SPRINGBOARD DATA SCIENCE COURSE

CAPSTONE PROJECT 1

Predicting the likelihood of hospital readmission for Diabetes Patients

Abhishek Sukhadia

12th April, 2020

Contents

1. Introduction

As per Wikipedia, hospital readmission refers to an “episode when a patient who had been discharged from a hospital is admitted again within a specified time interval.” These readmission rates are used as a metric to measure quality of health services in any country. The most common time frames used are 30-day, 90-day or 1 year.

Hospital Readmission rates were made part of (Patient Protection and Affordable Act, 2010 in the United States. Under this act, a hospital gets penalized if they have higher than expected readmission rates. Thus it has become an important measure for hospitals to monitor and reduce their readmissions. It has become an important indicator of hospital quality and affects the cost of care adversely. Being able to determine factors that lead to higher readmissions, and correspondingly being able to predict which patients will get readmitted can help hospitals save millions of dollars while improving quality of care.

The most important stakeholders are Government Healthcare regulatory authorities, Hospitals and Patients. Center for Medicare & Medicaid Services (CMS), a federal agency within the United States Department of Health and Human Service, has established a Hospital Readmissions Reduction Program. This project sees direct application to such programs run by government agencies. Hospitals can directly use the results of this project to identify factors that are causing high readmission rates and thus concrete action to reduce the same. This would help them save penalties and thus improve the quality of care for the patients. Currently CMS has included 6 diseases in this program and they are regularly adding new disease conditions to the list. I have taken diabetes as the disease to predict readmissions even though it is not part of the CMS list.

In US, diabetes affects nearly 34.2 million Americans - just over 1 in 10 have diabetes (National Diabetes Statistics Report, 2020) and this number is expected to grow substantially every year. It’s the fifth leading cause of death in America, more than breast cancer and AIDs combined. The American Diabetes Association released new research on March 22, 2018 estimating the total costs of diagnosed diabetes have risen to $327 billion in 2017 from $245 billion in 2012, when the cost was last examined. For the cost categories analysed, care for people with diagnosed diabetes accounts for 1 in 4 health care dollars in the U.S., and more than half of that expenditure is directly attributable to diabetes. This proves that there is huge burden both on patients and hospitals on the growing costs and readmissions contribute a significant portion of this burden. Reducing readmission rates of diabetic patients has the potential to greatly reduce health care costs while simultaneously improving care.

2. Data Acquisition and Cleaning

I have used data available on Kaggle which was in turn sourced from UCI repository which contains de-identified diabetes patient encounter data for 130 US hospitals (1999–2008) containing 101,766 observations over 10 years. The dataset has over 50 features including patient characteristics, conditions, tests and 23 medications.

Each row is a single encounter of a patient with details given below.

Patient Characteristics:

- Encounter ID, Patient Number, Race, Gender, Age, Weight

Patient Hospital Admission and Discharge Details:

- Admission Type, Discharge Disposition, Admission Source, Time in Hospital, Payer Code, Medical Specialty

Patient Encounter Details:

- Number of lab procedures, procedures other than lab tests, medications, outpatient visits, inpatient visits, emergency visits

Patient Medical Diagnosis Details:

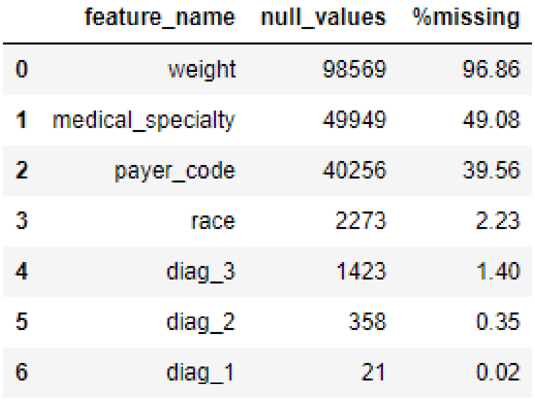
- Primary, secondary and additional secondary Diagnosis, Number of diagnoses, A1C result, Glucose serum test result,

Patient Medications:

- change of medications, diabetes medications, 24 features of drug medications (generic names)

Details about each field can be found in the research article "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records".

The above data was downloaded as a CSV and then imported in python. More details of this can be found in this **IPython Notebook**. Post importing, data was checked for missing values which were present as "?" in the data set. Table 1 is the summary of the missing values.



	feature_name	null_values	%missing
0	weight	98569	96.86
1	medical_specialty	49949	49.08
2	payer_code	40256	39.56
3	race	2273	2.23
4	diag_3	1423	1.40
5	diag_2	358	0.35
6	diag_1	21	0.02

Table 1: Features with missing values

Missing values were handled differently for each feature.

- **Weight** have ~97% missing values. There is poor interpretability of the missing values, so it is best to drop this column
- **medical_speciality** and **payer_code** have 40-50% missing values. I have decided to drop it but there are ways to deal with it. Alternatively, if these columns are not dropped then, it is best to create a separate category of 'missing'.
- **race, diag_3, diag_2, diag_1** have only <=2% missing values, so missing rows were dropped for diagnosis and replaced with mode of the column values for race.
- A missing first diagnosis while data have diagnosis second and third, is also a bad data. But second and third diagnosis can give us some idea about the readmission. So we can either keep this missing values under "missing" category or just ignore the rows. I am going with first one.
- In **gender**, there are values like "Unknown/Invalid" which are missing. These columns are also dropped

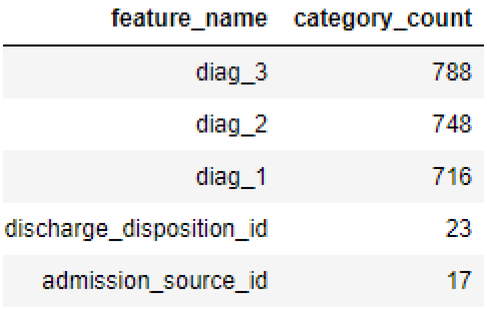
Apart from removing missing values, data required further cleaning based on domain knowledge of readmissions.

We ignored patients who died post hospital admission since they have zero probability of readmission and thus would bias the data. I also dropped two columns where all records have same values (examide and citoglipton) since it cannot provide any information about readmission. More details about the data cleaning process can be found in this **IPython Notebook**. The cleanest data is ready for data wrangling.

3. Data Wrangling

3.1 Data Wrangling

This data sets contains categorical data with number of categories ranging from 2 to 788. The features with high categories can increase the dimensionality of the data and make is sparse. So it was decided to reduce the number of categories through grouping. Table 2 shows the features identified for grouping of categories.



feature_name	category_count
diag_3	788
diag_2	748
diag_1	716
discharge_disposition_id	23
admission_source_id	17

Table 2: Features with high category count

Diagnosis columns (diag_1, diag_2 and diag_3) contain numbers that are ICD9 Codes describing specific diseases. They have been clubbed into larger categories in accordance with ICD9 to reduce the categories.

For discharge disposition, admission type and admission source, ~90% of data falls under 3-4 categories (Figure 1). Thus we can club the remaining categories with the existing category or under others. This would help reduce the noise from the data. Attempt was made to group items based on similar distribution of target variable and same feature characteristics like "urgent and Emergency", "Not Applicable, Not Available, Missing" etc (Figure 2). More Details of the grouping is available in this **IPython Notebook**.

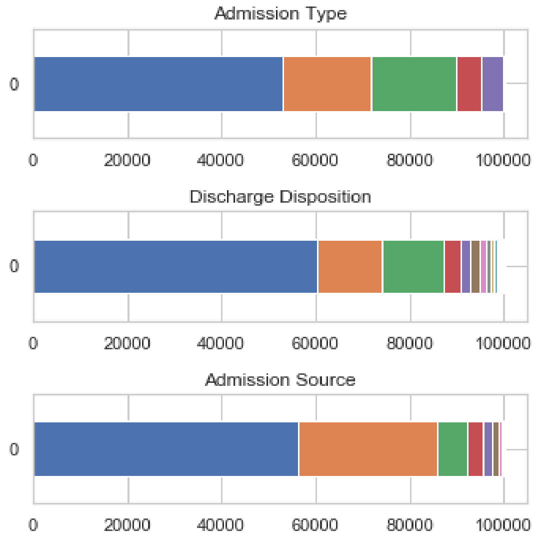


Figure 1: Pre-Grouping Hospital Admission Features

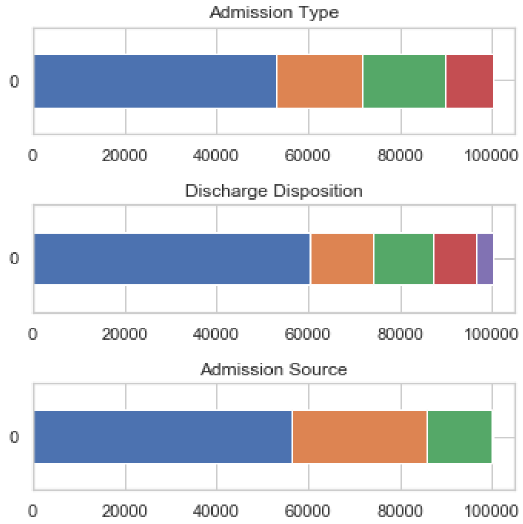


Figure 2: Post-Grouping Hospital Admission Features

By looking at the individual patient's id, it was observed that many of the patients have had multiple inpatient visits post readmission. This implies there are non-unique encounters for patient. This means, those observations are not statistically independent. The patient next visit data might get affected based on the patient's previous visits. Since it violates one of the assumptions of logistic regression, I have created a new feature column to deal with this multiple inpatient visits. The new column would be the 'visit_num' defined as the visit number for that patient i.e. whether it is his/her first visit, second visit, etc. The rank is determined by the ascending order of the encounter_id. The groups made were 1, 2, 3, 4-30 and > 30. Post this grouping patient and encounter index were dropped. More Details of the grouping is available in this **IPython Notebook**.

3.2 Distribution of Numeric Variables

The numerical variables in the data set are all discrete count data. Figure 3 shows the distribution of all discrete variables in the dataset. All the variables shows long tailed distribution skewed towards either right or left.

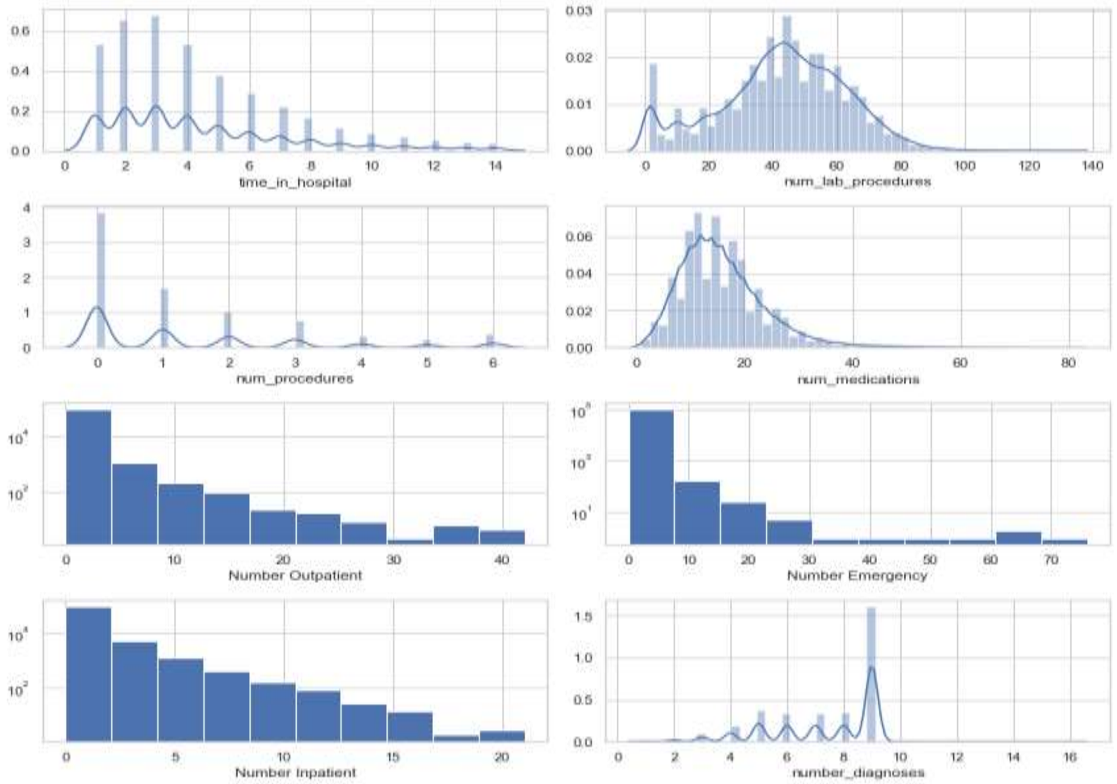


Figure 3: Distribution of Discrete Numeric Variables

4. Exploratory Data Analysis

In the cleaned data, there are now 96551 patient records, 44 features/columns and one target variable. We will go through most of the features in the dataset to explore the relation with the readmission rate.

The target variable for this project is called “readmitted” which contains three values:

- “<30” : Person gets readmitted within 30 days (0)
- “>30”: Person gets readmitted after 30 days (1)
- “NO”: Person doesn’t get admitted (2)

4.1. Hospital Readmission Rates

The distribution of the target variable in **Figure 4** shows that our dataset is highly imbalanced. The distribution of “<30” (class of interest) is just 10.7% compared to remaining two classes. This imbalance needs to be kept in mind while applying our machine learning algorithms. It would also require special treatment at many places.

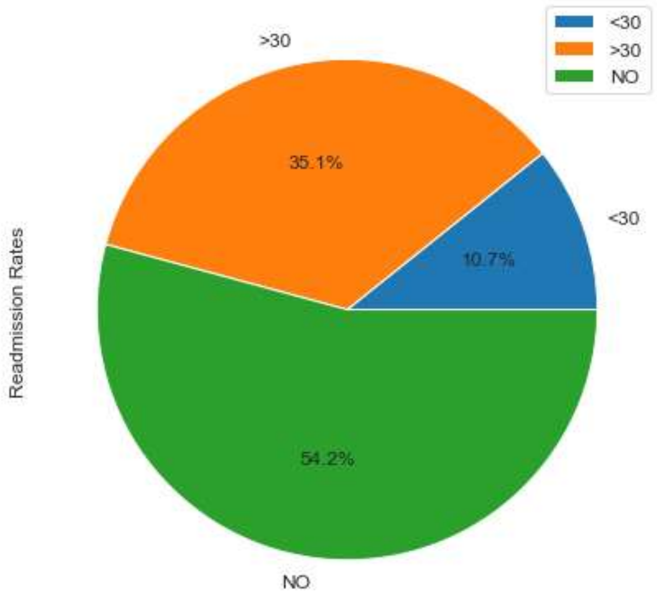


Figure 4: Distribution of Target Variable – Readmitted

We define the readmission rates as:

$$\text{Readmission rates} = \frac{\text{A person getting readmission within} < 30 \text{ or} > 30 \text{ days}}{\text{Total Number of Patient for a given scenario}}$$

Thus we will transform this multi-class problem to binary class.

Readmission would depend on lot of factors like type of medicine prescribed, patient medical conditions, patient age, race, results of laboratory tests, time in hospital, whether patient is visiting for the first time or multiple times, etc. In the following subsections, we will go through many of the above mentioned factors and explore the distribution of our readmission rates. As per Section 2, there are broadly 5 categories of information that are embedded in most of the fields.

1. Patient Medications
2. Patient Characteristics
3. Patient Hospital Admission and Discharge info
4. Patient Medical Diagnosis
5. Patient Encounter Details

Before we deep dive into above categories, we will look at one of the important feature that would affect our readmission rates.

4.2. Visit Number

A person admitting for the first time, his/her visit number would be 1. Post first readmission, if he gets readmitted again or visit the hospital, his/her visit number will then be 2. Thus visit number is the number of visit to hospital made by the patient post first readmission. This feature was created as per Section 3.1. ~30% of my data has visit number as >1.

On comparing visit number with readmission rates, we see that as the visits of the patient increases to the hospital, his/her patient readmission chances also increases (Figure 5). Common sense would say that after a particular number of visits, readmission rate will be zero either because patient gets died or gets cured or gets transferred to other hospitals. During our data cleaning and outlier analysis, we have removed all such cases. So that would not be visible in our trend.

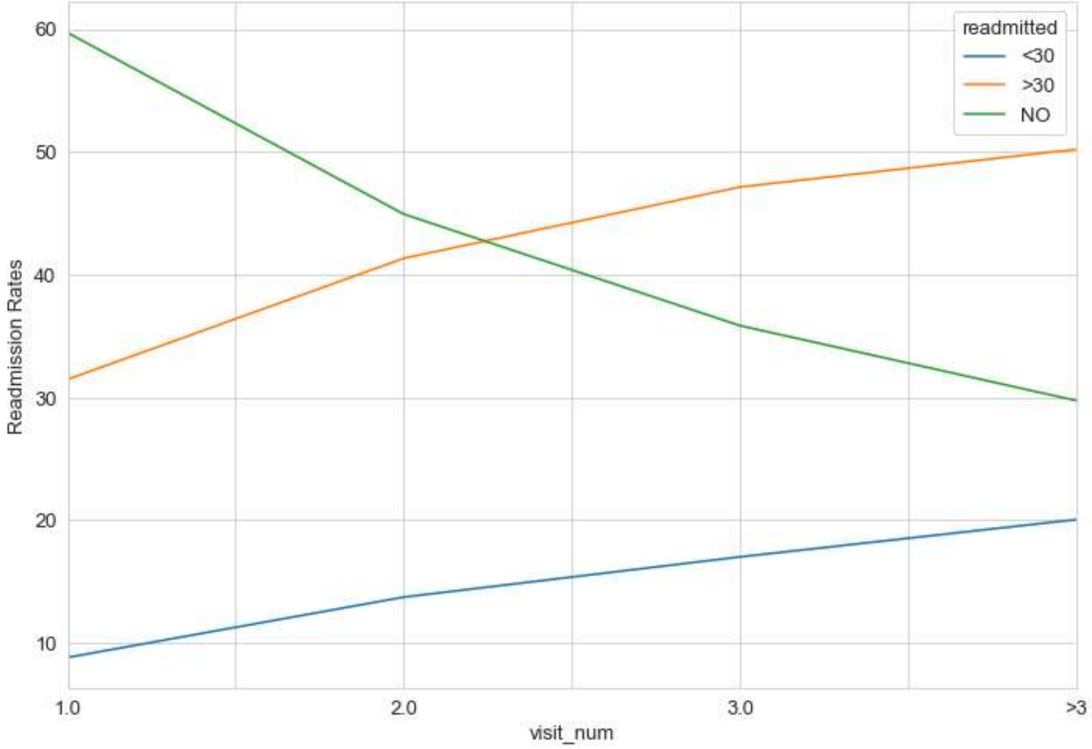


Figure 5: Readmission rates of different visit numbers

4.3. Patient Medications

Medications (21 Features)

For the 21 features, exploratory analysis in Figure 6 shows that there were few medicines not prescribed to 99.9% of the patient. This observation told us that even if we remove those features, it would not affect my target variable distribution. I didn't those features and left it for the L1 Regularization to figure it out. More Details is available in this **IPython Notebook**.

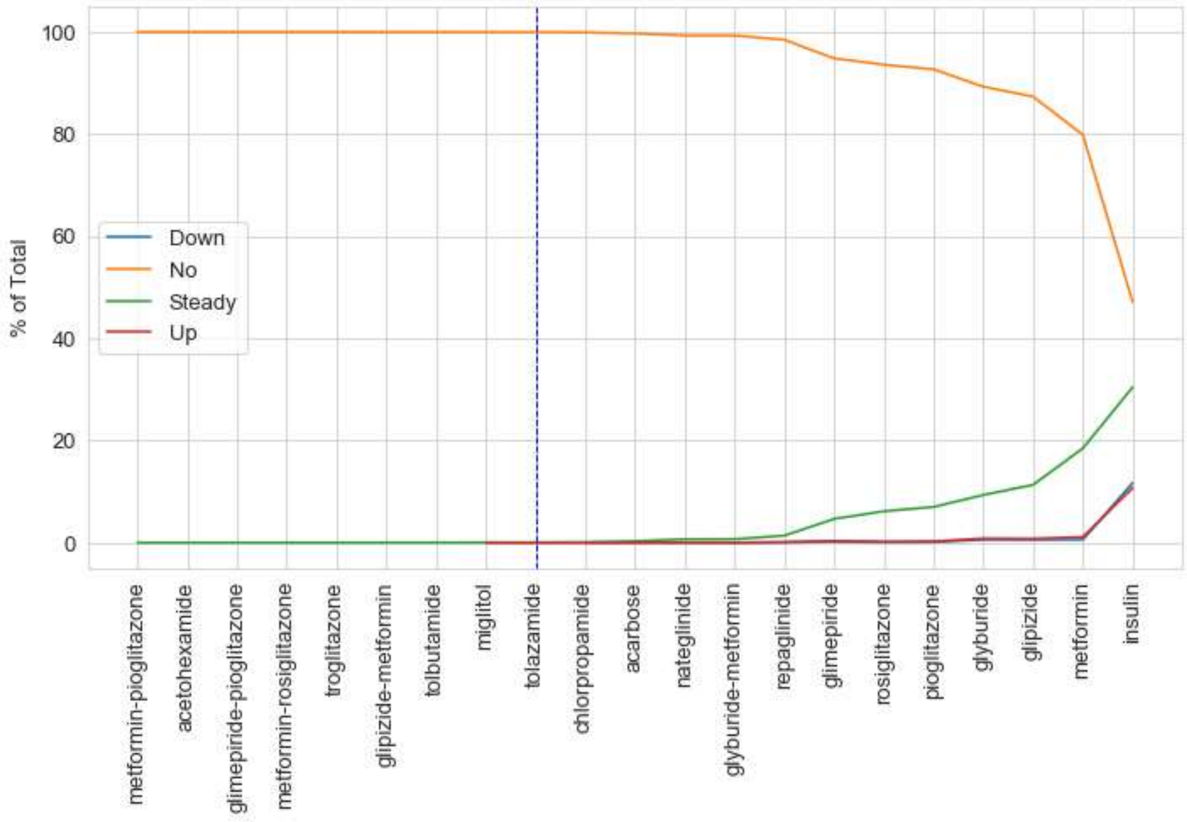


Figure 6: Medications and their prescription changes

Change of Medications

Patients whose medications were changes have higher chance of getting readmitted (Figure 7). Possible reasons may be wrong medicines prescribed, experimenting with medicines, wrong dosage of medicines prescribed, and person's immunity to the medicines.

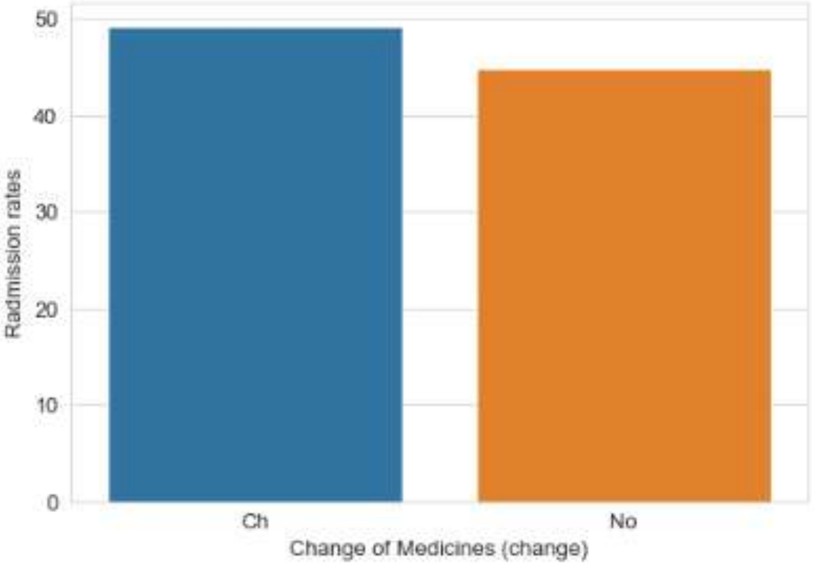


Figure 7: Change of Medicines affecting Readmission rates

4.4. Patient Characteristics

Race, Age and Gender

~77% of the patients are from Caucasian Race. This is as expected because the data set is from United States. Almost 90% of the patients more than 40 years of age, this implies that diabetes occurs in patient above the age of 40 years (Figure 8). The ratio of female and male is approximately uniform.

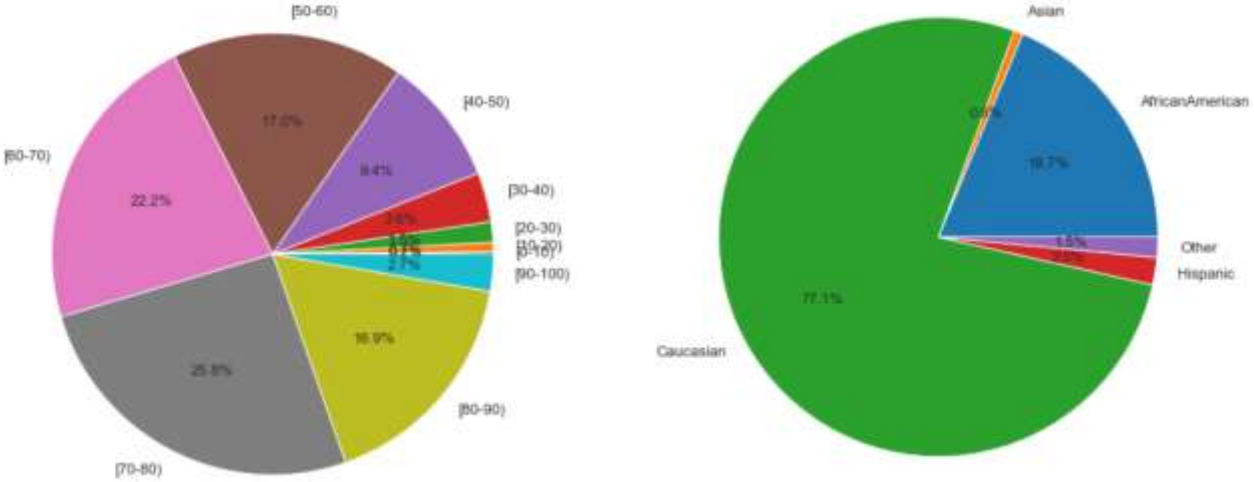


Figure 8: Age and Race Distribution of Patients

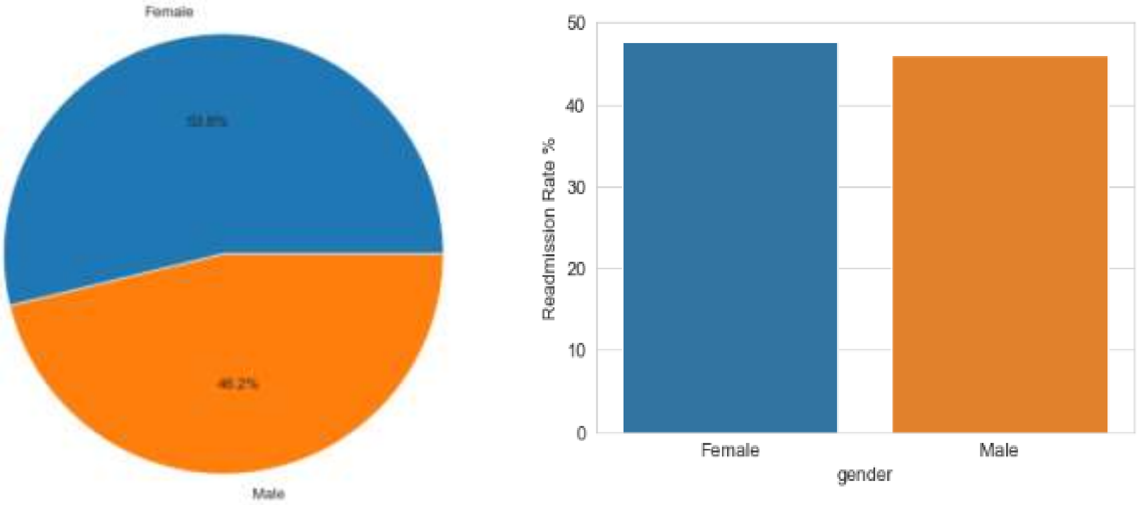


Figure 9: Gender Distribution of Patients and its relation with readmission rates

In

Figure 9, we observe that gender doesn't play any major role in determining readmission rates. On exploring relation of race with readmission we found that, Caucasians and AfricanAmerican are at highest risk of being readmitted. With the exception of 20-30 age category, we see in Figure 10 that as age increases, chances of readmission also increases. Thus old people are more likely to get readmitted.

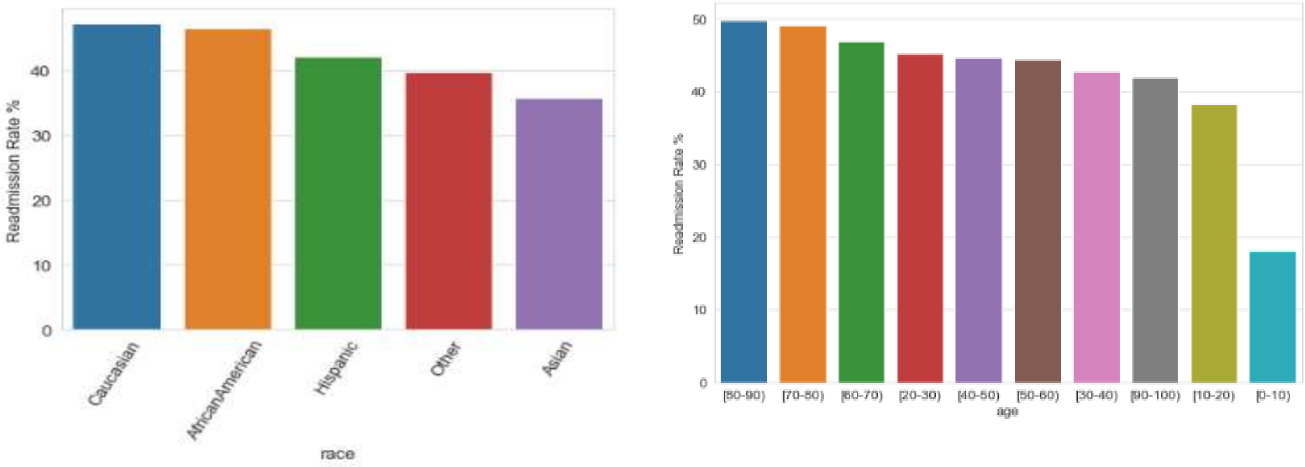


Figure 10: Readmission rates for age and race features

4.5. Patient Admission and Discharge Information

This information consists of reason for admission, admission source and how was the patient discharged. All the above three features affect the readmission rate because they may contain indirect information regarding the patient conditions at the time of admission and discharge. For example, a person coming for elective surgery would have a lesser chance of getting readmitted than a person coming as emergency. Similarly, a person getting discharged to home mean fully recovered compared to discharge to other facilities where he/she may not have recovered fully.

Exploratory analysis suggested that an Emergency Patient has high chance of getting readmitted compared to other reasons. In Figure 11, 7 and 1 represents emergency. ID mappings are available in the dataset. For discharges, a person getting discharged to home/hospice (Id = 1) have less chance of readmission compared to other categories (Figure 12).

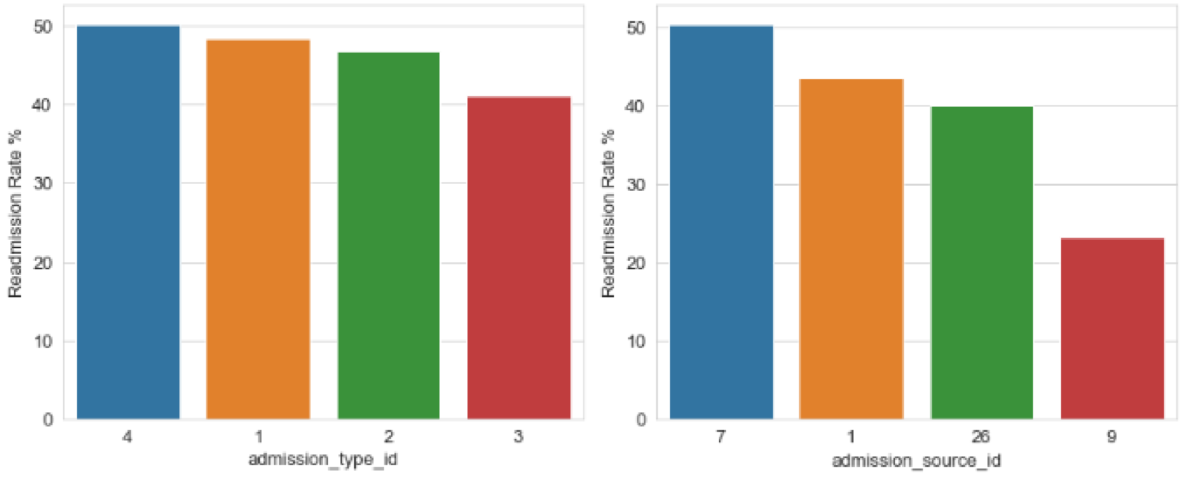


Figure 11: Admission Type and Source Readmission Rate

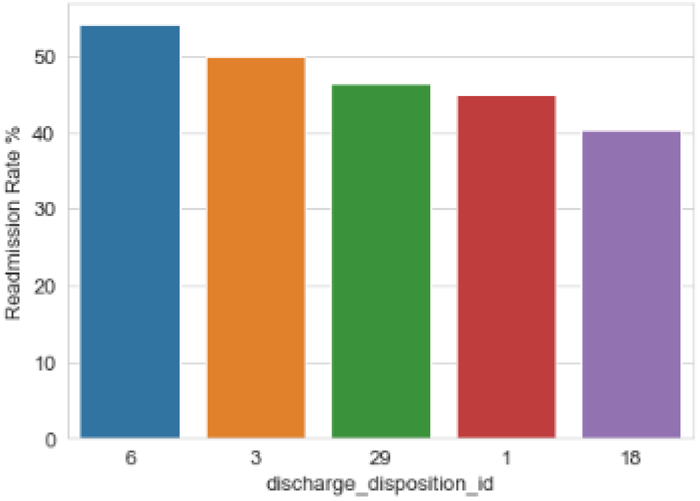


Figure 12: Readmission Rates for Discharge Disposition

4.6. Patient Medical Diagnosis

Glucose Serum and A1C Tests

Since we are dealing with diabetes patients, we will first look at the Glucose serum and A1C test results. Both these tests are performed to diagnose diabetes. A person with diabetes will show values of >200 in glucose test and >6% in A1c results. As both the value increases, intensity of diabetes also increases. Figure 13 and Figure 14 shows that diabetes patients whose lab results are >300 in glucose serum test and no A1C test was performed, have higher chance of readmission.

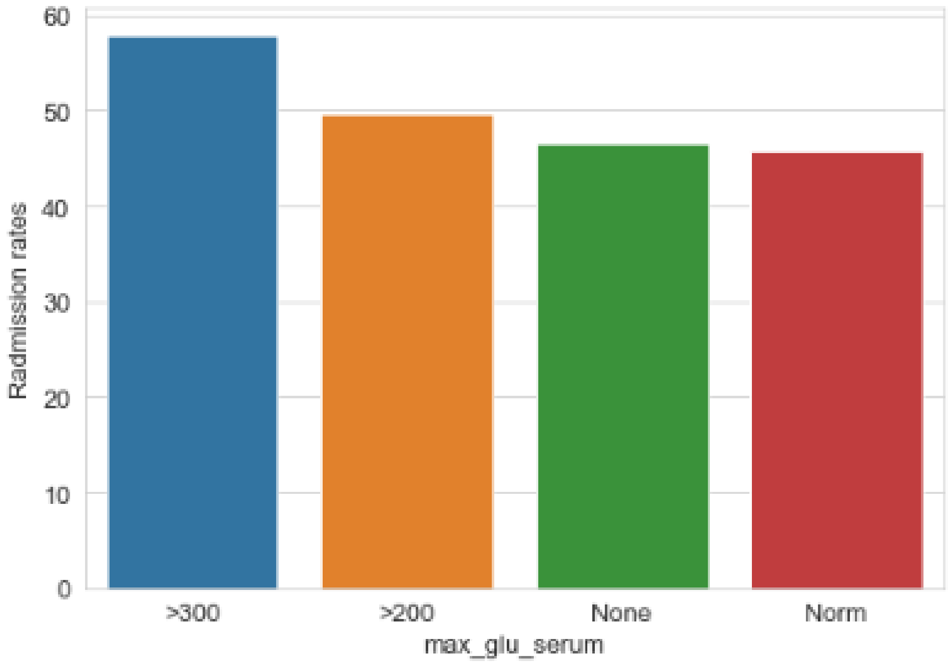


Figure 13: Glucose Serum Test Readmission Rates

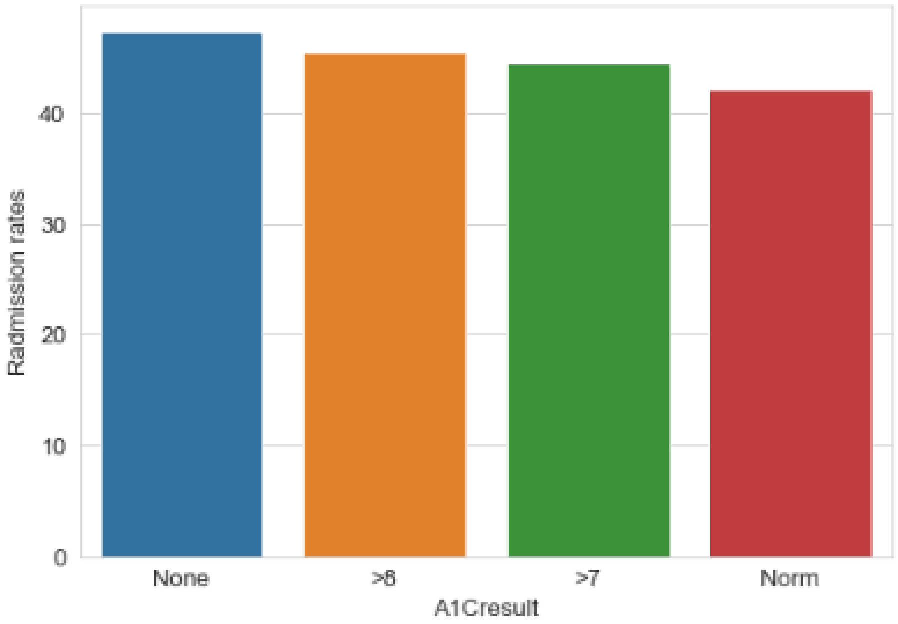


Figure 14: A1C Result Test Readmission Rates

Diagnosis

There are three diagnosis columns; primary, secondary and additional secondary diagnosis. This features represents the medical diseases of the patient i.e. whether patient has diabetes, cancer, etc. Just exploring the distribution of diagnosed diseases among the sample data, we found that Circulatory is the major diagnosis and present among 30% of the patients. Diabetes represents only ~8% among the patients (Figure 15).

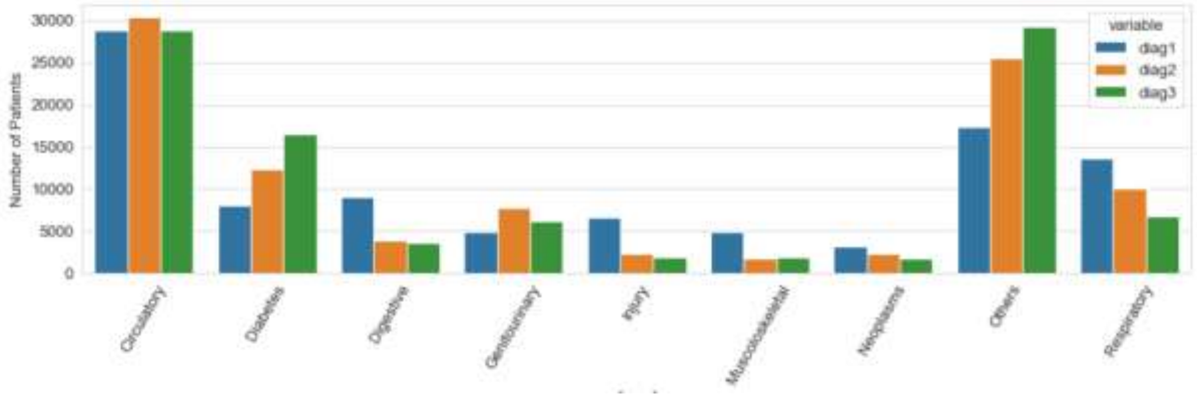


Figure 15: Types of Diagnoses among Patients

Figure 16 shows that a patient having primary diagnosis as diabetes have high readmission rates compared to other diseases. But when we see secondary and additional secondary diagnosis, diabetes is pushed towards the end and Neoplasms (Cancer Related) and Genitourinary (related to kidney) gets pushed up towards high readmission rates. This implies a particular combination of primary, secondary and additional secondary would affect the readmission rates. Our machine learning model will be further able to help us identify those combinations.

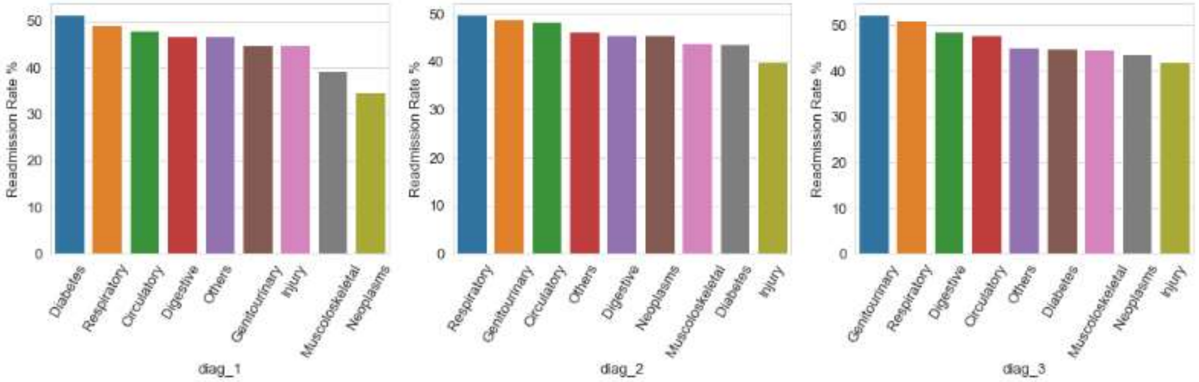


Figure 16: Readmission Rates across different diagnosis types

diag_1 – Primary Diagnosis

diag_2 – Secondary Diagnosis

diag_3 – Additional Secondary Diagnosis

4.7. Patient Encounters Details

This group contains following features:

- Number of lab procedures performed during the encounter
- Number of procedures other than lab tests during the encounter
- Number of distinct medications administered during the encounter
- Number of outpatient and Inpatient visits in the year preceding the encounter
- Number of emergency visits in the year preceding the encounter

- Number of diagnoses entered in system

All of the above features are discrete numeric variables i.e. count variables. The distribution of these variables is shown in **Error! Reference source not found.**. Except for diagnoses number, all variables are skewed towards right suggesting they follow lognormal distribution. Number of diagnoses is skewed towards left. To explore relation between above features with readmission rate, we did a simplification. We combined '>30' and 'NO' in to one class No-Readmission (0) and '<30' as class Readmission (1). This would help us better visualize the relation. Figure 17 shows that except for 'number of procedures other than lab tests', all features have different distribution for readmission and no-readmission. The distribution for readmission is shifted towards right. We explore in next section whether this variation is statistically significant or not for each of the features.

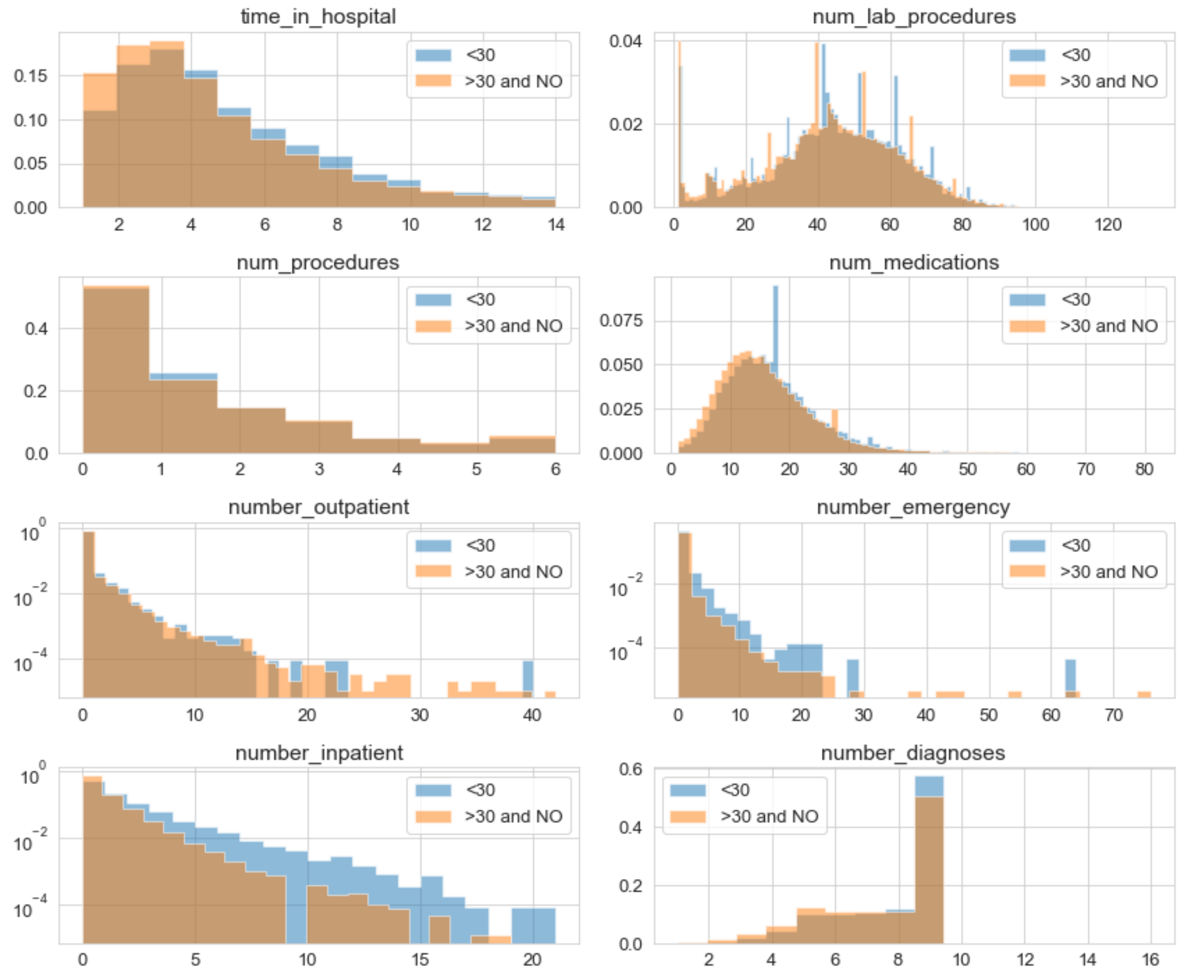


Figure 17: Distribution of patient encounter features between <30 and other classes

We also checked the distribution for the case what if "<30" and ">30" are combined as one class while "NO" as other class.

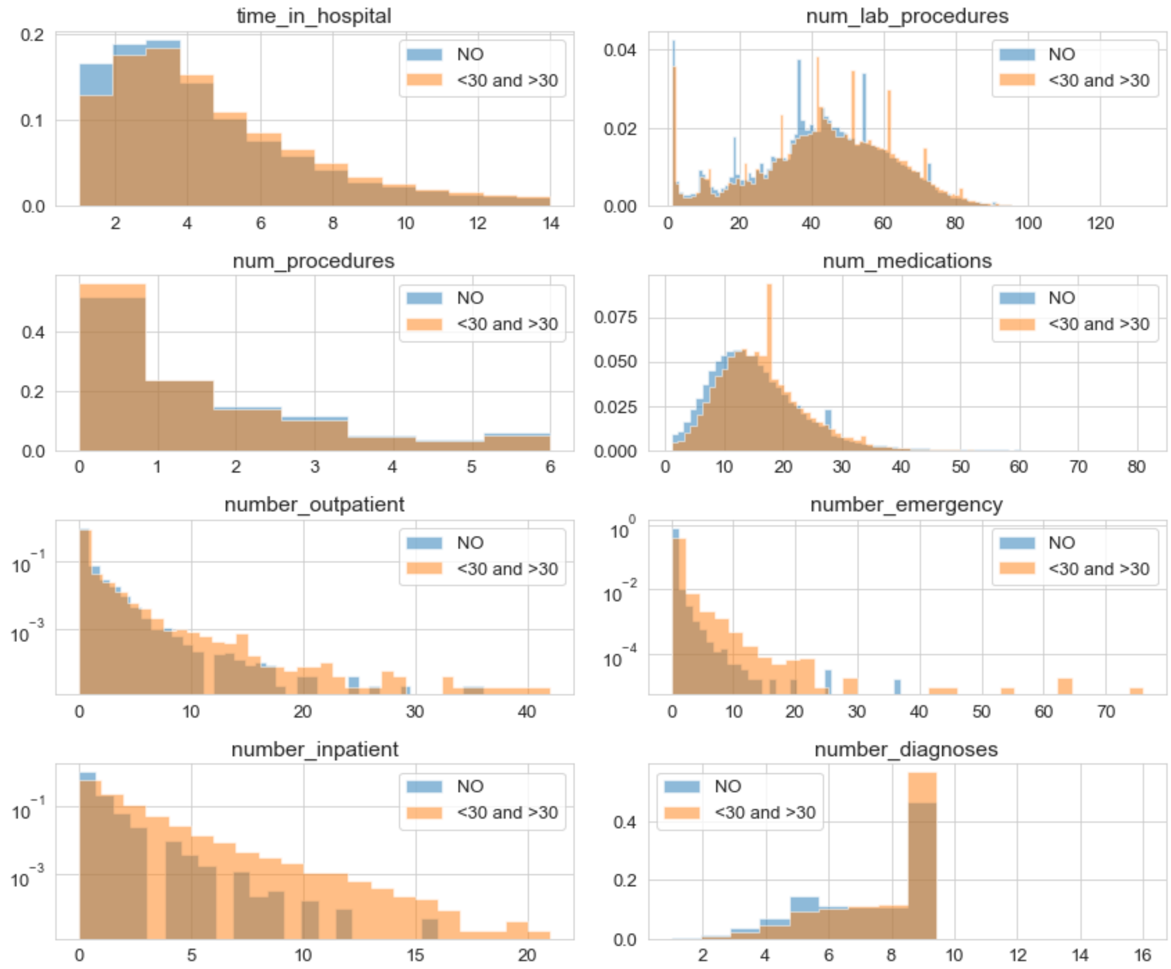


Figure 18: Distribution of patient encounter features between combined >30/>30 and other classes

Based on above distribution, we can easily define our problem from 3 multi-class problem to a binary problem.

4.8. Frequentist and Inferential Statistics

In this section we will see which count variables have statistically significant relation with readmission rates. The Null hypothesis was that the mean of the feature samples for readmission and non-readmission are equal. The method used to test this hypothesis was Bootstrapped Hypothesis Test with difference of mean as test statistic and 5% significance level. P-value and confidence interval was calculated for all the features of Patient encounter group. More details is available in this **IPython notebook**. Below are the results of the test.

```
time_in_hospital        p-value = 0.0
num_lab_procedures      p-value = 0.0
num_procedures          p-value = 0.0009
num_medications         p-value = 0.0
number_outpatient       p-value = 0.0
number_emergency        p-value = 0.0
number_inpatient        p-value = 0.0
number_diagnoses        p-value = 0.0
```

It shows that all variables are statistically significant i.e. Null Hypothesis was rejected. This proves that distribution of patient encounter features are definitely different for readmitted patients and non-readmitted patients. Figure 19 shows the results graphically with confidence interval.

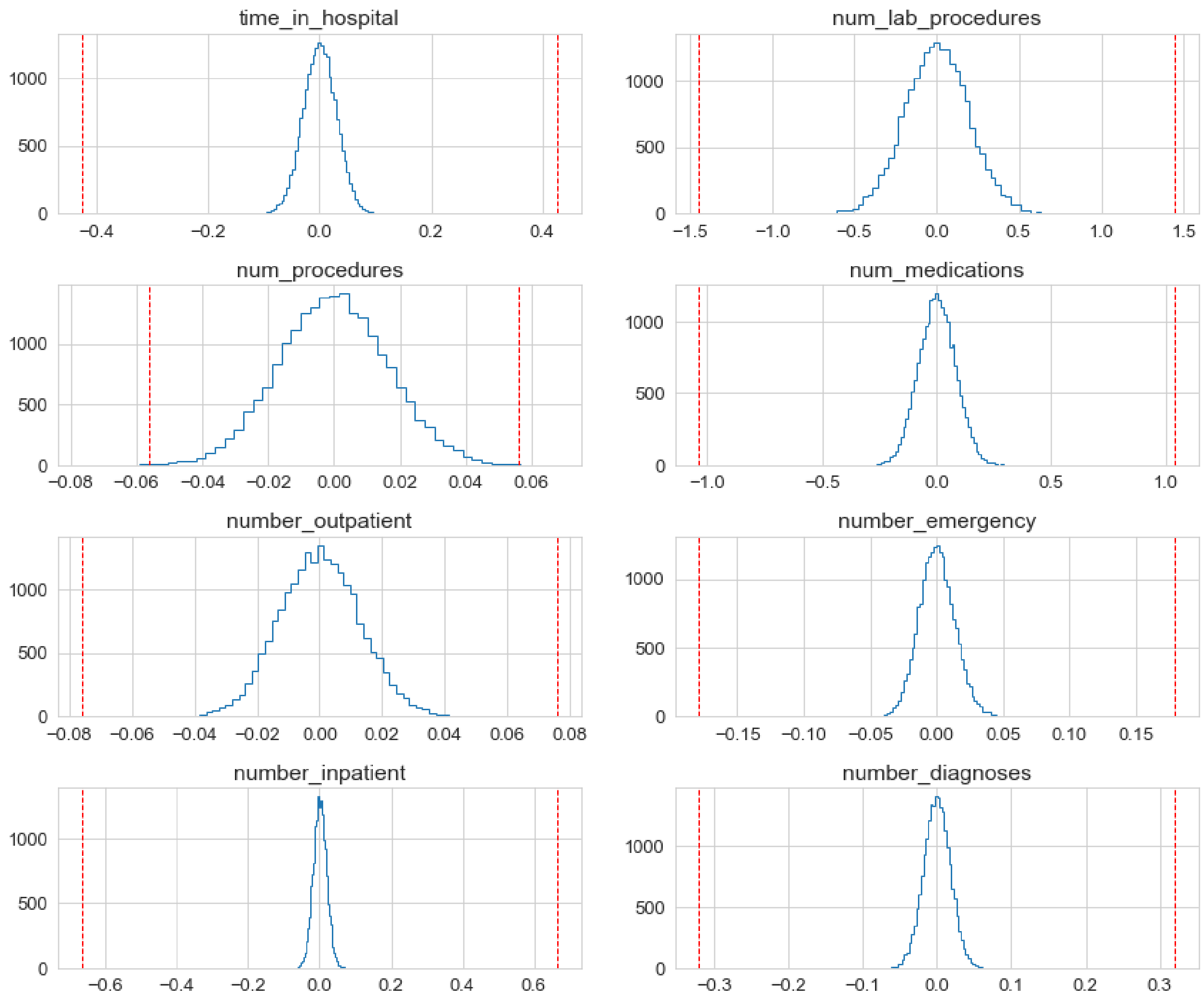
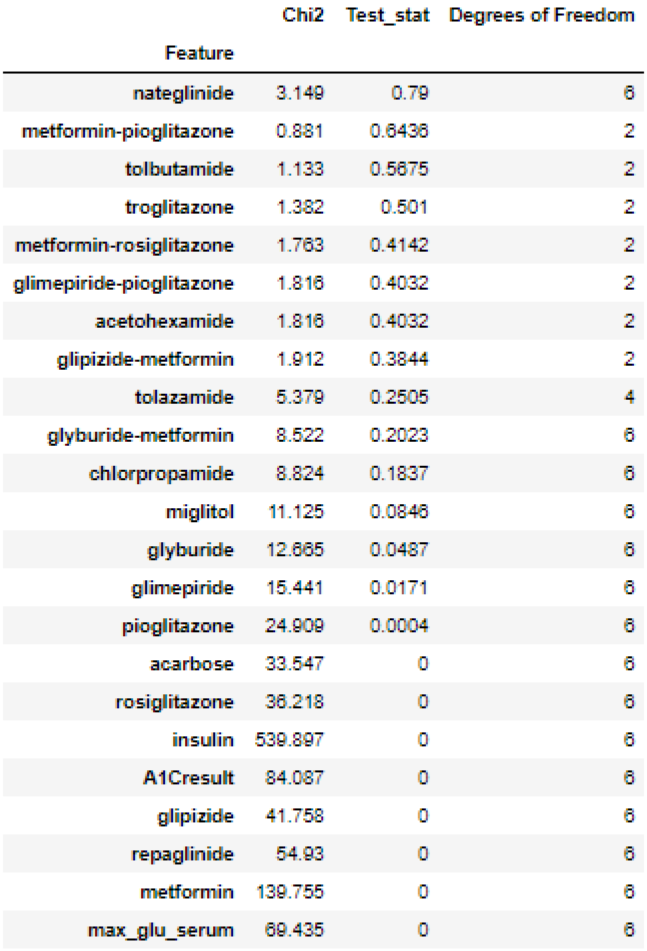


Figure 19: Results of Bootstrapped Hypothesis Tests

I also made use of Chi-square Independence Test to verify whether that columns of medicines that were dropped were insignificant and not related to Readmission rates. Chi-Square Test was used, since I was comparing two categorical variables i.e. medicine features and readmitted. The Null Hypothesis was that medicine features are independent of Readmission rates i.e. there is no relation between the two.

Below Table 3 shows that test statistic for all them medicine variables. There are specific medicines whose test statistics is <0.05 like insulin, A1Cresult and max_glu_serum.



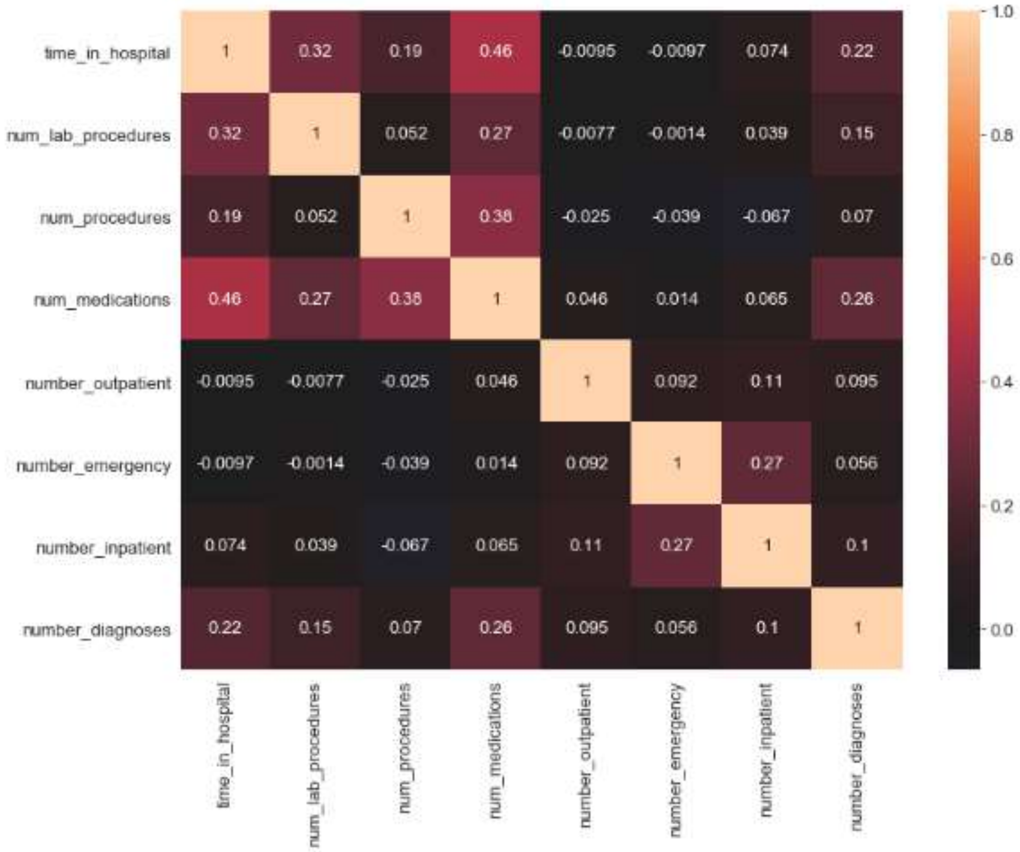
	Chi2	Test_stat	Degrees of Freedom
Feature			
nateglinide	3.149	0.79	6
metformin-pioglitazone	0.881	0.6436	2
tolbutamide	1.133	0.5675	2
troglitazone	1.382	0.501	2
metformin-rosiglitazone	1.763	0.4142	2
glimepiride-pioglitazone	1.816	0.4032	2
acetohexamide	1.816	0.4032	2
glipizide-metformin	1.912	0.3844	2
tolazamide	5.379	0.2505	4
glyburide-metformin	8.522	0.2023	6
chlorpropamide	8.824	0.1837	6
miglitol	11.125	0.0846	6
glyburide	12.665	0.0487	6
glimepiride	15.441	0.0171	6
pioglitazone	24.909	0.0004	6
acarbose	33.547	0	6
rosiglitazone	36.218	0	6
insulin	539.897	0	6
A1Cresult	84.087	0	6
glipizide	41.758	0	6
repaglinide	54.93	0	6
metformin	139.755	0	6
max_glu_serum	69.435	0	6

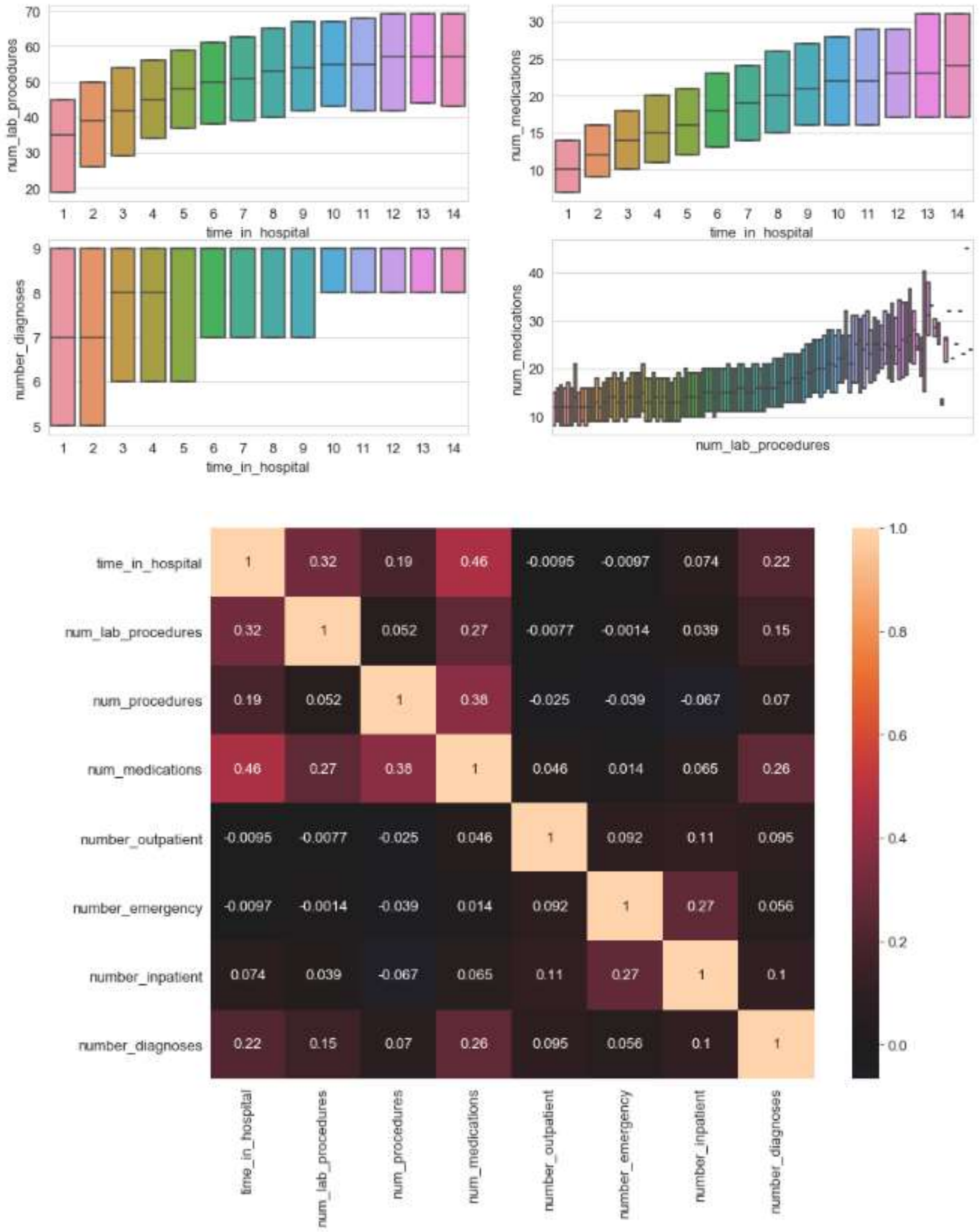
Table 3: Results of Chi-Square Independence Test on Medicine Features

4.9. Multi-Collinearity

Common sense would say that a person spending more time in hospital is highly likely to got high number of lab procedures performed and been prescribed large number of medicines (

Figure 20). Similarly, ‘change’ i.e. change in medications column derives itself from the 24 features of medicine. Thus they two will be highly related. Multi-Collinearity was also observed between variables of Patient admission and discharge. More details about their proof is given in this **IPython Notebook**. This means that multi-collinearity is expected between various independent variables in the given dataset.





	time_in_hospital	num_lab_procedures	num_procedures	num_medications	number_outpatient	number_emergency	number_inpatient	number_diagnoses
time_in_hospital	1	0.32	0.19	0.46	-0.0095	-0.0097	0.074	0.22
num_lab_procedures	0.32	1	0.052	0.27	-0.0077	-0.0014	0.039	0.15
num_procedures	0.19	0.052	1	0.38	-0.025	-0.039	-0.067	0.07
num_medications	0.46	0.27	0.38	1	0.046	0.014	0.065	0.26
number_outpatient	-0.0095	-0.0077	-0.025	0.046	1	0.092	0.11	0.095
number_emergency	-0.0097	-0.0014	-0.039	0.014	0.092	1	0.27	0.056
number_inpatient	0.074	0.039	-0.067	0.065	0.11	0.27	1	0.1
number_diagnoses	0.22	0.15	0.07	0.26	0.095	0.056	0.1	1

Figure 20: Multi-collinearity among features of Patient encounter group

5. Modelling

We have converted our problem to binary classification in Section 4.1. Labels for readmission is 1 while for not readmission is 0. We will use supervised machine learning algorithms to build a predictive model. Since there are only two class, we will use binary classification algorithms. We will use 70% of the data for training and 30% of the data as test set. Final Shape of the data set is 100110 rows and 44 columns. Of the ~1 lakh data, 46.8% of the patients were readmitted. This is approximately a balanced data set.

5.1. Data Pre-processing

Before we feed the data to a machine learning algorithm, we need to perform some pre-processing steps. Below is the outline of all the steps performed during pre-processing. Note that there are some steps which would not be required for some algorithms but for the sake of completeness we are mentioning all the steps in the order they are performed:

1) **Ordinal/One Hot Encoding:** There are both numerical and categorical values present in the data set. For numerical and binary class, there is no need for encoding. For categorical values, I have used Ordinal Encoding for Age so as to capture some ordinal nature of the feature. For other variables One Hot Encoding was used.
2) **Data Splitting:** In this second step, we split our labelled data into 70% training and 30% test data set. We ensure that fraction of both classes are same in training and test data set.
3) **Feature Interaction:** We found multi-collinearity presence in our data set during exploratory analysis. To account for that, feature interactions in logistic and other regression models can take care of some of the non-linearity. Not all models will require feature interaction. Degree 2 Feature interactions were only considered (a, b, ab). Only top 5 features were used based on Chi2 test.
4) **Scaling:** To remove the bias of large values, it is require to scale the numeric variables so that they lie within a specified range. We used Normal Standardization and MinMax scaling techniques depending upon which gives better results.

No Feature selection was done in this project to reduce the dimensionality of the problem, nor were new features created based on model output.

5.2. Modelling and Evaluation metrics

After pre-processing of the data, we the use the data to train and build our model. Thereafter, model is evaluated on test data. It was ensured that the same pre-processing steps used for training set were applied to test data. 5-fold cross validation was used for hyper-parameter tuning. Depending upon the time take to run the model, Grid Search and Random Search was used for hyper-parameter tuning. At times, manual search was needed to narrow down the range of parameters.

Accuracy and area under the curve (AUC) of receiver operating characteristics (ROC) curve were used as the metrics to evaluate the performance. They are popular metrics in this type of classification and balanced data sets.

5.3. Logistic Regression

All the 4 steps of pre-processing were followed. Scaling was attempted using both MinMax and Normal Standardization. GridSearchCV was used for hyper parameter tuning and obtaining the final result.

Top 5 Feature Interactions are as below:

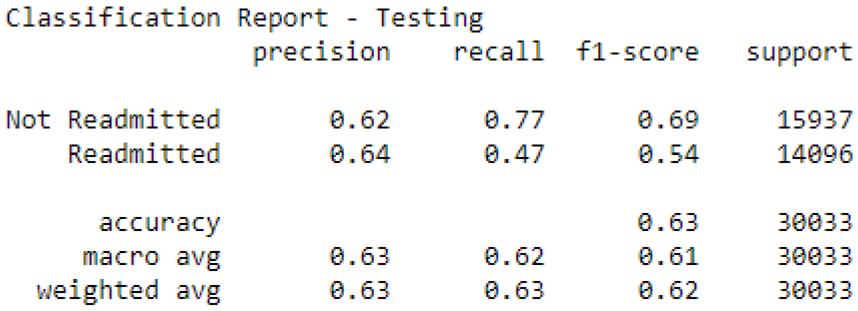
1) age * number_emergency
2) age * number_inpatient
3) num_lab_procedures * number_emergency
4) num_lab_procedures * number_inpatient
5) number_medications * number_inpatient

Hyper parameters tuned were model type, penalty (l1,l2) and C. Final Result post hyper parameter tuning on Feature Interactions pipeline is given in Table 4.

Pipeline Name	Metric	Training	Testing
Post – Hyper parameter Tuning	Accuracy	0.6292	0.6267
	ROC AUC	0.6739	**0.6731**

Table 4: Final Result post hyper parameter tuning - Logistic Regression

Above results suggest that our models does perform better for negative class and have high recall. Class of interest have 47% recall, and AUC is more than 0.5 (Figure 21 and Figure 22). This means that our model is better than luck or chance.



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

 Not Readmitted       0.62      0.77      0.69     15937
     Readmitted       0.64      0.47      0.54     14096

       accuracy                           0.63     30033
      macro avg       0.63      0.62      0.61     30033
   weighted avg       0.63      0.63      0.62     30033
```

Figure 21: Classification Report - Logistic Regression

We also performed L1 regularization to see parameters which are non-relevant. 14 features with zero coefficient were found and all of them were one hot encoded (OHE) variables from medicine groups. There are given in Figure 23.

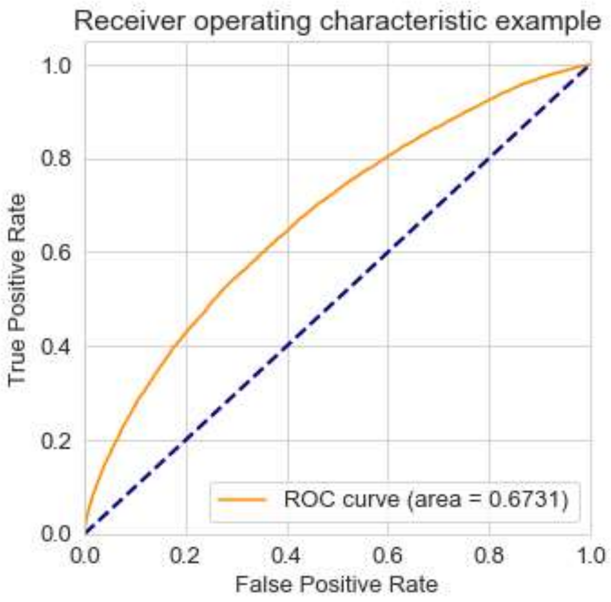
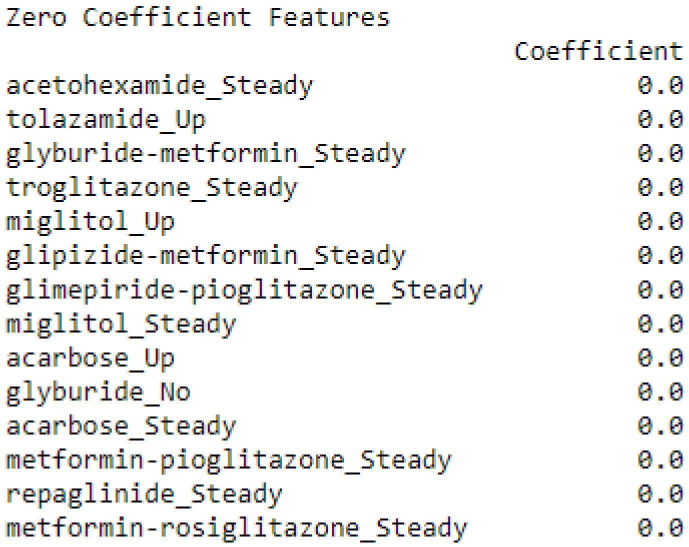


Figure 22: ROC AUC curve - Logistic Regression



```
Zero Coefficient Features
                                     Coefficient
acetohexamide_Steady                         0.0
tolazamide_Up                                0.0
glyburide-metformin_Steady                   0.0
troglitazone_Steady                          0.0
miglitol_Up                                  0.0
glipizide-metformin_Steady                   0.0
glimepiride-pioglitazone_Steady              0.0
miglitol_Steady                              0.0
acarbose_Up                                  0.0
glyburide_No                                 0.0
acarbose_Steady                              0.0
metformin-pioglitazone_Steady                0.0
repaglinide_Steady                           0.0
metformin-rosiglitazone_Steady               0.0
```

Figure 23: Zero Coefficient Features post L1 Regularization

5.4. Support Vector Machines (SVM)

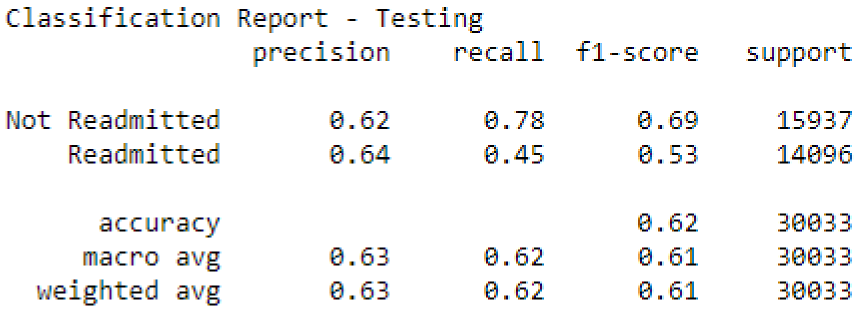
Another algorithm that we tried was support vector machines. It is one of the popularly used off the shelf classifier. Linear kernel was used. It is widely used in text classification, face recognition etc. Similar to logistic regression, all 4 pre-processing steps were followed. Normal Standardization was used for Scaling. Since feature interaction is independent of model used, same features were there as given in logistic regression.

Hyper parameters tuned were penalty (l1,l2) and C. Final Result post hyper parameter tuning on Feature Interactions pipeline is given in Table 5.

Pipeline Name	Metric	Training	Testing
Post – Hyper parameter Tuning	Accuracy	0.6281	0.6250
	ROC AUC	0.6728	**0.6720**

Table 5: Final result post hyper parameter tuning - Linear SVC

Linear SVC algorithm has problems with tolerance, so while hyper parameter tuning, tolerance had to be lowered to help model converge. Testing ROC AUC of SVC is lower than logistic regression. Thus it underperformed. Non-Linear SVC models with different kernels were not tried in this project, as it was assumed that based on above results, ensemble methods would give better results. SVC was faster than Logistic Regression in terms of computational time.



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

Not Readmitted       0.62      0.78      0.69     15937
    Readmitted       0.64      0.45      0.53     14096

      accuracy                           0.62     30033
     macro avg       0.63      0.62      0.61     30033
  weighted avg       0.63      0.62      0.61     30033
```

Figure 24: Classification Report – Linear SVC

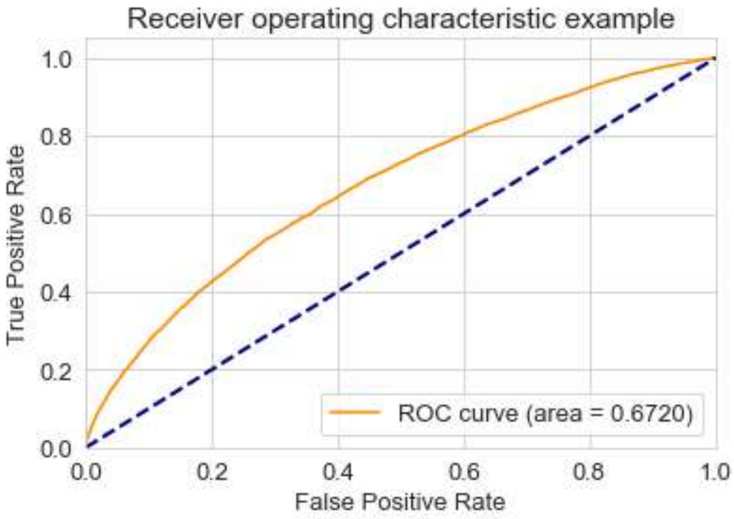


Figure 25: ROC AUC Curve - Linear SVC

5.5. Random Forest

Use of feature interactions in the linear models suggested that non-linear and ensemble models like Random forest would help us give better results. Scaling and Feature Interaction were not performed in this model as Random Forest algorithm takes care of it.

There are lot of parameters to be optimized for Random Parameters. Except for number of trees, all parameters were tuned first. Number of trees and Computation time are directly related. Also above a specified number of trees, roc auc score tends to stabilize i.e. doesn't increase more. So, number of trees were tuned after tuning all other parameters. Following is the list of parameters used for tuning.

- Max Depth
- Min samples split
- Max features
- Number of trees

Individual Manual search was used to narrow down the range for each parameter for tuning. Post that, Randomized Search CV was used for tuning. Optimum number of trees was found to be 300 (Figure 26).

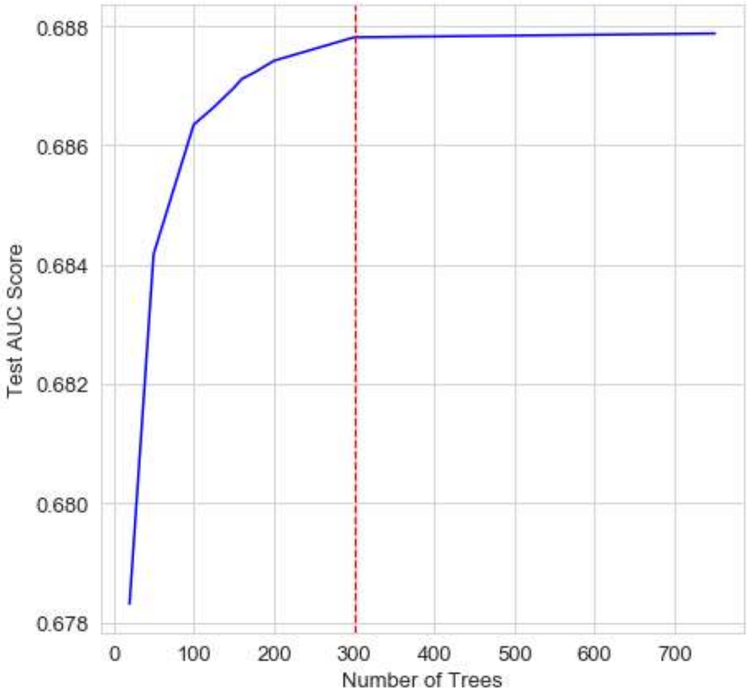


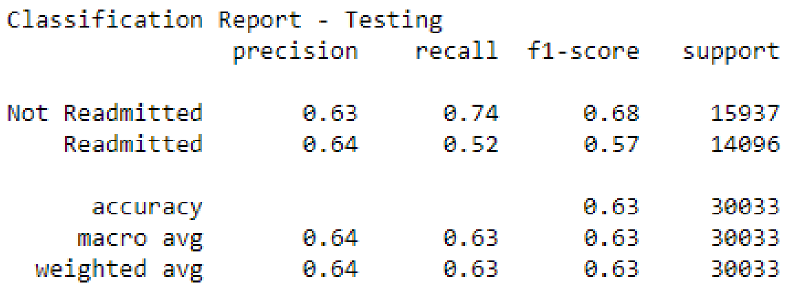
Figure 26: Optimizing Number of Trees - Random Forest

Final Result post hyper parameter tuning on Feature Interactions pipeline is given in Table 6.

Pipeline Name	Metric	Training	Testing
Post – Hyper parameter Tuning	Accuracy	0.7406	0.6349
	ROC AUC	0.8337	**0.6878**

Table 6: Final result post hyper parameter tuning - Random Forest

This model shows good improvement over SVM and Logistic Regression.



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

Not Readmitted       0.63      0.74      0.68     15937
    Readmitted       0.64      0.52      0.57     14096

      accuracy                           0.63     30033
     macro avg       0.64      0.63      0.63     30033
  weighted avg       0.64      0.63      0.63     30033
```

Figure 27: Classification Report - Random Forest

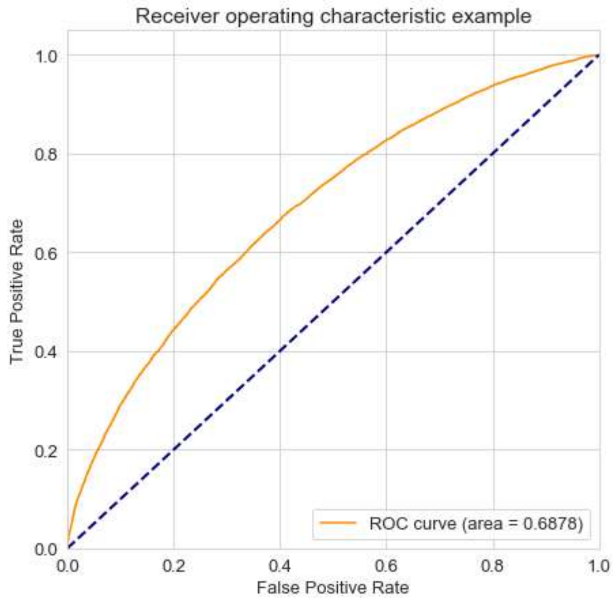


Figure 28: ROC AUC Curve - Random Forest

Top 25 important features as per Random forests are given in Figure 29. Patient encounter details and admission/discharge details seems to dominate the importance. Only Age from patient characteristics find itself in top 10. Also, diabetesMed from medicines category is important. Features from medicines group are important but are from 25 – 50 rank in terms of importance. A1C1_Result _ none and Insulin are the important ones from medicines group.

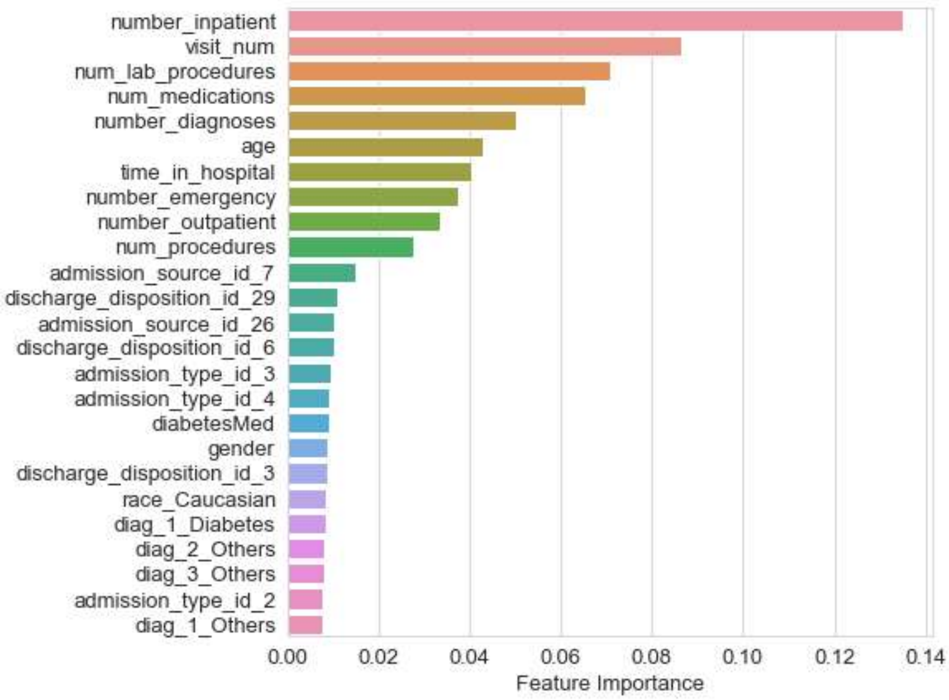


Figure 29: Top 25 important features - Random Forest

5.6. XGBoost

XGBoost is an upgraded version of Gradient Boosting. Similar to random forest, there is no need for scaling or feature interaction in this model. But as an experiment, scaling was done to see the impact. No major impact change in metric was found. Step 4 of pre-processing flow was not performed here.

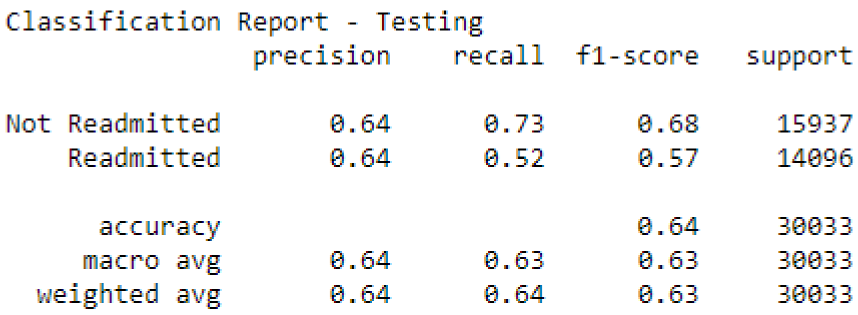
5 parameters were used for hyper parameter tuning. Log loss was used a metric to evaluate the cross validation. Individual Manual search was used to narrow down the range for each parameter for tuning. Post that, Randomized Search CV was used for tuning. Following is the list of parameters used for tuning. It took 188 mins to perform the hyper parameter tuning.

- Gamma
- Learning rate
- Max Depth
- No. of Trees
- Sub sample

Pipeline Name	Metric	Training	Testing
Post – Hyper parameter Tuning	Accuracy	0.6589	0.6353
	ROC AUC	0.7199	**0.6883**

Table 7: Final result post hyper parameter tuning - XGBoost

This model performs better than Random Forest by few margins.



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

Not Readmitted       0.64      0.73      0.68     15937
    Readmitted       0.64      0.52      0.57     14096

      accuracy                           0.64     30033
     macro avg       0.64      0.63      0.63     30033
  weighted avg       0.64      0.64      0.63     30033
```

Figure 30: Classification Report - XGBoost

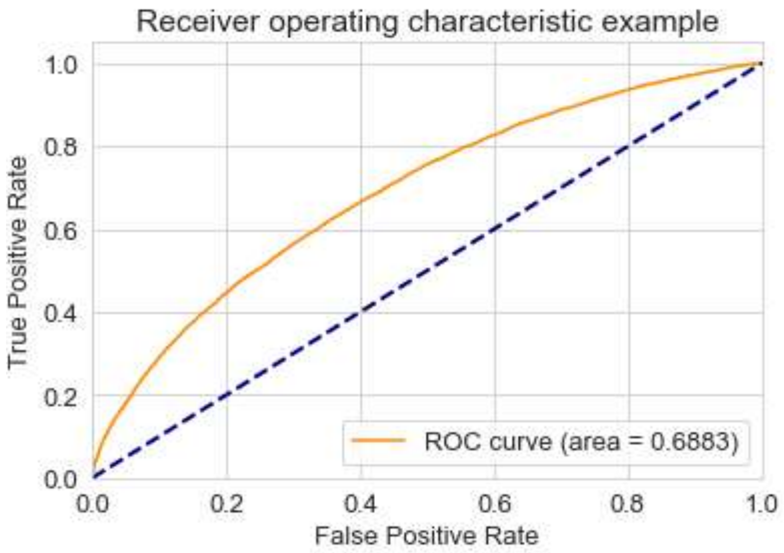


Figure 31: ROC AUC Curve - XGBoost

The list of top 25 important features are different from the Random Forest. Here it is a mix of medicines, patient characteristics, diagnosis, admission/discharge condition and patient encounter details. Here, number_inpatient has very high importance compared to other features. 15 Features common to both random forest and XGboost are number_inpatient, visit_num, age, number_diagnoses, number_emergency, number_oupatient, num_procedures, admission_source_26, discharge_disposition_29, discharge_disposition_6, admission_type_3, admission_type_4, diabetesMed, diag_1_diabetes and diag_1_others.

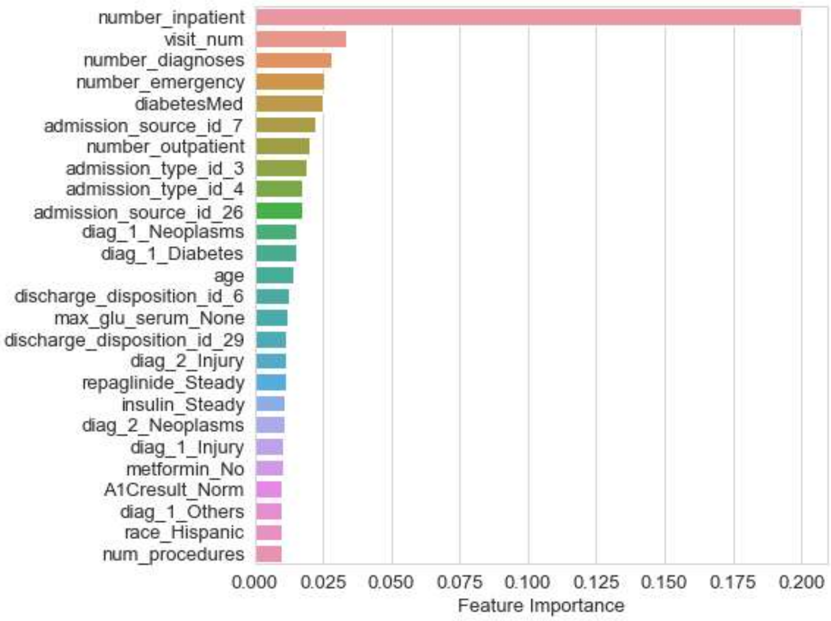


Figure 32: Top 25 Important features - XGBoost

5.7. CatBoost

My Dataset has lot of categorical variables compared to numeric variables. CatBoost is newly developed algorithm to handle categorical data. It is based on gradient boosting of decision trees. In this model there is no need of encoding categorical data. Without any pre-processing it converts categories into numbers using innovative algorithm. Thus for CatBoost only step 2 of data pre-processing was done. But as an experiment, scaling was done to see the impact. No major impact change in metric was found.

There were lots of parameters for tuning in CatBoost. Attempt was made to tune only important parameters. The list is given below.

- max_depth
- rsm/colsample_bylevel
- subsample
- l2_leaf_reg
- one_hot_max_size
- learning_rate
- number of trees

Learning rate and number of trees were tuned after all other parameters were tuned.

Pipeline Name	Metric	Training	Testing
Post – Hyper parameter Tuning	Accuracy	0.6774	0.6392
	ROC AUC	0.7435	**0.6938**

Table 8: Final Result post Hyper parameter Tuning - XGBoost

ROC AUC score is higher than XGboost and Random Forest.

It has a completely different lists of feature importance (no one_hot encoded variables). Common Features between all three ensemble models are visit_num, admission_type, admission_source, age, discharge_disposition, num_procedures, number_oupatient and race. This model gives more importance to medicine groups as can be seen from f

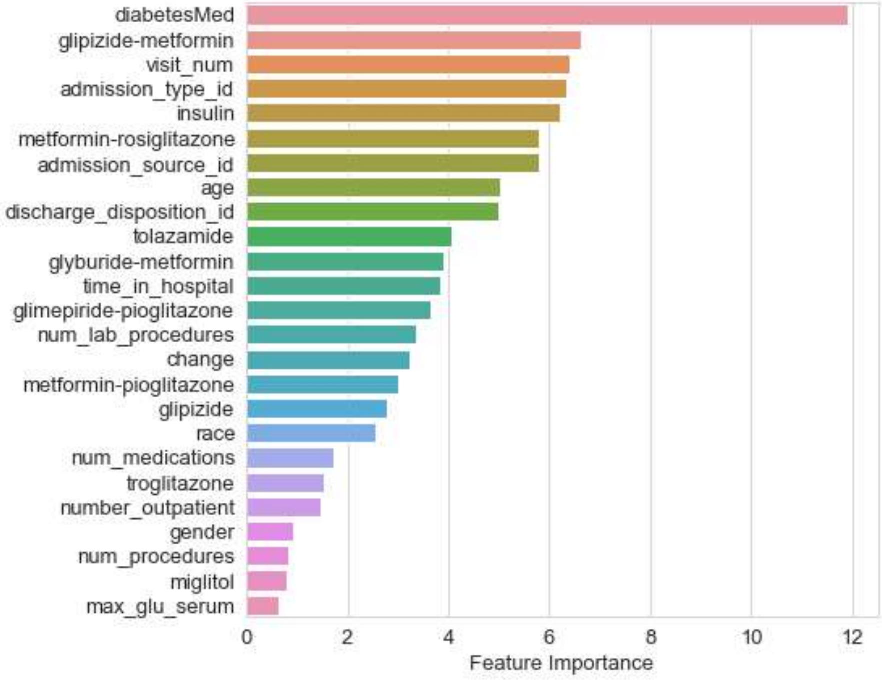


Figure 33: Top 25 Important Features - CatBoost

5.8. Model Comparisons

We have used Logistic Regression, Linear SVC, Random Forest, XGBoost and Catboost classifiers to build a model to predict the readmission rates of a patient. Based on testing the models on test data, there were differences in performance of all the models. The results are shown in Table 9.

Model Name	Accuracy	ROC AUC Score	Computational time
Logistic Regression	0.6267	0.6731	~17.2 secs
Linear SVC	0.6250	0.6720	~14 secs
Random Forest	0.6349	0.6878	~18.5 secs
XGBoost	0.6353	0.6883	~20 secs
CatBoost	0.6392	0.6938	~21.2 secs

Table 9: Comparison of different models. Red for the worst and Green for the best

Other metrics like Log Loss, Brier Loss, PR AUC score can also be used for comparison. They were not calculated in this project. Table shows that there is not much difference between the models.

The main difference between the models was found in terms of feature importances. All the models gave different set of important features. Random Forest gave more importance to patient encounter details while XGBoost was a mix of all groups and CatBoost gave more weightage to Medicines and patient encounter details. We chose CatBoost as the final model since it has the best accuracy and AUC score. But the AUC score for this data set is pretty low, which suggests that even though this is balanced data set, negative class is of dominating nature.

6. Using Model and Recommendations

To now use the best model, some cleaning and grouping of the variables like discharge disposition will be needed in terms of clubbing them into smaller groups. Apart from these, pre-processing steps of Splitting Data and Scaling would be needed. There would be no need of performing OHE since CatBoost takes care of it. All features would be used for the model except for features which were removed because of high % of missing values.

These model can give high false negatives, so it is recommended to keep those into consideration while taking the final decision. Also it is recommended to use probability values from the model to make categories to indicate the level of readmission rates. A sample is given in Table 10.

p(readmission) > 75%	High chance of readmission
P(readmission) (50% - 75%)	Moderate chance of readmission
p(readmission) (25% - 50 %)	Low chance of readmission
p(readmission) <25%	Very low chance of readmission

Table 10: Categorization of Readmission based on probability

7. Assumptions and Limitations

- This model is limited to predicting the readmission rates for only diabetes patients.
- Grouping of Patient admission and discharge details has been done based on similar distribution of Target variable. In practice for some ids where data is few, this might differ.
- Age has been considered as an ordinal variable.
- Target variable was of three class (<30, >30, NO) but we converted this problem into a binary classification by combining <30 and >30 as one class. The model could also have been built by combining >30 and NO as one class.

8. Future Work

- Including medical speciality data after considering missing data as a separate category
- Considering age as categorical data rather than ordinal data.
- Creating new features based on important features found from the current project i.e. combining admission and discharge data , patient encounter details
- Getting new hospital names, state and region data can better help in identifying the readmission rates and improving the model
- Building similar models for other categories of diseases which are chronic in nature

9. Conclusion

The data set for this project was sourced from Kaggle and was originally submitted by Virginia Commonwealth University. The first step in this project was to explore data for any missing values. Missing values for each of the column were handled differently. Then we did some data wrangling to make data more analysable. For some categorical columns, grouping was done to reduce the number of unique categories. Medical cause diagnoses ICD codes were replaced by corresponding names. Post this cleaning, we further explored the data through various visualization to understand the distribution of the features. It also helped us identify the correlation between target variables and input variables. Bootstrapping and Chi2 test were used to determine the relation. Multi-collinearity between variables of patient encounter were found.

Post the exploratory analysis, we build 5 different models consisting of both linear and ensemble methods. Every model required different sets of pre-processing steps. Hyper parameter tuning took the maximum computational time for the ensemble models. Based on accuracy and ROC AUC score, CatBoost was selected as the final model for implementation. It gave the highest ROC AUC score of 0.6938 and accuracy of 63.92%. It was recommended to use probability values from the model to make categories to indicate the level of readmission rates.

The accuracy of the model can be improved by including other data like medical specialty, region and hospital based data. Future projects also has scope of feature engineering based on domain knowledge.