CIND 820- Big Data Analytics Project

Using Machine Learning for Prediction of Early Readmission of Diabetic Patients

Supervised by: Ceni Babaoglu

Presented by: Nehal Gamal Mohamed (501278190)

In [ ]:
```
!pip install pandas
import sys
!pip install matplotlib
!pip install graphviz
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-p
ackages (2.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/p
ython3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/
dist-packages (from pandas) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.1
0/dist-packages (from pandas) (2024.1)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.1
0/dist-packages (from pandas) (1.25.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist
-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/di
st-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python
3.10/dist-packages (from matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/
dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python
3.10/dist-packages (from matplotlib) (4.53.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python
3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/d
ist-packages (from matplotlib) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.
10/dist-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.1
0/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python
3.10/dist-packages (from matplotlib) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/pyt
hon3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist
-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist
-packages (0.20.3)
```

In [1]:
```
#importing necessary libraries
import csv
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

In [2]:
```
#Uploading diabetic_data csv file
from google.colab import files
data = files.upload()
```

```
#Instintiating DataReader class
class DataReader:
    def read_csv(self, filename):
        df = pd.read_csv(filename)
        return df

filename = next(iter(data))
data_reader = DataReader()
df = data_reader.read_csv(filename)
#Displaying the first 10 records of the dataset
print(df.head(10))
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving diabetic_data (version 1).csv to diabetic_data (version 1).csv
   encounter_id  patient_nbr             race  gender       age weight  \
0       2278392      8222157        Caucasian  Female    [0-10)      ?
1        149190     55629189        Caucasian  Female   [10-20)      ?
2         64410     86047875  AfricanAmerican  Female   [20-30)      ?
3        500364     82442376        Caucasian    Male   [30-40)      ?
4         16680     42519267        Caucasian    Male   [40-50)      ?
5         35754     82637451        Caucasian    Male   [50-60)      ?
6         55842     84259809        Caucasian    Male   [60-70)      ?
7         63768    114882984        Caucasian    Male   [70-80)      ?
8         12522     48330783        Caucasian  Female   [80-90)      ?
9         15738     63555939        Caucasian  Female  [90-100)      ?

   admission_type_id  discharge_disposition_id  admission_source_id  \
0                  6                        25                    1
1                  1                         1                    7
2                  1                         1                    7
3                  1                         1                    7
4                  1                         1                    7
5                  2                         1                    2
6                  3                         1                    2
7                  1                         1                    7
8                  2                         1                    4
9                  3                         3                    4

   time_in_hospital  ... citoglipton insulin  glyburide-metformin  \
0                 1  ...          No      No                   No
1                 3  ...          No      Up                   No
2                 2  ...          No      No                   No
3                 2  ...          No      Up                   No
4                 1  ...          No  Steady                   No
5                 3  ...          No  Steady                   No
6                 4  ...          No  Steady                   No
7                 5  ...          No      No                   No
8                13  ...          No  Steady                   No
9                12  ...          No  Steady                   No

   glipizide-metformin  glimepiride-pioglitazone  metformin-rosiglitazone
\
0                   No                        No                       No
1                   No                        No                       No
2                   No                        No                       No
3                   No                        No                       No
4                   No                        No                       No
5                   No                        No                       No
6                   No                        No                       No
7                   No                        No                       No
8                   No                        No                       No
9                   No                        No                       No

   metformin-pioglitazone  change diabetesMed readmitted
0                      No      No          No         NO
1                      No      Ch         Yes        >30
2                      No      No         Yes         NO
3                      No      Ch         Yes         NO
4                      No      Ch         Yes         NO
5                      No      No         Yes        >30
6                      No      Ch         Yes         NO
7                      No      No         Yes        >30
8                      No      Ch         Yes         NO
9                      No      Ch         Yes         NO

[10 rows x 50 columns]
```

In [3]:
```
#Displaying variables data types
print("Dataset Information:")
print(df.info())
```

```
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
Data columns (total 50 columns):
 #   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             5346 non-null    object
 23  A1Cresult                 17018 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)
memory usage: 38.8+ MB
None
```

In [4]:
```
#Defining lists for categorical variables and numeric variables
categorical_columns = [
    'race',
```

```
    'gender',
    'age',
    'weight',
    'payer_code',
    'medical_specialty',
    'diag_1',
    'diag_2',
    'diag_3',
    'max_glu_serum',
    'A1Cresult',
    'metformin',
    'repaglinide',
    'nateglinide',
    'chlorpropamide',
    'glimepiride',
    'acetohexamide',
    'glipizide',
    'glyburide',
    'tolbutamide',
    'pioglitazone',
    'rosiglitazone',
    'acarbose',
    'miglitol',
    'troglitazone',
    'tolazamide',
    'examide',
    'citoglipton',
    'insulin',
    'glyburide-metformin',
    'glipizide-metformin',
    'glimepiride-pioglitazone',
    'metformin-rosiglitazone',
    'metformin-pioglitazone',
    'change',
    'diabetesMed',
    'readmitted'
]

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

In [5]:

```
#Dataset Description
print("Dataset Description:")
print(df.describe(include='all'))
```

Dataset Description:

	encounter_id	patient_nbr	race	gender	age	weight
count	1.017660e+05	1.017660e+05	101766	101766	101766	101766
unique	NaN	NaN	6	3	10	10
top	NaN	NaN	Caucasian	Female	[70-80)	?
freq	NaN	NaN	76099	54708	26068	98569
mean	1.652016e+08	5.433040e+07	NaN	NaN	NaN	NaN
std	1.026403e+08	3.869636e+07	NaN	NaN	NaN	NaN
min	1.252200e+04	1.350000e+02	NaN	NaN	NaN	NaN
25%	8.496119e+07	2.341322e+07	NaN	NaN	NaN	NaN
50%	1.523890e+08	4.550514e+07	NaN	NaN	NaN	NaN
75%	2.302709e+08	8.754595e+07	NaN	NaN	NaN	NaN
max	4.438672e+08	1.895026e+08	NaN	NaN	NaN	NaN

	admission_type_id	discharge_disposition_id	admission_source_id
count	101766.000000	101766.000000	101766.000000
unique	NaN	NaN	NaN
top	NaN	NaN	NaN
freq	NaN	NaN	NaN
mean	2.024006	3.715642	5.754437
std	1.445403	5.280166	4.064081
min	1.000000	1.000000	1.000000
25%	1.000000	1.000000	1.000000
50%	1.000000	1.000000	7.000000
75%	3.000000	4.000000	7.000000
max	8.000000	28.000000	25.000000

	time_in_hospital	...	citoglipton	insulin	glyburide-metformin
count	101766.000000	...	101766	101766	101766
unique	NaN	...	1	4	4
top	NaN	...	No	No	No
freq	NaN	...	101766	47383	101060
mean	4.395987	...	NaN	NaN	NaN
std	2.985108	...	NaN	NaN	NaN
min	1.000000	...	NaN	NaN	NaN
25%	2.000000	...	NaN	NaN	NaN
50%	4.000000	...	NaN	NaN	NaN
75%	6.000000	...	NaN	NaN	NaN
max	14.000000	...	NaN	NaN	NaN

	glipizide-metformin	glimepiride-pioglitazone
count	101766	101766
unique	2	2
top	No	No
freq	101753	101765
mean	NaN	NaN
std	NaN	NaN
min	NaN	NaN
25%	NaN	NaN
50%	NaN	NaN
75%	NaN	NaN
max	NaN	NaN

	metformin-rosiglitazone	metformin-pioglitazone	change	diabetesMed
count	101766	101766	101766	101766
unique	2	2	2	2
top	No	No	No	Yes
freq	101764	101765	54755	78363

```
63
mean                        NaN                     NaN      NaN          N
aN
std                         NaN                     NaN      NaN          N
aN
min                         NaN                     NaN      NaN          N
aN
25%                         NaN                     NaN      NaN          N
aN
50%                         NaN                     NaN      NaN          N
aN
75%                         NaN                     NaN      NaN          N
aN
max                         NaN                     NaN      NaN          N
aN

       readmitted
count      101766
unique          3
top            NO
freq        54864
mean          NaN
std           NaN
min           NaN
25%           NaN
50%           NaN
75%           NaN
max           NaN

[11 rows x 50 columns]
```

In [6]:

```
#Converted blank and "?" to NaN to represent missing values
# Replace '?' and blank cells with NaN
df.replace({'?': pd.NA, '': pd.NA}, inplace=True)

# Replaced 'None' in 'max_glu_serum' and 'A1Cresult' to be seen as 'No Te
df['max_glu_serum'].replace({pd.NA: 'No Test', 'None': 'No Test'}, inplac
df['A1Cresult'].replace({pd.NA: 'No Test', 'None': 'No Test'}, inplace=Tr
missing_values = df.isna().sum()
print("Number of missing values in each column:")
print(missing_values)
```

```
Number of missing values in each column:
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
```

In [7]:

```python
# converting numeric columns to numeric data types and categorical column
for col in numeric_columns:
    df[col] = pd.to_numeric(df[col], errors='coerce')

for col in categorical_columns:
    if col in df.columns:
        df[col] = df[col].astype('category')
print(df.dtypes)
print(df[numeric_columns].dtypes)
```

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

```
number_diagnoses            int64
dtype: object
```

In [8]:
```python
# Handle missing values by replacing with mode
for column in ['race', 'diag_1', 'diag_2', 'diag_3']:
    df[column].fillna(df[column].mode()[0], inplace=True)

# Verify changes
print(df.isnull().sum())
```

```
encounter_id                   0
patient_nbr                    0
race                           0
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                         0
diag_2                         0
diag_3                         0
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
```

In [9]:
```
# Drop unnecessary columns
df.drop(columns=['weight', 'payer_code', 'medical_specialty', 'encounter_

# Verify the remaining columns
print("Remaining columns after dropping unnecessary ones:")
print(df.columns)
```

```
Remaining columns after dropping unnecessary ones:
Index(['race', 'gender', 'age', 'admission_type_id',
       'discharge_disposition_id', 'admission_source_id', 'time_in_hospit
al',
       'num_lab_procedures', 'num_procedures', 'num_medications',
       'number_outpatient', 'number_emergency', 'number_inpatient', 'diag
_1',
       'diag_2', 'diag_3', 'number_diagnoses', 'max_glu_serum', 'A1Cresul
t',
       'metformin', 'repaglinide', 'nateglinide', 'chlorpropamide',
       'glimepiride', 'acetohexamide', 'glipizide', 'glyburide', 'tolbuta
mide',
       'pioglitazone', 'rosiglitazone', 'acarbose', 'miglitol', 'troglita
zone',
       'tolazamide', 'examide', 'citoglipton', 'insulin',
       'glyburide-metformin', 'glipizide-metformin',
       'glimepiride-pioglitazone', 'metformin-rosiglitazone',
       'metformin-pioglitazone', 'change', 'diabetesMed', 'readmitted'],
      dtype='object')
```

In [10]:
```
#Checking for duplicate records
duplicate_records = df.duplicated().sum()
print(f"Number of duplicate records: {duplicate_records}")
```

```
Number of duplicate records: 0
```

In [11]:
```
# Simplify and categorize diag_1, diag_2, diag_3
def simplify_diag(x):
    if pd.isna(x):
        return -1
    if 'V' in x or 'E' in x:
        return 0
    try:
        return int(x[:3])
    except:
        return -1

df['diag_1'] = df['diag_1'].apply(simplify_diag)
df['diag_2'] = df['diag_2'].apply(simplify_diag)
df['diag_3'] = df['diag_3'].apply(simplify_diag)

# Verify changes
print(df[['diag_1', 'diag_2', 'diag_3']].head())
```

```
   diag_1  diag_2  diag_3
0     250     276     250
1     276     250     255
2     648     250       0
3       8     250     403
4     197     157     250
```

In [12]:
```
# Combine 'NO' and '>30' categories into a single '>30' category
df['readmitted'] = df['readmitted'].replace({'NO': '>30', '>30': '>30', '

# Verify changes
print(df['readmitted'].value_counts())
```

```
readmitted
>30    90409
<30    11357
Name: count, dtype: int64
```

In [13]:

```
# Here I dropped the List of columns dominated by "No" category since the
columns_dominated_by_no = [
    'metformin', 'repaglinide', 'nateglinide', 'chlorpropamide',
    'glimepiride', 'acetohexamide', 'glipizide', 'glyburide',
    'tolbutamide', 'pioglitazone', 'rosiglitazone', 'acarbose',
    'troglitazone', 'tolazamide', 'examide', 'citoglipton',
    'glyburide-metformin', 'glipizide-metformin', 'glimepiride-pioglitazo
    'metformin-rosiglitazone', 'metformin-pioglitazone'
]
df.drop(columns=columns_dominated_by_no, inplace=True)

# Check the remaining columns
print("Remaining columns after dropping unnecessary ones:")
print(df.columns)
print(df.info())
```

```
Remaining columns after dropping unnecessary ones:
Index(['race', 'gender', 'age', 'admission_type_id',
       'discharge_disposition_id', 'admission_source_id', 'time_in_hospit
al',
       'num_lab_procedures', 'num_procedures', 'num_medications',
       'number_outpatient', 'number_emergency', 'number_inpatient', 'diag
_1',
       'diag_2', 'diag_3', 'number_diagnoses', 'max_glu_serum', 'A1Cresul
t',
       'miglitol', 'insulin', 'change', 'diabetesMed', 'readmitted'],
      dtype='object')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
Data columns (total 24 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   race                      101766 non-null  category
 1   gender                    101766 non-null  category
 2   age                       101766 non-null  category
 3   admission_type_id         101766 non-null  int64
 4   discharge_disposition_id  101766 non-null  int64
 5   admission_source_id       101766 non-null  int64
 6   time_in_hospital          101766 non-null  int64
 7   num_lab_procedures        101766 non-null  int64
 8   num_procedures            101766 non-null  int64
 9   num_medications           101766 non-null  int64
 10  number_outpatient         101766 non-null  int64
 11  number_emergency          101766 non-null  int64
 12  number_inpatient          101766 non-null  int64
 13  diag_1                    101766 non-null  int64
 14  diag_2                    101766 non-null  int64
 15  diag_3                    101766 non-null  int64
 16  number_diagnoses          101766 non-null  int64
 17  max_glu_serum             101766 non-null  category
 18  A1Cresult                 101766 non-null  category
 19  miglitol                  101766 non-null  category
 20  insulin                   101766 non-null  category
 21  change                    101766 non-null  category
 22  diabetesMed               101766 non-null  category
 23  readmitted                101766 non-null  category
dtypes: category(10), int64(14)
memory usage: 11.8 MB
None
```

In [14]:

```
# Ensure that 'admission_type_id', 'discharge_disposition_id', 'admission
df[['admission_type_id', 'discharge_disposition_id', 'admission_source_id

# List of categorical features
categorical_features = ['race', 'gender', 'age', 'max_glu_serum', 'A1Cres
                        'admission_type_id', 'discharge_disposition_id',

# Convert all categorical columns to strings
df[categorical_features] = df[categorical_features].astype(str)

# Verify conversion
print(df[categorical_features].dtypes)
```

```
race                        object
gender                      object
age                         object
max_glu_serum               object
A1Cresult                   object
miglitol                    object
insulin                     object
change                      object
diabetesMed                 object
admission_type_id           object
discharge_disposition_id    object
admission_source_id         object
dtype: object
```

In [15]:

```
# One-hot encode categorical features
df = pd.get_dummies(df, columns=categorical_features, drop_first=True)

# Verify the changes
print(df.head())
print(df.info())
```

```
   time_in_hospital  num_lab_procedures  num_procedures  num_medications
\
0                 1                  41               0                1
1                 3                  59               0               18
2                 2                  11               5               13
3                 2                  44               1               16
4                 1                  51               0                8

   number_outpatient  number_emergency  number_inpatient  diag_1  diag_2
\
0                  0                 0                 0     250     276
1                  0                 0                 0     276     250
2                  2                 0                 1     648     250
3                  0                 0                 0       8     250
4                  0                 0                 0     197     157

   diag_3  ...  admission_source_id_20 admission_source_id_22  \
0     250  ...                  False                 False
1     255  ...                  False                 False
2       0  ...                  False                 False
3     403  ...                  False                 False
4     250  ...                  False                 False

   admission_source_id_25  admission_source_id_3  admission_source_id_4
\
0                   False                  False                  False
1                   False                  False                  False
2                   False                  False                  False
3                   False                  False                  False
4                   False                  False                  False

   admission_source_id_5  admission_source_id_6  admission_source_id_7  \
0                  False                  False                  False
1                  False                  False                   True
2                  False                  False                   True
3                  False                  False                   True
4                  False                  False                   True

   admission_source_id_8  admission_source_id_9
0                  False                  False
1                  False                  False
2                  False                  False
3                  False                  False
4                  False                  False

[5 rows x 89 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
Data columns (total 89 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   time_in_hospital            101766 non-null  int64
 1   num_lab_procedures          101766 non-null  int64
 2   num_procedures              101766 non-null  int64
 3   num_medications             101766 non-null  int64
 4   number_outpatient           101766 non-null  int64
 5   number_emergency            101766 non-null  int64
 6   number_inpatient            101766 non-null  int64
 7   diag_1                      101766 non-null  int64
 8   diag_2                      101766 non-null  int64
 9   diag_3                      101766 non-null  int64
 10  number_diagnoses            101766 non-null  int64
 11  readmitted                  101766 non-null  category
```

```
12  race_Asian                    101766 non-null  bool
13  race_Caucasian                101766 non-null  bool
14  race_Hispanic                 101766 non-null  bool
15  race_Other                    101766 non-null  bool
16  gender_Male                   101766 non-null  bool
17  gender_Unknown/Invalid        101766 non-null  bool
18  age_[10-20)                   101766 non-null  bool
19  age_[20-30)                   101766 non-null  bool
20  age_[30-40)                   101766 non-null  bool
21  age_[40-50)                   101766 non-null  bool
22  age_[50-60)                   101766 non-null  bool
23  age_[60-70)                   101766 non-null  bool
24  age_[70-80)                   101766 non-null  bool
25  age_[80-90)                   101766 non-null  bool
26  age_[90-100)                  101766 non-null  bool
27  max_glu_serum_>300            101766 non-null  bool
28  max_glu_serum_No Test         101766 non-null  bool
29  max_glu_serum_Norm            101766 non-null  bool
30  A1Cresult_>8                  101766 non-null  bool
31  A1Cresult_No Test             101766 non-null  bool
32  A1Cresult_Norm                101766 non-null  bool
33  miglitol_No                   101766 non-null  bool
34  miglitol_Steady               101766 non-null  bool
35  miglitol_Up                   101766 non-null  bool
36  insulin_No                    101766 non-null  bool
37  insulin_Steady                101766 non-null  bool
38  insulin_Up                    101766 non-null  bool
39  change_No                     101766 non-null  bool
40  diabetesMed_Yes               101766 non-null  bool
41  admission_type_id_2           101766 non-null  bool
42  admission_type_id_3           101766 non-null  bool
43  admission_type_id_4           101766 non-null  bool
44  admission_type_id_5           101766 non-null  bool
45  admission_type_id_6           101766 non-null  bool
46  admission_type_id_7           101766 non-null  bool
47  admission_type_id_8           101766 non-null  bool
48  discharge_disposition_id_10   101766 non-null  bool
49  discharge_disposition_id_11   101766 non-null  bool
50  discharge_disposition_id_12   101766 non-null  bool
51  discharge_disposition_id_13   101766 non-null  bool
52  discharge_disposition_id_14   101766 non-null  bool
53  discharge_disposition_id_15   101766 non-null  bool
54  discharge_disposition_id_16   101766 non-null  bool
55  discharge_disposition_id_17   101766 non-null  bool
56  discharge_disposition_id_18   101766 non-null  bool
57  discharge_disposition_id_19   101766 non-null  bool
58  discharge_disposition_id_2    101766 non-null  bool
59  discharge_disposition_id_20   101766 non-null  bool
60  discharge_disposition_id_22   101766 non-null  bool
61  discharge_disposition_id_23   101766 non-null  bool
62  discharge_disposition_id_24   101766 non-null  bool
63  discharge_disposition_id_25   101766 non-null  bool
64  discharge_disposition_id_27   101766 non-null  bool
65  discharge_disposition_id_28   101766 non-null  bool
66  discharge_disposition_id_3    101766 non-null  bool
67  discharge_disposition_id_4    101766 non-null  bool
68  discharge_disposition_id_5    101766 non-null  bool
69  discharge_disposition_id_6    101766 non-null  bool
70  discharge_disposition_id_7    101766 non-null  bool
71  discharge_disposition_id_8    101766 non-null  bool
72  discharge_disposition_id_9    101766 non-null  bool
73  admission_source_id_10        101766 non-null  bool
74  admission_source_id_11        101766 non-null  bool
```

```
 75  admission_source_id_13       101766 non-null  bool
 76  admission_source_id_14       101766 non-null  bool
 77  admission_source_id_17       101766 non-null  bool
 78  admission_source_id_2        101766 non-null  bool
 79  admission_source_id_20       101766 non-null  bool
 80  admission_source_id_22       101766 non-null  bool
 81  admission_source_id_25       101766 non-null  bool
 82  admission_source_id_3        101766 non-null  bool
 83  admission_source_id_4        101766 non-null  bool
 84  admission_source_id_5        101766 non-null  bool
 85  admission_source_id_6        101766 non-null  bool
 86  admission_source_id_7        101766 non-null  bool
 87  admission_source_id_8        101766 non-null  bool
 88  admission_source_id_9        101766 non-null  bool
dtypes: bool(77), category(1), int64(11)
memory usage: 16.1 MB
None
```

In [16]:

```python
# Convert boolean columns to integer
bool_columns = df.select_dtypes(include='bool').columns
df[bool_columns] = df[bool_columns].astype(int)

# Verify the changes
print(df.head())
print(df.info())
```

```
   time_in_hospital  num_lab_procedures  num_procedures  num_medications
\
0                 1                  41               0                1
1                 3                  59               0               18
2                 2                  11               5               13
3                 2                  44               1               16
4                 1                  51               0                8

   number_outpatient  number_emergency  number_inpatient  diag_1  diag_2
\
0                  0                 0                 0     250     276
1                  0                 0                 0     276     250
2                  2                 0                 1     648     250
3                  0                 0                 0       8     250
4                  0                 0                 0     197     157

   diag_3  ...  admission_source_id_20 admission_source_id_22  \
0     250  ...                      0                     0
1     255  ...                      0                     0
2       0  ...                      0                     0
3     403  ...                      0                     0
4     250  ...                      0                     0

   admission_source_id_25  admission_source_id_3  admission_source_id_4
\
0                       0                      0                      0
1                       0                      0                      0
2                       0                      0                      0
3                       0                      0                      0
4                       0                      0                      0

   admission_source_id_5  admission_source_id_6  admission_source_id_7  \
0                      0                      0                      0
1                      0                      0                      1
2                      0                      0                      1
3                      0                      0                      1
4                      0                      0                      1

   admission_source_id_8  admission_source_id_9
0                      0                      0
1                      0                      0
2                      0                      0
3                      0                      0
4                      0                      0

[5 rows x 89 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
Data columns (total 89 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   time_in_hospital            101766 non-null  int64
 1   num_lab_procedures          101766 non-null  int64
 2   num_procedures              101766 non-null  int64
 3   num_medications             101766 non-null  int64
 4   number_outpatient           101766 non-null  int64
 5   number_emergency            101766 non-null  int64
 6   number_inpatient            101766 non-null  int64
 7   diag_1                      101766 non-null  int64
 8   diag_2                      101766 non-null  int64
 9   diag_3                      101766 non-null  int64
 10  number_diagnoses            101766 non-null  int64
 11  readmitted                  101766 non-null  category
```

```
12  race_Asian                    101766 non-null  int64
13  race_Caucasian                101766 non-null  int64
14  race_Hispanic                 101766 non-null  int64
15  race_Other                    101766 non-null  int64
16  gender_Male                   101766 non-null  int64
17  gender_Unknown/Invalid        101766 non-null  int64
18  age_[10-20)                   101766 non-null  int64
19  age_[20-30)                   101766 non-null  int64
20  age_[30-40)                   101766 non-null  int64
21  age_[40-50)                   101766 non-null  int64
22  age_[50-60)                   101766 non-null  int64
23  age_[60-70)                   101766 non-null  int64
24  age_[70-80)                   101766 non-null  int64
25  age_[80-90)                   101766 non-null  int64
26  age_[90-100)                  101766 non-null  int64
27  max_glu_serum_>300            101766 non-null  int64
28  max_glu_serum_No Test         101766 non-null  int64
29  max_glu_serum_Norm            101766 non-null  int64
30  A1Cresult_>8                  101766 non-null  int64
31  A1Cresult_No Test             101766 non-null  int64
32  A1Cresult_Norm                101766 non-null  int64
33  miglitol_No                   101766 non-null  int64
34  miglitol_Steady               101766 non-null  int64
35  miglitol_Up                   101766 non-null  int64
36  insulin_No                    101766 non-null  int64
37  insulin_Steady                101766 non-null  int64
38  insulin_Up                    101766 non-null  int64
39  change_No                     101766 non-null  int64
40  diabetesMed_Yes               101766 non-null  int64
41  admission_type_id_2           101766 non-null  int64
42  admission_type_id_3           101766 non-null  int64
43  admission_type_id_4           101766 non-null  int64
44  admission_type_id_5           101766 non-null  int64
45  admission_type_id_6           101766 non-null  int64
46  admission_type_id_7           101766 non-null  int64
47  admission_type_id_8           101766 non-null  int64
48  discharge_disposition_id_10   101766 non-null  int64
49  discharge_disposition_id_11   101766 non-null  int64
50  discharge_disposition_id_12   101766 non-null  int64
51  discharge_disposition_id_13   101766 non-null  int64
52  discharge_disposition_id_14   101766 non-null  int64
53  discharge_disposition_id_15   101766 non-null  int64
54  discharge_disposition_id_16   101766 non-null  int64
55  discharge_disposition_id_17   101766 non-null  int64
56  discharge_disposition_id_18   101766 non-null  int64
57  discharge_disposition_id_19   101766 non-null  int64
58  discharge_disposition_id_2    101766 non-null  int64
59  discharge_disposition_id_20   101766 non-null  int64
60  discharge_disposition_id_22   101766 non-null  int64
61  discharge_disposition_id_23   101766 non-null  int64
62  discharge_disposition_id_24   101766 non-null  int64
63  discharge_disposition_id_25   101766 non-null  int64
64  discharge_disposition_id_27   101766 non-null  int64
65  discharge_disposition_id_28   101766 non-null  int64
66  discharge_disposition_id_3    101766 non-null  int64
67  discharge_disposition_id_4    101766 non-null  int64
68  discharge_disposition_id_5    101766 non-null  int64
69  discharge_disposition_id_6    101766 non-null  int64
70  discharge_disposition_id_7    101766 non-null  int64
71  discharge_disposition_id_8    101766 non-null  int64
72  discharge_disposition_id_9    101766 non-null  int64
73  admission_source_id_10        101766 non-null  int64
74  admission_source_id_11        101766 non-null  int64
```

```
 75  admission_source_id_13       101766 non-null  int64
 76  admission_source_id_14       101766 non-null  int64
 77  admission_source_id_17       101766 non-null  int64
 78  admission_source_id_2        101766 non-null  int64
 79  admission_source_id_20       101766 non-null  int64
 80  admission_source_id_22       101766 non-null  int64
 81  admission_source_id_25       101766 non-null  int64
 82  admission_source_id_3        101766 non-null  int64
 83  admission_source_id_4        101766 non-null  int64
 84  admission_source_id_5        101766 non-null  int64
 85  admission_source_id_6        101766 non-null  int64
 86  admission_source_id_7        101766 non-null  int64
 87  admission_source_id_8        101766 non-null  int64
 88  admission_source_id_9        101766 non-null  int64
dtypes: category(1), int64(88)
memory usage: 68.4 MB
None
```

In [17]:
```
# Convert the target variable 'readmitted' into binary (1 for '<30', 0 fo
df['readmitted'] = df['readmitted'].apply(lambda x: 1 if x == '<30' else
```

In [18]:
```
# Separate the features (X) and the labels (y)
X = df.drop(columns=['readmitted'])
y = df['readmitted']
```

In [19]:
```
from sklearn.model_selection import train_test_split
# Perform the train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,

# Verify the split
print("Training set size:", X_train.shape)
print("Test set size:", X_test.shape)
```

```
Training set size: (71236, 88)
Test set size: (30530, 88)
```

In [20]:
```
from sklearn.preprocessing import StandardScaler

# Normalization stage
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [21]:
```
from imblearn.combine import SMOTETomek

# Resample the training dataset
smote_tomek = SMOTETomek(sampling_strategy='auto', random_state=42)
X_resampled, y_resampled = smote_tomek.fit_resample(X_train_scaled, y_tra

print("Original class distribution:", y_train.value_counts())
print("Resampled class distribution:", pd.Series(y_resampled).value_count
```

```
Original class distribution: readmitted
0    63304
1     7932
Name: count, dtype: int64
Resampled class distribution: readmitted
1    63230
0    63230
Name: count, dtype: int64
```

In [ ]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClas

# Initialize models
log_reg = LogisticRegression(max_iter=5000, solver='saga', class_weight='
decision_tree = DecisionTreeClassifier(class_weight='balanced')
random_forest = RandomForestClassifier(class_weight='balanced')
gradient_boosting = GradientBoostingClassifier()

# Fit models on resampled training data
log_reg.fit(X_resampled, y_resampled)
decision_tree.fit(X_resampled, y_resampled)
random_forest.fit(X_resampled, y_resampled)
gradient_boosting.fit(X_resampled, y_resampled)
```

Out[ ]:
```
▾ GradientBoostingClassifier
GradientBoostingClassifier()
```

In [ ]:
```
from sklearn.metrics import accuracy_score, precision_score, recall_score

models = {
    "Logistic Regression": log_reg,
    "Decision Tree": decision_tree,
    "Random Forest": random_forest,
    "Gradient Boosting": gradient_boosting
}

for model_name, model in models.items():
    y_pred = model.predict(X_test_scaled)
    print(f"\n{model_name} on Test Data:")
    print("Accuracy:", accuracy_score(y_test, y_pred))
    print("Precision:", precision_score(y_test, y_pred, pos_label=1))
    print("Recall:", recall_score(y_test, y_pred, pos_label=1))
    print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
    print("Classification Report:\n", classification_report(y_test, y_pre
```

```
Logistic Regression on Test Data:
Accuracy: 0.6512938093678349
Precision: 0.1737008585630366
Recall: 0.5611678832116789
Confusion Matrix:
 [[17962  9143]
 [ 1503  1922]]
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.66      0.77     27105
           1       0.17      0.56      0.27      3425

    accuracy                           0.65     30530
   macro avg       0.55      0.61      0.52     30530
weighted avg       0.84      0.65      0.71     30530


Decision Tree on Test Data:
Accuracy: 0.7933180478218146
Precision: 0.15232586165341047
Recall: 0.18452554744525548
Confusion Matrix:
 [[23588  3517]
 [ 2793   632]]
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.87      0.88     27105
           1       0.15      0.18      0.17      3425

    accuracy                           0.79     30530
   macro avg       0.52      0.53      0.52     30530
weighted avg       0.81      0.79      0.80     30530


Random Forest on Test Data:
Accuracy: 0.8869308876514903
Precision: 0.425414364640884
Recall: 0.022481751824817518
Confusion Matrix:
 [[27001   104]
 [ 3348    77]]
Classification Report:
              precision    recall  f1-score   support

           0       0.89      1.00      0.94     27105
           1       0.43      0.02      0.04      3425

    accuracy                           0.89     30530
   macro avg       0.66      0.51      0.49     30530
weighted avg       0.84      0.89      0.84     30530


Gradient Boosting on Test Data:
Accuracy: 0.8814281035047494
Precision: 0.2972972972972973
Recall: 0.04175182481751825
Confusion Matrix:
 [[26767   338]
 [ 3282   143]]
Classification Report:
              precision    recall  f1-score   support
```

```
              0       0.89      0.99      0.94     27105
              1       0.30      0.04      0.07      3425

       accuracy                           0.88     30530
      macro avg       0.59      0.51      0.50     30530
   weighted avg       0.82      0.88      0.84     30530
```

In [ ]:

```
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import accuracy_score, precision_score, recall_score

# Perform 10-fold cross-validation predictions for each model
y_pred_log_reg = cross_val_predict(log_reg, X_resampled, y_resampled, cv=
y_pred_decision_tree = cross_val_predict(decision_tree, X_resampled, y_re
y_pred_random_forest = cross_val_predict(random_forest, X_resampled, y_re
y_pred_gradient_boosting = cross_val_predict(gradient_boosting, X_resampl

# Calculate and print metrics for Logistic Regression
log_reg_accuracy = accuracy_score(y_resampled, y_pred_log_reg)
log_reg_precision = precision_score(y_resampled, y_pred_log_reg)
log_reg_recall = recall_score(y_resampled, y_pred_log_reg)
log_reg_conf_matrix = confusion_matrix(y_resampled, y_pred_log_reg)

print("Logistic Regression:")
print("Accuracy:", log_reg_accuracy)
print("Precision:", log_reg_precision)
print("Recall:", log_reg_recall)
print("Confusion Matrix:\n", log_reg_conf_matrix)

# Calculate and print metrics for Decision Tree
decision_tree_accuracy = accuracy_score(y_resampled, y_pred_decision_tree
decision_tree_precision = precision_score(y_resampled, y_pred_decision_tr
decision_tree_recall = recall_score(y_resampled, y_pred_decision_tree)
decision_tree_conf_matrix = confusion_matrix(y_resampled, y_pred_decision

print("\nDecision Tree:")
print("Accuracy:", decision_tree_accuracy)
print("Precision:", decision_tree_precision)
print("Recall:", decision_tree_recall)
print("Confusion Matrix:\n", decision_tree_conf_matrix)

# Calculate and print metrics for Random Forest
random_forest_accuracy = accuracy_score(y_resampled, y_pred_random_forest
random_forest_precision = precision_score(y_resampled, y_pred_random_fore
random_forest_recall = recall_score(y_resampled, y_pred_random_forest)
random_forest_conf_matrix = confusion_matrix(y_resampled, y_pred_random_f

print("\nRandom Forest:")
print("Accuracy:", random_forest_accuracy)
print("Precision:", random_forest_precision)
print("Recall:", random_forest_recall)
print("Confusion Matrix:\n", random_forest_conf_matrix)

# Calculate and print metrics for Gradient Boosting
gradient_boosting_accuracy = accuracy_score(y_resampled, y_pred_gradient_
gradient_boosting_precision = precision_score(y_resampled, y_pred_gradien
gradient_boosting_recall = recall_score(y_resampled, y_pred_gradient_boos
gradient_boosting_conf_matrix = confusion_matrix(y_resampled, y_pred_grad

print("\nGradient Boosting:")
print("Accuracy:", gradient_boosting_accuracy)
print("Precision:", gradient_boosting_precision)
```

```
print("Recall:", gradient_boosting_recall)
print("Confusion Matrix:\n", gradient_boosting_conf_matrix)
```

```
Logistic Regression:
Accuracy: 0.6194923295903844
Precision: 0.6334634611647912
Recall: 0.5671516685117823
Confusion Matrix:
 [[42480 20750]
 [27369 35861]]

Decision Tree:
Accuracy: 0.8765933892139807
Precision: 0.8704994554224366
Recall: 0.8848173335442037
Confusion Matrix:
 [[54907  8323]
 [ 7283 55947]]

Random Forest:
Accuracy: 0.9422267910801835
Precision: 0.9949201741654572
Recall: 0.8889925668195476
Confusion Matrix:
 [[62943   287]
 [ 7019 56211]]

Gradient Boosting:
Accuracy: 0.9127866519057409
Precision: 0.9869405421540642
Recall: 0.8366439981021667
Confusion Matrix:
 [[62530   700]
 [10329 52901]]
```

In [ ]:

```
from sklearn.model_selection import RandomizedSearchCV

# Define the parameter grid for each model with a narrower range or fewer
log_reg_params = {
    'solver': ['newton-cg', 'lbfgs', 'liblinear'],
    'C': [0.1, 1, 10]
}

decision_tree_params = {
    'max_depth': [10, 50],
    'min_samples_split': [2, 5],
    'min_samples_leaf': [1, 2]
}

random_forest_params = {
    'n_estimators': [100, 200],
    'max_depth': [10, 50],
    'min_samples_split': [2, 5]
}

gradient_boosting_params = {
    'n_estimators': [100, 200],
    'learning_rate': [0.01, 0.1],
    'max_depth': [3, 5]
}

# Perform randomized search for each model
log_reg_random = RandomizedSearchCV(LogisticRegression(), log_reg_params,
```

```
log_reg_random.fit(X_resampled, y_resampled)
print(f"Best Logistic Regression Parameters: {log_reg_random.best_params_

decision_tree_random = RandomizedSearchCV(DecisionTreeClassifier(), decis
decision_tree_random.fit(X_resampled, y_resampled)
print(f"Best Decision Tree Parameters: {decision_tree_random.best_params_

random_forest_random = RandomizedSearchCV(RandomForestClassifier(), rando
random_forest_random.fit(X_resampled, y_resampled)
print(f"Best Random Forest Parameters: {random_forest_random.best_params_

gradient_boosting_random = RandomizedSearchCV(GradientBoostingClassifier(
gradient_boosting_random.fit(X_resampled, y_resampled)
print(f"Best Gradient Boosting Parameters: {gradient_boosting_random.best
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.p
y:305: UserWarning: The total space of parameters 9 is smaller than n_ite
r=10. Running 9 iterations. For exhaustive searches, use GridSearchCV.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.p
y:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown i
n:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-re
gression
  n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.p
y:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown i
n:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-re
gression
  n_iter_i = _check_optimize_result(
Best Logistic Regression Parameters: {'solver': 'liblinear', 'C': 1}
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.p
y:305: UserWarning: The total space of parameters 8 is smaller than n_ite
r=10. Running 8 iterations. For exhaustive searches, use GridSearchCV.
  warnings.warn(
Best Decision Tree Parameters: {'min_samples_split': 5, 'min_samples_lea
f': 2, 'max_depth': 50}
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.p
y:305: UserWarning: The total space of parameters 8 is smaller than n_ite
r=10. Running 8 iterations. For exhaustive searches, use GridSearchCV.
  warnings.warn(
Best Random Forest Parameters: {'n_estimators': 200, 'min_samples_split':
2, 'max_depth': 50}
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.p
y:305: UserWarning: The total space of parameters 8 is smaller than n_ite
r=10. Running 8 iterations. For exhaustive searches, use GridSearchCV.
  warnings.warn(
Best Gradient Boosting Parameters: {'n_estimators': 200, 'max_depth': 5,
'learning_rate': 0.1}
```

In [22]:

```
# Convert to DataFrame for feature names consistency
X_resampled_df = pd.DataFrame(X_resampled, columns=X_train.columns)
X_test_df = pd.DataFrame(X_test_scaled, columns=X_train.columns)
# Print the first few rows of the DataFrames
print("First few rows of X_resampled_df:")
print(X_resampled_df.head())

print("\nFirst few rows of X_test_df:")
print(X_test_df.head())

# Print the column names of the DataFrames
print("\nColumn names in X_resampled_df:")
print(X_resampled_df.columns)

print("\nColumn names in X_test_df:")
print(X_test_df.columns)

# Check the shapes of the DataFrames
print("\nShape of X_resampled_df:", X_resampled_df.shape)
print("Shape of X_test_df:", X_test_df.shape)

# Ensure that column names match
print("\nDo column names match between X_resampled_df and X_train?")
print(X_resampled_df.columns.equals(X_train.columns))

print("\nDo column names match between X_test_df and X_train?")
print(X_test_df.columns.equals(X_train.columns))
```

```
First few rows of X_resampled_df:
   time_in_hospital  num_lab_procedures  num_procedures  num_medications
\
0          1.874229            0.096709       -0.784130        -0.246297
1         -0.133102           -0.207070       -0.784130        -0.615887
2         -1.136767           -0.308330        0.390684        -1.108675
3          1.205119            0.096709       -0.784130         0.985671
4          2.543340            0.400488       -0.784130        -0.492690

   number_outpatient  number_emergency  number_inpatient    diag_1    dia
g_2  \
0          -0.290147         -0.221346         -0.501837 -0.233059  0.335
502
1          -0.290147         -0.221346         -0.501837  2.398393  0.700
212
2          -0.290147         -0.221346         -0.501837  2.108090  1.900
055
3          -0.290147          0.904444          2.650621 -0.265835  0.906
352
4          -0.290147         -0.221346         -0.501837 -0.265835  0.340
788

     diag_3  ...  admission_source_id_20  admission_source_id_22  \
0  0.171779  ...               -0.040906               -0.012427
1 -0.606868  ...               -0.040906               -0.012427
2  0.812113  ...               -0.040906               -0.012427
3  0.166656  ...               -0.040906               -0.012427
4 -0.740057  ...               -0.040906               -0.012427

   admission_source_id_25  admission_source_id_3  admission_source_id_4
\
0               -0.005299              -0.042427              -0.181548
1               -0.005299              -0.042427              -0.181548
2               -0.005299              -0.042427              -0.181548
3               -0.005299              -0.042427              -0.181548
4               -0.005299              -0.042427              -0.181548

   admission_source_id_5  admission_source_id_6  admission_source_id_7  \
0              -0.091932              -0.151435               0.877004
1              -0.091932              -0.151435               0.877004
2              -0.091932              -0.151435              -1.140246
3              -0.091932              -0.151435               0.877004
4              -0.091932              -0.151435               0.877004

   admission_source_id_8  admission_source_id_9
0               -0.01351              -0.034767
1               -0.01351              -0.034767
2               -0.01351              -0.034767
3               -0.01351              -0.034767
4               -0.01351              -0.034767

[5 rows x 88 columns]

First few rows of X_test_df:
   time_in_hospital  num_lab_procedures  num_procedures  num_medications
\
0          2.208785            1.261196        -0.78413         0.492884
1         -1.136767           -1.169038        -0.78413        -1.108675
2         -0.133102           -1.118408         0.97809         0.862475
3          2.543340           -0.763999        -0.78413         0.369687
4         -1.136767           -1.118408        -0.78413        -1.231871

   number_outpatient  number_emergency  number_inpatient    diag_1    dia
```

```
g_2  \
0          -0.290147         -0.221346         -0.501837 -1.099284  0.906
352
1          -0.290147         -0.221346         -0.501837  1.382334 -0.002
780
2           0.497600         -0.221346          1.074392  1.077985  1.614
629
3          -0.290147         -0.221346          0.286277  0.043197 -0.795
627
4          -0.290147         -0.221346         -0.501837 -0.382893  0.827
067

     diag_3  ...  admission_source_id_20  admission_source_id_22  \
0 -0.673463  ...               -0.040906               -0.012427
1 -2.020726  ...               -0.040906               -0.012427
2  1.688091  ...               -0.040906               -0.012427
3 -1.421373  ...               -0.040906               -0.012427
4  0.171779  ...               -0.040906               -0.012427

   admission_source_id_25  admission_source_id_3  admission_source_id_4
\
0               -0.005299              -0.042427              -0.181548
1               -0.005299              -0.042427              -0.181548
2               -0.005299              -0.042427              -0.181548
3               -0.005299              -0.042427              -0.181548
4               -0.005299              -0.042427              -0.181548

   admission_source_id_5  admission_source_id_6  admission_source_id_7  \
0              -0.091932               6.603494              -1.140246
1              -0.091932              -0.151435              -1.140246
2              -0.091932              -0.151435              -1.140246
3              -0.091932              -0.151435              -1.140246
4              -0.091932              -0.151435               0.877004

   admission_source_id_8  admission_source_id_9
0               -0.01351              -0.034767
1               -0.01351              -0.034767
2               -0.01351              -0.034767
3               -0.01351              -0.034767
4               -0.01351              -0.034767

[5 rows x 88 columns]

Column names in X_resampled_df:
Index(['time_in_hospital', 'num_lab_procedures', 'num_procedures',
       'num_medications', 'number_outpatient', 'number_emergency',
       'number_inpatient', 'diag_1', 'diag_2', 'diag_3', 'number_diagnose
s',
       'race_Asian', 'race_Caucasian', 'race_Hispanic', 'race_Other',
       'gender_Male', 'gender_Unknown/Invalid', 'age_[10-20)', 'age_[20-3
0)',
       'age_[30-40)', 'age_[40-50)', 'age_[50-60)', 'age_[60-70)',
       'age_[70-80)', 'age_[80-90)', 'age_[90-100)', 'max_glu_serum_>30
0',
       'max_glu_serum_No Test', 'max_glu_serum_Norm', 'A1Cresult_>8',
       'A1Cresult_No Test', 'A1Cresult_Norm', 'miglitol_No', 'miglitol_St
eady',
       'miglitol_Up', 'insulin_No', 'insulin_Steady', 'insulin_Up',
       'change_No', 'diabetesMed_Yes', 'admission_type_id_2',
       'admission_type_id_3', 'admission_type_id_4', 'admission_type_id_
5',
       'admission_type_id_6', 'admission_type_id_7', 'admission_type_id_
8',
```

```
       'discharge_disposition_id_10', 'discharge_disposition_id_11',
       'discharge_disposition_id_12', 'discharge_disposition_id_13',
       'discharge_disposition_id_14', 'discharge_disposition_id_15',
       'discharge_disposition_id_16', 'discharge_disposition_id_17',
       'discharge_disposition_id_18', 'discharge_disposition_id_19',
       'discharge_disposition_id_2', 'discharge_disposition_id_20',
       'discharge_disposition_id_22', 'discharge_disposition_id_23',
       'discharge_disposition_id_24', 'discharge_disposition_id_25',
       'discharge_disposition_id_27', 'discharge_disposition_id_28',
       'discharge_disposition_id_3', 'discharge_disposition_id_4',
       'discharge_disposition_id_5', 'discharge_disposition_id_6',
       'discharge_disposition_id_7', 'discharge_disposition_id_8',
       'discharge_disposition_id_9', 'admission_source_id_10',
       'admission_source_id_11', 'admission_source_id_13',
       'admission_source_id_14', 'admission_source_id_17',
       'admission_source_id_2', 'admission_source_id_20',
       'admission_source_id_22', 'admission_source_id_25',
       'admission_source_id_3', 'admission_source_id_4',
       'admission_source_id_5', 'admission_source_id_6',
       'admission_source_id_7', 'admission_source_id_8',
       'admission_source_id_9'],
      dtype='object')

Column names in X_test_df:
Index(['time_in_hospital', 'num_lab_procedures', 'num_procedures',
       'num_medications', 'number_outpatient', 'number_emergency',
       'number_inpatient', 'diag_1', 'diag_2', 'diag_3', 'number_diagnose
s',
       'race_Asian', 'race_Caucasian', 'race_Hispanic', 'race_Other',
       'gender_Male', 'gender_Unknown/Invalid', 'age_[10-20)', 'age_[20-3
0)',
       'age_[30-40)', 'age_[40-50)', 'age_[50-60)', 'age_[60-70)',
       'age_[70-80)', 'age_[80-90)', 'age_[90-100)', 'max_glu_serum_>30
0',
       'max_glu_serum_No Test', 'max_glu_serum_Norm', 'A1Cresult_>8',
       'A1Cresult_No Test', 'A1Cresult_Norm', 'miglitol_No', 'miglitol_St
eady',
       'miglitol_Up', 'insulin_No', 'insulin_Steady', 'insulin_Up',
       'change_No', 'diabetesMed_Yes', 'admission_type_id_2',
       'admission_type_id_3', 'admission_type_id_4', 'admission_type_id_
5',
       'admission_type_id_6', 'admission_type_id_7', 'admission_type_id_
8',
       'discharge_disposition_id_10', 'discharge_disposition_id_11',
       'discharge_disposition_id_12', 'discharge_disposition_id_13',
       'discharge_disposition_id_14', 'discharge_disposition_id_15',
       'discharge_disposition_id_16', 'discharge_disposition_id_17',
       'discharge_disposition_id_18', 'discharge_disposition_id_19',
       'discharge_disposition_id_2', 'discharge_disposition_id_20',
       'discharge_disposition_id_22', 'discharge_disposition_id_23',
       'discharge_disposition_id_24', 'discharge_disposition_id_25',
       'discharge_disposition_id_27', 'discharge_disposition_id_28',
       'discharge_disposition_id_3', 'discharge_disposition_id_4',
       'discharge_disposition_id_5', 'discharge_disposition_id_6',
       'discharge_disposition_id_7', 'discharge_disposition_id_8',
       'discharge_disposition_id_9', 'admission_source_id_10',
       'admission_source_id_11', 'admission_source_id_13',
       'admission_source_id_14', 'admission_source_id_17',
       'admission_source_id_2', 'admission_source_id_20',
       'admission_source_id_22', 'admission_source_id_25',
       'admission_source_id_3', 'admission_source_id_4',
       'admission_source_id_5', 'admission_source_id_6',
       'admission_source_id_7', 'admission_source_id_8',
```

```
       'admission_source_id_9'],
      dtype='object')

Shape of X_resampled_df: (126460, 88)
Shape of X_test_df: (30530, 88)

Do column names match between X_resampled_df and X_train?
True

Do column names match between X_test_df and X_train?
True
```

In [23]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClas
from sklearn.metrics import classification_report, confusion_matrix

# Re-train Logistic Regression
log_reg_best = LogisticRegression(solver='liblinear', C=1)
log_reg_best.fit(X_resampled_df, y_resampled)
y_pred_log_reg_best = log_reg_best.predict(X_test_df)
print("Logistic Regression (Best Params) Classification Report:")
print(classification_report(y_test, y_pred_log_reg_best))
print("Logistic Regression (Best Params) Confusion Matrix:")
print(confusion_matrix(y_test, y_pred_log_reg_best))

# Re-train Decision Tree
decision_tree_best = DecisionTreeClassifier(min_samples_split=5, min_samp
decision_tree_best.fit(X_resampled_df, y_resampled)
y_pred_decision_tree_best = decision_tree_best.predict(X_test_df)
print("\nDecision Tree (Best Params) Classification Report:")
print(classification_report(y_test, y_pred_decision_tree_best))
print("Decision Tree (Best Params) Confusion Matrix:")
print(confusion_matrix(y_test, y_pred_decision_tree_best))

# Re-train Random Forest
random_forest_best = RandomForestClassifier(n_estimators=200, min_samples
random_forest_best.fit(X_resampled_df, y_resampled)
y_pred_random_forest_best = random_forest_best.predict(X_test_df)
print("\nRandom Forest (Best Params) Classification Report:")
print(classification_report(y_test, y_pred_random_forest_best))
print("Random Forest (Best Params) Confusion Matrix:")
print(confusion_matrix(y_test, y_pred_random_forest_best))

# Re-train Gradient Boosting
gradient_boosting_best = GradientBoostingClassifier(n_estimators=200, max
gradient_boosting_best.fit(X_resampled_df, y_resampled)
y_pred_gradient_boosting_best = gradient_boosting_best.predict(X_test_df)
print("\nGradient Boosting (Best Params) Classification Report:")
print(classification_report(y_test, y_pred_gradient_boosting_best))
print("Gradient Boosting (Best Params) Confusion Matrix:")
print(confusion_matrix(y_test, y_pred_gradient_boosting_best))
```

```
Logistic Regression (Best Params) Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.66      0.77     27105
           1       0.17      0.56      0.27      3425

    accuracy                           0.65     30530
   macro avg       0.55      0.61      0.52     30530
weighted avg       0.84      0.65      0.71     30530

Logistic Regression (Best Params) Confusion Matrix:
[[17961  9144]
 [ 1502  1923]]

Decision Tree (Best Params) Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.91      0.90     27105
           1       0.16      0.14      0.15      3425

    accuracy                           0.82     30530
   macro avg       0.53      0.52      0.52     30530
weighted avg       0.81      0.82      0.82     30530

Decision Tree (Best Params) Confusion Matrix:
[[24553  2552]
 [ 2944   481]]

Random Forest (Best Params) Classification Report:
              precision    recall  f1-score   support

           0       0.89      1.00      0.94     27105
           1       0.48      0.02      0.04      3425

    accuracy                           0.89     30530
   macro avg       0.68      0.51      0.49     30530
weighted avg       0.84      0.89      0.84     30530

Random Forest (Best Params) Confusion Matrix:
[[27020    85]
 [ 3348    77]]

Gradient Boosting (Best Params) Classification Report:
              precision    recall  f1-score   support

           0       0.89      1.00      0.94     27105
           1       0.55      0.03      0.05      3425

    accuracy                           0.89     30530
   macro avg       0.72      0.51      0.50     30530
weighted avg       0.85      0.89      0.84     30530

Gradient Boosting (Best Params) Confusion Matrix:
[[27029    76]
 [ 3331    94]]
```

In [24]:

```python
from sklearn.feature_selection import RFE

# Initialize the model for RFE
rfe_model = LogisticRegression(max_iter=5000, solver='liblinear', class_w

# Initialize RFE
rfe = RFE(estimator=rfe_model, n_features_to_select=10)
```

```
# Fit RFE
rfe.fit(X_resampled_df, y_resampled)

# Get the ranking of features
feature_ranking = rfe.ranking_
selected_features = X_train.columns[rfe.support_]

print("Selected Features by RFE:", selected_features)
print("Feature Ranking:", feature_ranking)
```

```
Selected Features by RFE: Index(['number_inpatient', 'number_diagnoses',
'age_[70-80)', 'age_[80-90)',
       'admission_type_id_7', 'discharge_disposition_id_11',
       'discharge_disposition_id_12', 'discharge_disposition_id_16',
       'discharge_disposition_id_17', 'discharge_disposition_id_22'],
      dtype='object')
Feature Ranking: [56  9 18 39 32 49  1 15 41 64  1 59 55 79 63 60 47 30
7  4  6  3  2  1
  1  5 73 45 54 12  8 13 62 61 44 20 21 22 31 10 27 28 76 40 65  1 71 23
  1  1 11 29 37  1  1 19 17 24 42  1 38 74 68 34 25 16 58 14 48 69 57 46
 26 33 43 36 51 78 75 66 35 77 53 50 72 52 70 67]
```

In [28]:

```
from sklearn.ensemble import RandomForestClassifier
import numpy as np
import pandas as pd

# Initialize RandomForestClassifier
random_forest = RandomForestClassifier()

# Fit Random Forest
random_forest.fit(X_resampled_df, y_resampled)

# Get feature importances
importances = random_forest.feature_importances_
indices = np.argsort(importances)[::-1]

# Print the feature ranking
print("Feature ranking:")

for f in range(X_resampled_df.shape[1]):
    print("%d. feature %s (%f)" % (f + 1, X_resampled_df.columns[indices[

# Select top features
top_features = X_resampled_df.columns[indices[:10]]  # Select top 10 feat
print("Top Features:", top_features)
```

Feature ranking:
1. feature number_inpatient (0.135676)
2. feature num_procedures (0.076402)
3. feature time_in_hospital (0.074801)
4. feature diag_3 (0.070299)
5. feature number_diagnoses (0.059890)
6. feature diag_2 (0.058606)
7. feature diag_1 (0.055265)
8. feature num_medications (0.053535)
9. feature num_lab_procedures (0.048763)
10. feature gender_Male (0.042827)
11. feature number_outpatient (0.031861)
12. feature number_emergency (0.028954)
13. feature change_No (0.024881)
14. feature race_Caucasian (0.019020)
15. feature admission_source_id_7 (0.018122)
16. feature insulin_No (0.015137)
17. feature insulin_Steady (0.014735)
18. feature diabetesMed_Yes (0.012537)
19. feature admission_type_id_2 (0.011842)
20. feature age_[70-80) (0.011623)
21. feature age_[60-70) (0.009930)
22. feature discharge_disposition_id_3 (0.009853)
23. feature A1Cresult_No Test (0.009793)
24. feature age_[50-60) (0.008708)
25. feature discharge_disposition_id_6 (0.008624)
26. feature age_[80-90) (0.008531)
27. feature admission_type_id_3 (0.007803)
28. feature discharge_disposition_id_22 (0.007749)
29. feature insulin_Up (0.006039)
30. feature age_[40-50) (0.005328)
31. feature A1Cresult_>8 (0.003962)
32. feature discharge_disposition_id_11 (0.003688)
33. feature discharge_disposition_id_18 (0.003487)
34. feature A1Cresult_Norm (0.003028)
35. feature discharge_disposition_id_2 (0.002781)
36. feature admission_type_id_6 (0.002732)
37. feature age_[30-40) (0.002722)
38. feature admission_source_id_17 (0.002599)
39. feature age_[90-100) (0.002414)
40. feature admission_type_id_5 (0.002251)
41. feature max_glu_serum_No Test (0.002119)
42. feature discharge_disposition_id_5 (0.002070)
43. feature admission_source_id_4 (0.002051)
44. feature age_[20-30) (0.001792)
45. feature admission_source_id_6 (0.001612)
46. feature max_glu_serum_Norm (0.001489)
47. feature race_Hispanic (0.001486)
48. feature race_Other (0.001196)
49. feature admission_source_id_2 (0.001003)
50. feature discharge_disposition_id_4 (0.000949)
51. feature max_glu_serum_>300 (0.000910)
52. feature admission_source_id_5 (0.000839)
53. feature race_Asian (0.000687)
54. feature discharge_disposition_id_7 (0.000664)
55. feature discharge_disposition_id_25 (0.000653)
56. feature discharge_disposition_id_28 (0.000576)
57. feature admission_type_id_8 (0.000462)
58. feature age_[10-20) (0.000366)
59. feature discharge_disposition_id_13 (0.000360)
60. feature discharge_disposition_id_23 (0.000344)
61. feature discharge_disposition_id_14 (0.000339)
62. feature discharge_disposition_id_15 (0.000251)

```
63. feature admission_source_id_20 (0.000200)
64. feature admission_source_id_3 (0.000185)
65. feature discharge_disposition_id_8 (0.000129)
66. feature admission_source_id_9 (0.000105)
67. feature discharge_disposition_id_9 (0.000091)
68. feature discharge_disposition_id_24 (0.000059)
69. feature discharge_disposition_id_12 (0.000049)
70. feature miglitol_No (0.000035)
71. feature admission_source_id_22 (0.000033)
72. feature admission_source_id_8 (0.000028)
73. feature miglitol_Steady (0.000021)
74. feature admission_type_id_4 (0.000020)
75. feature admission_type_id_7 (0.000009)
76. feature discharge_disposition_id_16 (0.000006)
77. feature discharge_disposition_id_17 (0.000006)
78. feature discharge_disposition_id_19 (0.000003)
79. feature admission_source_id_11 (0.000003)
80. feature discharge_disposition_id_10 (0.000001)
81. feature discharge_disposition_id_27 (0.000001)
82. feature admission_source_id_13 (0.000000)
83. feature admission_source_id_10 (0.000000)
84. feature miglitol_Up (0.000000)
85. feature admission_source_id_25 (0.000000)
86. feature discharge_disposition_id_20 (0.000000)
87. feature gender_Unknown/Invalid (0.000000)
88. feature admission_source_id_14 (0.000000)
Top Features: Index(['number_inpatient', 'num_procedures', 'time_in_hospi
tal', 'diag_3',
       'number_diagnoses', 'diag_2', 'diag_1', 'num_medications',
       'num_lab_procedures', 'gender_Male'],
      dtype='object')
```

In [30]:

```python
from sklearn.metrics import accuracy_score, precision_score, recall_score
# Select top features
top_features = X_resampled_df.columns[indices[:10]]

# Filter the dataset to include only the top features
X_train_selected = X_train[top_features]
X_test_selected = X_test[top_features]

# Resample the selected features
X_resampled_selected, y_resampled_selected = smote_tomek.fit_resample(X_t

# Initialize and train models with the selected features
log_reg = LogisticRegression(max_iter=5000, solver='saga', class_weight='
decision_tree = DecisionTreeClassifier(class_weight='balanced')
random_forest = RandomForestClassifier(class_weight='balanced')
gradient_boosting = GradientBoostingClassifier()

log_reg.fit(X_resampled_selected, y_resampled_selected)
decision_tree.fit(X_resampled_selected, y_resampled_selected)
random_forest.fit(X_resampled_selected, y_resampled_selected)
gradient_boosting.fit(X_resampled_selected, y_resampled_selected)

# Evaluate models
models = {
    "Logistic Regression": log_reg,
    "Decision Tree": decision_tree,
    "Random Forest": random_forest,
    "Gradient Boosting": gradient_boosting
}

for model_name, model in models.items():
```

```
    y_pred = model.predict(X_test_selected)
    print(f"\n{model_name} on Test Data:")
    print("Accuracy:", accuracy_score(y_test, y_pred))
    print("Precision:", precision_score(y_test, y_pred, pos_label=1))
    print("Recall:", recall_score(y_test, y_pred, pos_label=1))
    print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
    print("Classification Report:\n", classification_report(y_test, y_pre
```

```
Logistic Regression on Test Data:
Accuracy: 0.551850638716017
Precision: 0.12893423051877578
Recall: 0.5202919708029197
Confusion Matrix:
 [[15066 12039]
 [ 1643  1782]]
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.56      0.69     27105
           1       0.13      0.52      0.21      3425

    accuracy                           0.55     30530
   macro avg       0.52      0.54      0.45     30530
weighted avg       0.81      0.55      0.63     30530


Decision Tree on Test Data:
Accuracy: 0.6825417622011136
Precision: 0.12522425547183352
Recall: 0.30569343065693433
Confusion Matrix:
 [[19791  7314]
 [ 2378  1047]]
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.73      0.80     27105
           1       0.13      0.31      0.18      3425

    accuracy                           0.68     30530
   macro avg       0.51      0.52      0.49     30530
weighted avg       0.81      0.68      0.73     30530


Random Forest on Test Data:
Accuracy: 0.7760890926957091
Precision: 0.1515832482124617
Recall: 0.21664233576642336
Confusion Matrix:
 [[22952  4153]
 [ 2683   742]]
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.85      0.87     27105
           1       0.15      0.22      0.18      3425

    accuracy                           0.78     30530
   macro avg       0.52      0.53      0.52     30530
weighted avg       0.81      0.78      0.79     30530


Gradient Boosting on Test Data:
Accuracy: 0.6376678676711431
Precision: 0.14047641465022126
Recall: 0.43562043795620436
Confusion Matrix:
 [[17976  9129]
 [ 1933  1492]]
Classification Report:
              precision    recall  f1-score   support
```

0	0.90	0.66	0.76	27105
1	0.14	0.44	0.21	3425
accuracy			0.64	30530
macro avg	0.52	0.55	0.49	30530
weighted avg	0.82	0.64	0.70	30530