Prediction of hospital Readmission: Analysis of 70,000 Clinical Database Patient Records

Abstract:

One of the most critical problems in healthcare is predicting the likelihood of hospital readmission in case of chronic diseases such as diabetes to be able to allocate necessary resources such as beds, rooms, specialists, and medical staff, for an acceptable quality of service. Unfortunately, relatively few research studies in the literature attempted to tackle this problem; most of the research studies are concerned with predicting the likelihood of the diseases themselves. Numerous machine learning techniques are suitable for prediction. Nevertheless, there is also shortage in adequate comparative studies that specify the most suitable techniques for the prediction process. Towards this goal, this paper presents a comparative study among five common techniques in the literature for predicting the likelihood of hospital readmission. Those techniques are Naïve Bayesian (NB) classifier, decision tree, and support vector machine (SVM), Ensembles (Random forest, Extra trees), K-Nearest neighbor. The comparative study is based on realistic data gathered from several hospitals in the United States. The comparative study revealed that Extra tree showed best performance, while the NB classifier were the worst.

Introduction

Nowadays, numerous chronic diseases, such as diabetes, are widespread in the world; and the number of patients is increasing continuously. The estimated number of diabetic adults in 2014 was 422 million versus 108 million in 1980. Such patients visit hospitals frequently, requiring continuous preparation for ensuring the availability of required resources including hospital beds, rooms, and enough medical staff for an acceptable quality of service. Accordingly, predicting the likelihood of readmission of a given patient is of ultimate importance. In fact readmission during a one month period (30 days) of discharge indicates "a high-priority healthcare quality measure" and the goal is to address this problem .

Databases of clinical data contain valuable but heterogeneous and difficult data in terms of missing values, incomplete or inconsistent records, and high dimensionality understood not only by number of features but also their complexity. [8]. Additionally, analyzing external data is more challenging than analysis of results of a carefully designed experiment or trial, because one has no impact on how and what type of information was collected. Nonetheless, it is important to utilize these huge amounts of data to find new information/knowledge that is possibly not available anywhere.
Various machine learning algorithms are applied to predict the readmission.

Data exploration

1 . Data assembly

The Health Facts data we used was an extract representing 10 years (1999–2008) of clinical care at 130 hospitals and integrated delivery networks throughout the United States. The database consists of 41 tables in a fact-dimension schema and a total of 117 features. The database includes 74,036,643 unique encounters (visits) that correspond to 17,880,231 unique patients and 2,889,571 providers. Because this data represents integrated delivery network health systems in addition to stand-alone hospitals, the data contains both inpatient and outpatient data, including emergency department, for the same group of patients. The dataset was created in two steps. First, encounters of interest were extracted from the database with 50 attributes. Second, preliminary analysis and preprocessing of the data were performed resulting in retaining only these features (attributes) and encounters that could be used in further analysis, that is, contain sufficient information. Both steps are described in the following subsections.

2. Extraction of the Initial Dataset from the Database

Information was extracted from the database for encounters that satisfied the following criteria.

- It is an inpatient encounter (a hospital admission).
- It is a "diabetic" encounter, that is, one during which any kind of diabetes was entered to the system as a diagnosis.
- The length of stay was at least 1 day and at most 14 days.
- Laboratory tests were performed during the encounter.
- Medications were administered during the encounter.

Criteria 3-4 were applied to remove admissions for procedures and so forth, which were of less than 23 hours of duration and in which changes in diabetes management were less likely to have occurred. It should be noted that the diabetic encounters are not all encounters of diabetic patients but rather only these encounters where diabetes was coded as an existing health condition.

101,766 encounters were identified to fulfill all the above five inclusion criteria and were used in further analysis. Attribute/feature selection was performed by our clinical experts and only attributes that were potentially associated with the diabetic condition or management were retained. From the information available in the database, we extracted 50 features describing the diabetic encounters, including demographics, diagnoses, diabetic medications, number of visits in the year preceding the encounter, and payer information.

Below is the full list of features and their description:

Encounter ID : Unique identifier of an encounter

Patient number : Unique identifier of a patient

Race : Values: Caucasian, Asian, African American, Hispanic, and other

Gender : Values: male, female, and unknown/invalid

Age : Grouped in 10-year intervals in the range of 0 to 100

Weight : Weight in pounds

Admission type : Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available

Discharge disposition : Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available

Admission source : Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital

Time in hospital : Integer number of days between admission and discharge

Payer code : Integer identifier corresponding to 23 distinct values, for example, Blue Cross/Blue Shield, Medicare, and self-pay Medical

Medical specialty : Integer identifier of a specialty of the admitting physician, corresponding to 84 distinct values, for example, cardiology, internal medicine, family/general practice, and surgeon

Number of lab procedures : Number of lab tests performed during the encounter

Number of procedures : Numeric Number of procedures (other than lab tests) performed during the encounter

Number of medications: Number of distinct generic names administered during the encounter

Number of outpatient visits : Number of outpatient visits of the patient in the year preceding the encounter

Number of emergency visits : Number of emergency visits of the patient in the year preceding the encounter

Number of inpatient visits : Number of inpatient visits of the patient in the year preceding the encounter

Diagnosis 1 : The primary diagnosis (coded as first three digits of ICD9); 848 distinct values

Diagnosis 2 : Secondary diagnosis (coded as first three digits of ICD9); 923 distinct values

Diagnosis 3 : Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values

Number of diagnoses : Number of diagnoses entered to the system 0%

Glucose serum test result : 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 : 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 : Indicates if there was a change in diabetic medications (either dosage or generic name). Values: “change” and “no change”

Diabetes medications : Indicates if there was any diabetic medication prescribed. Values: “yes” and “no”

24 features for medications 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

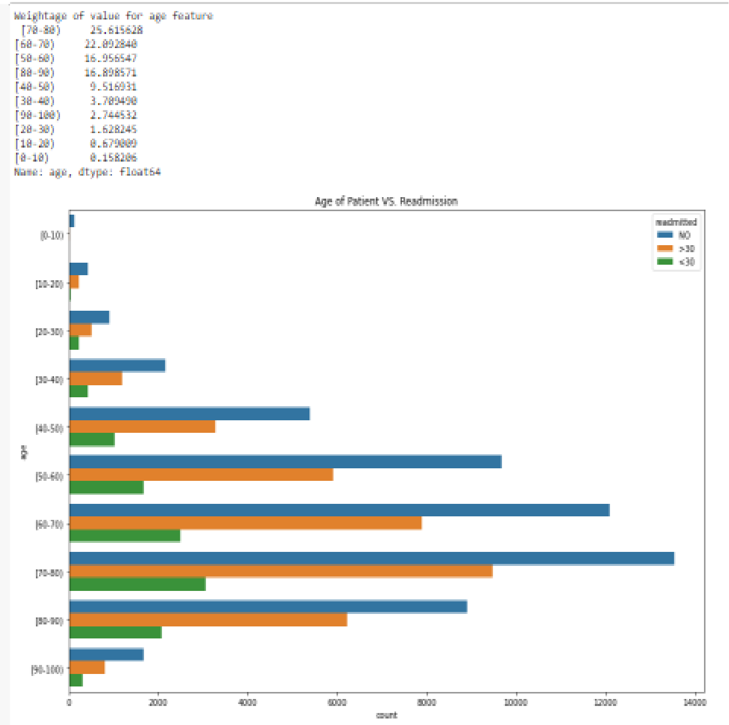
Readmitted : Days to inpatient readmission. Values: “<30” if the patient was readmitted in less than 30 days, “>30” if the patient was readmitted in more than 30 days, and “No” for no record of readmission

Since we are primarily interested in factors that lead to readmission either in less than 30 days or more than 30 days or not readmitted. We defined ‘readmitted’ has three values i.e. ‘No’, ’<30’, ’>30’.

2.3. Preliminary Analysis and the Final Dataset.

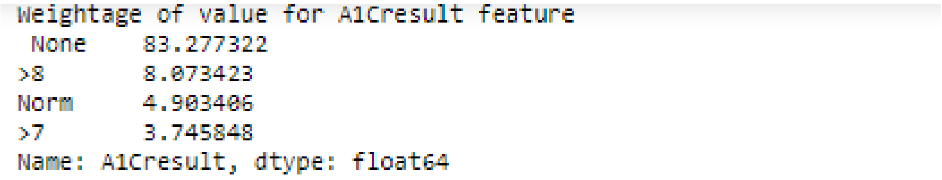
The original database contains incomplete, redundant, and noisy information as expected in any real-world data.

2.3.1 Data Analysis:

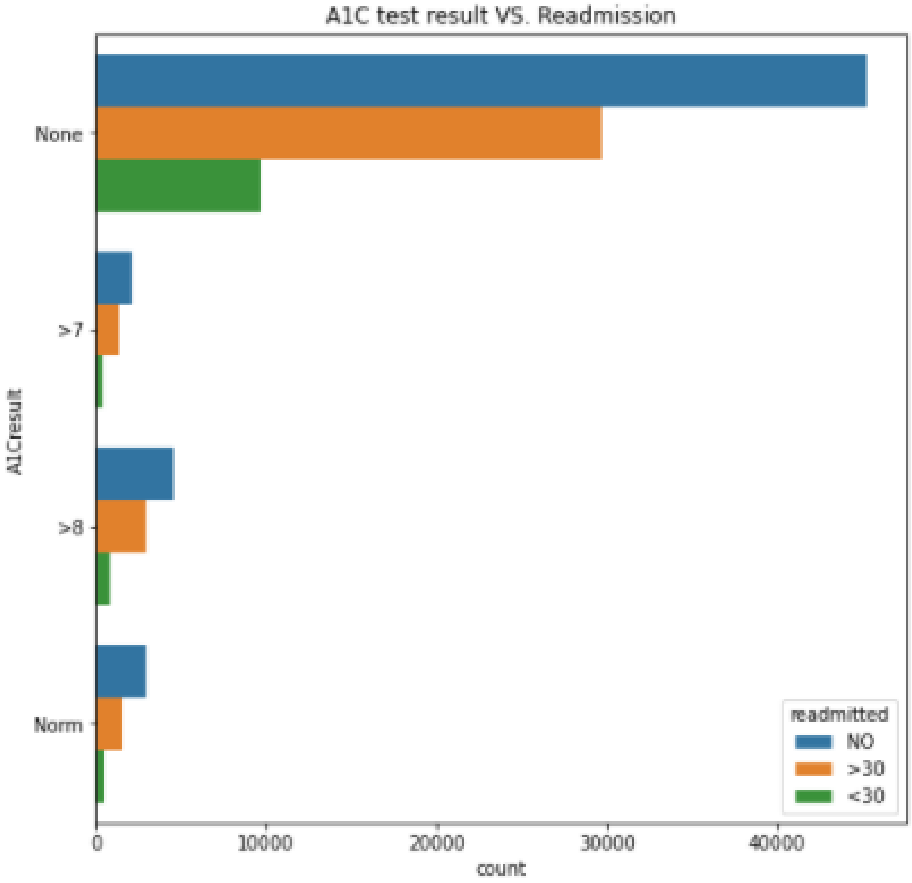


```
Weightage of value for age feature
 [70-80)     25.615628
[60-70)     22.092840
[50-60)     16.956547
[80-90)     16.898571
[40-50)      9.516931
[30-40)      3.709490
[90-100)     2.744532
[20-30)      1.628245
[10-20)      0.679009
[0-10)       0.158206
Name: age, dtype: float64
```

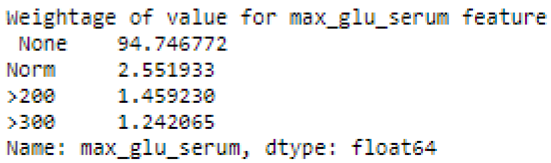
Age in interval of [70-80) readmitted more compared to any age group. Age in interval of [0-10), [10-20),[20-30) are combinedly less than 5% of total number of records.



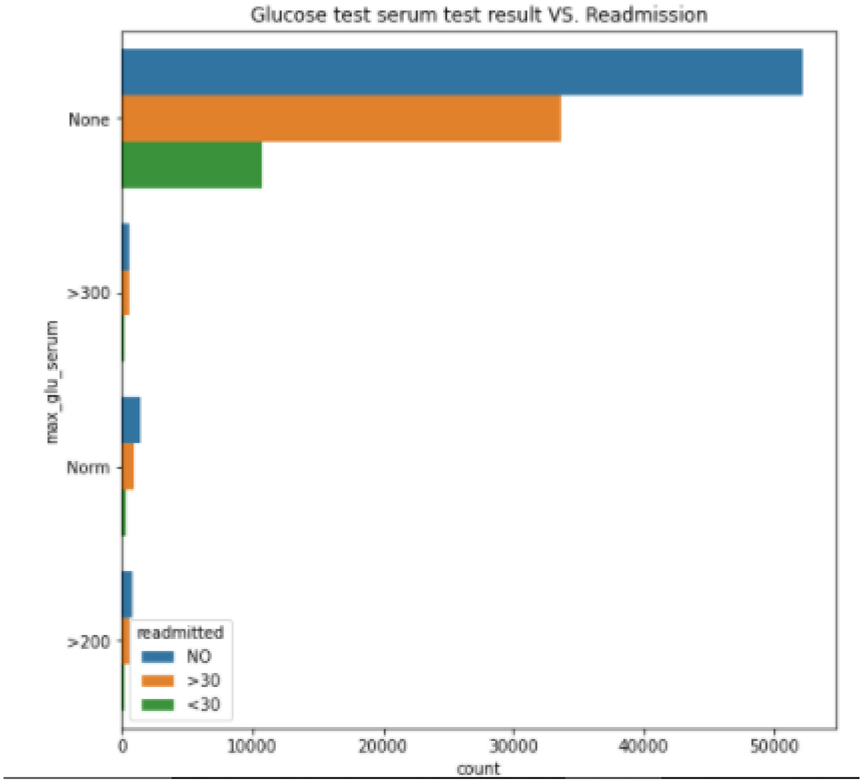
```
Weightage of value for A1Cresult feature
 None     83.277322
>8        8.073423
Norm      4.903406
>7        3.745848
Name: A1Cresult, dtype: float64
```



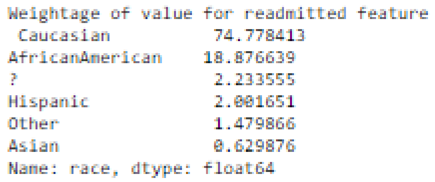
Patient who didn’t took the A1C test were more readmitted.



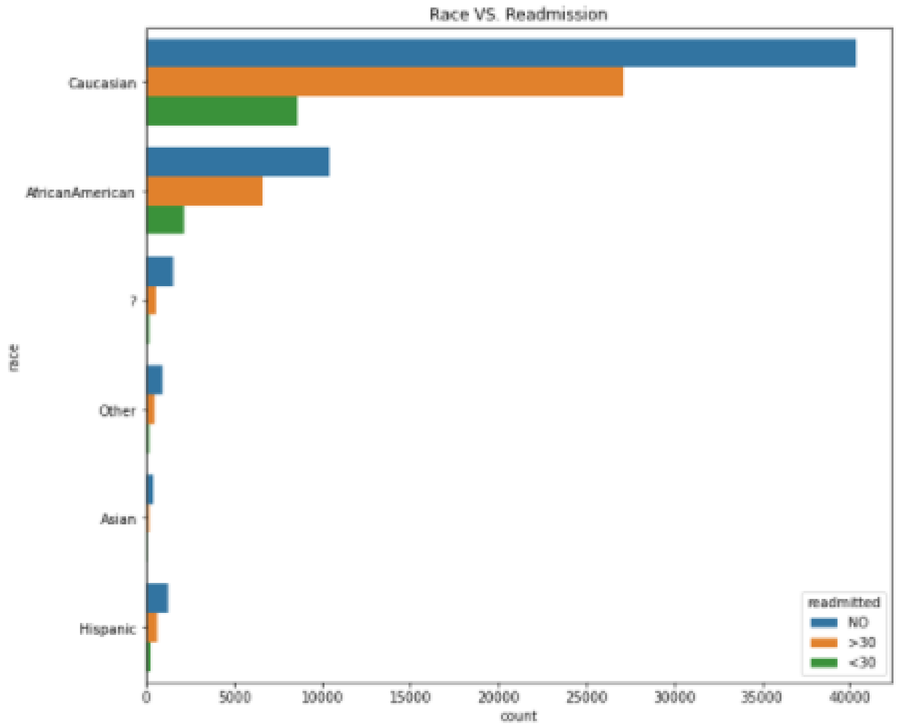
```
Weightage of value for max_glu_serum feature
 None     94.746772
Norm      2.551933
>200      1.459230
>300      1.242065
Name: max_glu_serum, dtype: float64
```



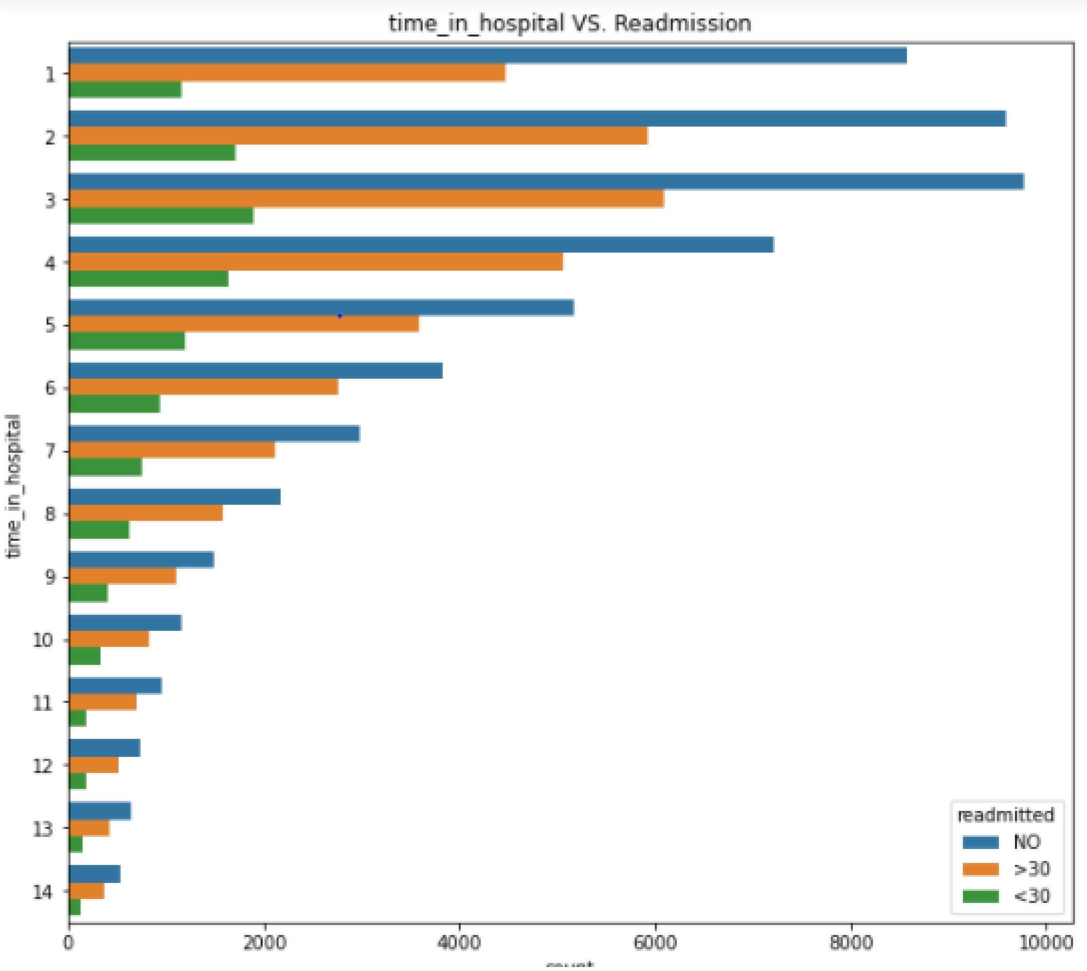
Patient who didn’t take glucose serum test are more admitted.



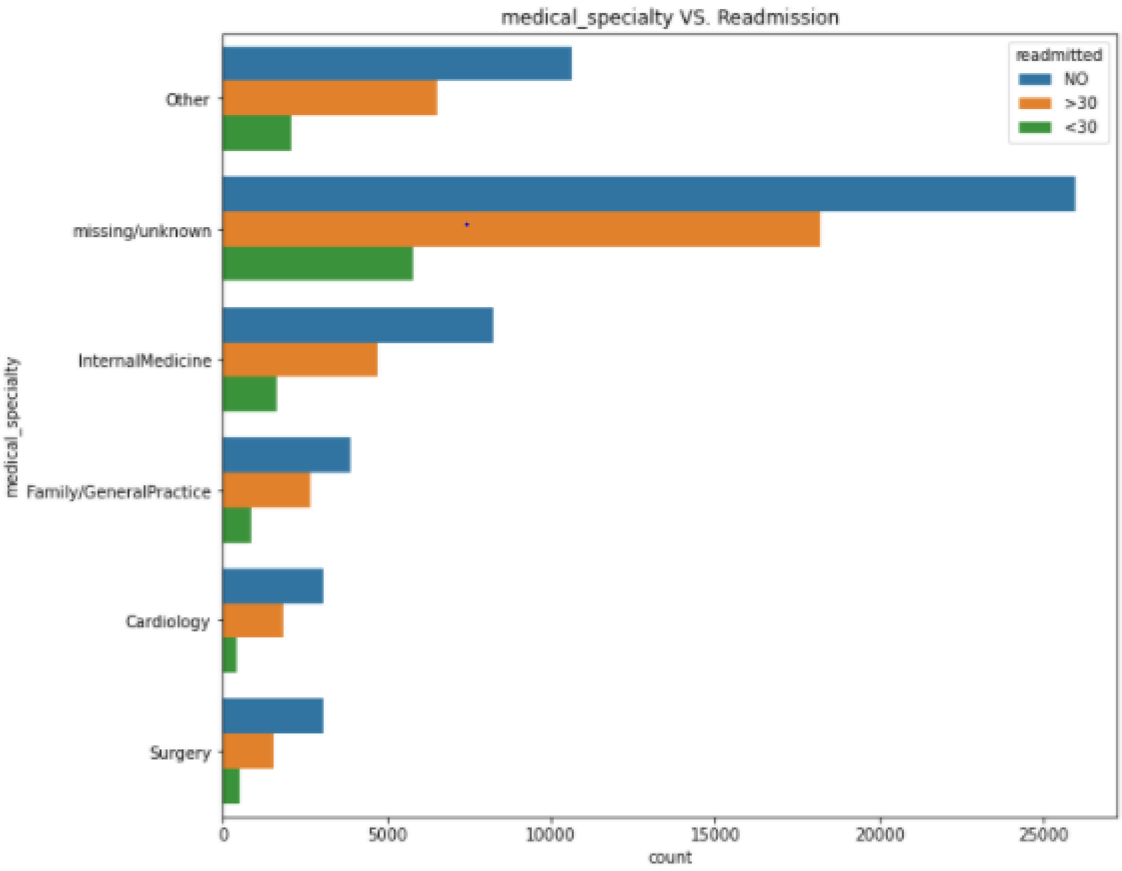
```
Weightage of value for readmitted feature
 Caucasian          74.778413
AfricanAmerican    18.876639
?                   2.233555
Hispanic            2.001651
Other               1.479866
Asian               0.629876
Name: race, dtype: float64
```



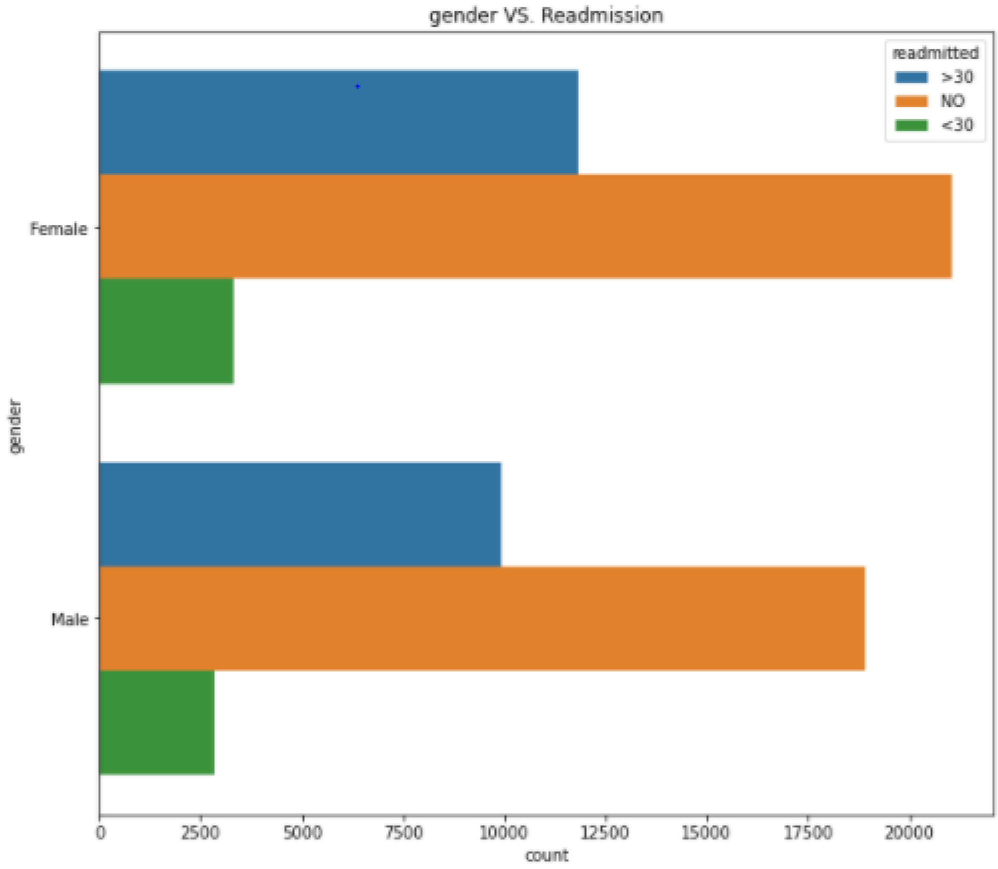
Caucassian race patients were more readmitted. Then comes the AfricanAmerican and the others are not likely readmitted.



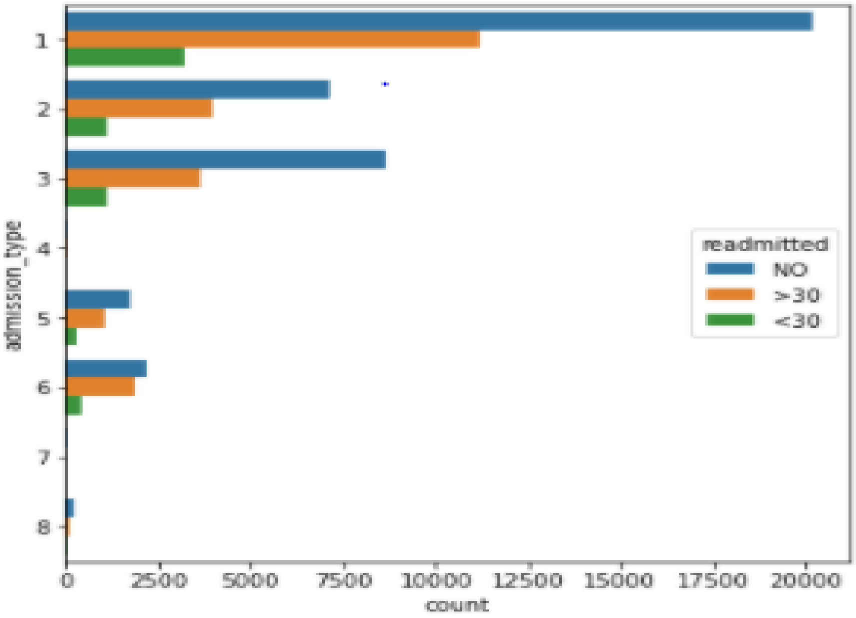
Patient who spent more time in the hospital are less likely readmitted and the patient who spent less time in hospital are more readmitted.



Patient whose admitting physician is missing/unknown are more readmitted.



Compared to female, male are less likely to be readmitted.



Patient who were admitted as emergency cases are more likely to be readmitted.

2.3.1 Handling missing values:

There were several features that could not be treated directly since they had a high percentage of missing values. Initial dataset has features like weight with 97% missing values, payer code with 40% and medical specialty with 47% missing values. Weight attribute was 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. Medical specialty attribute was maintained, adding the value "missing" in order to account for missing values as it has impact on the target variable.

2.3.2 Feature engineering

Below is the detailed compilation of techniques, transformations applied to the initial dataset to extract the relevant and important features which has impact on the outcome or target variable 'readmitted'.

As part of this, firstly, 'encounter_id' feature is dropped as it is irrelevant to the outcome 'readmission'. We only considered those patients who are alive, as the dead or hospice patients can't be readmitted. So the records where the 'discharge_disposition_id' feature has values 11, 13, 14, 19, 20, 21 are not considered for further analysis. After this step, dataset has 99343 records with 47 features.

The preliminary dataset contains multiple encounters for some patients. We thus only one counter per patient; in particular, we considered only the first encounter for each patient as the primary admission and dropped 'patient_nbr' feature. Feature 'Gender' has three values i.e., Male, Female, Unknown/Invalid. Count of records with gender value 'Unknown/Invalid' are only 3. So, these records are not considered. Feature 'Age' is grouped/ categorized into 8 groups and they are replaced with mean value of the age interval for each record. For example, age with [40-50) is replaced with 35. The first three intervals [0-10), [10-20), [20-30) are grouped together as [0-30) and replaced with 15. Feature 'discharge_disposition_id' has 28 integer values. These 28 values are grouped into 2 groups i.e., the value with '1' is replaced with 'discharge to home' and the rest values are grouped as 'Other'. At this point, the dataset has 69987 records with 46 features.

Feature 'diag_1' has more than 650 unique values. It will be difficult for the learner to manage too many values. So these values are grouped into 9 groups namely Circulatory, Respiratory, Digestive, Diabetes, Injury, Musculoskeletal, Genitourinary, Neoplasms , others according to the ICD-9 codes as follows:

Group name	ICD 9 codes	Number of encounters	Description
Circulatory	390–459, 785	21389	Diseases of the circulatory system
Respiratory	460-519,786	9491	Diseases of the respiratory system
Digestive	520-579, 787	6488	Diseases of the digestive system
Diabetes	250.xx	5748	Diabetes mellitus
Injury	800-999	4694	Injury and poisoning
Musculoskeletal	710-739	4064	Diseases of the musculoskeletal

			system and connective tissue
Genitourinary	580–629, 788	3441	Diseases of the genitourinary system
Neoplasms	140–239	2538	Neoplasms
Others	780, 781, 784, 790–799, 240–279, without 250, 680–709, 782, 001–139, 290–319, E–V, 280–289, 320–359, 630–679	12124	Other symptoms, signs, and ill-defined conditions, Endocrine, nutritional, and metabolic diseases and immunity disorders, without diabetes, Diseases of the skin and subcutaneous tissue, Infectious and parasitic diseases, Mental disorders, External causes of injury and supplemental classification, Diseases of the blood and blood-forming organs, Diseases of the nervous system, Complications of pregnancy, childbirth, and the puerperium, Diseases of the sense organs, Congenital anomalies

Dropped ‘diag_2’, ‘diag_2’ features. At this point, the dataset has 68061 records with 44 features.

Dropping the features glyburide-metformin','glipizide-metformin','glimepiride-pioglitazone','glimepiride-pioglitazone','metformin-pioglitazone','metformin-rosiglitazone', 'metformin', 'repaglinide', 'nateglinide', 'chlorpropamide', 'acetohexamide', 'tolbutamide', 'acarbose', 'miglitol', 'troglitazone', 'tolazamide', 'glimepiride', 'glipizide', 'glyburide', 'pioglitazone', 'rosiglitazone', 'examide', 'citoglipton' as they not prescribed is more than 75% cases and including these features can bias the model. This resulted in the dataset of shape (68061, 22).

Normalization:

All the numeric features are normalized on the scale of 0-1 to make sure the model is not altered in classifying correctly.

Grouping/ categorizing features values:

Feature ‘admission_type_id’ has 8 unique integer values. These 8 values are categorized into 4 groups i.e., 1 as emergency, 2 as urgent, 3 as elective, rest as other (has all values which are combinedly less than 5% of total records). In the similar way, ‘race’ feature is categorized into 3 categories namely, Caucasian, AfricanAmerican and Other(has all values which are combinedly less than 5% of total records). Also, ‘admission_source_id’ is categorized into 3 categories i.e, 7 as emergency room, 1 as physician referred and other (has all values which are combinedly less than 5% of total records).

Encoding on feature values:

For feature ‘A1Cresult’, ‘>7’ and ‘>8’ representing the positive results of the test as ‘1’, ‘Norm’ as 0 and ‘None’ as -1000. For feature ‘max_glu_serum’, ‘>200’ and ‘>300’ representing the high levels of glucose are encoded as ‘1’, ‘Norm’ representing normal levels of glucose is encoded as’0’ and ‘None’ as -1000.

As a result of the above preprocessing, the final dataset that can be applied for further analysis has 67836 records with 22 features.

Handling imbalanced data:

Number of records for each class value of the target variable is as follows:

- No (Not readmitted) : 39953
- >30 (Readmitted after 30 days) : 21751
- <30 (Readmitted within 30 days) : 6132

The above data clearly states that the data is imbalanced. This can bias the model. So, performed balanced sampling technique using SMOTETomek to balance the data by reducing the majority class and duplicating the records of minority class. After performing sampling, the results are as follows:

- 'NO': 39953,
- '<30': 39572,
- '>30': 21751

Feature selection:

All the features other than the target variable is considered as features denoted as x and the target denoted as y. one-hot encoding technique is applied to all the categorical features which resulted in a total of 36 features.

Test and train split of data:

Total data of size (101276, 21) is divided into train data and test data by shuffling the data to avoid biasing. Test data contributes to 33% and the train data contributes to 77% of the data.

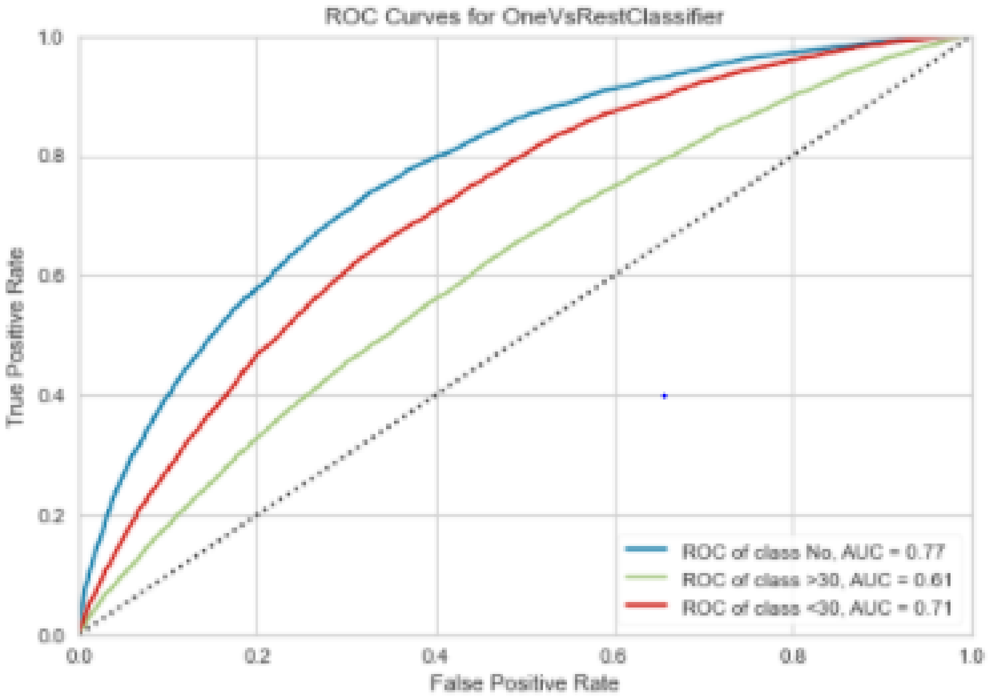
TASK 1: Model construction, metrics and results for ‘Readmitted’ target:

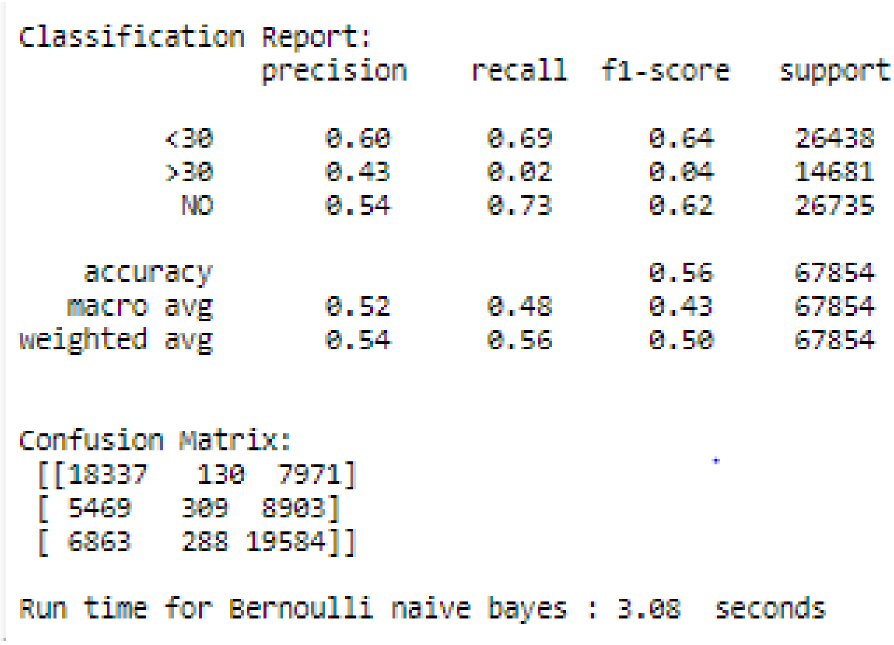
OnevsRestClassifier is applied for all the algorithms to handle multiclass classification. The following supervised algorithms are applied to the multiclass classification problem:

1. **Bernouli’s naïve bayes**

   Bernoulis version of naïve bayes fitted on train data resulted in accuracy of 56%. Runtime calculated is 3.08 seconds.

   Results:



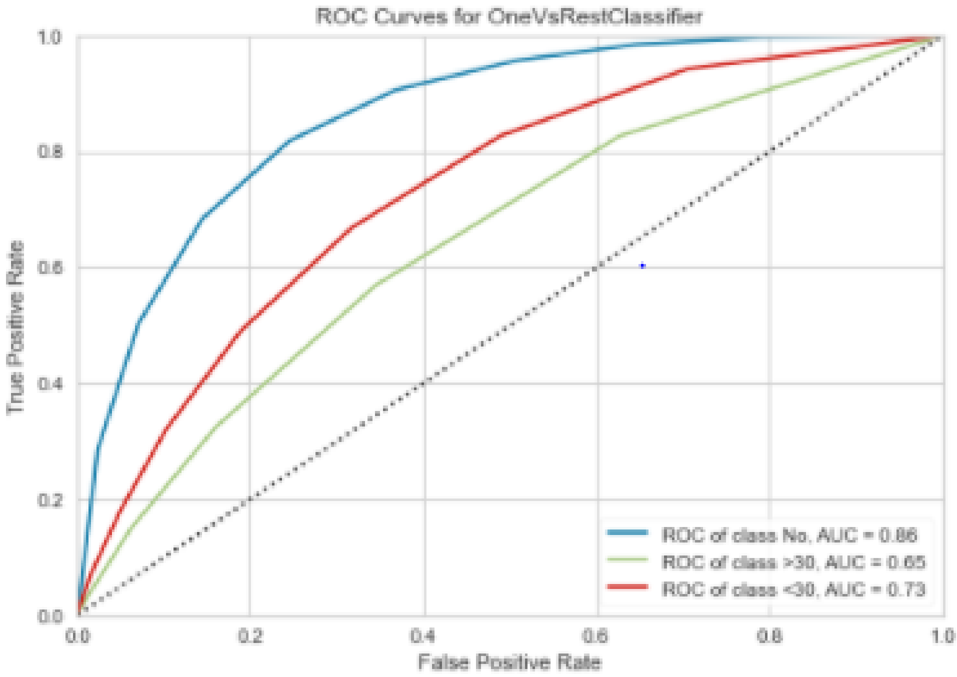


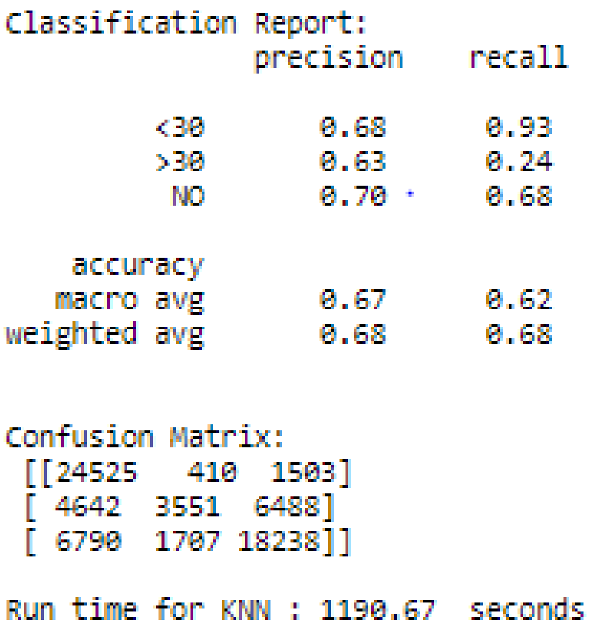
```
Classification Report:
               precision    recall  f1-score   support

         <30       0.60      0.69      0.64     26438
         >30       0.43      0.02      0.04     14681
          NO       0.54      0.73      0.62     26735

    accuracy                           0.56     67854
   macro avg       0.52      0.48      0.43     67854
weighted avg       0.54      0.56      0.50     67854


Confusion Matrix:
 [[18337   130  7971]
 [ 5469   309  8903]
 [ 6863   288 19584]]

Run time for Bernoulli naive bayes : 3.08  seconds
```

1. **K-Nearest Neighbour**

   K-NN fitted on the train data resulted in the accuracy of 68%. When different values of K is given to the model, the accuracy fluctuates. But the best accuracy was given at a value of 8. Runtime calculated is 1190.67 seconds.

   Results:



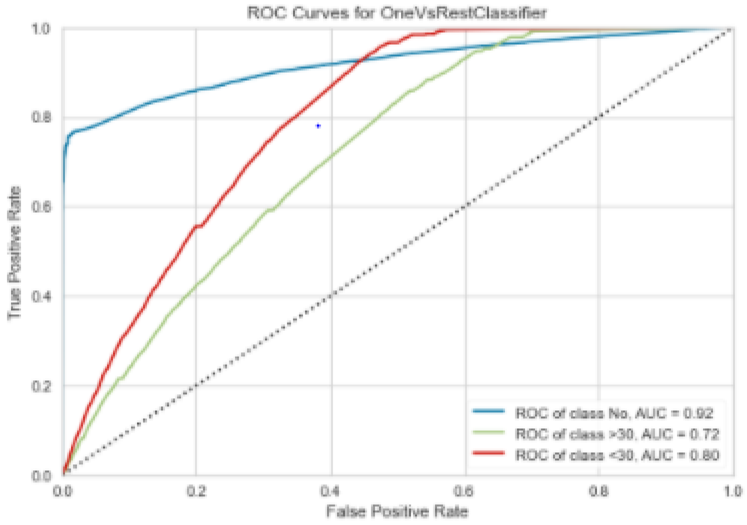


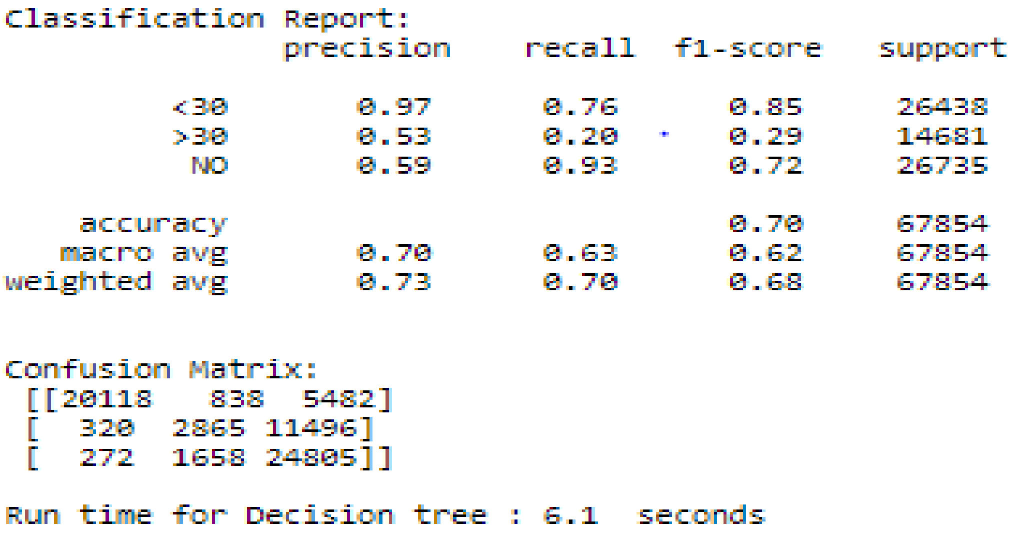
```
Classification Report:
               precision    recall  f1-score   support

         <30       0.68      0.93      0.79     26438
         >30       0.63      0.24      0.35     14681
          NO       0.70      0.68      0.69     26735

    accuracy                           0.68     67854
   macro avg       0.67      0.62      0.61     67854
weighted avg       0.68      0.68      0.65     67854


Confusion Matrix:
 [[24525   410  1503]
 [ 4642  3551  6488]
 [ 6790  1707 18238]]

Run time for KNN : 1190.67  seconds
```

2. Decision tree

DecisionTree Classifier with entropy criterion is fitted on the train data, it resulted in 70.4%. when the parameters of the classifier were tuned (for example, max_depth increased), the classifier projects the best accuracy at a depth of 10. Runtime calculated is 6.1 seconds.

Results:



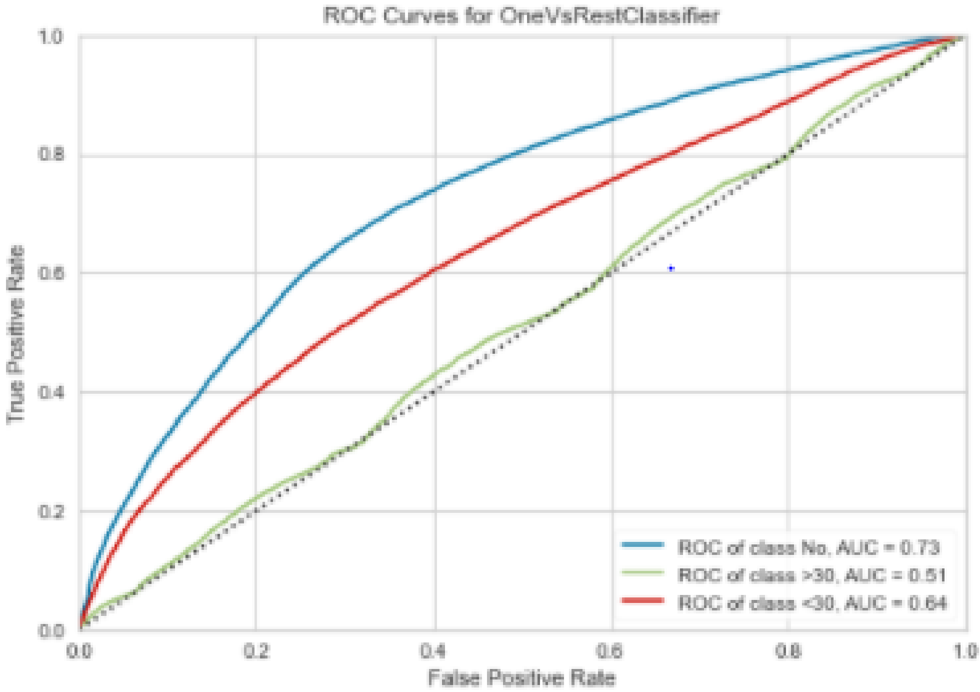


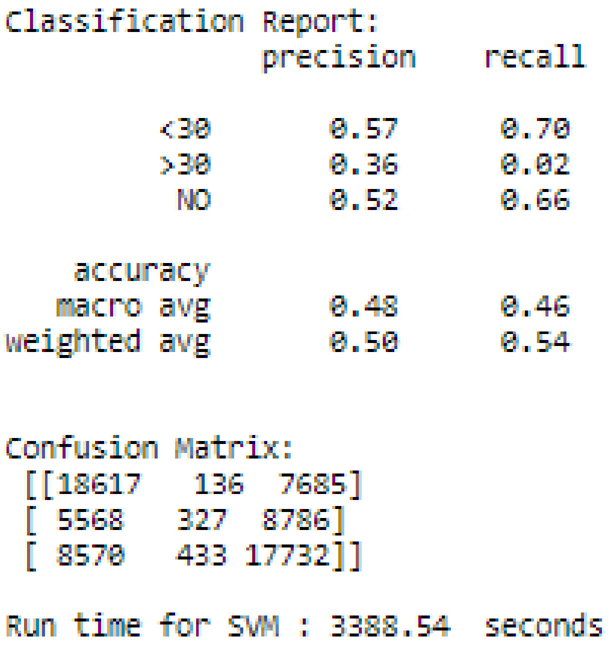
```
Classification Report:
              precision    recall  f1-score   support

         <30       0.97      0.76      0.85     26438
         >30       0.53      0.20      0.29     14681
          NO       0.59      0.93      0.72     26735

    accuracy                           0.70     67854
   macro avg       0.70      0.63      0.62     67854
weighted avg       0.73      0.70      0.68     67854


Confusion Matrix:
 [[20118   838  5482]
 [  320  2865 11496]
 [  272  1658 24805]]

Run time for Decision tree : 6.1  seconds
```

3. **Linear models – SVM**

SVM with linear kernel fitted on the train data, resulted the best accuracy of 54%. Runtime calculated is 3388.54 seconds

Results:



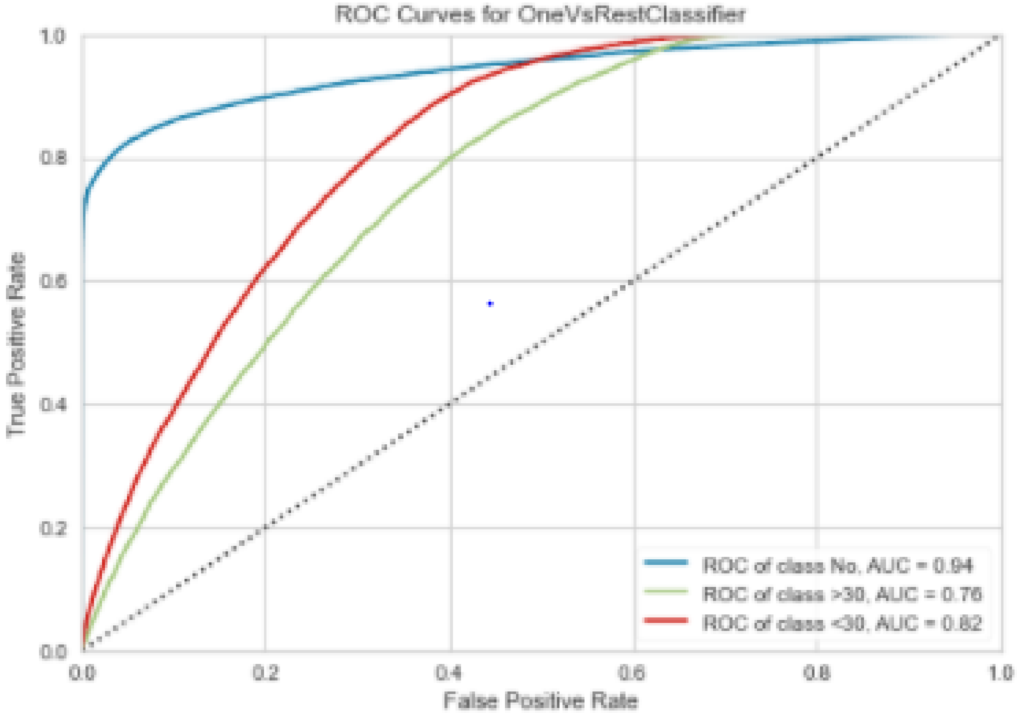


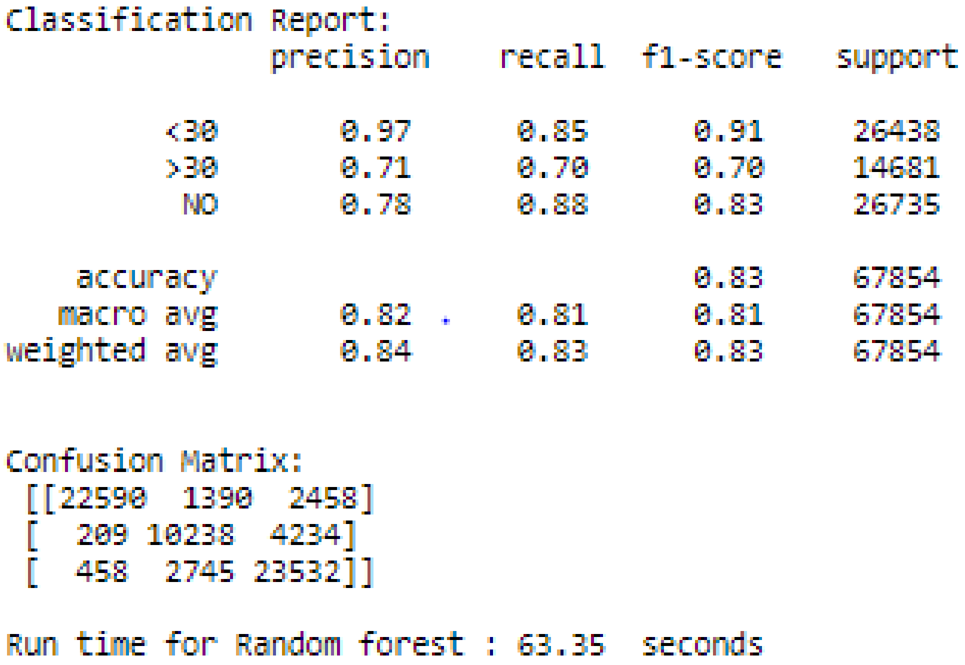
```
Classification Report:
              precision    recall  f1-score   support

         <30       0.57      0.70      0.63     26438
         >30       0.36      0.02      0.04     14681
          NO       0.52      0.66      0.58     26735

    accuracy                           0.54     67854
   macro avg       0.48      0.46      0.42     67854
weighted avg       0.50      0.54      0.48     67854


Confusion Matrix:
 [[18617   136  7685]
 [ 5568   327  8786]
 [ 8570   433 17732]]

Run time for SVM : 3388.54  seconds
```

4. **Ensembles - Random forest**

Randomforest classifier fitted on train data resulted the best accuracy of 83%. When the parameters of the classifier like n_estimators, max_depth are tuned, the accuracy goes up. At 100 estimators with depth of 15, considering the balanced class weight, random forest shows best accuracy. Runtime calculated is 63.35 seconds

Results:



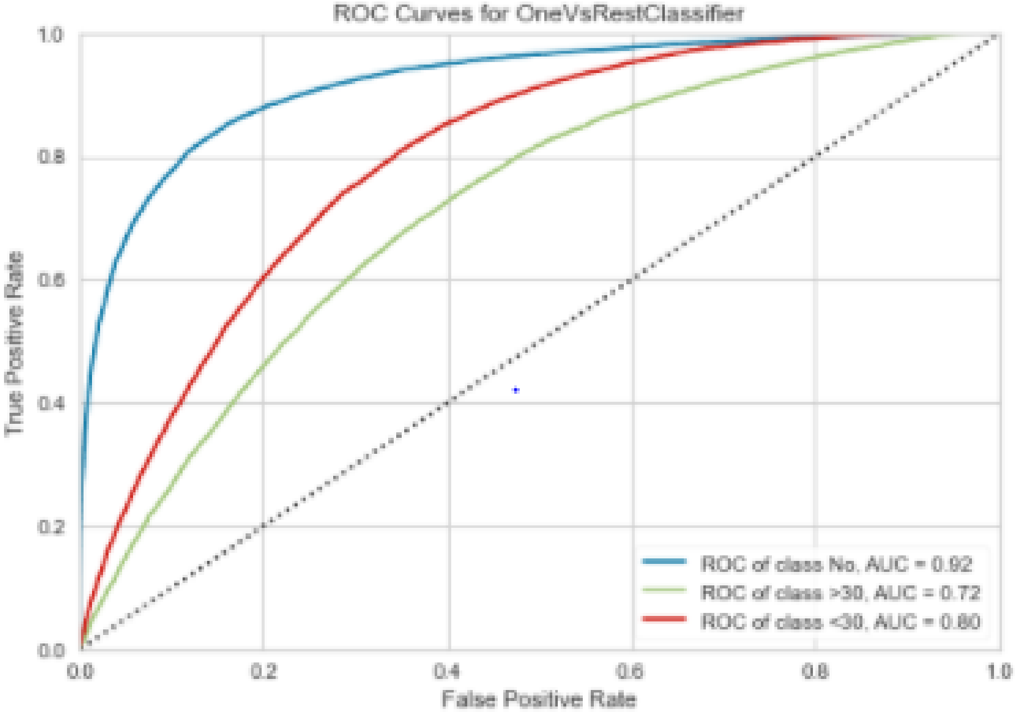


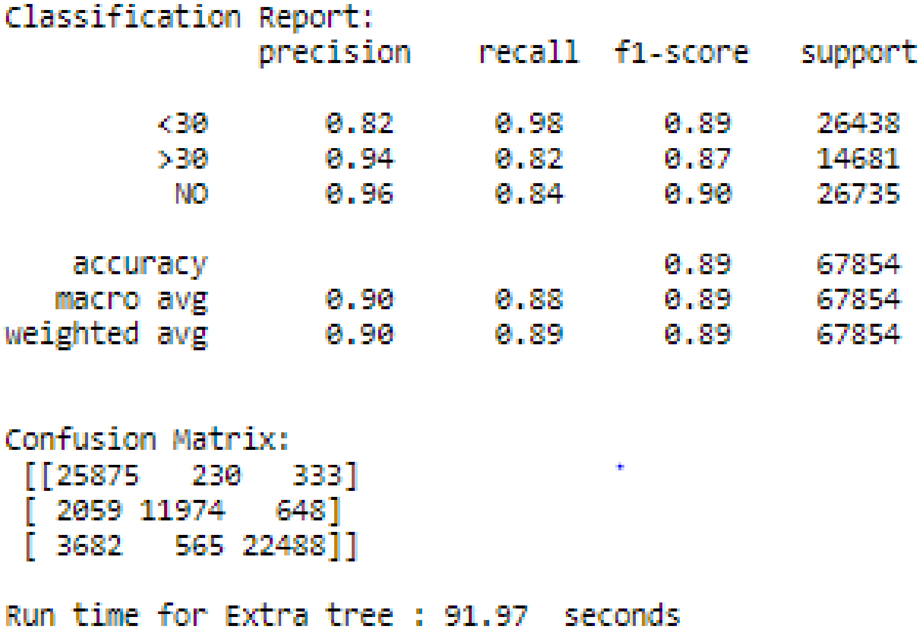
```
Classification Report:
              precision    recall  f1-score   support

         <30       0.97      0.85      0.91     26438
         >30       0.71      0.70      0.70     14681
          NO       0.78      0.88      0.83     26735

    accuracy                           0.83     67854
   macro avg       0.82      0.81      0.81     67854
weighted avg       0.84      0.83      0.83     67854


Confusion Matrix:
 [[22590  1390  2458]
 [  209 10238  4234]
 [  458  2745 23532]]

Run time for Random forest : 63.35  seconds
```

5. **Ensembles – Extra Tree**

   ExtraTree classifier fitted on train data resulted the best accuracy of 88.9%. When the parameters of the classifier like n_estimators, max_depth, criterion are tuned, the accuracy goes up. At 100 estimators with depth of 20, considering the balanced class weight, random forest shows best accuracy. Runtime calculated is 91.97 seconds.

   Results:



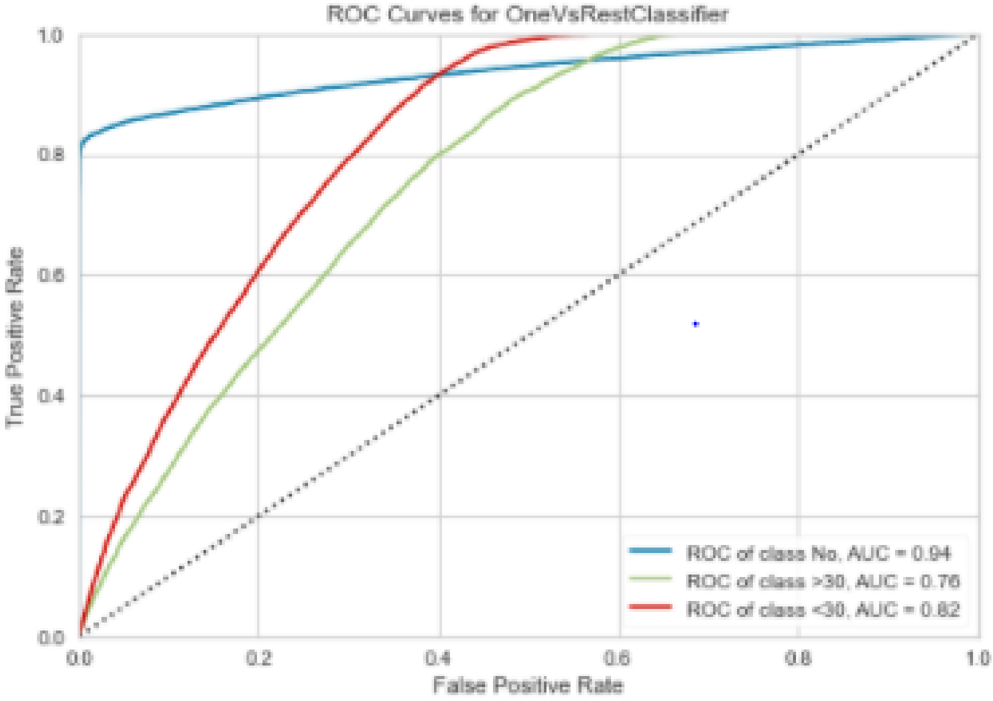


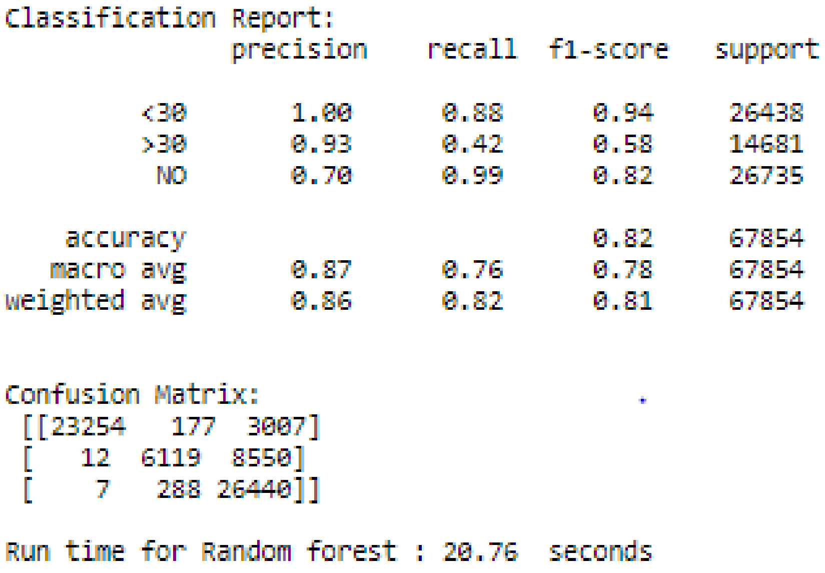
```
Classification Report:
              precision    recall  f1-score   support

         <30       0.82      0.98      0.89     26438
         >30       0.94      0.82      0.87     14681
          NO       0.96      0.84      0.90     26735

    accuracy                           0.89     67854
   macro avg       0.90      0.88      0.89     67854
weighted avg       0.90      0.89      0.89     67854


Confusion Matrix:
 [[25875   230   333]
 [ 2059 11974   648]
 [ 3682   565 22488]]

Run time for Extra tree : 91.97  seconds
```

6. **Ensembles – Bagging with decision tree**

Bagging classifier with base_estimator as decision tree classifier resulted the best accuracy of 82.2 % when the max_depth parameter of decision tree is 17 and entropy is considered as criterion. Runtime calculated is 20.76 seconds.

Results:

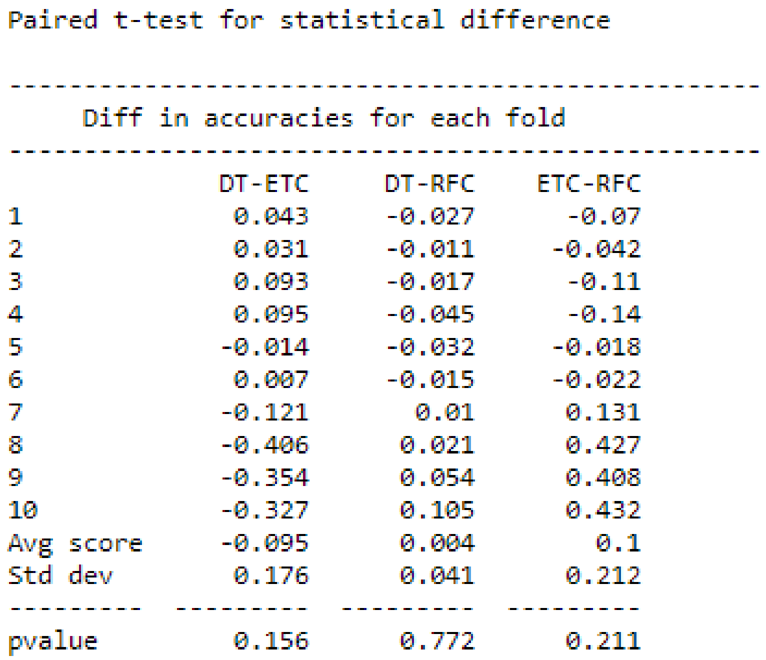




```
Classification Report:
              precision    recall  f1-score   support

         <30       1.00      0.88      0.94     26438
         >30       0.93      0.42      0.58     14681
          NO       0.70      0.99      0.82     26735

    accuracy                           0.82     67854
   macro avg       0.87      0.76      0.78     67854
weighted avg       0.86      0.82      0.81     67854


Confusion Matrix:
 [[23254   177  3007]
 [   12  6119  8550]
 [    7   288 26440]]

Run time for Random forest : 20.76  seconds
```

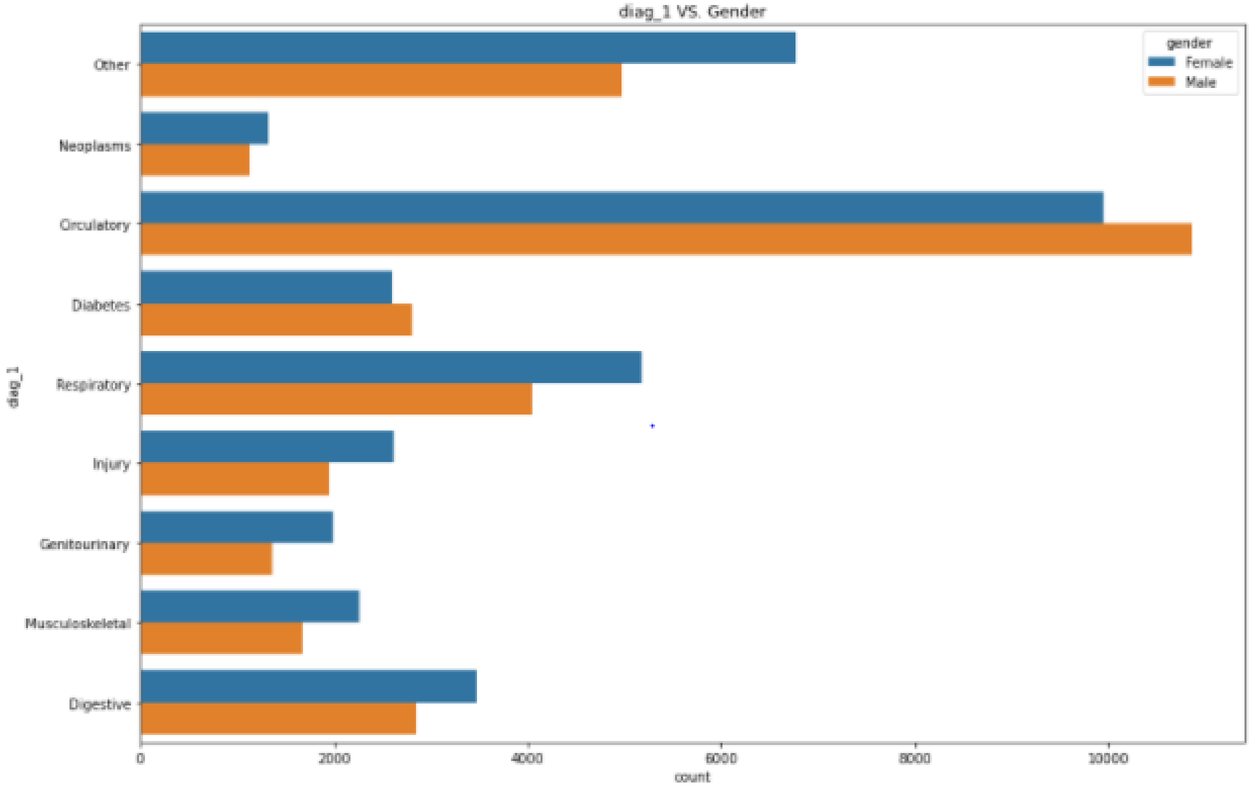
Statistical difference between the algorithms

Out of all the supervised algorithms applied, we consider extra trees classifier, random forest and decision tree models. Once again, cross validation is performed on the dataset with 67k records without performing resampling. All the three classifiers are applied to the OnevsRest Classifier and cross validation is applied. The average accuracies were 55.1 % for decision tree, 54.6 % for random forest and 64.6% for extra tree classifier. After obtaining these results, paired t-test is applied on these results to calculate the statistical difference. The p- tree value is 0.156 for decision tree and Extra tree, 0.772 for decision tree and random forest and 0.211 for extra tree and random forest.

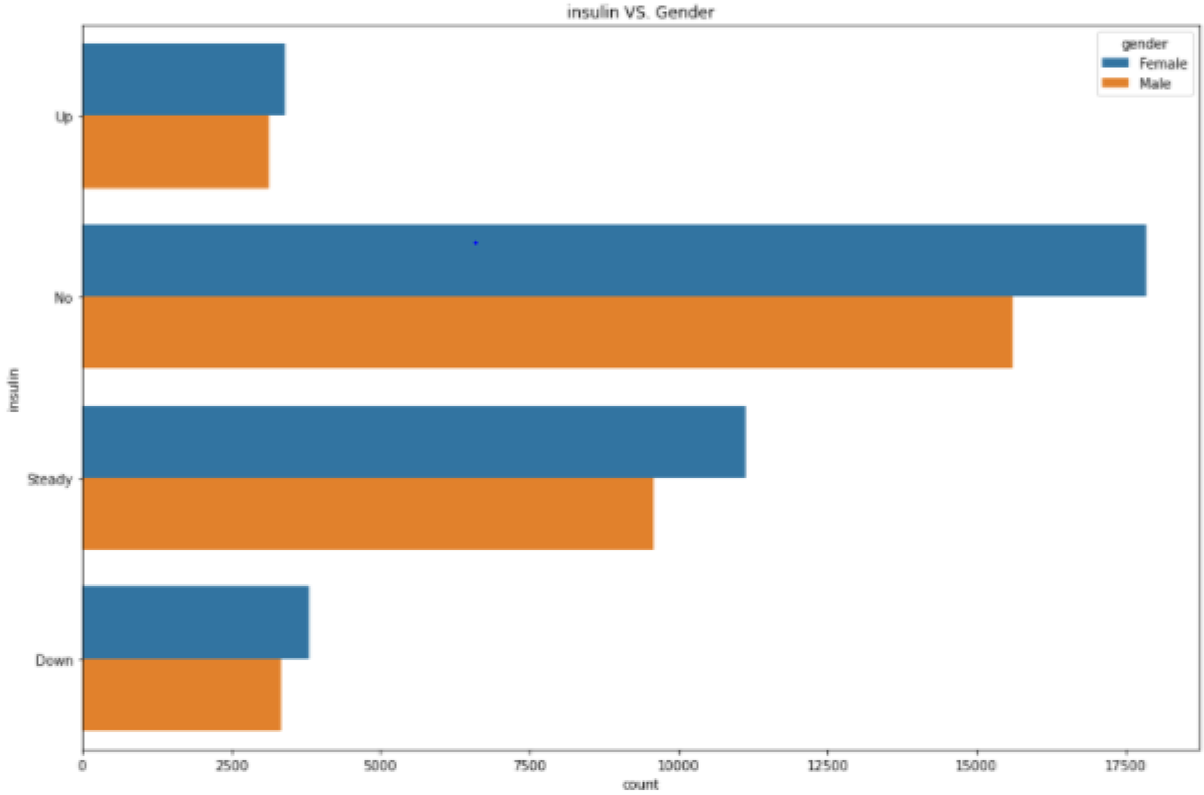


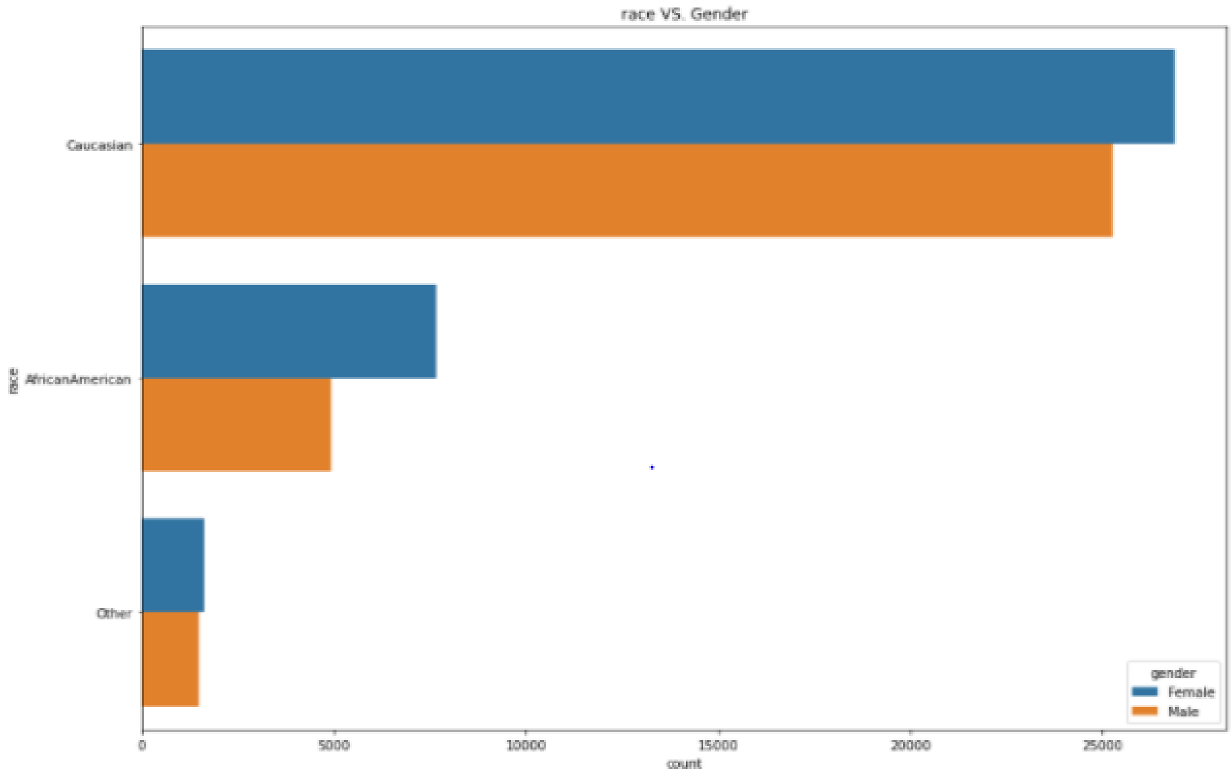
```
Paired t-test for statistical difference

--------------------------------------------------
     Diff in accuracies for each fold
--------------------------------------------------
                DT-ETC      DT-RFC     ETC-RFC
1                0.043      -0.027       -0.07
2                0.031      -0.011      -0.042
3                0.093      -0.017       -0.11
4                0.095      -0.045       -0.14
5               -0.014      -0.032      -0.018
6                0.007      -0.015      -0.022
7               -0.121        0.01       0.131
8               -0.406       0.021       0.427
9               -0.354       0.054       0.408
10              -0.327       0.105       0.432
Avg score       -0.095       0.004         0.1
Std dev          0.176       0.041       0.212
---------    ---------   ---------   ---------
pvalue           0.156       0.772       0.211
```

TASK 2: Data analysis



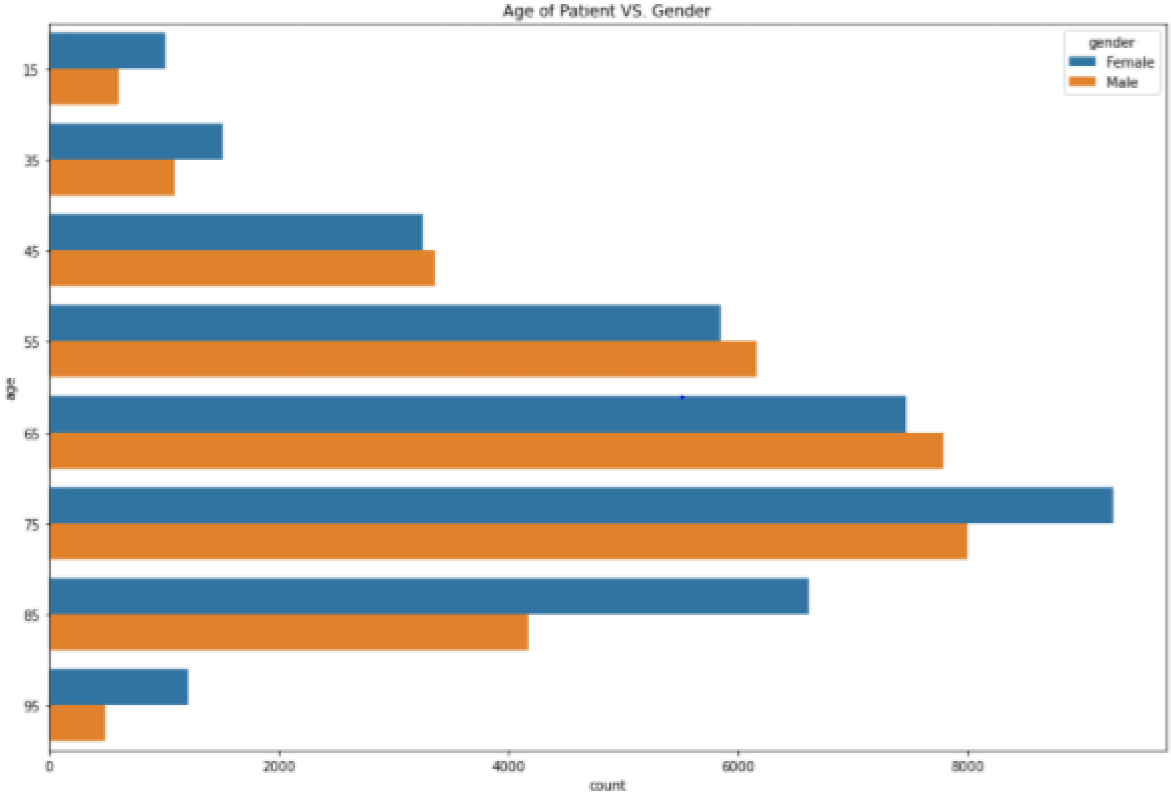
Both male and female are more prone to circulatory diagnosis. Both male and female less go for neoplasms diagnosis.





Most of patients are from causcassian race. Very less portion of patients are from other groups like Asian, othe r.

In majority case, both male and female are less prescribed for insulin. For very few male and female patients, i nsulin medication is either increased or decreased.



Most of the patients are belong to the age interval [70-80), [60-70), 950-60).

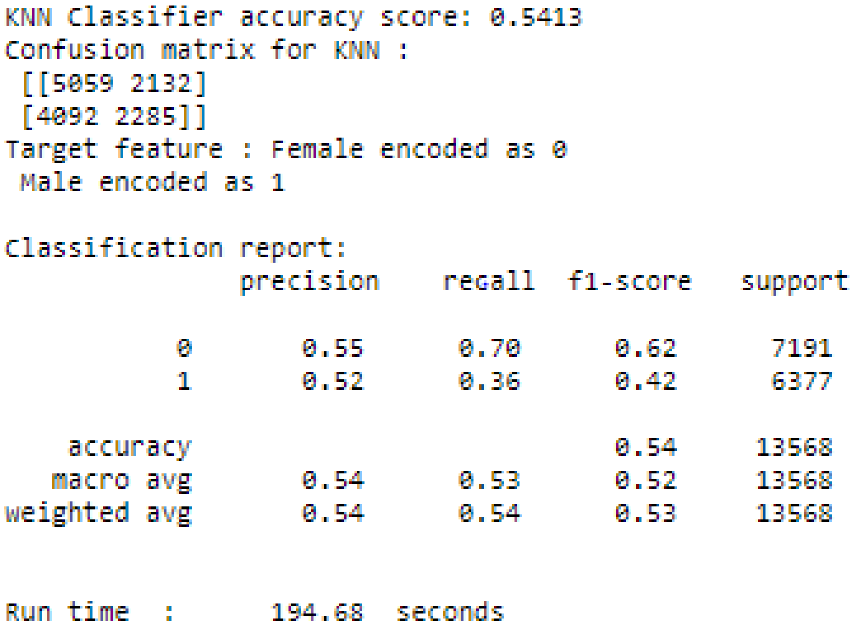
TASK 2: Model construction, metrics and results for ‘gender’ target:

The following supervised algorithms are applied to the binary classification problem:

1. **K-NN (K=8)**

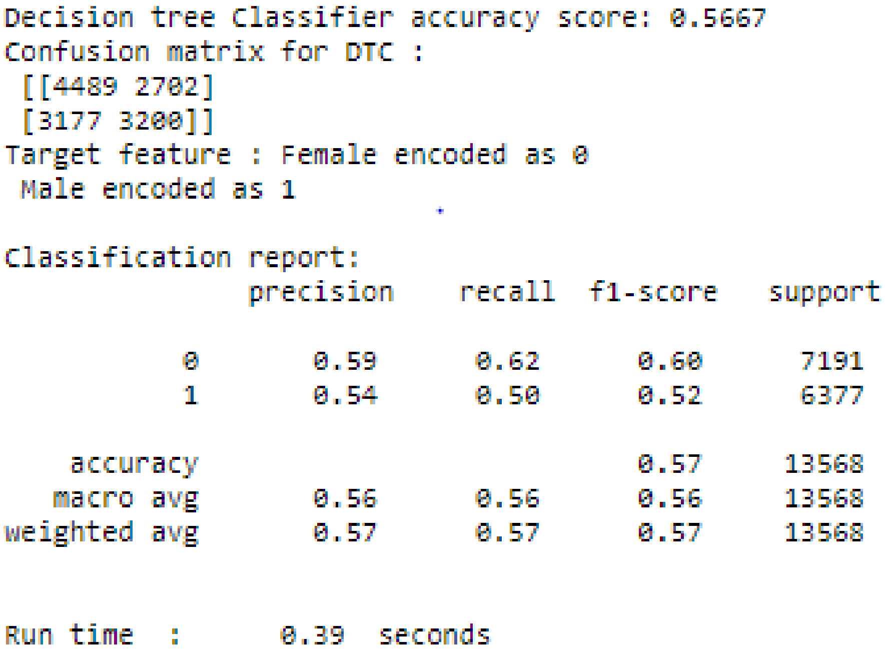
   K-NN fitted on the train data resulted in the accuracy of 54%. When different values of K is given to the model, the accuracy fluctuates. But the best accuracy was given at a value of 8. Runtime calculated is 194.68 seconds.

   Results



```
KNN Classifier accuracy score: 0.5413
Confusion matrix for KNN :
 [[5059 2132]
 [4092 2285]]
Target feature : Female encoded as 0
 Male encoded as 1

Classification report:
               precision    recall  f1-score   support

           0       0.55      0.70      0.62      7191
           1       0.52      0.36      0.42      6377

    accuracy                           0.54     13568
   macro avg       0.54      0.53      0.52     13568
weighted avg       0.54      0.54      0.53     13568


Run time  :      194.68  seconds
```

2. **Decision Tree**

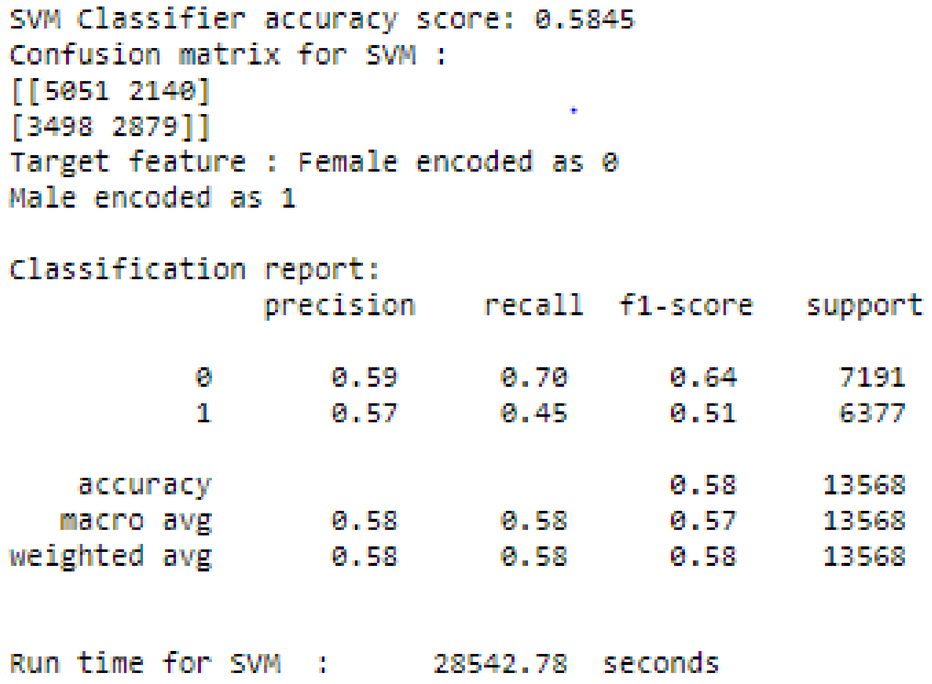


```
Decision tree Classifier accuracy score: 0.5667
Confusion matrix for DTC :
 [[4489 2702]
 [3177 3200]]
Target feature : Female encoded as 0
 Male encoded as 1

Classification report:
               precision    recall  f1-score   support

           0       0.59      0.62      0.60      7191
           1       0.54      0.50      0.52      6377

    accuracy                           0.57     13568
   macro avg       0.56      0.56      0.56     13568
weighted avg       0.57      0.57      0.57     13568


Run time  :      0.39   seconds
```

3. **SVM**

SVM with linear kernel fitted on the train data, resulted the best accuracy of 58%. Runtime calculated is 28542.78 seconds

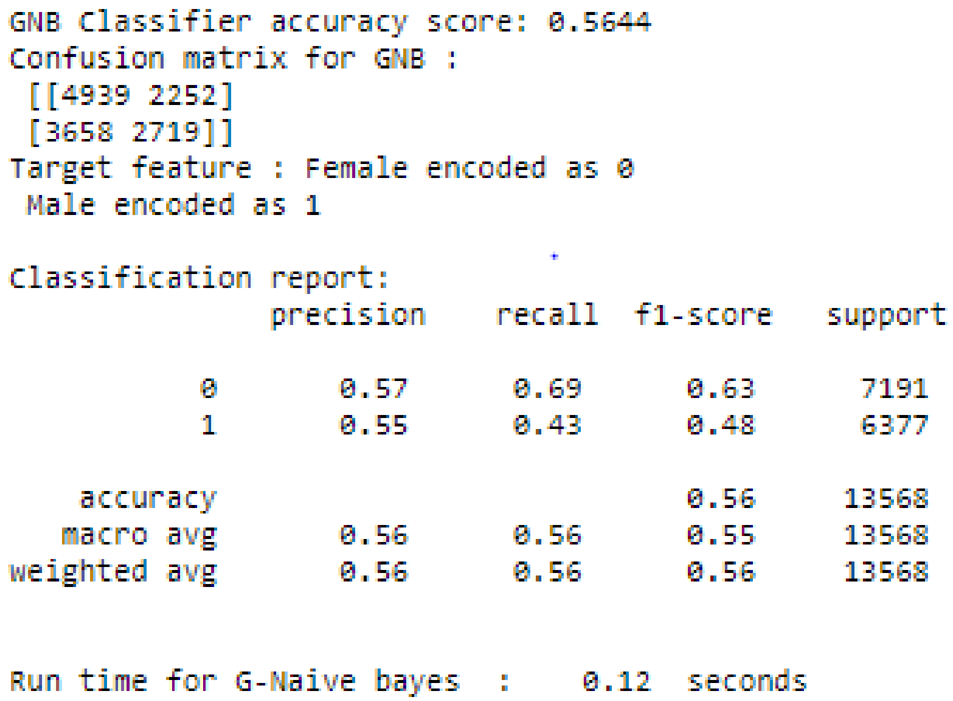
Results:



```
SVM Classifier accuracy score: 0.5845
Confusion matrix for SVM :
[[5051 2140]
[3498 2879]]
Target feature : Female encoded as 0
Male encoded as 1

Classification report:
               precision    recall  f1-score   support

           0       0.59      0.70      0.64      7191
           1       0.57      0.45      0.51      6377

    accuracy                           0.58     13568
   macro avg       0.58      0.58      0.57     13568
weighted avg       0.58      0.58      0.58     13568


Run time for SVM  :      28542.78   seconds
```

4. **Bernoulli Naïve Bayes**

Bernoulli Naïve Bayes fitted on the train data, resulted the best accuracy of 56 %. Runtime calcula ted is 0.12 seconds

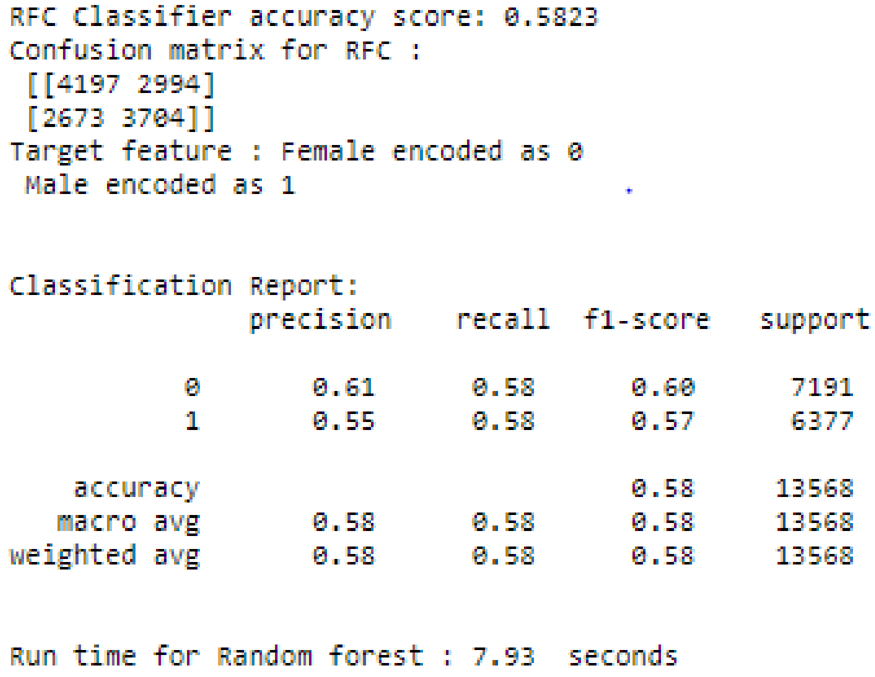


```
GNB Classifier accuracy score: 0.5644
Confusion matrix for GNB :
 [[4939 2252]
 [3658 2719]]
Target feature : Female encoded as 0
 Male encoded as 1

Classification report:
               precision    recall  f1-score   support

           0       0.57      0.69      0.63      7191
           1       0.55      0.43      0.48      6377

    accuracy                           0.56     13568
   macro avg       0.56      0.56      0.55     13568
weighted avg       0.56      0.56      0.56     13568


Run time for G-Naive bayes  :    0.12  seconds
```

5. **Random forest**

Randomforest classifier fitted on train data resulted the best accuracy of 58%. When the parameters of the classifier like n_estimators, max_depth are tuned, the accuracy goes up. At 100 estimators with depth of 15, considering the balanced class weight, random forest shows best accuracy. Runtime calculated is 7.93 seconds

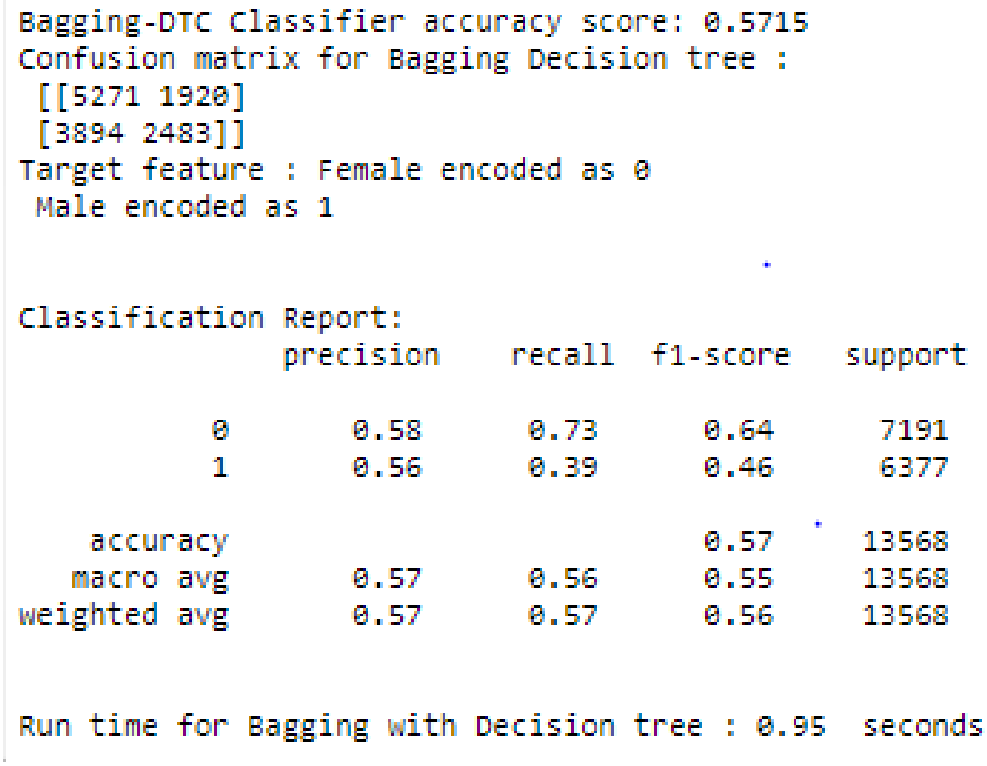
Results:



```
RFC Classifier accuracy score: 0.5823
Confusion matrix for RFC :
 [[4197 2994]
 [2673 3704]]
Target feature : Female encoded as 0
 Male encoded as 1


Classification Report:
               precision    recall  f1-score   support

           0       0.61      0.58      0.60      7191
           1       0.55      0.58      0.57      6377

    accuracy                           0.58     13568
   macro avg       0.58      0.58      0.58     13568
weighted avg       0.58      0.58      0.58     13568


Run time for Random forest : 7.93  seconds
```

5. **Bagging with decision tree**

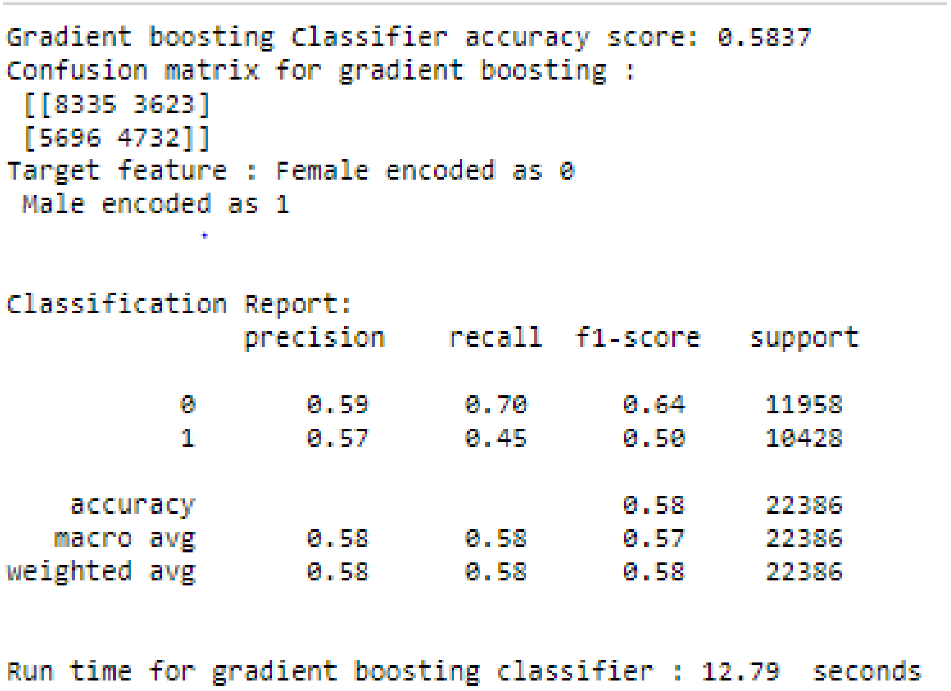
   Bagging classifier with base_estimator as decision tree classifier resulted the best accuracy of 57 % when the max_depth parameter of decision tree is 17 and entropy is considered as criterion. Runti me calculated is 0.95 seconds



```
Bagging-DTC Classifier accuracy score: 0.5715
Confusion matrix for Bagging Decision tree :
 [[5271 1920]
 [3894 2483]]
Target feature : Female encoded as 0
 Male encoded as 1


Classification Report:
              precision    recall  f1-score   support

           0       0.58      0.73      0.64      7191
           1       0.56      0.39      0.46      6377

    accuracy                           0.57     13568
   macro avg       0.57      0.56      0.55     13568
weighted avg       0.57      0.57      0.56     13568


Run time for Bagging with Decision tree : 0.95  seconds
```

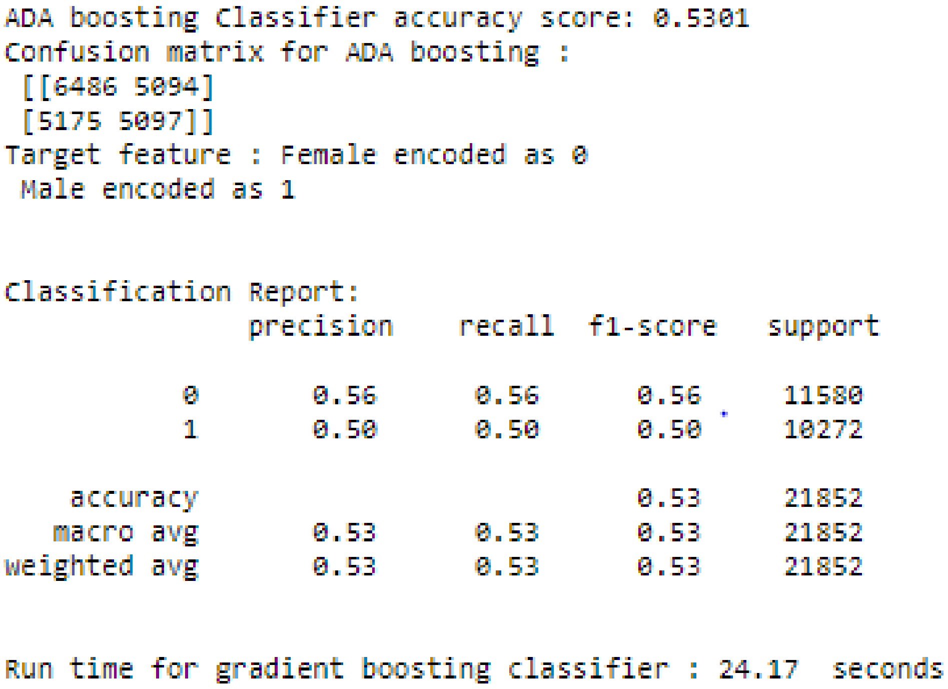
6. **Gradient boosting**

   Bagging classifier resulted the best accuracy of 58 %. Runtime calculated is 12.79 seconds



```
Gradient boosting Classifier accuracy score: 0.5837
Confusion matrix for gradient boosting :
 [[8335 3623]
 [5696 4732]]
Target feature : Female encoded as 0
 Male encoded as 1


Classification Report:
               precision    recall  f1-score   support

           0       0.59      0.70      0.64     11958
           1       0.57      0.45      0.50     10428

    accuracy                           0.58     22386
   macro avg       0.58      0.58      0.57     22386
weighted avg       0.58      0.58      0.58     22386


Run time for gradient boosting classifier : 12.79  seconds
```

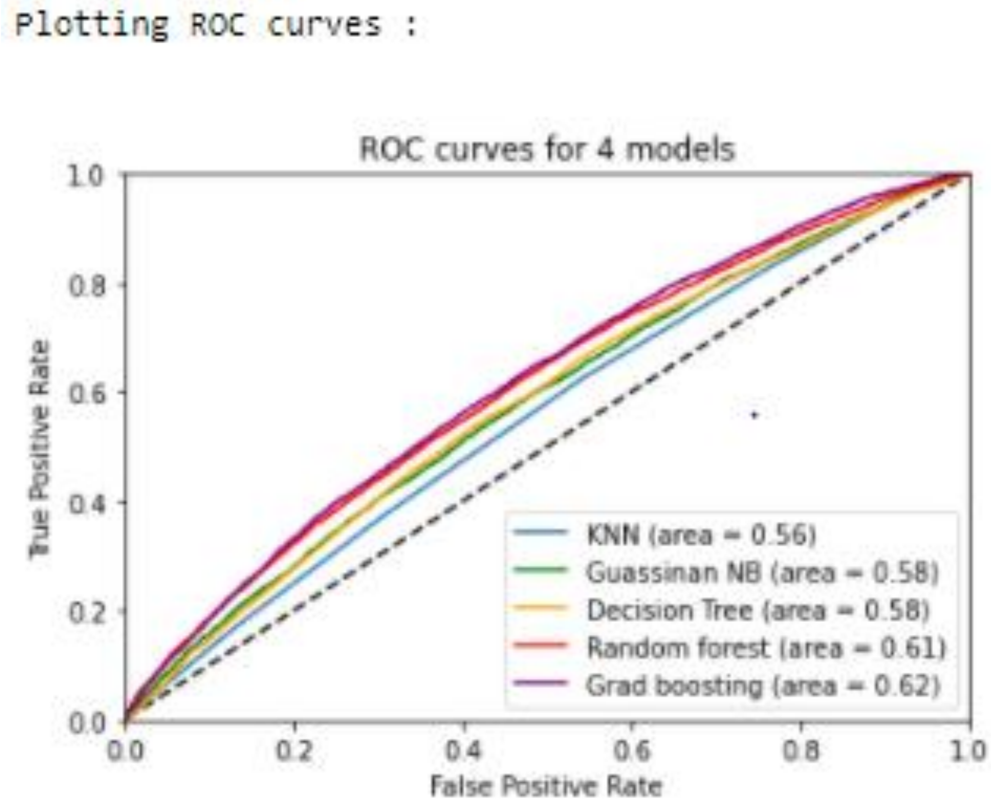
7. **Adaboosting**



```
ADA boosting Classifier accuracy score: 0.5301
Confusion matrix for ADA boosting :
 [[6486 5094]
 [5175 5097]]
Target feature : Female encoded as 0
 Male encoded as 1


Classification Report:
               precision    recall  f1-score   support

           0       0.56      0.56      0.56     11580
           1       0.50      0.50      0.50     10272

    accuracy                           0.53     21852
   macro avg       0.53      0.53      0.53     21852
weighted avg       0.53      0.53      0.53     21852


Run time for gradient boosting classifier : 24.17  seconds
```

Plotting ROC curves for 5 models:

Plotting ROC curves for KNN, Naïve bayes, decision tree, random forest, gradient boosting algorithms.



```
Plotting ROC curves :
```

B. Semi supervised learning:

The same preprocessing steps are performed to retrieve the final dataset. Semi supervised learning is performed in 2 ways. The first way is by performing resampling technique for the imbalanced data and the second way is not by performing any resampling technique. In this project, I succeeded in fully implementing two semi-supervised learning algorithms for task 1 (target variable 'readmitted') of supervised learning task.

The two-graph based transductive semi-supervised algorithms implemented are the label propagation and label spreading using scikit-learn module. For all the algorithms, label encoder is applied to the target variable for encoding as 0, 1, 2 for classes No, >30, <30. Two functions called maskfunc(true_target, percentage) and report_conf(unlabelled_data_Set, percen) are implemented.

- maskfunc(true_target, percentage) : Used for unlabelling the target feature by given percentage.by replacing the target feature value with -1.
- report_conf(unlabelled_data_Set, percen) : used for displaying the classification report and confusion matrix based on the unlabelled data and the percentage of unlabelled data.

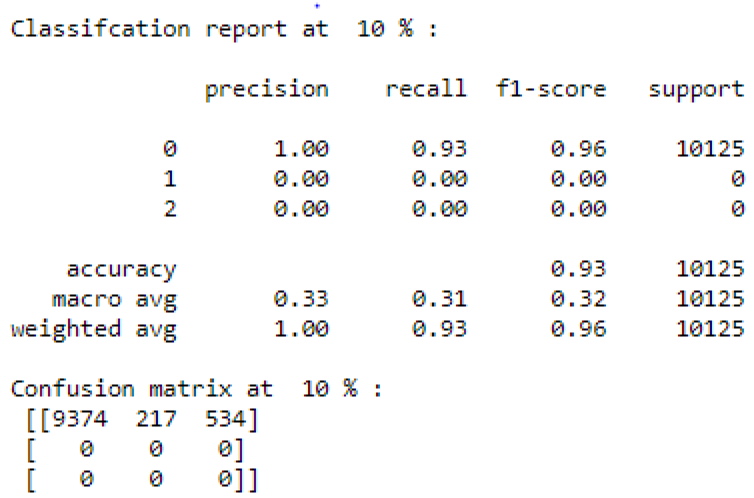
First discuss the model, results, evaluation about the Label propagation technique on resampled data and original final dataset at various level of unlabelled data like 0%, 10%, 20%, 50%, 90%, 95%.

1. **Label propagation (Resampled data)**

- **At 0% unlabelled data:**
  When considered 101250 records as labelled and 0 records unlabelled data, model was 93 % accu rate. Below are the results:

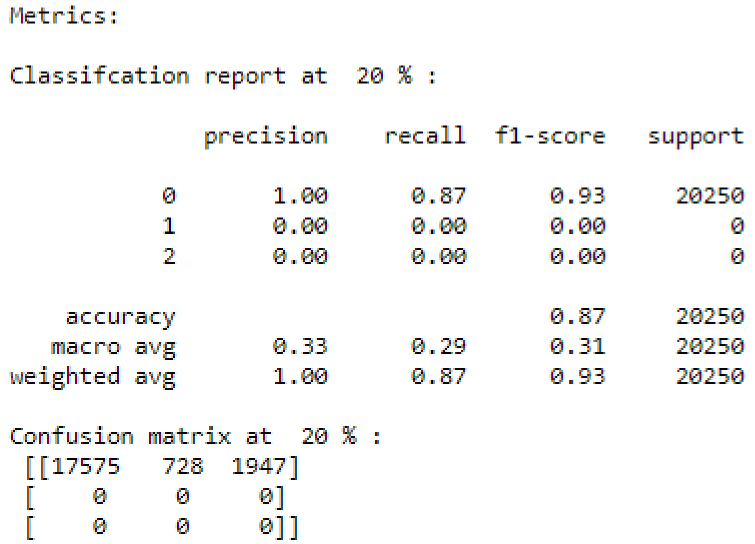
```
Total samples considered : 101250
Count of labelled points at  0 % : 101250
Count of unlabelled points at  0 % : 0

Accuracy with 0 % unlabelled data 0.9333
```

- **At 10% unlabelled data:**
  When considered 91125 records as labelled and 10125 records unlabelled data, model was 93 % accurate. Below are the results:



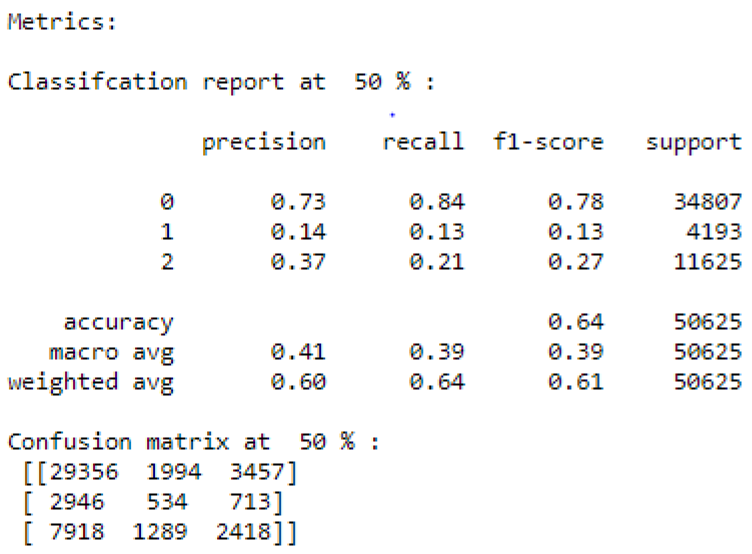
```
Classifcation report at  10 % :

              precision    recall  f1-score   support

           0       1.00      0.93      0.96     10125
           1       0.00      0.00      0.00         0
           2       0.00      0.00      0.00         0

    accuracy                           0.93     10125
   macro avg       0.33      0.31      0.32     10125
weighted avg       1.00      0.93      0.96     10125

Confusion matrix at  10 % :
 [[9374  217  534]
 [   0    0    0]
 [   0    0    0]]
```

- **At 20% unlabelled data:**
  When considered 81000 records as labelled and 20250 records unlabelled data, model was 87 % accurate. Below are the results:



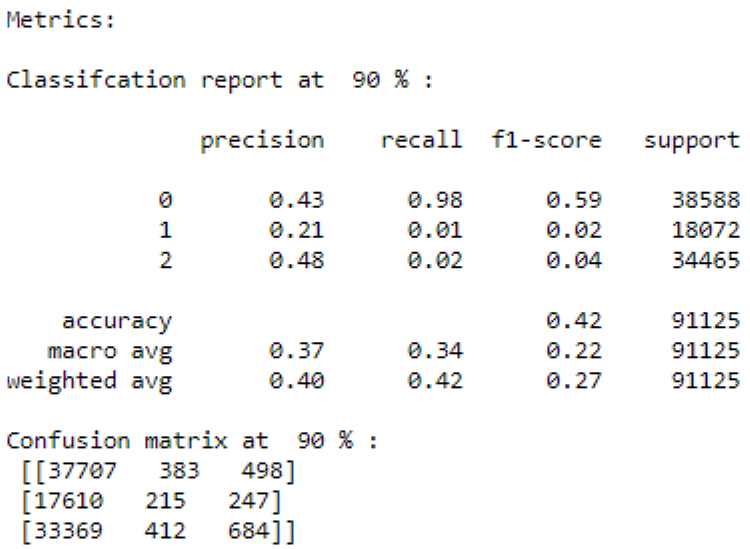
```
Metrics:

Classifcation report at  20 % :

              precision    recall  f1-score   support

           0       1.00      0.87      0.93     20250
           1       0.00      0.00      0.00         0
           2       0.00      0.00      0.00         0

    accuracy                           0.87     20250
   macro avg       0.33      0.29      0.31     20250
weighted avg       1.00      0.87      0.93     20250

Confusion matrix at  20 % :
 [[17575   728  1947]
 [    0     0     0]
 [    0     0     0]]
```

- **At 50% unlabelled data:**
  When considered 50625 records as labelled and 50625 records unlabelled data, model was 64 % accurate. Below are the results:



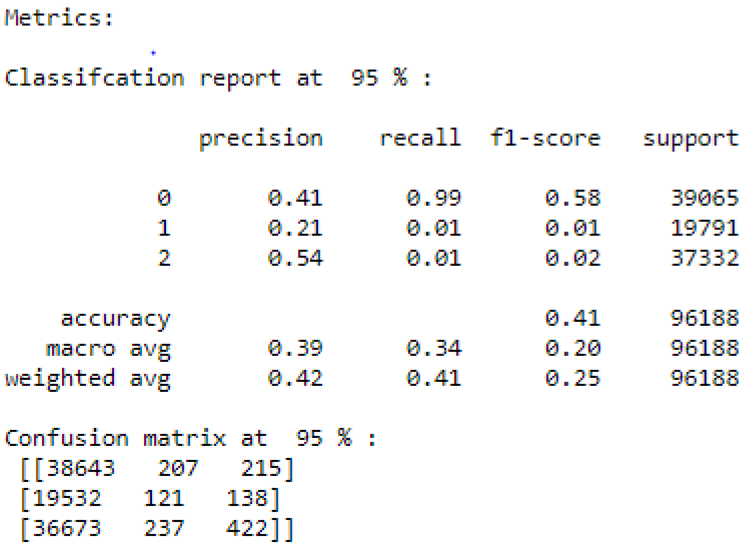
```
Metrics:

Classifcation report at  50 % :

              precision    recall  f1-score   support

           0       0.73      0.84      0.78     34807
           1       0.14      0.13      0.13      4193
           2       0.37      0.21      0.27     11625

    accuracy                           0.64     50625
   macro avg       0.41      0.39      0.39     50625
weighted avg       0.60      0.64      0.61     50625

Confusion matrix at  50 % :
 [[29356  1994  3457]
 [ 2946   534   713]
 [ 7918  1289  2418]]
```

- **At 90% unlabelled data:**
  When considered 10125 records as labelled and 91125 records unlabelled data, model was 42 % accurate. Below are the results:



```
Metrics:

Classifcation report at  90 % :

              precision    recall  f1-score   support

           0       0.43      0.98      0.59     38588
           1       0.21      0.01      0.02     18072
           2       0.48      0.02      0.04     34465

    accuracy                           0.42     91125
   macro avg       0.37      0.34      0.22     91125
weighted avg       0.40      0.42      0.27     91125

Confusion matrix at  90 % :
 [[37707   383   498]
 [17610   215   247]
 [33369   412   684]]
```

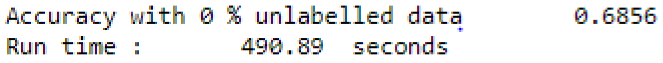
- **At 95% unlabelled data:**
  When considered 5062 records as labelled and 96188 records unlabelled data, model was 41 % a ccurate. Below are the results:



```
Metrics:

Classifcation report at  95 % :

              precision    recall  f1-score   support

           0       0.41      0.99      0.58     39065
           1       0.21      0.01      0.01     19791
           2       0.54      0.01      0.02     37332

    accuracy                           0.41     96188
   macro avg       0.39      0.34      0.20     96188
weighted avg       0.42      0.41      0.25     96188

Confusion matrix at  95 % :
 [[38643   207   215]
 [19532   121   138]
 [36673   237   422]]
```

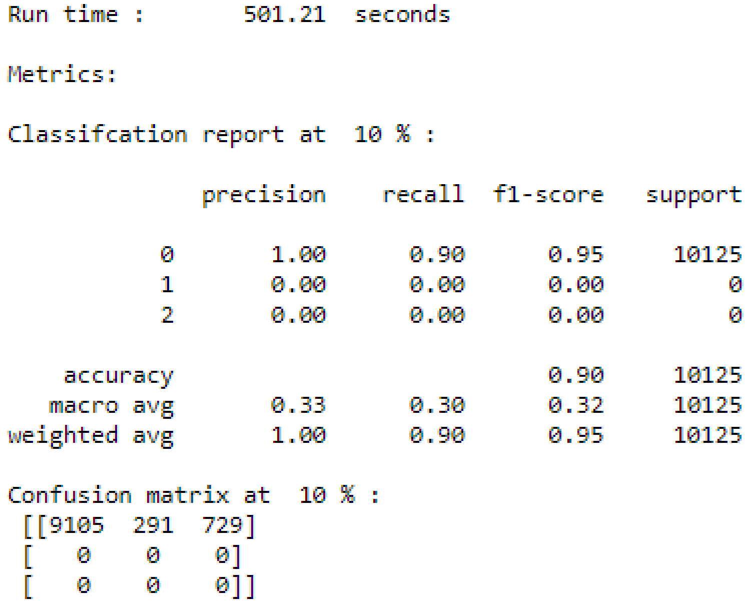
8. Label spreading (Resampled data)

- **At 0% unlabelled data:**
  When considered 101250 records as labelled and 0 records unlabelled data, model was 91 % accu rate. Below are the results:



```
Accuracy with 0 % unlabelled data        0.6856
Run time :       490.89  seconds
```

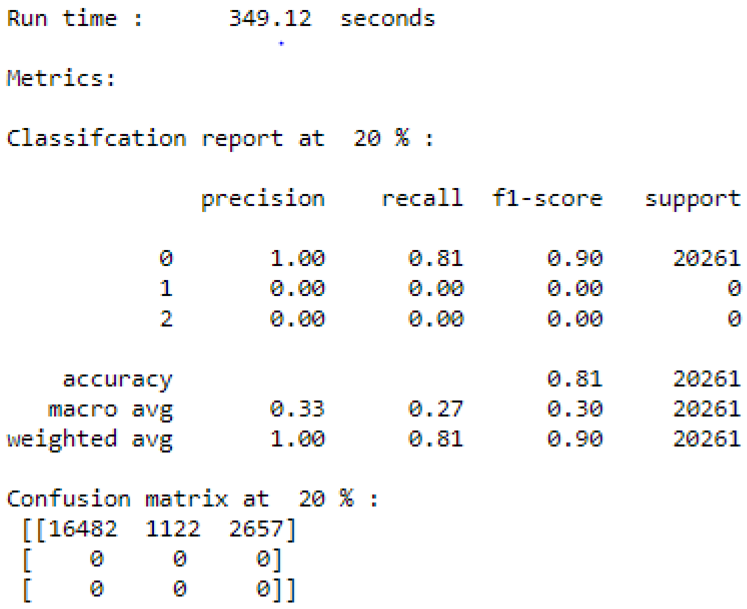
- **At 10% unlabelled data:**
  When considered 91125 records as labelled and 10125 records unlabelled data, model was 90 % accurate. Below are the results:



```
Run time :       501.21  seconds

Metrics:

Classifcation report at  10 % :

              precision    recall  f1-score   support

           0       1.00      0.90      0.95     10125
           1       0.00      0.00      0.00         0
           2       0.00      0.00      0.00         0

    accuracy                           0.90     10125
   macro avg       0.33      0.30      0.32     10125
weighted avg       1.00      0.90      0.95     10125

Confusion matrix at  10 % :
 [[9105  291  729]
 [   0    0    0]
 [   0    0    0]]
```

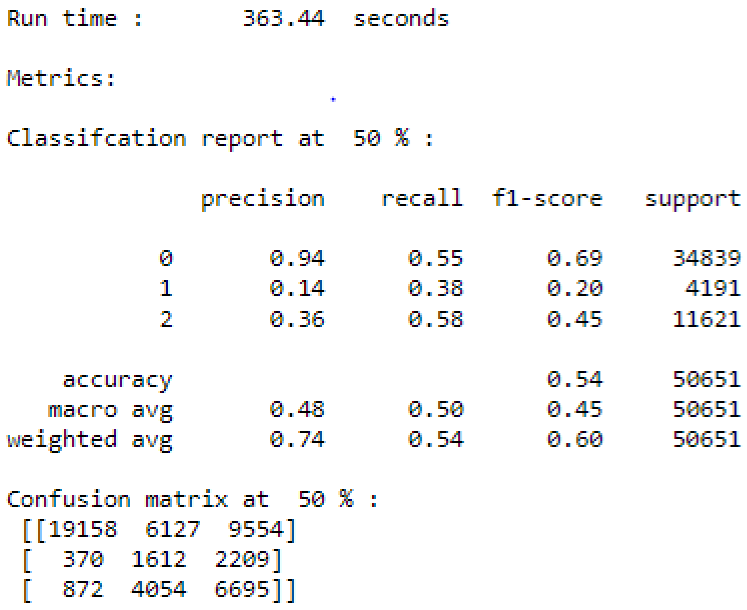
- **At 20% unlabelled data:**

When considered 81041 records as labelled and 20261 records unlabelled data, model was 81 % accurate. Below are the results:



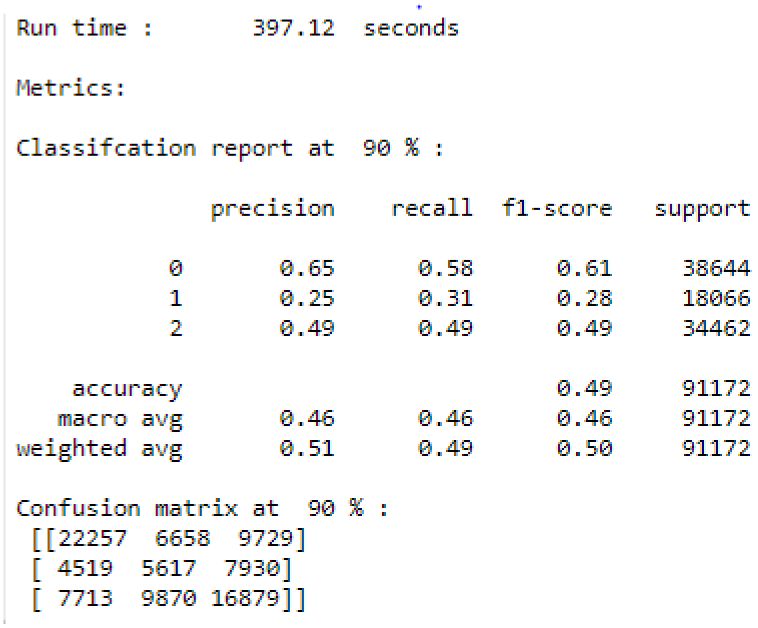
```
Run time :      349.12  seconds

Metrics:

Classifcation report at  20 % :

              precision    recall  f1-score   support

           0       1.00      0.81      0.90     20261
           1       0.00      0.00      0.00         0
           2       0.00      0.00      0.00         0

    accuracy                           0.81     20261
   macro avg       0.33      0.27      0.30     20261
weighted avg       1.00      0.81      0.90     20261

Confusion matrix at  20 % :
 [[16482  1122  2657]
 [    0     0     0]
 [    0     0     0]]
```

- **At 50% unlabelled data:**
  When considered 50651 records as labelled and 50651 records unlabelled data, model was 54 % accurate. Below are the results:



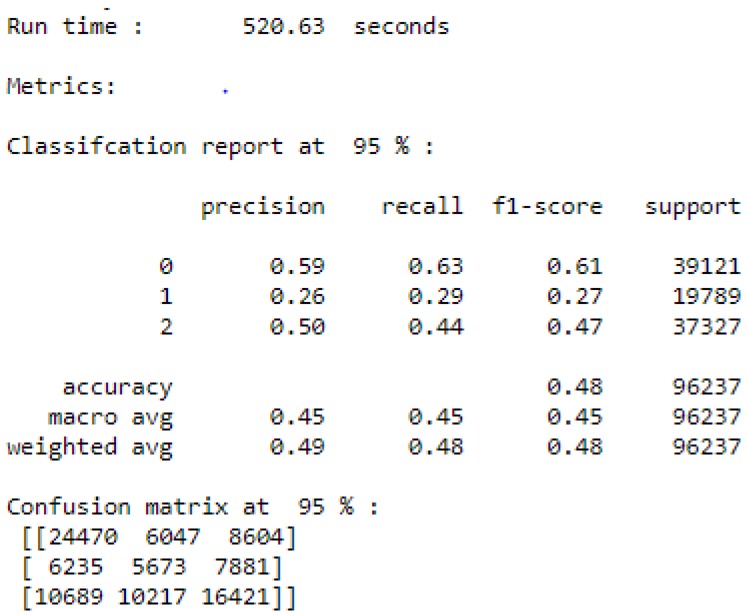
```
Run time :      363.44  seconds

Metrics:

Classifcation report at  50 % :

              precision    recall  f1-score   support

           0       0.94      0.55      0.69     34839
           1       0.14      0.38      0.20      4191
           2       0.36      0.58      0.45     11621

    accuracy                           0.54     50651
   macro avg       0.48      0.50      0.45     50651
weighted avg       0.74      0.54      0.60     50651

Confusion matrix at  50 % :
 [[19158  6127  9554]
 [  370  1612  2209]
 [  872  4054  6695]]
```

- **At 90% unlabelled data:**
  When considered 10130 records as labelled and 91172 records unlabelled data, model was 49 % accurate. Below are the results:



```
Run time :       397.12  seconds

Metrics:

Classifcation report at  90 % :

              precision    recall  f1-score   support

           0       0.65      0.58      0.61     38644
           1       0.25      0.31      0.28     18066
           2       0.49      0.49      0.49     34462

    accuracy                           0.49     91172
   macro avg       0.46      0.46      0.46     91172
weighted avg       0.51      0.49      0.50     91172

Confusion matrix at  90 % :
 [[22257  6658  9729]
 [ 4519  5617  7930]
 [ 7713  9870 16879]]
```

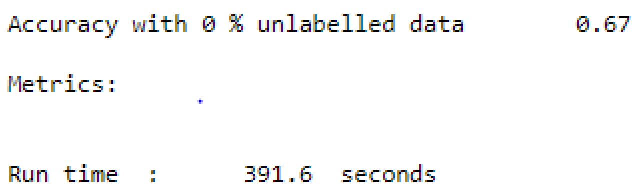
- **At 95% unlabelled data:**
  When considered 5065 records as labelled and 92367 records unlabelled data, model was 48 % a ccurate. Below are the results:



```
Run time :       520.63  seconds

Metrics:

Classifcation report at  95 % :

              precision    recall  f1-score   support

           0       0.59      0.63      0.61     39121
           1       0.26      0.29      0.27     19789
           2       0.50      0.44      0.47     37327

    accuracy                           0.48     96237
   macro avg       0.45      0.45      0.45     96237
weighted avg       0.49      0.48      0.48     96237

Confusion matrix at  95 % :
 [[24470  6047  8604]
 [ 6235  5673  7881]
 [10689 10217 16421]]
```

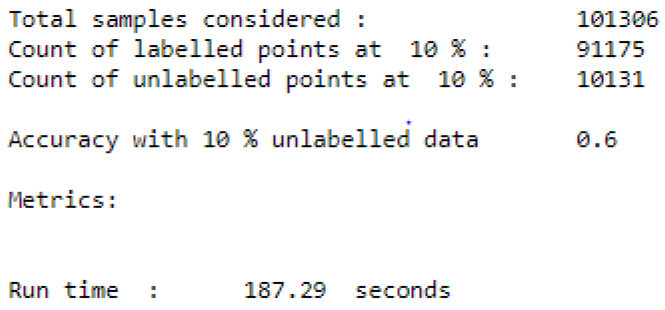
3. K-NN (Resampled data)

- **At 0% unlabelled data:**
  When considered 101250 records as labelled and 0 records unlabelled data, model was 67 % accu rate. Below are the results:



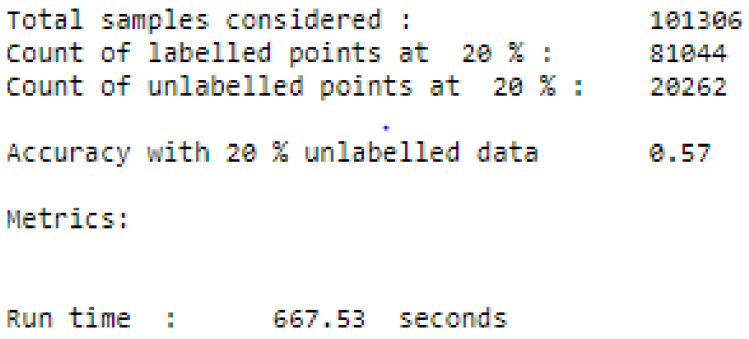
```
Accuracy with 0 % unlabelled data        0.67

Metrics:


Run time  :      391.6   seconds
```

- **At 10% unlabelled data:**
  When considered 91175 records as labelled and 10131 records unlabelled data, model was 60 % accurate. Below are the results:



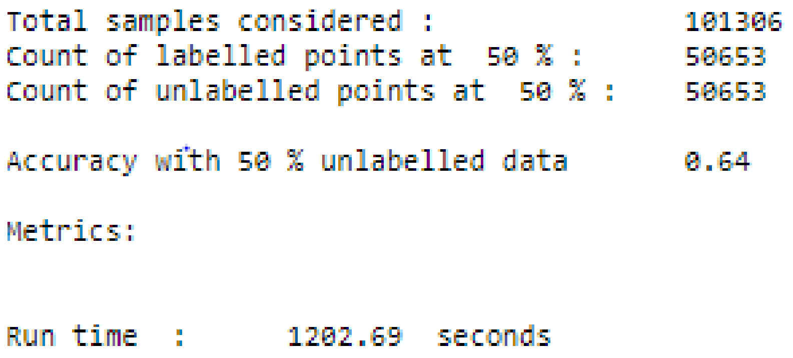
```
Total samples considered :                101306
Count of labelled points at  10 % :       91175
Count of unlabelled points at  10 % :     10131

Accuracy with 10 % unlabelled data        0.6

Metrics:


Run time  :      187.29   seconds
```

- **At 20% unlabelled data:**
  When considered 81044 records as labelled and 20262 records unlabelled data, model was 57 % accurate. Below are the results:



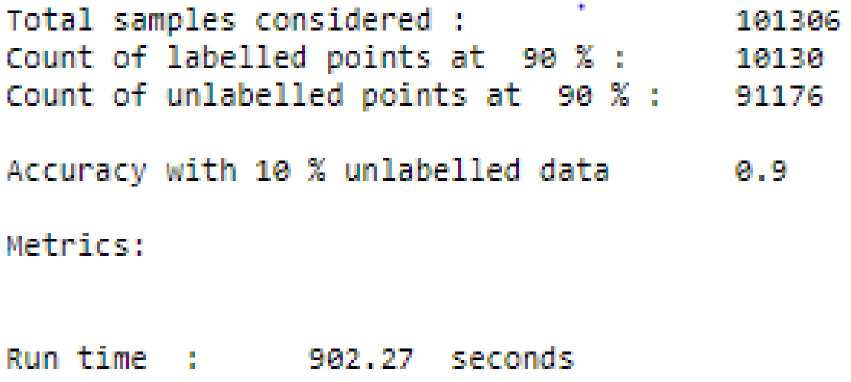
```
Total samples considered :                101306
Count of labelled points at  20 % :       81044
Count of unlabelled points at  20 % :     20262

Accuracy with 20 % unlabelled data        0.57

Metrics:


Run time  :      667.53   seconds
```

- **At 50% unlabelled data:**
  When considered 50653 records as labelled and 50653 records unlabelled data, model was 54 % accurate. Below are the results:



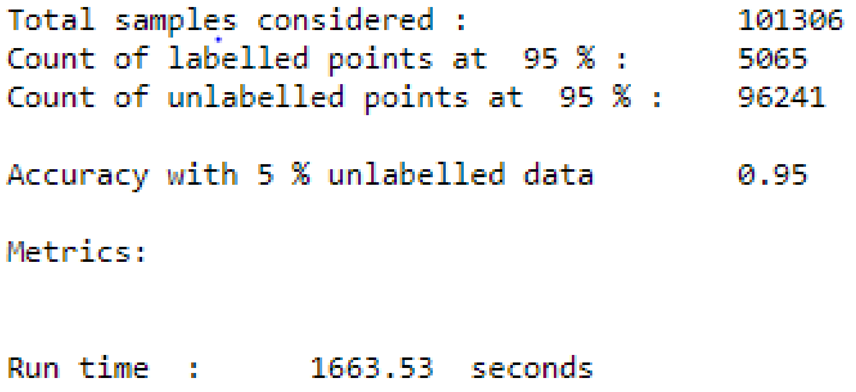
```
Total samples considered :             101306
Count of labelled points at  50 % :    50653
Count of unlabelled points at  50 % :  50653

Accuracy with 50 % unlabelled data     0.64

Metrics:


Run time  :      1202.69  seconds
```

- **At 90% unlabelled data:**
  When considered 10130 records as labelled and 91176 records unlabelled data, model was 9 % a ccurate. Below are the results:



```
Total samples considered :              101306
Count of labelled points at  90 % :     10130
Count of unlabelled points at  90 % :   91176

Accuracy with 10 % unlabelled data      0.9

Metrics:


Run time  :      902.27  seconds
```

- **At 95% unlabelled data:**
  When considered 5065 records as labelled and 92367 records unlabelled data, model was 48 % a ccurate. Below are the results:



```
Total samples considered :              101306
Count of labelled points at  95 % :     5065
Count of unlabelled points at  95 % :   96241

Accuracy with 5 % unlabelled data       0.95

Metrics:


Run time  :      1663.53  seconds
```

Conclusion:

Below are the key notes:

1. Readmission differed significantly in patients where Hba1c was checked in the setting of a primary diabetes diagnosis
2. For task 1, where the target variable is 'readmitted', extra tree classifier performed the best at 88% and the least performing algorithm was bernoulli's naïve bayes at 56% accuracy.
3. When applied for gradient boosting algorithm on task 1, the model was overfitting.
4. SVM model was a time-consuming algorithm and was 68% accurate.
5. When resampling was performed, accuracy of all the models increased by 10%.
6. When the parameters of the supervised algorithms are tuned, it resulted in improving the performance of the algo.
7. Bagging improved in the accuracy of the weak learner.
8. Label propagation and label spreading algorithms resulted almost the same accuracy for the task 1.
9. When there is a large amount of unlabelled data given to the model, it resulted in less accuracy.

Future work:

The dataset can be more tuned by applying more feature transformation techniques which can improve the accuracy of the model. Inductive Semi-supervised learning algorithms can be applied to identify the trade-offs among them.

References:

[1] https://www.hindawi.com/journals/bmri/2014/781670/ - Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records.

[2] https://scikitlearn.org/stable/modules/label_propagation.html# - Two semi-supervised techniques in Scikit Learn: