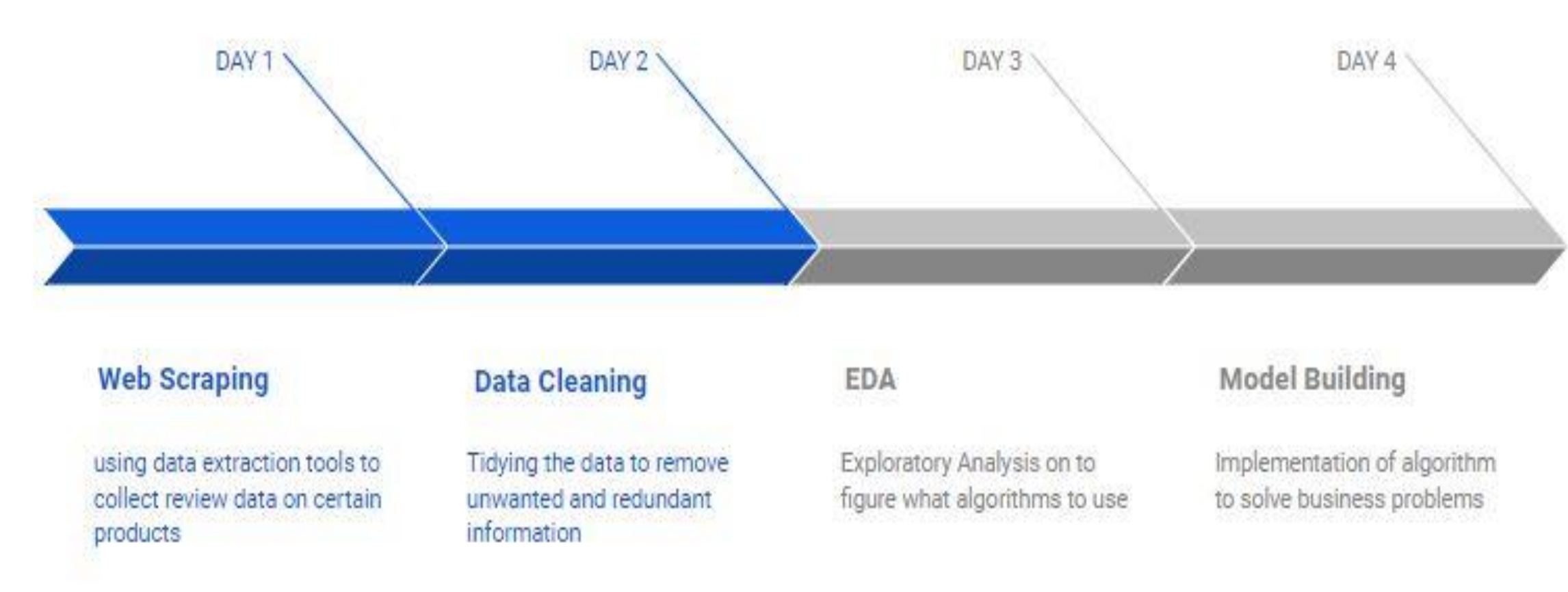
FRAUD ANALYTICS

By:-
NAVEEN

BUSINESS OBJECTIVE

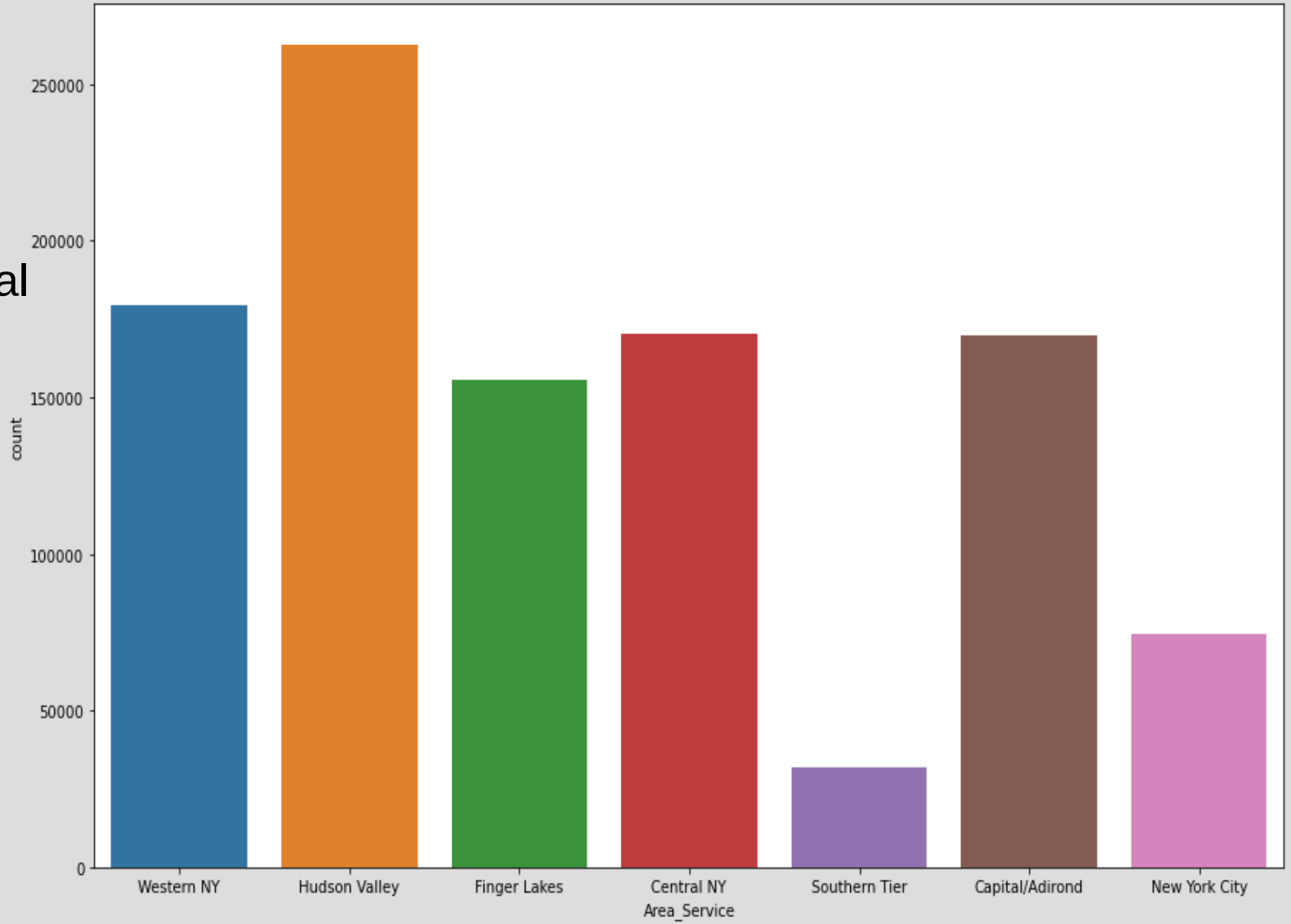
To Predict whether the customer claim is Genuine or Fraud

PROJECT ARCHITECTURE/FLOW

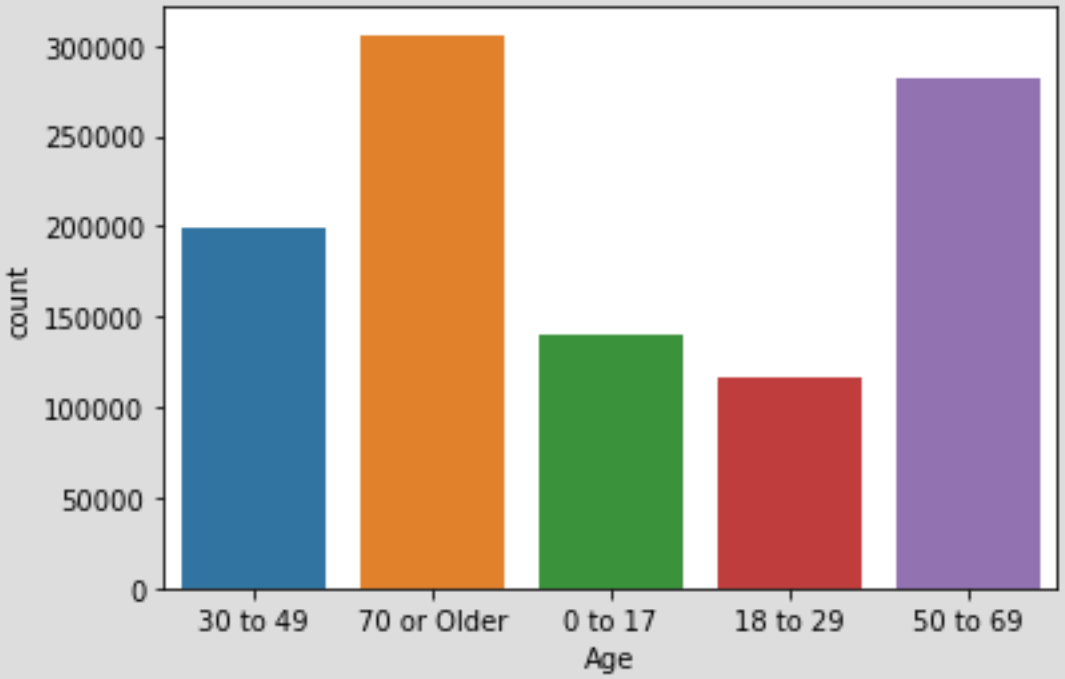


This is a count plot for the area_service column. We can see that Hudson Valley is the most served area followed by Western NY, Central NY, Capital and Finger Lakes respectively where the data on hospital facilities was collected.

From this plot, we observe highest peak (count) in Hudson Valley and the lowest in the Southern tier.



Here is a count plot for the age column which shows patients of different age groups starting from 0 to 70.

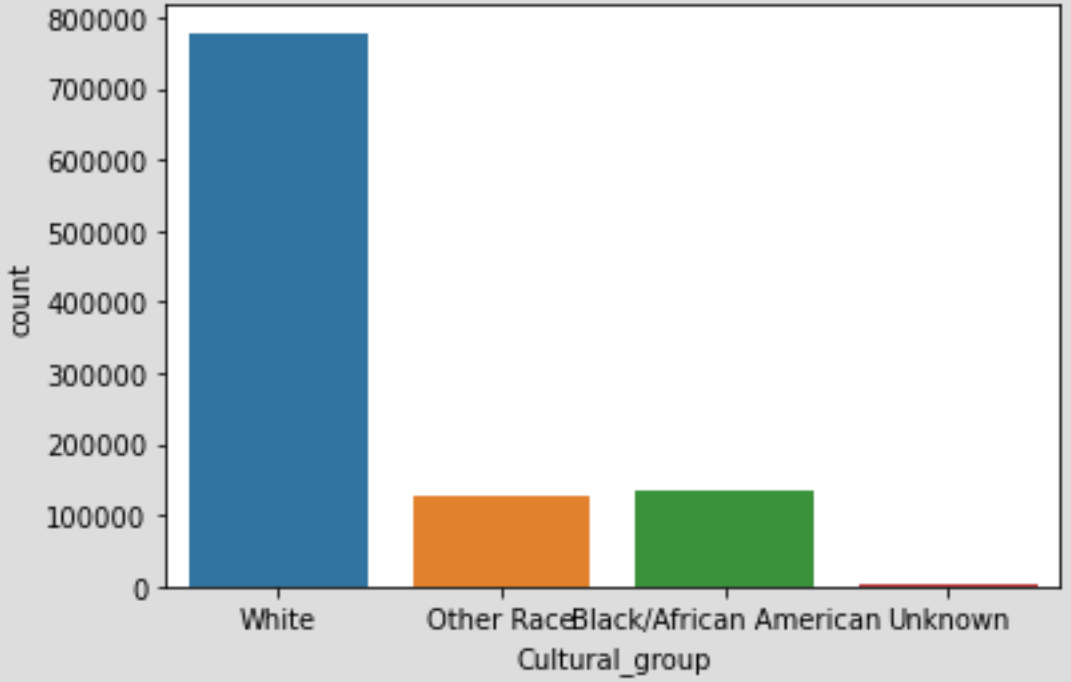


We can see that most of the patients are of age group 70 or older followed by group 50 to 69. The minimum amount of patients are in group 0 to 17, followed by group 18 to 29.

The highest peak is observed in ages groups 70 or older and lowest in the age groups between 18-29.

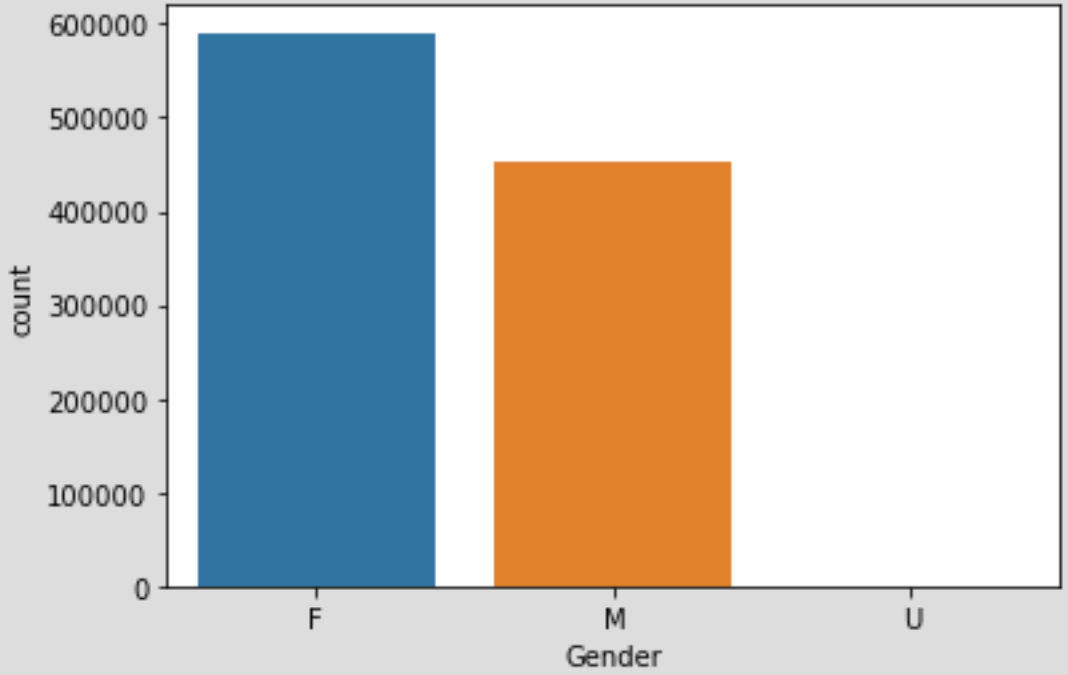
This is a count plot for the cultural_group column. Black depicts African Americans, White depicts Americans and the Other Race is the group other than these two.

We can see that the most number of patients are White. Blacks and Other Races are almost equal in number.Unknown are almost negligible.



Here is a count plot for the Gender column which shows the gender of the patients.

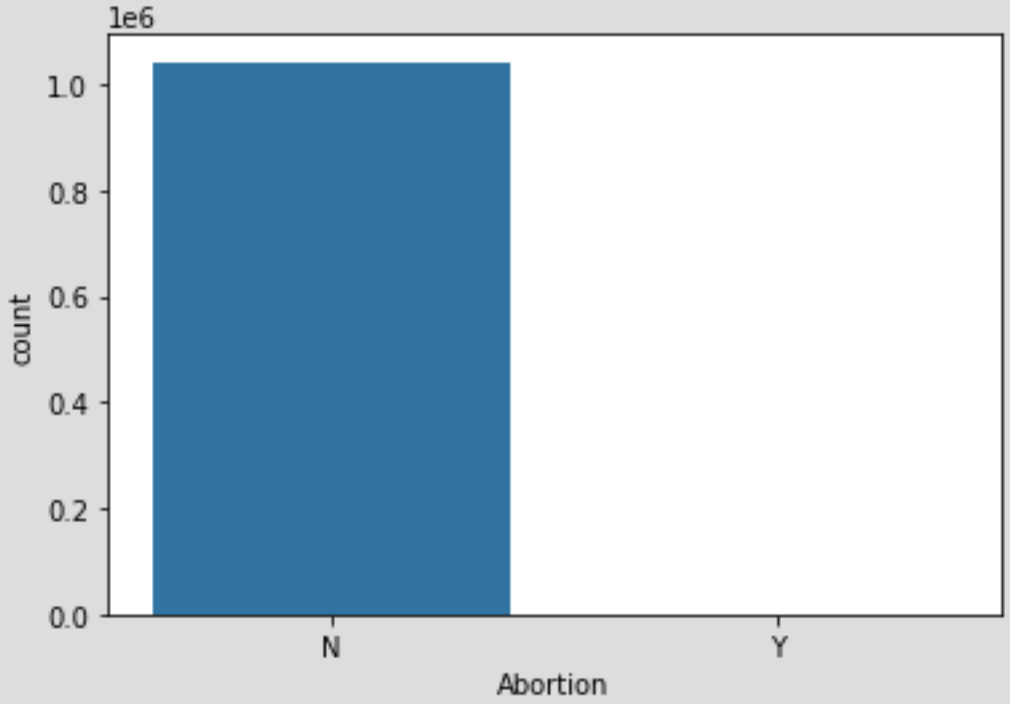
We can see that female patients are more in number than the male patients.



Here is a count plot for the abortion column.

We can see that none of the patients went for abortion.

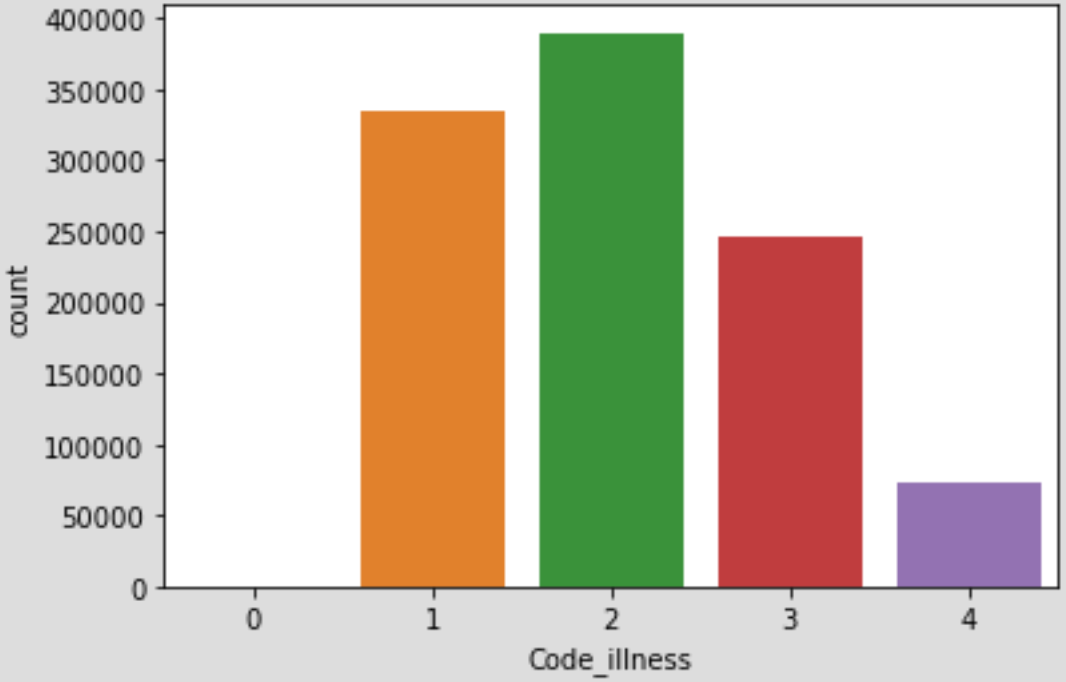
This depicts that there is an imbalance in the data which could create a bias in the model building.



Here is a count plot for the illness code column. 1 depicts mild illness, 2 is for moderate, 3 for severe and 4 for indeterminate.

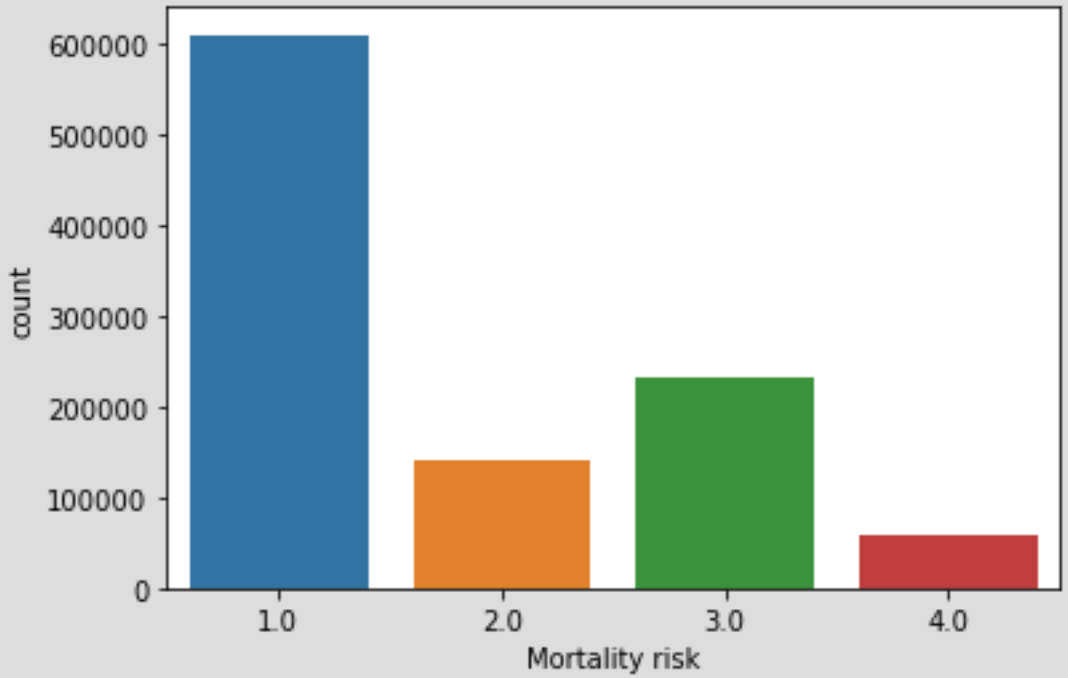
We can see that most number of patients have moderate illness followed by mild illness & severe.

The least number lies in indeterminate illness group.



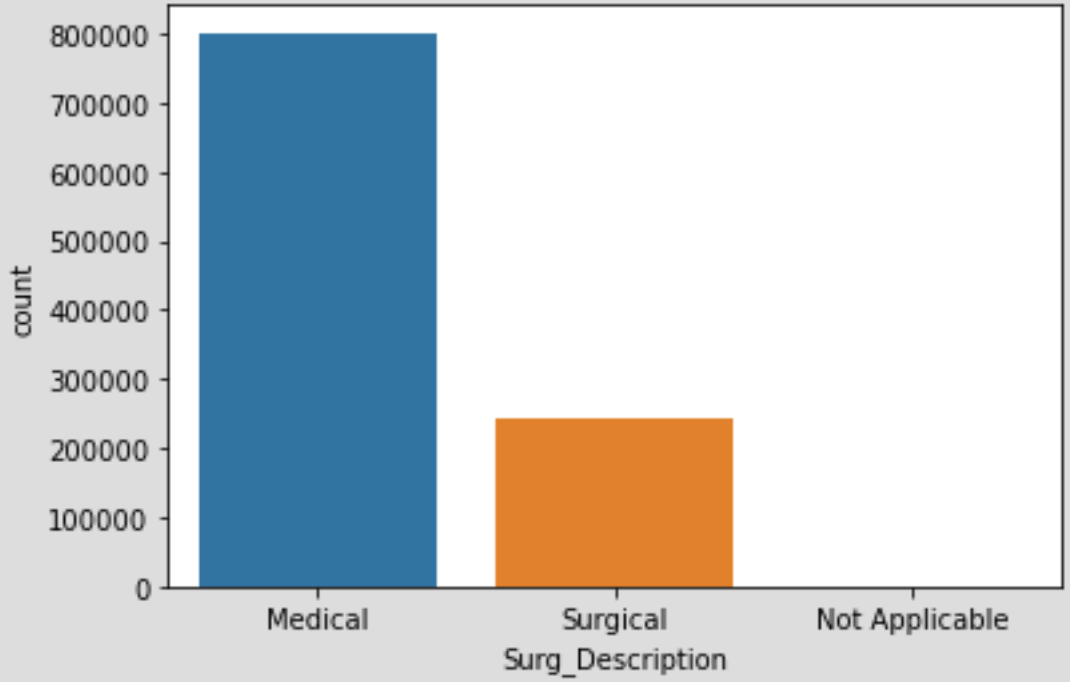
Here is a count plot for the mortality risk column which shows the mortality risk for the patients. 1 is for minor risk, 2 for moderate, 3 for major and 4 for severe.

We can see that risk is minor for most of the patients followed by major risk.



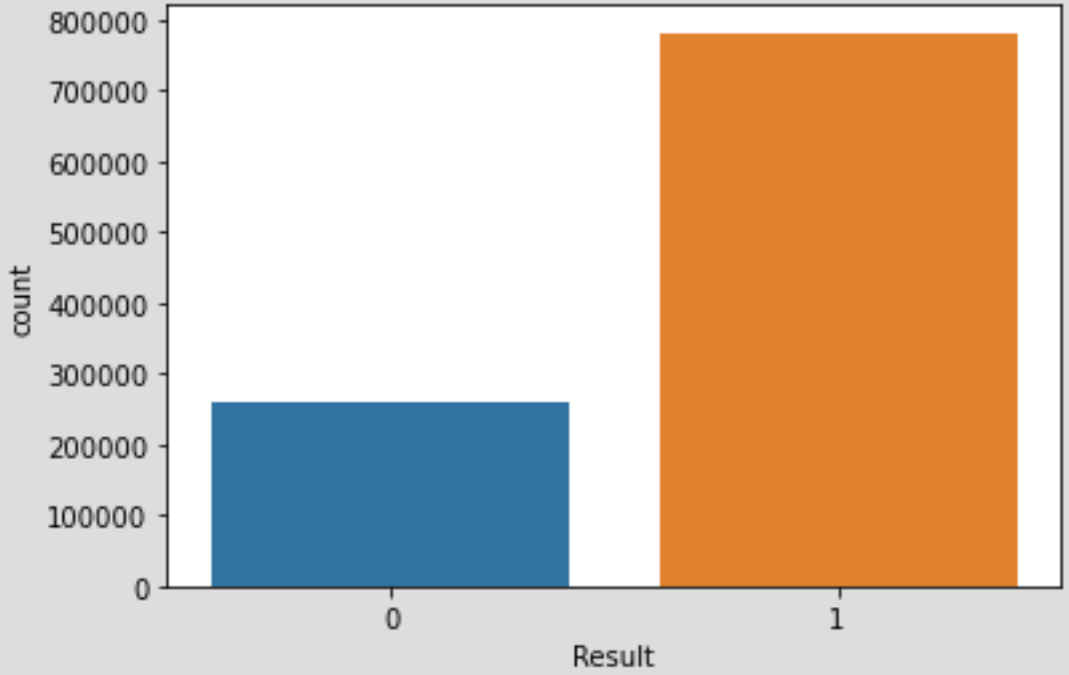
This is the count plot for the Surg_Description column which shows the different types of treatment provided to patients.

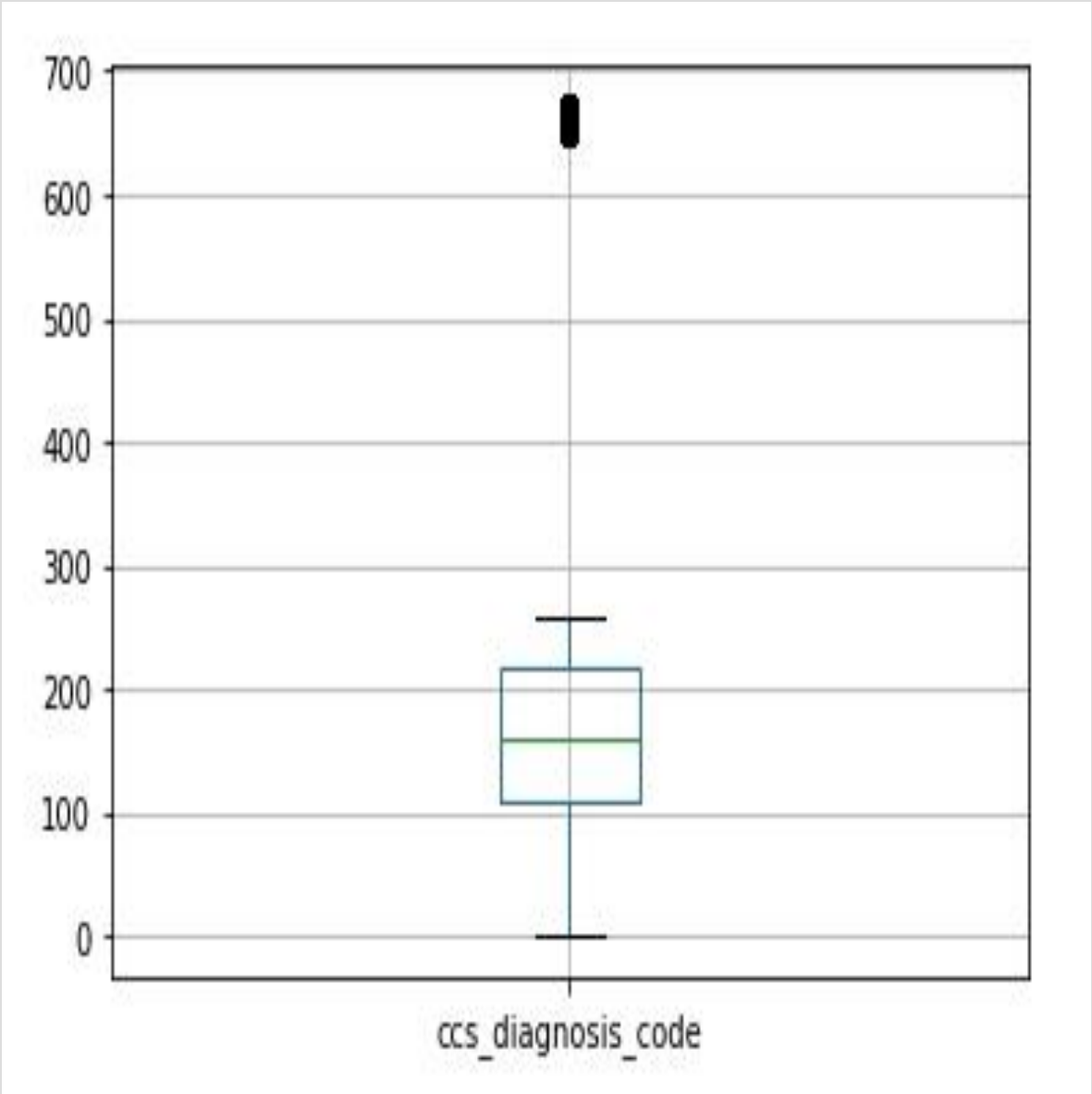
We can see that more amount of patients were given medical treatment as compared to surgical treatment.

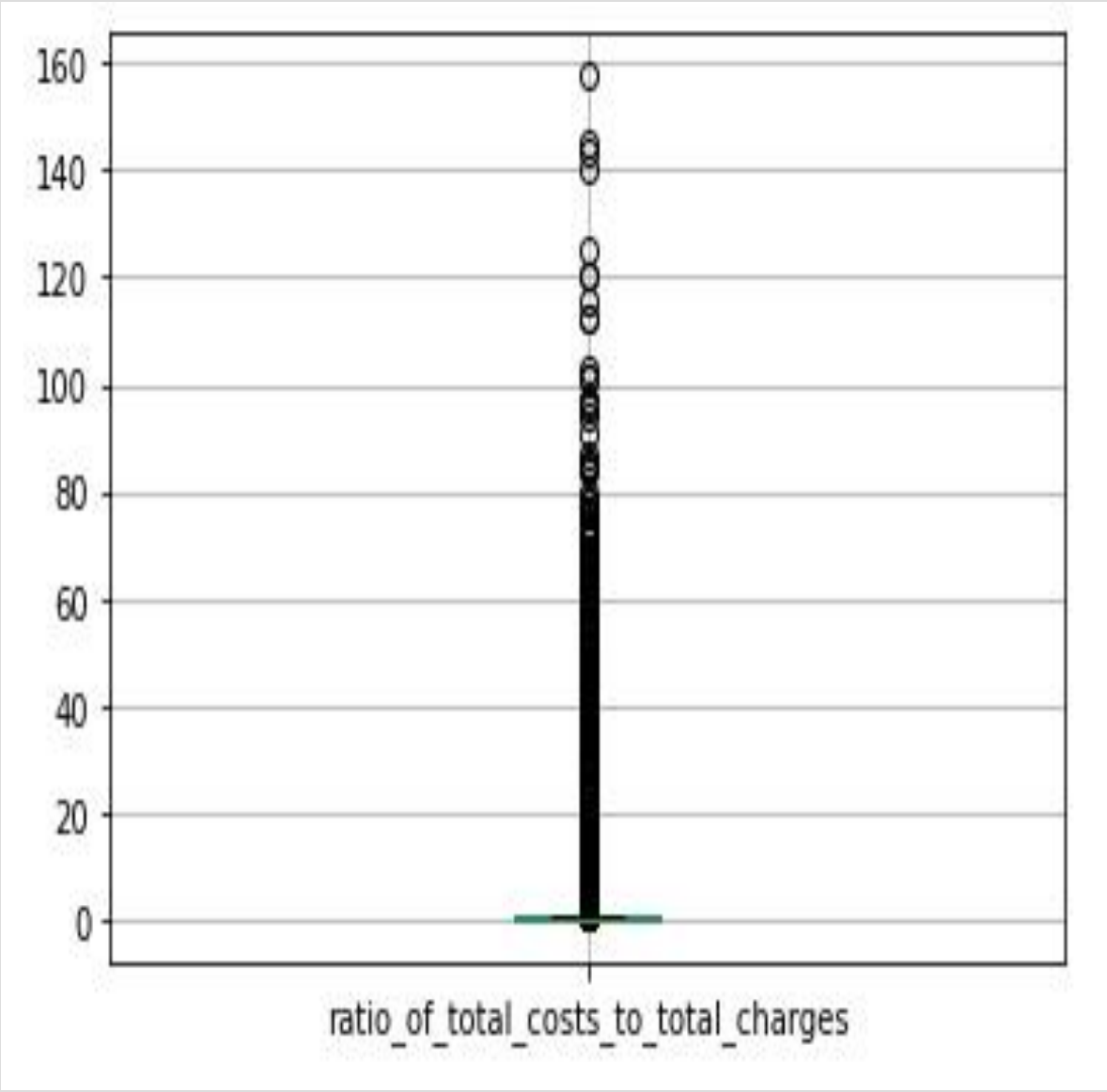


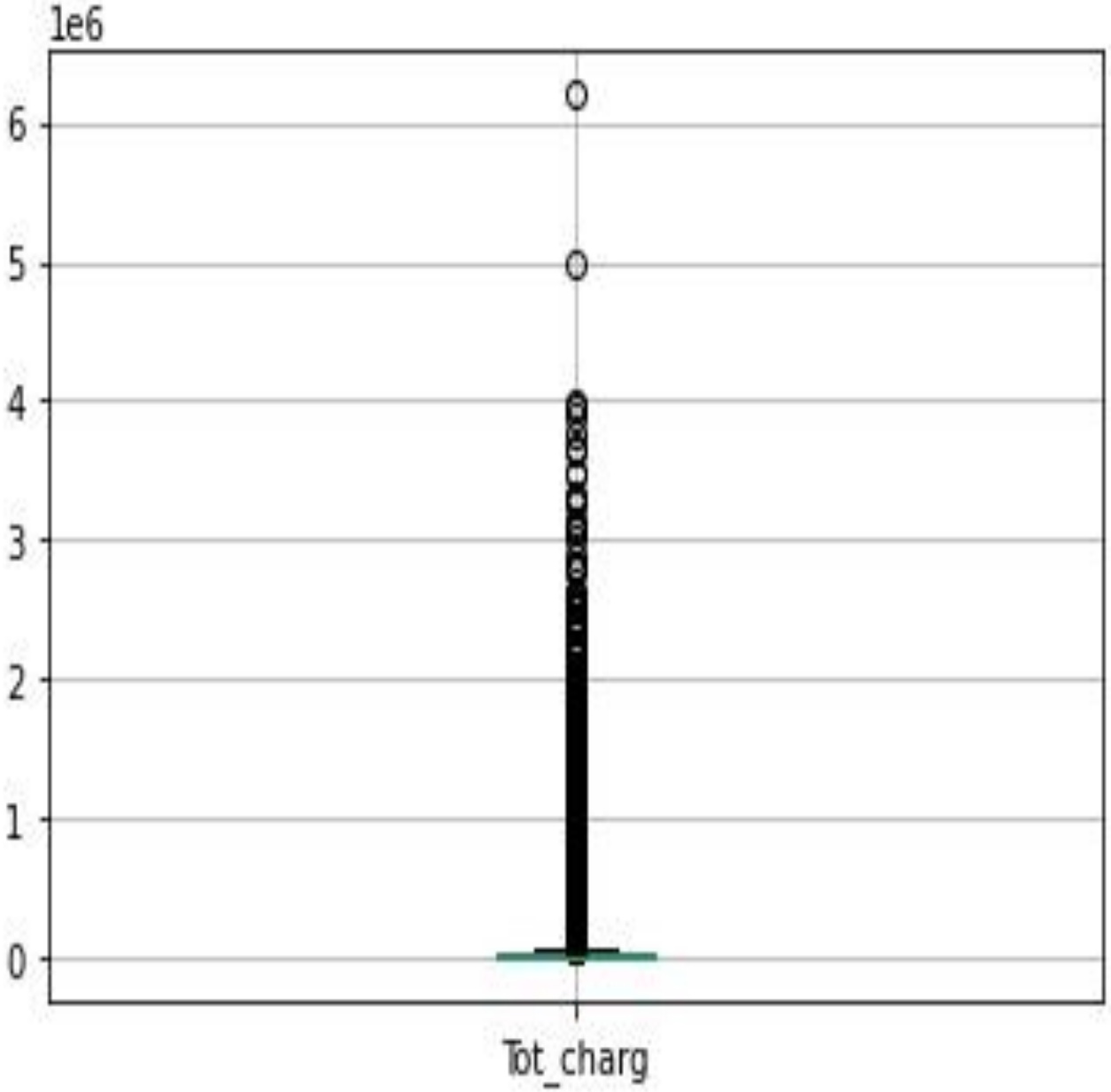
Here is a count plot for the result column. Here 1 shows the count of genuine claims and 0 shows the count of fraud claims.

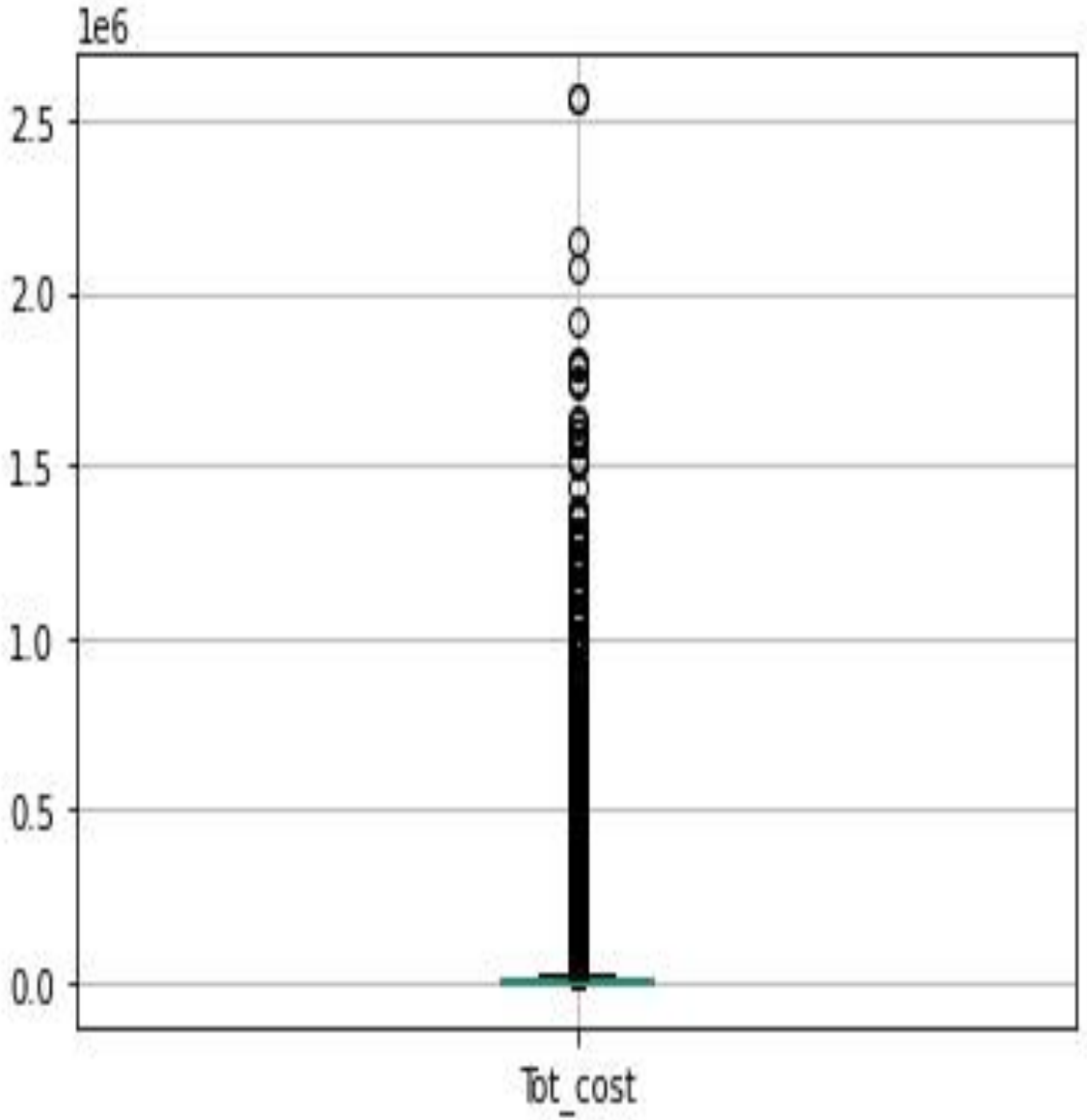
We can clearly see the imbalance in the result as the count of genuine claims is much greater than the count of fraud claims.

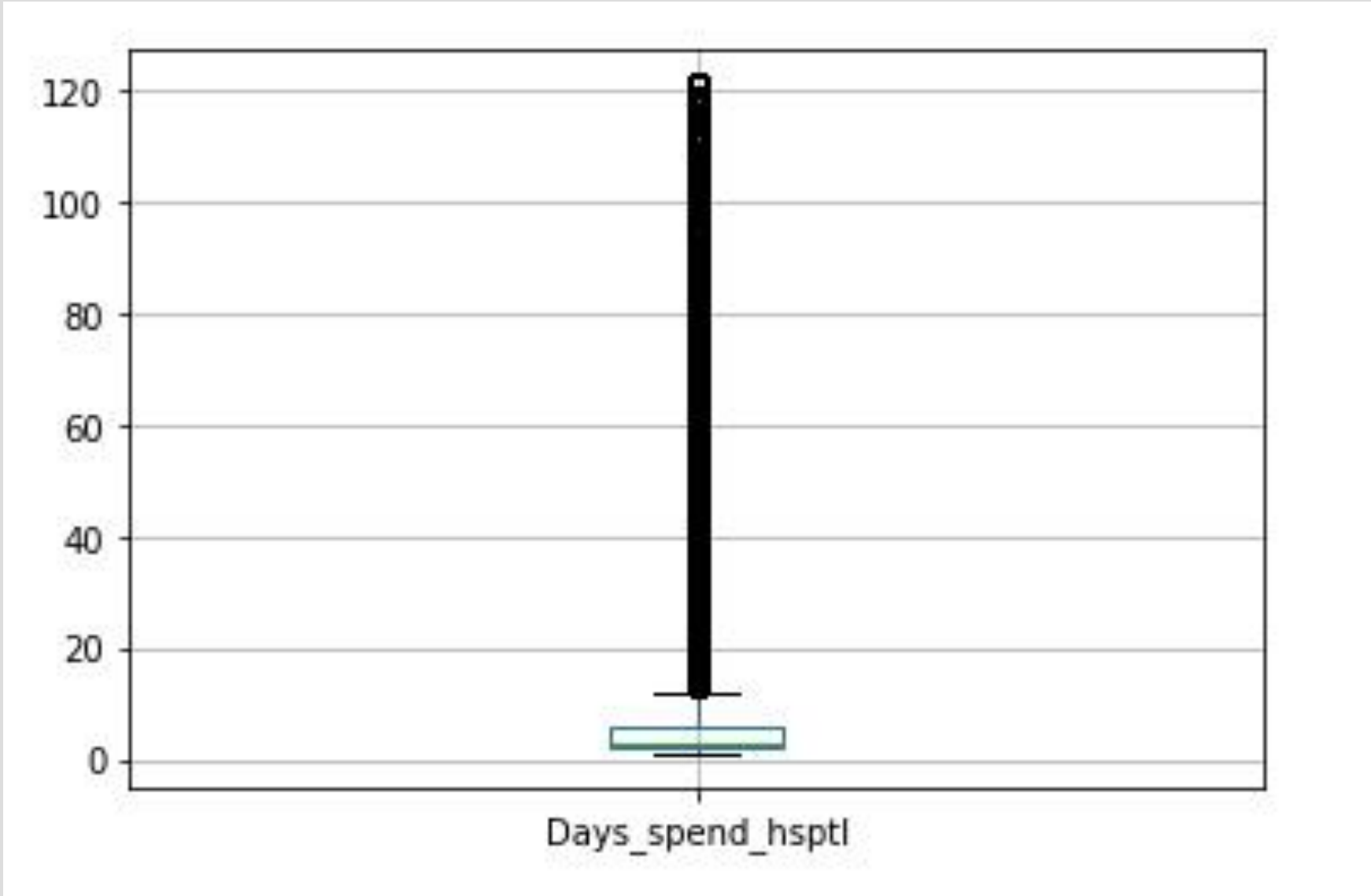


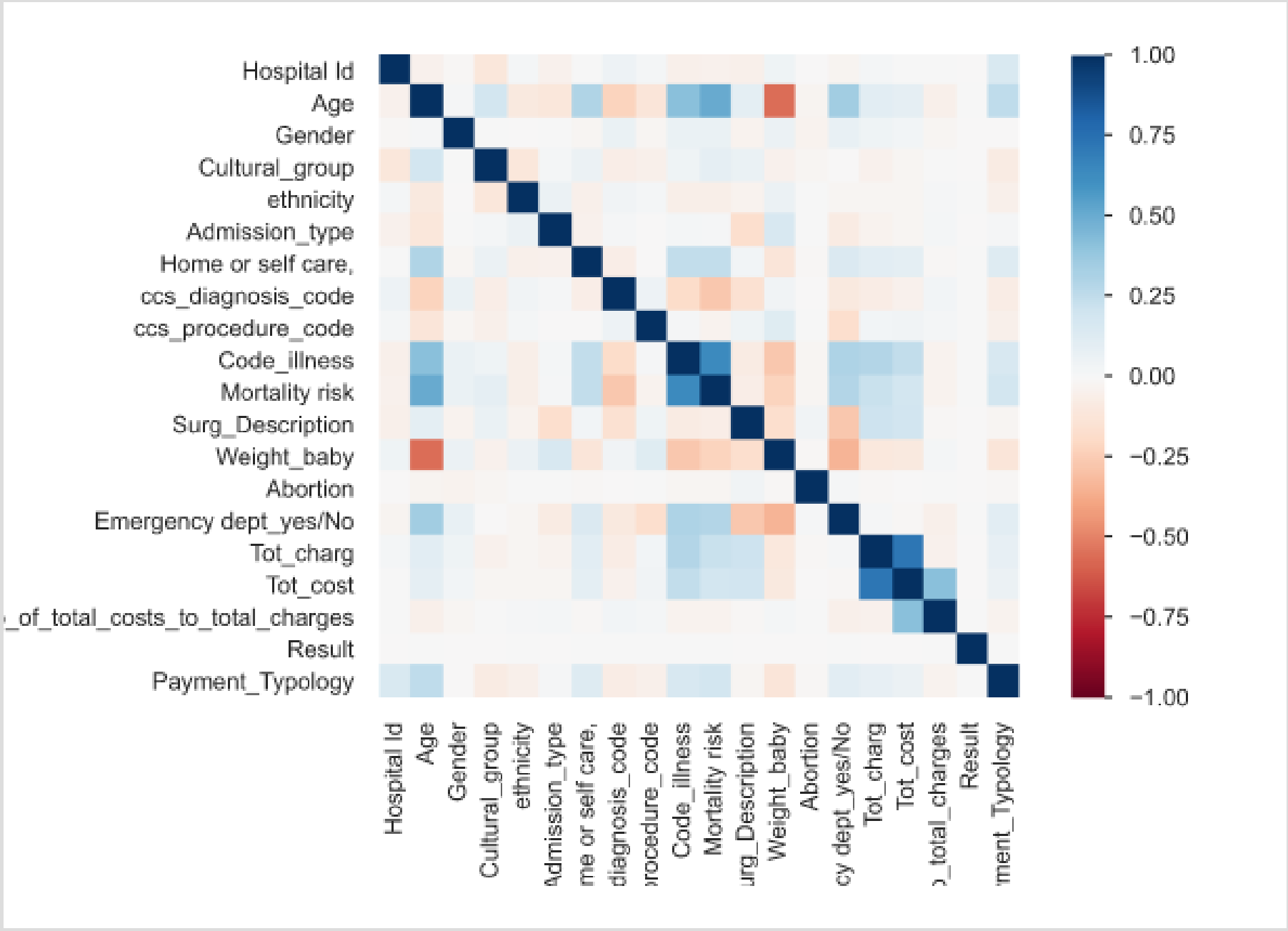

700
600
500
400
300
200
100
0
ccs_diagnosis_code


160
140
120
100
80
60
40
20
0
ratio_of_total_costs_to_total_charges


1e6
6
5
4
3
2
1
0
Tot_charg

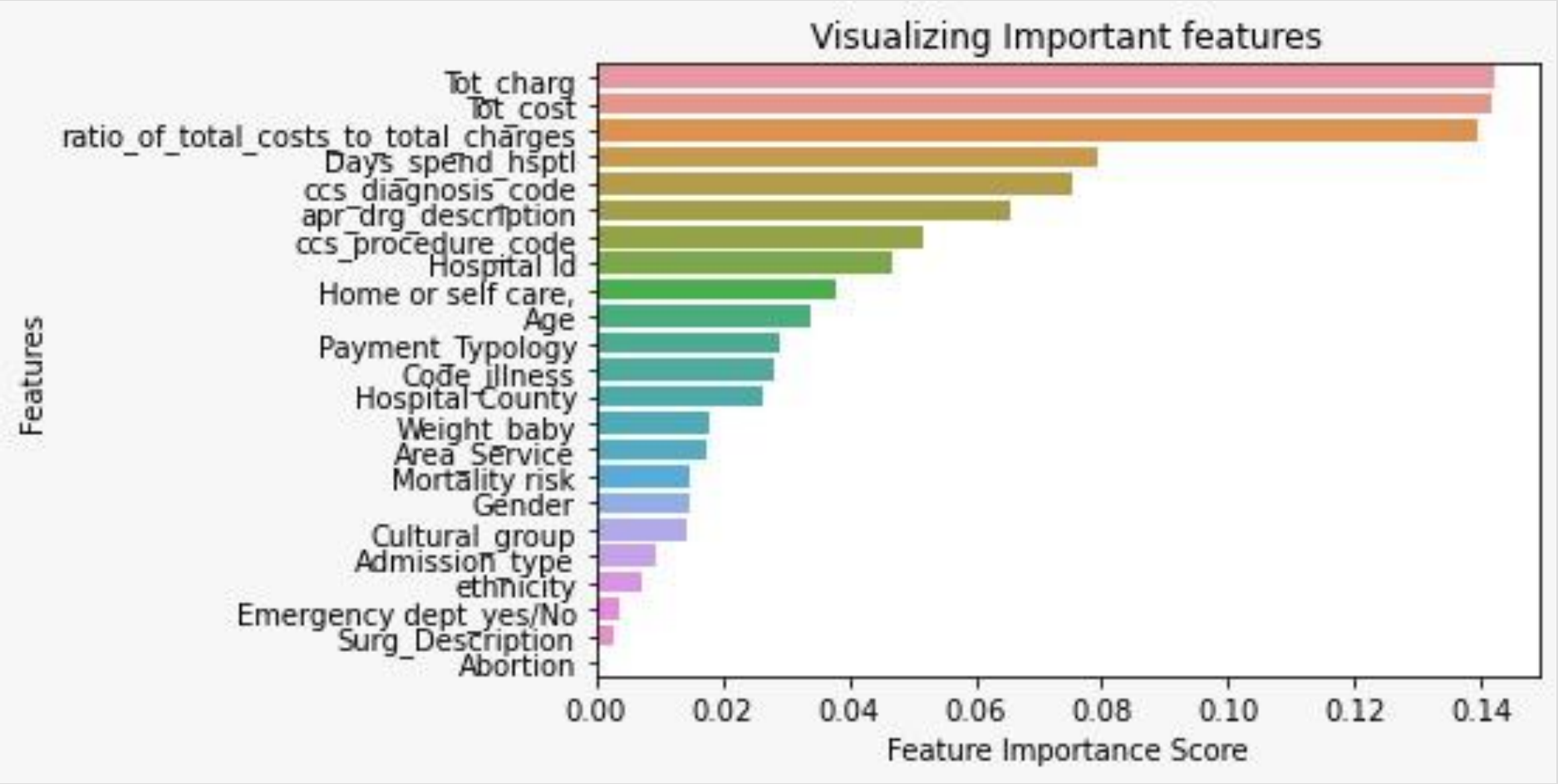

1e6
2.5
2.0
1.5
1.0
0.5
0.0
Tot_cost


120
100
80
60
40
20
0
Days_spend_hsptl



Heatmap of Correlation

FEATURE ENGINEERING



- We remove the columns Hospital County and Hospital ID since irrespective of which hospital it is, there will be claims.
- We remove the weight_baby column to as 90% of the rows are zeros and also it has low score as predicted by the ExtraTrees model.
- We remove the apr_drug_description as it is just the description of the disease and does not have a great impact in the model building.
- We remove the ratio_of_total_cost_to_charge column as it just indicates whether the hospital is making profit or not and it would not be an important feature for model building.
- The feature Abortion has only one value and hence would create biasness in the model, also it has the lowest score as predicted by the ExtraTrees model. Thus we would also drop that column.

```
Tot_charg                              0.141988
Tot_cost                               0.141878
ratio_of_total_costs_to_total_charges  0.139632
Days_spend_hsptl                       0.079202
ccs_diagnosis_code                     0.075467
apr_drg_description                    0.065754
ccs_procedure_code                     0.051622
Hospital Id                            0.046678
Home or self care,                     0.037825
Age                                    0.034121
Payment_Typology                       0.028896
Code_illness                           0.028076
Hospital County                        0.026566
Weight_baby                            0.017975
Area_Service                           0.017434
Mortality risk                         0.014795
Gender                                 0.014787
Cultural_group                         0.014544
Admission_type                         0.009291
ethnicity                              0.007147
Emergency dept_yes/No                  0.003737
Surg_Description                       0.002538
Abortion                               0.000046
dtype: float64
```

- The observation shows that the Days_spent_hsptl column is numerical but it is shown as object type so we use type conversion to convert it into integer type.

```
#we see 'Days_spent_hsptl' is numerical but is shown as object type. So converting its data type
data['Days_spend_hsptl'].replace(to_replace='120 +',value='121',inplace=True)
data['Days_spend_hsptl']=data['Days_spend_hsptl'].astype('int64')
```

- We use Label Encoder to convert the categorical variables into numeric.

```
#using label encoder to convert categorical variables to numerical
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
data['Gender']=encoder.fit_transform(data['Gender'])
data['Surg_Description']=encoder.fit_transform(data['Surg_Description'])
data['Abortion']=encoder.fit_transform(data['Abortion'])
data['Emergency dept_yes/No']=encoder.fit_transform(data['Emergency dept_yes/No'])
data['Admission_type']=encoder.fit_transform(data['Admission_type'])
data['Home or self care,']=encoder.fit_transform(data['Home or self care,'])
data['Cultural_group']=encoder.fit_transform(data['Cultural_group'])
data['ethnicity']=encoder.fit_transform(data['ethnicity'])
data['Age']=encoder.fit_transform(data['Age'])
data['apr_drg_description']=encoder.fit_transform(data['apr_drg_description'])
data['Hospital County']=encoder.fit_transform(data['Hospital County'])
data['Area_Service']=encoder.fit_transform(data['Area_Service'])
```

#	Column	Non-Null Count	Dtype
0	Area_Service	1048575 non-null	int32
1	hospital_county	1048575 non-null	int32
2	hospital_id	1048575 non-null	float64
3	Age	1048575 non-null	int32
4	Gender	1048575 non-null	int32
5	Cultural_group	1048575 non-null	int32
6	ethnicity	1048575 non-null	int32
7	Days_spend_hsptl	1048575 non-null	object
8	Admission_type	1048575 non-null	int32
9	home_self_care	1048575 non-null	int32
10	ccs_diagnosis_code	1048575 non-null	int64
11	ccs_procedure_code	1048575 non-null	int64
12	apr_drg_description	1048575 non-null	int32
13	Code_illness	1048575 non-null	int64
14	mortality_risk	1048575 non-null	float64
15	Surg_Description	1048575 non-null	int32
16	Weight_baby	1048575 non-null	int64
17	Abortion	1048575 non-null	int32
18	emergency_dept_yes_no	1048575 non-null	int32
19	Tot_charg	1048575 non-null	float64
20	Tot_cost	1048575 non-null	float64
21	ratio_of_total_costs_to_total_charges	1048575 non-null	float64
22	Result	1048575 non-null	int64
23	Payment_Typology	1048575 non-null	int64

Check whether data is Balanced or Imbalanced

```
Y.value_counts()
```

```
1    3876
0    1343
Name: Result, dtype: int64
```

From the data we came to know that data is containing 75% of Genuine (1) data and 25% of Fraudulent (0) data.

This imbalance may create biasness in the model towards Genuine data, So Over Sampling is done to balance the data.

```
Counter({1: 782188, 0: 261573})
Counter({1: 782188, 0: 782188})
```

```python
from imblearn.over_sampling import RandomOverSampler
os =  RandomOverSampler(sampling_strategy='minority')
X_over, Y_over = os.fit_resample(X, Y)
X_over.shape,Y_over.shape
```

```
((1564376, 17), (1564376,))
```

```python
from collections import Counter
print(Counter(Y))
print(Counter(Y_over))
```

```
Counter({1: 782188, 0: 261573})
Counter({1: 782188, 0: 782188})
```

EDA and Feature Engineering Summary

i. The data set were checked for null values and imputation was used to eliminate them correspondingly.

ii. Different visualizations were plotted to get a deep insight into the data.

iii. Label Encoder was used on the categorical columns to convert them into numeric.

iv. Extra trees classifier was used to get the feature scores and the features which had little impact on the model building were eliminated.

v. Random over sampling was used to create the balance in the result column.

MODEL BUILDING

Comparing accuracies of different models

Model Name	Training Accuracy	Testing Accuracy
Decision Tree with Entropy Criteria	99.06%	99.08%
Decision Tree with Gini Criteria	99.06%	99.06
Logistic Regression	74.99%	74.99%
Ada Boost	50%	55%
Random Forest with under sampling	50.75%	48.33%
Random Forest with over sampling	90.5%	70.83%

Building the Decision Tree Classifier with over sampling as it gave the highest accuracy

```
from sklearn.tree import  DecisionTreeClassifier
model2 = DecisionTreeClassifier(criterion = 'gini',random_state=3)
model2.fit(X_over,Y_over)
```

executed in 12.4s, finished 19:05:32 2021-05-11

```
DecisionTreeClassifier(random_state=3)
```

```
y_pred1 = model2.predict(x_test)
y_pred1
```

executed in 109ms, finished 19:05:35 2021-05-11

```
array([1, 1, 0, ..., 0, 1, 0], dtype=int64)
```

```
train_acc=round(model2.score(x_train, y_train) * 100, 2)
train_acc
```

executed in 391ms, finished 19:05:39 2021-05-11

```
99.97
```

```
from sklearn.metrics import accuracy_score
test_acc=round(accuracy_score(y_test,y_pred1)*100,2)
test_acc
```

executed in 24ms, finished 19:06:21 2021-05-11

```
99.98
```

ROC Score

```
# Plotting of AUC
import matplotlib.pyplot as plt
plt.plot(fpr, tpr, color='red', label='logit model ( area  = %0.2f)'%auc)
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
```

executed in 1.15s, finished 15:10:56 2021-05-11

```
Text(0, 0.5, 'True Positive Rate')
```