Medicare Fraud Detection

November 27, 2021

[1]:
```
import numpy as np
import pandas as pd
import scipy as sc
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
from sklearn.preprocessing import StandardScaler,MinMaxScaler
from imblearn.combine import SMOTETomek
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from collections import Counter
from sklearn.metrics import␣
↪confusion_matrix,roc_curve,accuracy_score,roc_auc_score,classification_report
import pickle
from scipy import stats
import time
from sklearn.model_selection import GridSearchCV,KFold

from pylab import rcParams

%matplotlib inline
sns.set(style='whitegrid', palette='muted', font_scale=1.5)
rcParams['figure.figsize'] = 14, 8
RANDOM_SEED = 42

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

[2]:
```
# Load Train Dataset

Train=pd.read_csv("Train-1542865627584.csv")
Train_Beneficiarydata=pd.read_csv("Train_Beneficiarydata-1542865627584.csv")
Train_Inpatientdata=pd.read_csv("Train_Inpatientdata-1542865627584.csv")
Train_Outpatientdata=pd.read_csv("Train_Outpatientdata-1542865627584.csv")

# Load Test Dataset

Test=pd.read_csv("Test-1542969243754.csv")
Test_Beneficiarydata=pd.read_csv("Test_Beneficiarydata-1542969243754.csv")
Test_Inpatientdata=pd.read_csv("Test_Inpatientdata-1542969243754.csv")
Test_Outpatientdata=pd.read_csv("Test_Outpatientdata-1542969243754.csv")
```

[3]:
```
## Lets Check Shape of datasets

print('Shape of Train data :',Train.shape)
print('Shape of Train_Beneficiarydata data :',Train_Beneficiarydata.shape)
print('Shape of Train_Inpatientdata data :',Train_Inpatientdata.shape)
print('Shape of Train_Outpatientdata data :',Train_Outpatientdata.shape)

print('Shape of Test data :',Test.shape)
print('Shape of Test_Beneficiarydata data :',Test_Beneficiarydata.shape)
print('Shape of Test_Inpatientdata data :',Test_Inpatientdata.shape)
print('Shape of Test_Outpatientdata data :',Test_Outpatientdata.shape)
```

```
Shape of Train data : (5410, 2)
Shape of Train_Beneficiarydata data : (138556, 25)
Shape of Train_Inpatientdata data : (40474, 30)
Shape of Train_Outpatientdata data : (517737, 27)
Shape of Test data : (1353, 1)
Shape of Test_Beneficiarydata data : (63968, 25)
Shape of Test_Inpatientdata data : (9551, 30)
Shape of Test_Outpatientdata data : (125841, 27)
```

Train and Test Dataset understanding

[4]:
```
print('\033[1m'"Train Dataset"+ "\033[0m","\n",Train.head(4),'\n')

print('\033[1m'+"Test Dataset"+ "\033[0m")

print(Test.head(4)) # We don't have Target Variable Fraud in the test dataset
 ↪and this target variable we need to predict
```

```
Train Dataset
    Provider PotentialFraud
0  PRV51001             No
1  PRV51003            Yes
2  PRV51004             No
```

```
3  PRV51005            Yes

Test Dataset
   Provider
0  PRV51002
1  PRV51006
2  PRV51009
3  PRV51010
```

[5]:
```
#To Check the summary of the train dataset

Train.describe()
```

[5]:
```
         Provider PotentialFraud
count        5410           5410
unique       5410              2
top     PRV51001             No
freq            1           4904
```

[6]:
```
## Lets check whether  providers details are unique or not in train data
print(Train.Provider.value_counts(sort=True,ascending=False).head(5))  # number␣
 ↪of unique providers in train data.Check for duplicates

print('\n Total missing values in Train :',Train.isna().sum().sum())

print('\n Total missing values in Train :',Test.isna().sum().sum())
```

```
PRV51001    1
PRV55516    1
PRV55527    1
PRV55525    1
PRV55523    1
Name: Provider, dtype: int64

 Total missing values in Train : 0

 Total missing values in Train : 0
```

0.0.1 Data Preprocessing on Beneficiary Dataset

[7]:
```
print('\033[1m'+"Train Dataset"+ "\033[0m")

display(Train_Beneficiarydata.head(5))

print('\033[1m'+"Test Dataset"+ "\033[0m")

display(Test_Beneficiarydata.head(5))
```

```
Train Dataset
```

```
      BeneID         DOB  DOD  Gender  Race RenalDiseaseIndicator  State  \
0  BENE11001  1943-01-01  NaN       1     1                     0     39
1  BENE11002  1936-09-01  NaN       2     1                     0     39
2  BENE11003  1936-08-01  NaN       1     1                     0     52
3  BENE11004  1922-07-01  NaN       1     1                     0     39
4  BENE11005  1935-09-01  NaN       1     1                     0     24

   County  NoOfMonths_PartACov  NoOfMonths_PartBCov  …  \
0     230                   12                   12  …
1     280                   12                   12  …
2     590                   12                   12  …
3     270                   12                   12  …
4     680                   12                   12  …

   ChronicCond_Depression  ChronicCond_Diabetes  ChronicCond_IschemicHeart  \
0                       1                     1                          1
1                       2                     2                          2
2                       2                     2                          1
3                       2                     1                          1
4                       2                     1                          2

   ChronicCond_Osteoporasis  ChronicCond_rheumatoidarthritis  \
0                         2                                1
1                         2                                2
2                         2                                2
3                         1                                1
4                         2                                2

   ChronicCond_stroke  IPAnnualReimbursementAmt  IPAnnualDeductibleAmt  \
0                   1                     36000                   3204
1                   2                         0                      0
2                   2                         0                      0
3                   2                         0                      0
4                   2                         0                      0

   OPAnnualReimbursementAmt  OPAnnualDeductibleAmt
0                        60                     70
1                        30                     50
2                        90                     40
3                      1810                    760
4                      1790                   1200

[5 rows x 25 columns]
```

Test Dataset

```
      BeneID         DOB         DOD  Gender  Race RenalDiseaseIndicator  \
0  BENE11001  1943-01-01         NaN       1     1                     0
1  BENE11007  1940-09-01  2009-12-01       1     2                     0
```

```
2  BENE11010  1936-07-01         NaN       2     1                    0
3  BENE11011  1914-03-01         NaN       2     2                    0
4  BENE11014  1938-04-01         NaN       2     1                    Y

   State  County  NoOfMonths_PartACov  NoOfMonths_PartBCov  …  \
0     39     230                   12                   12  …
1     45     610                   12                   12  …
2     41      30                   12                   12  …
3      1     360                   12                   12  …
4     45     780                   12                   12  …

   ChronicCond_Depression  ChronicCond_Diabetes  ChronicCond_IschemicHeart  \
0                       1                     1                          1
1                       2                     1                          2
2                       2                     1                          1
3                       1                     1                          2
4                       1                     2                          1

   ChronicCond_Osteoporasis  ChronicCond_rheumatoidarthritis  \
0                         2                                1
1                         1                                1
2                         1                                2
3                         2                                1
4                         2                                2

   ChronicCond_stroke  IPAnnualReimbursementAmt  IPAnnualDeductibleAmt  \
0                   1                     36000                   3204
1                   2                         0                      0
2                   2                         0                      0
3                   1                      5000                   1068
4                   2                     21260                   2136

   OPAnnualReimbursementAmt  OPAnnualDeductibleAmt
0                        60                     70
1                      1490                    160
2                      1170                    660
3                       250                    320
4                       120                    100

[5 rows x 25 columns]
```

[8]:
```
#Lets Check missing values in each column in beneficiary data :


print('\033[1m'+"Train Beneficiary Dataset"+ "\033[0m")

print(Train_Beneficiarydata.isna().sum())
```

```
print('\033[1m'+"Test Beneficiary Dataset"+ "\033[0m")

print(Train_Beneficiarydata.isna().sum())
```

```
Train Beneficiary Dataset
BeneID                                0
DOB                                   0
DOD                              137135
Gender                                0
Race                                  0
RenalDiseaseIndicator                 0
State                                 0
County                                0
NoOfMonths_PartACov                   0
NoOfMonths_PartBCov                   0
ChronicCond_Alzheimer                 0
ChronicCond_Heartfailure              0
ChronicCond_KidneyDisease             0
ChronicCond_Cancer                    0
ChronicCond_ObstrPulmonary            0
ChronicCond_Depression                0
ChronicCond_Diabetes                  0
ChronicCond_IschemicHeart             0
ChronicCond_Osteoporasis              0
ChronicCond_rheumatoidarthritis       0
ChronicCond_stroke                    0
IPAnnualReimbursementAmt              0
IPAnnualDeductibleAmt                 0
OPAnnualReimbursementAmt              0
OPAnnualDeductibleAmt                 0
dtype: int64
Test Beneficiary Dataset
BeneID                                0
DOB                                   0
DOD                              137135
Gender                                0
Race                                  0
RenalDiseaseIndicator                 0
State                                 0
County                                0
NoOfMonths_PartACov                   0
NoOfMonths_PartBCov                   0
ChronicCond_Alzheimer                 0
ChronicCond_Heartfailure              0
ChronicCond_KidneyDisease             0
ChronicCond_Cancer                    0
ChronicCond_ObstrPulmonary            0
```

```
ChronicCond_Depression                0
ChronicCond_Diabetes                  0
ChronicCond_IschemicHeart             0
ChronicCond_Osteoporasis              0
ChronicCond_rheumatoidarthritis       0
ChronicCond_stroke                    0
IPAnnualReimbursementAmt              0
IPAnnualDeductibleAmt                 0
OPAnnualReimbursementAmt              0
OPAnnualDeductibleAmt                 0
dtype: int64
```

[9]:
```
# Lets check data types of each column in beneficiary data

Train_Beneficiarydata.dtypes
```

[9]:
```
BeneID                               object
DOB                                  object
DOD                                  object
Gender                                int64
Race                                  int64
RenalDiseaseIndicator                object
State                                 int64
County                                int64
NoOfMonths_PartACov                   int64
NoOfMonths_PartBCov                   int64
ChronicCond_Alzheimer                 int64
ChronicCond_Heartfailure              int64
ChronicCond_KidneyDisease             int64
ChronicCond_Cancer                    int64
ChronicCond_ObstrPulmonary            int64
ChronicCond_Depression                int64
ChronicCond_Diabetes                  int64
ChronicCond_IschemicHeart             int64
ChronicCond_Osteoporasis              int64
ChronicCond_rheumatoidarthritis       int64
ChronicCond_stroke                    int64
IPAnnualReimbursementAmt              int64
IPAnnualDeductibleAmt                 int64
OPAnnualReimbursementAmt              int64
OPAnnualDeductibleAmt                 int64
dtype: object
```

[10]:
```
Train_Beneficiarydata.describe(include='all')
```

[10]:

	BeneID	DOB	DOD	Gender	Race	\
count	138556	138556	1421	138556.000000	138556.000000	
unique	138556	900	11	NaN	NaN	

top	BENE11001	1939-10-01	2009-12-01	NaN	NaN
freq	1	540	182	NaN	NaN
mean	NaN	NaN	NaN	1.570932	1.254511
std	NaN	NaN	NaN	0.494945	0.717007
min	NaN	NaN	NaN	1.000000	1.000000
25%	NaN	NaN	NaN	1.000000	1.000000
50%	NaN	NaN	NaN	2.000000	1.000000
75%	NaN	NaN	NaN	2.000000	1.000000
max	NaN	NaN	NaN	2.000000	5.000000

	RenalDiseaseIndicator	State	County	\
count	138556	138556.000000	138556.000000	
unique	2	NaN	NaN	
top	0	NaN	NaN	
freq	118978	NaN	NaN	
mean	NaN	25.666734	374.424745	
std	NaN	15.223443	266.277581	
min	NaN	1.000000	0.000000	
25%	NaN	11.000000	141.000000	
50%	NaN	25.000000	340.000000	
75%	NaN	39.000000	570.000000	
max	NaN	54.000000	999.000000	

	NoOfMonths_PartACov	NoOfMonths_PartBCov	…	ChronicCond_Depression	\
count	138556.000000	138556.000000	…	138556.000000	
unique	NaN	NaN	…	NaN	
top	NaN	NaN	…	NaN	
freq	NaN	NaN	…	NaN	
mean	11.907727	11.910145	…	1.644476	
std	1.032332	0.936893	…	0.478674	
min	0.000000	0.000000	…	1.000000	
25%	12.000000	12.000000	…	1.000000	
50%	12.000000	12.000000	…	2.000000	
75%	12.000000	12.000000	…	2.000000	
max	12.000000	12.000000	…	2.000000	

	ChronicCond_Diabetes	ChronicCond_IschemicHeart	\
count	138556.000000	138556.000000	
unique	NaN	NaN	
top	NaN	NaN	
freq	NaN	NaN	
mean	1.398142	1.324143	
std	0.489517	0.468056	
min	1.000000	1.000000	
25%	1.000000	1.000000	
50%	1.000000	1.000000	
75%	2.000000	2.000000	

```
max                 2.000000                    2.000000

       ChronicCond_Osteoporasis  ChronicCond_rheumatoidarthritis  \
count                138556.000000                    138556.000000
unique                         NaN                              NaN
top                            NaN                              NaN
freq                           NaN                              NaN
mean                      1.725317                         1.743180
std                       0.446356                         0.436881
min                       1.000000                         1.000000
25%                       1.000000                         1.000000
50%                       2.000000                         2.000000
75%                       2.000000                         2.000000
max                       2.000000                         2.000000

       ChronicCond_stroke  IPAnnualReimbursementAmt  IPAnnualDeductibleAmt  \
count         138556.000000             138556.000000          138556.000000
unique                  NaN                       NaN                    NaN
top                     NaN                       NaN                    NaN
freq                    NaN                       NaN                    NaN
mean               1.920942               3660.346502             399.847296
std                0.269831               9568.621827             956.175202
min                1.000000              -8000.000000               0.000000
25%                2.000000                  0.000000               0.000000
50%                2.000000                  0.000000               0.000000
75%                2.000000               2280.000000            1068.000000
max                2.000000             161470.000000           38272.000000

       OPAnnualReimbursementAmt  OPAnnualDeductibleAmt
count                138556.000000          138556.000000
unique                         NaN                    NaN
top                            NaN                    NaN
freq                           NaN                    NaN
mean                   1298.219348             377.718258
std                    2493.901134             645.530187
min                     -70.000000               0.000000
25%                     170.000000              40.000000
50%                     570.000000             170.000000
75%                    1500.000000             460.000000
max                  102960.000000           13840.000000

[11 rows x 25 columns]
```

[11]:
```
##Replacing 2 with 0 for chronic conditions ,that means chronic condition No is␣
↪0 and yes is 1
```

```
Train_Beneficiarydata = Train_Beneficiarydata.replace({'ChronicCond_Alzheimer':␣
↪2, 'ChronicCond_Heartfailure': 2, 'ChronicCond_KidneyDisease': 2,
                          'ChronicCond_Cancer': 2,␣
↪'ChronicCond_ObstrPulmonary': 2, 'ChronicCond_Depression': 2,
                          'ChronicCond_Diabetes': 2,␣
↪'ChronicCond_IschemicHeart': 2, 'ChronicCond_Osteoporasis': 2,
                          'ChronicCond_rheumatoidarthritis': 2,␣
↪'ChronicCond_stroke': 2 }, 0)

Train_Beneficiarydata = Train_Beneficiarydata.replace({'RenalDiseaseIndicator':␣
↪'Y'}, 1)


## Same thing do in the Test Dataset also
Test_Beneficiarydata = Test_Beneficiarydata.replace({'ChronicCond_Alzheimer':␣
↪2, 'ChronicCond_Heartfailure': 2, 'ChronicCond_KidneyDisease': 2,
                          'ChronicCond_Cancer': 2,␣
↪'ChronicCond_ObstrPulmonary': 2, 'ChronicCond_Depression': 2,
                          'ChronicCond_Diabetes': 2,␣
↪'ChronicCond_IschemicHeart': 2, 'ChronicCond_Osteoporasis': 2,
                          'ChronicCond_rheumatoidarthritis': 2,␣
↪'ChronicCond_stroke': 2 }, 0)

Test_Beneficiarydata = Test_Beneficiarydata.replace({'RenalDiseaseIndicator':␣
↪'Y'}, 1)
```

Feature Engineering on Beneficiary Dataset

[12]:
```
## Lets Create Age column to the Train and Test dataset

Train_Beneficiarydata['DOB'] = pd.to_datetime(Train_Beneficiarydata['DOB'] )
Train_Beneficiarydata['DOD'] = pd.
↪to_datetime(Train_Beneficiarydata['DOD'],errors='ignore')
Train_Beneficiarydata['Age'] = round(((Train_Beneficiarydata['DOD'] -␣
↪Train_Beneficiarydata['DOB']).dt.days)/365)


Test_Beneficiarydata['DOB'] = pd.to_datetime(Test_Beneficiarydata['DOB'])
Test_Beneficiarydata['DOD'] = pd.
↪to_datetime(Test_Beneficiarydata['DOD'],errors='ignore')
Test_Beneficiarydata['Age'] = round(((Test_Beneficiarydata['DOD'] -␣
↪Test_Beneficiarydata['DOB']).dt.days)/365)
```

[13]:
```
Train_Beneficiarydata.head(10)
```

[13]:

	BeneID	DOB	DOD	Gender	Race	RenalDiseaseIndicator	State	\
0	BENE11001	1943-01-01	NaT	1	1	0	39	

```
1  BENE11002 1936-09-01        NaT       2     1                    0     39
2  BENE11003 1936-08-01        NaT       1     1                    0     52
3  BENE11004 1922-07-01        NaT       1     1                    0     39
4  BENE11005 1935-09-01        NaT       1     1                    0     24
5  BENE11006 1976-09-01        NaT       2     1                    0     23
6  BENE11007 1940-09-01 2009-12-01       1     2                    0     45
7  BENE11008 1934-02-01        NaT       2     1                    0     15
8  BENE11009 1929-06-01        NaT       1     1                    1     44
9  BENE11010 1936-07-01        NaT       2     1                    0     41

   County  NoOfMonths_PartACov  NoOfMonths_PartBCov  …  \
0     230                   12                   12  …
1     280                   12                   12  …
2     590                   12                   12  …
3     270                   12                   12  …
4     680                   12                   12  …
5     810                   12                   12  …
6     610                   12                   12  …
7     140                   12                   12  …
8     230                   12                   12  …
9      30                   12                   12  …

   ChronicCond_Diabetes  ChronicCond_IschemicHeart  ChronicCond_Osteoporasis  \
0                     1                          1                         0
1                     0                          0                         0
2                     0                          1                         0
3                     1                          1                         1
4                     1                          0                         0
5                     0                          0                         0
6                     1                          0                         1
7                     1                          0                         0
8                     1                          0                         0
9                     1                          1                         1

   ChronicCond_rheumatoidarthritis  ChronicCond_stroke  \
0                                1                   1
1                                0                   0
2                                0                   0
3                                1                   0
4                                0                   0
5                                0                   0
6                                1                   0
7                                0                   0
8                                0                   0
9                                0                   0

   IPAnnualReimbursementAmt  IPAnnualDeductibleAmt  OPAnnualReimbursementAmt  \
```

```
0                      36000                    3204                      60
1                          0                       0                      30
2                          0                       0                      90
3                          0                       0                    1810
4                          0                       0                    1790
5                          0                       0                     500
6                          0                       0                    1490
7                          0                       0                      30
8                          0                       0                     100
9                          0                       0                    1170

   OPAnnualDeductibleAmt   Age
0                     70   NaN
1                     50   NaN
2                     40   NaN
3                    760   NaN
4                   1200   NaN
5                      0   NaN
6                    160  69.0
7                      0   NaN
8                      0   NaN
9                    660   NaN

[10 rows x 26 columns]
```

[14]:
```
## As we can see above Age column have some Nan values, This is due to DOD is␣
 ↪Nan for that record.
## As we see that last DOD value is 2017-12-01 ,which means Beneficiary Details␣
 ↪data is of year 2017.
## so we will calculate age of other benficiaries for year 2017.

Train_Beneficiarydata.Age.fillna(round(((pd.to_datetime('2017-12-01' ) -␣
 ↪Train_Beneficiarydata['DOB']).dt.days)/365),
                                 inplace=True)


Test_Beneficiarydata.Age.fillna(round(((pd.to_datetime('2017-12-01') -␣
 ↪Test_Beneficiarydata['DOB']).dt.days)/365),
                                 inplace=True)
```

[15]:
```
Train_Beneficiarydata.head(5)
```

[15]:
```
      BeneID        DOB DOD  Gender  Race RenalDiseaseIndicator  State  \
0  BENE11001 1943-01-01 NaT       1     1                      0     39
1  BENE11002 1936-09-01 NaT       2     1                      0     39
2  BENE11003 1936-08-01 NaT       1     1                      0     52
3  BENE11004 1922-07-01 NaT       1     1                      0     39
```

```
4  BENE11005 1935-09-01 NaT       1     1                    0     24

   County  NoOfMonths_PartACov  NoOfMonths_PartBCov  …  \
0     230                   12                   12  …
1     280                   12                   12  …
2     590                   12                   12  …
3     270                   12                   12  …
4     680                   12                   12  …

   ChronicCond_Diabetes  ChronicCond_IschemicHeart  ChronicCond_Osteoporasis  \
0                     1                          1                         0
1                     0                          0                         0
2                     0                          1                         0
3                     1                          1                         1
4                     1                          0                         0

   ChronicCond_rheumatoidarthritis  ChronicCond_stroke  \
0                                1                   1
1                                0                   0
2                                0                   0
3                                1                   0
4                                0                   0

   IPAnnualReimbursementAmt  IPAnnualDeductibleAmt  OPAnnualReimbursementAmt  \
0                     36000                   3204                        60
1                         0                      0                        30
2                         0                      0                        90
3                         0                      0                      1810
4                         0                      0                      1790

   OPAnnualDeductibleAmt   Age
0                     70  75.0
1                     50  81.0
2                     40  81.0
3                    760  95.0
4                   1200  82.0

[5 rows x 26 columns]
```

Add Flag column 'WhetherDead' using DOD values to tell whether beneficiary is dead on not

[16]:
```
#Lets create a new variable 'WhetherDead' with flag 1 means Dead and 0 means␣
↪not Dead

Train_Beneficiarydata.loc[Train_Beneficiarydata.DOD.isna(),'WhetherDead']=0
Train_Beneficiarydata.loc[Train_Beneficiarydata.DOD.notna(),'WhetherDead']=1
```

```
Test_Beneficiarydata.loc[Test_Beneficiarydata.DOD.isna(),'WhetherDead']=0
Test_Beneficiarydata.loc[Test_Beneficiarydata.DOD.notna(),'WhetherDead']=1
```

[17]:
```
print('\033[1m'+"Train Dataset"+ "\033[0m")

print(Train_Beneficiarydata.loc[:,'WhetherDead'].head(7))

print('\033[1m'+"Test Dataset"+ "\033[0m")

print(Train_Beneficiarydata.loc[:,'WhetherDead'].head(7))
```

```
Train Dataset
0    0.0
1    0.0
2    0.0
3    0.0
4    0.0
5    0.0
6    1.0
Name: WhetherDead, dtype: float64
Test Dataset
0    0.0
1    0.0
2    0.0
3    0.0
4    0.0
5    0.0
6    1.0
Name: WhetherDead, dtype: float64
```

0.0.2 Data Preprocessing on Inpatient Dataset

[18]:
```
# Summary of Inpatient Dataset

print('\033[1m'+"Train Inpatient Dataset"+ "\033[0m")

display(Train_Inpatientdata.head(5))

print('\033[1m'+"Test Inpatient Dataset"+ "\033[0m")

display(Train_Inpatientdata.head(5))
```

```
Train Inpatient Dataset
```

	BeneID	ClaimID	ClaimStartDt	ClaimEndDt	Provider	\
0	BENE11001	CLM46614	2009-04-12	2009-04-18	PRV55912	
1	BENE11001	CLM66048	2009-08-31	2009-09-02	PRV55907	

```
2  BENE11001  CLM68358   2009-09-17  2009-09-20  PRV56046
3  BENE11011  CLM38412   2009-02-14  2009-02-22  PRV52405
4  BENE11014  CLM63689   2009-08-13  2009-08-30  PRV56614

   InscClaimAmtReimbursed AttendingPhysician OperatingPhysician  \
0                   26000          PHY390922                NaN
1                    5000          PHY318495          PHY318495
2                    5000          PHY372395                NaN
3                    5000          PHY369659          PHY392961
4                   10000          PHY379376          PHY398258

  OtherPhysician AdmissionDt  … ClmDiagnosisCode_7  ClmDiagnosisCode_8  \
0            NaN  2009-04-12  …                2724               19889
1            NaN  2009-08-31  …                 NaN                 NaN
2     PHY324689  2009-09-17  …                 NaN                 NaN
3     PHY349768  2009-02-14  …               25062               40390
4            NaN  2009-08-13  …                5119               29620

  ClmDiagnosisCode_9 ClmDiagnosisCode_10 ClmProcedureCode_1  \
0               5849                 NaN                NaN
1                NaN                 NaN             7092.0
2                NaN                 NaN                NaN
3               4019                 NaN              331.0
4              20300                 NaN             3893.0

  ClmProcedureCode_2 ClmProcedureCode_3 ClmProcedureCode_4 ClmProcedureCode_5  \
0                NaN                NaN                NaN                NaN
1                NaN                NaN                NaN                NaN
2                NaN                NaN                NaN                NaN
3                NaN                NaN                NaN                NaN
4                NaN                NaN                NaN                NaN

  ClmProcedureCode_6
0                NaN
1                NaN
2                NaN
3                NaN
4                NaN

[5 rows x 30 columns]
```

Test Inpatient Dataset

```
      BeneID   ClaimID ClaimStartDt  ClaimEndDt  Provider  \
0  BENE11001  CLM46614   2009-04-12  2009-04-18  PRV55912
1  BENE11001  CLM66048   2009-08-31  2009-09-02  PRV55907
2  BENE11001  CLM68358   2009-09-17  2009-09-20  PRV56046
3  BENE11011  CLM38412   2009-02-14  2009-02-22  PRV52405
4  BENE11014  CLM63689   2009-08-13  2009-08-30  PRV56614
```

```
   InscClaimAmtReimbursed AttendingPhysician OperatingPhysician  \
0                   26000          PHY390922                NaN
1                    5000          PHY318495          PHY318495
2                    5000          PHY372395                NaN
3                    5000          PHY369659          PHY392961
4                   10000          PHY379376          PHY398258

  OtherPhysician AdmissionDt  … ClmDiagnosisCode_7  ClmDiagnosisCode_8  \
0            NaN  2009-04-12  …               2724               19889
1            NaN  2009-08-31  …                NaN                 NaN
2      PHY324689  2009-09-17  …                NaN                 NaN
3      PHY349768  2009-02-14  …              25062               40390
4            NaN  2009-08-13  …               5119               29620

  ClmDiagnosisCode_9 ClmDiagnosisCode_10 ClmProcedureCode_1  \
0               5849                 NaN                NaN
1                NaN                 NaN             7092.0
2                NaN                 NaN                NaN
3               4019                 NaN              331.0
4              20300                 NaN             3893.0

  ClmProcedureCode_2 ClmProcedureCode_3 ClmProcedureCode_4 ClmProcedureCode_5  \
0                NaN                NaN                NaN                NaN
1                NaN                NaN                NaN                NaN
2                NaN                NaN                NaN                NaN
3                NaN                NaN                NaN                NaN
4                NaN                NaN                NaN                NaN

  ClmProcedureCode_6
0                NaN
1                NaN
2                NaN
3                NaN
4                NaN

[5 rows x 30 columns]
```

[19]:
```
#Lets check missing values in each column in inpatient data

print('\033[1m'+"Train Inpatient Dataset"+ "\033[0m")

print(Train_Inpatientdata.isna().sum())

print('\033[1m'+"Test Inpatient Dataset"+ "\033[0m")

print(Test_Inpatientdata.isna().sum())
```

Train Inpatient Dataset

BeneID	0
ClaimID	0
ClaimStartDt	0
ClaimEndDt	0
Provider	0
InscClaimAmtReimbursed	0
AttendingPhysician	112
OperatingPhysician	16644
OtherPhysician	35784
AdmissionDt	0
ClmAdmitDiagnosisCode	0
DeductibleAmtPaid	899
DischargeDt	0
DiagnosisGroupCode	0
ClmDiagnosisCode_1	0
ClmDiagnosisCode_2	226
ClmDiagnosisCode_3	676
ClmDiagnosisCode_4	1534
ClmDiagnosisCode_5	2894
ClmDiagnosisCode_6	4838
ClmDiagnosisCode_7	7258
ClmDiagnosisCode_8	9942
ClmDiagnosisCode_9	13497
ClmDiagnosisCode_10	36547
ClmProcedureCode_1	17326
ClmProcedureCode_2	35020
ClmProcedureCode_3	39509
ClmProcedureCode_4	40358
ClmProcedureCode_5	40465
ClmProcedureCode_6	40474

dtype: int64

Test Inpatient Dataset

BeneID	0
ClaimID	0
ClaimStartDt	0
ClaimEndDt	0
Provider	0
InscClaimAmtReimbursed	0
AttendingPhysician	31
OperatingPhysician	3962
OtherPhysician	8538
AdmissionDt	0
ClmAdmitDiagnosisCode	0
DeductibleAmtPaid	196
DischargeDt	0
DiagnosisGroupCode	0
ClmDiagnosisCode_1	0

```
ClmDiagnosisCode_2          54
ClmDiagnosisCode_3         169
ClmDiagnosisCode_4         404
ClmDiagnosisCode_5         719
ClmDiagnosisCode_6        1197
ClmDiagnosisCode_7        1736
ClmDiagnosisCode_8        2360
ClmDiagnosisCode_9        3238
ClmDiagnosisCode_10       8664
ClmProcedureCode_1        4118
ClmProcedureCode_2        8297
ClmProcedureCode_3        9328
ClmProcedureCode_4        9522
ClmProcedureCode_5        9549
ClmProcedureCode_6        9551
dtype: int64
```

Feature Engineering on Inpatient Dataset Create new column 'AdmitForDays' indicating number of days patient was admitted in hospital

[20]:
```
## As patient can be admitted for only for 1 day,we will add 1 to the␣
 ↪difference of Discharge Date and Admission Date

Train_Inpatientdata['AdmissionDt'] = pd.
 ↪to_datetime(Train_Inpatientdata['AdmissionDt'])
Train_Inpatientdata['DischargeDt'] = pd.
 ↪to_datetime(Train_Inpatientdata['DischargeDt'])
Train_Inpatientdata['AdmitForDays'] = ((Train_Inpatientdata['DischargeDt'] -␣
 ↪Train_Inpatientdata['AdmissionDt']).dt.days.abs())+1


Test_Inpatientdata['AdmissionDt'] = pd.
 ↪to_datetime(Test_Inpatientdata['AdmissionDt'])
Test_Inpatientdata['DischargeDt'] = pd.
 ↪to_datetime(Test_Inpatientdata['DischargeDt'])
Test_Inpatientdata['AdmitForDays'] = ((Test_Inpatientdata['DischargeDt'] -␣
 ↪Test_Inpatientdata['AdmissionDt']).dt.days.abs())+1
```

[21]:
```
Train_Inpatientdata.loc[:,['AdmissionDt','DischargeDt','AdmitForDays']]
```

[21]:

	AdmissionDt	DischargeDt	AdmitForDays
0	2009-04-12	2009-04-18	7
1	2009-08-31	2009-09-02	3
2	2009-09-17	2009-09-20	4
3	2009-02-14	2009-02-22	9
4	2009-08-13	2009-08-30	18
…	…	…	…

```
40469  2009-09-28  2009-10-02             5
40470  2009-11-03  2009-11-06             4
40471  2009-11-18  2009-11-22             5
40472  2009-12-17  2009-12-18             2
40473  2009-09-28  2009-10-06             9

[40474 rows x 3 columns]
```

[22]:
```
## Lets check Min and Max values of AdmitforDays column in Train and Test.
print('Min AdmitForDays Train:- ',Train_Inpatientdata.AdmitForDays.min())
print('Max AdmitForDays Train:- ',Train_Inpatientdata.AdmitForDays.max())
print(Train_Inpatientdata.AdmitForDays.isnull().sum() )  #Check Null values.

print('Min AdmitForDays Test:- ',Test_Inpatientdata.AdmitForDays.min())
print('Max AdmitForDays Test:- ',Test_Inpatientdata.AdmitForDays.max())
print(Test_Inpatientdata.AdmitForDays.isnull().sum())   #Check Null values.
```

```
Min AdmitForDays Train:-  1
Max AdmitForDays Train:-  36
0
Min AdmitForDays Test:-  1
Max AdmitForDays Test:-  36
0
```

0.0.3 Data Preprocessing on Outpatient Dataset

[23]:
```
# Summary of Outpatient Dataset

print('\033[1m'+"Train Outpatient Dataset"+ "\033[0m")

display(Train_Outpatientdata.head(5))

print('\033[1m'+"Test Outpatient Dataset"+ "\033[0m")

display(Train_Outpatientdata.head(5))
```

Train Outpatient Dataset

```
      BeneID    ClaimID ClaimStartDt  ClaimEndDt  Provider  \
0  BENE11002  CLM624349   2009-10-11  2009-10-11  PRV56011
1  BENE11003  CLM189947   2009-02-12  2009-02-12  PRV57610
2  BENE11003  CLM438021   2009-06-27  2009-06-27  PRV57595
3  BENE11004  CLM121801   2009-01-06  2009-01-06  PRV56011
4  BENE11004  CLM150998   2009-01-22  2009-01-22  PRV56011

   InscClaimAmtReimbursed AttendingPhysician OperatingPhysician  \
0                      30          PHY326117                NaN
1                      80          PHY362868                NaN
2                      10          PHY328821                NaN
```

```
3                     40          PHY334319                NaN
4                    200          PHY403831                NaN

  OtherPhysician ClmDiagnosisCode_1  … ClmDiagnosisCode_9  \
0            NaN              78943  …                NaN
1            NaN               6115  …                NaN
2            NaN               2723  …                NaN
3            NaN              71988  …                NaN
4            NaN              82382  …                NaN

  ClmDiagnosisCode_10 ClmProcedureCode_1 ClmProcedureCode_2  \
0                 NaN                NaN                NaN
1                 NaN                NaN                NaN
2                 NaN                NaN                NaN
3                 NaN                NaN                NaN
4                 NaN                NaN                NaN

  ClmProcedureCode_3 ClmProcedureCode_4 ClmProcedureCode_5 ClmProcedureCode_6  \
0                NaN                NaN                NaN                NaN
1                NaN                NaN                NaN                NaN
2                NaN                NaN                NaN                NaN
3                NaN                NaN                NaN                NaN
4                NaN                NaN                NaN                NaN

  DeductibleAmtPaid  ClmAdmitDiagnosisCode
0                 0                 56409
1                 0                 79380
2                 0                   NaN
3                 0                   NaN
4                 0                 71947

[5 rows x 27 columns]
```

Test Outpatient Dataset

```
      BeneID    ClaimID ClaimStartDt  ClaimEndDt  Provider  \
0  BENE11002  CLM624349   2009-10-11  2009-10-11  PRV56011
1  BENE11003  CLM189947   2009-02-12  2009-02-12  PRV57610
2  BENE11003  CLM438021   2009-06-27  2009-06-27  PRV57595
3  BENE11004  CLM121801   2009-01-06  2009-01-06  PRV56011
4  BENE11004  CLM150998   2009-01-22  2009-01-22  PRV56011

   InscClaimAmtReimbursed AttendingPhysician OperatingPhysician  \
0                      30          PHY326117                NaN
1                      80          PHY362868                NaN
2                      10          PHY328821                NaN
3                      40          PHY334319                NaN
4                     200          PHY403831                NaN
```

```
  OtherPhysician ClmDiagnosisCode_1  … ClmDiagnosisCode_9  \
0            NaN              78943  …                NaN
1            NaN               6115  …                NaN
2            NaN               2723  …                NaN
3            NaN              71988  …                NaN
4            NaN              82382  …                NaN

  ClmDiagnosisCode_10 ClmProcedureCode_1 ClmProcedureCode_2  \
0                 NaN                NaN                NaN
1                 NaN                NaN                NaN
2                 NaN                NaN                NaN
3                 NaN                NaN                NaN
4                 NaN                NaN                NaN

  ClmProcedureCode_3 ClmProcedureCode_4 ClmProcedureCode_5 ClmProcedureCode_6  \
0                NaN                NaN                NaN                NaN
1                NaN                NaN                NaN                NaN
2                NaN                NaN                NaN                NaN
3                NaN                NaN                NaN                NaN
4                NaN                NaN                NaN                NaN

  DeductibleAmtPaid  ClmAdmitDiagnosisCode
0                 0                 56409
1                 0                 79380
2                 0                   NaN
3                 0                   NaN
4                 0                 71947

[5 rows x 27 columns]
```

[24]:
```
# Lets check the null values in each column of Outpatient Dataset

print('\033[1m'+"Train Outpatient Dataset"+ "\033[0m")

print(Train_Outpatientdata.isna().sum())

print('\033[1m'+"Test Outpatient Dataset"+ "\033[0m")

print(Test_Outpatientdata.isna().sum())
```

```
Train Outpatient Dataset
BeneID                         0
ClaimID                        0
ClaimStartDt                   0
ClaimEndDt                     0
Provider                       0
InscClaimAmtReimbursed         0
AttendingPhysician          1396
```

```
OperatingPhysician        427120
OtherPhysician            322691
ClmDiagnosisCode_1         10453
ClmDiagnosisCode_2        195380
ClmDiagnosisCode_3        314480
ClmDiagnosisCode_4        392141
ClmDiagnosisCode_5        443393
ClmDiagnosisCode_6        468981
ClmDiagnosisCode_7        484776
ClmDiagnosisCode_8        494825
ClmDiagnosisCode_9        502899
ClmDiagnosisCode_10       516654
ClmProcedureCode_1        517575
ClmProcedureCode_2        517701
ClmProcedureCode_3        517733
ClmProcedureCode_4        517735
ClmProcedureCode_5        517737
ClmProcedureCode_6        517737
DeductibleAmtPaid              0
ClmAdmitDiagnosisCode     412312
dtype: int64
```

Test Outpatient Dataset

```
BeneID                         0
ClaimID                        0
ClaimStartDt                   0
ClaimEndDt                     0
Provider                       0
InscClaimAmtReimbursed         0
AttendingPhysician           316
OperatingPhysician        104237
OtherPhysician             78222
ClmDiagnosisCode_1          2578
ClmDiagnosisCode_2         47731
ClmDiagnosisCode_3         76575
ClmDiagnosisCode_4         95371
ClmDiagnosisCode_5        107875
ClmDiagnosisCode_6        114035
ClmDiagnosisCode_7        117871
ClmDiagnosisCode_8        120310
ClmDiagnosisCode_9        122278
ClmDiagnosisCode_10       125578
ClmProcedureCode_1        125807
ClmProcedureCode_2        125832
ClmProcedureCode_3        125839
ClmProcedureCode_4        125841
ClmProcedureCode_5        125841
ClmProcedureCode_6        125841
DeductibleAmtPaid              0
```

```
ClmAdmitDiagnosisCode      100036
dtype: int64
```

[25]:
```
## Lets Check Shape of datasets after adding new variables

print('Shape of Train data :',Train.shape)
print('Shape of Train_Beneficiarydata data :',Train_Beneficiarydata.shape)
print('Shape of Train_Inpatientdata data :',Train_Inpatientdata.shape)
print('Shape of Train_Outpatientdata data :',Train_Outpatientdata.shape)

print('Shape of Test data :',Test.shape)
print('Shape of Test_Beneficiarydata data :',Test_Beneficiarydata.shape)
print('Shape of Test_Inpatientdata data :',Test_Inpatientdata.shape)
```

```
Shape of Train data : (5410, 2)
Shape of Train_Beneficiarydata data : (138556, 27)
Shape of Train_Inpatientdata data : (40474, 31)
Shape of Train_Outpatientdata data : (517737, 27)
Shape of Test data : (1353, 1)
Shape of Test_Beneficiarydata data : (63968, 27)
Shape of Test_Inpatientdata data : (9551, 31)
```

0.0.4 Merge Beneficiary, Inpatient and Outpatient Dataset into a single dataset

Merging of Train Datasets

[26]:
```
Train_patient_merge_id = [i for i in Train_Outpatientdata.columns if i in␣
↪Train_Inpatientdata.columns]

# Merge Inpatient, Outpatient and beneficiary dataframe into a single patient␣
↪dataset
Train_Patient_data = pd.merge(Train_Inpatientdata, Train_Outpatientdata,
                    left_on = Train_patient_merge_id,
                    right_on = Train_patient_merge_id,
                    how = 'outer').\
        ␣
↪merge(Train_Beneficiarydata,left_on='BeneID',right_on='BeneID',how='inner')
```

Merging of Test Dataset

[27]:
```
Test_patient_merge_id = [i for i in Test_Outpatientdata.columns if i in␣
↪Test_Inpatientdata.columns]

# Merge Inpatient, Outpatient and beneficiary dataframe into a single patient␣
↪dataset
Test_Patient_data = pd.merge(Test_Inpatientdata, Test_Outpatientdata,
                    left_on = Test_patient_merge_id,
                    right_on = Test_patient_merge_id,
                    how = 'outer').\
```

```
␣
↪merge(Test_Beneficiarydata,left_on='BeneID',right_on='BeneID',how='inner')
```

[28]:
```
# Shape of Merging Dataset

print("Train Dataset Shape after merge:",Train_Patient_data.shape)

print("Test Dataset Shape after merge:",Test_Patient_data.shape)
```

```
Train Dataset Shape after merge: (558211, 57)
Test Dataset Shape after merge: (135392, 57)
```

0.0.5 Exploratory Data Analysis on Train_Patient_data dataset

[29]:
```
Train_Patient_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 558211 entries, 0 to 558210
Data columns (total 57 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   BeneID                          558211 non-null  object
 1   ClaimID                         558211 non-null  object
 2   ClaimStartDt                    558211 non-null  object
 3   ClaimEndDt                      558211 non-null  object
 4   Provider                        558211 non-null  object
 5   InscClaimAmtReimbursed          558211 non-null  int64
 6   AttendingPhysician              556703 non-null  object
 7   OperatingPhysician              114447 non-null  object
 8   OtherPhysician                  199736 non-null  object
 9   AdmissionDt                     40474 non-null   datetime64[ns]
 10  ClmAdmitDiagnosisCode           145899 non-null  object
 11  DeductibleAmtPaid               557312 non-null  float64
 12  DischargeDt                     40474 non-null   datetime64[ns]
 13  DiagnosisGroupCode              40474 non-null   object
 14  ClmDiagnosisCode_1              547758 non-null  object
 15  ClmDiagnosisCode_2              362605 non-null  object
 16  ClmDiagnosisCode_3              243055 non-null  object
 17  ClmDiagnosisCode_4              164536 non-null  object
 18  ClmDiagnosisCode_5              111924 non-null  object
 19  ClmDiagnosisCode_6              84392 non-null   object
 20  ClmDiagnosisCode_7              66177 non-null   object
 21  ClmDiagnosisCode_8              53444 non-null   object
 22  ClmDiagnosisCode_9              41815 non-null   object
 23  ClmDiagnosisCode_10             5010 non-null    object
 24  ClmProcedureCode_1              23310 non-null   float64
 25  ClmProcedureCode_2              5490 non-null    float64
 26  ClmProcedureCode_3              969 non-null     float64
```

```
 27  ClmProcedureCode_4                 118 non-null     float64
 28  ClmProcedureCode_5                 9 non-null       float64
 29  ClmProcedureCode_6                 0 non-null       float64
 30  AdmitForDays                       40474 non-null   float64
 31  DOB                                558211 non-null  datetime64[ns]
 32  DOD                                4131 non-null    datetime64[ns]
 33  Gender                             558211 non-null  int64
 34  Race                               558211 non-null  int64
 35  RenalDiseaseIndicator              558211 non-null  object
 36  State                              558211 non-null  int64
 37  County                             558211 non-null  int64
 38  NoOfMonths_PartACov                558211 non-null  int64
 39  NoOfMonths_PartBCov                558211 non-null  int64
 40  ChronicCond_Alzheimer              558211 non-null  int64
 41  ChronicCond_Heartfailure           558211 non-null  int64
 42  ChronicCond_KidneyDisease          558211 non-null  int64
 43  ChronicCond_Cancer                 558211 non-null  int64
 44  ChronicCond_ObstrPulmonary         558211 non-null  int64
 45  ChronicCond_Depression             558211 non-null  int64
 46  ChronicCond_Diabetes               558211 non-null  int64
 47  ChronicCond_IschemicHeart          558211 non-null  int64
 48  ChronicCond_Osteoporasis           558211 non-null  int64
 49  ChronicCond_rheumatoidarthritis    558211 non-null  int64
 50  ChronicCond_stroke                 558211 non-null  int64
 51  IPAnnualReimbursementAmt           558211 non-null  int64
 52  IPAnnualDeductibleAmt              558211 non-null  int64
 53  OPAnnualReimbursementAmt           558211 non-null  int64
 54  OPAnnualDeductibleAmt              558211 non-null  int64
 55  Age                                558211 non-null  float64
 56  WhetherDead                        558211 non-null  float64
dtypes: datetime64[ns](4), float64(10), int64(22), object(21)
memory usage: 247.0+ MB
```

Handling Missing values

[30]:
```
# To check the number of missing values in the Train_Pateint_data

Train_Patient_data.isnull().sum()
```

[30]:
```
BeneID                         0
ClaimID                        0
ClaimStartDt                   0
ClaimEndDt                     0
Provider                       0
InscClaimAmtReimbursed         0
AttendingPhysician          1508
OperatingPhysician        443764
OtherPhysician            358475
```

AdmissionDt	517737
ClmAdmitDiagnosisCode	412312
DeductibleAmtPaid	899
DischargeDt	517737
DiagnosisGroupCode	517737
ClmDiagnosisCode_1	10453
ClmDiagnosisCode_2	195606
ClmDiagnosisCode_3	315156
ClmDiagnosisCode_4	393675
ClmDiagnosisCode_5	446287
ClmDiagnosisCode_6	473819
ClmDiagnosisCode_7	492034
ClmDiagnosisCode_8	504767
ClmDiagnosisCode_9	516396
ClmDiagnosisCode_10	553201
ClmProcedureCode_1	534901
ClmProcedureCode_2	552721
ClmProcedureCode_3	557242
ClmProcedureCode_4	558093
ClmProcedureCode_5	558202
ClmProcedureCode_6	558211
AdmitForDays	517737
DOB	0
DOD	554080
Gender	0
Race	0
RenalDiseaseIndicator	0
State	0
County	0
NoOfMonths_PartACov	0
NoOfMonths_PartBCov	0
ChronicCond_Alzheimer	0
ChronicCond_Heartfailure	0
ChronicCond_KidneyDisease	0
ChronicCond_Cancer	0
ChronicCond_ObstrPulmonary	0
ChronicCond_Depression	0
ChronicCond_Diabetes	0
ChronicCond_IschemicHeart	0
ChronicCond_Osteoporasis	0
ChronicCond_rheumatoidarthritis	0
ChronicCond_stroke	0
IPAnnualReimbursementAmt	0
IPAnnualDeductibleAmt	0
OPAnnualReimbursementAmt	0
OPAnnualDeductibleAmt	0
Age	0

```
WhetherDead                              0
dtype: int64
```

[31]:
```
### There are missing values in AttendingPhysician, OperatingPhysician and␣
↪OtherPhysician columns, so we need to handle these varaibles

Train_Patient_data[['AttendingPhysician', 'OperatingPhysician',␣
↪'OtherPhysician']]
```

[31]:

	AttendingPhysician	OperatingPhysician	OtherPhysician
0	PHY390922	NaN	NaN
1	PHY318495	PHY318495	NaN
2	PHY372395	NaN	PHY324689
3	PHY369659	PHY392961	PHY349768
4	PHY379398	NaN	NaN
…	…	…	…
558206	PHY364188	PHY364188	PHY385752
558207	PHY423019	PHY332284	NaN
558208	PHY361063	NaN	NaN
558209	PHY403198	NaN	PHY419379
558210	PHY419379	NaN	PHY419379

[558211 rows x 3 columns]

[32]:
```
Train_Patient_data[['AttendingPhysician','OperatingPhysician',␣
↪'OtherPhysician']].describe()
```

[32]:

	AttendingPhysician	OperatingPhysician	OtherPhysician
count	556703	114447	199736
unique	82063	35315	46457
top	PHY330576	PHY330576	PHY412132
freq	2534	424	1247

[33]:
```
## We are replacing these columns value with 0 and 1 where we have value we are␣
↪replacing it with 1 and in place of null value we replace it with 0.


Train_Patient_data[['AttendingPhysician', 'OperatingPhysician',␣
↪'OtherPhysician']] = np.where(Train_Patient_data[['AttendingPhysician',␣
↪'OperatingPhysician', 'OtherPhysician']].isnull(), 0, 1)
```

[34]:
```
Train_Patient_data[['AttendingPhysician', 'OperatingPhysician',␣
↪'OtherPhysician']]
```

[34]:

	AttendingPhysician	OperatingPhysician	OtherPhysician
0	1	0	0
1	1	1	0

```
2                       1                   0               1
3                       1                   1               1
4                       1                   0               0
…                       …                   …               …
558206                  1                   1               1
558207                  1                   1               0
558208                  1                   0               0
558209                  1                   0               1
558210                  1                   0               1

[558211 rows x 3 columns]
```

[35]:
```
### Add a new variable in which it tells us how many total types of physicians used for the particular claim or patient.


Train_Patient_data['N_Types_Physicians'] = Train_Patient_data['AttendingPhysician'] + Train_Patient_data['OperatingPhysician'] + Train_Patient_data['OtherPhysician']

Train_Patient_data['N_Types_Physicians']
```

[35]:
```
0         1
1         2
2         2
3         3
4         1
         ..
558206    3
558207    2
558208    1
558209    2
558210    2
Name: N_Types_Physicians, Length: 558211, dtype: int32
```

[36]:
```
Train_Patient_data.isnull().sum() #We can see here new variable "N_Type_Physicians" is added
```

[36]:
```
BeneID                          0
ClaimID                         0
ClaimStartDt                    0
ClaimEndDt                      0
Provider                        0
InscClaimAmtReimbursed          0
AttendingPhysician              0
OperatingPhysician              0
```

OtherPhysician	0
AdmissionDt	517737
ClmAdmitDiagnosisCode	412312
DeductibleAmtPaid	899
DischargeDt	517737
DiagnosisGroupCode	517737
ClmDiagnosisCode_1	10453
ClmDiagnosisCode_2	195606
ClmDiagnosisCode_3	315156
ClmDiagnosisCode_4	393675
ClmDiagnosisCode_5	446287
ClmDiagnosisCode_6	473819
ClmDiagnosisCode_7	492034
ClmDiagnosisCode_8	504767
ClmDiagnosisCode_9	516396
ClmDiagnosisCode_10	553201
ClmProcedureCode_1	534901
ClmProcedureCode_2	552721
ClmProcedureCode_3	557242
ClmProcedureCode_4	558093
ClmProcedureCode_5	558202
ClmProcedureCode_6	558211
AdmitForDays	517737
DOB	0
DOD	554080
Gender	0
Race	0
RenalDiseaseIndicator	0
State	0
County	0
NoOfMonths_PartACov	0
NoOfMonths_PartBCov	0
ChronicCond_Alzheimer	0
ChronicCond_Heartfailure	0
ChronicCond_KidneyDisease	0
ChronicCond_Cancer	0
ChronicCond_ObstrPulmonary	0
ChronicCond_Depression	0
ChronicCond_Diabetes	0
ChronicCond_IschemicHeart	0
ChronicCond_Osteoporasis	0
ChronicCond_rheumatoidarthritis	0
ChronicCond_stroke	0
IPAnnualReimbursementAmt	0
IPAnnualDeductibleAmt	0
OPAnnualReimbursementAmt	0
OPAnnualDeductibleAmt	0

```
Age                                    0
WhetherDead                            0
N_Types_Physicians                     0
dtype: int64
```

[37]:
```
### Handling Missing values on"DiagnosisGroupCode"

Train_Patient_data['DiagnosisGroupCode'].describe()
```

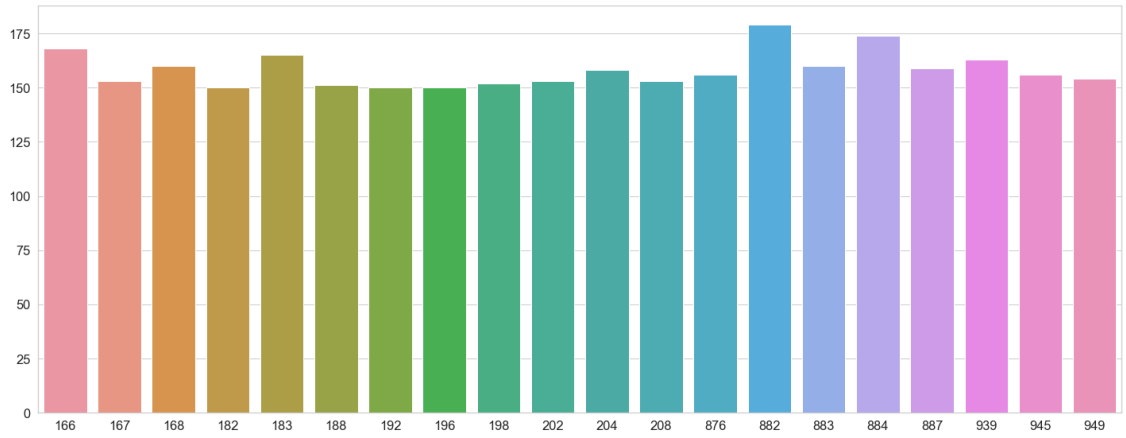
[37]:
```
count     40474
unique      736
top         882
freq        179
Name: DiagnosisGroupCode, dtype: object
```

[38]:
```
# Here we are finding out each DignosisGroupCode Count

Count_DiagnosisGroupCode=Train_Patient_data['DiagnosisGroupCode'].value_counts()
Count_DiagnosisGroupCode=Count_DiagnosisGroupCode[:20] # To show only top 20␣
 ↪codes
Count_DiagnosisGroupCode
```

[38]:
```
882    179
884    174
166    168
183    165
939    163
168    160
883    160
887    159
204    158
945    156
876    156
949    154
167    153
208    153
202    153
198    152
188    151
196    150
192    150
182    150
Name: DiagnosisGroupCode, dtype: int64
```

[39]:
```
### Visualization of top 20 DignosisGroupCode

fig=plt.figure(figsize=(20,8))
```

```
sns.barplot(Count_DiagnosisGroupCode.index,Count_DiagnosisGroupCode.values)
fig.tight_layout()

## From here we can see that DignosisGroupCode 882 has maximum count that is␣
 ↪179
```



[40]:
```
## Since in this columns we have maximum values as null, so we are handling␣
 ↪this by creating a new column
## so we are creating a new variable/column "IsDiagnosisCode" in which value␣
 ↪will either "1" or "0"
## if in a claim there is a groupDiagnosiscode  has null value then in␣
 ↪"IsDiagnosisCode" column value is 0 otherwise 1

Train_Patient_data['IsDiagnosisCode'] = np.where(Train_Patient_data.
 ↪DiagnosisGroupCode.notnull(), 1, 0)
Train_Patient_data = Train_Patient_data.drop(['DiagnosisGroupCode'], axis = 1)␣
 ↪# We are droping the column "DiagnosisGroupCode"
```

[41]:
```
Train_Patient_data['IsDiagnosisCode']
```

[41]:
```
0         1
1         1
2         1
3         1
4         0
         ..
558206    0
558207    0
558208    0
558209    0
558210    0
```

```
Name: IsDiagnosisCode, Length: 558211, dtype: int32
```

[42]:
```
### Handling missing values for "DeductibleamtPaid" column

Train_Patient_data['DeductibleAmtPaid'].isnull().sum()  #Check number of␣
↪missing values in this variable
```

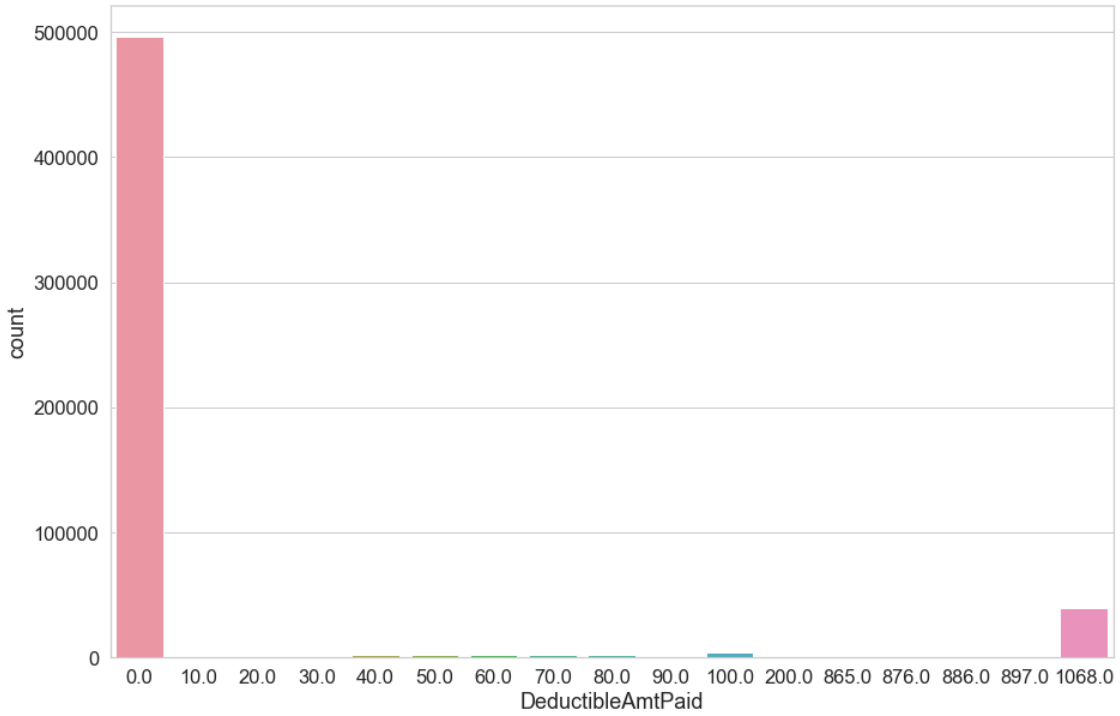
[42]:
```
899
```

[43]:
```
# Describing this column by omiting the Nan, to check mean , variance ,␣
↪skewness etc

sc.stats.describe(Train_Patient_data['DeductibleAmtPaid'],nan_policy='omit')
```

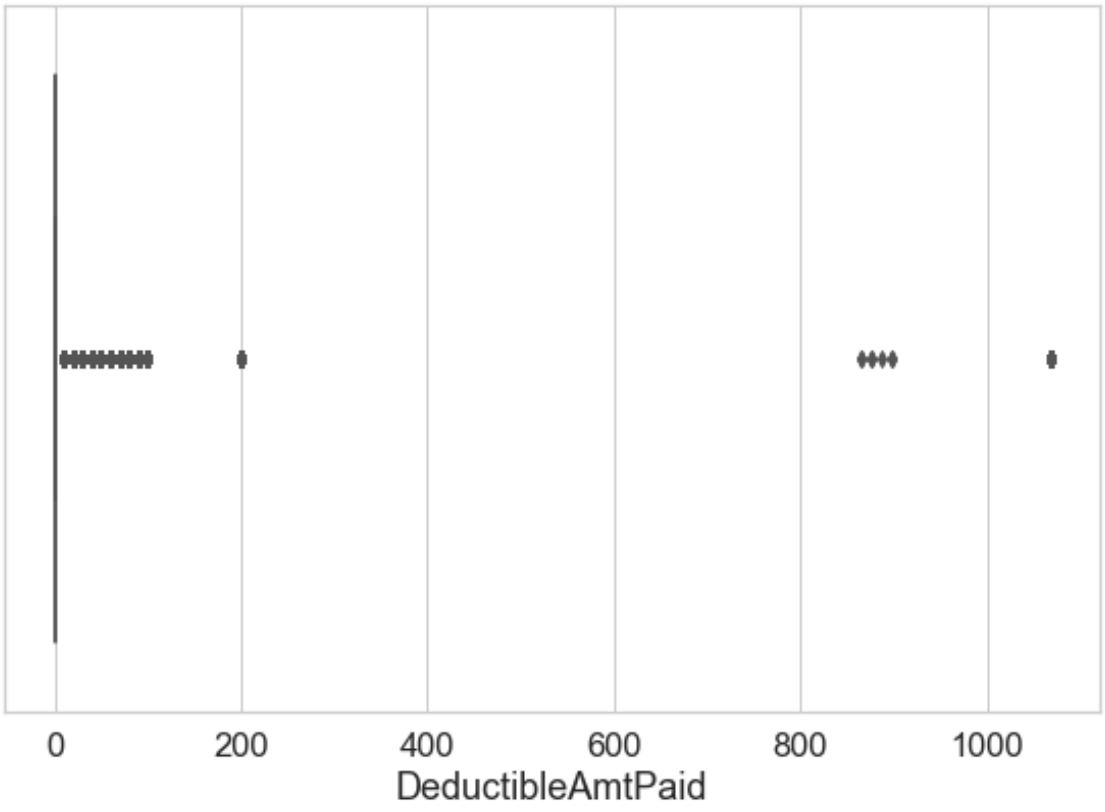
[43]:
```
DescribeResult(nobs=557312, minmax=(masked_array(data=0.,
             mask=False,
       fill_value=1e+20), masked_array(data=1068.,
             mask=False,
       fill_value=1e+20)), mean=78.42108549609554, variance=75085.21352232435,
skewness=masked_array(data=3.32405005,
             mask=False,
       fill_value=1e+20), kurtosis=9.085581103391615)
```

[44]:
```
## Count Plot of "DeductibleAmtPaid" maximum values are 0 in this

fig=plt.figure(figsize=(15,10))
sns.countplot(Train_Patient_data['DeductibleAmtPaid'])
```

[44]:
```
<AxesSubplot:xlabel='DeductibleAmtPaid', ylabel='count'>
```



[45]:
```
## Box plot of this "DeductibleAmtPaid", maximum values are 0 that shows here.

fig=plt.figure(figsize=(8,6))
sns.boxplot(Train_Patient_data['DeductibleAmtPaid'])
fig.tight_layout()
```



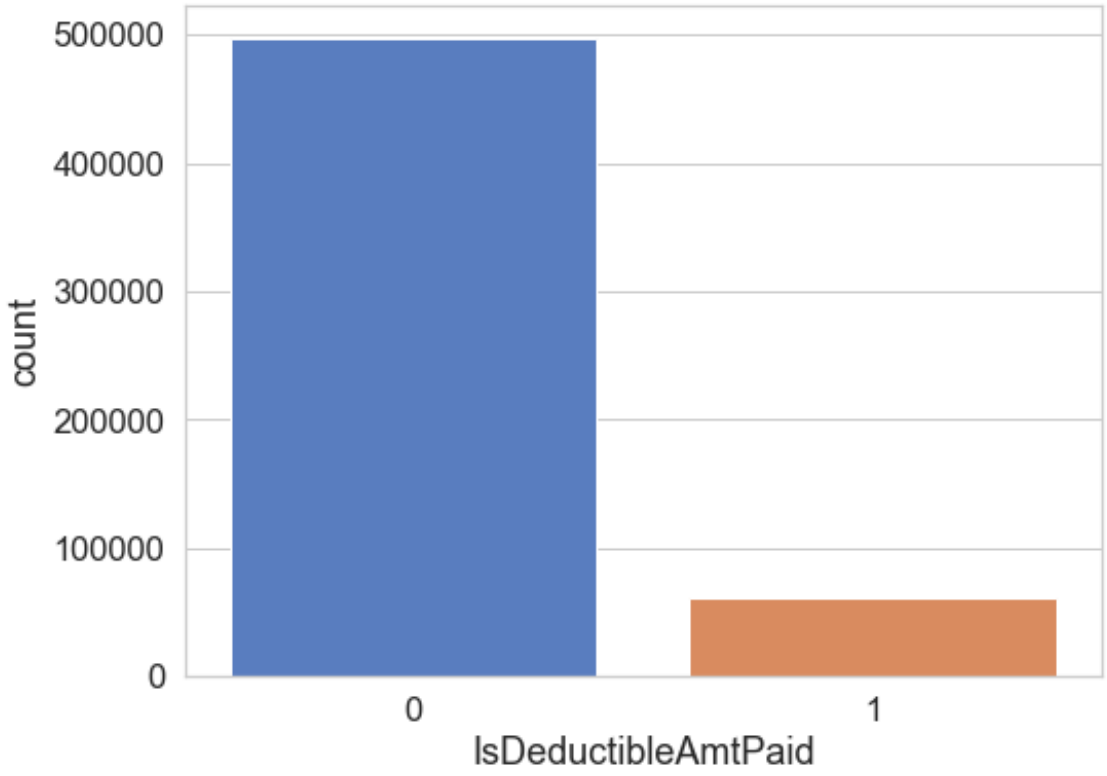
[46]:
```
## So from the above analysis we can reach to the conclusion that we replace␣
 ↪missing values with 0

Train_Patient_data['DeductibleAmtPaid'].fillna(0,inplace=True)
```

[47]:
```
### We are also creating one new variable "IsDeductibleAmtPaid" which tells us␣
 ↪that particular claim has any DeductibleAmtPaid or not

Train_Patient_data['IsDeductibleAmtPaid']=np.
 ↪where(Train_Patient_data['DeductibleAmtPaid']==0,0,1)
```

[48]:
```
# So from this plot we can say that maximum claims doesn't have any␣
 ↪"DeductibleAmtPaid"

fig=plt.figure(figsize=(8,6))
sns.countplot(Train_Patient_data['IsDeductibleAmtPaid'])

print(Train_Patient_data['IsDeductibleAmtPaid'].value_counts())
```

```
0    497600
1     60611
```

```
Name: IsDeductibleAmtPaid, dtype: int64
```



[49]:
```
### Handling missing values for "AdmitForDays" column

Train_Patient_data['AdmitForDays'].isnull().sum() # Count of missing values in␣
↪this column
```

[49]:
```
517737
```

[50]:
```
# Replace all value with 0 as these all are the patients that didn't admit in␣
↪the hospital

Train_Patient_data['AdmitForDays'].fillna(0,inplace=True)
```

[51]:
```
Train_Patient_data['AdmitForDays'].isnull().sum()
```

[51]:
```
0
```

[52]:
```
#In this dataset now we have some Date columns in which missing values are␣
↪there, which we do not need to handle and we can drop those columns also.

Train_Patient_data.isnull().sum()
```

```
# Now we need to handle missing values of ClmDiagnosisCodes and␣
↪ClmProcedureCode columns
```

[52]:
```
BeneID                          0
ClaimID                         0
ClaimStartDt                    0
ClaimEndDt                      0
Provider                        0
InscClaimAmtReimbursed          0
AttendingPhysician              0
OperatingPhysician              0
OtherPhysician                  0
AdmissionDt                517737
ClmAdmitDiagnosisCode      412312
DeductibleAmtPaid               0
DischargeDt                517737
ClmDiagnosisCode_1          10453
ClmDiagnosisCode_2         195606
ClmDiagnosisCode_3         315156
ClmDiagnosisCode_4         393675
ClmDiagnosisCode_5         446287
ClmDiagnosisCode_6         473819
ClmDiagnosisCode_7         492034
ClmDiagnosisCode_8         504767
ClmDiagnosisCode_9         516396
ClmDiagnosisCode_10        553201
ClmProcedureCode_1         534901
ClmProcedureCode_2         552721
ClmProcedureCode_3         557242
ClmProcedureCode_4         558093
ClmProcedureCode_5         558202
ClmProcedureCode_6         558211
AdmitForDays                    0
DOB                             0
DOD                        554080
Gender                          0
Race                            0
RenalDiseaseIndicator           0
State                           0
County                          0
NoOfMonths_PartACov             0
NoOfMonths_PartBCov             0
ChronicCond_Alzheimer           0
ChronicCond_Heartfailure        0
ChronicCond_KidneyDisease       0
```

```
ChronicCond_Cancer                    0
ChronicCond_ObstrPulmonary            0
ChronicCond_Depression                0
ChronicCond_Diabetes                  0
ChronicCond_IschemicHeart             0
ChronicCond_Osteoporasis              0
ChronicCond_rheumatoidarthritis       0
ChronicCond_stroke                    0
IPAnnualReimbursementAmt              0
IPAnnualDeductibleAmt                 0
OPAnnualReimbursementAmt              0
OPAnnualDeductibleAmt                 0
Age                                   0
WhetherDead                           0
N_Types_Physicians                    0
IsDiagnosisCode                       0
IsDeductibleAmtPaid                   0
dtype: int64
```

[53]:
```
## First we handle ClmProcedureCodes variables

ClmProcedure_vars = ['ClmProcedureCode_{}'.format(x) for x in range(1,7)]
ClmProcedure_vars
```

[53]:
```
['ClmProcedureCode_1',
 'ClmProcedureCode_2',
 'ClmProcedureCode_3',
 'ClmProcedureCode_4',
 'ClmProcedureCode_5',
 'ClmProcedureCode_6']
```

[54]:
```
Train_Patient_data[ClmProcedure_vars]
```

[54]:
```
        ClmProcedureCode_1  ClmProcedureCode_2  ClmProcedureCode_3  \
0                      NaN                 NaN                 NaN
1                   7092.0                 NaN                 NaN
2                      NaN                 NaN                 NaN
3                    331.0                 NaN                 NaN
4                      NaN                 NaN                 NaN
…                      …                   …                   …
558206                 NaN                 NaN                 NaN
558207                 NaN                 NaN                 NaN
558208                 NaN                 NaN                 NaN
558209                 NaN                 NaN                 NaN
558210                 NaN                 NaN                 NaN

        ClmProcedureCode_4  ClmProcedureCode_5  ClmProcedureCode_6
```

```
0                    NaN                  NaN                  NaN
1                    NaN                  NaN                  NaN
2                    NaN                  NaN                  NaN
3                    NaN                  NaN                  NaN
4                    NaN                  NaN                  NaN
…                    …                    …                    …
558206               NaN                  NaN                  NaN
558207               NaN                  NaN                  NaN
558208               NaN                  NaN                  NaN
558209               NaN                  NaN                  NaN
558210               NaN                  NaN                  NaN

[558211 rows x 6 columns]
```

[55]:
```
## To Check how many null values are in each Clmprocedurecodes
## By this we find out that in code_6 column all are Nan values

Train_Patient_data[ClmProcedure_vars].isnull().sum()
```

[55]:
```
ClmProcedureCode_1    534901
ClmProcedureCode_2    552721
ClmProcedureCode_3    557242
ClmProcedureCode_4    558093
ClmProcedureCode_5    558202
ClmProcedureCode_6    558211
dtype: int64
```

[56]:
```
Train_Patient_data[ClmProcedure_vars].describe()
```

[56]:
```
       ClmProcedureCode_1  ClmProcedureCode_2  ClmProcedureCode_3  \
count        23310.000000         5490.000000          969.000000
mean          5896.154612         4106.358106         4221.123839
std           3050.489933         2031.640878         2281.849885
min             11.000000           42.000000           42.000000
25%           3848.000000         2724.000000         2724.000000
50%           5363.000000         4019.000000         4019.000000
75%           8669.000000         4439.000000         5185.000000
max           9999.000000         9999.000000         9999.000000

       ClmProcedureCode_4  ClmProcedureCode_5  ClmProcedureCode_6
count          118.000000            9.000000                 0.0
mean          4070.262712         5269.444444                 NaN
std           2037.626990         2780.071632                 NaN
min             42.000000         2724.000000                 NaN
25%           2754.250000         4139.000000                 NaN
50%           4019.000000         4139.000000                 NaN
75%           4439.000000         5185.000000                 NaN
```

```
max           9986.000000          9982.000000                  NaN
```

[57]:
```
# This function helps us find the length of unique values in each row/record
def N_unique_values(df):
    return np.array([len(set([i for i in x[~pd.isnull(x)]])) for x in df.
 ↪values])
```

[58]:
```
# We count the number of procedureCode for each claim and store these value in␣
 ↪a new variable
Train_Patient_data['N_Procedure'] =␣
 ↪N_unique_values(Train_Patient_data[ClmProcedure_vars])

## So from here we get to know that 534901 claims/records has 0 claim procedure␣
 ↪codes, 17820 claims/records has 1 claimprocedurecodes and so on

Train_Patient_data['N_Procedure'].value_counts()
```

[58]:
```
0    534901
1     17820
2      4521
3       851
4       109
5         9
Name: N_Procedure, dtype: int64
```

[59]:
```
### Handling of 'ClmDiagnosisCode'

# We count the number of claims
ClmDiagnosisCode_vars =['ClmAdmitDiagnosisCode'] + ['ClmDiagnosisCode_{}'.
 ↪format(x) for x in range(1, 11)]


ClmDiagnosisCode_vars
```

[59]:
```
['ClmAdmitDiagnosisCode',
 'ClmDiagnosisCode_1',
 'ClmDiagnosisCode_2',
 'ClmDiagnosisCode_3',
 'ClmDiagnosisCode_4',
 'ClmDiagnosisCode_5',
 'ClmDiagnosisCode_6',
 'ClmDiagnosisCode_7',
 'ClmDiagnosisCode_8',
 'ClmDiagnosisCode_9',
 'ClmDiagnosisCode_10']
```

[60]:
```
# We count the number of CLMDiagnosisCode for each claim and store these value␣
↪in a new variable

Train_Patient_data['N_UniqueDiagnosis_Claims'] =␣
↪N_unique_values(Train_Patient_data[ClmDiagnosisCode_vars])


Train_Patient_data['N_UniqueDiagnosis_Claims'].value_counts()
```

[60]:
```
1     152275
2     132264
3      86573
4      57288
5      30338
10     22219
9      20821
6      19652
7      13770
8      11576
0       8319
11      3116
Name: N_UniqueDiagnosis_Claims, dtype: int64
```

EDA on other remaining variables

1.Gender

[61]:
```
Train_Patient_data.Gender.describe()
```

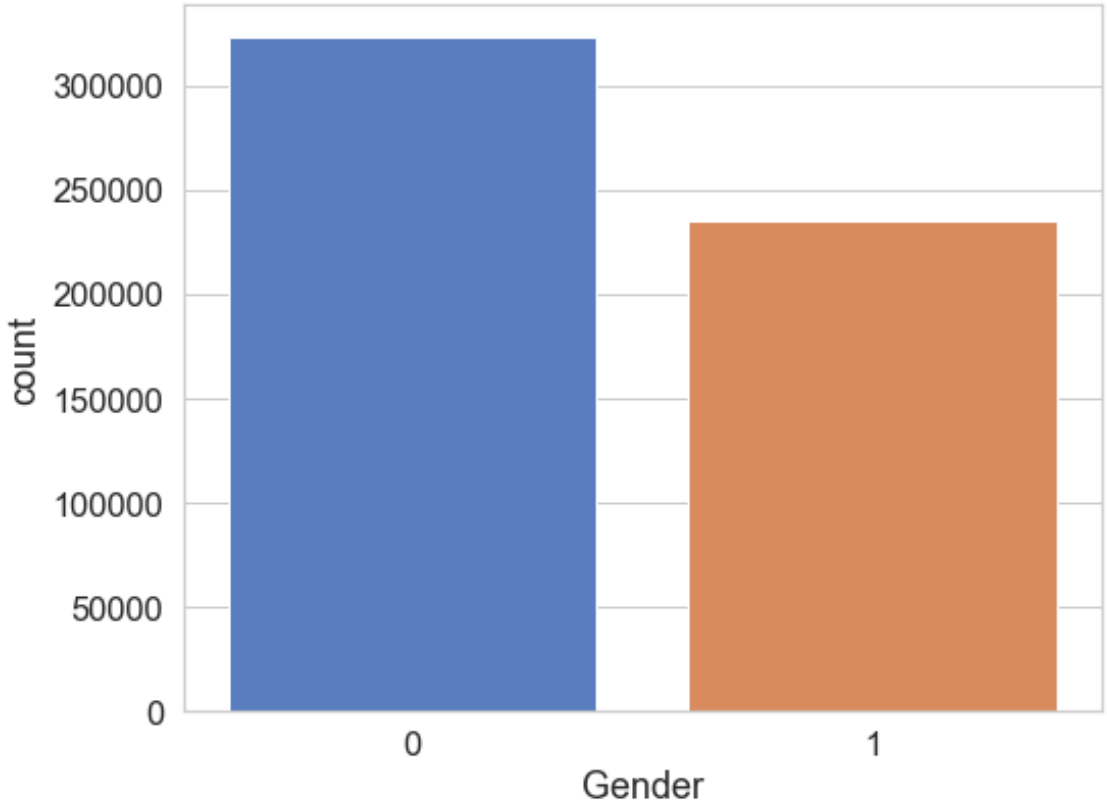
[61]:
```
count    558211.000000
mean          1.578838
std           0.493746
min           1.000000
25%           1.000000
50%           2.000000
75%           2.000000
max           2.000000
Name: Gender, dtype: float64
```

[62]:
```
Train_Patient_data.Gender.value_counts() # here we have only 1 and 2, so we can␣
↪change it to binary as 0 or 1
```

[62]:
```
2    323114
1    235097
Name: Gender, dtype: int64
```

[63]:
```

```

```
Train_Patient_data['Gender']=Train_Patient_data['Gender'].replace(2,0) #␣
 ↪replacing 2 with 0



## Countplot of Gender Column, Here we can consider 0 as Female and 1 as Male

fig=plt.figure(figsize=(8,6))
sns.countplot(Train_Patient_data['Gender'])
fig.tight_layout()
```



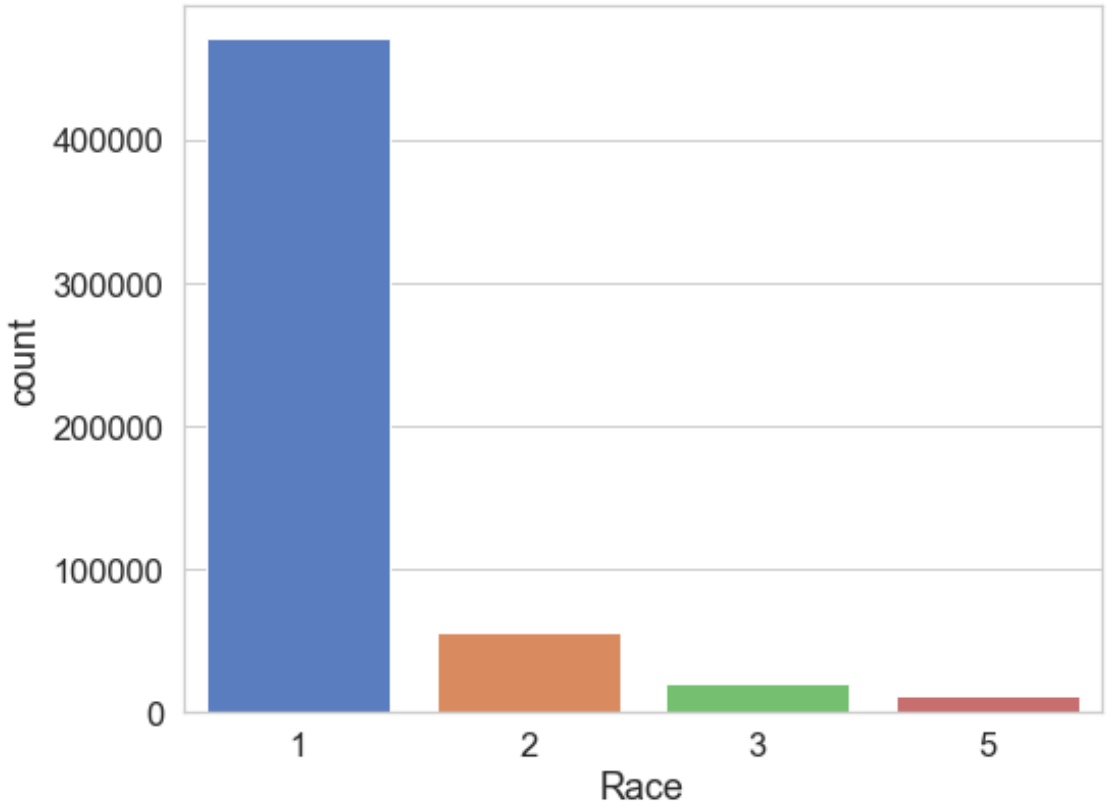
2.Race

[64]:
```
Train_Patient_data['Race'].describe()
```

[64]:
```
count    558211.000000
mean          1.255011
std           0.717437
min           1.000000
25%           1.000000
50%           1.000000
75%           1.000000
```

```
max           5.000000
Name: Race, dtype: float64
```

[65]:
```
### Countplot of Race variable
### From here we can find out that majority of claims are from Race 1
fig=plt.figure(figsize=(8,6))
sns.countplot(Train_Patient_data['Race'])
fig.tight_layout()
```



[66]:
```
### Now in Race column we do 'one hot encoding' so that ranking of values␣
 ↪doesn't occur here

from sklearn.preprocessing import OneHotEncoder
onehotencoder = OneHotEncoder()
x = onehotencoder.fit_transform(Train_Patient_data.Race.values.reshape(-1, 1)).
 ↪toarray()
```

[67]:
```
df_OneHot = pd.DataFrame(x, columns = ["Race_"+str(int(i)) for i in range(1,5)])
df_OneHot
```

[67]:
```
        Race_1  Race_2  Race_3  Race_4
0          1.0     0.0     0.0     0.0
1          1.0     0.0     0.0     0.0
2          1.0     0.0     0.0     0.0
3          0.0     1.0     0.0     0.0
4          0.0     1.0     0.0     0.0
…          …       …       …       …
558206     1.0     0.0     0.0     0.0
558207     1.0     0.0     0.0     0.0
558208     1.0     0.0     0.0     0.0
558209     1.0     0.0     0.0     0.0
558210     1.0     0.0     0.0     0.0

[558211 rows x 4 columns]
```

[68]:
```
df_OneHot.drop('Race_1',axis=1,inplace=True) ## Drop the first column "Race_1"
 ↪this we need to drop when we do oneHotEncoding
df_OneHot
```

[68]:
```
        Race_2  Race_3  Race_4
0          0.0     0.0     0.0
1          0.0     0.0     0.0
2          0.0     0.0     0.0
3          1.0     0.0     0.0
4          1.0     0.0     0.0
…          …       …       …
558206     0.0     0.0     0.0
558207     0.0     0.0     0.0
558208     0.0     0.0     0.0
558209     0.0     0.0     0.0
558210     0.0     0.0     0.0

[558211 rows x 3 columns]
```

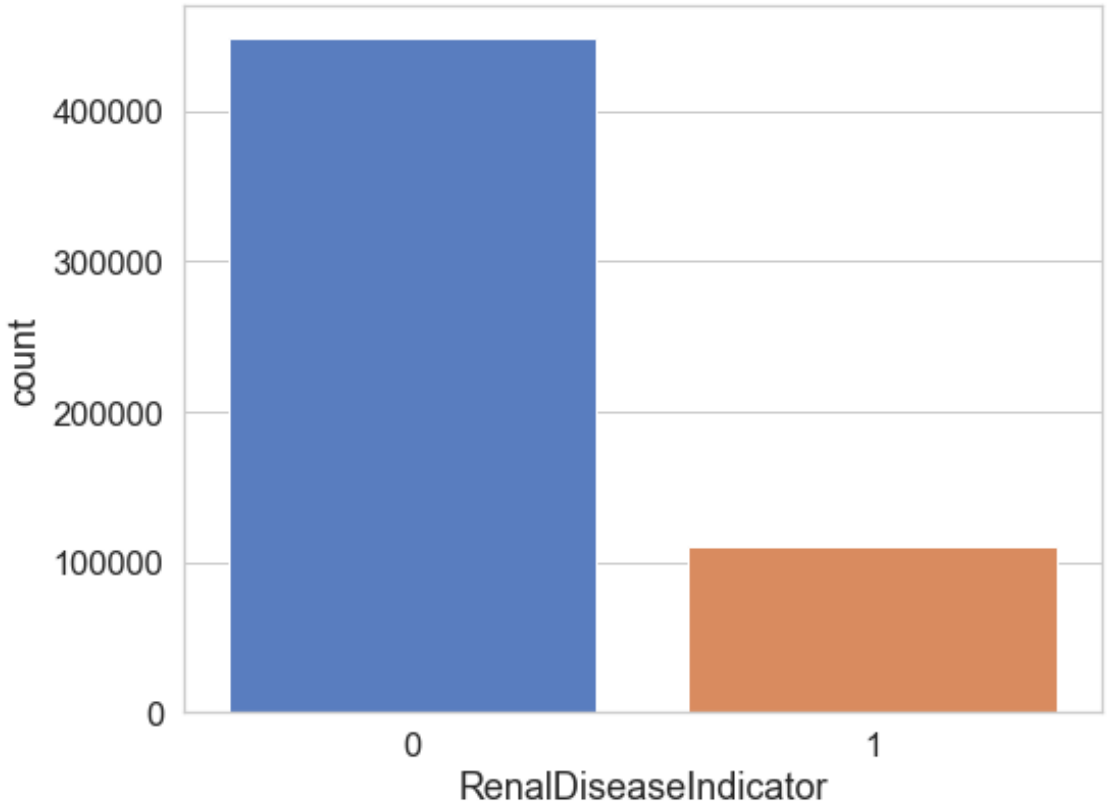
[69]:
```
## Concatenation of dataframe "df_oneHot" that we created above in our main
 ↪dataset

Train_Patient_data = pd.concat([Train_Patient_data, df_OneHot], axis=1)


Train_Patient_data.drop(['Race'], axis=1,inplace=True)  #So now we do not need
 ↪this race column so we are droping this also
```

3. RenealDiseaseIndicator

[70]:
```
Train_Patient_data['RenalDiseaseIndicator'].describe()
```

[70]:
```
count     558211
unique         2
top            0
freq      448363
Name: RenalDiseaseIndicator, dtype: object
```

[71]:
```
## Countplot of "RenalDiseaseIndicator" variable from here we can findout that␣
 ↪maximu disease doesn't have any RenalDisease
fig=plt.figure(figsize=(8,6))
sns.countplot(Train_Patient_data['RenalDiseaseIndicator'])
fig.tight_layout()
```



[72]:
```
Train_Patient_data['RenalDiseaseIndicator']=Train_Patient_data.
 ↪RenalDiseaseIndicator.astype(int) # Change of datatype from object to int


Train_Patient_data['RenalDiseaseIndicator'].describe()
```

[72]:
```
count    558211.000000
mean          0.196786
std           0.397569
```

```
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max           1.000000
Name: RenalDiseaseIndicator, dtype: float64
```

4. State and County

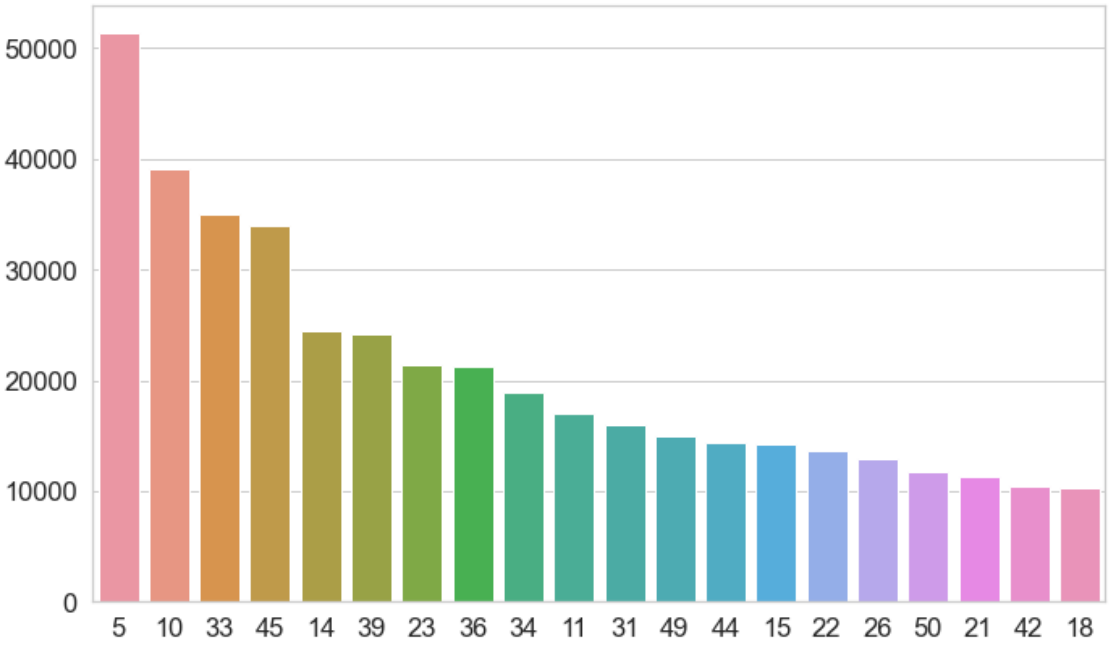
[73]:
```
Train_Patient_data[['State','County']].describe()
```

[73]:
```
                State         County
count  558211.000000  558211.000000
mean       25.446969     378.588195
std        15.192784     265.215531
min         1.000000       0.000000
25%        11.000000     150.000000
50%        24.000000     350.000000
75%        38.000000     570.000000
max        54.000000     999.000000
```

[74]:
```
#Find out which state has maximum count of claims

state_count=Train_Patient_data['State'].value_counts()
state_count=state_count[:20]
state_count
```

[74]:
```
5     51350
10    39073
33    35024
45    34022
14    24417
39    24251
23    21343
36    21291
34    18905
11    17003
31    15940
49    14997
44    14418
15    14213
22    13624
26    12911
50    11740
21    11261
42    10491
18    10322
Name: State, dtype: int64
```

[75]:
```
##Count plot of top 20 states which have maximum claims

## from here we can see that state code 5 has maximum number of claims

fig=plt.figure(figsize=(10,6))
sns.barplot(state_count.index,state_count.values,order=state_count.index)
fig.tight_layout()
```

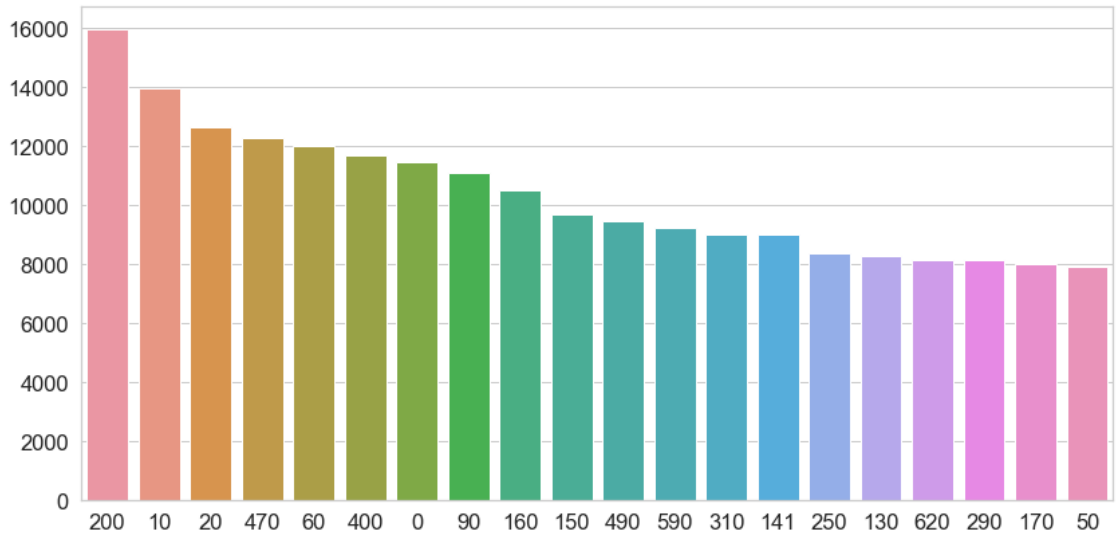


[76]:
```
#Find out which County has maximum count of claims
county_count=Train_Patient_data['County'].value_counts()
county_count=county_count[:20]
county_count
```

[76]:
```
200    15957
10     13982
20     12632
470    12278
60     11995
400    11697
0      11481
90     11086
160    10533
150     9687
490     9487
590     9227
310     9027
```

```
141     8995
250     8384
130     8283
620     8170
290     8142
170     8017
50      7934
Name: County, dtype: int64
```

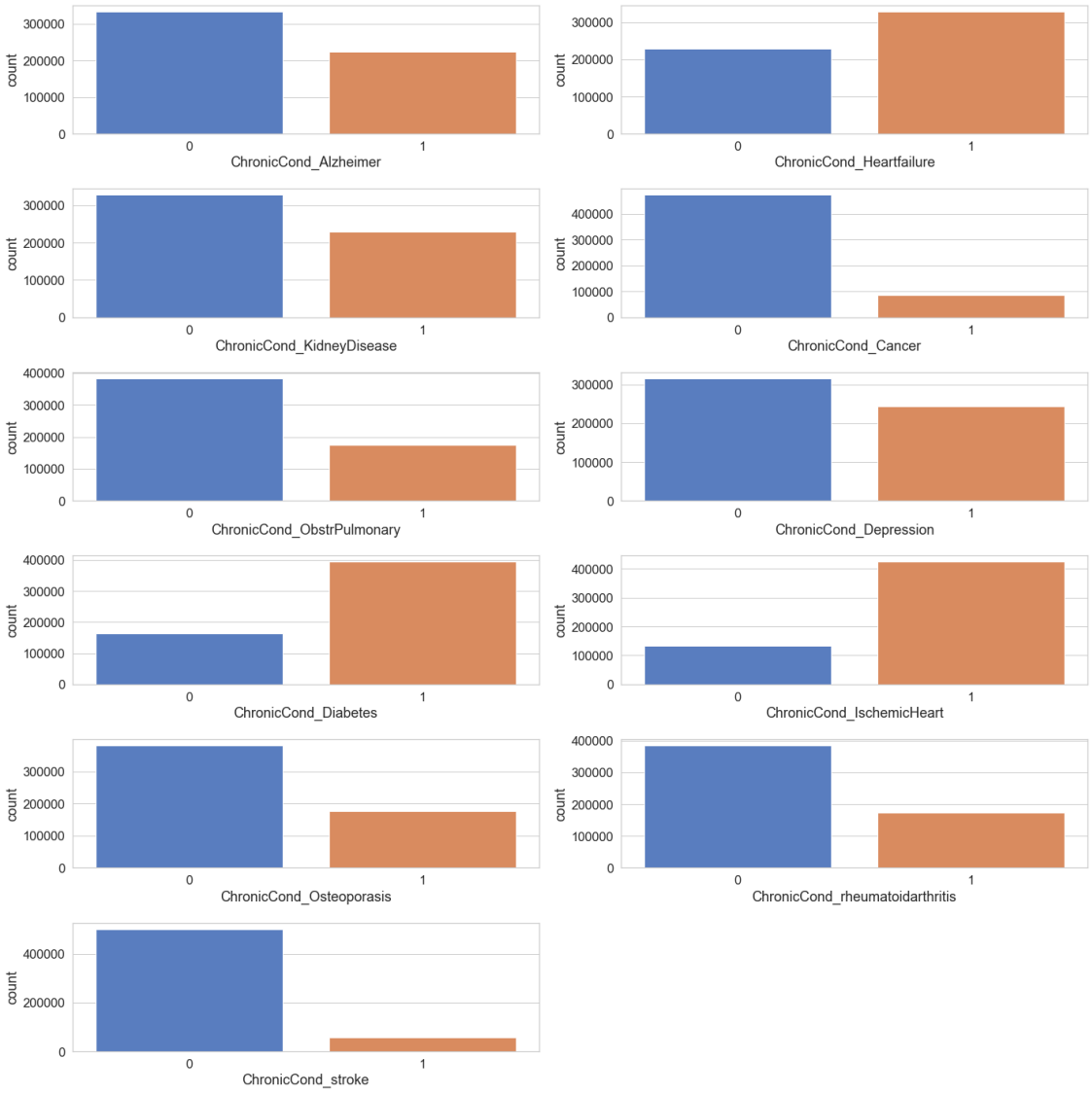
[77]:
```
##Count plot of top 20 County which have maximum claims

## from here we can see that County code 200 has maximum number of claims


fig=plt.figure(figsize=(12,6))
sns.barplot(county_count.index,county_count.values,order=county_count.index)
fig.tight_layout()
```



[78]:
```
#### 5. Chronic_cond
```

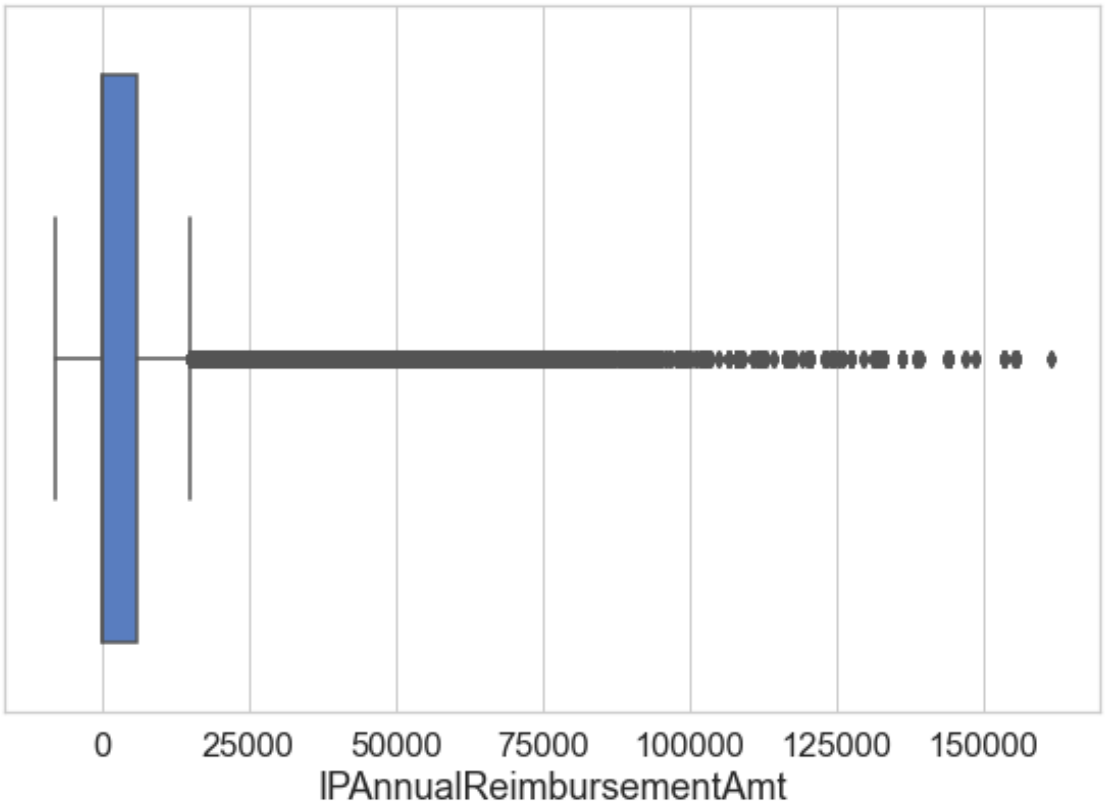
[79]:
```
## Visulization of ChronicCond Variables

## From this we can findout that how many claims has ChronicCond diseases, for␣
 ↪eg: In ChronicCond_Alzheimer more than 3 lacs claims doesn't have this and␣
 ↪remaining claims approx( 2 lacs) have ChronicCond_Alzheimer

fig=plt.figure(figsize=(20,20))

for col in range(1,12):
```

```
    plt.subplot(6,2,col)
    sns.countplot(Train_Patient_data.iloc[:,37+col])

fig.tight_layout()
```

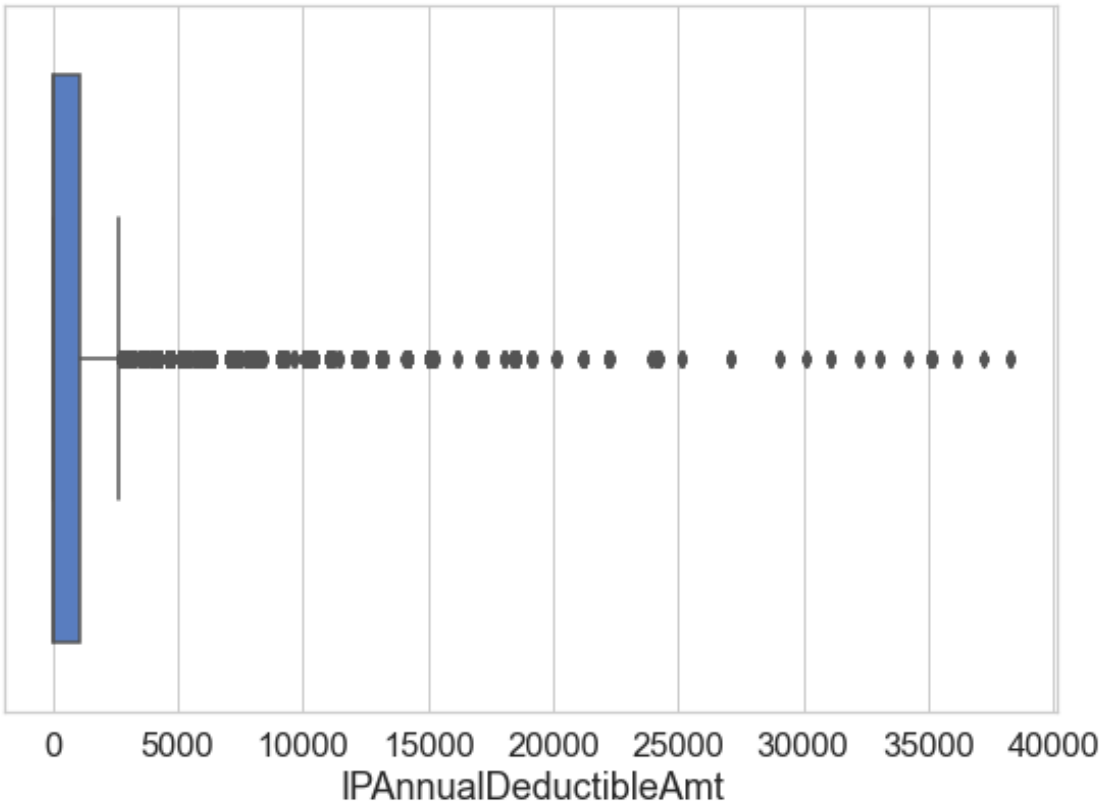


Boxplots of some numerical features to check the distribution of data

[80]:
```
## Boxplot of "IPAnnualReimbursementAmt" and we can see in this boxplot data is␣
 ↪not normally distributed and it is left skewed

fig=plt.figure(figsize=(8,6))
sns.boxplot(Train_Patient_data['IPAnnualReimbursementAmt'])
fig.tight_layout()
```



[81]:
```
## Boxplot of "IPAnnualDeductibleAmt" and we can see in this boxplot data is␣
 ↪not normally distributed and it is left skewed


fig=plt.figure(figsize=(8,6))
sns.boxplot(Train_Patient_data['IPAnnualDeductibleAmt'])
fig.tight_layout()
```



0.0.6 Handling Missing values and add new features in Test_Patient_data

[82]:
```
Test_Patient_data.isnull().sum()
```

[82]:
```
BeneID                         0
ClaimID                        0
ClaimStartDt                   0
ClaimEndDt                     0
Provider                       0
InscClaimAmtReimbursed         0
AttendingPhysician           347
OperatingPhysician        108199
OtherPhysician             86760
AdmissionDt               125841
ClmAdmitDiagnosisCode     100036
DeductibleAmtPaid            196
DischargeDt               125841
DiagnosisGroupCode        125841
ClmDiagnosisCode_1          2578
ClmDiagnosisCode_2         47785
ClmDiagnosisCode_3         76744
```

```
ClmDiagnosisCode_4                 95775
ClmDiagnosisCode_5                108594
ClmDiagnosisCode_6                115232
ClmDiagnosisCode_7                119607
ClmDiagnosisCode_8                122670
ClmDiagnosisCode_9                125516
ClmDiagnosisCode_10               134242
ClmProcedureCode_1                129925
ClmProcedureCode_2                134129
ClmProcedureCode_3                135167
ClmProcedureCode_4                135363
ClmProcedureCode_5                135390
ClmProcedureCode_6                135392
AdmitForDays                      125841
DOB                                    0
DOD                               134352
Gender                                 0
Race                                   0
RenalDiseaseIndicator                  0
State                                  0
County                                 0
NoOfMonths_PartACov                    0
NoOfMonths_PartBCov                    0
ChronicCond_Alzheimer                  0
ChronicCond_Heartfailure               0
ChronicCond_KidneyDisease              0
ChronicCond_Cancer                     0
ChronicCond_ObstrPulmonary             0
ChronicCond_Depression                 0
ChronicCond_Diabetes                   0
ChronicCond_IschemicHeart              0
ChronicCond_Osteoporasis               0
ChronicCond_rheumatoidarthritis        0
ChronicCond_stroke                     0
IPAnnualReimbursementAmt               0
IPAnnualDeductibleAmt                  0
OPAnnualReimbursementAmt               0
OPAnnualDeductibleAmt                  0
Age                                    0
WhetherDead                            0
dtype: int64
```

[83]:
```
## We are replacing these columns value with 0 and 1 where we have value we are␣
↪replacing it with 1 and in place of null value we replace it with 0.
```

```
Test_Patient_data[['AttendingPhysician', 'OperatingPhysician',␣
↪'OtherPhysician']] = np.where(Test_Patient_data[['AttendingPhysician',␣
↪'OperatingPhysician', 'OtherPhysician']].isnull(), 0, 1)

Test_Patient_data['N_Types_Physicians'] =␣
↪Test_Patient_data['AttendingPhysician'] + ␣
↪Test_Patient_data['OperatingPhysician'] + Test_Patient_data['OtherPhysician']
```

[84]:
```
Test_Patient_data['IsDiagnosisCode'] = np.where(Test_Patient_data.
↪DiagnosisGroupCode.notnull(), 1, 0)
Test_Patient_data = Test_Patient_data.drop(['DiagnosisGroupCode'], axis = 1)
```

[85]:
```
Test_Patient_data.isnull().sum()
```

[85]:
```
BeneID                         0
ClaimID                        0
ClaimStartDt                   0
ClaimEndDt                     0
Provider                       0
InscClaimAmtReimbursed         0
AttendingPhysician             0
OperatingPhysician             0
OtherPhysician                 0
AdmissionDt               125841
ClmAdmitDiagnosisCode     100036
DeductibleAmtPaid            196
DischargeDt               125841
ClmDiagnosisCode_1          2578
ClmDiagnosisCode_2         47785
ClmDiagnosisCode_3         76744
ClmDiagnosisCode_4         95775
ClmDiagnosisCode_5        108594
ClmDiagnosisCode_6        115232
ClmDiagnosisCode_7        119607
ClmDiagnosisCode_8        122670
ClmDiagnosisCode_9        125516
ClmDiagnosisCode_10       134242
ClmProcedureCode_1        129925
ClmProcedureCode_2        134129
ClmProcedureCode_3        135167
ClmProcedureCode_4        135363
ClmProcedureCode_5        135390
ClmProcedureCode_6        135392
AdmitForDays              125841
DOB                            0
DOD                       134352
Gender                         0
```

```
Race                                  0
RenalDiseaseIndicator                 0
State                                 0
County                                0
NoOfMonths_PartACov                   0
NoOfMonths_PartBCov                   0
ChronicCond_Alzheimer                 0
ChronicCond_Heartfailure              0
ChronicCond_KidneyDisease             0
ChronicCond_Cancer                    0
ChronicCond_ObstrPulmonary            0
ChronicCond_Depression                0
ChronicCond_Diabetes                  0
ChronicCond_IschemicHeart             0
ChronicCond_Osteoporasis              0
ChronicCond_rheumatoidarthritis       0
ChronicCond_stroke                    0
IPAnnualReimbursementAmt              0
IPAnnualDeductibleAmt                 0
OPAnnualReimbursementAmt              0
OPAnnualDeductibleAmt                 0
Age                                   0
WhetherDead                           0
N_Types_Physicians                    0
IsDiagnosisCode                       0
dtype: int64
```

[86]:
```
Test_Patient_data['DeductibleAmtPaid'].describe()
```

[86]:
```
count    135196.000000
mean         76.499194
std         270.779562
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max        1068.000000
Name: DeductibleAmtPaid, dtype: float64
```

[87]:
```
Test_Patient_data['DeductibleAmtPaid'].fillna(0,inplace=True)

Test_Patient_data['IsDeductibleAmtPaid']=np.
 ↪where(Test_Patient_data['DeductibleAmtPaid']==0,0,1)


Test_Patient_data['IsDeductibleAmtPaid'].value_counts()
```

[87]:
```
0    120907
1     14485
Name: IsDeductibleAmtPaid, dtype: int64
```

[88]:
```
Test_Patient_data['AdmitForDays'].isnull().sum()
```

[88]:
```
125841
```

[89]:
```
Test_Patient_data['AdmitForDays'].fillna(0,inplace=True)
```

[90]:
```
Test_Patient_data.Gender.describe()
```

[90]:
```
count    135392.000000
mean          1.576231
std           0.494157
min           1.000000
25%           1.000000
50%           2.000000
75%           2.000000
max           2.000000
Name: Gender, dtype: float64
```

[91]:
```
Test_Patient_data['Gender']=Test_Patient_data['Gender'].replace(2,0)
```

[92]:
```
Test_Patient_data['Race'].describe()
```

[92]:
```
count    135392.000000
mean          1.240605
std           0.695578
min           1.000000
25%           1.000000
50%           1.000000
75%           1.000000
max           5.000000
Name: Race, dtype: float64
```

[93]:
```
onehotencoder = OneHotEncoder()
x = onehotencoder.fit_transform(Test_Patient_data.Race.values.reshape(-1, 1)).
 ↪toarray()

df_test_OneHot = pd.DataFrame(x, columns = ["Race_"+str(int(i)) for i in␣
 ↪range(1,5)])
df_test_OneHot
```

[93]:

	Race_1	Race_2	Race_3	Race_4
0	1.0	0.0	0.0	0.0
1	1.0	0.0	0.0	0.0
2	1.0	0.0	0.0	0.0

```
3          1.0     0.0     0.0     0.0
4          1.0     0.0     0.0     0.0
…          …       …       …       …
135387     1.0     0.0     0.0     0.0
135388     1.0     0.0     0.0     0.0
135389     1.0     0.0     0.0     0.0
135390     1.0     0.0     0.0     0.0
135391     1.0     0.0     0.0     0.0

[135392 rows x 4 columns]
```

[94]:
```
df_test_OneHot.drop('Race_1',axis=1,inplace=True)


Test_Patient_data = pd.concat([Test_Patient_data, df_test_OneHot], axis=1)

#droping the country column
```

[95]:
```
Test_Patient_data.drop(['Race'], axis=1,inplace=True)
```

[96]:
```
Test_Patient_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 135392 entries, 0 to 135391
Data columns (total 61 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   BeneID                          135392 non-null  object
 1   ClaimID                         135392 non-null  object
 2   ClaimStartDt                    135392 non-null  object
 3   ClaimEndDt                      135392 non-null  object
 4   Provider                        135392 non-null  object
 5   InscClaimAmtReimbursed          135392 non-null  int64
 6   AttendingPhysician              135392 non-null  int32
 7   OperatingPhysician              135392 non-null  int32
 8   OtherPhysician                  135392 non-null  int32
 9   AdmissionDt                     9551 non-null    datetime64[ns]
 10  ClmAdmitDiagnosisCode           35356 non-null   object
 11  DeductibleAmtPaid               135392 non-null  float64
 12  DischargeDt                     9551 non-null    datetime64[ns]
 13  ClmDiagnosisCode_1              132814 non-null  object
 14  ClmDiagnosisCode_2              87607 non-null   object
 15  ClmDiagnosisCode_3              58648 non-null   object
 16  ClmDiagnosisCode_4              39617 non-null   object
 17  ClmDiagnosisCode_5              26798 non-null   object
 18  ClmDiagnosisCode_6              20160 non-null   object
 19  ClmDiagnosisCode_7              15785 non-null   object
 20  ClmDiagnosisCode_8              12722 non-null   object
```

```
 21  ClmDiagnosisCode_9               9876 non-null    object
 22  ClmDiagnosisCode_10              1150 non-null    object
 23  ClmProcedureCode_1               5467 non-null    float64
 24  ClmProcedureCode_2               1263 non-null    float64
 25  ClmProcedureCode_3               225 non-null     float64
 26  ClmProcedureCode_4               29 non-null      float64
 27  ClmProcedureCode_5               2 non-null       float64
 28  ClmProcedureCode_6               0 non-null       float64
 29  AdmitForDays                     135392 non-null  float64
 30  DOB                              135392 non-null  datetime64[ns]
 31  DOD                              1040 non-null    datetime64[ns]
 32  Gender                           135392 non-null  int64
 33  RenalDiseaseIndicator            135392 non-null  object
 34  State                            135392 non-null  int64
 35  County                           135392 non-null  int64
 36  NoOfMonths_PartACov              135392 non-null  int64
 37  NoOfMonths_PartBCov              135392 non-null  int64
 38  ChronicCond_Alzheimer            135392 non-null  int64
 39  ChronicCond_Heartfailure         135392 non-null  int64
 40  ChronicCond_KidneyDisease        135392 non-null  int64
 41  ChronicCond_Cancer               135392 non-null  int64
 42  ChronicCond_ObstrPulmonary       135392 non-null  int64
 43  ChronicCond_Depression           135392 non-null  int64
 44  ChronicCond_Diabetes             135392 non-null  int64
 45  ChronicCond_IschemicHeart        135392 non-null  int64
 46  ChronicCond_Osteoporasis         135392 non-null  int64
 47  ChronicCond_rheumatoidarthritis  135392 non-null  int64
 48  ChronicCond_stroke               135392 non-null  int64
 49  IPAnnualReimbursementAmt         135392 non-null  int64
 50  IPAnnualDeductibleAmt            135392 non-null  int64
 51  OPAnnualReimbursementAmt         135392 non-null  int64
 52  OPAnnualDeductibleAmt            135392 non-null  int64
 53  Age                              135392 non-null  float64
 54  WhetherDead                      135392 non-null  float64
 55  N_Types_Physicians               135392 non-null  int32
 56  IsDiagnosisCode                  135392 non-null  int32
 57  IsDeductibleAmtPaid              135392 non-null  int32
 58  Race_2                           135392 non-null  float64
 59  Race_3                           135392 non-null  float64
 60  Race_4                           135392 non-null  float64
dtypes: datetime64[ns](4), float64(13), int32(6), int64(21), object(17)
memory usage: 60.9+ MB
```

[97]:
```
Test_Patient_data['RenalDiseaseIndicator'].describe()
```

[97]:
```
count     135392
unique         2
```

```
top            0
freq      109143
Name: RenalDiseaseIndicator, dtype: object
```

[98]:
```
Test_Patient_data['RenalDiseaseIndicator']=Test_Patient_data.
 ↪RenalDiseaseIndicator.astype(int)
```

[99]:
```
Test_Patient_data[ClmProcedure_vars].describe()
```

[99]:

	ClmProcedureCode_1	ClmProcedureCode_2	ClmProcedureCode_3	\
count	5467.000000	1263.000000	225.000000	
mean	5905.430766	4138.790182	4182.213333	
std	3057.976988	2042.016095	2165.057828	
min	14.000000	42.000000	185.000000	
25%	3891.000000	2749.000000	2724.000000	
50%	5369.000000	4019.000000	4019.000000	
75%	8741.000000	4439.000000	5121.000000	
max	9999.000000	9998.000000	9984.000000	

	ClmProcedureCode_4	ClmProcedureCode_5	ClmProcedureCode_6
count	29.000000	2.000000	0.0
mean	4509.931034	7055.500000	NaN
std	2571.379659	4124.553855	NaN
min	260.000000	4139.000000	NaN
25%	3320.000000	5597.250000	NaN
50%	4263.000000	7055.500000	NaN
75%	5781.000000	8513.750000	NaN
max	9971.000000	9972.000000	NaN

[100]:
```
# We count the number of procedures for each claim
Test_Patient_data['N_Procedure'] =␣
 ↪N_unique_values(Test_Patient_data[ClmProcedure_vars])
```

[101]:
```
Test_Patient_data['N_Procedure'].value_counts()
```

[101]:
```
0    129925
1      4204
2      1038
3       196
4        27
5         2
Name: N_Procedure, dtype: int64
```

[102]:
```
# We count the number of CLMDiagnosisCode for each claim and store these value␣
 ↪in a new variable
```

```
Test_Patient_data['N_UniqueDiagnosis_Claims'] =␣
 ↪N_unique_values(Test_Patient_data[ClmDiagnosisCode_vars])

Test_Patient_data['N_UniqueDiagnosis_Claims'].value_counts()
```

[102]:
```
1     37149
2     32041
3     20966
4     13979
5      7368
10     5193
9      5095
6      4696
7      3367
8      2754
0      2086
11      698
Name: N_UniqueDiagnosis_Claims, dtype: int64
```

[103]:
```
print('\033[1m'+"Train Patient Dataset"+ "\033[0m")

print(Train_Patient_data.info())

print('\033[1m'+"Test Patient Dataset"+ "\033[0m")

print(Test_Patient_data.info())
```

```
Train Patient Dataset
<class 'pandas.core.frame.DataFrame'>
Int64Index: 558211 entries, 0 to 558210
Data columns (total 63 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   BeneID                          558211 non-null  object
 1   ClaimID                         558211 non-null  object
 2   ClaimStartDt                    558211 non-null  object
 3   ClaimEndDt                      558211 non-null  object
 4   Provider                        558211 non-null  object
 5   InscClaimAmtReimbursed          558211 non-null  int64
 6   AttendingPhysician              558211 non-null  int32
 7   OperatingPhysician              558211 non-null  int32
 8   OtherPhysician                  558211 non-null  int32
 9   AdmissionDt                     40474 non-null   datetime64[ns]
 10  ClmAdmitDiagnosisCode           145899 non-null  object
 11  DeductibleAmtPaid               558211 non-null  float64
 12  DischargeDt                     40474 non-null   datetime64[ns]
 13  ClmDiagnosisCode_1              547758 non-null  object
 14  ClmDiagnosisCode_2              362605 non-null  object
```

```
15  ClmDiagnosisCode_3              243055 non-null  object
16  ClmDiagnosisCode_4              164536 non-null  object
17  ClmDiagnosisCode_5              111924 non-null  object
18  ClmDiagnosisCode_6              84392 non-null   object
19  ClmDiagnosisCode_7              66177 non-null   object
20  ClmDiagnosisCode_8              53444 non-null   object
21  ClmDiagnosisCode_9              41815 non-null   object
22  ClmDiagnosisCode_10             5010 non-null    object
23  ClmProcedureCode_1              23310 non-null   float64
24  ClmProcedureCode_2              5490 non-null    float64
25  ClmProcedureCode_3              969 non-null     float64
26  ClmProcedureCode_4              118 non-null     float64
27  ClmProcedureCode_5              9 non-null       float64
28  ClmProcedureCode_6              0 non-null       float64
29  AdmitForDays                    558211 non-null  float64
30  DOB                             558211 non-null  datetime64[ns]
31  DOD                             4131 non-null    datetime64[ns]
32  Gender                          558211 non-null  int64
33  RenalDiseaseIndicator           558211 non-null  int32
34  State                           558211 non-null  int64
35  County                          558211 non-null  int64
36  NoOfMonths_PartACov             558211 non-null  int64
37  NoOfMonths_PartBCov             558211 non-null  int64
38  ChronicCond_Alzheimer           558211 non-null  int64
39  ChronicCond_Heartfailure        558211 non-null  int64
40  ChronicCond_KidneyDisease       558211 non-null  int64
41  ChronicCond_Cancer              558211 non-null  int64
42  ChronicCond_ObstrPulmonary      558211 non-null  int64
43  ChronicCond_Depression          558211 non-null  int64
44  ChronicCond_Diabetes            558211 non-null  int64
45  ChronicCond_IschemicHeart       558211 non-null  int64
46  ChronicCond_Osteoporasis        558211 non-null  int64
47  ChronicCond_rheumatoidarthritis 558211 non-null  int64
48  ChronicCond_stroke              558211 non-null  int64
49  IPAnnualReimbursementAmt        558211 non-null  int64
50  IPAnnualDeductibleAmt           558211 non-null  int64
51  OPAnnualReimbursementAmt        558211 non-null  int64
52  OPAnnualDeductibleAmt           558211 non-null  int64
53  Age                             558211 non-null  float64
54  WhetherDead                     558211 non-null  float64
55  N_Types_Physicians              558211 non-null  int32
56  IsDiagnosisCode                 558211 non-null  int32
57  IsDeductibleAmtPaid             558211 non-null  int32
58  N_Procedure                     558211 non-null  int32
59  N_UniqueDiagnosis_Claims        558211 non-null  int32
60  Race_2                          558211 non-null  float64
61  Race_3                          558211 non-null  float64
62  Race_4                          558211 non-null  float64
```

```
dtypes: datetime64[ns](4), float64(13), int32(9), int64(21), object(16)
memory usage: 269.5+ MB
None
```
Test Patient Dataset
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 135392 entries, 0 to 135391
Data columns (total 63 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   BeneID                          135392 non-null  object
 1   ClaimID                         135392 non-null  object
 2   ClaimStartDt                    135392 non-null  object
 3   ClaimEndDt                      135392 non-null  object
 4   Provider                        135392 non-null  object
 5   InscClaimAmtReimbursed          135392 non-null  int64
 6   AttendingPhysician              135392 non-null  int32
 7   OperatingPhysician              135392 non-null  int32
 8   OtherPhysician                  135392 non-null  int32
 9   AdmissionDt                     9551 non-null    datetime64[ns]
 10  ClmAdmitDiagnosisCode           35356 non-null   object
 11  DeductibleAmtPaid               135392 non-null  float64
 12  DischargeDt                     9551 non-null    datetime64[ns]
 13  ClmDiagnosisCode_1              132814 non-null  object
 14  ClmDiagnosisCode_2              87607 non-null   object
 15  ClmDiagnosisCode_3              58648 non-null   object
 16  ClmDiagnosisCode_4              39617 non-null   object
 17  ClmDiagnosisCode_5              26798 non-null   object
 18  ClmDiagnosisCode_6              20160 non-null   object
 19  ClmDiagnosisCode_7              15785 non-null   object
 20  ClmDiagnosisCode_8              12722 non-null   object
 21  ClmDiagnosisCode_9              9876 non-null    object
 22  ClmDiagnosisCode_10             1150 non-null    object
 23  ClmProcedureCode_1              5467 non-null    float64
 24  ClmProcedureCode_2              1263 non-null    float64
 25  ClmProcedureCode_3              225 non-null     float64
 26  ClmProcedureCode_4              29 non-null      float64
 27  ClmProcedureCode_5              2 non-null       float64
 28  ClmProcedureCode_6              0 non-null       float64
 29  AdmitForDays                    135392 non-null  float64
 30  DOB                             135392 non-null  datetime64[ns]
 31  DOD                             1040 non-null    datetime64[ns]
 32  Gender                          135392 non-null  int64
 33  RenalDiseaseIndicator           135392 non-null  int32
 34  State                           135392 non-null  int64
 35  County                          135392 non-null  int64
 36  NoOfMonths_PartACov             135392 non-null  int64
 37  NoOfMonths_PartBCov             135392 non-null  int64
 38  ChronicCond_Alzheimer           135392 non-null  int64
```

```
 39  ChronicCond_Heartfailure          135392 non-null  int64
 40  ChronicCond_KidneyDisease         135392 non-null  int64
 41  ChronicCond_Cancer                135392 non-null  int64
 42  ChronicCond_ObstrPulmonary        135392 non-null  int64
 43  ChronicCond_Depression            135392 non-null  int64
 44  ChronicCond_Diabetes              135392 non-null  int64
 45  ChronicCond_IschemicHeart         135392 non-null  int64
 46  ChronicCond_Osteoporasis          135392 non-null  int64
 47  ChronicCond_rheumatoidarthritis   135392 non-null  int64
 48  ChronicCond_stroke                135392 non-null  int64
 49  IPAnnualReimbursementAmt          135392 non-null  int64
 50  IPAnnualDeductibleAmt             135392 non-null  int64
 51  OPAnnualReimbursementAmt          135392 non-null  int64
 52  OPAnnualDeductibleAmt             135392 non-null  int64
 53  Age                               135392 non-null  float64
 54  WhetherDead                       135392 non-null  float64
 55  N_Types_Physicians                135392 non-null  int32
 56  IsDiagnosisCode                   135392 non-null  int32
 57  IsDeductibleAmtPaid               135392 non-null  int32
 58  Race_2                            135392 non-null  float64
 59  Race_3                            135392 non-null  float64
 60  Race_4                            135392 non-null  float64
 61  N_Procedure                       135392 non-null  int32
 62  N_UniqueDiagnosis_Claims          135392 non-null  int32
dtypes: datetime64[ns](4), float64(13), int32(9), int64(21), object(16)
memory usage: 61.5+ MB
None
```

0.1 Merging of Train and Test dataframe with Train_Patient_data and Test_Patient_data respectively to create a Final Dataframe for Train and Test for modelling

[104]:
```
### Count number of records
## From here we get the count of BeneID and ClaimId for each provider

## For Train
Train_Count = Train_Patient_data[['BeneID', 'ClaimID']].
 ↪groupby(Train_Patient_data['Provider']).nunique().reset_index()
Train_Count.rename(columns={'BeneID':'BeneID_count','ClaimID':
 ↪'ClaimID_count'},inplace=True)


## For Test
Test_Count = Test_Patient_data[['BeneID', 'ClaimID']].
 ↪groupby(Test_Patient_data['Provider']).nunique().reset_index()
Test_Count.rename(columns={'BeneID':'BeneID_count','ClaimID':
 ↪'ClaimID_count'},inplace=True)
```

[105]:
```
Train_Count
```

[105]:

	Provider	BeneID_count	ClaimID_count
0	PRV51001	24	25
1	PRV51003	117	132
2	PRV51004	138	149
3	PRV51005	495	1165
4	PRV51007	58	72
…	…	…	…
5405	PRV57759	24	28
5406	PRV57760	9	22
5407	PRV57761	67	82
5408	PRV57762	1	1
5409	PRV57763	70	118

[5410 rows x 3 columns]

[106]:
```
Test_Count
```

[106]:

	Provider	BeneID_count	ClaimID_count
0	PRV51002	169	205
1	PRV51006	81	102
2	PRV51009	30	39
3	PRV51010	25	38
4	PRV51018	146	190
…	…	…	…
1348	PRV57713	10	11
1349	PRV57726	8	8
1350	PRV57745	2	2
1351	PRV57749	45	49
1352	PRV57750	94	105

[1353 rows x 3 columns]

[107]:
```
Train_Data_Sum = Train_Patient_data.groupby(['Provider'], as_index =␣
↪False)[['InscClaimAmtReimbursed', 'DeductibleAmtPaid',␣
↪'RenalDiseaseIndicator',
                                                          ␣
↪'AttendingPhysician','OperatingPhysician','OtherPhysician','AdmitForDays',
                                                            'ChronicCond_Alzheimer',␣
↪'ChronicCond_Heartfailure','ChronicCond_Cancer',
                                                         ␣
↪'ChronicCond_KidneyDisease', 'ChronicCond_ObstrPulmonary',
                                                        ␣
↪'ChronicCond_Depression','ChronicCond_Diabetes', 'ChronicCond_IschemicHeart',
                                                            'ChronicCond_Osteoporasis',␣
↪'ChronicCond_rheumatoidarthritis',
```

```
                                                        'ChronicCond_stroke',␣
↪'IPAnnualReimbursementAmt','IPAnnualDeductibleAmt',
                                                        'OPAnnualReimbursementAmt',␣
↪'OPAnnualDeductibleAmt', 'WhetherDead',
                                                       ␣
↪'N_Types_Physicians','IsDiagnosisCode', 'N_Procedure',␣
↪'N_UniqueDiagnosis_Claims']].sum()

Test_Data_Sum = Test_Patient_data.groupby(['Provider'], as_index =␣
↪False)[['InscClaimAmtReimbursed', 'DeductibleAmtPaid',␣
↪'RenalDiseaseIndicator',
                                                       ␣
↪'AttendingPhysician','OperatingPhysician','OtherPhysician','AdmitForDays',
                                                        'ChronicCond_Alzheimer',␣
↪'ChronicCond_Heartfailure','ChronicCond_Cancer',
                                                       ␣
↪'ChronicCond_KidneyDisease', 'ChronicCond_ObstrPulmonary',
                                                       ␣
↪'ChronicCond_Depression','ChronicCond_Diabetes', 'ChronicCond_IschemicHeart',
                                                        'ChronicCond_Osteoporasis',␣
↪'ChronicCond_rheumatoidarthritis',
                                                        'ChronicCond_stroke',␣
↪'IPAnnualReimbursementAmt','IPAnnualDeductibleAmt',
                                                        'OPAnnualReimbursementAmt',␣
↪'OPAnnualDeductibleAmt', 'WhetherDead',
                                                       ␣
↪'N_Types_Physicians','IsDiagnosisCode', 'N_Procedure',␣
↪'N_UniqueDiagnosis_Claims']].sum()
```

[108]:
```
Train_Data_Sum
```

[108]:

	Provider	InscClaimAmtReimbursed	DeductibleAmtPaid	\
0	PRV51001	104640	5340.0	
1	PRV51003	605670	66286.0	
2	PRV51004	52170	310.0	
3	PRV51005	280910	3700.0	
4	PRV51007	33710	3264.0	
…	…	…	…	
5405	PRV57759	10640	130.0	
5406	PRV57760	4770	0.0	
5407	PRV57761	18470	370.0	
5408	PRV57762	1900	0.0	
5409	PRV57763	43610	390.0	

	RenalDiseaseIndicator	AttendingPhysician	OperatingPhysician	\
0	8	25	5	

```
1                        29                  132                   45
2                        23                  149                   27
3                       259                 1163                  222
4                        11                   72                   12
…                       …                    …                     …
5405                      5                   28                    1
5406                      0                   22                    6
5407                     23                   82                   14
5408                      0                    1                    0
5409                     25                  118                   19

      OtherPhysician  AdmitForDays  ChronicCond_Alzheimer  \
0                 10          30.0                     15
1                 25         382.0                     56
2                 63           0.0                     64
3                478           0.0                    426
4                 26          19.0                     26
…                …             …                       …
5405              12           0.0                     14
5406               9           0.0                      3
5407              36           0.0                     36
5408               0           0.0                      0
5409              49           0.0                     45

      ChronicCond_Heartfailure  …  ChronicCond_stroke  \
0                           19  …                   6
1                           80  …                  12
2                           88  …                  17
3                          680  …                 124
4                           40  …                  12
…                          …  …                   …
5405                        20  …                   4
5406                        11  …                   0
5407                        56  …                  10
5408                         0  …                   0
5409                        63  …                   6

      IPAnnualReimbursementAmt  IPAnnualDeductibleAmt  \
0                       440150                  22428
1                       999000                 122948
2                       648430                  64808
3                      4221950                 441724
4                       219600                  32040
…                          …                    …
5405                    110940                  12816
5406                     61280                   9612
5407                    576180                  48060
```

```
5408                    15000                   1068
5409                   424710                  71148

      OPAnnualReimbursementAmt  OPAnnualDeductibleAmt  WhetherDead  \
0                        65380                  11598          0.0
1                       353520                  97300          1.0
2                       327040                  92790          1.0
3                      2457840                 741323          4.0
4                       124540                  33820          1.0
…                          …                      …            …
5405                     90770                  24830          0.0
5406                     32840                  17720          0.0
5407                    240130                  58000          1.0
5408                      2540                    400          0.0
5409                    366850                  94790          0.0

      N_Types_Physicians  IsDiagnosisCode  N_Procedure  \
0                     40                5            3
1                    202               62           48
2                    239                0            0
3                   1863                0            0
4                    110                3            1
…                      …                …            …
5405                  41                0            0
5406                  37                0            0
5407                 132                0            0
5408                   1                0            0
5409                 186                0            0

      N_UniqueDiagnosis_Claims
0                           91
1                          761
2                          410
3                         3246
4                          231
…                            …
5405                        61
5406                        59
5407                       235
5408                         2
5409                       315

[5410 rows x 28 columns]
```

[109]:
```
Test_Data_Sum
```

[109]:

	Provider	InscClaimAmtReimbursed	DeductibleAmtPaid	\
0	PRV51002	53790	380.0	
1	PRV51006	30720	0.0	
2	PRV51009	27230	1238.0	
3	PRV51010	64580	5340.0	
4	PRV51018	61620	670.0	
…	…	…	…	
1348	PRV57713	860	0.0	
1349	PRV57726	1590	0.0	
1350	PRV57745	510	0.0	
1351	PRV57749	9980	370.0	
1352	PRV57750	27020	230.0	

	RenalDiseaseIndicator	AttendingPhysician	OperatingPhysician	\
0	32	205	30	
1	10	102	24	
2	12	38	12	
3	5	38	9	
4	41	190	30	
…	…	…	…	
1348	2	11	0	
1349	0	8	0	
1350	1	2	0	
1351	9	49	8	
1352	17	105	23	

	OtherPhysician	AdmitForDays	ChronicCond_Alzheimer	\
0	77	0.0	79	
1	38	0.0	35	
2	10	8.0	8	
3	10	29.0	21	
4	72	0.0	73	
…	…	…	…	
1348	6	0.0	6	
1349	3	0.0	4	
1350	2	0.0	1	
1351	21	0.0	18	
1352	35	0.0	39	

	ChronicCond_Heartfailure	…	ChronicCond_stroke	\
0	108	…	19	
1	69	…	8	
2	17	…	3	
3	23	…	5	
4	109	…	13	
…	…	…	…	
1348	7	…	0	

```
1349                          5  …                    0
1350                          2  …                    0
1351                         22  …                    4
1352                         62  …                    9

      IPAnnualReimbursementAmt  IPAnnualDeductibleAmt  \
0                      1062090                 112392
1                       384290                  48924
2                       117160                   9612
3                       200200                  20292
4                       900400                 101460
…                          …                      …
1348                      5000                   1068
1349                    105000                   2136
1350                     20060                   2136
1351                    172930                  17088
1352                    355890                  39516

      OPAnnualReimbursementAmt  OPAnnualDeductibleAmt  WhetherDead  \
0                       480740                 138980          3.0
1                       244970                  61800          0.0
2                        95200                  25210          0.0
3                        67480                  25230          0.0
4                       534460                 156960          2.0
…                          …                      …          …
1348                     15710                   5510          0.0
1349                     21210                   9010          0.0
1350                      7430                   1420          0.0
1351                     85060                  24780          2.0
1352                    259740                  64360          0.0

      N_Types_Physicians  IsDiagnosisCode  N_Procedure  \
0                    312                0            0
1                    164                0            0
2                     60                2            3
3                     57                6            3
4                    292                0            0
…                      …                …            …
1348                  17                0            0
1349                  11                0            0
1350                   4                0            0
1351                  78                0            0
1352                 163                0            0

      N_UniqueDiagnosis_Claims
0                          584
1                          306
```

```
2                          127
3                          128
4                          541
…                          …
1348                        21
1349                        23
1350                         4
1351                       114
1352                       269

[1353 rows x 28 columns]
```

[110]:
```
## Here we are calculating the mean of values for some variables for each␣
↪unique provider.

Train_Data_Mean=round(Train_Patient_data.groupby(['Provider'], as_index =␣
↪False)[['NoOfMonths_PartACov', 'NoOfMonths_PartBCov',

                                                                        ␣
↪'Age']].mean())


Test_Data_Mean=round(Test_Patient_data.groupby(['Provider'], as_index =␣
↪False)[['NoOfMonths_PartACov', 'NoOfMonths_PartBCov',

                                                                        ␣
↪'Age']].mean())
```

[111]:
```
Train_Data_Mean
```

[111]:
```
      Provider  NoOfMonths_PartACov  NoOfMonths_PartBCov   Age
0     PRV51001                 12.0                 12.0  87.0
1     PRV51003                 12.0                 12.0  78.0
2     PRV51004                 12.0                 12.0  80.0
3     PRV51005                 12.0                 12.0  78.0
4     PRV51007                 12.0                 12.0  77.0
…        …                     …                    …     …
5405  PRV57759                 12.0                 12.0  82.0
5406  PRV57760                 12.0                 12.0  69.0
5407  PRV57761                 12.0                 12.0  80.0
5408  PRV57762                 12.0                 12.0  76.0
5409  PRV57763                 12.0                 12.0  82.0

[5410 rows x 4 columns]
```

[112]:
```
#### Now we merge Count,sum and mean dataframes with the main train dataframe
```

[113]:
```
## Merging of Train Datasets
Train_df=pd.merge(Train_Count,Train_Data_Sum,on='Provider',how='left').\
```

```
                merge(Train_Data_Mean,on='Provider',how='left').\
                merge(Train,on='Provider',how='left')

## Merging of Test Datasets

Test_df=pd.merge(Test_Count,Test_Data_Sum,on='Provider',how='left').\
                merge(Test_Data_Mean,on='Provider',how='left').\
                merge(Test,on='Provider',how='left')
```

[114]:
```
Train_df #Target column PotentialFraud is avaialble here
```

[114]:

	Provider	BeneID_count	ClaimID_count	InscClaimAmtReimbursed	\
0	PRV51001	24	25	104640	
1	PRV51003	117	132	605670	
2	PRV51004	138	149	52170	
3	PRV51005	495	1165	280910	
4	PRV51007	58	72	33710	
…	…	…	…	…	
5405	PRV57759	24	28	10640	
5406	PRV57760	9	22	4770	
5407	PRV57761	67	82	18470	
5408	PRV57762	1	1	1900	
5409	PRV57763	70	118	43610	

	DeductibleAmtPaid	RenalDiseaseIndicator	AttendingPhysician	\
0	5340.0	8	25	
1	66286.0	29	132	
2	310.0	23	149	
3	3700.0	259	1163	
4	3264.0	11	72	
…	…	…	…	
5405	130.0	5	28	
5406	0.0	0	22	
5407	370.0	23	82	
5408	0.0	0	1	
5409	390.0	25	118	

	OperatingPhysician	OtherPhysician	AdmitForDays	…	\
0	5	10	30.0	…	
1	45	25	382.0	…	
2	27	63	0.0	…	
3	222	478	0.0	…	
4	12	26	19.0	…	
…	…	…	…	…	
5405	1	12	0.0	…	
5406	6	9	0.0	…	
5407	14	36	0.0	…	

```
5408                   0               0          0.0  …
5409                  19              49          0.0  …

      OPAnnualDeductibleAmt  WhetherDead  N_Types_Physicians  IsDiagnosisCode  \
0                     11598          0.0                  40                5
1                     97300          1.0                 202               62
2                     92790          1.0                 239                0
3                    741323          4.0                1863                0
4                     33820          1.0                 110                3
…                     …          …                   …              …
5405                  24830          0.0                  41                0
5406                  17720          0.0                  37                0
5407                  58000          1.0                 132                0
5408                    400          0.0                   1                0
5409                  94790          0.0                 186                0

      N_Procedure  N_UniqueDiagnosis_Claims  NoOfMonths_PartACov  \
0               3                        91                 12.0
1              48                       761                 12.0
2               0                       410                 12.0
3               0                      3246                 12.0
4               1                       231                 12.0
…           …                         …                  …
5405            0                        61                 12.0
5406            0                        59                 12.0
5407            0                       235                 12.0
5408            0                         2                 12.0
5409            0                       315                 12.0

      NoOfMonths_PartBCov   Age  PotentialFraud
0                    12.0  87.0              No
1                    12.0  78.0             Yes
2                    12.0  80.0              No
3                    12.0  78.0             Yes
4                    12.0  77.0              No
…                   …   …             …
5405                 12.0  82.0              No
5406                 12.0  69.0              No
5407                 12.0  80.0              No
5408                 12.0  76.0              No
5409                 12.0  82.0              No

[5410 rows x 34 columns]
```

[115]:
```
Test_df #Target column PotentialFraud is not avaialble here
```

[115]:

	Provider	BeneID_count	ClaimID_count	InscClaimAmtReimbursed	\
0	PRV51002	169	205	53790	
1	PRV51006	81	102	30720	
2	PRV51009	30	39	27230	
3	PRV51010	25	38	64580	
4	PRV51018	146	190	61620	
…	…	…	…	…	
1348	PRV57713	10	11	860	
1349	PRV57726	8	8	1590	
1350	PRV57745	2	2	510	
1351	PRV57749	45	49	9980	
1352	PRV57750	94	105	27020	

	DeductibleAmtPaid	RenalDiseaseIndicator	AttendingPhysician	\
0	380.0	32	205	
1	0.0	10	102	
2	1238.0	12	38	
3	5340.0	5	38	
4	670.0	41	190	
…	…	…	…	
1348	0.0	2	11	
1349	0.0	0	8	
1350	0.0	1	2	
1351	370.0	9	49	
1352	230.0	17	105	

	OperatingPhysician	OtherPhysician	AdmitForDays	…	\
0	30	77	0.0	…	
1	24	38	0.0	…	
2	12	10	8.0	…	
3	9	10	29.0	…	
4	30	72	0.0	…	
…	…	…	…	…	
1348	0	6	0.0	…	
1349	0	3	0.0	…	
1350	0	2	0.0	…	
1351	8	21	0.0	…	
1352	23	35	0.0	…	

	OPAnnualReimbursementAmt	OPAnnualDeductibleAmt	WhetherDead	\
0	480740	138980	3.0	
1	244970	61800	0.0	
2	95200	25210	0.0	
3	67480	25230	0.0	
4	534460	156960	2.0	
…	…	…	…	
1348	15710	5510	0.0	

```
1349                    21210                   9010          0.0
1350                     7430                   1420          0.0
1351                    85060                  24780          2.0
1352                   259740                  64360          0.0

      N_Types_Physicians  IsDiagnosisCode  N_Procedure  \
0                    312                0            0
1                    164                0            0
2                     60                2            3
3                     57                6            3
4                    292                0            0
…                    …                  …            …
1348                  17                0            0
1349                  11                0            0
1350                   4                0            0
1351                  78                0            0
1352                 163                0            0

      N_UniqueDiagnosis_Claims  NoOfMonths_PartACov  NoOfMonths_PartBCov   Age
0                          584                 12.0                 12.0  80.0
1                          306                 12.0                 12.0  83.0
2                          127                 12.0                 12.0  78.0
3                          128                 12.0                 12.0  83.0
4                          541                 12.0                 12.0  81.0
…                          …                   …                    …     …
1348                        21                 12.0                 12.0  87.0
1349                        23                 12.0                 12.0  76.0
1350                         4                 12.0                 12.0  85.0
1351                       114                 12.0                 12.0  79.0
1352                       269                 12.0                 12.0  81.0

[1353 rows x 33 columns]
```

[116]:
```
Train_df.isnull().sum() ## No null value is present in this dataset
```

[116]:
```
Provider                        0
BeneID_count                    0
ClaimID_count                   0
InscClaimAmtReimbursed          0
DeductibleAmtPaid               0
RenalDiseaseIndicator           0
AttendingPhysician              0
OperatingPhysician              0
OtherPhysician                  0
AdmitForDays                    0
ChronicCond_Alzheimer           0
ChronicCond_Heartfailure        0
```

```
ChronicCond_Cancer                0
ChronicCond_KidneyDisease         0
ChronicCond_ObstrPulmonary        0
ChronicCond_Depression            0
ChronicCond_Diabetes              0
ChronicCond_IschemicHeart         0
ChronicCond_Osteoporasis          0
ChronicCond_rheumatoidarthritis   0
ChronicCond_stroke                0
IPAnnualReimbursementAmt          0
IPAnnualDeductibleAmt             0
OPAnnualReimbursementAmt          0
OPAnnualDeductibleAmt             0
WhetherDead                       0
N_Types_Physicians                0
IsDiagnosisCode                   0
N_Procedure                       0
N_UniqueDiagnosis_Claims          0
NoOfMonths_PartACov               0
NoOfMonths_PartBCov               0
Age                               0
PotentialFraud                    0
dtype: int64
```

[117]:
```
#In Train Dataset Target variable PotentialFraud has value in category i.e
 ↪"Yes" and "No" need to replace with 1 and 0.

Train_df['PotentialFraud']=np.where(Train_df.PotentialFraud == "Yes", 1, 0)
```

[118]:
```
Train_df
```

[118]:

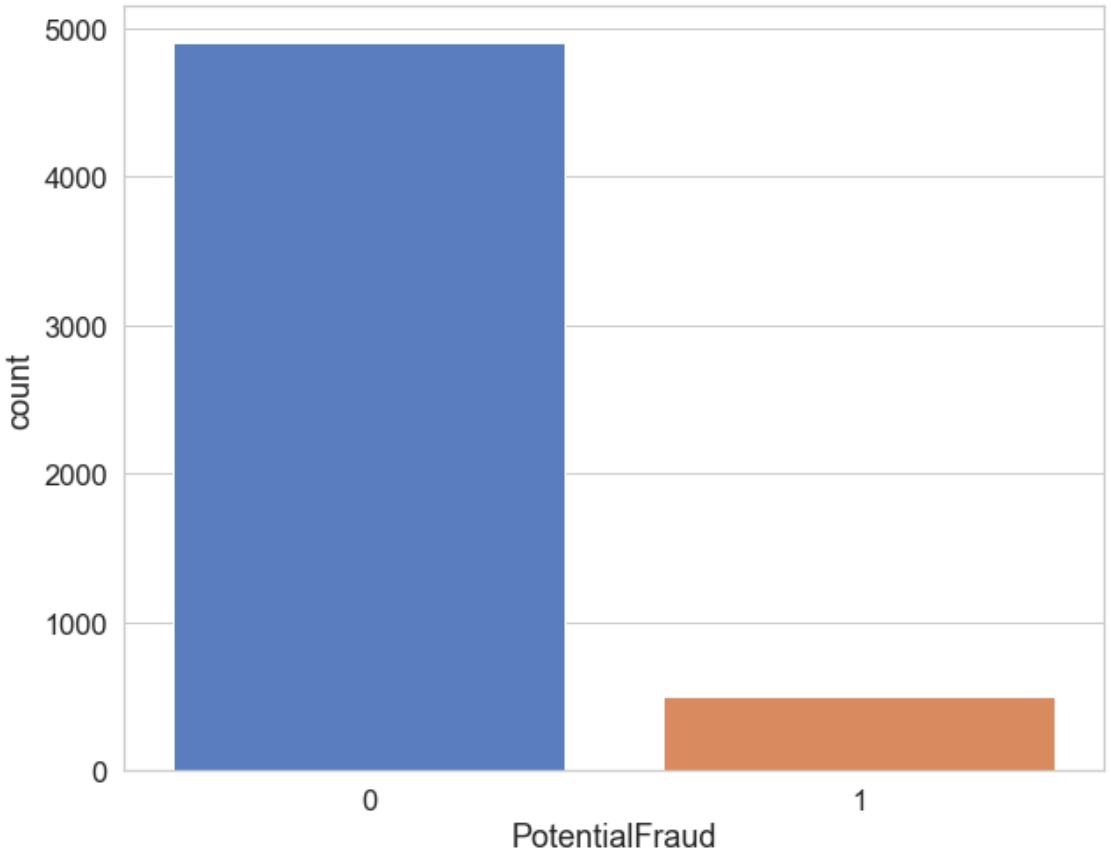
	Provider	BeneID_count	ClaimID_count	InscClaimAmtReimbursed	DeductibleAmtPaid	RenalDiseaseIndicator	AttendingPhysician
0	PRV51001	24	25	104640	5340.0	8	25
1	PRV51003	117	132	605670	66286.0	29	132
2	PRV51004	138	149	52170			
3	PRV51005	495	1165	280910			
4	PRV51007	58	72	33710			
…	…	…	…	…			
5405	PRV57759	24	28	10640			
5406	PRV57760	9	22	4770			
5407	PRV57761	67	82	18470			
5408	PRV57762	1	1	1900			
5409	PRV57763	70	118	43610			

```
2                310.0                    23                149
3               3700.0                   259               1163
4               3264.0                    11                 72
…                  …                      …                  …
5405             130.0                     5                 28
5406               0.0                     0                 22
5407             370.0                    23                 82
5408               0.0                     0                  1
5409             390.0                    25                118

      OperatingPhysician  OtherPhysician  AdmitForDays  …  \
0                      5              10          30.0  …
1                     45              25         382.0  …
2                     27              63           0.0  …
3                    222             478           0.0  …
4                     12              26          19.0  …
…                    …               …            …  …
5405                   1              12           0.0  …
5406                   6               9           0.0  …
5407                  14              36           0.0  …
5408                   0               0           0.0  …
5409                  19              49           0.0  …

      OPAnnualDeductibleAmt  WhetherDead  N_Types_Physicians  IsDiagnosisCode  \
0                     11598          0.0                  40                5
1                     97300          1.0                 202               62
2                     92790          1.0                 239                0
3                    741323          4.0                1863                0
4                     33820          1.0                 110                3
…                     …              …                   …                …
5405                  24830          0.0                  41                0
5406                  17720          0.0                  37                0
5407                  58000          1.0                 132                0
5408                    400          0.0                   1                0
5409                  94790          0.0                 186                0

      N_Procedure  N_UniqueDiagnosis_Claims  NoOfMonths_PartACov  \
0               3                        91                 12.0
1              48                       761                 12.0
2               0                       410                 12.0
3               0                      3246                 12.0
4               1                       231                 12.0
…             …                         …                   …
5405            0                        61                 12.0
5406            0                        59                 12.0
5407            0                       235                 12.0
5408            0                         2                 12.0
```

```
5409            0                      315                 12.0

      NoOfMonths_PartBCov   Age  PotentialFraud
0                    12.0  87.0               0
1                    12.0  78.0               1
2                    12.0  80.0               0
3                    12.0  78.0               1
4                    12.0  77.0               0
…                     …     …                 …
5405                 12.0  82.0               0
5406                 12.0  69.0               0
5407                 12.0  80.0               0
5408                 12.0  76.0               0
5409                 12.0  82.0               0

[5410 rows x 34 columns]
```

[119]:
```
# Here we can the count of Dependent variable values
plt.figure(figsize=(10,8))
sns.countplot(Train_df.PotentialFraud)
```

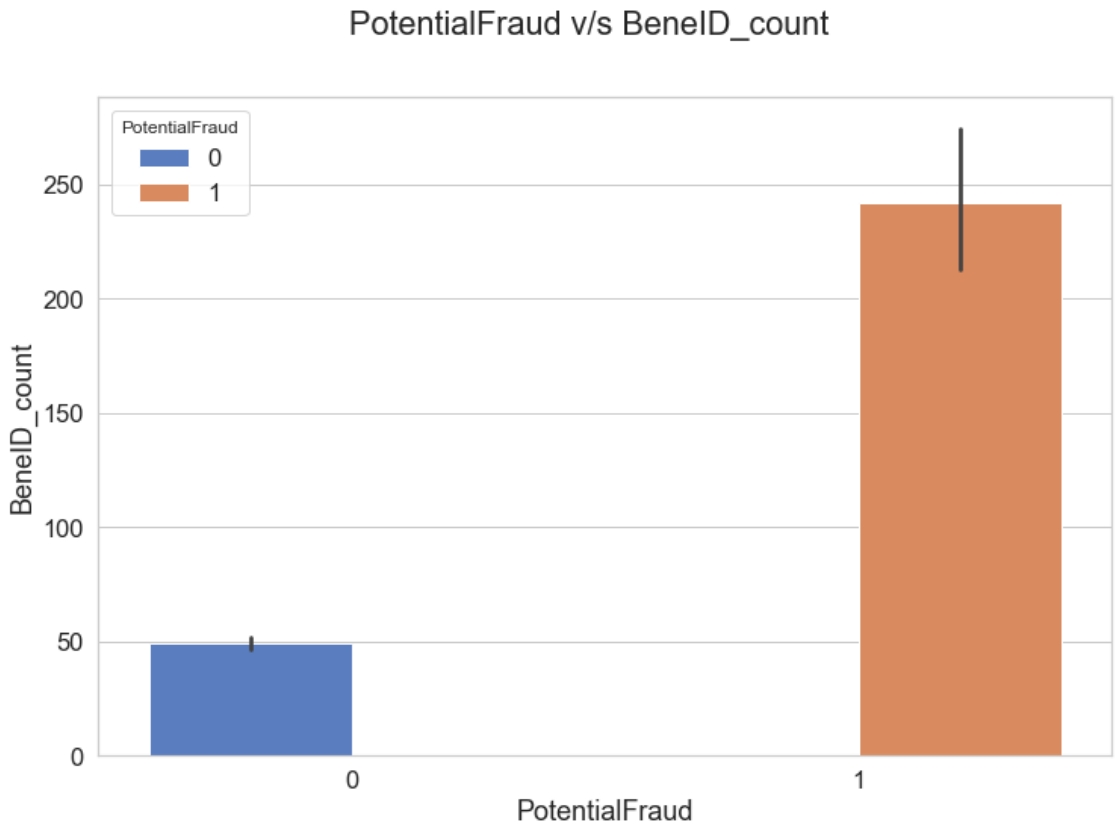
[119]: <AxesSubplot:xlabel='PotentialFraud', ylabel='count'>



0.1.1 Bivariant Data Analysis

[120]:
```
## Here we can se the barplot of PotentialFraud v/s BeneID_Count and here bar␣
 ↪shows mean of BeneID_Count for Potential Fraud value 1 and 0
## From this barplot we can conclude that there is a Potential Fraud when the␣
 ↪BeneID_Count is more as its mean is more as shown.

plt.figure(figsize=(12,8))
sns.barplot(Train_df["PotentialFraud"],Train_df["BeneID_count"],␣
 ↪hue=Train_df["PotentialFraud"])
plt.suptitle('PotentialFraud v/s BeneID_count')
plt.xlabel('PotentialFraud')
plt.ylabel('BeneID_count')
```

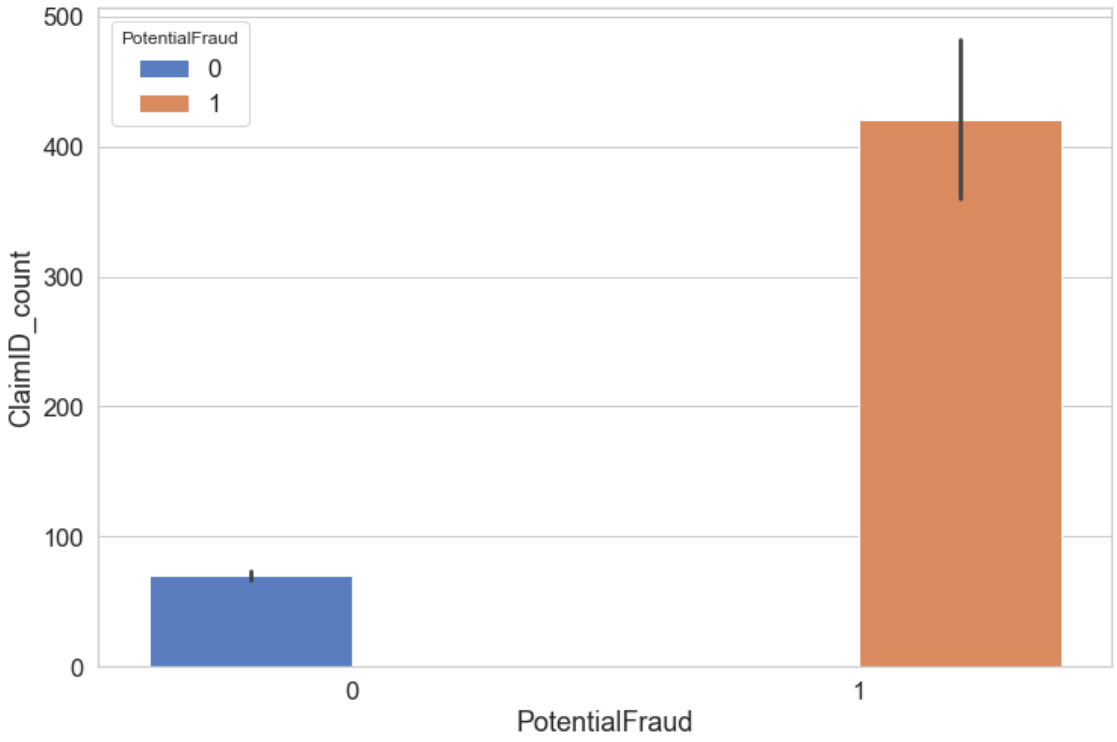
[120]:
```
Text(0, 0.5, 'BeneID_count')
```



As we can see Fraudulant claims have higher number of Beneficiary ID as they tend to commit fraud with multiple beneficiary id.

[121]:
```
## Here we can se the barplot of PotentialFraud v/s ClaimID_Count and here bar␣
↪shows mean of ClaimID_Count for Potential Fraud value 1 and 0
## From this barplot we can conclude that there is a Potential Fraud when the␣
↪ClaimID_Count is more as its mean is more as shown.

plt.figure(figsize=(12,8))
sns.barplot(Train_df["PotentialFraud"],Train_df["ClaimID_count"],␣
↪hue=Train_df["PotentialFraud"])
```

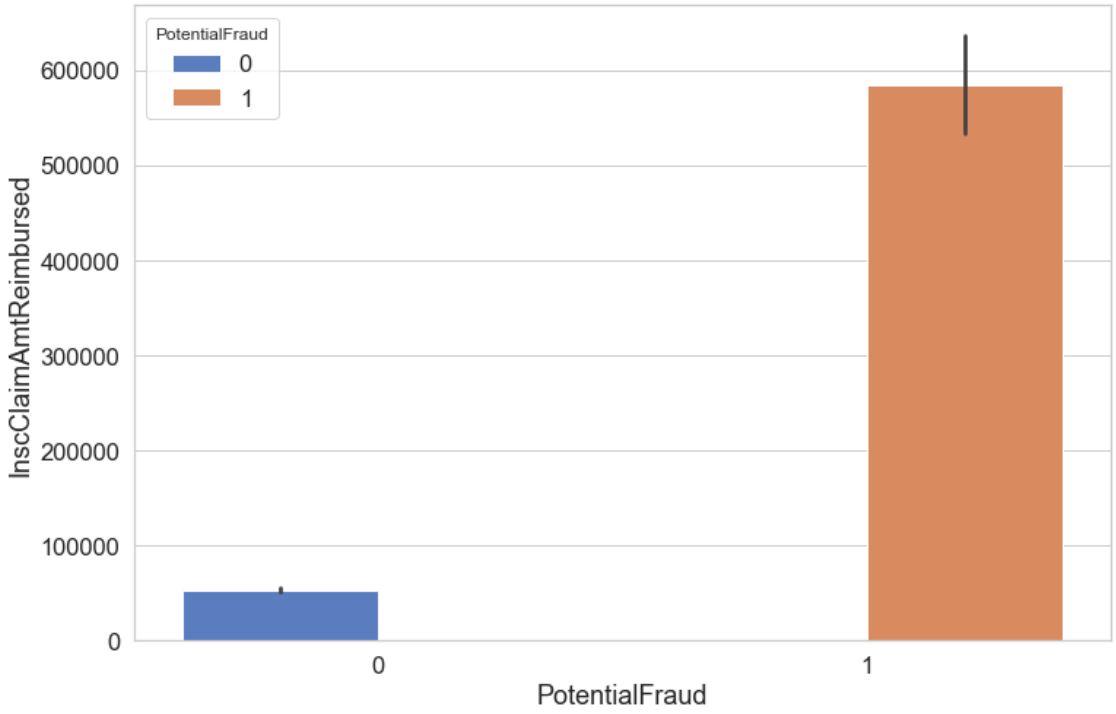
[121]:
```
<AxesSubplot:xlabel='PotentialFraud', ylabel='ClaimID_count'>
```



Same as the above observation, potential fraud claims tend to have higher number of Claim ID.

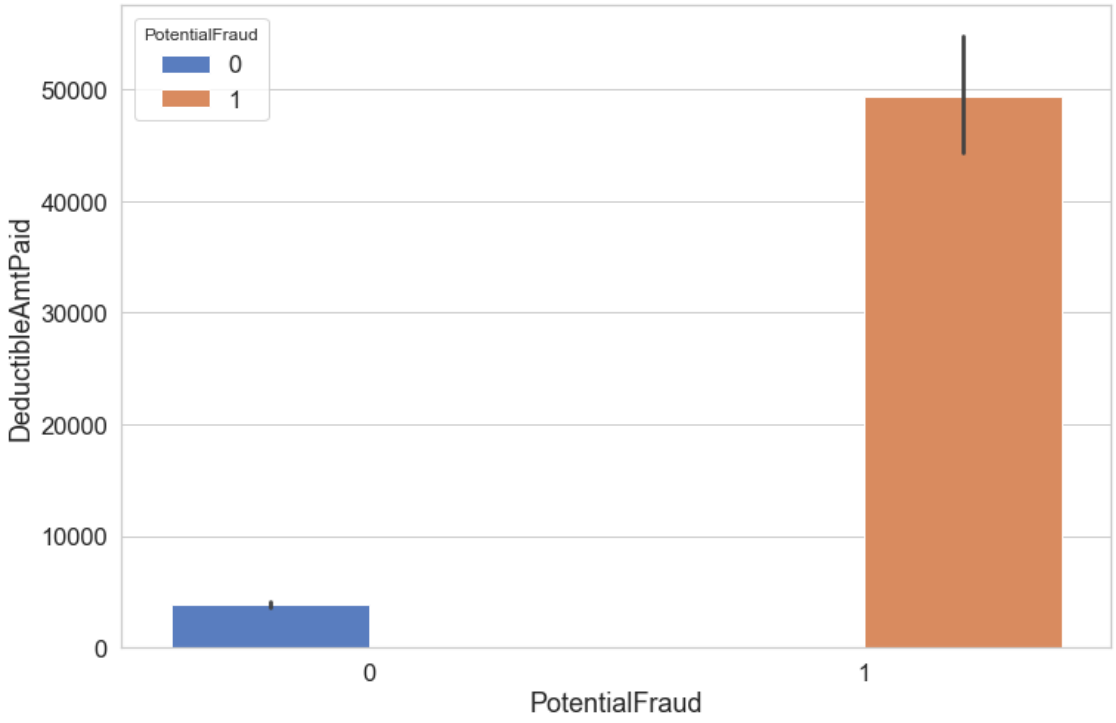
[122]:
```
## Here we can se the barplot of PotentialFraud v/s InscClaimAmtReimbursed and␣
↪here bar shows mean of InscClaimAmtReimbursed for Potential Fraud value 1␣
↪and 0
## From this barplot we can conclude that there is a Potential Fraud when the␣
↪InscClaimAmtReimbursed is more as its mean is more as shown.

plt.figure(figsize=(12,8))

sns.barplot(Train_df["PotentialFraud"],Train_df["InscClaimAmtReimbursed"],␣
↪hue=Train_df["PotentialFraud"])
```

[122]: <AxesSubplot:xlabel='PotentialFraud', ylabel='InscClaimAmtReimbursed'>



[123]:
```
plt.figure(figsize=(12,8))

sns.barplot(Train_df["PotentialFraud"],Train_df["DeductibleAmtPaid"],␣
↪hue=Train_df["PotentialFraud"])
```

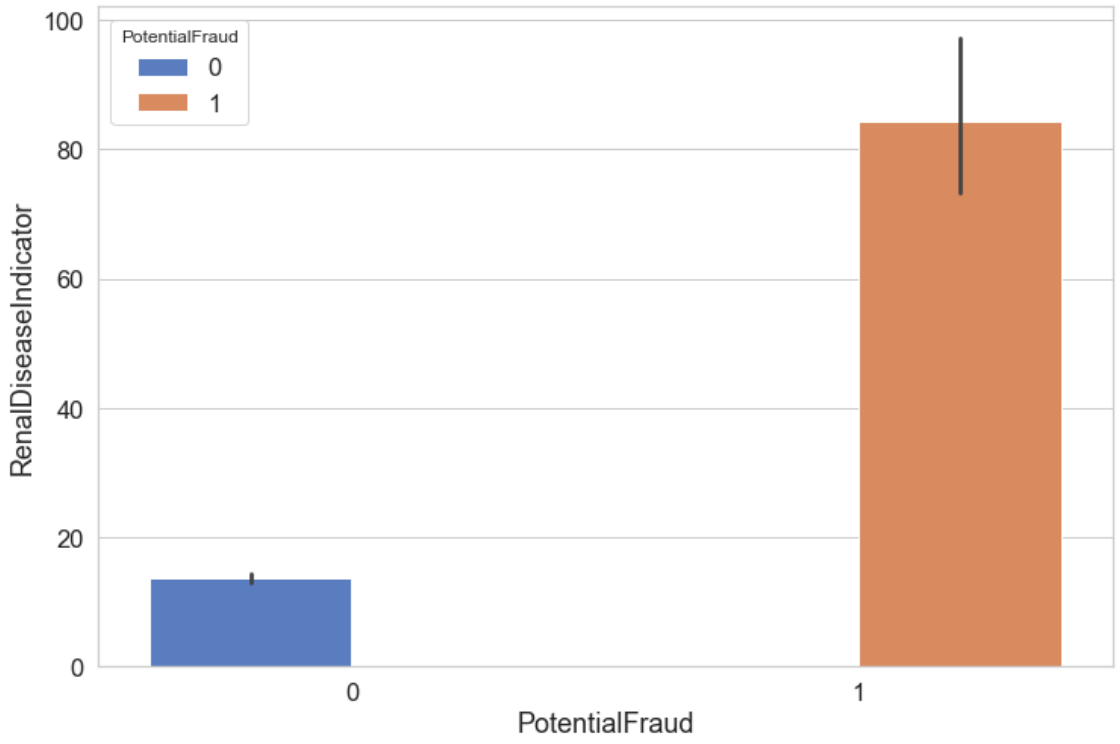
[123]: <AxesSubplot:xlabel='PotentialFraud', ylabel='DeductibleAmtPaid'>



As we have observed both in InscClaimAmtReimbursed and DeductibleAmtPaid are way higher than the legitimate claims.

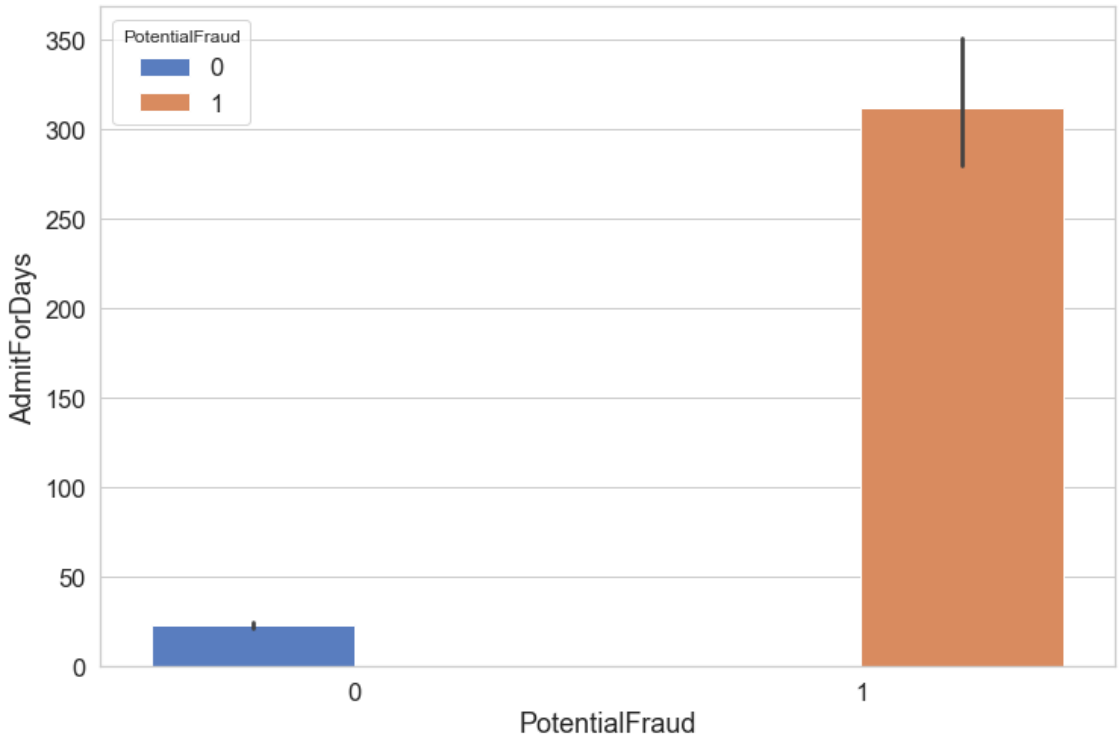
[124]:
```
plt.figure(figsize=(12,8))
sns.barplot(Train_df["PotentialFraud"],Train_df["RenalDiseaseIndicator"],␣
↪hue=Train_df["PotentialFraud"])
```

[124]:
```
<AxesSubplot:xlabel='PotentialFraud', ylabel='RenalDiseaseIndicator'>
```



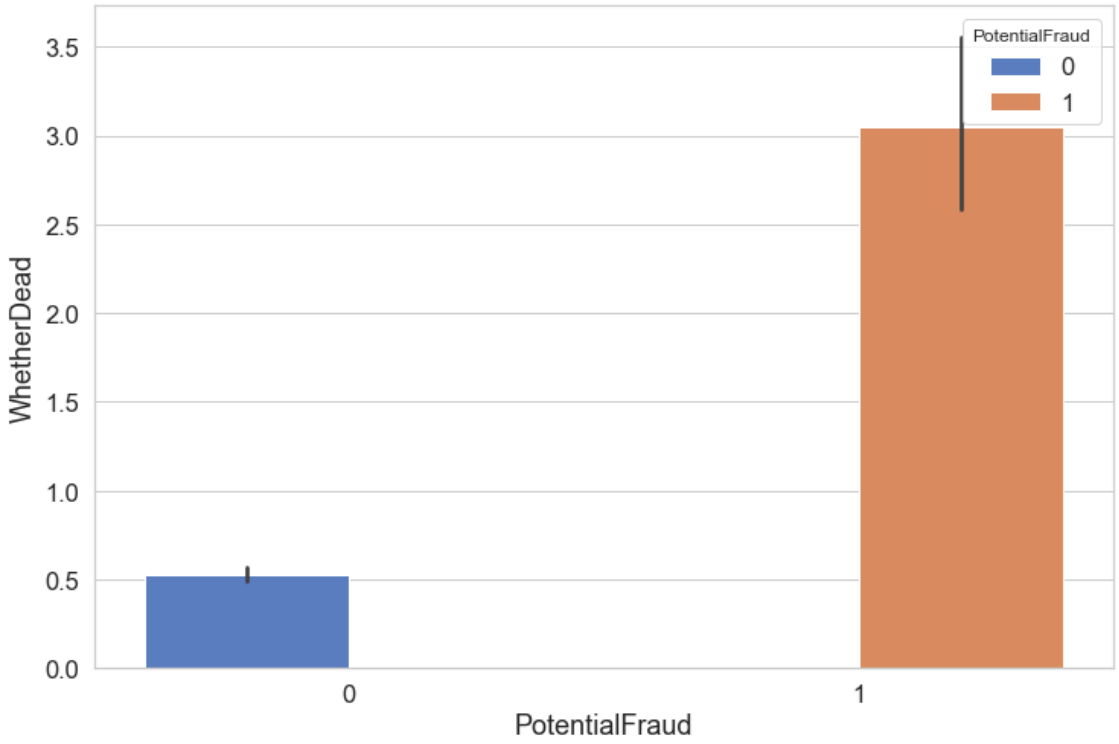
[125]:
```
plt.figure(figsize=(12,8))
sns.barplot(Train_df["PotentialFraud"],Train_df["AdmitForDays"],␣
 ↪hue=Train_df["PotentialFraud"])
```

[125]: <AxesSubplot:xlabel='PotentialFraud', ylabel='AdmitForDays'>



[126]:
```
plt.figure(figsize=(12,8))
sns.barplot(Train_df["PotentialFraud"],Train_df["WhetherDead"],␣
↪hue=Train_df["PotentialFraud"])
```

[126]: <AxesSubplot:xlabel='PotentialFraud', ylabel='WhetherDead'>

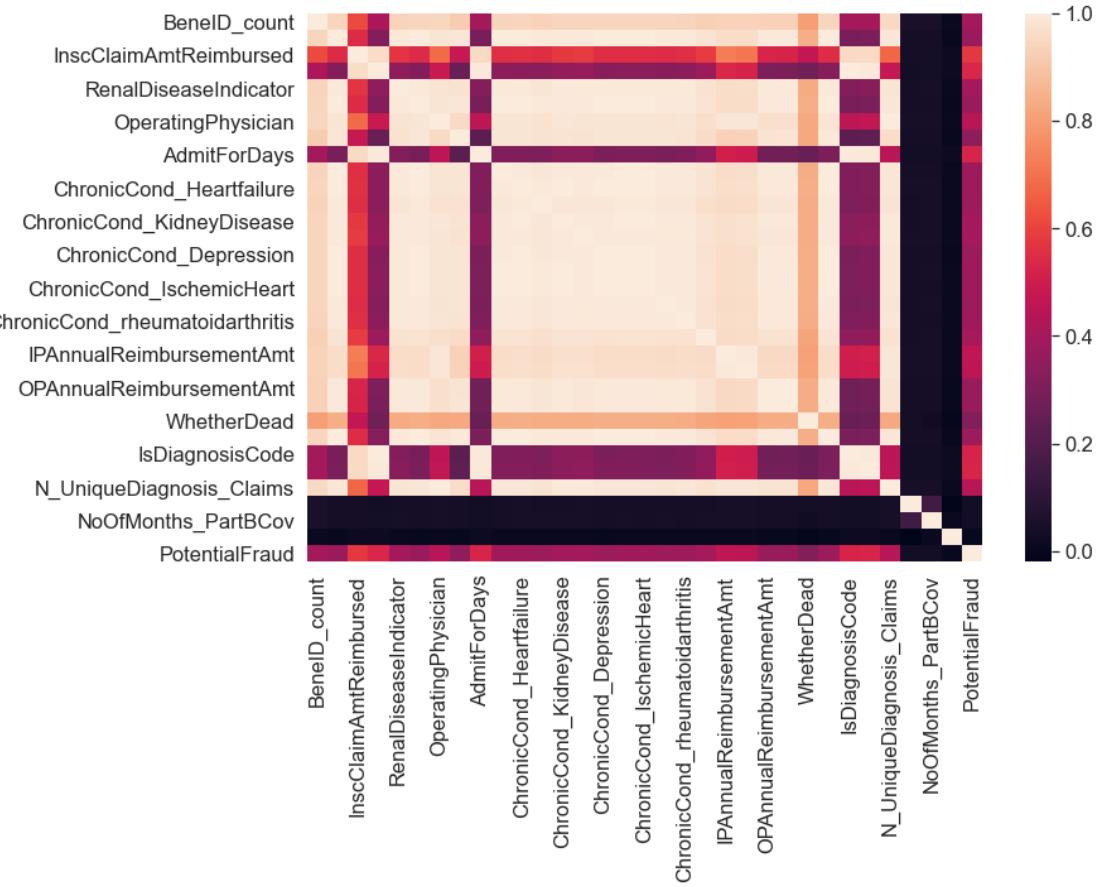


In category 0, the bar is between 0 and 1 because there are some people who are dead and some are alive, but in category 1 the bar has gone above 3 that means fraudulant claims are more likely to happen where people are dead.

0.1.2 Correlation Matrix

[127]:
```
plt.figure(figsize=(12,8))
Train_corr=Train_df.corr()
sns.heatmap(Train_corr)
```

[127]: <AxesSubplot:>



[128]:
```
Train_corr=Train_df.corr()
Train_corr['PotentialFraud']
```

[128]:
```
BeneID_count                    0.393531
ClaimID_count                   0.374197
InscClaimAmtReimbursed          0.575558
DeductibleAmtPaid               0.532070
RenalDiseaseIndicator           0.391002
AttendingPhysician              0.374253
OperatingPhysician              0.445086
OtherPhysician                  0.342673
AdmitForDays                    0.526338
ChronicCond_Alzheimer           0.380344
ChronicCond_Heartfailure        0.384131
ChronicCond_Cancer              0.376945
ChronicCond_KidneyDisease       0.394239
ChronicCond_ObstrPulmonary      0.396191
ChronicCond_Depression          0.377411
ChronicCond_Diabetes            0.378881
```

```
ChronicCond_IschemicHeart          0.380093
ChronicCond_Osteoporasis           0.378274
ChronicCond_rheumatoidarthritis    0.380161
ChronicCond_stroke                 0.399206
IPAnnualReimbursementAmt           0.461978
IPAnnualDeductibleAmt              0.454921
OPAnnualReimbursementAmt           0.368562
OPAnnualDeductibleAmt              0.368196
WhetherDead                        0.317546
N_Types_Physicians                 0.377036
IsDiagnosisCode                    0.525393
N_Procedure                        0.527376
N_UniqueDiagnosis_Claims           0.441150
NoOfMonths_PartACov                0.029799
NoOfMonths_PartBCov                0.030174
Age                                0.001096
PotentialFraud                     1.000000
Name: PotentialFraud, dtype: float64
```

So from here we can see that Age, NoOfMonths_PartBCov and NoOfMonths_PartACov are not making any pattern/relationship with dependent variable 'PotentialFraud', hence we will not consider these variables in our model

We will make a final dataset on which we will do modelling,In this dataset we keep only those variable which we will use in our machine learning modelling algorithms. So from our Train_df dataset we will remove all ID type variables like Provider,BeneID_count and ClaimID_count and also remove those variable which are not making any pattern with the dependent variable this we can see correlation matrix that is shown above

[129]:
```
df_clf=Train_df.iloc[:,3:]
df_clf
```

[129]:

	InscClaimAmtReimbursed	DeductibleAmtPaid	RenalDiseaseIndicator	\
0	104640	5340.0	8	
1	605670	66286.0	29	
2	52170	310.0	23	
3	280910	3700.0	259	
4	33710	3264.0	11	
…	…	…	…	
5405	10640	130.0	5	
5406	4770	0.0	0	
5407	18470	370.0	23	
5408	1900	0.0	0	
5409	43610	390.0	25	

	AttendingPhysician	OperatingPhysician	OtherPhysician	AdmitForDays	\
0	25	5	10	30.0	
1	132	45	25	382.0	

```
2                    149                   27              63           0.0
3                   1163                  222             478           0.0
4                     72                   12              26          19.0
…                      …                    …               …             …
5405                  28                    1              12           0.0
5406                  22                    6               9           0.0
5407                  82                   14              36           0.0
5408                   1                    0               0           0.0
5409                 118                   19              49           0.0

      ChronicCond_Alzheimer  ChronicCond_Heartfailure  ChronicCond_Cancer  \
0                        15                        19                   5
1                        56                        80                  10
2                        64                        88                  16
3                       426                       680                 165
4                        26                        40                  12
…                         …                         …                   …
5405                     14                        20                   4
5406                      3                        11                   0
5407                     36                        56                  14
5408                      0                         0                   1
5409                     45                        63                  24

      …  OPAnnualDeductibleAmt  WhetherDead  N_Types_Physicians  \
0     …                  11598          0.0                  40
1     …                  97300          1.0                 202
2     …                  92790          1.0                 239
3     …                 741323          4.0                1863
4     …                  33820          1.0                 110
…   …                        …            …                   …
5405  …                  24830          0.0                  41
5406  …                  17720          0.0                  37
5407  …                  58000          1.0                 132
5408  …                    400          0.0                   1
5409  …                  94790          0.0                 186

      IsDiagnosisCode  N_Procedure  N_UniqueDiagnosis_Claims  \
0                   5            3                        91
1                  62           48                       761
2                   0            0                       410
3                   0            0                      3246
4                   3            1                       231
…                   …            …                         …
5405                0            0                        61
5406                0            0                        59
5407                0            0                       235
5408                0            0                         2
```

```
5409                0            0                      315

      NoOfMonths_PartACov  NoOfMonths_PartBCov   Age  PotentialFraud
0                    12.0                 12.0  87.0               0
1                    12.0                 12.0  78.0               1
2                    12.0                 12.0  80.0               0
3                    12.0                 12.0  78.0               1
4                    12.0                 12.0  77.0               0
…                     …                    …    …                  …
5405                 12.0                 12.0  82.0               0
5406                 12.0                 12.0  69.0               0
5407                 12.0                 12.0  80.0               0
5408                 12.0                 12.0  76.0               0
5409                 12.0                 12.0  82.0               0

[5410 rows x 31 columns]
```

[130]:
```
df_clf.
↪drop(['NoOfMonths_PartACov','NoOfMonths_PartBCov','Age'],axis=1,inplace=True)
```

0.1.3 Final Train Dataset on which we trained our model

[131]:
```
df_clf #This is final Trained Dataset
```

[131]:
```
      InscClaimAmtReimbursed  DeductibleAmtPaid  RenalDiseaseIndicator  \
0                     104640             5340.0                      8
1                     605670            66286.0                     29
2                      52170              310.0                     23
3                     280910             3700.0                    259
4                      33710             3264.0                     11
…                       …                  …                      …
5405                   10640              130.0                      5
5406                    4770                0.0                      0
5407                   18470              370.0                     23
5408                    1900                0.0                      0
5409                   43610              390.0                     25

      AttendingPhysician  OperatingPhysician  OtherPhysician  AdmitForDays  \
0                     25                   5              10          30.0
1                    132                  45              25         382.0
2                    149                  27              63           0.0
3                   1163                 222             478           0.0
4                     72                  12              26          19.0
…                      …                   …               …           …
5405                  28                   1              12           0.0
5406                  22                   6               9           0.0
5407                  82                  14              36           0.0
```

```
5408                   1                    0              0           0.0
5409                 118                   19             49           0.0

      ChronicCond_Alzheimer  ChronicCond_Heartfailure  ChronicCond_Cancer  \
0                        15                        19                   5
1                        56                        80                  10
2                        64                        88                  16
3                       426                       680                 165
4                        26                        40                  12
…                       …                         …                   …
5405                     14                        20                   4
5406                      3                        11                   0
5407                     36                        56                  14
5408                      0                         0                   1
5409                     45                        63                  24

      …  IPAnnualReimbursementAmt  IPAnnualDeductibleAmt  \
0     …                    440150                  22428
1     …                    999000                 122948
2     …                    648430                  64808
3     …                   4221950                 441724
4     …                    219600                  32040
…   …                       …                      …
5405  …                    110940                  12816
5406  …                     61280                   9612
5407  …                    576180                  48060
5408  …                     15000                   1068
5409  …                    424710                  71148

      OPAnnualReimbursementAmt  OPAnnualDeductibleAmt  WhetherDead  \
0                        65380                  11598          0.0
1                       353520                  97300          1.0
2                       327040                  92790          1.0
3                      2457840                 741323          4.0
4                       124540                  33820          1.0
…                       …                      …          …
5405                     90770                  24830          0.0
5406                     32840                  17720          0.0
5407                    240130                  58000          1.0
5408                      2540                    400          0.0
5409                    366850                  94790          0.0

      N_Types_Physicians  IsDiagnosisCode  N_Procedure  \
0                     40                5            3
1                    202               62           48
2                    239                0            0
3                   1863                0            0
```

```
4                    110                3            1
…                    …                  …            …
5405                  41                0            0
5406                  37                0            0
5407                 132                0            0
5408                   1                0            0
5409                 186                0            0

      N_UniqueDiagnosis_Claims  PotentialFraud
0                           91               0
1                          761               1
2                          410               0
3                         3246               1
4                          231               0
…                          …                 …
5405                        61               0
5406                        59               0
5407                       235               0
5408                         2               0
5409                       315               0

[5410 rows x 28 columns]
```

0.1.4 Final Test Dataset on which we will do final Prediction

[132]:
```
Test_df
```

[132]:
```
      Provider  BeneID_count  ClaimID_count  InscClaimAmtReimbursed  \
0     PRV51002           169            205                   53790
1     PRV51006            81            102                   30720
2     PRV51009            30             39                   27230
3     PRV51010            25             38                   64580
4     PRV51018           146            190                   61620
…        …          …             …                      …
1348  PRV57713            10             11                     860
1349  PRV57726             8              8                    1590
1350  PRV57745             2              2                     510
1351  PRV57749            45             49                    9980
1352  PRV57750            94            105                   27020

      DeductibleAmtPaid  RenalDiseaseIndicator  AttendingPhysician  \
0                 380.0                     32                 205
1                   0.0                     10                 102
2                1238.0                     12                  38
3                5340.0                      5                  38
4                 670.0                     41                 190
…                 …                         …                  …
```

```
1348               0.0                     2                  11
1349               0.0                     0                   8
1350               0.0                     1                   2
1351             370.0                     9                  49
1352             230.0                    17                 105

      OperatingPhysician  OtherPhysician  AdmitForDays  …  \
0                     30              77           0.0  …
1                     24              38           0.0  …
2                     12              10           8.0  …
3                      9              10          29.0  …
4                     30              72           0.0  …
…                    …               …            …  …
1348                   0               6           0.0  …
1349                   0               3           0.0  …
1350                   0               2           0.0  …
1351                   8              21           0.0  …
1352                  23              35           0.0  …

      OPAnnualReimbursementAmt  OPAnnualDeductibleAmt  WhetherDead  \
0                       480740                 138980          3.0
1                       244970                  61800          0.0
2                        95200                  25210          0.0
3                        67480                  25230          0.0
4                       534460                 156960          2.0
…                          …                      …            …
1348                    15710                   5510          0.0
1349                    21210                   9010          0.0
1350                     7430                   1420          0.0
1351                    85060                  24780          2.0
1352                   259740                  64360          0.0

      N_Types_Physicians  IsDiagnosisCode  N_Procedure  \
0                    312                0            0
1                    164                0            0
2                     60                2            3
3                     57                6            3
4                    292                0            0
…                    …                  …            …
1348                  17                0            0
1349                  11                0            0
1350                   4                0            0
1351                  78                0            0
1352                 163                0            0

      N_UniqueDiagnosis_Claims  NoOfMonths_PartACov  NoOfMonths_PartBCov   Age
0                          584                 12.0                 12.0  80.0
```

```
1                          306                 12.0                  12.0  83.0
2                          127                 12.0                  12.0  78.0
3                          128                 12.0                  12.0  83.0
4                          541                 12.0                  12.0  81.0
…                          …                   …                     …     …
1348                        21                 12.0                  12.0  87.0
1349                        23                 12.0                  12.0  76.0
1350                         4                 12.0                  12.0  85.0
1351                       114                 12.0                  12.0  79.0
1352                       269                 12.0                  12.0  81.0

[1353 rows x 33 columns]
```

[133]:
```
def test(test_data):
    test_data=test_data.iloc[:,3:]
    test_data=test_data.
 ↪drop(['NoOfMonths_PartACov','NoOfMonths_PartBCov','Age'],axis=1)
    return test_data
```

[134]:
```
Test_data=test(Test_df)
Test_data ## In this target varaible is not there we need to predict this after␣
 ↪we trained our model
```

[134]:
```
      InscClaimAmtReimbursed  DeductibleAmtPaid  RenalDiseaseIndicator  \
0                      53790              380.0                     32
1                      30720                0.0                     10
2                      27230             1238.0                     12
3                      64580             5340.0                      5
4                      61620              670.0                     41
…                      …                  …                         …
1348                     860                0.0                      2
1349                    1590                0.0                      0
1350                     510                0.0                      1
1351                    9980              370.0                      9
1352                   27020              230.0                     17

      AttendingPhysician  OperatingPhysician  OtherPhysician  AdmitForDays  \
0                    205                  30              77           0.0
1                    102                  24              38           0.0
2                     38                  12              10           8.0
3                     38                   9              10          29.0
4                    190                  30              72           0.0
…                    …                    …               …            …
1348                  11                   0               6           0.0
1349                   8                   0               3           0.0
1350                   2                   0               2           0.0
1351                  49                   8              21           0.0
```

```
1352                105                  23              35           0.0

      ChronicCond_Alzheimer  ChronicCond_Heartfailure  ChronicCond_Cancer  \
0                        79                       108                  25
1                        35                        69                  15
2                         8                        17                   1
3                        21                        23                   7
4                        73                       109                  25
…                       …                         …                   …
1348                      6                         7                   1
1349                      4                         5                   1
1350                      1                         2                   1
1351                     18                        22                   6
1352                     39                        62                  13

      …  ChronicCond_stroke  IPAnnualReimbursementAmt  \
0     …                   19                   1062090
1     …                    8                    384290
2     …                    3                    117160
3     …                    5                    200200
4     …                   13                    900400
…   …                     …                         …
1348  …                    0                      5000
1349  …                    0                    105000
1350  …                    0                     20060
1351  …                    4                    172930
1352  …                    9                    355890

      IPAnnualDeductibleAmt  OPAnnualReimbursementAmt  OPAnnualDeductibleAmt  \
0                    112392                    480740                 138980
1                     48924                    244970                  61800
2                      9612                     95200                  25210
3                     20292                     67480                  25230
4                    101460                    534460                 156960
…                       …                         …                      …
1348                  1068                     15710                   5510
1349                  2136                     21210                   9010
1350                  2136                      7430                   1420
1351                 17088                     85060                  24780
1352                 39516                    259740                  64360

      WhetherDead  N_Types_Physicians  IsDiagnosisCode  N_Procedure  \
0             3.0                 312                0            0
1             0.0                 164                0            0
2             0.0                  60                2            3
3             0.0                  57                6            3
4             2.0                 292                0            0
```

```
…          …                  …               …          …
1348         0.0                17              0          0
1349         0.0                11              0          0
1350         0.0                 4              0          0
1351         2.0                78              0          0
1352         0.0               163              0          0

      N_UniqueDiagnosis_Claims
0                          584
1                          306
2                          127
3                          128
4                          541
…                          …
1348                        21
1349                        23
1350                         4
1351                       114
1352                       269

[1353 rows x 27 columns]
```

0.1.5 Working on our Train Dataset

[135]:
```
#Split the dataset into Independent and Dependent Features

x=df_clf.drop("PotentialFraud",axis=1)
y=df_clf.PotentialFraud
```

[136]:
```
print("Independent Variable shape:",x.shape)
print("Dependent Variable shape:",y.shape)
```

```
Independent Variable shape: (5410, 27)
Dependent Variable shape: (5410,)
```

[137]:
```
### Train Test Split

x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.
 ↪7,random_state=42)



print("Independent variables train:",x_train.shape)
print("Target variable train:",y_train.shape)
print("Independent variables test:",x_test.shape)
print("Target variables test:",y_test.shape)
```

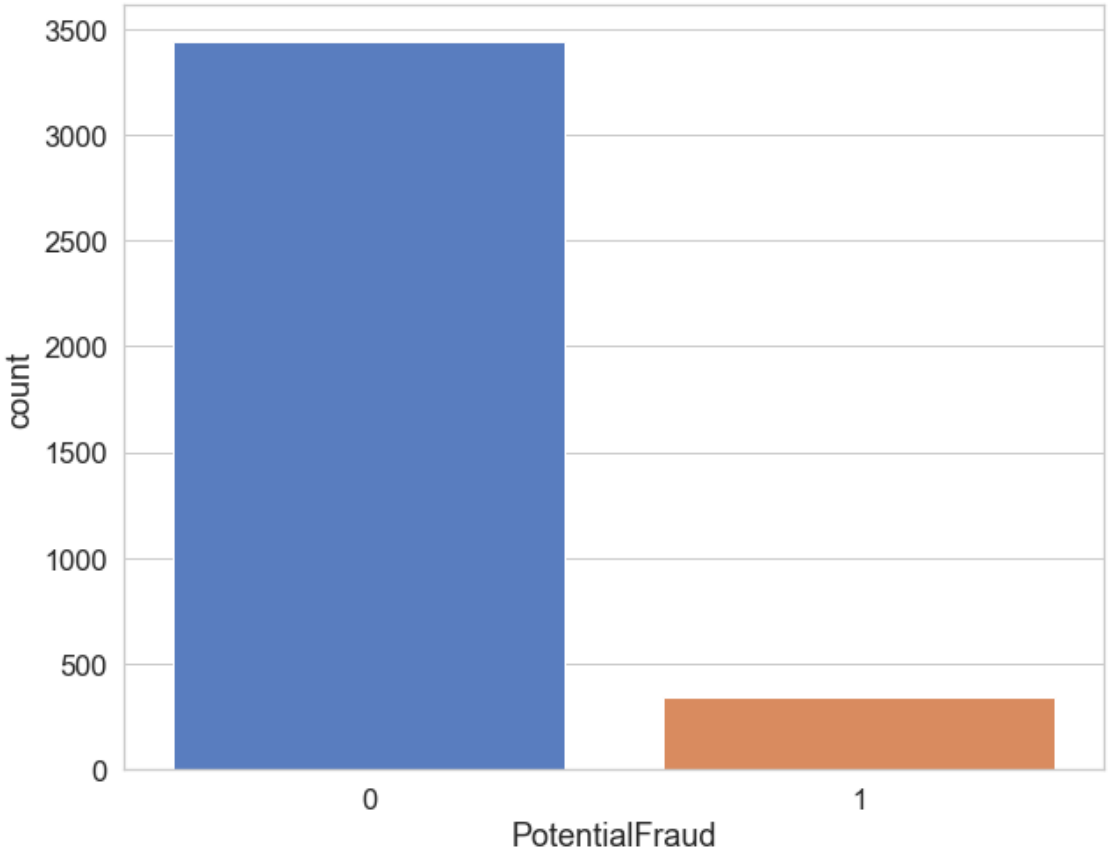
```
Independent variables train: (3786, 27)
Target variable train: (3786,)
```

```
Independent variables test: (1624, 27)
Target variables test: (1624,)
```

0.2 Modeling

[138]:
```
#Here we can see that our target vairable is imbalanced as "0" class is is majority and "1" class is in minority
plt.figure(figsize=(10,8))
sns.countplot(y_train)
```

[138]: <AxesSubplot:xlabel='PotentialFraud', ylabel='count'>



[139]:
```
from sklearn.ensemble import RandomForestClassifier


clf=RandomForestClassifier()
clf_fit=clf.fit(x_train,y_train)

y_pred_rf=clf_fit.predict(x_test)
```

[140]:
```
print('\033[1m'+"Confusion Matrix␣
 ↪\n"+'\033[0m',confusion_matrix(y_test,y_pred_rf))
print('\033[1m'+"\n Accuracy Score␣
 ↪\n"+'\033[0m',accuracy_score(y_test,y_pred_rf))
print('\033[1m'+"\n Classification Report␣
 ↪\n"+'\033[0m',classification_report(y_test,y_pred_rf))
```

Confusion Matrix

```
[[1444   21]
 [  91   68]]
```

Accuracy Score

```
0.9310344827586207
```

Classification Report

```
              precision    recall  f1-score   support

           0       0.94      0.99      0.96      1465
           1       0.76      0.43      0.55       159

    accuracy                           0.93      1624
   macro avg       0.85      0.71      0.76      1624
weighted avg       0.92      0.93      0.92      1624
```

[141]:
```
from sklearn.svm import SVC


clf_svc=SVC()
clf_svc_fit=clf_svc.fit(x_train,y_train)

y_pred_svc=clf_svc_fit.predict(x_test)
```

[142]:
```
print('\033[1m'+"Confusion Matrix␣
 ↪\n"+'\033[0m',confusion_matrix(y_test,y_pred_svc))
print('\033[1m'+"\n Accuracy Score␣
 ↪\n"+'\033[0m',accuracy_score(y_test,y_pred_svc))
print('\033[1m'+"\n Classification Report␣
 ↪\n"+'\033[0m',classification_report(y_test,y_pred_svc))
```

Confusion Matrix

```
[[1454   11]
 [ 103   56]]
```

Accuracy Score

0.9298029556650246

Classification Report

	precision	recall	f1-score	support
0	0.93	0.99	0.96	1465
1	0.84	0.35	0.50	159
accuracy			0.93	1624
macro avg	0.88	0.67	0.73	1624
weighted avg	0.92	0.93	0.92	1624