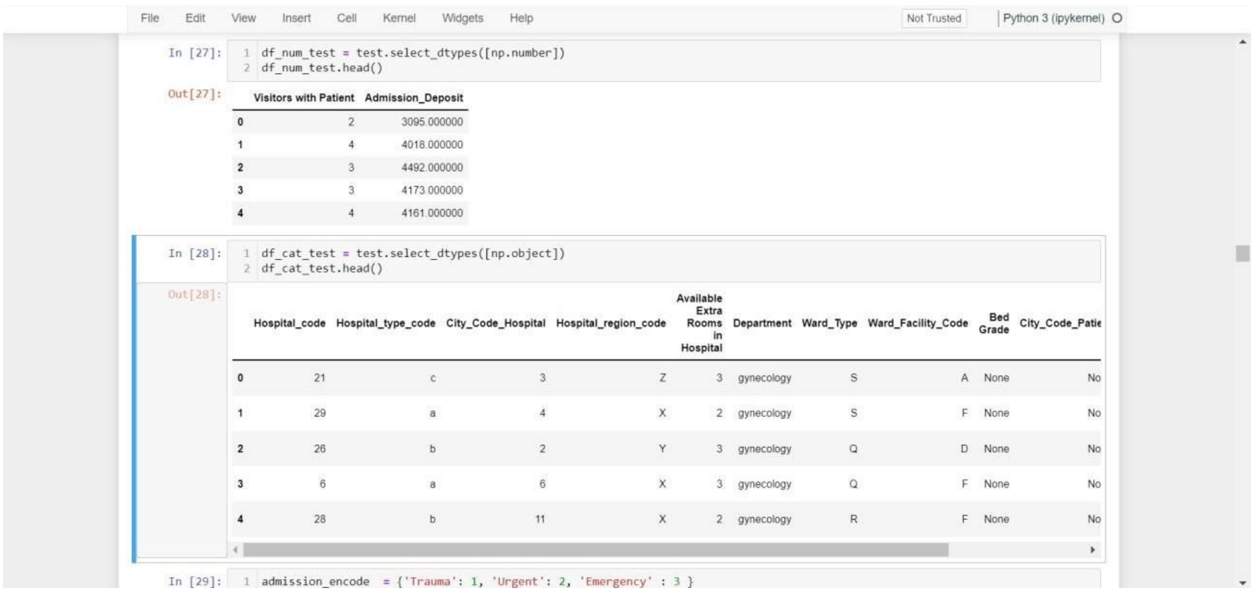
ANALYTICS FOR HOSPITALAND HEALTHCARE DATA

TEAM ID : **PNT2022TMID16345**

DATA ANALYSIS



In [27]:
```
df_num_test = test.select_dtypes([np.number])
df_num_test.head()
```

Out[27]:

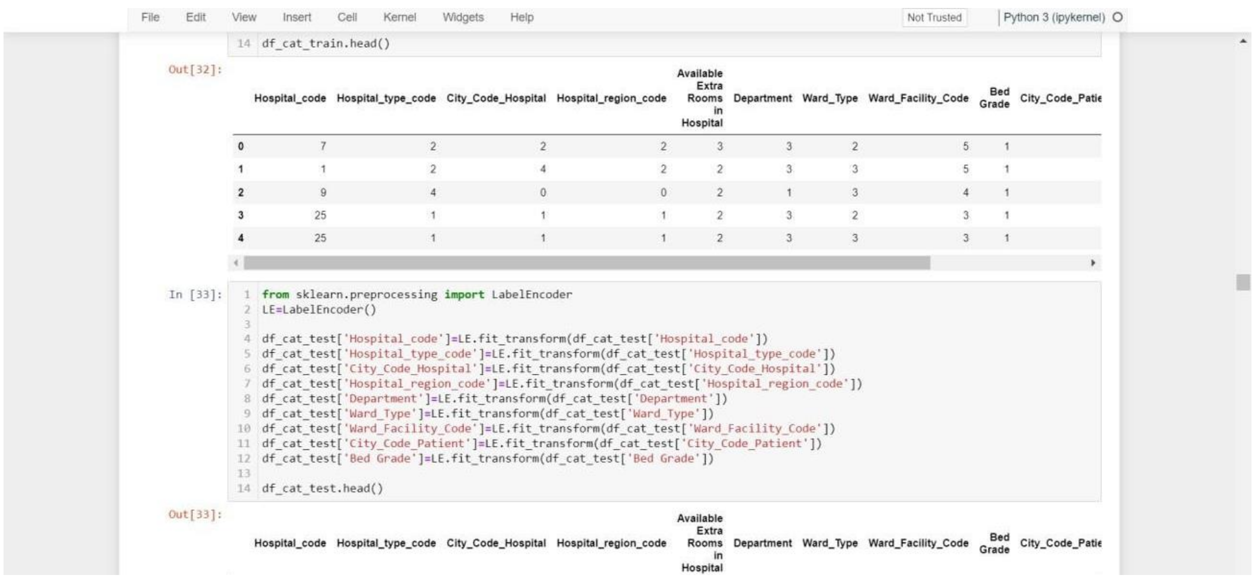
	Visitors with Patient	Admission_Deposit
0	2	3095.000000
1	4	4018.000000
2	3	4492.000000
3	3	4173.000000
4	4	4161.000000

In [28]:
```
df_cat_test = test.select_dtypes([np.object])
df_cat_test.head()
```

Out[28]:

	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Available Extra Rooms in Hospital	Department	Ward_Type	Ward_Facility_Code	Bed Grade	City_Code_Patie
0	21	c	3	Z	3	gynecology	S	A	None	No
1	29	a	4	X	2	gynecology	S	F	None	No
2	26	b	2	Y	3	gynecology	Q	D	None	No
3	6	a	6	X	3	gynecology	Q	F	None	No
4	28	b	11	X	2	gynecology	R	F	None	No

In [29]:
```
admission_encode  = {'Trauma': 1, 'Urgent': 2, 'Emergency' : 3 }
```



```
df_cat_train.head()
```

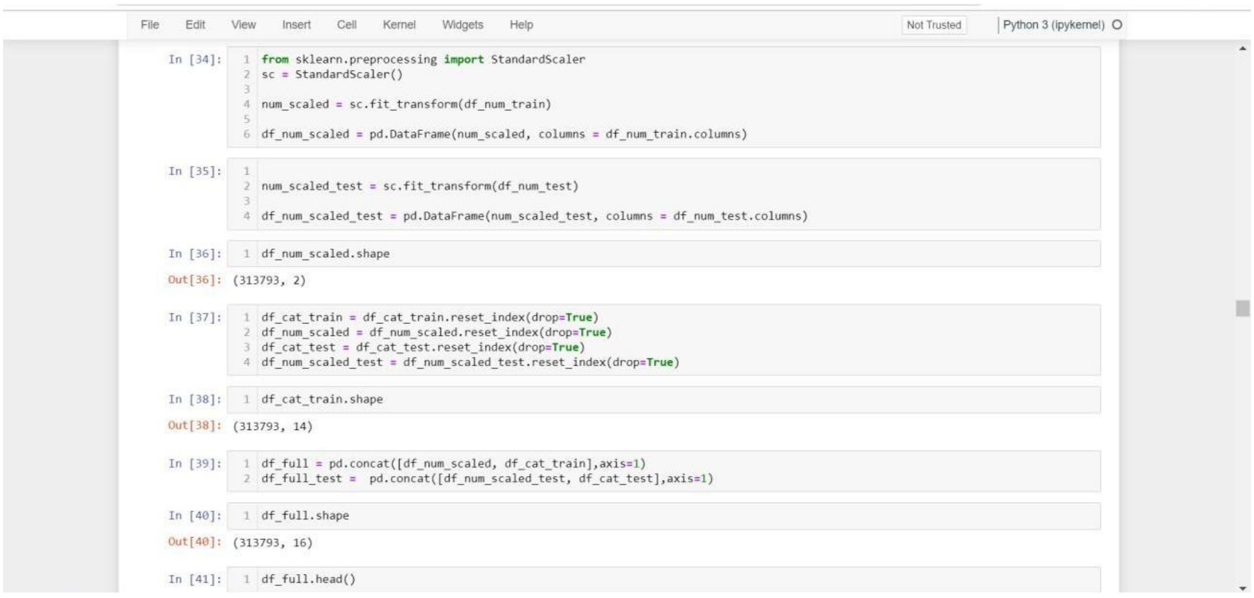
Out[32]:

	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Available Extra Rooms in Hospital	Department	Ward_Type	Ward_Facility_Code	Bed Grade	City_Code_Patie
0	7	2	2	2	3	3	2	5	1	
1	1	2	4	2	2	3	3	5	1	
2	9	4	0	0	2	1	3	4	1	
3	25	1	1	1	2	3	2	3	1	
4	25	1	1	1	2	3	3	3	1	

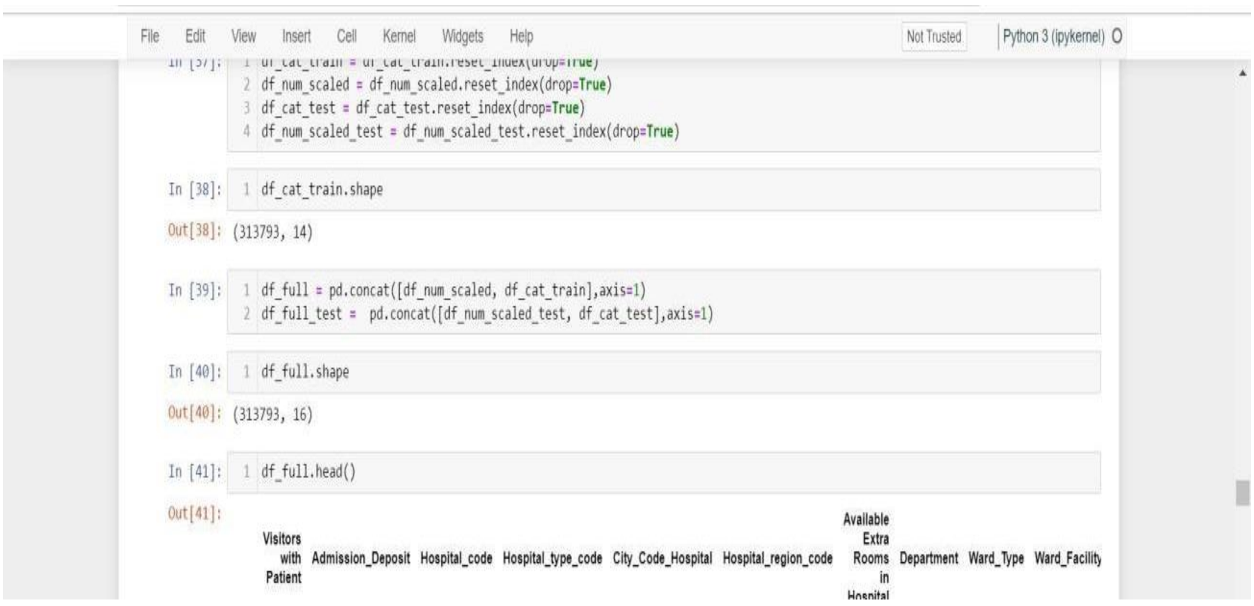
In [33]:
```
from sklearn.preprocessing import LabelEncoder
LE=LabelEncoder()

df_cat_test['Hospital_code']=LE.fit_transform(df_cat_test['Hospital_code'])
df_cat_test['Hospital_type_code']=LE.fit_transform(df_cat_test['Hospital_type_code'])
df_cat_test['City_Code_Hospital']=LE.fit_transform(df_cat_test['City_Code_Hospital'])
df_cat_test['Hospital_region_code']=LE.fit_transform(df_cat_test['Hospital_region_code'])
df_cat_test['Department']=LE.fit_transform(df_cat_test['Department'])
df_cat_test['Ward_Type']=LE.fit_transform(df_cat_test['Ward_Type'])
df_cat_test['Ward_Facility_Code']=LE.fit_transform(df_cat_test['Ward_Facility_Code'])
df_cat_test['City_Code_Patient']=LE.fit_transform(df_cat_test['City_Code_Patient'])
df_cat_test['Bed Grade']=LE.fit_transform(df_cat_test['Bed Grade'])

df_cat_test.head()
```

Out[33]:

	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Available Extra Rooms in Hospital	Department	Ward_Type	Ward_Facility_Code	Bed Grade	City_Code_Patie

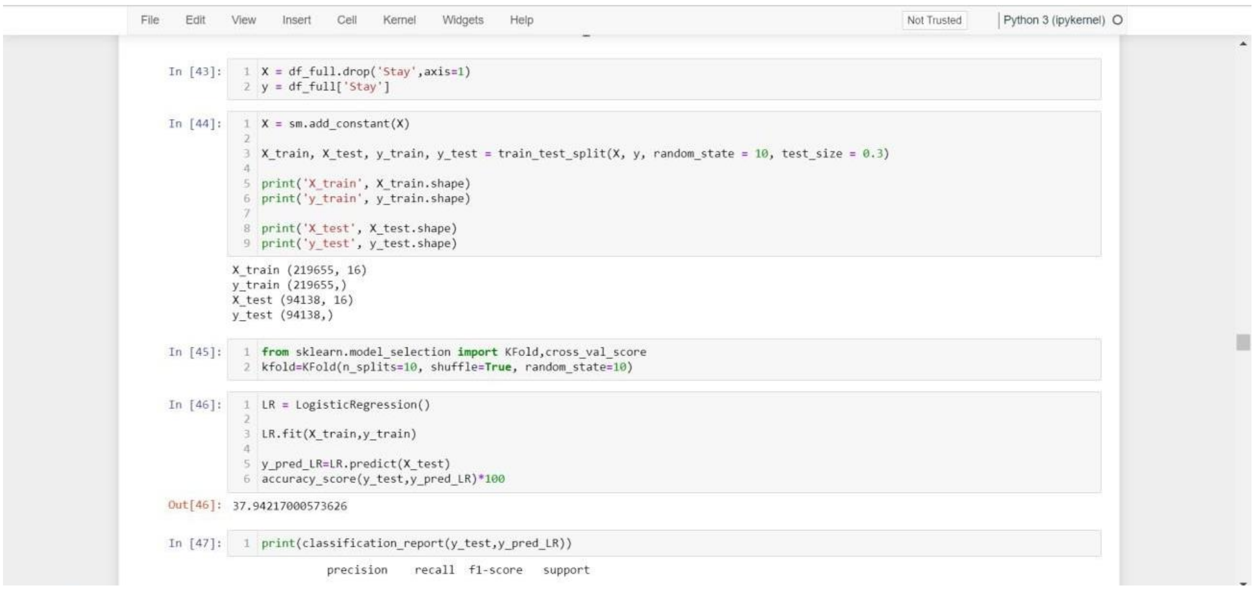


```
In [34]:  from sklearn.preprocessing import StandardScaler
          sc = StandardScaler()

          num_scaled = sc.fit_transform(df_num_train)

          df_num_scaled = pd.DataFrame(num_scaled, columns = df_num_train.columns)

In [35]:
          num_scaled_test = sc.fit_transform(df_num_test)

          df_num_scaled_test = pd.DataFrame(num_scaled_test, columns = df_num_test.columns)

In [36]:  df_num_scaled.shape

Out[36]:  (313793, 2)

In [37]:  df_cat_train = df_cat_train.reset_index(drop=True)
          df_num_scaled = df_num_scaled.reset_index(drop=True)
          df_cat_test = df_cat_test.reset_index(drop=True)
          df_num_scaled_test = df_num_scaled_test.reset_index(drop=True)

In [38]:  df_cat_train.shape

Out[38]:  (313793, 14)

In [39]:  df_full = pd.concat([df_num_scaled, df_cat_train],axis=1)
          df_full_test =  pd.concat([df_num_scaled_test, df_cat_test],axis=1)

In [40]:  df_full.shape

Out[40]:  (313793, 16)

In [41]:  df_full.head()
```



```
In [37]:  df_cat_train = df_cat_train.reset_index(drop=True)
          df_num_scaled = df_num_scaled.reset_index(drop=True)
          df_cat_test = df_cat_test.reset_index(drop=True)
          df_num_scaled_test = df_num_scaled_test.reset_index(drop=True)

In [38]:  df_cat_train.shape

Out[38]:  (313793, 14)

In [39]:  df_full = pd.concat([df_num_scaled, df_cat_train],axis=1)
          df_full_test =  pd.concat([df_num_scaled_test, df_cat_test],axis=1)

In [40]:  df_full.shape

Out[40]:  (313793, 16)

In [41]:  df_full.head()

Out[41]:
```

Visitors with Patient	Admission_Deposit	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Available Extra Rooms in Hospital	Department	Ward_Type	Ward_Facility



In [43]:
```python
X = df_full.drop('Stay',axis=1)
y = df_full['Stay']
```

In [44]:
```python
X = sm.add_constant(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 10, test_size = 0.3)

print('X_train', X_train.shape)
print('y_train', y_train.shape)

print('X_test', X_test.shape)
print('y_test', y_test.shape)
```

```
X_train (219655, 16)
y_train (219655,)
X_test (94138, 16)
y_test (94138,)
```

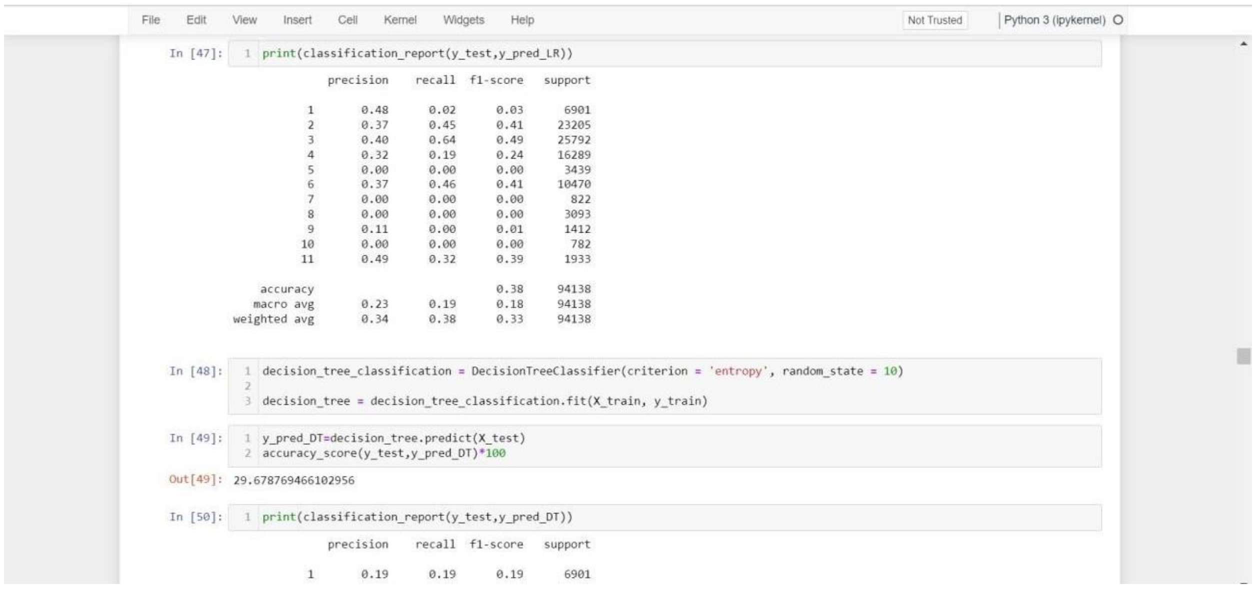
In [45]:
```python
from sklearn.model_selection import KFold,cross_val_score
kfold=KFold(n_splits=10, shuffle=True, random_state=10)
```

In [46]:
```python
LR = LogisticRegression()

LR.fit(X_train,y_train)

y_pred_LR=LR.predict(X_test)
accuracy_score(y_test,y_pred_LR)*100
```

Out[46]:
```
37.94217000573626
```

In [47]:
```python
print(classification_report(y_test,y_pred_LR))
```

```
              precision    recall  f1-score   support
```



In [47]:
```python
print(classification_report(y_test,y_pred_LR))
```

```
              precision    recall  f1-score   support

           1       0.48      0.02      0.03      6901
           2       0.37      0.45      0.41     23205
           3       0.40      0.64      0.49     25792
           4       0.32      0.19      0.24     16289
           5       0.00      0.00      0.00      3439
           6       0.37      0.46      0.41     10470
           7       0.00      0.00      0.00       822
           8       0.00      0.00      0.00      3093
           9       0.11      0.00      0.01      1412
          10       0.00      0.00      0.00       782
          11       0.49      0.32      0.39      1933

    accuracy                           0.38     94138
   macro avg       0.23      0.19      0.18     94138
weighted avg       0.34      0.38      0.33     94138
```

In [48]:
```python
decision_tree_classification = DecisionTreeClassifier(criterion = 'entropy', random_state = 10)

decision_tree = decision_tree_classification.fit(X_train, y_train)
```

In [49]:
```python
y_pred_DT=decision_tree.predict(X_test)
accuracy_score(y_test,y_pred_DT)*100
```

Out[49]:
```
29.678769466102956
```

In [50]:
```python
print(classification_report(y_test,y_pred_DT))
```

```
              precision    recall  f1-score   support

           1       0.19      0.19      0.19      6901
```