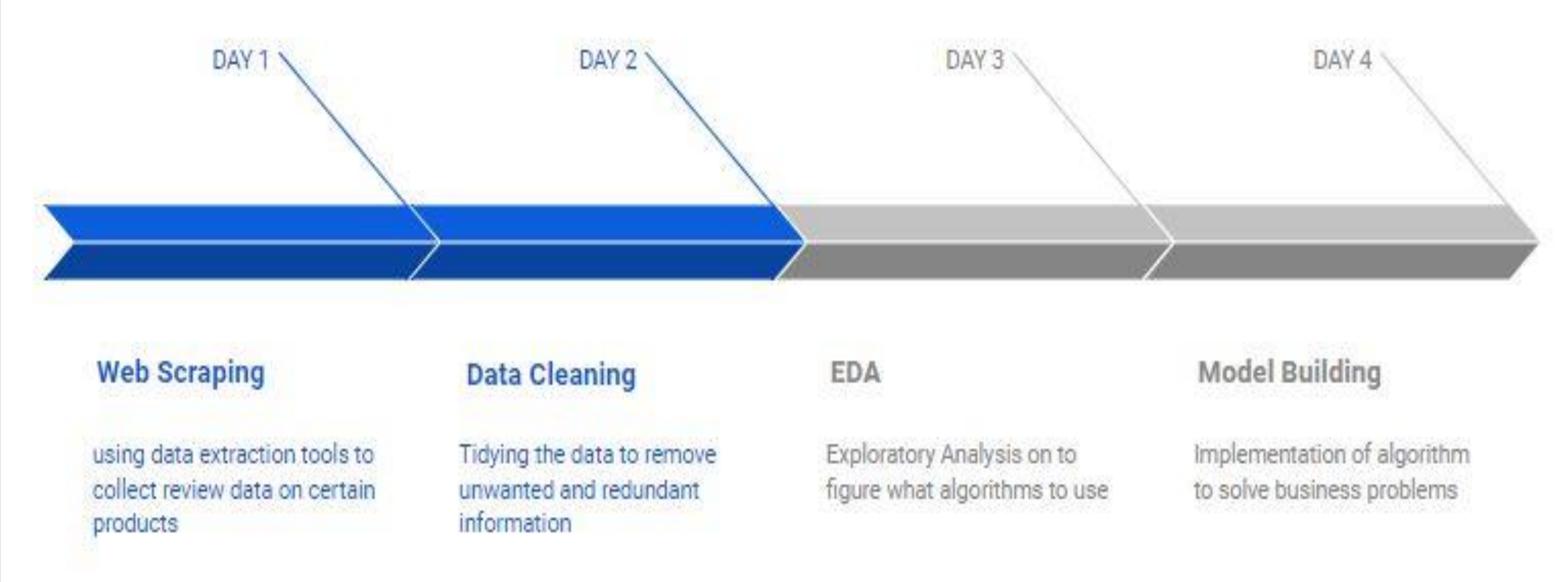
FRAUD ANALYTICS

By:-
NAVEEN
SURAJ

BUSINESS OBJECTIVE

To Predict whether the customer claim is Genuine or Fraud

PROJECT ARCHITECTURE/FLOW



DAY 1

Web Scraping

using data extraction tools to collect review data on certain products

DAY 2

Data Cleaning

Tidying the data to remove unwanted and redundant information

DAY 3

EDA

Exploratory Analysis on to figure what algorithms to use

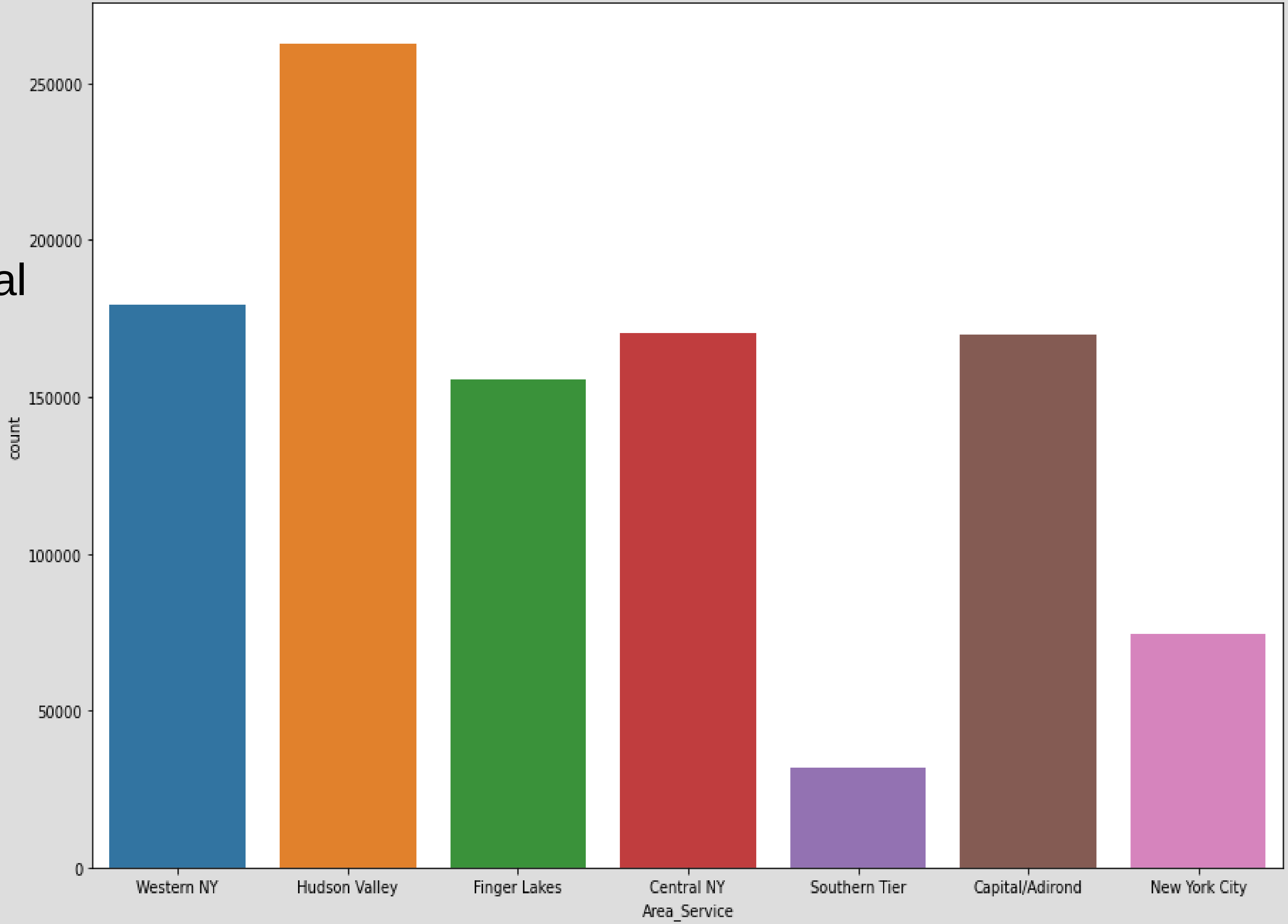
DAY 4

Model Building

Implementation of algorithm to solve business problems

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.

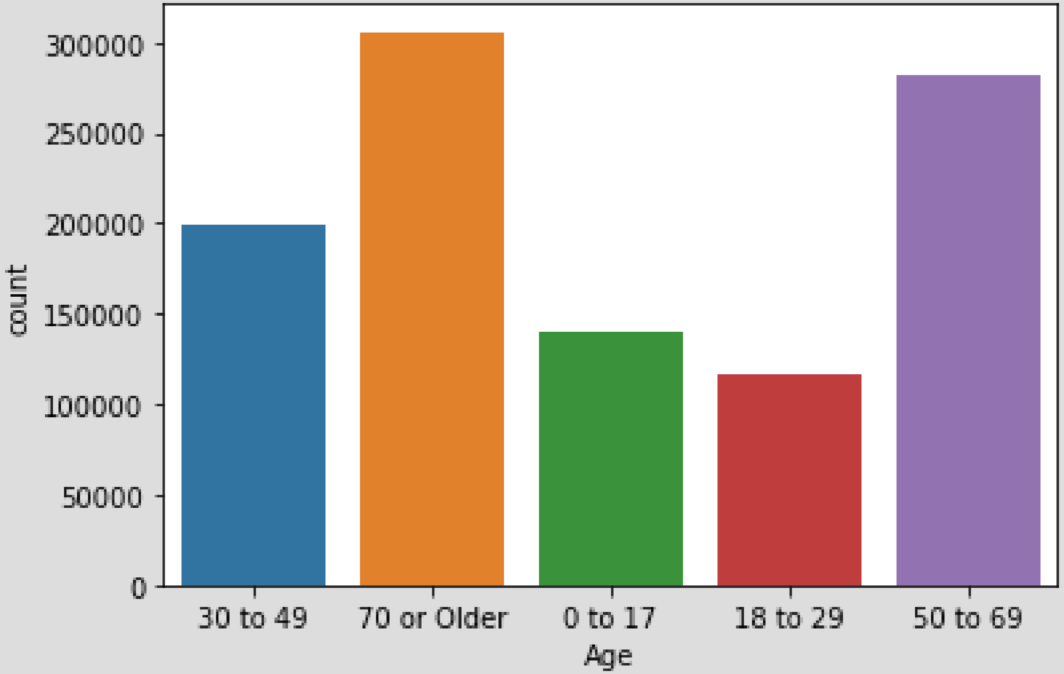
From this plot, we observe highest peak (count) in Hudson Valley and the lowest in the Southern tier.



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

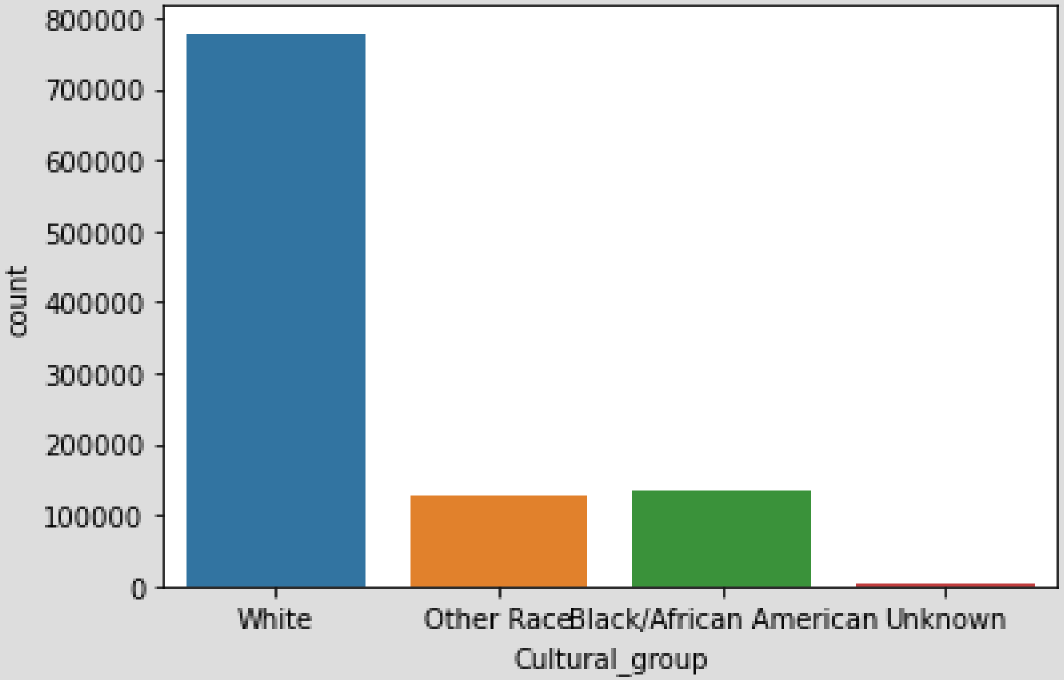
We can see that most of the patients are of age group 70 or older followed by group 50 to 69. The minimum amount of patients are in group 0 to 17, followed by group 18 to 29.

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



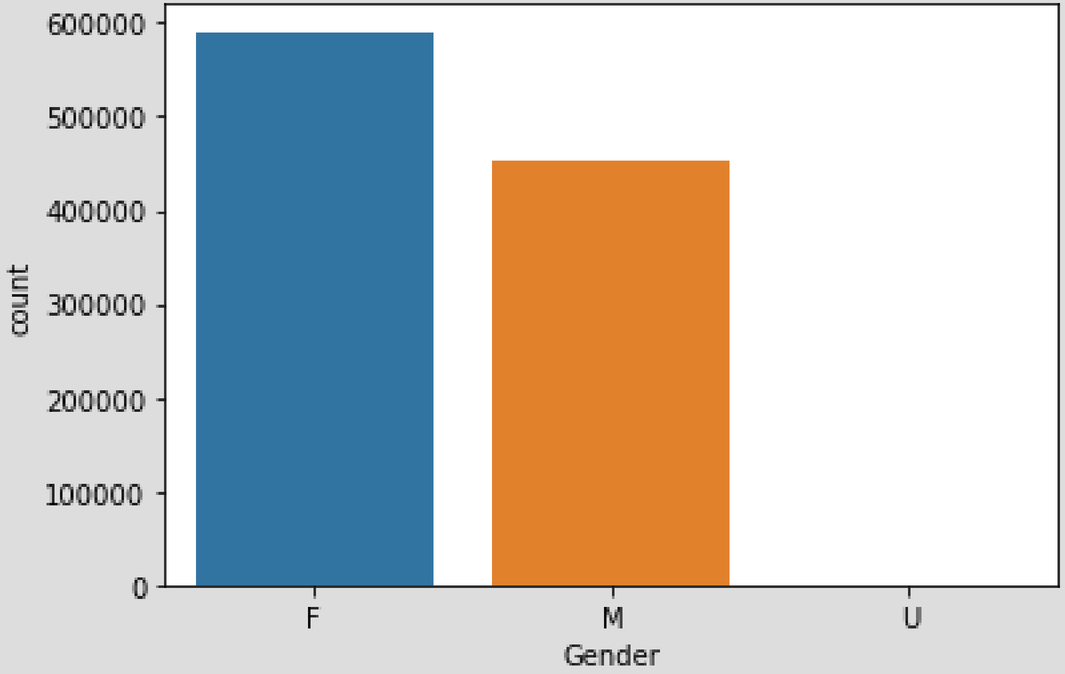
This is a count plot for the cultural_group column. Black depicts African Americans, White depicts Americans and the Other Race is the group other than these two.

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.

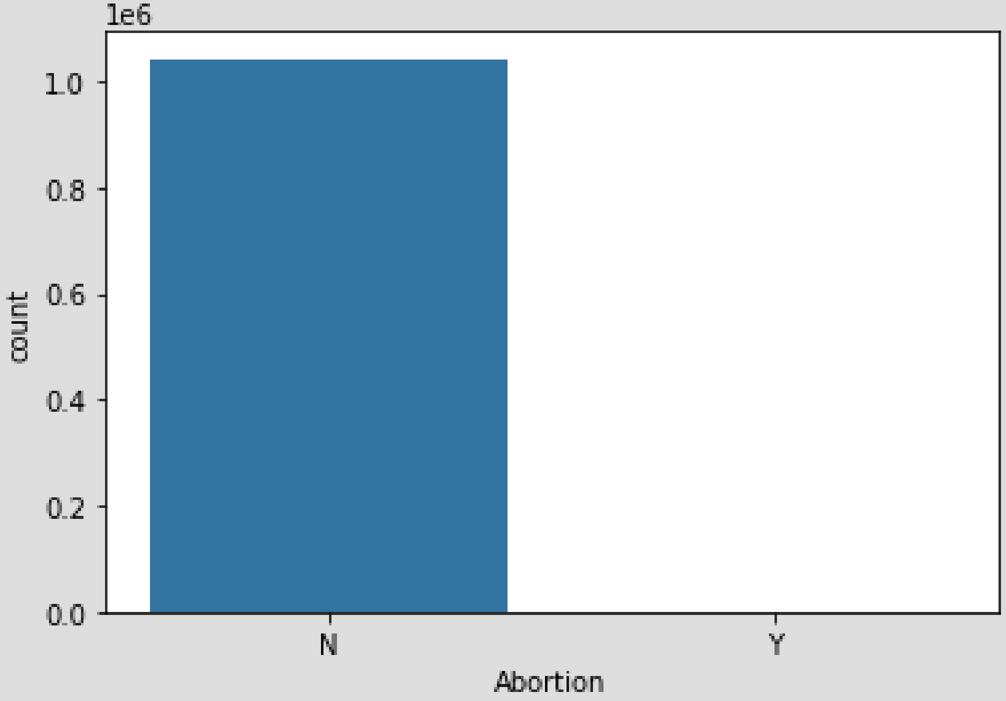
We can see that female patients are more in number than the male patients.



Here is a count plot for the abortion column.

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

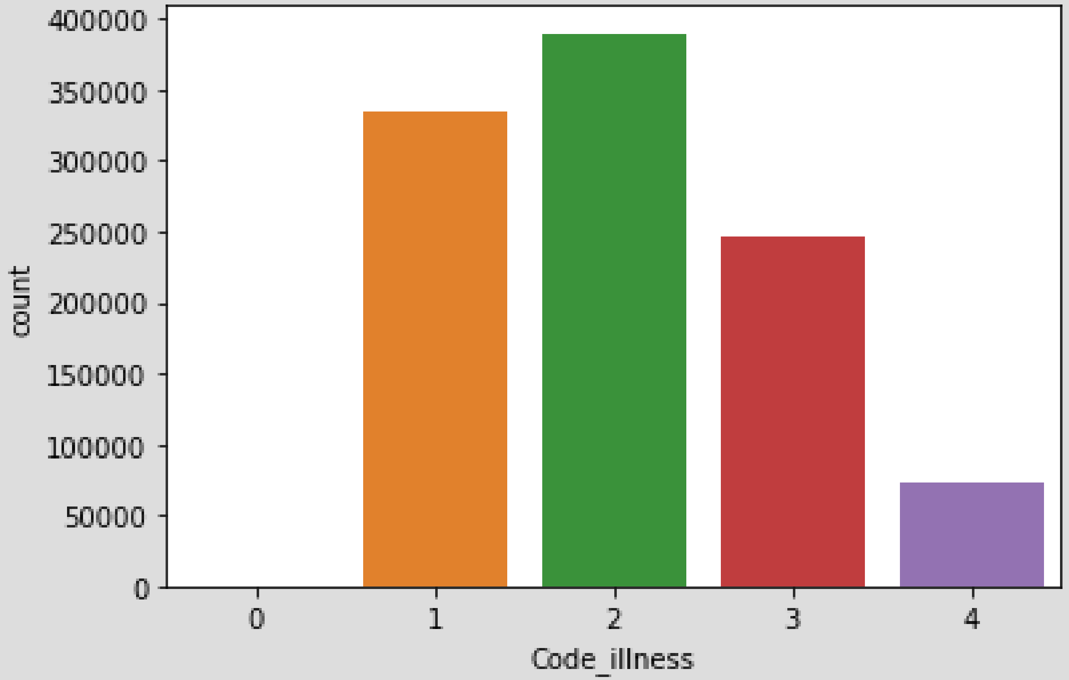
This depicts that there is an imbalance in the data which could create a bias in the model building.



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.

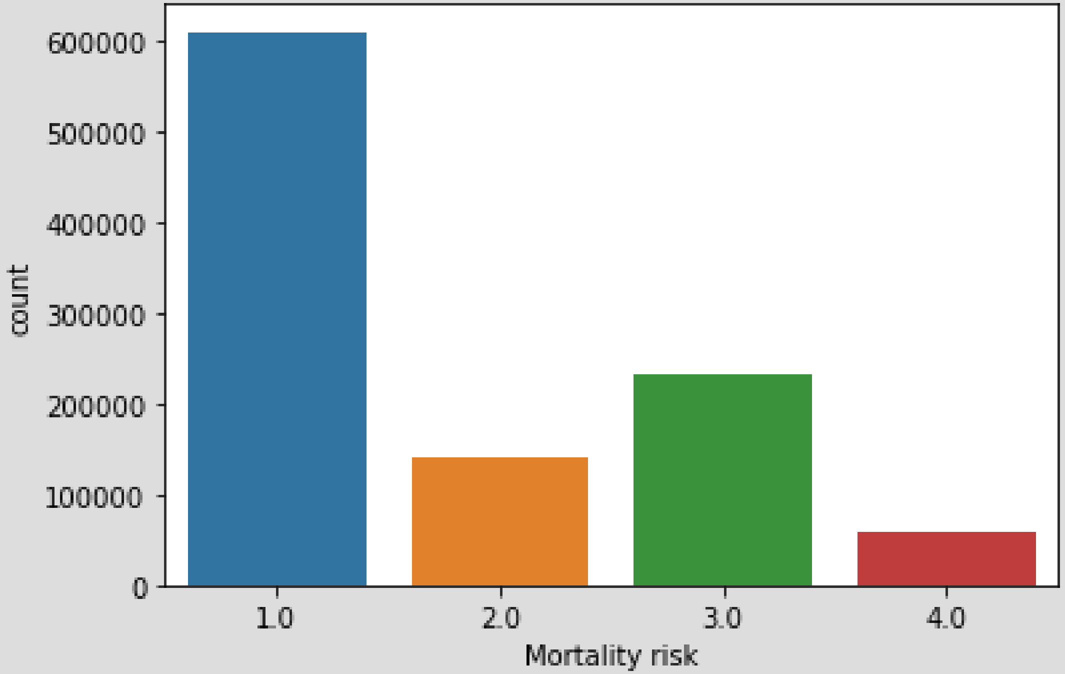
We can see that most number of patients have moderate illness followed by mild illness & severe.

The least number lies in indeterminate illness group.



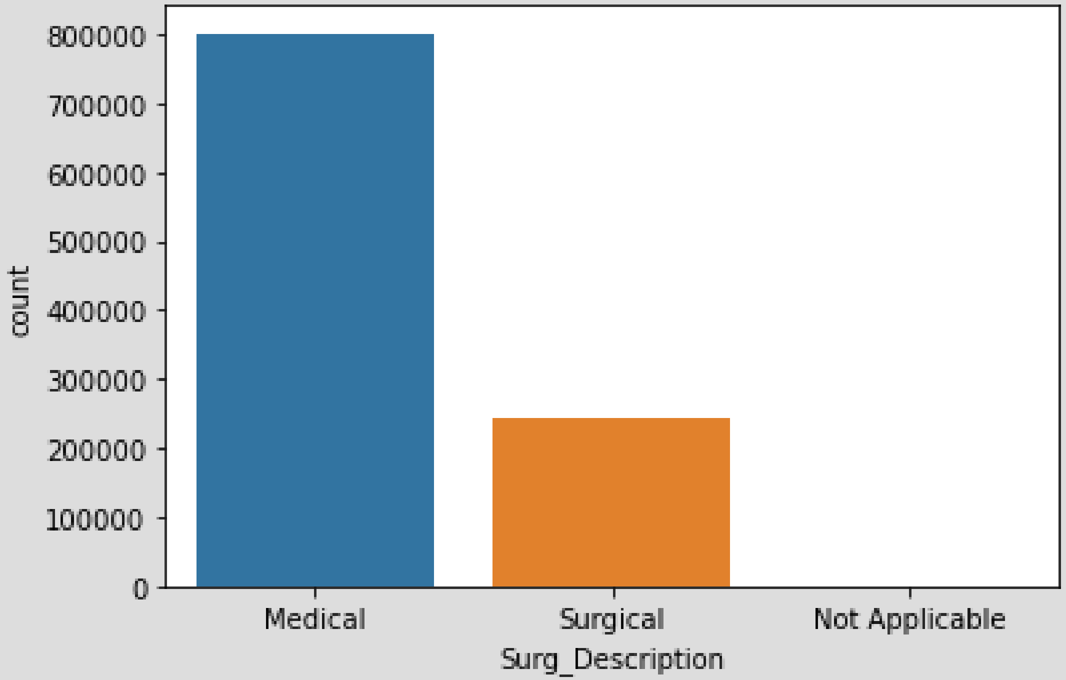
Here is a count plot for the mortality risk column which shows the mortality risk for the patients. 1 is for minor risk, 2 for moderate, 3 for major and 4 for severe.

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



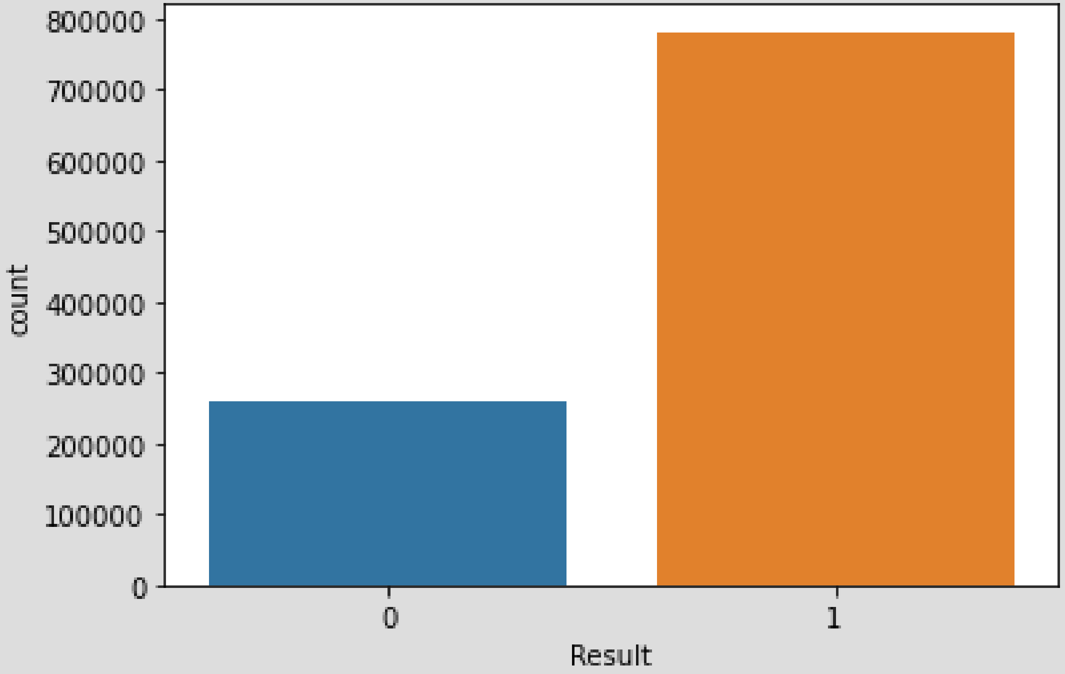
This is the count plot for the Surg_Description column which shows the different types of treatment provided to patients.

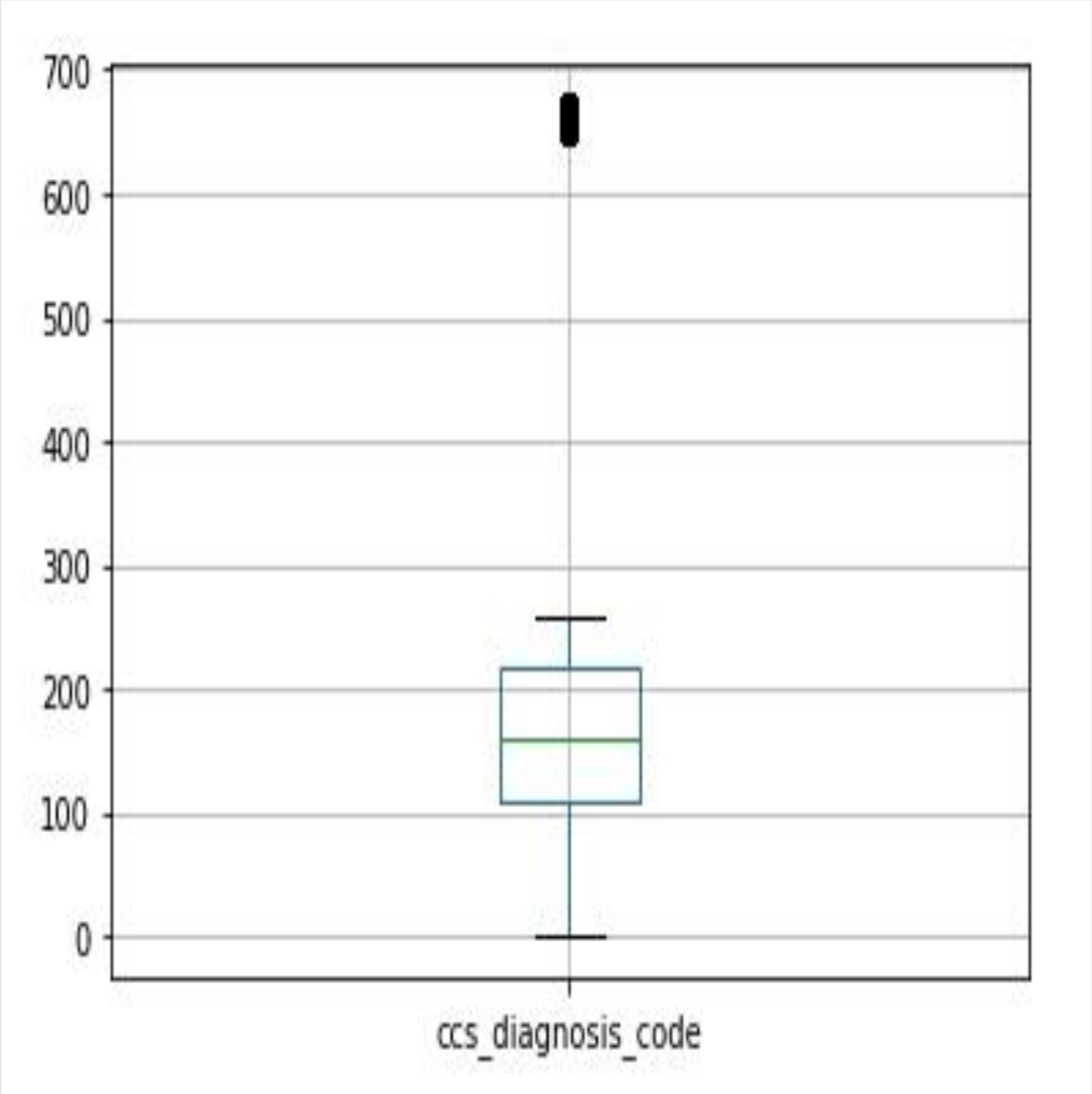
We can see that more amount of patients were given medical treatment as compared to surgical treatment.

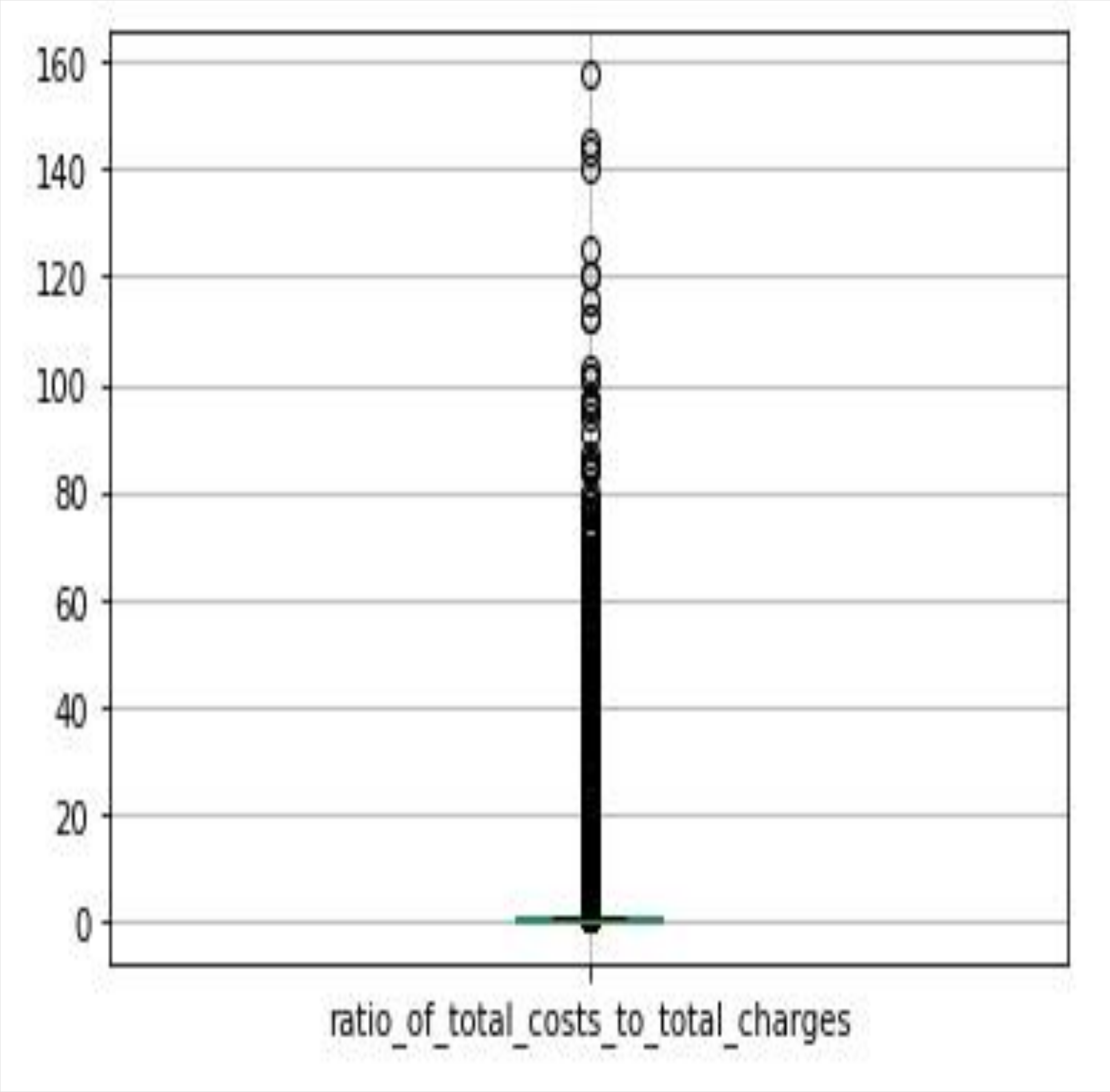


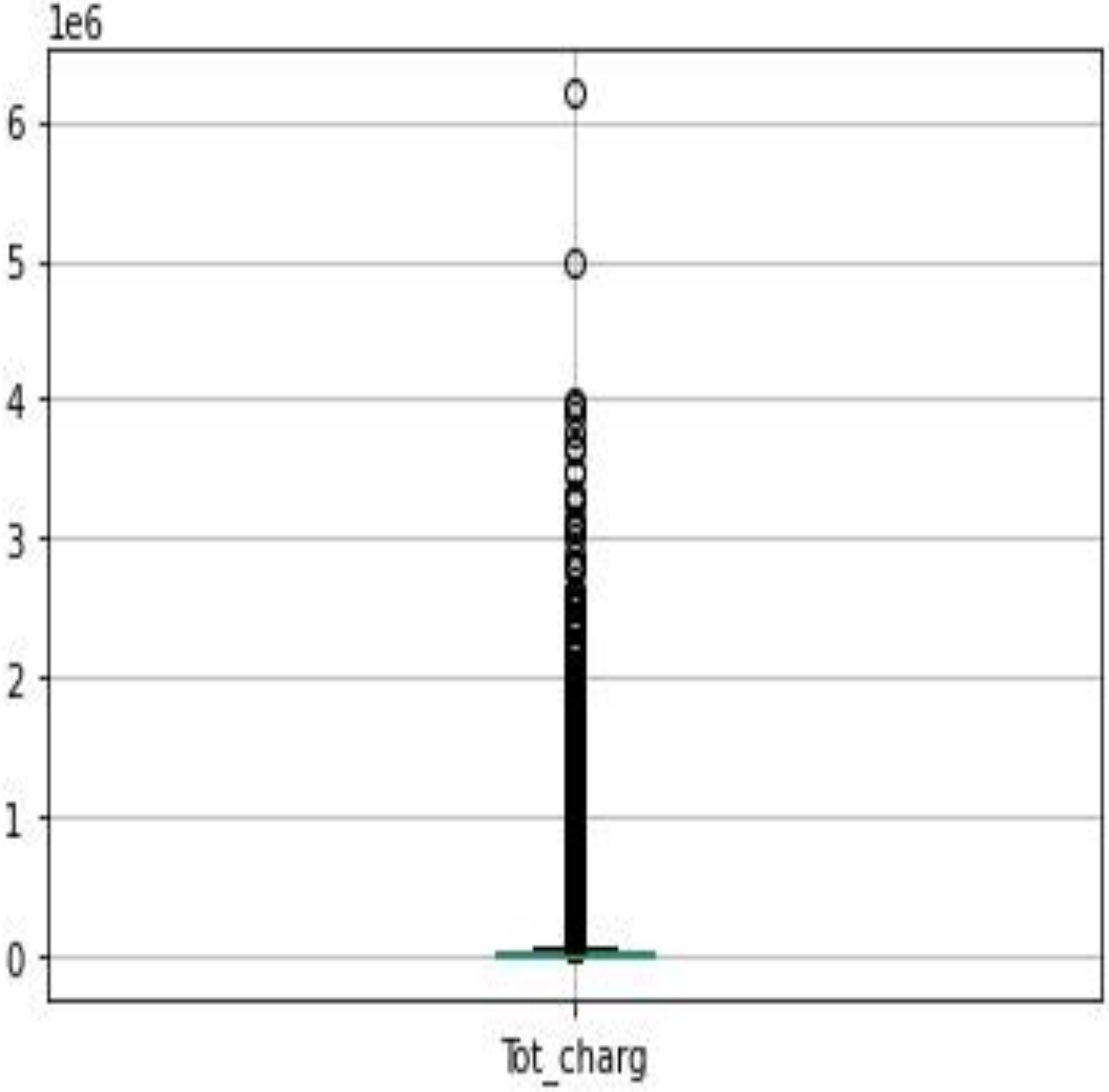
Here is a count plot for the result column. Here 1 shows the count of genuine claims and 0 shows the count of fraud claims.

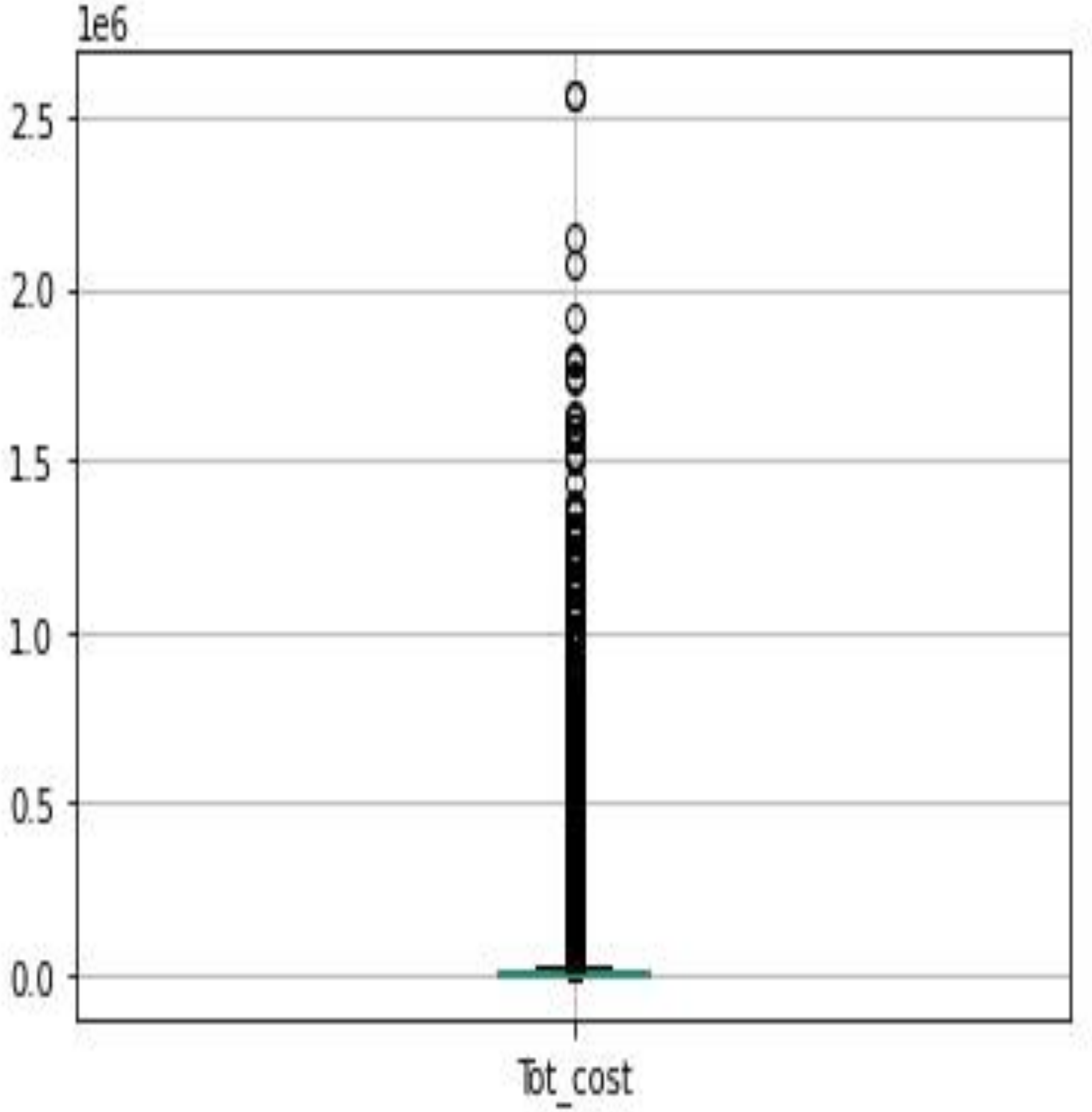
We can clearly see the imbalance in the result as the count of genuine claims is much greater than the count of fraud claims.

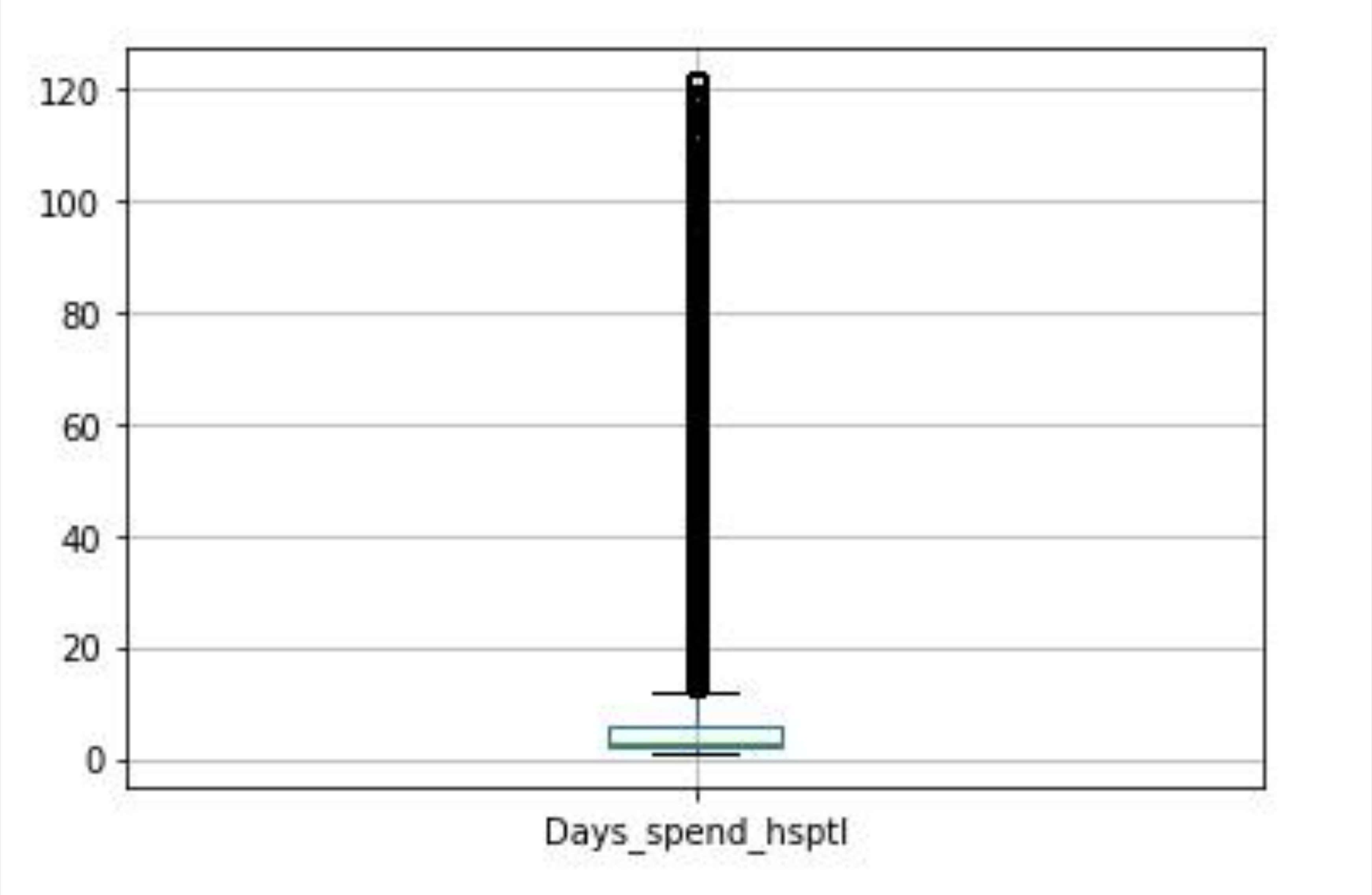


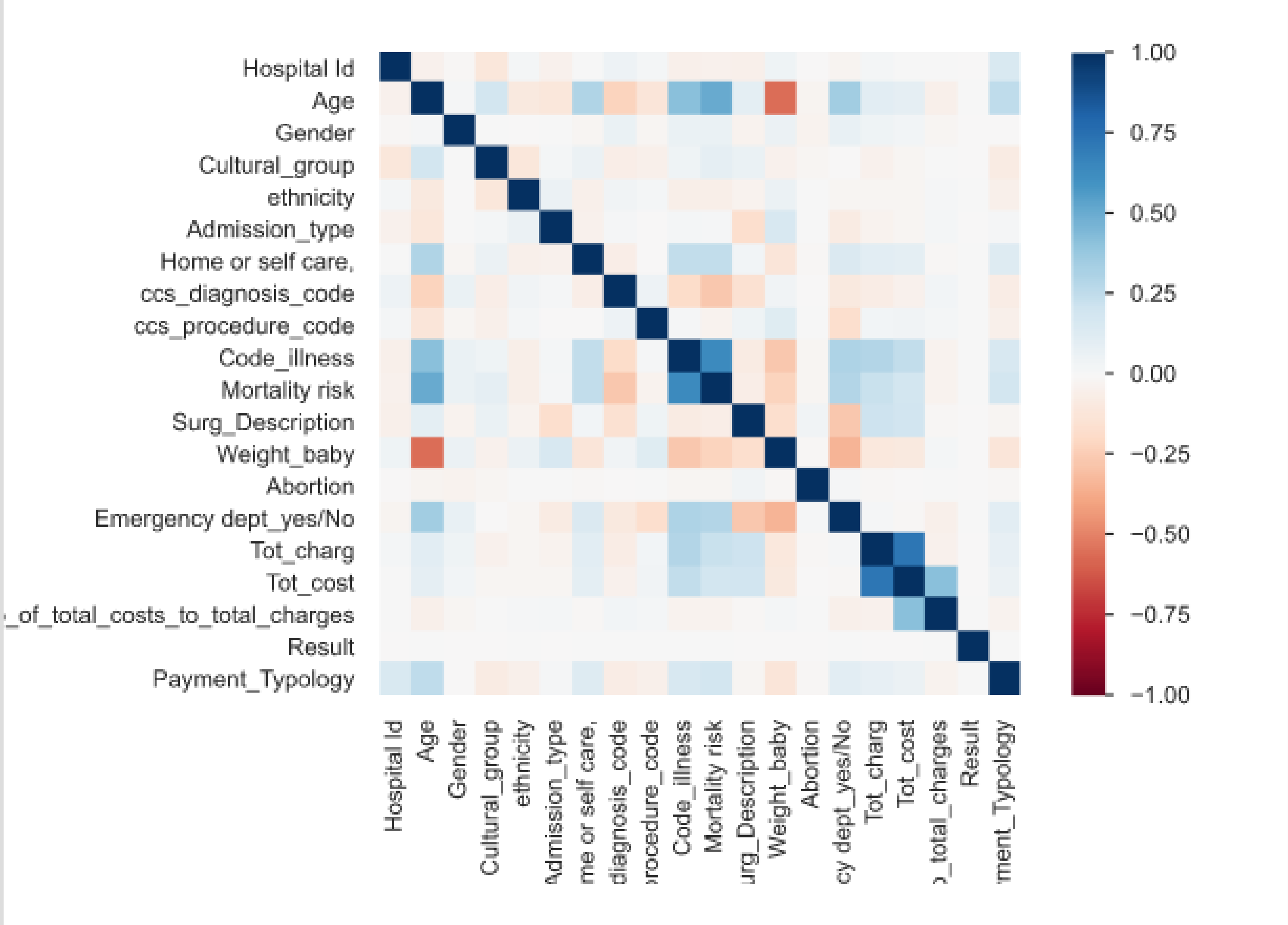

700
600
500
400
300
200
100
0
ccs_diagnosis_code


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

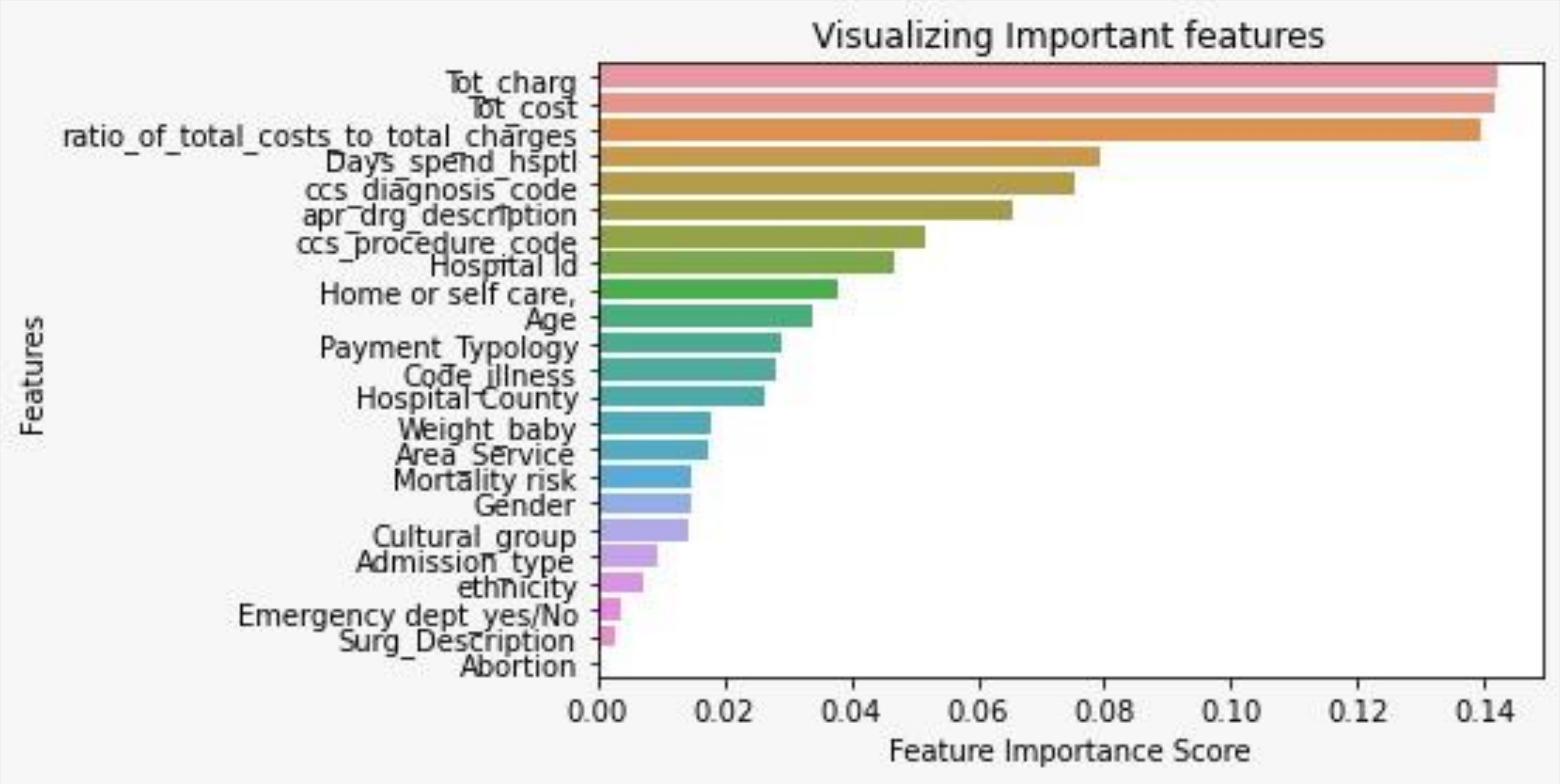

1e6
2.5
2.0
1.5
1.0
0.5
0.0
Tot_cost


120
100
80
60
40
20
0
Days_spend_hsptl

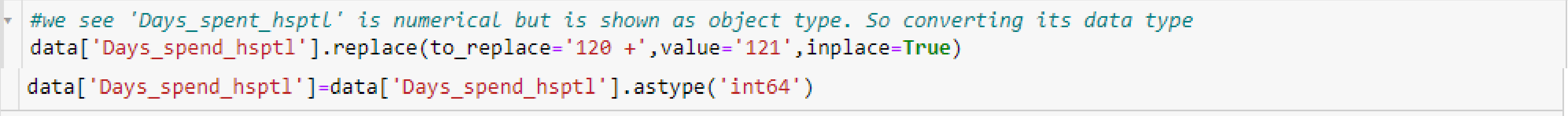


Heatmap of Correlation

FEATURE ENGINEERING



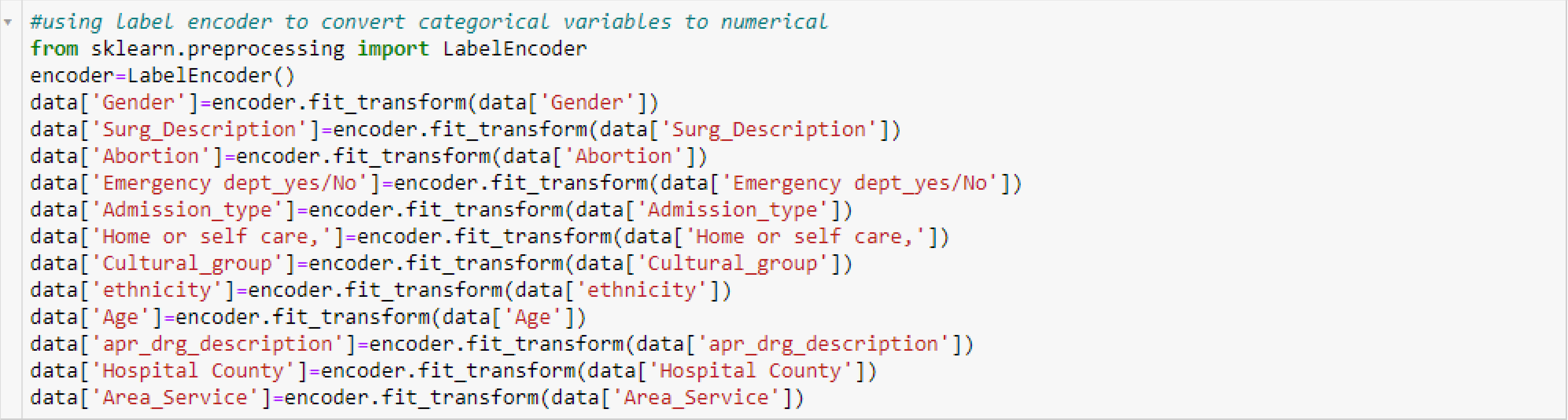
- The observation shows that the Days_spent_hsptl column is numerical but it is shown as object type so we use type conversion to convert it into integer type.



```
#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)
```

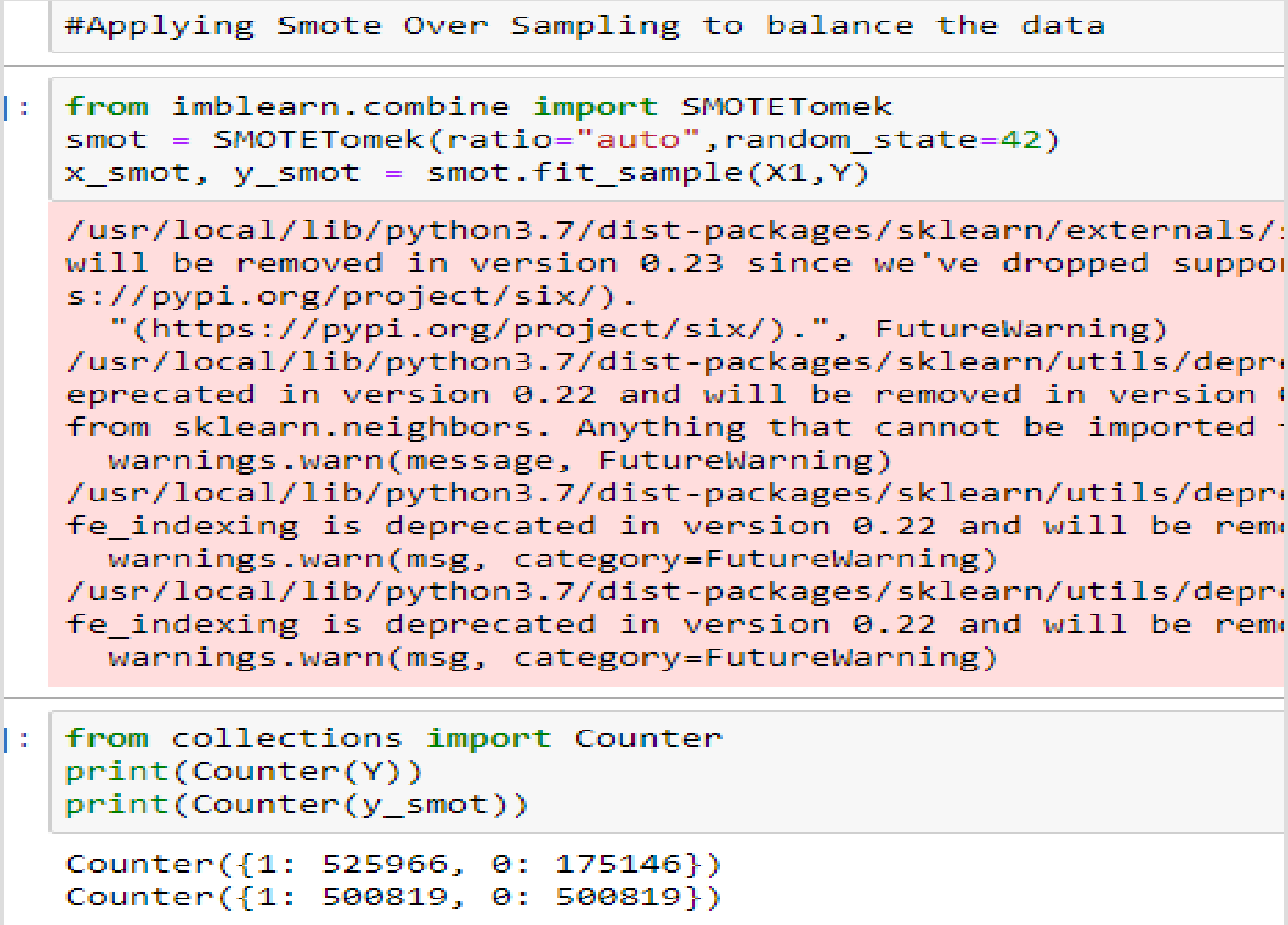
```
data['Days_spend_hsptl']=data['Days_spend_hsptl'].astype('int64')
```

- 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'])
```

We have used smote over sampling to balance the data..



```
#Applying Smote Over Sampling to balance the data
```

```
from imblearn.combine import SMOTETomek
smot = SMOTETomek(ratio="auto",random_state=42)
x_smot, y_smot = smot.fit_sample(X1,Y)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/externals/
will be removed in version 0.23 since we've dropped suppo
s://pypi.org/project/six/).
  "(https://pypi.org/project/six/).", FutureWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/depr
eprecated in version 0.22 and will be removed in version
from sklearn.neighbors. Anything that cannot be imported
  warnings.warn(message, FutureWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/depr
fe_indexing is deprecated in version 0.22 and will be rem
  warnings.warn(msg, category=FutureWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/depr
fe_indexing is deprecated in version 0.22 and will be rem
  warnings.warn(msg, category=FutureWarning)
```

```
from collections import Counter
print(Counter(Y))
print(Counter(y_smot))
```

```
Counter({1: 525966, 0: 175146})
Counter({1: 500819, 0: 500819})
```

EDA and Feature Engineering Summary

i. Different visualizations were plotted with the help of seaborn and matplotlib libraries to get a deep insight into the data.

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

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

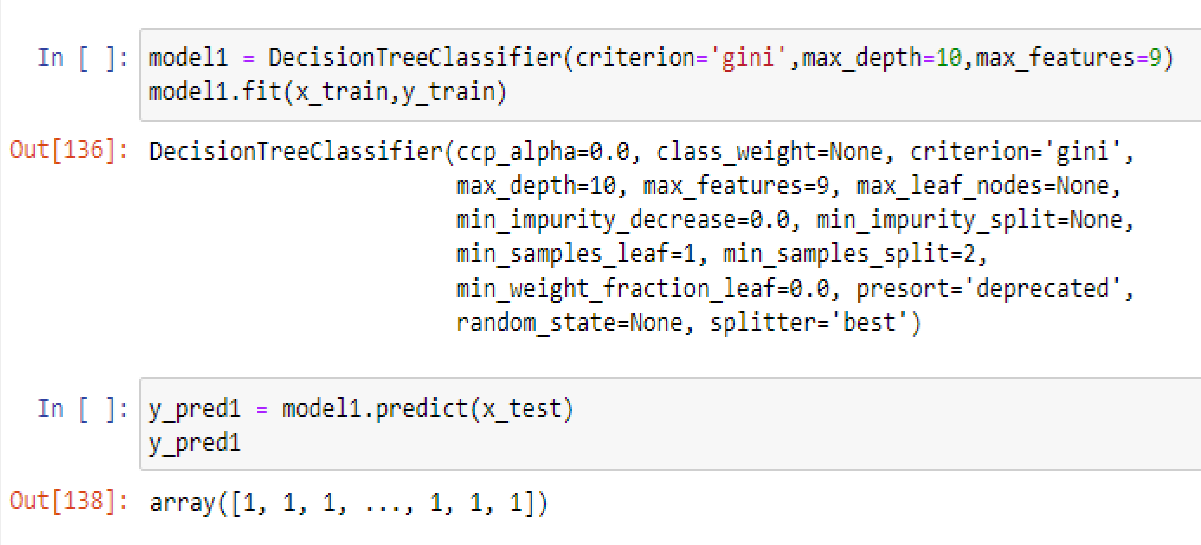
iv. Smote oversampling technique was used to create the balance in the result column.

MODEL BUILDING

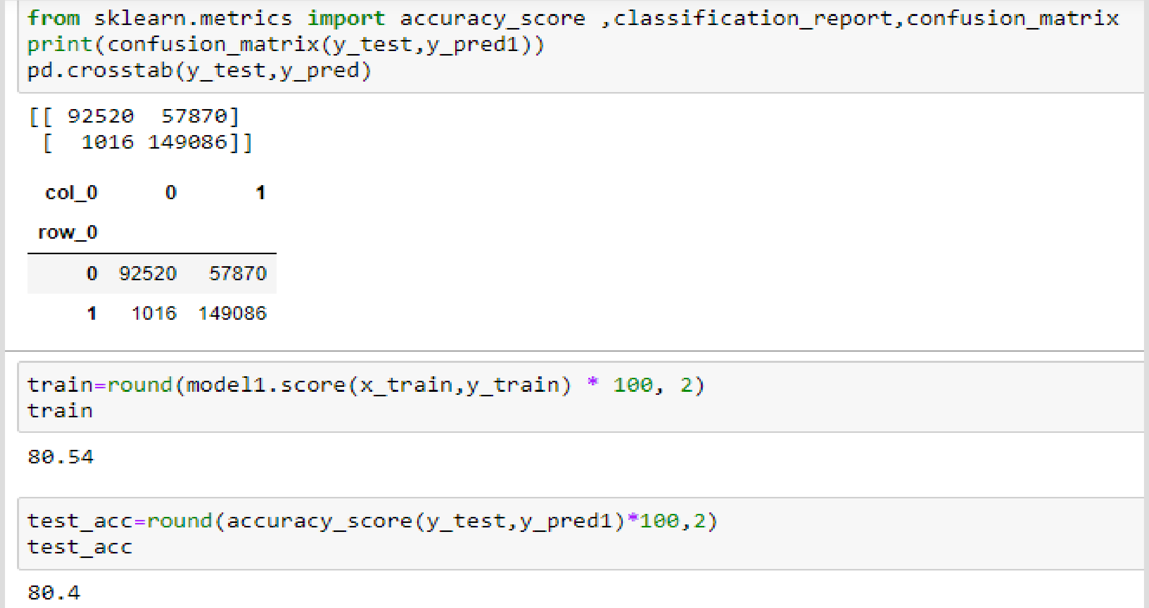
Comparing accuracies of different models

Algorithm Name	Training Accuracy	Testing Accuracy
Decision Tree Classifier	80.4%	80.4%
Random Forest	99.06%	99.06
Logistic Regression	50.8%	80.4%
Ada Boost	84.1%	81.2%
XG Boost	83.39%	82.29%

Model Building using Decision Tree

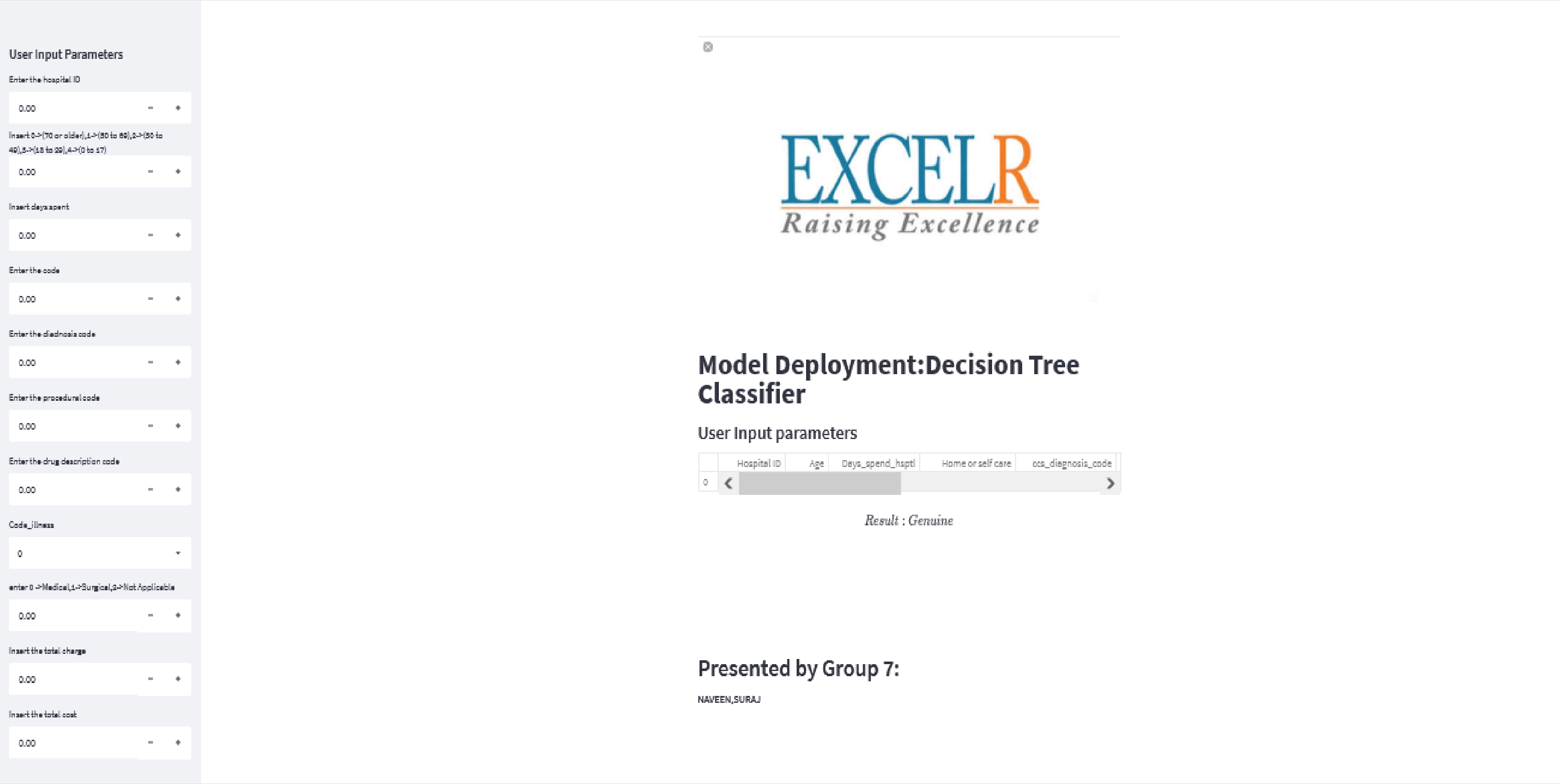


```
In [ ]: model1 = DecisionTreeClassifier(criterion='gini',max_depth=10,max_features=9)
        model1.fit(x_train,y_train)

Out[136]: DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                                 max_depth=10, max_features=9, max_leaf_nodes=None,
                                 min_impurity_decrease=0.0, min_impurity_split=None,
                                 min_samples_leaf=1, min_samples_split=2,
                                 min_weight_fraction_leaf=0.0, presort='deprecated',
                                 random_state=None, splitter='best')

In [ ]: y_pred1 = model1.predict(x_test)
        y_pred1

Out[138]: array([1, 1, 1, ..., 1, 1, 1])
```



```
from sklearn.metrics import accuracy_score ,classification_report,confusion_matrix
print(confusion_matrix(y_test,y_pred1))
pd.crosstab(y_test,y_pred)

[[ 92520  57870]
 [  1016 149086]]

 col_0      0       1
 row_0
     0  92520   57870
     1   1016  149086

train=round(model1.score(x_train,y_train) * 100, 2)
train

80.54

test_acc=round(accuracy_score(y_test,y_pred1)*100,2)
test_acc

80.4
```

Model Deployment



User Input Parameters

Enter the hospital ID

0.00 − +

Insert 0->(70 or older),1->(50 to 69),2->(30 to 49),3->(18 to 29),4->(0 to 17)

0.00 − +

Insert days spent

0.00 − +

Enter the code

0.00 − +

Enter the diadnosis code

0.00 − +

Enter the procedural code

0.00 − +

Enter the drug description code

0.00 − +

Code_illness

0

enter 0 ->Medical,1->Surgical,2->Not Applicable

0.00 − +

Insert the total charge

0.00 − +

Insert the total cost

0.00 − +

Model Deployment:Decision Tree Classifier

User Input parameters

	Hospital ID	Age	Days_spend_hsptl	Home or self care	ccs_diagnosis_code
0					

Result : Genuine

Presented by Group 7:

NAVEEN,SURAJ