# National Institute of Technology, Calicut

## Department of Computer Science and Engineering


## Data Mining Project Report
## Winter Semester 2016

**Faculty Incharge:**
**Mr.Bharath Narayanan**

**Team 13:**
**1) Aakanksha N S - B130203CS**
**2) Ajith Saravanan - B130163CS**
**3) Akshaye A P - B130236CS**
**4) Reshmi Sriram - B130152CS**
**5) Sruthi Chandrasenan - B130754CS**

# Contents

# 1. INTRODUCTION

## 1.1 Project Overview

This Project involves applying various data cleaning techniques using OpenRefine and Weka for preprocessing. The preprocessed data is then mined using Weka.

## 1.2 Project Deliverables

**Data Pre processing**

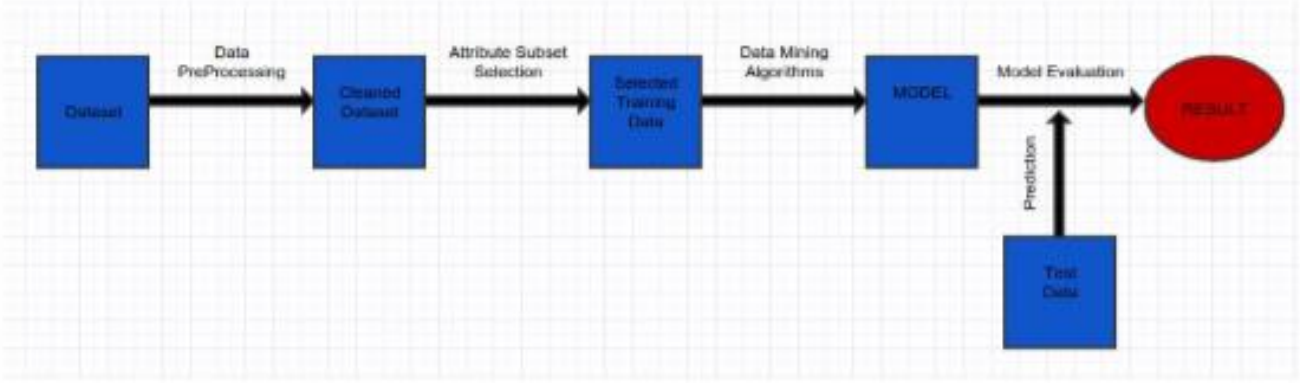
- Noisy Data
- Outliers
- Use a measure of central tendency for the attribute (e.g., the mean or median) to fill in the missing value. Use the most probable value to fill in the missing value

**Data Mining**

- Subset selection
- Classification:
  Naive Bayesian Classification Method
  Random forests
  Meta Naive Bayesian classifier ( using Chi-square attribute-evaluator and attribute selection using ranker)
- Clustering
  Simple K-means method (Manhattan and Euclidean distance).
- Association
  Apriori method

# 2 .PROCESS ORGANIZATION

## 2.1 Process Model




## 2.2 Role and Responsibility

### Preprocessing:

- Aakanksha did Data Selection. The data set was discovered from UCI Learning Repository.
- Ajith and Reshmi took up the Data Cleaning part.
- Akshaye and Sruthi were involved in Data Smoothening.

### Mining:

- Ajith did the attribute subset selection.
- Aakanksha did the Classification.
- Reshmi and Sruthi were involved in Clustering.
- Akshaye did the Association part
- Everyone was equally involved in creating the report.

## 2.3 Tools and Techniques

**TOOLS**:

### Preprocessing:

- Weka
- OpenRefine

### Data mining:

- Weka

**TECHNIQUES**:

■ Data cleaning: handles missing values by filling with mode for nominal attributes and median for numerical attributes.
■ Dimensionality reduction.
■ Attribute subset selection.

**Data Mining**:
■ Classification: - classifying data based on Naive Bayesian Model, random forests and compared the models using various factors like accuracy, precision, F-score etc.
■ Clustering - A simple K-means algorithm was ran on the data set to cluster the same on the basis of distance metrics between the data sets.

# 3. PROJECT MANAGEMENT

## 3.1 Data Set Description

The dataset represents 10 years (19992008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes. Information was extracted from the database for encounters that satisfied the following criteria.

(1) It is an inpatient encounter (a hospital admission).
(2) It is a diabetic encounter, that is, one during which any kind of diabetes was entered to the system as a diagnosis.
(3) The length of stay was at least 1 day and at most 14 days.
(4) Laboratory tests were performed during the encounter.
(5) Medications were administered during the encounter.
The data contains such attributes as patient number, race, gender, age, admission type, time in hospital, medical specialty of admitting physician, number of lab test performed, HbA1c test result, diagnosis, number of medication, diabetic medications, number of outpatient, inpatient, and emergency visits in the year before the hospitalization, etc. This data set had missing values only for six attributes whose information is presented below.

| **Feature name** | **Type** | **Description and values** | **Percentage Missing** |
|---|---|---|---|
| Race | Nominal | Values: Caucasian, Asian, African American, Hispanic, and other | 2% |
| Weight | Numeric | Weight in Pounds | 97% |

| | | | |
|---|---|---|---|
| Payer Code | Nominal | Integer identifier corresponding to 23 distinct values, for example, Blue Cross/Blue Shield, Medicare, and | 52% |
| | | selfpay | |
| Medical Speciality | Nominal | Integer identifier of a specialty of the admitting physician, corresponding to 84 distinct values, for example, cardiology, internal medicine, family/general practice, and surgeon | 53% |
| Diagnosis 3 | Nominal | Additional secondary diagnosis (coded as first three Digits of ICD9); 954 distinct values | 1% |

## 3.2 Tasks

## Preprocessing:

1. Handling Noisy data
2. Handling Missing data
3. Handling largely distributed valued attributes
4. Handling outliers
5. Data Discretization

## Mining:

1. Subset Selection
2. Classification
3. Clustering
4. Association

## 3.3 Description of Plan

## Preprocessing:

1. **Data Reduction**
   - The original database contains incomplete, redundant, and noisy information as expected in any real-world data. There were several features that could not be treated directly since they had a high percentage of missing values. These features were weight (97% values missing), payer code (40%), and medical specialty (47%). Weight attribute was considered to be too sparse and it was not included in further analysis. Payer code was removed since it had a high percentage of missing values and it was not considered relevant to the outcome.

2. **Handling Noisy data**
   - All the values were trimmed.

3. **Data Cleaning**
   - Missing values were replaced by mode for nominal attributes and median for numerical attributes. Medical specialty attribute was maintained, adding the value “missing” in order to account for missing values.

4. **Data discretization**
    - Since none of our attributes were continuous, there was no need for discretization to be applied on the data set.

5. **Data smoothening**
    - Various attributes were missing values from it's domain so tuples were added to include those values and the missing values were again filled up using the above methods.

## Mining:

1. **Subset selection**
2. **Classification**
    - Naive Bayesian Classification Method
    - Random forests
    - Meta Naive Bayesian classifier ( using Chi-square attribute-evaluator and attribute selection using ranker)
3. **Clustering**
    Simple K-means method (Manhattan and Euclidean distance).
4. **Association**
    Apriori method

## 3.5 Limitations of the Tool

- Doesn’t scale well with increase in size of data.
- Hard to take advantage of multiple processors available in the system
- Weka Server is hard to set up
- Not very stable; Does not handle huge memory loads well

# 4. SOFTWARE

## 4.1 Introduction

**Weka** is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data preprocessing, classification, regression, clustering, association rules, and visualization. It is also well suited for developing new machine learning schemes. Weka Server provides you with an interface to run Weka on one or more servers thereby speeding up computation. These "servers" can also be one or more processors of a single machine.

**OpenRefine**, formerly called *Google Refine*, is a standalone open source desktop application for data cleanup and transformation to other formats, the activity known as data wrangling. It is similar to spreadsheet applications (and can work with spreadsheet file formats); however, it behaves more like a database. It operates on rows of data which have cells under columns, which is very similar to relational database tables. One OpenRefine project is one table. The user can filter the rows to display using facets that define filtering criteria (for example, showing rows where a given column is not empty). Unlike spreadsheets, most operations in OpenRefine are done on all visible rows: transformation of all cells in all rows under one column, creation of a new column based on existing column data, etc. All actions that were done on a dataset are stored in a project and can be replayed on another dataset.

## 4.2 Reliability

**Weka** however is not as reliable as OpenRefine. If you lose your data during an execution, there is no way to recover the results that you had already obtained. Also,

Weka Server showed tendencies to crash when ran on large inputs (or when it ran out of memory).

## 4.3 Availability

**Weka** is an open source software that is published under the GNU GPL. This makes the software easy to obtain, use and modify if need be. Similar to OpenRefine, it is implemented using Java and hence can run on Windows, Linux and Mac.

## 4.4 Maintainability

**Weka:** Virtually no programming need take place and results are obtained by simple point and click (if you're using the GUI). In addition to the fact that the software is open source, this makes Weka very maintainable. You can also integrate algorithms in Weka in your Java code.

## 4.3.5 Portability

**Weka:** As mentioned previously, it is an open source software, and is distributed freely under the GNU General Public License. The software is fully portable and runs on any system that has a working implementation of the Java Runtime Environment.

## 4.3.6 Performance

**Weka:** Weka has more features than OpenRefine but it is very hard to scale it to use with Big Data. We were able to obtain many results using the same but encountered many crashes while doing so.

# 5. DESIGN SPECIFICATIONS

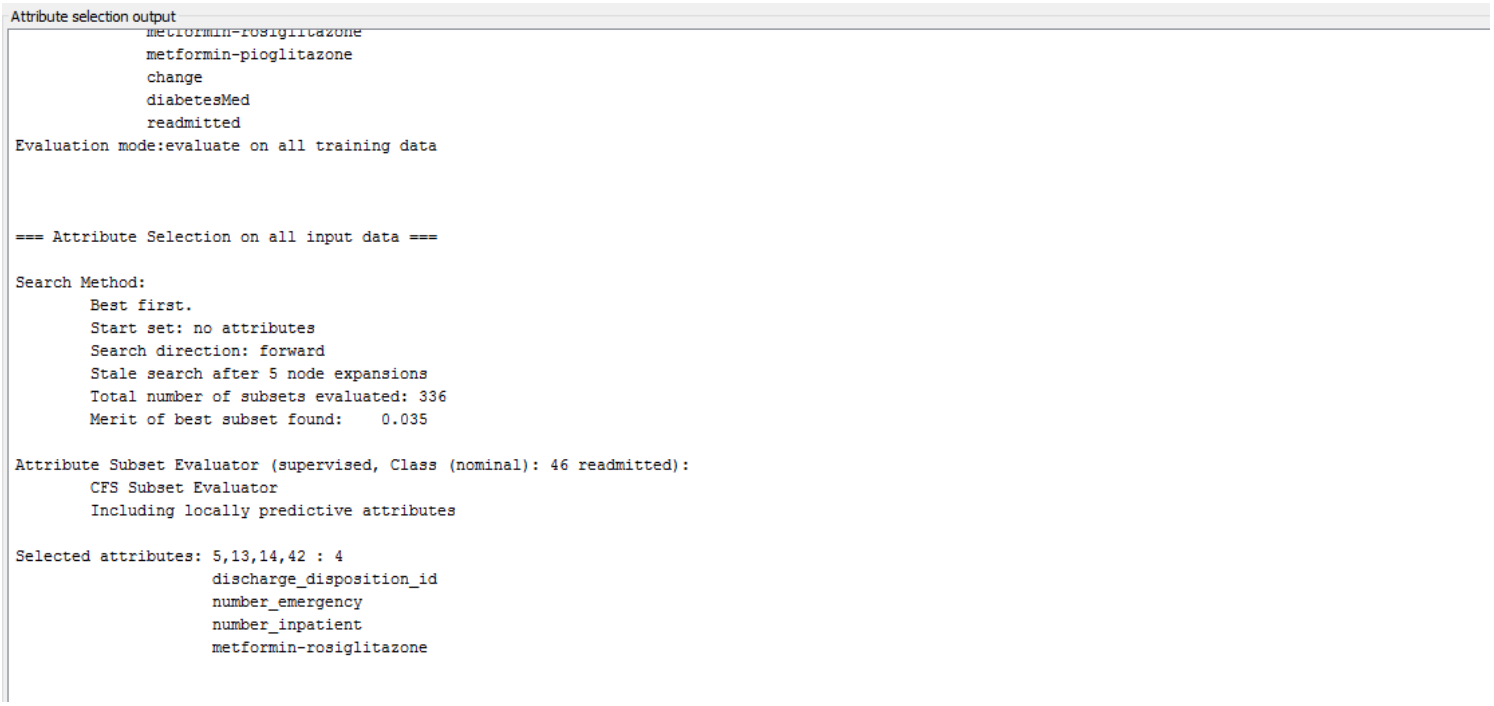
## 5.1 Design Overview

The data set was initially cleaned to make it suitable for drawing useful results. This was done using DataCleaner and Weka. This included prediction of missing values (using methods like Naive Bayes), dimensionality reduction etc. Once the clean data set was obtained, we had different versions of the data set which differed in their number of attributes and tuple values based on attribute subset selection and some steps of data cleaning (which we will discuss below). We then compare our results obtained when using these data sets to predict the readmission status of a diabetes patient.

## 5.2 Work Done

### 5.2.1 Attribute subset selection

Attribute selection was performed on the full data set using CFS attribute evaluator and best first search method.



## 5.2.2 Classification

1) Naive Bayesian Classifier

Classifier output

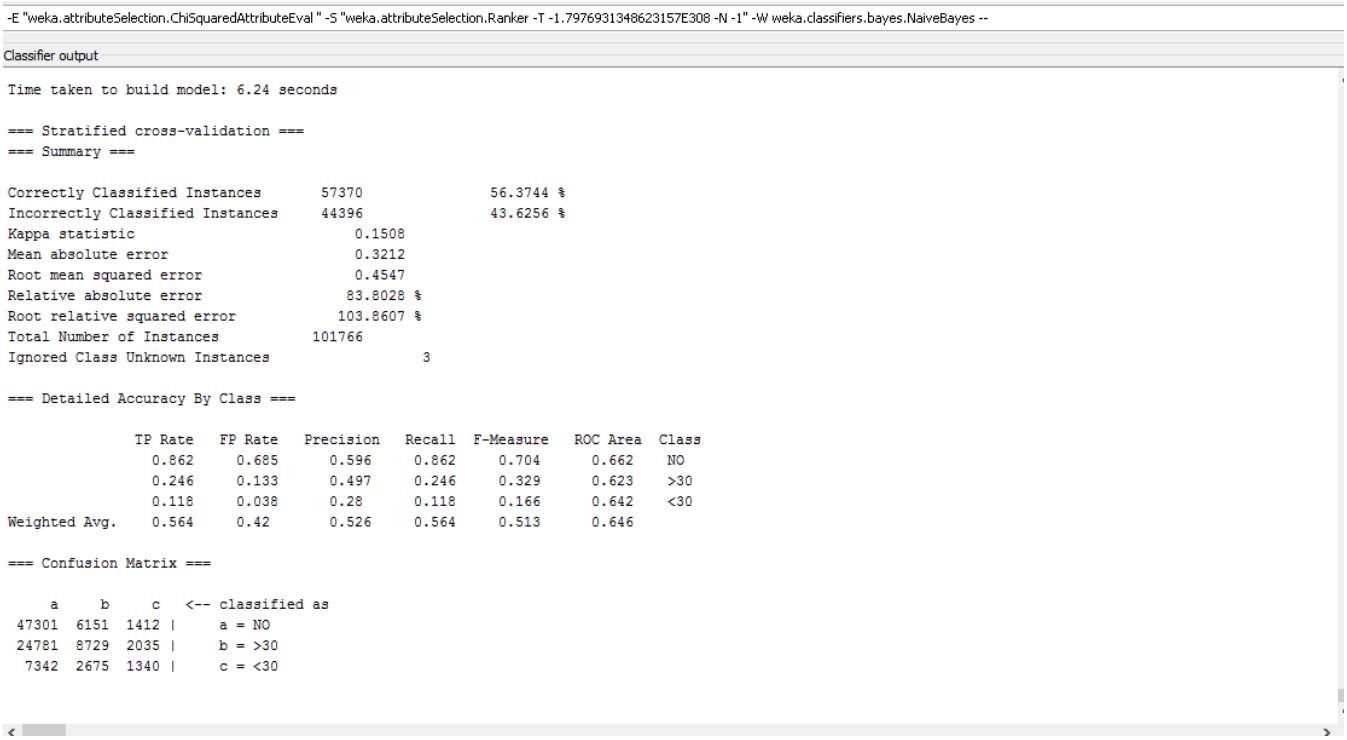
```
Time taken to build model: 0.99 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       57370               56.3744 %
Incorrectly Classified Instances     44396               43.6256 %
Kappa statistic                          0.1508
Mean absolute error                      0.3212
Root mean squared error                  0.4547
Relative absolute error                 83.8028 %
Root relative squared error            103.8607 %
Total Number of Instances           101766
Ignored Class Unknown Instances                3

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.862     0.685      0.596     0.862      0.704      0.662    NO
                 0.246     0.133      0.497     0.246      0.329      0.623    >30
                 0.118     0.038      0.28      0.118      0.166      0.642    <30
Weighted Avg.    0.564     0.42       0.526     0.564      0.513      0.646

=== Confusion Matrix ===

     a     b     c   <-- classified as
 47301  6151  1412 |     a = NO
 24781  8729  2035 |     b = >30
  7342  2675  1340 |     c = <30
```

2) Random forests

Classifier output

```
Time taken to build model: 14.87 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       58594               57.5772 %
Incorrectly Classified Instances     43172               42.4228 %
Kappa statistic                          0.1513
Mean absolute error                      0.3632
Root mean squared error                  0.4264
Relative absolute error                 94.7636 %
Root relative squared error             97.4008 %
Total Number of Instances           101766
Ignored Class Unknown Instances                3

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.884     0.703      0.595     0.884      0.712      0.632    NO
                 0.278     0.15       0.498     0.278      0.356      0.597    >30
                 0.019     0.003      0.445     0.019      0.036      0.611    <30
Weighted Avg.    0.576     0.432      0.545     0.576      0.512      0.617

=== Confusion Matrix ===

     a     b     c   <-- classified as
 48514  6269    81 |     a = NO
 25494  9868   183 |     b = >30
  7463  3682   212 |     c = <30
```

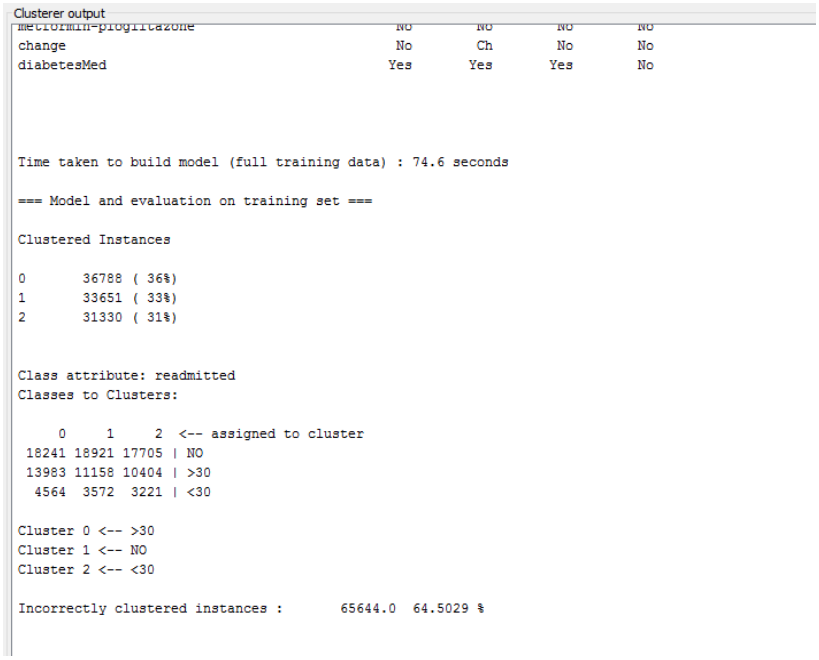
3) Meta Naive Bayesian classifier (using Chi-square attribute-evaluator and attribute selection using ranker)



```
-E "weka.attributeSelection.ChiSquaredAttributeEval " -S "weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1" -W weka.classifiers.bayes.NaiveBayes --

Classifier output

Time taken to build model: 6.24 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       57370               56.3744 %
Incorrectly Classified Instances     44396               43.6256 %
Kappa statistic                          0.1508
Mean absolute error                      0.3212
Root mean squared error                  0.4547
Relative absolute error                 83.8028 %
Root relative squared error            103.8607 %
Total Number of Instances           101766
Ignored Class Unknown Instances                3

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.862     0.685      0.596     0.862      0.704      0.662    NO
                 0.246     0.133      0.497     0.246      0.329      0.623    >30
                 0.118     0.038      0.28      0.118      0.166      0.642    <30
Weighted Avg.    0.564     0.42       0.526     0.564      0.513      0.646

=== Confusion Matrix ===

     a     b     c   <-- classified as
 47301  6151  1412 |     a = NO
 24781  8729  2035 |     b = >30
  7342  2675  1340 |     c = <30
```

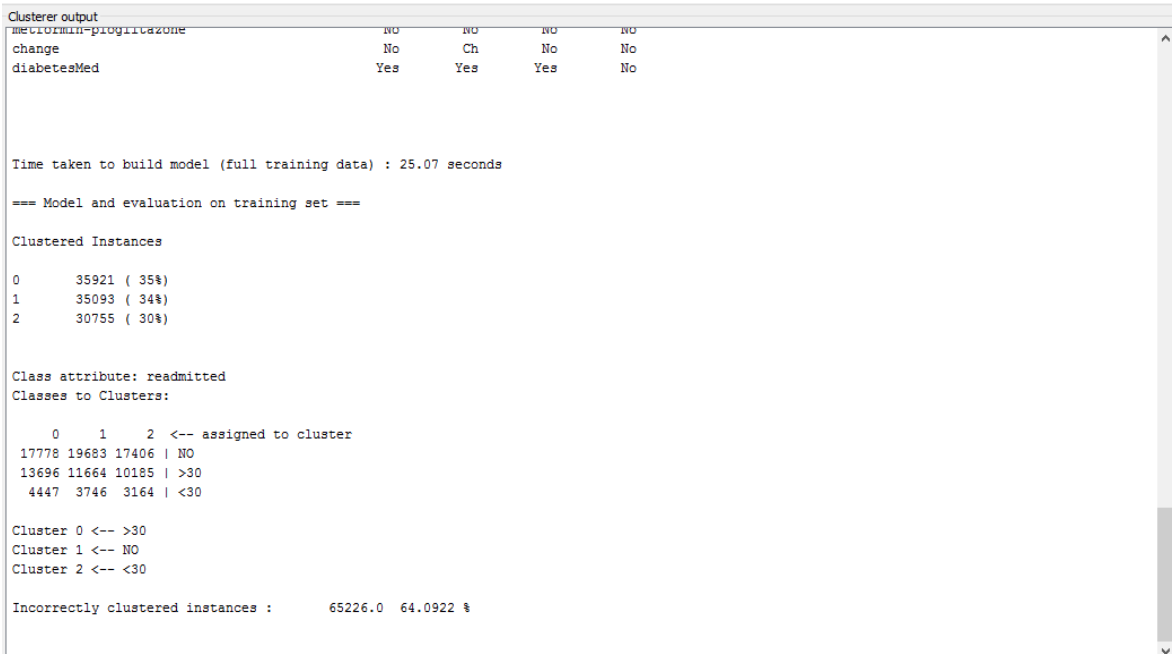
# 5.2.3 Clustering (simple k-means method)

1) Euclidean distance



```
Clusterer output

metformin-pioglitazone                   No        No        No        No
change                                   No        Ch        No        No
diabetesMed                             Yes       Yes       Yes        No



Time taken to build model (full training data) : 74.6 seconds

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

Clustered Instances

0       36788 ( 36%)
1       33651 ( 33%)
2       31330 ( 31%)


Class attribute: readmitted
Classes to Clusters:

     0     1     2  <-- assigned to cluster
 18241 18921 17705 | NO
 13983 11158 10404 | >30
  4564  3572  3221 | <30

Cluster 0 <-- >30
Cluster 1 <-- NO
Cluster 2 <-- <30

Incorrectly clustered instances :       65644.0  64.5029 %
```

2) Manhattan distance



```
Clusterer output
metformin-pioglitazone                    No       No       No       No
change                                    No       Ch       No       No
diabetesMed                              Yes      Yes      Yes       No



Time taken to build model (full training data) : 25.07 seconds

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

Clustered Instances

0       35921 ( 35%)
1       35093 ( 34%)
2       30755 ( 30%)


Class attribute: readmitted
Classes to Clusters:

     0     1     2  <-- assigned to cluster
 17778 19683 17406 | NO
 13696 11664 10185 | >30
  4447  3746  3164 | <30

Cluster 0 <-- >30
Cluster 1 <-- NO
Cluster 2 <-- <30

Incorrectly clustered instances :       65226.0  64.0922 %
```

## 5.2.4 Association (Apriori method)

We performed association using apriori method on the attributes selected by the Attribute subset selection.

```
Apriori
=======

Minimum support: 0.9 (91592 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 2

Generated sets of large itemsets:

Size of set of large itemsets L(1): 3

Size of set of large itemsets L(2): 3

Size of set of large itemsets L(3): 1

Best rules found:

 1. metformin-rosiglitazone=No 101765 ==> glimepiride-pioglitazone=No 101763    conf:(1)
 2. number_inpatient='(-inf-2.1]' metformin-rosiglitazone=No 94716 ==> glimepiride-pioglitazone=No 94714    conf:(1)
 3. glimepiride-pioglitazone=No 101766 ==> metformin-rosiglitazone=No 101763    conf:(1)
 4. number_inpatient='(-inf-2.1]' 94720 ==> glimepiride-pioglitazone=No 94717    conf:(1)
 5. number_inpatient='(-inf-2.1]' glimepiride-pioglitazone=No 94717 ==> metformin-rosiglitazone=No 94714    conf:(1)
 6. number_inpatient='(-inf-2.1]' 94720 ==> metformin-rosiglitazone=No 94716    conf:(1)
 7. number_inpatient='(-inf-2.1]' 94720 ==> glimepiride-pioglitazone=No metformin-rosiglitazone=No 94714    conf:(1)
 8. glimepiride-pioglitazone=No 101766 ==> number_inpatient='(-inf-2.1]' 94717    conf:(0.93)
 9. metformin-rosiglitazone=No 101765 ==> number_inpatient='(-inf-2.1]' 94716    conf:(0.93)
10. glimepiride-pioglitazone=No metformin-rosiglitazone=No 101763 ==> number_inpatient='(-inf-2.1]' 94714    conf:(0.93)
```

## 5.3 Inference

**5.3.1 Classification:** We chose the simple Naive Bayes Classifier from the classifiers available in Weka, which, on execution, yielded the confusion matrix. We see that Naive Bayes Classification has an accuracy of 56%. Meanwhile, Random forests method gave an accuracy of 58% and Meta Naive Bayesian classifier showed an accuracy of 56%. Hence, we conclude that the Random forests method was the best method to classify the given data.

**5.3.2 Clustering using K-means method**: We clustered the dataset into three distinct clusters. The ratio of incorrectly clustered Euclidean and Manhattan distances are 64.5% and 64.09% correspondingly. Hence, we derive that the Manhattan distance provides slightly better results than that of the Euclidean Distance.

**5.3.3 Association:**
The Apriori method was used for association mining. As this method need a lot of full data scans we were not able to obtain the results for all the data sets. Only the datasets obtained after attribute subset selection yielded results and thus the transaction set was obtained. The result as shown in the screenshot predicts the best association rules. As we can see, every rule has confidence 1 and the 10 rules are ordered according to their support.

**********