PREDICTION OF READMISSION OF DIABETIC PATIENTS

A Thesis submitted to

Great Lakes Institute of Management

In partial fulfilment of requirements for

the award of

Post Graduate Programme in Data Science and Engineering

By

Akhila Panabaka,

Bharath Kumar,

Bhargav Shale,

Rithvik Kandukuri,

Yashwant Vishnu

Under guidance of

Dr Pankaj Agarwal

Table of Contents

PROBLEM STATEMENT

Whether a Diabetic patient will be readmitted within 30 days or not

With respect to morbidity and mortality, management of hyperglycaemia has a significant impact. This led to standardizing protocols regarding monitoring Glucose levels in ICUs (Intensive Care Units). This is not the scenario for most non-ICU inpatient admissions, where management is arbitrary and wide fluctuations in glucose levels are observed. For effective and safe treatment, Protocol driven inpatient strategies are necessary.

The data consists of features about demographics, type of admission, test results, type of medications prescribed etc.

The input to each algorithm would be data of each patient**.** We use the algorithm to output a binary prediction of whether a Diabetic patient will be readmitted within 30 days or not.

For this project we hope to gain the following insights:

1. Which factors have more impact on Readmission probability?
2. Which Age group patients have higher chances of readmission?
3. Which Gender patients have higher chances of readmission?
4. Effect of changing medications dosage on readmission probability

PROJECT OUTCOME

The outcome of our project is to build a robust machine learning algorithm that helps the business client understand what are the factors determine if a diabetic patient is going to be readmitted within 30 days or not

DATA SET

- A dataset is a collection of data, and it can be structured or unstructured.
- A structured data is represented in a tabular format, where every column of the table represents a particular variable, and each row corresponds to a given record of the dataset.
- Unsupervised/unstructured data is not represented in a tabular form, data that we fetch from Kaggle.

DATA DESCRIPTION:

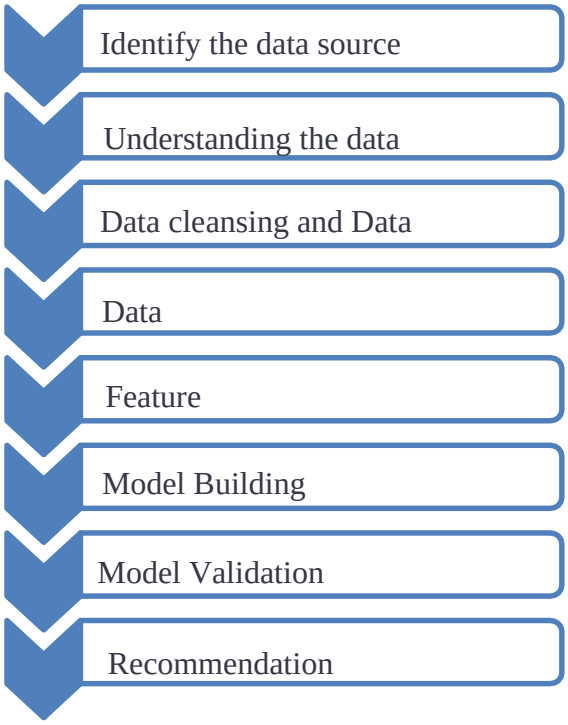
The data contains such attributes as patient number, race, gender, age, admission type, time in hospital, medical specialty of admitting physician, number of lab test performed, HbA1c test result, diagnosis, number of medications, diabetic medications, number of outpatient, inpatient, and emergency visits in the year before the hospitalisation, etc. The data set consists of 101766 observations and 50 features before the cleaning.

Variables	Variable Information	Data Type
Encounter ID	Unique identifier of an encounter	Numerical
Patient number	Unique identifier of a patient	Numerical
Race	Values: Caucasian, Asian, African American, Hispanic, and other	Object
Gender	Values: male, female, and unknown/invalid	Object
Age	Grouped in 10-year intervals: [0, 10), [10, 20), . . ., [90, 100)	Object
Number of lab procedures	Number of lab tests performed during the encounter	Numerical
Number of procedures	Number of procedures (other than lab tests) performed during the encounter	Numerical
Number of outpatient visits	Number of outpatient visits of the patient in the year preceding the encounter	Numerical

Number of emergency visits	Number of emergency visits of the patient in the year preceding the encounter	Numerical
Number of inpatient visits	Number of inpatient visits of the patient in the year preceding the encounter	Numerical
Diagnosis 1	The primary diagnosis (coded as first three digits of ICD9)	Object
Diagnosis 2	Secondary diagnosis (coded as first three digits of ICD9)	Object
Diagnosis 3	Additional secondary diagnosis (coded as first three digits of ICD9)	Object
Glucose serum test result	Test for diabetes	Object
A1c test result	Test for diabetes	Object
Number of diagnoses	Number of diagnoses entered to the system	Numerical
Change of medications	Indicates if there was a change in diabetic medications (either dosage or generic name)	Object

medications	For the generic chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, sitagliptin, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, and metformin-pioglitazone, the feature indicates whether the drug was prescribed or there was a change in the dosage	Object
Days to inpatient readmission	If the patient was readmitted in less than or more than 30 days, or no record of readmission	Object

PROJECT METHODOLOGY



Methodology to be Followed:

CRISP-DM stands for **the industry standard process** for **data mining and** is a methodology **developed for designing** data mining projects. **Describes** the **various phases / tasks of** the project and **outlines the** data mining life cycle.

Business Understanding -

The focus of this process is on understanding the business requirements and objectives, and determining what outcome to achieve.

What business units are affected by this problem?

What are the possible causes of this problem?

What are the possible solutions to this problem?

Identify what is needed in order to satisfy the needs of the customer.

- Situation assessment: Determine resource availability, project requirements, risk assessment and emergency, and perform a cost-benefit analysis.

- Define data mining objectives: turn a business problem into a data mining problem and Recognize the type of data mining problem such as classification, regression or aggregation, etc.
- Develop a project plan to ensure successful completion of the project.

Data Understanding:

The first step in data analysis is collecting the data. After that, you examine the data for its surface properties, such as its format and number of records. The next step is to analyze the data to learn more about each attribute and to perform basic statistical measures on them. Check the data for quality by looking for missing values, outliers, duplicates, etc.Then determine the quality of the data.

- Collect initial data: Acquire the data and load it into the analysis tool to be used.
- Describe data: Examine the data and document its surface properties like data format, number of records, or field identities. Understand the meaning of each attribute and attribute value in business terms. For each attribute, compute basic statistics so as to get a higher-level understanding.
- Explore data: Find insights from the data. Query it, visualize it, and identify relationships among the data.
- Verify data quality: Identify special values, missing attributes and null data. Determine how clean/dirty is the data.

Data Preparation:

The goal of data wrangling is to develop a final data set for EDA and modeling. This tool covers all of the steps necessary to create the final dataset from the raw data. Some of the tasks include selecting tables, records, and attributes for modelling, as well as transformation and cleaning of data for modelling tools.

Select Data: Determine which attributes/features will be used and document reasons for inclusion/exclusion.

- **Clean data:** Correct, impute and remove the improper data.
- **Extract data:** Derive new attributes from the existing ones
- **Integrate data:** Create features by combining data from multiple sources.
- **Format data:** Re-format data as necessary. For example, convert string values to numeric

values so as to perform mathematical operations.

Modelling: In this step, we build and evaluate different models built using different methods from the training dataset.

- **Select modelling technique:** Determine the algorithms to be used to model the data based on the business requirement.
- **Generate test design:** To train and test the model, we need to divide the dataset into a training data set and a testing data set. In this step, we divide the data into a training data set and a test data set.
- **Build model:** Based on the modelling technique selected, build the model on the input data set.
- **Assess model:** Compare the results of different models based on **the** confusion matrix. The outcome of this step **often** leads to **iterations of** model tuning until the best model is found.

Evaluation: Evaluate the models and review the steps executed to construct the model to be certain it properly achieves the business objectives.

- **Evaluate results:** Understand the data mining results and check how impactful they are in achieving the data mining goal. Select appropriate model based on confusion matrix.
- **Review process:** Review the work accomplished and make sure that nothing was overlooked and all steps were properly executed. Summarise the findings and correct anything if needed.
- **Determine next steps:** Based on the previous three tasks, determine whether to proceed to deployment, iterate further, or initiate new projects.

Data Preprocessing:

Data pre-processing is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data pre-processing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for building and training Machine Learning models.

Data Preparation is the process of collecting, cleaning, and consolidating data into one file or data table, primarily for use in the analysis.

Acquire the dataset:

https://www.kaggle.com/datasets/brandao/diabetes/metadata

Missing/Null Values:

Impute or drop features with missing values based on the percentage of missing values and relevance for model building.

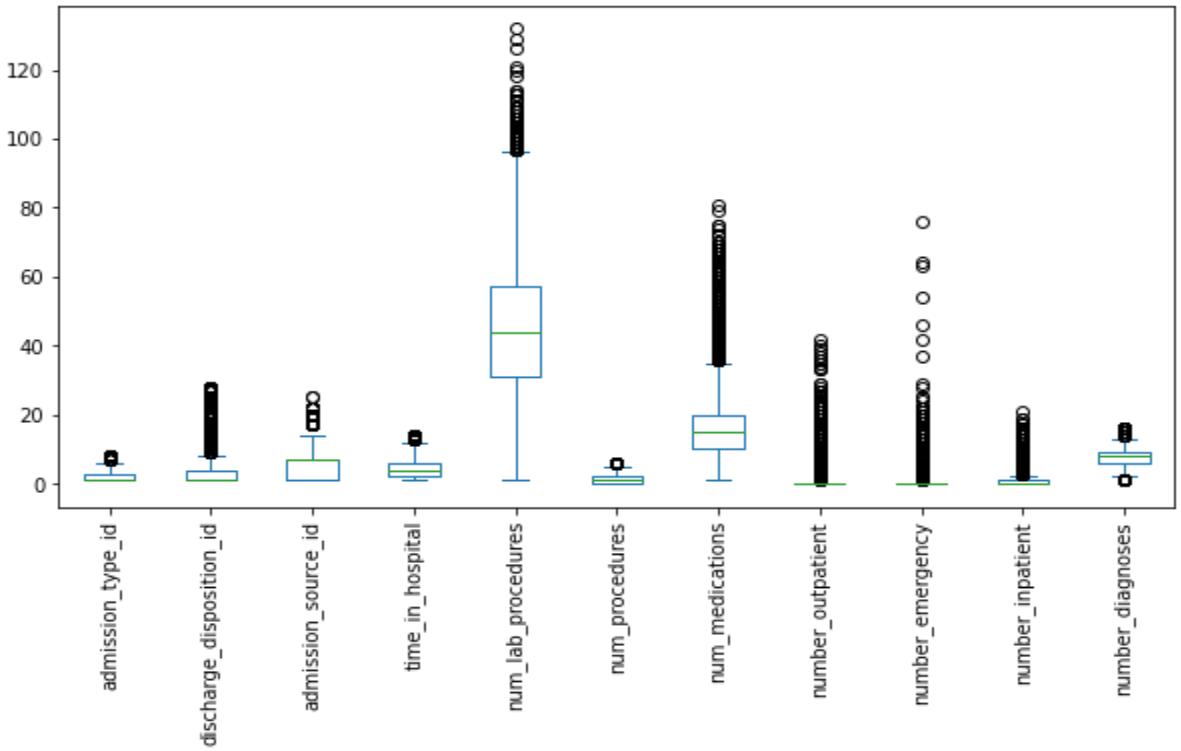
There are null values in columns

- race: 2.233%
- weight: 96.85%
- payer_code: 39.55%
- medical_specialty: 49.08%
- diag_1: 0.02%
- diag_2: 0.35%
- diag_3: 1.39%

- Dropping weight, payer_code, medical_speciality columns as they have 97%, 40%, 49% missing values
- Binning has been done on Race column and treated the null values with mode imputation
- For diag_1, diag_2, diag_3, the values have been binned according to ICD9 Codes. We replaced the null values as 'other' class which can indicate other diagnosis or no diagnosis also.

OUTLIERS:

Outliers are the extreme that deviates from other observations on data, they may indicate variability in measurement, experimental errors or a novelty.



There are many outliers in numerical columns. We are not going to treat them or drop them for our base model as it would lead to data manipulation or data loss.

Exploratory Data Analysis:

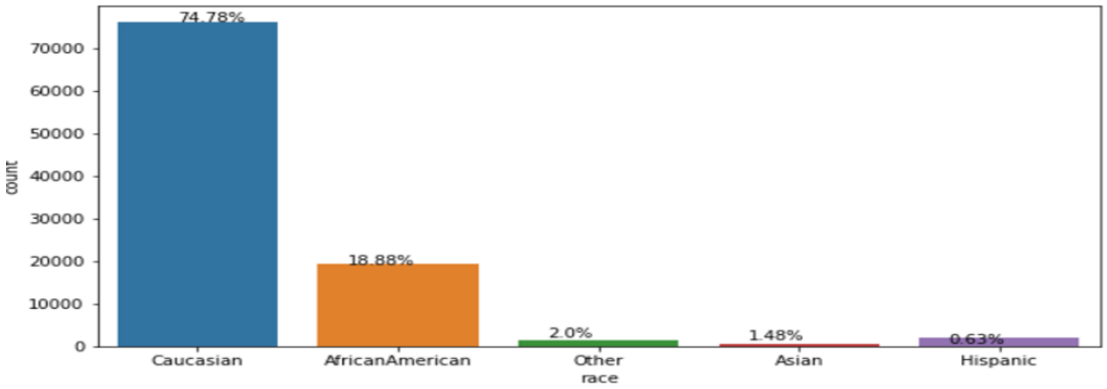
Exploratory Data Analysis (EDA) is an approach to analyzing datasets to summarize their main characteristics, often with visual methods.

Univariate Analysis

Univariate analysis is the simplest form of analyzing data. Uni means one, so in other words the data has only one variable. Univariate data requires analyzing each variable separately.

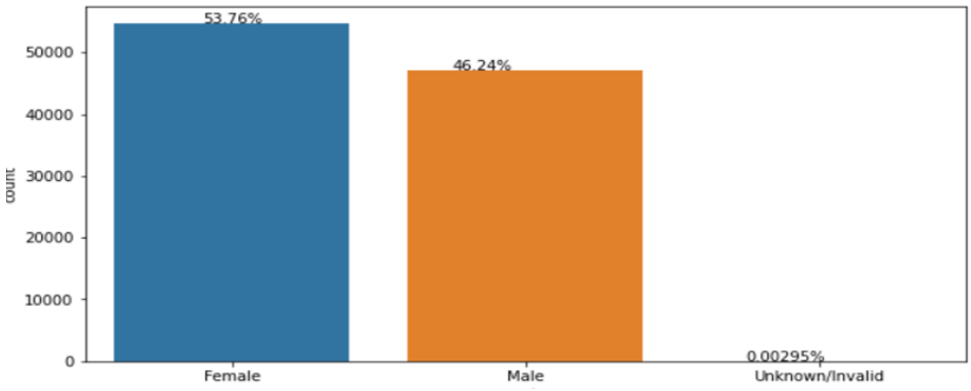
Race :

This feature explains about different races like Caucasian which are 74.78% following AfricaAmerica are 18.18%, Hispanic 2.0% ,others 1.48% and Asian 0.63% are suffering from diabetes.



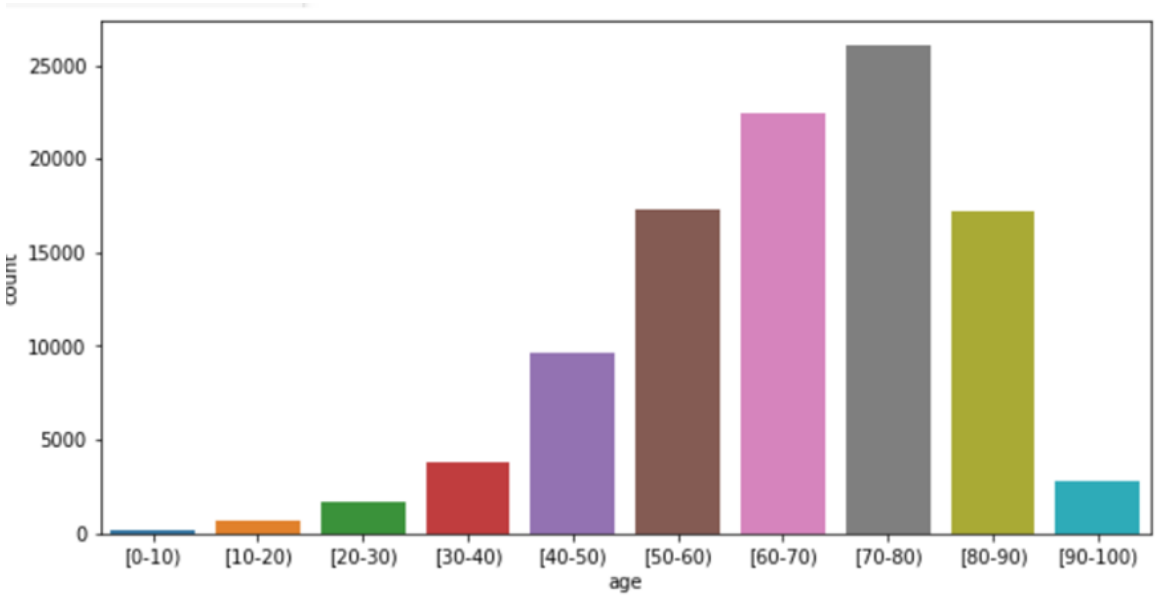
Gender :

The variable consists of female of 53.76% , Male of 46.24% and unknown category of 0.00295% .The major classes are nearly balanced.



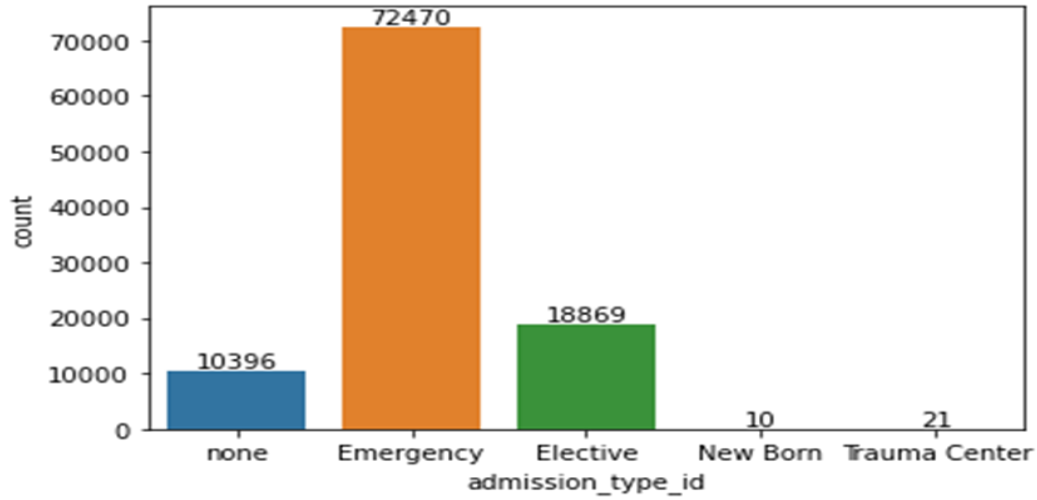
Age :

The highest number of patients are from age groups 70-80, 60-70 and 50-60 in the same order



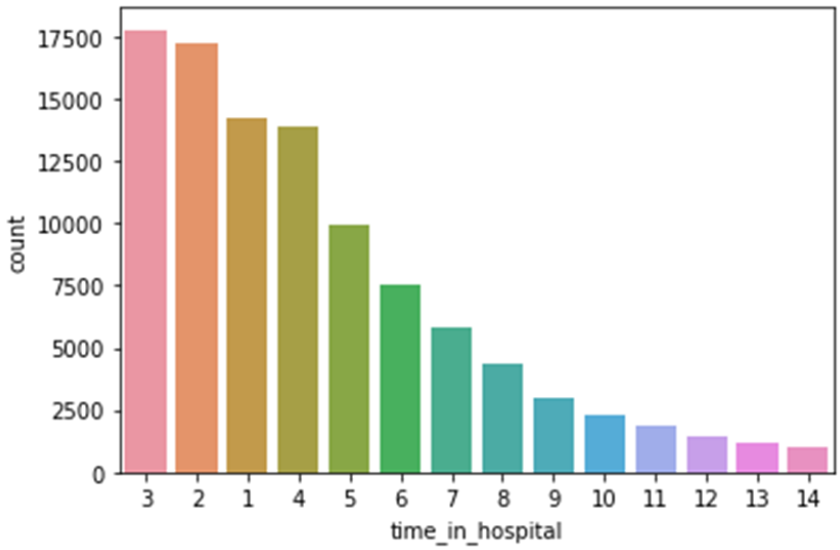
Admission_type_id :

Most of the patient's admission type is emergency.



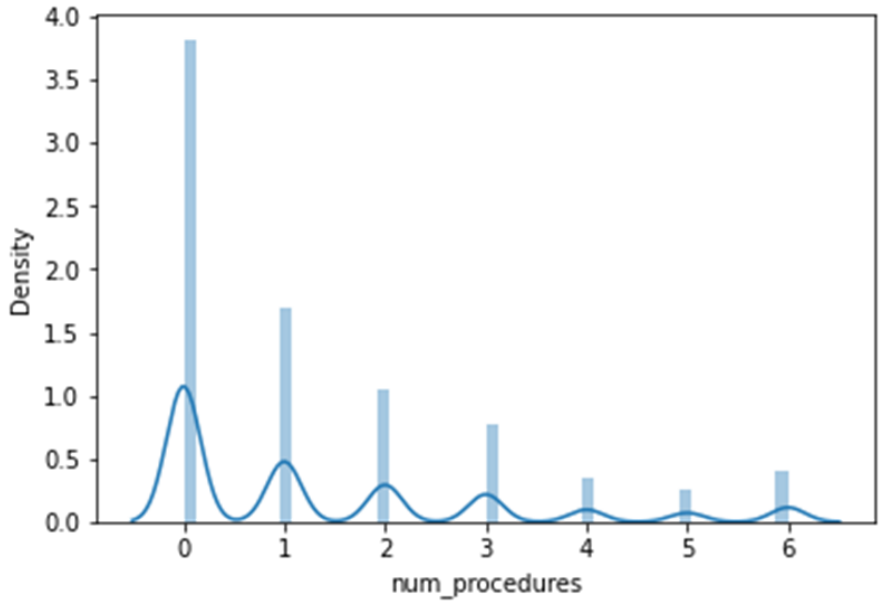
Time_in_hospital:

Nearly 60 % of patients stayed between 1 to 4 days.



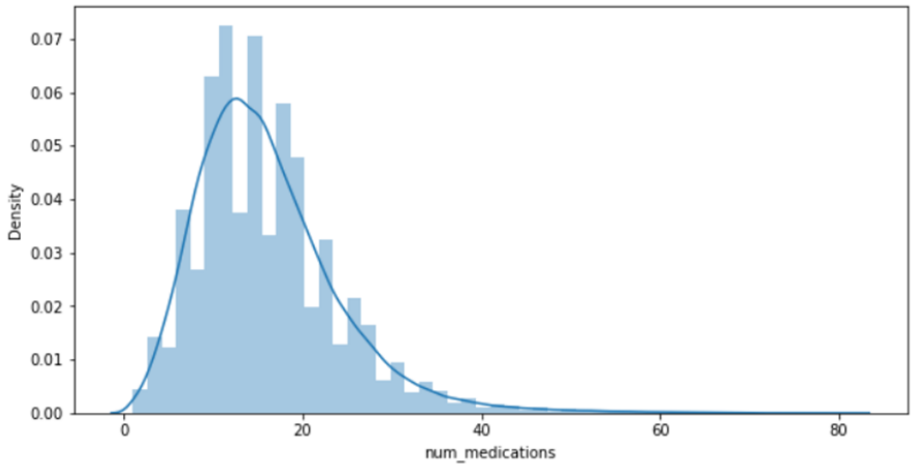
Number of procedures :

Most procedures are in the range 0 to 2.



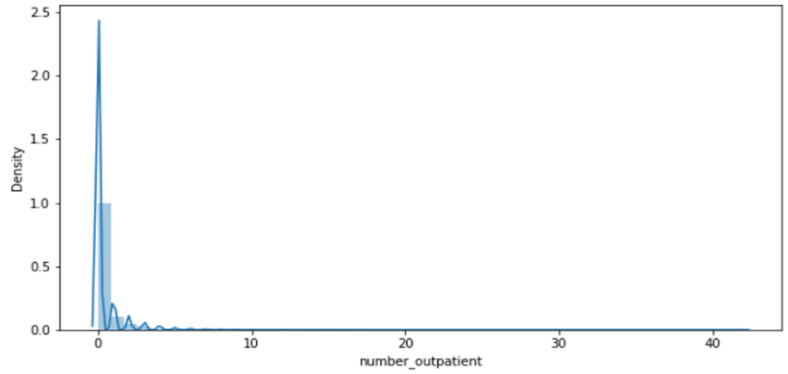
Number_of_medications :

Most number of medications are in the range of 10 to 20



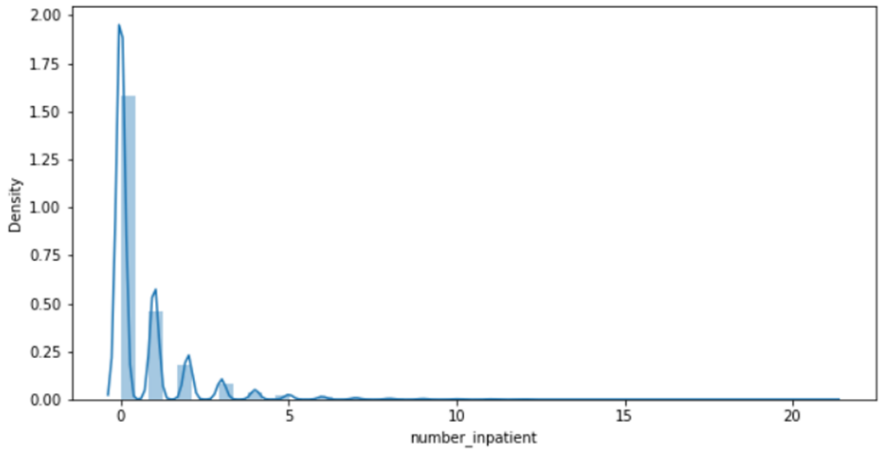
Number_Outpatient:

Most number of outpatient visits are 0.



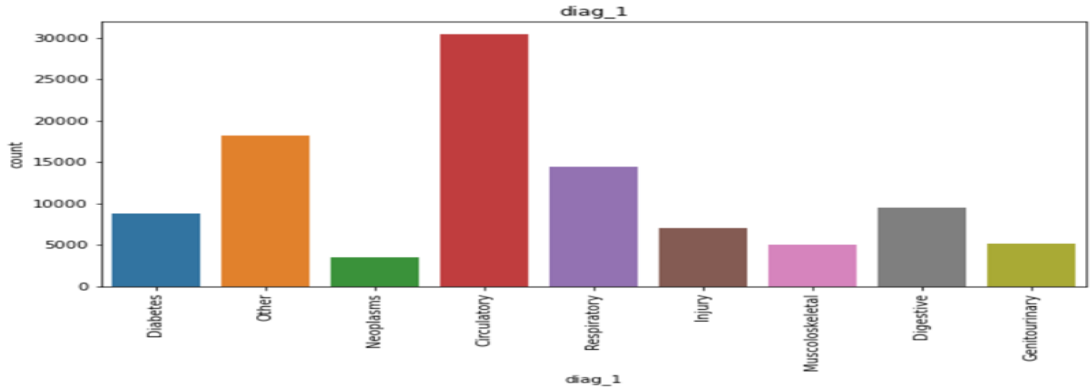
Number of inpatient :

Most number of inpatient visits are in the order 0, 1, 2, 3



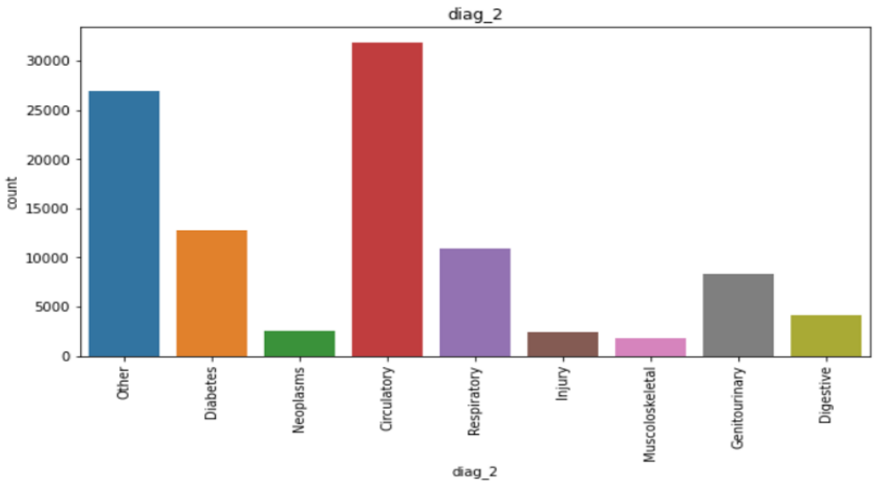
Diagnosis 1 :

Circulatory, Respiratory and other types are highest in number in primary diagnosis.



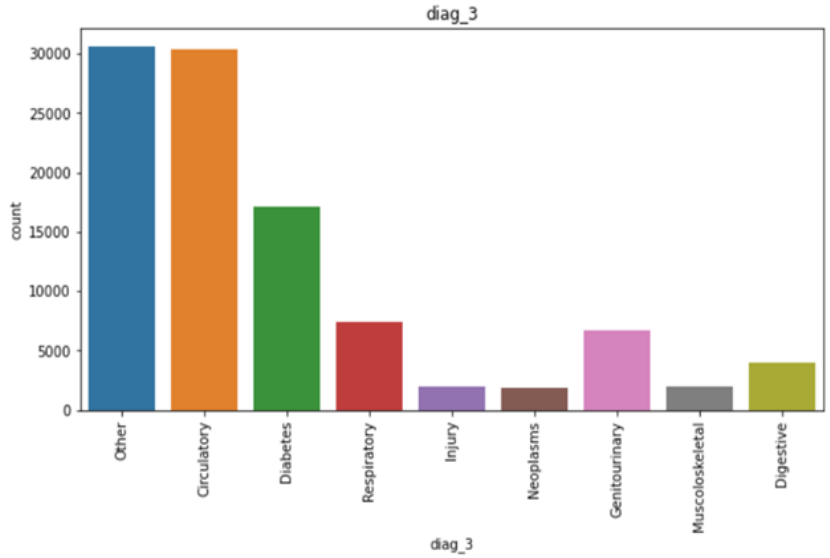
Diagnosis 2 :

Circulatory, other types and diabetes are highest in number in secondary diagnosis



Diagnosis 3 :

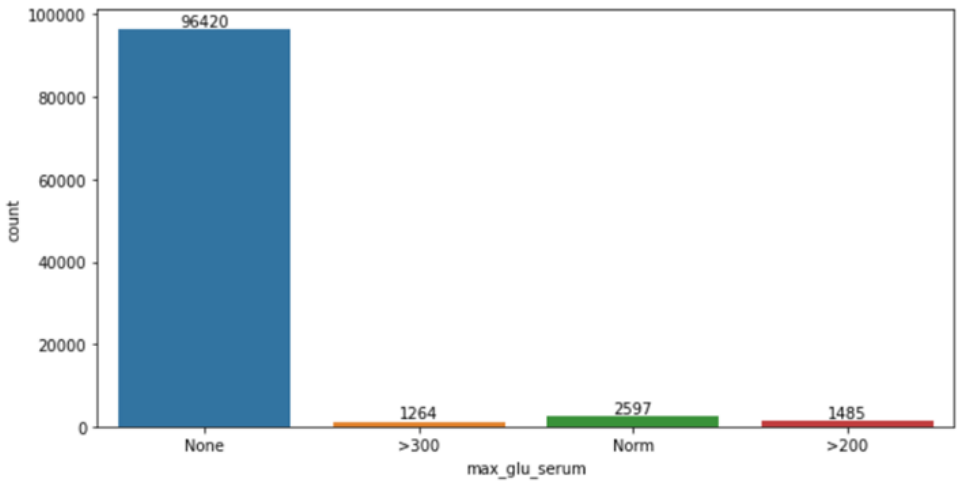
Circulatory, Other types and diabetes are highest in number in Tertiary diagnosis



Glucose serum test result :

Glucose serum test result is a numeric variable.

- The glucose serum is the simplest and most direct single test available to test for diabetes.
- The test measures the amount of glucose in the fluid portion of the blood. Results were normal for most of the patients followed by >200 and then >300.



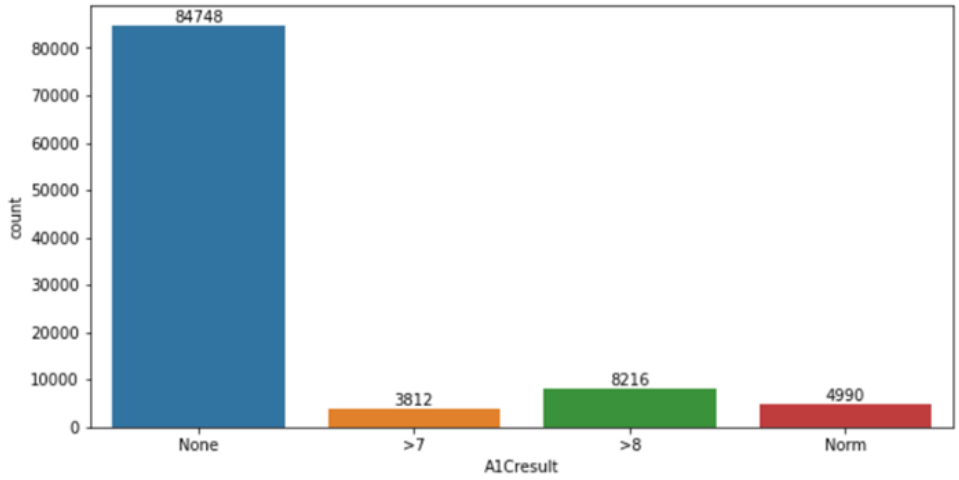
A1c test result:

A1c test result is a numeric variable

· It is simple blood test that measures your average blood sugar levels over the past 3 months

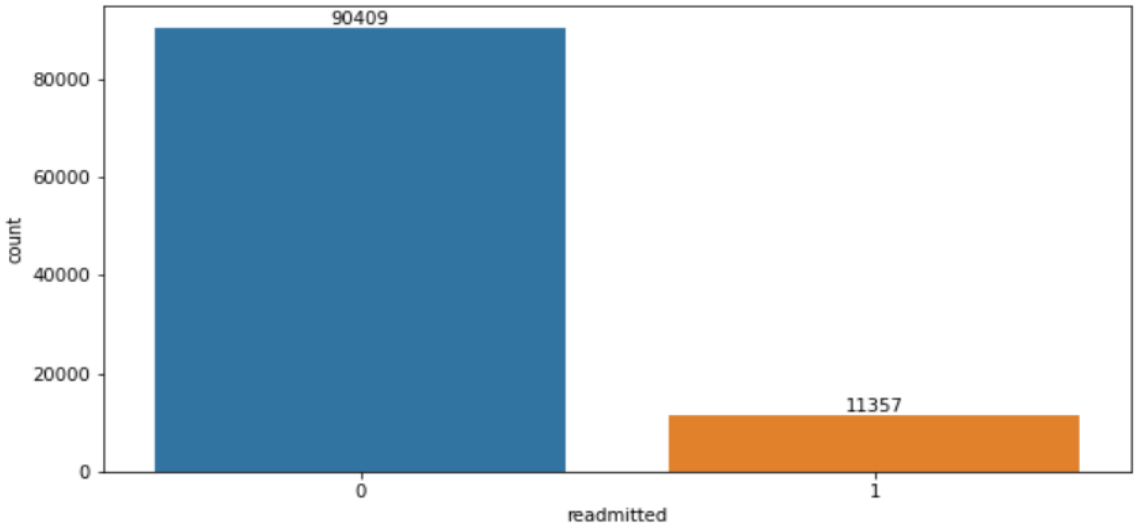
For most of the patients AIC test was not conducted

Results were >8 for most of the patients followed by normal and then >7



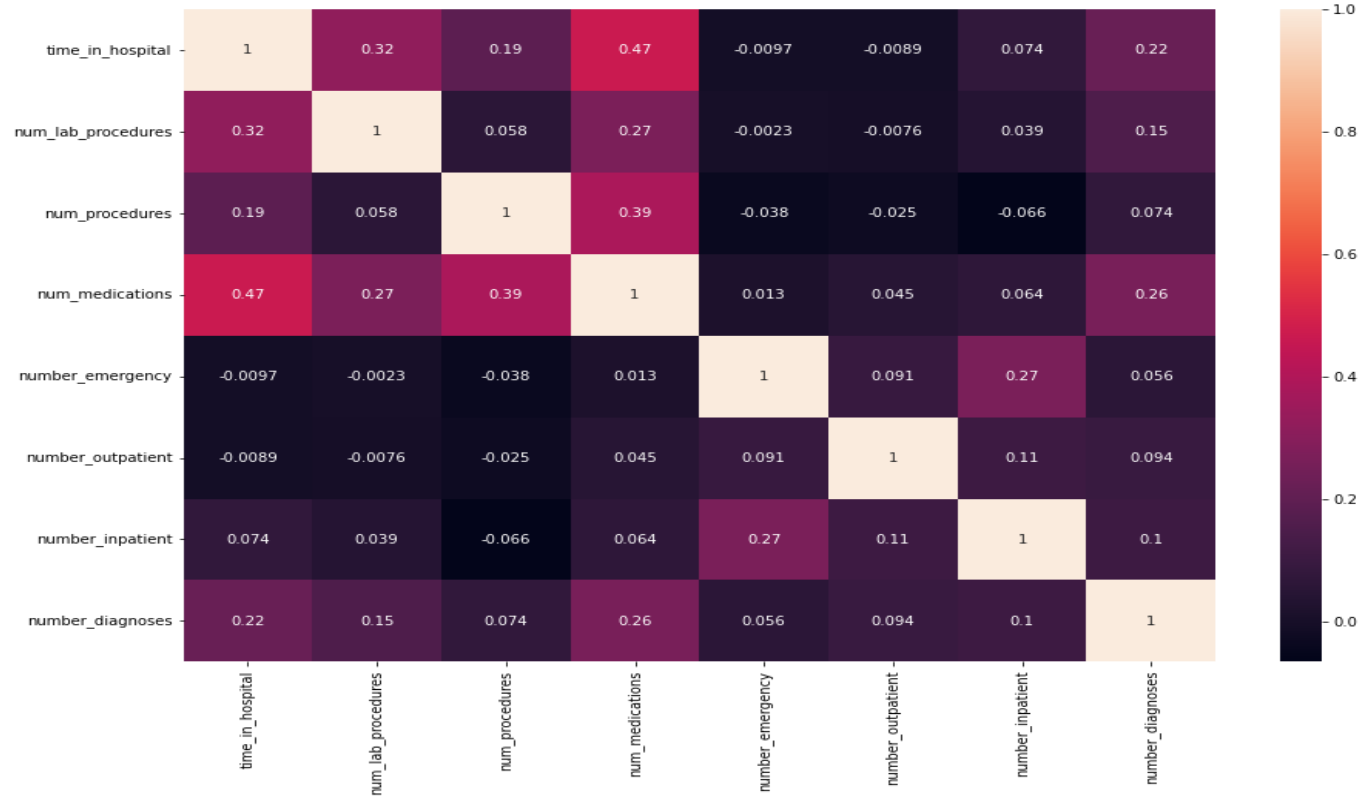
Readmitted :

Readmitted is the Target variable. As we want to predict if a patient will get readmitted or not. It is an imbalanced data.




```
Percentages of each class:
0    88.840084
1    11.159916
```

HeatMap for Numerical Values:

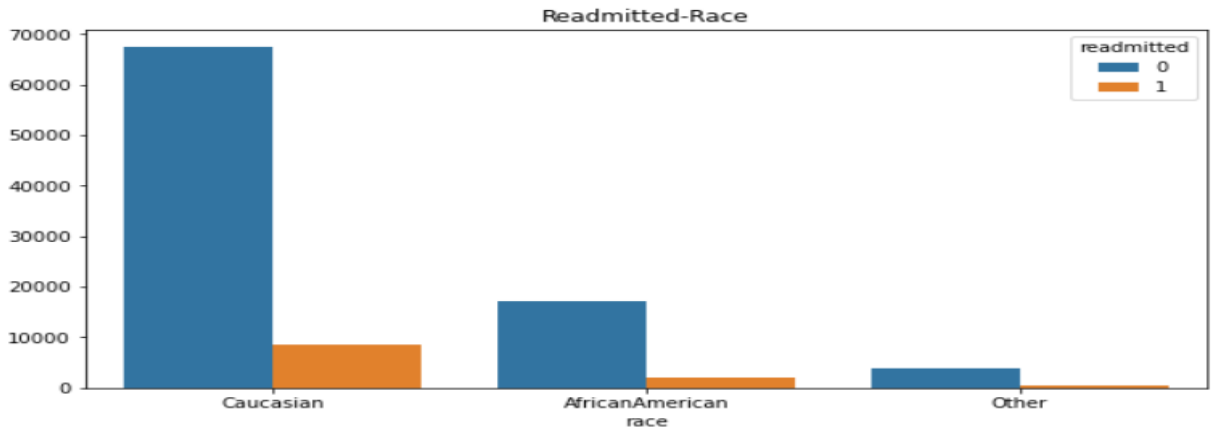


We can see the correlation between the numerical columns, there is no high correlation between them. The highest correlation for the given variables is 0.47 between num_mediacation and time_in_hospital.

Bi-variate analysis:

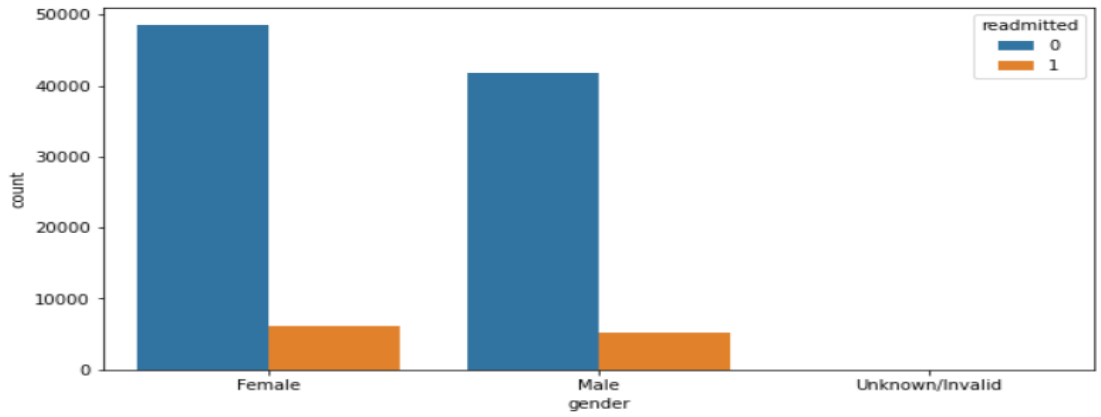
Race Vs Target Variable:

The Caucasian has more probability of readmission followed by African American and other.



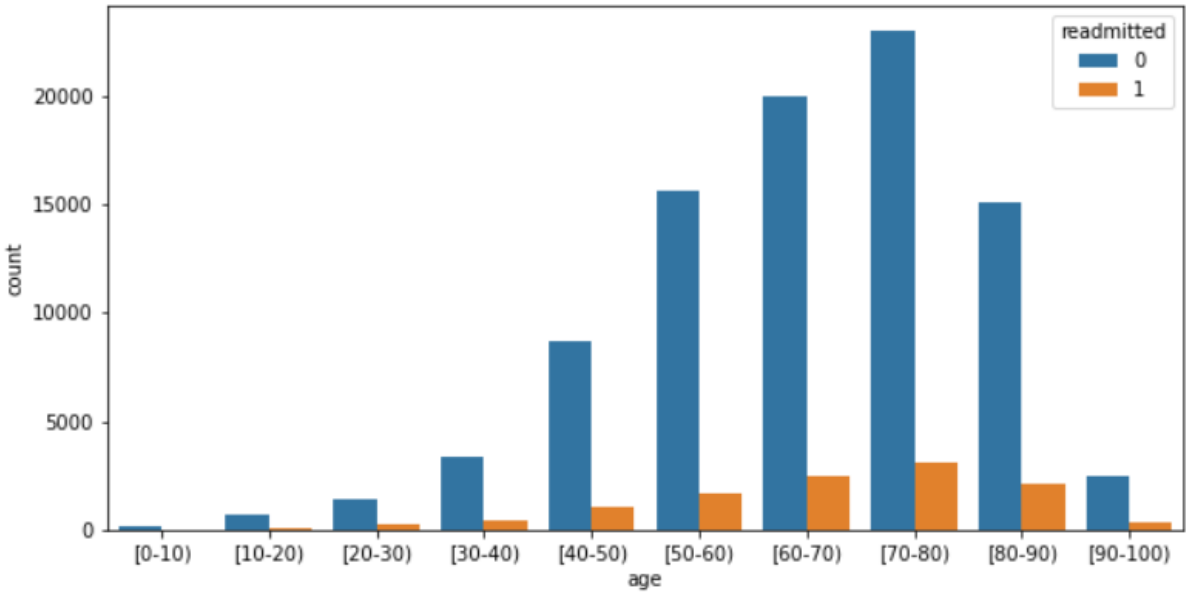
Gender with the target variable:

Readmission probability is slightly higher for Females than Males



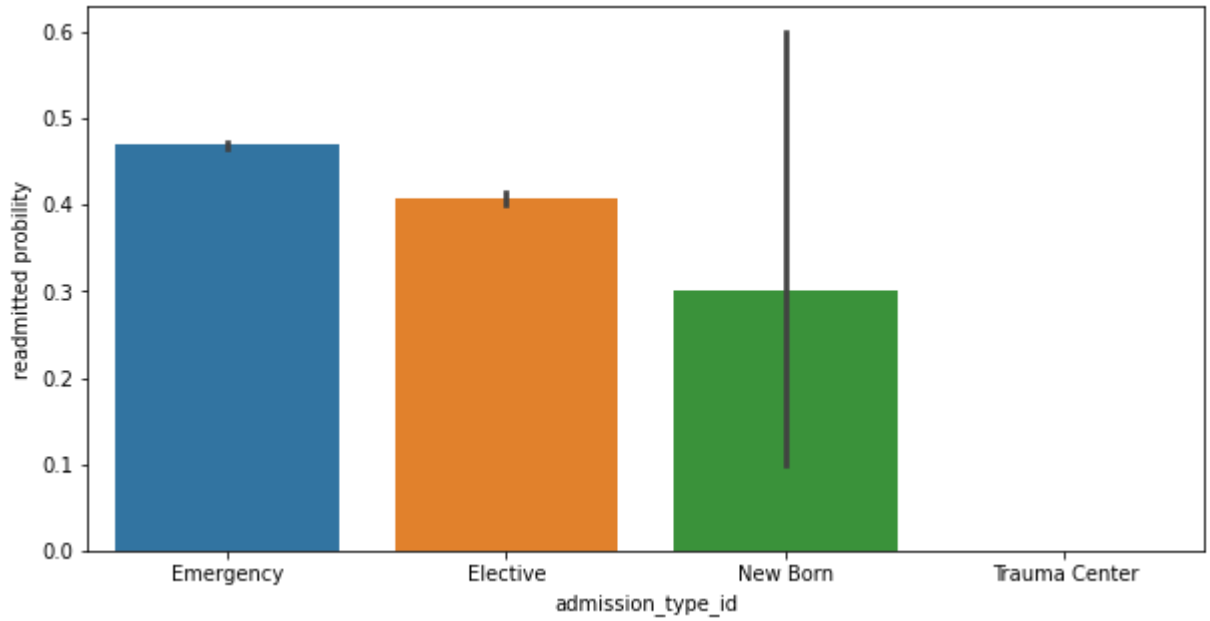
Age vs target variable:

Elderly age group has more readmission probability than other groups



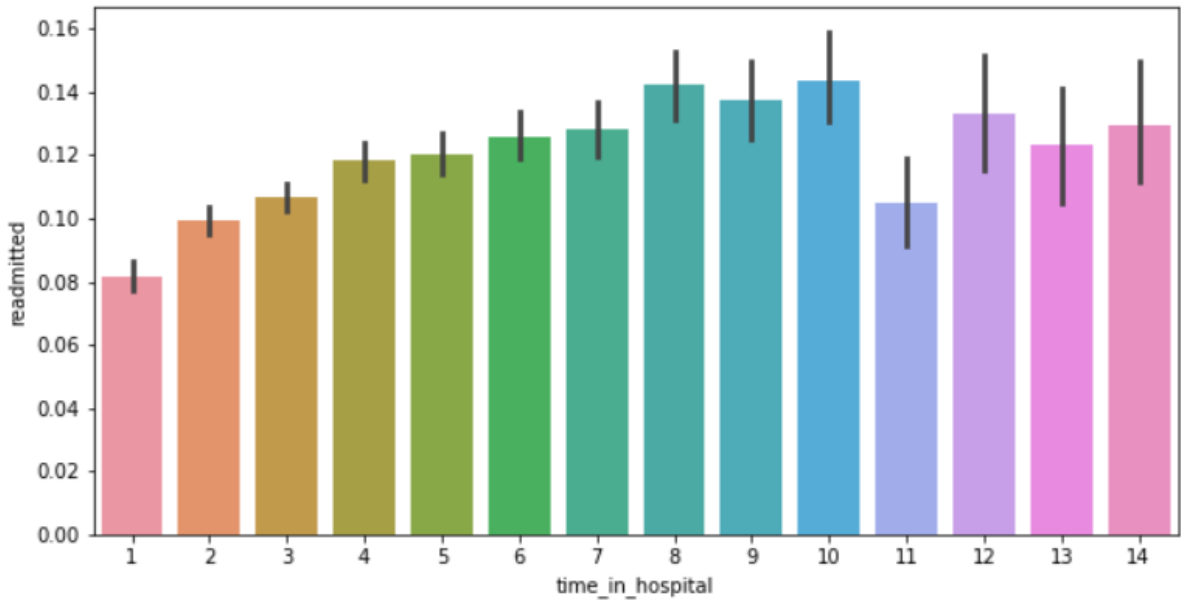
Admission type id with the target variable:

Readmission probability is higher for Emergency followed by elective.



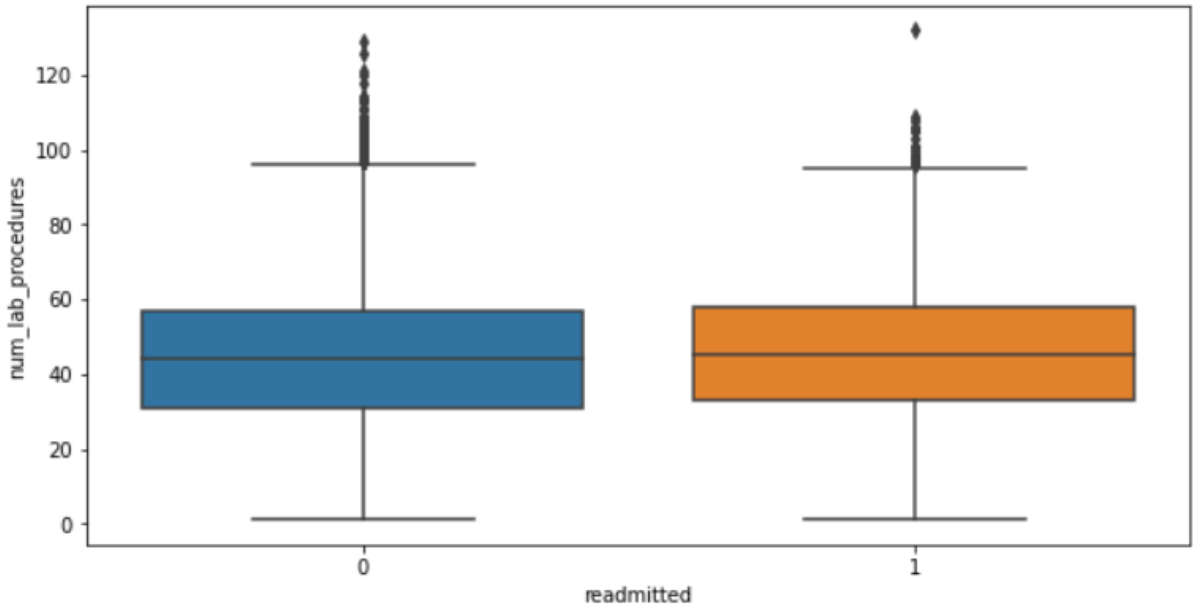
Time_in_hospital vs target variable:

Most people stayed for 2 to 3 days in the hospital.



Num_lab_procedures vs target variable:

Number of procedures is almost the same for both readmitted and not readmitted patients.

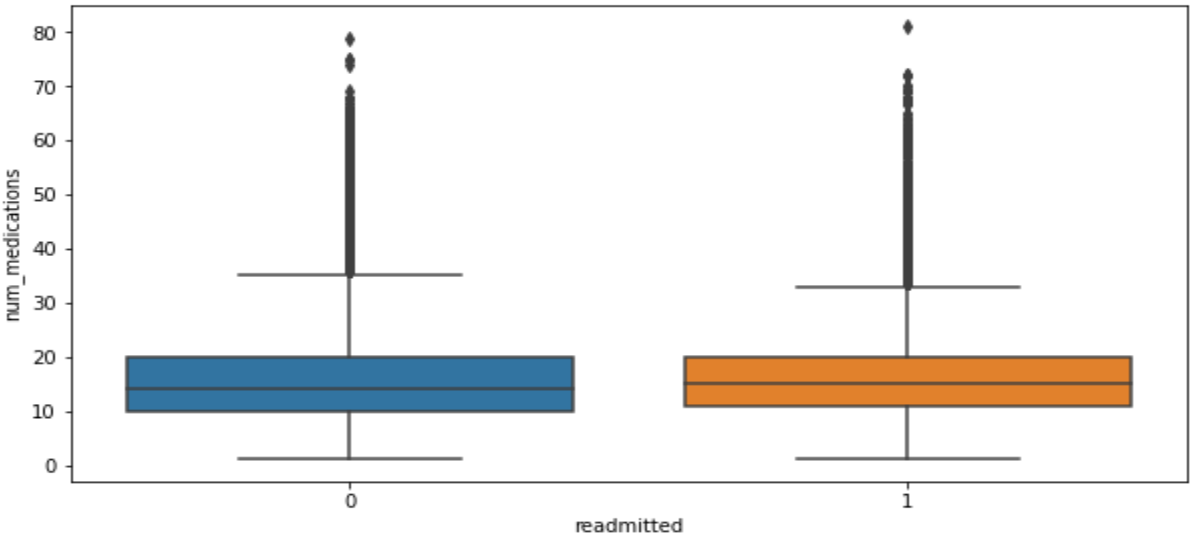


The greatest number of lab procedures are in the range 35 to 60

Number of lab procedures is slightly higher for readmitted than for not readmitted patients

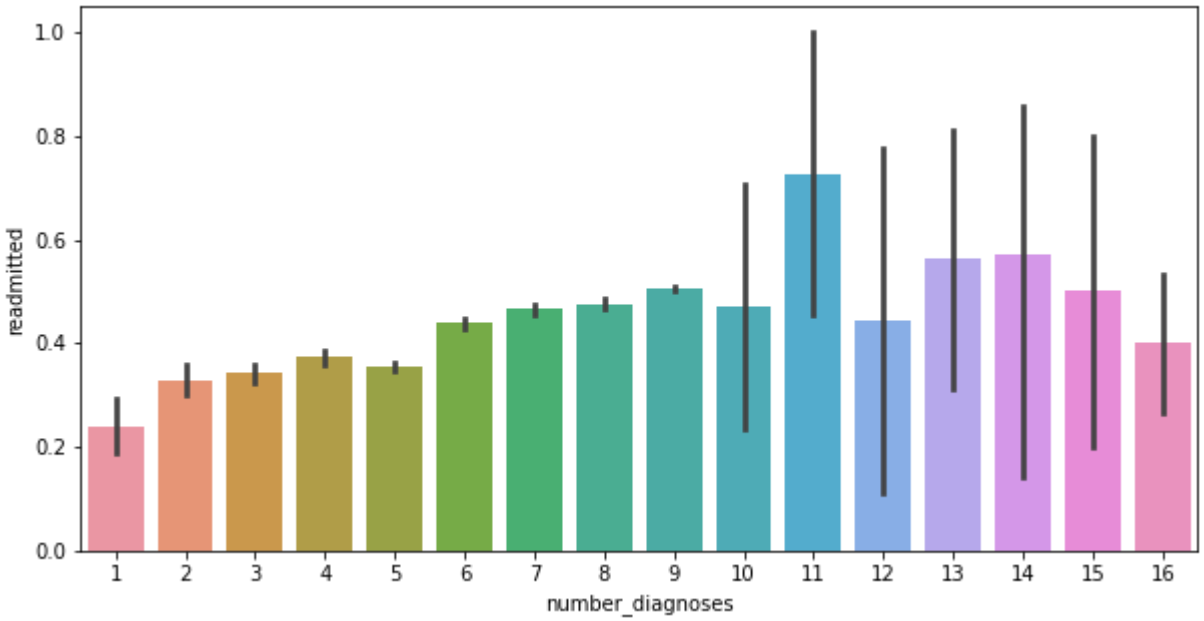
Num_medications vs target variable:

Number of medications is slightly higher for readmitted than for not readmitted patients



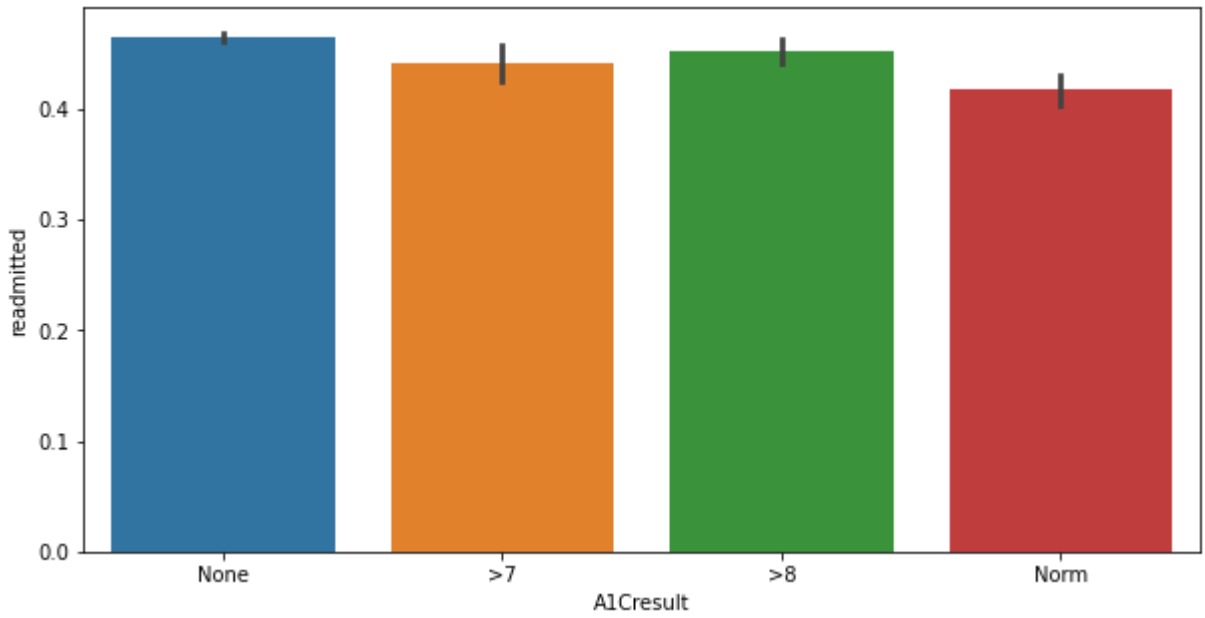
Number_diagnoses with the target variable:

Readmission probability is highest for higher number of diagnoses.



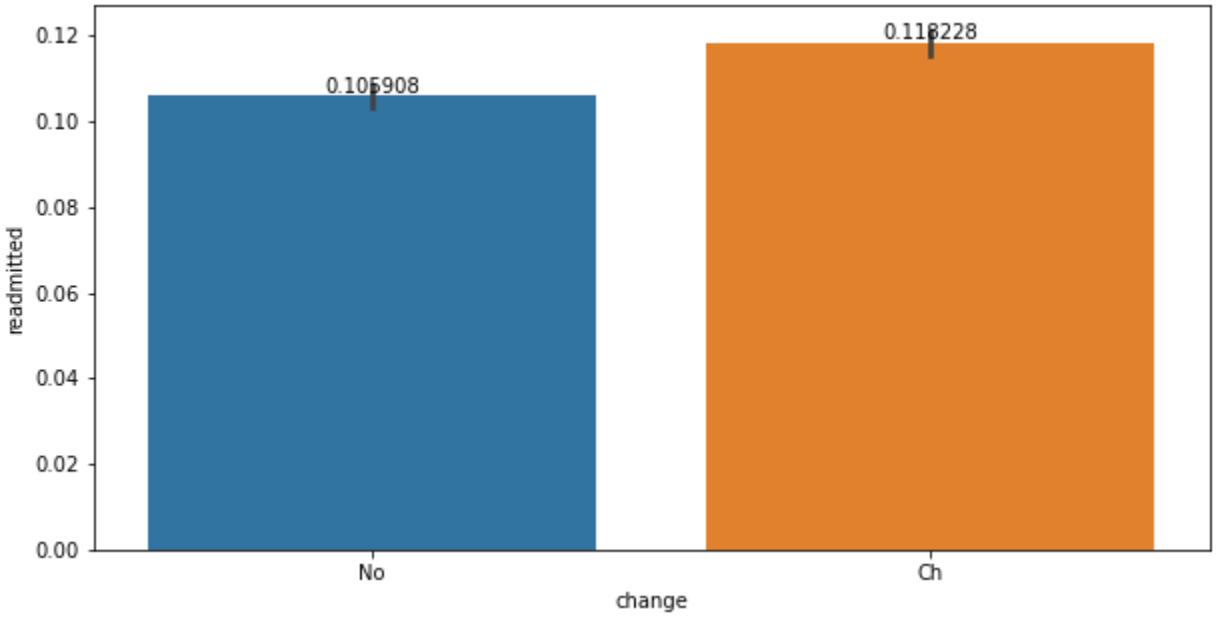
A1Cresults with the target variable:

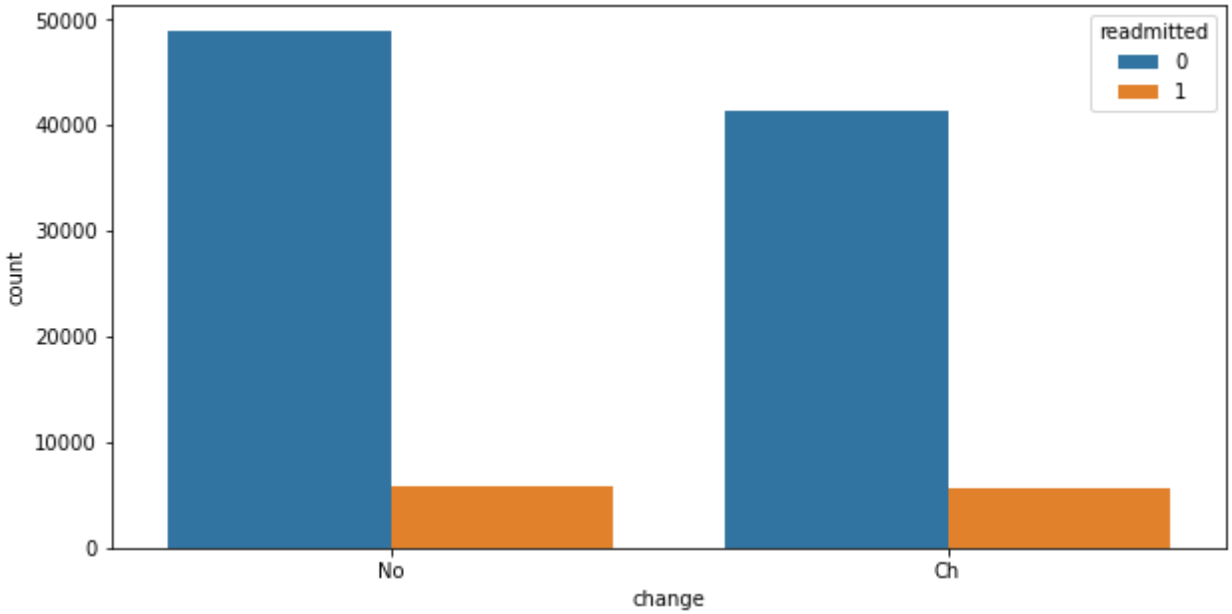
Readmission probability was highest for patients with no test conducted, followed by test results >8 and >7.



Change vs target variable:

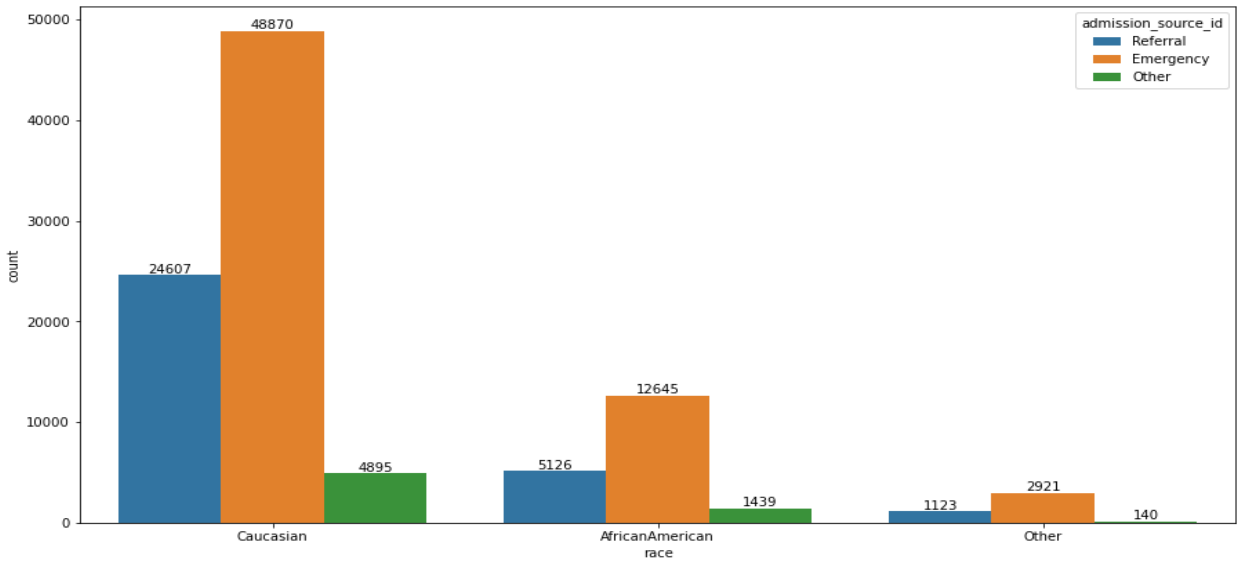
Medication dosage was changed for very few patients. Readmission Probability is higher for patients whose prescription was changed.

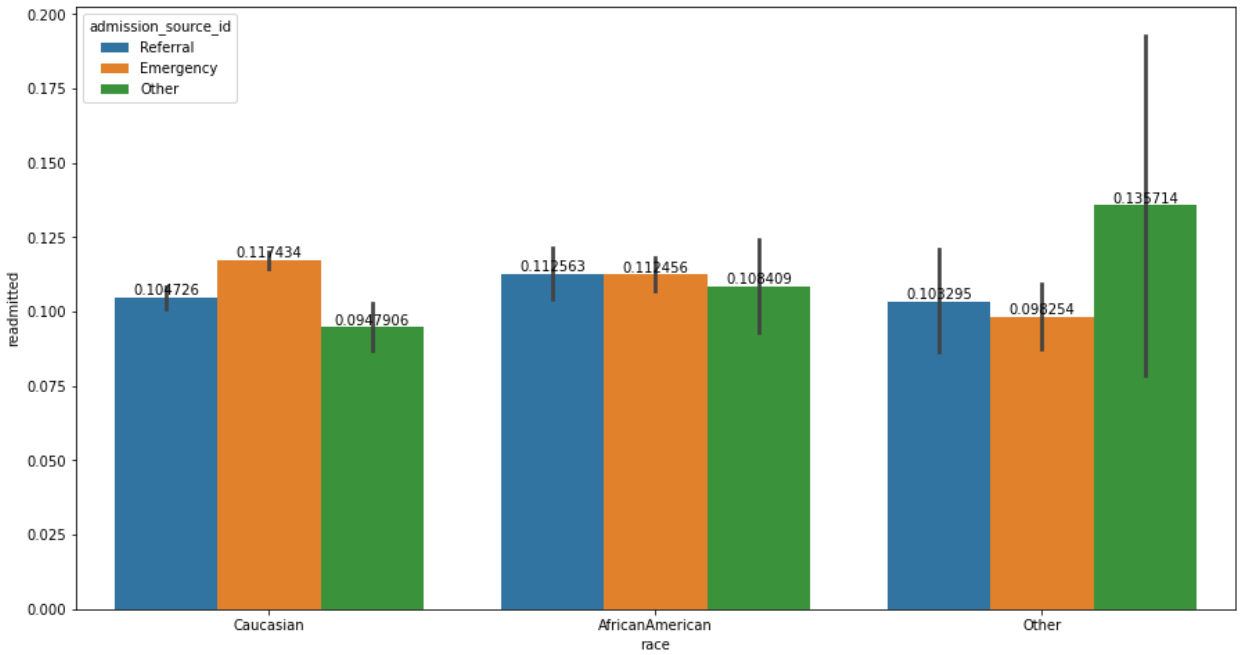




Race vs Admission_type_id vs Readmission:

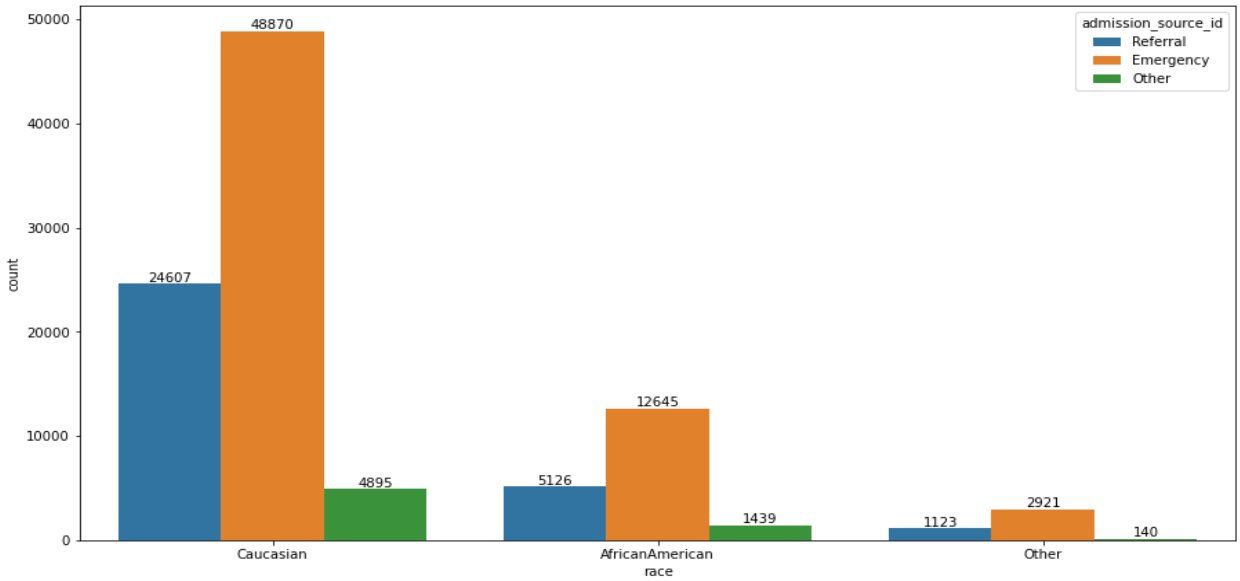
69% of the Caucasian of total Caucasians are of emergency type cases while 79% of African Americans of total African Americans are of emergency type cases





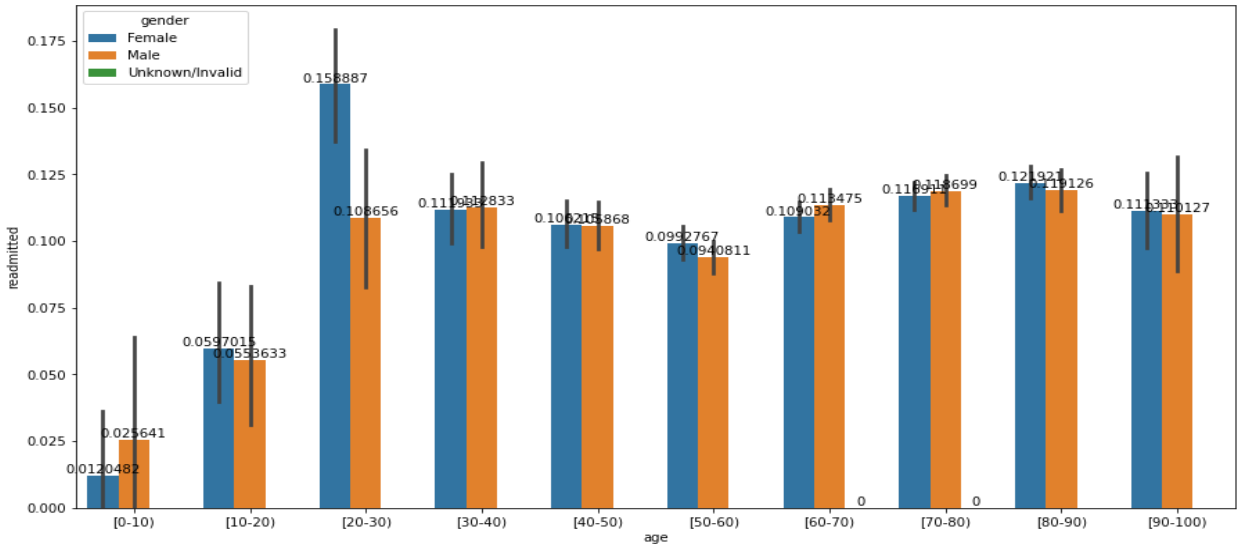
Race vs Admission_source_id:

60 % Caucasian are of emergency type, the probability of readmission is more for emergency type irrespective of race.

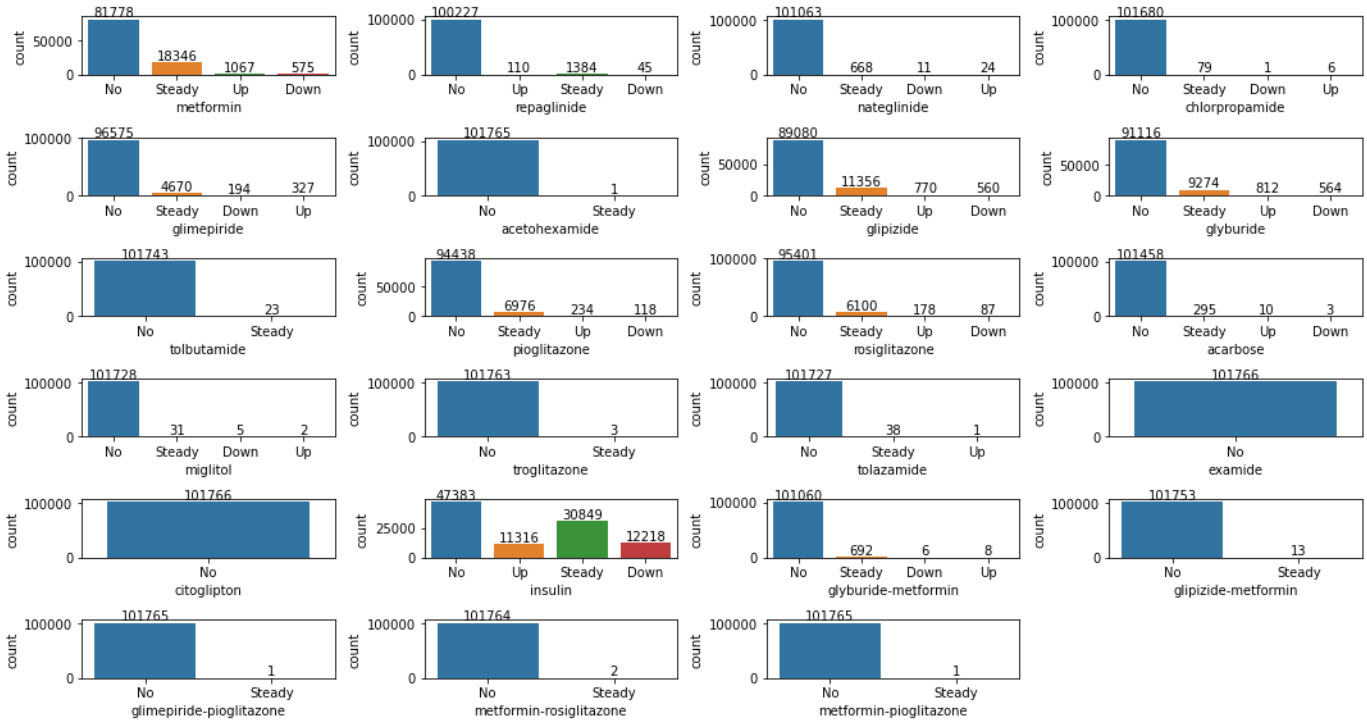


Gender vs Age vs Readmission:

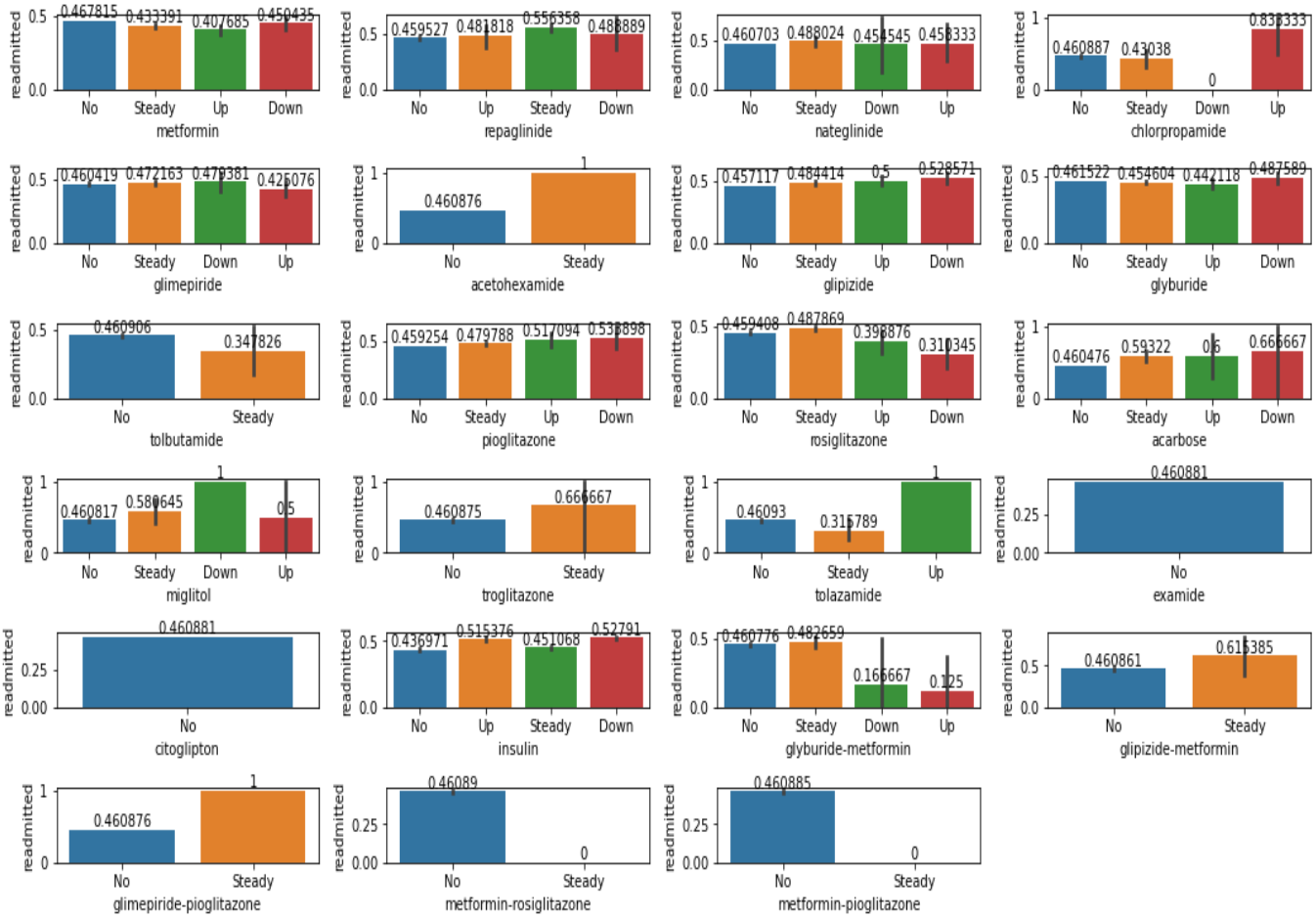
Irrespective of age in most of the age groups the probability of readmission is more in females.



Univariate and Bivariate analysis of medications:



For most of the patients, the medication was not prescribed Medications like examide, citoglipton, acetohexamide, troglitazone, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, metformin-pioglitazone have highly imbalance data. Very few patients were prescribed these medications, Variance is very low which makes these features redundant. Hence, dropped these features.



Readmission probability is higher for patients:

- who were not prescribed Metformin and whose dosage was decreased
- whose repaglinide dosage was not changed and whose dosage was decreased
- whose nateglinide dosage was not changed
- whose chlorpropamide dosage was increased and for who those who were not prescribed
- whose glipizide dosage was changed
- whose glyburide dosage was decreased
- whose pioglitazone dosage was changed
- whose rosiglitazone dosage was not changed and for those not prescribed
- whose insulin dosage was changed
- whose glyburide-metformin dosage was steady and for whom drug was not prescribed.

FEATURE ENGINEERING (STATISTICAL TESTING):

Chi-Square test of Independence:

- The Chi-Square test of independence is used to determine if there is a significant relationship between two nominal (categorical) variables.
- The hypothesis to test the independence of attributes

H_0: The attributes are independent

against

H_a: The attributes are dependent

	Feature	P-Value
0	race	0.177
1	gender	0.539
2	age	0.0
3	diag_1	0.0
4	diag_2	0.0
5	diag_3	0.0
6	max_glu_serum	0.001
7	A1Cresult	0.0
8	metformin	0.0
9	repaglinide	0.007
10	nateglinide	0.703
11	chlorpropamide	0.433
12	glimepiride	0.073
13	acetohexamide	0.723
14	glipizide	0.009
15	glyburide	0.201
16	tolbutamide	0.299
17	pioglitazone	0.094
18	rosiglitazone	0.142
19	acarbose	0.219
20	miglitol	0.162
21	troglitazone	0.539
22	tolazamide	0.766
23	examide	1.0
24	citoglipton	1.0
25	insulin	0.0
26	glyburide-metformin	0.756
27	glipizide-metformin	0.691
28	glimepiride-pioglitazone	0.723
29	metformin-rosiglitazone	0.616
30	metformin-pioglitazone	0.723
31	change	0.0
32	diabetesMed	0.0

- It is observed that nateglinide, chlorpropamide, glimepiride, acetohexamide, glyburide, tolbutamide, miglitol, troglitazone, tolazamide, examide, citoglipton, glyburide-

metformin, glipizide-metformin, glimepiride-pioglitazone, metformin- rosiglitazone, metformin-pioglitazone have the p value greater than 0.05, more than significance level, hence we accept the null hypothesis. Hence there is no significant impact of these features on the target variable. These can be dropped.

- The rest of the features have p value less than 0.05 significance level, hence we fail to accept the null hypothesis. Hence there is a significant impact of those on target variable.

KRUSKAL WALLIS TEST ON TARGET AND OTHER NUMERICAL FEATURES:

	Feature	P-Value
0	admission_type_id	0.0
1	discharge_disposition_id	0.0
2	admission_source_id	0.0
3	time_in_hospital	0.0
4	num_lab_procedures	0.0
5	num_procedures	0.051
6	num_medications	0.0
7	number_outpatient	0.0
8	number_emergency	0.0
9	number_inpatient	0.0
10	number_diagnoses	0.0

- It is observed that the pvalue for most independent numeric variables is less than 0.05 significance level except num_procedure feature having 0.051 p-value, hence we fail to accept the null hypothesis. They are all significant.
- All the columns are not normally distributed. Hence, we can't do ANOVA test. We are doing the Kruskal Wallis Test, which is a non- parametric version of ANOVA.

DATA PRE-PROCESSING:

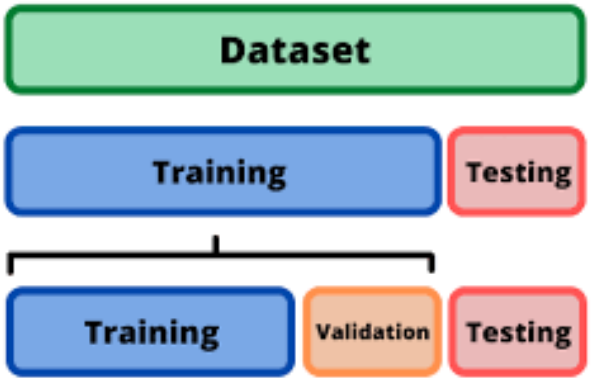
- Scaling done using MinMax Scaler from sklearn library on Numerical Columns

- Nominal Encoding done using get_dummies from Pandas library on Categorical columns with less number of Classes
- For Categorical columns with a greater number of classes, Encoding done using Weight of Evidence (WOE) from Xverse package

MODEL BUILDING:

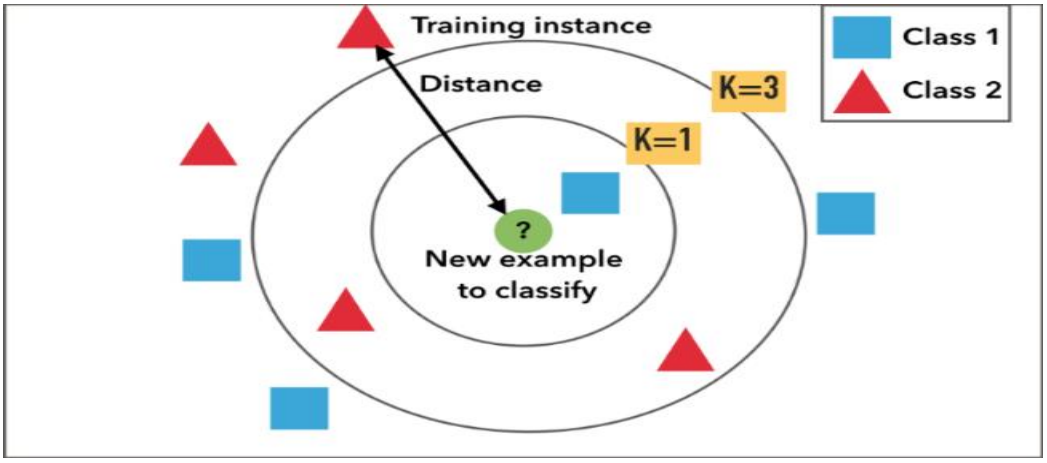
Setting up methods for data collection, comprehending and paying attention to what is significant in the data to address the questions you are posing, and creating a statistical, mathematical, or simulation model to acquire understanding and make predictions are all part of the model-building process.

We divided the data into train data and test data after completing Nominal Encoding.



BASE MODEL :K-Nearest Neighbour (KNN) Algorithm:

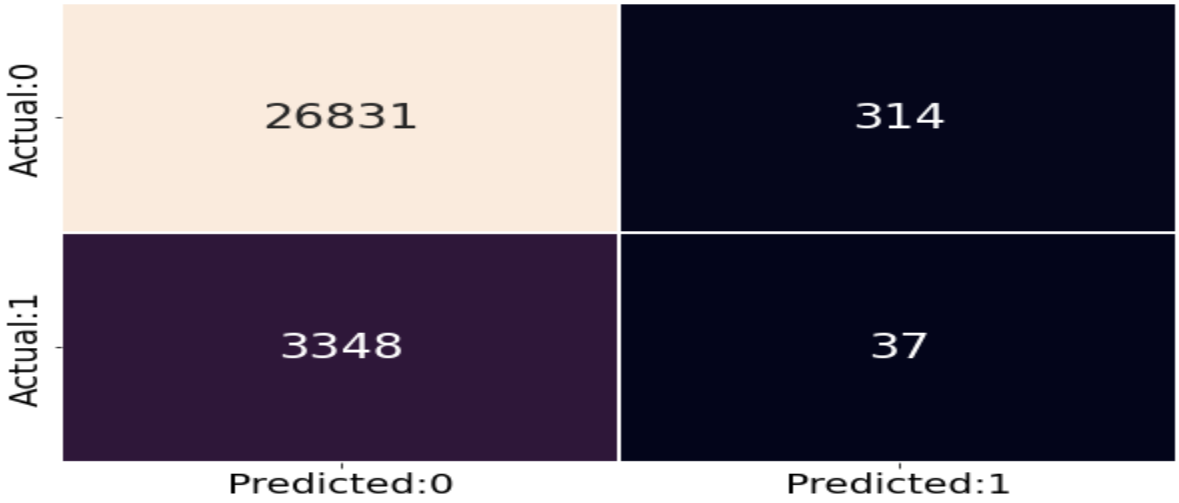
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories
- We are using K-NN as it is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.



PERFORMANCE METRICS:

- **Confusion Matrix:**

  It is the performance measure for the classification problem. It is a table used to compare predicted and actual values of the target variable.



- An ideal model for our problem statement should reduce the number of false negatives as it will predict that a patient will not be readmitted when he/she is likely to be readmitted
- We'll need to tune the models so as to reduce the number of false positives and false negatives

Classification Report :

For Train Data:

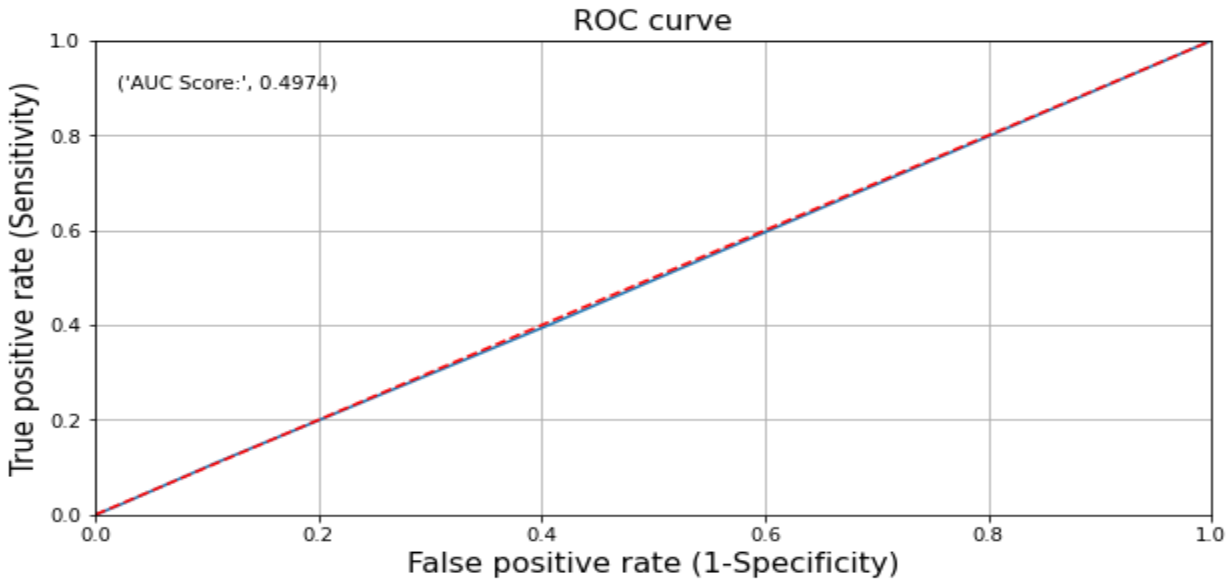
	precision	recall	f1-score	support
0	0.89	0.99	0.94	63264
1	0.60	0.06	0.11	7972
accuracy			0.89	71236
macro avg	0.75	0.53	0.53	71236
weighted avg	0.86	0.89	0.85	71236

For Test Data:

	precision	recall	f1-score	support
0	0.89	0.99	0.94	27145
1	0.11	0.01	0.02	3385
accuracy			0.88	30530
macro avg	0.50	0.50	0.48	30530
weighted avg	0.80	0.88	0.83	30530

Receiver Operating Characteristics Curve:

ROC curve, also known as Receiver Operating Characteristics Curve, is a metric used to measure the performance of a classifier model. The ROC curve depicts the rate of true positives with respect to the rate of false positives, therefore highlighting the sensitivity of the classifier model.



AUC-ROC is the valued metric used for evaluating the performance in classification models. The AUC-ROC metric clearly helps determine and tell us about the capability of a model in distinguishing the classes.

Interpretation:

The red dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner).

From the above plot, we can see that our classifier (logistic regression) is on the dotted line; with the AUC score 0.4974

PERFORMANCE EVALUATION METRICS:

- **Accuracy:**

  Accuracy is the fraction of predictions that our model got correct. Higher the accuracy of the model the better is the model.

  The accuracy is 0.88 for test data.

- **Precision:**

  Precision is the proportion of positive cases that were correctly predicted.

  The precision score is 0.11 for readmitted class

- **Recall:**

A recall is the proportion of actual positive cases that were correctly predicted.

Recall for readmitted is 0.01

Recall for not readmitted class is 0.99

- **F1 score:**

  F1 score is the harmonic mean of precision and recall values for a classification model.

  F1 score for not readmitted class is 0.94

  F1 score for readmitted class is 0.02

Inferences from Base Model:

For our model, the cost we want to reduce is **False negatives** mainly and then False positives. Also, our data is **heavily imbalanced** which is leading to an **accuracy paradox** i.e., even though the overall accuracy is very high, the model is able to predict only the majority class properly and not the other one. This can be addressed using SMOTE and considering f1 score as performance metric.

As False negatives impact our problem statement more, we are considering recall as our performance metric. We can see that **Recall for our interested class (1) is 0.01** which is very low.

So, we worked on improving the **Accuracy, f1 score, AUC score and mainly Recall score**.

Over-Sampling and Under-Sampling:

An imbalanced classification problem is an example of a classification problem where the distribution of examples across the known classes is biased or skewed. The distribution can vary from a slight bias to a severe imbalance where there is one example in the minority class for hundreds, thousands, or millions of examples in the majority class or classes.

Imbalanced classifications pose a challenge for predictive modeling as most of the machine learning algorithms used for classification were designed around the assumption of an equal number of examples for each class. This results in models that have poor predictive performance, specifically for the minority class. This is a problem because typically, the minority class is more important and therefore the problem is more sensitive to classification errors for the minority class than the majority class.

Imbalanced classification refers to a classification predictive modelling problem where the number of examples in the training dataset for each class label is not balanced. That

is, where the class distribution is not equal or close to equal, and is instead biased or skewed.

One approach to addressing the problem of class imbalance is to randomly resample the training dataset. The two main approaches to randomly resampling an imbalanced dataset are to delete examples from the majority class, called under-sampling, and to duplicate examples from the minority class, called over-sampling.

Random over-sampling involves randomly duplicating examples from the minority class and adding them to the training dataset.

Examples from the training dataset are selected randomly with replacement. This means that examples from the minority class can be chosen and added to the new “more balanced” training

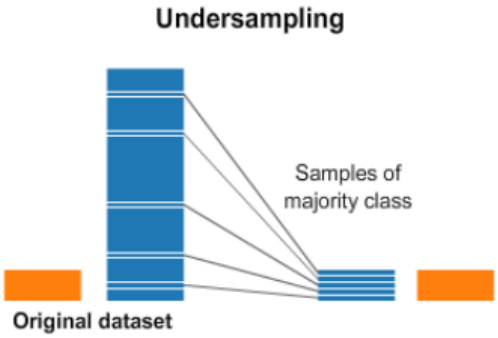
dataset multiple times; they are selected from the original training dataset, added to the new training dataset, and then returned or “replaced” in the original dataset, allowing them to be selected again.

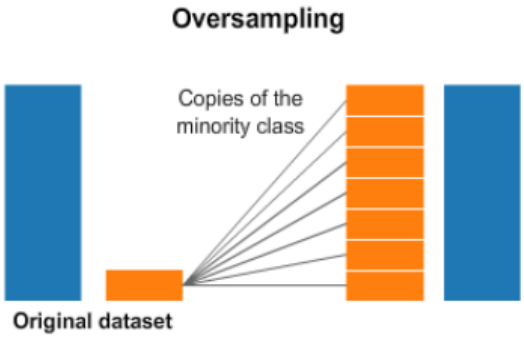
In some cases, seeking a balanced distribution for a severely imbalanced dataset can cause affected algorithms to overfit the minority class, leading to increased generalization error. The effect can be better performance on the training dataset, but worse performance on the holdout or test dataset.

Random under-sampling involves randomly selecting examples from the majority class to delete from the training dataset.

This has the effect of reducing the number of examples in the majority class in the transformed version of the training dataset. This process can be repeated until the desired class distribution is achieved, such as an equal number of examples for each class.

A limitation of under-sampling is that examples from the majority class are deleted that may be useful, important, or perhaps critical to fitting a robust decision boundary. Given that examples are deleted randomly, there is no way to detect or preserve “good” or more information-rich examples from the majority class.





BEFORE SMOT

```
y.value_counts(normalize=True)*100
```

```
0    90.938915
1     9.061085
Name: readmitted, dtype: float64
```

AFTER SMOT

```
from imblearn.over_sampling import SMOTE
```

```
x1 = pd.concat((xtrain,xtest),0)
y1 = pd.concat((ytrain,ytest),0)
y1 = y1.astype(int)
sme = SMOTE(sampling_strategy=0.4)
x_sme,y_sme = sme.fit_resample(x1,y1)
```

```
0    71.428897
1    28.571103
Name: readmitted, dtype: float64
```

CHANGES MADE AFTER MODEL:

- Binning on following columns:
    1. Admission Source - Binned 26 distinct subclasses based on domain knowledge as

```
Emergency    56799
Referral     30307
Other         6976
Transfer      6162
```

2. Admission Type ID: Binned 8 distinct subclasses based on domain knowledge as

Emergency	69489
Elective	18355
NA	9981

3. Discharge Disposition ID: Binned 30 distinct subclasses based on domain knowledge into 6 distinct categories. We later dropped the Hospice/Expired column as they don't make sense when we are dealing with readmission giving 5 distinct categories.

Home	59008
Discharged_AHC	20678
Home_HC	12937
Unknown	4598
Hospice/Ex	2419
AMA	604

Home	59008
Discharged_AHC	20678
Home_HC	12937
Unknown	4598
AMA	604

- Feature Engineering:
    1. From the 23 medication columns, examide and citoglipton were dropped as they had single value in all the rows making them redundant
    2. In the rest 21 columns, Up and Down which represent medication dosage changed to 1 and No and Steady to 0
    3. Then all these 21 columns were added together to give Number of changes in medications column and then dropped
- Transforming independent variables:
    1. As we've observed that numerical columns are skewed, we've done different transformation techniques
    2. We've down square root transformation on time_in_hospital, num_procedures, num_medications columns and 1/x transformation on number_outpatient, number_emergency, number_inpatient, num_change columns

Logistic Regression model:

By estimating probabilities using the underlying logistic function, logistic regression determines the relation between the dependent variable (our label, what we want to predict), and one or more independent variables (our features).

Test Data

	precision	recall	f1-score	support
0	0.72	0.98	0.83	15563
1	0.55	0.06	0.10	6363
accuracy			0.71	21926
macro avg	0.63	0.52	0.47	21926
weighted avg	0.67	0.71	0.62	21926

Decision Tree Model:

It is a tree structured classifier, where internal node represents the feature of a dataset, branches represent the decision rules and each leaf node represent outcome.

For Test Data:

	precision	recall	f1-score	support
0	0.90	0.89	0.89	15563
1	0.73	0.77	0.75	6363
accuracy			0.85	21926
macro avg	0.82	0.83	0.82	21926
weighted avg	0.85	0.85	0.85	21926

Naive Bayes Classifiers:

Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e., every pair of features being classified is independent of each other.

For Test Data:

	precision	recall	f1-score	support
0	0.85	0.05	0.09	15563
1	0.30	0.98	0.45	6363
accuracy			0.32	21926
macro avg	0.57	0.51	0.27	21926
weighted avg	0.69	0.32	0.20	21926

From the classification report accuracy was decreased when compare to the other models and recall was improved for readmission and recall was decreased for no readmission

Hyperparameter tuning random forest classifier:

- n_estimators = number of trees in the forest
- max_depth = max number of levels in each decision tree
- min_samples_split = min number of data points placed in a node before the node is split
- min_samples_leaf = min number of data points allowed in a leaf node

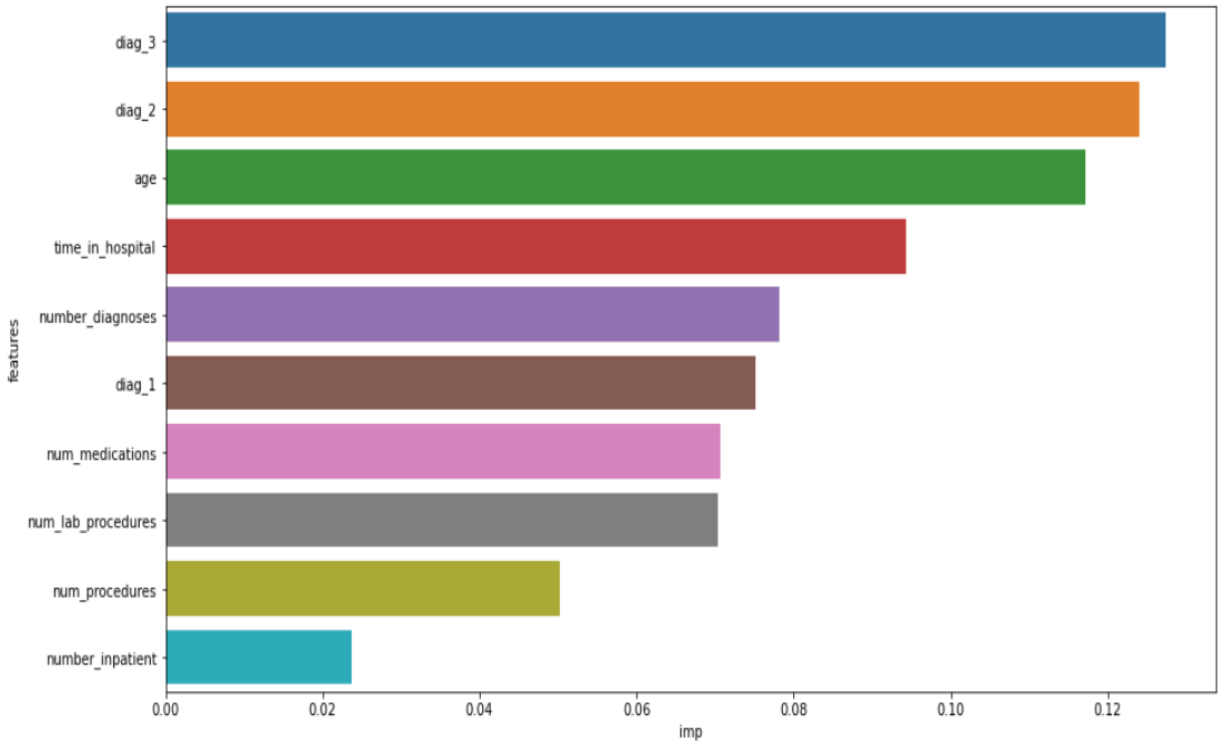
```
gd.best_params_
```

```
{'criterion': 'gini',
 'max_depth': 50,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'n_estimators': 50}
```

For Test Data:

	precision	recall	f1-score	support
0	0.91	1.00	0.95	15563
1	1.00	0.74	0.85	6363
accuracy			0.92	21926
macro avg	0.95	0.87	0.90	21926
weighted avg	0.93	0.92	0.92	21926

Feature Importances:



Adaboost classifier:

Adaboost is the ensemble learning method. The most common algorithm used with Adaboost is decision trees with one level that means with Decision trees with only one split. AdaBoost uses an iterative approach to learn from the mistakes of weak classifiers and turn them into strong ones.

For Test Data:

	precision	recall	f1-score	support
0	0.86	1.00	0.93	15563
1	0.98	0.62	0.76	6363
accuracy			0.89	21926
macro avg	0.92	0.81	0.84	21926
weighted avg	0.90	0.89	0.88	21926

Hyperparameter tuning XGboost classifier:

XGBoost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

We can use XGBoost for any supervised machine learning task when satisfies the following criteria:

- When you have large number of observations in training data.
- Number features < number of observations in training data.
- It performs well when data has mixture numerical and categorical features or just numeric features.
- When the model performance metrics are to be considered.

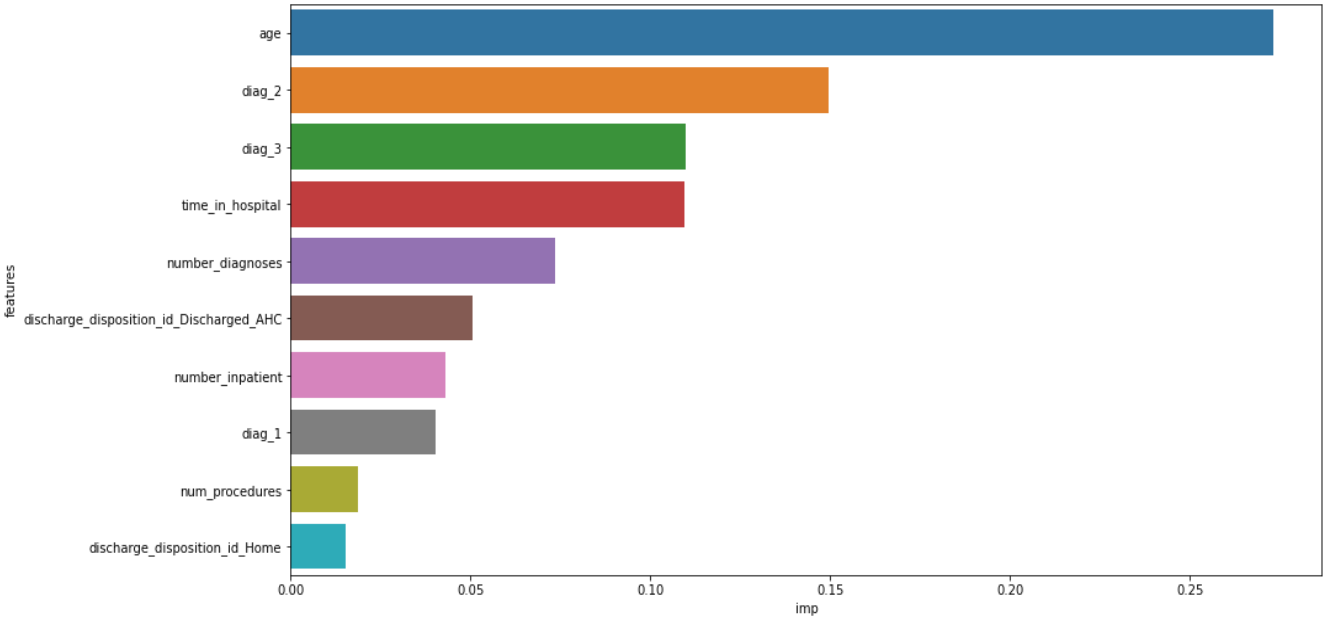
```
xgb_grid.best_params_
```

```
{'gamma': 0, 'learning_rate': 0.4, 'max_depth': 4}
```

For Test Data:

	precision	recall	f1-score	support
0	0.91	1.00	0.95	15563
1	1.00	0.75	0.86	6363
accuracy			0.93	21926
macro avg	0.95	0.88	0.90	21926
weighted avg	0.93	0.93	0.92	21926

Feature Importances:



In XGBoost, Hyper parameter tuned model, we can observe increased accuracy and also the recall for the readmission compared to the base model.

Stacking Classifier using Hyper Parameter Tuned Random Forest and XGBoost:

Stack of estimators with a final classifier. Stacked classifier stacks the output of individual estimators and uses a final classifier at the end to compute the final prediction. Stacking allows us to use the strength of each individual estimator by using their output as input of a final estimator.

Estimators used:

1. Above Hyperparameter tuned Random Forest
2. Above Hyperparameter tuned XGBoost Classifier

Final Estimator used: Logistic Regression

For Test Data:

	precision	recall	f1-score	support
0	0.91	0.99	0.95	15563
1	0.98	0.77	0.86	6363
accuracy			0.93	21926
macro avg	0.95	0.88	0.91	21926
weighted avg	0.93	0.93	0.93	21926

For Train Data:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	47083
1	1.00	1.00	1.00	18695
accuracy			1.00	65778
macro avg	1.00	1.00	1.00	65778
weighted avg	1.00	1.00	1.00	65778

Inferences:

The Test scores are best for this **Stacking classifier** using hyperparameter tuned random forest and XGBoost. We can see that all the scores are 1 for train data which are varying from test scores indicating overfitting. Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on test/new data. **Overfitting** can be addressed by decreasing the number of columns, decreasing the model complexity or regularisation.

Hence, we built a Stacking Classifier using Significant columns given by Recursive Feature Elimination.

FINAL MODEL

We've done Recursive Feature elimination (RFE) on XGBoost Classifier. RFE is an algorithm which is effective at selecting those features (columns) in a training dataset that are more or most relevant in predicting the target variable and got the following columns as the most important features.

```
Index(['age', 'time_in_hospital', 'num_procedures', 'num_medications',
       'number_inpatient', 'diag_1', 'diag_2', 'diag_3', 'number_diagnoses',
       'discharge_disposition_id_Discharged_AHC',
       'discharge_disposition_id_Home', 'diabetesMed_Yes'],
      dtype='object')
```

Then we've applied the Stacking classifier (Hyperparameter tuned Random Forest and XGBoost) on these columns.

For Test Data:

	precision	recall	f1-score	support
0	0.91	1.00	0.95	15563
1	0.99	0.76	0.86	6363
accuracy			0.93	21926
macro avg	0.95	0.88	0.90	21926
weighted avg	0.93	0.93	0.92	21926

For Train Data:

	precision	recall	f1-score	support
0	0.92	1.00	0.96	47083
1	1.00	0.78	0.88	18695
accuracy			0.94	65778
macro avg	0.96	0.89	0.92	65778
weighted avg	0.94	0.94	0.93	65778

We can observe that Overfitting has reduced significantly, almost to None. The Recall improved from 0.01 in the base model to 0.76 with overall accuracy 0.93 in the final model. From the Scorecard below, we can see that overall scores are best for the Stacking Classifier with Important features.

	Model	Accuracy	Recall	Precision	F1Score	Cohen_kappa_score	AUCScore
0	M1: Base Model-KNeighbours	0.88	0.01	0.11	0.02	0.00	0.50
1	M2: Decision Tree	0.85	0.77	0.73	0.75	0.65	0.83
2	M3: Logistic Regression with significant varia...	0.71	0.07	0.54	0.13	0.06	0.52
3	M4: Random Forest	0.92	0.74	1.00	0.85	0.80	0.87
4	M5: DT Hyperparameter tuned	0.85	0.77	0.74	0.75	0.65	0.83
5	M6: RF - Hyper Parameter Tuned	0.92	0.74	1.00	0.85	0.80	0.87
6	M7: Naive Bayes	0.32	0.98	0.30	0.46	0.02	0.52
7	M8: AdaBoost	0.89	0.62	0.99	0.76	0.69	0.81
8	M9: Gradient Boosting	0.92	0.72	1.00	0.84	0.78	0.86
9	M10: GB Hyperparameter Tuned	0.93	0.75	0.99	0.85	0.81	0.87
10	M11: XGB Hyperparameter tuned	0.93	0.75	1.00	0.86	0.81	0.88
11	M12: Stacking(RF+XGB,LR)	0.93	0.77	0.98	0.86	0.82	0.88
12	Final Model: RFE Stacking(RF+XGB,LR)	0.93	0.76	0.99	0.86	0.81	0.88

MODEL VALIDATION:

Model validation refers to the process of confirming that the model actually achieves its intended purpose i.e., how effective our model is.

KFold cross Validation

KFold cross validation is a resampling procedure used to evaluate machine learning models on a limited data sample

With Number of splits = 10,

Mean Accuracy: 0.9277455

Mean Recall: 0.749098

INFERENCES:

- The following features had the most impact on readmission probability:

  Age, Time spent in hospital, number of procedures, number of medications, number of inpatient visits, primary diagnosis, secondary diagnosis, tertiary diagnosis, number of diagnoses, Discharged to Another Health Care, Discharged to Home, On Diabetes medications or not
- Elderly age group 50 to 80 have higher chances of getting readmitted within 30 days
- People who have spent a greater number of days in hospital have higher chances of getting readmitted within 30 days
- Most number of medications are in the range of 10 to 20. Number of medications is slightly higher for readmitted than for not readmitted patients
- Patients with higher number of inpatient visits have higher chances of getting readmitted within 30 days
- Primary, Secondary and Tertiary diagnosis have an impact on readmission rates. People diagnosed with Respiratory, Circulatory and Diabetes type have higher chances of getting readmitted within 30 days
- Before discharging patients to home or another health care facility, proper care has to be taken to check if their Sugar levels are in control

BUSINESS JUSTIFICATION:

- The main motive is to improve the performance of the model. We aim to predict the hospital readmission rates using many variables that are influencing the outcome.
- The model is built to evaluate historical trends of diabetes care in diabetic patients admitted to a US hospital and to identify future paths that could lead to patient safety improvements. We are looking at the use of two tests, Maximum Glucose serum and HbA1c test as a marker of diabetes care in a large number of people who have been diagnosed with diabetes.

LIMITATIONS OF DATA:

Few of the limitations are: -

1. The dataset belongs to only the United States of America. The model will be more robust if the data would have belonged from different regions of the world.
2. Also, the duration of data collected is from 1999 to 2008. Due to this there isn't even distribution of the data and doesn't represent the present scenario well
3. The time stamps for the encounters aren't mentioned which was challenging as there were multiple encounters for each patient
4. The quantities of dosages weren't mentioned for medications due to which we couldn't analyse the effects of a particular medicine on readmission
5. Data of patient's medical history wasn't there, which could have been better for understanding the model and model building

CHALLENGES:

Few of the challenges faced are: -

1. High cardinality results in huge training effort in model tuning due to increase in model complexity (i.e., more number of features)
2. Due to high imbalance in the data, we used oversampling technique SMOTE. This added about 20,000 rows of artificial data. Using an under-sampling technique would've reduced the overall size of the data by a huge number. Dealing with imbalanced data was challenging
3. As we have knowledge only about machine learning models, we were limited by those. Better models could've been built using deep learning models for this dataset.

SCOPE:

Scope for some future work is: -

1. Couldn't give more hyperparameters while performing hyper parameter tuning for the Random Forest and XGBoost model since due to lower processing power of our laptops. By giving more number of hyper parameters with varied values would probably give us better hyper parameters

2. Exploring Google collab as an option for model training and tuning with faster lead time.
3. Exploring different Deep Learning Models
4. Exploring some robust data sampling technique other than SMOTE (a true representation of population data) from the population data.

References:

- **https://www.kaggle.com/datasets/brandao/diabetes**
- Research Article Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records
- https://miro.medium.com/max/650/0*Sk18h9op6uK9EpT8.