EE 559 Project

(05/02/2016)

PATTERN RECOGNITION SYSTEM FOR DIABETES DATASET

Name: Kusha Sridhar Huliyar Sridhara Murthy

USC ID: 3954054993

E-mail: huliyars@usc.edu

CONTENTS

ABSTRACT

This project analyses and does classification, learning and testing on the diabetes dataset [1]. The dataset was retrieved from the UCI Machine Learning repository. The objective of this project is to develop a pattern recognition system that operates on a given real-world dataset. Different tools and techniques of supervised learning have been applied to check the performance of different classifiers on the dataset. The performance of different classifiers like Naïve Bayes, KNN, Linear discriminant are evaluated using metrics discussed in the rest of the report that follows.

1. APPROACH

First, the given dataset is inspected for uninformative features (according to the information provided in [2] and discarded cleverly. Second, features are recasted and missing data in each feature column is filled using techniques discussed in the preprocessing section. Next comes the normalization techniques to suit different classifiers for training. And finally, the performance of the classifier is analyzed by using a metric called mean F1 score.

2. PREPROCESSING

The Test set is random 50% selection from the entire dataset. The method employed for randomizing the entire data set without bias on either training data or the test data is '2-fold cross validation'. There are 3 class labels (readmission rate) – '>30', '<30', 'No'. The string data type of the class labels is converted to double data type - 1, 2, 3 respectively. The features are first sorted according to their class labels and 2-fold cross validation is run on each class to randomly pick 50% of the samples of each class for the test set. Thus, the frequency of occurrence of samples within each class will be the same in the test set as in the training set.

i. Discarding uninformative features

- Weight: Since there are more than 97% missing values, this feature can be conveniently discarded from the feature space.
- Payer code: same reason as above but with 52% missing data and I don't think this has much effect on the readmission rate.
- Medical Specialty: 53% missing data and indexing needs to be done for 84 different value. It may be significant for the classification problem but let's see if it matters later.
- Drug types: Out of the 24 different types of drugs involved in the treatment of diabetes, most of them were found to be not prescribed in most of the medical cases except for insulin which was found to be a significant contributor in the treatment for diabetes over the rest and was coincidentally found to be prescribed in most of the cases and the dosage level was also fluctuating frequently.

ii. Handling missing data

- Race: Caucasian and African American were the dominant races among all and since missing data was around 1% of the total data, it was probabilistically filled into Caucasian and African American categories. Example: if 70% were Caucasian and 29% were African American – fill 70% of the missing data to Caucasian and the rest to African American. This statistical approach was employed to all the features with missing data. In the end the new race vector was left with 5 different races which were expanded into 5 binary columns (later this recasting helped in classification).
- Gender: Had very few unknown values, therefore categorized them into female category since the female gender has more occurrences in the given data set.
- Similar approach as above is used to fill the missing data for age as well. And then age vector was also expanded into 10 binary columns.

- Admission type id: Null, Not mapped and Not available are considered missing and filled into Emergency, Urgent, Newborn, trauma center and Elective categories as these 5 are found to be statistically dominant categories. Finally left with 8 levels which were expanded into 8 binary columns.
- Admission source id: Same operation as that done for admission type id is done on this feature except that it has 25 categories. Chose the top 5 most occurring categories and expanded them to binary after taking care of the missing data in the same way as explained above.
- Discharge disposition id: Following the above trend to fill in the missing values, they are included into transferred to SNF, discharged home or discharged with health service categories as these three are statistically significant.
- Diagnoses 1, 2, 3 : Based on the disease tabular indices (guidelines) given by the International Statistical Classification of Diseases and Health Related Problems, the current feature is put into a range between 1 and 19 corresponding to the ranges mentioned below. There are missing values in this as well, but here they are given the value 20 as it is difficult to categorize the missing values into any one of the below categories as they are found to be correlated to one another and have a very fuzzy boundary between them.

iii. Reacasting the representation of features and the final features used:

Post the step of discarding uninformative features, features are recasted into binary and integer. The real valued features are retained as it is. The missing values are taken care of by including them to the most probable triats that appear in a particular feature.

- Race: Type string → Binary (5 columns) representing 'Caucasian' (74.78%) and 'African American' (18.88%) and the other three since they were the dominant ones among the six races given, some missing values.
- Gender: Type string → Binary (1 column) representing 'Male' or 'Female' after taking care of the missing values.
- Age: Intervals → Integers → Binary, (10 columns).
- Admission type id: Integer → Binary, (expanded to 8 columns), keeping only statistically significant ones and also considering the kind of information they are providing.
- Discharge disposition id: Integer → Binary (expanded to 7 columns), keeping only statistically significant ones and also considering the kind of information they are providing.
- Admission source id: Integer → Binary (expanded to 5 features), keeping only statistically significant ones and also considering the kind of information they are providing.
- Time in hospital: Retain as it is → Integer (1 column).
- Number of lab procedures, Number of procedures and Number of medications: Retained as it is → Integer
- Number of inpatient, outpatient, emergency visits: Retained as an integer → 3 columns.
- Diagnosis1, 2 and 3: Integers between 1 and 20. (Expanded into 60 columns → Binary).[3]
- Number of Diagnoses: Retained as an Integer data type. It perfectly makes sense to retain as it is.
- Medications administered: Among the 24 features only Insulin is retained because it has the least amount of 'No' → drug not administered. And the rest 23 have like around 80% and above of no usage, hence removed. The four different levels are coded as 1, 2, 3, 4 representing 'Up', 'Steady', 'Down', 'No'. Therefore it is an integer data type. Then it is expanded to 4 columns → Binary.
- Change in medication: String → Binary representing 1 if changed and 0 if unchanged.
- Diabetes Medication: String → Binary representation either prescribed (1) or not (0).

Finally, the class labels are recasted as explained above for convenience.

iv. Normalization:

- Method 1: The whole reduced training data set of dimension 50883x27 is normalized (between -1 and 1) with their respective column means and standard deviations (using zscore MATLAB function). These means and standard deviations are then used to normalize the test data set aside.
- Method 2: Normalized only the features categorized as real and integer between 0 and 1 (may be helpful for statistical classification methods where the probability densities are non-negative) and let alone the binary features. (makes sense).

$$\boldsymbol{normalized\ sample} = \frac{\boldsymbol{Sample(i) - min}}{\boldsymbol{Max - Min}}$$

Where min and max are the minimum and maximum values of the respective column being normalized.

- Another way used was to normalize with respect to the class means and class standard deviations.

All these techniques failed to give F1 score above the baseline (0.5) for all the classifiers used. Some of the classifiers could not predict all the class labels when the normalized dataset was used. Hence, the idea of normalization was dropped and the raw reduced dataset without normalization was used for classification, training and further analysis.

v. Feature Selection

Looking at the readmission rate column, the features that were most explanatory were retained. This [2] source provided as part of the project helped in considering features that proved to be important in terms of their nature, statistical significance or simply made sense. Other factors used to select the features were using the distribution of the features among the three classes and the proportion of missing values they contained. Thus, the final set of features that were obtained is tabulated in the appendix. Note: The final 16 features are expanded to binary and categorical (selectively) to get 110 features over which the rest of the analysis is done.

To get clarity on the distribution of the classes among each feature column, WEKA machine learning software tool was used. (Appendix 3).

3. CLASSIFIERS USED

	Classifier Name	Function Name	MATLAB Toolbox name
	Statistical Classifiers		
1	Naive Bayes	fitcnb with multivariate multinomial distribution	
2	KNN	fitcknn with 16 nearest neighbors	
3	Linear discriminant	fitcdisc	Classification Learner
4	Bagged Trees		Classification Learner
5	Boosted Trees		Classification Learner
	Distribution Free Classifiers		
1	Multiclass classifier using Least Squares estimation. Criterion function is SSE	Least Squares (LS) Parameters: LS, OAA = One vs all	
2	Minimum distance to class means classifier	Code included in the MATLAB code	

4. Data Set usage

The whole data set is checked for repeated samples using patient number as a reference column and only one of them is retained and the rest deleted to avoid introducing overdispersion in the data. This technique can avoid inflation of standard errors and variances in the data. Different classification and learning algorithms mentioned above are used on the train dataset to get predicted labels, which are then compared with the test labels to find the classification accuracy, precision, recall and the mean F1 score. The algorithms used the train, train labels and the test dataset as their input arguments and give predicted labels as the output.

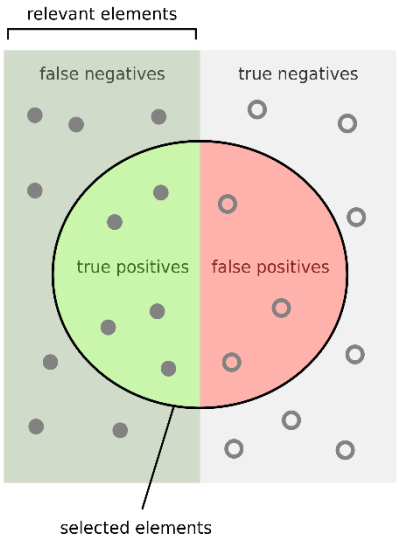
5. Performance Evaluation Techniques

Evaluations metrics like precision, recall, accuracy and mean F1 score are used to check the performance of the model (classifier). Precision (also called positive predictive value) is the fraction of the returned samples that are relevant. Recall (also called sensitivity) is the fraction of the relevant samples that are returned. Bothe precision and recall are based on an understanding and measure of relevance. Concisely, precision is “how useful the classification results” and recall is “how complete the results are”. In statistical terms (hypothesis testing), maximum precision means no false positives (type I errors) and maximum recall means no false negatives (type II errors).

True Positive = TP, True Negative = TN, False Positive = FP, False Negative = FN

- Precision in the confusion matrix (shown for different classifiers in the “results” section) is the number in the column corresponding to a particular row (which gives the class) divided by the sum of the numbers in that column.

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}}$$

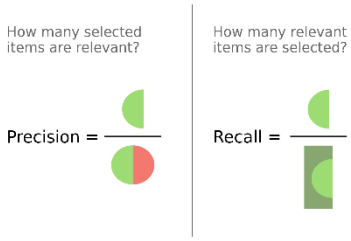


- Recall in the confusion matrix is the number in column to a particular row divided by the sum of the numbers in that row.

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}}$$

- Mean F1 score is in the range [0,1], the larger the better. It will have a large value when both the precision and recall are relatively large, but much smaller when one of the two is very large and the other is very small.

$$\boldsymbol{F1} = 2\frac{\boldsymbol{Precision} * \boldsymbol{Recall}}{\boldsymbol{Precision} + \boldsymbol{Recall}}$$



$$\boldsymbol{Mean\ F1\ score} = \frac{\sum_{i=1}^{3} \boldsymbol{N_i}.\boldsymbol{F1_i}}{\sum_{i=1}^{3} \boldsymbol{N_i}}$$

- Accuracy is not so robust, hence not always used for evaluating the performance of the classifiers when other strong metrics like the ones mentioned above are present.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP} + \boldsymbol{TN}}{\boldsymbol{TP} + \boldsymbol{TN} + \boldsymbol{FP} + \boldsymbol{FN}}$$

6. RESULTS

Statistical Classifiers

- **Naive Bayes Classifier using Multivariate Multinomial distribution**

==========

Precision	Recall	F1-score	Support
0.52	0.30	0.38	12069
0.29	0.01	0.02	2323
0.65	0.87	0.74	21367
0.58	0.62	0.58	35759

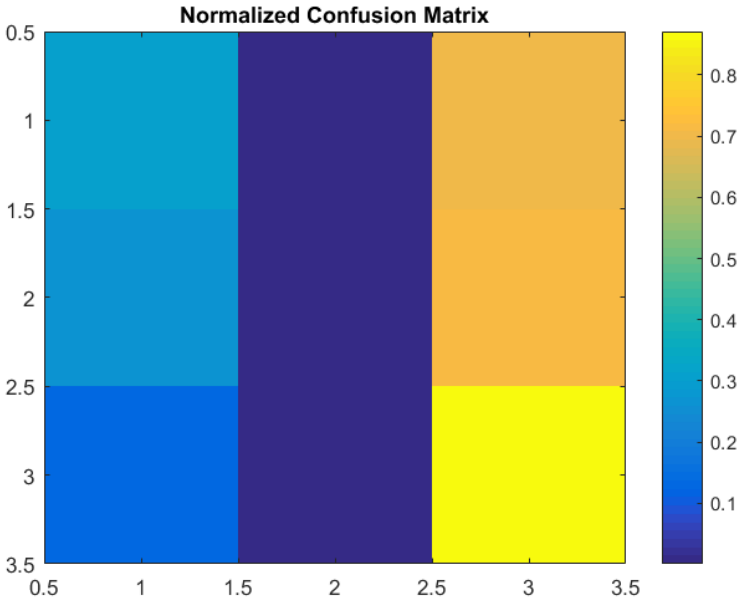
==========

0.575167

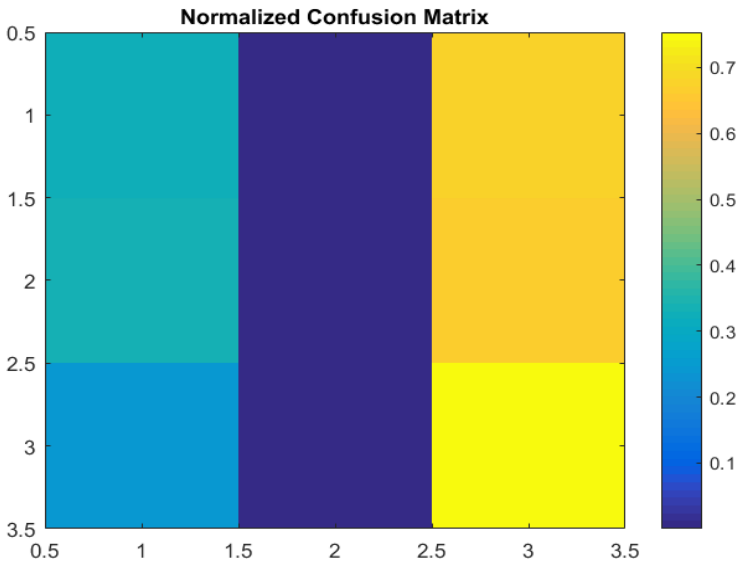
m_f1 = 0.575166744131649

conf =

3648	42	8379
619	29	1675
2731	30	18606



- **K-Nearest Neighbors (KNN)**



==========

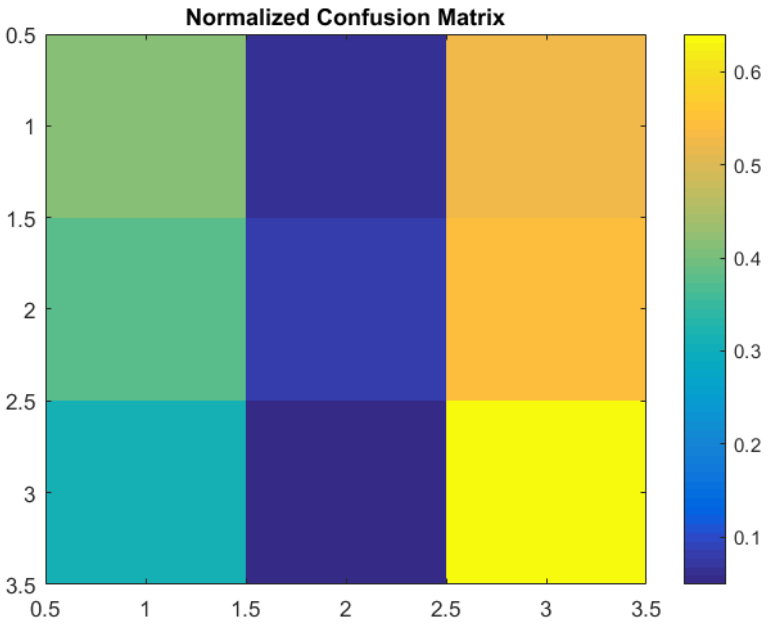
Precision	Recall	F1-score	Support
0.40	0.33	0.36	12069
0.06	0.00	0.00	2323
0.62	0.75	0.68	21367
0.51	0.56	0.53	35759

==========

0.529109

m_f1 = 0.529108872560129

- **Decision Tree classifier**



==========

Precision	Recall	F1-score	Support
0.40	0.41	0.41	12069
0.09	0.08	0.08	2323
0.64	0.64	0.64	21367
0.52	0.53	0.53	35759

==========

0.526043

m_f1 = 0.526043169029382

- **Bagged Trees (Ensemble classifier)**

==========

Precision	Recall	F1-score	Support
0.51	0.31	0.39	12069
0.52	0.00	0.01	2323
0.65	0.86	0.74	21367
0.60	0.62	0.57	35759

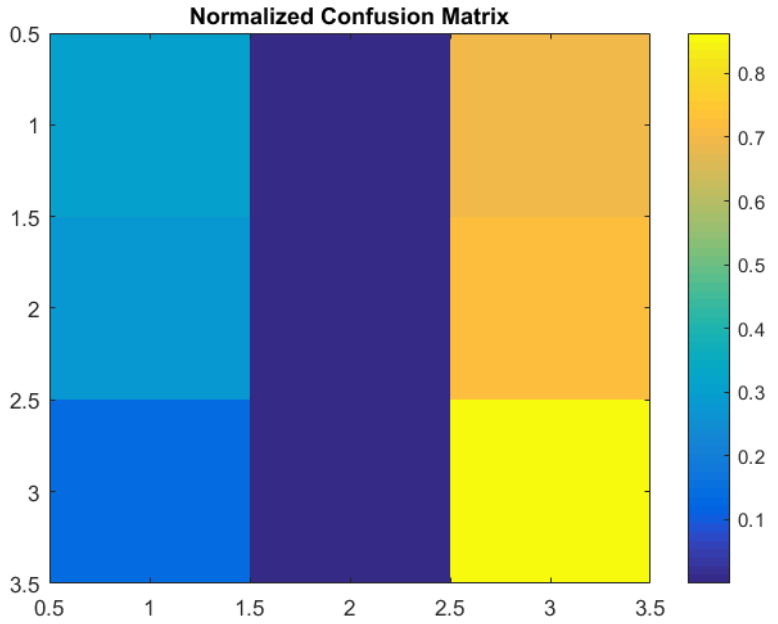
==========

0.574121

m_f1 = 0.574121112695055

conf =

3766	5	8298
622	11	1690
2932	5	18430



Distribution Free Classifiers

- **Minimum distance to class means classifier using Maximal Value Method**

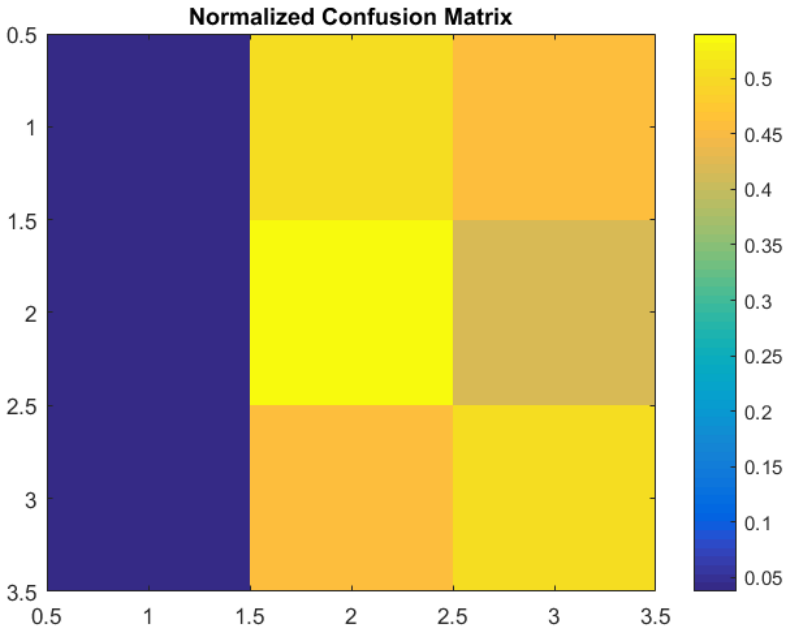
==========

Precision	Recall	F1-score	Support
0.35	0.04	0.07	12069
0.07	0.54	0.13	2323
0.62	0.51	0.56	21367
0.50	0.35	0.37	35759

==========

0.367031

m_f1 = 0.367030516445074



- **Linear Discriminant**

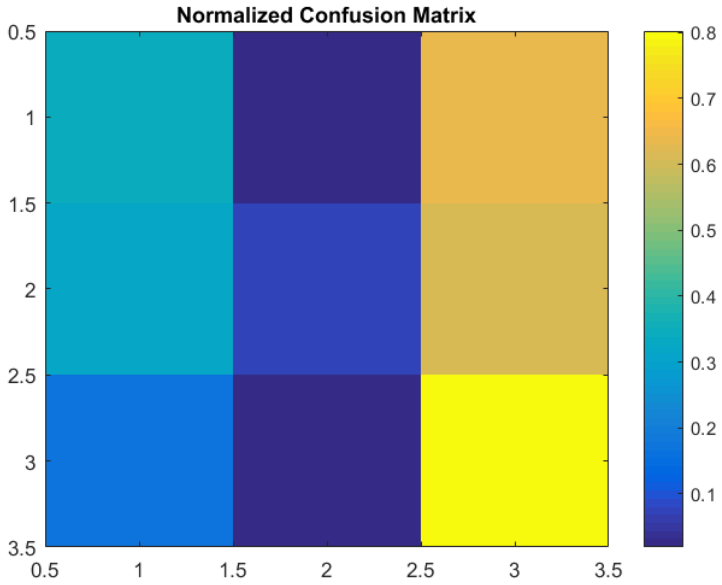
==========

Precision	Recall	F1-score	Support
0.47	0.34	0.40	12069
0.20	0.07	0.11	2323
0.65	0.80	0.72	21367
0.56	0.60	0.57	35759

==========

0.570740

m_f1 = 0.570739746374235



conf =

4097	272	7700
737	174	1412
3811	417	17139

INTERPRETATION OF RESULTS

- **Naive Bayes Classifier:** This classifier gave the highest mean F1 score of 0.58. The precision for classes 1 and 3 are look good but the recall for class 2 affects the F1 score. Since this classifier, according to the Bayes theorem, assumes a strong independence between features, binary expansion of the features may be helping this method to classify with a high mean F1 score. Also from Bayes theorem, the probability of evidence and the prior probabilities are constant through any feature, so the posterior probabilities are easy to compute.

  Function: model = fitcnb (train, label, 'DistributionNmes', 'mvmn'); → using multivariate multinomial distribution.

- **KNN:** The mean precision and recall were 0.51 and 0.56 respectively. But the recall for class 2 is really low which can introduce a skew in the data and hence can cause overfitting problems but the recall for the other two classes seem to balance the other one.

  The mean F1 score from the KNN classifier is 0.53.

Function: model = fitcknn (train, label, 'NumNeighbors', '15'); → with 15 nearest neighbors.

- **Decision Tree:** A decision tree is a non-parametric supervised learning method that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm. Decision tree has a flow-chart like structure in which each internal node represents a test on an attribute (like an event occurring), each branch represents the outcome of the test and each leaf node represents a class label. The path from the root to leaf represents classification rules.[4]

  Some advantages of decision trees:

  - Requires little data preparation. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed.
  - Able to handle both numerical and categorical data. Other techniques are usually specialized in analyzing datasets that have only one type of variable.

  Similar to the above classifiers the recall for class 2 (that is the readmission rate <30) is low (0.08) indicating that very few of the relevant samples were returned. This indicates that there were insufficient samples for training leading to drop in the value of recall.

  Function: model = fitctree (train, label, 'prune', 'on'): → this functions finds the optimal nodes/levels to prune the tree and classifies.

- **Bagged Tree:** This is one of the ensemble methods to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability/robustness over a single estimator.

  The algorithm for bagging goes something like this:[4]

  - Training phase:
    1) Initialize the parameters
       - D = the ensemble
       - L, the number of classifiers to train
    2) For k = 1,……..,L
       - Take a bootstrap sample S_k from Z.
       - Build a classifier D_k using S_k as the training set.
       - Add the classifier to the current ensemble, $D = D \cup D_k$.
    3) Return D.
  - Classification phase

4) Run D_1,……,D_L on the input x.
5) The class with the maximum number of votes is chosen as the label for x.

As observer from the confusion matrix, again the recall for class 2 is zero indicating that none of the relevant samples were returned. This further has affected the F1 score of class 2 to go low (complies with the formula for F1 score). However, there is an indication that greater than 50% of the returned samples are relevant in every class which favours the classifier to give good precision.

Function: Using the inbuilt model in the classification toolbox.

- **Minimum distance to class means classifier using Maximal Value Method:** This classifier performs poorly on class 1, same goes for class 2 as well but the recall for class 2 is high indicating a good fit. However, the performance metrics for class 3 proves a good fit. But the overall F1 score reduces due to inconsistency in all the three performance metrics in any particular class.

- **Linear Discriminant:** The precision, recall and F1 score for class 1 and class 3 are high but for class 2 they are 0.2, 0.07 and 0.11 respectively. The analysis for this classifier can be thought in similar lines to the previously mentioned classifiers.

It is observed that the precision, recall and F1 scores for class 2 are consistently performing poor on all classifiers. This may be because of the high difference in proportion of the split of the data from each of the three classes. This high imbalance in classes can be solved by stratified cross-validation which takes into account the proportion of samples from each class. However, this method also performed poorly on my preprocessed dataset.

IMPORTANT FEATURES AND THEIR INFLUENCE

1) Some of the features like number of lab procedures and number of procedures were found to have a great influence on the mean F1 score of all the classifiers used. When these feature were included in the model after initial preprocessing, they increased the F1 score by a small but significant amount above the baseline.
2) Some feature like diagnisis1, 2 and 3 when expanded, blew up the dimensionality but increased the F1 score as well. This is due to the fact that there is no need of external scaling for the expanded binary features, they are naturally scaled.
3) The binary representation and recasting of features reduced the risk of multicollinearity in the model as the features was found to be almost independent of each other and this also helped in using bayes classifier which works well with the normality assumption.

4) Some features age, number of times a patient has been inpatient and number of times a patient has been admitted as an inpatient increased the precision of readmission >30 days. This indicates that there was a high correlation between the readmission rate the features mentioned earlier.

5) Some features like patients transferred to home had a positive effect on the readmission rate.

REFERENCES

[1]. http://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008

[2]. http://www.hindawi.com/journals/bmri/2014/781670/tab1/

[3]. http://www.icd9.chrisendres.com/index.php?action=contents

[4]. Python scikit learn

[5]. Richard O. Duda., *Pattern Classification with Computer Manual*. Wiley - 2006

APPENDIX

APPENDIX 1

MATLAB CODE

```
%%
clc;
clear all;
load('project_559');


%%
%%---------------------------------------
% RACE
u = unique(race);
for i = 1:length(u)
    race2(strcmp(u(i),race)) = i;
end

for j = 1:length(race2)
    if race2(j) == 1
        race2(j) = mode(race2);
    end
end

race2 = race2';
race2_1 = zeros(length(race2),1);
race2_2 = zeros(length(race2),1);
race2_3 = zeros(length(race2),1);
race2_4 = zeros(length(race2),1);
race2_5 = zeros(length(race2),1);

race2_1(race2(:)==2)=1;race2_2(race2(:)==3)=1;
race2_3(race2(:)==4)=1;race2_4(race2(:)==5)=1;
race2_5(race2(:)==6)=1;

new_race = [race2_1 race2_2 race2_3 race2_4 race2_5];


%%
% GENDER
u = unique(gender);
for i = 1:length(u)
    gender2(strcmp(u(i),gender)) = i;
```

```
end

length(find(gender2==1))/length(find(gender2==2));

for j = 1:length(gender2)
    if gender2(j) == 3;
        gender2(j) = 1;
    end
end

gender2 = gender2';

gender2_1 = zeros(length(gender2),1);

gender2_1(gender2(:)==1)=1;

new_gender = gender2_1;

%%
% AGE
u = unique(age);
for i = 1:length(u)
    age2(strcmp(u(i),age)) = i;
end

age2 = age2';

ten=zeros(length(age2),1);
twenty=zeros(length(age2),1);
thirty=zeros(length(age2),1);
forty=zeros(length(age2),1);
fifty=zeros(length(age2),1);
sixty=zeros(length(age2),1);
seventy=zeros(length(age2),1);
eighty=zeros(length(age2),1);
ninety=zeros(length(age2),1);
hundred=zeros(length(age2),1);

ten(age2==1)=1;twenty(age2==2)=1;thirty(age2==3)=1;forty(age2==4)=
1;fifty(age2==5)=1;sixty(age2==6)=1;
seventy(age2==7)=1;eighty(age2==8)=1;ninety(age2==9)=1;hundred(age
2==10)=1;
```

```
new_age = [ten twenty thirty forty fifty sixty seventy eighty
ninety hundred];


%%
% WEIGHT - removed

%%
% ADMISSION TYPE ID
id1 = admission_type_id;
a = find(id1 == 5 | id1 == 6 | id1 == 8);
id_array1 = randperm(length(find(id1 == 5 | id1 == 6 | id1 ==
8)),10396);
one = 0.6*length(id_array1);
id1(a(id_array1(1:6238)))=1;
id1(a(id_array1(6239:8317)))=2;
id1(a(id_array1(8317:10396)))=3;

id_1 = zeros(length(id1),1);
id_2 = zeros(length(id1),1);
id_3 = zeros(length(id1),1);
id_4 = zeros(length(id1),1);
id_5 = zeros(length(id1),1);
id_6 = zeros(length(id1),1);
id_7 = zeros(length(id1),1);
id_8 = zeros(length(id1),1);

id_1(id1(:)==1)=1;id_2(id1(:)==2)=1;id_3(id1(:)==3)=1;id_4(id1(:)=
=4)=1;id_5(id1(:)==5)=1;
id_6(id1(:)==6)=1;id_7(id1(:)==7)=1;id_8(id1(:)==8)=1;

new_admission_type_id = [id_1 id_2 id_3 id_4 id_5 id_6 id_7 id_8];


%%
% DISCHARGE DISPOSITION ID
id2 = discharge_disposition_id;
missing_values2 = find(discharge_disposition_id==18 |
discharge_disposition_id==25);
id_array2 =
randperm(length(missing_values2),length(missing_values2));
two = 0.6*length(id_array2);
id2(missing_values2(id_array2(1:2770)))=1;
id2(missing_values2(id_array2(2771:3411)))=3;
```

```
id2(missing_values2(id_array2(3412:end)))=6;

transferred_to_snf = zeros(length(id2),1);
back_home = zeros(length(id2),1);
health_service = zeros(length(id2),1);
short_term = zeros(length(id2),1);
inpatient = zeros(length(id2),1);
expired = zeros(length(id2),1);
rehab = zeros(length(id2),1);

transferred_to_snf(id2 == 1) = 1;
short_term(id2 == 2) = 1;
back_home(id2 == 3) = 1;
inpatient(id2 == 5) = 1;
health_service(id2 == 6) = 1;
expired(id2 == 11) = 1;
rehab(id2 == 22) = 1;

new_discharge_disposition_id  =[transferred_to_snf short_term
back_home inpatient health_service expired rehab];


%%
% ADMISSION SOURCE ID
for i = 1:25
    occurrence3(i) = length(find(admission_source_id==i));
end

id3 = admission_source_id;
missing_values3 = find(admission_source_id==9 |
admission_source_id==17 | admission_source_id==20);
id_array3 =
randperm(length(missing_values3),length(missing_values3));
three1 = 0.56*length(id_array3);
three2 = 0.29*length(id_array3);
id3(missing_values3(id_array3(1:3957)))=7;
id3(missing_values3(id_array3(3958:6007)))=1;
id3(missing_values3(id_array3(6008:end)))=4;

source_id1 = zeros(length(admission_source_id),1);
source_id2 = zeros(length(admission_source_id),1);
source_id3 = zeros(length(admission_source_id),1);
source_id4 = zeros(length(admission_source_id),1);
source_id5 = zeros(length(admission_source_id),1);
```

```
source_id1(admission_source_id(:)==7)=1;
source_id2(admission_source_id(:)==1)=1;
source_id3(admission_source_id(:)==4)=1;
source_id4(admission_source_id(:)==2)=1;
source_id5(admission_source_id(:)==6)=1;

new_admission_source_id = [source_id1 source_id2 source_id3
source_id4 source_id5];


%%
% TIME IN HOSPITAL - retain as it is

%%
% PAYER CODE - removed

%%
% MEDICAL SPECIALITY - removed

%%
% NUMBER OF LAB PROCEDURES, NUMBER OF PROCEDURES AND NUMBER OF
MEDICATIONS - retained as it is


%%
% NUMBER OF INPATIENT, OUTPATIENT AND EMERGENCY VISITS - retained
as it is

%%
% DIAGNOSIS - includes all three
ICD9_1 = zeros(19,2);
ICD9_1(1,:) = [0 139];ICD9_1(2,:) = [140 239];ICD9_1(3,:) = [240
279];
ICD9_1(4,:) = [280 289];ICD9_1(5,:) = [290 319];ICD9_1(6,:) = [320
389];
ICD9_1(7,:) = [390 459];ICD9_1(8,:) = [460 519];ICD9_1(9,:) = [520
579];
ICD9_1(10,:) = [580 629];ICD9_1(11,:) = [630 679];ICD9_1(12,:) =
[680 709];
ICD9_1(13,:) = [710 739];ICD9_1(14,:) = [740 759];ICD9_1(15,:) =
[760 779];
ICD9_1(16,:) = [780 799];ICD9_1(17,:) = [800 999];
```

```
a = cellfun(@(y) strcmp(y(1),'?'),diabeticdata);
%
b = cellfun(@(y) strcmp(y(1),'V'),diabeticdata);
%
c = cellfun(@(y) strcmp(y(1),'E'),diabeticdata);

diabeticdata(a==1) = cellstr('-20');
diabeticdata(b==1) = cellstr('-18');
diabeticdata(c==1) = cellstr('-19');
d = cellfun(@(x) str2double(x),diabeticdata);

for i = 1:17
    d(d>=ICD9_1(i,1) & d<=ICD9_1(i,2))=i;
    e(i) = length(find(d==i));
end
(max(e)/101766)*100
d(d==-20)=20;
d(d==-18)=18;
d(d==-19)=19;



new_d1 = zeros(101766,20);
new_d2 = zeros(101766,20);
new_d3 = zeros(101766,20);
for k = 1:20
    new_d1(d(:,1)==k,k)=1;
    new_d2(d(:,2)==k,k)=1;
    new_d3(d(:,3)==k,k)=1;
end

new_d = [new_d1 new_d2 new_d3];

%%
% NUMBER OF DIAGNOSIS - retain as it is

%%
% Insulin : 'Up'     = 1
%           'Steady' = 2
%           'Down'   = 3
%           'No'     = 4 or 0
u = unique(insulin);
for i = 1:length(u)
```

```
    new_insulin(strcmp(u(i),insulin)) = i;
end

for j = 1:length(new_insulin)
    if new_insulin(j) == 1
        new_insulin(j) = 3;
    elseif new_insulin(j) == 2
        new_insulin(j) = 0;
    elseif new_insulin(j) == 3
        new_insulin(j) = 2;
    else
        new_insulin(j) = 1;
    end
end

new_insulin = new_insulin';

insulin1 = zeros(length(new_insulin),1);
insulin2 = zeros(length(new_insulin),1);
insulin3 = zeros(length(new_insulin),1);
insulin4 = zeros(length(new_insulin),1);

insulin1(new_insulin(:) == 1) = 1;
insulin2(new_insulin(:) == 2) = 1;
insulin3(new_insulin(:) == 3) = 1;
insulin4(new_insulin(:) == 4) = 1;

insulin_new = [insulin1 insulin2 insulin3 insulin4];


%%
% CHANGE
new_change = zeros(length(change),1);

new_change(strcmp(change(:),'Ch')) = 1;
new_change(strcmp(change(:),'No')) = 0;


%%
% DIABETEDMED

new_diabetesMed = zeros(length(diabetesMed),1);
new_diabetesMed(strcmp(diabetesMed(:),'Yes')) = 1;
new_diabetesMed(strcmp(diabetesMed(:),'No')) = 0;
```

```
%%

new_final_project_data1 = [new_race new_gender new_age
new_admission_type_id new_discharge_disposition_id
new_admission_source_id];
new_final_project_data2 = [time_in_hospital num_lab_procedures
num_procedures];
new_final_project_data3 = [num_medications number_emergency
number_inpatient number_outpatient new_d number_diagnoses
insulin_new new_change new_diabetesMed];
new_final_project_data = [new_final_project_data1
new_final_project_data2 new_final_project_data3];

%%
% READMITTED
new_readmitted = readmitted;
new_readmitted1 = cellfun(@(z) strcmp(z(1),'N'),readmitted);
new_readmitted2 = cellfun(@(z) strcmp(z(1),'>'),readmitted);
new_readmitted3 = cellfun(@(z) strcmp(z(1),'<'),readmitted);

new_readmitted(new_readmitted1==1) = cellstr('3');
new_readmitted(new_readmitted2==1) = cellstr('1');
new_readmitted(new_readmitted3==1) = cellstr('2');

label = cellfun(@(w) str2double(w),new_readmitted);



%%
% NORMALIZATION - finally train has 110 features/dimensions
(degrees of freedom)
%                          test has 110 features

% Normalization between -1 and 1
% for i = 1:110
%     train_mean(i) = mean(train(:,i));
%     train_std(i) = std(train(:,i));
% end
%
% norm_train = zeros(size(train));
% for j = 1:110
%     norm_train(:,j) = (train(:,j) -
train_mean(j))./train_std(j);
```

```
% end
%
% norm_test = zeros(size(test));
% for j = 1:110
%     norm_test(:,j) = (test(:,j) - train_mean(j))./train_std(j);
% end


% Normalization between 0 and 1
for i = 1:110
    minimum(i) = min(train(:,i));
    maximum(i) = max(train(:,i));
end

norm_train1 = zeros(size(train));
for j = 1:110
    norm_train1(:,j) = (train(:,j)-minimum(j))/(maximum(j)-
minimum(j));
end


norm_test1 = zeros(size(test));
for k = 1:110
    norm_test1(:,k) = (test(:,k)-minimum(k))/(maximum(k)-
minimum(k));
end




%%
finally = [patient_nbr new_final_project_data label];
finally_sort = sortrows(finally,112);
[C,IA,IC] = unique(finally_sort(:,1),'rows','first');
reduced_dataset = finally_sort(IA,:);
reduced_dataset(:,1) = [];

%%
    reduced_sort = sortrows(reduced_dataset,111);

class1(1:length(find(reduced_sort(:,111)==1))) =
crossvalind('Kfold',length(find(reduced_sort(:,111)==1)),2);
```

```
class2(1:length(find(reduced_sort(:,111)==2))) =
crossvalind('Kfold',length(find(reduced_sort(:,111)==2)),2);
class3(1:length(find(reduced_sort(:,111)==3))) =
crossvalind('Kfold',length(find(reduced_sort(:,111)==3)),2);

class3(class3==2)=3;
class = [class1';class2';class3'];
dummy_model = [reduced_sort class];


for i = 1:71518
    if dummy_model(i,111) == dummy_model(i,112)
        final_model(i,:) = dummy_model(i,:);
    end
end

final_model( ~any(final_model,2), : ) = [];  %rows
train = final_model;
train(:,112) = [];
label1 = train(:,111);
train(:,111) = [];

for j = 1:71518
    if dummy_model(j,111) ~= dummy_model(j,112)
        test_model(j,:) = dummy_model(j,:);
    end
end

test_model( ~any(test_model,2), : ) = [];  %rows
test = test_model;
test(:,112) = [];
label2 = test(:,111);
test(:,111) = [];

%%
norm_train = zscore(train);
norm_test = zscore(test);

%%
% Maximal value method - minimum distance to class means
classifier

means1 = mean(train(1:12070,:));
means2 = mean(train(12071:14392,:));
```

```
means3 = mean(train(14393:end,:));
means = [means1;means2;means3];

for i = 1:3
    DIST(:,i) = pdist2(test,means(i,:));
end

new_label = zeros(length(DIST),1);

for j = 1:length(DIST)
    if DIST(j,1) < DIST(j,2) && DIST(j,1) < DIST(j,3)
        new_label(j) = 1;
    elseif DIST(j,2) < DIST(j,1) && DIST(j,2) < DIST(j,3)
        new_label(j) = 2;
    elseif DIST(j,3) < DIST(j,1) && DIST(j,3) < DIST(j,2)
        new_label(j) = 3;
    end
end

accuracy = 0;
for k = 1:length(new_label)
    if new_label(k) == label2(k)
        accuracy = accuracy+1;
    end
end

display((accuracy/length(new_label))*100);
```

APPENDIX 2

ICD-9 Values

1. INFECTIOUS AND PARASITIC DISEASES (001-139)

2. NEOPLASMS (140-239)

3. ENDOCRINE, NUTRITIONAL AND METABOLIC DISEASES, AND IMMUNITY DISORDERS (240-279)

4. DISEASES OF THE BLOOD AND BLOOD-FORMING ORGANS (280-289)

5. MENTAL DISORDERS (290-319)

6. DISEASES OF THE NERVOUS SYSTEM AND SENSE ORGANS (320-389)

7. DISEASES OF THE CIRCULATORY SYSTEM (390-459)

8. DISEASES OF THE RESPIRATORY SYSTEM (460-519)

9. DISEASES OF THE DIGESTIVE SYSTEM (520-579)

10. DISEASES OF THE GENITOURINARY SYSTEM (580-629)

11. COMPLICATIONS OF PREGNANCY, CHILDBIRTH, AND THE PUERPERIUM (630-679)

12. DISEASES OF THE SKIN AND SUBCUTANEOUS TISSUE (680-709)

13. DISEASES OF THE MUSCULOSKELETAL SYSTEM AND CONNECTIVE TISSUE (710-739)

14. CONGENITAL ANOMALIES (740-759)

15. CERTAIN CONDITIONS ORIGINATING IN THE PERINATAL PERIOD (760-779)

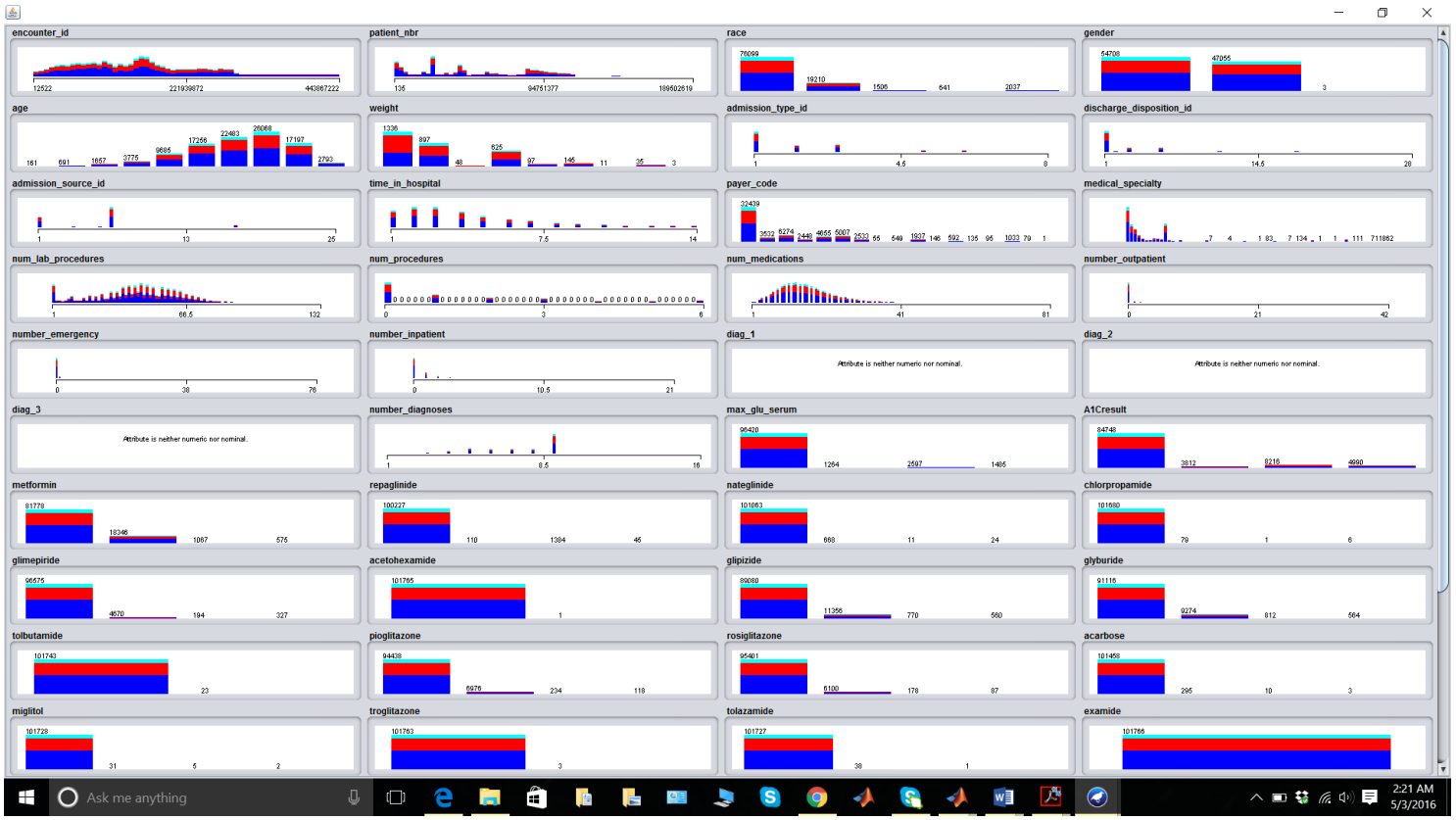
16. SYMPTOMS, SIGNS, AND ILL-DEFINED CONDITIONS (780-799)

17. INJURY AND POISONING (800-999)

SUPPLEMENTARY CLASSIFICATION OF FACTORS INFLUENCING HEALTH STATUS AND CONTACT WITH HEALTH SERVICES (V01-V89)

APPENDIX 3

WEKA EXPLORER OUTPUTS FOR FEATURE VISUALIZATION AND FEATURE DISTRIBUTION.



This gives a visual representation of how the classes are divided in each feature. Very useful in feature selection and extraction of the best explanatory features.

APPENDIX 4

Features	Type	Expanded into = # of columns	Descriptions
Race	Binary	5	N/A
Gender	Binary	1	N/A
Age	Binary	10	N/A
Admission type id	Binary	8	N/A
Discharge disposition	Binary	7	N/A
Admission source id	Binary	5	N/A
Time in hospital	Real	1	Min:1,Max:14,Mean:4.39,STD:2.98
Number of procedures	Real	1	Min:0,Max:6,Mean:1.339,STD:1.7
Number of lab procedures	Real	1	Min:1,Max:132,Mean:43.09,STD:19.67
Number of medications	Real	1	Min:0,Max:80,Mean:16.02,STD:12.76
Number of inpatient	Real	1	Min:0,Max:80,Mean:0.635,STD:1.26
Number of outpatient	Real	1	Min:1,Max:14,Mean:0.369,STD:1.26
Number of emergencies	Real	1	Min:1,Max:14,Mean:0.1978,STD:0.9304
Diagnosis 1	Binary	20	N/A
Diagnosis 2	Binary	20	N/A
Diagnosis 3	Binary	20	N/A
Number of Diagnoses	Real	1	
Insulin	Binary	4	N/A
Change	Binary	1	N/A
Diabetes medication	Binary	1	N/A
Total 20 features used		Total = 110	