PROJECT REPORT

Statistical Learning for Engineering
IE 7300

Prof. Ramin Mohammadi

Readmission Analysis of a Diabetes Patient

Group 8

Anshita Aishwarya
Amar Sai Kiran Poosarla
Bijin Rao
Manoj Kondlay

ABSTRACT:

Diabetes is a chronic disease and one of the leading factors of deaths in the recent times. It is widespread among a vast population and majorly contributes to several other diseases. The term 'Hospital Readmission' refers to when a patient discharged from a hospital has to be re-admitted within a certain time interval. The patients hospitalized with diabetes have a higher risk of being readmitted than the ones who weren't hospitalized. The rate of hospital readmission is an indicator of the hospital quality as well as the treatment being performed. With this project, we aim to identify the likelihood of hospital readmission of a diabetic patient. This analysis will help the hospitals in identifying the factors that lead to higher readmission of patients and thereby, improving the quality of their care or changing medications being offered.

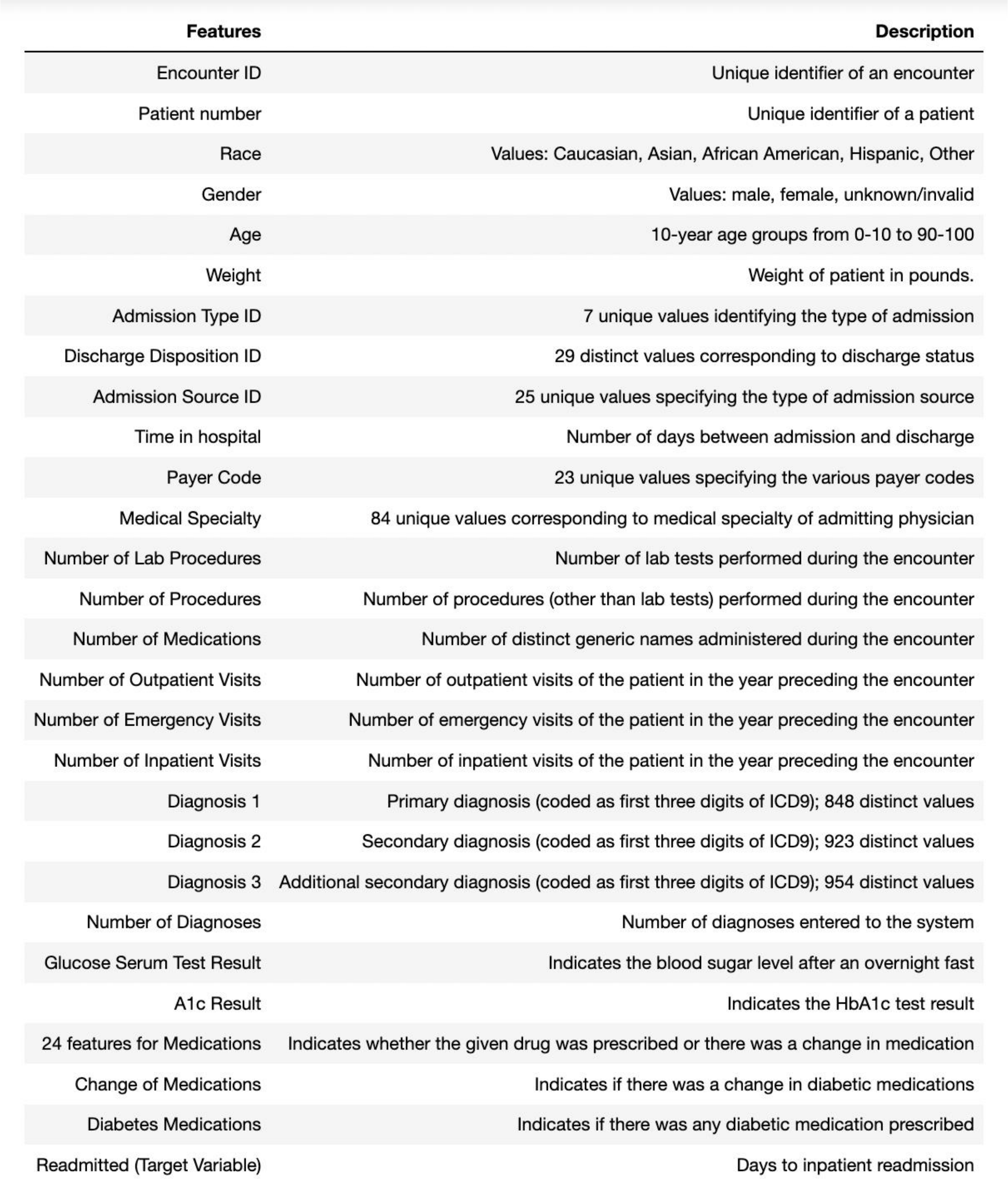
INTRODUCTION:

This dataset has been obtained from the following UCI Machine Learning Repository:
https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008
It contains data of over 130 US hospitals and delivery networks of 10 consecutive years from 1999 to 2008. It stores the in-patient record details such as age, gender, race, admission type, medications that were administered during the encounter, duration for which patient was admitted, number of laboratory test performed, laboratory test results, diagnosis and emergency visits in the year before hospitalization. All of these diverse features contain valuable and heterogeneous data about the patients and also increase its complexity. We use various Machine Learning models to predict whether the diabetic patient is readmitted or not.

DATA DESCRIPTION:

1) Dataset consists of more than 50 features and 100K records of patients.
2) The dataset has 13 numerical columns 37 categorical columns.
3) It contains several features like race, age, patient id, lab test performed, gender, emergency visits, medication details etc. which affect the patient and hospital outcomes.
4) Duration of admitted patient, medication administered, and laboratory test were recorded and performed.
5) Medical specialty of admitting physician, number of lab tests performed, diagnosis given were also used which contributes to change in treatment
6) There are 24 features for indicating medications such as metformin, insulin, Nateglinide, chlorpropamide, etc. which indicate whether the drug was prescribed or there was a change in the dosage.
7) The target column 'readmitted' tells us whether the patient was readmitted or not. This column has 3 different values

i) If the patient was readmitted in less than 30 days, the value is **"<30"**
ii) If the patient was readmitted in more than 30 days, the value is **">30"**
iii) If there is no record, the value is **"NO"**



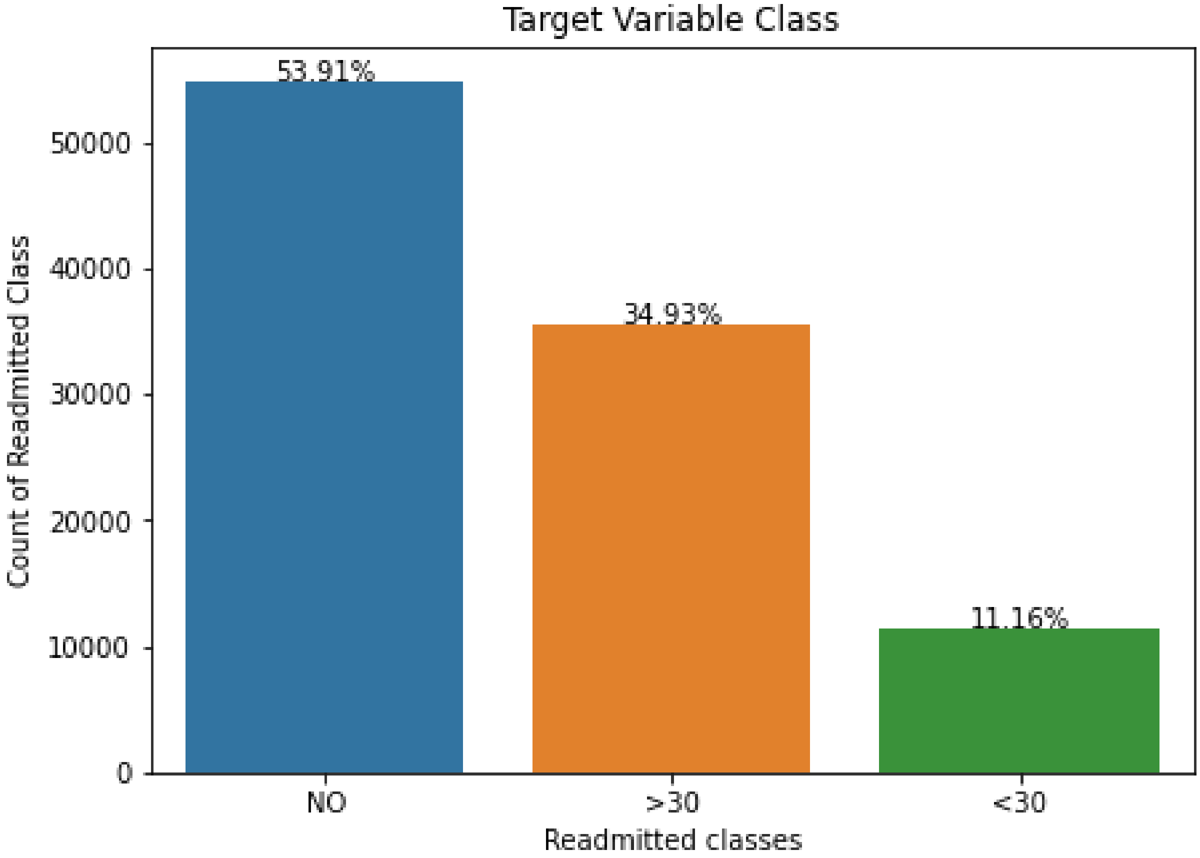
Features	Description
Encounter ID	Unique identifier of an encounter
Patient number	Unique identifier of a patient
Race	Values: Caucasian, Asian, African American, Hispanic, Other
Gender	Values: male, female, unknown/invalid
Age	10-year age groups from 0-10 to 90-100
Weight	Weight of patient in pounds.
Admission Type ID	7 unique values identifying the type of admission
Discharge Disposition ID	29 distinct values corresponding to discharge status
Admission Source ID	25 unique values specifying the type of admission source
Time in hospital	Number of days between admission and discharge
Payer Code	23 unique values specifying the various payer codes
Medical Specialty	84 unique values corresponding to medical specialty of admitting physician
Number of Lab Procedures	Number of lab tests performed during the encounter
Number of Procedures	Number of procedures (other than lab tests) performed during the encounter
Number of Medications	Number of distinct generic names administered during the encounter
Number of Outpatient Visits	Number of outpatient visits of the patient in the year preceding the encounter
Number of Emergency Visits	Number of emergency visits of the patient in the year preceding the encounter
Number of Inpatient Visits	Number of inpatient visits of the patient in the year preceding the encounter
Diagnosis 1	Primary diagnosis (coded as first three digits of ICD9); 848 distinct values
Diagnosis 2	Secondary diagnosis (coded as first three digits of ICD9); 923 distinct values
Diagnosis 3	Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values
Number of Diagnoses	Number of diagnoses entered to the system
Glucose Serum Test Result	Indicates the blood sugar level after an overnight fast
A1c Result	Indicates the HbA1c test result
24 features for Medications	Indicates whether the given drug was prescribed or there was a change in medication
Change of Medications	Indicates if there was a change in diabetic medications
Diabetes Medications	Indicates if there was any diabetic medication prescribed
Readmitted (Target Variable)	Days to inpatient readmission

EXPLORATORY DATA ANALYSIS:

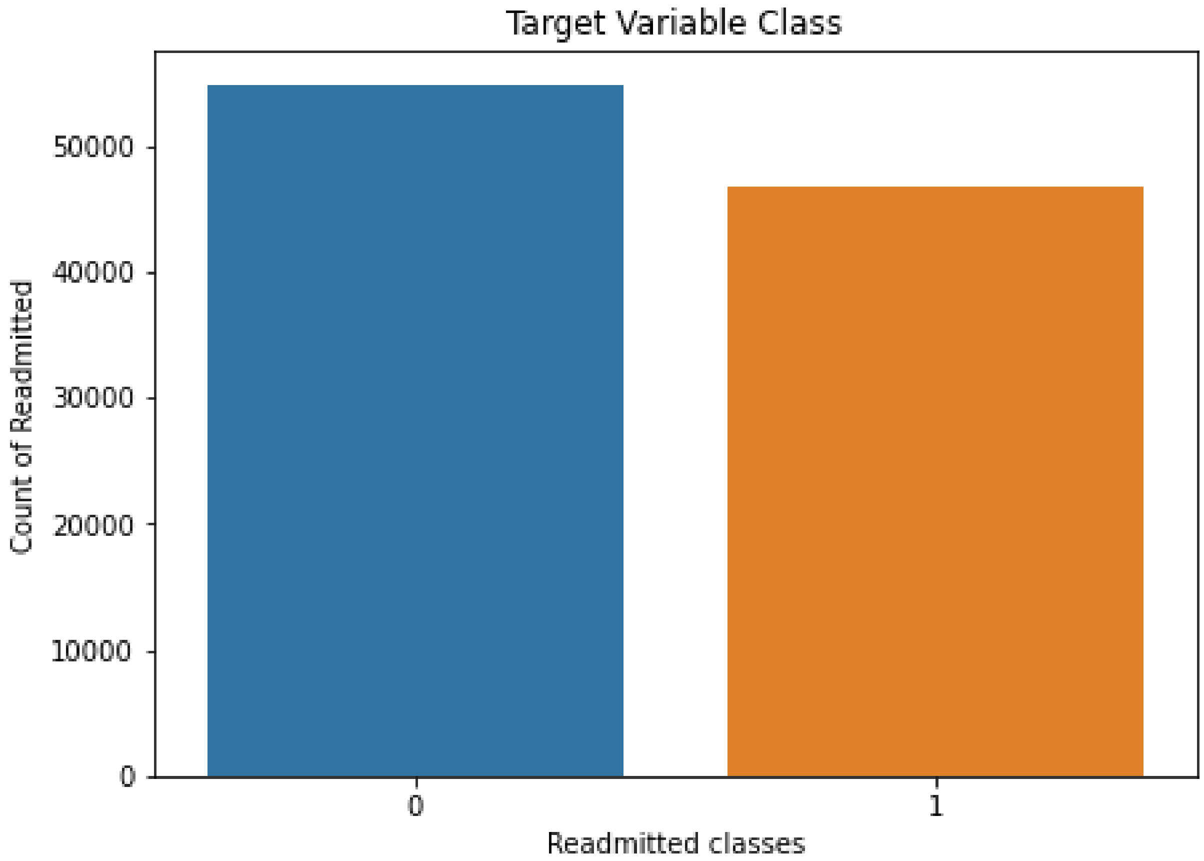
It is an essential step before we start building models. This technique helps in identifying trends and patterns in the data. It includes checking for missing values, identifying distributions of the different variables, checking for outliers, understanding the correlation between variables etc. This analysis further aids us in the process of data cleaning and processing.

Target Variable:

The target variable 'readmitted' has 3 different classes, '<30', '>30' and 'No'.



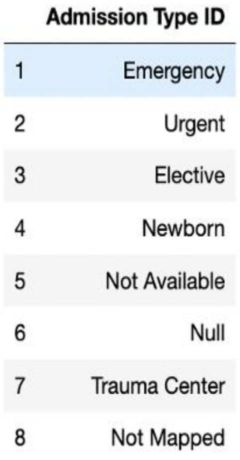
We observe that there is a lot of imbalances in the target variable classes. We know that the class >30 signifies that the patient was readmitted in more than 30 days and <30 signifies that the patient was readmitted in less than 30 days. Therefore, we combine these 2 classes and call it class 1 (patient was readmitted). On the other hand, class 'No' means that the patient was not readmitted at all, and we call it class 0 (patient was not readmitted).



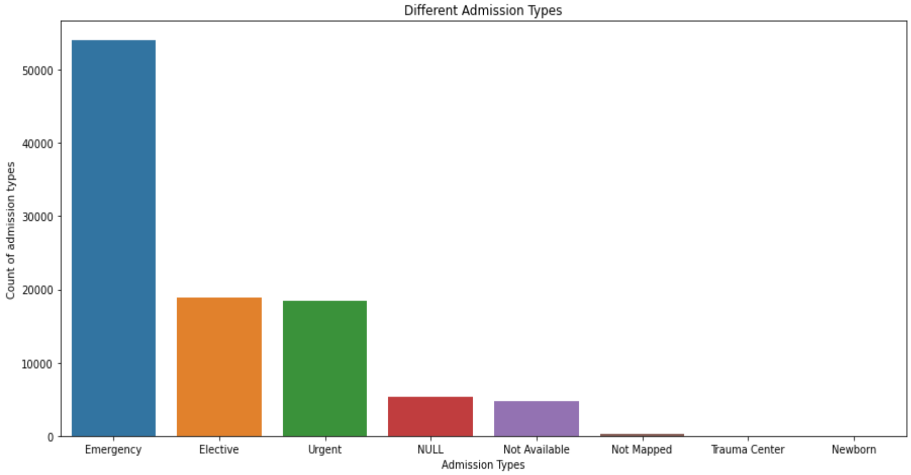
<u>Categorical Columns:</u>

- The columns ‘admission_type_id’, discharge_disposition_id’ and ‘admission_source_id’ are nominal categorical columns which demonstrate the type of admission to hospital, type of discharge provided and the source of admission to hospital respectively. These 3 columns have been encoded with numerical values for the different categories. We try to have a look at these values:

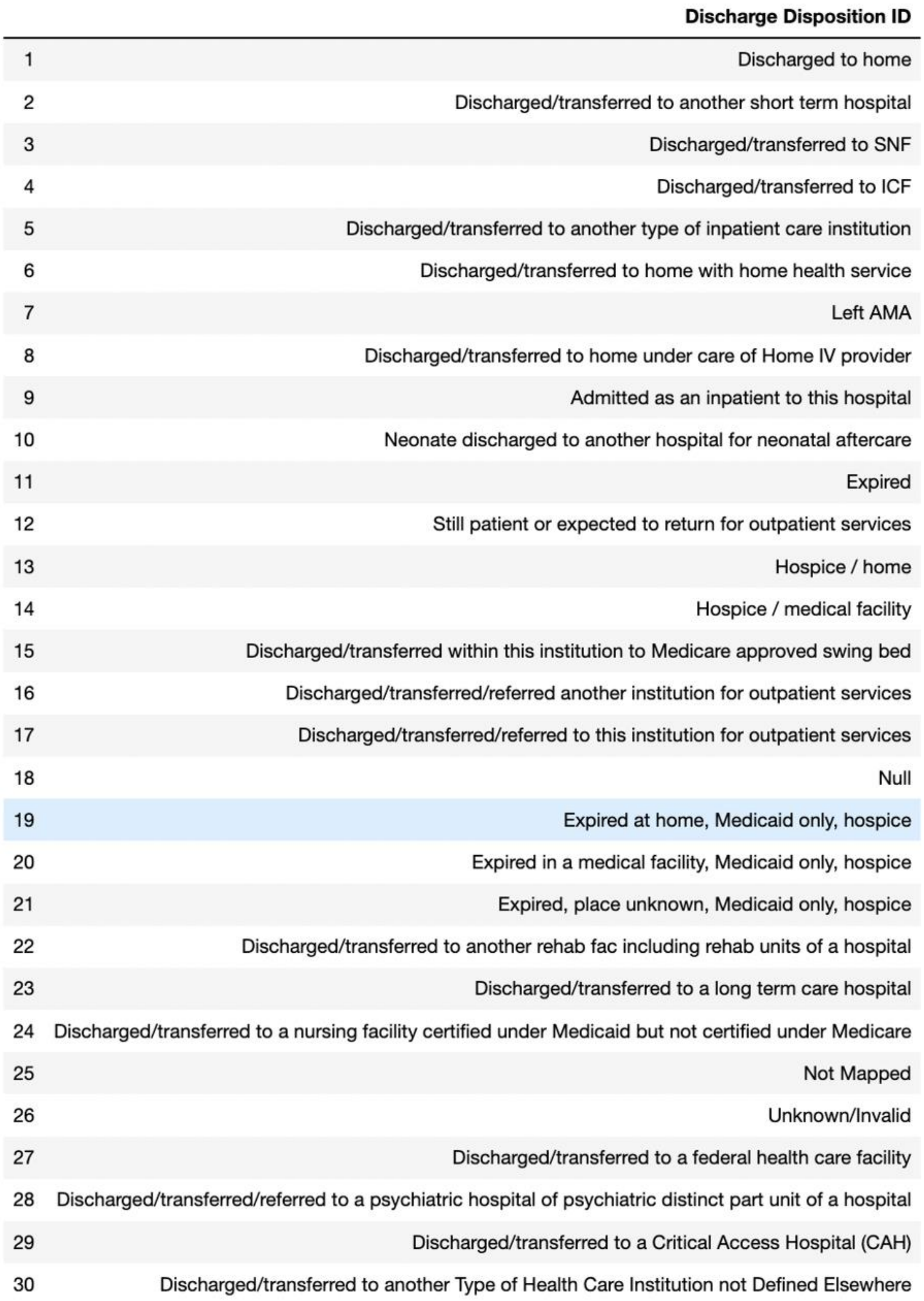
  Admission Type contains 8 distinct values.



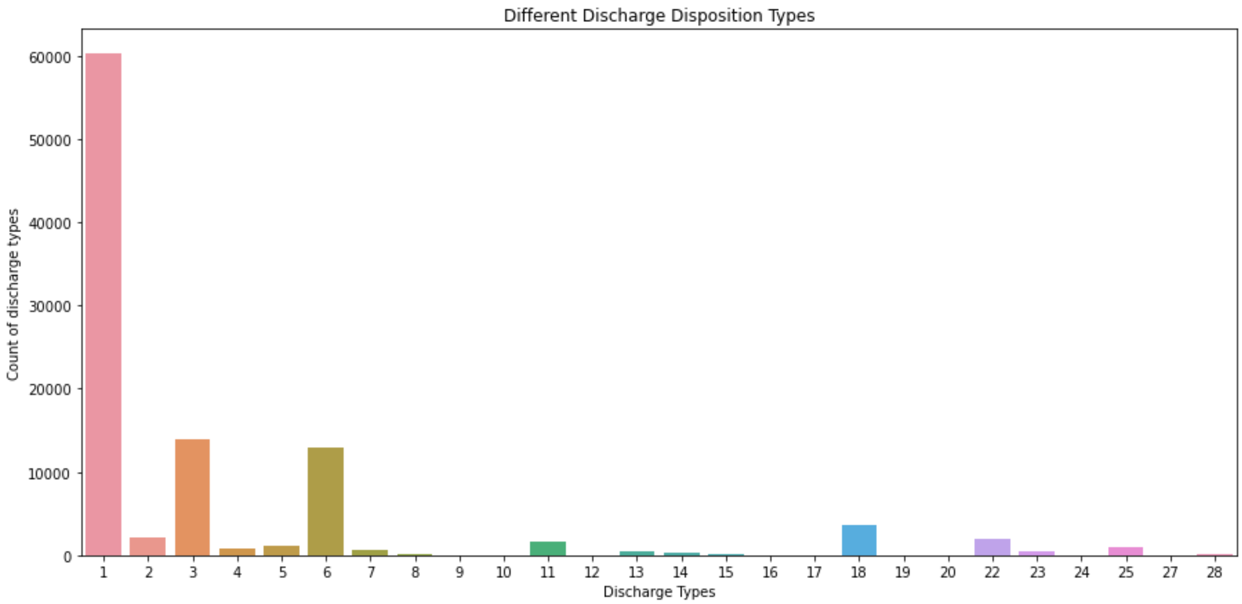
	Admission Type ID
1	Emergency
2	Urgent
3	Elective
4	Newborn
5	Not Available
6	Null
7	Trauma Center
8	Not Mapped



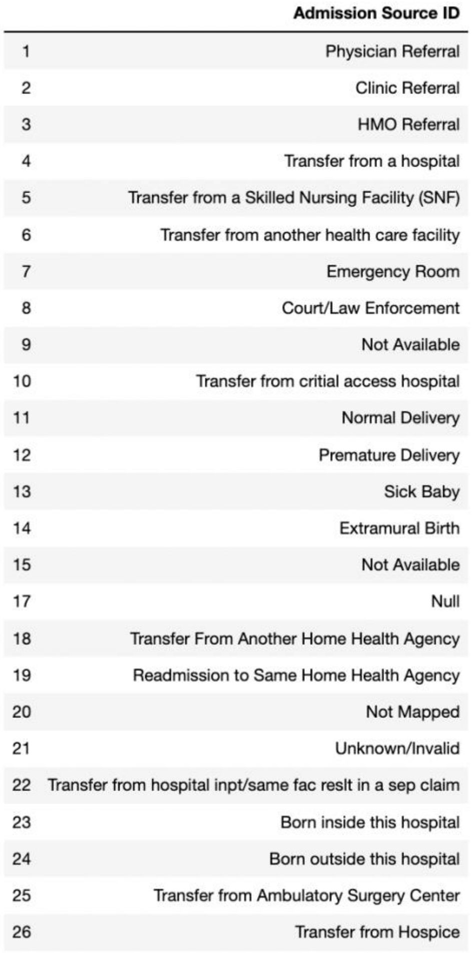
Discharge Disposition type contains 30 unique values that are self-explanatory.



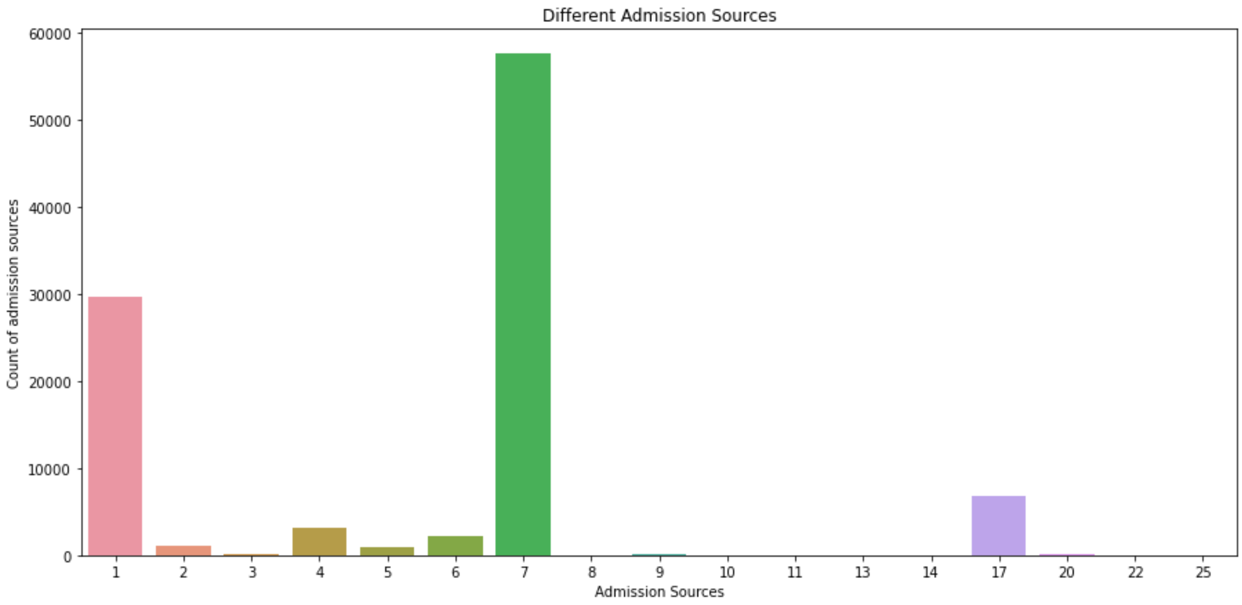
	Discharge Disposition ID
1	Discharged to home
2	Discharged/transferred to another short term hospital
3	Discharged/transferred to SNF
4	Discharged/transferred to ICF
5	Discharged/transferred to another type of inpatient care institution
6	Discharged/transferred to home with home health service
7	Left AMA
8	Discharged/transferred to home under care of Home IV provider
9	Admitted as an inpatient to this hospital
10	Neonate discharged to another hospital for neonatal aftercare
11	Expired
12	Still patient or expected to return for outpatient services
13	Hospice / home
14	Hospice / medical facility
15	Discharged/transferred within this institution to Medicare approved swing bed
16	Discharged/transferred/referred another institution for outpatient services
17	Discharged/transferred/referred to this institution for outpatient services
18	Null
19	Expired at home, Medicaid only, hospice
20	Expired in a medical facility, Medicaid only, hospice
21	Expired, place unknown, Medicaid only, hospice
22	Discharged/transferred to another rehab fac including rehab units of a hospital
23	Discharged/transferred to a long term care hospital
24	Discharged/transferred to a nursing facility certified under Medicaid but not certified under Medicare
25	Not Mapped
26	Unknown/Invalid
27	Discharged/transferred to a federal health care facility
28	Discharged/transferred/referred to a psychiatric hospital of psychiatric distinct part unit of a hospital
29	Discharged/transferred to a Critical Access Hospital (CAH)
30	Discharged/transferred to another Type of Health Care Institution not Defined Elsewhere



Similarly, the column Admission Source has 26 self-explanatory categories.

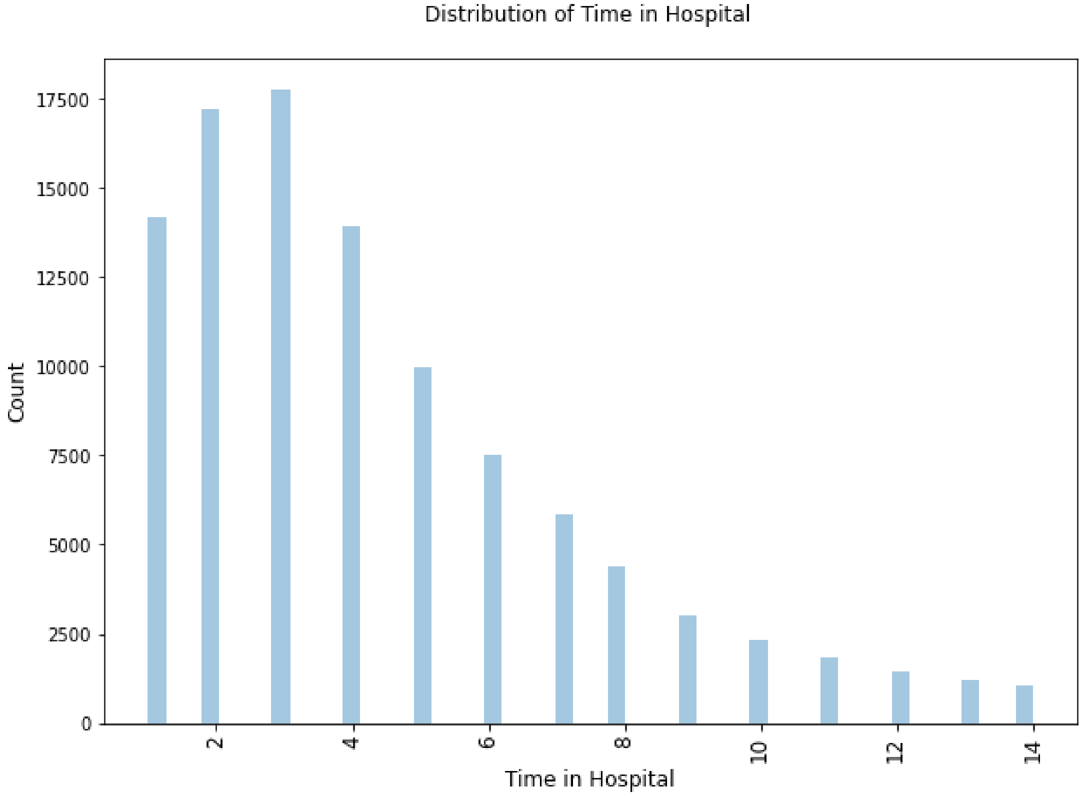


	Admission Source ID
1	Physician Referral
2	Clinic Referral
3	HMO Referral
4	Transfer from a hospital
5	Transfer from a Skilled Nursing Facility (SNF)
6	Transfer from another health care facility
7	Emergency Room
8	Court/Law Enforcement
9	Not Available
10	Transfer from critial access hospital
11	Normal Delivery
12	Premature Delivery
13	Sick Baby
14	Extramural Birth
15	Not Available
17	Null
18	Transfer From Another Home Health Agency
19	Readmission to Same Home Health Agency
20	Not Mapped
21	Unknown/Invalid
22	Transfer from hospital inpt/same fac reslt in a sep claim
23	Born inside this hospital
24	Born outside this hospital
25	Transfer from Ambulatory Surgery Center
26	Transfer from Hospice

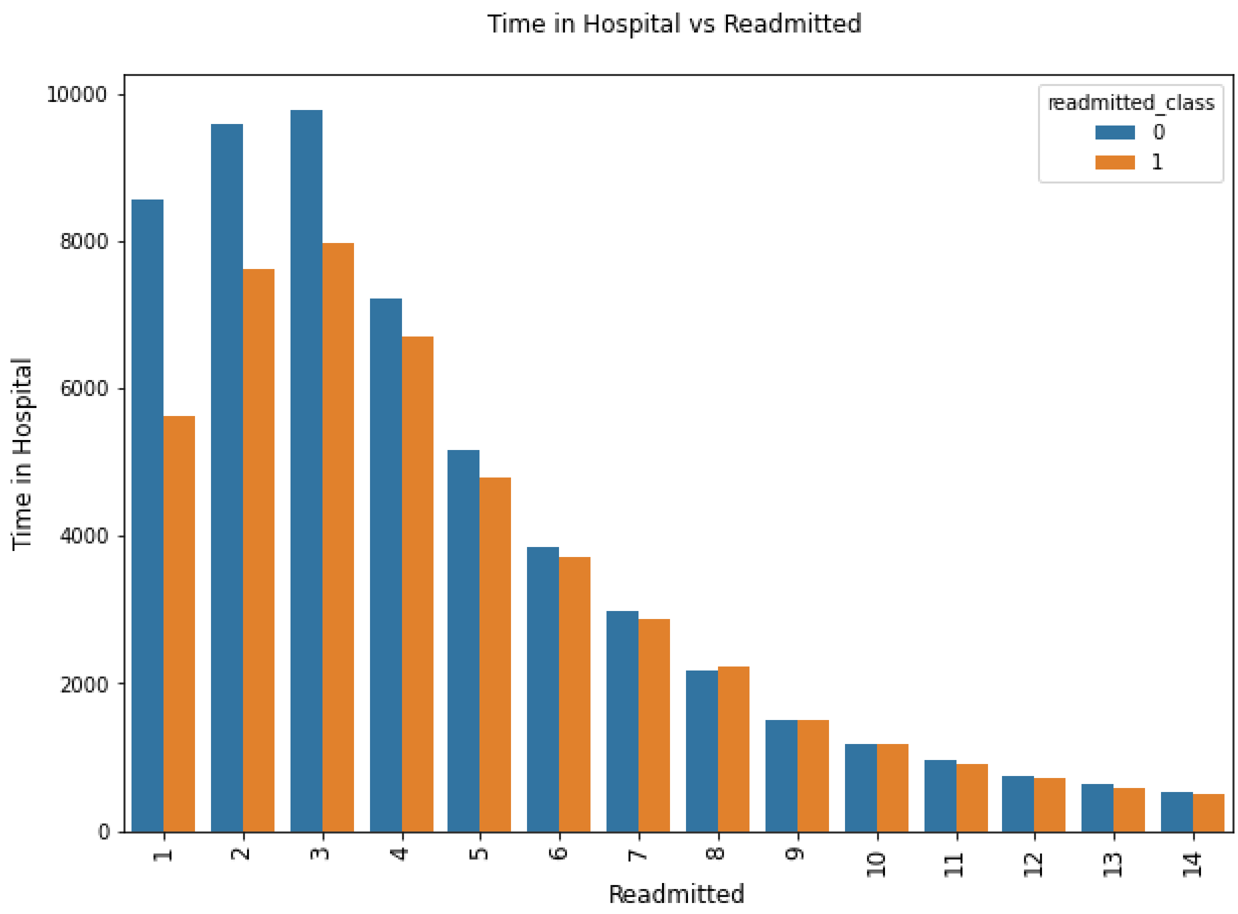


Numerical Columns:

- The column ‘time_in_hosptital’ tell us how many days the patient spent at the hospital. We observe the distribution of this variable, and we conclude that the average number of days between hospital admission and discharge is around 4 days.

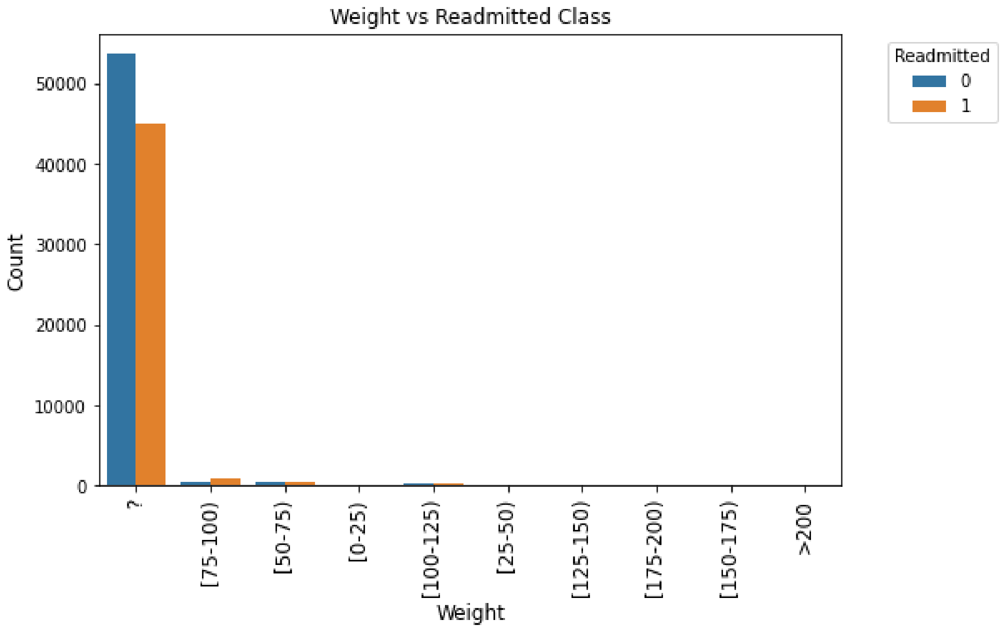


Further, we also visualize the time in hospital against the target variable.



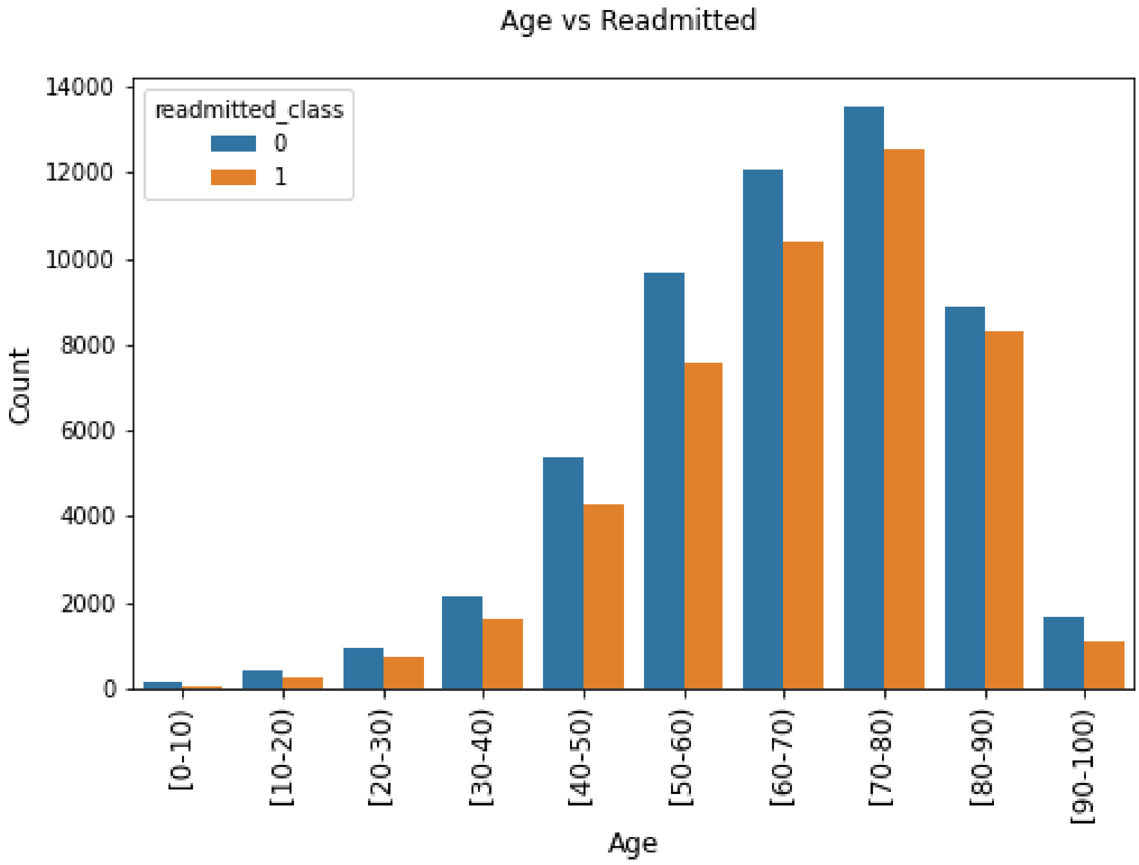
- The columns ‘age’ and ‘weight’ contain binned values.

We have weight of the patient ranging from 0 to above 200 pounds with a bin interval of 25 pounds.



We can see that this column has a lot of unknown '?' values which means that these values are missing.

Age of the patient ranges from 0-100 years and have a bin width of 10 years.



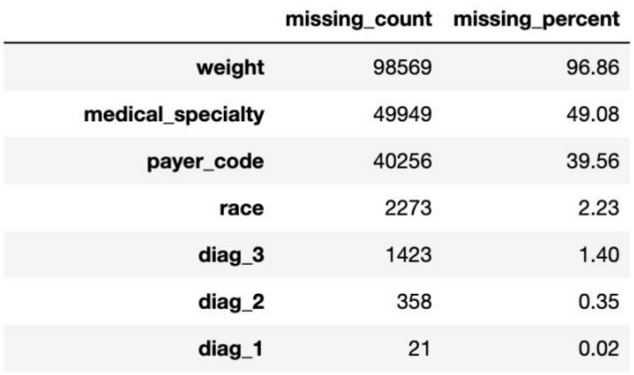
Most of the patients that have diabetes lie in the 50 to 90-year span.

DATA CLEANING AND PROCESSING:

Missing Values:

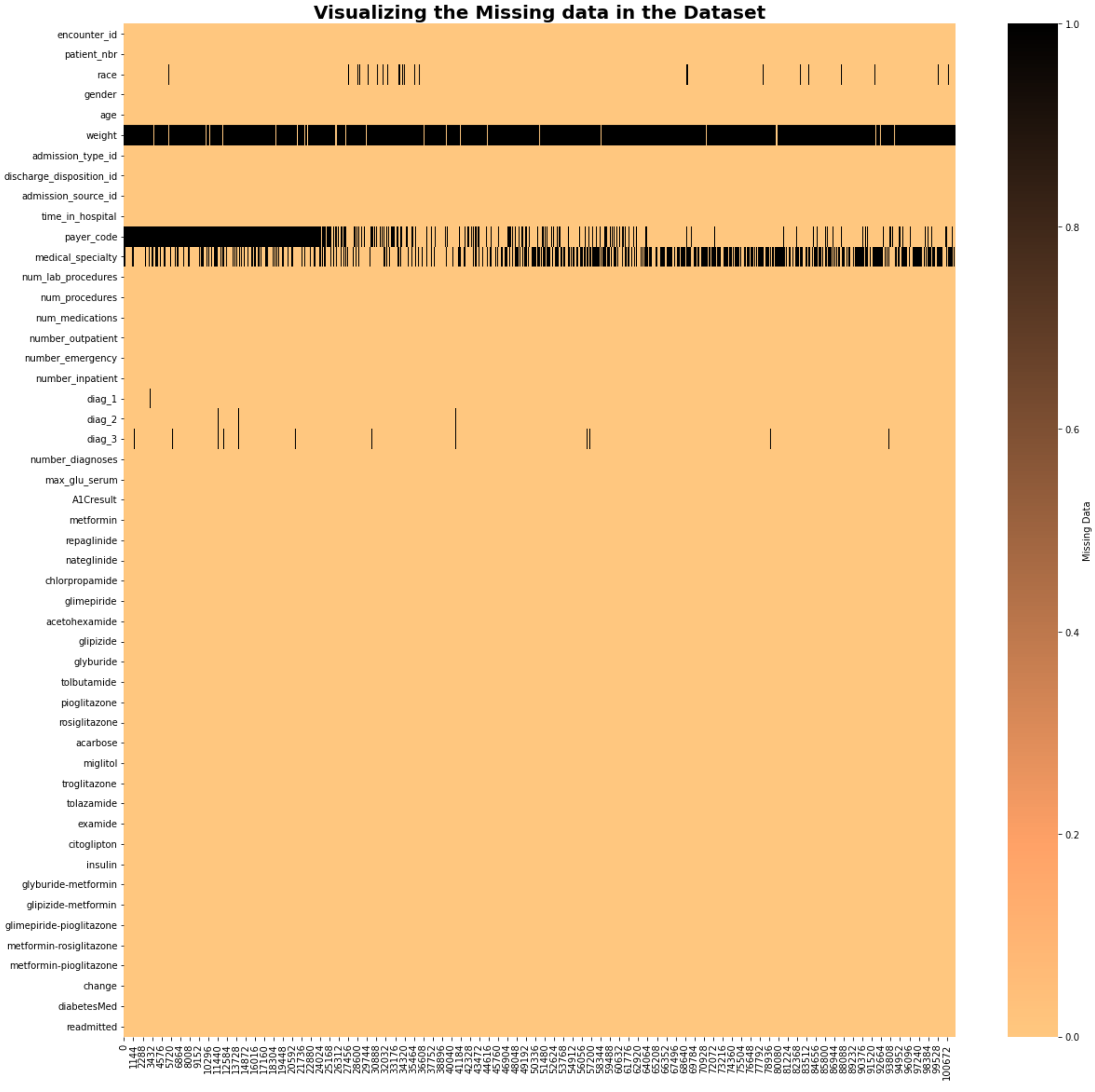
Since this is historical medical data, we expect to find missing values in the dataset. Having missing data can lead to discrepancies during analysis, thus we either try to impute them or drop the columns that have more than 90% missing data (because even if we impute them, the data will not be accurate and representative of the distribution).

In our dataset, we observed a lot of unknown values '?'. These values account for the missing data in our dataset. The following table shows the columns having missing values and their corresponding percent of missing values.



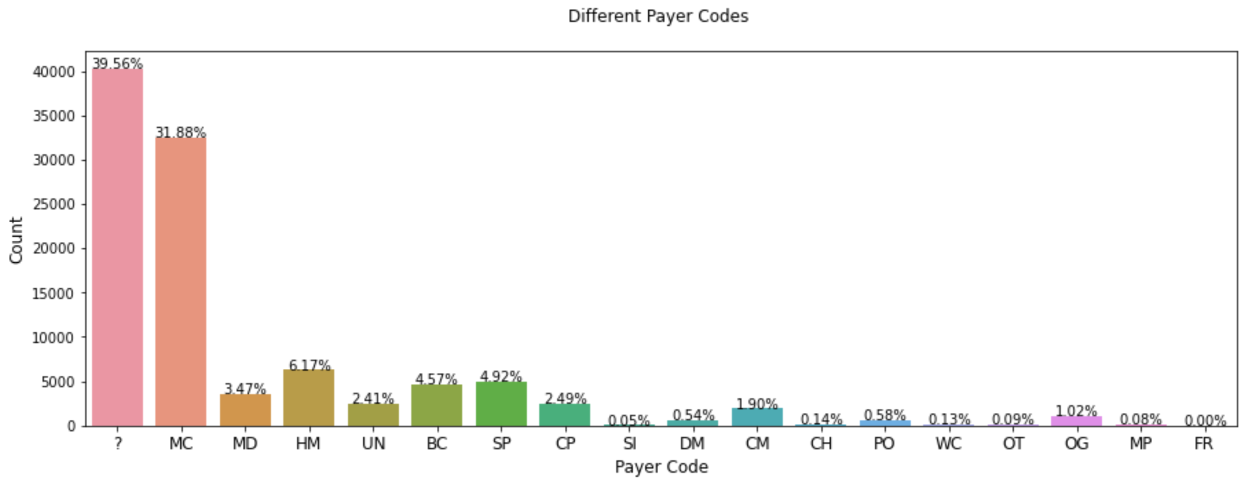
	missing_count	missing_percent
weight	98569	96.86
medical_specialty	49949	49.08
payer_code	40256	39.56
race	2273	2.23
diag_3	1423	1.40
diag_2	358	0.35
diag_1	21	0.02

The following heatmap is plotted for easy visualization of this missing data.



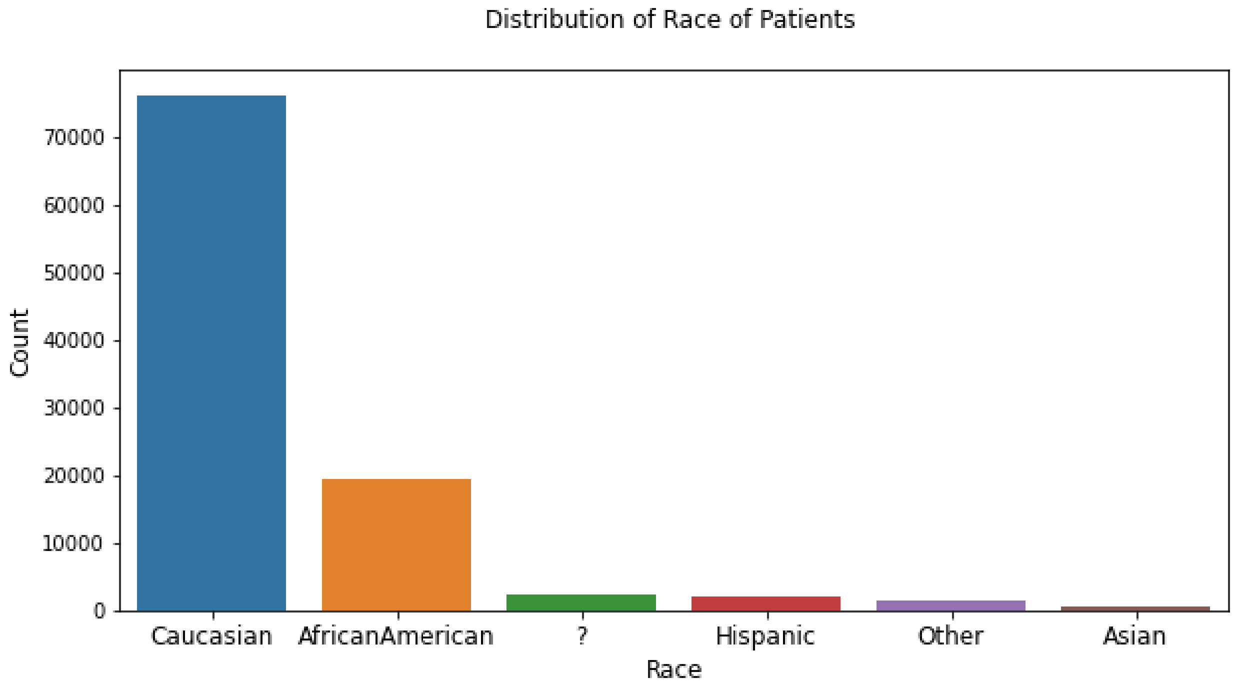
We observe that the weight column has more than 95% missing data and medical specialty has almost 50% missing data. Hence, we drop these 2 columns.

Next, we take a look at the payer_code column. Payer code is an identifier corresponding to 23 distinct values, for example, Blue Cross/Blue Shield, Medicare, Self-pay etc.

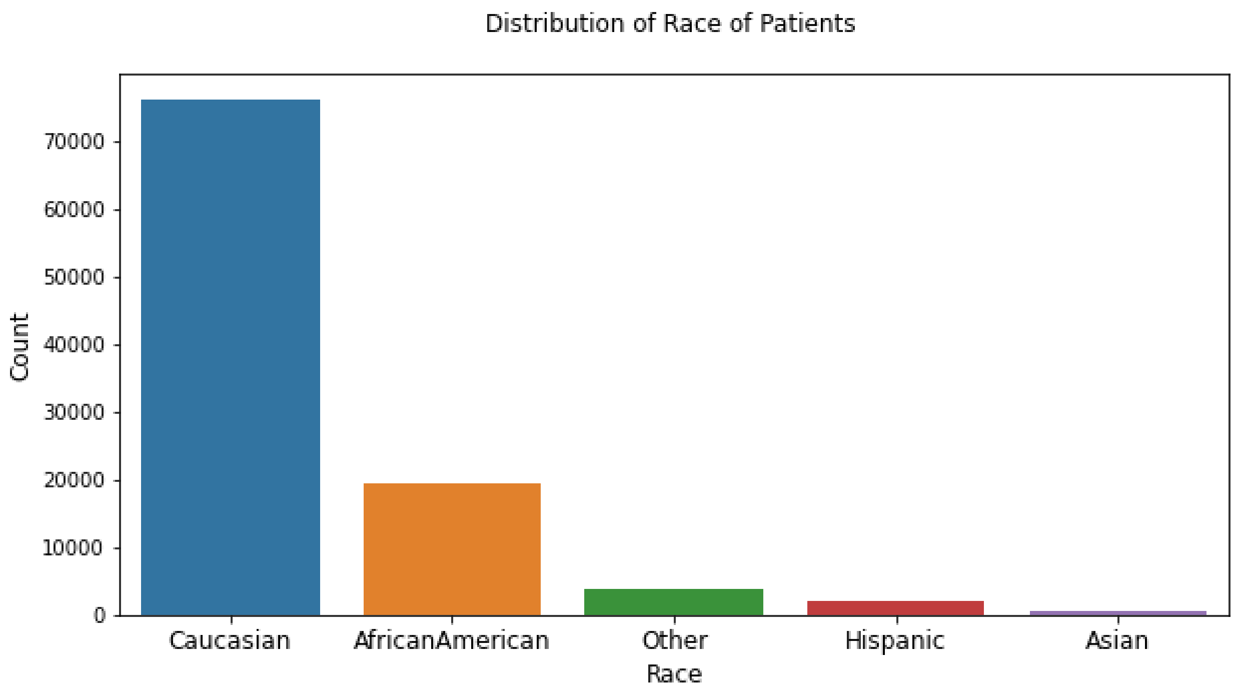


We observe that each of the categories account for less than 35% of total and almost 40% of the data is missing. Hence, we drop this column as well.

We now analyze the next column containing missing values which is ‘race’.

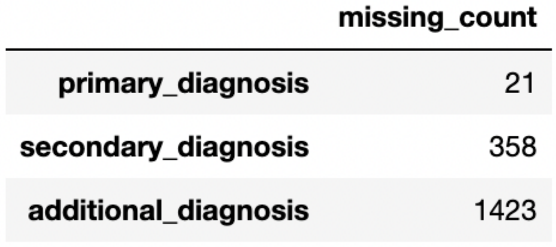


A small portion of this column has missing data. Therefore, we impute the column and set the unknown values ‘?’ as ‘Other’ category.



Finally, we analyze the diagnosis columns ‘diag_1’, ‘diag_2’ and ‘diag_3’. These columns correspond to the primary diagnosis, secondary diagnosis and additional secondary diagnosis. These columns have been coded as first three digits of ICD9. The primary diagnosis contains 848 distinct values, the secondary diagnosis has 923 distinct values whereas additional secondary diagnosis has 954 distinct values. We rename the columns as ‘primary_diagnosis’, ‘secondary_diagnosis’ and ‘additional_diagnosis’ for better interpretability.

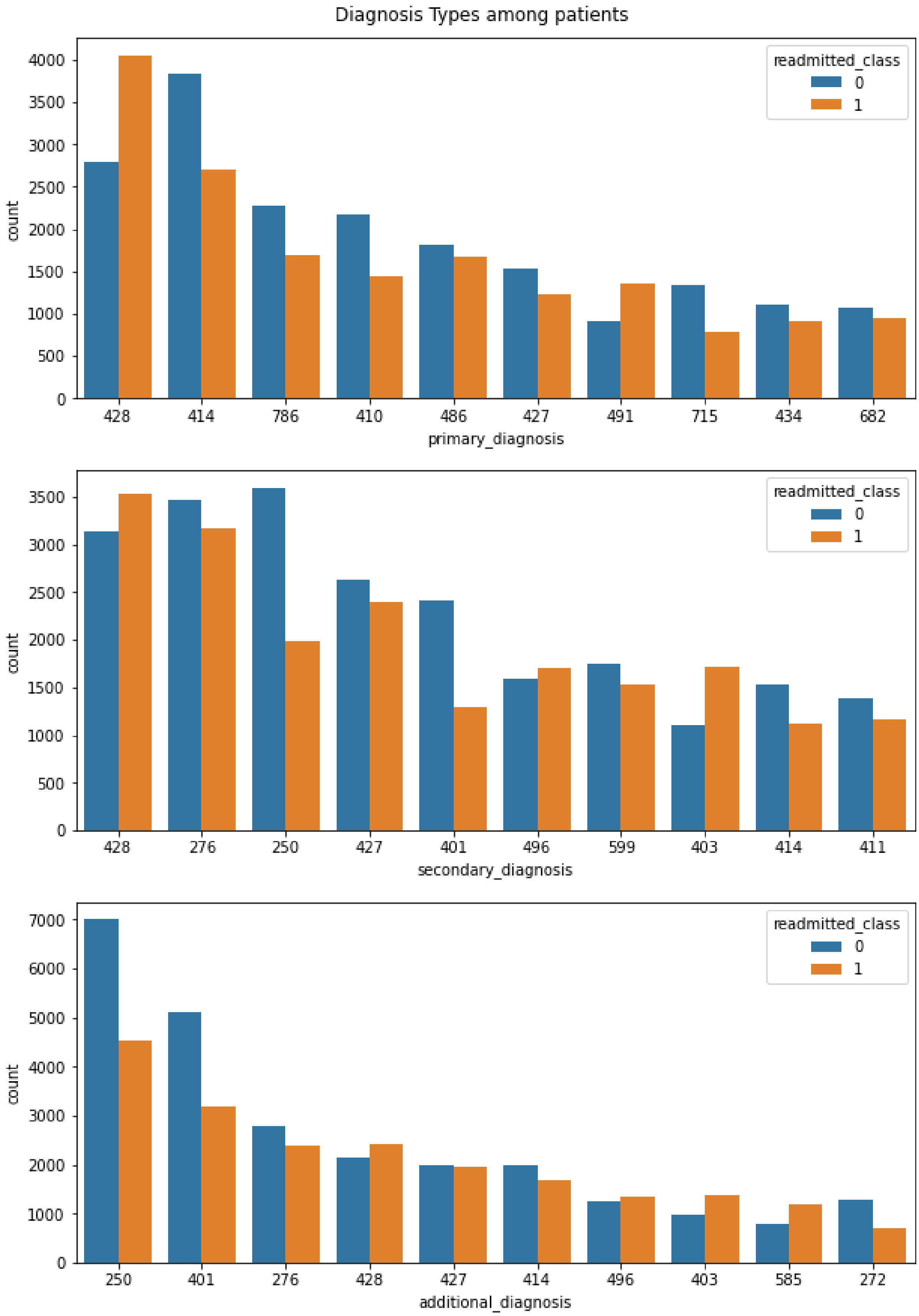
The missing count in these 3 columns is shown in the table below:



	missing_count
primary_diagnosis	21
secondary_diagnosis	358
additional_diagnosis	1423

The number of missing values in these 3 columns is considerably very small as compared to the number of rows in our dataset. Hence, we drop the rows containing missing values in these columns.

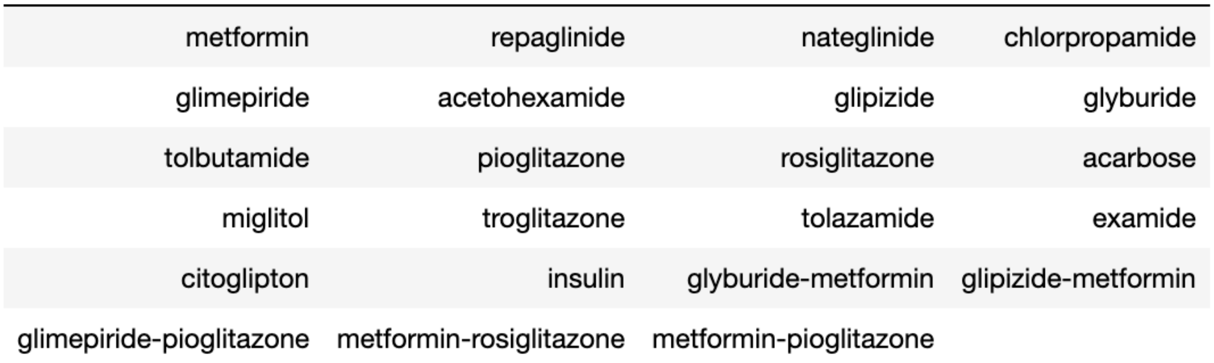
We visualize the different unique values present in these columns using the following bar plots given below:


Diagnosis Types among patients
count
0
500
1000
1500
2000
2500
3000
3500
4000
readmitted_class
0
1
428
414
786
410
486
427
491
715
434
682
primary_diagnosis
count
0
500
1000
1500
2000
2500
3000
3500
readmitted_class
0
1
428
276
250
427
401
496
599
403
414
411
secondary_diagnosis
count
0
1000
2000
3000
4000
5000
6000
7000
readmitted_class
0
1
250
401
276
428
427
414
496
403
585
272
additional_diagnosis

Most patients admitted are of 428 (Congestive heart failure) and 414 (Ischemic heart disease) categories of diagnosis.

Data Processing:

We have 24 features that showcase the medications given to the patient while being admitted in the hospital. It includes the details of the following medications.

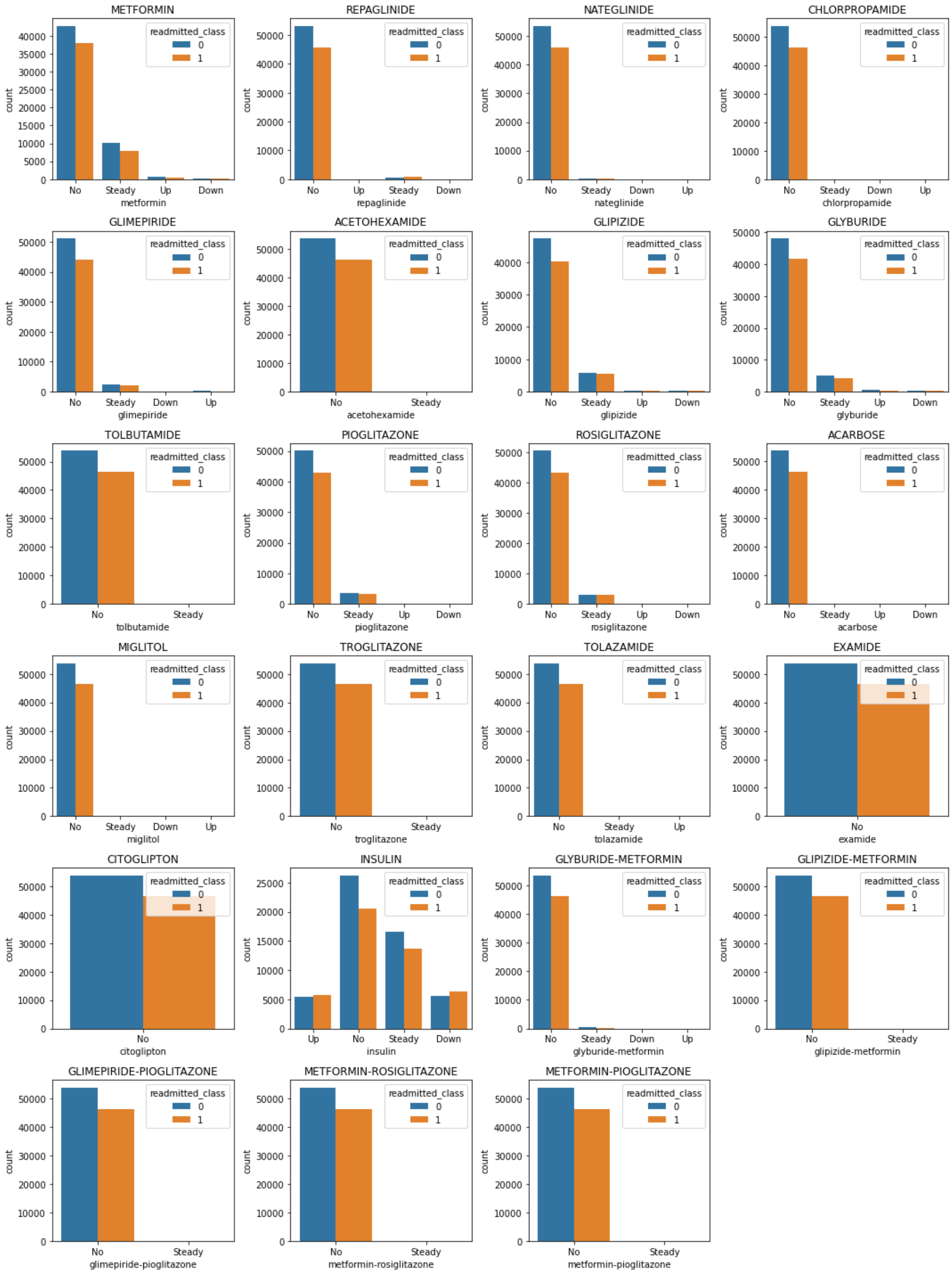


metformin	repaglinide	nateglinide	chlorpropamide
glimepiride	acetohexamide	glipizide	glyburide
tolbutamide	pioglitazone	rosiglitazone	acarbose
miglitol	troglitazone	tolazamide	examide
citoglipton	insulin	glyburide-metformin	glipizide-metformin
glimepiride-pioglitazone	metformin-rosiglitazone	metformin-pioglitazone	

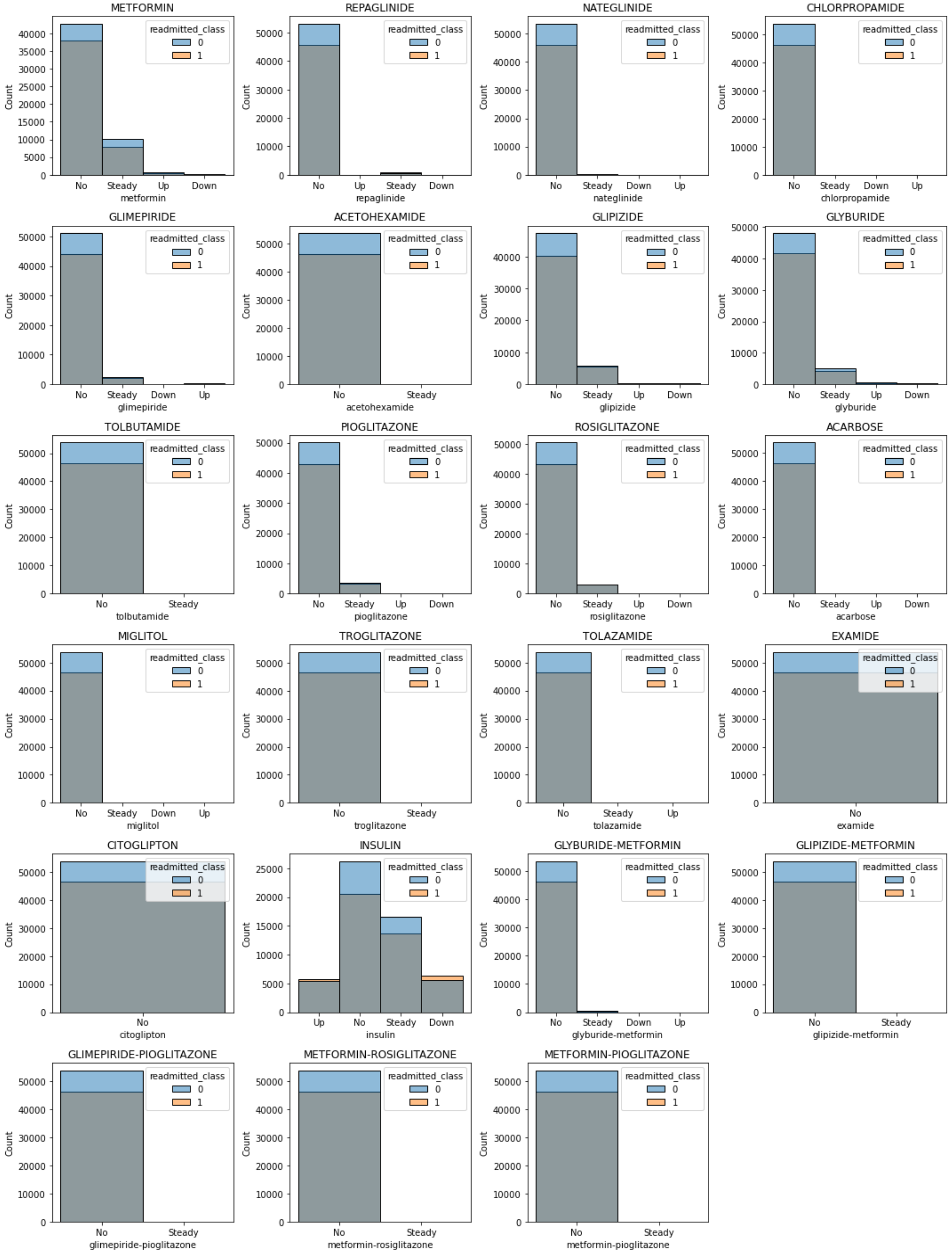
These features indicate whether the drug was prescribed or there was a change in the dosage.

The values that they hold are "up" if the dosage was increased during the encounter, "down" if the dosage was decreased, "steady" if the dosage did not change, and "no" if the drug was not prescribed.

We plot a bar graph for all the 24 features and observe the different values each feature takes and their respective frequencies.

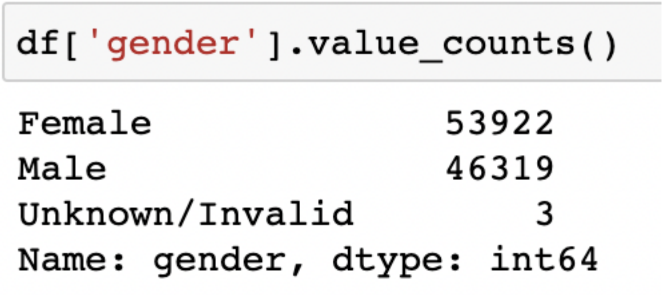

METFORMIN
count
40000
35000
30000
25000
20000
15000
10000
5000
0
No
Steady
Up
Down
metformin
readmitted_class
0
1
REPAGLINIDE
No
Up
Steady
Down
repaglinide
NATEGLINIDE
No
Steady
Down
Up
nateglinide
CHLORPROPAMIDE
No
Steady
Down
Up
chlorpropamide
GLIMEPIRIDE
No
Steady
Down
Up
glimepiride
ACETOHEXAMIDE
No
Steady
acetohexamide
GLIPIZIDE
No
Steady
Up
Down
glipizide
GLYBURIDE
No
Steady
Up
Down
glyburide
TOLBUTAMIDE
No
Steady
tolbutamide
PIOGLITAZONE
No
Steady
Up
Down
pioglitazone
ROSIGLITAZONE
No
Steady
Up
Down
rosiglitazone
ACARBOSE
No
Steady
Up
Down
acarbose
MIGLITOL
No
Steady
Down
Up
miglitol
TROGLITAZONE
No
Steady
troglitazone
TOLAZAMIDE
No
Steady
Up
tolazamide
EXAMIDE
No
examide
CITOGLIPTON
No
citoglipton
INSULIN
25000
20000
15000
10000
5000
Up
No
Steady
Down
insulin
GLYBURIDE-METFORMIN
No
Steady
Down
Up
glyburide-metformin
GLIPIZIDE-METFORMIN
No
Steady
glipizide-metformin
GLIMEPIRIDE-PIOGLITAZONE
No
Steady
glimepiride-pioglitazone
METFORMIN-ROSIGLITAZONE
No
Steady
metformin-rosiglitazone
METFORMIN-PIOGLITAZONE
No
Steady
metformin-pioglitazone
50000
40000
30000
20000
10000

Similarly, we plot a histogram of these features to show the frequency distribution with respect to the target variable.



Out of these features, we can observe that ‘acetohexamide’, ‘tolbutamide’, ‘troglitazone’, ‘tolazamide’, ‘examide’, ‘citoglipton’, ‘glipizide-metformin’, ‘glimepiride-pioglitazone’, ‘metformin-rosiglitazone’ and ‘metformin-pioglitazone’ columns mostly have patients who were never readmitted and very less values for the success class. Hence, we drop these columns.

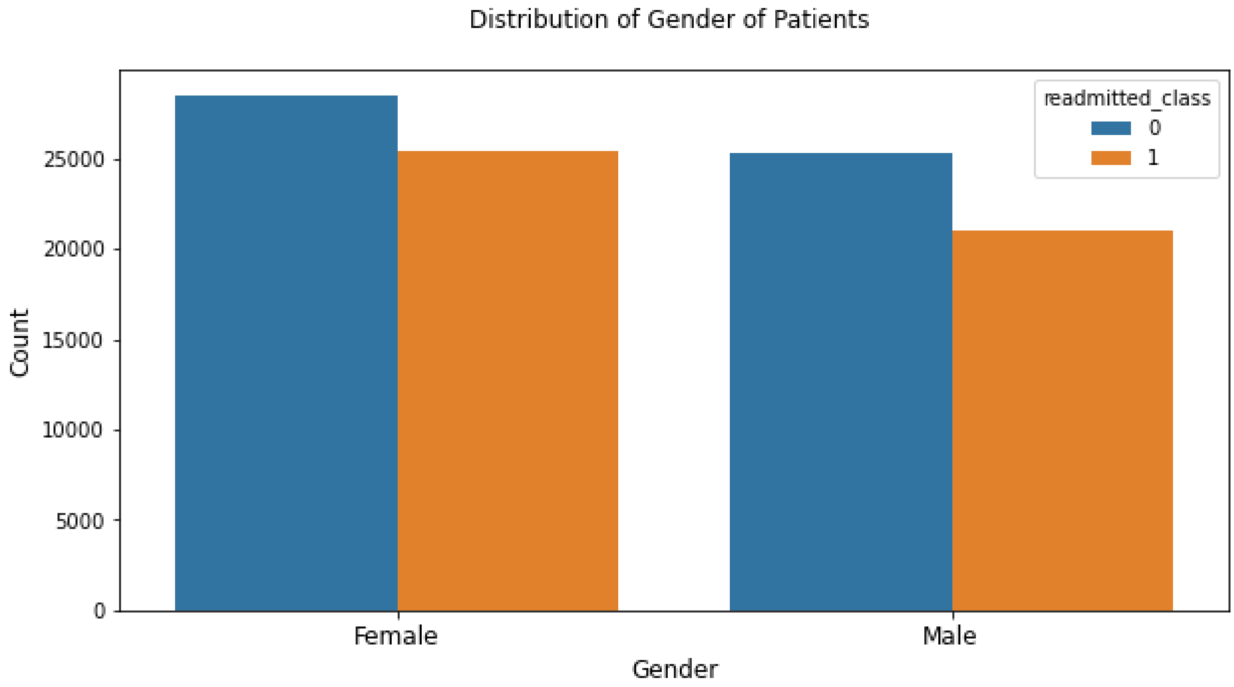
Next, we analyze the column ‘gender’ and we identify 3 different categories for the column.



```
df['gender'].value_counts()
```

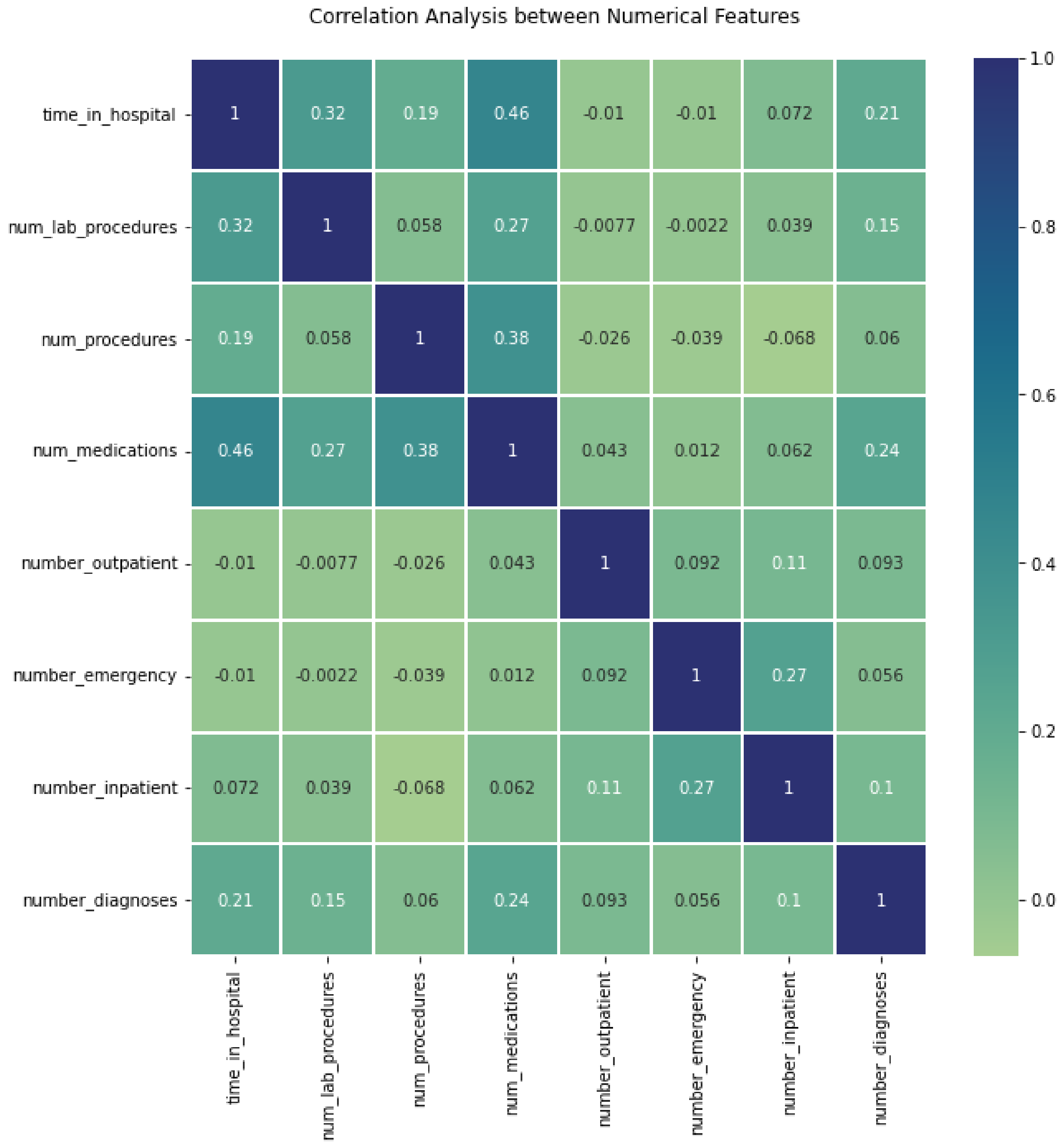
```
Female             53922
Male               46319
Unknown/Invalid        3
Name: gender, dtype: int64
```

The number of records with the ‘Unknown/Invalid’ category are extremely small as compared to the 2 other categories, Hence, we drop the 3 rows containing this value.



Correlation Analysis:

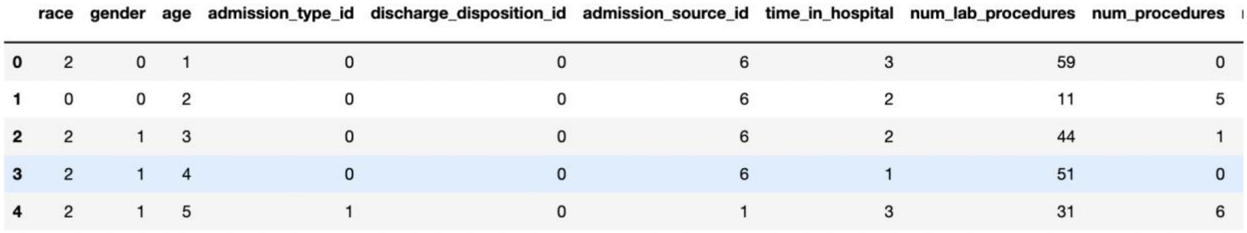
Correlation analysis is performed as part of dimension reduction, where we calculate the correlation between every 2 numerical columns using Pearson’s coefficient. Pearson’s correlation coefficient is a bivariate correlation that measures the linear correlation between two sets of data, whose value ranges from -1 to 1. Essentially, it is a normalized measurement of their covariances. If a pair of variables are highly correlated (for example, setting 0.8 as the cutoff for highly correlated variables), then we drop one of the column-pair.



Since, we do not see any high correlation between any 2 features, hence we retain all the columns.

Data Encoding:

Since we are dealing with categorical columns in our dataset, we map these unique values into numerical values as few Machine Learning models can only deal with numeric data. All of the categorical columns present in our dataset have ordinal categories i.e., they do not have any particular order. In this case, we have used Label encoding for the categorical columns.

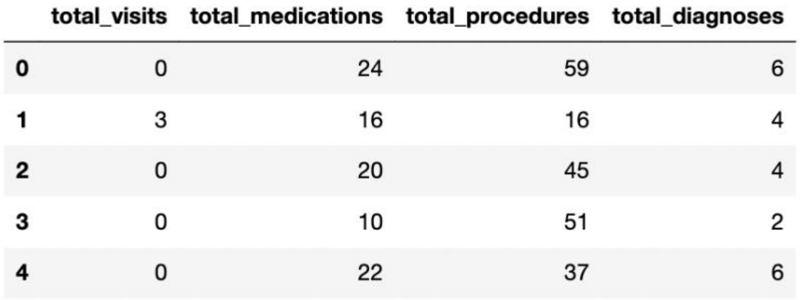


	race	gender	age	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	num_lab_procedures	num_procedures
0	2	0	1	0	0	6	3	59	0
1	0	0	2	0	0	6	2	11	5
2	2	1	3	0	0	6	2	44	1
3	2	1	4	0	0	6	1	51	0
4	2	1	5	1	0	1	3	31	6

Feature Engineering:

This methodology is used to extract or derive new features from existing data using domain knowledge. Here, we have used this technique to come up with new columns that includes:

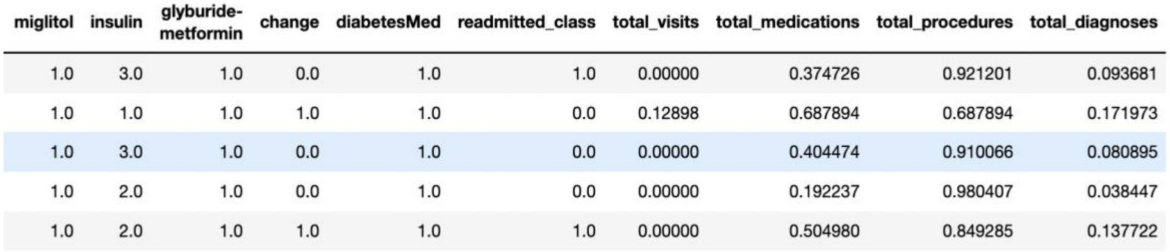
1. Total hospital visits = number of outpatient visits + number of emergency visits + number of inpatient visits
2. Total medications = number of medications given + number of diagnoses
3. Total procedures completed = number of lab procedures + number of other procedures
4. Total diagnoses = number of diagnoses + number of inpatient visits



	total_visits	total_medications	total_procedures	total_diagnoses
0	0	24	59	6
1	3	16	16	4
2	0	20	45	4
3	0	10	51	2
4	0	22	37	6

Data Scaling/ Normalization:

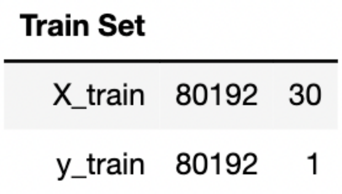
The goal of this step is to bring all the features to a similar scale. This is an important step since it ensures equal consideration of all the features, thus improving the numerical stability of our model. It may also speed up the training process. We use sklearn's normalize function to scale our dataset.



miglitol	insulin	glyburide-metformin	change	diabetesMed	readmitted_class	total_visits	total_medications	total_procedures	total_diagnoses
1.0	3.0	1.0	0.0	1.0	1.0	0.00000	0.374726	0.921201	0.093681
1.0	1.0	1.0	1.0	1.0	0.0	0.12898	0.687894	0.687894	0.171973
1.0	3.0	1.0	0.0	1.0	0.0	0.00000	0.404474	0.910066	0.080895
1.0	2.0	1.0	0.0	1.0	0.0	0.00000	0.192237	0.980407	0.038447
1.0	2.0	1.0	1.0	1.0	1.0	0.00000	0.504980	0.849285	0.137722

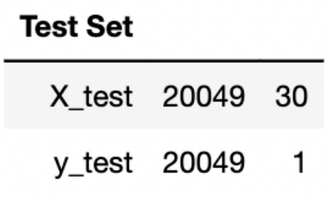
Data Partitioning/ Splitting:

Data partitioning or data splitting is the process of dividing the dataset into 2 or more parts, typically to train the model on one part and validate on the other. This step is performed to avoid overfitting. Data should be split in a way that we have more data for the training purpose. Since we have a medium sized dataset, hence we split it into train and test. In this case, we have performed an 80-20 split for creating train and test sets.



Train Set

X_train	80192	30
y_train	80192	1



Test Set

X_test	20049	30
y_test	20049	1

MODEL BUILDING AND MODEL SELECTION:

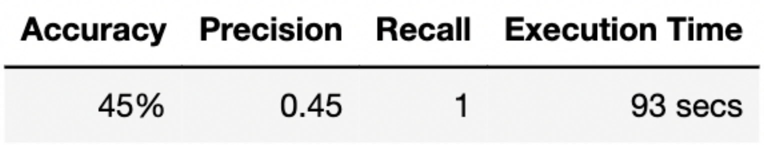
Since we have a classification problem, we have explored different classification models including logistic regression, naive bayes and support vector machines. We choose the model that performs the best on the dataset, and finally assess the performance of the chosen model.

Logistic Regression:

Logistic regression is a classification model which uses a set of features to predict a categorical outcome variable. It applies the logistic sigmoid function to weighted input values to generate a prediction of the data class.

We chose this model as our base model against which all other models will be compared. This model is chosen because it can interpret the model coefficients as indicators of feature importance, it performs well when the dataset is linearly separable.

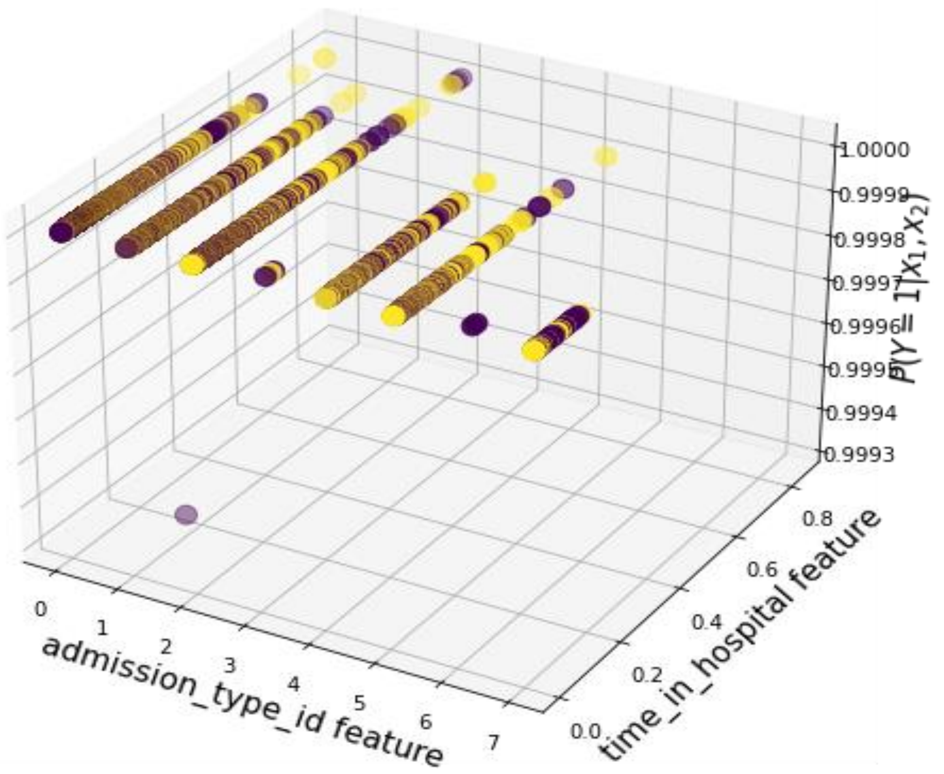
In this case, we have used logistic regression using gradient descent with a maximum number of iterations as 10000. For the model training, we have used the values of Learning Rate as 0.0000001 and Tolerance as 0.0000001.



Accuracy	Precision	Recall	Execution Time
45%	0.45	1	93 secs

This resulted in achieving an accuracy of 45%, precision of 0.45.

We have used the class's plot function to check the model's features (2 at a time, since the dataset is in higher dimension) in 3D space.

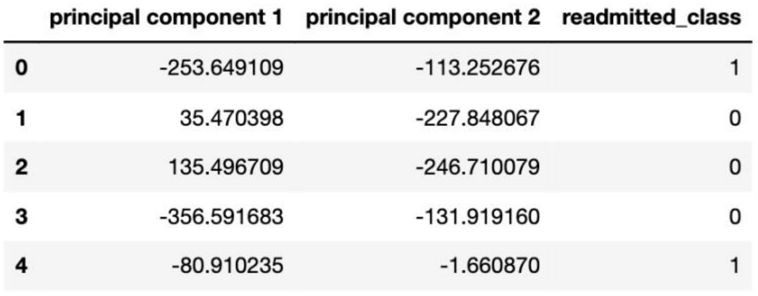


Naïve Bayes with PCA:

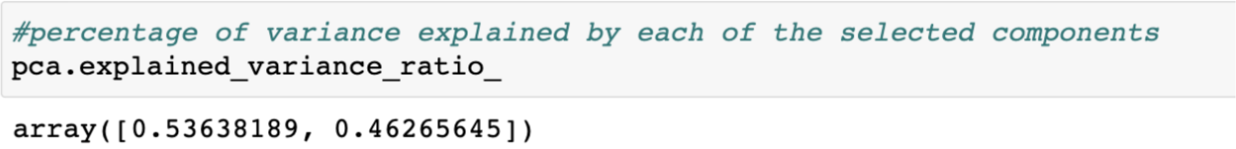
A Naive Bayes classifier is a probabilistic machine learning model that is based on the Bayes theorem. We have used this classification method, since it is fast and efficient while giving accurate results even for large datasets. We have also applied Laplace smoothing that handles the problem of zero probability in the Naive Bayes model.

Principal Component Analysis (PCA) is a technique for reducing the dimensionality of large datasets. This method creates new uncorrelated variables that successively maximize the variance. Since, we have almost 30 features in our dataset, we try using PCA and see if the model performs any better than our base model.

After performing PCA, we consider the first 2 components which covers 99% (53% and 46% of components 1 and 2 respectively) of the variance of the dataset.



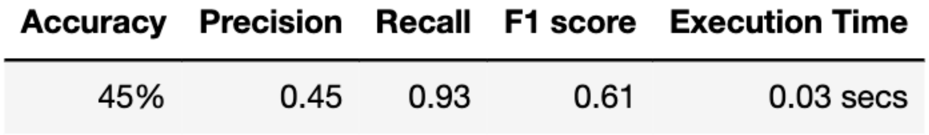
	principal component 1	principal component 2	readmitted_class
0	-253.649109	-113.252676	1
1	35.470398	-227.848067	0
2	135.496709	-246.710079	0
3	-356.591683	-131.919160	0
4	-80.910235	-1.660870	1



```
#percentage of variance explained by each of the selected components
pca.explained_variance_ratio_
```

```
array([0.53638189, 0.46265645])
```

Using the Naïve Bayes classifier, we obtain the following result:



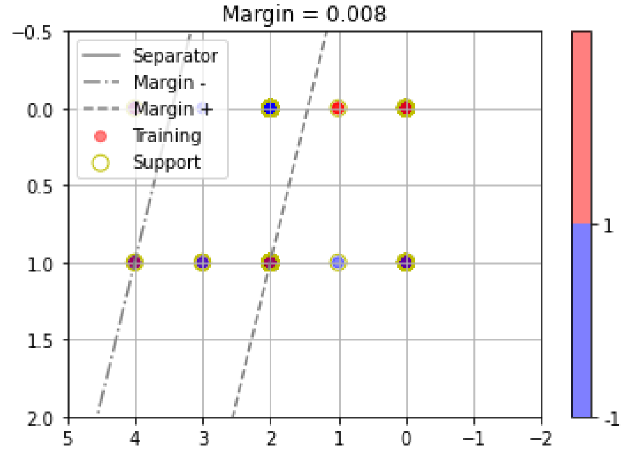
Accuracy	Precision	Recall	F1 score	Execution Time
45%	0.45	0.93	0.61	0.03 secs

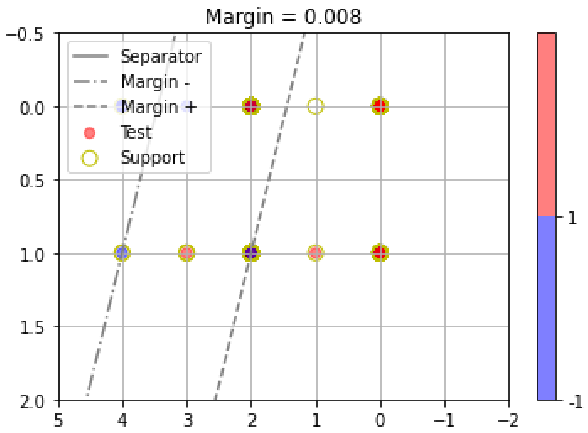
Support Vector Machines:

It is mainly a classification model in which we plot each data item as a point in n-dimensional space and perform classification by finding the hyper-plane that differentiates the two classes well. We have chosen this model for our analysis as it is an effective algorithm in high dimensional spaces.

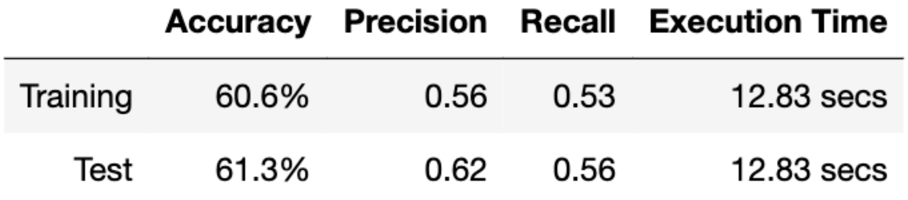
Since the computation time and complexity increases with the number of rows in the dataset, therefore in our case, we have used a sampled dataset of 500 records.

The decision boundary line is plotted by predicting the target variable class on both the training set and test and they are shown below:





With this model, we obtained the following results:



	Accuracy	Precision	Recall	Execution Time
Training	60.6%	0.56	0.53	12.83 secs
Test	61.3%	0.62	0.56	12.83 secs

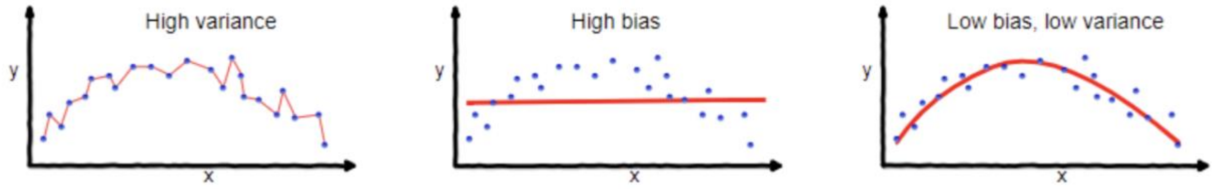
Even though, SVM is performing the best out of all the 3 models that we have explored, however the computation time is higher. Even when we sampled the dataset to 500 records, the time taken by SVM is around 20 seconds. If we take the complete dataset (~100k records), then the computation time would be much higher. This is not ideal.

Finally, we choose Logistic Regression as the best performing model with the following performance measures:

BIAS-VARIANCE TRADE-OFF:

The term 'bias' refers to the difference between the average prediction of our model and the correct value that we are trying to predict. High bias leads to high error on the training and test data.
Moreover, the term 'variance' refers to the variability of the model prediction for a give data that describes the spread of our data. Models with high variance perform very well on the training data but does poorly on the test data.

Overfitting occurs when the model captures the noise in the data along with the underlying pattern. Overfit models have a low bias and a high variance. On the other hand, underfitting occurs when the model is unable to capture the underlying patterns in the data. Underfit models are very simple to capture the complexities in the data and have a high bias. Bias-variance trade-off signifies that there should be good balance in the model performance without overfitting or underfitting the data.

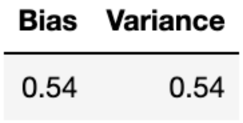


overfitting

underfitting

Good balance

In our chosen model that is, Logistic Regression, we have calculated the bias and variance of the model as follows:



Bias	Variance
0.54	0.54

We observe that there is a proper balance between the 2 values, hence the model we have chosen is neither overfitting nor underfitting.

SUMMARY:

For the classification problem, we have used the Logistic Regression Model as the base model and considering the following factors, we have chosen this as the best performing model on our dataset:

- The algorithm is efficient and takes only about 93 seconds to provide the results for over 100k records.
- It has been able to accurately predict the new data (test data) with a recall score of 0.92.