DIABETES HOSPITAL READMISSION PREDICTION

Project Report

By

GAURAV KHAMPARIYA

ANMOL ABHAY KALE

PRIYANKA B PATIL

ABHIJITH C

NITIN H S

Table of Contents

DIABETES HOSPITAL READMISSION PREDICTION

PROBLEM DESCRIPTION

Dataset Summary

The dataset was obtained from the UCI Machine Learning Repository. It is listed under the name Diabetes 130 – US Hospitals. According to the dataset description, the data has been prepared to analyse factors related to readmission as well as other outcomes pertaining to patients with diabetes. The dataset represents 10 years (1999-2008) of clinical care at 130 U.S. hospitals and integrated delivery networks. This dataset contains 101,766 unique inpatients encounters (instances) with 50 attributes, making the size of this dataset a total of 5,088,300 cells.

Encounters (Records)

As stated on the UCI's dataset information page, the dataset contains 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.

The domain of this Capstone Project falls under the purview of "Healthcare Analytics". Healthcare Analytics primarily involves the exploration of actionable insights from sets of patient data collected from four areas within healthcare:

- Claims and cost data.
- Pharmaceutical and R&D data
- Clinical data collected from electronic medical records (EHRs)
- Patient behaviour and sentiment data

According to a recent Research and Markets report1, healthcare analytics is poised to grow into a $34.27 billion industry by the end of 2022.

The capstone project will present the analysis of a large clinical database (III) that was undertaken to examine the historical patterns of diabetes care in patients with diabetes admitted to a US hospital and to indicate future directions which will lead to a reduction in hospital readmission rate and improvements in patient care.

OVERVIEW OF THE FINAL PROCESS

Features (Attributes)

The attributes represent patient and hospital outcomes. This data set mostly contains nominal attributes such as medical specialty and gender, but also includes a few ordinal attributes such as age and weight and continues attributes such as time(days) in hospital and number of medications.

The following table list each attribute, its description, and the percentage of missing information pertaining to each attribute.

Feature name	**Type**	**Description and values**	**% missing**
Encounter ID	Numeric	Unique identifier of an encounter	0%
Patient number	Numeric	Unique identifier of a patient	0%
Race	Nominal	Values: Caucasian, Asian, African American, Hispanic, and other	2%
Gender	Nominal	Values: male, female, and unknown/invalid	0%
Age	Ordinal	Grouped in 10-year intervals: [0, 10), [10, 20), ..., [90, 100)	0%
Weight	Numeric	Weight in pounds.	97%
Admission type	Nominal	Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available	0%
Discharge disposition	Nominal	Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available	0%
Admission source	Nominal	Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital	0%
Time in hospital	Numeric	Integer number of days between admission and discharge	0%
Payer code	Nominal	Integer identifier corresponding to 23 distinct values, for example, Blue Cross\Blue Shield, Medicare, and self-pay	52%
Medical specialty	Nominal	Integer identifier of a specialty of the admitting physician, corresponding to 84 distinct values, for example, cardiology, internal medicine, family\general practice, and surgeon	53%
Number of lab procedures	Numeric	Number of lab tests performed during the encounter	0%
Number of procedures	Numeric	Number of procedures (other than lab tests) performed during the encounter	0%
Number of medications	Numeric	Number of distinct generic names administered during the encounter	0%

Number of outpatient visits	Numeric	Number of outpatient visits of the patient in the year preceding the encounter	0%
Number of emergency visits	Numeric	Number of emergency visits of the patient in the year preceding the encounter	0%
Number of inpatient visits	Numeric	Number of inpatient visits of the patient in the year preceding the encounter	0%
Diagnosis 1	Nominal	The primary diagnosis (coded as first three digits of ICD9); 848 distinct values	0%
Diagnosis 2	Nominal	Secondary diagnosis (coded as first three digits of ICD9); 923 distinct values	0%
Diagnosis 3	Nominal	Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values	1%
Number of diagnoses	Numeric	Number of diagnoses entered to the system	0%
Glucose serum test result	Ordinal	Indicates the range of the result or if the test was not taken. Values: “>200,” “>300,” “normal,” and “none” if not measured	0%
A1c test result	Ordinal	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.	0%
Change of medications	Nominal	Indicates if there was a change in diabetic medications (either dosage or generic name). Values: “change” and “no change”	0%
Diabetes medications	Nominal	Indicates if there was any diabetic medication prescribed. Values: “yes” and “no”	0%
24 features for medications	Nominal	For the generic names: metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, sitagliptin, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, and metformin-pioglitazone, the feature indicates whether the drug was prescribed or there was a change in the dosage. Values: “up” if the dosage was increased during the encounter, “down” if the dosage was decreased, “steady” if the dosage did not change, and “no” if the drug was not prescribed	0%

Readmitted	Nominal	Days to inpatient readmission. Values: “30” if the patient was readmitted in more than 30 days, and “No” for no record of readmission.0	0%

Target Variable

The last attribute in the previous table is the class attribute, which in this case is Readmission.

The distribution of the class attribute is as follows:

- Encounters of patients who were not readmitted (No) to the hospital. There are 54, 864 of such encounters.
- Encounters of patients who were readmitted to the hospital after 30 days of discharge (>30). There are 35,545 of such encounters.
- Encounters of patients who were readmitted to the hospital within 30 days of discharge (<30). There are 11, 357 of such encounters.
- Since our aim is to predict the patients who are getting readmitted to the hospital within 30 days, we will consider <30 days as 1 and others i.e. no and >30 as 0

DATA PRE-PROCESSING

Data Cleaning Process

Data cleaning is commonly defined as the process of detecting and correcting corrupt or inaccurate records from a dataset, table, or database. Data quality is an important component in any data mining efforts. For this reason, many data scientists spend from 50% to 80% of their time preparing and cleaning their data before it can be mined for insights. There are four broad categories of data quality problems: missing data, abnormal data (outliers), departure from models, and goodness-of-fit. For our project, our team mainly dealt with missing data. Our team will also address the imbalance in the class variable using SMOTE.

Missing Values

Therefore, the first step in addressing missing values will be to encode them properly. As a general rule, variables with 50% or more missing values should be dropped from the analysis. The variable medical specialty comprises 49% of missing observations. In term of proportion, the whole column should be dropped. However, based on background understanding and recommendation from previous researches such variable is of prime importance when predicting readmission. Hence, the missing values were encoded as a new category labelled “Missing”. Moreover, the social economic status of the patient is a critical factor in predicting readmissions; therefore, variable such as “Payer code” should be preserved in the dataset. In addition, the list wise deletion was performed for variables with very few missing values as the dataset is large enough to maintain significant weight. For the rest of the variables with low to average missing rate, imputation was conducted in order to maintain as much data as possible for further modelling.

Irrelevant Data

The class attribute determines whether a patient is readmitted in the hospital within 30 days, over 30 days, or not readmitted at all. The attribute, discharge disposition, corresponds to 29 distinct values that indicate patients are discharged to home or another hospital, to hospice for terminally ill patients, or indicate that the patients have passed away.

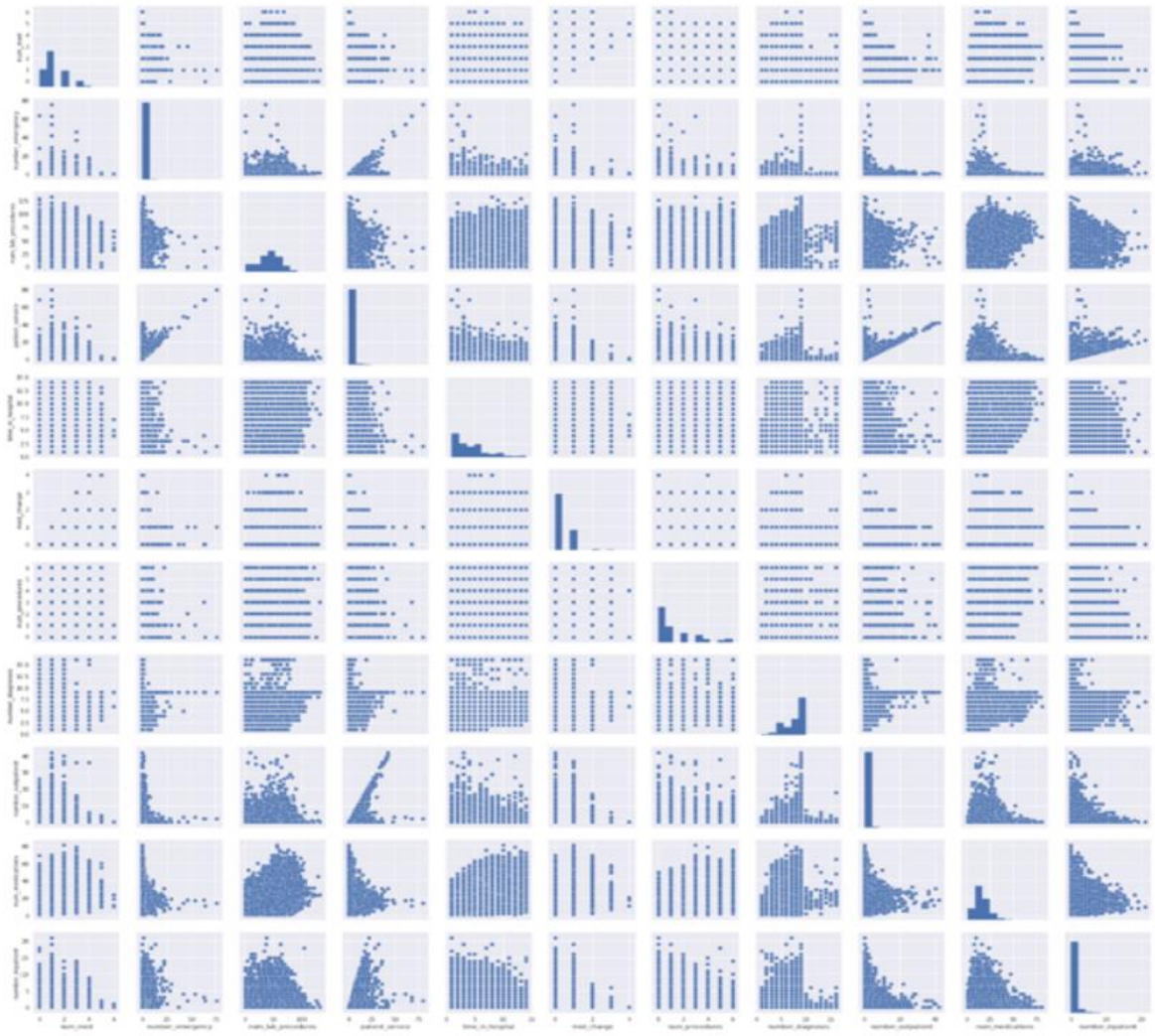
To correctly include only active (alive) patients and not in hospice, we removed records that had Discharge Disposition codes of 11, 13, 14, 19, 20, and 21. These discharge codes matched the instances of patients who were deceased or sent to hospital.

DATA EXPLORATION (EDA)

- **Relationship between variables**

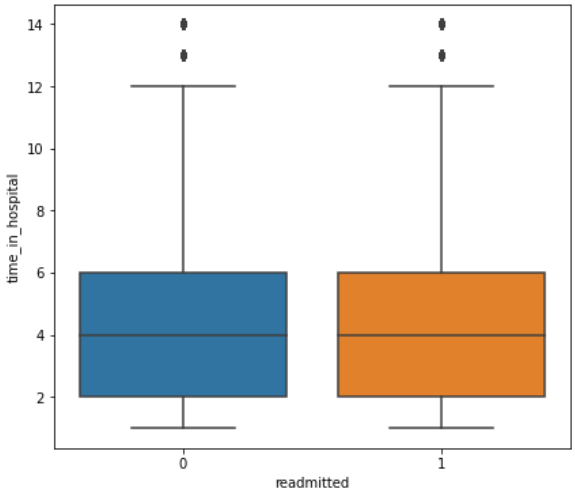
Prior to performing any analysis, we conducted exploratory analysis to preview the data type, attributes, and overall patterns of the data. We are interested in the class label

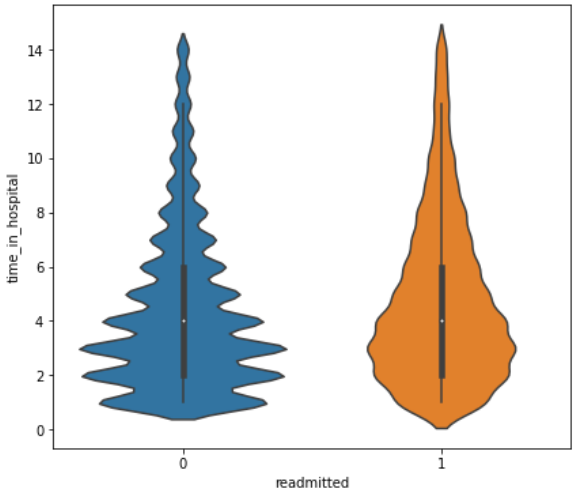
"Readmitted" so we checked the distribution of the readmitted, and several categorical variables. For the numerical variables, we used pair plot to demonstrate the relationship between numerical variables and their distributions



Bivariate Analysis

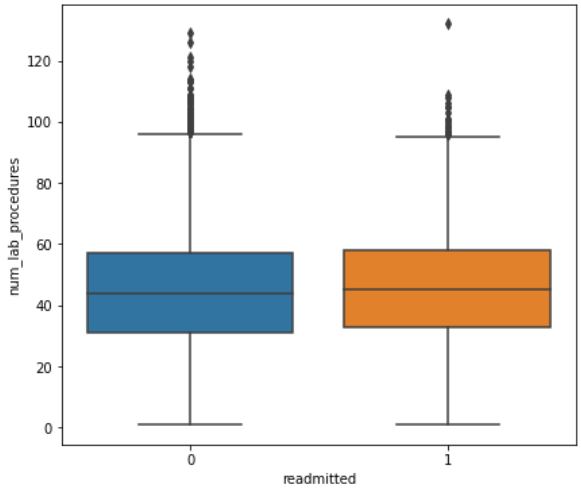
1- Box plot and Violin Plot between Readmitted and Time in hospital

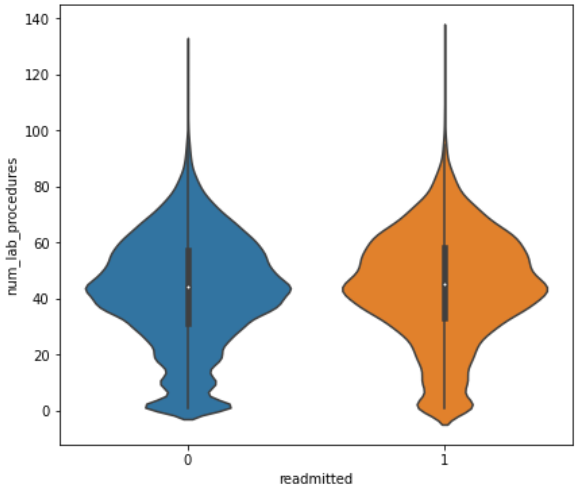




Above box plot shows that for time in hospital there is a similar distribution for people who are readmitted within 30 days and for people not being readmitted within 30days few outliers.

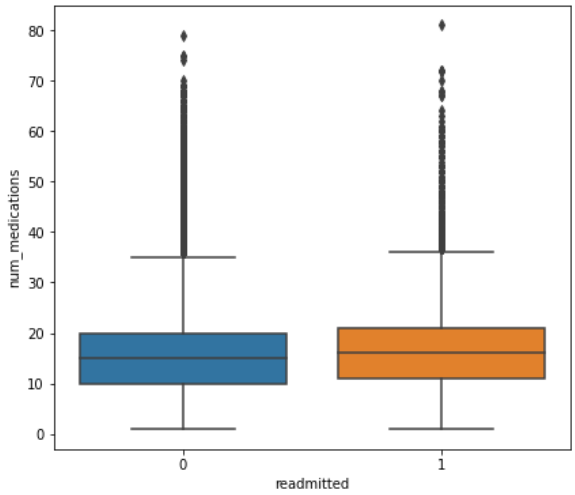
2- Box and Violin Plot between Readmitted and number of lab procedure

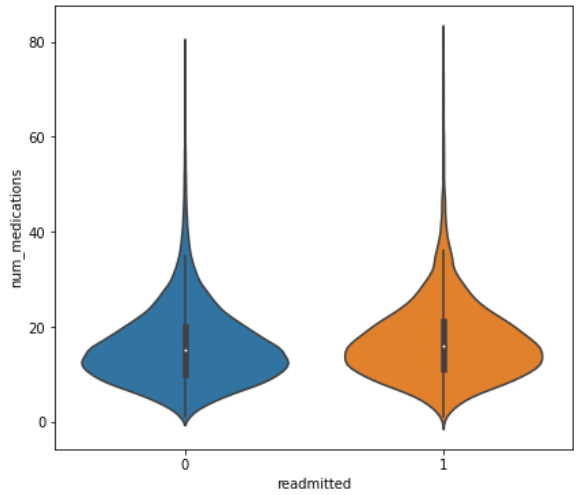




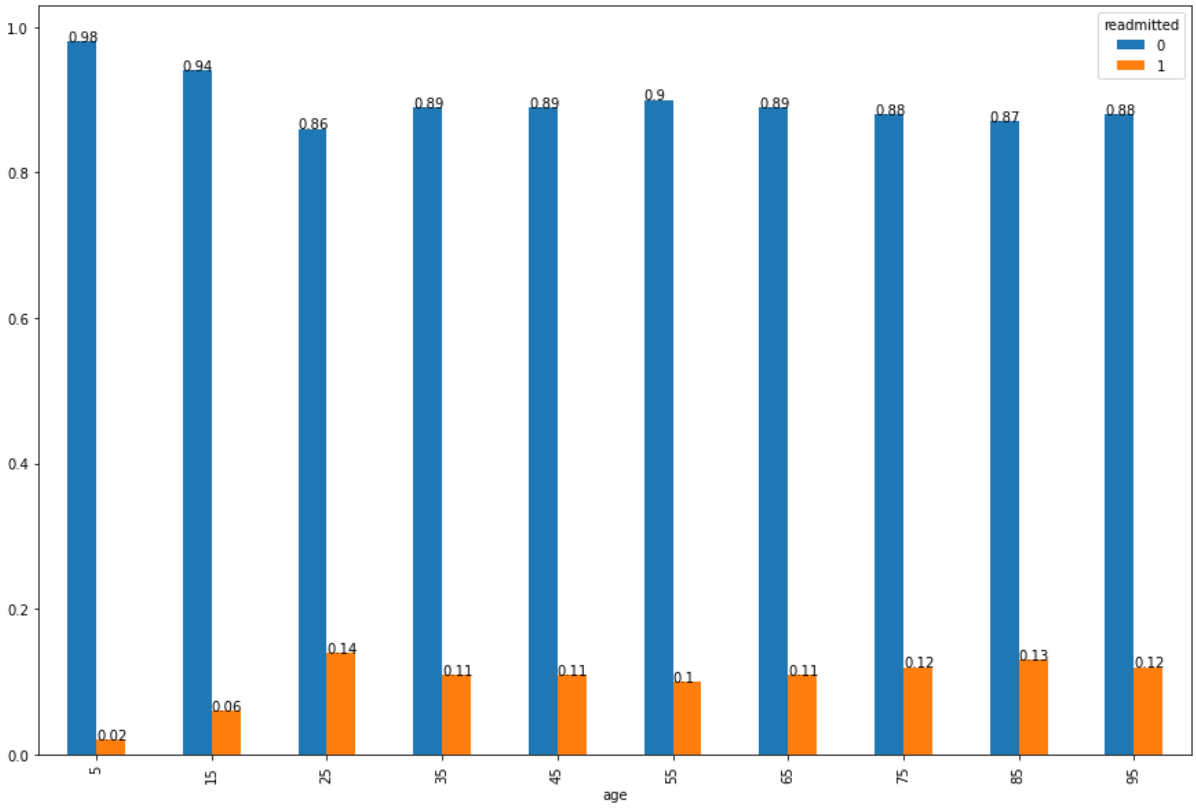
Just as we have seen in above boxplot, there is a similar distribution of number of lab procedures for both the values of readmitted. However, we can clearly see from the box plot that for patients who are not being readmitted within 30days, there are comparatively more outliers than patients who are being readmitted within 30 days.

3- Stacked Bar Graph between Race and Age With Readmitted.



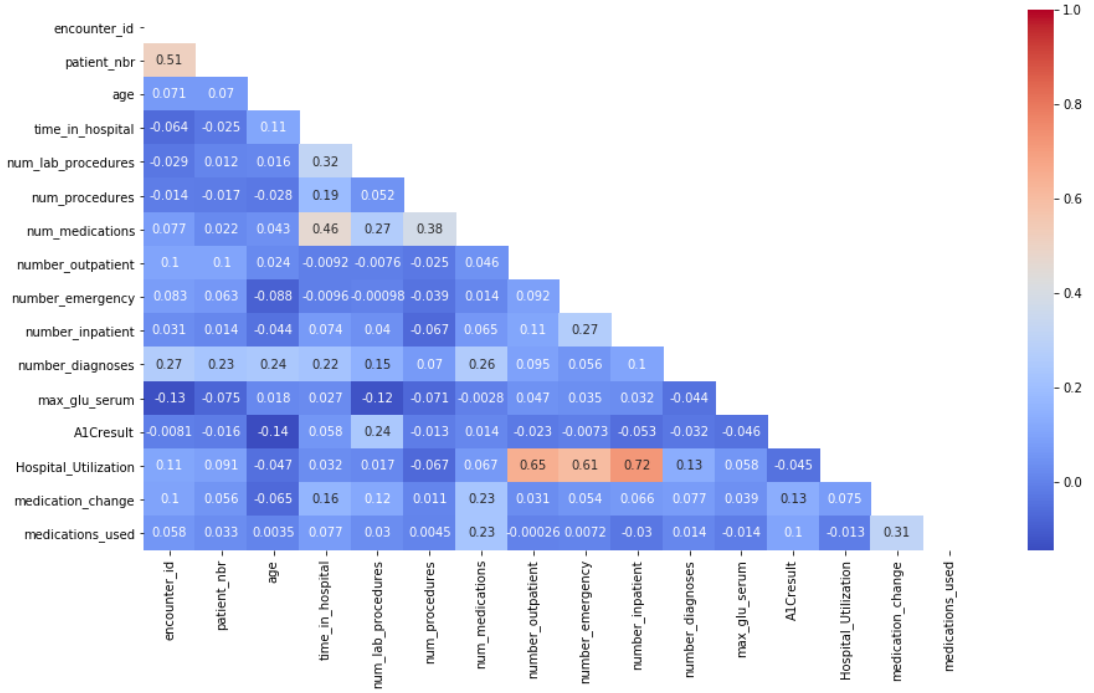


From the above crosstab we can see that the percentage of people not being readmitted is highest in Asian and Other race. While it is the lowest in Caucasian, followed by African American. Similarly, Caucasian and African American have the highest percentage of readmission within 30 days.



For age feature as expected people who belong to age category of within 40 years, there chances of being not readmitted is very high compared to the older group. Most interesting is that people who are being readmitted within 30 days is highest for the age group of 20-30. Quite surprising as they should have higher immunity when compared with other age groups.

Multivariate Analysis



From the heatmap we can see that most of the features are not correlated to each other with the exception of hospital utilization, number outpatient, number inpatient, number emergency. In case of high correlation between independent features, there exists a strong multicollinearity (one independent feature influences another independent feature) between independent features. After checking multicollinearity we have decide what action should be taken on these features.

Feature Engineering

From the dataset we can see that few features like time_in_hospital, num_medications have some interaction between them. So we will create new features from our dataset which have interaction between them.

```
interaction = [('num_medications','time_in_hospital'),
('num_medications','num_procedures'),
('time_in_hospital','num_lab_procedures'),
('num_medications','num_lab_procedures'),
('num_medications','number_diagnoses'),
('age','number_diagnoses'),
('change','num_medications'),
('number_diagnoses','time_in_hospital')]
```

```
for i in interaction:
    name = i[0] + '|' + i[1]
    df[name] = df[i[0]] * df[i[1]]
```

For categorical data which are ordinal in nature we have replaced them using label encoder for nominal categorical data we have performed one hot encoding, which will replace categories with numbers.

```
df['race'] = df['race'].replace({'Caucasian':1,'AfricanAmerican':2,'Hispanic':3,'Other':4, 'Asian':5})
df['gender'] = df['gender'].replace({'Male':1, 'Female':2})
```

Using Selectkbest to select the best number of features for our model. We will select features whose chi2_scores are above 10 from a total of 29 features to proceed with our model building which will increase our accuracy and saves our computational time.

```
df_scores = SelectKBest(score_func=chi2, k=15)
fit = df_scores.fit(X,y)
feature_Scores = pd.DataFrame(fit.scores_, index = X.columns, columns = ['chi2 score'])
feature_Scores.sort_values(by = 'chi2 score', ascending = False)
```

	chi2 score
time_in_hospital|num_lab_procedures	33982.074568
age|number_diagnoses	15832.857200
number_diagnoses|time_in_hospital	5201.297745
num_medications|num_lab_procedures	4566.620230
num_medications|time_in_hospital	1761.104135
num_lab_procedures	663.940871
age	604.928388
num_medications|number_diagnoses	569.703523
time_in_hospital	439.304943
Hospital_Utilization	310.131414
discharge_disposition_id	101.608333
number_diagnoses	80.689225
change|num_medications	44.773561
medication_change	20.178467
num_medications	14.874242
diabetesMed	12.821971
num_medications|num_procedures	11.796369

MODEL BUILDING

****Step-by-step walk through of the solution**

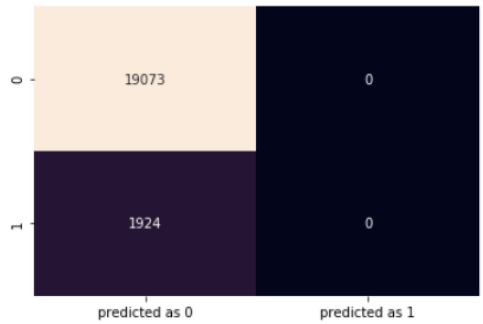
Train Test Splitting of data into independent and dependent features

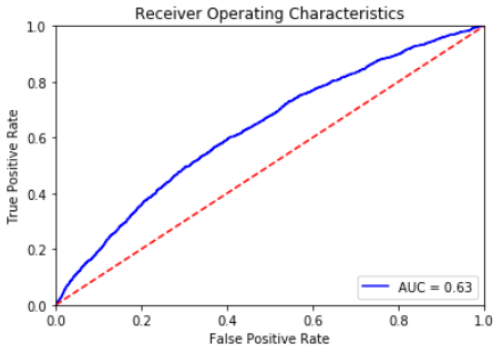
Performing Power Transforming X_train and X_test numerical features to bring them into one scale i.e which will normalize and standardize the numerical features.

Building a base model

Logistic Regression - Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

```
Accuracy Score : 0.9083678620755345
Precision Score : 0.0
Recall Score : 0.0
F1 Score : 0.0
AUC Score : 0.6295351114598219
```





Model is doing a good job predicting the majority class and not at all able to predict the minority class. We have 10% records corresponding to the readmission category. As with most models, algorithms

work best when the number of samples in each class are about equal. Standard classification algorithms have a bias towards classes which have higher number instances. They tend to predict only the majority class data. Thus, there is a high probability of misclassification of the minority class as compared to the majority class.

Treating the Imbalance Data

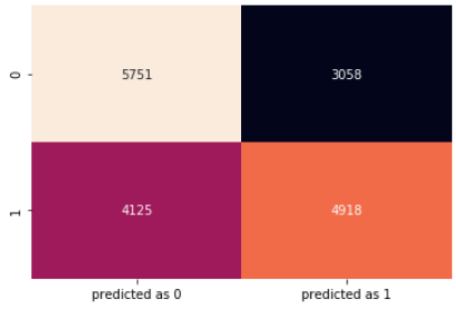
There are number of ways for treating imbalanced dataset, we will be using SMOTE technique to treat the imbalanced dataset. But before doing that we will split the dataset into train and test data and then implement SMOTE

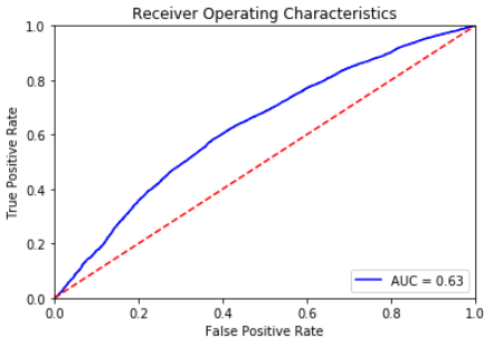
Applying Various Algorithms

Applying various algorithms to our X_train and Y_train ensures the difference in there Accuracy and there Precision Score , by comparing all the models we can the best model which have high accuracy and precision and f1 score .

Logistic Regression - Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

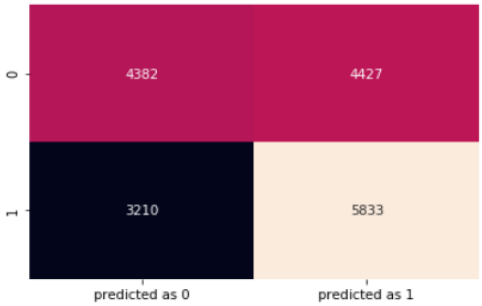
```
Accuracy Score : 0.5976361192023303
Precision Score : 0.6165997993981945
Recall Score : 0.5438460687824836
F1 Score : 0.5779422997825958
AUC Score : 0.6293524736640332
```

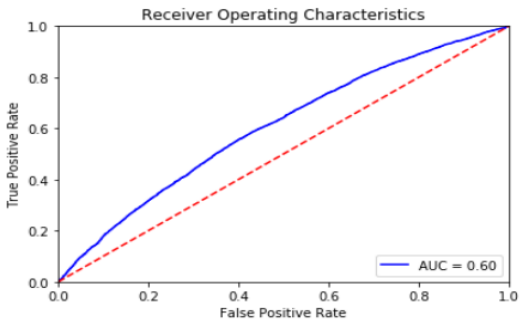




Naive Bayes Classifier - A **Naive Bayes classifier** is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

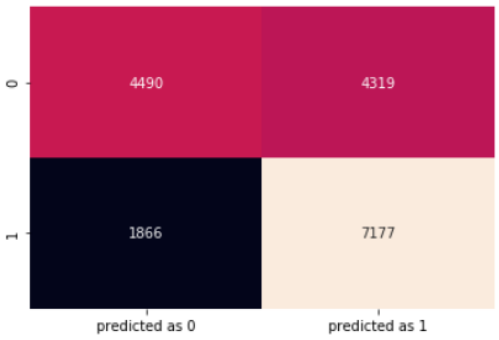
```
Accuracy Score : 0.5722047949809546
Precision Score : 0.5685185185185185
Recall Score : 0.6450293044343691
F1 Score : 0.6043620162669014
AUC Score : 0.6043052311952579
```

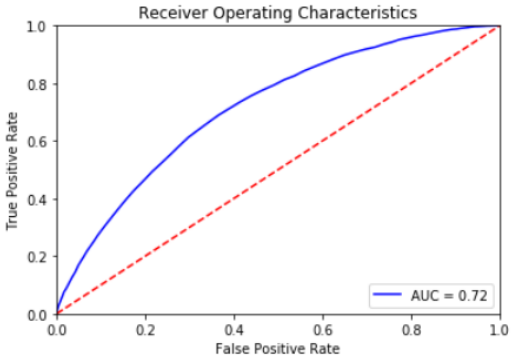




KNearestNeighbors Classifier - K nearest neighbours is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). **KNN works** by finding the distances between a query and all the examples in the data, selecting the specified number examples (**K**) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

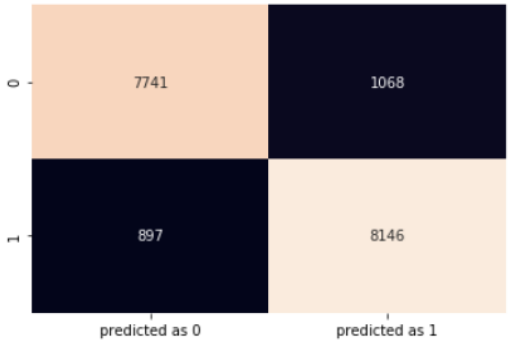
```
Accuracy Score : 0.6535402195832399
Precision Score : 0.6243041057759221
Recall Score : 0.7936525489328763
F1 Score : 0.6988655728126978
AUC Score : 0.7165186055543933
```

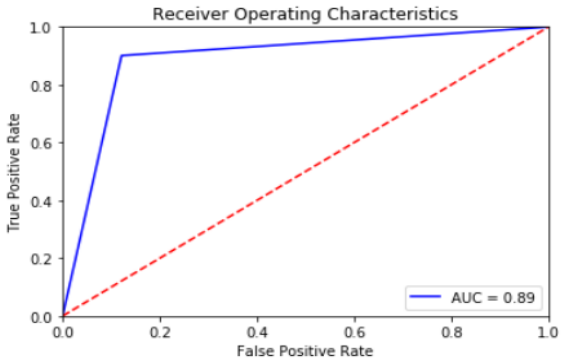




Decision Tree Classifier - Decision tree learning is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

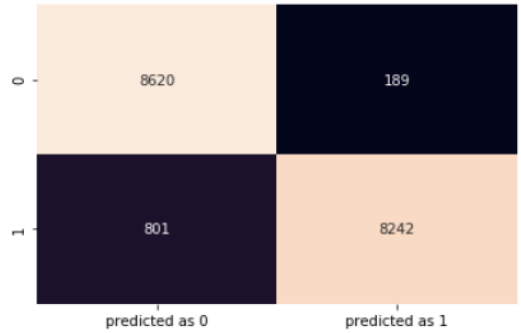
```
Accuracy Score : 0.8899282993502129
Precision Score : 0.8840894291295854
Recall Score : 0.900807254229791
F1 Score : 0.8923700498438956
AUC Score : 0.8898584614593559
```

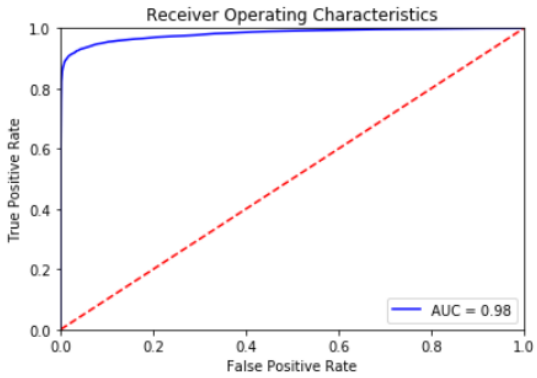




Random Forest - The **random forest** is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated **forest** of trees whose prediction by committee is more accurate than that of any individual tree.

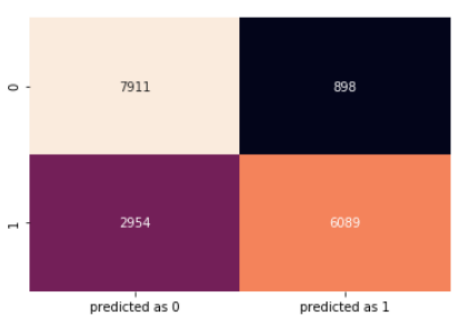
```
Accuracy Score : 0.9445440286802599
Precision Score : 0.9775827303997153
Recall Score : 0.9114232002653987
F1 Score : 0.9433443973904087
AUC Score : 0.9817032275017257
```

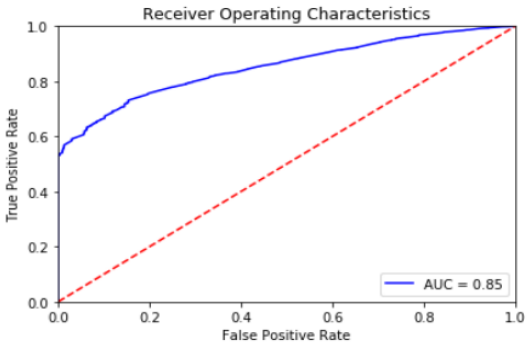




AdaBoost Classifier - AdaBoost or Adaptive Boosting is one of the ensemble boosting classifier proposed by Yoav Freund and Robert Schapire in 1996.It combines multiple weak classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get high accuracy strong classifier. The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations. Any machine learning algorithm can be used as base classifier if it accepts weights on the training set.

```
Accuracy Score : 0.7842258570468295
Precision Score : 0.8714755975382854
Recall Score : 0.6733384938626562
F1 Score : 0.7597005614472863
AUC Score : 0.8527235831047352
```

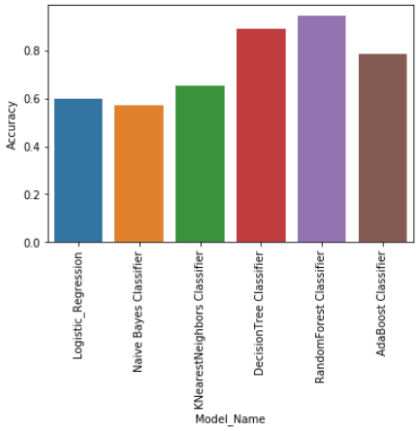


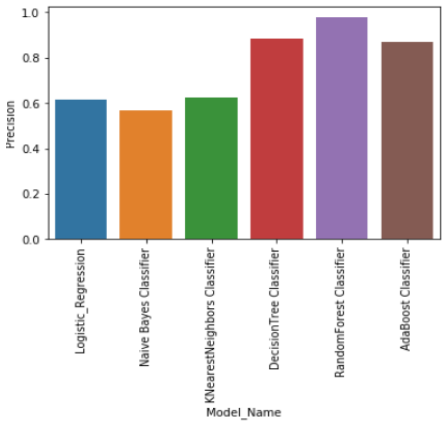


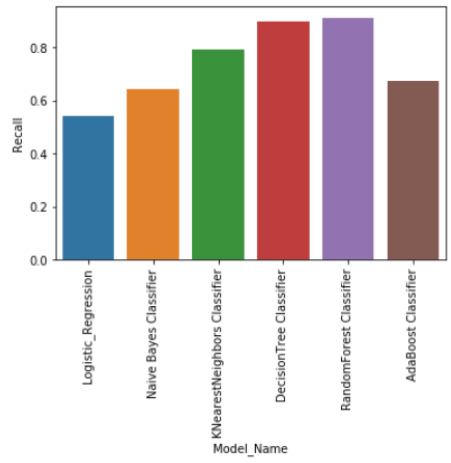
MODEL EVALUATION

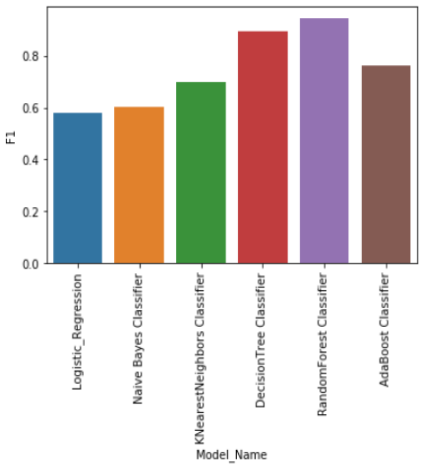
We will select the best algorithm out of the base models by doing a comparative study of the algorithms against all the performance metrics.

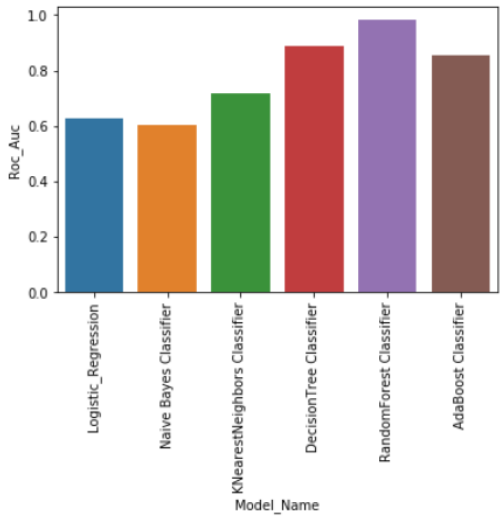
Model_Name	Accuracy	Precision	Recall	F1	Roc_Auc
Logistic_Regression	0.597636	0.616600	0.543846	0.577942	0.629352
Naive Bayes Classifier	0.572205	0.568519	0.645029	0.604362	0.604305
KNearestNeighbors Classifier	0.653540	0.624304	0.793653	0.698866	0.716519
DecisionTree Classifier	0.889928	0.884089	0.900807	0.892370	0.889858
RandomForest Classifier	0.944544	0.977583	0.911423	0.943344	0.981703
AdaBoost Classifier	0.784226	0.871476	0.673338	0.759701	0.852724











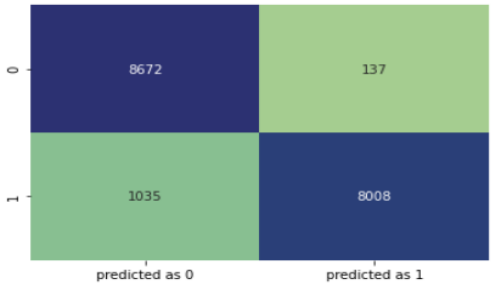
The objective is to get good accuracy while predicting. From the metrics we can infer that RandomForest Classifier is performing very good when compared with other models.

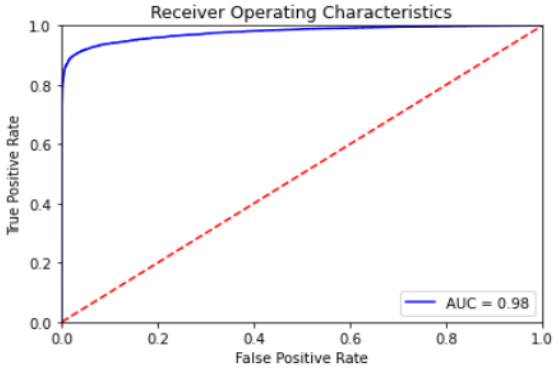
Hyperparameter Tuning RandomForest model

Improving the model accuracy through hyperparameter tuning (RandomizedSearchCV) and see if we can improve the accuracy of the model.

- Finding Optimal hyperparameters
- Result:

```
Accuracy Score : 0.9343490925386512
Precision Score : 0.9831798649478207
Recall Score : 0.885546831803605
F1 Score : 0.9318128927158483
AUC Score : 0.9767723832854336
```





Comparison to benchmark

Current health care providers in US use LACE (Length of Stay, Accuity, Comorbidities) to identify high risk patients. LACE is a logistic regression model that makes use of a small set of features. LACE itself was derived from a set of 4812 patients, and validated on 1,000,000 patients using patient records from 2004 to 2008 (van Walraven et al., 2010. The LACE index may not accurately predict unplanned readmissions within 30 days from hospital discharge in CHF patients. The LACE high risk index may have utility as a screening tool to predict high risk ED revisits after hospital discharge.

Implications

- Our solution is instrumental in predicting the patient readmission to the hospital due to diabetes in US healthcare setup specifically.
- The hospital management can better scrutinize these important features for a patient record and bring down the overall readmission rates for the hospital.
- Provider/Hospital getting the direct benefit of understanding the repeating patients and they should be able to arrange the medical infrastructure and facilities accordingly.
-

Limitations

- Databases of clinical data can present difficulties related to missing values, incomplete or inconsistent records, high dimensionality, and complexity of features.
- Model is designed for major diagnostic codes for ICD 9 diagnostic codes as given in the datasets. Need to be returned for ICD 10 diagnosis codes.
- Analyzing external data is more challenging than analysis of data collected during a carefully designed study, as features that may be important may simply not be available in an external dataset.
- Models only focus on the readmission which assumes that they are admitted to the same facility. In real world, data from all the sources should have been consolidated to make it accurate.
- Readmission is an important yet somewhat arbitrary measure which is influenced by a potentially infinite number of factors related to a patient's health and care received during a hospital admission.

Closing Reflections

There is huge learning in terms of understanding the domain and how important is the understanding the dataset is imperative. As the lack of understanding of the dataset can lead to catastrophic results.

- Significant amount of domain knowledge is required to analyse each feature in the dataset.
- We need to carefully choose the technique and they are prone to outliers.
- Handling categorical features is to be learned from experience.
- In feature engineering we have implemented basic technique but there is huge scope to venture with advanced technique like polynomial kernel, RBF kernel, mathematical transformation on the features.