In [110]:
```
import pandas as pd
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv("train_data1.csv", engine= "python")
data.head(10)
```

ospital_region_code	Available Extra Rooms in Hospital	Department	Ward_Type	Ward_Facility_Code	Bed Grade	patientid	City_Code_Patient
Z	3	radiotherapy	R	F	2.0	31397	7.0
Z	2	radiotherapy	S	F	2.0	31397	7.0
X	2	anesthesia	S	E	2.0	31397	7.0
Y	2	radiotherapy	R	D	2.0	31397	7.0
Y	2	radiotherapy	S	D	2.0	31397	7.0

In [4]:
```
test_data = pd.read_csv('test_data.csv')
test_data.head()
```

Out[4]:

	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Avai R Hos
0	318439	21	c	3	Z	
1	318440	29	a	4	X	
2	318441	26	b	2	Y	
3	318442	6	a	6	X	
4	318443	28	b	11	X	

In [5]:
```
age_lst = data["Age"].unique()

age_lst.sort()
age_dict = dict(zip(age_lst, range(len(age_lst))))
data["Age"]=data["Age"].replace(age_dict)
print(age_dict)

stay_list = data["Stay"].unique()
stay_list.sort()
dept_Stay = dict(zip(stay_list, range(len(stay_list))))
data["Stay"]= data["Stay"].replace(dept_Stay)
print(dept_Stay)

data.head()
```

```
{'0-10': 0, '11-20': 1, '21-30': 2, '31-40': 3, '41-50': 4, '51-60': 5, '61
-70': 6, '71-80': 7, '81-90': 8, '91-100': 9}
{'0-10': 0, '11-20': 1, '21-30': 2, '31-40': 3, '41-50': 4, '51-60': 5, '61
-70': 6, '71-80': 7, '81-90': 8, '91-100': 9, 'More than 100 Days': 10}
```

Out[5]:

	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Avai Ro Hos
0	1	8	c	3	Z	
1	2	2	c	5	Z	
2	3	10	e	1	X	
3	4	26	b	2	Y	
4	5	26	b	2	Y	

In [7]:
```
new1 = data.drop(['case_id','patientid']
            , axis = 1)
```

In [15]:
```
import numpy as np

new_dept = new1["Department"].unique()
new_dept.sort()
new_dept = dict(zip(new_dept, range(len(new_dept))))
new1.Department.replace(new_dept, inplace= True)
print(new_dept)

new_hosp_code = new1["Hospital_region_code"].unique()
new_hosp_code.sort()
new_hosp_code= dict(zip(new_hosp_code, range(len(new_hosp_code))))
new1.Hospital_region_code.replace(new_hosp_code, inplace = True)
print(new_hosp_code)

new_ward_type = new1["Ward_Type"].unique()
new_ward_type.sort()
new_ward_type = dict(zip(new_ward_type, range(len(new_ward_type))))
new1.replace(new_ward_type, inplace=True)
print(new_ward_type)

new_type_admiss = new1["Type of Admission"].unique()
new_type_admiss.sort()
new_type_admiss = dict(zip(new_type_admiss, range(len(new_type_admiss))))
new1["Type of Admission"].replace(new_type_admiss, inplace=True)
print(new_type_admiss)

new_severity = new1["Severity of Illness"].unique()
new_severity .sort()
new_severity  = dict(zip(new_severity, range(len(new_severity ))))
new1["Severity of Illness"].replace(new_severity , inplace=True)
print(new_severity )

new_Hospital_type_code = new1["Hospital_type_code"].unique()
new_Hospital_type_code .sort()
new_Hospital_type_code  = dict(zip(new_Hospital_type_code, range(len(new_Hosp
new1["Hospital_type_code"].replace(new_Hospital_type_code , inplace=True)
print(new_Hospital_type_code )

new_Ward_Facility_Code = new1["Ward_Facility_Code"].unique()
new_Ward_Facility_Code .sort()
new_Ward_Facility_Code  = dict(zip(new_Ward_Facility_Code, range(len(new_Ward
new1["Ward_Facility_Code"].replace(new_Ward_Facility_Code , inplace=True)
print(new_Ward_Facility_Code )
```

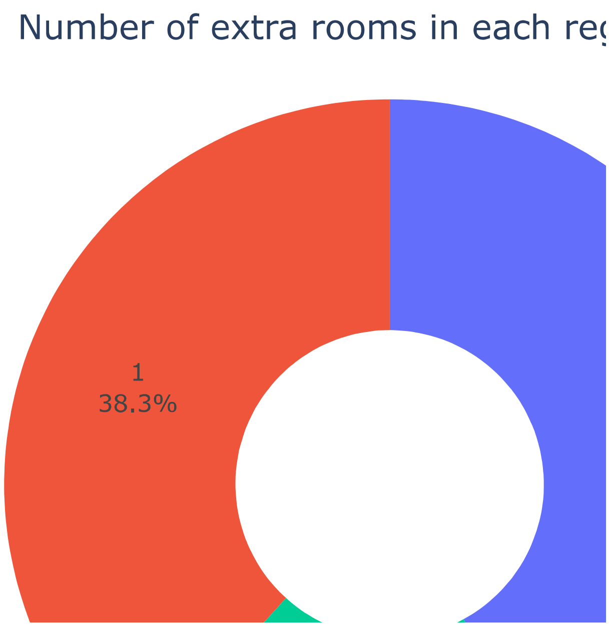
```
{0: 0, 1: 1, 2: 2, 3: 3, 4: 4}
{0: 0, 1: 1, 2: 2}
{0: 0, 1: 1, 2: 2, 3: 3, 4: 4, 5: 5}
{0: 0, 1: 1, 2: 2}
{0: 0, 1: 1, 2: 2}
{0: 0, 1: 1, 2: 2, 3: 3, 4: 4, 5: 5, 6: 6}
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5}
```

In [16]:
```
new1.head()
```

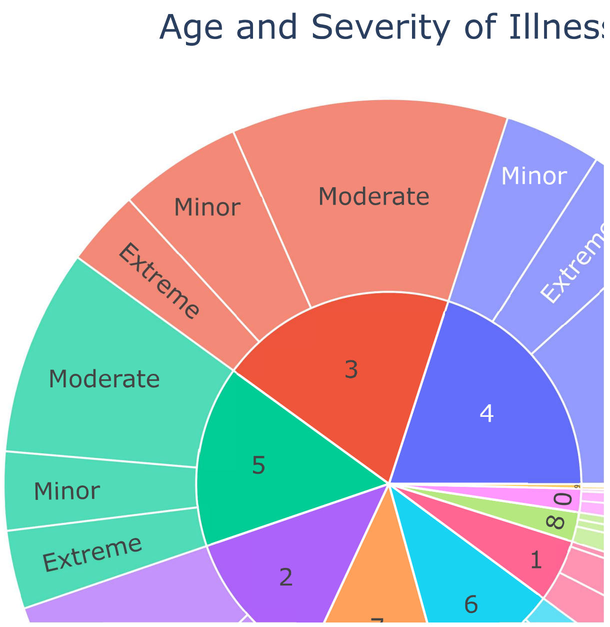
Out[16]:

ɡion_code	Available Extra Rooms in Hospital	Department	Ward_Type	Ward_Facility_Code	Bed Grade	City_Code_Patient	A
2	3	3	2	5	2.0	7.0	
2	2	3	3	5	2.0	7.0	
0	2	1	3	4	2.0	7.0	
1	2	3	2	3	2.0	7.0	
1	2	3	3	3	2.0	7.0	

In [94]:
```
import plotly.express as px
extra_room=new1.groupby('Hospital_region_code')['Available Extra Rooms in Hos
fig4=px.pie(extra_room,values='Available Extra Rooms in Hospital',names='Hosp
fig4.update_layout(title='Number of extra rooms in each region code',title_x=
fig4.update_traces(textinfo='percent+label')
```



In [106]:
```
age_plot=px.sunburst(data, path=['Age','Severity of Illness'])
age_plot.update_layout(title='Age and Severity of Illness',title_x=0.5)
age_plot.show()
```



In [17]:
```
new2 = test_data.drop(['case_id','patientid']
           , axis = 1)
```

In [20]:

```
new_dept = new2["Department"].unique()
new_dept.sort()
new_dept = dict(zip(new_dept, range(len(new_dept))))
new2.Department.replace(new_dept, inplace= True)
print(new_dept)

new_hosp_code = new2["Hospital_region_code"].unique()
new_hosp_code.sort()
new_hosp_code= dict(zip(new_hosp_code, range(len(new_hosp_code))))
new2.Hospital_region_code.replace(new_hosp_code, inplace = True)
print(new_hosp_code)

new_ward_type = new2["Ward_Type"].unique()
new_ward_type.sort()
new_ward_type = dict(zip(new_ward_type, range(len(new_ward_type))))
new2.replace(new_ward_type, inplace=True)
print(new_ward_type)

new_type_admiss = new2["Type of Admission"].unique()
new_type_admiss.sort()
new_type_admiss = dict(zip(new_type_admiss, range(len(new_type_admiss))))
new2["Type of Admission"].replace(new_type_admiss, inplace=True)
print(new_type_admiss)

new_severity = new2["Severity of Illness"].unique()
new_severity .sort()
new_severity  = dict(zip(new_severity, range(len(new_severity ))))
new2["Severity of Illness"].replace(new_severity , inplace=True)
print(new_severity )

new_Hospital_type_code = new2["Hospital_type_code"].unique()
new_Hospital_type_code .sort()
new_Hospital_type_code  = dict(zip(new_Hospital_type_code, range(len(new_Hosp
new2["Hospital_type_code"].replace(new_Hospital_type_code , inplace=True)
print(new_Hospital_type_code )

new_Ward_Facility_Code = new2["Ward_Facility_Code"].unique()
new_Ward_Facility_Code .sort()
new_Ward_Facility_Code  = dict(zip(new_Ward_Facility_Code, range(len(new_Ward
new2["Ward_Facility_Code"].replace(new_Ward_Facility_Code , inplace=True)
print(new_Ward_Facility_Code )

new_age = new2["Age"].unique()
new_age .sort()
new_age = dict(zip(new_age, range(len(new_age))))
new2["Age"].replace(new_age , inplace=True)
print(new_age )
```

```
{0: 0, 1: 1, 2: 2, 3: 3, 4: 4}
{0: 0, 1: 1, 2: 2}
{0: 0, 1: 1, 2: 2, 3: 3, 4: 4, 5: 5}
{0: 0, 1: 1, 2: 2}
{0: 0, 1: 1, 2: 2}
```

```
{0: 0, 1: 1, 2: 2, 3: 3, 4: 4, 5: 5, 6: 6}
{0: 0, 1: 1, 2: 2, 3: 3, 4: 4, 5: 5}
{'0-10': 0, '11-20': 1, '21-30': 2, '31-40': 3, '41-50': 4, '51-60': 5,
'61-70': 6, '71-80': 7, '81-90': 8, '91-100': 9}
```

In [21]:

```
new2.head()
```

Out[21]:

Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Available Extra Rooms in Hospital	Depar
21	2	3	2	3	
29	0	4	0	2	
26	1	2	1	3	
6	0	6	0	3	
28	1	11	0	2	

In [25]:

```
pip install catboost
```

```
Collecting catboost
  Downloading catboost-0.24.3-cp38-none-win_amd64.whl (65.4 MB)
Requirement already satisfied: six in c:\users\moiez\anaconda3\lib\site-pac
kages (from catboost) (1.15.0)
Requirement already satisfied: plotly in c:\users\moiez\anaconda3\lib\site-
packages (from catboost) (4.12.0)
Requirement already satisfied: pandas>=0.24.0 in c:\users\moiez\anaconda3\l
ib\site-packages (from catboost) (1.0.5)
Requirement already satisfied: numpy>=1.16.0 in c:\users\moiez\anaconda3\li
b\site-packages (from catboost) (1.18.5)
Requirement already satisfied: matplotlib in c:\users\moiez\anaconda3\lib\s
ite-packages (from catboost) (3.2.2)
Collecting graphviz
  Downloading graphviz-0.15-py2.py3-none-any.whl (18 kB)
Requirement already satisfied: scipy in c:\users\moiez\anaconda3\lib\site-p
ackages (from catboost) (1.5.0)
Requirement already satisfied: retrying>=1.3.3 in c:\users\moiez\anaconda3
\lib\site-packages (from plotly->catboost) (1.3.3)
Requirement already satisfied: pytz>=2017.2 in c:\users\moiez\anaconda3\lib
\site-packages (from pandas>=0.24.0->catboost) (2020.1)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\users\moiez\ana
conda3\lib\site-packages (from pandas>=0.24.0->catboost) (2.8.1)
Requirement already satisfied: cycler>=0.10 in c:\users\moiez\anaconda3\lib
\site-packages (from matplotlib->catboost) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
c:\users\moiez\anaconda3\lib\site-packages (from matplotlib->catboost) (2.
4.7)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\moiez\anaconda
3\lib\site-packages (from matplotlib->catboost) (1.2.0)
Installing collected packages: graphviz, catboost
Successfully installed catboost-0.24.3 graphviz-0.15
Note: you may need to restart the kernel to use updated packages.
```

CATBOOST MODEL

In [30]:

```
from catboost import CatBoostClassifier, Pool
from catboost import CatBoostRegressor
from sklearn.model_selection import train_test_split
```

In [23]:

```
X = new1.drop(columns=['Stay'])
Y = new1['Stay']


# selecting features for test data

test_X = new2
```

In [31]:
```
X_train , X_test , Y_train , Y_test = train_test_split(X,Y,test_size = 0.2 ,
```

In [32]:
```
train_dataset = Pool(data=X_train, label=Y_train)

eval_dataset = Pool(data=X_test, label=Y_test)
```

In [33]:
```
model = CatBoostClassifier(iterations=750,
                           learning_rate=0.08,
                           depth=7,
                           loss_function='MultiClass',
                           eval_metric='Accuracy')
```

In [34]:
```
model.fit(train_dataset)

# validation

eval_pred = model.predict(eval_dataset)
```

```
21:     learn: 0.4001060        total: 13s      remaining: 7m 9s
22:     learn: 0.4015819        total: 13.6s    remaining: 7m 9s
23:     learn: 0.4026104        total: 14.2s    remaining: 7m 8s
24:     learn: 0.4037213        total: 14.7s    remaining: 7m 6s
25:     learn: 0.4043101        total: 15.3s    remaining: 7m 6s
26:     learn: 0.4043925        total: 15.9s    remaining: 7m 6s
27:     learn: 0.4047105        total: 16.5s    remaining: 7m 5s
28:     learn: 0.4058999        total: 17.1s    remaining: 7m 5s
29:     learn: 0.4057429        total: 17.7s    remaining: 7m 3s
30:     learn: 0.4065201        total: 18.2s    remaining: 7m 2s
31:     learn: 0.4067242        total: 18.8s    remaining: 7m 2s
32:     learn: 0.4076271        total: 19.4s    remaining: 7m 1s
33:     learn: 0.4077645        total: 20s      remaining: 7m
34:     learn: 0.4079607        total: 20.6s    remaining: 6m 59s

35:     learn: 0.4082041        total: 21.1s    remaining: 6m 59s
36:     learn: 0.4088361        total: 21.7s    remaining: 6m 58s
37:     learn: 0.4089225        total: 22.3s    remaining: 6m 57s
38:     learn: 0.4092208        total: 22.9s    remaining: 6m 56s
39:     learn: 0.4094171        total: 23.4s    remaining: 6m 55s
```

In [35]:
```

model.get_best_score()
```

Out[35]:
```
{'learn': {'Accuracy': 0.4592423945044161, 'MultiClass': 1.405211229645959
8}}
```

In [43]:
```
from catboost.utils import get_confusion_matrix
from sklearn.metrics import confusion_matrix
```

In [44]:
```
cm = confusion_matrix(Y_test, eval_pred)
cm
```

Out[44]:
```
array([[  805,  2265,  1607,    27,     0,     9,     0,     0,     0,
            0,     0],
       [  427,  7991,  6105,   685,     3,   452,     1,     2,     3,
            0,     3],
       [  305,  5111, 11540,   258,     4,   216,     1,     6,    27,
            0,    27],
       [  187,  1877,  4448,  2628,    11,  1736,     1,    19,    12,
            1,    21],
       [   97,   413,  1474,   141,     9,   176,     0,     2,    10,
            0,    16],
       [   72,   541,   955,  1609,     4,  3513,     2,    27,   129,
            2,   110],
       [   13,    78,   301,    43,     1,   102,     0,     3,    11,
            0,    20],
       [   31,   135,   205,   417,     0,  1086,     0,    52,    37,
            2,   134],
       [   11,    42,    49,    85,     0,   493,     0,    11,   184,
            1,   115],
       [    8,    38,    62,    82,     1,   267,     0,     6,     5,
           10,    91],
       [   14,    61,    64,    79,     3,   396,     0,    35,   101,
            5,   575]], dtype=int64)
```

In [51]:
```

new_dataset = Pool(test_X)

y_pred = model.predict(new_dataset)
```

In [52]:
```

output = pd.DataFrame(test_data['case_id'].values,columns=['case_id'])
output['Stay'] = y_pred
swap_dict_stay = dict([(value, key) for key, value in dept_Stay.items()])
output['Stay'].replace(swap_dict_stay, inplace=True)
```

In [55]:
```
output.head(5)
```

Out[55]:

	case_id	Stay
0	318439	0-10
1	318440	51-60
2	318441	21-30
3	318442	21-30
4	318443	51-60

In [72]:

```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set(style='whitegrid')

# Modeling
from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedKFold

from sklearn.linear_model import SGDClassifier

from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifi
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import roc_curve, auc, accuracy_score, roc_auc_score, f1


from sklearn.model_selection import RandomizedSearchCV

from sklearn.ensemble import RandomForestClassifier
from sklearn import tree
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import KFold, cross_val_score, train_test_split
from sklearn.metrics import accuracy_score
from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
```

In [73]:
```
new1.dropna()
```

Out[73]:

	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Availabl Extr Room i Hospita
0	8	2	3	2	
1	2	2	5	2	
2	10	4	1	0	
3	26	1	2	1	
4	26	1	2	1	
...	...	...	...	...	.
318433	6	0	6	0	
318434	24	0	1	0	
318435	7	0	4	0	
318436	11	1	2	1	
318437	19	0	7	1	

313793 rows × 16 columns

In [80]:
```
new1.isna().sum()
```

Out[80]:
```
Hospital_code                           0
Hospital_type_code                      0
City_Code_Hospital                      0
Hospital_region_code                    0
Available Extra Rooms in Hospital       0
Department                              0
Ward_Type                               0
Ward_Facility_Code                      0
Bed Grade                             113
City_Code_Patient                    4532
Type of Admission                       0
Severity of Illness                     0
Visitors with Patient                   0
Age                                     0
Admission_Deposit                       0
Stay                                    0
dtype: int64
```

KNN MODEL

In [85]:
```
x = new1.drop(["Stay",'Bed Grade','City_Code_Patient','Hospital_code', 'Hospi
       'Hospital_region_code' ], axis=1).to_numpy()
y = new1['Stay'].values
```

In [86]:
```
X_train, X_val, Y_train, Y_val = train_test_split(x, y, test_size = 0.2, ranc
```

In [107]:
```
neighbors = KNeighborsClassifier(n_neighbors=11) # 11 different values of Sta
neighbors.fit(X_train, Y_train)
new_Y_pred= neighbors.predict(X_val)
# get the accuracy score
acc_neigh = accuracy_score(new_Y_pred, Y_val)
print(acc_neigh)
```

```
0.29609973621404345
```

RANDOM FOREST MODEL

In [108]:
```
randfor = RandomForestClassifier(n_estimators=200, max_depth=15)

randfor.fit(X_train, Y_train)

pred_randfor = randfor.predict(X_val)
# get the accuracy score
accuracy = accuracy_score(Y_pred_rf, Y_val)
print(accuracy)
```

```
0.4003736967717623
```

LINEAR REGRESSION

In [109]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
linear_reg = LinearRegression()

linear_reg.fit(X_train, Y_train)

pred_linear_reg = linear_reg.predict(X_val)

accuracy = linear_reg.score(X_val,Y_val)
print(accuracy)
```

```
0.361929490302932
```

In [ ]: