cluster 1 seems to be roughly equal as we have seen in classes to cluster evaluation. Few clusters seem to be a bit far away from their group, while majority are clustered tightly. In the other visualization with colour as readmitted we can clearly see that cluster 0 consists of more patients with no risk of readmission (0) – 2828 and cluster 1 consists of equal instances of patients with risk of readmission and no risk of readmission(1) – 2659 instances .
6. From the cluster visualization with colour as race, cluster 0 and cluster 1 is clearly dominated by Caucasians and Hispanic is the least present race. More number of lab procedures are included in cluster 1 which is also evident in the results (54 in cluster 1 when compared to 35 in cluster 0). And cluster 1 consists mainly of patients with 'change=Ch', i.e, change of medication. This is also indicated by the centroid.

2.1.4. AddCluster Filter

In Weka, unsupervised filter called AddCluster filter can be used to generate a new attribute called cluster as shown below.

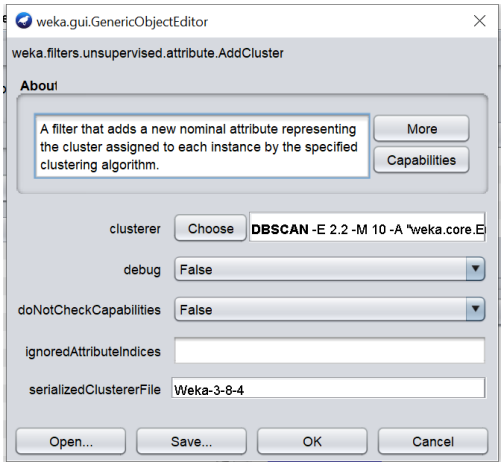


Figure 112 AddCluster Filter

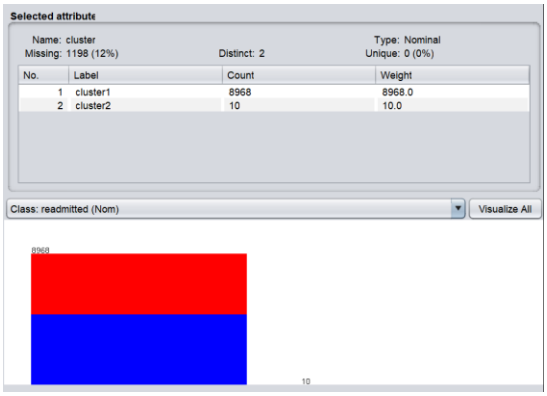


Figure 113 cluster attribute with graph

It can be seen that there are 1198 or12% unclustered instances. 8968 instances are assigned to cluster 1 and only 10 instances are assigned to cluster 2.

Figure 114 Figure 112 Visualization of all attributes according to cluster

From the above visualization, the distribution of each attribute in the clusters can be clearly seen.

2.2. Clustering: DBSCAN – 10%

Clustering based on density are useful for discovering clusters of arbitrary shapes, can handle noise. Clusters are dense regions in the data space, separated by regions of lower object density. DBSCAN in Weka uses clustering based on density. There are two parameters that are extremely important in DBSCAN. (Dr. Abubakr Siddig- Clustering lecture , 2020).

- ε-Neighborhood – Objects within a radius of εfrom an object.
- "High density" - ε-Neighborhood of an object contains at least minPoints of objects.

We will be varying these parameters in each of the three experiments.

There are three categories of points:

A point is a **core point** if it has more than a specified number of points (MinPts) within Eps—These are points that are at the interior of a cluster. **A border point** has fewer than minPoints within Eps, but is in the neighborhood of a core point. A **noise point** is any point that is not a core point nor a border point. (Dr. Abubakr Siddig- Clustering lecture , 2020).

Three experiments will be conducted on dataset_clustering.arff

2.2.1. Experiment 1

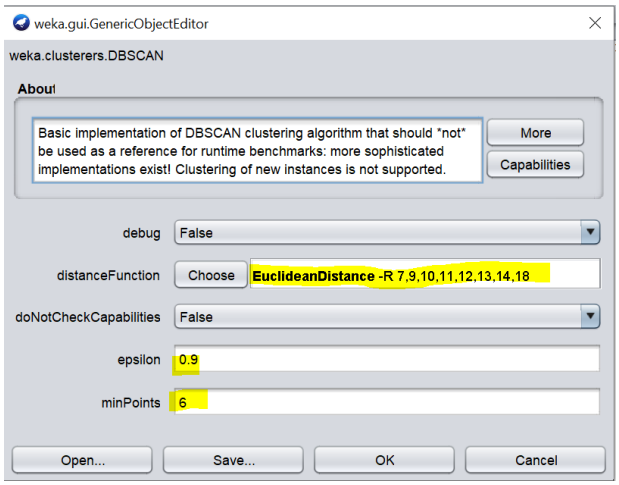


Figure 115 DBSCAN - Experiment 1

For this experiment, EuclideanDistance function is used with default values for epsilon and minPoints.

Result

Clusterer output

```
DBSCAN clustering results
==========================================================================================

Clustered DataObjects: 10176
Number of attributes: 24
Epsilon: 0.9; minPoints: 6
Distance-type:
Number of generated clusters: 1
Elapsed time: 24.28

(    0.) Caucasian,Female,[70-80),2,1,1,1,InternalMedicine,36,6,14,0,0,0,410,V  -->  0
(    1.) AfricanAmerican,Female,[80-90),1,1,7,2,InternalMedicine,12,0,9,0,0,0,  -->  0
(    2.) Caucasian,Female,[60-70),2,1,7,7,Emergency/Trauma,49,0,15,0,0,2,428,4  -->  0
(    3.) Caucasian,Male,[80-90),1,1,7,1,InternalMedicine,20,0,18,0,0,0,411,585  -->  0
(    4.) Caucasian,Female,[70-80),1,1,6,2,InternalMedicine,59,0,12,0,0,0,250.8  -->  0
(    5.) Other,Male,[60-70),1,2,7,3,InternalMedicine,56,3,15,0,0,0,414,411,530  -->  0
(    6.) Caucasian,Female,[50-60),1,1,4,9,InternalMedicine,52,3,20,0,0,0,444,4  -->  0
(    7.) Caucasian,Male,[60-70),2,22,7,2,Surgery-General,1,0,16,0,2,4,198,276,  -->  0
(    8.) AfricanAmerican,Female,[70-80),3,3,1,8,InternalMedicine,40,3,39,0,0,2  -->  0
(    9.) Caucasian,Male,[70-80),1,6,7,3,InternalMedicine,66,2,21,0,2,0,518,584  -->  0
(   10.) Caucasian,Female,[80-90),1,15,7,11,InternalMedicine,68,2,8,1,0,0,733,  -->  0
(   11.) Caucasian,Female,[70-80),1,6,7,7,InternalMedicine,70,0,26,0,0,1,466,2  -->  0
(   12.) Caucasian,Male,[80-90),2,1,7,3,Emergency/Trauma,39,0,7,0,0,1,780,276,  -->  0
(   13.) Caucasian,Male,[70-80),1,22,7,6,InternalMedicine,37,0,16,11,0,4,584,4  -->  0
(   14.) Caucasian,Male,[70-80),1,6,7,6,InternalMedicine,69,1,32,1,0,0,162,486  -->  0
```

Clusterer output

```
(10175.) AfricanAmerican,Male,[70-80),3,18,1,1,InternalMedicine,1,1,19,0,0,0,4  -->  0



Time taken to build model (full training data) : 24.28 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      10176 (100%)


Class attribute: readmitted
Classes to Clusters:

    0  <-- assigned to cluster
 5487 | 0
 4689 | 1

Cluster 0 <-- 0

Incorrectly clustered instances :       4689.0   46.079  %
```

Figure 116 Experiment 1 result with classes to cluster evaluation

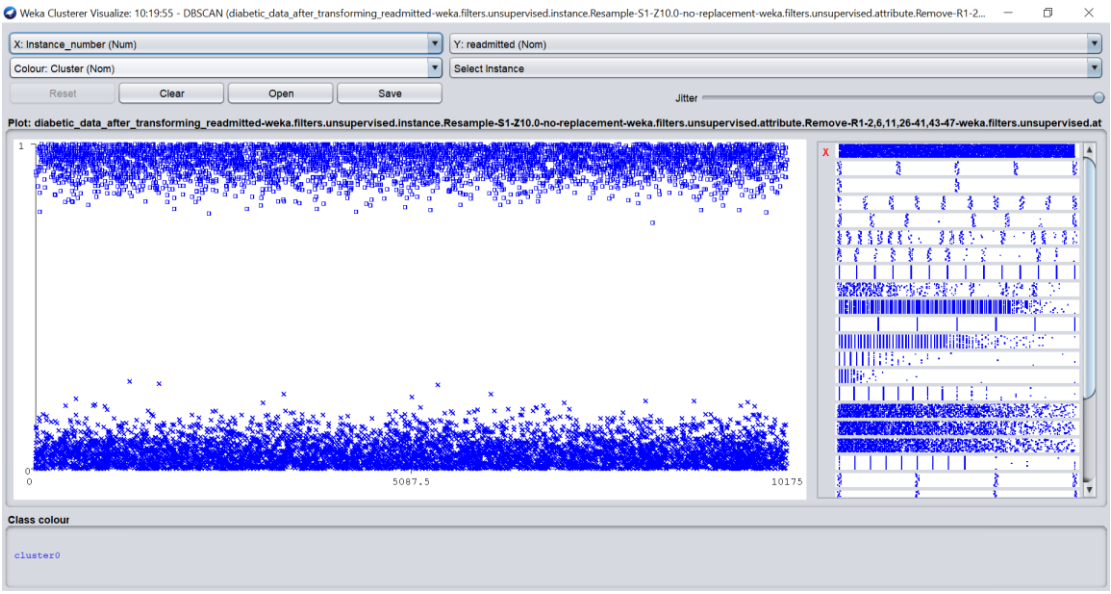


Figure 117 DBSCAN CLUSTER

Findings

1. Firstly, DBSCAN algorithm with epsilon =0.9 and minPoints= 6 generated only 1 cluster. If p is a core point, then it forms a cluster together with all points (core or non-core) that are reachable from it, and If a point is density-reachable from some point of the cluster, it is part of the cluster as well. So, in this case all of the points are grouped together to one cluster. This can be either due to the nature of the dataset with all points behaving similarly or due to the epsilon and minPoints value, which will be observed in further experiments.
2. There is no NOISE point, 100% of instances are assigned to cluster 0.
3. 46.079% of instances are incorrectly classified with classes to cluster evaluation.

2.2.2. Experiment 2

DBSCAN algorithm is highly dependent on the density in the data and the scale of the data. We have 24 attributes including class. This can also be the reason for production of just one single cluster, it is called 'curse of dimensionality' and mainly happens with Euclidean distance metric (Wikipedia DBSCAN, 2020). In order to generate more than one cluster it is useful to remove some attributes.

The preprocessed dataset contains 24 attributes. 'Ignore attributes' option is used to ignore few attributes that contain more than 75% of instances as No or None to see if there is a difference in the clustering algorithm's performance. Attributes containing more than 75% of the data as one value, may lead to identical behavior of the dataset and hence the formation of a single cluster. This can be verified by ignoring these attributes. The selected attributes as shown below are ignored.

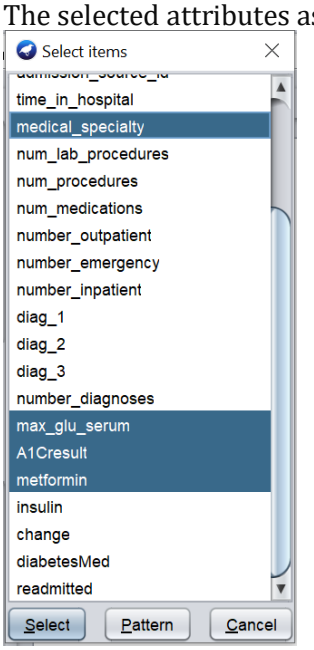


Figure 118 Ignoring few attributes

Now there are 20 attributes. DBSCAN algorithm is highly dependent on the distance function and parameter estimation for epsilon and minPoints. Choosing too small minPoints will create noise. The rule of thumb is to choose minPoints as 2*dimension (Wikipedia DBSCAN, 2020). So we can choose our minPoints as 2*20=40. In order to determine epsilon, we have to plot sorted distance of every point to its kth nearest neighbour. In order to find, the nearest neighbours IBk algorithm can be used with Manhattan distance and KNN=30 with crossvalidate= True, so optimal k will be determined by LinearNN search algorithm.

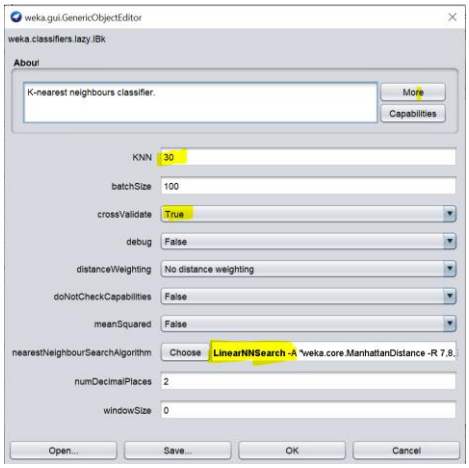
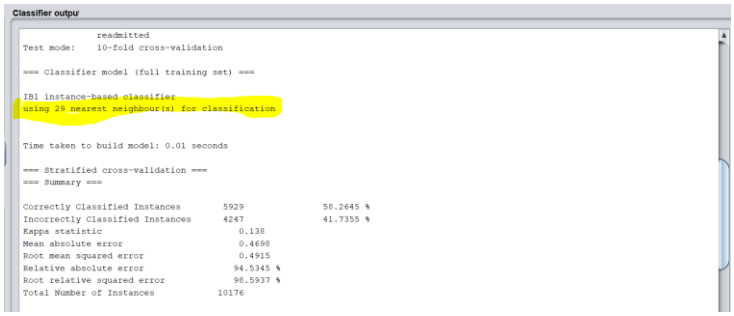


Figure 119 IBk with LinearNNSearch



The results show k=29 to be the optimum. K-dist plotting can determine the epsilon from the knee. Choosen epsilon is 9 in this experiment. The distance measure that will be used in this experiment is Manhattan distance as Manhattan distance is expected to work slightly better than Euclidean with high dimensional data.

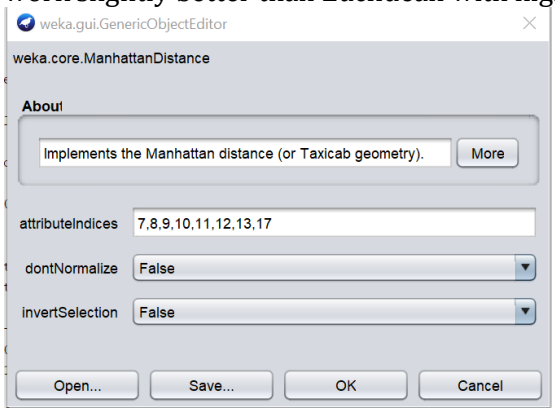


Figure 120 Manhattan distance

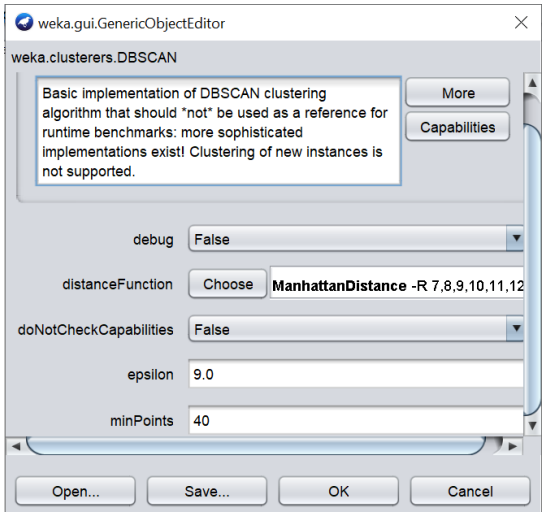


Figure 121 DBSCAN- Experiment 2

Results

```
=== Clustering model (full training set) ===

DBSCAN clustering results
========================================================================================

Clustered DataObjects: 10176
Number of attributes: 20
Epsilon: 9.0; minPoints: 40
Distance-type:
Number of generated clusters: 1
Elapsed time: 31.84

(    0.) Caucasian,Female,[70-80),2,1,1,1,36,6,14,0,0,0,410,V85,250,9,Steady,N  -->  0
(    1.) AfricanAmerican,Female,[80-90),1,1,7,2,12,0,9,0,0,0,250.8,424,780,7,U  -->  0
(    2.) Caucasian,Female,[60-70),2,1,7,7,49,0,15,0,0,2,428,425,427,9,No,No,Ye  -->  0
(    3.) Caucasian,Male,[80-90),1,1,7,1,20,0,18,0,0,0,411,585,428,9,Down,Ch,Ye  -->  0
(    4.) Caucasian,Female,[70-80),1,1,6,2,59,0,12,0,0,0,250.81,707,427,5,Stead  -->  0
(    5.) Other,Male,[60-70),1,2,7,3,56,3,15,0,0,0,414,411,530,8,Steady,No,Yes   -->  0
(    6.) Caucasian,Female,[50-60),1,1,4,9,52,3,20,0,0,0,444,414,440,5,No,No,No  -->  0
(    7.) Caucasian,Male,[60-70),2,22,7,2,1,0,16,0,2,4,198,276,304,9,No,Ch,Yes   -->  0
```

Figure 122 DBSCAN - Experiment 2- results

```
Time taken to build model (full training data) : 31.84 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      10176 (100%)


Class attribute: readmitted
Classes to Clusters:

    0  <-- assigned to cluster
 5487 | 0
 4689 | 1

Cluster 0 <-- 0

Incorrectly clustered instances :       4689.0    46.079  %
```

Figure 123 DBSCAN - Experiment 2- Results- Classes to Cluster

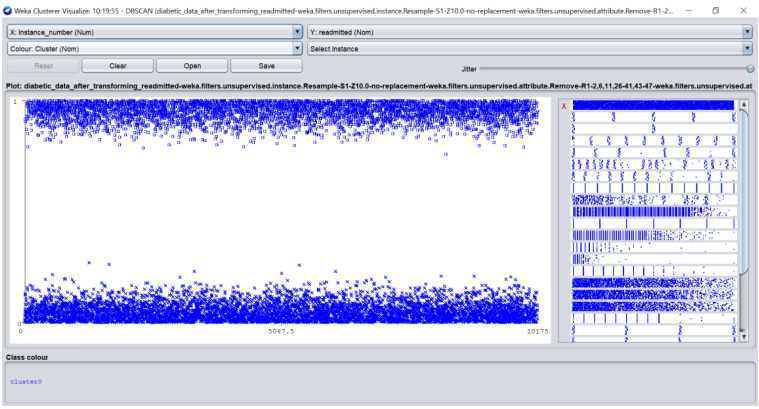


Figure 124 Cluster

Findings

1. Even after varying the parameters and ignoring certain attributes, DBSCAN could detect only one cluster. Choosing epsilon 9 and minPoints 40 with Manhattan distance didn't make a difference in this experiment.
2. There is no NOISE point, 100% of instances are assigned to cluster 0.
3. 46.079% of instances are incorrectly classified with classes to cluster evaluation.

2.2.3. Experiment 3

Until now, in both the experiments DBSCAN produced only 1 large cluster. This suggests that most of the data behaves similary, and low density groups get combined to the large density groups. In order to confirm this, in experiment 3 slight variations are made. **Epsilon is set to 2.2** which is the optimum epsilon that has to be used. This epsilon was chosen after going through several journals which is mentioned in the references section and minPoints is set to 10. Euclidean distance measure is set to first-last for the algorithm to choose to deal with the attributes accordingly.

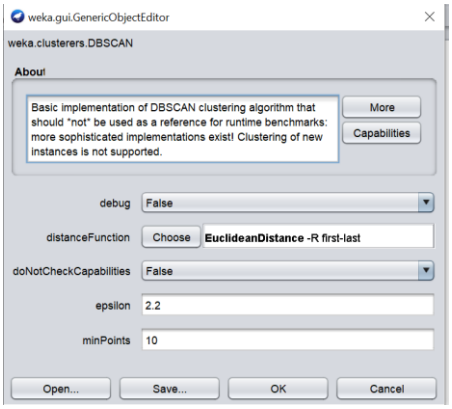


Figure 125 Experiment 3

Results

```
=== Clustering model (full training set) ===

DBSCAN clustering results
==========================================================================================

Clustered DataObjects: 10176
Number of attributes: 24
Epsilon: 2.2; minPoints: 10
Distance-type:
Number of generated clusters: 2
Elapsed time: 50.62

(    0.) Caucasian,Female,[70-80),2,1,1,1,InternalMedicine,36,6,14,0,0,0,410,V  -->  0
(    1.) AfricanAmerican,Female,[80-90),1,1,7,2,InternalMedicine,12,0,9,0,0,0,  -->  0
(    2.) Caucasian,Female,[60-70),2,1,7,7,Emergency/Trauma,49,0,15,0,0,2,428,4  -->  0
(    3.) Caucasian,Male,[80-90),1,1,7,1,InternalMedicine,20,0,18,0,0,0,411,585  -->  0
```

Figure 126 Experiment 3- DBSCAN - Result

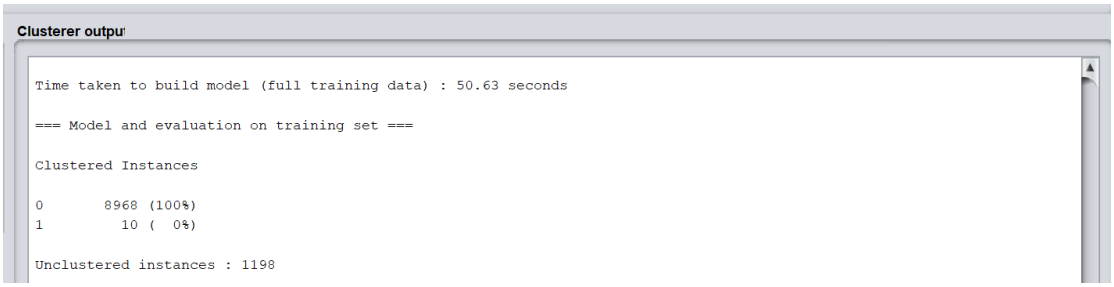


Figure 127 Experiment 3- DBSCAN - Result

```
Class attribute: readmitted
Classes to Clusters:

    0    1  <-- assigned to cluster
 4822    6 | 0
 4146    4 | 1

Cluster 0 <-- 0
Cluster 1 <-- 1

Incorrectly clustered instances :       4152.0    40.8019 %
```

Figure 128 Classes to Cluster Evaluation

Two clusters are generated when epsilon is set to 2.2.

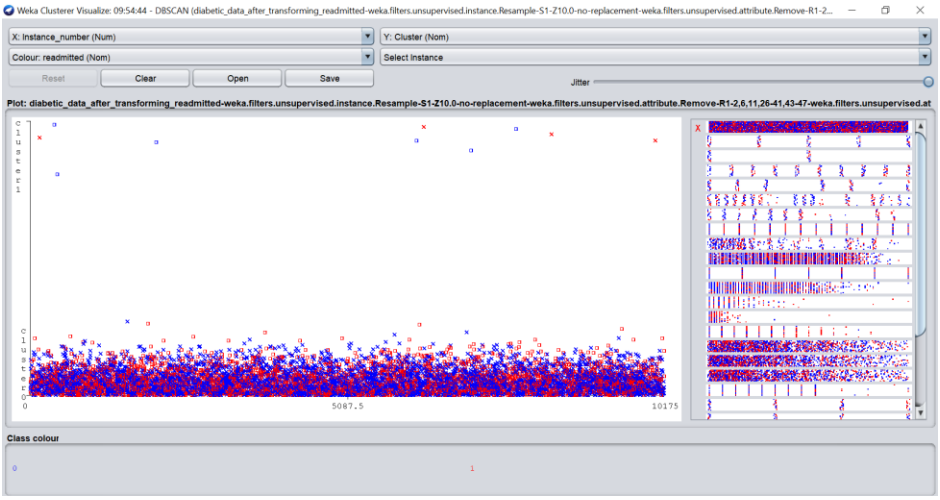


Figure 129 Cluster

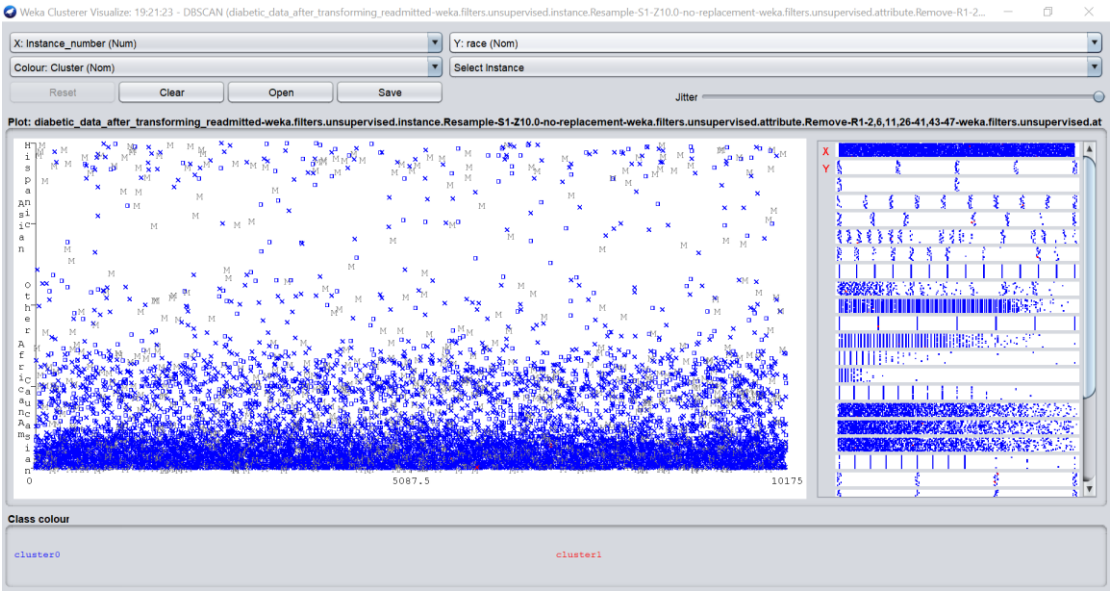


Figure 130 Clusters with race

Findings

1. **DBSCAN is extremely sensitive to parameters**, for any epsilon greater or lesser than 2.2, only one cluster is generated. With epsilon=2.2 and minPoints=10, 2 clusters are produced.
2. There are 1198 unclustered instances. There are few NOISE points.
3. 8968 instances are assigned to cluster 0 and 10 instances are assigned to cluster 1.
4. From the classes to cluster evaluation, it is evident that patients with no risk of readmission(0) are assigned to cluster 0 4822 instances and to cluster 1, 6 instances. Patients with risk of readmission(1) are assigned to cluster 0 4146 instances and to cluster 1, 4 instances.
5. Thus, 40.8019% of instances are incorrectly clustered. Given the complexity of the dataset, this is not a poor clustering.
6. From the visualization, we can see that in cluster 1 there are few points, and it is mostly dominated by patients with no risk of readmission. It can also be seen that race 'Caucasian' dominates cluster 0.

Part 3 - Overall Evaluation

1. Report Quality and presentation of knowledge 10%

2. References – 5%

1. A. Siddig, Data Mining Algorithms and Techniques Clustering, D., 2020. *Data Mining Algorithms And Techniques -Clustering*.
2. A. Siddig, Data Mining Algorithms and Techniques Clustering, D., 2020. *Data Mining Algorithms And Techniques -Association*.
3. A. Siddig, Data Mining Algorithms and Techniques Clustering, D., 2020. *Data Mining Algorithms And Techniques -Decision Tree* 2
4. A. Siddig, Datasets, Datasets,EDA and altering data structure D., 2020. *Data Mining Algorithms And Techniques* - Datasets,EDA and altering data structure
5. Archive.ics.uci.edu. n.d. *UCI Machine Learning Repository: Diabetes 130-US Hospitals For Years 1999-2008 Data Set*. [online] Available at: <http://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008> [Accessed 14 March 2020].
6. Brownlee, J., 2016. *Boosting And Adaboost For Machine Learning*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/> [Accessed 17 March 2020].
7. Brownlee, J., 2020. *SMOTE Oversampling For Imbalanced Classification With Python*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/> [Accessed 24 March 2020].
8. Cdc.gov. 2020. *Health Insurance Portability And Accountability Act Of 1996 (HIPAA) | CDC*. [online] Available at: <https://www.cdc.gov/phlp/publications/topic/hipaa.html> [Accessed 24 March 2020].
9. Chioka.in. 2013. *Class Imbalance Problem*. [online] Available at: <http://www.chioka.in/class-imbalance-problem/> [Accessed 22 March 2020].
10. Cs.ccsu.edu. n.d. *Using Weka 3 For Clustering*. [online] Available at: <http://www.cs.ccsu.edu/~markov/ccsu_courses/datamining-ex3.html> [Accessed 24 March 2020].
11. Cs.waikato.ac.nz. 2020. *Data Mining: Practical Machine Learning Tools And Techniques*. [online] Available at: <https://www.cs.waikato.ac.nz/~ml/weka/book.html> [Accessed 24 March 2020].
12. Docs.displayr.com. 2020. *Data Preparation For Cluster-Based Segmentation - Displayr*. [online] Available at: <https://docs.displayr.com/wiki/Data_Preparation_for_Cluster-Based_Segmentation> [Accessed 23 March 2020].
13. EduPristine. 2020. *K-Means Algorithm: Data Pre-Processing Before Running The K-Means Algorithm.*. [online] Available at: <https://www.edupristine.com/blog/k-means-algorithm> [Accessed 22 March 2020].
14. En.wikipedia.org. 2020. *Association Rule Learning*. [online] Available at: <https://en.wikipedia.org/wiki/Association_rule_learning> [Accessed 22 March 2020].
15. En.wikipedia.org. 2020. *Canopy Clustering Algorithm*. [online] Available at: <https://en.wikipedia.org/wiki/Canopy_clustering_algorithm> [Accessed 24 March 2020].
16. En.wikipedia.org. 2020. *DBSCAN*. [online] Available at: <https://en.wikipedia.org/wiki/DBSCAN> [Accessed 25 March 2020].
17. En.wikipedia.org. 2020. *Decision Tree Pruning*. [online] Available at: <https://en.wikipedia.org/wiki/Decision_tree_pruning> [Accessed 20 March 2020].
18. En.wikipedia.org. 2020. *Hospital Readmission*. [online] Available at: <https://en.wikipedia.org/wiki/Hospital_readmission> [Accessed 14 March 2020].

19. En.wikipedia.org. 2020. *Outlier*. [online] Available at: <https://en.wikipedia.org/wiki/Outlier> [Accessed 18 March 2020].
20. Hammoudeh, A., Al-Naymat, G., Ghannam, I. and Obied, N., 2018. Predicting Hospital Readmission among Diabetics using Deep Learning. *Procedia Computer Science*, 141, pp.484-489.
21. Harvard, H., 2020. *Diabetes - Harvard Health*. [online] Harvard Health. Available at: <https://www.health.harvard.edu/topics/diabetes> [Accessed 24 March 2020].
22. Kharroubi, A., 2015. Diabetes mellitus: The epidemic of the century. *World Journal of Diabetes*, 6(6).
23. Medium. 2017. *About Train, Validation And Test Sets In Machine Learning*. [online] Available at: <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7> [Accessed 16 March 2020].
24. Medium. 2019. *DBSCAN Python Example: The Optimal Value For Epsilon (EPS)*. [online] Available at: <https://towardsdatascience.com/machine-learning-clustering-dbscan-determine-the-optimal-value-for-epsilon-eps-python-example-3100091cfbc> [Accessed 29 March 2020].
25. National Institutes of Health (NIH). 2017. *Diabetes Increasing In Youths*. [online] Available at: <https://www.nih.gov/news-events/nih-research-matters/diabetes-increasing-youths> [Accessed 24 March 2020].
26. Rahmah, N. and Sitanggang, I., 2016. Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra. *IOP Conference Series: Earth and Environmental Science*, 31, p.012012.
27. SC CTSI. 2020. *Cerner Health Facts | SC CTSI*. [online] Available at: <https://sc-ctsi.org/resources/cerner-health-facts> [Accessed 15 March 2020].
28. ScienceDaily. 2020. *Diabetes Complications Are A Risk Factor For Repeat Hospitalizations, Study Shows*. [online] Available at: <https://www.sciencedaily.com/releases/2017/07/170707135143.htm> [Accessed 20 March 2020].
29. Strack, B., DeShazo, J., Gennings, C., Olmo, J., Ventura, S., Cios, K. and Clore, J., 2014. Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records. *BioMed Research International*, 2014, pp.1-11.
30. Weka.8497.n7.nabble.com. 2012. *WEKA - The Use Of Verbose In Association Rules*. [online] Available at: <https://weka.8497.n7.nabble.com/The-use-of-verbose-in-Association-Rules-td26573.html> [Accessed 24 March 2020].
31. Who.int. 2020. *Diabetes*. [online] Available at: <https://www.who.int/health-topics/diabetes> [Accessed 15 March 2020].