Comprehensive Analysis of Diabetic Readmissions

Milestone: Data Collection, Data Visualization, Data Exploration and Data Processing

Group 18

Mallika Gaikwad
Divya Babulal Shah

gaikwad.mal@northeastern.edu
shah.divyab@northeastern.edu

Percentage of Effort Contributed by Student 1: 50%

Percentage of Effort Contributed by Student 2: 50%

Signature of Student 1: Mallika Gaikwad

Signature of Student 2: Divya Babulal Shah

Submission Date: 16th February 2024

DATASET LINK:

https://archive.ics.uci.edu/dataset/296/diabetes+130-us+hospitals+for+years+1999-2008

DATASET DESCRIPTION:

Encounter ID: Unique identifier of an encounter

Patient number: Unique identifier of a patient

Race Values: Caucasian, Asian, African American, Hispanic, and other

Gender Values: male, female, and unknown/invalid

Age: Grouped in 10-year intervals: 0, 10), 10, 20), …, 90, 100)

Weight: Weight in pounds

Admission type: Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available

Discharge disposition: Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available

Admission source: Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital

Time in hospital: Integer number of days between admission and discharge

Payer code: Integer identifier corresponding to 23 distinct values, for example, Blue Cross/Blue Shield, Medicare, and self-pay Medical

Medical specialty: Integer identifier of a specialty of the admitting physician, corresponding to 84 distinct values, for example, cardiology, internal medicine, family/general practice, and surgeon

Number of lab procedures: Number of lab tests performed during the encounter

Number of procedures: Numeric 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: The 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 0%

Glucose serum test : result Indicates the range of the result or if the test was not taken. Values: “>200,” “>300,” “normal,” and “none” if not measured

A1c test result : Indicates the range of the result or if the test was not taken. Values: “>8” if the result was greater than 8%, “>7” if the result was greater than 7% but less than 8%, “normal” if the result was less than 7%, and “none” if not measured

Change of medications : Indicates if there was a change in diabetic medications (either dosage or generic name). Values: “change” and “no change”

Diabetes medications : Indicates if there was any diabetic medication prescribed. Values: “yes” and “no” 24 features for medications For the generic names: metformin, repaglinide, nateglinide, 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. Values: “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

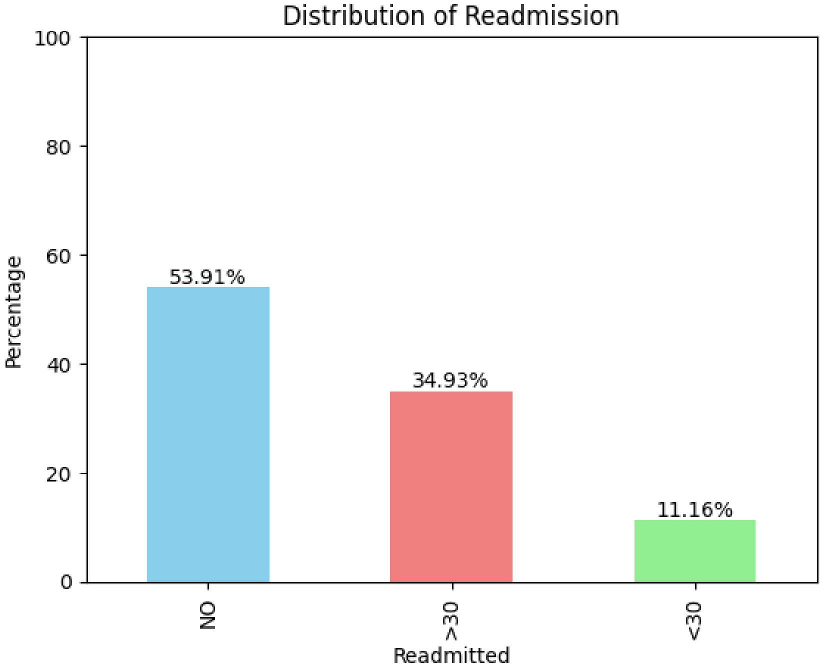
Readmitted: Days to inpatient readmission. Values: “<30” if the patient was readmitted in less than 30 days, “>30” if the patient was readmitted in more than 30 days, and “No” for no record of readmission

Shape of data: (101766, 50)

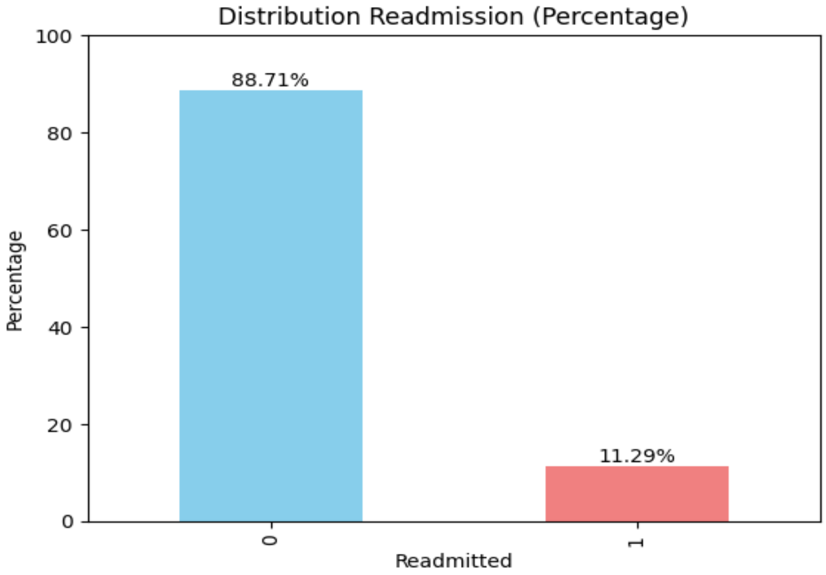
EXPLORATORY DATA ANALYSIS

Total number of unique values in data each readmitted column:

3 Unique values in readmitted column: ['NO' '>30' '<30']



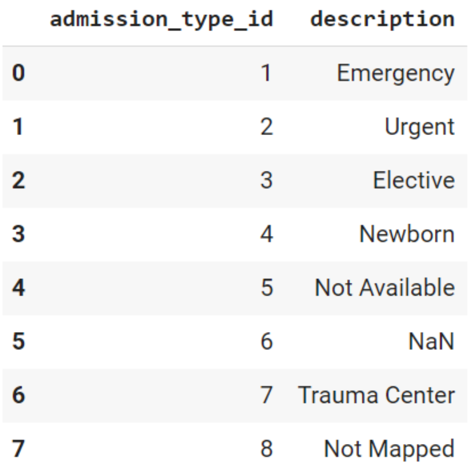
Encoding the outcome variable: The outcome we are looking at is whether the patient gets readmitted to the hospital within 30 days or not. The variable has “< 30”, “> 30” and “NO” Readmission categories. To reduce our problem to a binary classification, we combined the readmission after 30 days and no readmission into a single category:



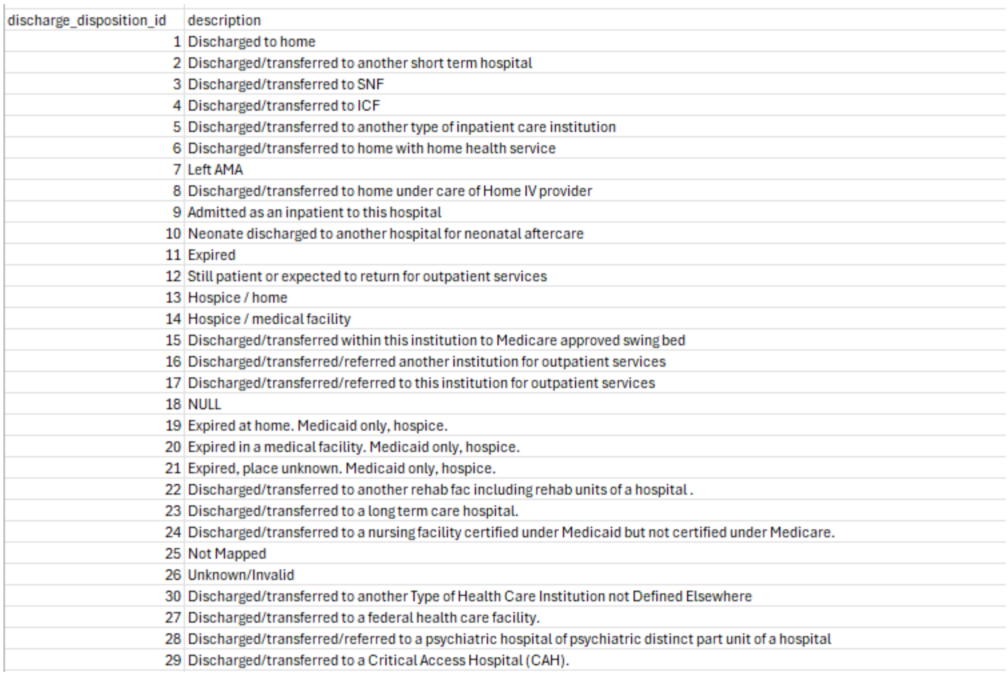
The numerical columns in the dataset are:

['encounter_id', 'patient_nbr', 'admission_type_id', 'discharge_disposition_id', 'admission_source_id', 'time_in_hospital','num_lab_procedures', 'num_procedures', 'num_medications','number_outpatient', 'number_emergency', 'number_inpatient', 'number_diagnoses', 'readmitted_class']

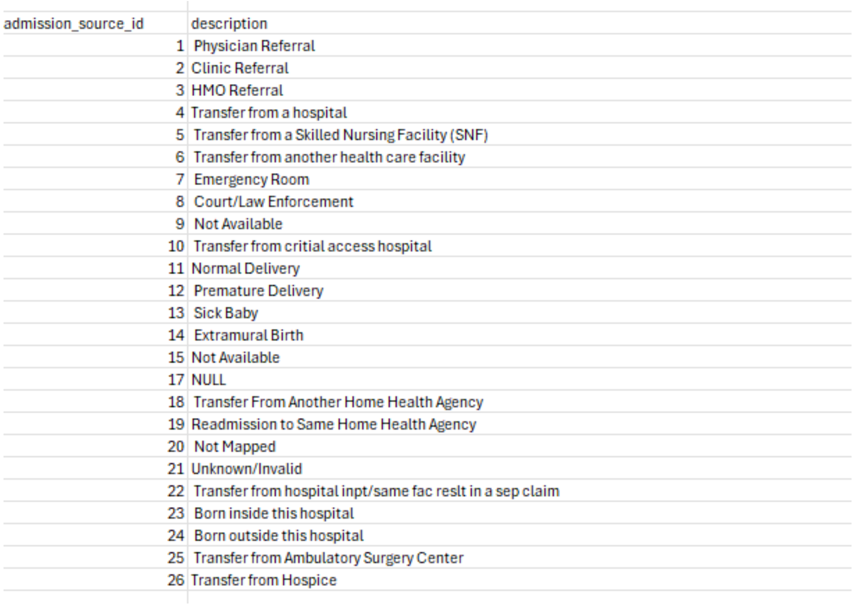
We observe that the dataset has 14 numerical columns. However, the columns admission_type_id, discharge_disposition_id, admission_source_id are mappings to various categories. Below are the tables corresponding to these columns and what each value specifies.



	admission_type_id	description
0	1	Emergency
1	2	Urgent
2	3	Elective
3	4	Newborn
4	5	Not Available
5	6	NaN
6	7	Trauma Center
7	8	Not Mapped

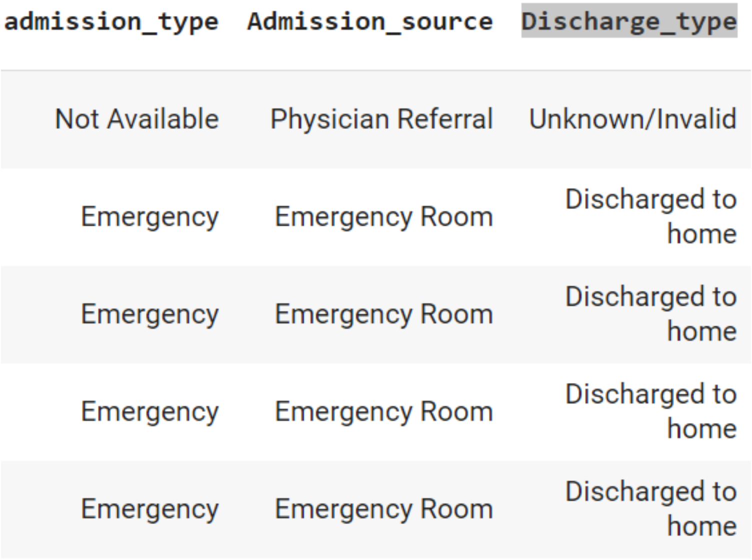


discharge_disposition_id	description
1	Discharged to home
2	Discharged/transferred to another short term hospital
3	Discharged/transferred to SNF
4	Discharged/transferred to ICF
5	Discharged/transferred to another type of inpatient care institution
6	Discharged/transferred to home with home health service
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
30	Discharged/transferred to another Type of Health Care Institution not Defined Elsewhere
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).

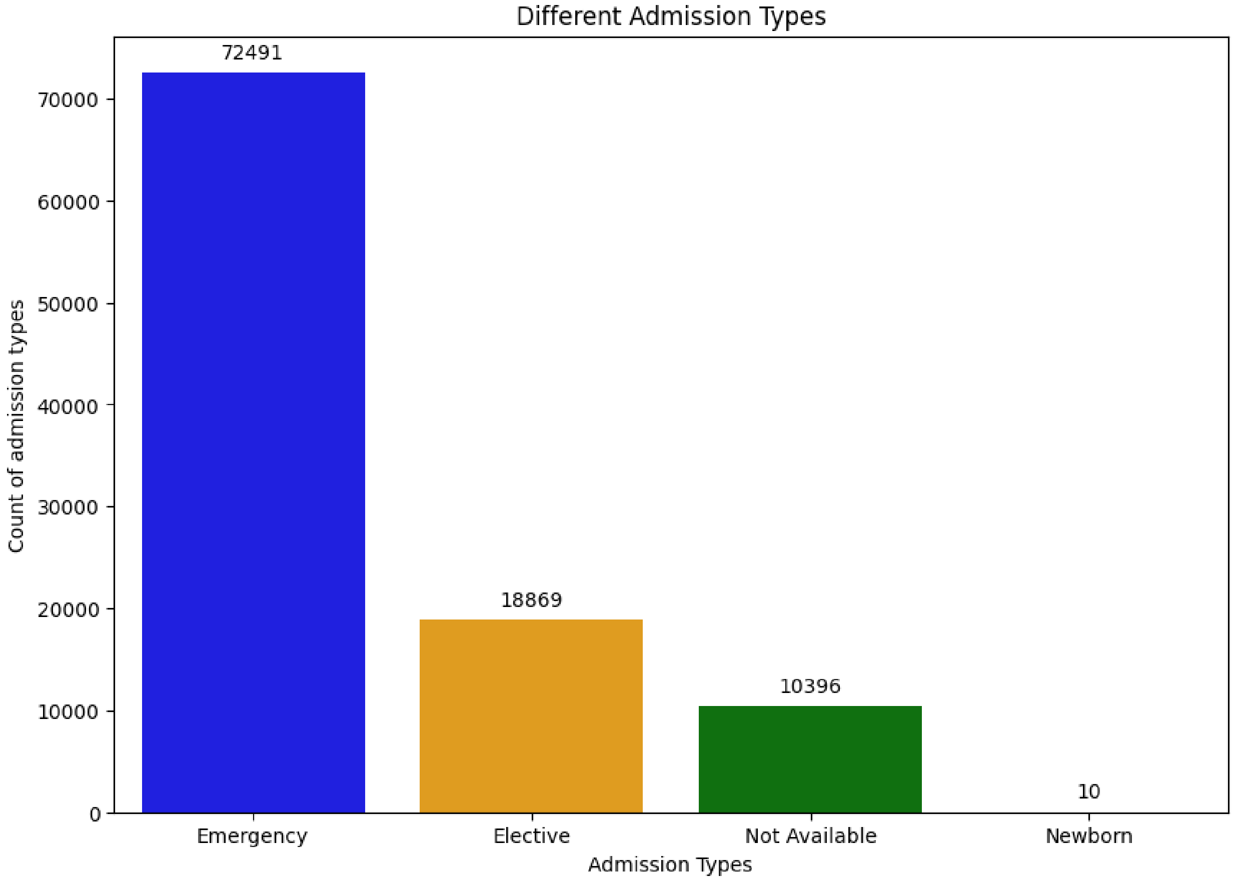


admission_source_id	description
1	Physician Referral
2	Clinic Referral
3	HMO Referral
4	Transfer from a hospital
5	Transfer from a Skilled Nursing Facility (SNF)
6	Transfer from another health care facility
7	Emergency Room
8	Court/Law Enforcement
9	Not Available
10	Transfer from critial access hospital
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

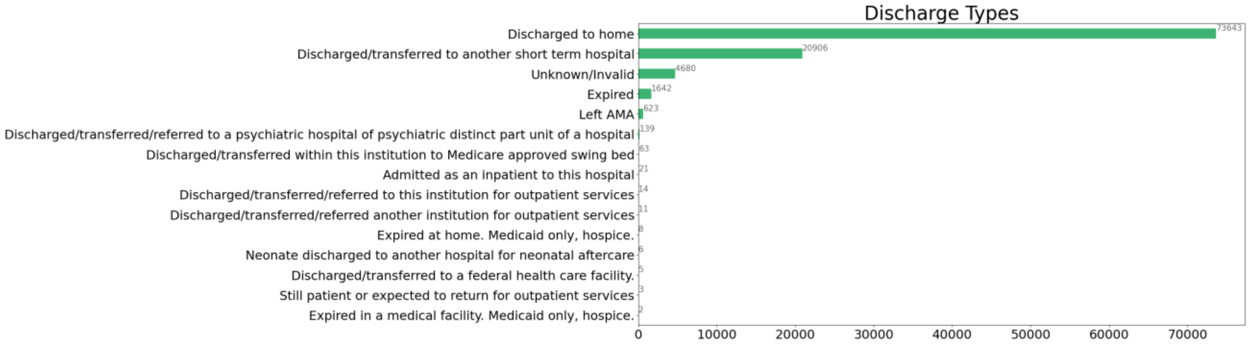
The original numeric identifiers are replaced with descriptive categories, enhancing the interpretability of the data. The new columns are named "admission_type," "Admission_source", “Discharge_type” and include mapped descriptions for admission type, admission source, and discharge disposition, respectively.



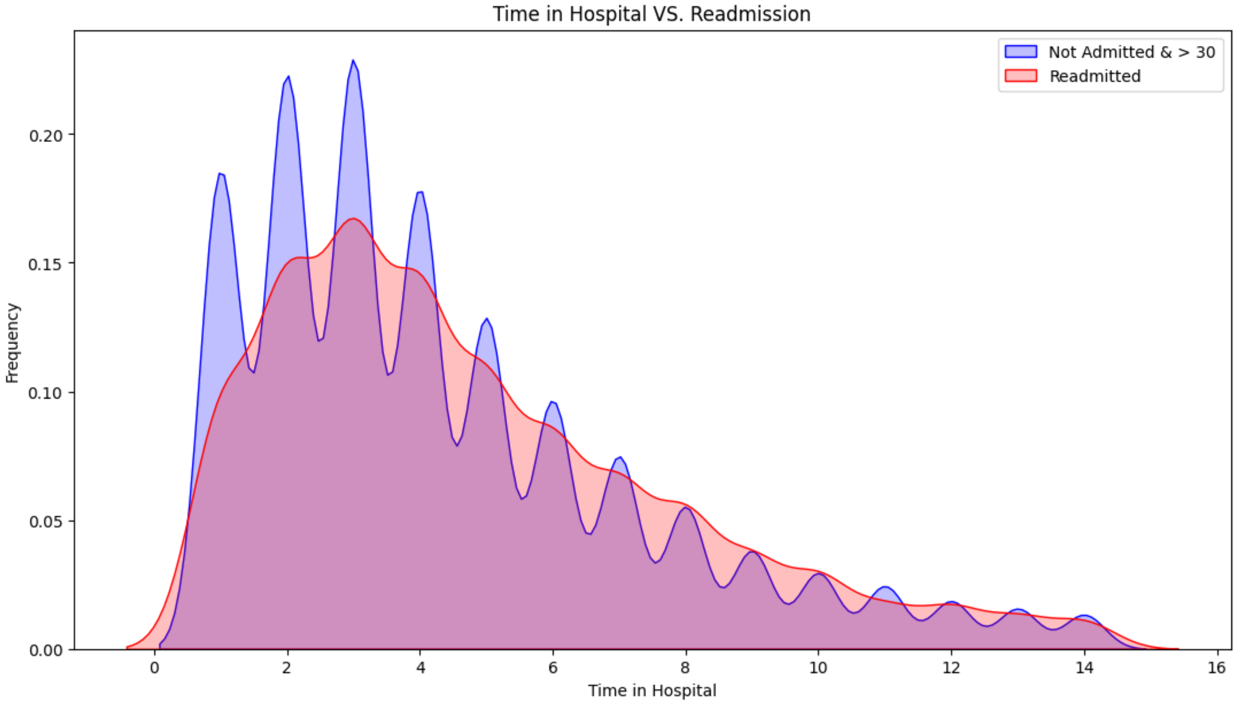
admission_type	Admission_source	Discharge_type
Not Available	Physician Referral	Unknown/Invalid
Emergency	Emergency Room	Discharged to home
Emergency	Emergency Room	Discharged to home
Emergency	Emergency Room	Discharged to home
Emergency	Emergency Room	Discharged to home



We can interpret from the above graph that the Emergency Admission Type has the highest frequency in the dataset, while Newborn has the least.



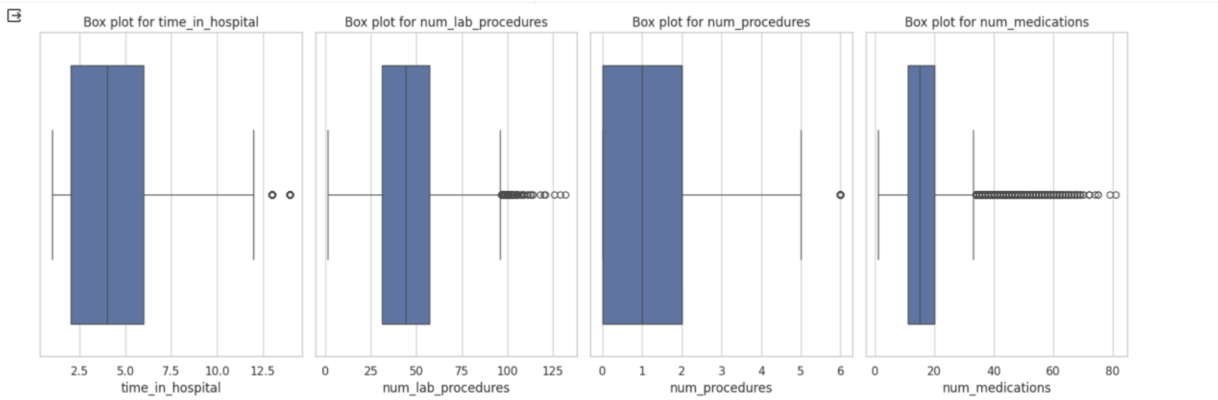
The above chart shows that the most common discharge type is "Discharged to home," followed by "Discharged/transferred to another short term hospital," and "Unknown/Invalid". We will handle these unknown values in the data preprocessing part.



The above KDE Plot graph shows that there is a positive correlation between the length of a patient's hospital stay and their risk of being readmitted. This means that patients who stay in the hospital for longer are more likely to be readmitted within 30 days.

There are several possible explanations for this correlation. For example, patients who stay in the hospital for longer may have more complex medical conditions that make them more likely to be readmitted. Additionally, patients who stay in the hospital for longer may be more likely to experience complications during their stay, which could also lead to readmission.

Below, we are plotting a boxplot which represents 4 numerical columns (‘time_in_hospital’, ‘num_lab_procedures’, ‘num_procedures’, ‘num_medications’)



After plotting, we could infer that:

Time in hospital:

- The distribution is skewed to the right, indicating that most patients have shorter stays, but some have much longer stays.
- The median time in hospital is around 4 days.
- There are some outliers, which could represent patients with complex medical conditions or complications.

Number of lab procedures:

- The distribution is roughly symmetrical, with most patients undergoing between 25 and 50 procedures.
- There are quite a few outliers with a very high number of procedures.

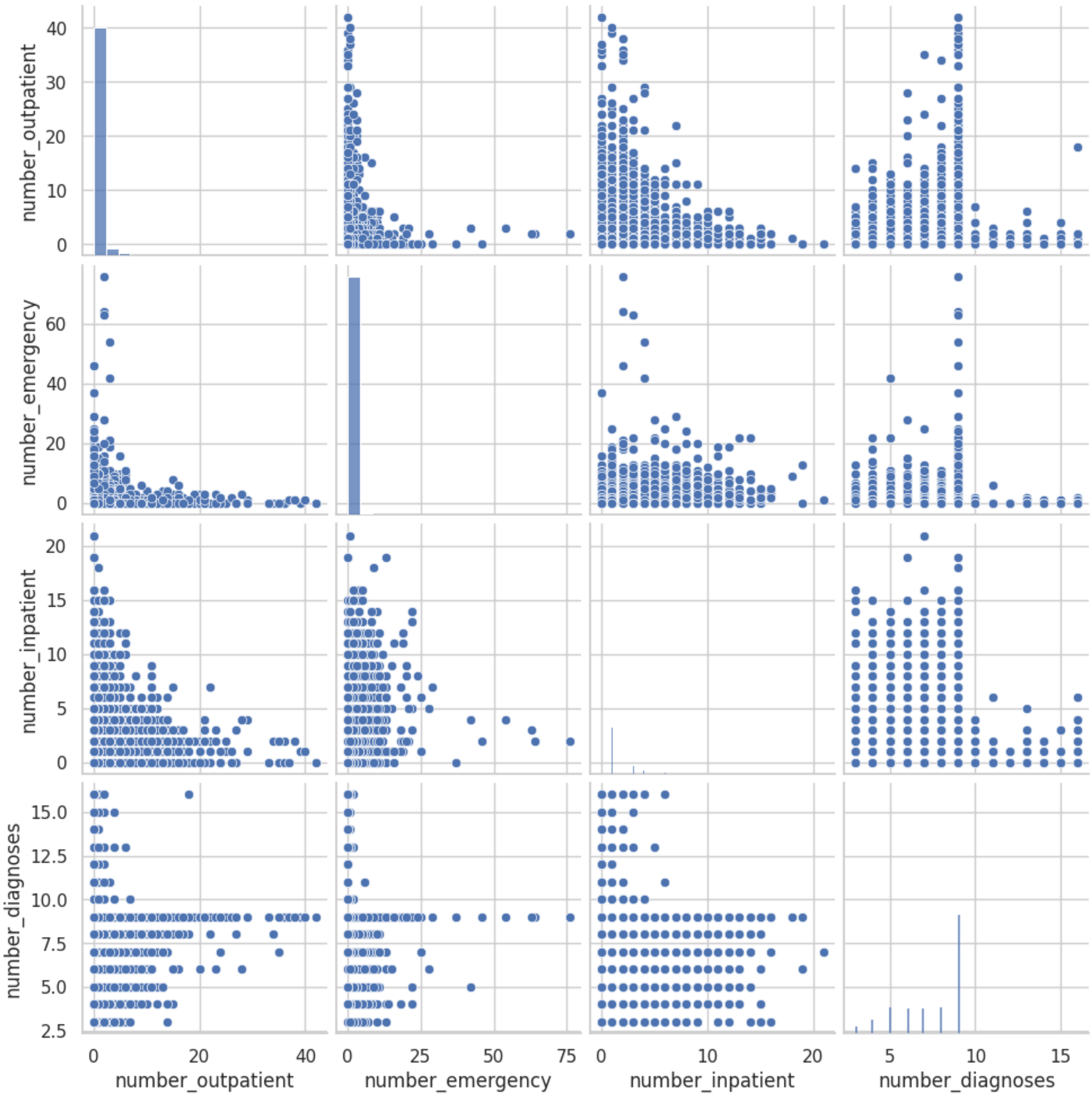
Number of procedures:

- The distribution is skewed to the right, with most patients having few procedures but some having many.
- The median number of procedures is around 2.
- There are some outliers with a very high number of procedures.

Number of medications:

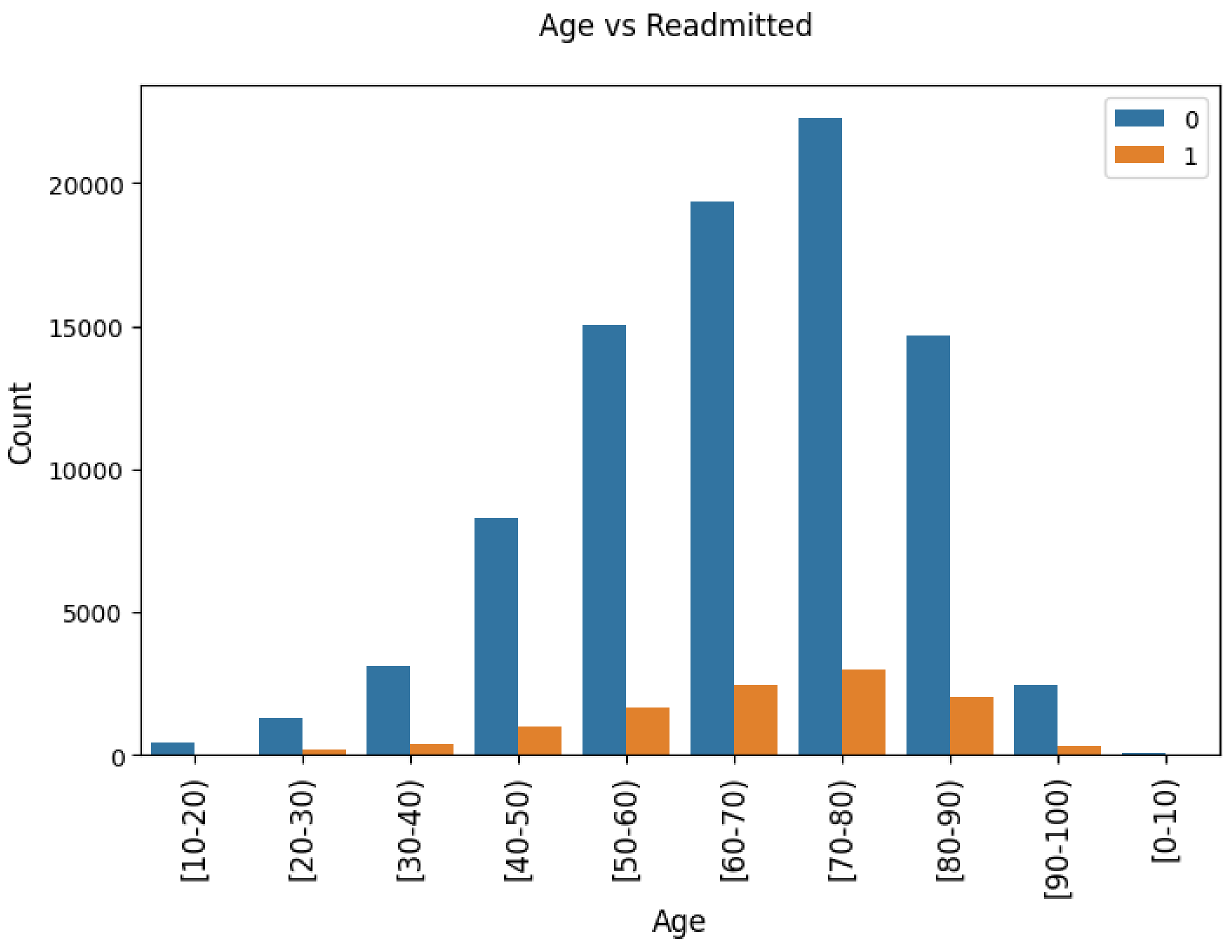
- The distribution is roughly symmetrical, with most patients taking between 20 and 40 medications.
- There are a few outliers taking many more medications.

Below, we are plotting a pairplot drawing relations between 4 other columns (‘number_outpatient','number_emergency','number_inpatient','number_diagnoses’)


number_outpatient
number_emergency
number_inpatient
number_diagnoses
0
10
20
30
40
0
20
40
60
0
5
10
15
20
2.5
5.0
7.5
10.0
12.5
15.0
0
20
40
0
25
50
75
0
10
20
5
10
15
number_outpatient
number_emergency
number_inpatient
number_diagnoses

From the above pairplot, we can infer that:

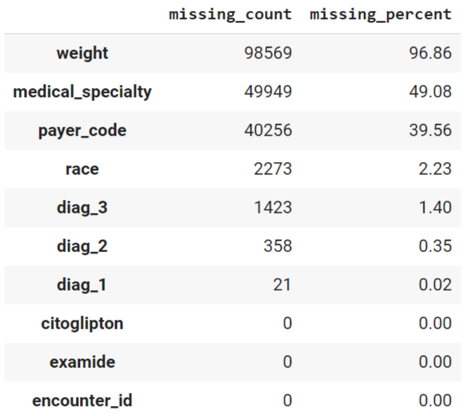
- **Number of patients with emergency case vs number of outpatients:** There is a weak positive correlation between the number of patients with emergency cases and the number of outpatients. This means that as the number of outpatients increases, the number of patients with emergency cases also tends to increase, but not very strongly.
- **Number of patients with emergency cases vs number of inpatients:** There is a weak positive correlation between the number of patients with emergency cases and the number of inpatients. This means that as the number of inpatients increases, the number of patients with emergency calls also tends to increase, but not very strongly.
- **Number of patients with emergency calls vs number of diagnoses:** There is a weak positive correlation between the number of patients with emergency calls and the number of diagnoses. This means that as the number of diagnoses increases, the number of patients with emergency calls also tends to increase, but not very strongly.
- **Number of outpatients vs number of inpatients:** There is a weak positive correlation between the number of outpatients and the number of inpatients. This means that as the number of outpatients increases, the number of inpatients also tends to increase, but not very strongly.
- **Number of outpatients vs number of diagnoses:** There is a weak positive correlation between the number of outpatients and the number of diagnoses. This means that as the number of outpatients increases, the number of diagnoses also tends to increase, but not very strongly.
- **Number of inpatients vs number of diagnoses:** There is a weak positive correlation between the number of inpatients and the number of diagnoses. This means that as the number of inpatients increases, the number of diagnoses also tends to increase, but not very strongly.



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

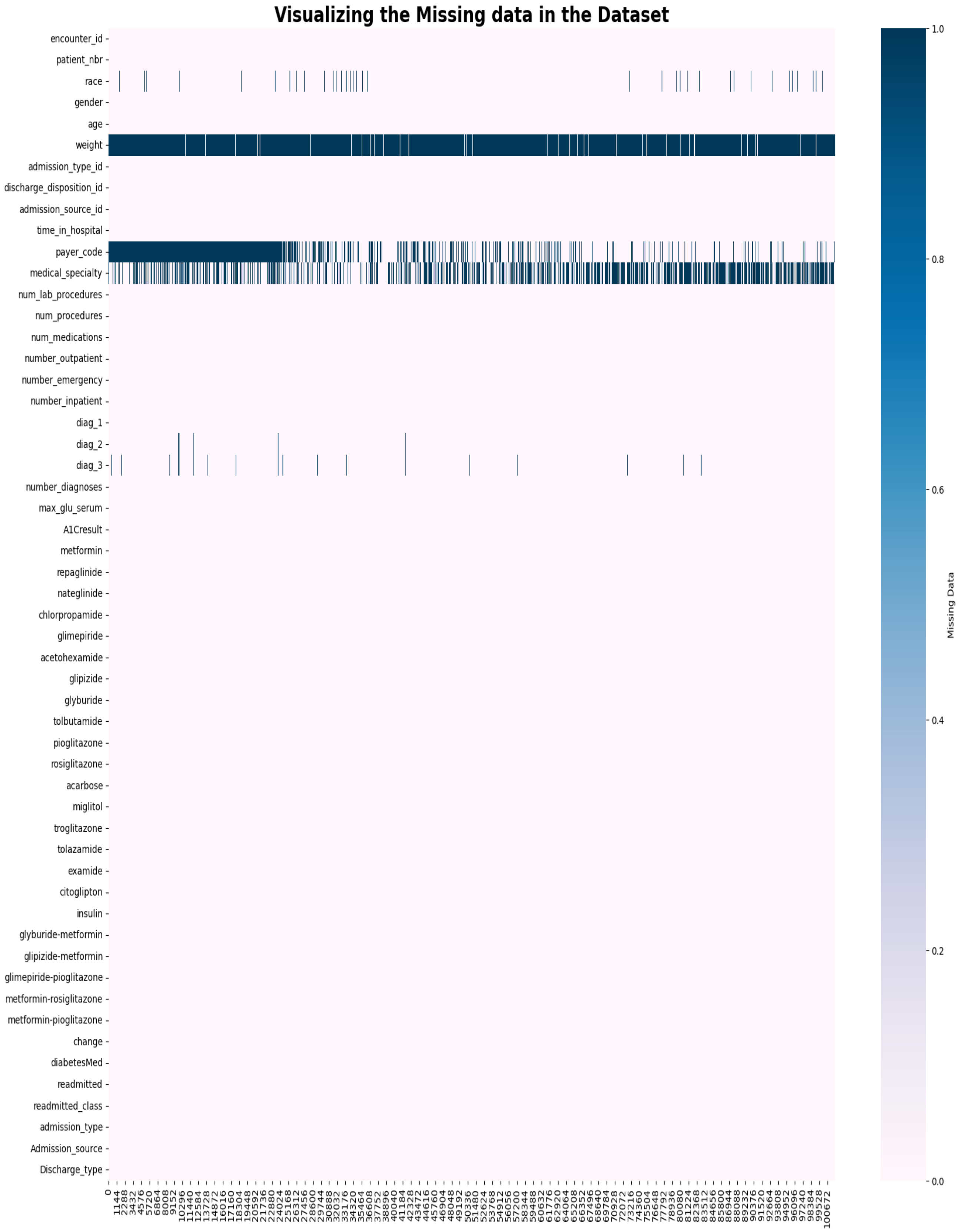
DATA CLEANING

Handling 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
citoglipton	0	0.00
examide	0	0.00
encounter_id	0	0.00

Visualizing the Missing Data using Heatmap:



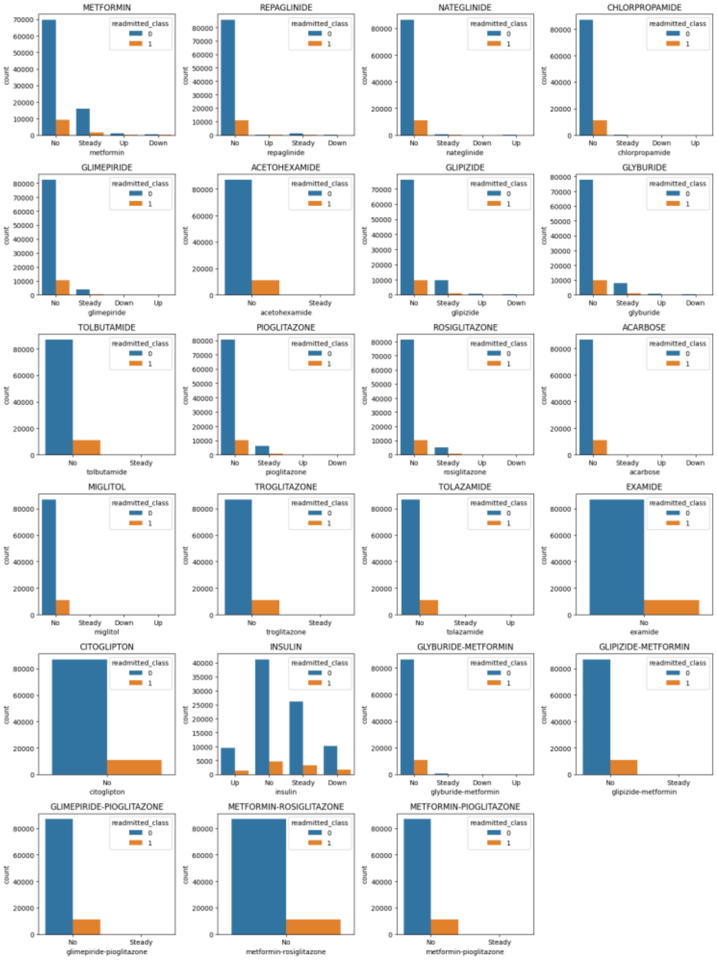
Since the variable 'weight' accounts for almost 97% of the missing values, we chose to eliminate this variable because there is no benefit to filling in the missing values. We also removed the variables 'Payer code' and 'medical_speciality' since they had roughly 40% and 50% missing entries, respectively. In comparison to other attributes we omitted, the variables 'race,' 'diag_1,' 'diag_2,' 'diag_3,' and 'gender' had comparatively few missing values. Therefore, we will drop the rows that have these values missing.

Number of instances after dropping missing rows: 98052

DATA PROCESSING

In our dataset, we have 24 features (columns) that showcase the medications given to the patient while being admitted in the hospital. These features indicate if the drug was prescribed or a change in the dosage was noted or not.

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.



From the above column chart, we see that the variables (drugs named ‘citoglipton’ and ‘examide’, ‘acetohexamide', 'tolbutamide', 'troglitazone', 'tolazamide', 'glipizide metformin', 'glimepiride-pioglitazone', 'metformin-rosiglitazone', 'metformin-pioglitazone'), all have the same value. So essentially these cannot provide any interpretive or discriminatory information for predicting readmission, so we decided to drop all these variables.

DATA ENCODING

Why do we perform Data Encoding?

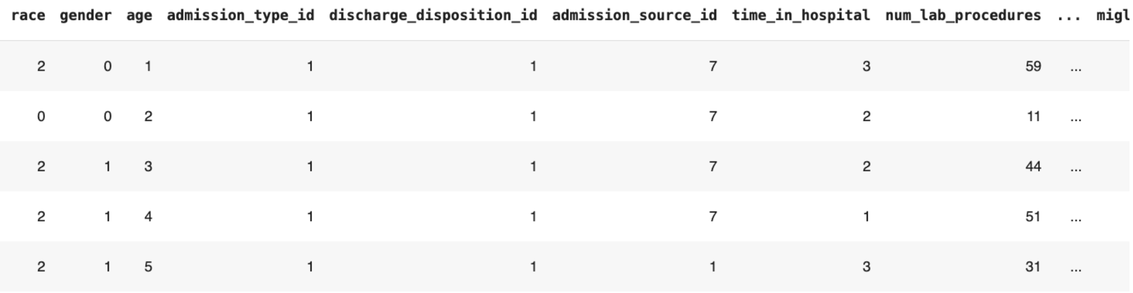
- **Encoding Ordinal Data**: Sometimes, categorical variables have an inherent order or ranking (e.g., low, medium, high). Label encoding assigns numerical values to categories in a way that preserves this order. It helps algorithms understand the ordinal nature of the data, allowing them to learn from it more effectively.
- **Improving Model Performance**: Label encoding can improve the performance of machine learning models. While some algorithms can handle categorical data directly (e.g., decision trees), others may perform better with numerical inputs. By converting categorical variables into numerical representations, we provide the model with more informative and discriminative features, potentially leading to better predictions.
- **Reducing Memory Usage**: Encoding categorical variables as integers can reduce the memory footprint of the dataset. Integer representations typically require less memory compared to storing categorical labels as strings, especially when dealing with large datasets.
- **Handling Textual Data**: In natural language processing tasks, label encoding is often used to convert text categories (e.g., sentiment labels like "positive", "neutral", "negative") into numerical representations that can be processed by machine learning models.

For the data encoding part in our dataset, we are using the “LabelEncoder” module.

LabelEncoder is a utility class provided by the **sklearn.preprocessing** module in scikit-learn, a popular machine learning library in Python. It is used for converting categorical labels (strings or integers) into numerical representations (integer labels).

- **Fit**: When you instantiate a **LabelEncoder** object and call its **fit ()** method with a list or array of categorical labels, it internally learns the mapping between unique labels and integers. It assigns a unique integer to each unique label encountered in the input data.
- **Transform**: After fitting, you can use the **transform ()** method to transform categorical labels into their corresponding integer representations based on the mapping learned during the fitting stage. This method takes a list or array of categorical labels as input and returns an array of corresponding integer labels.
- **Inverse Transform (Optional)**: Additionally, you can use the **inverse_transform()** method to convert the integer labels back to their original categorical labels.

In our code, we are using the LabelEncoder.fit_transform() to transform the categorical variables into numerical variables.



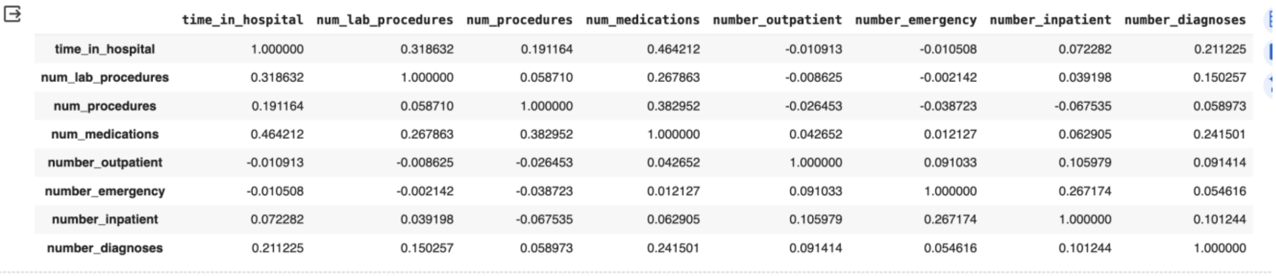
race	gender	age	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	num_lab_procedures	...	migl
2	0	1	1	1	7	3	59	...	
0	0	2	1	1	7	2	11	...	
2	1	3	1	1	7	2	44	...	
2	1	4	1	1	7	1	51	...	
2	1	5	1	1	1	3	31	...	

CORREALTION ANALYSIS

Correlation analysis is performed as part of dimension reduction, where we calculate the correlation between every 2 numerical columns using Pearson's coefficient.

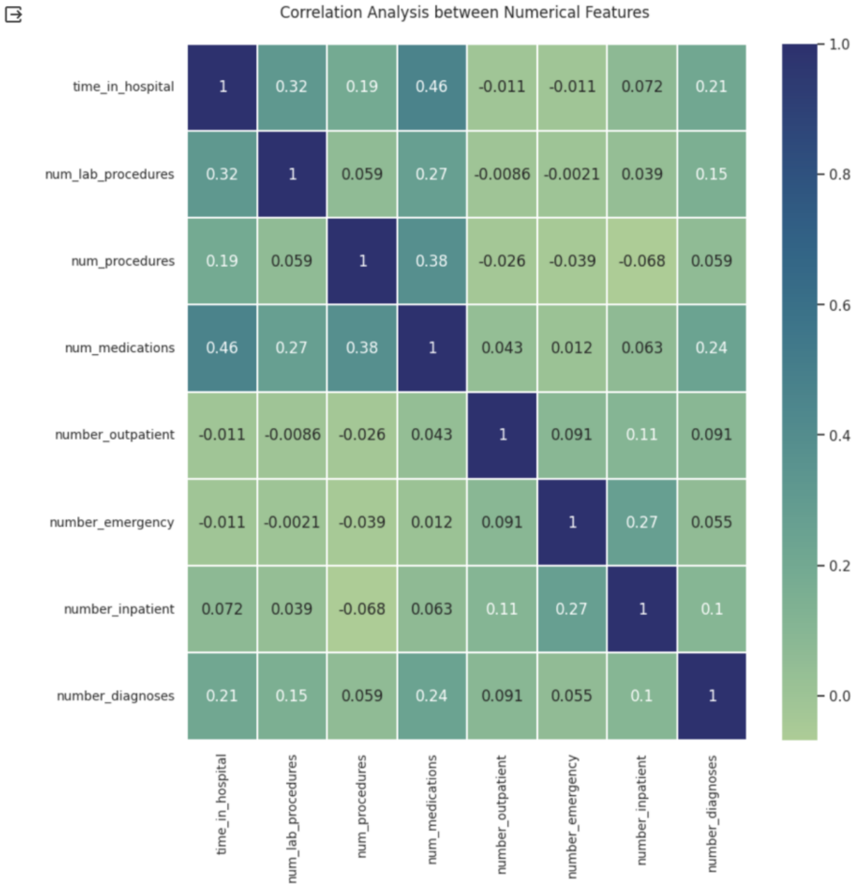
Pearson's correlation coefficient is a bivariate correlation that measures the linear correlation between two sets of data, whose value ranges from -1 to 1. Essentially, it is a normalized measurement of their covariances. If a pair of variables are highly correlated (for example, setting 0.8 as the cutoff for highly correlated variables), then we drop one of the column-pair.

Below is the correlation matrix, giving the relation between the 8 numerical variables from the dataset.



	time_in_hospital	num_lab_procedures	num_procedures	num_medications	number_outpatient	number_emergency	number_inpatient	number_diagnoses
time_in_hospital	1.000000	0.318632	0.191164	0.464212	-0.010913	-0.010508	0.072282	0.211225
num_lab_procedures	0.318632	1.000000	0.058710	0.267863	-0.008625	-0.002142	0.039198	0.150257
num_procedures	0.191164	0.058710	1.000000	0.382952	-0.026453	-0.038723	-0.067535	0.058973
num_medications	0.464212	0.267863	0.382952	1.000000	0.042652	0.012127	0.062905	0.241501
number_outpatient	-0.010913	-0.008625	-0.026453	0.042652	1.000000	0.091033	0.105979	0.091414
number_emergency	-0.010508	-0.002142	-0.038723	0.012127	0.091033	1.000000	0.267174	0.054616
number_inpatient	0.072282	0.039198	-0.067535	0.062905	0.105979	0.267174	1.000000	0.101244
number_diagnoses	0.211225	0.150257	0.058973	0.241501	0.091414	0.054616	0.101244	1.000000

The graph below is a heat-map depicting the same correlation between the 8 numerical values.



	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	-0.011	-0.011	0.072	0.21
num_lab_procedures	0.32	1	0.059	0.27	-0.0086	-0.0021	0.039	0.15
num_procedures	0.19	0.059	1	0.38	-0.026	-0.039	-0.068	0.059
num_medications	0.46	0.27	0.38	1	0.043	0.012	0.063	0.24
number_outpatient	-0.011	-0.0086	-0.026	0.043	1	0.091	0.11	0.091
number_emergency	-0.011	-0.0021	-0.039	0.012	0.091	1	0.27	0.055
number_inpatient	0.072	0.039	-0.068	0.063	0.11	0.27	1	0.1
number_diagnoses	0.21	0.15	0.059	0.24	0.091	0.055	0.1	1

FEATURE ENGINEERING

Feature engineering is the process of selecting, creating, or modifying features (variables) in a dataset to improve the performance of machine learning models. It involves transforming raw data into a format that better represents the underlying problem to the predictive models, thereby enhancing their performance.

Feature engineering is crucial in data mining and machine learning for several reasons:

1. Improving Model Performance

2. Dimensionality Reduction

3. Handling Non-linear Relationships

4. Dealing with Missing Data

5. Addressing Skewed Distributions

6. Domain-Specific Knowledge Incorporation

For our dataset, we are using Feature Engineering to address missing values, and mostly for dimension reduction, forming new columns by clubbing 2 or more columns which have similar values or are related to each other.

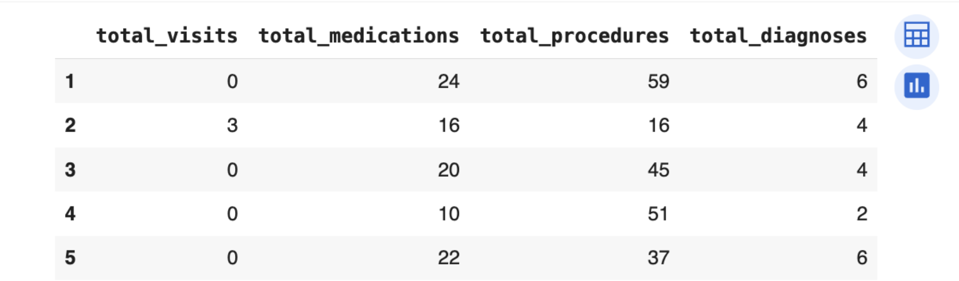
Below we are executing the technique to come up with new columns by clubbing various columns:

Total hospital visits = number of outpatient visits + number of emergency visits + number of inpatient visits

Total medications = number of medications given + number of diagnoses

Total procedures completed = number of lab procedures + number of other procedures

Total diagnoses = number of diagnoses + number of inpatient visits



	total_visits	total_medications	total_procedures	total_diagnoses
1	0	24	59	6
2	3	16	16	4
3	0	20	45	4
4	0	10	51	2
5	0	22	37	6

We will also use feature engineering techniques to reduce the categories within a single column.

We found that “Admission Type”, “Discharge Disposition” and “Admission Source” consists similar categories which are related to each other, so we club them into one main category.

This involves the re-encoding of admission type, discharge type, and admission source into fewer categories to simplify and consolidate the information. The specific mappings are as follows:

Admission Type:

- Merge categories 2 and 7 into category 1.
- Merge categories 6 and 8 into category 5.

Discharge Disposition:

- Merge categories 6, 8, and 13 into category 1.
- Merge categories 3, 4, 5, 14, 22, 23, and 24 into category 2.
- Merge categories 18 and 25 into category 26.

Admission Source:

- Merge categories 2 and 3 into category 1.
- Merge categories 5, 6, 10, 22, and 25 into category 4.
- Merge categories 15, 17, 20, and 21 into category 9.

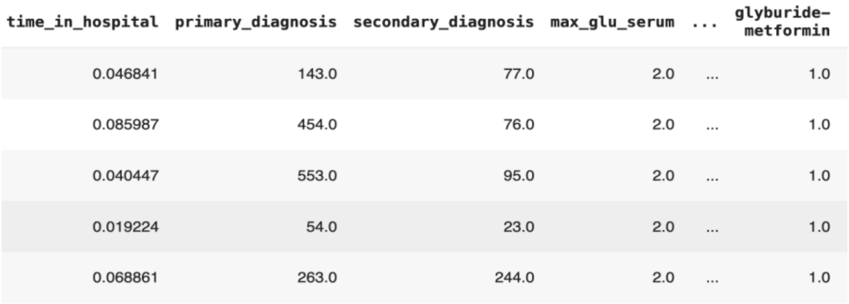
DATA SCALING/ NORMALIZATION

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.

For this step, we are going to use the “preprocessing” module from sklearn library

The **sklearn.preprocessing** module in Python's scikit-learn library is used for preprocessing data before feeding it into machine learning models.

Preprocessing is a crucial step in the machine learning pipeline as it helps to prepare the data in a format that is suitable for the chosen model and improves the model's performance.



time_in_hospital	primary_diagnosis	secondary_diagnosis	max_glu_serum	...	glyburide-metformin
0.046841	143.0	77.0	2.0	...	1.0
0.085987	454.0	76.0	2.0	...	1.0
0.040447	553.0	95.0	2.0	...	1.0
0.019224	54.0	23.0	2.0	...	1.0
0.068861	263.0	244.0	2.0	...	1.0

For our dataset, after using the preprocessing.normalize module, the numerical columns are standardize which is essential for better model performance.