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

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

```python
#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) (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 [ ]: #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 [ ]:
```python
#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 [ ]:
```
#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 diabetesM
ed  \
count                    101766                  101766  101766      1017
66
unique                        2                       2       2
2
top                          No                      No      No         Y
es
freq                     101764                  101765   54755       783
```

```
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 [ ]:

```
#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 [ ]:

```
# 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 [ ]:
```
# 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 [ ]: # Check for missing values
        missing_values = df.isnull().sum()
        print(missing_values)
```

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

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

In [ ]:
```
# 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 [ ]:
```
# 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 [ ]:
```
# 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 [ ]:
```
# 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 [ ]:
```
# 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 [ ]:
```
# 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 [ ]:

```
# 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 [ ]:
```
# 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 [ ]:
```
# To assess the normality of the numerical variables, I conducted the Sha
# Despite the test results indicating that none of the variables followed
# I chose to retain the outliers. The reason behind this decision was tha
# across the numeric variables was not substantial, and removing them cou
# the variance and valuable information in the dataset.
from scipy.stats import shapiro

# Copy the DataFrame to avoid modifying the original
df_clustering = df.copy()

# List of numeric columns
numeric_columns = ['time_in_hospital', 'num_lab_procedures', 'num_procedu
                   'number_outpatient', 'number_emergency', 'number_inpat
                   'diag_1', 'diag_2', 'diag_3']

# Dropping the class label 'readmitted' for Shapiro-Wilk test
data_for_shapiro = df_clustering[numeric_columns]

# Run Shapiro-Wilk test on numeric columns to check for normality
shapiro_results = {}
for column in numeric_columns:
    stat, p = shapiro(data_for_shapiro[column])
    shapiro_results[column] = {'statistic': stat, 'p_value': p}

# Display Shapiro-Wilk test results
print("Shapiro-Wilk Test Results:")
for column, result in shapiro_results.items():
    print(f"Column: {column}, Statistic: {result['statistic']}, p-value:
```

```
Shapiro-Wilk Test Results:
Column: time_in_hospital, Statistic: 0.8869192600250244, p-value: 0.0
Column: num_lab_procedures, Statistic: 0.9848576188087463, p-value: 0.0
Column: num_procedures, Statistic: 0.7742846012115479, p-value: 0.0
Column: num_medications, Statistic: 0.9244745969772339, p-value: 0.0
Column: number_outpatient, Statistic: 0.3123285174369812, p-value: 0.0
Column: number_emergency, Statistic: 0.19770395755767822, p-value: 0.0
Column: number_inpatient, Statistic: 0.5557276010513306, p-value: 0.0
Column: diag_1, Statistic: 0.9683600068092346, p-value: 0.0
Column: diag_2, Statistic: 0.9584629535675049, p-value: 0.0
Column: diag_3, Statistic: 0.9462312459945679, p-value: 0.0
```

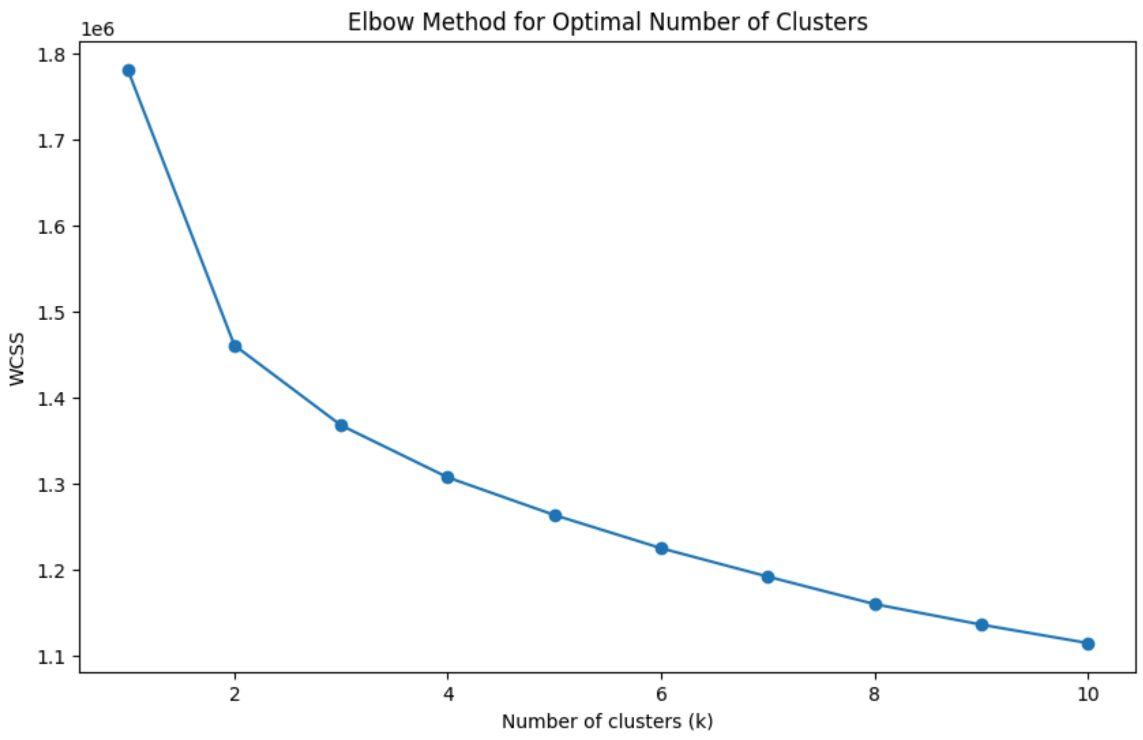
```
/usr/local/lib/python3.10/dist-packages/scipy/stats/_morestats.py:1882: U
serWarning: p-value may not be accurate for N > 5000.
  warnings.warn("p-value may not be accurate for N > 5000.")
```

In [ ]:

```
from sklearn.preprocessing import StandardScaler

# Normalize numeric columns
scaler = StandardScaler()
df_clustering[numeric_columns] = scaler.fit_transform(df_clustering[numer

# Verify normalization
print("Normalized Data:")
print(pd.DataFrame(df_clustering[numeric_columns], columns=numeric_column
```

```
Normalized Data:
       time_in_hospital  num_lab_procedures  num_procedures  num_medicati
ons  \
count      1.017660e+05        1.017660e+05    1.017660e+05      1.017660e
+05
mean       5.082986e-17        1.111554e-16   -1.717602e-17     -1.323811e
-16
std        1.000005e+00        1.000005e+00    1.000005e+00      1.000005e
+00
min       -1.137649e+00       -2.139630e+00   -7.853977e-01     -1.848268e
+00
25%       -8.026506e-01       -6.147950e-01   -7.853977e-01     -7.409197e
-01
50%       -1.326548e-01        4.596660e-02   -1.991621e-01     -1.257264e
-01
75%        5.373411e-01        7.067282e-01    3.870736e-01      4.894670e
-01
max        3.217324e+00        4.518815e+00    2.732016e+00      7.994826e
+00

       number_outpatient  number_emergency  number_inpatient        diag_
1  \
count       1.017660e+05      1.017660e+05      1.017660e+05  1.017660e+0
5
mean        2.010851e-17      3.044206e-17     -2.115583e-17 -1.273365e-1
6
std         1.000005e+00      1.000005e+00      1.000005e+00  1.000005e+0
0
min        -2.914615e-01     -2.126202e-01     -5.032762e-01 -2.265768e+0
0
25%        -2.914615e-01     -2.126202e-01     -5.032762e-01 -3.805559e-0
1
50%        -2.914615e-01     -2.126202e-01     -5.032762e-01 -2.358985e-0
1
75%        -2.914615e-01     -2.126202e-01      2.885790e-01  5.060539e-0
1
max         3.285094e+01      8.146673e+01      1.612568e+01  2.395933e+0
0

             diag_2        diag_3
count  1.017660e+05  1.017660e+05
mean  -3.239705e-17 -3.072134e-17
std    1.000005e+00  1.000005e+00
min   -2.255769e+00 -2.026482e+00
25%   -7.982336e-01 -7.439787e-01
50%   -1.665646e-02  3.065351e-02
75%    4.797506e-01  5.064624e-01
max    3.019876e+00  3.098402e+00
```

In [ ]:
```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Prepare the data for clustering (excluding the 'readmitted' column)
clustering_data = df_clustering.drop(columns=['readmitted'])

# Calculate WCSS for different numbers of clusters
wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=50)
    kmeans.fit(clustering_data)
    wcss.append(kmeans.inertia_)

# Visualize the Elbow Method
plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), wcss, marker='o')
plt.title('Elbow Method for Optimal Number of Clusters')
plt.xlabel('Number of clusters (k)')
plt.ylabel('WCSS')
plt.show()
```



In [ ]:
```
# Verify that clustering_data contains all the columns
print("Columns in clustering_data:")
print(clustering_data.columns)
# Get the count of columns
column_count = len(clustering_data.columns)
print(f"Number of columns in clustering_data: {column_count}")
```

```
Columns in clustering_data:
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')
Number of columns in clustering_data: 88
```

In [ ]:

```python
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import pandas as pd

# Perform K-Means clustering with the optimal number of clusters (K=4)
optimal_clusters = 4  # Based on the Elbow Method
kmeans = KMeans(n_clusters=optimal_clusters, n_init=50, random_state=42)
kmeans.fit(clustering_data)

# Add the cluster labels to the DataFrame
df_clustering['Cluster'] = kmeans.labels_

# Print the DataFrame after adding the cluster labels
print(df_clustering)

# Calculate the silhouette score to evaluate the clustering performance
silhouette_avg = silhouette_score(clustering_data, kmeans.labels_)
print(f'Silhouette Score: {silhouette_avg}')
```

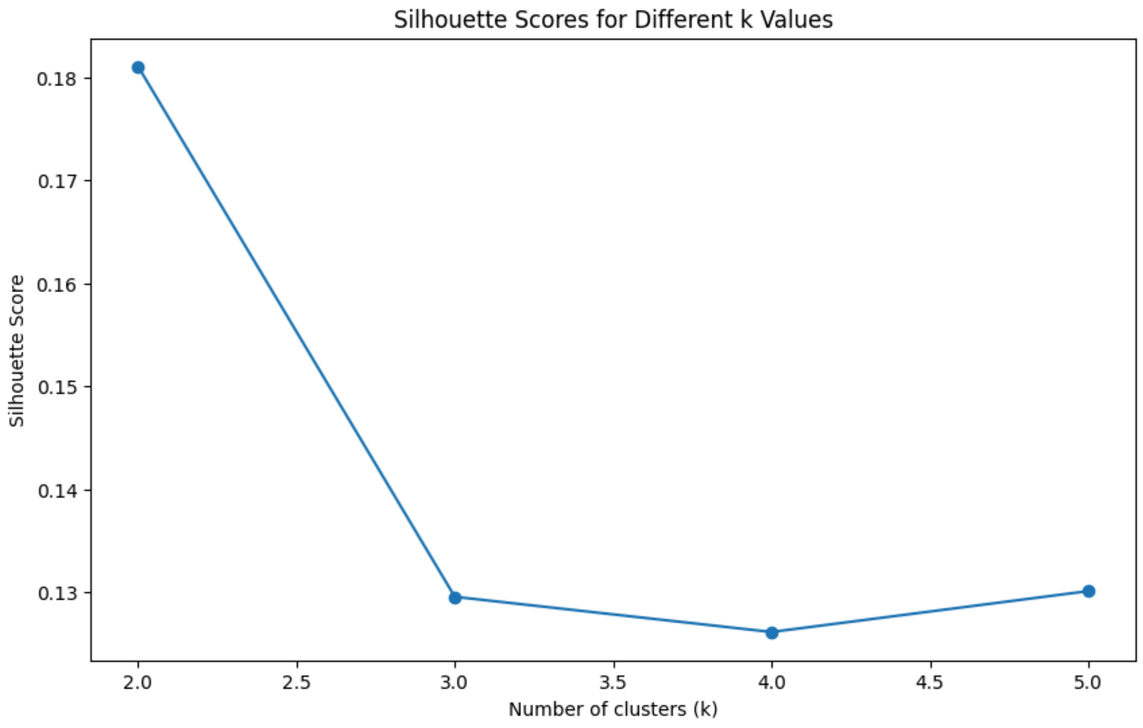
```
        time_in_hospital  num_lab_procedures  num_procedures  num_medicat
ions  \
0              -1.137649           -0.106517       -0.785398        -1.84
8268
1              -0.467653            0.808384       -0.785398         0.24
3390
2              -0.802651           -1.631351        2.145781        -0.37
1804
3              -0.802651            0.045967       -0.199162        -0.00
2688
4              -1.137649            0.401761       -0.785398        -0.98
6997
...                  ...                 ...             ...
...
101761         -0.467653            0.401761       -0.785398        -0.00
2688
101762          0.202343           -0.513139        0.973309         0.24
3390
101763         -1.137649            0.503417       -0.785398        -0.86
3958
101764          1.877333            0.096794        0.387074         0.61
2506
101765          0.537341           -1.529696        0.973309        -1.60
2190

        number_outpatient  number_emergency  number_inpatient    diag_1
\
0               -0.291461          -0.21262         -0.503276 -1.099176
1               -0.291461          -0.21262         -0.503276 -0.977851
2                1.286748          -0.21262          0.288579  0.758038
3               -0.291461          -0.21262         -0.503276 -2.228437
4               -0.291461          -0.21262         -0.503276 -1.346494
...                   ...               ...               ...       ...
101761          -0.291461          -0.21262         -0.503276 -1.099176
101762          -0.291461          -0.21262          0.288579  0.347397
101763           0.497643          -0.21262         -0.503276 -2.088446
101764          -0.291461          -0.21262          0.288579  2.381934
101765          -0.291461          -0.21262         -0.503276  0.207406

          diag_2    diag_3  ...  admission_source_id_22  \
0      -0.798234 -0.743979  ...                       0
1      -0.935538 -0.718329  ...                       0
2      -0.935538 -2.026482  ...                       0
3      -0.935538  0.040914  ...                       0
4      -1.426664 -0.743979  ...                       0
...          ...       ...  ...                     ...
101761 -0.719020  0.323064  ...                       0
101762 -0.798234  2.010839  ...                       0
101763  0.859977 -0.507998  ...                       0
101764 -0.750705  3.093272  ...                       0
101765  0.543122  2.010839  ...                       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
...                            ...                     ...
...
101761                           0                       0
0
101762                           0                       0
0
101763                           0                       0
0
101764                           0                       0
0
101765                           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
...                       ...                     ...
...
101761                      0                       0
1
101762                      1                       0
0
101763                      0                       0
1
101764                      0                       0
1
101765                      0                       0
1

        admission_source_id_8  admission_source_id_9  Cluster
0                           0                       0        2
1                           0                       0        3
2                           0                       0        2
3                           0                       0        3
4                           0                       0        2
...                       ...                     ...      ...
101761                      0                       0        3
101762                      0                       0        3
101763                      0                       0        3
101764                      0                       0        1
101765                      0                       0        3

[101766 rows x 90 columns]
Silhouette Score: 0.12615024043888765
```

In [ ]:

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

# Function to evaluate and plot silhouette scores for different k values
def evaluate_k_means(data, k_values, n_init=50, random_state=42):
    silhouette_scores = []
    for k in k_values:
        kmeans = KMeans(n_clusters=k, random_state=random_state, n_init=n
        kmeans.fit(data)
        score = silhouette_score(data, kmeans.labels_)
        silhouette_scores.append(score)
        print(f'For n_clusters = {k}, Silhouette Score = {score}')

    plt.figure(figsize=(10, 6))
    plt.plot(k_values, silhouette_scores, marker='o')
    plt.title('Silhouette Scores for Different k Values')
    plt.xlabel('Number of clusters (k)')
    plt.ylabel('Silhouette Score')
    plt.show()

# Define range of k values to evaluate
k_values = range(2, 6)

# Evaluate k-means clustering
evaluate_k_means(clustering_data, k_values)
```

```
For n_clusters = 2, Silhouette Score = 0.18104549510866558
For n_clusters = 3, Silhouette Score = 0.12956704345897635
For n_clusters = 4, Silhouette Score = 0.12615024043888765
For n_clusters = 5, Silhouette Score = 0.13011944653549468
```



In [ ]:
```
# Get the cluster centers
centers = kmeans.cluster_centers_
centers_df = pd.DataFrame(centers, columns=clustering_data.columns)

print("\nCluster Centers:")
print(centers_df)
```

```
Cluster Centers:
   time_in_hospital  num_lab_procedures  num_procedures  num_medications
\
0         -0.124821            0.074534       -0.283150        -0.283080
1         -0.493989           -1.084364       -0.405609        -0.419612
2          0.101573            0.653746       -0.439452         0.037352
3          1.747294            0.868241        1.087644         1.752257
4         -0.307474           -0.224819        1.877981         0.189714
5         -0.565761           -0.485532       -0.080871        -0.558479
6         -0.020726            0.059744       -0.171299         0.187226
7          0.189131            0.184523       -0.246129         0.199210
8         -0.002749           -0.014285       -0.224957        -0.113821
9         -0.148754           -0.061326       -0.288018        -0.023839

   number_outpatient  number_emergency  number_inpatient    diag_1    dia
g_2  \
0          -0.161802         -0.118401         -0.185763 -0.000730 -0.024
152
1          -0.057230         -0.043332         -0.176297  0.312814 -0.051
197
2          -0.105205         -0.047106         -0.145427 -0.108873 -0.031
963
3          -0.114240         -0.081855         -0.096228 -0.190869  0.263
911
4          -0.134249         -0.113021         -0.260529 -0.031398  0.133
717
5          -0.152540         -0.107597         -0.258944  0.037006 -0.481
932
6           4.866870          0.222834          0.260631 -0.048029  0.135
032
7           0.125006          0.470208          3.055249 -0.118085  0.032
696
8          -0.085547         -0.050226         -0.137136  0.007271  0.449
355
9           0.440758          8.883833          2.744648 -0.126580 -0.071
193

     diag_3  ...  admission_source_id_20  admission_source_id_22  \
0 -0.154916  ...            9.067481e-04            4.772358e-05
1 -0.325354  ...            1.719937e-03            1.433281e-04
2 -0.256922  ...            1.813659e-03            2.072754e-04
3  0.261574  ...            3.039195e-03            2.095997e-04
4 -0.024582  ...            4.113111e-03            1.285347e-04
5 -0.436995  ...            3.773015e-04            3.876023e-18
6  0.163059  ...           -1.691355e-17            1.138412e-18
7  0.033645  ...            8.259343e-04            2.064836e-04
8  1.661158  ...            1.371742e-03            1.055186e-04
9 -0.040067  ...            1.734723e-18           -5.827587e-19

   admission_source_id_25  admission_source_id_3  admission_source_id_4
\
0            8.131516e-20           2.147561e-03               0.028682
1            2.093865e-18           9.316325e-04               0.021929
2           -5.454892e-19           8.809203e-04               0.016271
3            1.047998e-04           2.515196e-03               0.060050
4            8.402567e-19           2.442159e-03               0.077378
5            3.184844e-19           3.697555e-03               0.034033
6           -2.473336e-19           1.149254e-17               0.007529
7           -3.828589e-19           1.032418e-03               0.020442
8            1.055186e-04           1.582779e-03               0.023531
9            1.084202e-19           9.107298e-18               0.003697

   admission_source_id_5  admission_source_id_6  admission_source_id_7  \
0              0.006538          6.762432e-02                0.540326
1              0.004443          3.869858e-03                0.505661
2              0.013473          7.410094e-03                0.755726
3              0.013624          7.440788e-03                0.521065
4              0.005784          3.084833e-03                0.389846
5              0.002188          3.471174e-02                0.431407
6              0.004235          6.117647e-03                0.573647
7              0.012183          7.020442e-03                0.697089
8              0.012873          5.064894e-03                0.608737
9              0.003697          1.387779e-17                0.824399

   admission_source_id_8  admission_source_id_9
0           2.386179e-04           3.531545e-03
1           7.166404e-05           2.866562e-04
2           5.181884e-05          -2.474149e-16
3           2.095997e-04           1.047998e-04
4           6.722053e-18          -2.905662e-17
5           3.773015e-04           3.395714e-03
6          -1.978669e-18           2.797242e-17
7          -3.062871e-18           3.989864e-17
8           1.055186e-04           1.055186e-04
9           1.848429e-03          -4.987330e-18

[10 rows x 88 columns]
```

In [ ]:

```python
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import pandas as pd
# Perform K-Means clustering with the optimal number of clusters (K=2) si
optimal_clusters = 2
kmeans = KMeans(n_clusters=optimal_clusters, n_init=50, random_state=42)
kmeans.fit(clustering_data)

# Add the cluster labels to the clustering_data DataFrame
clustering_data['Cluster'] = kmeans.labels_

# Calculate the mean values for each feature (including one-hot encoded c
cluster_means = clustering_data.groupby('Cluster').mean()

print("\nMean values for each feature within each cluster:")
print(cluster_means)

# Calculate the silhouette score
silhouette_avg = silhouette_score(clustering_data.drop(columns=['Cluster'
print(f"\nSilhouette Score for K={optimal_clusters}: {silhouette_avg}")

# Optional: Save the clustering results
clustering_data.to_csv('clustering_results.csv', index=False)
```

```
Mean values for each feature within each cluster:
         time_in_hospital  num_lab_procedures  num_procedures  \
Cluster
0               -0.332139           -0.236397       -0.115492
1                0.178380            0.126960        0.062027

         num_medications  number_outpatient  number_emergency  \
Cluster
0              -0.386925          -0.126819         -0.079964
1               0.207804           0.068110          0.042946

         number_inpatient    diag_1    diag_2    diag_3  ...  \
Cluster                                                  ...
0               -0.150618  0.020951 -0.215357 -0.264754  ...
1                0.080892 -0.011252  0.115661  0.142190  ...

         admission_source_id_20  admission_source_id_22  \
Cluster
0                      0.000562                0.000028
1                      0.002130                0.000166

         admission_source_id_25  admission_source_id_3  admission_source_
id_4  \
Cluster
0                       0.00000               0.002840                0.03
3185
1                       0.00003               0.001299                0.03
0314

         admission_source_id_5  admission_source_id_6  admission_source_i
d_7  \
Cluster
0                     0.003712               0.053490                0.478
570
1                     0.010920               0.005468                0.611
361

         admission_source_id_8  admission_source_id_9
Cluster
0                     0.000281               0.003347
1                     0.000091               0.000091

[2 rows x 88 columns]

Silhouette Score for K=2: 0.18104549510866558
```

In [ ]:

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import pandas as pd

# Perform K-Means clustering with the optimal number of clusters (K=4)
optimal_clusters = 4  # Based on the Elbow Method
kmeans = KMeans(n_clusters=optimal_clusters, n_init=50, random_state=42)
kmeans.fit(clustering_data)

# Add the cluster labels to the clustering_data DataFrame
clustering_data['Cluster'] = kmeans.labels_

# Calculate the mean values for each feature (including one-hot encoded c
cluster_means = clustering_data.groupby('Cluster').mean()

print("\nMean values for each feature within each cluster:")
print(cluster_means)
```

```python
# Calculate the silhouette score
silhouette_avg = silhouette_score(clustering_data.drop(columns=['Cluster'
print(f"\nSilhouette Score for K={optimal_clusters}: {silhouette_avg}")

# Optional: Save the clustering results
clustering_data.to_csv('clustering_results.csv', index=False)
```

```
Mean values for each feature within each cluster:
         time_in_hospital  num_lab_procedures  num_procedures  \
Cluster
0                1.188584            0.702316        0.788902
1               -0.339837           -0.196827       -0.262453
2               -0.319039           -0.208386       -0.108835
3                0.055019            0.116061       -0.248414

         num_medications  number_outpatient  number_emergency  \
Cluster
0               1.106119          -0.097712         -0.085151
1              -0.274204          -0.071718         -0.079213
2              -0.387190          -0.144489         -0.115751
3               0.213187           1.927709          1.770374

         number_inpatient    diag_1    diag_2    diag_3  ...  \
Cluster                                                  ...
0               -0.052875 -0.184521  0.245164  0.308356  ...
1               -0.123027  0.089893  0.020994  0.017894  ...
2               -0.189744  0.009810 -0.211165 -0.255412  ...
3                2.479069 -0.106375  0.059963  0.091354  ...

         admission_source_id_20  admission_source_id_22  \
Cluster
0                      0.002773                0.000382
1                      0.001865                0.000044
2                      0.000562                0.000033
3                      0.000199                0.000199

         admission_source_id_25  admission_source_id_3  admission_source_
id_4  \
Cluster
0                      0.000096               0.002056                0.05
1630
1                      0.000000               0.001163                0.02
2838
2                      0.000000               0.002910                0.03
3395
3                      0.000000               0.000597                0.01
1149

         admission_source_id_5  admission_source_id_6  admission_source_i
d_7  \
Cluster
0                     0.013386               0.006693               0.562
673
1                     0.009434               0.005046               0.617
670
2                     0.003505               0.061830               0.466
373
3                     0.007764               0.004778               0.689
827

         admission_source_id_8  admission_source_id_9
Cluster
0                     0.000143               0.000143
1                     0.000110               0.000066
2                     0.000231               0.003935
3                     0.000199               0.000000

[4 rows x 88 columns]

Silhouette Score for K=4: 0.1254200111853815
```

In [ ]:
```
# Calculate the mean values for each feature (including one-hot encoded c
cluster_means = clustering_data.groupby('Cluster').mean()

print("\nMean values for each feature within each cluster:")
display(cluster_means)

clustering_data.to_csv('clustering_results.csv', index=False)
```

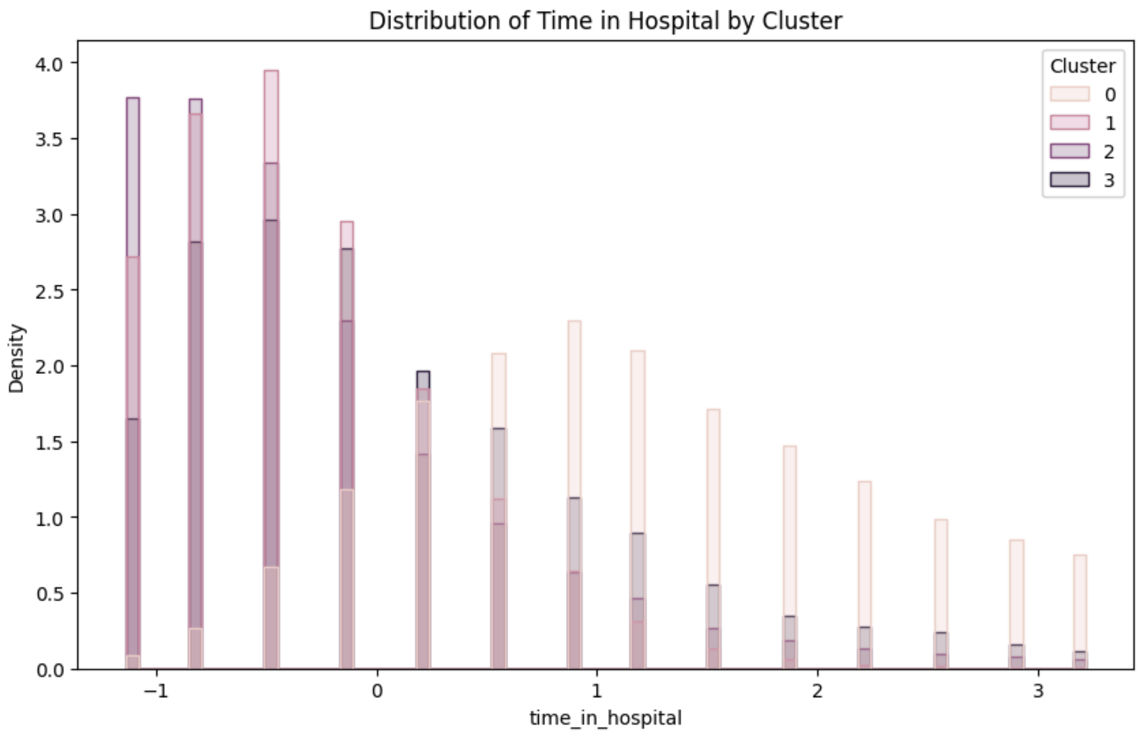
```
Mean values for each feature within each cluster:
```

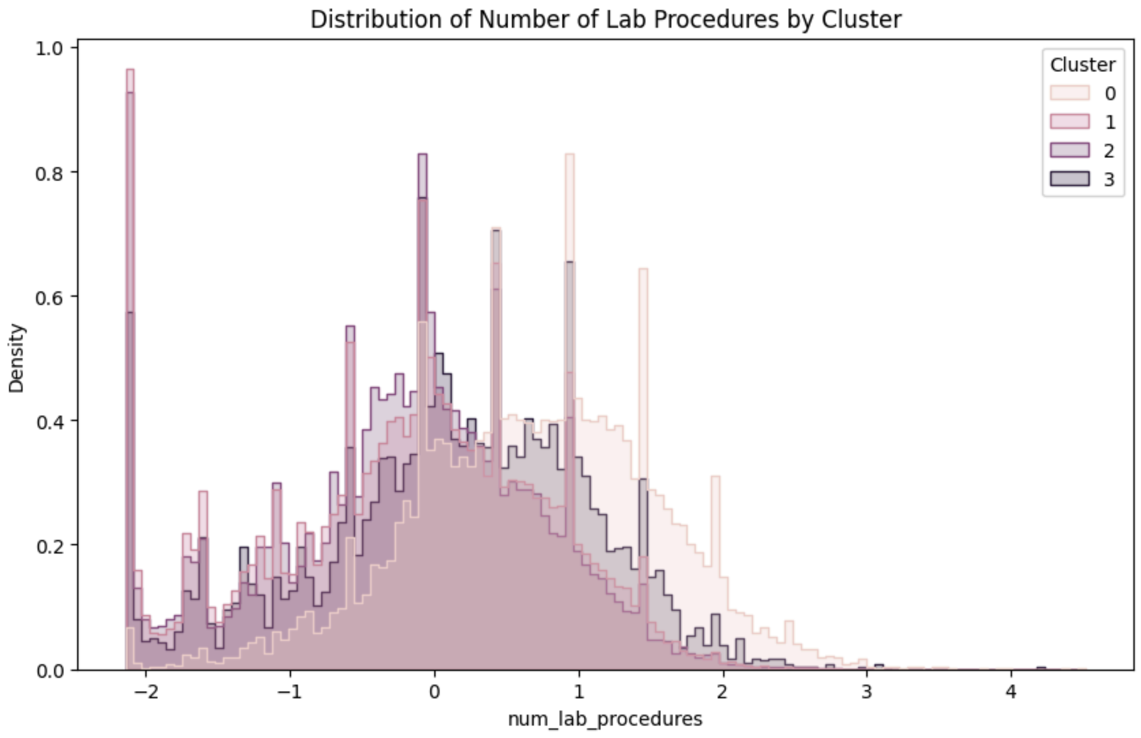
	time_in_hospital	num_lab_procedures	num_procedures	num_medications	number_
Cluster					
0	1.188584	0.702316	0.788902	1.106119	
1	-0.339837	-0.196827	-0.262453	-0.274204	
2	-0.319039	-0.208386	-0.108835	-0.387190	
3	0.055019	0.116061	-0.248414	0.213187	

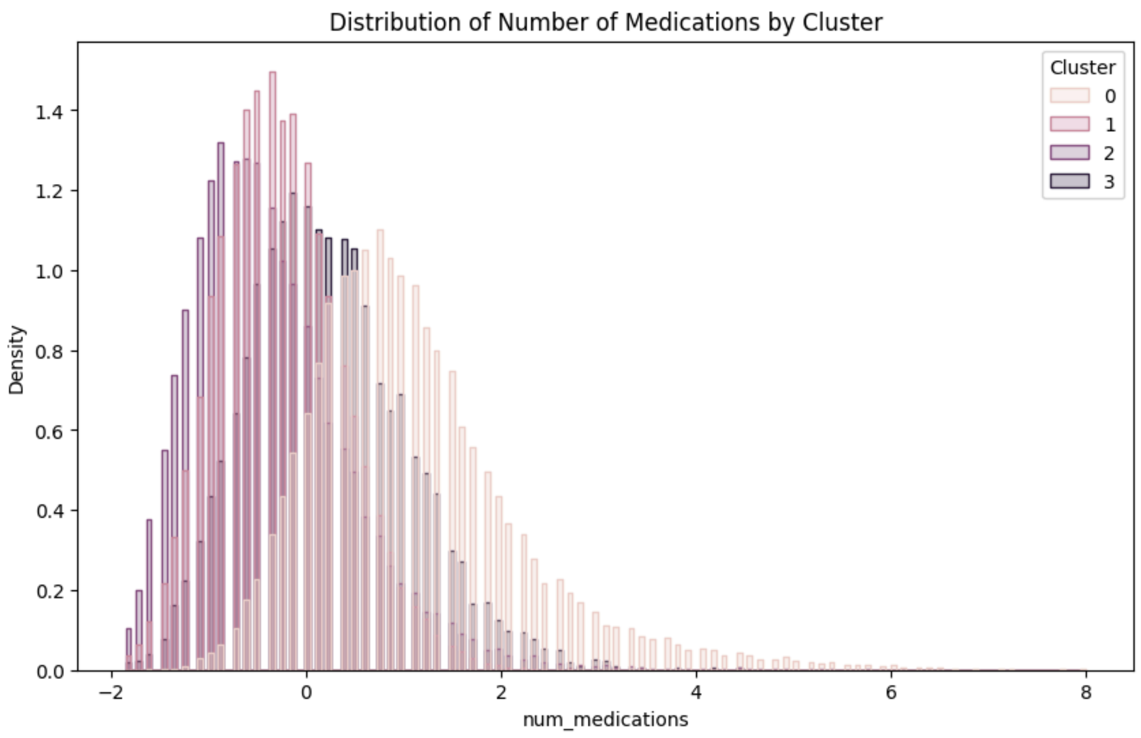
4 rows × 88 columns

```
Warning: Total number of columns (88) exceeds max_columns (20) limiting t
o first (20) columns.
```

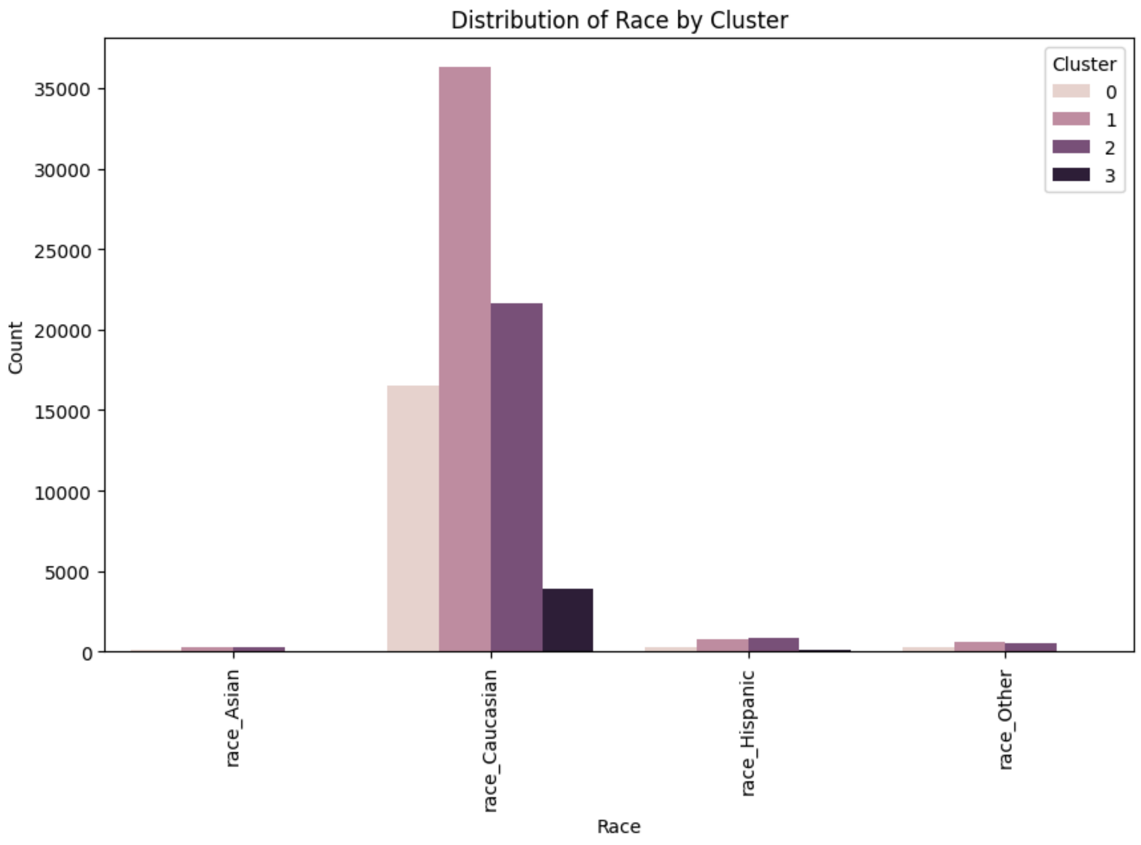
In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Ensure the Cluster column is present in clustering_data
clustering_data['Cluster'] = kmeans.labels_

# Visualize the distribution of 'time_in_hospital' by cluster
plt.figure(figsize=(10, 6))
sns.histplot(data=clustering_data, x='time_in_hospital', hue='Cluster', e
plt.title('Distribution of Time in Hospital by Cluster')
plt.show()

# Visualize the distribution of 'num_lab_procedures' by cluster
plt.figure(figsize=(10, 6))
sns.histplot(data=clustering_data, x='num_lab_procedures', hue='Cluster',
plt.title('Distribution of Number of Lab Procedures by Cluster')
plt.show()

# Visualize the distribution of 'num_medications' by cluster
plt.figure(figsize=(10, 6))
sns.histplot(data=clustering_data, x='num_medications', hue='Cluster', el
plt.title('Distribution of Number of Medications by Cluster')
plt.show()

# Sum the one-hot encoded 'race' columns by cluster
race_columns = [col for col in clustering_data.columns if col.startswith(
race_distribution = clustering_data.groupby('Cluster')[race_columns].sum(

# Convert to long format for easier plotting
race_distribution = race_distribution.reset_index().melt(id_vars='Cluster

# Visualize the distribution of 'race' by cluster
plt.figure(figsize=(10, 6))
sns.barplot(data=race_distribution, x='Race', y='Count', hue='Cluster')
plt.title('Distribution of Race by Cluster')
plt.xticks(rotation=90)
plt.show()

# Sum the one-hot encoded 'age' columns by cluster
age_columns = [col for col in clustering_data.columns if col.startswith('
age_distribution = clustering_data.groupby('Cluster')[age_columns].sum()

# Convert to long format for easier plotting
age_distribution = age_distribution.reset_index().melt(id_vars='Cluster',

# Visualize the distribution of 'age' by cluster
plt.figure(figsize=(10, 6))
sns.barplot(data=age_distribution, x='Age', y='Count', hue='Cluster')
plt.title('Distribution of Age by Cluster')
plt.xticks(rotation=90)
plt.show()
```



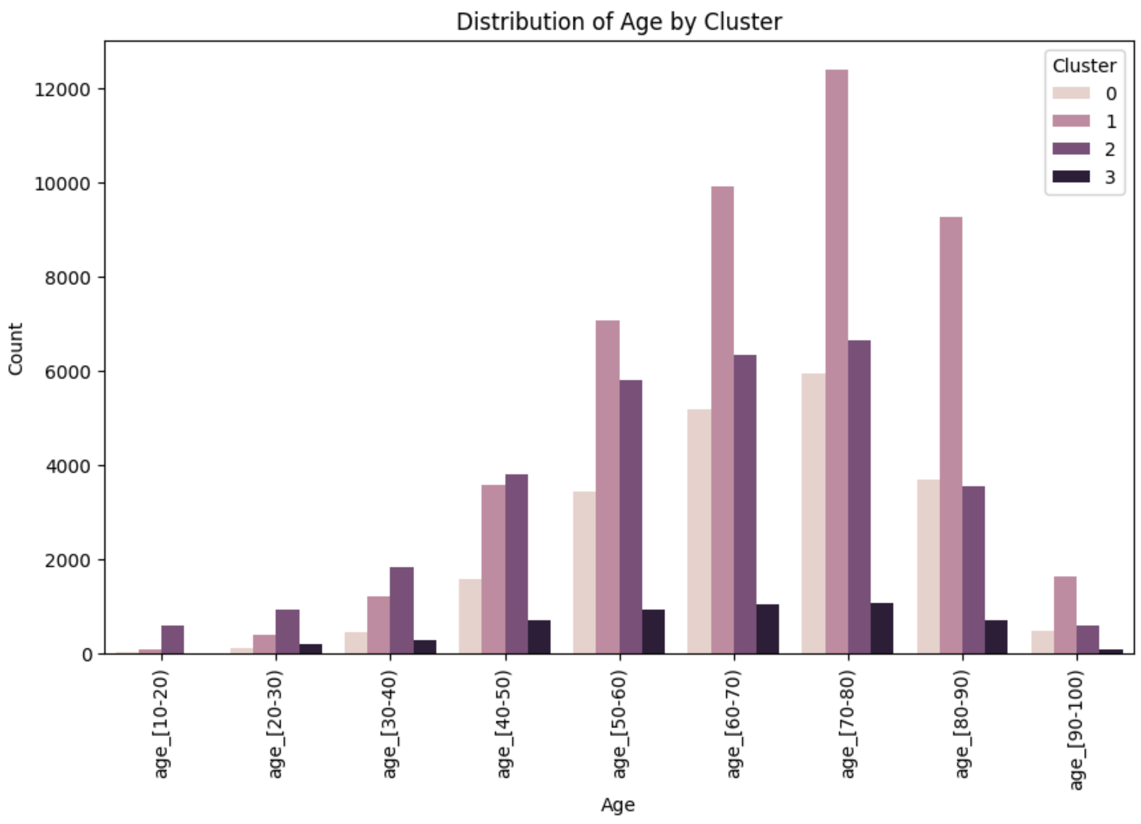




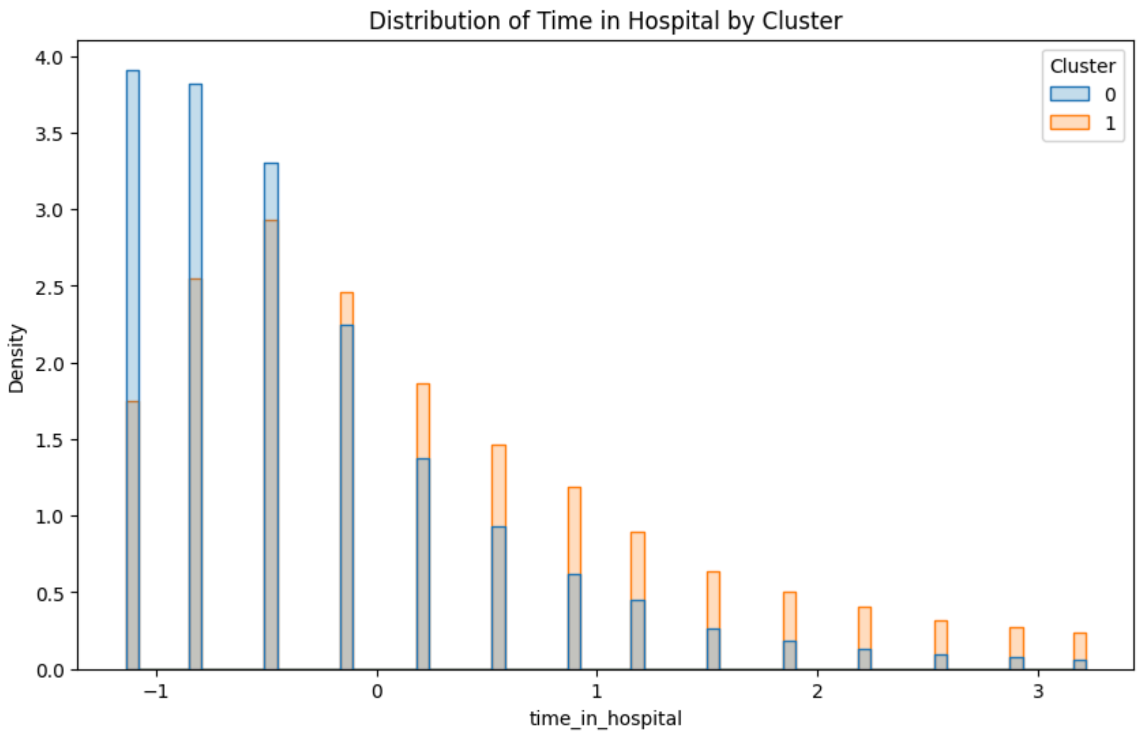
Distribution of Number of Medications by Cluster
Cluster
0
1
2
3
Density
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−2
0
2
4
6
8
num_medications

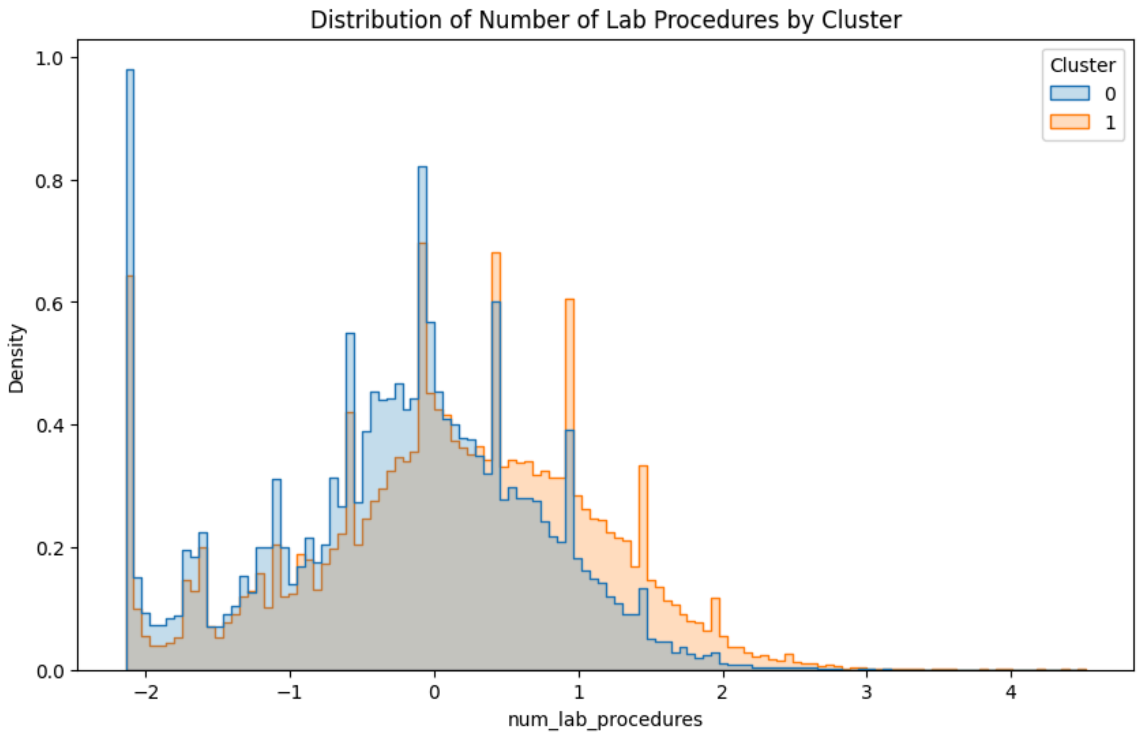


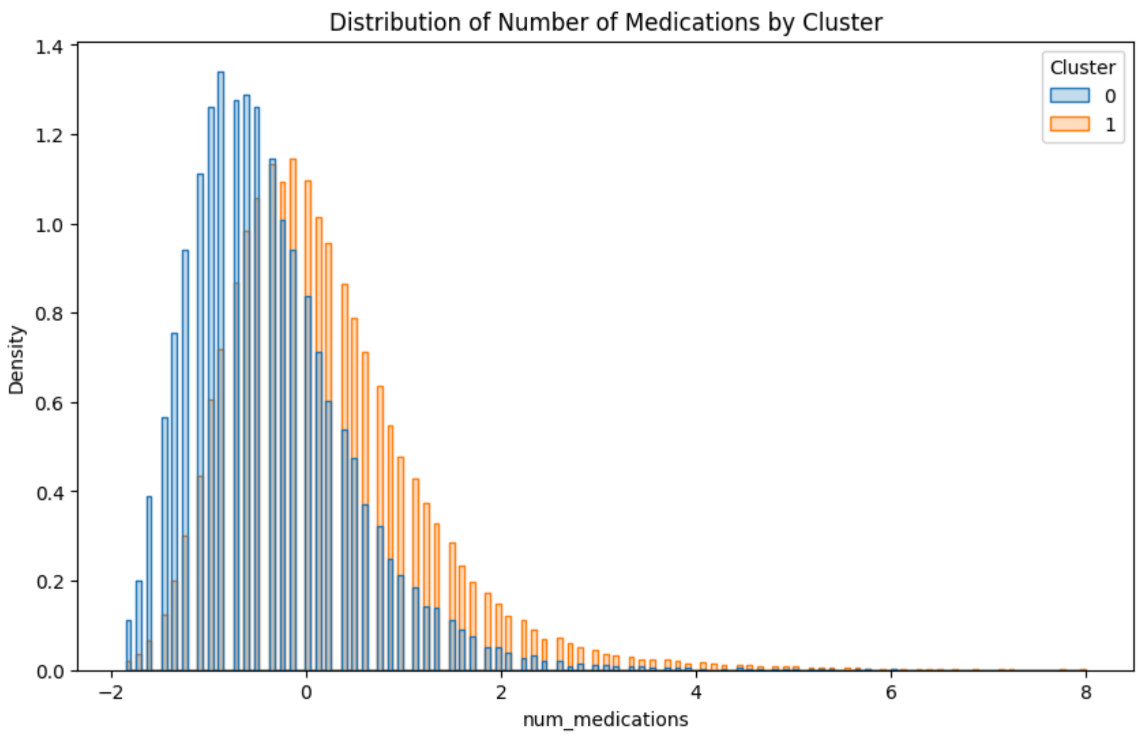
Distribution of Race by Cluster
Cluster
0
1
2
3
Count
35000
30000
25000
20000
15000
10000
5000
0
race_Asian
race_Caucasian
race_Hispanic
race_Other
Race



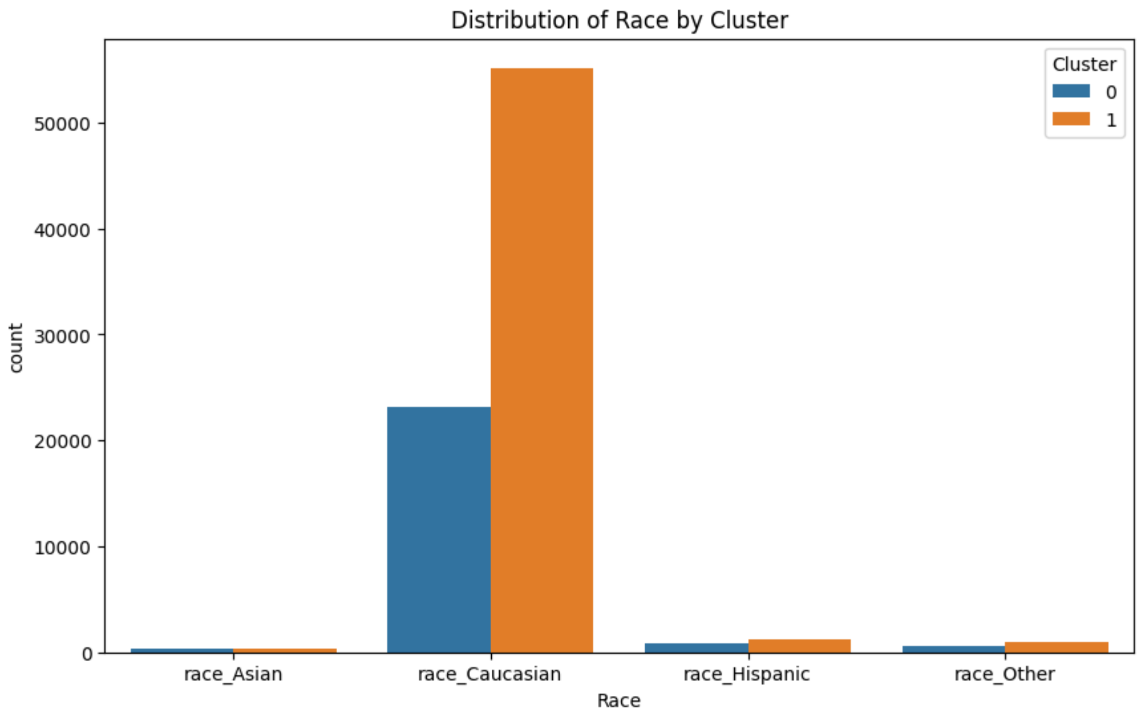
In [ ]:
```
from sklearn.cluster import KMeans
import seaborn as sns
import matplotlib.pyplot as plt

# Perform K-Means clustering with K=2
kmeans = KMeans(n_clusters=2, n_init=50, random_state=42)
kmeans.fit(clustering_data)

# Add the cluster labels to the DataFrame
clustering_data['Cluster'] = kmeans.labels_

# Visualize the distribution of 'time_in_hospital' by cluster
plt.figure(figsize=(10, 6))
sns.histplot(data=clustering_data, x='time_in_hospital', hue='Cluster', e
plt.title('Distribution of Time in Hospital by Cluster')
plt.show()

# Visualize the distribution of 'num_lab_procedures' by cluster
plt.figure(figsize=(10, 6))
sns.histplot(data=clustering_data, x='num_lab_procedures', hue='Cluster',
plt.title('Distribution of Number of Lab Procedures by Cluster')
plt.show()

# Visualize the distribution of 'num_medications' by cluster
plt.figure(figsize=(10, 6))
sns.histplot(data=clustering_data, x='num_medications', hue='Cluster', el
plt.title('Distribution of Number of Medications by Cluster')
plt.show()

# Visualize the distribution of 'race' by cluster
plt.figure(figsize=(10, 6))
race_columns = ['race_Asian', 'race_Caucasian', 'race_Hispanic', 'race_Ot
race_data = clustering_data.melt(id_vars='Cluster', value_vars=race_colum
race_data = race_data[race_data['Count'] == 1]
sns.countplot(data=race_data, x='Race', hue='Cluster')
plt.title('Distribution of Race by Cluster')
plt.show()

# Visualize the distribution of 'age' by cluster
plt.figure(figsize=(10, 6))
age_columns = ['age_[10-20)', 'age_[20-30)', 'age_[30-40)', 'age_[40-50)'
age_data = clustering_data.melt(id_vars='Cluster', value_vars=age_columns
age_data = age_data[age_data['Count'] == 1]
sns.countplot(data=age_data, x='Age', hue='Cluster')
plt.title('Distribution of Age by Cluster')
plt.show()
```



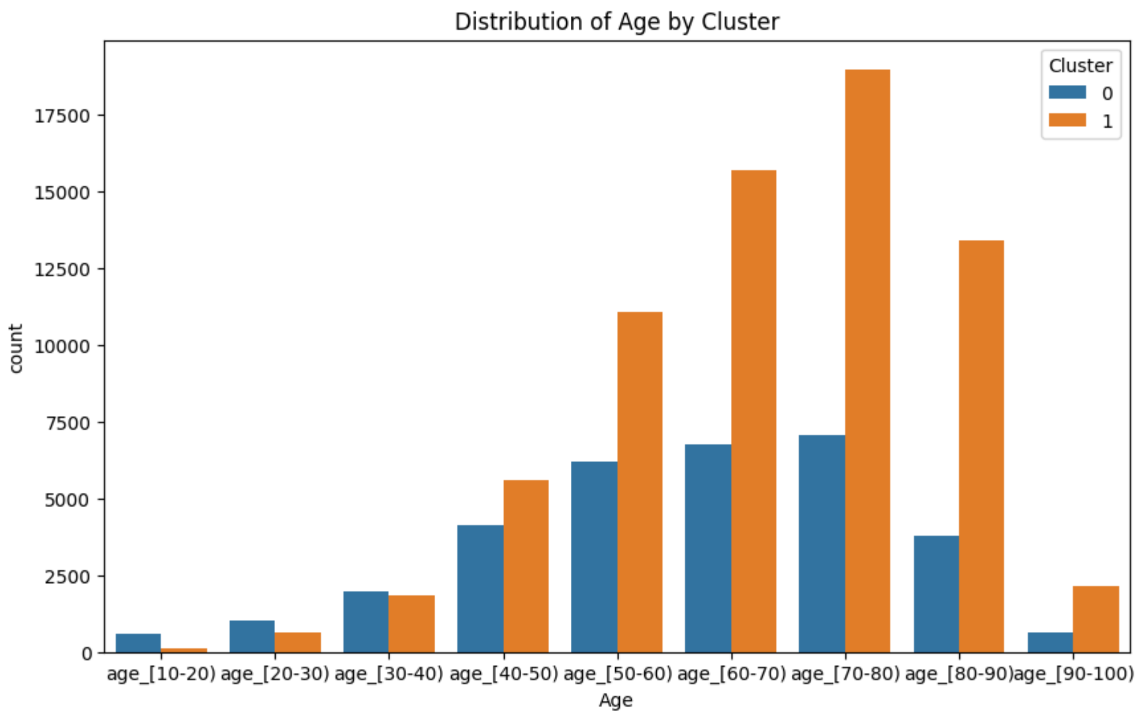




Distribution of Number of Medications by Cluster
Cluster
0
1
Density
num_medications
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−2
0
2
4
6
8



Distribution of Race by Cluster
Cluster
0
1
count
Race
50000
40000
30000
20000
10000
0
race_Asian
race_Caucasian
race_Hispanic
race_Other


Distribution of Age by Cluster
Cluster
0
1
count
17500
15000
12500
10000
7500
5000
2500
0
age_[10-20)
age_[20-30)
age_[30-40)
age_[40-50)
age_[50-60)
age_[60-70)
age_[70-80)
age_[80-90)
age_[90-100)
Age