Final Project Report

Title - Diabetic patient readmission prediction.

Group 2

Mentor: Vidya K

Members: V Keerthi Vikram, Sarvesh Mankar, Kajal Chopda, Nishad Kharwade, Vidhushi Vanjari.

Problem Statement

To help the hospital prioritize attention towards certain diabetic patients who have a high probability of getting readmitted to the hospital based on past information from the data collected.

Literature Survey

- “Diabetes 130-US hospitals dataset” by Weiyan Wu, Yao Jin, Jiwei Wang, Lingyun Mao, Han W.
- “Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records” By Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios and John N. Clore.

Introduction

Type 1 diabetes is a disease in which the body does not make enough insulin to control blood sugar levels. Type 1 diabetes was previously called insulin-dependent diabetes or juvenile diabetes. Type 1 diabetes is an autoimmune condition. It happens when your body attacks your pancreas with antibodies. The organ is damaged and doesn't make insulin.

When you have type 2 diabetes, your pancreas usually creates some insulin. But either it’s not enough or your body doesn’t use it like it should. Insulin resistance, when your cells don’t respond to insulin, usually happens in fat, liver, and muscle cells. Type 2 diabetes is often milder than type 1. But it can still cause major health complications, especially in the tiny blood vessels in your kidneys, nerves, and eyes. Type 2 also raises your risk of heart disease and stroke. Type 2 diabetes used to be called non-insulin-dependent or adult-onset diabetes. But it’s become more common in children and teens over the past 20 years, largely because more young people are overweight or obese.

Descriptive Statistics

The shape of the dataset was checked which was found to be 1,01,763 rows and 50 columns.

Following this the data type of columns was found out.

It could be seen that there were 9 numerical and 41 categorical columns.

Upon observation it was found that 3 columns were pre encoded.

It was also observed that we had 24 columns related to the dosage (Increase, Decrease, Steady or not given) for the drugs. Upon performing boxplot, we could also observe there were outliers.

Also, outliers could be observed looking at the difference in mean and median value.

Our target column is readmitted which is a categorical column.

EDA

Missing value imputation

Missing values were present in the dataset in Weight, Race, Diag_1, Diag_2, Diag_3, Payer code, Medical Specialty.

Weight, Payer code, Medical Speciality columns were dropped due to above 40% missing values.

Missing values in Race was imputed using mode.

Diag_1, Diag_2, Diag_3 had missing values replaced by No Diagnosis Done.

Statistical test

Statistical tests such as chi2 contingency test was done on categorical columns and logistic regression (Logit) was done on numerical columns and non-significant columns were dropped.

Some more columns were dropped due to Insufficient data based on domain knowledge.

Columns dropped are

- acarbose
- acetohexamide
- chlorpropamide
- citoglipton
- examide
- glimepiride-pioglitazone
- glipizide-metformin
- glyburide-metformin
- metformin-pioglitazone
- metformin-rosiglitazone
- miglitol
- tolazamide
- troglitazone
- tolbutamide
- nateglinide
- glyburide

Feature Engineering

We combined num_lab_procedures and num_procedures to get num_ovrll_prcdrs and combined number_outpatient , number_emergency and number_inpatient to get num_prvs_vists.

We further generalized diag_1, diag_2 and diag_3 columns based on ICD-9 2008 code to strings.

We dropped the rows containing others subclass of gender column as it had very less data points and could have caused bias.

Correlation

It was checked and none of the columns were found to be highly correlated with each other. Pearson correlation was used for numerical columns and Cramers_v correlation was used for categorical columns

Pearson Correlation

In statistics, the Pearson correlation coefficient also referred to as Pearson's r, the Pearson product-moment correlation coefficient (PPMCC), or the bivariate correlation, is a statistic that measures linear correlation between two variables X and Y. It has a value between +1 and −1. A value of +1 is total positive linear correlation, 0 is no linear correlation, and −1 is total negative linear correlation.

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

Where,

cov is the covariance

σ_X is the standard deviation of X

σ_Y is the standard deviation of Y

Cramer's V

In statistics, Cramer's V is a measure of association between two nominal variables, giving a value between 0 and +1 (inclusive). It is based on Pearson's chi-squared statistic.

Cramer's V may also be applied to goodness of fit chi-squared models when there is a 1 × k table (in this case r = 1).

$$V = \sqrt{\frac{\varphi^2}{min(k-1, r-1)}} = \sqrt{\frac{\chi^2/n}{min(k-1, r-1)}}$$

Where,

φ is the phi coefficient

χ^2 is derived from Pearson's chi-squared test

n is the grand total observations

k is the number of columns

r is the number of rows

Multicollinearity

It was checked for and the columns were found to not have multicollinearity.

Variance Inflation Factor

In statistics, multicollinearity (also collinearity) is a phenomenon in which one predictor variable in a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy. In this situation, the coefficient estimates of the multiple regression may change erratically in response to small changes in the model or the data. Multicollinearity does not reduce the predictive power or reliability of the model as a whole, at least within the sample data set; it only affects calculations regarding individual predictors. That is, a multivariate regression model with collinear predictors can indicate how well the entire bundle of predictors predicts the outcome variable, but it may not give valid results about any individual predictor, or about which predictors are redundant with respect to others.

In statistics, the variance inflation factor (VIF) is the quotient of the variance in a model with multiple terms by the variance of a model with one term alone.[1] It quantifies the severity of multicollinearity in an ordinary least squares regression analysis. It provides an index that measures how much the variance (the square of the estimate's standard deviation) of an estimated regression coefficient is increased because of collinearity.

$$VIF_i = \frac{1}{1 - R_i^2}$$

where R_i^2 is the coefficient of determination of the regression equation.

New Shape

Following this our new number of rows is 1,01,760 and our new number of columns is 26.

Skewness and Transformation

We checked for skewness in the dataset and the numerical columns were found to have some skewness.

Boxcox Transformation

A Box Cox transformation is a way to transform non-normal dependent variables into a normal shape. Normality is an important assumption for many statistical techniques; if your data isn't normal, applying a Box-Cox means that you are able to run a broader number of tests.

At the core of the Box Cox transformation is an exponent, lambda (λ), which varies from -5 to 5. All values of λ are considered and the optimal value for your data is selected; The "optimal value" is the one which results in the best approximation of a normal distribution curve. The transformation of Y has the form:

$$y(\lambda) = \frac{y^{\lambda} - 1}{\lambda}\ ,\ if\ \lambda \neq 0$$
$$y(\lambda) = log(y),\ if\ \lambda = 0$$

Standardization

Following this the numerical columns were scaled down to the same scale using standard scaler.

Standard Scaler

Standardize generally means changing the values so that the distribution standard deviation from the mean equals one. It outputs something very close to a normal distribution. Standard Scaler standardizes a feature by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation. Standard Scaler results in a distribution with a standard deviation equal to 1. The variance is equal to 1 also, because *variance = standard deviation squared*. And 1 squared = 1. Standard Scaler makes the mean of the distribution 0. About 68% of the values will lie be between -1 and 1.

Encoding

We have done mean encoding on the diagnosis columns as they have high cardinality. This was followed by one hot encoding wherever possible and the remaining areas mean encoding.

Mean Encoding

```
from sklearn.base import BaseEstimator
class MeanEncoding(BaseEstimator):

    """   In Mean Encoding we take the number
    of labels into account along with the target variable
    to encode the labels into machine comprehensible values    """

    def __init__(self, feature, C=0.1):
        self.C = C
        self.feature = feature

    def fit(self, X_train, y_train):

        df = pd.DataFrame(
            {'feature': X_train[self.feature], 'target': y_train}).dropna()

        self.global_mean = df.target.mean()
        mean = df.groupby('feature').target.mean()
        size = df.groupby('feature').target.size()

        self.encoding = (self.global_mean * self.C +
                         mean * size) / (self.C + size)

    def transform(self, X_test):

        X_test[self.feature] = X_test[self.feature].map(
            self.encoding).fillna(self.global_mean).values

        return X_test

    def fit_transform(self, X_train, y_train):

        df = pd.DataFrame(
            {'feature': X_train[self.feature], 'target': y_train}).dropna()

        self.global_mean = df.target.mean()
        mean = df.groupby('feature').target.mean()
        size = df.groupby('feature').target.size()
        self.encoding = (self.global_mean * self.C +
                         mean * size) / (self.C + size)

        X_train[self.feature] = X_train[self.feature].map(
            self.encoding).fillna(self.global_mean).values

        return X_train

for f in cat_cols:
    me = MeanEncoding(f, C=0.01 * len(X[f].unique()))
    me.fit(X, y)
    X = me.transform(X)
    test = me.transform(test)
```

In Mean Encoding we take the number of labels into account along with the target variable to encode the labels into machine comprehensible values.

Mean encoding can embody the target in the label whereas label encoding has no correlation with the target.

In case of large number of features, mean encoding could prove to be a much simpler alternative.

A histogram of predictions using label & mean encoding show that mean encoding tend to group the classes together whereas the grouping is random in case of label encoding.

One-Hot Encoding

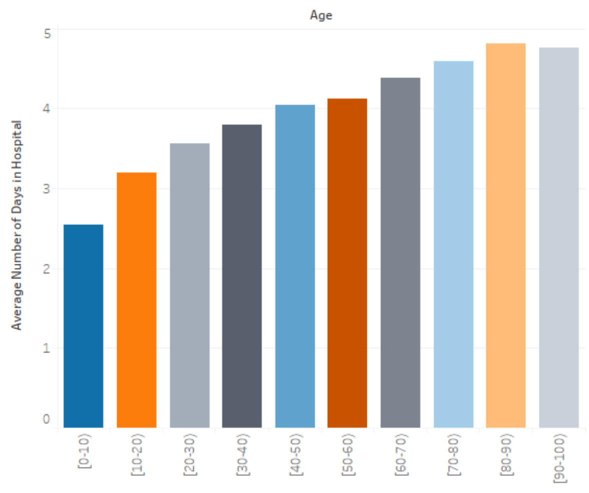
For categorical variables where no ordinal relationship exists, the integer encoding may not be enough, at best, or misleading to the model at worst.

Forcing an ordinal relationship via an ordinal encoding and allowing the model to assume a natural ordering between categories may result in poor performance or unexpected results (predictions halfway between categories).

In this case, a one-hot encoding can be applied to the ordinal representation. This is where the integer encoded variable is removed and one new binary variable is added for each unique integer value in the variable.

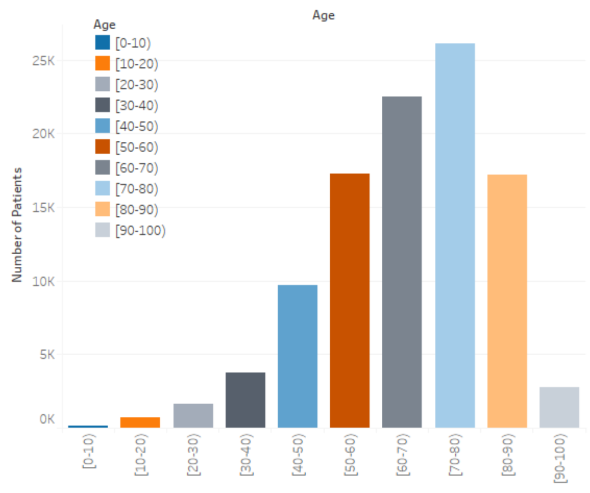
Visualizations

Age vs Average number of days



As the age increases the number of days spent in the hospital increases.

Age vs number of patients



As the age increases the number of patients in that age group visiting the hospital increases.

Insulin effectivness

	Insulin			
Readmitted	Down	No	Steady	Up
<30	1,698	2,510	3,433	1,470
>30	4,752	8,722	10,482	4,362
NO	5,768	12,748	16,934	5,484

Metformin effectivness

	Metformin			
Readmitted	Down	No	Steady	Up
<30	69	7,172	1,782	88
>30	190	21,612	6,169	347
NO	316	29,591	10,395	632

Glipzide effectivness

	Glipizide			
Readmitted	Down	No	Steady	Up
<30	85	7,659	1,268	99
>30	211	23,588	4,233	286
NO	264	34,430	5,855	385

Pioglitazone effectivness

	Pioglitazone			
Readmitted	Down	No	Steady	Up
<30	18	8,337	727	29
>30	45	25,561	2,620	92
NO	55	37,137	3,629	113

Glimepiride effectivness

	Glimepiride			
Readmitted	Down	No	Steady	Up
<30	25	8,581	468	37
>30	68	26,411	1,737	102
NO	101	38,180	2,465	188

Glyburide effectiveness

	Glyburide			
Readmitted	Down	No	Steady	Up
<30	52	7,979	995	85
>30	223	24,600	3,221	274
NO	289	35,134	5,058	453

Insulin effectivness

	Insulin			
Readmitted	Down	No	Steady	Up
<30	1,698	2,510	3,433	1,470
>30	4,752	8,722	10,482	4,362
NO	5,768	12,748	16,934	5,484

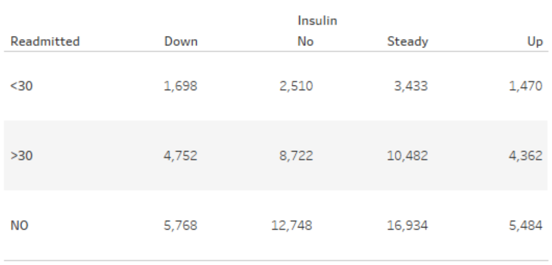
Among those taking this drug only 12.1% were readmitted within 30 days. 36% of the patients were readmitted after 30 days. The remaining patients were not readmitted. It can be seen that this drug is useful in reducing the patient readmission rate. Around 13.9% of those whose dosage was lowered were readmitted within 30 days. 38.8% were readmitted after 30 days. The remaining patients were not readmitted. Around 11.1% of those whose dosage was steady were readmitted within 30 days. 33.9% were readmitted after 30 days. The remaining patients were not readmitted. Around 12.9% of those whose dosage was upped were readmitted within 30 days. 38.5% were readmitted after 30 days. The remaining patients were not readmitted.

Metformin effectivness

	Metformin			
Readmitted	Down	No	Steady	Up
<30	69	7,172	1,782	88
>30	190	21,612	6,169	347
NO	316	29,591	10,395	632

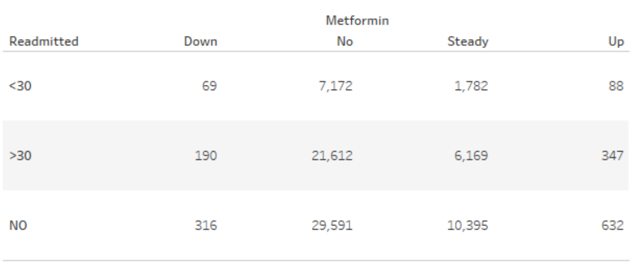
Among those taking this drug only 9.7% were readmitted within 30 days. 33.5% of the patients were readmitted after 30 days. The remaining patients were not readmitted. It can be seen that this drug is useful in reducing the patient readmission rate. Around 12% of those whose dosage was lowered were readmitted within 30 days. 33% were readmitted after 30 days. The remaining patients were not readmitted. Around 9.7% of those whose dosage was steady were readmitted within 30 days. 33.6% were readmitted after 30 days. The remaining patients were not readmitted. Around 8.2% of those whose dosage was upped were readmitted within 30 days. 32.5% were readmitted after 30 days. The remaining patients were not readmitted.

Insulin effectivness

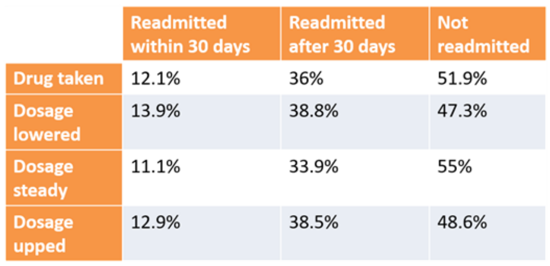


Readmitted	Insulin Down	Insulin No	Insulin Steady	Insulin Up
<30	1,698	2,510	3,433	1,470
>30	4,752	8,722	10,482	4,362
NO	5,768	12,748	16,934	5,484

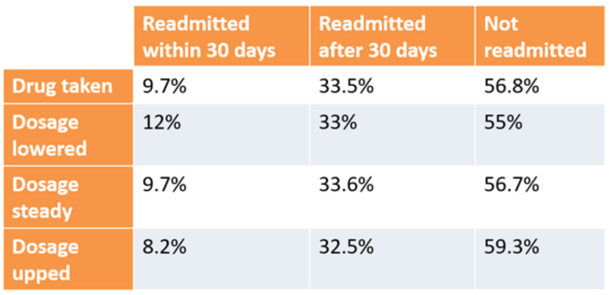
Metformin effectivness



Readmitted	Metformin Down	Metformin No	Metformin Steady	Metformin Up
<30	69	7,172	1,782	88
>30	190	21,612	6,169	347
NO	316	29,591	10,395	632

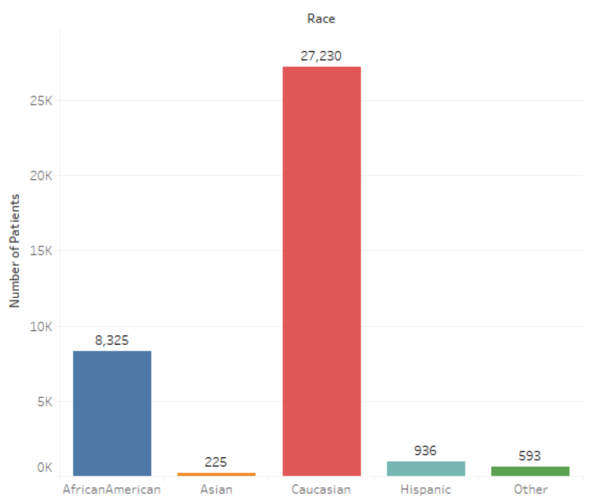


	Readmitted within 30 days	Readmitted after 30 days	Not readmitted
Drug taken	12.1%	36%	51.9%
Dosage lowered	13.9%	38.8%	47.3%
Dosage steady	11.1%	33.9%	55%
Dosage upped	12.9%	38.5%	48.6%

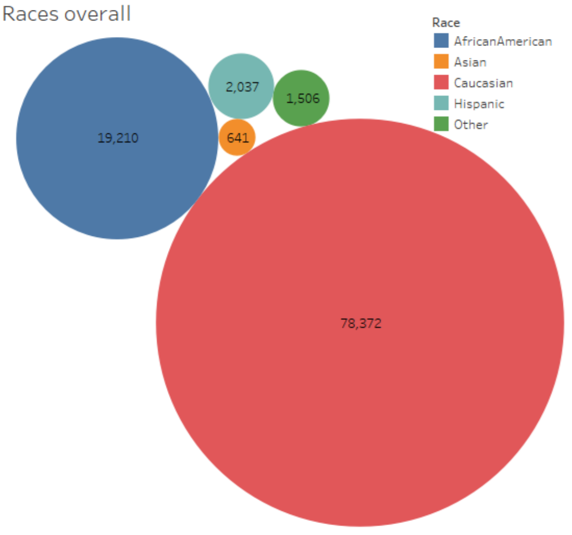


	Readmitted within 30 days	Readmitted after 30 days	Not readmitted
Drug taken	9.7%	33.5%	56.8%
Dosage lowered	12%	33%	55%
Dosage steady	9.7%	33.6%	56.7%
Dosage upped	8.2%	32.5%	59.3%

Races with diabetes



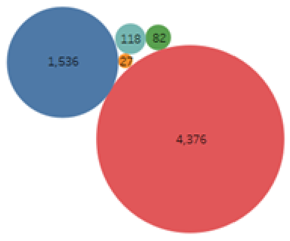
Races overall



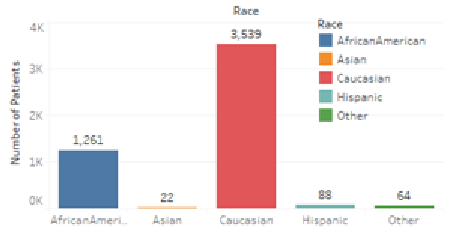
Hispanics (45.94%) and African-Americans (43.33%) are more probable to get diabetes.

Diabetes percent - Caucasian (34.74), Asian (35.10), Hispanic (45.94), African-Americans (43.33), Other (39.37)

Type 1 diabetes race wise

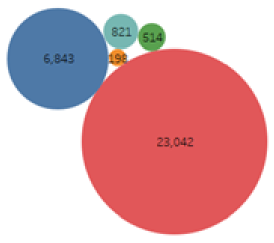


Diabetic medicine taken race wise type 1

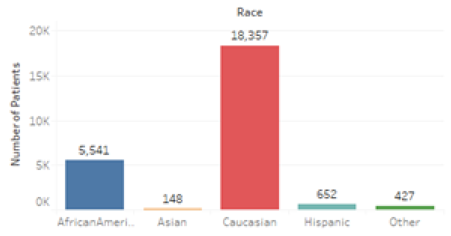


We can see that more people have diabetes type 2.

Type 2 diabetes race wise

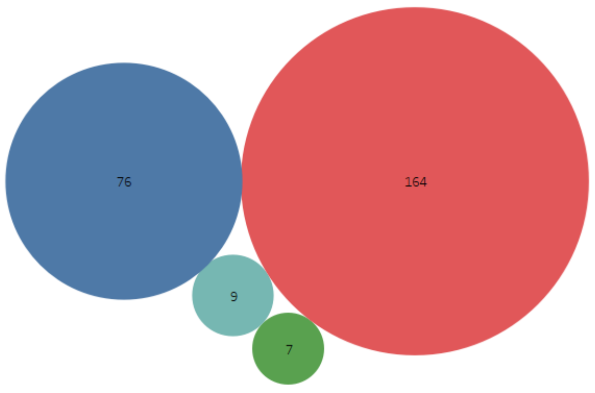


Diabetic medicine taken race wise type 2

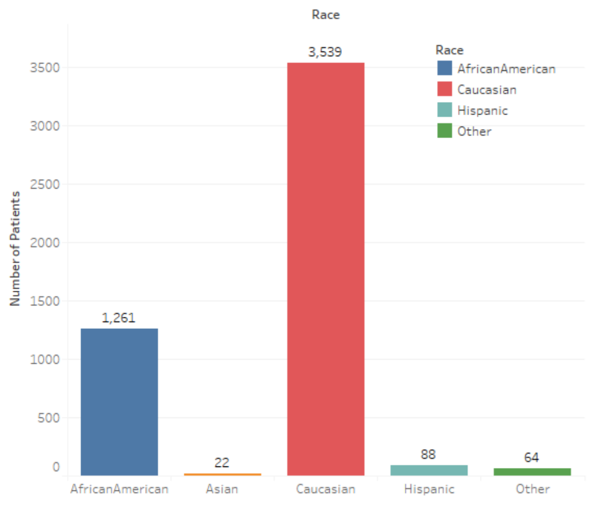


Not taking medicine - Caucasian (Typ1 - 19.12, Typ2 - 20.33), Asian (Typ1 - 18.51, Typ2 - 25.25), Hispanic (Typ1 - 25.42, Typ2 - 20.58), African-Americans (Typ1 - 17.90, Typ2 - 19.02), Other (Typ1 - 21.95, Typ2 - 16.92)

Type 1 diabetes not taking insulin

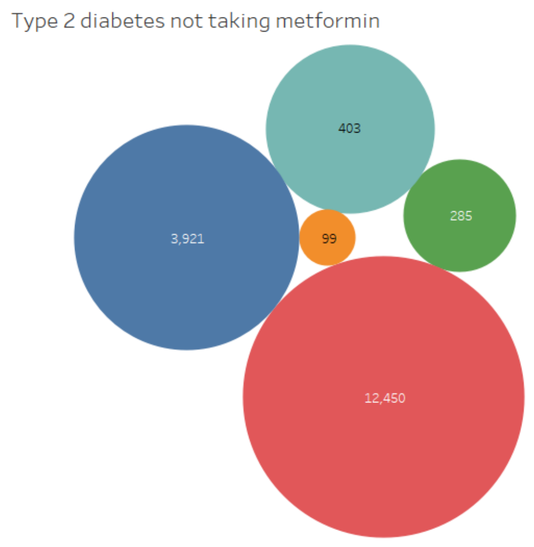


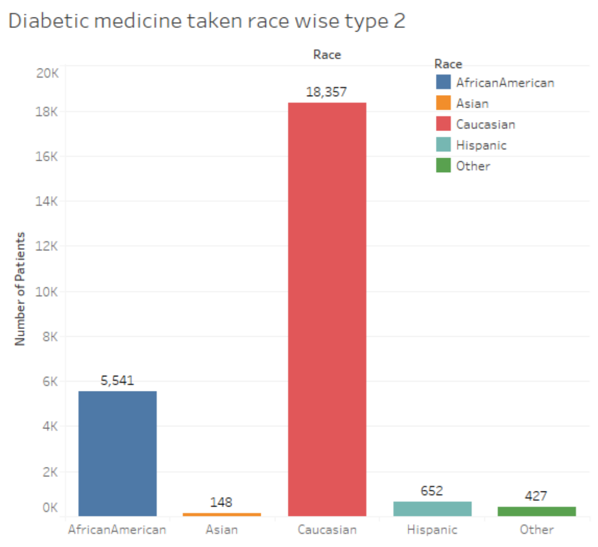
Diabetic medicine taken race wise type 1



Around 10.22% Hispanics and 10.93% of other races are not taking Insulin the popular drug in diabetes 1.

Medicine takers not insulin - Caucasian (4.63), Asian (0), Hispanic (10.22), African-Americans (6.02), Other (10.93)





Around 70.76% African-Americans and around 67.82% Caucasians do not take Metformin.

Medicine takers not metformin- Caucasian (67.82), Asian (66.89), Hispanic (61.80), African-Americans (70.76), Other (66.74)

Splitting the dataset into Train and Test

The dataset is split into Train and Test in the ratio of $70 : 30$ using train_test_splitter().

Balancing the Imbalanced dataset

Different balancing techniques (such as Random Oversampling, Random Under sampling, SMOTE) were performed. It was observed that SMOTE gives the best performance for the dataset during model building.

SMOTE

One approach to addressing imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples don't add any new information to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling Technique, or SMOTE for short. SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line. Specifically, a random example from the minority class is first chosen. Then *k* of the nearest neighbors for that example are found (typically *k=5*). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space.

Model Building

Three scenarios were created for model building

- Scenario 1 - There are 3 sub classes (NO, >30, <30).
- Scenario 2 – There are 2 Sub classes (NO, Readmitted [>30,<30]).
- Scenario 3 – There are 2 sub classes (>30, <30).

Models Used

Logistic Regression

The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

Decision Tree Classifier

Decision Tree Classifier is a simple representation for classifying examples where the data is continuously split according to a parameter. It is built through a process of binary recursive partitioning, where an iterative process of splitting the data into partitions and further splitting on each of the branches occurs. Each root node represents a single input variable (x) and a split point on that variable (assuming the variable is numeric). The leaf nodes of the tree contain an output variable (y) which is used to make a prediction

Support Vector Classifier

SVC is a non-parametric clustering algorithm that does not make any assumption on the number of shape of the clusters in the data. It works best for low dimensional data. Incase of high dimension data principal component analysis(PCA) is usually required. There are specialized optimization procedures that re-formulate the optimization problem to be a Quadratic

Programming problem. The most popular method for fitting SVM is the Sequential Minimal Optimization (SMO) method that is very efficient. It breaks the problem down into sub-problems that can be solved analytically (by calculating) rather than numerically (by searching or optimizing).

KNN

KNN is a lazy learning and non-parametric algorithm because it does not have a specialized training phase and it uses all the data for training in classification. It uses feature similarity to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. Because the entire training dataset is stored, one may want to think carefully about the consistency of the training data. It might be a good idea to curate it, update it often as new data becomes available and remove erroneous and outlier data.

Random Forest Classifier

Random Forest as its name implies,consists of a large number of individual decision trees that operates as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

Decision trees can suffer from high variance which makes their results fragile to the specific training data used. Building multiple models from samples of your training data, called bagging, can reduce this variance, but the trees are highly correlated. Random Forest is an extension of bagging that in addition to building trees based on multiple samples of your training data, it also constrains the features that can be used to build the trees, forcing trees to be different. This, in turn, can give a lift in performance.

XGBoost Classifier

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. XGBoost and Gradient Boosting Machines (GBMs) are both ensemble tree methods that apply the principle of boosting weak learners (CARTs generally) using the gradient descent architecture. However, XGBoost improves upon the base GBM framework through systems optimization and algorithmic enhancements.

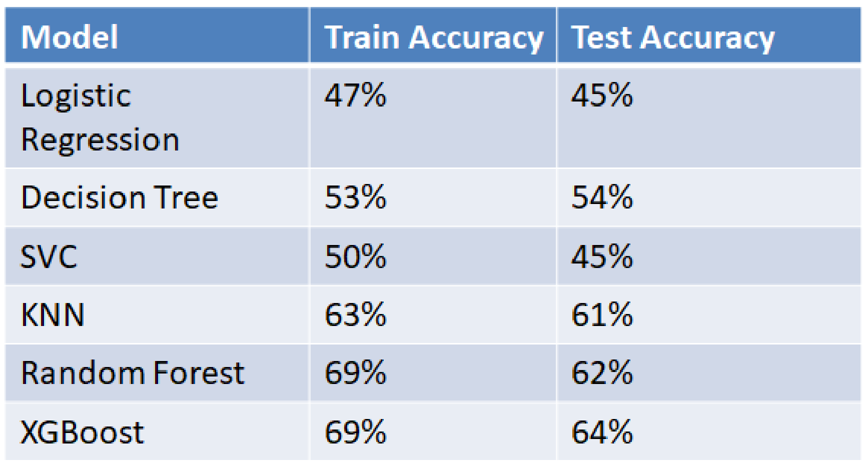
LightGBM

Light GBM is a gradient boosting framework that uses tree based learning algorithm. Light GBM grows tree vertically while other algorithm grows trees horizontally meaning that Light GBM grows tree leaf-wise while other algorithm grows level-wise. It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm. The size of data is increasing day by day and it is becoming difficult for traditional data science algorithms to give faster results. Light GBM is prefixed as 'Light' because of its high speed. Light GBM can handle the large size of data and takes lower memory to run. Another reason of why Light GBM is popular is because it focuses on accuracy of results. LGBM also supports GPU learning and thus data scientists are widely using LGBM for data science application development. Light GBM is sensitive to overfitting and can easily overfit small data.

Scenario 1

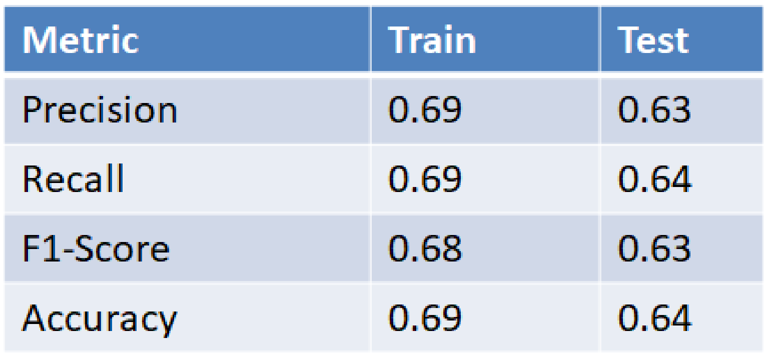
We have 3 subclasses in the target variable. The subclasses are in the ratio of NO : >30 : <30 is to 53% : 35% : 12%. Hence it can be seen that there is class imbalance present in this dataset.

Oversampling was performed to remove this imbalance. As a result the new shape of the dataset is 1,64,000 rows and 26 columns.



Model	Train Accuracy	Test Accuracy
Logistic Regression	47%	45%
Decision Tree	53%	54%
SVC	50%	45%
KNN	63%	61%
Random Forest	69%	62%
XGBoost	69%	64%

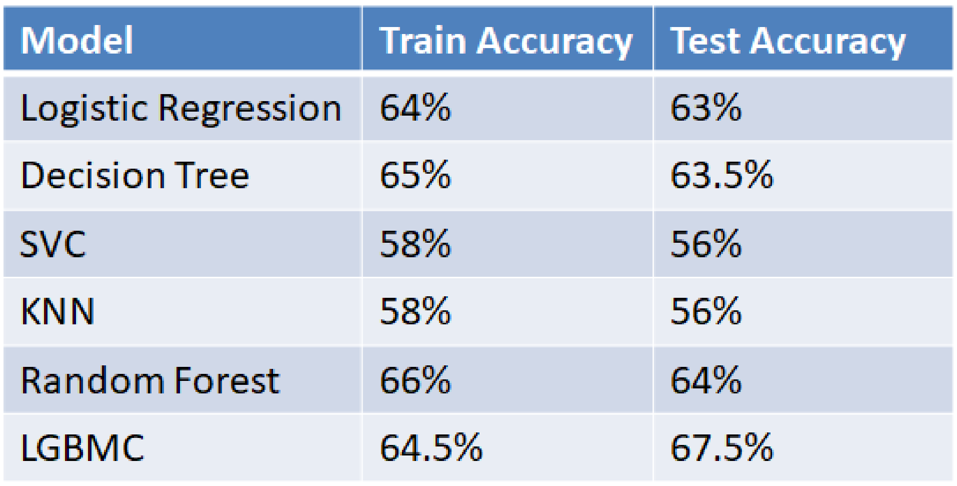
XGBoost has the best accuracy.



Metric	Train	Test
Precision	0.69	0.63
Recall	0.69	0.64
F1-Score	0.68	0.63
Accuracy	0.69	0.64

Scenario 2

We have 2 subclasses in the target variable. The subclasses are in the ratio of NO : Readmitted is to 54% : 46%



Model	Train Accuracy	Test Accuracy
Logistic Regression	64%	63%
Decision Tree	65%	63.5%
SVC	58%	56%
KNN	58%	56%
Random Forest	66%	64%
LGBMC	64.5%	67.5%

LGBM has the best accuracy.

Metric	Train	Test
Precision	0.67	0.64
Recall	0.67	0.65
F1-Score	0.67	0.64
Accuracy	0.67	0.65

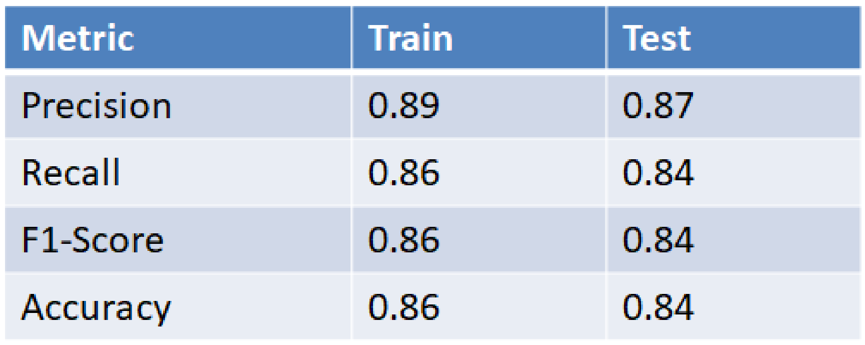
Scenario 3

We have 2 subclasses in the target variable. The subclasses are in the ratio of >30 : <30 is to 70% : 30%. Hence it can be seen that there is class imbalance present in this dataset.

Oversampling was performed to remove this imbalance. As a result the new shape of the dataset is 71090 rows and 26 columns.

Model	Train Accuracy	Test Accuracy
Logistic Regression	66.7%	66.5%
Decision Tree	80%	74.7%
SVC	69.7%	69.4%
KNN	71.3%	63.3%
Random Forest	83%	84%
LGBMC	84.2%	86%

LGBM has the best accuracy.



Metric	Train	Test
Precision	0.89	0.87
Recall	0.86	0.84
F1-Score	0.86	0.84
Accuracy	0.86	0.84

Business Inference

- The Hospital can use our model in scenario 1 to find out whether they should give higher care to a diabetic patient who might get readmitted in the next 30 days to increase their recovery rate.
- The hospital can further use our model in scenario 2 to find out whether a diabetic patient will get readmitted or not to prioritize care.
- If the hospital knows a diabetic patient will get readmitted they can use our model in scenario 3 to find out if the patient will get readmitted before or after 30 days to prioritize care.
- A pharmaceutical company producing metformin can focus on marketing it much more as despite being most effective in type 2 diabetes it is not used by a majority of people.
- A pharmaceutical company can target Hispanics and African Americans as they are more probable to get diabetes.
- A pharmaceutical company can target Hispanics for sale of more insulin.