cva2_final-Copy1

March 2, 2024

[ ]:

[2]:
```
import pandas as pd
import numpy as np
from prettytable import PrettyTable
from collections import Counter
import matplotlib.pyplot as plt
import seaborn as sns
import scipy
import warnings
warnings.filterwarnings("ignore")

from sklearn.neighbors import LocalOutlierFactor, KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix,␣
 ↪accuracy_score, roc_auc_score, precision_recall_fscore_support, roc_curve,␣
 ↪f1_score
from sklearn.preprocessing import StandardScaler
from sklearn.tree import  DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import cross_val_score, cross_validate
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from catboost import CatBoostClassifier
from sklearn.linear_model import LogisticRegression
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import RandomOverSampler
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import roc_auc_score, f1_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve, auc
```

[3]:
```
data = pd.read_csv('diabetic_data.csv')
data
```

[3]:

	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)	?	
…	…	…	…	…	…	…	…
101761	443847548	100162476	AfricanAmerican	Male	[70-80)	?	
101762	443847782	74694222	AfricanAmerican	Female	[80-90)	?	
101763	443854148	41088789	Caucasian	Male	[70-80)	?	
101764	443857166	31693671	Caucasian	Female	[80-90)	?	
101765	443867222	175429310	Caucasian	Male	[70-80)	?	

	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	
…	…	…	…	
101761	1	3	7	
101762	1	4	5	
101763	1	1	7	
101764	2	3	7	
101765	1	1	7	

	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	
…	…	…	…	…	…	
101761	3	…	No	Down	No	
101762	5	…	No	Steady	No	
101763	1	…	No	Down	No	
101764	10	…	No	Up	No	
101765	6	…	No	No	No	

	glipizide-metformin	glimepiride-pioglitazone	\
0	No	No	
1	No	No	
2	No	No	
3	No	No	
4	No	No	
…	…	…	
101761	No	No	

```
101762                    No                         No
101763                    No                         No
101764                    No                         No
101765                    No                         No

        metformin-rosiglitazone  metformin-pioglitazone  change diabetesMed  \
0                            No                      No      No          No
1                            No                      No      Ch         Yes
2                            No                      No      No         Yes
3                            No                      No      Ch         Yes
4                            No                      No      Ch         Yes
…                             …                       …       …          …
101761                       No                      No      Ch         Yes
101762                       No                      No      No         Yes
101763                       No                      No      Ch         Yes
101764                       No                      No      Ch         Yes
101765                       No                      No      No          No

       readmitted
0              NO
1             >30
2              NO
3              NO
4              NO
…             …
101761        >30
101762         NO
101763         NO
101764         NO
101765         NO

[101766 rows x 50 columns]
```

[4]:
```
print(np.unique(data['age']))

replaceDict = {'[0-10)' : 5,
'[10-20)' : 15,
'[20-30)' : 25,
'[30-40)' : 35,
'[40-50)' : 45,
'[50-60)' : 55,
'[60-70)' : 65,
'[70-80)' : 75,
'[80-90)' : 85,
'[90-100)' : 95}

data['age'] = data['age'].apply(lambda x : replaceDict[x])
```

```
print(data['age'].head())
```

```
['[0-10)' '[10-20)' '[20-30)' '[30-40)' '[40-50)' '[50-60)' '[60-70)'
 '[70-80)' '[80-90)' '[90-100)']
0     5
1    15
2    25
3    35
4    45
Name: age, dtype: int64
```

[ ]:

[5]:
```
# data.dropna(inplace = True)
print('Total data = ', len(data))
print('Unique entries = ', len(np.unique(data['patient_nbr'])))
data.drop_duplicates(['patient_nbr'], keep = 'first', inplace = True)
print('Length after removing Duplicates:', len(data))
```

```
Total data =  101766
Unique entries =  71518
Length after removing Duplicates: 71518
```

[6]:
```
t = PrettyTable(['Column Name', 'Missing Values', 'Missing Percentage'])
for col in data.columns :
    lenn = len(data[data[col] == '?'])
    t.add_row([col,lenn,lenn / len(data)])
print(t)
```

```
+--------------------------+----------------+------------------------+
|       Column Name        | Missing Values |   Missing Percentage   |
+--------------------------+----------------+------------------------+
|       encounter_id       |       0        |          0.0           |
|       patient_nbr        |       0        |          0.0           |
|           race           |      1948      |  0.027237898151514305  |
|          gender          |       0        |          0.0           |
|           age            |       0        |          0.0           |
|          weight          |     68665      |   0.960107944853044    |
|    admission_type_id     |       0        |          0.0           |
| discharge_disposition_id |       0        |          0.0           |
|   admission_source_id    |       0        |          0.0           |
|     time_in_hospital     |       0        |          0.0           |
|        payer_code        |     31043      |   0.434058558684527    |
|    medical_specialty     |     34477      |   0.4820744427976174   |
|    num_lab_procedures    |       0        |          0.0           |
|      num_procedures      |       0        |          0.0           |
|     num_medications      |       0        |          0.0           |
|    number_outpatient     |       0        |          0.0           |
|     number_emergency     |       0        |          0.0           |
```

```
|     number_inpatient     |       0        |          0.0          |
|          diag_1          |       11       | 0.0001538074330937666 |
|          diag_2          |      294       | 0.0041108532117788525 |
|          diag_3          |      1225      |  0.017128555049078554 |
|     number_diagnoses     |       0        |          0.0          |
|      max_glu_serum       |       0        |          0.0          |
|        A1Cresult         |       0        |          0.0          |
|        metformin         |       0        |          0.0          |
|       repaglinide        |       0        |          0.0          |
|       nateglinide        |       0        |          0.0          |
|      chlorpropamide      |       0        |          0.0          |
|       glimepiride        |       0        |          0.0          |
|      acetohexamide       |       0        |          0.0          |
|        glipizide         |       0        |          0.0          |
|        glyburide         |       0        |          0.0          |
|       tolbutamide        |       0        |          0.0          |
|       pioglitazone       |       0        |          0.0          |
|      rosiglitazone       |       0        |          0.0          |
|         acarbose         |       0        |          0.0          |
|         miglitol         |       0        |          0.0          |
|       troglitazone       |       0        |          0.0          |
|        tolazamide        |       0        |          0.0          |
|         examide          |       0        |          0.0          |
|       citoglipton        |       0        |          0.0          |
|         insulin          |       0        |          0.0          |
|   glyburide-metformin    |       0        |          0.0          |
|   glipizide-metformin    |       0        |          0.0          |
| glimepiride-pioglitazone |       0        |          0.0          |
| metformin-rosiglitazone  |       0        |          0.0          |
|  metformin-pioglitazone  |       0        |          0.0          |
|          change          |       0        |          0.0          |
|       diabetesMed        |       0        |          0.0          |
|        readmitted        |       0        |          0.0          |
+--------------------------+----------------+-----------------------+
```

[7]:
```
high_frequency = ['InternalMedicine', 'Family/GeneralPractice', 'Cardiology',␣
↪'Surgery-General', 'Orthopedics', 'Orthopedics-Reconstructive',
                  'Emergency/Trauma',␣
↪'Urology','ObstetricsandGynecology','Psychiatry','Pulmonology␣
↪','Nephrology','Radiologist']

low_frequency =␣
↪['Surgery-PlasticwithinHeadandNeck','Psychiatry-Addictive','Proctology','Dermatology','Spor
                ␣
↪'Neurophysiology','Resident','Pediatrics-Hematology-Oncology','Pediatrics-EmergencyMedicine
↪Adolescent',\
```

```
                ␣
 ↪'Pediatrics-Pulmonology','Surgery-Pediatric','AllergyandImmunology','Pediatrics-Neurology',
                ␣
 ↪'Endocrinology-Metabolism','PhysicianNotFound','Surgery-Colon&Rectal','OutreachServices',\
                ␣
 ↪'Surgery-Maxillofacial','Rheumatology','Anesthesiology-Pediatric','Obstetrics','Obsterics&G

pediatrics =␣
 ↪['Pediatrics','Pediatrics-CriticalCare','Pediatrics-EmergencyMedicine','Pediatrics-Endocrin
                'Pediatrics-Neurology','Pediatrics-Pulmonology',␣
 ↪'Anesthesiology-Pediatric', 'Cardiology-Pediatric', 'Surgery-Pediatric']

psychic = ['Psychiatry-Addictive', 'Psychology', 'Psychiatry', ␣
 ↪'Psychiatry-Child/Adolescent', 'PhysicalMedicineandRehabilitation',␣
 ↪'Osteopath']


neurology = ['Neurology', 'Surgery-Neuro',  'Pediatrics-Neurology',␣
 ↪'Neurophysiology']


surgery = ['Surgeon', 'Surgery-Cardiovascular', \
          'Surgery-Cardiovascular/Thoracic', 'Surgery-Colon&Rectal',␣
 ↪'Surgery-General', 'Surgery-Maxillofacial', \
             'Surgery-Plastic', 'Surgery-PlasticwithinHeadandNeck', ␣
 ↪'Surgery-Thoracic',\
             'Surgery-Vascular', 'SurgicalSpecialty', 'Podiatry']

ungrouped =␣
 ↪['Endocrinology','Gastroenterology','Gynecology','Hematology','Hematology/
 ↪Oncology','Hospitalist','InfectiousDiseases',\
          ␣
 ↪'Oncology','Ophthalmology','Otolaryngology','Pulmonology','Radiology']


missing = ['?']

colMedical = []

for val in data['medical_specialty'] :
    if val in pediatrics :
        colMedical.append('pediatrics')
    elif val in psychic :
        colMedical.append('psychic')
    elif val in neurology :
        colMedical.append('neurology')
```

```
    elif val in surgery :
        colMedical.append('surgery')
    elif val in high_frequency :
        colMedical.append('high_freq')
    elif val in low_frequency :
        colMedical.append('low_freq')
    elif val in ungrouped :
        colMedical.append('ungrouped')
    elif val in missing :
        colMedical.append('missing')

data['medical_specialty'] = colMedical
```

[8]:
```
diag_1 = Counter(list(data['diag_1'])).most_common(1)[0][0]
diag_2 = Counter(list(data['diag_2'])).most_common(1)[0][0]
diag_3 = Counter(list(data['diag_3'])).most_common(1)[0][0]

data['diag_1'] = data['diag_1'].apply(lambda x : diag_1 if x == '?' else x)
data['diag_2'] = data['diag_1'].apply(lambda x : diag_2 if x == '?' else x)
data['diag_3'] = data['diag_3'].apply(lambda x : diag_3 if x == '?' else x)
```

[9]:
```
print(len(np.unique(data['diag_1'])))
print(len(np.unique(data['diag_2'])))
print(len(np.unique(data['diag_3'])))

data['diag_1'] = data['diag_1'].apply(lambda x : 'other' if (str(x).find('V') !
 ↪= -1 or str(x).find('E') != -1)
                                        else ('circulatory' if int(float(x)) in␣
 ↪range(390, 460) or int(float(x)) == 785
                                        else     ('respiratory' if␣
 ↪int(float(x)) in range(460, 520) or int(float(x)) == 786
                                        else     ('digestive'   if␣
 ↪int(float(x)) in range(520, 580) or int(float(x)) == 787
                                        else     ('diabetes'    if␣
 ↪int(float(x)) == 250
                                        else     ('injury'      if␣
 ↪int(float(x)) in range(800, 1000)
                                        else ('musculoskeletal' if␣
 ↪int(float(x)) in range(710, 740)
                                        else ('genitourinary'   if␣
 ↪int(float(x)) in range(580, 630) or int(float(x)) == 788
                                        else ('neoplasms'       if␣
 ↪int(float(x)) in range(140, 240)
                                        else ('pregnecy'        if␣
 ↪int(float(x)) in range(630, 680)
                                        else 'other'))))))))))
```

```
data['diag_2'] = data['diag_2'].apply(lambda x : 'other' if (str(x).find('V') !
 ↪= -1 or str(x).find('E') != -1)
                                        else ('circulatory' if int(float(x)) in␣
 ↪range(390, 460) or int(float(x)) == 785
                                        else     ('respiratory' if␣
 ↪int(float(x)) in range(460, 520) or int(float(x)) == 786
                                        else     ('digestive'   if␣
 ↪int(float(x)) in range(520, 580) or int(float(x)) == 787
                                        else     ('diabetes'    if␣
 ↪int(float(x)) == 250
                                        else     ('injury'      if␣
 ↪int(float(x)) in range(800, 1000)
                                        else ('musculoskeletal' if␣
 ↪int(float(x)) in range(710, 740)
                                        else ('genitourinary'   if␣
 ↪int(float(x)) in range(580, 630) or int(float(x)) == 788
                                        else ('neoplasms'       if␣
 ↪int(float(x)) in range(140, 240)
                                        else ('pregnecy'        if␣
 ↪int(float(x)) in range(630, 680)
                                        else 'other'))))))))))

data['diag_3'] = data['diag_3'].apply(lambda x : 'other' if (str(x).find('V') !
 ↪= -1 or str(x).find('E') != -1)
                                        else ('circulatory' if int(float(x)) in␣
 ↪range(390, 460) or int(float(x)) == 785
                                        else     ('respiratory' if␣
 ↪int(float(x)) in range(460, 520) or int(float(x)) == 786
                                        else     ('digestive'   if␣
 ↪int(float(x)) in range(520, 580) or int(float(x)) == 787
                                        else     ('diabetes'    if␣
 ↪int(float(x)) == 250
                                        else     ('injury'      if␣
 ↪int(float(x)) in range(800, 1000)
                                        else ('musculoskeletal' if␣
 ↪int(float(x)) in range(710, 740)
                                        else ('genitourinary'   if␣
 ↪int(float(x)) in range(580, 630) or int(float(x)) == 788
                                        else ('neoplasms'       if␣
 ↪int(float(x)) in range(140, 240)
                                        else ('pregnecy'        if␣
 ↪int(float(x)) in range(630, 680)
                                        else 'other'))))))))))
```

```
print(np.unique(data['diag_1']), '\n')
print(np.unique(data['diag_2']), '\n')
print(np.unique(data['diag_3']), '\n')
```

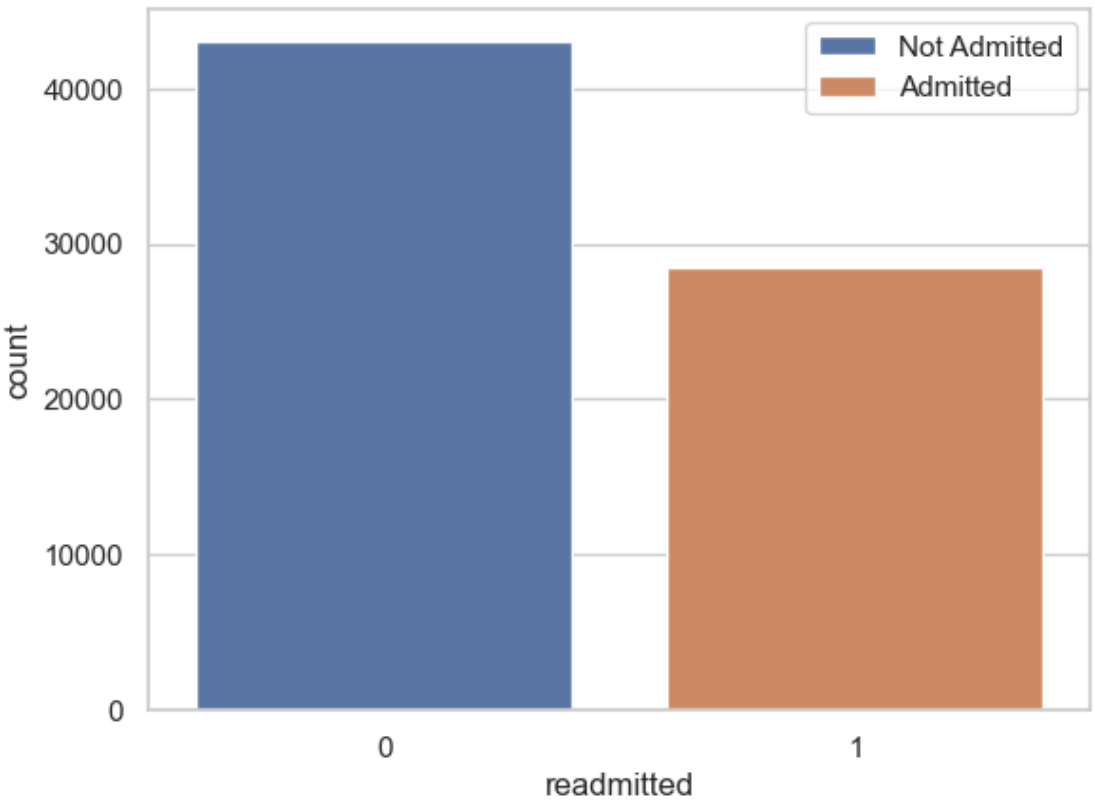
```
696
696
758
['circulatory' 'diabetes' 'digestive' 'genitourinary' 'injury'
 'musculoskeletal' 'neoplasms' 'other' 'pregnecy' 'respiratory']

['circulatory' 'diabetes' 'digestive' 'genitourinary' 'injury'
 'musculoskeletal' 'neoplasms' 'other' 'pregnecy' 'respiratory']

['circulatory' 'diabetes' 'digestive' 'genitourinary' 'injury'
 'musculoskeletal' 'neoplasms' 'other' 'pregnecy' 'respiratory']
```

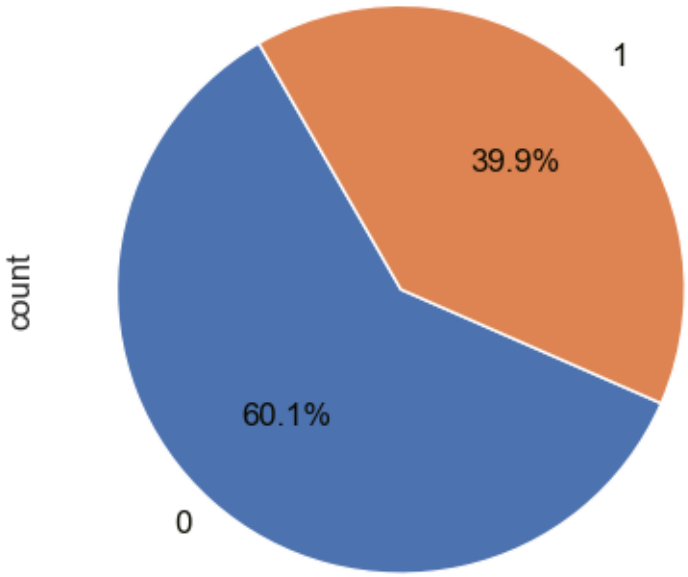
[10]:
```
print('BEFORE : ',np.unique(data['readmitted'].values))

data['readmitted'] = data['readmitted'].apply(lambda x : 0 if (x == 'NO') else␣
 ↪1)

print('AFTER : ',np.unique(data['readmitted'].values))
```

```
BEFORE :  ['<30' '>30' 'NO']
AFTER :  [0 1]
```

[11]:
```
##
plt.figure()
sns.set_theme(style="whitegrid")
ax = sns.countplot(x = 'readmitted', data = data, hue = 'readmitted')
handles, labels = ax.get_legend_handles_labels()
ax.legend(handles, labels = ['Not Admitted', 'Admitted'])
plt.figure()
data.readmitted.value_counts().plot.pie(autopct="%1.1f%%", startangle=120,
                                        textprops={'fontsize': 12, 'color':
 ↪'#0a0a00'})
```

[11]:
```
<Axes: ylabel='count'>
```


Not Admitted
Admitted
40000
30000
20000
10000
0
count
0
1
readmitted


1
39.9%
60.1%
0
count

[12]:
```
data.drop(['encounter_id', 'patient_nbr'], axis = 1, inplace = True)
data.drop(data[data.gender == 'Unknown/Invalid'].index, inplace = True)
```

[13]:
```
### lets analyse numeric features
from tqdm import tqdm

data['health_index'] = data.apply(lambda x:  1 / (x['number_emergency'] +␣
 ↪x['number_inpatient'] + x['number_outpatient'])
                                  if x['number_emergency'] != 0 or␣
 ↪x['number_inpatient'] != 0 or x['number_outpatient'] != 0
                                  else 1, axis = 1)



total = data['time_in_hospital'].sum() + data['num_procedures'].sum() + \
                               data['num_medications'].sum() +␣
 ↪data['num_lab_procedures'].sum() + \
                               data['number_diagnoses'].sum()

data['severity_of_disease'] = (data['time_in_hospital'] +␣
 ↪data['num_procedures'] + \
                               data['num_medications'] +␣
 ↪data['num_lab_procedures'] + \
                               data['number_diagnoses']) / total

drugList =␣
 ↪['metformin','repaglinide','nateglinide','chlorpropamide','glimepiride','acetohexamide',\
          ␣
 ↪'glipizide','glyburide','tolbutamide','pioglitazone','rosiglitazone','acarbose','miglitol',
          ␣
 ↪'troglitazone','tolazamide','examide','citoglipton','insulin','glyburide-metformin','glipiz
          ␣
 ↪'glimepiride-pioglitazone','metformin-rosiglitazone','metformin-pioglitazone']


number_of_changes = []
for i in tqdm(range(len(data))) :
    changeCount = 0
    for col in drugList :
        if data.iloc[i][col] in ['Down', 'Up'] :
            changeCount += 1
    number_of_changes.append(changeCount)

data['number_of_changes'] = number_of_changes
```

```
100%|      | 71515/71515 [02:16<00:00, 522.57it/s]
```

[14]:
```
data_checkpoint = data
```

[15]:
```
data_checkpoint.to_csv("proj2_cp2.csv", index = False)
```

[16]:
```
data_checkpoint = pd.read_csv("proj2_cp2.csv")
```

[17]:
```
data.drop(['weight','race', 'payer_code'], axis = 1, inplace = True)   ##␣
↪change here
```

[18]:
```
data['discharge_disposition_id'] = data['discharge_disposition_id'].
↪apply(lambda x : 1 if int(x) in [6, 8, 9, 13]
                                                                    else␣
↪( 2 if int(x) in [3, 4, 5, 14, 22, 23, 24]
                                                                    else␣
↪( 10 if int(x) in [12, 15, 16, 17]
                                                                    else␣
↪( 11 if int(x) in [19, 20, 21]
                                                                    else␣
↪( 18 if int(x) in [25, 26]
                                                                    else␣
↪int(x) )))))

data = data[~data.discharge_disposition_id.isin([11,13,14,19,20,21])]
```

[19]:
```
data['admission_type_id'] = data['admission_type_id'].apply(lambda x : 1 if␣
↪int(x) in [2, 7]
                                                        else ( 5 if int(x)␣
↪in [6, 8]
                                                        else int(x) ))
```

[20]:
```
data['admission_source_id'] = data['admission_source_id'].apply(lambda x : 1 if␣
↪int(x) in [2, 3]
                                                        else ( 4 if int(x)␣
↪in [5, 6, 10, 22, 25]
                                                        else ( 9 if int(x)␣
↪in [15, 17, 20, 21]
                                                        else ( 11 if int(x)␣
↪in [13, 14]
                                                        else int(x) ))))
```

[21]:
```
data['max_glu_serum'] = data['max_glu_serum'].apply(lambda x : 200 if x ==␣
↪'>200'
                                                        else ( 300 if x ==␣
↪'>300'
```

```
                                                        else ( 100 if x ==␣
↪'Norm'
                                                        else  0)))
```

[22]:
```
data['A1Cresult'] = data['A1Cresult'].apply(lambda x : 7 if x == '>7'
                                                    else (8 if  x == '>8'
                                                    else ( 5 if x == 'Norm'
                                                    else  0)))
```

[23]:
```
for col in ["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"]:
    data[col] = data[col].apply(lambda x : 10 if x == 'Up'
                                            else ( -10 if x == 'Down'
                                            else ( 0 if x == 'Steady'
                                            else  -20)))


data['change'] = data['change'].apply(lambda x : 1 if x == 'Ch'
                                              else -1)


data['diabetesMed'] = data['diabetesMed'].apply(lambda x : -1 if x == 'No'
                                            else 1)
```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[24]:
```
#data = pd.read_csv('/content/diabetic_dataset3.csv')
```

[25]:
```
data.columns
```

[25]:
```
Index(['gender', 'age', 'admission_type_id', 'discharge_disposition_id',
       'admission_source_id', 'time_in_hospital', 'medical_specialty',
       'num_lab_procedures', 'num_procedures', 'num_medications',
       'number_outpatient', 'number_emergency', 'number_inpatient', 'diag_1',
       'diag_2', 'diag_3', 'number_diagnoses', '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',
       'health_index', 'severity_of_disease', 'number_of_changes'],
      dtype='object')
```

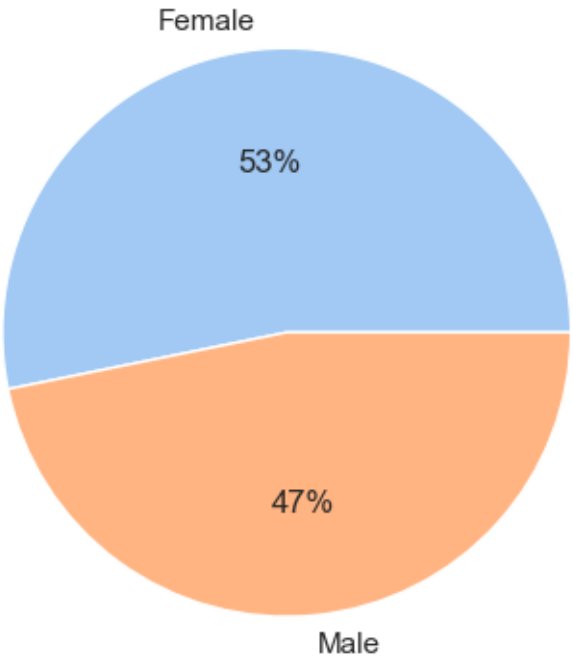
[26]:
```
data['readmitted']
```

[26]:
```
0         0
1         1
2         0
3         0
4         0
         ..
101754    1
101755    1
101756    1
101758    0
101765    0
Name: readmitted, Length: 70431, dtype: int64
```

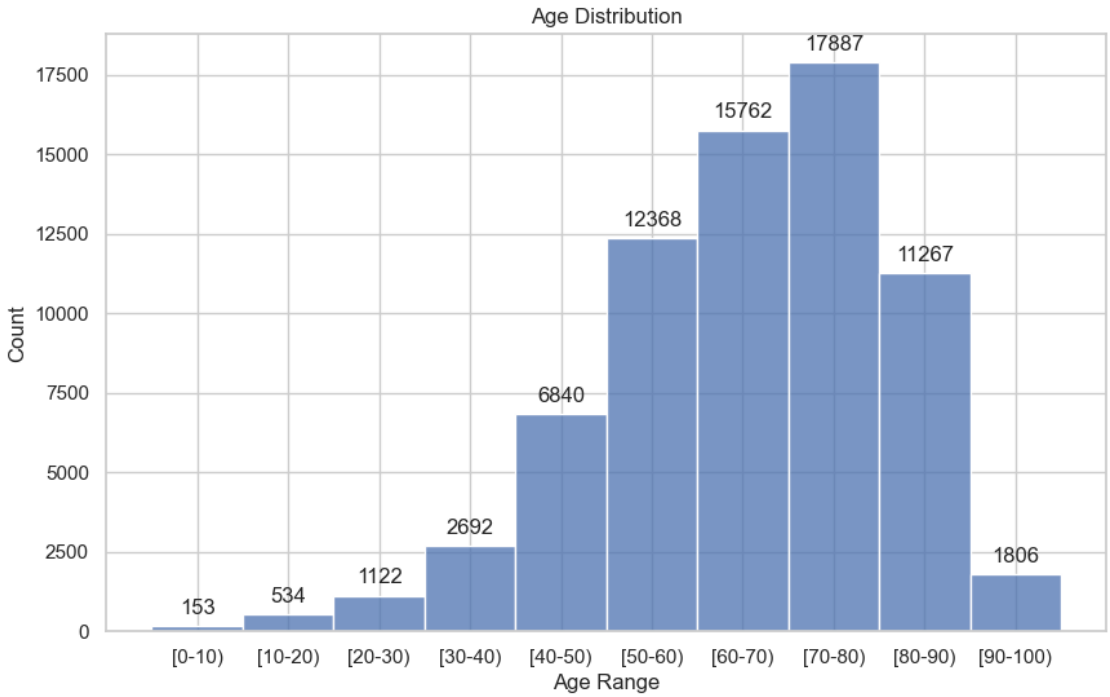
[27]:
```
data.columns
```

[27]:
```
Index(['gender', 'age', 'admission_type_id', 'discharge_disposition_id',
       'admission_source_id', 'time_in_hospital', 'medical_specialty',
       'num_lab_procedures', 'num_procedures', 'num_medications',
       'number_outpatient', 'number_emergency', 'number_inpatient', 'diag_1',
       'diag_2', 'diag_3', 'number_diagnoses', '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',
       'health_index', 'severity_of_disease', 'number_of_changes'],
      dtype='object')
```

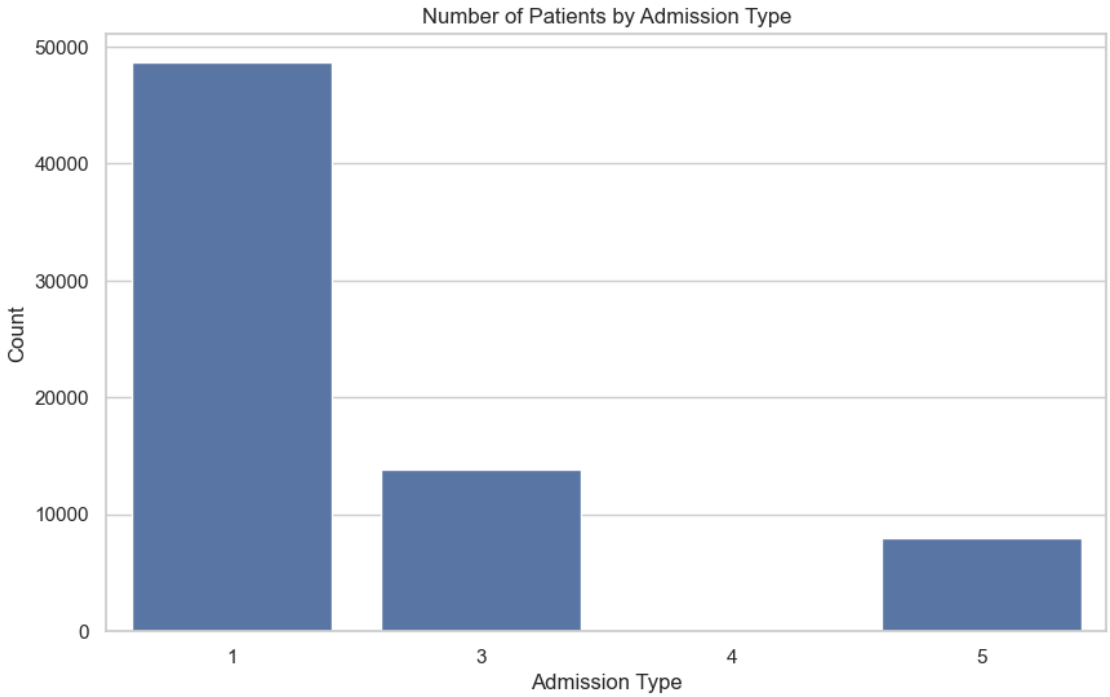
[28]:
```
gender_counts = data['gender'].value_counts()

palette_color = sns.color_palette('pastel')
plt.pie(gender_counts, labels=gender_counts.index, colors=palette_color,␣
 ↪autopct='%.0f%%')
plt.show()
```



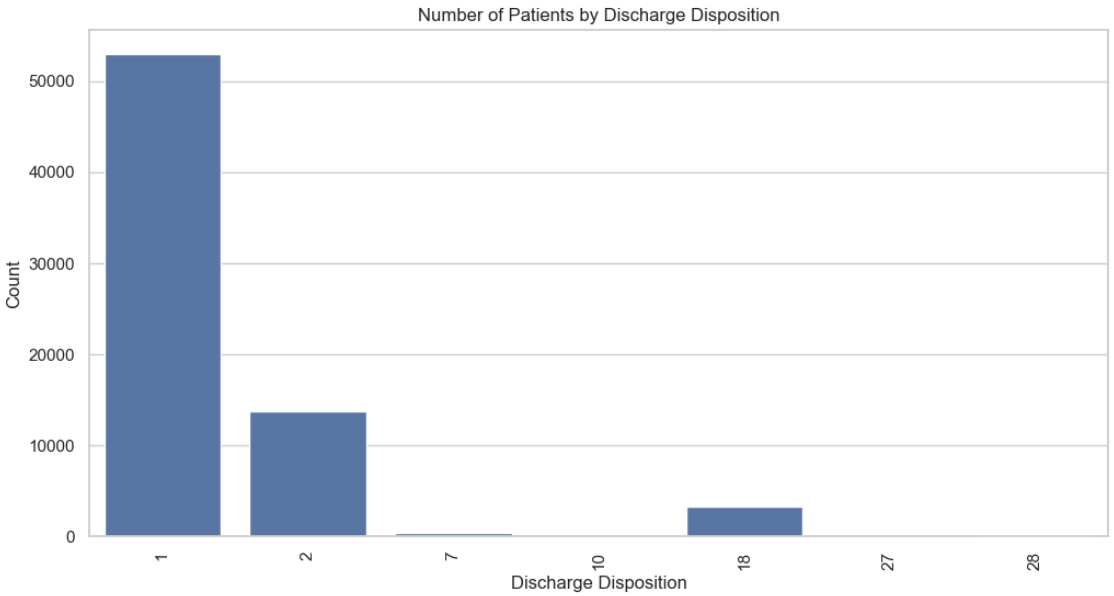
[29]:
```
reverseDict = {5: '[0-10)', 15: '[10-20)', 25: '[20-30)', 35: '[30-40)', 45:␣
 ↪'[40-50)', 55: '[50-60)', 65: '[60-70)', 75: '[70-80)', 85: '[80-90)', 95:␣
 ↪'[90-100)'}

data['age_labels'] = data['age'].apply(lambda x : reverseDict[x])

plt.figure(figsize=(10, 6))
plot = sns.histplot(data['age_labels'], bins=len(reverseDict), discrete=True)
plt.title('Age Distribution')
plt.xlabel('Age Range')
plt.ylabel('Count')

for p in plot.patches:
    plot.annotate(f'{p.get_height()}',
                  (p.get_x() + p.get_width() / 2., p.get_height()),
                  ha = 'center', va = 'center',
                  xytext = (0, 10),
                  textcoords = 'offset points')

plt.show()
```



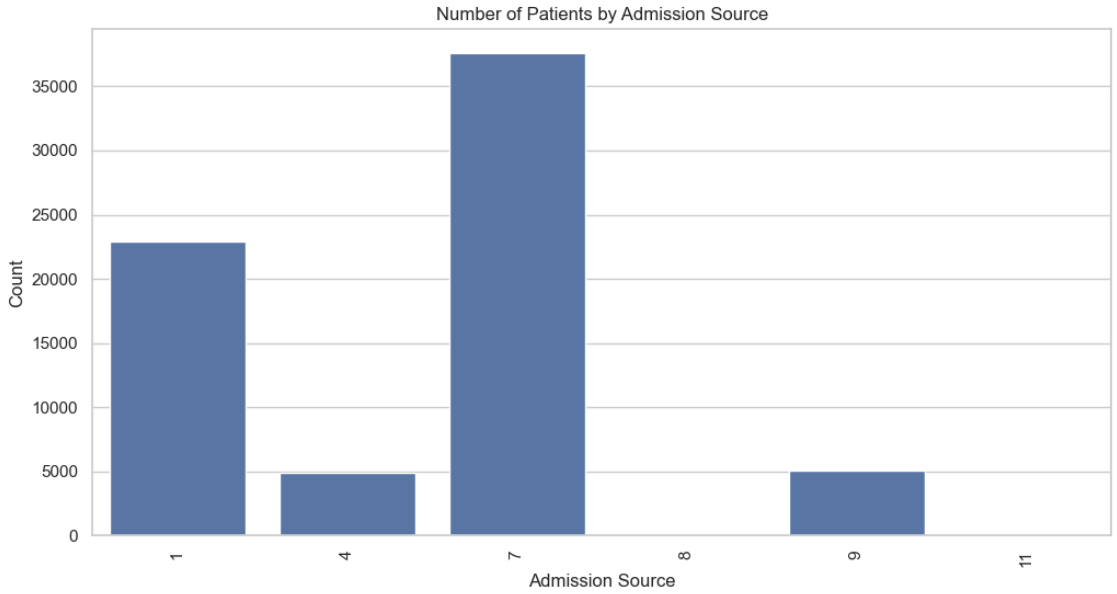
[30]:
```
plt.figure(figsize=(10, 6))
sns.countplot(x='admission_type_id', data=data)
plt.title('Number of Patients by Admission Type')
plt.xlabel('Admission Type')
plt.ylabel('Count')
plt.show()
```



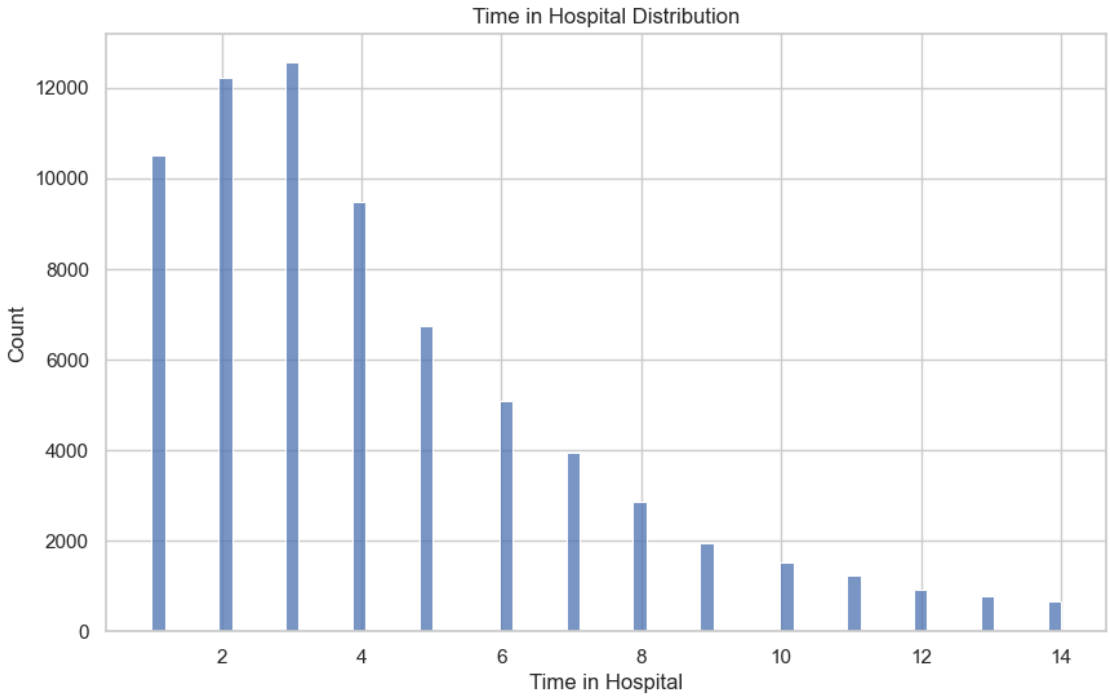
[31]:
```
plt.figure(figsize=(12, 6))
sns.countplot(x='discharge_disposition_id', data=data)
plt.title('Number of Patients by Discharge Disposition')
plt.xlabel('Discharge Disposition')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



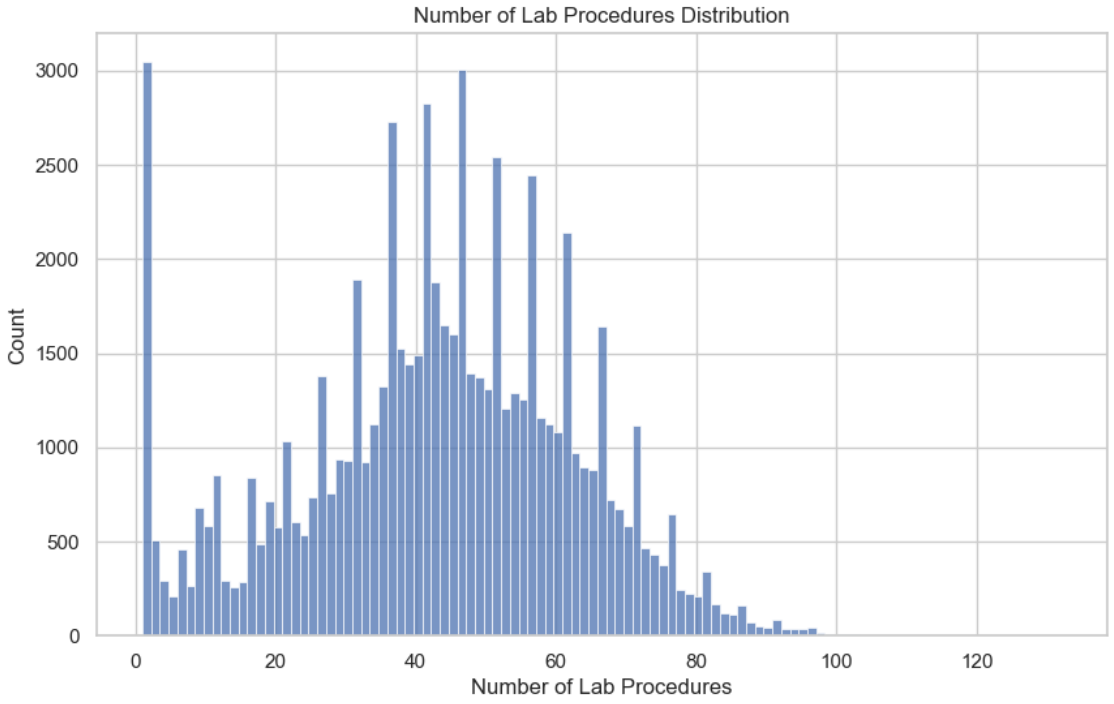
[32]:
```
plt.figure(figsize=(12, 6))
sns.countplot(x='admission_source_id', data=data)
plt.title('Number of Patients by Admission Source')
plt.xlabel('Admission Source')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



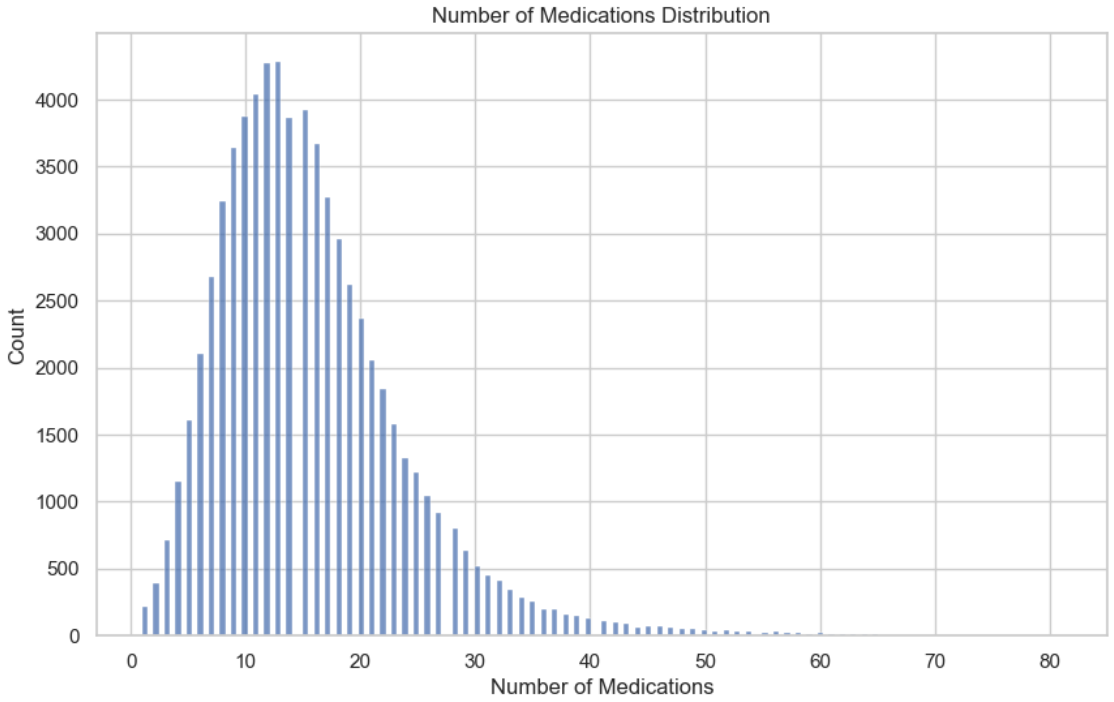
[33]:
```
plt.figure(figsize=(10, 6))
sns.histplot(data['time_in_hospital'])
plt.title('Time in Hospital Distribution')
plt.xlabel('Time in Hospital')
plt.ylabel('Count')
plt.show()
```



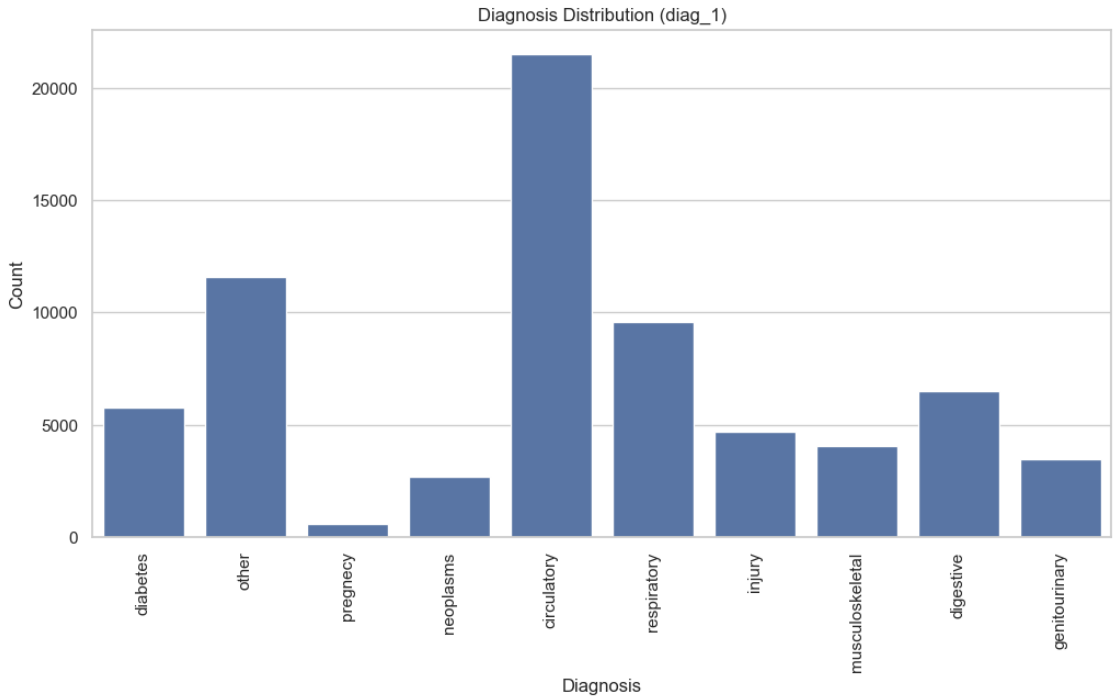
[34]:
```
plt.figure(figsize=(10, 6))
sns.histplot(data['num_lab_procedures'])
plt.title('Number of Lab Procedures Distribution')
plt.xlabel('Number of Lab Procedures')
plt.ylabel('Count')
plt.show()
```



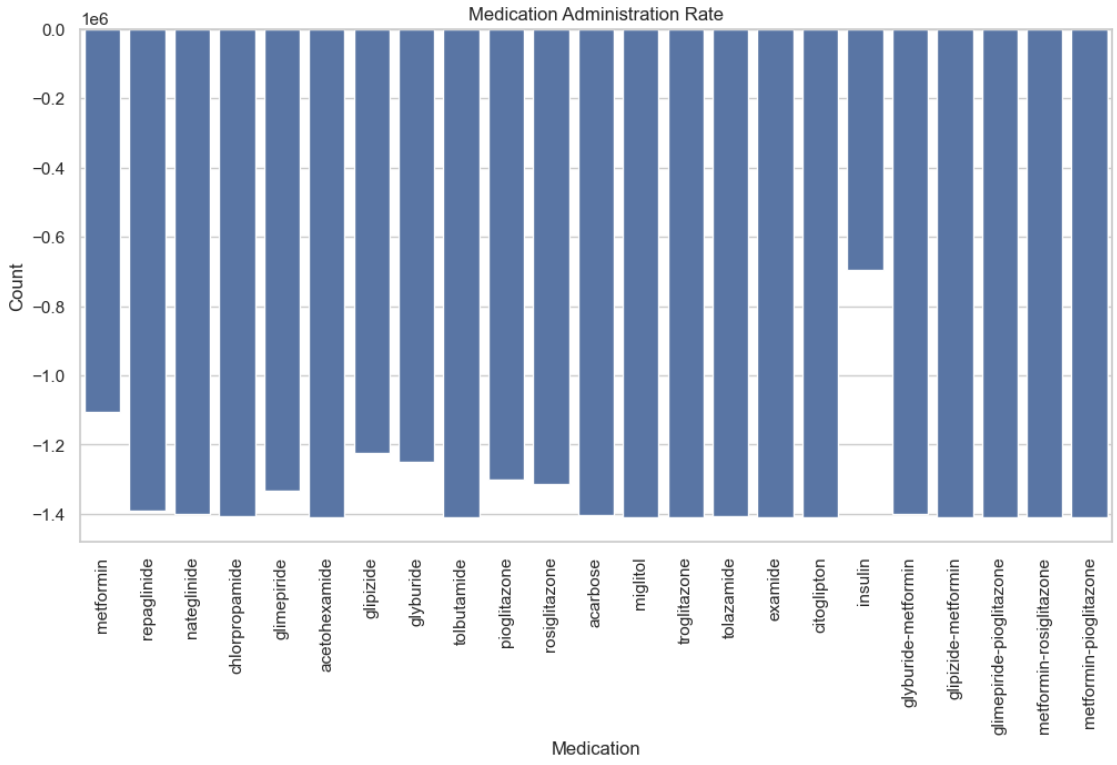
[35]:
```
plt.figure(figsize=(10, 6))
sns.histplot(data['num_medications'])
plt.title('Number of Medications Distribution')
plt.xlabel('Number of Medications')
plt.ylabel('Count')
plt.show()
```



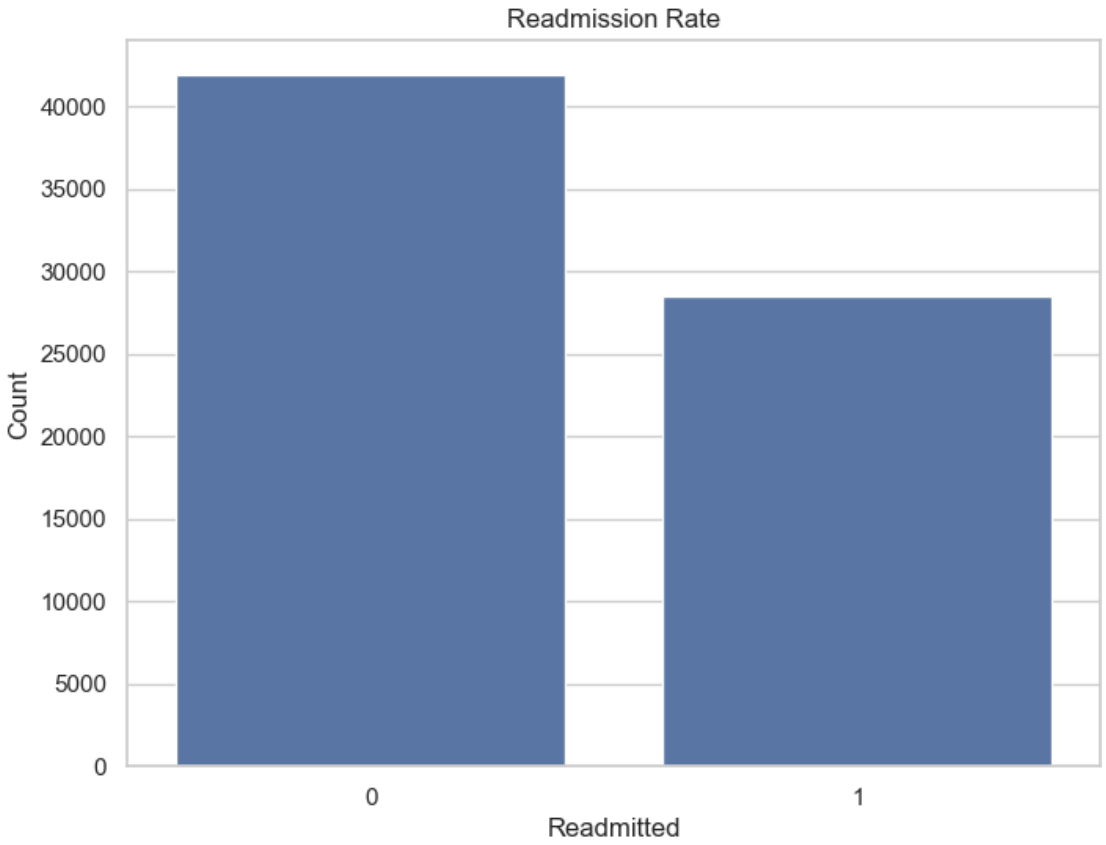
[36]:
```
plt.figure(figsize=(12, 6))
sns.countplot(x='diag_1', data=data)
plt.title('Diagnosis Distribution (diag_1)')
plt.xlabel('Diagnosis')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



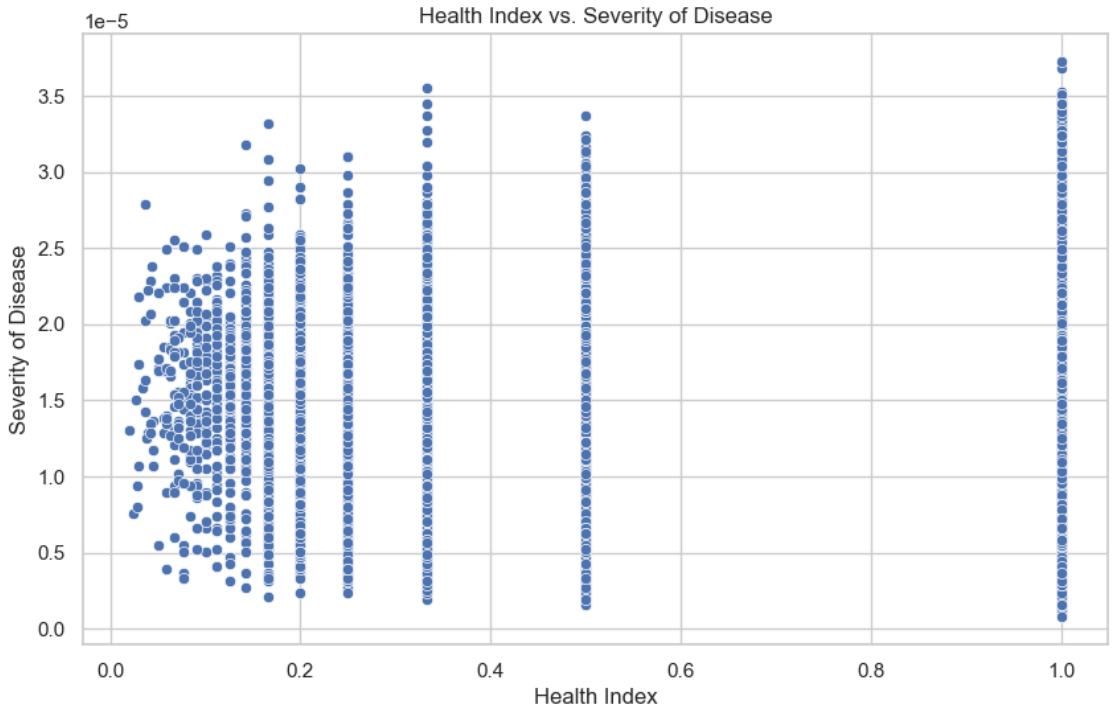
[37]:
```
medication_columns = ['metformin', 'repaglinide', 'nateglinide',␣
 ↪'chlorpropamide',
                      'glimepiride', 'acetohexamide', 'glipizide', 'glyburide',␣
 ↪'tolbutamide',
                      'pioglitazone', 'rosiglitazone', 'acarbose', 'miglitol',␣
 ↪'troglitazone',
                      'tolazamide', 'examide', 'citoglipton', 'insulin',
                      'glyburide-metformin', 'glipizide-metformin',
                      'glimepiride-pioglitazone', 'metformin-rosiglitazone',
                      'metformin-pioglitazone']
medication_data = data[medication_columns].sum().reset_index()
medication_data.columns = ['Medication', 'Count']
plt.figure(figsize=(12, 6))
sns.barplot(x='Medication', y='Count', data=medication_data)
plt.title('Medication Administration Rate')
plt.xlabel('Medication')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



[38]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x='readmitted', data=data)
plt.title('Readmission Rate')
plt.xlabel('Readmitted')
plt.ylabel('Count')
plt.show()
```



[39]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='health_index', y='severity_of_disease', data=data)
plt.title('Health Index vs. Severity of Disease')
plt.xlabel('Health Index')
plt.ylabel('Severity of Disease')
plt.show()
```



[ ]:

[ ]:

[40]:
```
data['age']
```

[40]:
```
0          5
1         15
2         25
3         35
4         45
          ..
101754    75
101755    45
101756    65
101758    85
101765    75
Name: age, Length: 70431, dtype: int64
```

[41]:
```
def map_age_to_group(age):
    if 0 <= age < 30:
        return 'Young'
    elif 30 <= age < 60:
        return 'Middle-Aged'
```

```
    else:
        return 'Senior'

# data['age'] = data['age'].map(replaceDict)
data['age_group'] = data['age'].apply(map_age_to_group)
print(data['age_group'].value_counts())
```

```
age_group
Senior         46722
Middle-Aged    21900
Young           1809
Name: count, dtype: int64
```

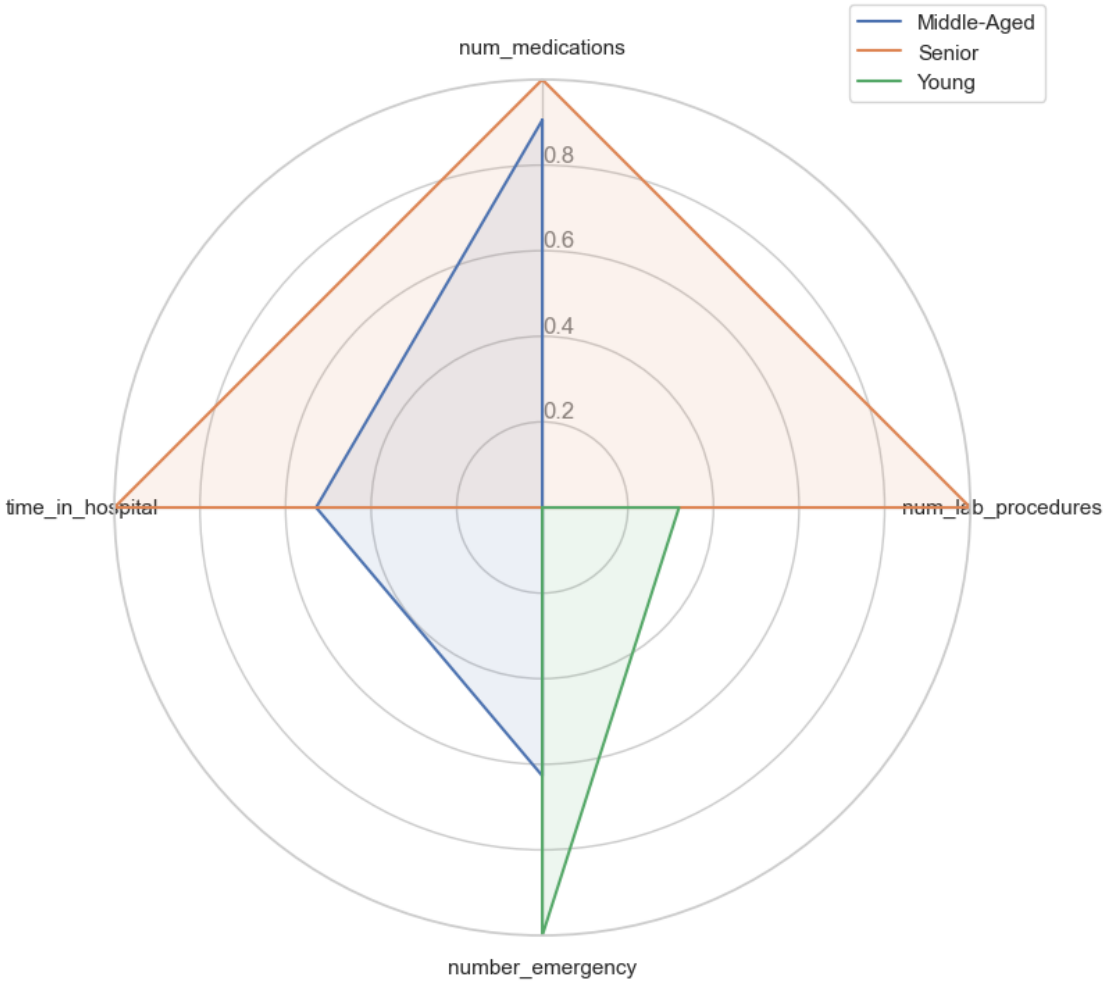
[42]:
```
categories = ['num_medications', 'num_lab_procedures', 'number_emergency',␣
 ↪'time_in_hospital']

average_metrics_by_age_group = data.groupby('age_group')[categories].mean().
 ↪reset_index()

print(average_metrics_by_age_group)
```

```
     age_group  num_medications  num_lab_procedures  number_emergency  \
0  Middle-Aged        15.429954           42.439498          0.136758
1       Senior        15.996704           43.150529          0.085998
2        Young        10.010503           42.666667          0.166943

   time_in_hospital
0          3.897032
1          4.506335
2          3.213930
```

[43]:
```
data_to_normalize = average_metrics_by_age_group[categories].values

scaler = MinMaxScaler()
normalized_data = scaler.fit_transform(data_to_normalize)

normalized_df = pd.DataFrame(normalized_data, columns=categories)
normalized_df['age_group'] = average_metrics_by_age_group['age_group']
```

[44]:
```
def create_radar_chart(data_frame, groups, categories):
    num_vars = len(categories)

    angles = np.linspace(0, 2 * np.pi, num_vars, endpoint=False).tolist()

    angles += angles[:1]

    fig, ax = plt.subplots(figsize=(8, 8), subplot_kw=dict(polar=True))
```

```
    ax.set_theta_offset(np.pi / 2)
    ax.set_theta_direction(-1)
    plt.xticks(angles[:-1], categories)

    ax.set_rlabel_position(0)
    plt.yticks([0.2, 0.4, 0.6, 0.8], ["0.2", "0.4", "0.6", "0.8"],␣
 ↪color="grey", size=12)
    plt.ylim(0,1)

    for idx, group in enumerate(groups):
        values = data_frame.loc[data_frame['age_group'] == group].
 ↪drop('age_group', axis=1).values.flatten().tolist()
        values += values[:1]
        ax.plot(angles, values, label=group)
        ax.fill(angles, values, alpha=0.1)

    plt.legend(loc='upper right', bbox_to_anchor=(1.1, 1.1))

create_radar_chart(normalized_df, average_metrics_by_age_group['age_group'],␣
 ↪categories)
plt.show()
```



[ ]:

[45]:
```
numeric_features = ['age','time_in_hospital', 'num_lab_procedures',␣
↪'num_procedures', 'num_medications',
                    'number_outpatient', 'number_emergency',␣
↪'number_inpatient', 'number_diagnoses',
                    'health_index', 'severity_of_disease', 'number_of_changes',␣
↪'A1Cresult', 'max_glu_serum',
                    ␣
↪'metformin','repaglinide','nateglinide','chlorpropamide','glimepiride','acetohexamide',\
          ␣
↪'glipizide','glyburide','tolbutamide','pioglitazone','rosiglitazone','acarbose','miglitol',
          ␣
↪'troglitazone','tolazamide','examide','citoglipton','insulin','glyburide-metformin','glipiz
```

```
␣
↪'glimepiride-pioglitazone','metformin-rosiglitazone','metformin-pioglitazone',␣
↪'change', 'diabetesMed']


categorical_features = []
rejected_features = []

for col in data.columns :
    if col not in numeric_features :
        categorical_features.append(col)

for col in categorical_features :
    data_crosstab = pd.crosstab(data['readmitted'],
                                data[col],
                                margins = False)

    stat, p, dof, expected = scipy.stats.chi2_contingency(data_crosstab)
    if p < 0.4 :
            print(p, col, 'is significant')
    else:
        print(p, col, 'is not significant')
        rejected_features.append(col)

print('\n\n', rejected_features)
data.drop(rejected_features, axis = 1, inplace=True)
```

```
1.2094958159533304e-05 gender is significant
5.93295699577205e-64 admission_type_id is significant
6.724160950719454e-32 discharge_disposition_id is significant
4.327667904789753e-108 admission_source_id is significant
5.063372923699428e-32 medical_specialty is significant
8.536250735746521e-62 diag_1 is significant
8.536250735746521e-62 diag_2 is significant
1.4328222569529509e-36 diag_3 is significant
0.0 readmitted is significant
3.057366271920308e-101 age_labels is significant
9.4160561762657e-76 age_group is significant


 []
```

[46]:
```
rejected_features = []
for col in numeric_features :
    rho , pval = scipy.stats.spearmanr(data['readmitted'], data[col])
    print(col, rho, pval)
```

```
print('\n\n')
for col in numeric_features :
    rho , pval = scipy.stats.spearmanr(data['readmitted'], data[col])
    if pval < 0.4 :
        print(col, 'is significant')
    else :
        print(col, 'is not significant')
        rejected_features.append(col)

#rejected_features.remove('max_glu_serum')
#rejected_features.remove('A1Cresult')
print(rejected_features)
data.drop(rejected_features, axis = 1, inplace=True)
```

```
age 0.07135095583173226 3.683089700271019e-80
time_in_hospital 0.06777643981394682 1.7023616788306636e-72
num_lab_procedures 0.05475941845560129 6.453546138251782e-48
num_procedures -0.028817825076789774 2.0196771653591713e-14
num_medications 0.06062970528831338 2.3548529960357112e-58
number_outpatient 0.08573905864893631 5.054707370065862e-115
number_emergency 0.0899417082340514 2.0118449675641115e-126
number_inpatient 0.14643684449123884 0.0
number_diagnoses 0.10435461008286794 1.0034232068159713e-169
health_index -0.13454813666646742 8.943785818004163e-282
severity_of_disease 0.0730349404508841 6.578749868776572e-84
number_of_changes 0.04178300458245632 1.3509740985122952e-28
A1Cresult -0.01916599178342482 3.64138782358839e-07
max_glu_serum 0.018280109904924083 1.224396239705935e-06
metformin -0.022499839068227927 2.3458376493645316e-09
repaglinide 0.020469095162959115 5.5497410486889906e-08
nateglinide 0.004601999634355191 0.22197166411999703
chlorpropamide 0.0020398121883154147 0.5882777773070488
glimepiride 0.0036424505298749623 0.33371895599214574
acetohexamide 0.004566085967021777 0.22560003922484595
glipizide 0.026946120347313723 8.529894649537796e-13
glyburide 0.008350961995168299 0.02667454244733254
tolbutamide -0.0016515127532465371 0.6611798153175513
pioglitazone 0.015506626236655805 3.8641970440543765e-05
rosiglitazone 0.020469805585416872 5.5438765129801194e-08
acarbose 0.011154319242551634 0.003073731051653799
miglitol 0.008407012945125691 0.025672949862151302
troglitazone 0.0034772089414928224 0.35611284289499634
tolazamide -0.0030186163095068724 0.42307760965257346
examide nan nan
citoglipton nan nan
insulin 0.030323780170860744 8.327748086459169e-16
glyburide-metformin 0.002093082301771121 0.5785728814678739
```

```
glipizide-metformin 0.0033775131458245546 0.3700715011478256
glimepiride-pioglitazone nan nan
metformin-rosiglitazone -0.004397605922094254 0.24318674364764511
metformin-pioglitazone -0.00310955489276415 0.4092426538559898
change 0.036866794252806194 1.2782015578963928e-22
diabetesMed 0.061665344610234964 2.6328206696789474e-60



age is significant
time_in_hospital is significant
num_lab_procedures is significant
num_procedures is significant
num_medications is significant
number_outpatient is significant
number_emergency is significant
number_inpatient is significant
number_diagnoses is significant
health_index is significant
severity_of_disease is significant
number_of_changes is significant
A1Cresult is significant
max_glu_serum is significant
metformin is significant
repaglinide is significant
nateglinide is significant
chlorpropamide is not significant
glimepiride is significant
acetohexamide is significant
glipizide is significant
glyburide is significant
tolbutamide is not significant
pioglitazone is significant
rosiglitazone is significant
acarbose is significant
miglitol is significant
troglitazone is significant
tolazamide is not significant
examide is not significant
citoglipton is not significant
insulin is significant
glyburide-metformin is not significant
glipizide-metformin is significant
glimepiride-pioglitazone is not significant
metformin-rosiglitazone is significant
metformin-pioglitazone is not significant
change is significant
diabetesMed is significant
```

```
['chlorpropamide', 'tolbutamide', 'tolazamide', 'examide', 'citoglipton',
'glyburide-metformin', 'glimepiride-pioglitazone', 'metformin-pioglitazone']
```

[47]:
```
data.columns
```

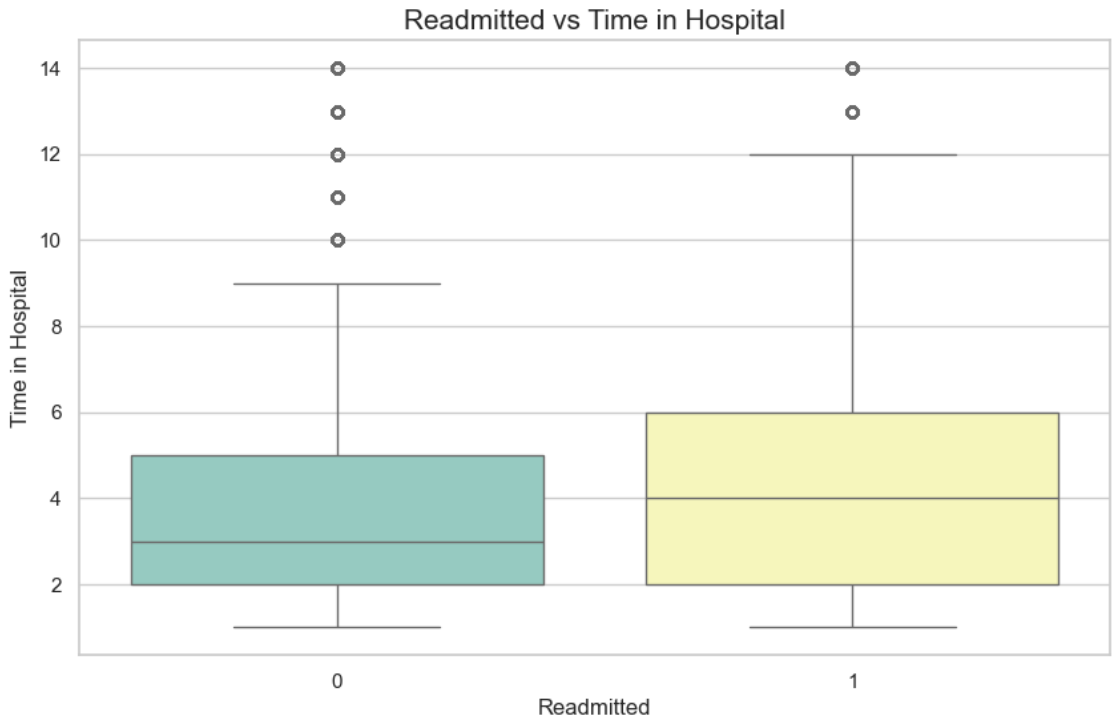
[47]:
```
Index(['gender', 'age', 'admission_type_id', 'discharge_disposition_id',
       'admission_source_id', 'time_in_hospital', 'medical_specialty',
       'num_lab_procedures', 'num_procedures', 'num_medications',
       'number_outpatient', 'number_emergency', 'number_inpatient', 'diag_1',
       'diag_2', 'diag_3', 'number_diagnoses', 'max_glu_serum', 'A1Cresult',
       'metformin', 'repaglinide', 'nateglinide', 'glimepiride',
       'acetohexamide', 'glipizide', 'glyburide', 'pioglitazone',
       'rosiglitazone', 'acarbose', 'miglitol', 'troglitazone', 'insulin',
       'glipizide-metformin', 'metformin-rosiglitazone', 'change',
       'diabetesMed', 'readmitted', 'health_index', 'severity_of_disease',
       'number_of_changes', 'age_labels', 'age_group'],
      dtype='object')
```

[ ]:

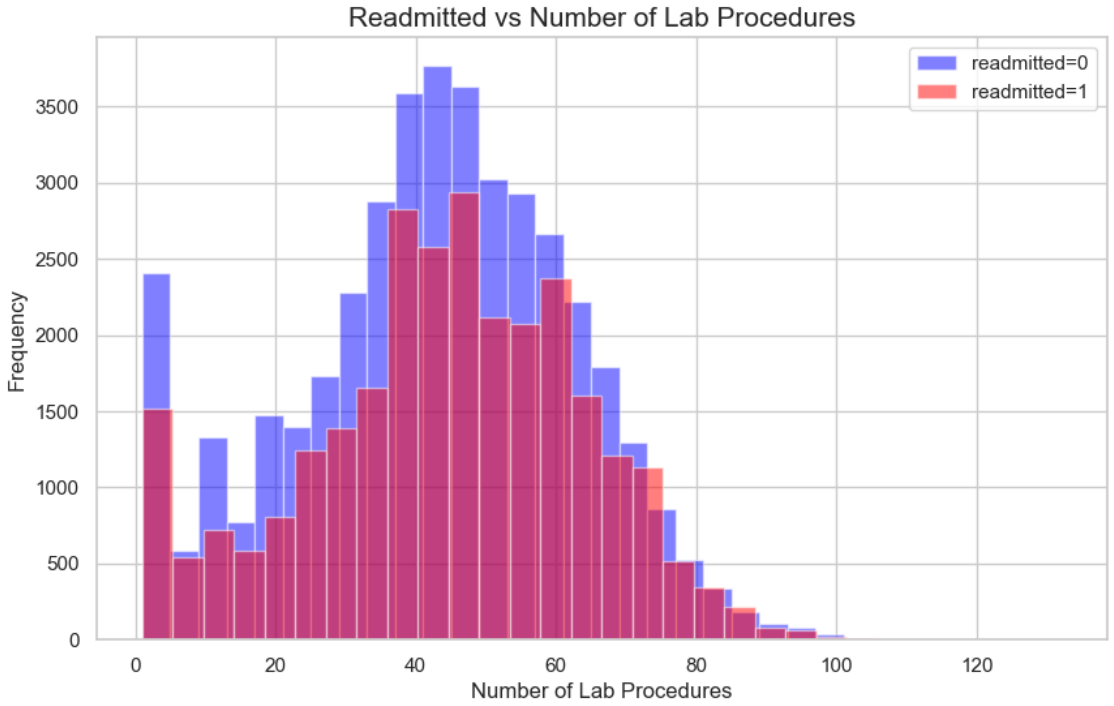
[ ]:

[48]:
```
# plt.figure(figsize=(10, 6))
# sns.countplot(x='gender', hue='readmitted', data=data, palette='Set2')
# plt.title('Readmitted vs Gender', fontsize=15)
# plt.xlabel('Gender', fontsize=12)
# plt.ylabel('Count', fontsize=12)
# plt.show()
```

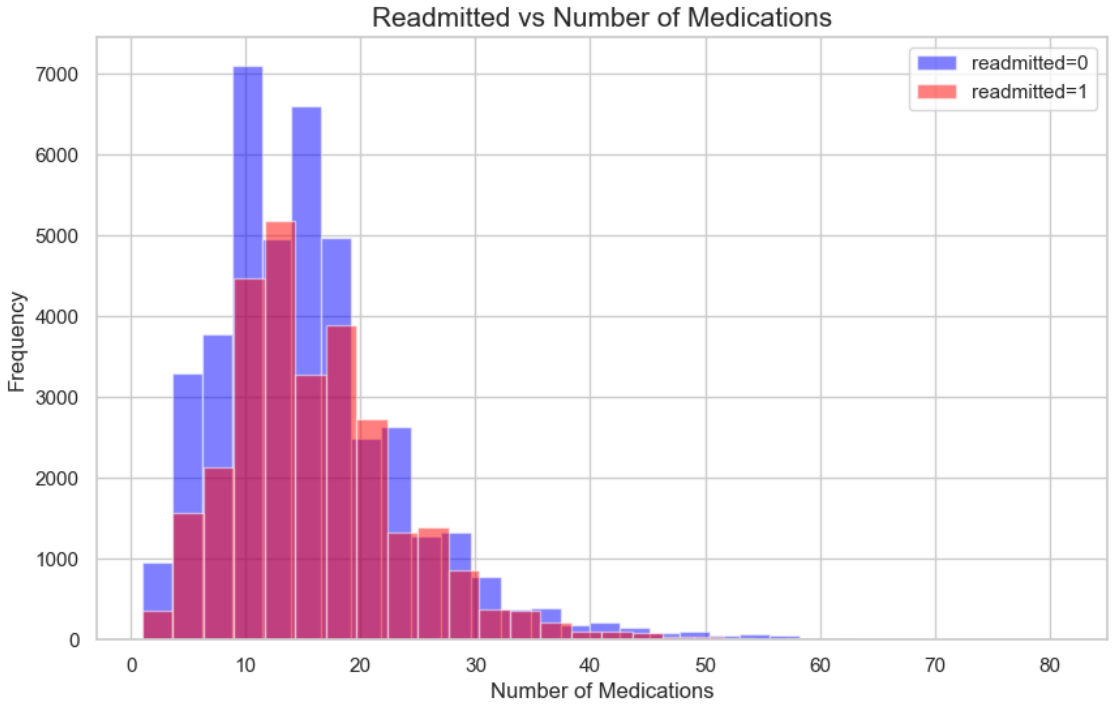
[49]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='readmitted', y='time_in_hospital', data=data, palette='Set3')
plt.title('Readmitted vs Time in Hospital', fontsize=15)
plt.xlabel('Readmitted', fontsize=12)
plt.ylabel('Time in Hospital', fontsize=12)
plt.show()
```



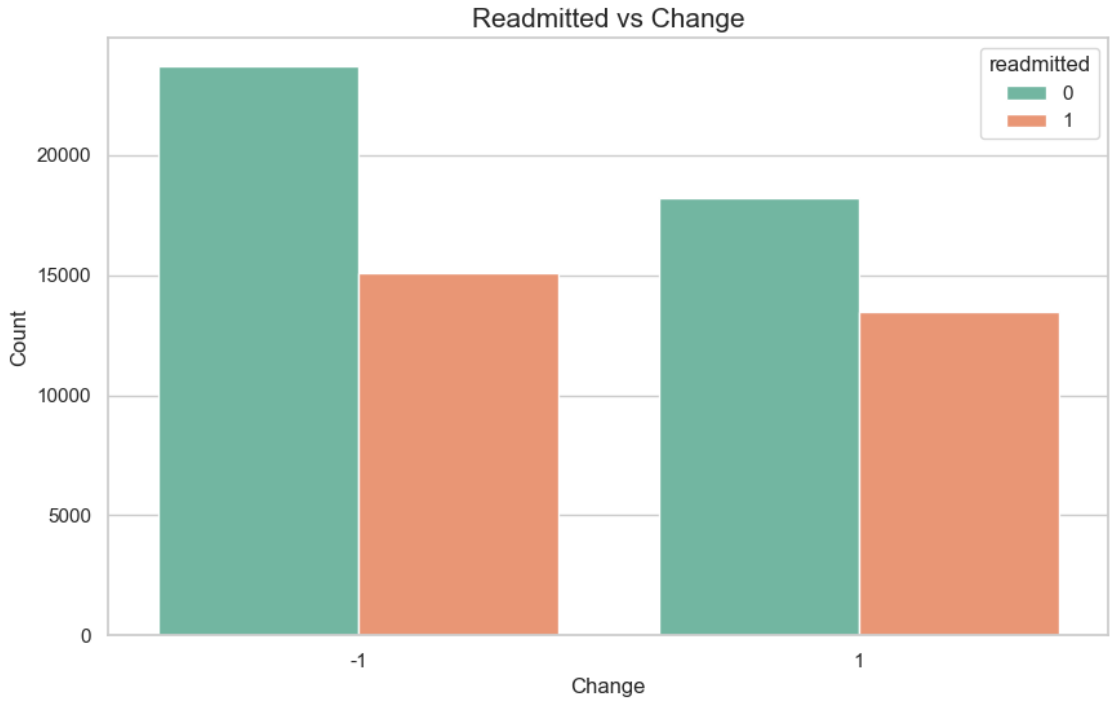
[50]:
```
plt.figure(figsize=(10, 6))
data[data['readmitted'] == 0]['num_lab_procedures'].hist(alpha=0.5,␣
 ↪color='blue', bins=30, label='readmitted=0')
data[data['readmitted'] == 1]['num_lab_procedures'].hist(alpha=0.5,␣
 ↪color='red', bins=30, label='readmitted=1')
plt.title('Readmitted vs Number of Lab Procedures', fontsize=15)
plt.xlabel('Number of Lab Procedures', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.legend()
plt.show()
```



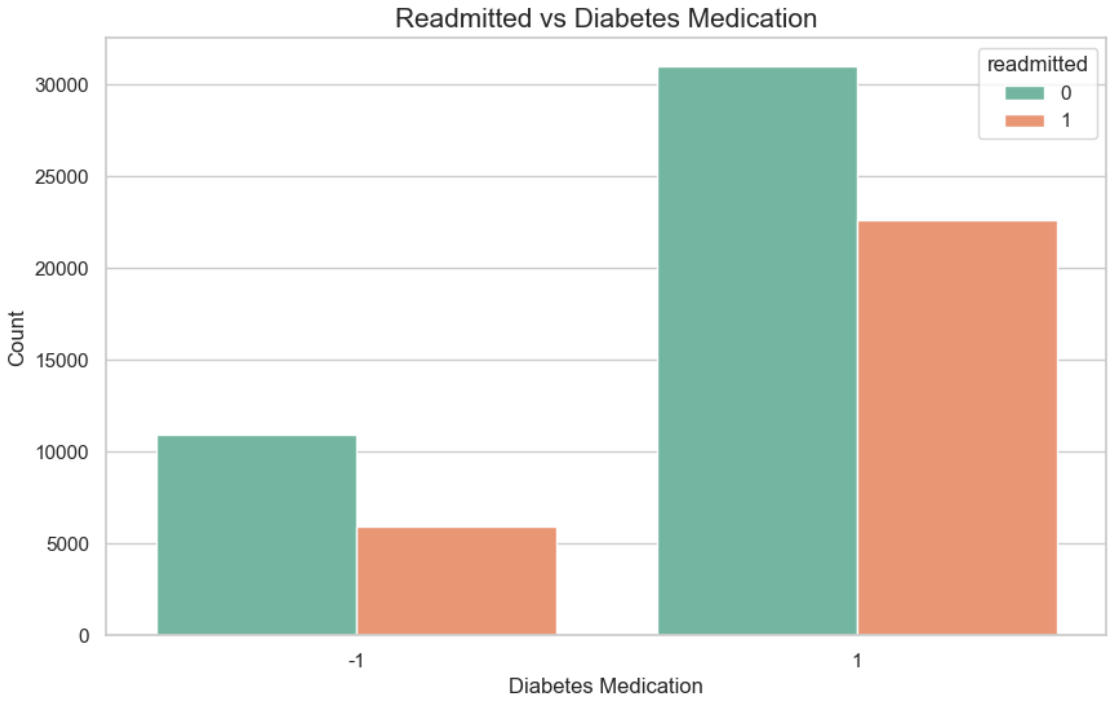
[51]:
```
plt.figure(figsize=(10, 6))
data[data['readmitted'] == 0]['num_medications'].hist(alpha=0.5, color='blue',␣
 ↪bins=30, label='readmitted=0')
data[data['readmitted'] == 1]['num_medications'].hist(alpha=0.5, color='red',␣
 ↪bins=30, label='readmitted=1')
plt.title('Readmitted vs Number of Medications', fontsize=15)
plt.xlabel('Number of Medications', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.legend()
plt.show()
```



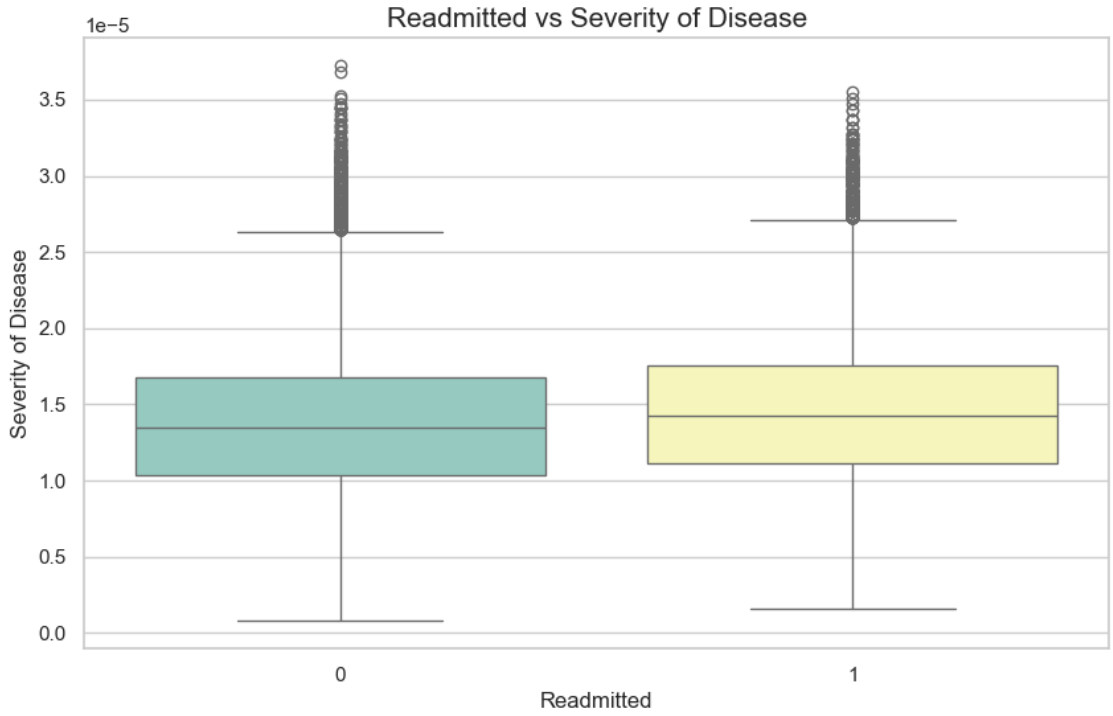
[52]:
```
plt.figure(figsize=(10, 6))
sns.countplot(x='change', hue='readmitted', data=data, palette='Set2')
plt.title('Readmitted vs Change', fontsize=15)
plt.xlabel('Change', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.show()
```



[53]:
```
plt.figure(figsize=(10, 6))
sns.countplot(x='diabetesMed', hue='readmitted', data=data, palette='Set2')
plt.title('Readmitted vs Diabetes Medication', fontsize=15)
plt.xlabel('Diabetes Medication', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.show()
```



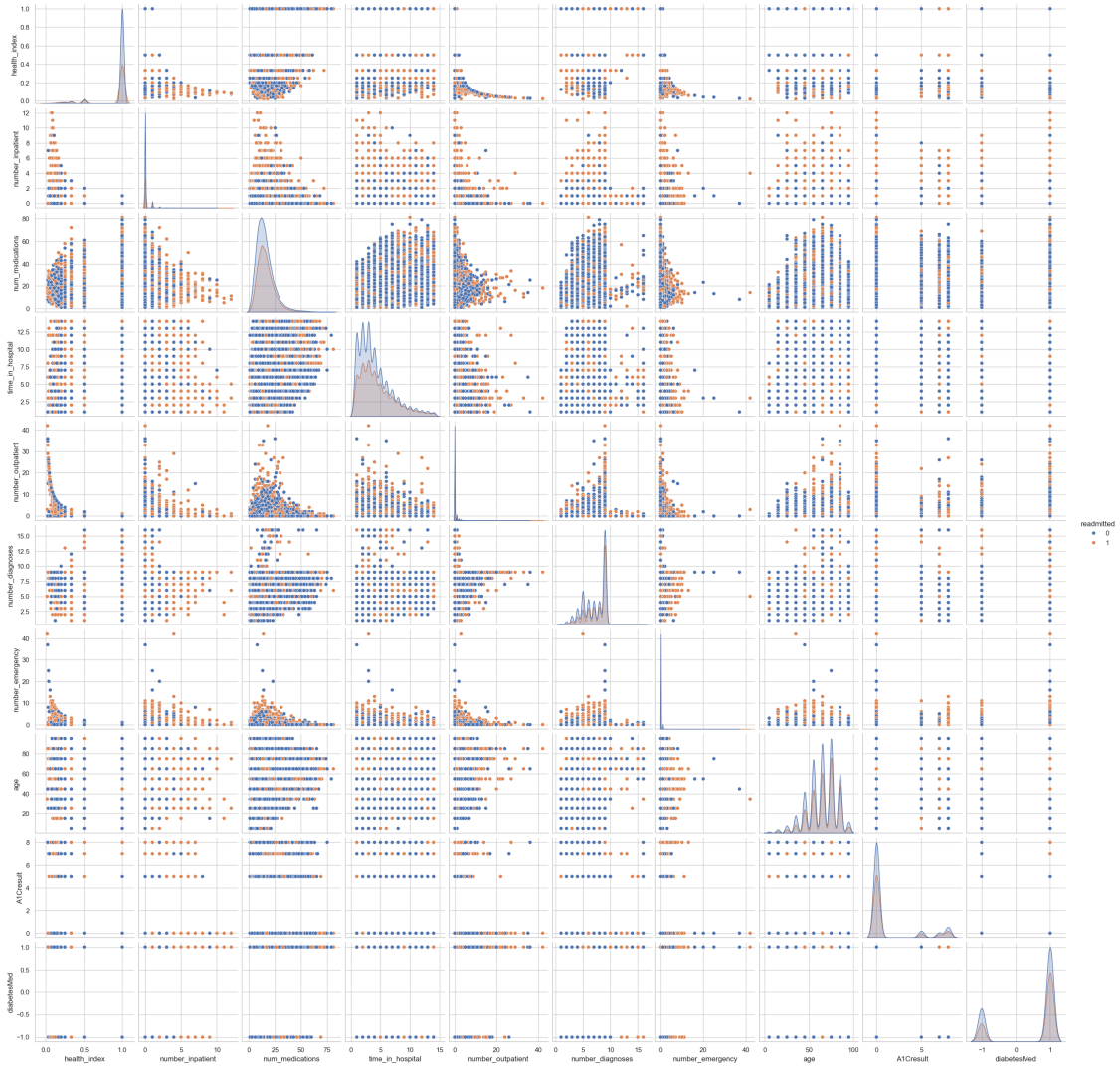
[54]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='readmitted', y='severity_of_disease', data=data, palette='Set3')
plt.title('Readmitted vs Severity of Disease', fontsize=15)
plt.xlabel('Readmitted', fontsize=12)
plt.ylabel('Severity of Disease', fontsize=12)
plt.show()
```



[ ]:

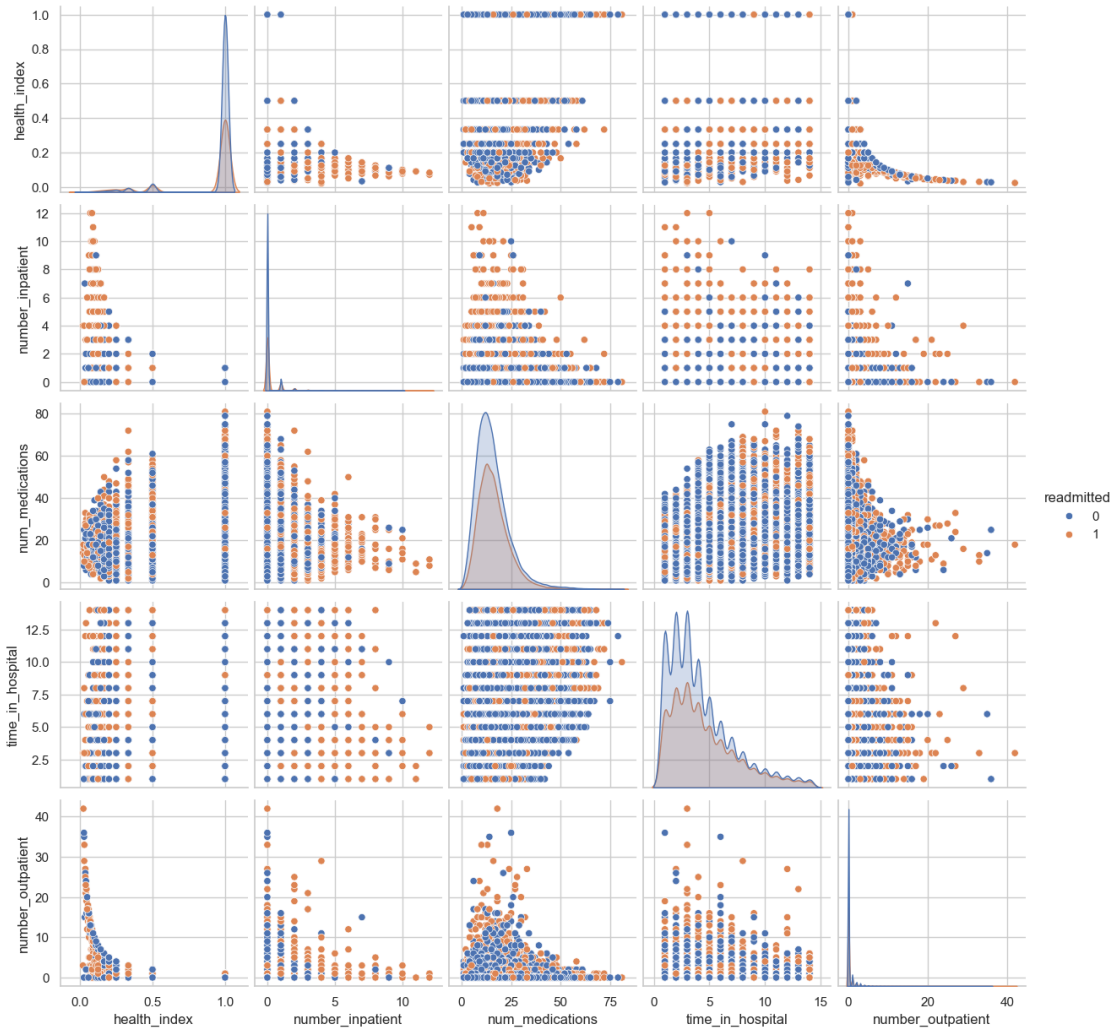
[55]:
```
selected_features = ['readmitted', 'health_index', 'number_inpatient',␣
 ↪'num_medications', 'time_in_hospital', 'number_outpatient',␣
 ↪'number_diagnoses','number_emergency', 'age', 'A1Cresult', 'diabetesMed']
sns.pairplot(data[selected_features], hue='readmitted')
```

[55]: <seaborn.axisgrid.PairGrid at 0x209cb6a9ed0>



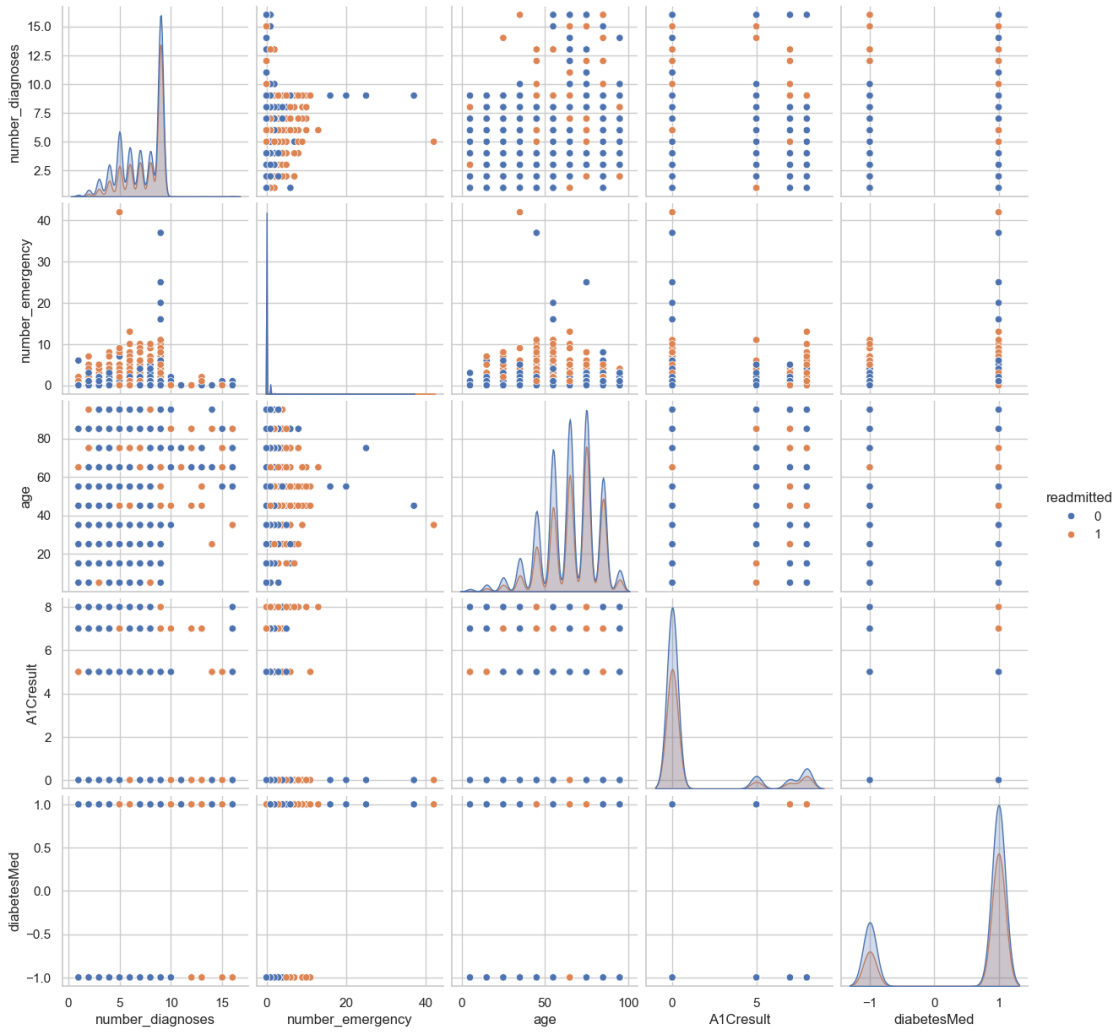
[56]:
```
selected_features_1 = ['readmitted', 'health_index', 'number_inpatient',␣
 ↪'num_medications', 'time_in_hospital', 'number_outpatient']
sns.pairplot(data[selected_features_1], hue='readmitted')
```

[56]: <seaborn.axisgrid.PairGrid at 0x209c9f047d0>



[57]:
```
selected_features_2 = ['readmitted', 'number_diagnoses','number_emergency',␣
↪'age', 'A1Cresult', 'diabetesMed']
sns.pairplot(data[selected_features_2], hue='readmitted')
```

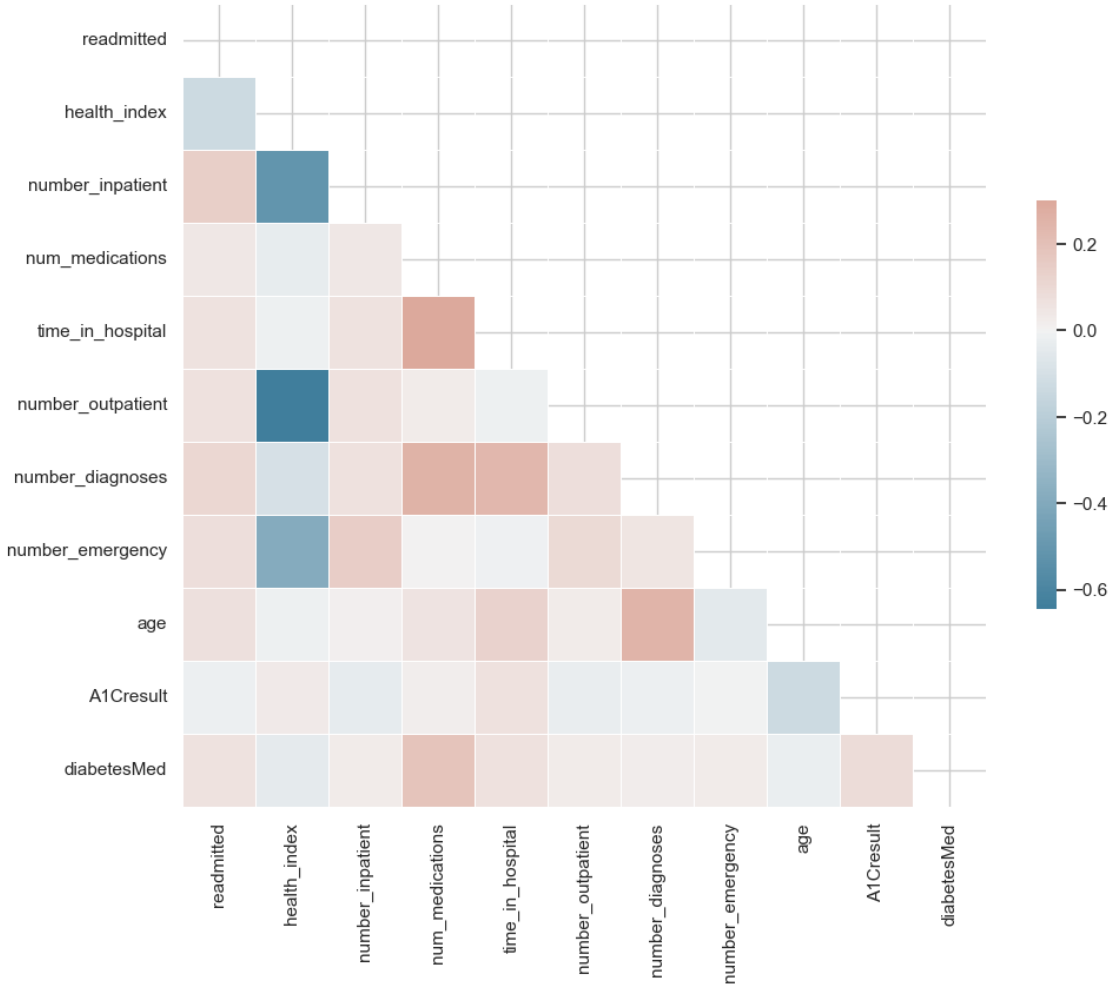
[57]: <seaborn.axisgrid.PairGrid at 0x209eb43d350>



[ ]:

[58]:
```
corr = data[selected_features].corr()

mask = np.triu(np.ones_like(corr, dtype=bool))

f, ax = plt.subplots(figsize=(11, 9))

cmap = sns.diverging_palette(230, 20, as_cmap=True)

sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
```

[58]: <Axes: >



[ ]:

[59]:
```
cat_new = []
for col in categorical_features :
    if col in data.columns :
        print(col, '-->', np.unique(data[col]), '\n')
        cat_new.append(col)

numeric_features = []

for col in data.columns :
    if col  not in cat_new :
        numeric_features.append(col)
```

```
gender --> ['Female' 'Male']

admission_type_id --> [1 3 4 5]
```

```
discharge_disposition_id --> [ 1  2  7 10 18 27 28]

admission_source_id --> [ 1  4  7  8  9 11]

medical_specialty --> ['high_freq' 'low_freq' 'missing' 'neurology' 'pediatrics'
'psychic'
 'surgery' 'ungrouped']

diag_1 --> ['circulatory' 'diabetes' 'digestive' 'genitourinary' 'injury'
 'musculoskeletal' 'neoplasms' 'other' 'pregnecy' 'respiratory']

diag_2 --> ['circulatory' 'diabetes' 'digestive' 'genitourinary' 'injury'
 'musculoskeletal' 'neoplasms' 'other' 'pregnecy' 'respiratory']

diag_3 --> ['circulatory' 'diabetes' 'digestive' 'genitourinary' 'injury'
 'musculoskeletal' 'neoplasms' 'other' 'pregnecy' 'respiratory']

readmitted --> [0 1]

age_labels --> ['[0-10)' '[10-20)' '[20-30)' '[30-40)' '[40-50)' '[50-60)'
'[60-70)'
 '[70-80)' '[80-90)' '[90-100)']

age_group --> ['Middle-Aged' 'Senior' 'Young']
```

[ ]:

[60]:
```
data[cat_new]
```

[60]:
```
        gender  admission_type_id  discharge_disposition_id  \
0       Female                  5                        18
1       Female                  1                         1
2       Female                  1                         1
3         Male                  1                         1
4         Male                  1                         1
…          …                  …                         …
101754  Female                  1                         1
101755  Female                  1                         1
101756  Female                  1                         1
101758  Female                  1                         1
101765    Male                  1                         1

        admission_source_id medical_specialty         diag_1         diag_2  \
0                         1       pediatrics       diabetes       diabetes
1                         7          missing          other          other
```

```
2                        7           missing       pregnecy       pregnecy
3                        7           missing          other          other
4                        7           missing      neoplasms      neoplasms
…                      …             …            …            …
101754                   7           missing      digestive      digestive
101755                   7           missing  genitourinary  genitourinary
101756                   7           missing         injury         injury
101758                   7           missing          other          other
101765                   7           missing      digestive      digestive

             diag_3  readmitted age_labels    age_group
0          diabetes           0     [0-10)        Young
1             other           1    [10-20)        Young
2             other           0    [20-30)        Young
3       circulatory           0    [30-40)  Middle-Aged
4          diabetes           0    [40-50)  Middle-Aged
…               …         …        …           …
101754     diabetes           1    [70-80)       Senior
101755  respiratory           1    [40-50)  Middle-Aged
101756  circulatory           1    [60-70)       Senior
101758        other           0    [80-90)       Senior
101765    digestive           0    [70-80)       Senior

[70431 rows x 11 columns]
```

[ ]:

[61]:
```
datare = data['readmitted']
```

[62]:
```
data.drop(columns = 'readmitted', inplace = True)
```

[63]:
```
data
```

[63]:
```
        gender  age  admission_type_id  discharge_disposition_id  \
0       Female    5                  5                        18
1       Female   15                  1                         1
2       Female   25                  1                         1
3         Male   35                  1                         1
4         Male   45                  1                         1
…        …  …               …                        …
101754  Female   75                  1                         1
101755  Female   45                  1                         1
101756  Female   65                  1                         1
101758  Female   85                  1                         1
101765    Male   75                  1                         1

        admission_source_id  time_in_hospital medical_specialty  \
```

```
0                        1                 1        pediatrics
1                        7                 3           missing
2                        7                 2           missing
3                        7                 2           missing
4                        7                 1           missing
…                      …                 …             …
101754                   7                 9           missing
101755                   7                14           missing
101756                   7                 2           missing
101758                   7                 5           missing
101765                   7                 6           missing

        num_lab_procedures  num_procedures  num_medications  …  insulin  \
0                       41               0                1  …      -20
1                       59               0               18  …       10
2                       11               5               13  …      -20
3                       44               1               16  …       10
4                       51               0                8  …        0
…                     …                …                …  …      …
101754                  50               2               33  …        0
101755                  73               6               26  …       10
101756                  46               6               17  …        0
101758                  76               1               22  …       10
101765                  13               3                3  …      -20

        glipizide-metformin  metformin-rosiglitazone  change  diabetesMed  \
0                       -20                      -20      -1           -1
1                       -20                      -20       1            1
2                       -20                      -20      -1            1
3                       -20                      -20       1            1
4                       -20                      -20       1            1
…                     …                        …     …         …
101754                  -20                      -20       1            1
101755                  -20                      -20       1            1
101756                  -20                      -20      -1            1
101758                  -20                      -20       1            1
101765                  -20                      -20      -1           -1

       health_index  severity_of_disease  number_of_changes  age_labels  \
0          1.000000             0.000009                  0      [0-10)
1          1.000000             0.000017                  1     [10-20)
2          0.333333             0.000007                  0     [20-30)
3          1.000000             0.000014                  1     [30-40)
4          1.000000             0.000013                  0     [40-50)
…             …                   …                    …         …
101754     1.000000             0.000020                  1     [70-80)
101755     1.000000             0.000025                  1     [40-50)
```

```
101756     0.333333             0.000016                  0     [60-70)
101758     1.000000             0.000022                  1     [80-90)
101765     1.000000             0.000007                  0     [70-80)

          age_group
0             Young
1             Young
2             Young
3       Middle-Aged
4       Middle-Aged
…               …
101754       Senior
101755  Middle-Aged
101756       Senior
101758       Senior
101765       Senior

[70431 rows x 41 columns]
```

[64]:
```
columns_to_encode = [col for col in cat_new if col != 'readmitted']
data_encoded = pd.get_dummies(data, columns=columns_to_encode)
```

[65]:
```
data_encoded
```

[65]:
```
         age  time_in_hospital  num_lab_procedures  num_procedures  \
0          5                 1                  41               0
1         15                 3                  59               0
2         25                 2                  11               5
3         35                 2                  44               1
4         45                 1                  51               0
…     …                  …                  …               …
101754   75                 9                  50               2
101755   45                14                  73               6
101756   65                 2                  46               6
101758   85                 5                  76               1
101765   75                 6                  13               3

         num_medications  number_outpatient  number_emergency  \
0                      1                  0                 0
1                     18                  0                 0
2                     13                  2                 0
3                     16                  0                 0
4                      8                  0                 0
…                    …                  …               …
101754                33                  0                 0
101755                26                  0                 1
101756                17                  1                 1
```

```
101758              22                   0                   1
101765               3                   0                   0

        number_inpatient  number_diagnoses  max_glu_serum  …  \
0                      0                 1              0  …
1                      0                 9              0  …
2                      1                 6              0  …
3                      0                 7              0  …
4                      0                 5              0  …
…                    …                 …              …  …
101754                 0                 9              0  …
101755                 0                 9              0  …
101756                 1                 9              0  …
101758                 0                 9              0  …
101765                 0                 9              0  …

        age_labels_[30-40)  age_labels_[40-50)  age_labels_[50-60)  \
0                    False               False               False
1                    False               False               False
2                    False               False               False
3                     True               False               False
4                    False                True               False
…                    …                 …                   …
101754               False               False               False
101755               False                True               False
101756               False               False               False
101758               False               False               False
101765               False               False               False

        age_labels_[60-70)  age_labels_[70-80)  age_labels_[80-90)  \
0                    False               False               False
1                    False               False               False
2                    False               False               False
3                    False               False               False
4                    False               False               False
…                    …                 …                   …
101754               False                True               False
101755               False               False               False
101756                True               False               False
101758               False               False                True
101765               False                True               False

        age_labels_[90-100)  age_group_Middle-Aged  age_group_Senior  \
0                     False                  False             False
1                     False                  False             False
2                     False                  False             False
3                     False                   True             False
```

```
4                    False                    True             False
…                    …                     …                …
101754               False                   False             True
101755               False                    True             False
101756               False                   False             True
101758               False                   False             True
101765               False                   False             True

        age_group_Young
0                  True
1                  True
2                  True
3                 False
4                 False
…                 …
101754            False
101755            False
101756            False
101758            False
101765            False

[70431 rows x 101 columns]
```

[66]:
```
data_encoded.columns
```

[66]:
```
Index(['age', 'time_in_hospital', 'num_lab_procedures', 'num_procedures',
       'num_medications', 'number_outpatient', 'number_emergency',
       'number_inpatient', 'number_diagnoses', 'max_glu_serum',
       …
       'age_labels_[30-40)', 'age_labels_[40-50)', 'age_labels_[50-60)',
       'age_labels_[60-70)', 'age_labels_[70-80)', 'age_labels_[80-90)',
       'age_labels_[90-100)', 'age_group_Middle-Aged', 'age_group_Senior',
       'age_group_Young'],
      dtype='object', length=101)
```

[67]:
```
concatenated_data = pd.concat([data_encoded, datare], axis=1)
```

[68]:
```
concatenated_data.columns
```

[68]:
```
Index(['age', 'time_in_hospital', 'num_lab_procedures', 'num_procedures',
       'num_medications', 'number_outpatient', 'number_emergency',
       'number_inpatient', 'number_diagnoses', 'max_glu_serum',
       …
       'age_labels_[40-50)', 'age_labels_[50-60)', 'age_labels_[60-70)',
       'age_labels_[70-80)', 'age_labels_[80-90)', 'age_labels_[90-100)',
       'age_group_Middle-Aged', 'age_group_Senior', 'age_group_Young',
       'readmitted'],
```

```
      dtype='object', length=102)
```

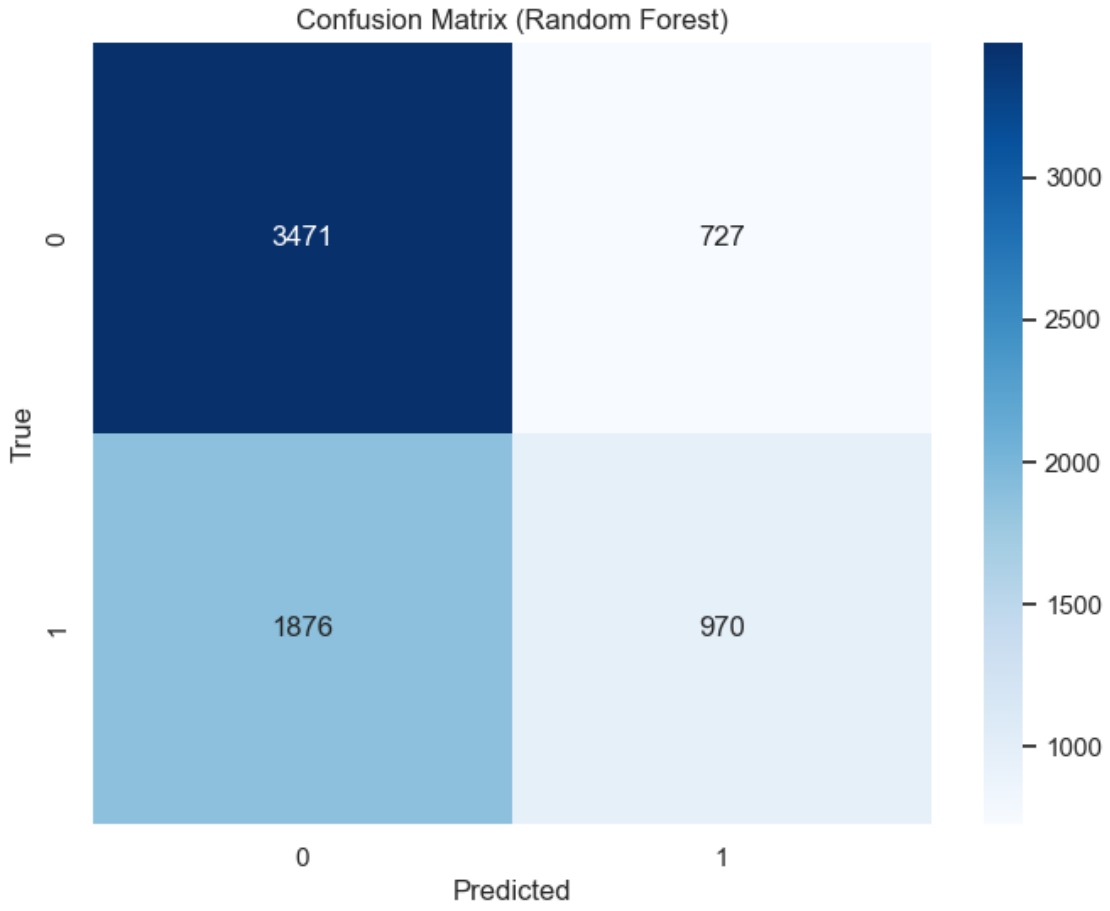
[69]:
```
X_train, X_temp, y_train, y_temp = train_test_split(concatenated_data.
 ↪drop('readmitted', axis=1), concatenated_data['readmitted'], test_size=0.2,␣
 ↪random_state=42)

X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5,␣
 ↪random_state=42)

print("Training set shape:", X_train.shape)
print("Validation set shape:", X_val.shape)
print("Testing set shape:", X_test.shape)
```

```
Training set shape: (56344, 101)
Validation set shape: (7043, 101)
Testing set shape: (7044, 101)
```

[70]:
```
rf_classifier = RandomForestClassifier(random_state=42)

rf_classifier.fit(X_train, y_train)

val_predictions = rf_classifier.predict(X_val)

val_accuracy = accuracy_score(y_val, val_predictions)
print("Validation Accuracy:", val_accuracy)

test_predictions = rf_classifier.predict(X_test)

test_accuracy = accuracy_score(y_test, test_predictions)
print("Test Accuracy:", test_accuracy)


AUC_RF = roc_auc_score(y_test, test_predictions)

F1_POS = f1_score(y_test, test_predictions, pos_label=1)

F1_NEG = f1_score(y_test, test_predictions, pos_label=0)

Harmonic_F1_RF = 2 * (F1_POS * F1_NEG) / (F1_POS + F1_NEG)

print("AUC_RF:", AUC_RF)
print("F1_POS:", F1_POS)
print("F1_NEG:", F1_NEG)
print("Harmonic_F1_RF:", Harmonic_F1_RF)


conf_matrix = confusion_matrix(y_test, test_predictions)
```

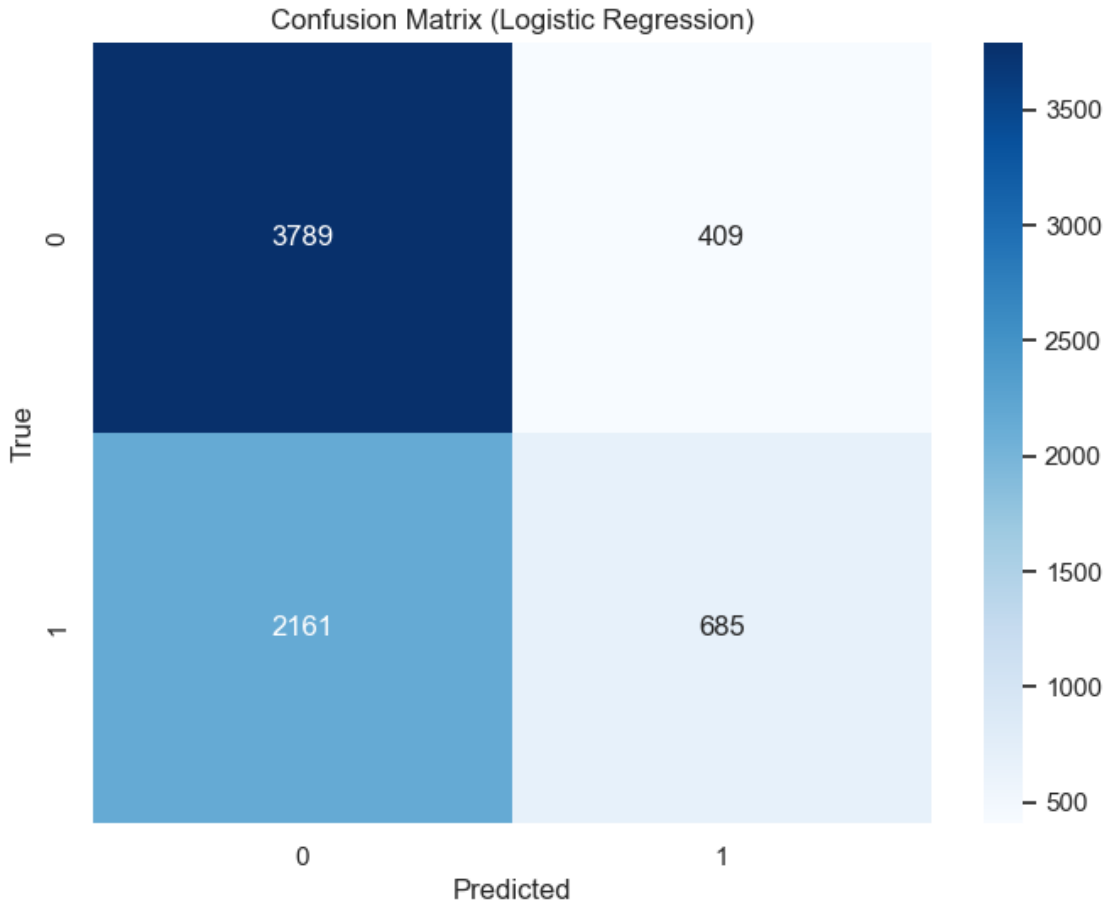
```
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues",
            xticklabels=np.unique(y_test),
            yticklabels=np.unique(y_test))
plt.title("Confusion Matrix (Random Forest)")
plt.xlabel("Predicted")
plt.ylabel("True")
plt.show()
```

```
Validation Accuracy: 0.6169246059917649
Test Accuracy: 0.630465644520159
AUC_RF: 0.5838257651721178
F1_POS: 0.42703059652212194
F1_NEG: 0.7272917757988475
Harmonic_F1_RF: 0.53810936756003
```



[71]:
```
logreg_classifier = LogisticRegression(random_state=42)

logreg_classifier.fit(X_train, y_train)

val_predictions_logreg = logreg_classifier.predict(X_val)

val_accuracy_logreg = accuracy_score(y_val, val_predictions_logreg)
print("Validation Accuracy (Logistic Regression):", val_accuracy_logreg)

test_predictions_logreg = logreg_classifier.predict(X_test)

test_accuracy_logreg = accuracy_score(y_test, test_predictions_logreg)
print("Test Accuracy (Logistic Regression):", test_accuracy_logreg)

AUC_logreg = roc_auc_score(y_test, test_predictions_logreg)

F1_POS_logreg = f1_score(y_test, test_predictions_logreg, pos_label=1)

F1_NEG_logreg = f1_score(y_test, test_predictions_logreg, pos_label=0)

Harmonic_F1_logreg = 2 * (F1_POS_logreg * F1_NEG_logreg) / (F1_POS_logreg +␣
 ↪F1_NEG_logreg)

print("AUC_logreg:", AUC_logreg)
print("F1_POS_logreg:", F1_POS_logreg)
print("F1_NEG_logreg:", F1_NEG_logreg)
print("Harmonic_F1_logreg:", Harmonic_F1_logreg)

conf_matrix_logreg = confusion_matrix(y_test, test_predictions_logreg)

plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_logreg, annot=True, fmt="d", cmap="Blues",
            xticklabels=np.unique(y_test),
            yticklabels=np.unique(y_test))
plt.title("Confusion Matrix (Logistic Regression)")
plt.xlabel("Predicted")
plt.ylabel("True")
plt.show()
```

```
Validation Accuracy (Logistic Regression): 0.6271475223626296
Test Accuracy (Logistic Regression): 0.6351504826802953
AUC_logreg: 0.5716306697597524
F1_POS_logreg: 0.3477157360406091
F1_NEG_logreg: 0.7467481277098936
Harmonic_F1_logreg: 0.47448999179160545
```

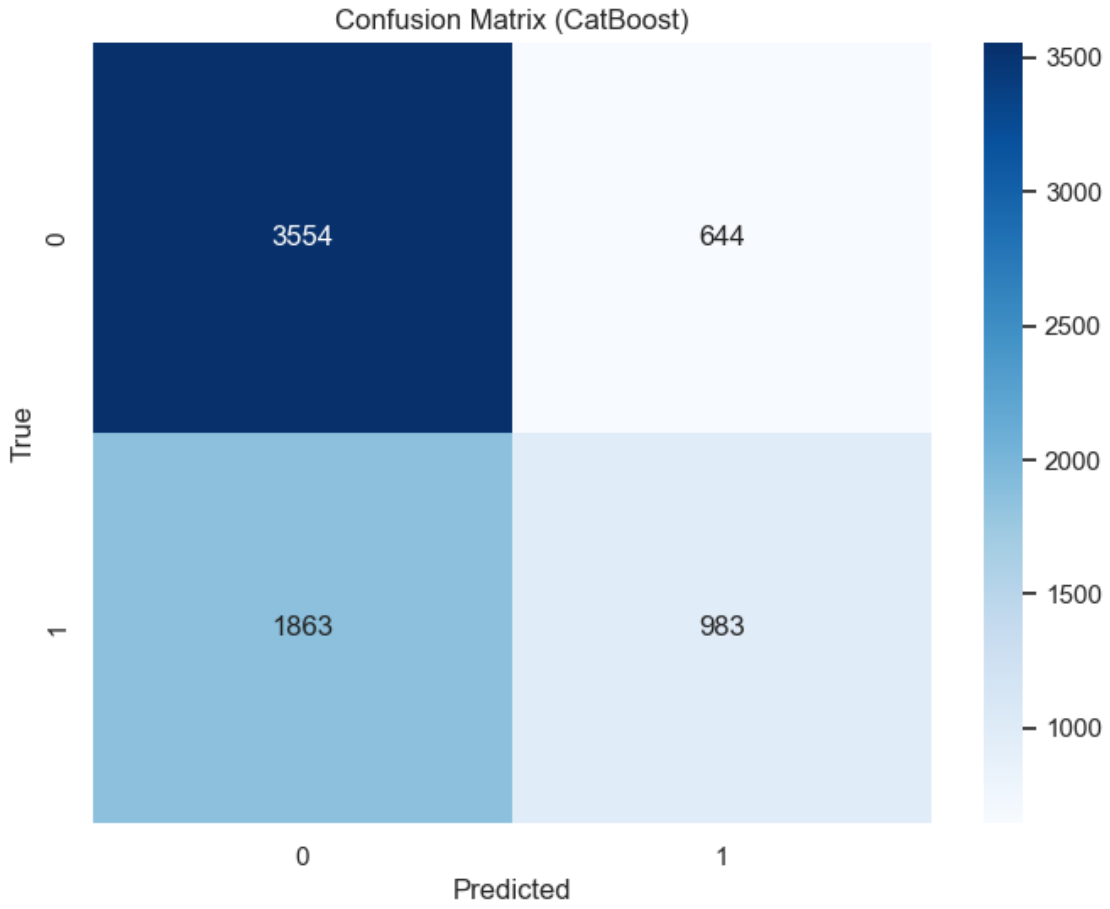


[72]:
```
# !pip install catboost
```

[73]:
```
catboost_classifier = CatBoostClassifier(random_state=42, verbose=0)

catboost_classifier.fit(X_train, y_train)

val_predictions_catboost = catboost_classifier.predict(X_val)

val_accuracy_catboost = accuracy_score(y_val, val_predictions_catboost)
print("Validation Accuracy (CatBoost):", val_accuracy_catboost)

test_predictions_catboost = catboost_classifier.predict(X_test)

test_accuracy_catboost = accuracy_score(y_test, test_predictions_catboost)
print("Test Accuracy (CatBoost):", test_accuracy_catboost)

AUC_catboost = roc_auc_score(y_test, test_predictions_catboost)
```

```
F1_POS_catboost = f1_score(y_test, test_predictions_catboost, pos_label=1)

F1_NEG_catboost = f1_score(y_test, test_predictions_catboost, pos_label=0)

Harmonic_F1_catboost = 2 * (F1_POS_catboost * F1_NEG_catboost) /␣
 ↪(F1_POS_catboost + F1_NEG_catboost)

print("AUC_catboost:", AUC_catboost)
print("F1_POS_catboost:", F1_POS_catboost)
print("F1_NEG_catboost:", F1_NEG_catboost)
print("Harmonic_F1_catboost:", Harmonic_F1_catboost)

conf_matrix_catboost = confusion_matrix(y_test, test_predictions_catboost)

plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_catboost, annot=True, fmt="d", cmap="Blues",
            xticklabels=np.unique(y_test),
            yticklabels=np.unique(y_test))
plt.title("Confusion Matrix (CatBoost)")
plt.xlabel("Predicted")
plt.ylabel("True")
plt.show()
```

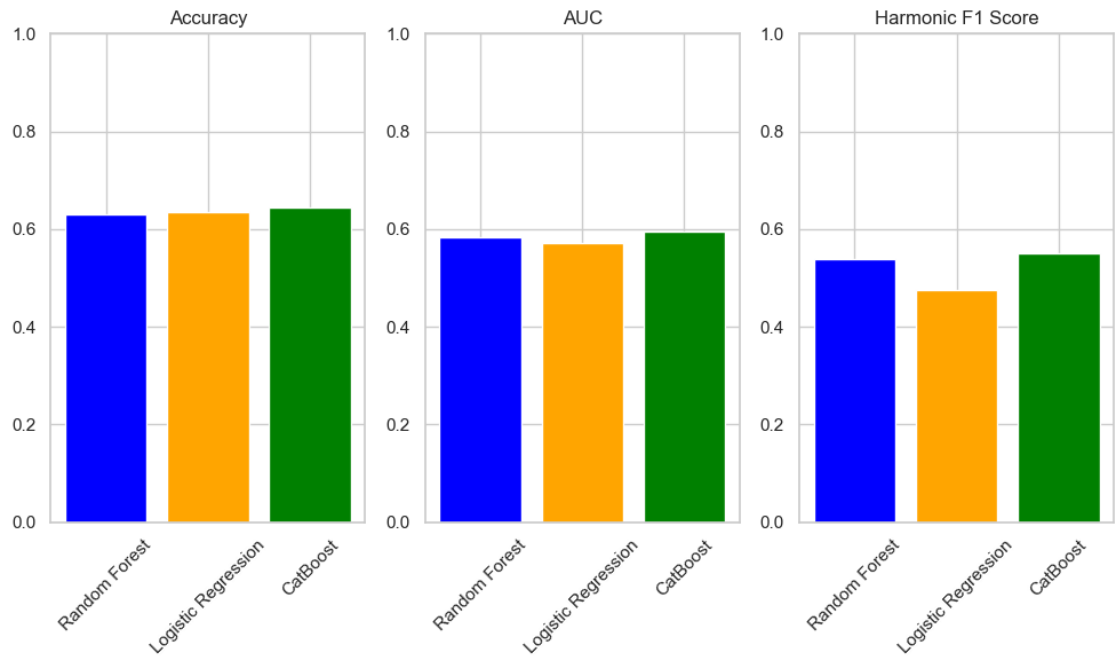
```
Validation Accuracy (CatBoost): 0.6368024989351129
Test Accuracy (CatBoost): 0.6440942646223736
AUC_catboost: 0.5959953322483651
F1_POS_catboost: 0.439526045159848
F1_NEG_catboost: 0.7392615704628185
Harmonic_F1_catboost: 0.5512862709073291
```



[74]:
```
metrics = {
    "Random Forest": {"Accuracy": test_accuracy, "AUC": AUC_RF, "Harmonic F1":␣
 ↪Harmonic_F1_RF},
    "Logistic Regression": {"Accuracy": test_accuracy_logreg, "AUC":␣
 ↪AUC_logreg, "Harmonic F1": Harmonic_F1_logreg},
    "CatBoost": {"Accuracy": test_accuracy_catboost, "AUC": AUC_catboost,␣
 ↪"Harmonic F1": Harmonic_F1_catboost}
}

print("Comparison of Models:")
print("=====================")
for model, scores in metrics.items():
    print(f"Model: {model}")
    print(f"Accuracy: {scores['Accuracy']:.4f}")
    print(f"AUC: {scores['AUC']:.4f}")
    print(f"Harmonic F1: {scores['Harmonic F1']:.4f}")
    print()
```

```
Comparison of Models:
======================
Model: Random Forest
Accuracy: 0.6305
AUC: 0.5838
Harmonic F1: 0.5381

Model: Logistic Regression
Accuracy: 0.6352
AUC: 0.5716
Harmonic F1: 0.4745

Model: CatBoost
Accuracy: 0.6441
AUC: 0.5960
Harmonic F1: 0.5513
```

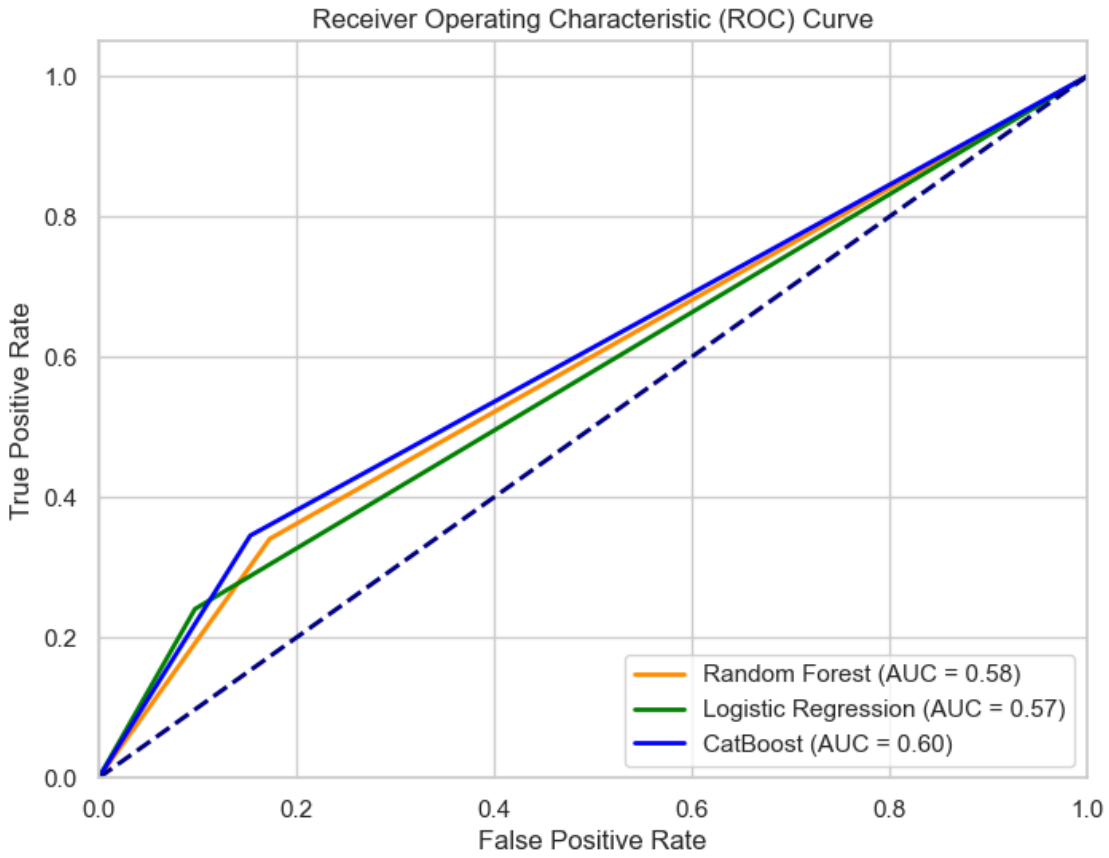
[75]:
```python
models = ["Random Forest", "Logistic Regression", "CatBoost"]
accuracies = [test_accuracy, test_accuracy_logreg, test_accuracy_catboost]
aucs = [AUC_RF, AUC_logreg, AUC_catboost]
harmonic_f1_scores = [Harmonic_F1_RF, Harmonic_F1_logreg, Harmonic_F1_catboost]

plt.figure(figsize=(10, 6))

plt.subplot(1, 3, 1)
plt.bar(models, accuracies, color=['blue', 'orange', 'green'])
plt.title('Accuracy')
plt.ylim(0, 1)
plt.xticks(rotation=45)

plt.subplot(1, 3, 2)
plt.bar(models, aucs, color=['blue', 'orange', 'green'])
plt.title('AUC')
plt.ylim(0, 1)
plt.xticks(rotation=45)

plt.subplot(1, 3, 3)
plt.bar(models, harmonic_f1_scores, color=['blue', 'orange', 'green'])
plt.title('Harmonic F1 Score')
plt.ylim(0, 1)
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```



[ ]:

[76]:
```
fpr_rf, tpr_rf, _ = roc_curve(y_test, test_predictions)
roc_auc_rf = auc(fpr_rf, tpr_rf)

fpr_logreg, tpr_logreg, _ = roc_curve(y_test, test_predictions_logreg)
roc_auc_logreg = auc(fpr_logreg, tpr_logreg)

fpr_catboost, tpr_catboost, _ = roc_curve(y_test, test_predictions_catboost)
roc_auc_catboost = auc(fpr_catboost, tpr_catboost)

plt.figure(figsize=(8, 6))
plt.plot(fpr_rf, tpr_rf, color='darkorange', lw=2, label=f'Random Forest (AUC =␣
 ↪{roc_auc_rf:.2f})')
plt.plot(fpr_logreg, tpr_logreg, color='green', lw=2, label=f'Logistic␣
 ↪Regression (AUC = {roc_auc_logreg:.2f})')
plt.plot(fpr_catboost, tpr_catboost, color='blue', lw=2, label=f'CatBoost (AUC␣
 ↪= {roc_auc_catboost:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
```

```
plt.legend(loc="lower right")
plt.show()
```



[ ]:

[ ]: