



**Project Summary**

|  |  |
| --- | --- |
| Batch details | PG-DSE June2021 |
| Team members | Vipul Inaparty  Nihar Penchala  Naveen Kumar Chukka |
| Domain of Project | Health Care |
| Proposed project title | Prediction on Hospital Readmission |
| Group Number | 3 |
| Team Leader | Vipul Inaparty |
| Mentor Name | Mr. Animesh Tiwari |

Date:28-10-2021

Signature of the Mentor Signature of the Team Leader

Contents

[Business Objective/ Understanding 3](#_heading=h.gjdgxs)

[Data Description and Preprocessing 5](#_heading=h.30j0zll)

[Data Preparation 7](#_heading=h.1fob9te)

[Exploratory Data Analysis & Business Insights 8](#_heading=h.3znysh7)

[Model](#_heading=h.2et92p0) Building [19](#_heading=h.2et92p0)

Conclusion 37

[REFERENCES:](#_heading=h.tyjcwt) 38

## **INTRODUCTION**

# **Overview :**

Diabetes is a long-term condition characterized by hyperglycemia when the pancreas is unable to produce enough insulin or when the body is not able to use the insulin effectively to regulate blood sugar level. The former is known as type 1 diabetes mellitus (T1DM) and the latter as type 2 diabetes mellitus (T2DM). With advances in diagnosis and treatment, lifespan for patients with diabetes mellitus (DM), which commonly includes both types of diabetes, is projected to be longer. Increased lifespan and the high prevalence of obesity worldwide have quadrupled the number of adults living with DM from 108 million in 1980 to 422 million in 2014. Globally, DM accounts for 1.9% of total disability-adjusted life years and approximately 30% of hospitalized adult patients with DM had two or more readmissions within the next calendar year. Complications from diabetes are a serious threat to healthcare systems and also one of the top 10 causes of public hospital readmissions worldwide. In 2017, the hospitalization cost of patients with DM in the USA was $123 billion. Based on a 20% readmission rate, it was estimated that $24.6 billion would be attributed to 30-day readmission. Patients with DM represent one-fifth of the overall 30-day hospital readmissions although some may be preventable through better continuity of care.

Systematic reviews (SR) of DM have mainly focused on the relationship between glycemic control and surgical outcomes and the economic cost to the health system. To our knowledge, no SR with meta-analysis (MA) has been conducted to evaluate the effect of patient characteristics on hospital readmission among adult patients with DM. Current literature suggests conflicting results in common risk factors such as gender with some studies showing a significant effect while others demonstrating no evidence of relationship. Patient characteristics such as gender, age, race, and comorbidities may affect the outcomes of self-management. Thus, the US National Standards for Diabetes Self-Management Education and Support Task Force emphasizes their importance in self-management education (SME), to achieve better control of glycated hemoglobin (HbA1c) levels and reduce macrovascular complications of DM.

The primary objective of this SR with MA is to synthesize evidence concerning the association between 30-day unplanned hospital readmissions and patient characteristics (namely gender, age and race), affordability of medical insurance (as a proxy measurement of socioeconomic status (SES)), comorbidities, diabetes-related medications and inpatient factors such as length of stay (LOS) of index admission in adult patients .



# **Business Objective/ Understanding**

1. **Business Problem Understanding**

Decreasing hospital readmission rates through the program came at a price for some hospitals. CMS penalized over 2,500 hospitals by more than $564 million in 2017 for excessive 30-day hospital readmission rates. We want to build a model which can predict factors affecting readmission rate.

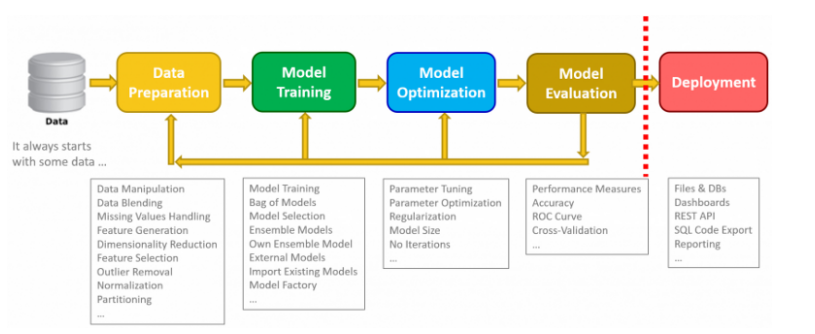
Among US hospitals in the Hospital Readmissions Reduction Program (HRRP), the authors calculated daily readmission rates for elderly Medicare fee-for-service beneficiaries through day 60 post-discharge following a hospitalization for acute myocardial infarction (AMI), heart failure (HF), and pneumonia. The time period examined was from July 2012 to June 2019. The primary outcomes were hospital readmissions and mortality.

Under the hospital readmission program established by the patient protection and affordable care Act, Centers for Medicare and Medicaid Services will decrease payments to hospitals that have greater than 30 days admission rate.

**Business Objective**

To obtain a measurement of patients with diabetes to predict readmission rates which may prove valuable in the development of strategies to reduce readmission rates and costs for the care of hospital.

**METHODOLOGY :**



# 

# **Data Description and Preprocessing:**

# As our dataset contains over 50+ variables, our dataset includes variety of Diagnosis results and test results.

# 

|  |  |
| --- | --- |
| **Variable Name** | **Variable 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: [0, 10), [10, 20), ..., [90, 100) |
| Weight | Weight in pounds. |
| Medical specialty | values, for example, cardiology, internal medicine, family\general practice, and surgeon |
| Number of procedures | Number of procedures (other than lab tests) performed during the encounter |
| 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. |
| 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 |
| Number of diagnoses | Number of diagnoses entered to the system |
| 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 |
| Readmitted | Days to inpatient readmission. Values: “30” if the patient was readmitted in more than 30 days, and “No” for no record of readmission. |

**Preprocessing:**

* Null values are observed in three columns, namely weight**,** race, payer\_code, medical\_specialty, diag\_1,diag\_2,diag\_3.
* Dropped all the null values in columns as the count was comparatively small.

# **Data Preparation**

The data that is used in this project originally comes from the UCI machine learning repository. The data consists of over 100000 hospital admissions from patients with diabetes from 130 US hospitals between 1999–2008.

Link-<https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008>

The most important column here is readmitted, which tells us if a patient was hospitalized within 30 days, greater than 30 days or not readmitted.

Another column that is important is discharge\_disposition\_id, which tells us where the patient went after the hospitalization. If we look at the IDs\_mapping.csv provided by UCI we can see that 11,13,14,19,20,21 are related to death or hospice. We should remove these samples from the predictive model since they cannot be readmitted.

From further analysis of the columns, we see there are a mix of categorical (non-numeric) and numerical data. A few things to point out are as follows,

* encounter\_id and patient\_nbr: these are just identifiers and not useful variables.
* age and weight: are categorical in this data set
* admission\_type\_id, discharge\_disposition\_id, admission\_source\_id: are numerical here, but are IDs. They should be considered categorical.
* examide and citoglipton only have 1 value, so we will not use these variables
* diag1, diag2, diag3 — are categorical and have a lot of values. We will not be using these as part of this project, but we can group these ICD codes to reduce the dimension. We will use number\_diagnoses to capture some of this information.
* medical\_speciality — has many categorical variables, so we will consider this when making features.

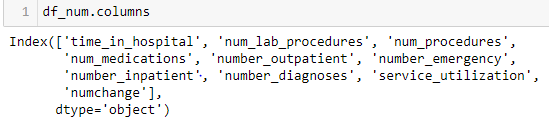
**Feature Engineering:**

we will be creating features for our predictive model. For each section, we will be adding new variables to the data frame and then keep track of which columns of the data frame we want to use as part of the predictive model features. We will be breaking down this section into numerical features, categorical features and extra features.

****

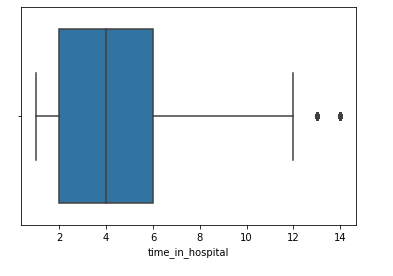
**Numerical Features**:

The easiest type of features to use is numerical features. These features do not need any modification.

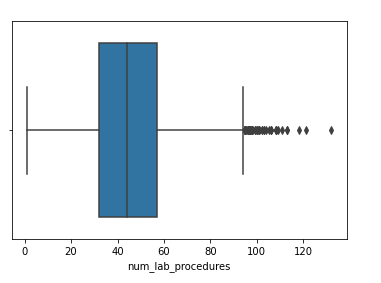


**Numerical Columns Description:**

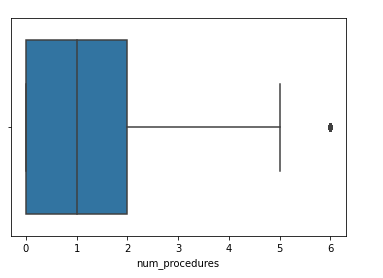
* **patient\_nbr:** Collapsing of Multiple Encounters for same patient Some patients in the dataset had more than one encounter. We could not count them as independent encounters because that bias the results towards those patients who had multiple encounters. Thus we tried multiple techniques to collapse and consolidate multiple encounters for same patient such as: Considering more than 2 readmissions across multiple encounters as readmission for collapsed record. Considering average stay at hospital across multiple encounters. Considering the percentage of the medication changes across multiple encounters Considering the total number of the encounters to replace the encounter unique ID Considering the combination of diagnoses across multiple encounters as a list However, taking the features such as “diagnosis”, for instance, we did not find it not meaningful to combine multiple categorical values into an array for building data model. We then considered first encounter and last encounter separately as possible representations of multiple encounters. However, last encounters gave extremely imbalanced data for readmissions (96/4 Readmissions vs No Readmissions) and thus, we decided to use first encounters of patients with multiple encounters. This resulted in dataset being reduced to about 70,000 encounters:
* **time\_in\_hospital**: Number of days between admission and discharge for the patient.



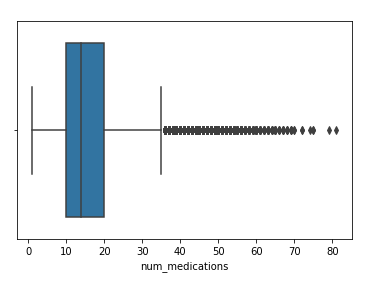
* **num\_lab\_procedures**:  Number of lab procedures performed in the current encounter



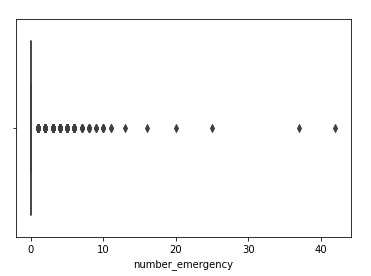
* **num\_procedures**: Number of non-lab procedures performed in the current encounter



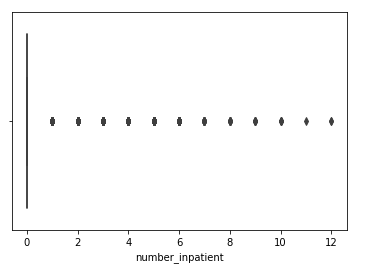
* **num\_medications**: Number of distinct medications performed in the current encounter.



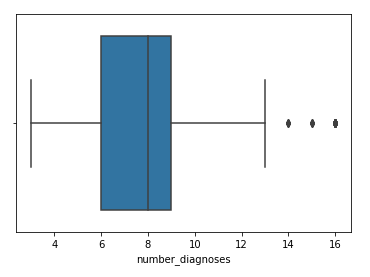
* **number\_emergency:** Number of emergency visits by the patient in the year prior to the current encounter



* **number\_inpatient**:  Number of inpatient visits by the patient in the year prior to the current encounter



* **number\_diagnoses**: Total number of diagnoses entered for the patient.



**Categorical Features:**

The next type of features we want to create are categorical variables. Categorical variables are non-numeric data such as race and gender. To turn these non-numerical data into variables, the simplest thing is to use a technique called one-hot encoding.

# 

# **Categorical Columns Description:**

# **race, gender:** Basic demographic information associated with each patient

# 

# 

# **admission\_source\_id and admission\_type\_id**: identify who referred the patient to the hospital (physical referral, emergency room, transfer from a hospital, etc.) and what type of admission this was (emergency, urgent, elective, etc.

# 

# 

# **discharge\_disposition\_id:** identifies where the patient was discharged to after treatment (discharged to home, expired, etc.)

# 

# **max\_glu\_serum**: Indicates the results of the glucose serum test

# 

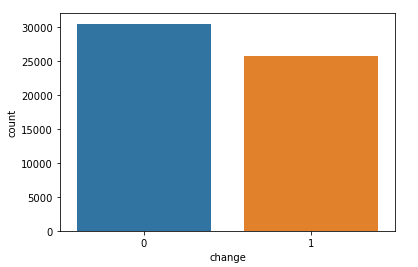
# **A1cresult**: Indicates results of the A1c test

# 

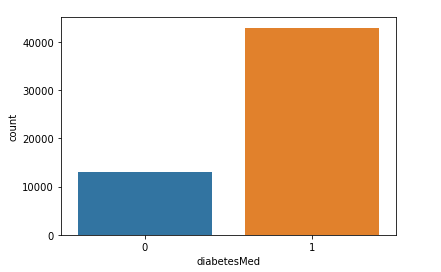
# **metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, metformin-pioglitazone**: Features for medication Indicates whether the dosage for the medicines was changed in any manner during the encounter

# The dataset contains 23 features for 23 drugs (or combos) which indicate for each of these, whether a change in that medication was made or not during the current hospital stay of patient. Medication change for diabetics upon admission has been shown by previous research to be associated with lower readmission rates. We decided to count how many changes were made in total for each patient, and declared that a new feature. The reasoning here was to both simplify the model and possibly discover a relationship with number of changes regardless of which drug was changed

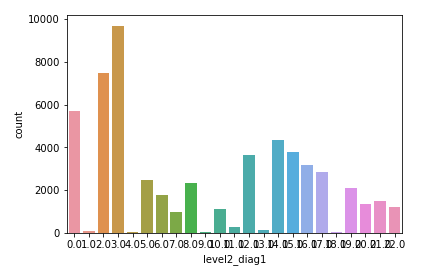
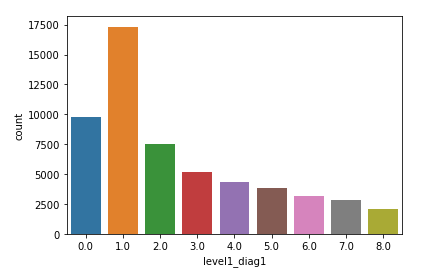
* **change**: Indicates if there was a change in diabetic medications

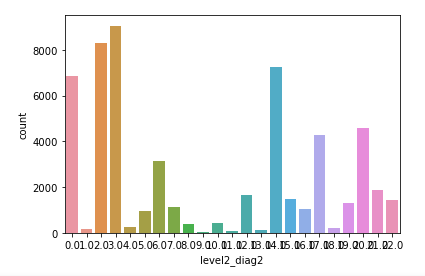
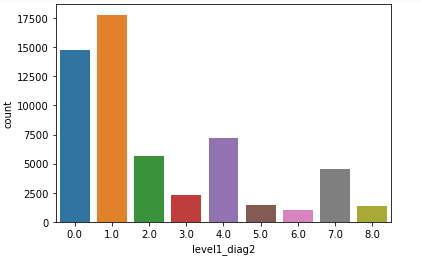


* **diabetesMed:** Indicates if any diabetes medication was prescribed



* Categorization of diagnoses: The dataset contained up to three diagnoses for a given patient (primary, secondary and additional). However, each of these had 700–900 unique ICD codes and it is extremely difficult to include them in the model and interpret meaningfully. Therefore, we collapsed these diagnosis codes into 9 disease categories in an almost similar fashion to that done in the original publication using this dataset. These 9 categories include Circulatory, Respiratory, Digestive, Diabetes, Injury, Musculoskeletal, Genitourinary, Neoplasms, and Others. Although we did this for primary, secondary and additional diagnoses, we eventually decided to use only the primary diagnosis in our model. Doing this in python was slightly cumbersome because, well, we are mapping the disease codes to certain category names. we have categories into 3 diag’ each into 2 levels





# 

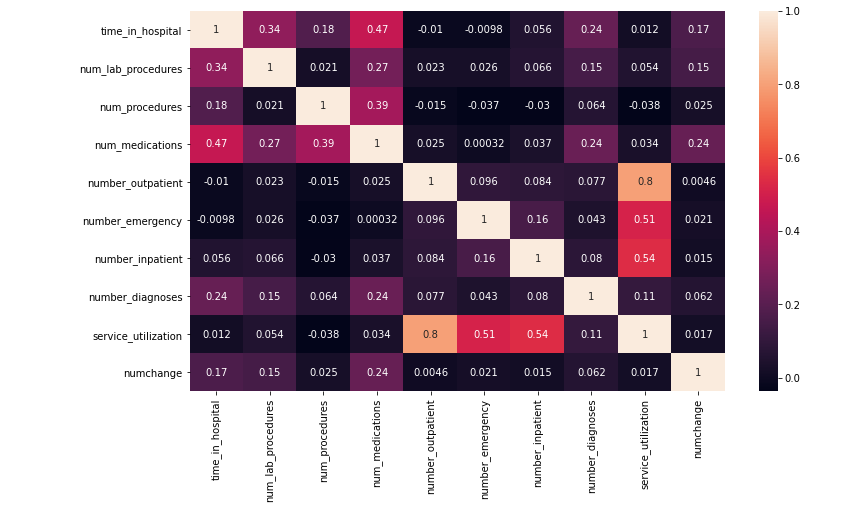
# Age\_group : We have segmented age into buckets for drawing inference.

# 

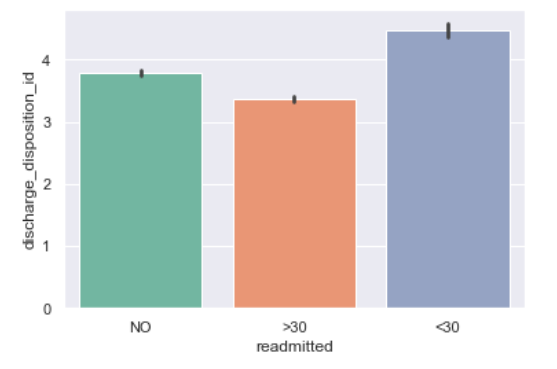
# **Exploratory Data Analysis & Business Insights:**

Correlation between Numerical columns:

From the below correlation matrix, we don’t see much collinearity within columns.The highest collinearity is seen in num\_medications.



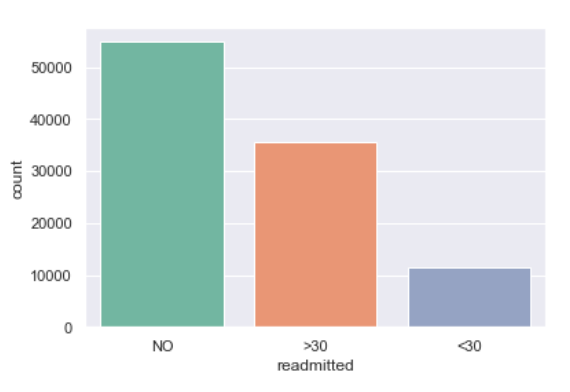
**discharge\_disposition\_id vs readmitted:**



This tells us where the patient went after the hospitalization. We can observe that less than 30 days of readmission is proportional to discharge id.

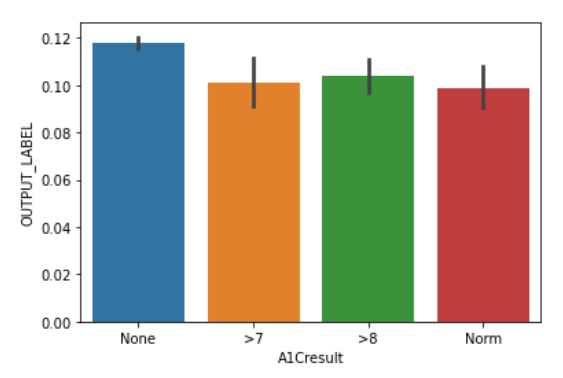
**Readmitted (Target)**

This tells us if a patient was hospitalized within 30 days, greater than 30 days or not readmitted. We can see that in our data set the ratio of patients who have not been readmitted is greater than patients who got readmitted in 30 days and ratio of patients who have been readmitted in greater than 30 days is more than patients who have been readmitted in less than 30 days.



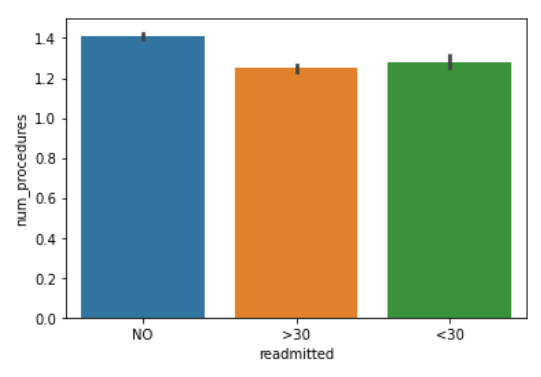
**AIC result vs Readmission:**

AIC 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.



**Number of Procedures vs Readmitted:**

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



We observe that number of procedures doesn’t have any significant effort on readmission of a patient.

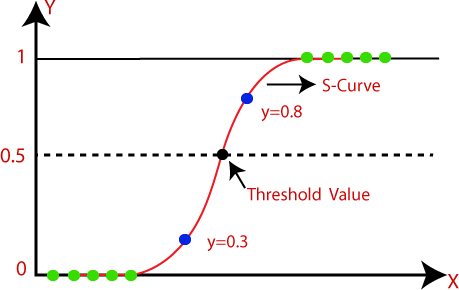
1. **MODELS FOR CLASSIFICATION:**

**LOGISTIC REGRESSION:**

Logistic Regression is a Machine learning Algorithm, which works as predictive analysis. It is a process of modeling the probability of a discrete outcome given an input feature. Logistic Regression is one of the basic and popular algorithms to solve a classification type of problem in Machine Learning. It predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical value. It can be either {Yes , No} or {0 or 1},{ true or False},etc.

In Logistic regression, we fit an “S” shaped logistic function instead of fitting a regression line, which predicts two maximum values( 0 or 1). Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. It can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.

The below image is showing the logistic function:



**Fig: Logistic Regression graph**

The sigmoid function is a mathematical function used to map the predicted values to probabilities. It is also known as the Logistic function.

**Assumptions for Logistic Regression:**

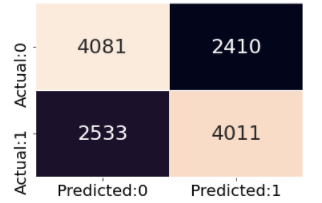
The dependent variable must be categorical in nature.

The Independent variable should not have Multi-Collinearity.

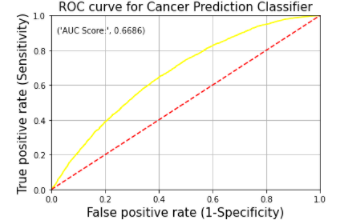
**Steps in Logistic Regression:** To implement the Logistic Regression using Python, we will use the same steps as we have done in previous topics of Regression. Below are the steps:

* + Data Pre-processing step
  + Fitting Logistic Regression to the Training
  + Predicting the Test Result
  + Test accuracy of the result(Creation of Confusion matrix)
  + Visualizing the test set result

**Confusion Matrix with decision Tree:**



**Roc curve with decision Tree:**



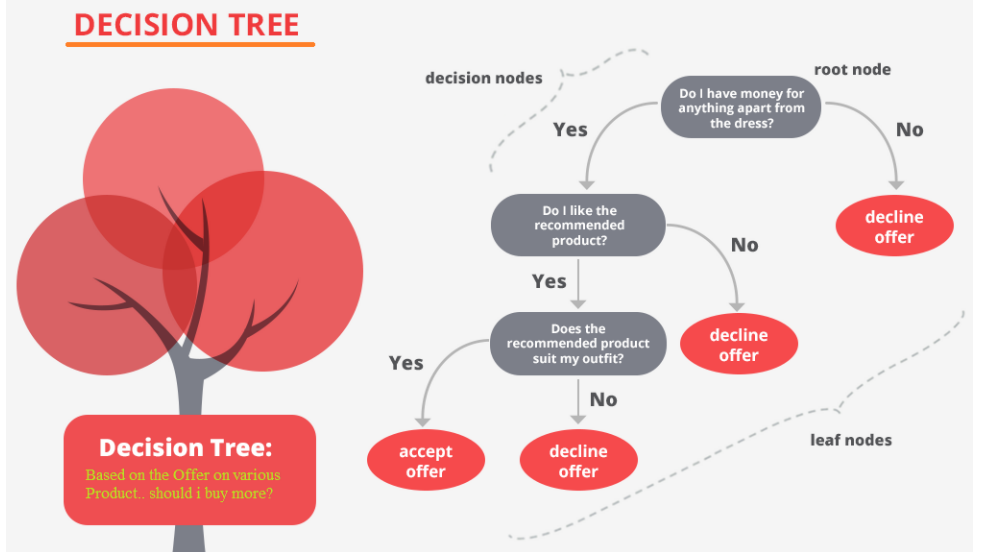
**DECISION TREE ALGORITHM:**

A Decision Tree Algorithm is a Machine Learning Algorithm. This algorithm can be used for the both Classification and Regression types of the Data. It is a tree-structured classified data and where internal nodes represent the features of a data set.

Branches represent the decision rules and the Leaf node represents the Outcome of Data.

In a Decision tree, there are two nodes ‘Decision Node’ and ‘Leaf Node’. The decisions are taken on the performance of a feature of a given data. It starts with a root node and which expands on further branches and constructs a tree-like structure.it built from top to bottom. A decision tree is built based on the simple answer{Yes, No}, which further split the tree into sub-trees. A decision tree can contain categorical data as well as numerical data.

The below diagram shows the general structure of a Decision Tree.



**Fig-6.2 Decision Tree split**

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from training data.

**Types of Decision Trees:**

Types of decision trees are based on the type of target variable we have. It can be of two types:

1. Categorical Variable Decision Tree**:** A decision Tree has a categorical target variable then it is

called a “Categorical variable decision tree”.

1. Continuous Variable Decision Tree**:** The decision Tree has a continuous target variable then it is

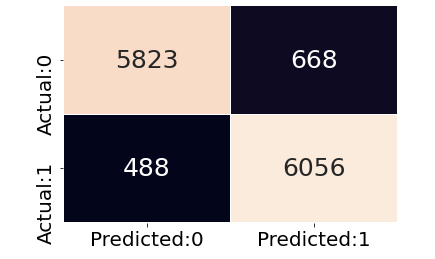
called” Continuous Variable Decision Tree”.

**Assumptions for a Decision Tree:**

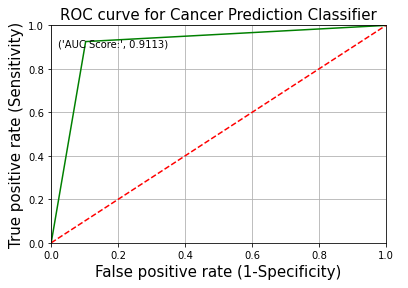
* + Firstly, the whole training set is considered as a Root.
  + Feature values are preferred to be categorical. If the values are continuous then they are discredited prior to building the model.
  + Records are distributed recursively on the basis of attribute values.
  + Order to placing attributes as root or internal node of the tree is done by using some statistical approach.

The decision of making strategic splits heavily affects a tree’s accuracy. The decision criteria are different for classification and regression trees. The algorithm selection is also based on the type of target variables. Let us look at some algorithms used in Decision Trees:

**Confusion Matrix with decision Tree:**



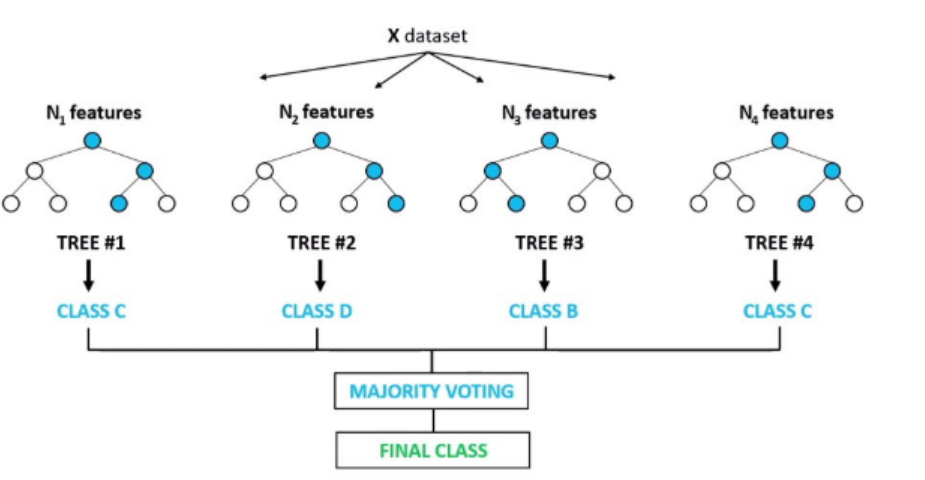
**Roc curve with decision Tree:**



**RANDOM FOREST ALGORITHM:**

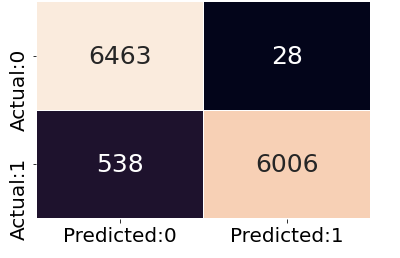
Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both classification and Regression problems in ML.it is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex Problem and to improve the performance of the model. Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions and it predicts the final output. The Greater the number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest Algorithm:

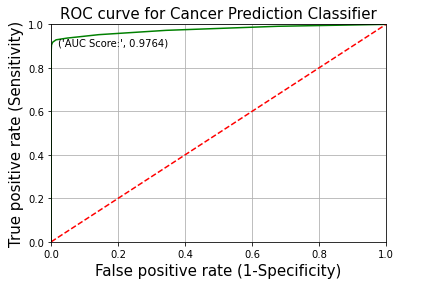


**Fig-6.3 Random Forest**

**Confusion Matrix with Random Forest:**



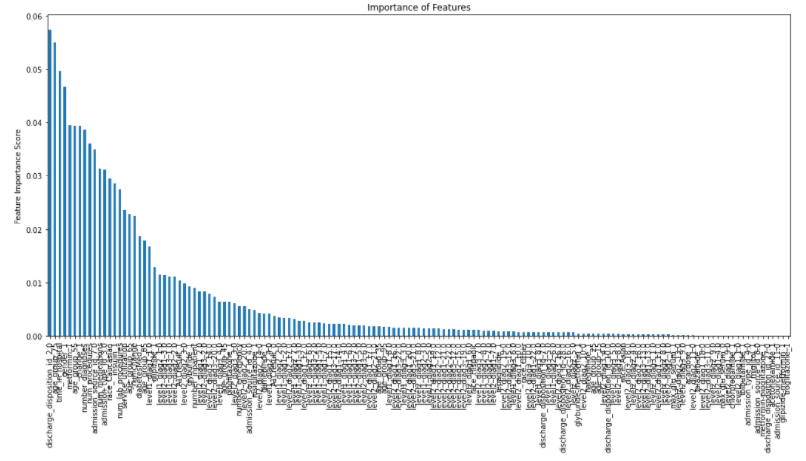
**Roc Curve with Random Forest:**



**Assumptions for the Random Forest** :

* + It takes less training time as compared to other algorithms.
  + It predicts output with high accuracy, even for the large dataset it runs efficiently.
  + It can also maintain accuracy when a large proportion of data is missing.

**Feature Importance:**

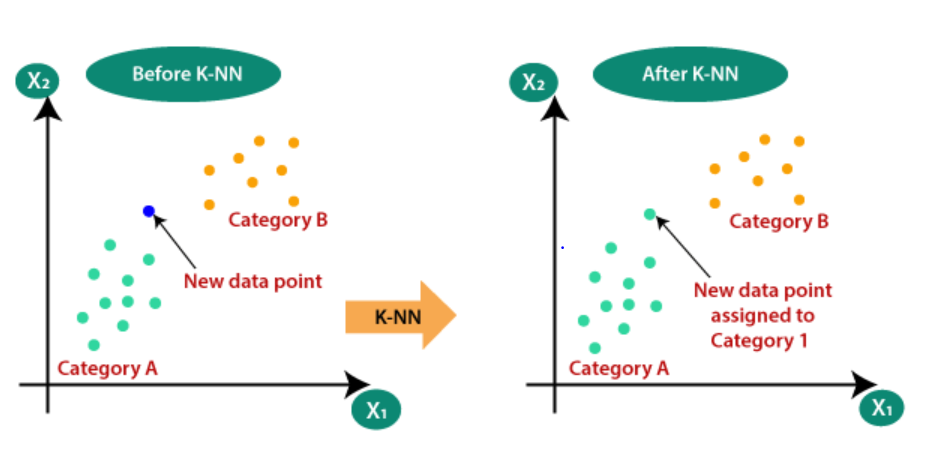


**K-NEAREST NEIGHBOR ALGORITHM (KNN ALGORITHM) :**

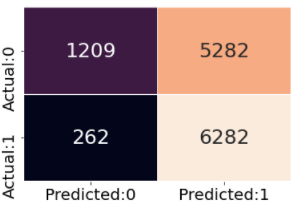
K-Nearest Neighbors(KNN) is one of the simplest algorithms used in Machine Learning for both regression and classification problems. KNN algorithms use data and classify new data points based on similarity measures .classification is done by a majority vote to its neighbors. The number of nearest neighbors to a new unknown variable that has to be predicted or classified is denoted by the symbol ‘K’. The KNN algorithm can compete with the most accurate models because it makes highly accurate predictions. Therefore, you can use the KNN algorithm for applications that require high accuracy but that do not require a human-readable model. The quality of the predictions depends on the distance measure. ‘K’in KNN is a parameter that refers to the number of nearest neighbors to include in the majority of the voting process.

**Working of KNN Algorithm :**

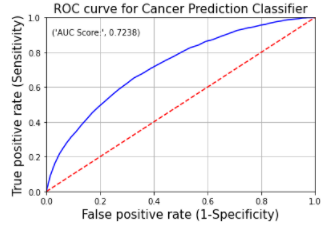
1. For implementing any algorithm, we need a dataset. So during the first step of KNN, we must load the training as well as test data.
2. Select the number K of the neighbors
3. Calculate the Euclidean distance of K number of neighbors
4. Take the K nearest neighbors as per the calculated Euclidean distance.
5. Among these k neighbors, count the number of the data points in each category.
6. Assign the new data points to that category for which the number of the neighbor is maximum.
7. And then, our model is Done.



**Confusion Matrix with KNN Algorithm:**



**Roc curve with KNN Algorithm:**



## BOOSTING ALGORITHMS:

Boosting is an ensemble learning method that combines a set of weak learners into a strong learner to minimize training errors. In boosting, a random sample of data is selected, fitted with a model, and then trained sequentially- that is, each model tries to compensate for the weakness of its predecessor.

The term ‘Boosting’ refers to a family of algorithms that converts weak learners to strong learners.

There are three types of Boosting Algorithms which are as follows:

## AdaBoost algorithm

1. **Gradient Boosting algorithm**

## XG Boost algorithm

**Adaptive Boosting Algorithm:**

AdaBoost Algorithm is a Boosting Technique used as an Ensemble method in Machine Learning.

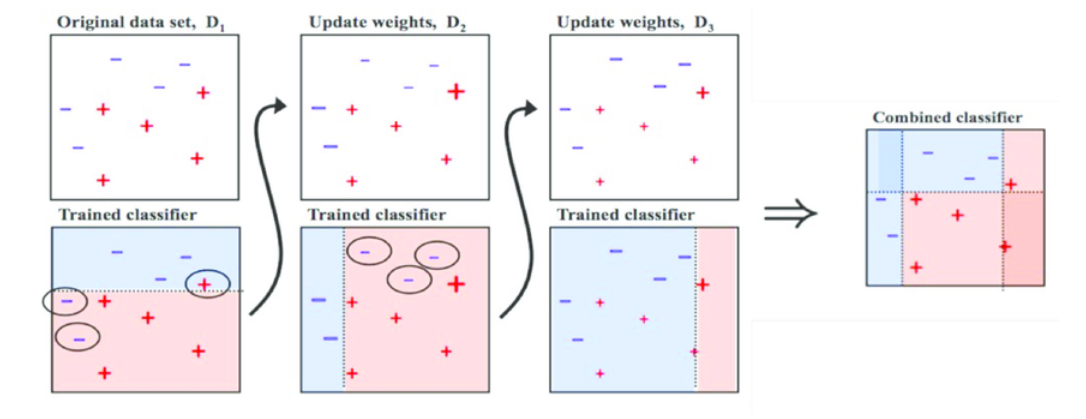
Adaboost helps you combine multiple “weak classifiers” into a single “strong classifier”. AdaBoost Algorithm can be used for the both Classification and Regression Problem. AdaBoost can be used to boost the performance of any machine learning algorithm. It is used with weak learners. These are models that achieve accuracy just above random chance on a classification problem. The most suited and therefore most common algorithm used with AdaBoost is decision trees with one level. It works on the principle of learners growing sequentially. Except for the first, each subsequent learner is

grown from previously grown learners.

## Working process of Adaptive Boosting Algorithm:

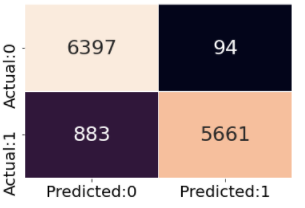
* + Importing the dataset
  + Splitting the dataset into training and test samples
  + Classifying the predictors and Target
  + Initializing the Adaboost classifier and fitting the training data.
  + Predicting the classes for the test set.
  + Attaching the predictions to test set for comparing.

The below figure explains the model of Ada Boost algorithm:

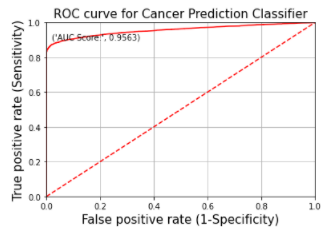


**Fig-6.5 Adaptive Boosting Ensemble technique**

**Confusion Matrix from ADA boosting:**



**ROC curve from ADA boosting:**



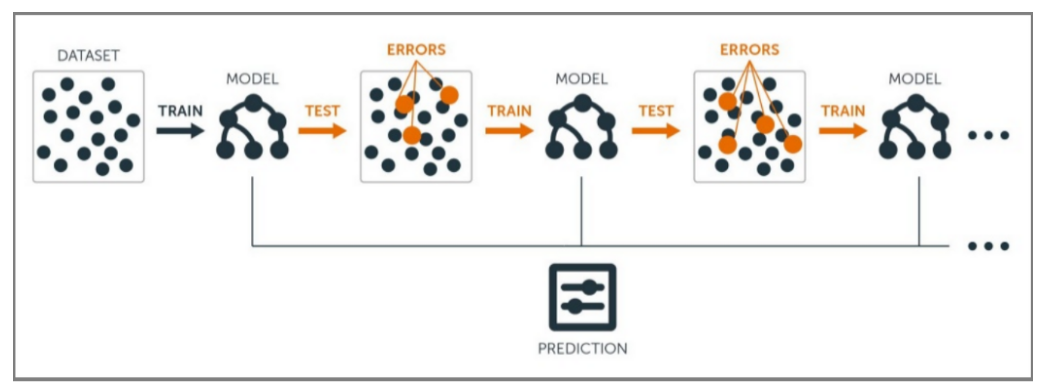
## GRADIENT BOOSTING IN ALGORITHM:

Gradient Boosting Algorithm is a Boosting Technique used as an Ensemble method in Machine Learning. Gradient boosting is a machine learning technique for regression, classification, and other tasks, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. Gradient Boosting algorithm is one of the most powerful algorithms in the field of machine learning. We know that the errors in Machine Learning algorithms are broadly classified into two categories i.e., Bias Error and Variance Error. As gradient boosting is one of the boosting algorithms, it is used to minimize the bias error of the model.

Gradient boosting algorithm can be used for predicting not only continuous target variable but also categorical target variable .when it is used as a regressor, the cost function is Mean Square Error(MSE) And when it is used as a classifier then the cost function is Log loss.

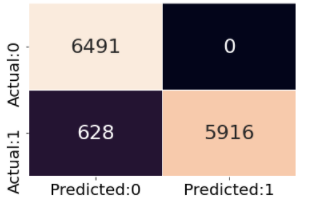
## Working steps of the Gradient Boosting Algorithm:

* + Calculate the average of the target Label.
  + Calculate the residuals
  + Construct a decision tree
  + Predict the target label using all of the trees within the ensemble
  + Compute the new residuals.

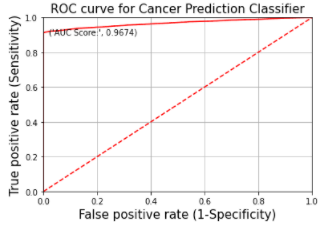


**Fig- Gradient Boosting Algorithm- Ensemble technique**

**Confusion matrix from Gradient Boosting:**



**ROC Curve from Gradient boosting:**

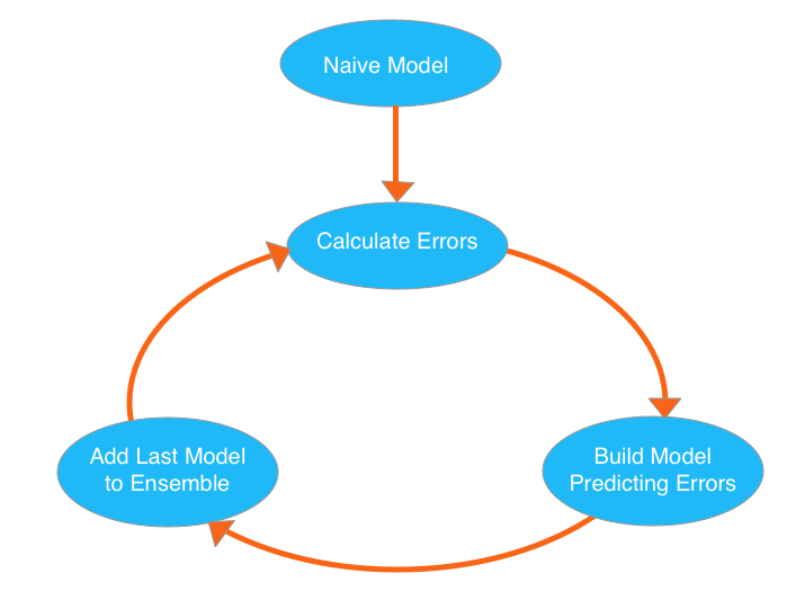


## XG BOOSTING ALGORITHM:

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. We use the XGBoost for only two reasons :

1. Execution of speed
2. And Model Performance.

Gradient Boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

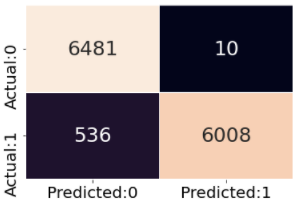


**Fig- XGBoost Algorithm features**

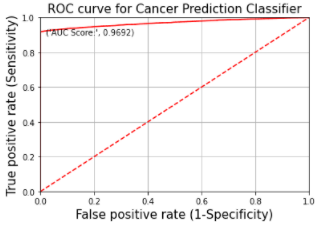
## Working of XG Boosting Algorithm:

* + Load the Libraries of an Xgboost
  + Load the Dataset
  + Data Cleaning and Feature Engineering.
  + Tune and Run the model.
  + These are the simple steps for a data problem to solve the boost Algorithm.

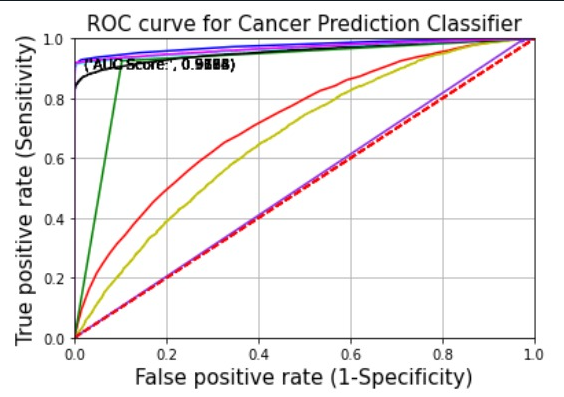
**Confusion matrix from XGboost:**



**ROC curve from XGboost:**



**Model Comparison:**



Blue: Random Forest Model

Green: Decision Tree Model

Cyan: GBoost Model

Black: ADA Boost Model

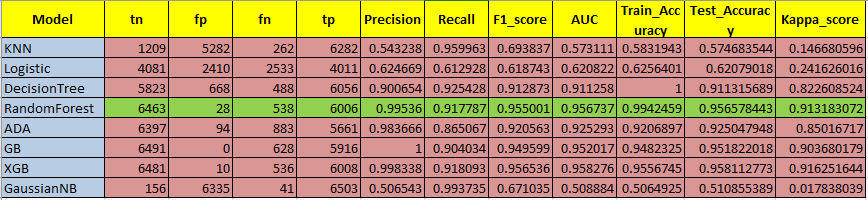
Red: Knn Model

Yellow: Logistic Regression Model

Magneta: XGBoost Model

Violet: Gaussian Naïve bayes

**Comparison table between different models:**



**Discussion:**

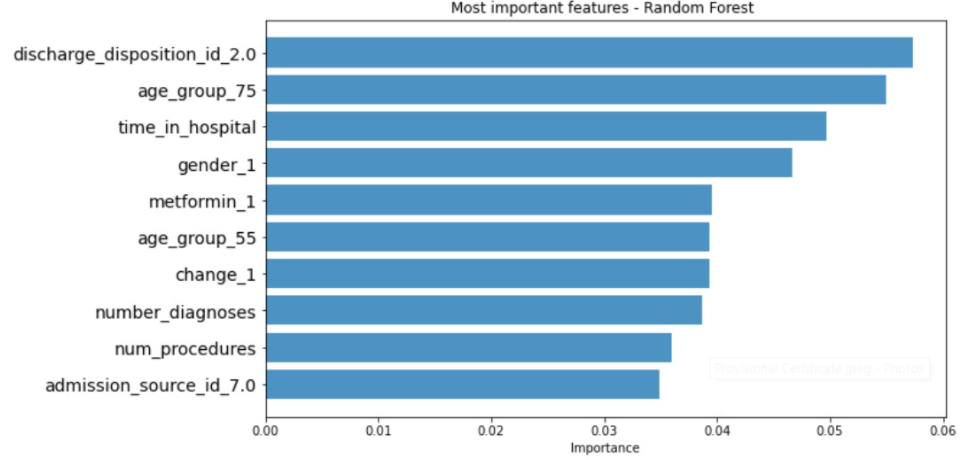
In this study, Random Forest, XGBoost, Gradient Boosting, GaussianNB, Decision Tree, Logistic Regression were used to construct a 30-day hospital readmission risk model. The dataset was used to train and verify the model through an 80% training set and a 20% test set. The RF algorithm showed good predictive performance in all three models. The complete process of the model design shown here included algorithm selection, which will be of reference significance for other similar predictive model designs in the future.

There are many classic ML algorithms in the classification of medical data. The RF algorithm can outperform the DT algorithm in most datasets, suggesting that it could be a method of feature importance computation. The distribution of the target variable was unbalanced. Most of the patients in the dataset had no readmission record (53.69%), with only 11.22% having been readmitted within 30 days (< 30), when the remaining patients (35.09%) being readmitted beyond 30 days (> 30) after the first discharge. In reality, the readmissions after > 30 days were difficult to measure because there was not much difference between admissions on day 30 and those on day 31. The reduction of overall classification accuracy is the main goal of the traditional ML algorithms. The major category gains too much attention in the process of classification when data imbalances occur, and the performance to identify minority sample decreased. However, the targeting category requiring prediction is a very small proportion of the overall quantity in medical data. The inconsistency between sensitivity and specificity was significantly reduced when the training set was balanced.

Besides, the analysis incorporates some of the factors provided in the dataset but lacks some key features, such as disease progression, family history, body mass index, and insurance information. Besides, inconsistencies existed between different genders from different races, for example, a previous study analyzed the readmission rates across non-Hispanic Whites, non-Hispanic Blacks, and Hispanics, revealing that the percentage of female patients varies among different ethnic groups . In addition, the lack of practical experience of doctors at the first diagnosis and the subjective choice of patients may also account for the determined readmission rates. So many known and unknown risk factors in medical activities can affect readmissions, and model performances will be greatly improved through the analysis of real-world data and the data-driven mining of potential risk factors affecting patient readmission rates.

RF was more suitable for predicting accidental readmissions in this study. As one of the most commonly used algorithms in current classification work, RF has better predictive performance and can give variable importance measures during classification.

Below mentioned are the most important features in determining readmission rate :



# **Conclusion :**

# In conclusion, the decision to obtain a measurement of HbA1c for patients with diabetes mellitus is a useful predictor of readmission rates which may prove valuable in the development of strategies to reduce readmission rates and costs for the care of individuals with diabetes mellitus. For instance, our analysis showed that the profile of readmission differed significantly in patients where Hba1c was checked in the setting of a primary diabetes diagnosis, when compared to those with a primary circulatory disorder. While readmission rates remained the highest for patients with circulatory diagnoses readmission rates for patients with diabetes appeared to be associated with the decision to test for HbA1c, rather than the values of the HbA1c result.

# **References :**

# [1] G. E. Umpierrez, S. D. Isaacs, N. Bazargan, X. You, L. M. Thaler,

# and A. E. Kitabchi, “Hyperglycemia: an independent marker

# of in-hospital mortality in patients with undiagnosed diabetes,”

# Journal of Clinical Endocrinology and Metabolism, vol. 87, no. 3,

# pp. 978–982, 2002.

# [2] C. S. Levetan, M. Passaro, K. Jablonski, M. Kass, and R. E. Ratner, “Unrecognized diabetes among hospitalized patients,” Diabetes Care, vol. 21, no. 2, pp. 246–249, 1998.

# [3] S. E. Siegelaar, J. B. L. Hoekstra, and J. H. Devries, “Special

# considerations for the diabetic patient in the ICU; targets

# for treatment and risks of hypoglycaemia,” Best Practice and

# Research: Clinical Endocrinology and Metabolism, vol. 25, no. 5,

# pp. 825–834, 2011.

# [4] A. G. Pittas, R. D. Siegel, and J. Lau, “Insulin therapy for critically ill hospitalized patients: a meta-analysis of randomized

# controlled trials,” Archives of Internal Medicine, vol. 164, no. 18,

# pp. 2005–2011, 2004.

# [5] A. C. Tricco, N. M. Ivers, J. M. Grimshaw et al., “Effectiveness of

# quality improvement strategies on the management of diabetes:

# a systematic review and meta-analysis,” The Lancet, vol. 379, no.

# 9833, pp. 2252–2261, 2012.

# [6] M. C. Lansang and G. E. Umpierrez, “Management of inpatient

# hyperglycemia in noncritically ill patients,” Diabetes Spectrum,

# vol. 21, no. 4, pp. 248–255, 2008.

# [7] R. Vinik and J. Clements, “Management of the hyperglycemic

# inpatient: tips, tools, and protocols for the clinician,” Hospital

# Practice, vol. 39, no. 2, pp. 40–46, 2011.

# [8] K. J. Cios and G. W. Moore, “Uniqueness of medical data

# mining,” Artificial Intelligence in Medicine, vol. 26, no. 1-2, pp.

# 1–24, 2002.

# [9] A. Frank and A. Asuncion, UCI Machine Learning Repository,

# University of California, School of Information and Computer

# Science, 2010.

# [10] R. M. Bergenstal, J. L. Fahrbach, S. R. Iorga, Y. Fan, and S. A.

# Foster, “Preadmission glycemic control and changes to diabetes

# mellitus treatment regimen after hospitalization,” Endocrine

# Practice, vol. 18, no. 3, pp. 371–375, 2012.