FRAUD DETECTION IN HEALTHCARE FOR MITIGATING MEDICARE SPENDING

Amila Viraj Mahinda Hewa Lunuwilage, Lohith Prasanna Teja Kakumanu, Monika Jangam Prabhudev

Background of the Problem:

The issue of provider fraud in Medicare is a serious problem because it has a significant impact on healthcare spending, increases insurance costs, and can affect the accessibility of quality healthcare services for people. The goal of the project is to tackle this problem by developing a system that can predict which healthcare providers might be engaging in fraudulent activities. Additionally, the project aims to understand the key factors that contribute to fraudulent behavior among providers and analyze patterns to better anticipate and prevent future fraudulent activities. Ultimately, the objective is to create a more secure and cost-effective healthcare system.

Problem Statement and significance:

This project's objective is to "predict the potentially fraudulent providers" based on the claims that these providers have submitted. In addition, we shall find significant factors that are useful in identifying the actions of possible fraud suppliers. To comprehend the future actions of providers, we will also examine patterns of deception in their claims.

The significance of this issue has serious consequences in two main ways. First, it makes healthcare more expensive for everyone by putting financial strain on the government and insurance companies. Second, it damages the trust in our healthcare system because resources are being misused for fraudulent activities instead of helping patients. To ensure that Medicare works fairly and effectively, we need to tackle and reduce fraud among healthcare providers.

Potential of the problem:

The potential Fraud in Medicare related is critical as it impacts healthcare spending, increases insurance costs, and can affect the accessibility of quality healthcare services. In order to solve this problem, we came up with a solution to understand the data from multiple insurance providers and healthcare providers where a particular beneficiary has undergone through a particular treatment including how many days he was admitted what all diseases were diagnosed and lot more parameters to build a predictive model to identify whether the transaction made by the beneficiary in this sector is Fraudulent or not.

Objective:

The primary objective of this study is to develop an advanced predictive model for identifying and mitigating provider fraud within the Medicare system. Additionally, we aim to pinpoint key patterns and indicators in Medicare claims data to enhance fraud detection capabilities and contribute to the overall integrity and sustainability of healthcare insurance systems.

Data:

We have picked the data from Kaggle.
Source:https://www.kaggle.com/datasets/rohitrox/healthcare-provider-fraud-detection-analysis/data
We're focusing on three main types of healthcare data for this project:

A) **Inpatient Data:**

This data gives us insights into the claims made for patients who are admitted to hospitals.
It includes additional details such as when they were admitted and discharged, as well as the diagnosis code for their admission.

B) **Outpatient Data:**

This data provides information about the claims made for patients who visit hospitals but are not admitted.

C) **Beneficiary Details Data:**

This data contains personal details about the beneficiaries, like their health conditions and the region they belong to.

Essentially, we're looking at the information related to patients admitted to hospitals, those who visit without admission, and the background details of the beneficiaries.

This dataset comprises of

```
Number of rows in beneficiary_df_train = 138556
Number of columns in beneficiary_df_train = 25
Number of rows in inpatient_df_train = 40474
Number of columns in inpatient_df_train = 30
Number of rows in outpatient_df_train = 517737
Number of columns in outpatient_df_train = 27
Number of rows in labels_df_train = 5410
Number of columns in labels_df_train = 2
Number of rows in beneficiary_df_test = 63968
Number of columns in beneficiary_df_test = 25
Number of rows in inpatient_df_test = 9551
Number of columns in inpatient_df_test = 30
Number of rows in outpatient_df_test = 125841
Number of columns in outpatient_df_test = 27
Number of rows in labels_df_test = 1353
Number of columns in labels_df_test = 1
```

Observations:

The below section gives more information on the features available in these datasets:

1. **Beneficiary dataset:**
This consists of below features:

"BeneID","DOB","DOD","Gender","Race","RenalDiseaseIndicator","State","County","NoOfMonths_PartACov","NoOfMonths_PartBCov","ChronicCond_Alzheimer","ChronicCond_Heartfailure","ChronicCond_KidneyDisease","ChronicCond_Cancer","ChronicCond_ObstrPulmonary","ChronicCond_Depression","ChronicCond_Diabetes","ChronicCond_IschemicHeart","ChronicCond_Osteoporasis","ChronicCond_rheumatoidarthritis","ChronicCond_stroke","IPAnnualReimbursementAmt","IPAnnualDeductibleAmt","OPAnnualReimbursementAmt","OPAnnualDeductibleAmt"

2. **Inpatient dataset:**
This consists of below features:

"BeneID","ClaimID","ClaimStartDt","ClaimEndDt","**Provider**","InscClaimAmtReimbursed","AttendingPhysician","OperatingPhysician","OtherPhysician","AdmissionDt","ClmAdmitDiagnosisCode","DeductibleAmtPaid","DischargeDt","DiagnosisGroupCode","ClmDiagnosisCode_1","ClmDiagnosisCode_2","ClmDiagnosisCode_3","ClmDiagnosisCode_4","ClmDiagnosisCode_5","ClmDiagnosisCode_6","ClmDiagnosisCode_7","ClmDiagnosisCode_8","ClmDiagnosisCode_9","ClmDiagnosisCode_10","ClmProcedureCode_1","ClmProcedureCode_2","ClmProcedureCode_3","ClmProcedureCode_4","ClmProcedureCode_5","ClmProcedureCode_6"

3. **Outpatient dataset:**
This consists of below features:

"BeneID","ClaimID","ClaimStartDt","ClaimEndDt","**Provider**","InscClaimAmtReimbursed","AttendingPhysician","OperatingPhysician","OtherPhysician","ClmDiagnosisCode_1","ClmDiagnosisCode_2","ClmDiagnosisCode_3","ClmDiagnosisCode_4","ClmDiagnosisCode_5","ClmDiagnosisCode_6","ClmDiagnosisCode_7","ClmDiagnosisCode_8","ClmDiagnosisCode_9","ClmDiagnosisCode_10","ClmProcedureCode_1","ClmProcedureCode_2","ClmProcedureCode_3","ClmProcedureCode_4","ClmProcedureCode_5","ClmProcedureCode_6","DeductibleAmtPaid","ClmAdmitDiagnosisCode"

4. **Target dataset:**
This consists of below features:

"Provider","PotentialFraud"

In this dataset we have **PotentialFraud** as our **target** feature which is being predicted based on **Provider** feature; present in Inpatient dataset and Outpatient dataset.

Part 1: Data Cleaning and processing

Objective: In this step the main objective is to understand the dataset, clean any inconsistencies or errors, and standardize the data to perform exploratory data analysis and derive insights.

Cleaning the data set ensures accuracy and reliability of the Machine Learning model that's going to be built. By identifying these inconsistencies or errors in the data we can prevent the Model from learning incorrect patterns. It also improves the performance of the model. The real-world data contains lots of misinformation along with the useful information, it often contains missing information as well cleaning this data involves fixing this problem as well where we have to impute or remove those missing information for making the model rely on the true information. Another problem is the data is not consistent throughout, so to address this issue we must standardize the format and restructure the data which involves in converting the textual information into the numerical format and scaling all the information for ensuring the consistency. Especially when dealing with health care or patient level data from multiple sources, ensuring consistency is a bit difficult but any mistake or issue results in huge impact on the Models prediction.

Overall, performing these standards on our dataset cleans the entire information and make it consistent, accurate ensuring the developers and stakeholders to have confidence in the insights and the decisions made by the model.

Some of the common operations performed to clean the data includes handling missing values, Dropping out the duplicate entries, formatting the data using different standardization techniques, dropping unwanted features or columns, removing the outliers, adding relevant features, renaming the labels of the columns, scaling the data termed as feature scaling, encoding the class labels to numbers which is often termed as Label Encoding, Data Normalization and Data validation.

In this section, we'll cover the various steps taken to clean and preprocess the data to make it suitable for analysis.

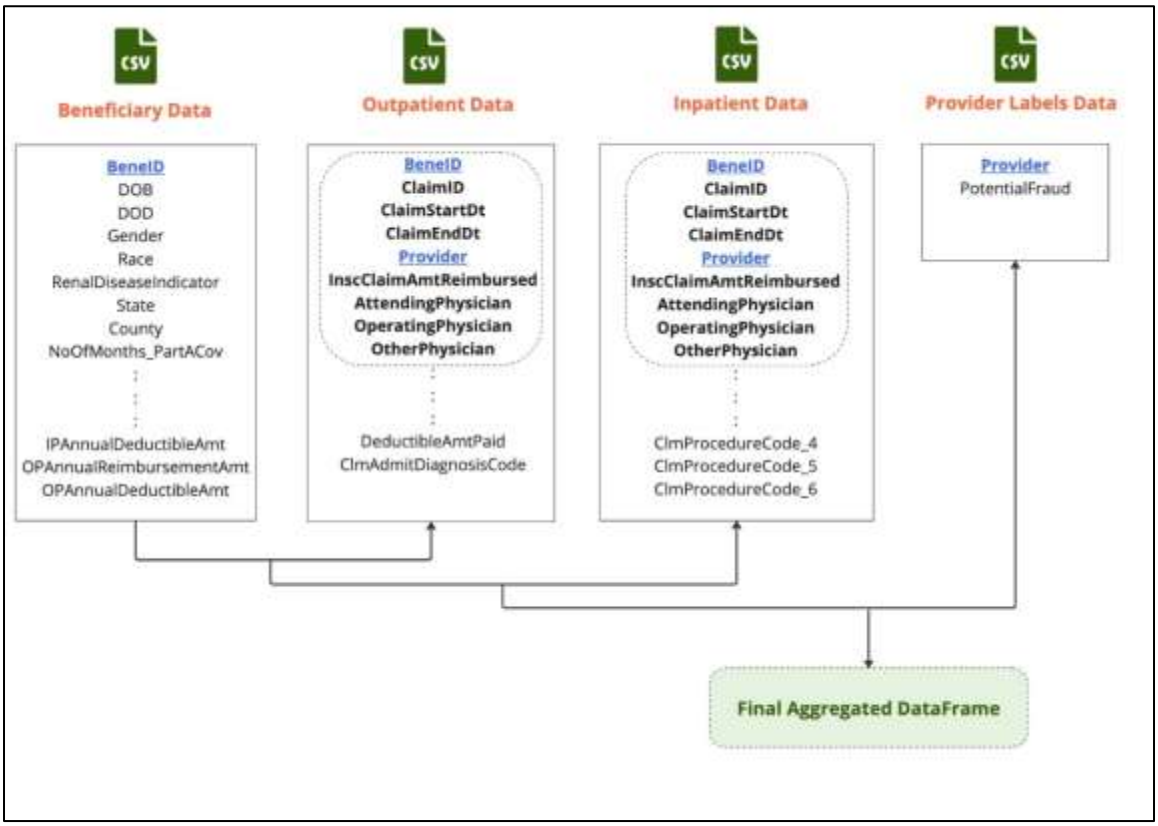
1. **Data Inspection**: After loading all the four types of data sets which includes patient's beneficiary data, inpatient, outpatient data and the labels. We have seen the number of missing information, structure which includes the number of non-null values, data types of the individual features available in each of the sources.

```
beneficiary_df = pd.read_csv('data/Train_Beneficiarydata-1542865627584.csv')
inpatient_df = pd.read_csv('data/Train_Inpatientdata-1542865627584.csv')
outpatient_df = pd.read_csv('data/Train_Outpatientdata-1542865627584.csv')
labels_df = pd.read_csv('data/Train-1542865627584.csv')
# shape of the dataset
print(beneficiary_data.shape)
```

2. **Combining the data to a single Data frame**: After the general inspection of the available information from all the sources we tried to identify the primary key and then combine all the data to a single pandas data frame which enables us to create a unified dataset and enriches features for further analysis and modelling.

```
# check common columns in inpatient, outpatient and beneficiary data
set(inpatient_data.columns) & set(outpatient_data.columns) & set(beneficiary_data.columns)
# merge the inpatient, outpatient and beneficiary data using the common columns
merged_data = pd.merge(inpatient_data, outpatient_data, on=list(set(inpatient_data.columns) & set(outpatient_data.columns)), how='outer')
# check the shape of the merged data
print(merged_data.shape)
# merge the merged data with the beneficiary data
merged_data = pd.merge(merged_data, beneficiary_data, on='BeneID')
# merge the merged data with the provider data
merged_data = pd.merge(merged_data, labels_data, on='Provider')
```

The below image shows the exact operation of how the merging is happening:



3. **Changing the Data Types**: On the merged dataset we tried to identify the type of data each column holds and fixed the data types, One of such wrong representations include the Date type which is stored as an object. So the necessary conversions into datetime was done using pandas library.

```
# Changing date columns to datetime
merged_data['ClaimStartDt'] = pd.to_datetime(merged_data['ClaimStartDt']) # claim start date
merged_data['ClaimEndDt'] = pd.to_datetime(merged_data['ClaimEndDt']) # claim end date
merged_data['DOB'] = pd.to_datetime(merged_data['DOB']) # date of birth
merged_data['DOD'] = pd.to_datetime(merged_data['DOD']) # date of death
merged_data['AdmissionDt'] = pd.to_datetime(merged_data['AdmissionDt']) # admission date
```

4. **Label Encoding**: we tried to identify the categorical columns along with the potential target variable and converted the values like Yes, No to numeric 1s and 0s along the data frame

```
# change the target variable to 0 and 1
merged_data['PotentialFraud'] = merged_data['PotentialFraud'].map({'Yes': 1, 'No': 0})
merged_data['PotentialFraud'].value_counts()
```

5. **Making the Data Consistent**: We tried to identify the data present in the columns like Renal Disease Indicator, Chronic Condition columns and observed that the values are not consistent, so we mapped the data 'Y' with 1 and '0' with 0 .

```
# check all columns with ChronicCond
chronic_columns = [col for col in merged_data.columns if 'ChronicCond' in col]
chronic_columns
```

6. **Structuring the Data**: We restructured the values present in the columns from '1' and '2' to binary structure which include '1' for 1 and '0' for 2 to make it consistent.

```
# since all the columns have values 1 and 2, it need to keep structure
# consistent as this is a binary column
# change the values of 2 to 0
for col in chronic_columns:
    merged_data[col] = merged_data[col].map({2: 0, 1: 1})
```

7. **Handling Missing Values**: the last DOD value is from 2009 which indicates the beneficiary data from the year 2009, therefore all the missing values are replaced with this datetime.

```
# check for missing values in DOD column
merged_data['DOD'].isnull().sum()
max_date = merged_data['DOD'].max()
# fill missing values in DOD column with the maximum date
merged_data['DOD'] = merged_data['DOD'].fillna(max_date)
# check again
merged_data['DOD'].isnull().sum()
```

8. **Feature Engineering**: As part of feature engineering, we are trying to extract or create new features based on the available information in the data frame. Created **Age** column from the available DOB and DOD, **AdmitPeriod** in days based on the Admission and discharge dates, **ClaimPeriod** based on the Claim start date and end date, **PhysiciansSame** to identify if the attending, operating and other physicians are same for a particular patient,

DiseaseCount to keep track of number of diseases for the patient, **TotalClaimProcedures** to identify the total number of medical procedures used on a beneficiary which gives an estimated cost incurred by beneficiary.

```
# create a new feature "Age"
merged_data['Age'] = (merged_data['DOD'] - merged_data['DOB']).dt.days // 365
# create a new feature "AdmitPeriod"
merged_data['AdmitPeriod'] = (merged_data['DischargeDt'] - merged_data['AdmissionDt']).dt.days
# create a new feature "ClaimPeriod"
merged_data['ClaimPeriod'] = (merged_data['ClaimEndDt'] - merged_data['ClaimStartDt']).dt.days
# create column with total number of claims procedure codes
claim_proc_columns = [col for col in merged_data.columns if 'ClmProcedureCode' in col]
merged_data['TotalClaimProcedures'] = merged_data[claim_proc_columns].nunique(axis=1)
```

9. **Data Validation:** We verify the data in the columns and validate that any non-meaningful information is not stored locally like age, Admit Period, Deductible Amount are non-negative for ensuring the Quality in the data.

```
# validate age
merged_data[merged_data['Age'] < 0]
# check for negative values in AdmitPeriod
merged_data[merged_data['AdmitPeriod'] < 0]
# check for negative values in deductible amount
merged_data[merged_data['DeductibleAmtPaid'] < 0]
```

10. **Handling Outliers**: We attempted to pinpoint the data points that deviate significantly from the norm, as managing outliers is crucial for eliminating genuine anomalies that could disproportionately impact our analysis or modeling. To achieve this, we employed Z scores, a statistical measure that helps identify outliers by quantifying how many standard deviations a data point is from the mean. By calculating the absolute Z scores for each numerical feature in the dataset and setting a threshold of 3 standard deviations, we flagged data points that fall beyond this threshold as potential outliers. This process aids in maintaining the integrity of our analysis by excluding extreme values that could skew our results. Finally, we determined the total number of identified outliers using the sum of Boolean values indicating outlier presence.

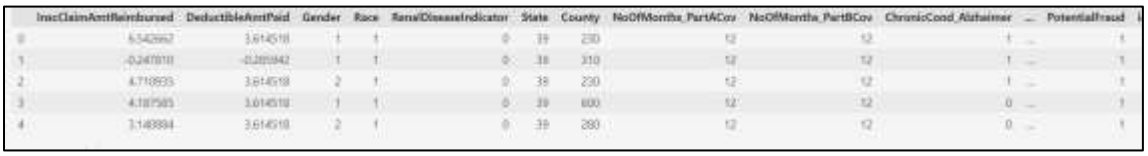
```
z_scores = np.abs((df[all_numerical_columns] - df[all_numerical_columns].mean()) / df[all_numerical_columns].std())
outliers_numerical = (z_scores > 3).any(axis=1)

# number of outliers
outliers_numerical.sum()
```

11. **Removing Unnecessary Data columns**: we removed the columns which contains missing values and kept the required features. It will help us to streamline the dataset and focus on the most relevant features for further analysis or modelling. It will also help in reducing the dimensionality and improve the efficiency of our model.

```
BeneID                    0
ClaimID                   0
ClaimStartDt              0
ClaimEndDt                0
Provider                  0
                         ..
Physician_Same            0
DiseasesCount             0
PhysiciansCount           0
TotalClaimCodes           0
TotalClaimProcedures      0
Length: 64, dtype: int64
```

12. **Data Normalization**: this technique is used to rescale all the values in the columns to a standard scale. This is solely performed since we will be having lot of features with different units or scales which should not dominate the analysis and the model shouldn't be biased towards certain features. We used Standard Scalar function and scaled all the numerical variables.



After **cleaning and preprocessing**, the dataset **appears** to be in a **structured format**, suitable for further analysis and modeling.

Part 2: Exploratory Data Analysis (EDA):

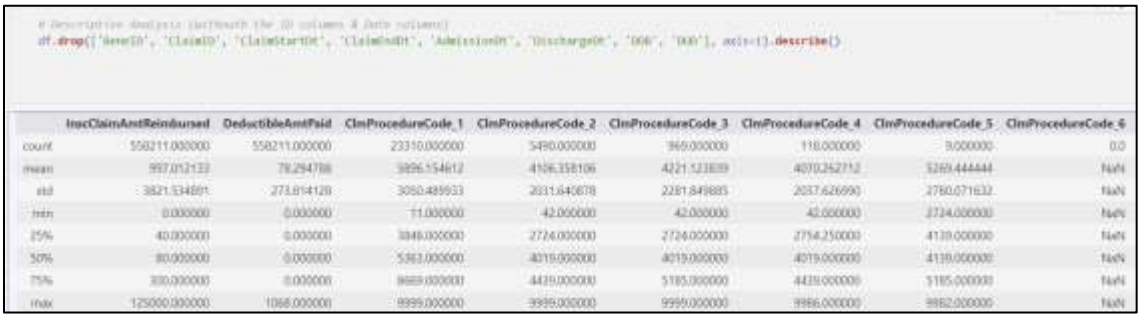
It's an approach to analyze the datasets and summarize the main characteristics, often employing graphical methods. It's the first major step done after preprocessing to understand the structure, patterns, distributions, relationships with the variables. The end outcome of the work is to build an ML Model which can do better predictions for the given problem scenario. To improve the accuracy of the model, preprocessing the data, selecting the significant features plays a major role.

In a nutshell it deals with exploring the entire dataset right from the kind of data we have with us to how do we treat the different types and what all the problems associated with the data and the original problem statement, how can we address and handle them. Basically, answering all the questions from the data.

After diving deep into the beneficiary or the patient data we segregated and merged, we tried to analyze them over different types including visual exploration to understand the patterns in the data, detect certain data points which doesn't follow the pattern significantly deviating from the rest of the data points available in the dataset.

In this section we will discuss the approaches we have covered in solving the questions we have with respect to the data through EDA:

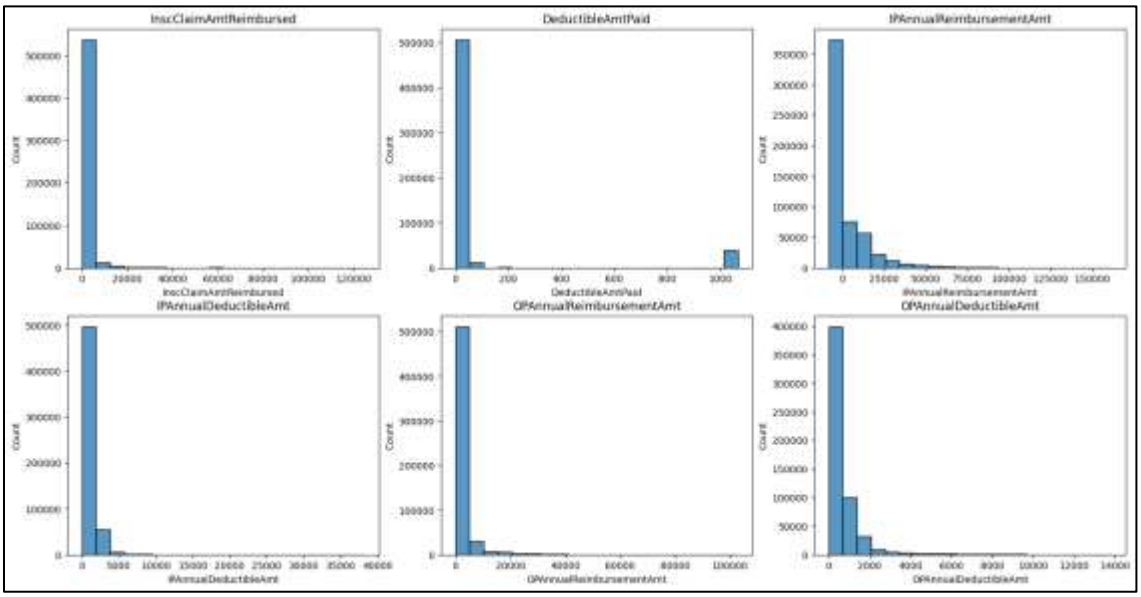
1. **Descriptive Analysis**: This is the foundational step in exploring the numerical variables in our datasets which summarizes the central tendencies, spread and distributions of the variables elucidating the prevalence of different categories. We used describe () function to see these variations. This helped us to answer some of the questions like what's the average claim amount reimbursed, what is the average age of the claimant, what is the mortality of the claimants. These questions were typically solved by this analysis, for instance, The average age of the claimants is around 73 years old with a standard deviation of 13 years which tells us that there is diverse Age range in the dataset, The average **reimbursement** amount is approximately **$997** with a **large standard deviation** of **$3827** indicating significant variability in the costs associated with insurance claims. Similarly, The **'Gender'** column has a mean around **1.58**, possibly indicating that the dataset is slightly skewed towards one gender. The most important observation was the '**PotentialFraud'** with a **mean** around **0.38** indicating approximately **38%** of the claims may involve Fraudulent Activities, this crucial identification allowed us to further explore the data more and motivates us to build a clean, significant reliable predictive model.



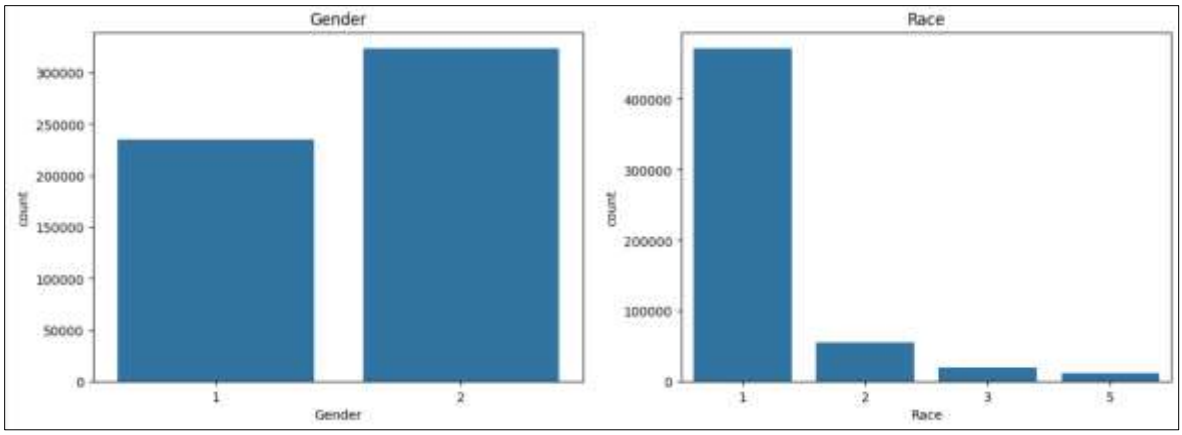
```
df.drop(['BeneID', 'ClaimID', 'ClaimStartDt', 'ClaimEndDt', 'AdmissionDt', 'DischargeDt', 'DOB', 'DOD'], axis=1).describe()
```

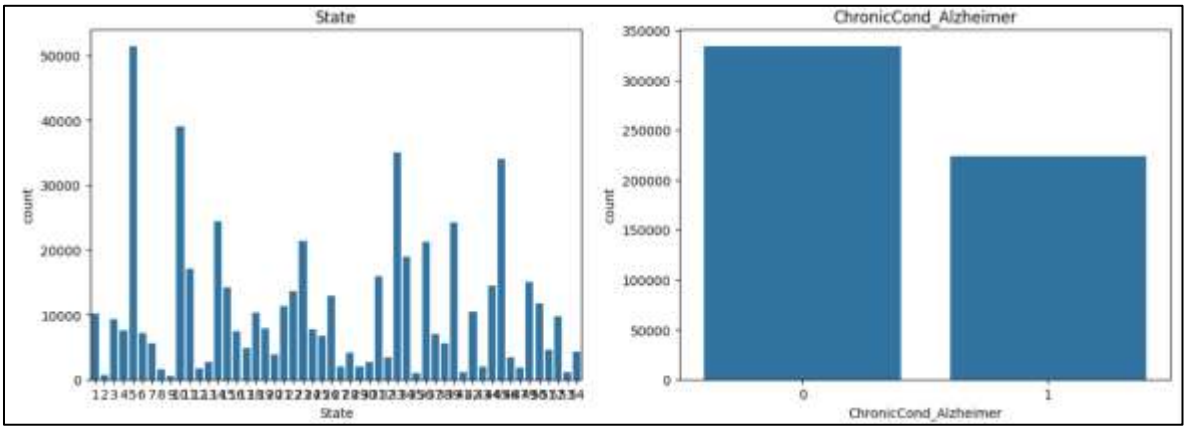
	IncClaimAnntReimbursed	DeductibleAmtPaid	ClmProcedureCode_1	ClmProcedureCode_2	ClmProcedureCode_3	ClmProcedureCode_4	ClmProcedureCode_5	ClmProcedureCode_6
count	558211.000000	558211.000000	23310.000000	5490.000000	969.000000	116.000000	9.000000	0.0
mean	997.012133	78.294788	3896.154612	4106.358106	4221.123839	4070.262712	3269.444444	NaN
std	3821.534891	273.814128	3050.489933	2031.640678	2281.849885	2037.626990	2780.071632	NaN
min	0.000000	0.000000	11.000000	42.000000	42.000000	42.000000	2724.000000	NaN
25%	40.000000	0.000000	3846.000000	2724.000000	2724.000000	2754.250000	4139.000000	NaN
50%	80.000000	0.000000	5363.000000	4019.000000	4019.000000	4019.000000	4139.000000	NaN
75%	300.000000	0.000000	9669.000000	4439.000000	5185.000000	4439.000000	5185.000000	NaN
max	125000.000000	1068.000000	9999.000000	9999.000000	9999.000000	9986.000000	9982.000000	NaN

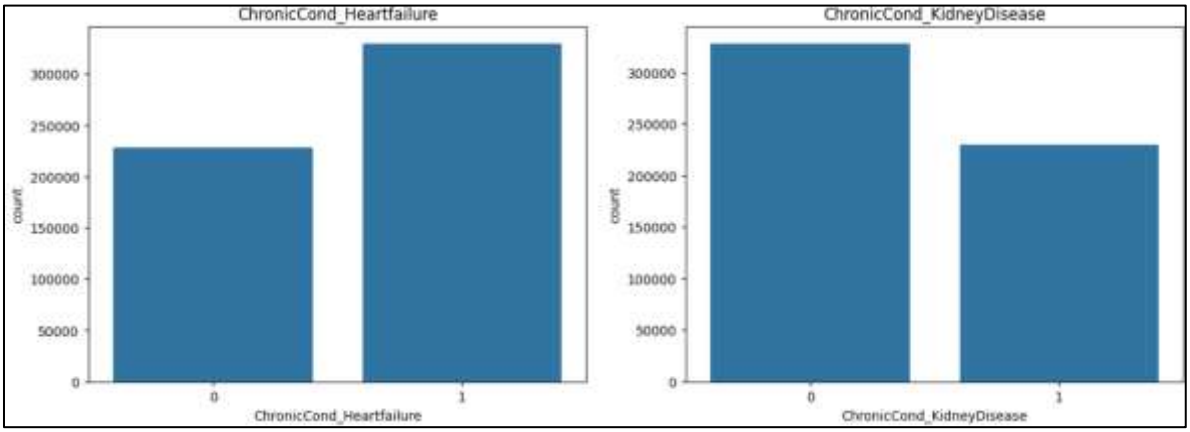
2. **Numerical Data Distribution**: We wanted to check the distribution and skewness of each of the feature using some of the common visualization techniques. We observed that some of the distributions like Insurance Claim Amount Reimbursed, Inpatient Annual reimbursement amount, **outpatient Annual Reimbursement** amount are **right skewed** indicating that there is some pattern across most of the common medical claims where larger number of claims are falling in the lower range of the reimbursement amounts and lower range of deductibles.

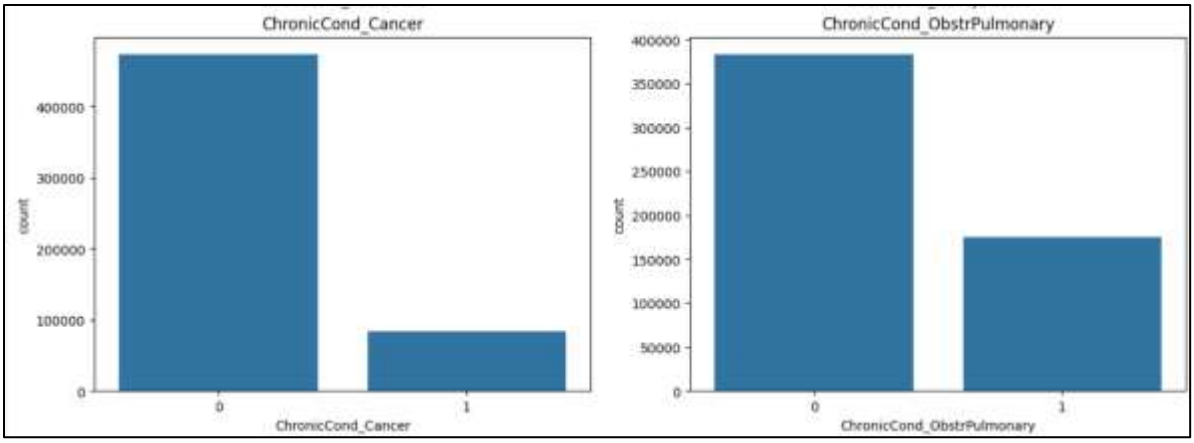


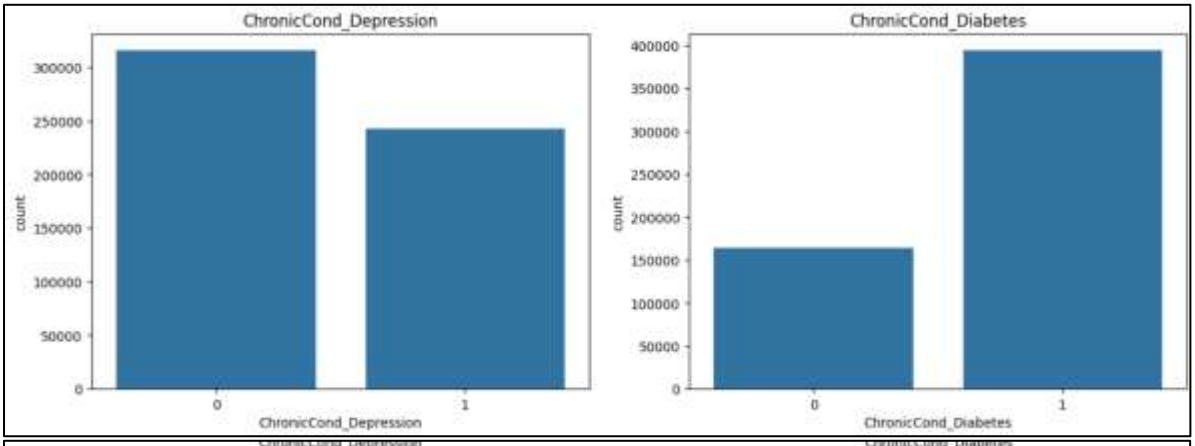
3. **Categorical Data Distribution**: this analysis plays a pivotal role in unraveling the intricate patterns associated with the qualitative variables. This analysis helped us to understand how the data is distributed across various genders, Race, states and various other chronic diseases. We found that the greatest number of males had raised for the claims with a count of 3,10,000 and females of 2,40,000 numbers. Most number of **Claimants** had **chronic Diabetes** with a **count of 3,80,000** followed by chronic Heart Failure with a count of 3,40,000. Distribution of Possible Fraudulent cases were also observed by the bar graph indicating there are **38%** of the cases are fraudulent in the dataset with a count of **2,10,000** Fraudulent and **3,46,000** Non-Fraudulent.

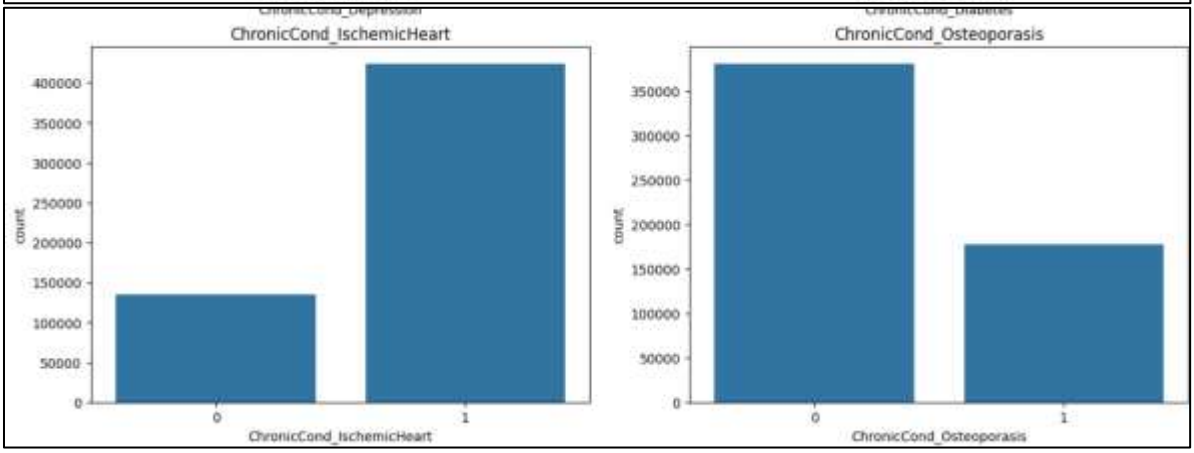

Gender
count
Gender
Race
count
Race


State
count
State
ChronicCond_Alzheimer
count
ChronicCond_Alzheimer

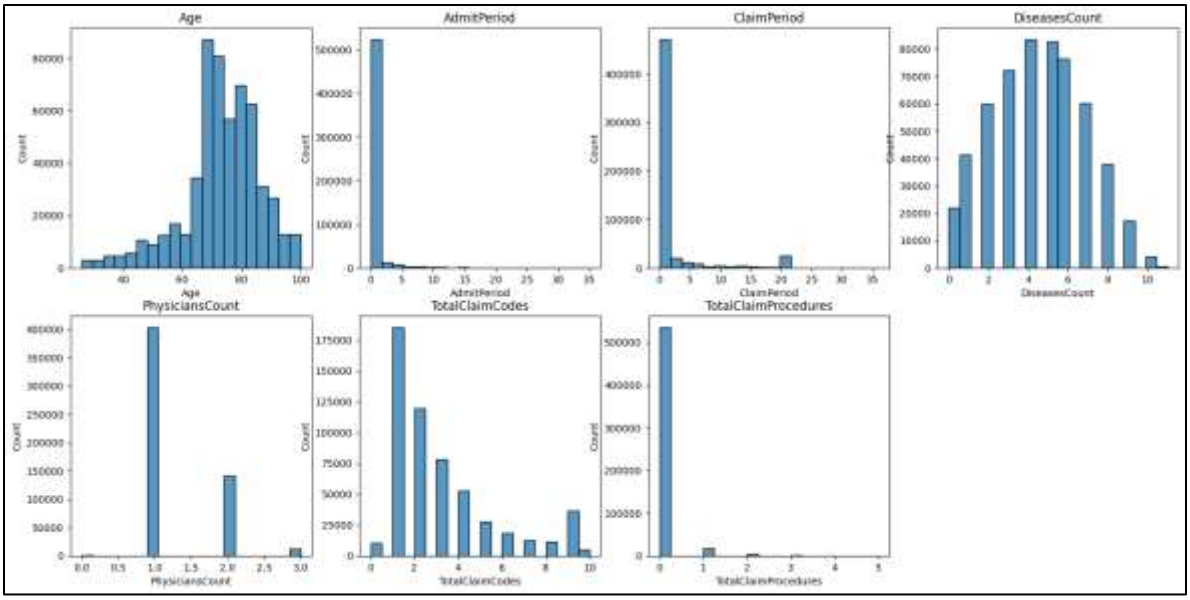

ChronicCond_Heartfailure
count
ChronicCond_Heartfailure
ChronicCond_KidneyDisease
count
ChronicCond_KidneyDisease





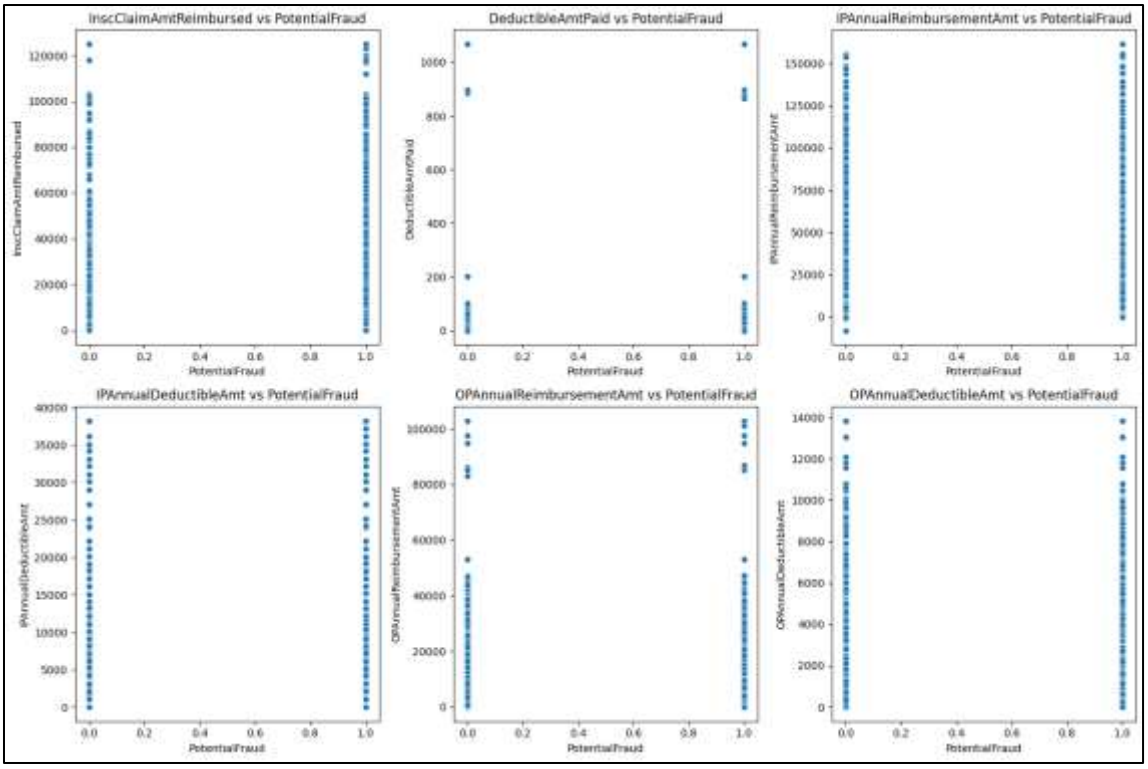


4. **EDA of Feature Engineering**: we tried to create the bar plots and histograms for the feature engineered variables like Age, **AdmitPeriod**, Disease count and few other. We observed that there are around **90,000 people** from the **age group 65s**, out of 5Lakh people there are very less people around less than 40,000 people who got admitted more than 4 more months.

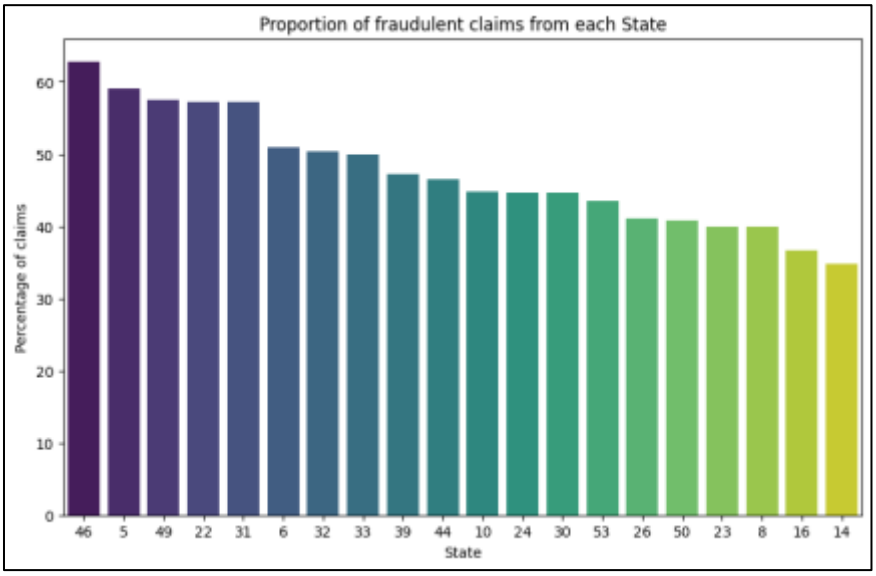


5. **Multivariate Analysis**: this statistical technique examines the relationship between multiple variables simultaneously. This enables us to understand the dependencies and associations among the variables. Here the axis are labelled with two variables which has to be compared and the color of the points in the scatter plot represents the value of third variable. We observed that there is a negative correlation between **IPAnnualDeductibleAmt** and **OPAnnualReimbursementAmt** for individual policies means that, on average, as the annual deductible for inpatient care increases, the annual reimbursement amount for outpatient care tends to decrease along with the Fraud cases.

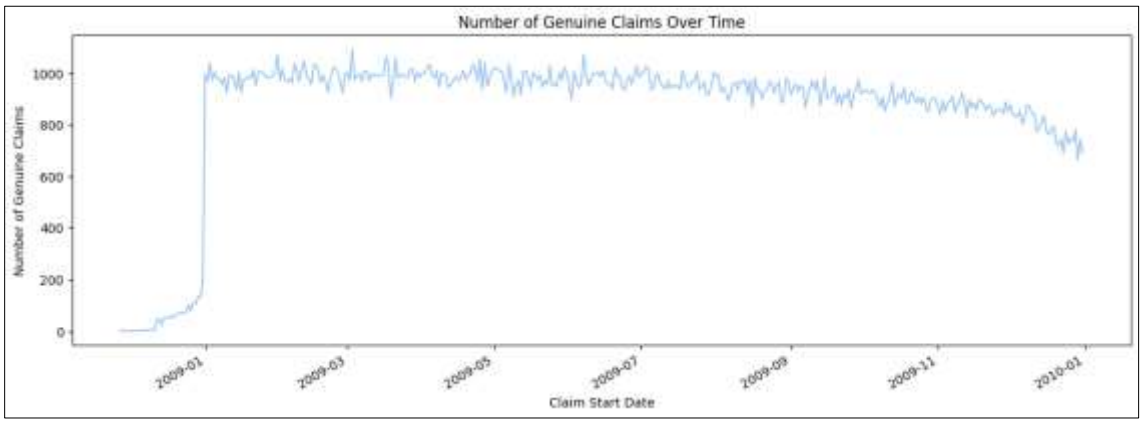
6. **Bivariate Analysis**: This technique is used to elucidate the relationship between any two variables in the dataset. We tried to identify the relationship between the individual variables and the target variable. We observed to be a **positive correlation** between the two variables as the **InscClaimAmtReimbursed** increases, the potential for fraud also increases. Also, there **does not appear** to be a clear **correlation** between **opAnnualReimbursementAmt** and **potential fraud**.



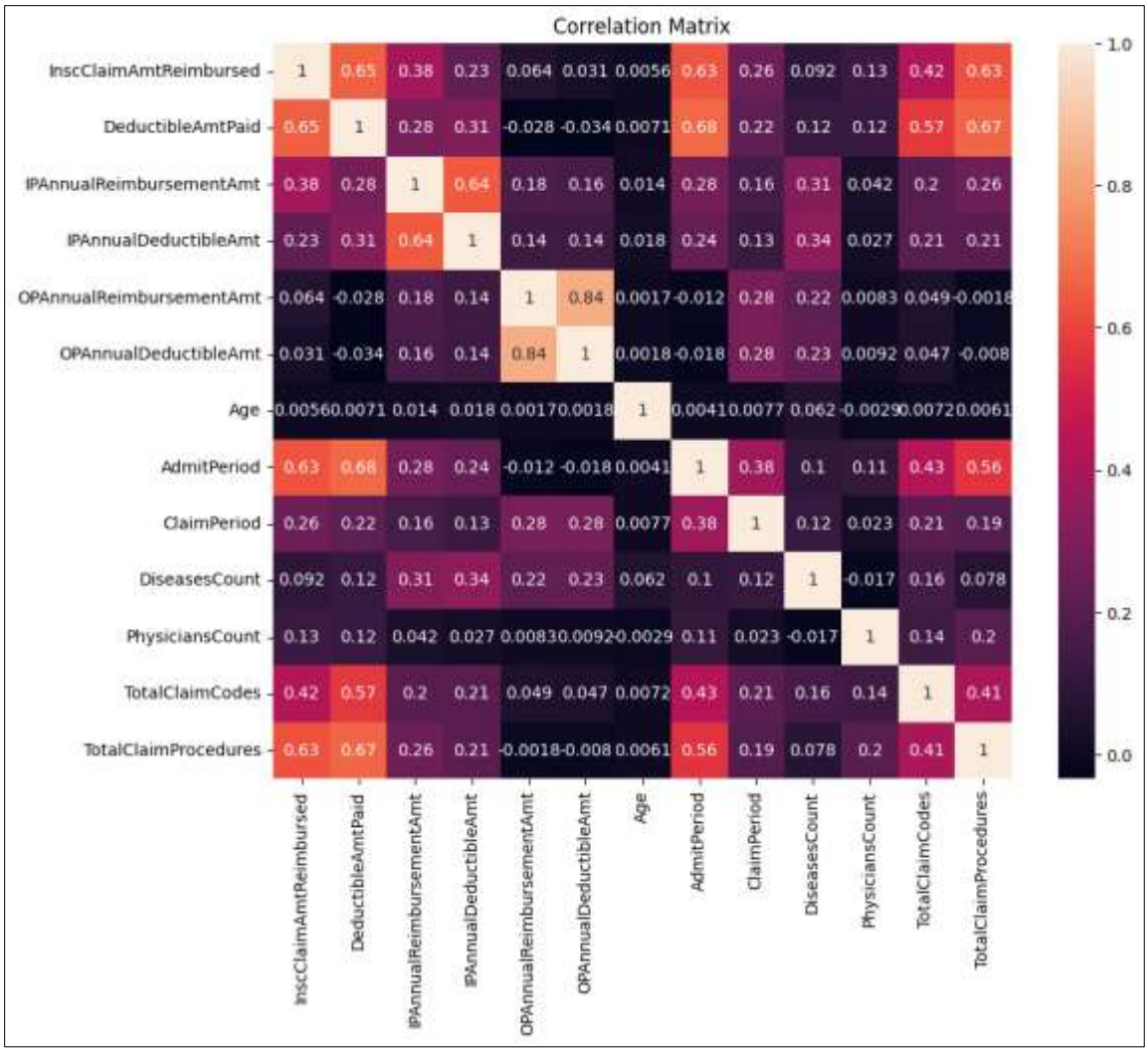
7. **Key Variable Insights**: in this we tried to cover the distributions of important variables within the dataset, offering valuable insights that can inform further analysis and decision-making. This helped us to answer some of the questions like what's the proportion of the **fraudulent claims across the states**, counties, and race. Similarly, we tried to identify the crucial information plotting Total clinical physicians across the potential fraud where we found out that there are many more physicians who have not been flagged for fraud than those who have been flagged.



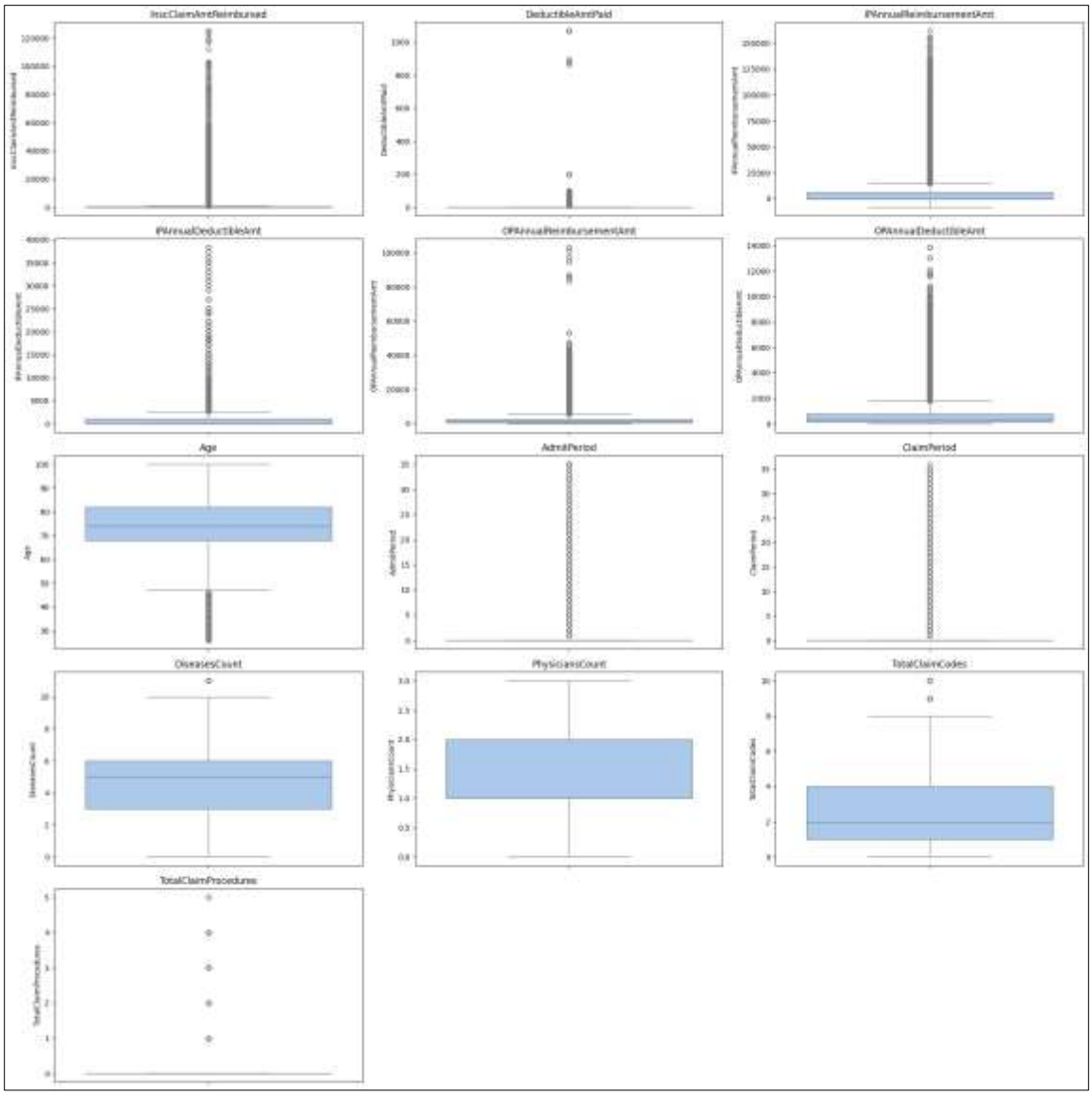
8. **Time Series Analysis**: this technique involves studying the patterns, trends, and relationships within the data to draw conclusions or insights about future behavior. We tried to plot number of claims over time, Number of Genuine claims and the Fraudulent claims over time. **We observed** that lot of claims were happened around June, 2009 out of which most of them were fraudulent and over the span of time in 2010 the **genuine claims got reduced** which clearly tells us **that yet there's increase in the fraudulent claims**.



9. **Correlation Matrix**: this is a table to indicate the correlation between the pairs of the variables. Each cell represents the value ranging from -1 to 1 which depicts the negative, uncorrelated and positively correlated values. Here, the correlation between "InscClaimAmtReimbursed" and "DeductibleAmtPaid" is 0.65, which indicates a moderately strong positive relationship which means that as the amount of reimbursed claim amount increases, the deductible amount paid also tends to increase. There is a **negative correlation** between **"PhysiciansCount"** and **"TotalClaimCodes"** (-0.2) which means that as the number of physicians increases, the number of claim codes tends to decrease. Also, we can't just assume because two variables are correlated does not mean that one variable causes the other it is possible that there is a third variable that is causing both variables to change so we tried to further identify the pattern in the data identifying and removing the outliers and removing one of the highly correlated variables which need not be required for modelling.



10. **Outlier Detection**: in this step we try to identify the points which significantly deviate from the rest of the data. Handling the outliers plays a key role in removing the genuine anomalies which doesn't unduly influence the analysis or modeling process. There are methods like visual inspection, statistical methods and some ML algorithms to identify them and they can be removed, transformed or imputed with certain suitable values. We tried to identify using Boxplot which contains the center line telling it's the median, the middle 50% is the IQR and the data points that fall 1.5* IQR(lines extending the whisker) represents the outliers. It can be seen that the distributions of some measures, like **InscClaimAmtReimburse** have a wider range than others, like Age. Additionally, it appears that some measures, like **IPAnnualReimbursementAmt have outliers**. Mathematically, we used Zscore for identifying them.



After diligently conducting data preprocessing and exploratory data analysis (EDA) procedures, we've meticulously examined the dataset's patterns and addressed various inquiries through our analysis, including normalization techniques. With these steps completed, the data stands poised and refined, fully prepared for the modeling phase.