# Diabetic Patient Readmission:
# A Comprehensive Analysis and Prediction

**Milestone: Project Report**


Group 4

Nivedita Ediga

Pranav Reddy

Santhosh Yedla

857-891-2703(Nivedita)
339-837-1453(Pranav)
857-313-4038(Santhosh)

ediga.n@northeastern.edu
reddy.prana@northeastern.edu
yedla.sa@northeastern.edu


**Percentage of Effort Contributed by Nivedita: _____33%_____**

**Percentage of Effort Contributed by Pranav: _____33%_____**

**Percentage of Effort Contributed by Santhosh: _____34%_____**

**Signature of Student 1:_____Nivedita_____**

**Signature of Student 2:_____Pranav_____**

**Signature of Student 3:_____Santhosh_____**

**Submission Date:__12/13/2023_____**

**Problem Setting and Problem Definition:**

Understanding the importance of effectively handling high blood sugar levels in hospitalized patients has increased, given its association with significant effects on both morbidity and mortality. This recognition has led to the establishment of structured protocols in the intensive care unit (ICU), where institutions frequently follow strict glucose targets. Nevertheless, a consistent systematic approach is not universally applied in non-ICU hospital admissions. The absence of standardized approaches in non-ICU settings presents difficulties in delivering reliable and efficient care to hospitalized individuals with elevated blood sugar.

Diabetes is a chronic health condition marked by prolonged elevation of blood glucose levels. Various factors like height, race, gender, and age play a role, but high sugar concentration is a significant contributor. A thorough examination of an extensive clinical database was carried out to scrutinize historical patterns in the care of diabetes patients admitted to a U.S. hospital. The objective is to extract insights that can shape future strategies for improving patient safety. A central policy concern revolves around the priority to decrease early hospital readmissions, aiming to elevate the overall quality of healthcare.

The infrequent measurement of HbA1c in hospitalized patients with diabetes impacts the management of hyperglycemia. So, it is very crucial to address the lack of national assessments of diabetes care during hospitalization and highlight the potential for improved patient outcomes and cost reduction through increased attention to HbA1c monitoring. Despite strong evidence supporting improved outcomes with proper interventions, there's a gap in diabetes management during hospitalization. This arbitrary care leads to increased readmissions, imposing financial burdens on hospitals and raising morbidity and mortality risks for patients.

Improving clinical outcomes for diabetic patients and easing the financial strain on healthcare providers has become quite essential. By pinpointing and addressing factors leading to early readmissions within 30 days of discharge, the goal is to enhance preventive and therapeutic interventions. This benefits patients by lowering the risk of diabetes related complications and streamlines healthcare delivery, making it more efficient and cost-effective for hospitals. The application extends beyond data analysis; it's about bridging the gap between evidence-based

practices and the practical challenges faced in hospital settings. By accurately predicting early readmission, healthcare providers can implement targeted strategies to improve glycemic control and overall diabetes management. Challenges in this effort include handling large and intricate datasets, necessitating advanced analytical tools. Safeguarding patient data privacy adds complexity. Accurately predicting readmission risk in the multifaceted realm of healthcare poses a significant challenge.

A significant share of inpatient services spending in hospitals is attributed to the expenses related to hospital readmissions. Diabetes, a leading global cause of death, stands out as the costliest chronic disease in the United States. Hospitalized individuals with diabetes face an increased risk of readmission compared to their non-diabetic counterparts. Consequently, lowering readmission rates for diabetic patients holds substantial promise for a notable reduction in medical expenditures. This project aims to predict the probability of readmission for patients with diabetes.

The goal is to use data analytics to address key questions:

1. What factors contribute to early readmission for diabetic patients?
2. Are there identifiable patterns in lab results indicating higher readmission risk?
3. Can predictive models accurately assess the risk of early readmission for individual diabetic patients?
4. What interventions or changes in diabetes management practices could reduce the risk of early readmission?

**Data Sources:**

The dataset that we are using for our project is “Diabetes 130-US hospitals for years 1999-2008”. Source: UCI Diabetes Dataset

The data was sourced from the Center for Machine Learning and Intelligent Systems at the University of California, Irvine. It comprises more than 100,000 attributes and includes 50 features, including metrics like the count of procedures, medications, and the duration of hospital stays, among others.

**Data Description:**

The dataset represents ten years (1999-2008) of clinical care (diabetic patient records) at 130 US hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes. The dataset contains attributes such as patient number, race, gender, age, admission type, time in hospital, medical specialty of admitting physician, number of lab tests performed, HbA1c test result, diagnosis, number of medications, diabetic medications, number of outpatient, inpatient, and emergency visits in the year before the hospitalization, etc. Every row pertains to the hospital records of patients diagnosed with diabetes, including information on laboratory tests, medications, and a maximum stay of 14 days.

Target variable: Readmitted

Number of Columns: 50

Number of rows: 101,766

A short description of each feature is provided below:

| Feature | 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 | 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. |

**Data Exploration and Data Mining Tasks:**

While performing exploratory data analysis, it is essential to get an overview of the dataset, especially when dealing with large datasets where understanding the structure and the data types is crucial. So, below is the summary of our dataset.

```
 #    Column                    Non-Null Count    Dtype
---   ------                    --------------    -----
 0    encounter_id              101766 non-null   int64
 1    patient_nbr               101766 non-null   int64
 2    race                      101766 non-null   object
 3    gender                    101766 non-null   object
 4    age                       101766 non-null   object
 5    weight                    101766 non-null   object
 6    admission_type_id         101766 non-null   int64
 7    discharge_disposition_id  101766 non-null   int64
 8    admission_source_id       101766 non-null   int64
 9    time_in_hospital          101766 non-null   int64
 10   payer_code                101766 non-null   object
 11   medical_specialty         101766 non-null   object
 12   num_lab_procedures        101766 non-null   int64
 13   num_procedures            101766 non-null   int64
 14   num_medications           101766 non-null   int64
 15   number_outpatient         101766 non-null   int64
 16   number_emergency          101766 non-null   int64
 17   number_inpatient          101766 non-null   int64
 18   diag_1                    101766 non-null   object
 19   diag_2                    101766 non-null   object
 20   diag_3                    101766 non-null   object
 21   number_diagnoses          101766 non-null   int64
 22   max_glu_serum             101766 non-null   object
 23   A1Cresult                 101766 non-null   object
 24   metformin                 101766 non-null   object
 25   repaglinide               101766 non-null   object
 26   nateglinide               101766 non-null   object
 27   chlorpropamide            101766 non-null   object
 28   glimepiride               101766 non-null   object
 29   acetohexamide             101766 non-null   object
 30   glipizide                 101766 non-null   object
 31   glyburide                 101766 non-null   object
 32   tolbutamide               101766 non-null   object
 33   pioglitazone              101766 non-null   object
 34   rosiglitazone             101766 non-null   object
 35   acarbose                  101766 non-null   object
 36   miglitol                  101766 non-null   object
 37   troglitazone              101766 non-null   object
 38   tolazamide                101766 non-null   object
 39   examide                   101766 non-null   object
 40   citoglipton               101766 non-null   object
 41   insulin                   101766 non-null   object
 42   glyburide-metformin       101766 non-null   object
 43   glipizide-metformin       101766 non-null   object
 44   glimepiride-pioglitazone  101766 non-null   object
 45   metformin-rosiglitazone   101766 non-null   object
 46   metformin-pioglitazone    101766 non-null   object
 47   change                    101766 non-null   object
 48   diabetesMed               101766 non-null   object
 49   readmitted                101766 non-null   object
dtypes: int64(13), object(37)
```

Next, statistics summary of numerical columns is used to understand the data distribution.

1. encounter_id and patient_nbr: These are identifiers for each encounter and patient, respectively. The range and spread of values indicate the variability in the dataset.

2. admission_type_id, discharge_disposition_id, and admission_source_id: These categorical variables seem to represent different aspects of the admission process. The mean and quartile values give an idea of the distribution.
3. time_in_hospital: On average, patients spend around 4.4 days in the hospital, with a range from 1 to 14 days.
4. num_lab_procedures: The average number of laboratory procedures is around 43, with a range from 1 to 132. This suggests variability in the level of testing or monitoring.
5. num_procedures: On average, patients undergo about 1.34 procedures during their hospital stay, with a maximum of 6.
6. num_medications: The average number of medications prescribed is approximately 16, with a range from 1 to 81.
7. number_outpatient, number_emergency, number_inpatient: These variables seem to represent the number of outpatient visits, emergency visits, and inpatient admissions, respectively. The average values and ranges indicate the frequency of these events.
8. number_diagnoses: On average, patients have around 7.42 recorded diagnoses, with a range from 1 to 16.

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| encounter_id | 101766.0 | 1.652016e+08 | 1.026403e+08 | 12522.0 | 84961194.0 | 152388987.0 | 2.302709e+08 | 443867222.0 |
| patient_nbr | 101766.0 | 5.433040e+07 | 3.869636e+07 | 135.0 | 23413221.0 | 45505143.0 | 8.754595e+07 | 189502619.0 |
| admission_type_id | 101766.0 | 2.024006e+00 | 1.445403e+00 | 1.0 | 1.0 | 1.0 | 3.000000e+00 | 8.0 |
| discharge_disposition_id | 101766.0 | 3.715642e+00 | 5.280166e+00 | 1.0 | 1.0 | 1.0 | 4.000000e+00 | 28.0 |
| admission_source_id | 101766.0 | 5.754437e+00 | 4.064081e+00 | 1.0 | 1.0 | 7.0 | 7.000000e+00 | 25.0 |
| time_in_hospital | 101766.0 | 4.395987e+00 | 2.985108e+00 | 1.0 | 2.0 | 4.0 | 6.000000e+00 | 14.0 |
| num_lab_procedures | 101766.0 | 4.309564e+01 | 1.967436e+01 | 1.0 | 31.0 | 44.0 | 5.700000e+01 | 132.0 |
| num_procedures | 101766.0 | 1.339730e+00 | 1.705807e+00 | 0.0 | 0.0 | 1.0 | 2.000000e+00 | 6.0 |
| num_medications | 101766.0 | 1.602184e+01 | 8.127566e+00 | 1.0 | 10.0 | 15.0 | 2.000000e+01 | 81.0 |
| number_outpatient | 101766.0 | 3.693572e-01 | 1.267265e+00 | 0.0 | 0.0 | 0.0 | 0.000000e+00 | 42.0 |
| number_emergency | 101766.0 | 1.978362e-01 | 9.304723e-01 | 0.0 | 0.0 | 0.0 | 0.000000e+00 | 76.0 |
| number_inpatient | 101766.0 | 6.355659e-01 | 1.262863e+00 | 0.0 | 0.0 | 0.0 | 1.000000e+00 | 21.0 |
| number_diagnoses | 101766.0 | 7.422607e+00 | 1.933600e+00 | 1.0 | 6.0 | 8.0 | 9.000000e+00 | 16.0 |

In summary, this dataset contains information about medical encounters, with details on patient identifiers, admission details, duration of hospital stay, procedures, medications, and various types of visits. The summary statistics provide an overview of the central tendency and variability of these variables in the dataset.

Next, we have identified and handled missing values as it can significantly impact the validity and reliability of our analysis.

```
encounter_id                    0
patient_nbr                     0
race                         2273
gender                          0
age                             0
weight                      98569
admission_type_id               0
discharge_disposition_id        0
admission_source_id             0
time_in_hospital                0
payer_code                  40256
medical_specialty           49949
num_lab_procedures              0
num_procedures                  0
num_medications                 0
number_outpatient               0
number_emergency                0
number_inpatient                0
diag_1                         21
diag_2                        358
diag_3                       1423
number_diagnoses                0
max_glu_serum                   0
A1Cresult                       0
metformin                       0
repaglinide                     0
nateglinide                     0
chlorpropamide                  0
glimepiride                     0
acetohexamide                   0
glipizide                       0
glyburide                       0
tolbutamide                     0
pioglitazone                    0
rosiglitazone                   0
acarbose                        0
miglitol                        0
troglitazone                    0
tolazamide                      0
examide                         0
citoglipton                     0
insulin                         0
glyburide-metformin             0
glipizide-metformin             0
glimepiride-pioglitazone        0
metformin-rosiglitazone         0
metformin-pioglitazone          0
change                          0
diabetesMed                     0
readmitted                      0
dtype: int64
```

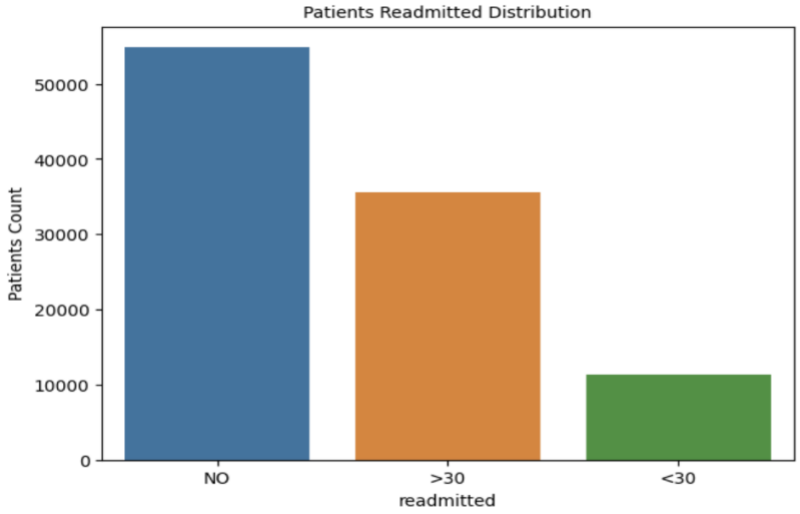
Based on the earlier results, we can deduce that the features like weight, payer_code, and medical_speciality contain a significant amount of missing data. Specifically, the weight feature exhibits an overwhelming 97% of missing values, while both payer_code and medical_speciality show approximately 50% of missing values each. So, we have eliminated the specified columns to improve the model performance. In addition, we have dropped off encounter_id and patient_nbr features as they don't contribute much towards the prediction.
Now, we have 101,766 observations and 45 features in our dataset.

In our pursuit of understanding early readmission factors, we categorized the readmission outcome into two facets: "readmitted," signifying a return within 30 days post-discharge, and "otherwise," encompassing both readmissions beyond 30 days and instances of no readmission. The 30-day threshold aligns with criteria commonly employed by funding agencies, underscoring the significance of this temporal window in unraveling the complexities of patient outcomes.

We have defined a function "plotCharts" to plot the charts to explore the data in detail. Additionally, a common function to plot charts for comparing the relevant column with "readmitted" column has also been defined. A function to remove outliers in dataset is quite essential when data exploration is being performed. Removing outliers in data mining is crucial to prevent distortions in summary statistics, model performance, and assumptions of analysis techniques. Outliers can mislead and negatively impact the overall quality and reliability of the dataset, so their careful identification and removal contribute to more accurate and robust data analysis.
Here, we have performed exploratory data analysis on every feature, including target variable as each feature might have its own story to tell, exploring them individually helped us to answer questions about the potential outliers and anomalies.

**Readmitted:**

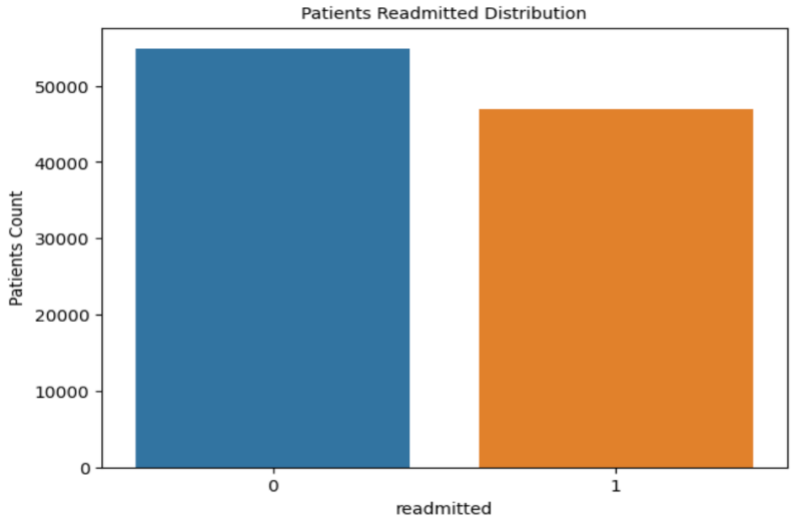
A bar chart is plotted between number of patients and readmitted feature.




From the above figure, the number of records of the readmitted feature outcomes are displayed as below:

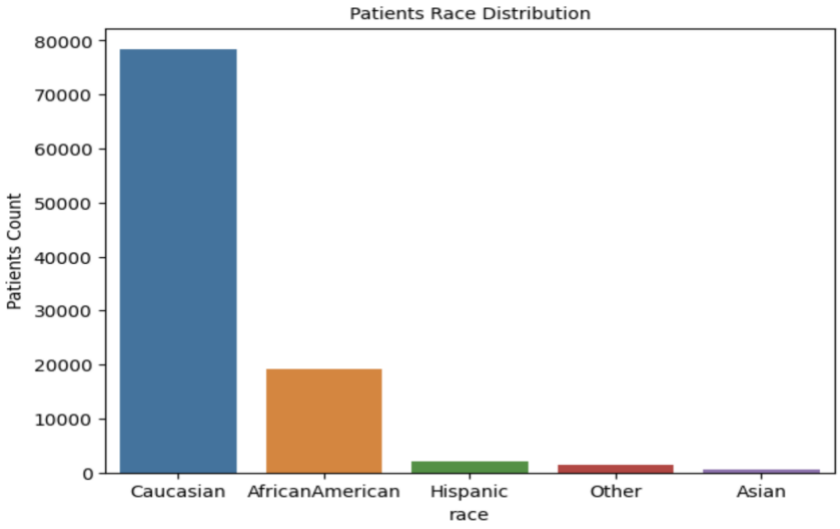
```
NO     54864
>30    35545
<30    11357
Name: readmitted, dtype: int64
```

Our main aim of the project is to predict if a patient would be readmitted or not. So, to make our work a bit easier we have classified the output into two classes ‘0’ for ‘No’ and ‘1’ for ‘Yes’. Therefore, we have merged ‘<30’ and ‘>30’ into class ‘1’.



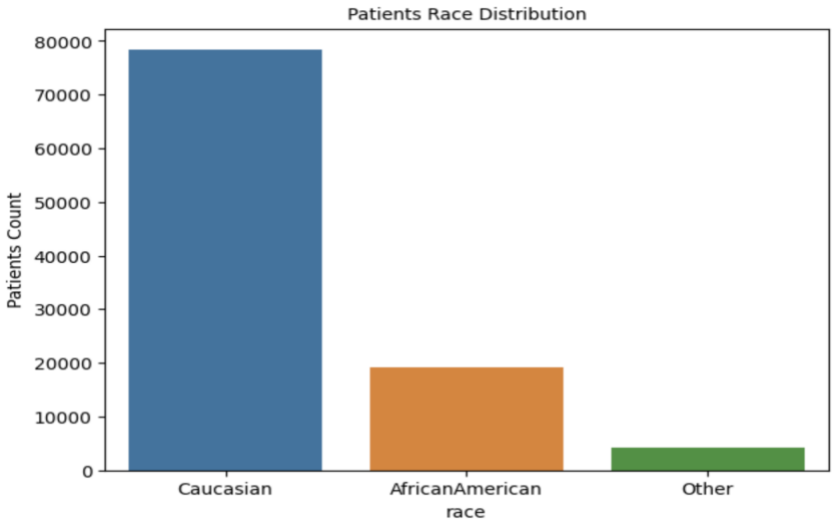

**Race:**

We have noticed that ‘Race’ feature has some missing values, so we have imputed those values with the mode. Mode imputation in data mining is beneficial because it preserves the distribution of existing data, providing a quick and simple way to fill in missing categorical values with the most common category. This method maintains relationships between variables and is particularly useful when dealing with large datasets, where computational efficiency is crucial.

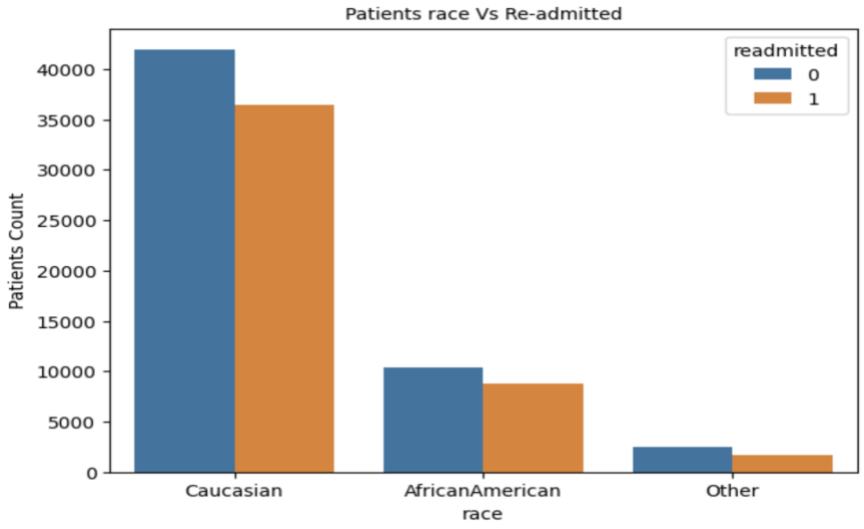



Since ‘Hispanic’ and ‘Asian’ has very a smaller number of records, we have merged these columns with the ‘Other’ column. Combining less common categories like 'Hispanic' and 'Asian' into an

'Other' category during data exploration and mining is beneficial for addressing sparse data issues, enhancing result interpretability, and improving the robustness of models. This consolidation simplifies analysis, improves visualizations, and guards against overfitting, ultimately contributing to a more focused and reliable understanding of the data.

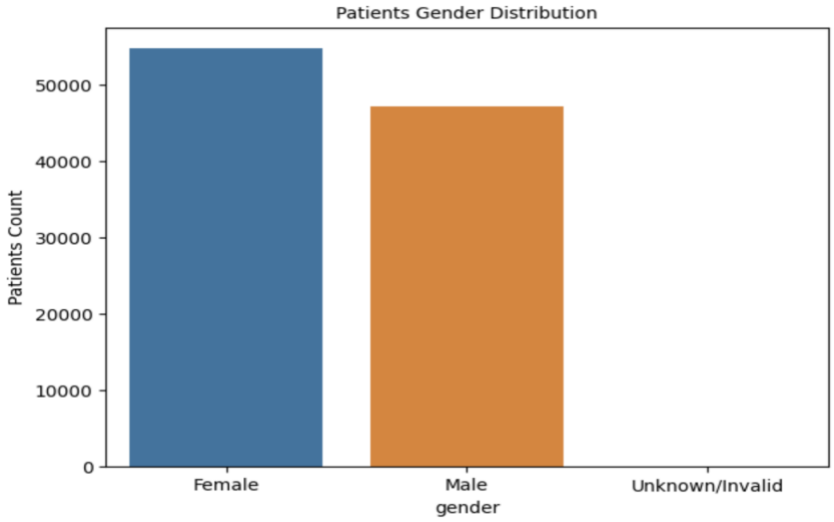



The above bar chart representation shows the number of patients by race in various hospitals. The majority of patients are Caucasian, followed by African American and other (consolidated ‘Hispanic’ and ‘Asian’) races. There are roughly 75,000 patients of Caucasian descent, 15,000 of African American descent, and fewer than 5,000 patients from other racial backgrounds.

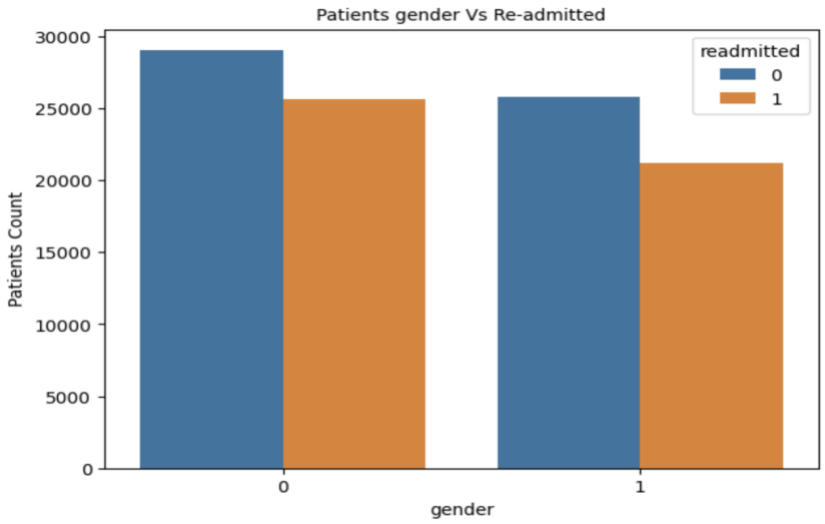


The clustered bar chart above illustrates the distribution of patients across different racial groups, showcasing the respective counts of readmitted and non-readmitted cases to the hospital. The trend in the readmission and non-readmission of patients followed a nearly identical pattern, with Caucasians comprising the largest count of patients in both categories.

**Gender:**

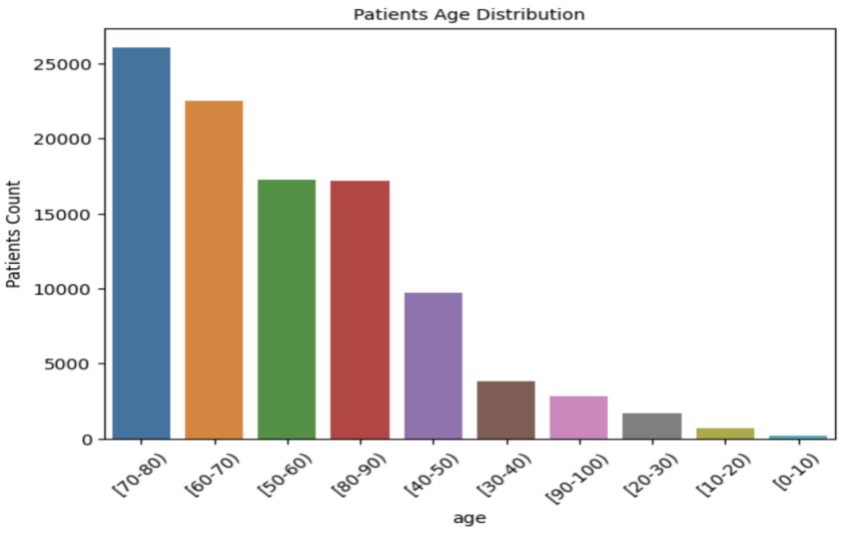



Based on the above visual representation, there are approximately 54,708 female patients and 47,055 male patients. However, the category labeled 'Unknown/Invalid' only consists of 3 patients, a notably smaller count compared to the other categories. Consequently, we have decided to eliminate those specific records. Next, we converted the column into a binary format, assigning 'Female' the value of 0 and 'Male' the value of 1.

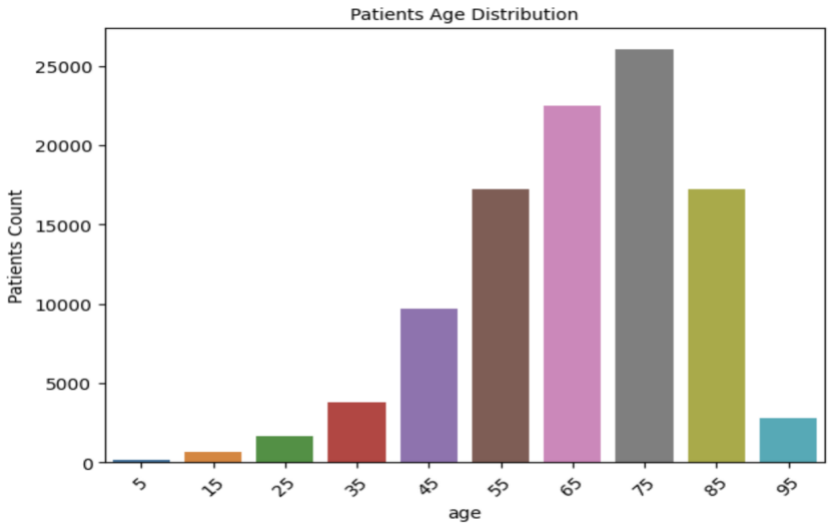



It is evident from the above graph that female patients exhibit a higher readmission rate than their male counterparts.

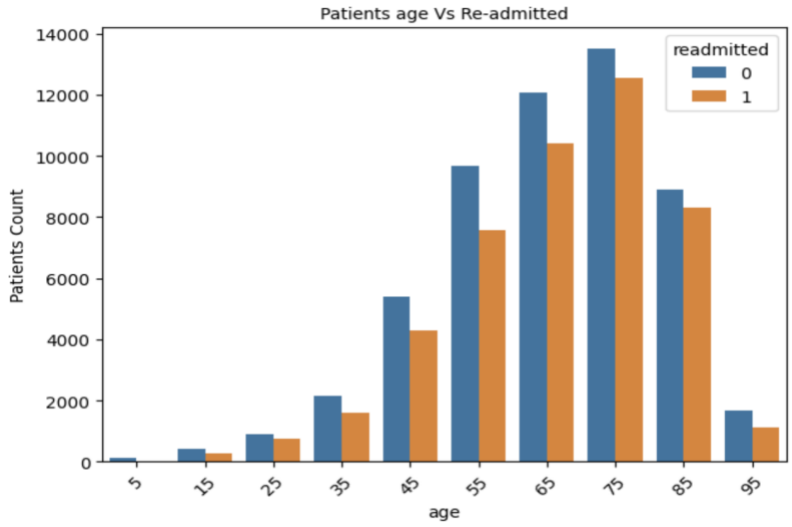
**Age:**




From the above illustration, notable observations include the highest concentration of patients (26,066) falling within the 70-80 age range. Conversely, the lowest count of patients (161) is observed in the 0-10 age bracket. Furthermore, the 50-60 and 80-90 age groups exhibit a nearly identical count of patients.




Regarding the age brackets, we will calculate the average by taking the midpoint between the lower and upper values within each bin. The visual representation above underscores that the dataset primarily comprises records from elderly patients, indicating a significant representation of this demographic in the dataset.




The distribution between the number of patients who were readmitted and those who were not readmitted is the same. It suggests that there's no significant imbalance in the number of cases for each outcome, and both outcomes are equally likely.

**Admission_type_id:**

Admission_type_id encompasses eight distinct categories, each corresponding to its respective numerical mapping as provided below.

Emergency – 1: 53988 records

Urgent – 2: 18480 records

Elective – 3: 18868 records

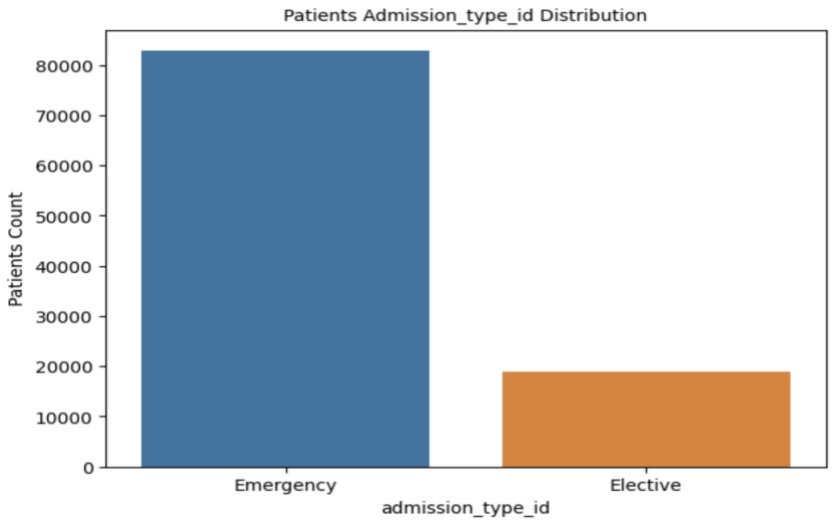
Newborn – 4: 10 records

Not Available – 5: 4785 records

NULL – 6: 5291 records

Trauma Center – 7: 21 records

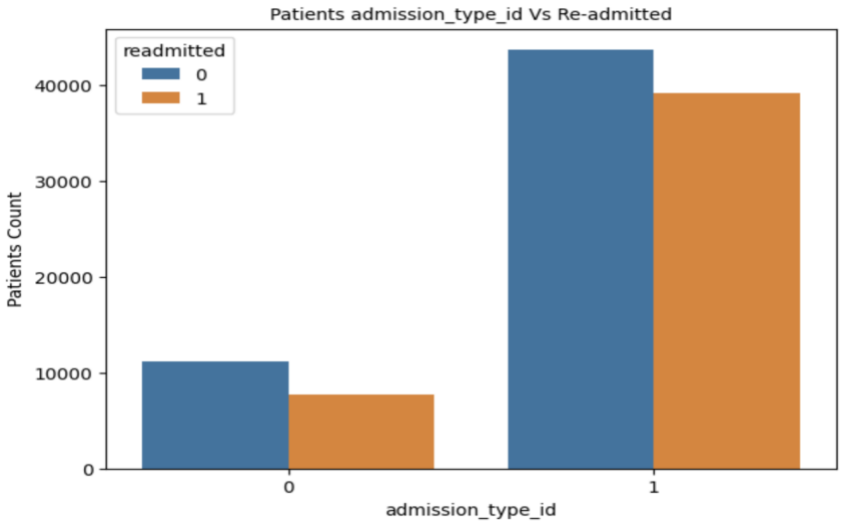
Not Mapped – 8: 320 records




We merged 'Emergency' and 'Urgent' care into a unified group labeled 'Emergency.' Additionally, 'Not Available,' 'Not Mapped,' and 'Null' have all been consolidated under the category 'Null.' Considering that 'Trauma Center' and 'Newborn' account for less than 0.05% of the data, we have opted to exclude them from the analysis.

There were 10,396 instances with missing values, therefore, we utilized the mode to substitute values in lieu of the Null entries.

Subsequently, we converted the column into a binary format by replacing 'Elective' with '0' and 'Emergency' with '1'. As a result, the count of records associated with the '0' class is 18,868, while the '1' class has 82,864 instances.



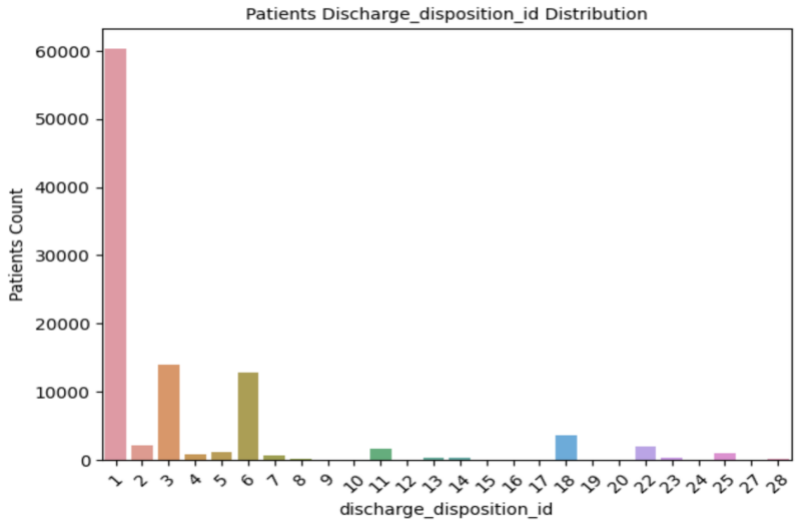

From the above representation, we can say that the distribution of patients is identical between those who were readmitted and those who were not readmitted.

**Discharge_disposition_id:**

Description:

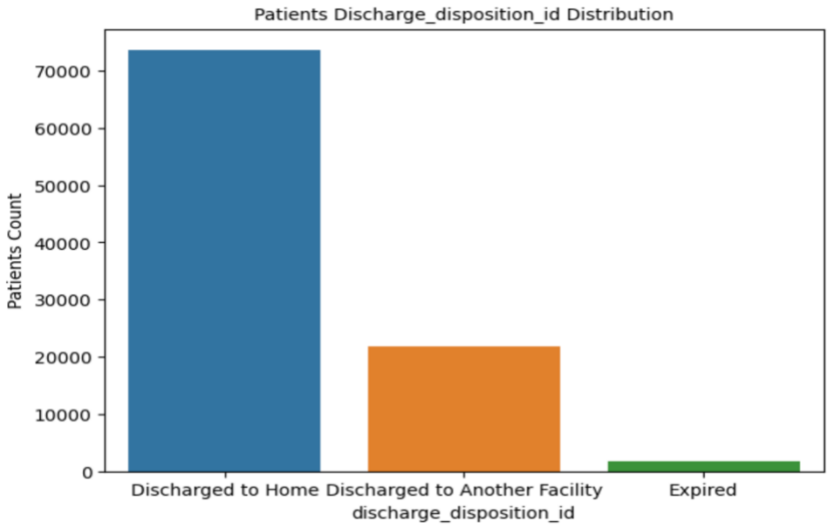
1. Discharged to home.
2. Discharged/transferred to another short-term hospital.
3. Discharged/transferred to SNF.
4. Discharged/transferred to ICF.
5. Discharged/transferred to another type of inpatient care institution.
6. Discharged/transferred to home with home health service.
7. Left AMA
8. Discharged/transferred 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 facility 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 another Type of Health Care Institution not Defined Elsewhere
28. Discharged/transferred to a federal health care facility.
29. Discharged/transferred/referred to a psychiatric hospital of psychiatric distinct part unit of a hospital.
30. Discharged/transferred to a Critical Access Hospital (CAH)




It is evident that there is an abundance of data points. To address this issue, we applied the following criteria:

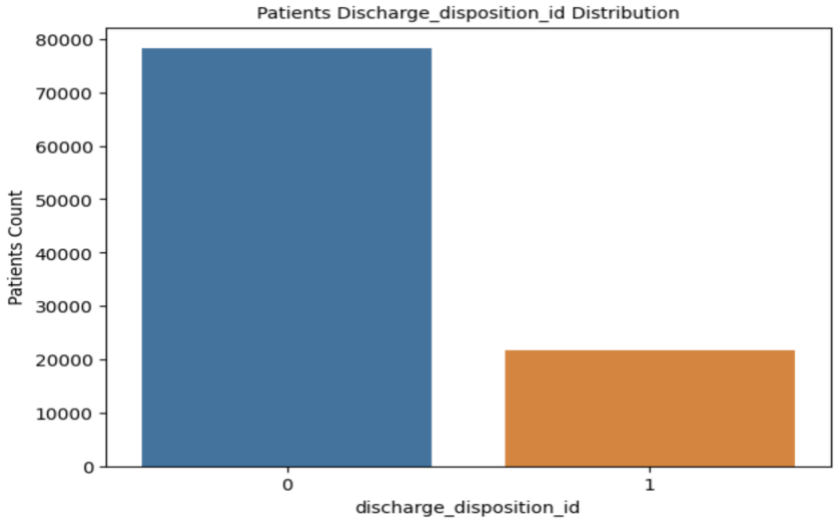
1. Patients discharged to home were grouped as "Home."
2. Those sent elsewhere were categorized as "Another Facility."
3. Patients marked as expired or in hospice were classified under "Expired."
4. Specific values such as 25, 26, and 18 were designated as null values.



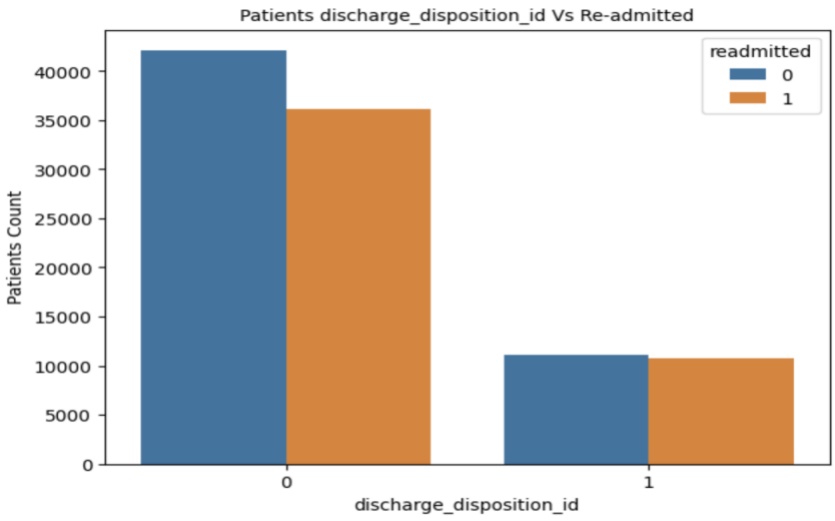

We have eliminated the patient data for individuals who have passed away, as they will not undergo readmission. Thus, the information of patients labeled as "Expired" has been omitted from

our dataset. Also, there were 10,396 instances with missing values; therefore, we have employed the mode imputation to replace these Null values.

Following the conversion of the column to a binary format, where 'Discharged to Another Facility' is represented as '1' and 'Discharged to Home' is denoted as '0', the ultimate visual depiction of the Discharge_disposition_id column is presented below.




Class ‘0’ has 78,303 instances, whereas Class ‘1’ has 21,780 instances.



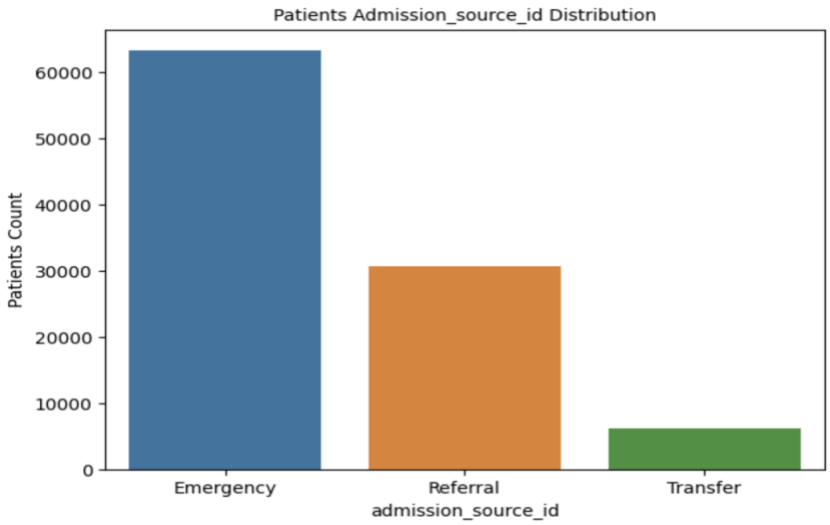
The distribution of patients between those who were readmitted and those who were not readmitted is identical.

**Admission_source_id:**

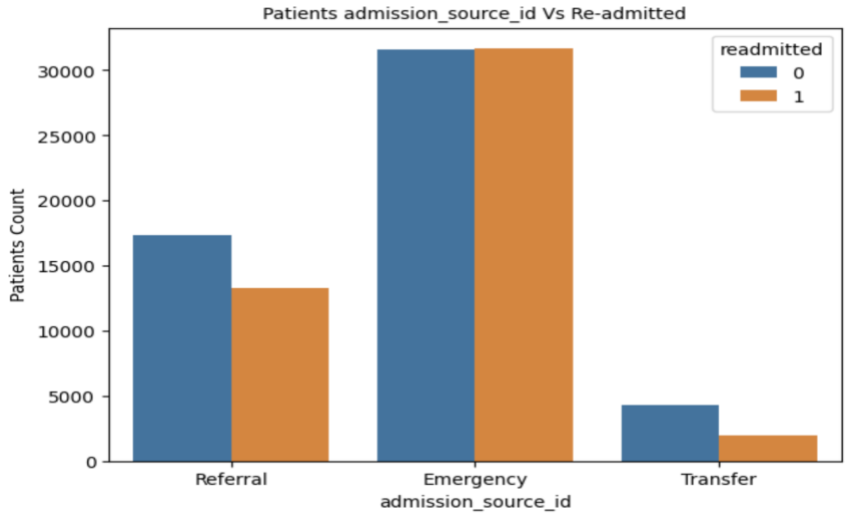
To enhance visualization and comprehension, we have consolidated the categories (25) within the Admission_source_id column, reducing the overall number as illustrated below.

1. Referral: Physician Referral, Clinic Referral, HMO Referral
2. Transfer: Transfer from a hospital, Transfer from a Skilled Nursing Facility (SNF), Transfer from another health care facility, Court/Law Enforcement, Transfer from critial access hospital, Normal Delivery, Premature Delivery, Sick Baby, Extramural Birth, Transfer from hospital inpt/same fac reslt in a sep claim, Born inside this hospital, Born outside this hospital, Transfer from Ambulatory Surgery Center, Transfer from Hospice, Transfer From Another Home Health Agency, Readmission to Same Home Health Agency.
3. Emergency: Emergency Room
4. Null: NULL, Not Available, Not Mapped, Unknown/Invalid.

It has been noticed that there are around 6929 missing values in the specified column. Mode imputation is performed to fill up the null values.



The above bar graph shows the distribution of patients' admission source IDs by emergency, referral, and transfer. The emergency IDs are the most common source of admission, with over 65,000 patients admitted. The referral IDs are the second most common source of admission, with over 30,000 patients admitted. The transfer IDs are the least common source of admission, with just over 5,000 patients admitted.




The distribution between the number of patients who were readmitted and those who were not readmitted is same.

**ICD-9 codes:**

001-139 Infectious And Parasitic Diseases

140-239 Neoplasms

240-279 Endocrine, Nutritional And Metabolic Diseases, And Immunity Disorders

280-289 Diseases Of The Blood And Blood-Forming Organs

290-319 Mental, Behavioral And Neurodevelopmental Disorders

320-389 Diseases Of The Nervous System And Sense Organs

390-459 Diseases Of The Circulatory System

460-519 Diseases Of The Respiratory System

520-579 Diseases Of The Digestive System

580-629 Diseases Of The Genitourinary System

630-679 Complications Of Pregnancy, Childbirth, And The Puerperium

680-709 Diseases Of The Skin And Subcutaneous Tissue

710-739 Diseases Of The Musculoskeletal System And Connective Tissue

740-759 Congenital Anomalies

760-779 Certain Conditions Originating In The Perinatal Period

780-799 Symptoms, Signs, And Ill-Defined Conditions

800-999 Injury And Poisoning

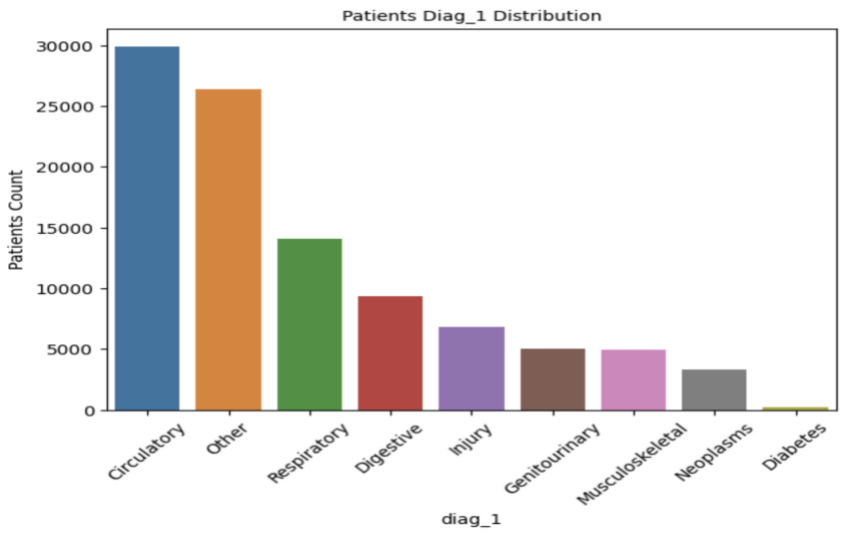
E000-E999 Supplementary Classification Of External Causes Of Injury And Poisoning

In each of the diag columns, there are over 750 unique ICD-9 codes. We categorized these codes based on their respective ICD-9 groups.

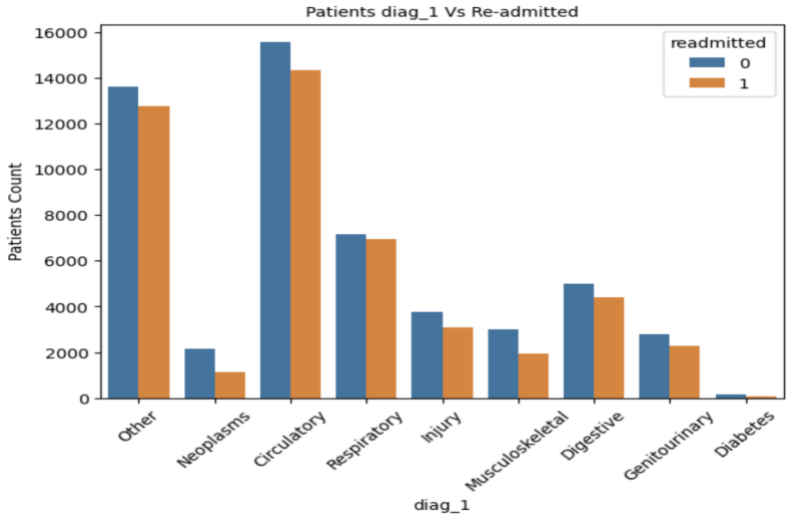
The number of records in diag_1, diag_2, and diag_3 is 714, 746, and 787.

Observing the presence of NaN values, we have chosen to fill these gaps by imputing the values with 'diabetes,' given that the dataset is predominantly centered around this condition.

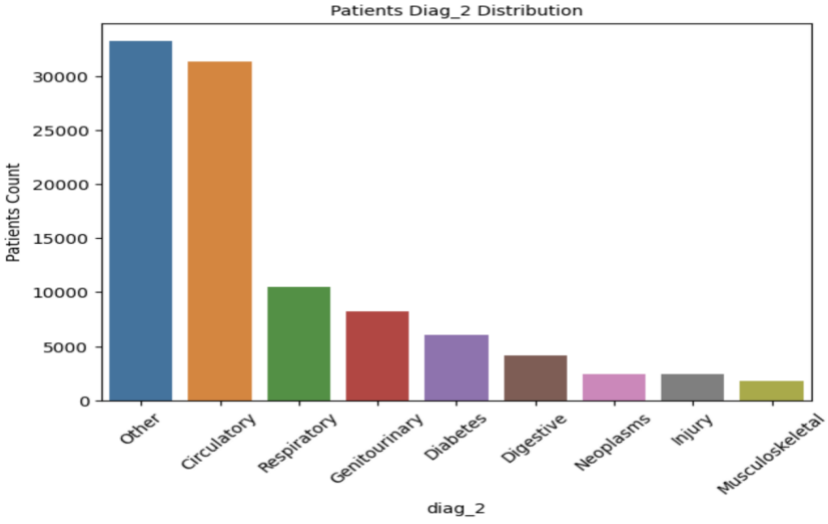
We have taken a numerical input representing a medical category code and categorized it into broader health categories such as Diabetes, Circulatory, Respiratory, Digestive, Injury, Musculoskeletal, Genitourinary, Neoplasms, or Other, based on specified ranges. For example, if the input is 250 or NaN, it's categorized as Diabetes. The function aims to provide a more interpretable classification for medical data.



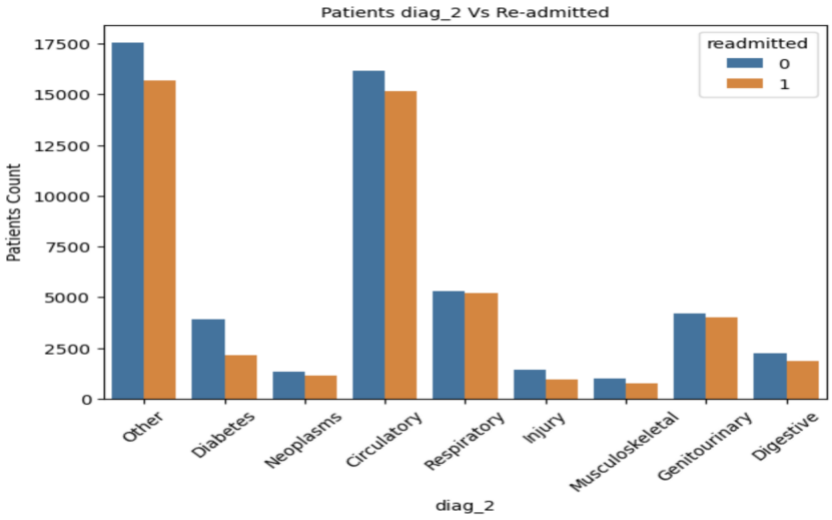
The primary diagnosis prevalent among patients in the dataset is associated with circulatory system diseases, and this correlation is linked to the diabetic condition as well.



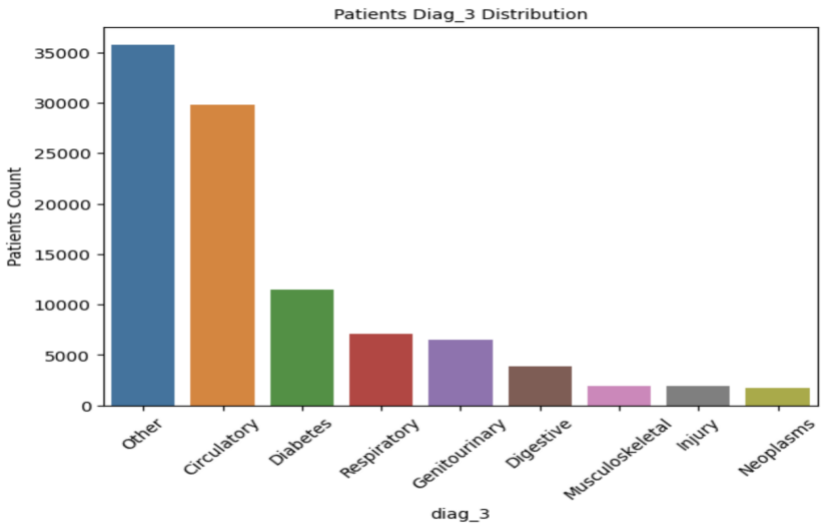

The distribution between the number of patients who were readmitted and those who were not readmitted is same.



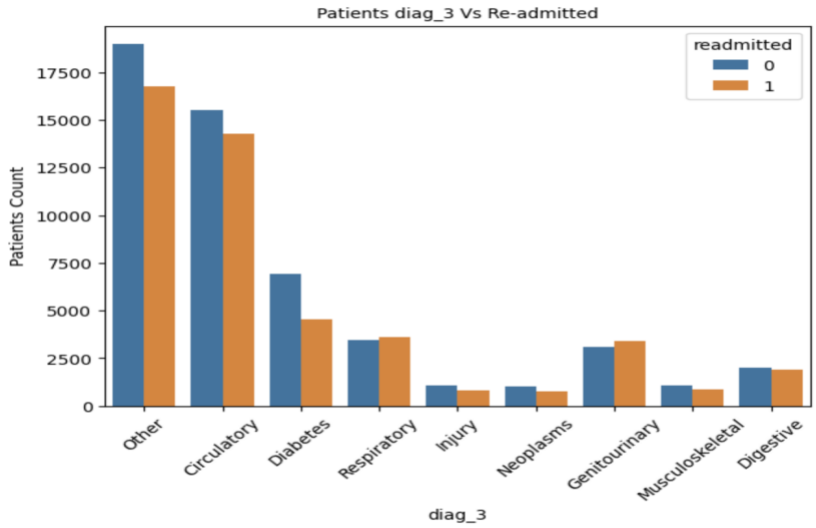

Majority of the patients in the dataset were diagnosed with ‘other’ diseases. Followed by ‘Circulatory’ diseases.




The distribution between the number of patients who were readmitted and those who were not readmitted is same.




Majority of the patients in the dataset were diagnosed with ‘other’ diseases. Followed by ‘Circulatory’ diseases.

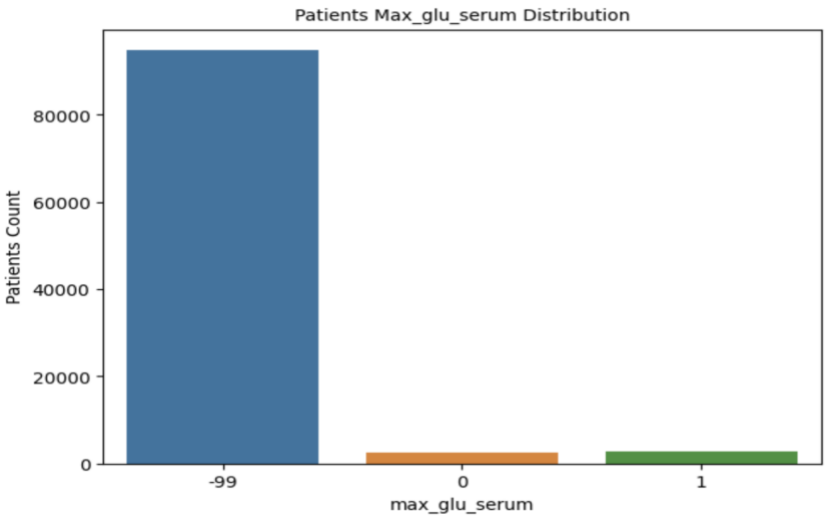



The distribution between the number of patients who were readmitted and those who were not readmitted is same.

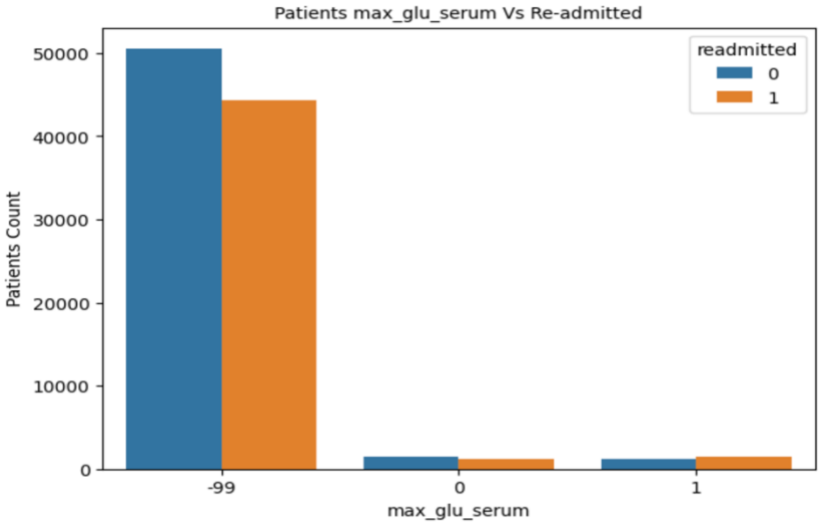
**Max_glu_serum:**

The Max_glu_serum column has indicated the following count of records in the dataset.

None – 94,859; Norm – 2,573; >200 – 1440; >300 – 1211.



Here, we have reassigned values based on the range of results for glucose serum tests, consolidating ">200" and ">300" into category 1, "Norm" into category 0, and "None" into category -99.

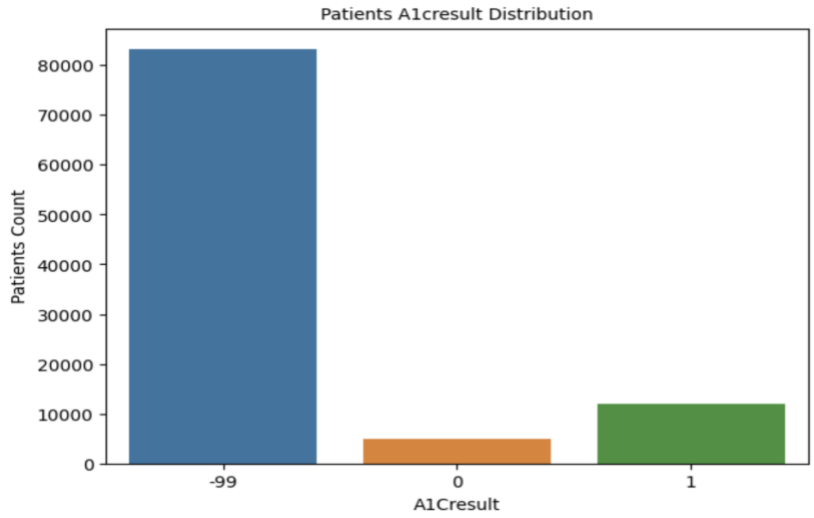



The distribution between the number of patients who were readmitted and those who were not readmitted is same.

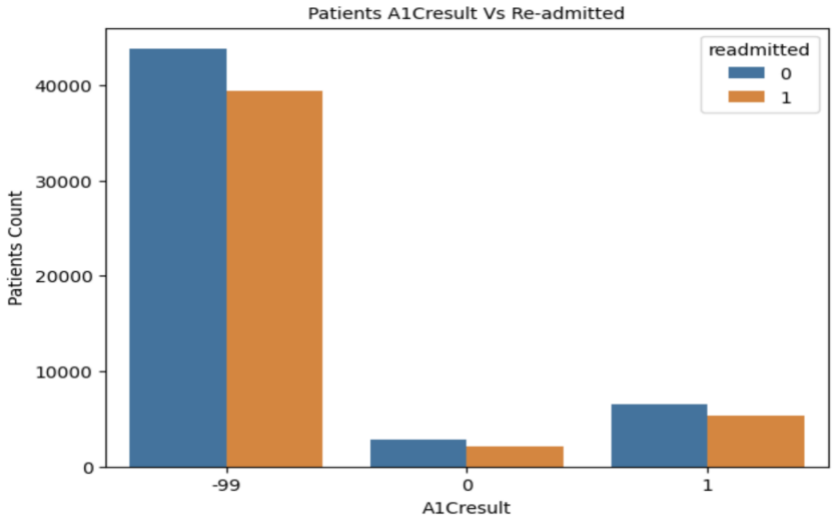
**A1C result:**

The A1C result column has reported the count of records as follows in the dataset.

None – 83213; >8 - 8151; Norm - 4937; >7 – 3782.

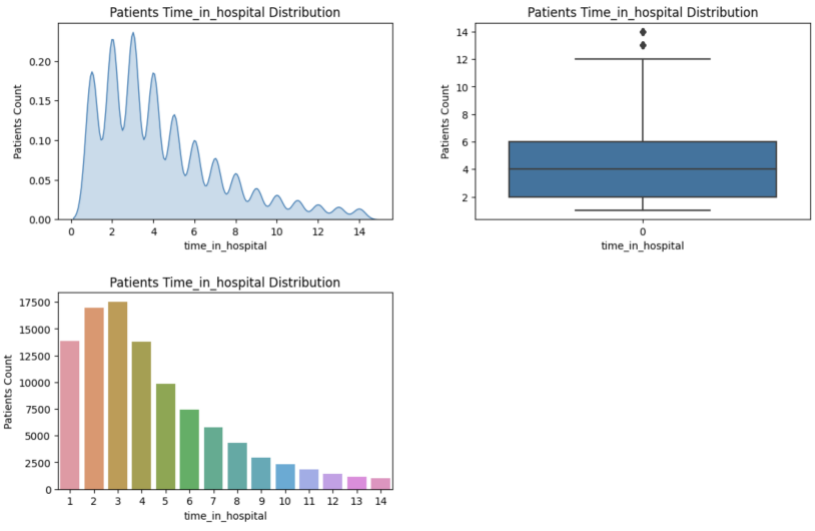


Here, we have reallocated values according to the result range for glucose serum tests, combining ">7" and ">8" into category 1, assigning "Norm" to category 0, and labeling "None" as -99.




The distribution between the number of patients who were readmitted and those who were not readmitted is same.

**Time_in_hospital:**



Central Tendency Shifts:

- The mean (average) will be greater than the median. The presence of some larger values on the right pulls the mean in that direction.
- The median, being the middle value, is less affected by extreme values, so it's a more robust measure of central tendency.

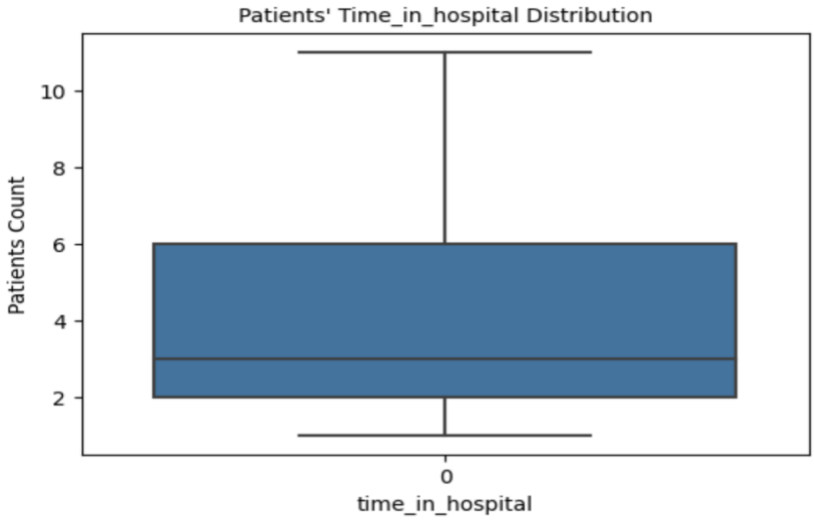
Data Spread:

- The spread or variability of the data tends to be larger on the right side due to those few larger values.

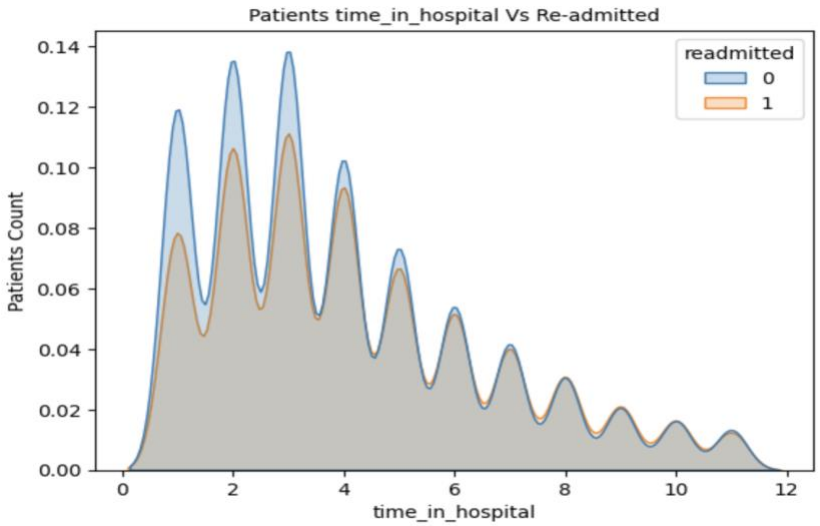
Skewness Measure:

- Skewness is a measure of the asymmetry of the distribution. In a right-skewed distribution, the skewness value is positive.

The column contains outliers, so we have eliminated them utilizing our defined outliers function.



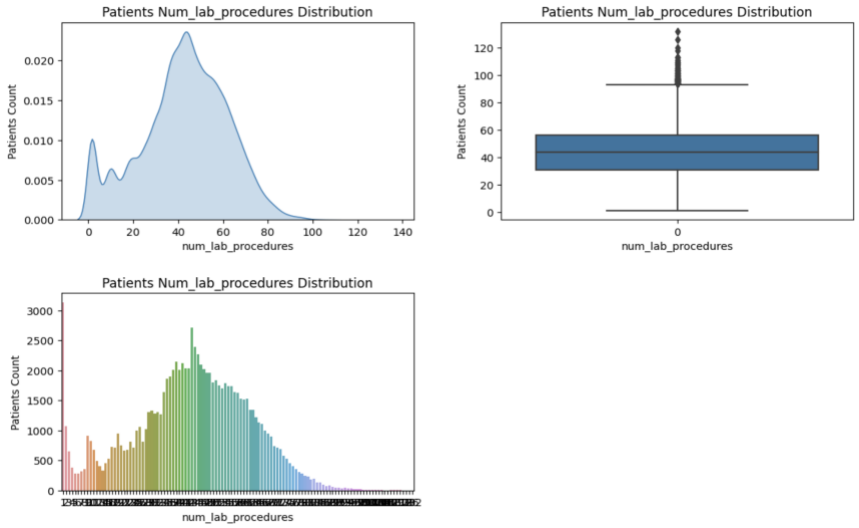

The above boxplot has been represented without any outliers.



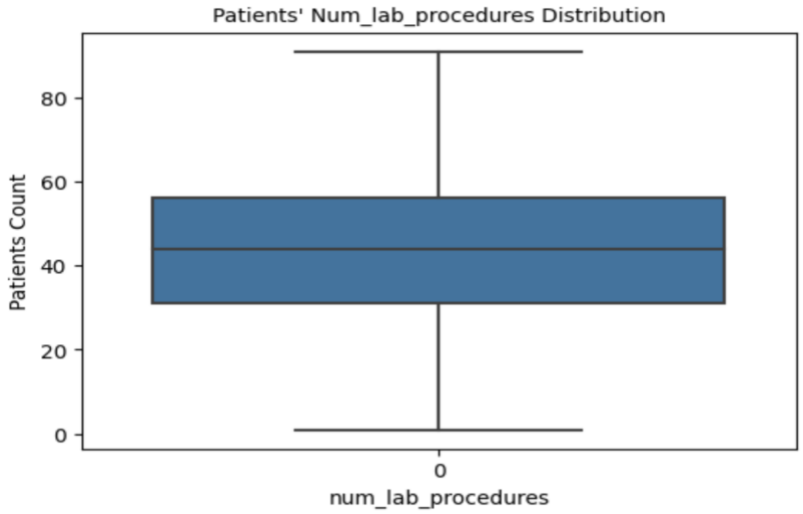

The probability density between the number of patients who were readmitted and those who were not readmitted is the same, it suggests that the likelihood of a patient being readmitted is comparable to the likelihood of a patient not being readmitted. In other words, there is no significant skew or bias towards either outcome in terms of probability.

This could indicate that the factors influencing readmission and non-readmission are relatively balanced or that the distribution of patients across these two categories is even. It's a neutral scenario where, statistically speaking, the chances of readmission are on par with the chances of not being readmitted.

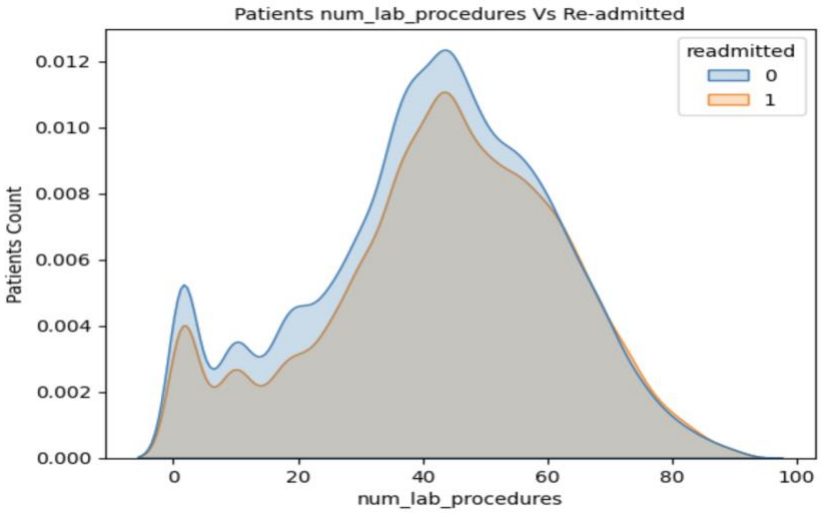
**Num_lab_procedures:**




We've identified outliers in the column and subsequently removed them using our pre-defined function for handling outliers.

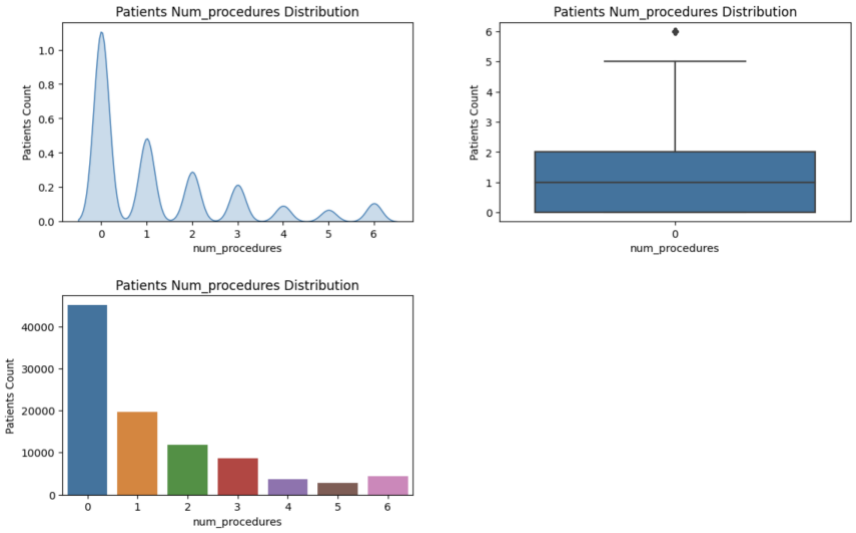



The above boxplot has been represented without any outliers.

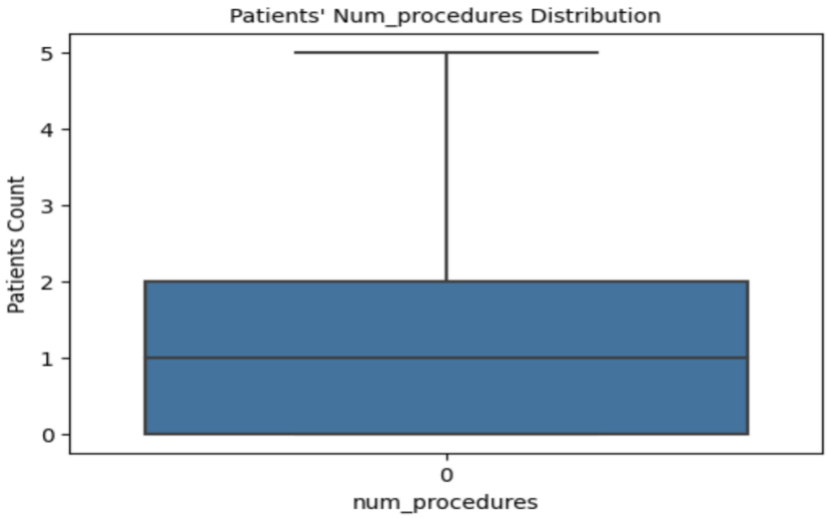



The probability density between the number of patients who were readmitted and those who were not readmitted is same.

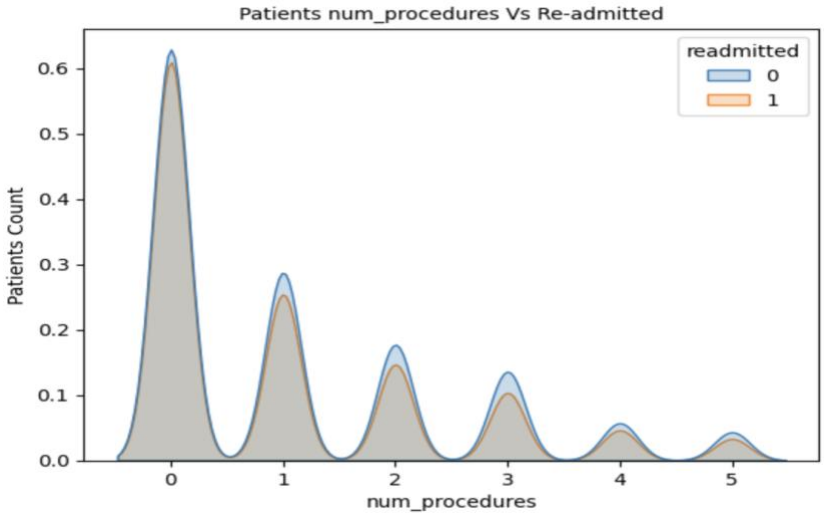
**Num_procedures:**




We detected outliers in the column and proceeded to eliminate them using our predefined function for outlier management.

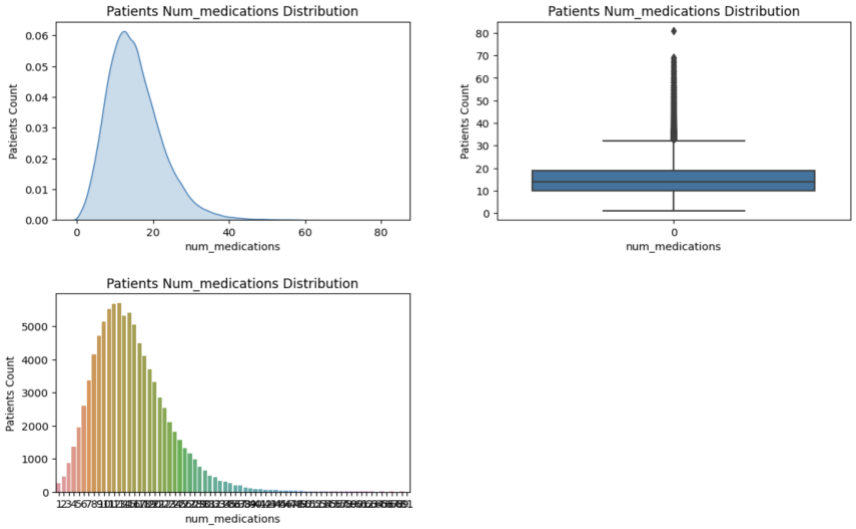



The above boxplot has been represented without any outliers.

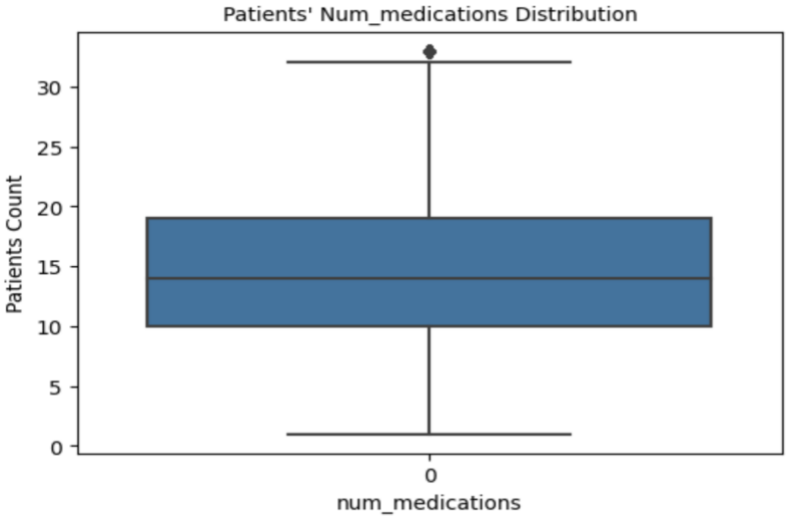



The probability density between the number of patients who were readmitted and those who were not readmitted is same.

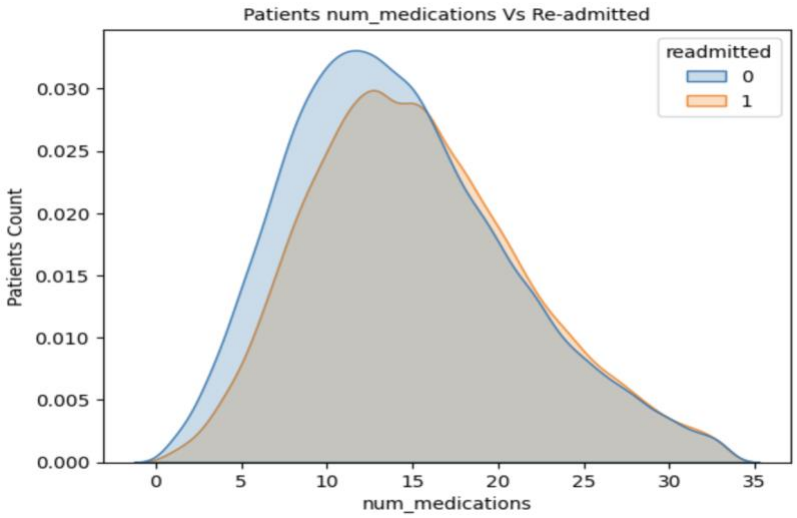
**Num_medications:**




We detected outliers in the column and proceeded to eliminate them using our predefined function for outlier management.

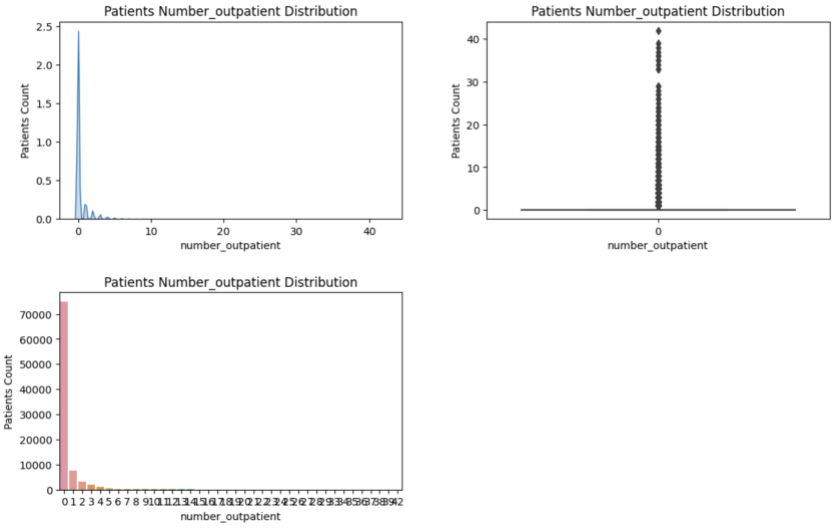



The above boxplot has been represented without any outliers.

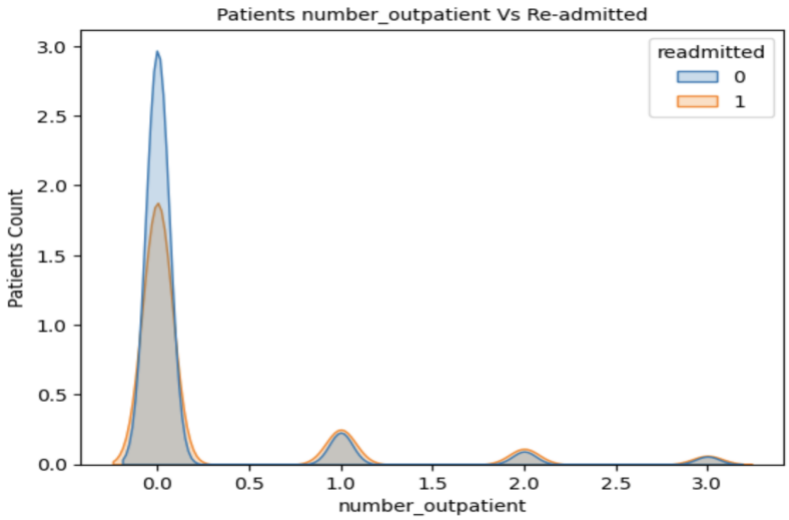



The probability density between the number of patients who were readmitted and those who were not readmitted is same.

**Number_outpatient:**

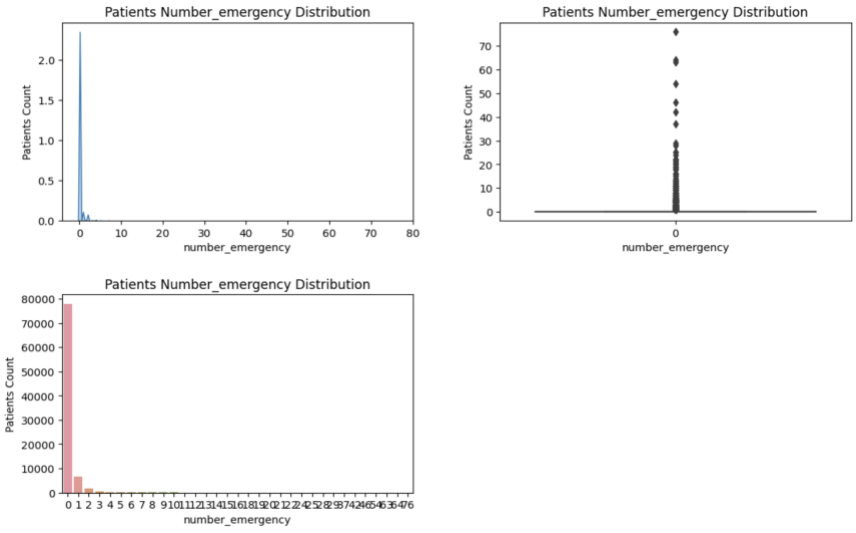



We detected outliers in the column and proceeded to eliminate them using our predefined function for outlier management.

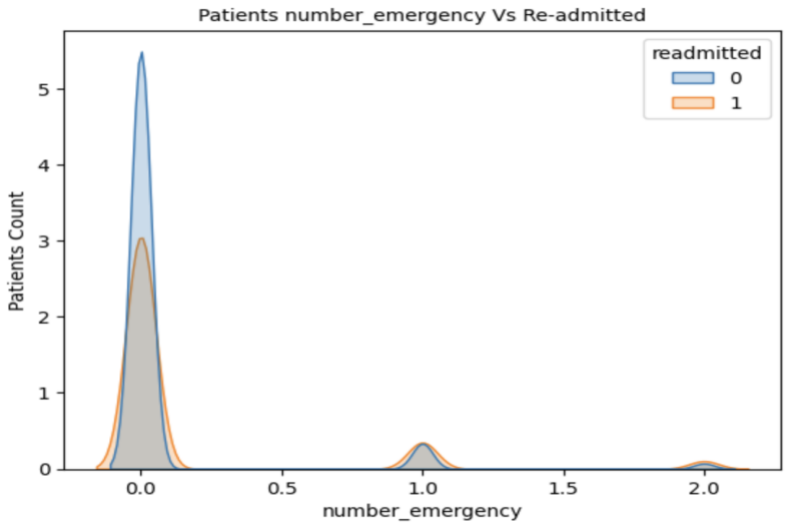



The probability density between the number of patients who were readmitted and those who were not readmitted is same.

**Number_emergency:**

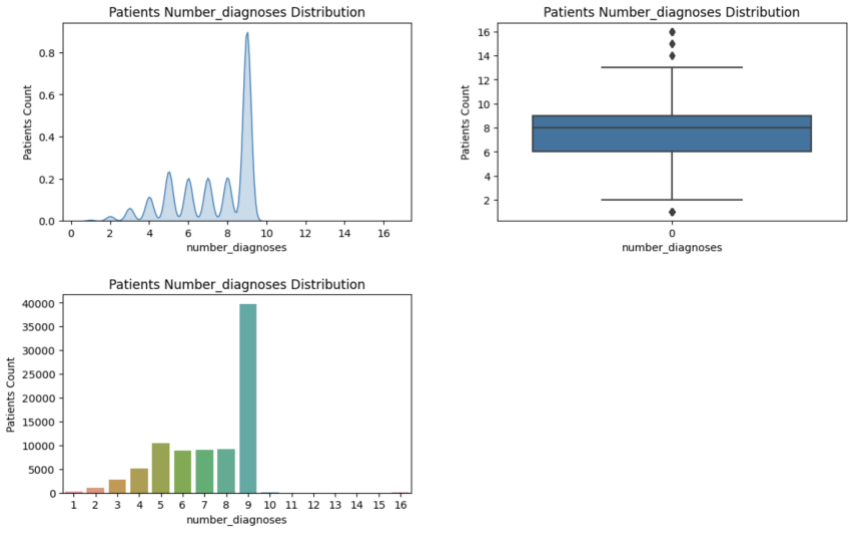



We detected outliers in the column and proceeded to eliminate them using our predefined function for outlier management.

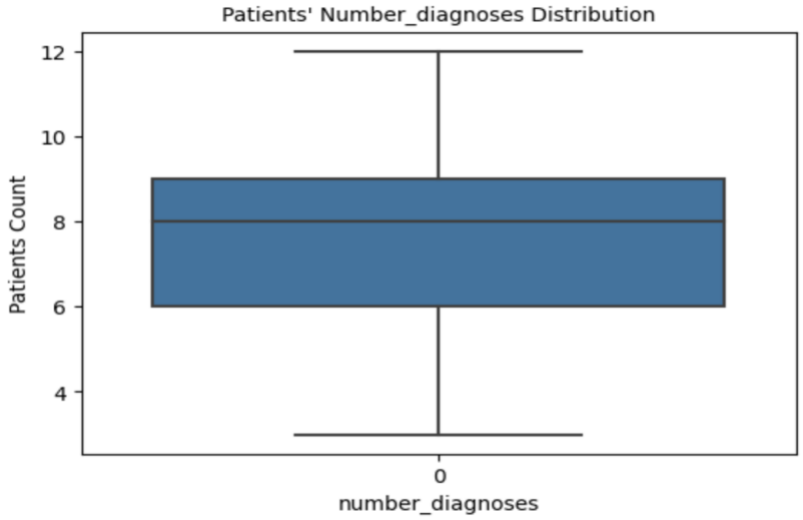



The probability density between the number of patients who were readmitted and those who were not readmitted is same.

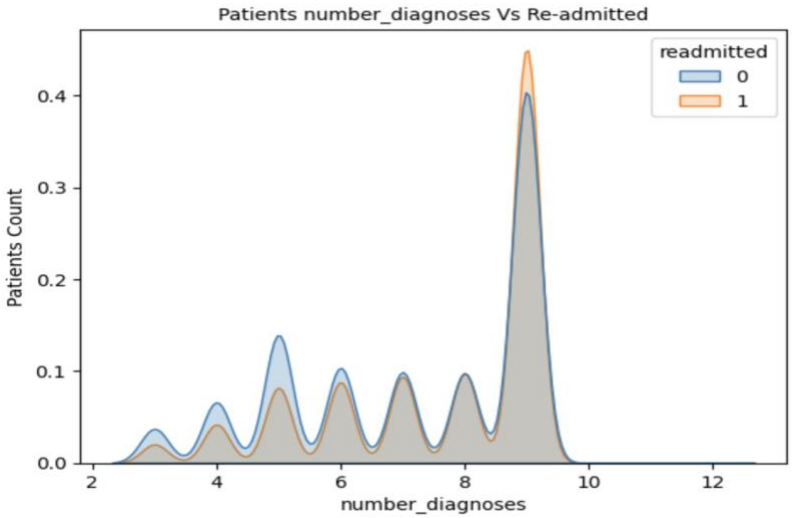
**Number_diagnoses:**




We detected outliers in the column and proceeded to eliminate them using our predefined function for outlier management.




The above boxplot has been represented without any outliers.

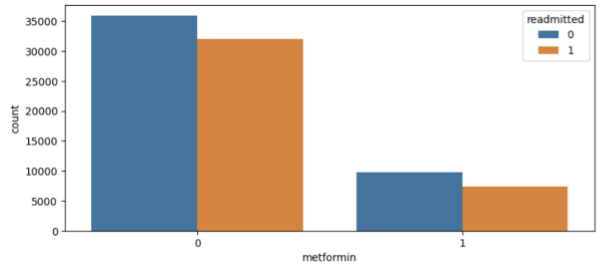


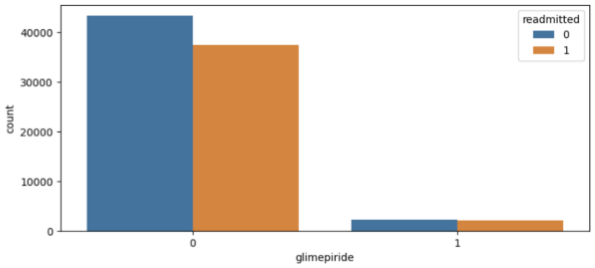

The probability density between the number of patients who were readmitted and those who were not readmitted is same.

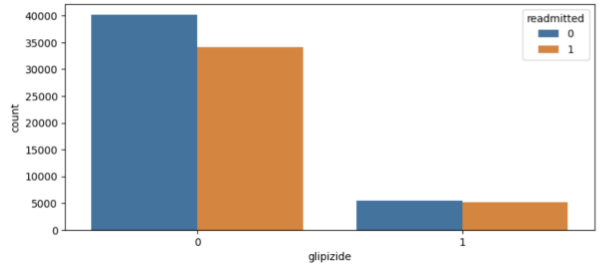
**Diabetes Medications:**

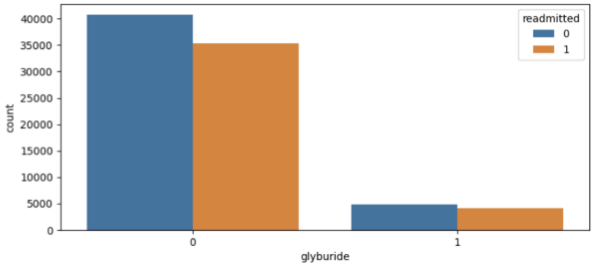
This feature indicates whether the drug was prescribed or if there was a dosage change. To simplify, we have categorized the values of the columns (diabetes medications) into binary form, assigning "No" to 0 and grouping "Steady," "Up," "Down," "Ch," and "Yes" into the category 1.

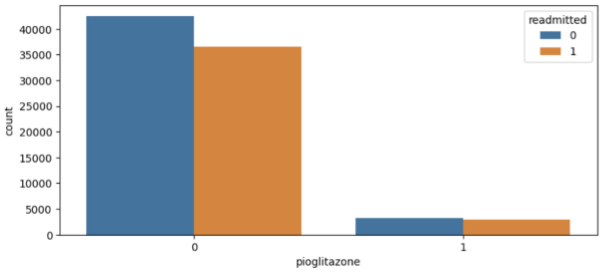
Upon observation, we noticed duplications in the diabetic medication columns, and consequently, we have removed those redundant columns.

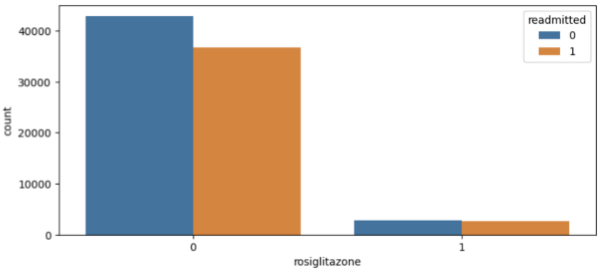


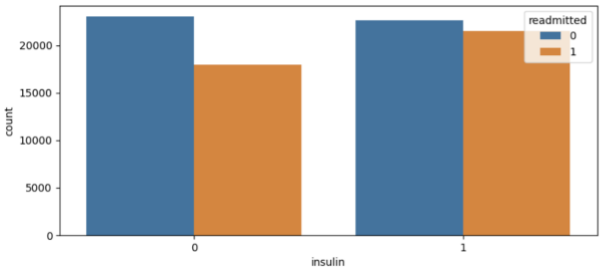



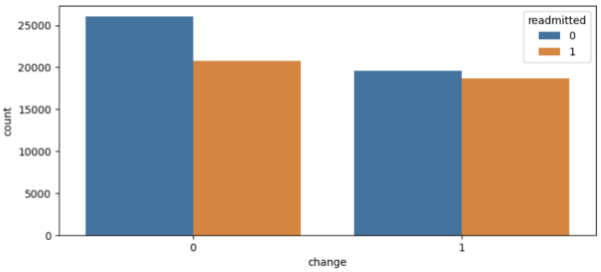



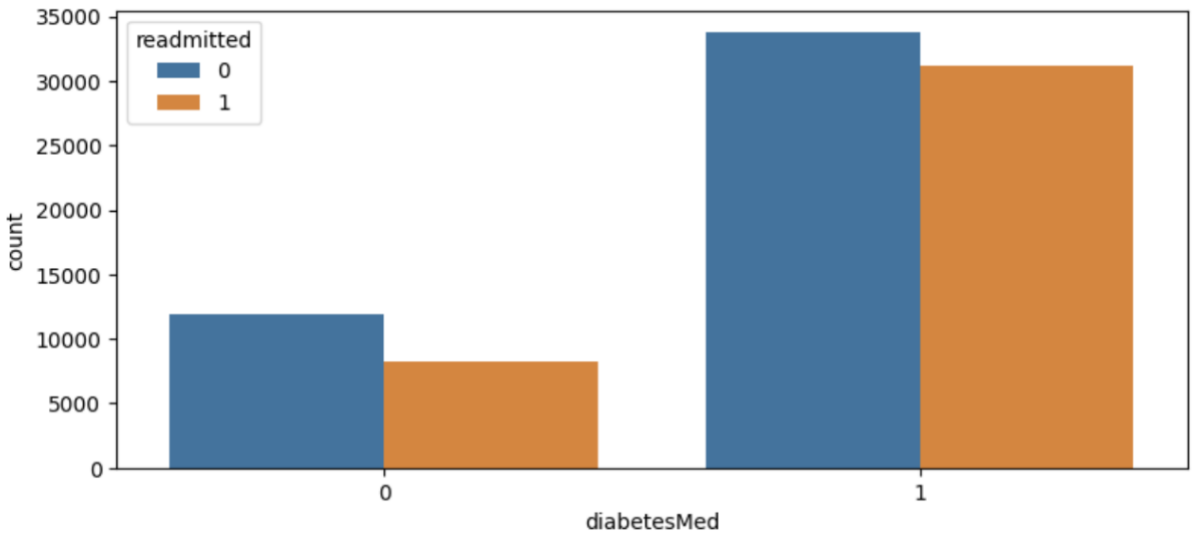






The distribution between the number of patients who were readmitted and those who were not readmitted is same.

Additional data mining tasks and data exploration have been performed below:

Diag_1

```
Circulatory         23730
Other               22470
Respiratory         12695
Digestive            8376
Injury               5999
Genitourinary        4537
Musculoskeletal      4299
Neoplasms            2746
Diabetes              188
Name: diag_1, dtype: int64
```

Diag_2

```
Other               28769
Circulatory         25922
Respiratory          9005
Genitourinary        7064
Diabetes             5141
Digestive            3617
Neoplasms            2145
Injury               1856
Musculoskeletal      1521
Name: diag_2, dtype: int64
```

Diag_3

```
Other               29991
Circulatory         25198
Diabetes            10369
Respiratory          5919
Genitourinary        5493
Digestive            3403
Musculoskeletal      1653
Neoplasms            1541
Injury               1473
Name: diag_3, dtype: int64
```

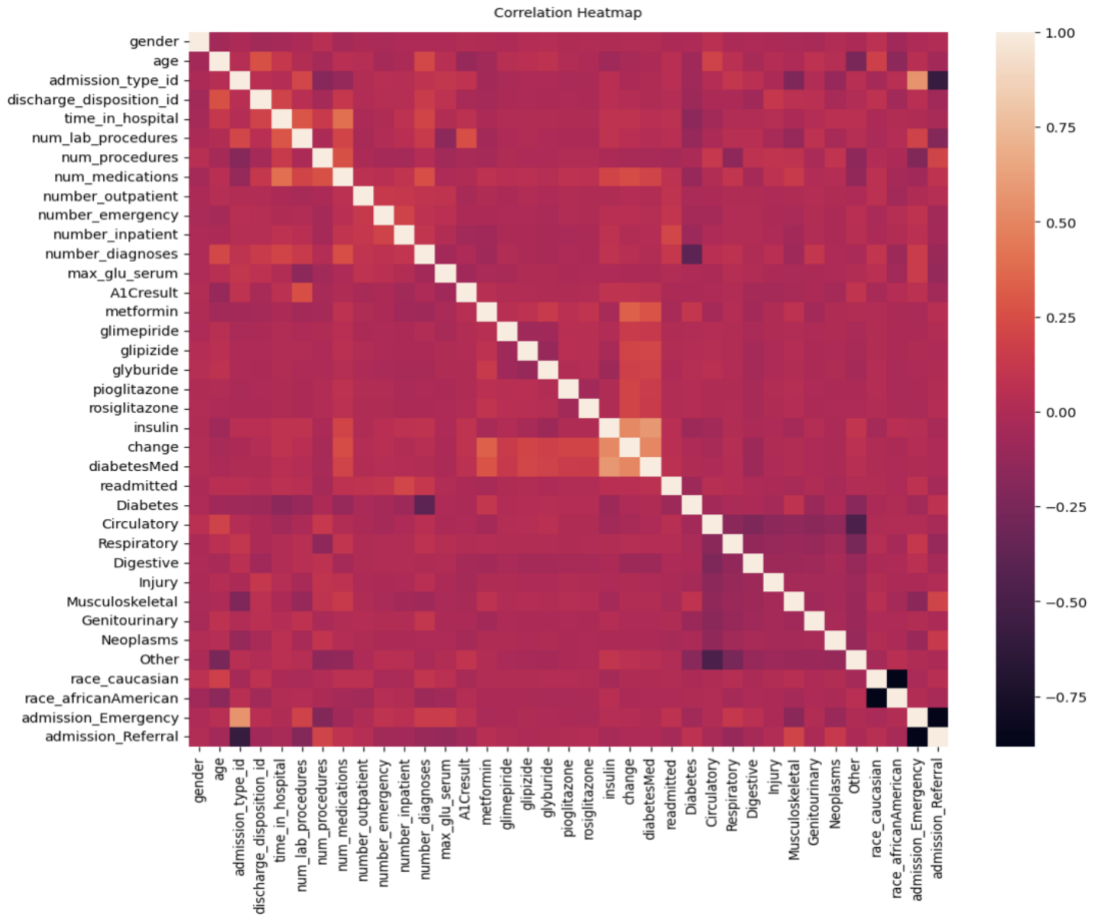
Transforming the dataset involves segregating the three diagnosis columns, namely diag_1, diag_2, and diag_3, into distinct sets of nine columns, each representing a specific diagnosis. In this process, diag_1 is designated to the third column, diag_2 to the second column, and diag_3 to the first column.

We have created two new binary columns, "race_caucasian" and "race_africanAmerican," based on the values in the original "race" column. If an individual is of Caucasian race, "race_caucasian" is set to 1; if African American, "race_africanAmerican" is set to 1. The original "race" column is

then dropped from the DataFrame, likely for the purpose of preparing the data for machine learning models that require numerical input.

Similarly, we have created two binary columns ("admission_Emergency" and "admission_Referral") based on conditions in the original "admission_source_id" column. Values are set to 1 if the conditions match, 0 otherwise. The original column is then dropped, likely for data preparation in machine learning or analysis.

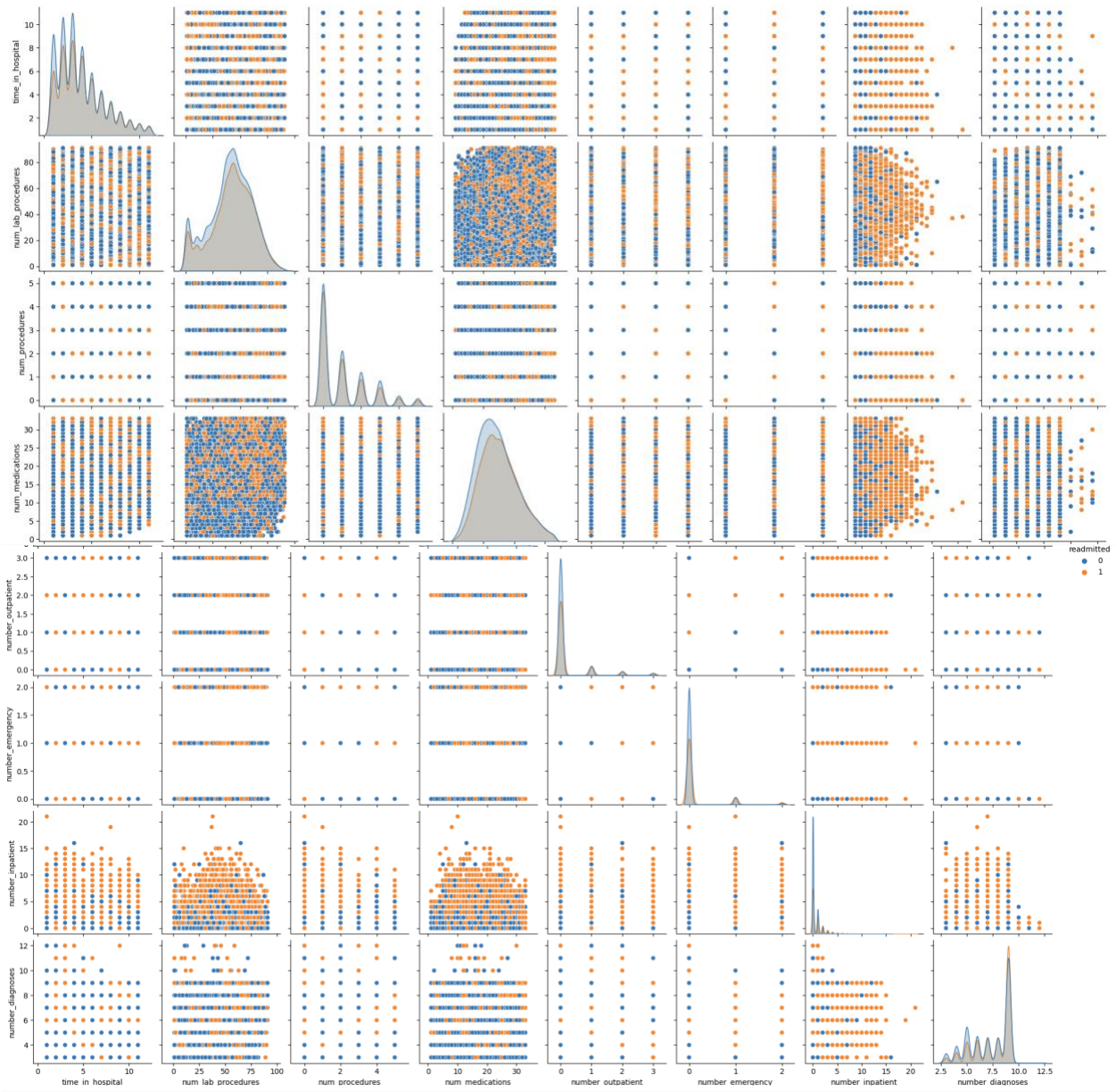
We have employed a correlation heatmap to visually represent the correlation coefficients between variables in our dataset. The colors indicate the strength and direction of these correlations, making it easier to identify patterns and relationships. It's a powerful tool for exploring and understanding the interdependencies within a dataset, providing a quick and intuitive overview of the data's structure.



Correlation values, which fall on a scale from -1 to +1, serve as indicators of the relationship between two variables. When the value approaches zero, it suggests a lack of a clear linear trend between the variables. As the value nears +1, it signals a strong positive correlation, indicating that the variables tend to move in the same direction. Conversely, when the value approaches -1, it points to a robust negative correlation, signifying that the variables typically move in opposite directions.

The correlation heatmap reveals diagonal values of one, signaling a perfect correlation. In the data context, this implies that each variable is perfectly correlated with itself, indicating a strong internal consistency within individual variables. This observation is crucial for understanding how each variable relates to its own values, providing insights into the inherent structure and coherence of the dataset. There are some positive values between 0 and 1 for some pair of variables which indicates some correlation; however, it isn't a strong one. In addition, there are some negative values too.

Pairplot, short for pairwise plot, is a useful tool in Exploratory Data Analysis (EDA) because it allows us to visualize the relationships between multiple variables in a dataset. The diagonal histograms in a pairplot reveal the distribution of individual variables, providing insights into their data patterns. Meanwhile, the scatter plots in the upper and lower triangles showcase the connections, or lack thereof, between pairs of variables, aiding in the exploration of their relationships.
Here, we have implemented a pair plot which includes numerical variables as follows:
time_in_hospital, num_lab_procedures, num_procedures, num_medications, number_outpatient, number_emergency, number_inpatient, number_diagnoses.




Based on the above scatterplots, it can be concluded that there is a lack of correlation between the respective numerical variables. The visual representations strongly suggest that changes in one variable do not correspond with predictable changes in the other, indicating an absence of a discernible relationship between them.

It has been observed from the diagonal plots that most of the variables are right skewed, whereas number_diagnoses is left-skewed. In this case, we should consider transforming the data to make the distribution more symmetric and improve the performance of certain statistical analyses.

Following the outlined procedures, our DataFrame has undergone modifications, resulting in an enriched dataset now comprising 85,040 rows and 35 columns.

Now, we've executed the following steps to ensure the consistency of our dataset throughout the entire process.

To address any class imbalance in the dataset, the majority and minority classes were separated. The majority class was down sampled to match the number of instances in the minority class. The resulting balanced dataset was created by combining the down sampled majority class with the original minority class. To ensure randomness, the combined dataset was shuffled. This approach aims to enhance the performance and fairness of machine learning models trained on the data.
Selected numerical columns—'age', 'time_in_hospital', 'num_lab_procedures', 'num_procedures', 'num_medications', 'number_inpatient', and 'number_diagnoses'—have been converted to the 'category' data type. This conversion is applied to columns that are deemed more categorical in nature, while leaving other numerical columns unchanged. The objective is to optimize the representation of certain features in the dataset for improved model performance.

Now, we have 78,788 rows and 35 columns.

Our DataFrame is split into features (X) and the target variable (y). X contains all the columns except 'readmitted', while y specifically contains the 'readmitted' column. This is a common setup for supervised machine learning, where we use features (X) to predict the target variable (y).

It has been observed that the target variable ‘Readmitted’ has a balanced set of values i.e., 0 – 39394; 1 – 39394.

We have split our data into training, validation, and testing sets. The training set typically comprises a larger portion (70% in this case), and the validation and testing set make up the remaining 30% (15% each). This split is crucial for evaluating our machine learning models' performance.

Numerical features in the training set (X_train) have been standardized. This process involves transforming the data so that the features have a mean of 0 and a standard deviation of 1. The same scaler is then applied to the validation set (X_valid) and the test set (X_test) to maintain consistency in the scaling across different subsets of the data. Standardizing numerical features is a common practice to enhance the performance and convergence of various machine learning algorithms.

**Data Mining Models and Performance Evaluation:**

**Logistic Regression**

Logistic regression is a preferred algorithm for binary classification due to its simplicity, interpretability, and efficiency. Its linear decision boundary and probabilistic output facilitate easy understanding and assessment of feature contributions. The model is less prone to overfitting, making it suitable for situations with a limited number of features. Logistic regression is efficient to train, especially with many observations, and offers regularization options for controlling model complexity. Its applicability spans diverse domains, making it a reliable and widely used choice in various industries, providing a balance between simplicity and effectiveness.

Model Initialization and Training:

- A logistic regression model is initialized with a random seed for reproducibility.
- The model is trained using the training data (X_train and y_train).

Validation Set Predictions and Evaluation:

- Predictions are made on the validation set (X_valid).
- The model's performance is evaluated on the validation set using accuracy and a classification report.

Test Set Predictions and Evaluation:

- Predictions are made on the test set (X_test).
- The model's performance is evaluated on the test set using accuracy and a classification report.

In a clinical dataset, accuracy and F1 score are crucial metrics for different reasons. Accuracy measures the overall correctness of your model, which is important in any scenario. However, in clinical datasets, where imbalances in class distribution are common, accuracy alone can be misleading.

This is where F1 score comes into play. F1 score considers both precision and recall, providing a balance between false positives and false negatives. In a clinical setting, false positives and false negatives can have serious consequences. For example, misdiagnosing a healthy patient as diseased (false positive) or vice versa (false negative) can lead to inappropriate treatments or missed interventions.

By focusing on both accuracy and F1 score, you aim for a model that not only makes correct overall predictions but also minimizes the risk of making critical errors, making it more reliable and safer for clinical use.

**Accuracy and Classification Report:**

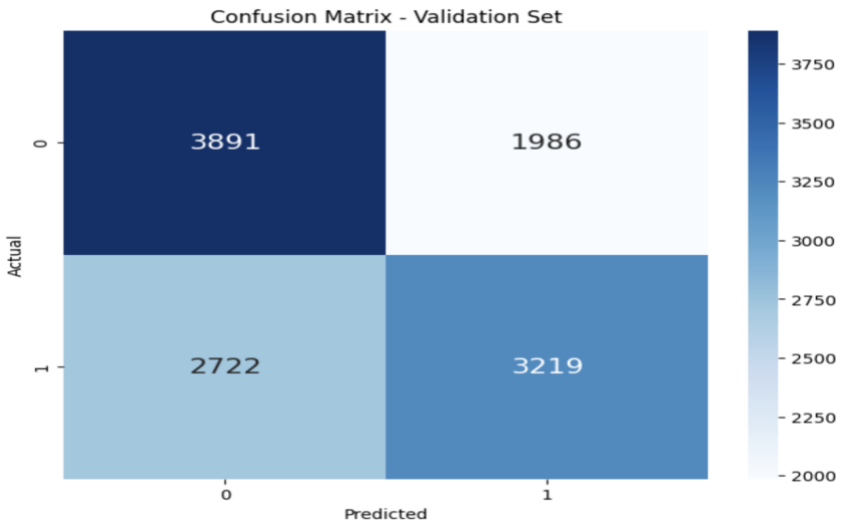
```
Validation Accuracy: 0.6016
Classification Report:
              precision    recall  f1-score   support

           0       0.59      0.66      0.62      5877
           1       0.62      0.54      0.58      5941

    accuracy                           0.60     11818
   macro avg       0.60      0.60      0.60     11818
weighted avg       0.60      0.60      0.60     11818


Test Set Performance:
Test Accuracy: 0.6098
Classification Report:
              precision    recall  f1-score   support

           0       0.60      0.68      0.64      5917
           1       0.63      0.54      0.58      5902

    accuracy                           0.61     11819
   macro avg       0.61      0.61      0.61     11819
weighted avg       0.61      0.61      0.61     11819
```

From the above values, we can infer that the validation set accuracy is ‘0.6016’, whereas test set accuracy is ‘0.6098’.

F1 = 2 * ((Precision * Recall)/ (Precision + Recall))

The F1-score ranges from 0 to 1, where 1 indicates perfect precision and recall, and 0 indicates the worst possible performance.
The f1-scores can be seen in the above classification report.

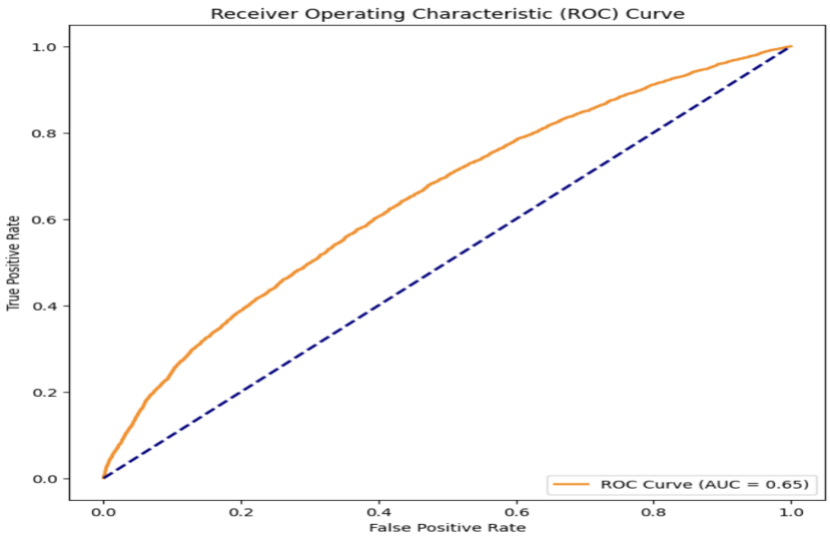
**Confusion matrix:**




True Negative instances: 3891
False Positive instances: 1986
False Negative instances: 2722
True Positive instances: 3219

**ROC Curve:**

ROC curve shows the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity) across different threshold values.
In simpler terms, the ROC curve helps us evaluate how well our model can distinguish between classes by plotting the relationship between true positive rate and false positive rate. The area under the ROC curve (AUC-ROC) is often used as a single metric to summarize the performance of the model. A higher AUC-ROC value indicates better discriminative ability.

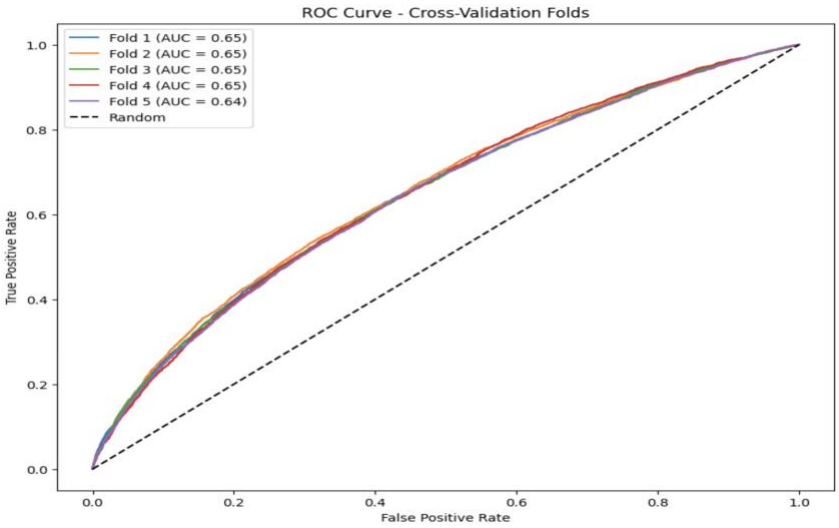



From the above graph, we can see that the AUC-ROC value is equal to 0.65.

**K – Fold Cross Validation for Logistic Regression:**

K-Fold Cross Validation is like a thorough check-up for our Logistic Regression model. Instead of just doing it once, K-Fold Cross Validation divides our data into K subsets (or "folds"). The model is then trained K times, each time using K-1 folds for training and the remaining fold for testing. This process repeats until each fold has been used as a test set.

By doing this, we get a more reliable estimate of our model's performance, as it gets evaluated on different portions of the data. It's like making sure our model is robust and not just good at handling one specific slice of your data. This helps us gauge how well our Logistic Regression model is likely to perform on new, unseen data.




After implementing K-Fold Cross validation, there is a slight improvement in accuracy and F1 score for the model with 65.2% and 71%. However, there isn't much improvement in the AUC-ROC value i.e., 0.65 in Fold 1, Fold 2, Fold 3, Fold 4, which eventually reduced to 0.64 in Fold 5.

**Random Forest:**

Random Forest is an ensemble learning method used in machine learning for both classification and regression tasks. It constructs a multitude of decision trees during training and outputs the mode (for classification) or the average prediction (for regression) of the individual trees. The randomness is introduced by training each tree on a random subset of the training data and by considering only a random subset of features at each split. This approach enhances generalization performance, mitigates overfitting, and provides robust predictions.

Model Initialization and Training:

- Created a Random Forest model with a random seed for reproducibility.
- Trained the model using the training set (X_train, y_train).

Validation Set Evaluation:

- Made predictions on the validation set (X_valid).
- Calculated and printed the accuracy and classification report for evaluating model performance on the validation set.

Test Set Evaluation:

- Made predictions on the test set (X_test).
- Calculated and printed the accuracy and classification report for evaluating model performance on the test set.

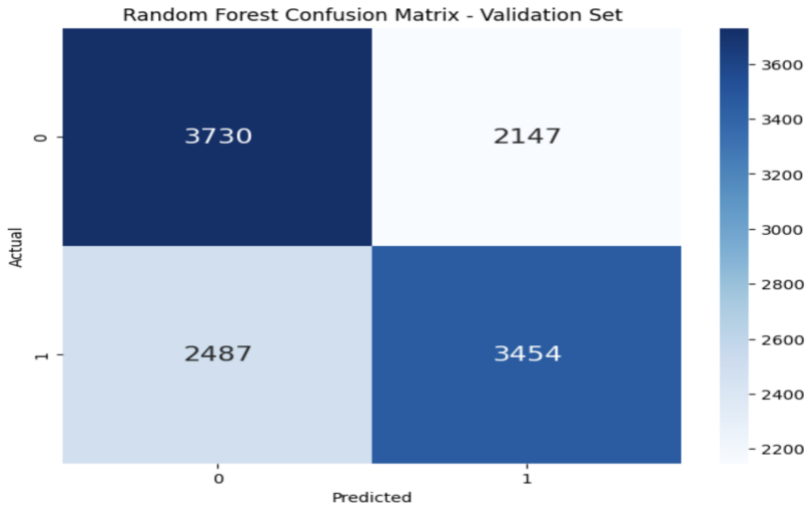
**Accuracy and Classification Report:**

```
Random Forest Validation Accuracy: 0.6079
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.60      0.63      0.62      5877
           1       0.62      0.58      0.60      5941

    accuracy                           0.61     11818
   macro avg       0.61      0.61      0.61     11818
weighted avg       0.61      0.61      0.61     11818


Random Forest Test Set Performance:
Random Forest Test Accuracy: 0.6121
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.61      0.64      0.62      5917
           1       0.62      0.58      0.60      5902

    accuracy                           0.61     11819
   macro avg       0.61      0.61      0.61     11819
weighted avg       0.61      0.61      0.61     11819
```

From the above values, we can infer that the validation set accuracy is ‘0.6079’, whereas test set accuracy is ‘0.6121’.

The f1-scores can be seen in the above classification report.

**Confusion matrix:**

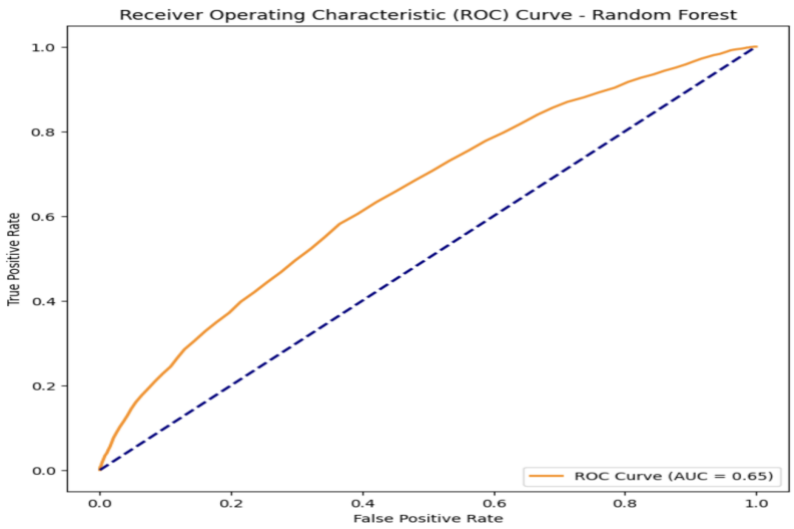


True Negative instances: 3730

False Positive instances: 2147

False Negative instances: 2487

True Positive instances: 3454

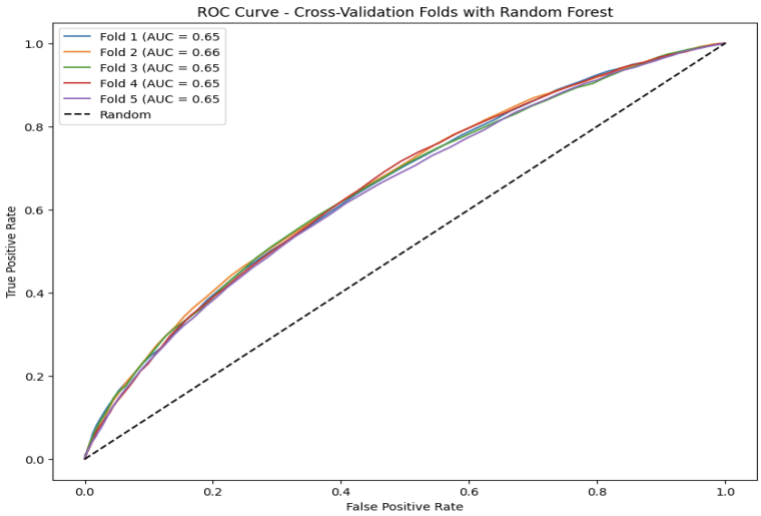



From the above graph, we can see that the AUC-ROC value is equal to 0.65. As compared to Logistic Regression model, there isn't much improvement in the AUC-ROC value.

**K – Fold Cross Validation for Random Forest:**

K-Fold Cross Validation is a resampling technique used to assess the performance of a machine learning model, particularly Random Forest in this context. The dataset is partitioned into K subsets, or folds. The model is then trained K times, each time using K-1 folds for training and the remaining fold for validation. This process is repeated until each fold has been utilized as a validation set.

The purpose of K-Fold Cross Validation is to provide a robust evaluation of the model's performance by exposing it to different subsets of the data. This helps to mitigate the impact of data variability and ensures a more reliable estimate of the model's generalization capabilities. It serves as a valuable technique for assessing how well the Random Forest model can handle diverse data scenarios, enhancing confidence in its ability to generalize effectively to new, unseen data.




From the above cure, we can say that the AUC-ROC curve values remain constant i.e., 0.65 using all the folds.

**Support Vector Machine:**

Support Vector Machine (SVM) is a supervised machine learning algorithm designed for classification and regression tasks. It seeks to find the optimal hyperplane in a high-dimensional space that maximizes the margin between different classes, with support vectors playing a key role. SVM can handle non-linear relationships by utilizing kernel functions to map data into a higher-dimensional space. Overall, it aims to achieve the best separation between classes by maximizing the margin and is effective for a variety of complex decision-making scenarios.

Initialization and Training:

- SVM model instantiated using scikit-learn's SVC.
- Training conducted on the provided training set (X_train, y_train).

Validation Set Evaluation:

- Predictions made on the validation set (X_valid).
- Performance assessed using accuracy and a classification report.
- Insights gained into the SVM model's classification effectiveness.

Test Set Performance:

- Applied the trained SVM model to the test set (X_test).

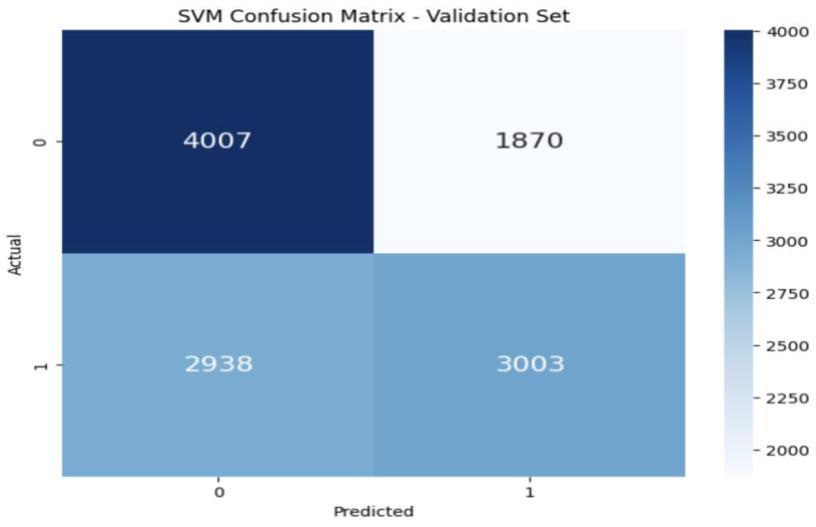
- Evaluated performance using accuracy and a classification report.

**Accuracy and Classification Report:**

```
SVM Validation Accuracy: 0.5932
SVM Classification Report:
              precision    recall  f1-score   support

           0       0.58      0.68      0.63      5877
           1       0.62      0.51      0.56      5941

    accuracy                           0.59     11818
   macro avg       0.60      0.59      0.59     11818
weighted avg       0.60      0.59      0.59     11818


SVM Test Set Performance:
SVM Test Accuracy: 0.5978
SVM Classification Report:
              precision    recall  f1-score   support

           0       0.58      0.70      0.63      5917
           1       0.62      0.50      0.55      5902

    accuracy                           0.60     11819
   macro avg       0.60      0.60      0.59     11819
weighted avg       0.60      0.60      0.59     11819
```

From the above values, we can infer that the validation set accuracy is ‘0.5932’, whereas test set accuracy is ‘0.5978’. The f1-scores can be seen in the above classification report.

**Confusion Matrix:**



True Negative instances: 4007

False Positive instances: 1870

False Negative instances: 2938

True Positive instances: 3003

**Neural Networks:**

Neural networks are favored for classification tasks due to their capacity to model complex, non-linear relationships in data using hidden layers and activation functions. They excel in automatically learning relevant features from raw input data, adapting to diverse data types, and scaling effectively for large and high-dimensional datasets. The end-to-end learning capability eliminates the need for manual feature engineering, while regularization techniques prevent overfitting. Additionally, transfer learning with pre-trained models enhances performance when labeled data is limited. Considerations include data availability, computational resources, and interpretability.

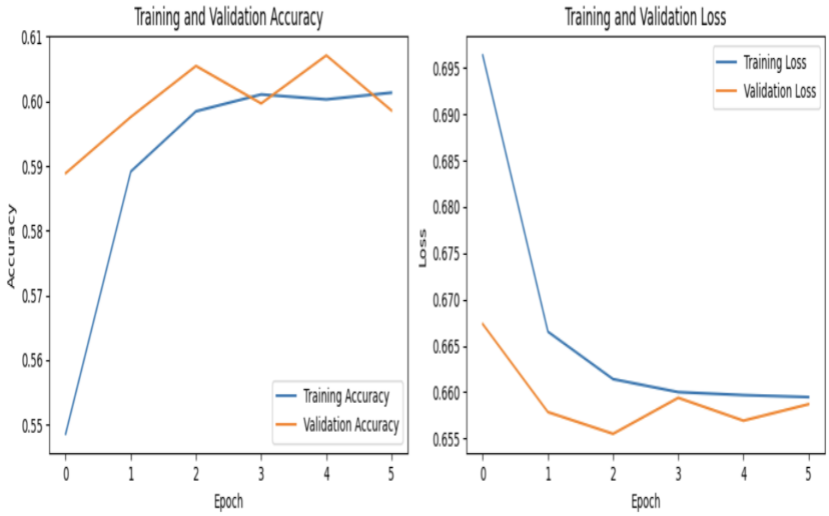
**Model 1:**

Dense layers = 4, epochs = 30, batch size = 512

Output for the Model 1 NN:

```
Epoch 1/30
87/87 [==============================] - 9s 7ms/step - loss: 0.6964 - accuracy: 0.5486 - val_loss: 0.6673 - val_accuracy: 0.5889
Epoch 2/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6665 - accuracy: 0.5891 - val_loss: 0.6578 - val_accuracy: 0.5976
Epoch 3/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6614 - accuracy: 0.5985 - val_loss: 0.6555 - val_accuracy: 0.6055
Epoch 4/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6600 - accuracy: 0.6011 - val_loss: 0.6594 - val_accuracy: 0.5997
Epoch 5/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6597 - accuracy: 0.6003 - val_loss: 0.6569 - val_accuracy: 0.6071
Epoch 6/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6595 - accuracy: 0.6013 - val_loss: 0.6587 - val_accuracy: 0.5986
370/370 [==============================] - 1s 2ms/step
              precision    recall  f1-score   support

           0       0.58      0.72      0.64      5917
           1       0.63      0.49      0.55      5902

    accuracy                           0.60     11819
   macro avg       0.61      0.60      0.60     11819
weighted avg       0.61      0.60      0.60     11819
```

The training loss reflects the error or discrepancy between the predicted outcomes and actual values during the model's learning phase on the training dataset. The objective is to minimize this loss, ensuring the model effectively captures underlying patterns in the training data. On the other hand, the validation loss measures the model's performance on a separate dataset that it has not encountered during training. This loss serves as an indicator of the model's ability to generalize to new, unseen data. Monitoring both training and validation loss is crucial for assessing the model's learning process and avoiding overfitting, where the model becomes too specific to the training data and performs poorly on new data. Striking a balance between minimizing training loss and preventing an increase in validation loss is essential for building a robust and generalizable neural network model.



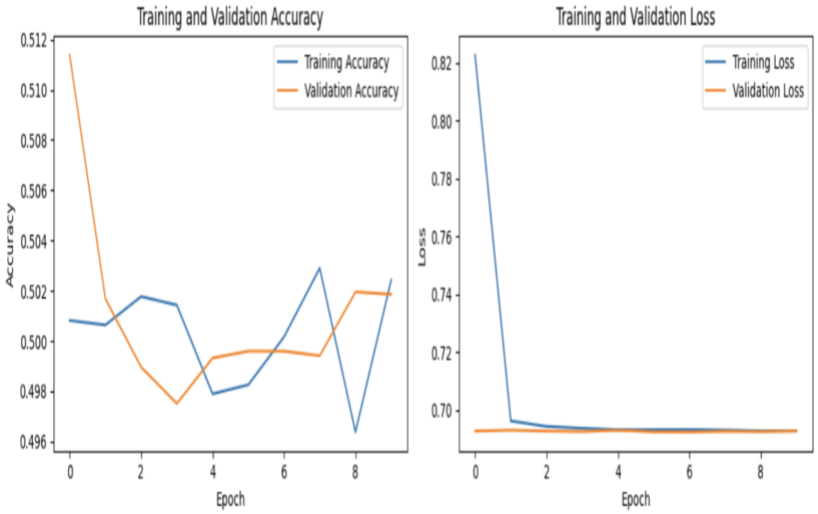

The above graphs represent the accuracy of training and validation sets. In addition, it also shows the training and validation losses.

**Model 2:**

Dense layers = 4, epochs = 20, batch size = 32, Regularization techniques used (drop out 0.5 after 1st layer and 0.3 after 2nd layer.)

Output for the Model 2 NN:

```
Epoch 1/20
1379/1379 [==============================] - 10s 5ms/step - loss: 0.8227 - accuracy: 0.5008 - val_loss: 0.6929 - val_accuracy: 0.5114
Epoch 2/20
1379/1379 [==============================] - 7s 5ms/step - loss: 0.6964 - accuracy: 0.5006 - val_loss: 0.6932 - val_accuracy: 0.5017
Epoch 3/20
1379/1379 [==============================] - 7s 5ms/step - loss: 0.6945 - accuracy: 0.5018 - val_loss: 0.6929 - val_accuracy: 0.4990
Epoch 4/20
1379/1379 [==============================] - 8s 6ms/step - loss: 0.6938 - accuracy: 0.5014 - val_loss: 0.6927 - val_accuracy: 0.4975
Epoch 5/20
1379/1379 [==============================] - 6s 5ms/step - loss: 0.6933 - accuracy: 0.4979 - val_loss: 0.6931 - val_accuracy: 0.4993
Epoch 6/20
1379/1379 [==============================] - 8s 5ms/step - loss: 0.6933 - accuracy: 0.4983 - val_loss: 0.6926 - val_accuracy: 0.4996
Epoch 7/20
1379/1379 [==============================] - 6s 5ms/step - loss: 0.6934 - accuracy: 0.5002 - val_loss: 0.6925 - val_accuracy: 0.4996
Epoch 8/20
1379/1379 [==============================] - 7s 5ms/step - loss: 0.6932 - accuracy: 0.5029 - val_loss: 0.6928 - val_accuracy: 0.4994
Epoch 9/20
1379/1379 [==============================] - 7s 5ms/step - loss: 0.6929 - accuracy: 0.4964 - val_loss: 0.6927 - val_accuracy: 0.5019
Epoch 10/20
1379/1379 [==============================] - 8s 6ms/step - loss: 0.6929 - accuracy: 0.5024 - val_loss: 0.6929 - val_accuracy: 0.5019
370/370 [==============================] - 1s 2ms/step
              precision    recall  f1-score   support

           0       0.42      0.00      0.00      5917
           1       0.50      1.00      0.67      5902

    accuracy                           0.50     11819
   macro avg       0.46      0.50      0.33     11819
weighted avg       0.46      0.50      0.33     11819
```

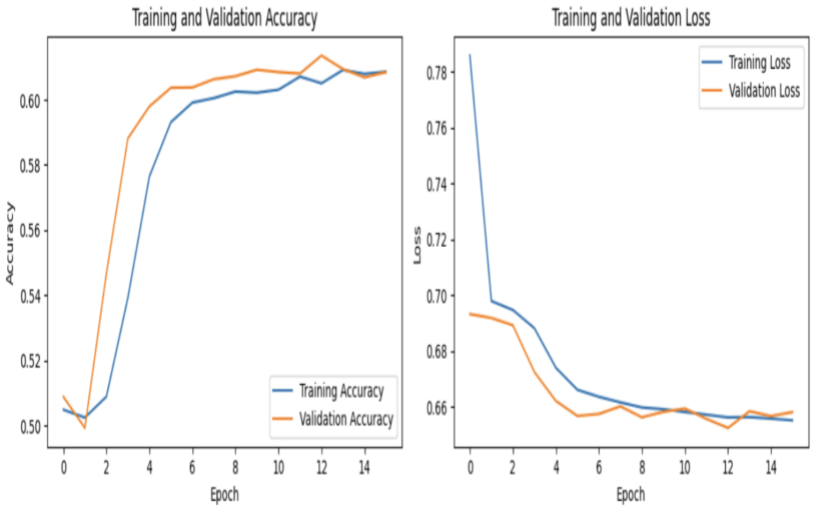


**Model 3:**

Dense layers = 4, epochs = 20, batch size = 64, Regularization techniques used (drop out 0.5 after 1st layer and 0.3 after 2nd layer.)

Output for the Model 3 NN:

```
Epoch 1/20
690/690 [==============================] - 5s 6ms/step - loss: 0.7858 - accuracy: 0.5050 - val_loss: 0.6933 - val_accuracy: 0.5089
Epoch 2/20
690/690 [==============================] - 4s 6ms/step - loss: 0.6980 - accuracy: 0.5026 - val_loss: 0.6919 - val_accuracy: 0.4994
Epoch 3/20
690/690 [==============================] - 3s 5ms/step - loss: 0.6948 - accuracy: 0.5090 - val_loss: 0.6893 - val_accuracy: 0.5467
Epoch 4/20
690/690 [==============================] - 3s 5ms/step - loss: 0.6882 - accuracy: 0.5393 - val_loss: 0.6726 - val_accuracy: 0.5882
Epoch 5/20
690/690 [==============================] - 4s 6ms/step - loss: 0.6741 - accuracy: 0.5765 - val_loss: 0.6622 - val_accuracy: 0.5980
Epoch 6/20
690/690 [==============================] - 3s 5ms/step - loss: 0.6662 - accuracy: 0.5931 - val_loss: 0.6568 - val_accuracy: 0.6037
Epoch 7/20
690/690 [==============================] - 3s 5ms/step - loss: 0.6637 - accuracy: 0.5991 - val_loss: 0.6576 - val_accuracy: 0.6038
Epoch 8/20
690/690 [==============================] - 3s 5ms/step - loss: 0.6617 - accuracy: 0.6005 - val_loss: 0.6603 - val_accuracy: 0.6063
Epoch 9/20
690/690 [==============================] - 4s 6ms/step - loss: 0.6599 - accuracy: 0.6025 - val_loss: 0.6564 - val_accuracy: 0.6072
Epoch 10/20
690/690 [==============================] - 4s 5ms/step - loss: 0.6592 - accuracy: 0.6022 - val_loss: 0.6583 - val_accuracy: 0.6092
Epoch 11/20
690/690 [==============================] - 3s 5ms/step - loss: 0.6583 - accuracy: 0.6031 - val_loss: 0.6595 - val_accuracy: 0.6085
Epoch 12/20
690/690 [==============================] - 3s 5ms/step - loss: 0.6573 - accuracy: 0.6071 - val_loss: 0.6558 - val_accuracy: 0.6080
Epoch 13/20
690/690 [==============================] - 4s 6ms/step - loss: 0.6563 - accuracy: 0.6050 - val_loss: 0.6526 - val_accuracy: 0.6135
Epoch 14/20
690/690 [==============================] - 3s 5ms/step - loss: 0.6565 - accuracy: 0.6091 - val_loss: 0.6585 - val_accuracy: 0.6094
Epoch 15/20
690/690 [==============================] - 3s 5ms/step - loss: 0.6559 - accuracy: 0.6079 - val_loss: 0.6568 - val_accuracy: 0.6068
Epoch 16/20
690/690 [==============================] - 3s 5ms/step - loss: 0.6553 - accuracy: 0.6086 - val_loss: 0.6582 - val_accuracy: 0.6085
370/370 [==============================] - 1s 2ms/step
              precision    recall  f1-score   support

           0       0.59      0.71      0.65      5917
           1       0.64      0.52      0.57      5902

    accuracy                           0.61     11819
   macro avg       0.62      0.61      0.61     11819
weighted avg       0.62      0.61      0.61     11819
```

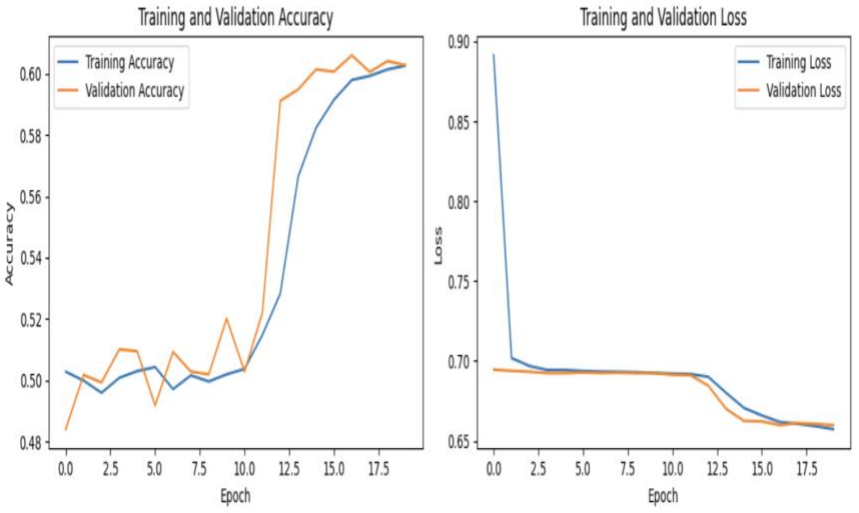


**Model 4:**

Dense layers = 4, epochs = 20, batch size = 128, Regularization techniques used (drop out 0.5 after 1st layer and 0.3 after 2nd layer.)

Output for the Model 4 NN:

```
Epoch 1/20
345/345 [==============================] - 3s 5ms/step - loss: 0.8913 - accuracy: 0.5028 - val_loss: 0.6946 - val_accuracy: 0.4841
Epoch 2/20
345/345 [==============================] - 2s 5ms/step - loss: 0.7018 - accuracy: 0.4999 - val_loss: 0.6938 - val_accuracy: 0.5018
Epoch 3/20
345/345 [==============================] - 2s 7ms/step - loss: 0.6968 - accuracy: 0.4959 - val_loss: 0.6932 - val_accuracy: 0.4993
Epoch 4/20
345/345 [==============================] - 2s 6ms/step - loss: 0.6944 - accuracy: 0.5008 - val_loss: 0.6924 - val_accuracy: 0.5101
Epoch 5/20
345/345 [==============================] - 2s 5ms/step - loss: 0.6943 - accuracy: 0.5030 - val_loss: 0.6924 - val_accuracy: 0.5095
Epoch 6/20
345/345 [==============================] - 2s 5ms/step - loss: 0.6937 - accuracy: 0.5043 - val_loss: 0.6927 - val_accuracy: 0.4918
Epoch 7/20
345/345 [==============================] - 2s 5ms/step - loss: 0.6933 - accuracy: 0.4971 - val_loss: 0.6924 - val_accuracy: 0.5093
Epoch 8/20
345/345 [==============================] - 2s 5ms/step - loss: 0.6932 - accuracy: 0.5016 - val_loss: 0.6926 - val_accuracy: 0.5029
Epoch 9/20
345/345 [==============================] - 2s 5ms/step - loss: 0.6929 - accuracy: 0.4996 - val_loss: 0.6924 - val_accuracy: 0.5019
Epoch 10/20
345/345 [==============================] - 2s 7ms/step - loss: 0.6925 - accuracy: 0.5019 - val_loss: 0.6924 - val_accuracy: 0.5202
Epoch 11/20
345/345 [==============================] - 2s 6ms/step - loss: 0.6920 - accuracy: 0.5037 - val_loss: 0.6914 - val_accuracy: 0.5029
Epoch 12/20
345/345 [==============================] - 2s 5ms/step - loss: 0.6918 - accuracy: 0.5148 - val_loss: 0.6911 - val_accuracy: 0.5221
Epoch 13/20
345/345 [==============================] - 2s 5ms/step - loss: 0.6901 - accuracy: 0.5283 - val_loss: 0.6846 - val_accuracy: 0.5912
Epoch 14/20
345/345 [==============================] - 2s 4ms/step - loss: 0.6800 - accuracy: 0.5664 - val_loss: 0.6700 - val_accuracy: 0.5949
Epoch 15/20
345/345 [==============================] - 2s 5ms/step - loss: 0.6706 - accuracy: 0.5824 - val_loss: 0.6625 - val_accuracy: 0.6014
Epoch 16/20
345/345 [==============================] - 2s 4ms/step - loss: 0.6659 - accuracy: 0.5915 - val_loss: 0.6622 - val_accuracy: 0.6007
Epoch 17/20
345/345 [==============================] - 2s 5ms/step - loss: 0.6618 - accuracy: 0.5980 - val_loss: 0.6598 - val_accuracy: 0.6060
Epoch 18/20
345/345 [==============================] - 2s 6ms/step - loss: 0.6607 - accuracy: 0.5993 - val_loss: 0.6611 - val_accuracy: 0.6006
Epoch 19/20
345/345 [==============================] - 2s 7ms/step - loss: 0.6592 - accuracy: 0.6014 - val_loss: 0.6605 - val_accuracy: 0.6041
Epoch 20/20
345/345 [==============================] - 2s 5ms/step - loss: 0.6573 - accuracy: 0.6026 - val_loss: 0.6598 - val_accuracy: 0.6028
370/370 [==============================] - 1s 2ms/step
              precision    recall  f1-score   support

           0       0.58      0.76      0.66      5917
           1       0.65      0.45      0.53      5902

    accuracy                           0.61     11819
   macro avg       0.62      0.61      0.60     11819
weighted avg       0.62      0.61      0.60     11819
```

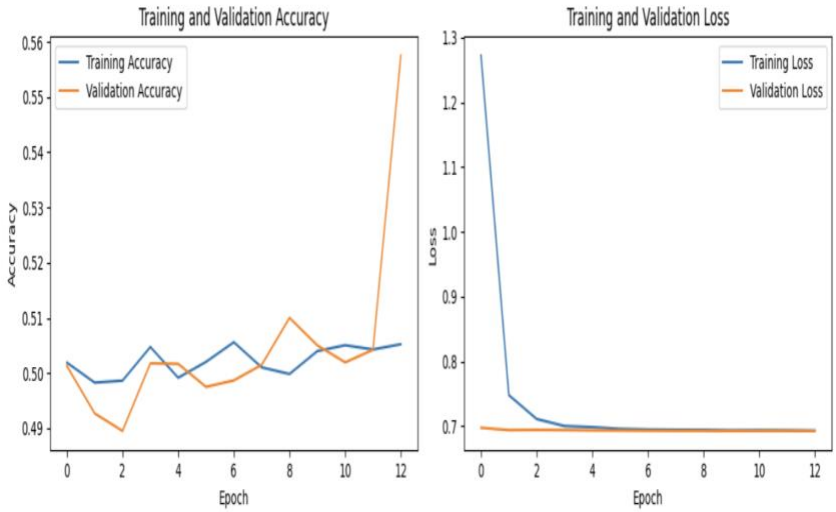


**Model 5:**

Dense layers = 4, epochs = 20, batch size = 32, Regularization techniques used (drop out 0.5 after 1st layer and 0.3 after 2nd layer.)

Output for the Model 5 NN:

```
Epoch 1/20
87/87 [==============================] - 3s 11ms/step - loss: 1.2721 - accuracy: 0.5019 - val_loss: 0.6972 - val_accuracy: 0.5013
Epoch 2/20
87/87 [==============================] - 1s 7ms/step - loss: 0.7478 - accuracy: 0.4983 - val_loss: 0.6937 - val_accuracy: 0.4927
Epoch 3/20
87/87 [==============================] - 0s 5ms/step - loss: 0.7108 - accuracy: 0.4986 - val_loss: 0.6940 - val_accuracy: 0.4895
Epoch 4/20
87/87 [==============================] - 0s 5ms/step - loss: 0.6999 - accuracy: 0.5047 - val_loss: 0.6937 - val_accuracy: 0.5018
Epoch 5/20
87/87 [==============================] - 0s 5ms/step - loss: 0.6983 - accuracy: 0.4992 - val_loss: 0.6932 - val_accuracy: 0.5017
Epoch 6/20
87/87 [==============================] - 0s 5ms/step - loss: 0.6955 - accuracy: 0.5020 - val_loss: 0.6930 - val_accuracy: 0.4975
Epoch 7/20
87/87 [==============================] - 0s 5ms/step - loss: 0.6945 - accuracy: 0.5056 - val_loss: 0.6927 - val_accuracy: 0.4987
Epoch 8/20
87/87 [==============================] - 0s 5ms/step - loss: 0.6941 - accuracy: 0.5010 - val_loss: 0.6926 - val_accuracy: 0.5015
Epoch 9/20
87/87 [==============================] - 0s 5ms/step - loss: 0.6940 - accuracy: 0.4998 - val_loss: 0.6926 - val_accuracy: 0.5100
Epoch 10/20
87/87 [==============================] - 0s 5ms/step - loss: 0.6932 - accuracy: 0.5040 - val_loss: 0.6924 - val_accuracy: 0.5050
Epoch 11/20
87/87 [==============================] - 0s 5ms/step - loss: 0.6936 - accuracy: 0.5051 - val_loss: 0.6925 - val_accuracy: 0.5019
Epoch 12/20
87/87 [==============================] - 0s 5ms/step - loss: 0.6932 - accuracy: 0.5043 - val_loss: 0.6927 - val_accuracy: 0.5042
Epoch 13/20
87/87 [==============================] - 0s 5ms/step - loss: 0.6927 - accuracy: 0.5052 - val_loss: 0.6924 - val_accuracy: 0.5575
370/370 [==============================] - 1s 2ms/step
              precision    recall  f1-score   support

           0       0.50      0.98      0.67      5917
           1       0.62      0.03      0.06      5902

    accuracy                           0.51     11819
   macro avg       0.56      0.51      0.36     11819
weighted avg       0.56      0.51      0.36     11819
```

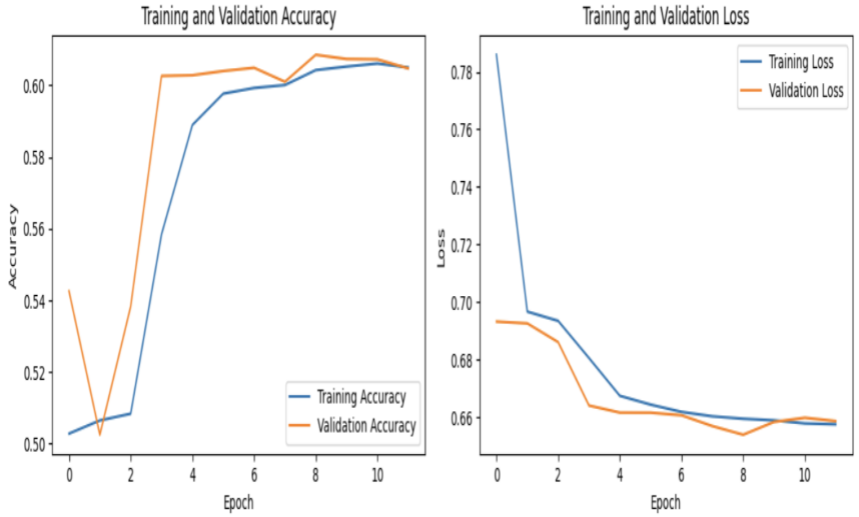


**Model 6:**

Dense layers = 4, epochs = 20, batch size = 512, Regularization techniques used (drop out 0.5 after 1st layer and 0.3 after 2nd layer.)

Output for the Model 6 NN:

```
Epoch 1/20
1379/1379 [==============================] - 8s 5ms/step - loss: 0.7859 - accuracy: 0.5028 - val_loss: 0.6933 - val_accuracy: 0.5426
Epoch 2/20
1379/1379 [==============================] - 7s 5ms/step - loss: 0.6967 - accuracy: 0.5065 - val_loss: 0.6926 - val_accuracy: 0.5025
Epoch 3/20
1379/1379 [==============================] - 6s 5ms/step - loss: 0.6935 - accuracy: 0.5084 - val_loss: 0.6861 - val_accuracy: 0.5383
Epoch 4/20
1379/1379 [==============================] - 7s 5ms/step - loss: 0.6806 - accuracy: 0.5584 - val_loss: 0.6641 - val_accuracy: 0.6027
Epoch 5/20
1379/1379 [==============================] - 6s 5ms/step - loss: 0.6675 - accuracy: 0.5890 - val_loss: 0.6617 - val_accuracy: 0.6028
Epoch 6/20
1379/1379 [==============================] - 7s 5ms/step - loss: 0.6644 - accuracy: 0.5977 - val_loss: 0.6616 - val_accuracy: 0.6040
Epoch 7/20
1379/1379 [==============================] - 6s 5ms/step - loss: 0.6619 - accuracy: 0.5993 - val_loss: 0.6607 - val_accuracy: 0.6049
Epoch 8/20
1379/1379 [==============================] - 9s 6ms/step - loss: 0.6604 - accuracy: 0.6001 - val_loss: 0.6570 - val_accuracy: 0.6010
Epoch 9/20
1379/1379 [==============================] - 7s 5ms/step - loss: 0.6595 - accuracy: 0.6043 - val_loss: 0.6540 - val_accuracy: 0.6086
Epoch 10/20
1379/1379 [==============================] - 7s 5ms/step - loss: 0.6590 - accuracy: 0.6053 - val_loss: 0.6584 - val_accuracy: 0.6074
Epoch 11/20
1379/1379 [==============================] - 7s 5ms/step - loss: 0.6579 - accuracy: 0.6061 - val_loss: 0.6599 - val_accuracy: 0.6073
Epoch 12/20
1379/1379 [==============================] - 7s 5ms/step - loss: 0.6576 - accuracy: 0.6050 - val_loss: 0.6587 - val_accuracy: 0.6048
370/370 [==============================] - 1s 3ms/step
              precision    recall  f1-score   support

           0       0.70      0.23      0.35      5917
           1       0.54      0.90      0.68      5902

    accuracy                           0.57     11819
   macro avg       0.62      0.57      0.51     11819
weighted avg       0.62      0.57      0.51     11819
```



In the investigation of various neural network models, a notable trend emerges as validation accuracy plateaus within the early epochs, specifically stabilizing between 5 to 8 epochs. During this period, the accuracy converges to approximately 60%, exhibiting minimal variance within a range of plus or minus 5%. This consistent pattern suggests a saturation point, indicating that the models cease to significantly learn from the provided data beyond this critical juncture.

**K-Fold Cross Validation for Neural Networks:**

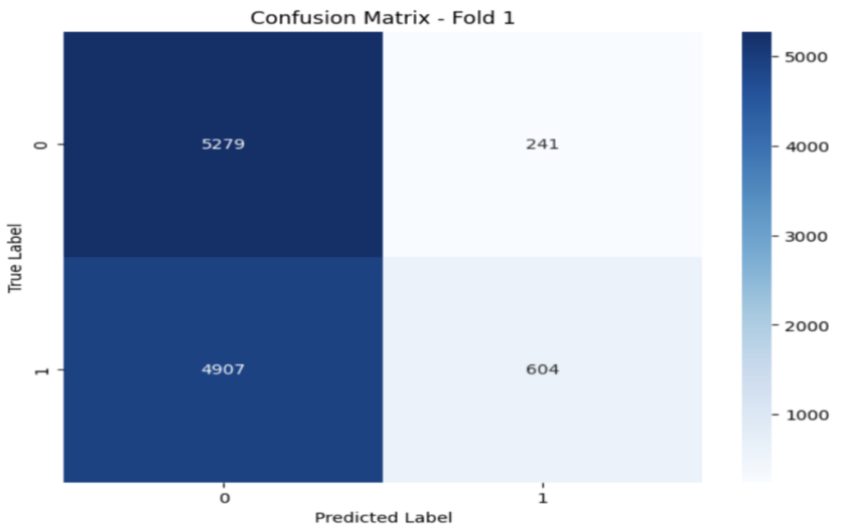
**Model 7:**

Dense layers = 4, epochs = 20, batch size = 32, Regularization techniques used (drop out 0.5 after 1st layer and 0.3 after 2nd layer.), 5 folds.

Output for the first fold NN:

```
Epoch 1/30
87/87 [==============================] - 2s 7ms/step - loss: 1.4465 - accuracy: 0.4974 - val_loss: 0.7104 - val_accuracy: 0.5017
Epoch 2/30
87/87 [==============================] - 0s 5ms/step - loss: 0.7767 - accuracy: 0.4986 - val_loss: 0.6940 - val_accuracy: 0.5014
Epoch 3/30
87/87 [==============================] - 0s 5ms/step - loss: 0.7249 - accuracy: 0.5040 - val_loss: 0.6966 - val_accuracy: 0.5013
Epoch 4/30
87/87 [==============================] - 0s 5ms/step - loss: 0.7104 - accuracy: 0.5029 - val_loss: 0.6928 - val_accuracy: 0.5011
Epoch 5/30
87/87 [==============================] - 0s 5ms/step - loss: 0.7031 - accuracy: 0.5029 - val_loss: 0.6932 - val_accuracy: 0.5015
Epoch 6/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6999 - accuracy: 0.4998 - val_loss: 0.6926 - val_accuracy: 0.5014
Epoch 7/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6978 - accuracy: 0.5023 - val_loss: 0.6926 - val_accuracy: 0.5039
Epoch 8/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6959 - accuracy: 0.5036 - val_loss: 0.6917 - val_accuracy: 0.5333
Epoch 9/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6958 - accuracy: 0.5009 - val_loss: 0.6928 - val_accuracy: 0.5047
Epoch 10/30
87/87 [==============================] - 1s 8ms/step - loss: 0.6944 - accuracy: 0.5056 - val_loss: 0.6926 - val_accuracy: 0.5011
Epoch 11/30
87/87 [==============================] - 1s 7ms/step - loss: 0.6946 - accuracy: 0.4997 - val_loss: 0.6929 - val_accuracy: 0.5035
345/345 [==============================] - 1s 2ms/step
```

**Confusion Matrix:**




True Negative instances: 5279

False Positive instances: 241

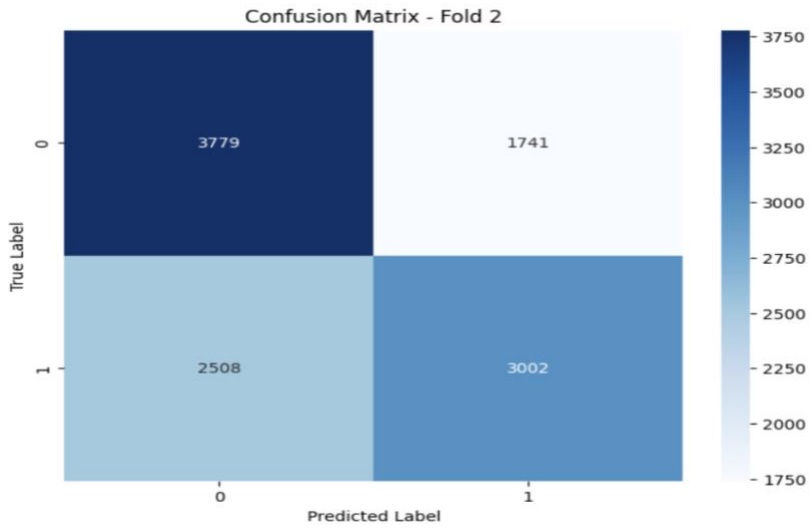
False Negative instances: 4907

True Positive instances: 604

Output for the second fold NN:

```
Epoch 1/30
87/87 [==============================] - 1s 9ms/step - loss: 0.6956 - accuracy: 0.5007 - val_loss: 0.6929 - val_accuracy: 0.5018
Epoch 2/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6948 - accuracy: 0.5036 - val_loss: 0.6923 - val_accuracy: 0.5157
Epoch 3/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6945 - accuracy: 0.5005 - val_loss: 0.6924 - val_accuracy: 0.5131
Epoch 4/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6935 - accuracy: 0.5068 - val_loss: 0.6919 - val_accuracy: 0.5617
Epoch 5/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6930 - accuracy: 0.5085 - val_loss: 0.6923 - val_accuracy: 0.5354
Epoch 6/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6924 - accuracy: 0.5155 - val_loss: 0.6909 - val_accuracy: 0.5849
Epoch 7/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6915 - accuracy: 0.5207 - val_loss: 0.6904 - val_accuracy: 0.5422
Epoch 8/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6884 - accuracy: 0.5345 - val_loss: 0.6811 - val_accuracy: 0.5888
Epoch 9/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6838 - accuracy: 0.5501 - val_loss: 0.6739 - val_accuracy: 0.5952
Epoch 10/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6786 - accuracy: 0.5646 - val_loss: 0.6720 - val_accuracy: 0.5868
Epoch 11/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6760 - accuracy: 0.5666 - val_loss: 0.6671 - val_accuracy: 0.6024
Epoch 12/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6721 - accuracy: 0.5776 - val_loss: 0.6643 - val_accuracy: 0.6067
Epoch 13/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6698 - accuracy: 0.5851 - val_loss: 0.6662 - val_accuracy: 0.6020
Epoch 14/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6669 - accuracy: 0.5884 - val_loss: 0.6617 - val_accuracy: 0.6110
Epoch 15/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6656 - accuracy: 0.5883 - val_loss: 0.6631 - val_accuracy: 0.6079
Epoch 16/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6624 - accuracy: 0.5930 - val_loss: 0.6615 - val_accuracy: 0.6118
Epoch 17/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6619 - accuracy: 0.5970 - val_loss: 0.6647 - val_accuracy: 0.6044
Epoch 18/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6622 - accuracy: 0.5942 - val_loss: 0.6622 - val_accuracy: 0.6092
Epoch 19/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6611 - accuracy: 0.5966 - val_loss: 0.6591 - val_accuracy: 0.6148
Epoch 20/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6592 - accuracy: 0.5985 - val_loss: 0.6600 - val_accuracy: 0.5990
Epoch 21/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6594 - accuracy: 0.6008 - val_loss: 0.6610 - val_accuracy: 0.6066
Epoch 22/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6586 - accuracy: 0.6012 - val_loss: 0.6607 - val_accuracy: 0.6121
345/345 [==============================] - 1s 2ms/step
```

**Confusion Matrix:**




True Negative instances: 3779

False Positive instances: 1741

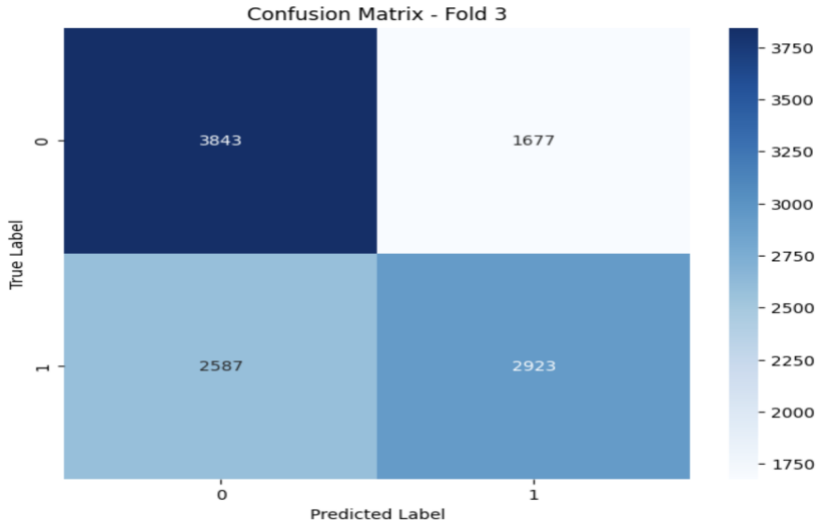
False Negative instances: 2508

True Positive instances: 3002

Output for the third fold NN:

```
Epoch 1/30
87/87 [==============================] - 1s 9ms/step - loss: 0.6588 - accuracy: 0.5994 - val_loss: 0.6585 - val_accuracy: 0.6126
Epoch 2/30
87/87 [==============================] - 1s 7ms/step - loss: 0.6593 - accuracy: 0.6023 - val_loss: 0.6559 - val_accuracy: 0.6122
Epoch 3/30
87/87 [==============================] - 1s 8ms/step - loss: 0.6587 - accuracy: 0.6013 - val_loss: 0.6574 - val_accuracy: 0.6110
Epoch 4/30
87/87 [==============================] - 1s 6ms/step - loss: 0.6594 - accuracy: 0.6030 - val_loss: 0.6536 - val_accuracy: 0.6134
Epoch 5/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6580 - accuracy: 0.6020 - val_loss: 0.6573 - val_accuracy: 0.6045
Epoch 6/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6574 - accuracy: 0.6030 - val_loss: 0.6550 - val_accuracy: 0.6161
Epoch 7/30
87/87 [==============================] - 0s 6ms/step - loss: 0.6566 - accuracy: 0.6056 - val_loss: 0.6573 - val_accuracy: 0.6140
345/345 [==============================] - 1s 2ms/step
```

**Confusion Matrix:**




True Negative instances: 3843

False Positive instances: 1677

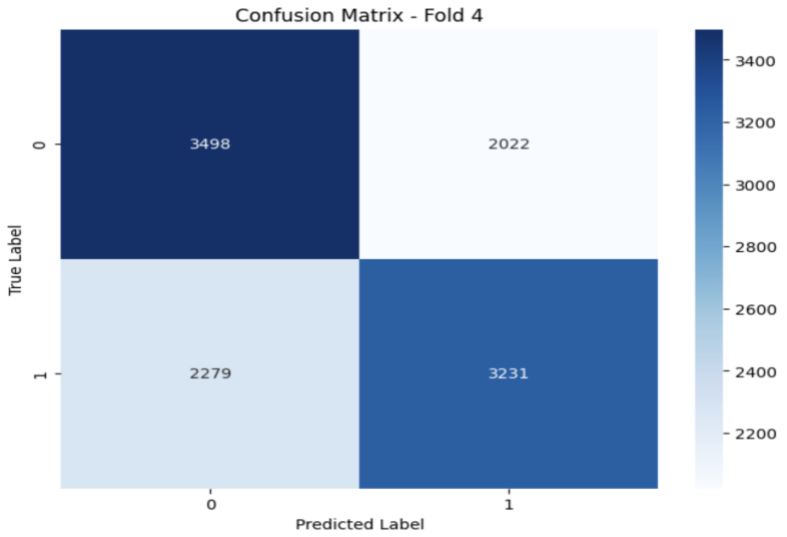
False Negative instances: 2587

True Positive instances: 2923

Output for the fourth fold NN:

```
Epoch 1/30
87/87 [==============================] - 1s 6ms/step - loss: 0.6576 - accuracy: 0.6029 - val_loss: 0.6601 - val_accuracy: 0.6086
Epoch 2/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6577 - accuracy: 0.6027 - val_loss: 0.6580 - val_accuracy: 0.6083
Epoch 3/30
87/87 [==============================] - 0s 6ms/step - loss: 0.6567 - accuracy: 0.6028 - val_loss: 0.6545 - val_accuracy: 0.6093
Epoch 4/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6567 - accuracy: 0.6074 - val_loss: 0.6602 - val_accuracy: 0.6061
Epoch 5/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6564 - accuracy: 0.6070 - val_loss: 0.6587 - val_accuracy: 0.6077
Epoch 6/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6556 - accuracy: 0.6041 - val_loss: 0.6542 - val_accuracy: 0.6101
Epoch 7/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6559 - accuracy: 0.6062 - val_loss: 0.6576 - val_accuracy: 0.6100
Epoch 8/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6553 - accuracy: 0.6062 - val_loss: 0.6554 - val_accuracy: 0.6091
Epoch 9/30
87/87 [==============================] - 0s 5ms/step - loss: 0.6549 - accuracy: 0.6092 - val_loss: 0.6567 - val_accuracy: 0.6092
345/345 [==============================] - 1s 2ms/step
```

**Confusion Matrix:**




True Negative instances: 3498

False Positive instances: 2022

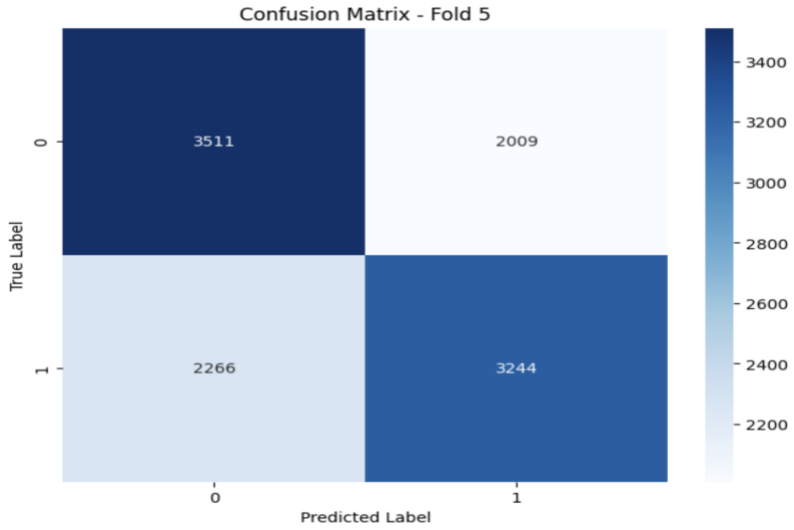
False Negative instances: 2279

True Positive instances: 3231

Output for the fifth fold NN:

```
Epoch 1/30
87/87 [==============================] - 1s 7ms/step - loss: 0.6552 - accuracy: 0.6072 - val_loss: 0.6562 - val_accuracy: 0.6124
Epoch 2/30
87/87 [==============================] - 1s 8ms/step - loss: 0.6552 - accuracy: 0.6084 - val_loss: 0.6572 - val_accuracy: 0.6103
Epoch 3/30
87/87 [==============================] - 1s 7ms/step - loss: 0.6546 - accuracy: 0.6094 - val_loss: 0.6589 - val_accuracy: 0.6084
Epoch 4/30
87/87 [==============================] - 1s 7ms/step - loss: 0.6556 - accuracy: 0.6037 - val_loss: 0.6563 - val_accuracy: 0.6122
345/345 [==============================] - 1s 2ms/step
```

**Confusion Matrix:**




True Negative instances: 3511

False Positive instances: 2009

False Negative instances: 2266

True Positive instances: 3244

```
Average Accuracy: 0.5967990332692183
Average Precision: 0.6431578797852627
Average Recall: 0.47201054087172956
Average F1 Score: 0.5114153696361845
```

The above values are calculated based on the average of the above K-folds implemented.

Despite high expectations for neural networks to outshine other models, the application fell short of delivering the anticipated boost in accuracy. Even with their advanced capabilities, neural networks couldn't provide the desired enhancement to the model's performance.

**Project Results:**

Comparison between Logistic Regression, Random Forest, and Support Vector Machine Models:

| Model | Validation-Accuracy | Test-Accuracy | Validation-F1 score | Test-F1 score |
|---|---|---|---|---|
| Logistic Regression | 0.6016 | 0.6098 | 0.62 | 0.64 |
| Random Forest | 0.6079 | 0.6121 | 0.62 | 0.62 |
| Support Vector Machine | 0.5932 | 0.5978 | 0.63 | 0.63 |

The following is the table for all the Neural Networks performed:

| Accuracy | F1-Score |
|---|---|
| 0.60 | 0.64 |
| 0.50 | 0.67 |
| 0.61 | 0.65 |
| 0.61 | 0.66 |
| 0.51 | 0.67 |
| 0.57 | 0.68 |

**Impact of the Project Outcomes:**

- Improving Patient Care: Studies using this dataset have led to a better understanding of factors contributing to readmissions in diabetic patients. This knowledge assists healthcare providers in improving patient care strategies, especially in managing chronic conditions like diabetes.

- Healthcare Policy Development: Insights gained from the data have informed healthcare policy, particularly in tailoring approaches to reduce readmission rates. This has implications for both patient outcomes and healthcare costs.

- Resource Allocation and Management: Insights from the data can help hospitals and healthcare systems in optimizing resource allocation. By identifying patterns and trends in hospital admissions, healthcare providers can better manage staffing, bed occupancy, and other critical resources.

- Global Health Implications: While the dataset is specific to the U.S., the insights gained can have broader implications for global health. The findings can inform diabetes care strategies in other countries, especially those with similar healthcare structures or patient demographics.