Predicting the readmission rate of a diabetic patient

Abstract

Readmissions to the hospital raise healthcare expenditures and damage the hospitals' reputation. Early readmission prediction enables patients at high risk of readmission to get immediate and intensive treatment, maximizing the efficiency of the healthcare system and reducing costs. Machine learning makes predictions that are more accurate than those made by conventional methods. In this paper, we built methods for predicting hospital readmission among diabetes patients. The main motivation is to come up with a solution to the readmission problem at the healthcare institutions on a significant level.

1. Introduction

Diabetes is a common chronic condition that is characterized by variable blood glucose levels as a result of insulin-related problems. The number of people with diabetes worldwide has increased from 108 million in 1980 to 422 million now. Diabetes prevalence is rising most quickly in low- and middle-income nations. For instance, the prevalence of Type 2 diabetes increased dramatically in Jordan, where it was 17.1% in 2004 and had increased by 30% during the previous ten years.

When a patient is readmitted to the same department within a specific time frame for the same ailment after being discharged, it is referred to as a "readmission." Numerous factors, including an incorrect initial diagnosis, a recurrence, an untimely release, and others, might result in an inadvertent readmission.

The Centers for Medicare and Medicaid Services now uses the 30-day readmission rate following an index hospitalization as a key hospital performance indicator, and it is coming under more scrutiny as a sign of subpar patient care. The Centers for Medicare & Medicaid Services fined 2610 hospitals a record-breaking amount in 2014 because too many patients had been readmitted to the hospital within a short period of time. Accidental readmission results in recurring medical resource waste in addition to raising the financial burden on the patient.

The methods now used to identify individuals with diabetes who are at risk for complications are subjective. A physician will examine the patient and determine the

best course of treatment. These strategies for predicting readmission have been demonstrated to be somewhat more accurate than arbitrary guesswork. On the other hand, a lot of prediction activities heavily rely on machine learning. Therefore, utilizing machine learning to forecast hospital readmissions seems like a promising implementation strategy.

2. Related work

The readmission rates of diabetic individuals were examined in several published articles. In several researches, machine learning models were employed to forecast the possibility of readmissions for any reason among diabetic patients. Recent work using data from 130 US hospitals brought up good performance across the different proposed models with accuracy and AUC up to 94% [1]. Furthermore, the Random Forest algorithm helped it achieve the best performance. The thorough pre-processing of the data and data balancing with the SMOTE algorithm may be responsible for the high performance. Similar outcomes were obtained in a recent study by [3] using a recurrent neural network technique that provided 0.80 c-statistics with 81.12% accuracy. The advantage of this strategy over Collins' model is the use of a larger, all-age database of 100,000 patients. The length of prior continuous hospital enrollment, which offers a higher level of predictability, is also not differentiated in this dataset. Although 33 out of 56 variables were used for analysis, the data used is outdated compared to other models, and there may be room for improvement in the factor selection procedure.

3. Methodology

First we performed detailed exploratory data analysis to analyze how the dataset is and how the features are distributed. We then preprocessed the data by handling the missing values, outliers, redundant features, duplicate values. Using this preprocessed data, we built the machine learning models to predict the readmission rate.

We used the Naive Bayes model as the baseline model. Naïve Bayes assumes that all the predictors are independent of each other. While is easy and faster to use, it can only deal with numerical features and has its assumptions. Further we built logistic regression, which can only be used when the features are linearly separable. We also built Random Forest and SVM models for the classification purpose.

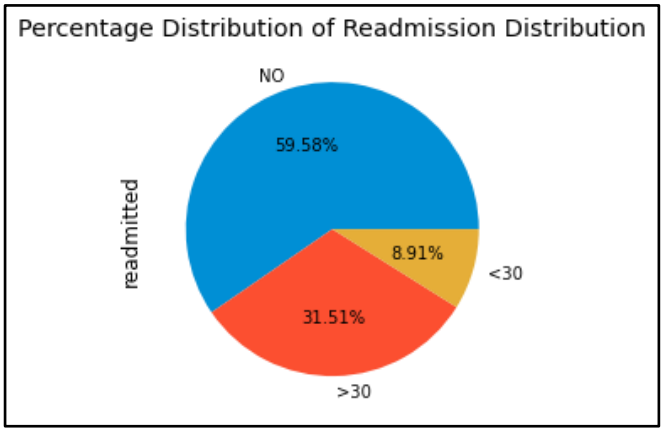
I. Dataset Description

For the project, we used the US Hospital readmission diabetes data set from UCI. The dataset represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes.

- Patient Identifiers
    - 'encounter_id' — A unique identifier for each admission
    - 'patient_nbr' — Unique identifier for each patient
- Patient Demographics
    - 'race', 'gender', 'age', 'weight' — Basic demographic information associated with each patient
    - 'payer_code' — Identifies which health insurance (Blue Cross / Blue Shield, Medicare and self-pay) the patient holds
- Admission and Discharge Details
    - '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.)
- Patient Medical History
    - 'number_outpatient' — Number of outpatient 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_emergency' — Number of emergency visits by the patient in the year prior to the current encounter
- Patient Admission Details
    - 'medical_speacialty' — Identifies the specialty of the physician admitting the patient (cardiology, internal medicine, family/general practice, etc.)
    - 'diag_1', 'diag_2' and 'diag_3' — ICD9 codes for the primary, secondary and tertiary diagnoses of the patient
    - 'time_in_hospital' — Number of days between admission and discharge for the patient
    - 'number_diagnoses' — Total number of diagnoses entered 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
- Clinical Results
    - 'max_glu_serum' — Indicates the results of the glucose serum test
    - 'A1c Result' — Indicates results of the A1c test
- Medication Details
    - diabetesMed' — Indicates if any diabetes medication was prescribed
    - 'change' — Indicates if there was a change in diabetic medications

II. Exploratory Data Analysis and Feature Engineering

We explored the data to understand how the data looks and how each feature is related to the target variable, how the target variable is distributed.



Target Variable Distribution

After dropping the duplicates, the length of the dataset reduced from 101766 to 71518.

In the dataset there are a lot of missing values for different features. We replace all the missing values with a nan representation.

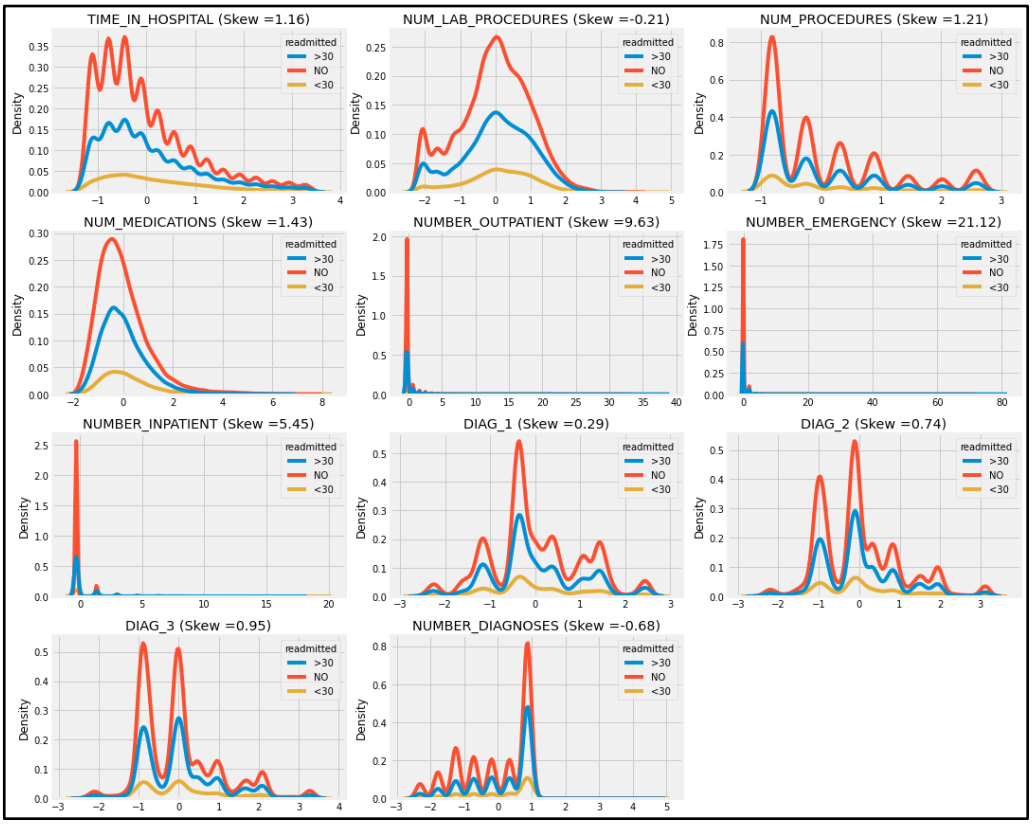
Variable Name	**Null Value Ratio**
Diagnose 1	0.02%
Diagnose 2	0.35%
Diagnose 3	1%
Race	2%
Weight	96%
payer_code	39%
Medical_Speciality	49%

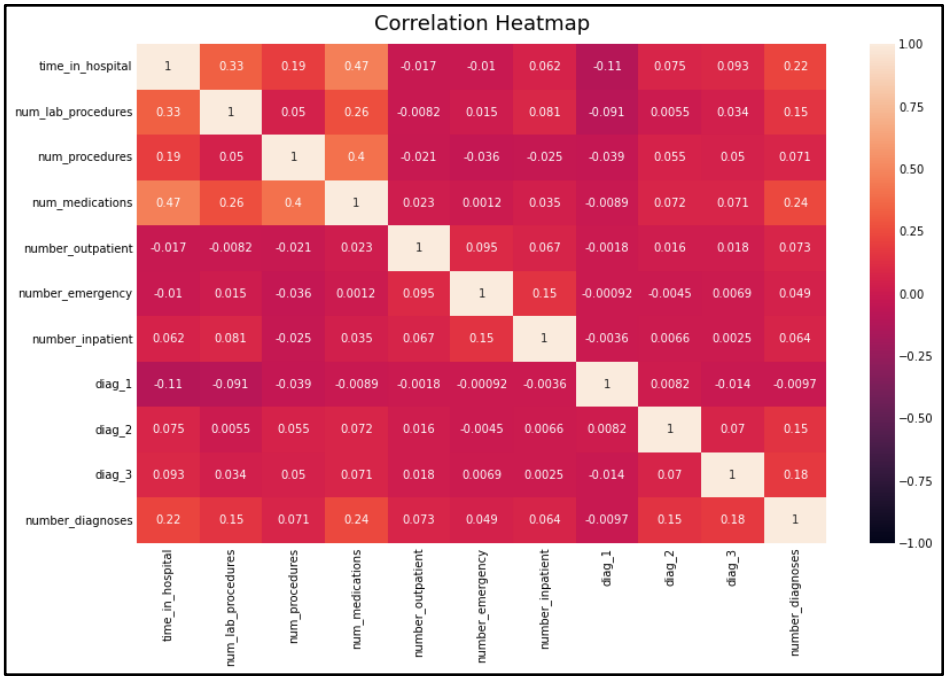
KNN-Imputation was used to impute the missing values (after Scaling the numeric features and encoding the categorical ones). We dropped the columns with more than 40% of missing values. Since race has 2% null values, we dropped the rows where ‘race’ value was null.

```
# knn imputation
imputer = KNNImputer(n_neighbors=5)
df_num_imp = pd.DataFrame(imputer.fit_t
df_num_imp.isnull().sum()
```

```
time_in_hospital      0
num_lab_procedures    0
num_procedures        0
num_medications       0
number_outpatient     0
number_emergency      0
number_inpatient      0
diag_1                0
diag_2                0
diag_3                0
number_diagnoses      0
dtype: int64
```

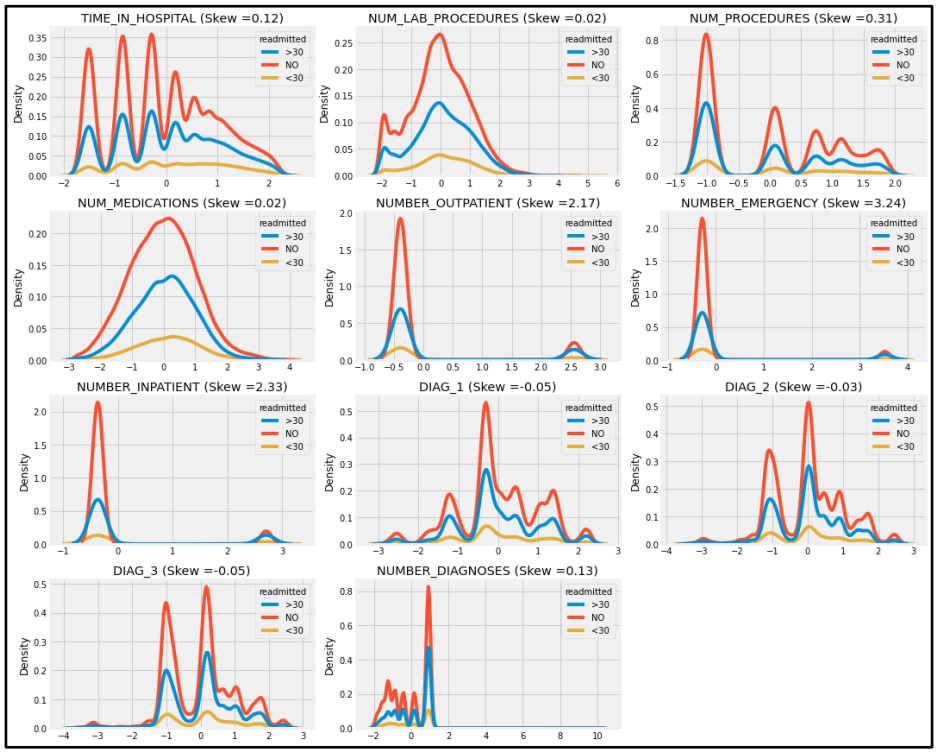
We checked the distribution of the numerical variables. We also checked visually whether there was any multicollinearity between the features using the heatmap. There was no visible multicollinearity . Also since VIF value is less than 2.5, we can clearly say that there is no multicollinearity amongst the numerical variables.

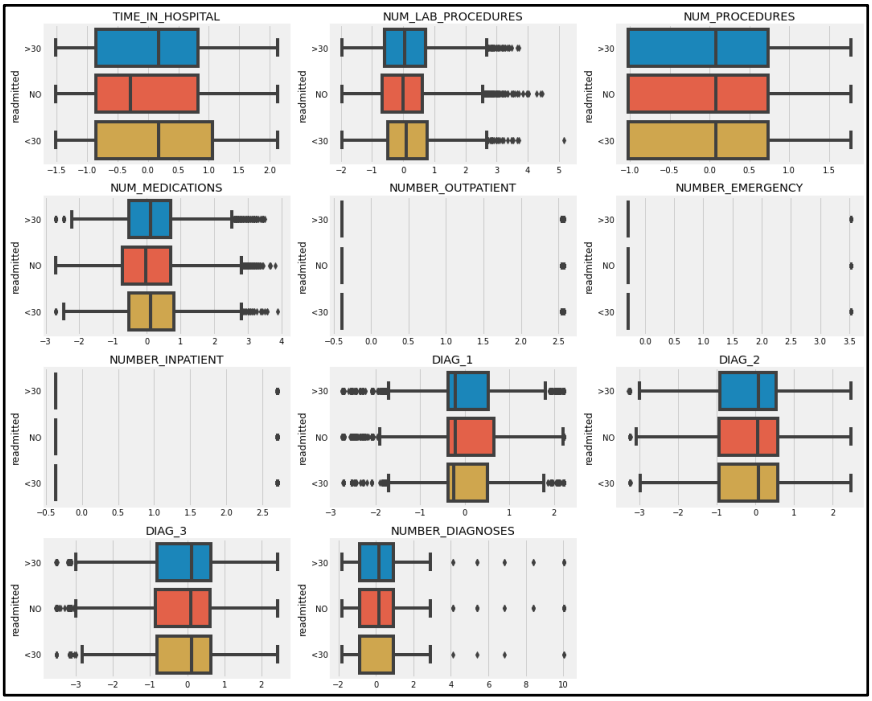




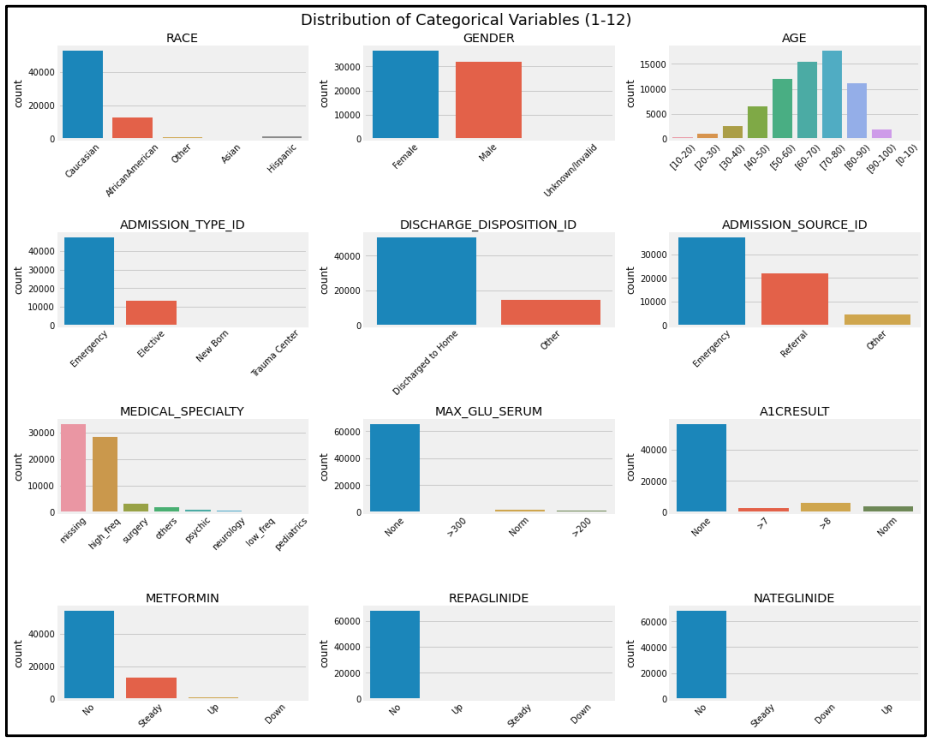
The box-plots say that there is not much difference across the 3 classes of the target variable with respect to the numerical features. Hence we need to build models to identify the features that impact the target variable.

We performed power transformation on the numerical data. This improved the skewness of the data and also eliminated the outliers. The below graph shows the skewness after power transformation.



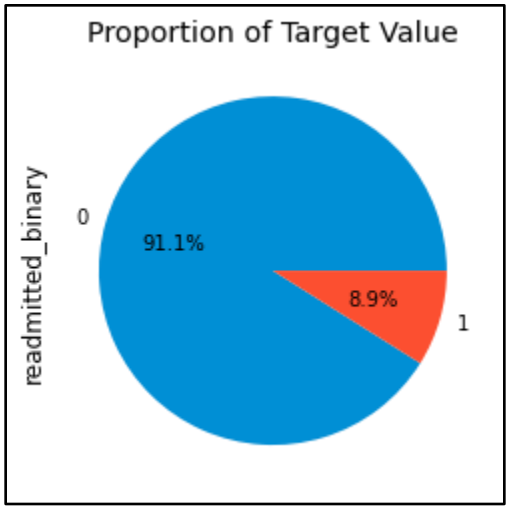


For the categorical data we remapped a few variables according to the data dictionary. In a few cases we reduced the vector size of the variable categories by grouping them. For variables having a huge number of classes, we've classified them into general categories. For example, in the column 'medical_speciality' we reduced the unique values from 73 to just 9.

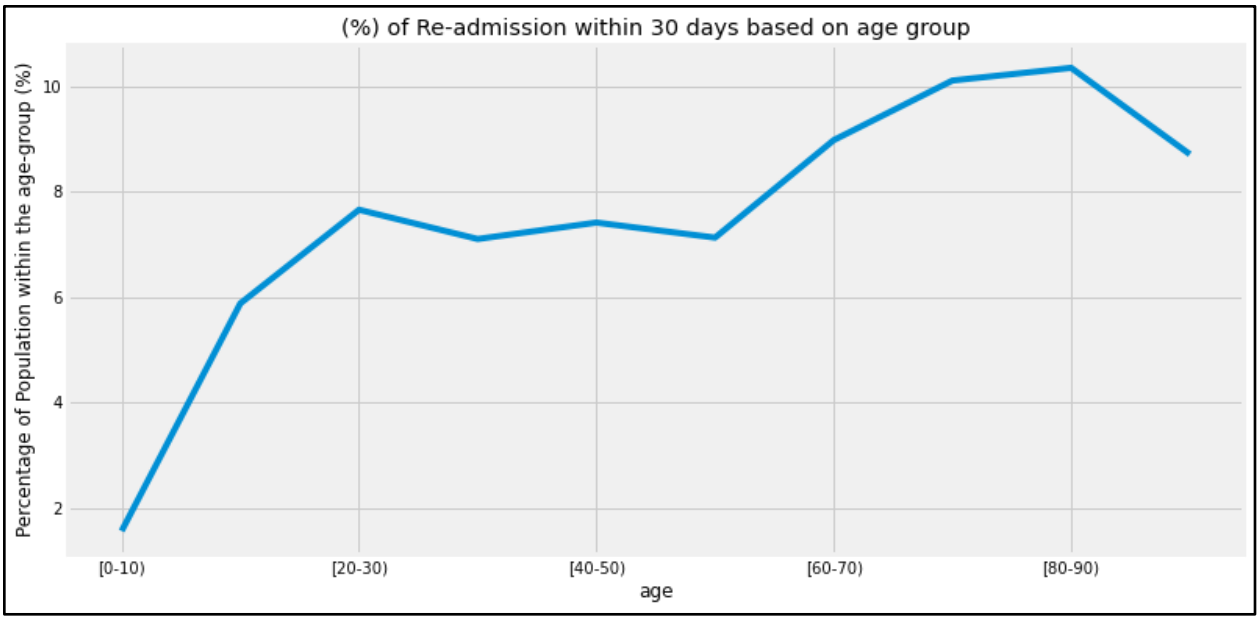


Distribution of few categorical variables

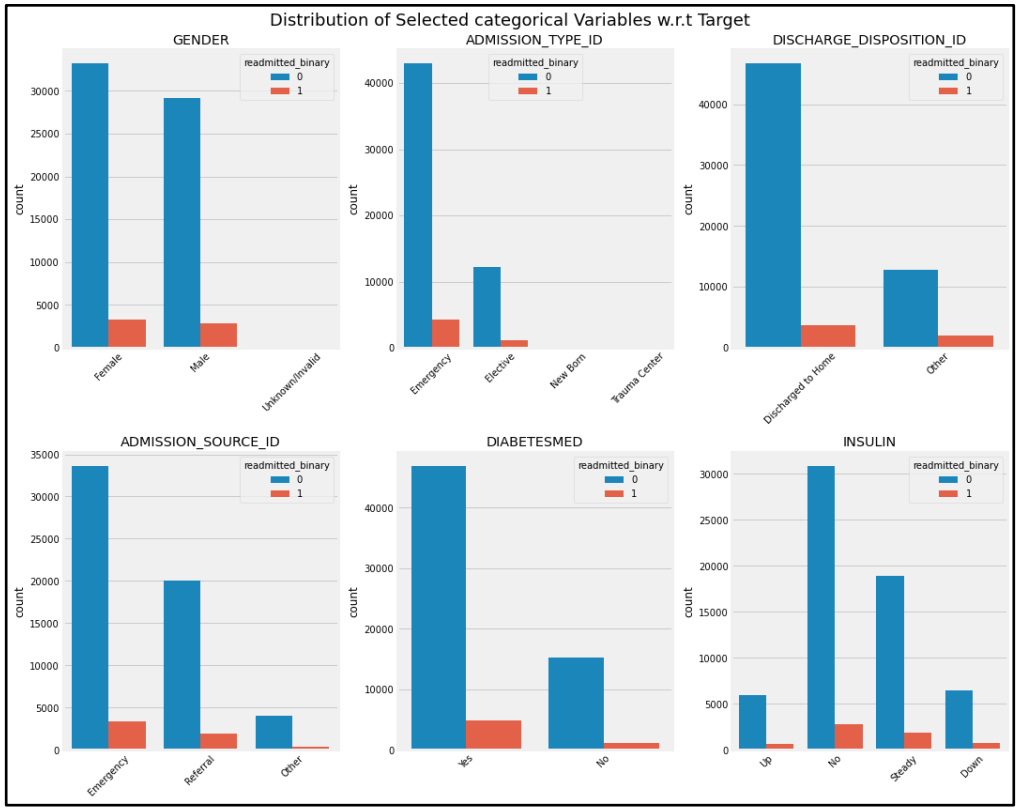
We wanted to make the problem a binary classification problem. So we changed the target variable as 1 if admitted within 30 days otherwise 0. The distribution was a shown



We observed how the target variable was distributed across various features. The below graphs depicts the same distribution



From the below graph we can say that females have a higher proportion of population, but the count of readmittance of Male seems to be equal to that of females. Similarly Discharge_disposition_id may also impact the re-admittance rate. For other variables though not evidently visible, there might still be some relation with the target.



To reduce column count after encoding we can combine categories which have similar % of readmittance. Gender has no impact on re-admittance rate. The variable discharge_disposition has the highest impact. The others have a very slight impact on readmittance rate.

III. Data preparation into train and test sets

Since the data is highly imbalanced, we tried to balance out the data using the SMOTE library. This technique is similar to up-sampling by creating synthetic samples. Here we will use imblearn's SMOTE or Synthetic Minority Oversampling Technique. SMOTE creates fresh, synthetic data using the closest neighbors algorithm that may be utilized to train our model. For our model to generalize effectively to new data, it is crucial to only produce new samples in the training set.

IV. Modeling

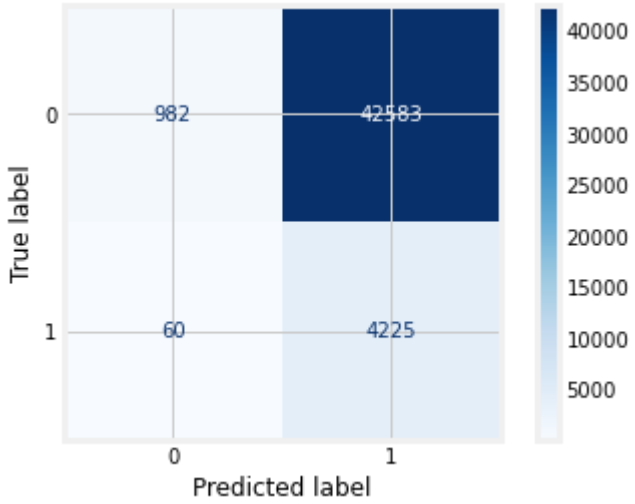
1. Naive Bayes Model (Base Model)

Naive Bayes classifier is a family of simple "probabilistic classifiers" based on Bayes' theorem. It assumes independence between the features. One way to pick the best hypothesis is using the MAP decision rule. We have used Gaussian Naive Bayes to solve the classification problem. Here the values of each class are distributed according to a normal/Gaussian distribution.

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}}\ p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k).$$

$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}}\, e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

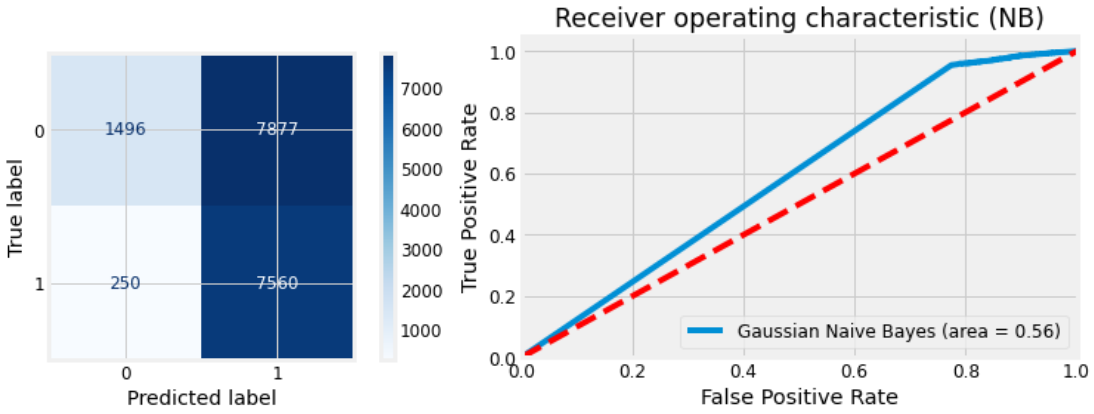
Model assessment before using SMOTE: The training accuracy was 10%



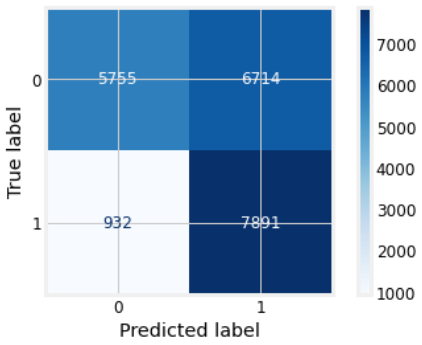
Model assessment after using SMOTE: The training accuracy was 10%

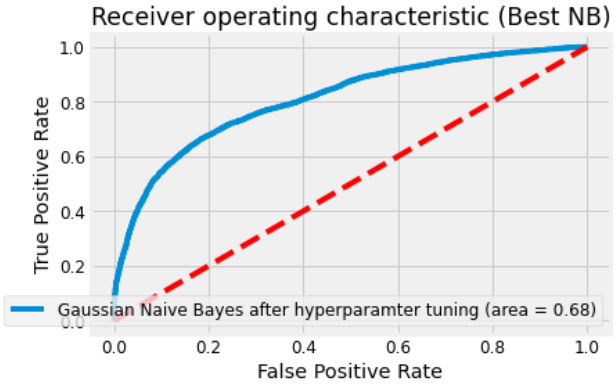
Naive Bayes	Before Tuning	After Tuning
Accuracy	53%	64%
ROC Score	56%	68%

Before Tuning:



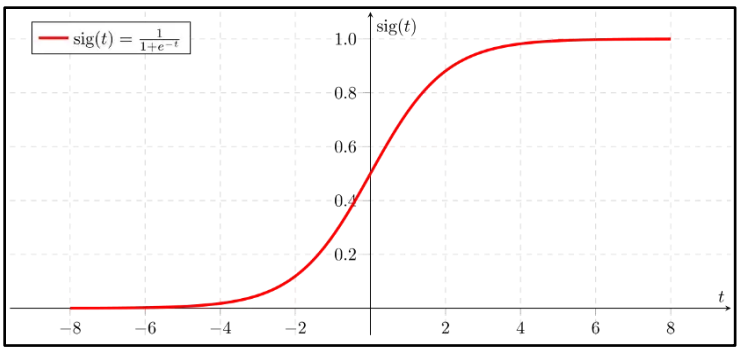
After Tuning:

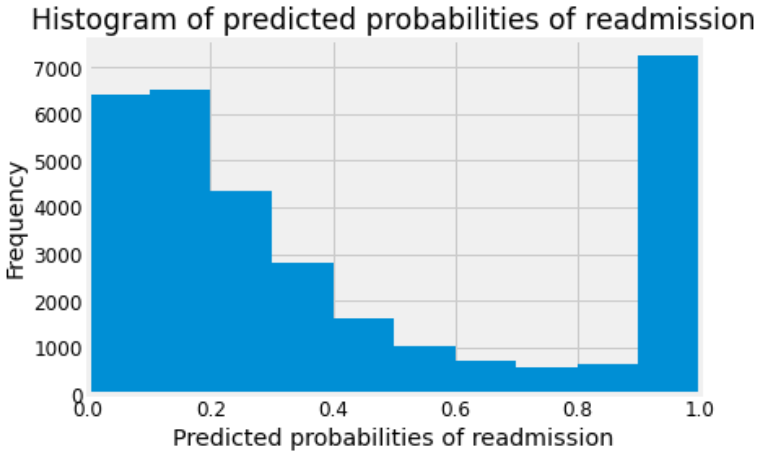




2. Logistic Regression

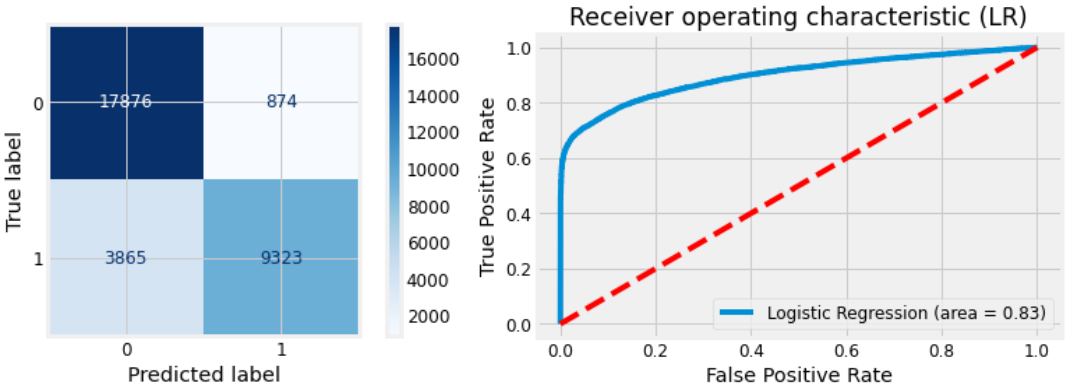
Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome.



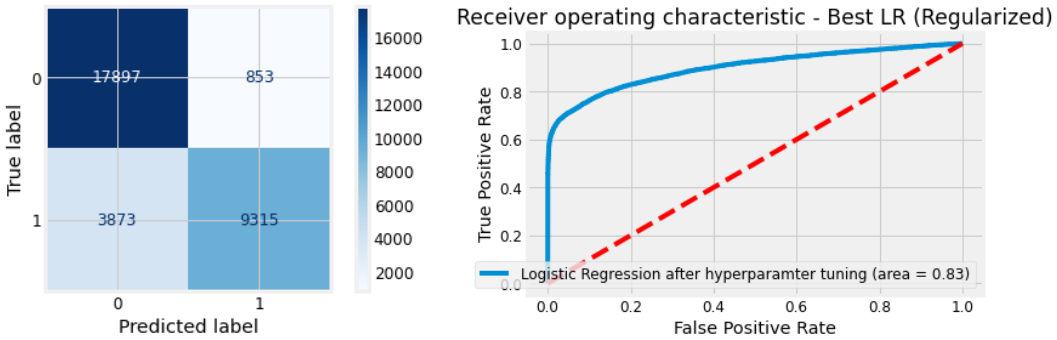


Logistic Regression	Before Tuning	After Tuning
Accuracy	85%	85%
ROC Score	83%	83%

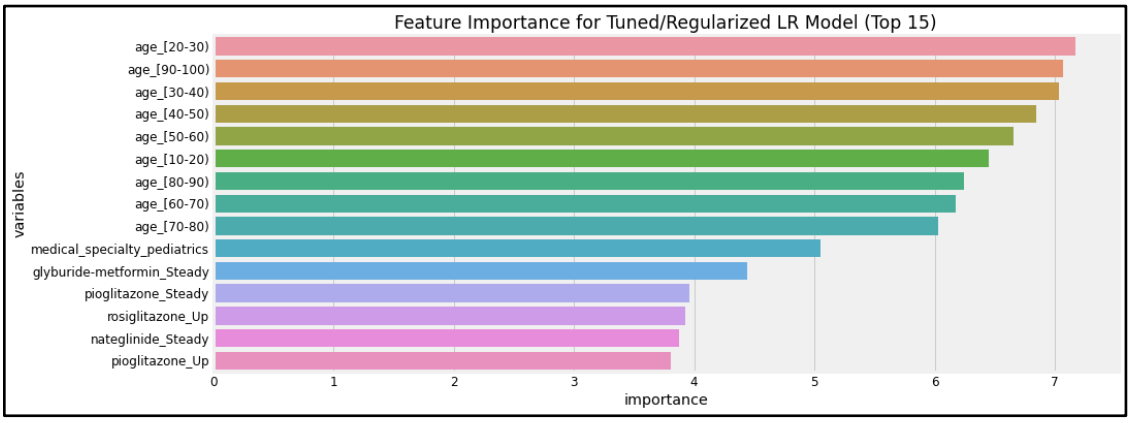
Before Tuning:



After Tuning:



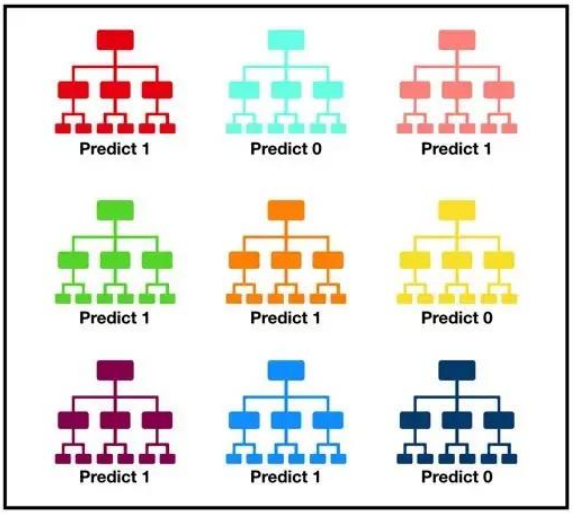
We got the feature importance for the linear regression model. We used bestlogistic_estimator.coef_ and also tried the recursive feature elimination. Both the methods gave us the same top 45 features. Hence using these top 45 features we split the data into train and test again.



After performing feature importance, the accuracy and the ROC score dropped by 2%. But the computation time for the model also reduced by 50%. So feature importance comes with a tradeoff between accuracy and computational time.

3. Random Forest Classifier

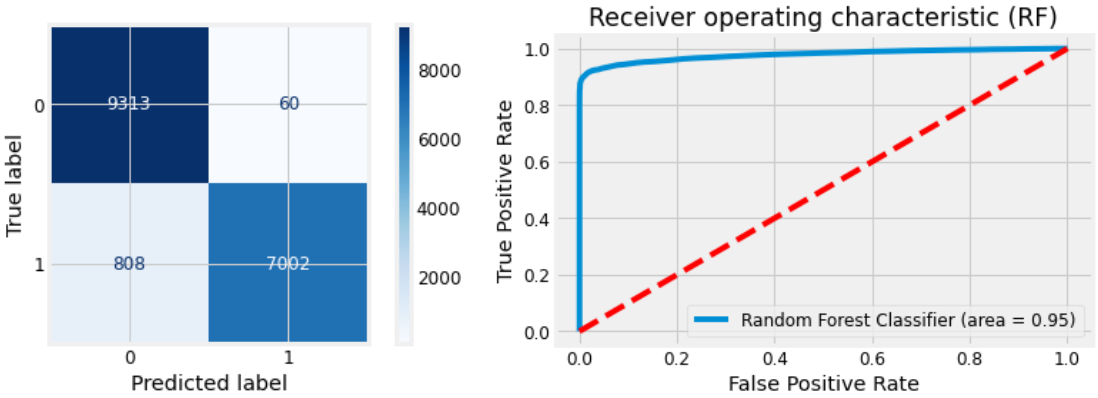
Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.



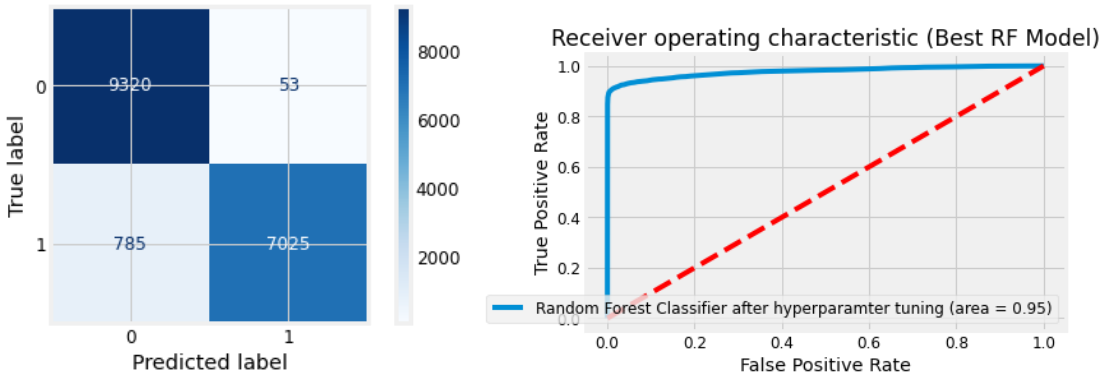
Example to depict the Random forest classifier

Random Forest	Before Tuning	After Tuning
Accuracy	95%	95%
ROC Score	95%	95%

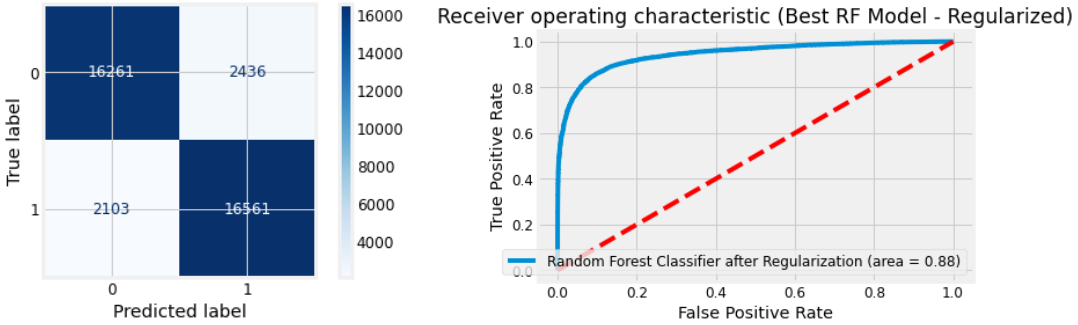
Before Tuning:



After Tuning:



Since there is no major change before and after hyperparameter tuning, we performed regularization



4. SVM

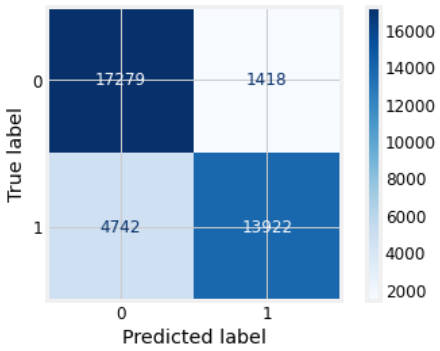
It is a supervised machine learning algorithm that can be used for both classification or regression challenges. In the SVM algorithm, each data item is plotted as a point in n-dimensional space. Then, for classification we find a hyper-plane that differentiates the two classes.

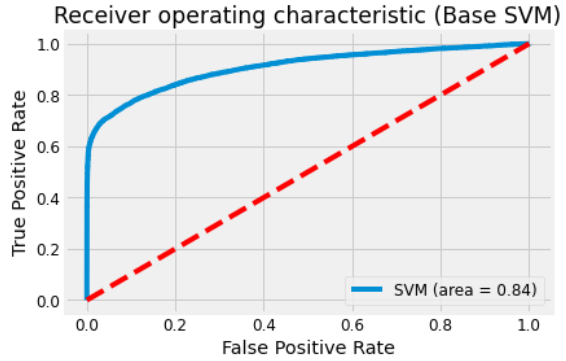
$$\text{minimize } \frac{1}{n}\sum_{i=1}^{n}\zeta_i + \lambda\|\mathbf{w}\|^2$$

$$\text{subject to } y_i\left(\mathbf{w}^\mathsf{T}\mathbf{x}_i - b\right) \geq 1 - \zeta_i \text{ and } \zeta_i \geq 0, \text{ for all } i.$$

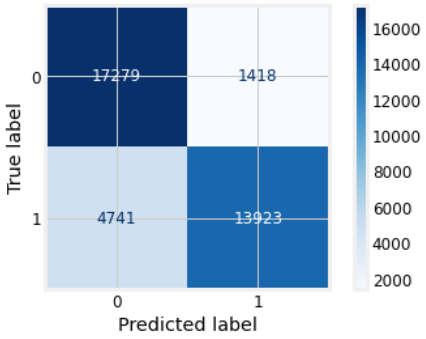
SVM	Before Tuning	After Tuning
Accuracy	83%	84%
ROC Score	84%	84%

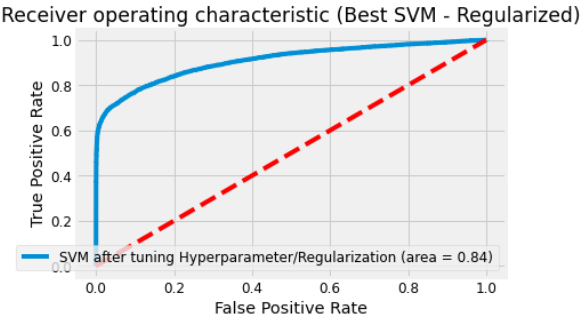
Before Tuning:



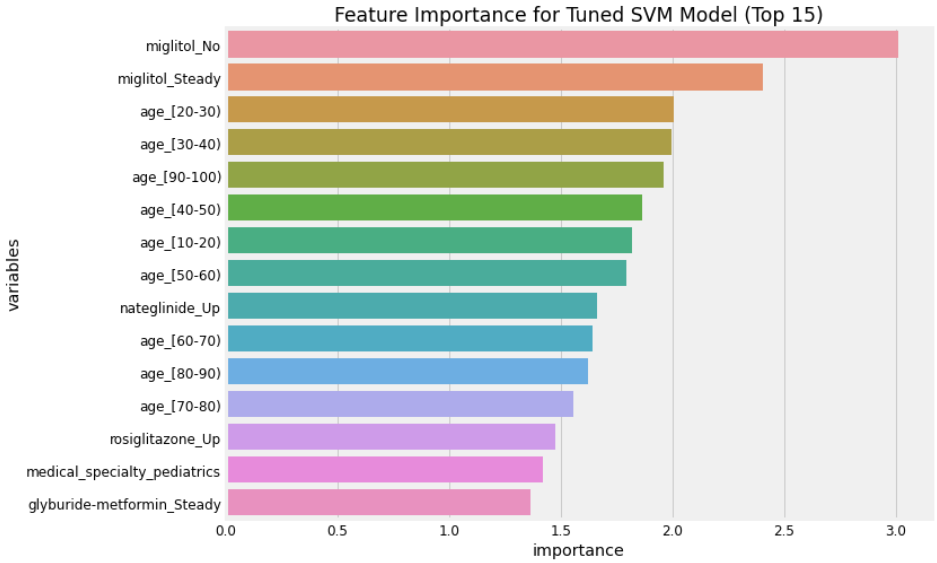


After Tuning:



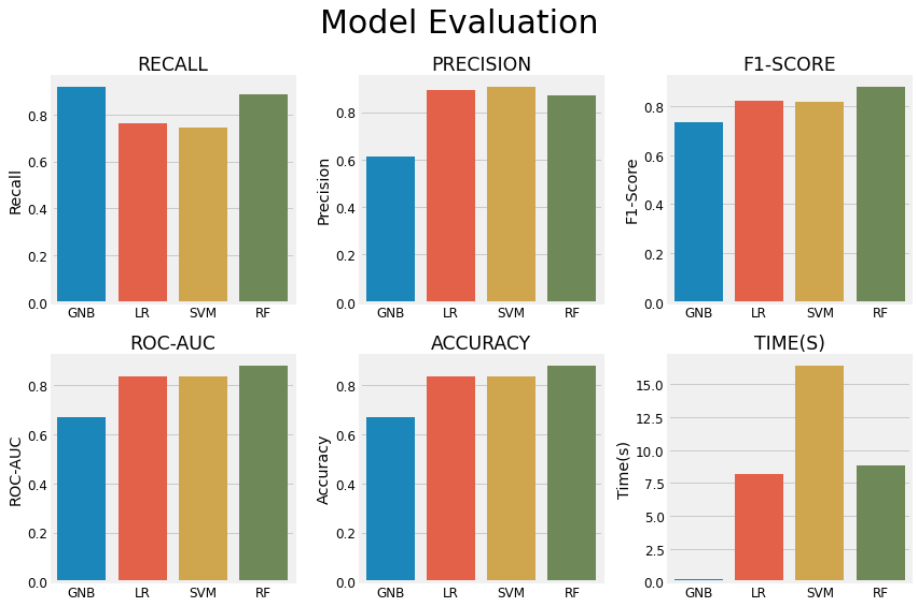


Most important features for the best SVM model are:



5. Model Comparison

	Recall	Precision	F1-Score	ROC-AUC	Accuracy	Time(s)
Gaussian Naive Bayes	0.919096	0.613168	0.735592	0.670143	0.669923	0.202185
Logistic Regression	0.761412	0.894561	0.822634	0.835913	0.835979	8.193740
Support Vector Machine	0.745928	0.907562	0.818845	0.835043	0.835122	16.397628
Random Forest	0.887323	0.871769	0.879477	0.878517	0.878510	8.845504



Top 15 feature for each model are as follows:

	Top Features (LR)	Top Features (SVM)	Top Features (RF)
1	age_[20-30)	miglitol_No	time_in_hospital
2	age_[30-40)	miglitol_Steady	number_inpatient
3	age_[90-100)	age_[20-30)	number_diagnoses
4	age_[40-50)	age_[30-40)	diag_3
5	age_[10-20)	age_[90-100)	num_medications
6	age_[50-60)	age_[40-50)	age_[50-60)
7	age_[60-70)	age_[10-20)	age_[40-50)
8	age_[80-90)	age_[50-60)	diag_2
9	medical_specialty_pediatrics	nateglinide_Up	diag_1
10	age_[70-80)	age_[60-70)	gender_Male
11	rosiglitazone_Up	age_[80-90)	num_procedures
12	glyburide-metformin_Steady	age_[70-80)	num_lab_procedures
13	nateglinide_Up	rosiglitazone_Up	age_[60-70)
14	rosiglitazone_Steady	medical_specialty_pediatrics	admission_source_id_Referral
15	nateglinide_Steady	glyburide-metformin_Steady	age_[70-80)

4. Conclusions

Throughout the confusion matrix it is evident that the True Positives have increased from Naive Bayes model to SVM model. This shows that our model is working without any bias, as the data is balanced. Moreover, It is important for us to find the model which has the right amount of accuracy and F1 score. Because if we just select a model based on the F1 score we would select Random Forest Classifier, but if we have to select a model just based on accuracy we would select SVM. But that is not the case, the class of utmost importance to us is 1(readmission). Out of all the linear models SVM has the best performance. If computational time is a critical factor then we could also go with Naives Bayes, however there's a significant trade-off with performance. Another option would be to feature selection and then LR or SVM. Regularized RF outperforms all other models but again that is to be expected since it's an ensemble mode

5. References

1. .C.-Y. Lin, H. S. Singh, R. Kar, and U. Raza, "What are Predictors of Medication Change and Hospital Readmission in Diabetic Patients?," Berkeley, 2018.
2. Ti'jay Goudjerkan, Manoj Jayabalan "Predicting 30-Day Hospital Readmission for Diabetes Patients using Multilayer Perceptron" International Journal of Advanced Computer Science and Applications, Vol. 10, No. 2, 2019
3. C. Chopra, S. Sinha, S. Jaroli, A. Shukla, and S. Maheshwari, "Recurrent Neural Networks with Non-Sequential Data to Predict Hospital Readmission of Diabetic Patients," in ICCBB 2017 Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics, 2017, pp. 18–23.