In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style("whitegrid")
plt.style.use("fivethirtyeight")
```

In [2]:

```
train = pd.read_csv('train_data.csv')
test = pd.read_csv('test_data.csv')
dictionary = pd.read_csv('train_data_dictionary.csv')
sample = pd.read_csv('sample_sub.csv')
```

EDA

In [5]:

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 318438 entries, 0 to 318437
Data columns (total 18 columns):
 #   Column                             Non-Null Count   Dtype  
---  ------                             --------------   -----  
 0   case_id                            318438 non-null  int64  
 1   Hospital_code                      318438 non-null  int64  
 2   Hospital_type_code                 318438 non-null  object 
 3   City_Code_Hospital                 318438 non-null  int64  
 4   Hospital_region_code               318438 non-null  object 
 5   Available Extra Rooms in Hospital  318438 non-null  int64  
 6   Department                         318438 non-null  object 
 7   Ward_Type                          318438 non-null  object 
 8   Ward_Facility_Code                 318438 non-null  object 
 9   Bed Grade                          318325 non-null  float64
 10  patientid                          318438 non-null  int64  
 11  City_Code_Patient                  313906 non-null  float64
 12  Type of Admission                  318438 non-null  object 
 13  Severity of Illness                318438 non-null  object 
 14  Visitors with Patient              318438 non-null  int64  
 15  Age                                318438 non-null  object 
 16  Admission_Deposit                  318438 non-null  float64
 17  Stay                               318438 non-null  object 
dtypes: float64(3), int64(6), object(9)
memory usage: 43.7+ MB
```

In [6]:

```
train.Stay.value_counts()
```

Out[6]:

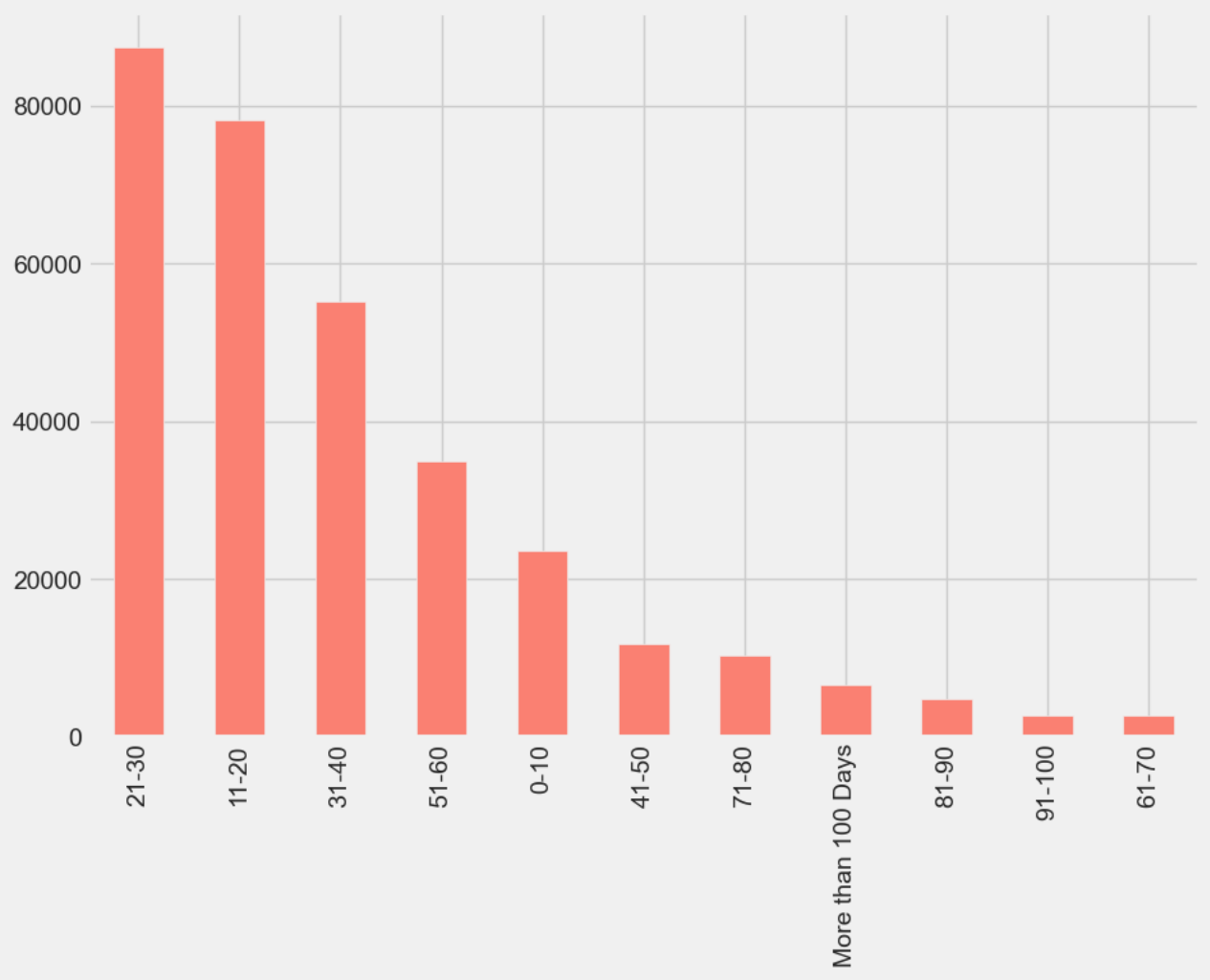
```
21-30                 87491
11-20                 78139
31-40                 55159
51-60                 35018
0-10                  23604
41-50                 11743
71-80                 10254
More than 100 Days     6683
81-90                  4838
91-100                 2765
61-70                  2744
Name: Stay, dtype: int64
```

In [7]:

```
# Distribution of target feature
plt.figure(figsize=(10,7))
train.Stay.value_counts().plot(kind="bar", color = ['Salmon'])
```

Out[7]:

<Axes: >



In [8]:

```
# Check for unique values in every column
for features in train.columns:
    print('Unique Values for {}'.format(features))
    print(train[features].unique())
    print('====================================')
    print()
```

```
Unique Values for case_id
[     1      2      3 ... 318436 318437 318438]
====================================

Unique Values for Hospital_code
[ 8  2 10 26 23 32  1 22 16  9  6 29 12  3 21 28 27 19  5 14 13 31 24 17
 25 15 11 30 18  4  7 20]
====================================

Unique Values for Hospital_type_code
['c' 'e' 'b' 'a' 'f' 'd' 'g']
====================================

Unique Values for City_Code_Hospital
[ 3  5  1  2  6  9 10  4 11  7 13]
====================================

Unique Values for Hospital_region_code
['Z' 'X' 'Y']
====================================

Unique Values for Available Extra Rooms in Hospital
[ 3  2  1  4  6  5  7  8  9 10 12  0 11 20 14 21 13 24]
====================================

Unique Values for Department
['radiotherapy' 'anesthesia' 'gynecology' 'TB & Chest disease' 'surgery']
====================================

Unique Values for Ward_Type
['R' 'S' 'Q' 'P' 'T' 'U']
====================================

Unique Values for Ward_Facility_Code
['F' 'E' 'D' 'B' 'A' 'C']
====================================

Unique Values for Bed Grade
[ 2.  3.  4.  1. nan]
====================================

Unique Values for patientid
[ 31397  63418   8088 ... 125235  91081  21641]
====================================

Unique Values for City_Code_Patient
[ 7.  8.  2.  5.  6.  3.  4.  1.  9. 14. nan 25. 15. 12. 10. 28. 24. 23.
 20. 11. 13. 21. 18. 16. 26. 27. 22. 19. 31. 34. 32. 30. 29. 37. 33. 35.
 36. 38.]
====================================

Unique Values for Type of Admission
['Emergency' 'Trauma' 'Urgent']
====================================

Unique Values for Severity of Illness
['Extreme' 'Moderate' 'Minor']
====================================

Unique Values for Visitors with Patient
[ 2  4  3  8  6  7 13  5  1 10 15 11 12  9 24 16 14 20  0 19 18 17 23 21
 32 30 22 25]
====================================

Unique Values for Age
['51-60' '71-80' '31-40' '41-50' '81-90' '61-70' '21-30' '11-20' '0-10'
 '91-100']
====================================

Unique Values for Admission_Deposit
[4911. 5954. 4745. ... 1937. 9439. 2349.]
====================================

Unique Values for Stay
['0-10' '41-50' '31-40' '11-20' '51-60' '21-30' '71-80'
 'More than 100 Days' '81-90' '61-70' '91-100']
====================================
```

In [9]:

```
# Check for null values
train.isna().sum()
```

Out[9]:

```
case_id                               0
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
patientid                             0
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
```

Data Processing & Feature engineering

In [11]:

```
train = train.drop(['Hospital_region_code', 'Bed Grade', 'patientid', 'City_Code_Patient'], axis = 1)
test = test.drop(['Hospital_region_code', 'Bed Grade', 'patientid', 'City_Code_Patient'], axis = 1)
```

In [12]:

```
# Combine test and train dataset for processing
combined = [train, test]
```

In [13]:

```
from sklearn.preprocessing import LabelEncoder

for dataset in combined:
    label = LabelEncoder()
    dataset['Department'] = label.fit_transform(dataset['Department'])
```

In [14]:

```
combined[1].Department.unique()
```

Out[14]:

```
array([2, 1, 0, 3, 4])
```

In [15]:

```
for dataset in combined:
    label = LabelEncoder()
    dataset['Hospital_type_code'] = label.fit_transform(dataset['Hospital_type_code'])
    dataset['Ward_Facility_Code'] = label.fit_transform(dataset['Ward_Facility_Code'])
    dataset['Ward_Type'] = label.fit_transform(dataset['Ward_Type'])
    dataset['Type of Admission'] = label.fit_transform(dataset['Type of Admission'])
    dataset['Severity of Illness'] = label.fit_transform(dataset['Severity of Illness'])
```

In [16]:

```
combined[0]
```

Out[16]:

	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Available Extra Rooms in Hospital	Department	Ward_Type	Ward_Facility_Code	Type of Admission	Severity of Illness
0	1	8	2	3	3	3	2	5	0	0
1	2	2	2	5	2	3	3	5	1	0
2	3	10	4	1	2	1	3	4	1	0
3	4	26	1	2	2	3	2	3	1	0
4	5	26	1	2	2	3	3	3	1	0
...	...	...	...	...	...	...	...	...	...	...
318433	318434	6	0	6	3	3	1	5	0	2
318434	318435	24	0	1	2	1	1	4	2	2
318435	318436	7	0	4	3	2	2	5	0	1
318436	318437	11	1	2	3	1	1	3	1	1
318437	318438	19	0	7	5	2	1	2	0	1

318438 rows × 14 columns

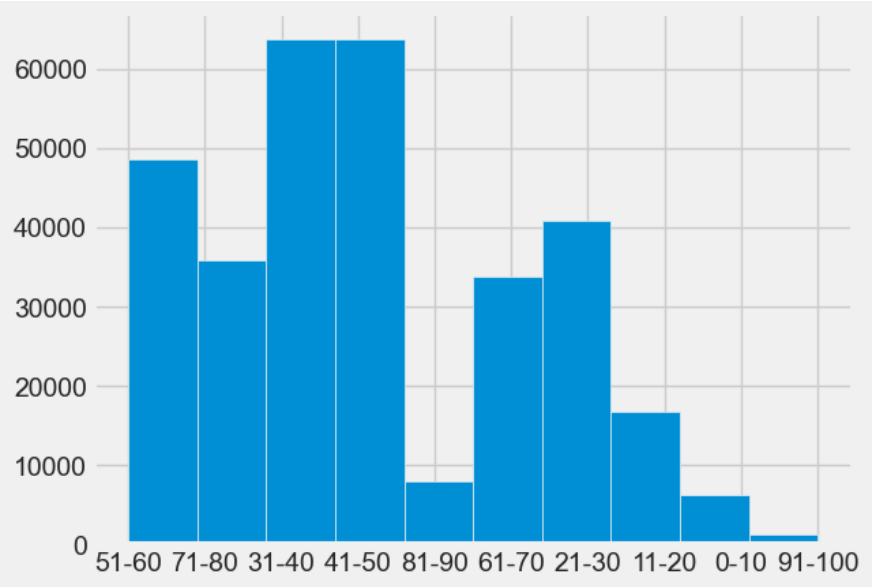
In [17]:

```
# Check age distribution
combined[0].Age.hist()

```

Out[17]:

<Axes: >



In [19]:

```
age_dict = {'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 [20]:

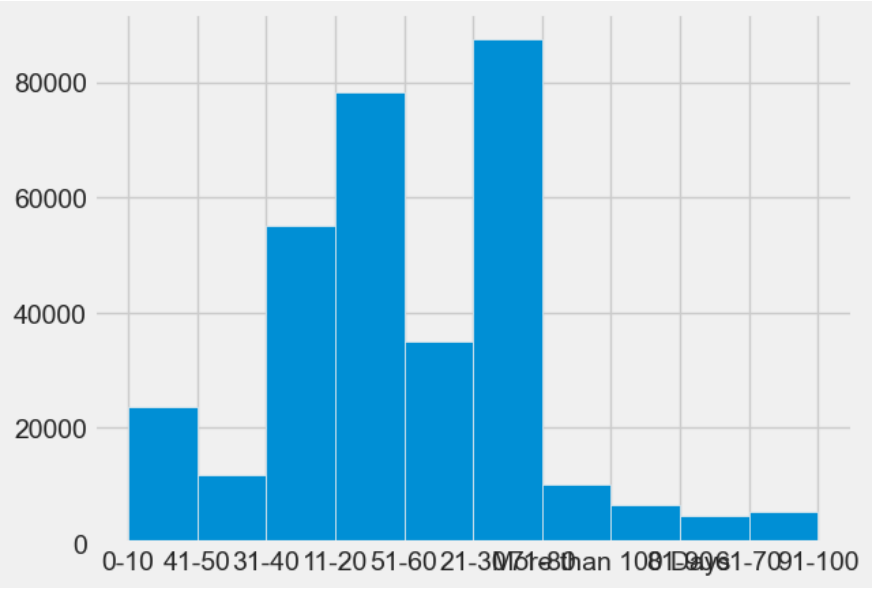
```
for dataset in combined:
    dataset['Age'] = dataset['Age'].replace(age_dict.keys(), age_dict.values())
```

In [21]:

```
combined[0].Stay.hist()
```

Out[21]:

<Axes: >



In [22]:

```
combined[0].Stay.unique()
```

Out[22]:

```
array(['0-10', '41-50', '31-40', '11-20', '51-60', '21-30', '71-80',
       'More than 100 Days', '81-90', '61-70', '91-100'], dtype=object)
```

In [24]:

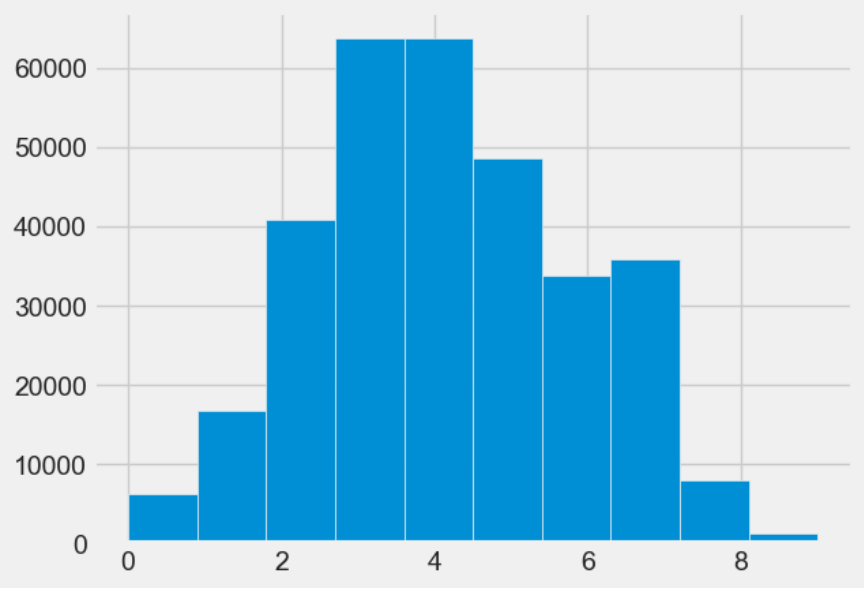
```
stay_dict = {'0-10': 0, '11-20': 1, '21-30': 2, '31-40': 3, '41-50': 4, '51-60': 5, '61-70': 6, '71-80': 7, '81-90': 8, '91-100': 
combined[0]['Stay'] = combined[0]['Stay'].replace(stay_dict.keys(), stay_dict.values())
```

In [25]:

```
combined[0].Age.hist()
```

Out[25]:

<Axes: >



In [26]:

```
for dataset in combined:
    print(dataset.shape)
```

```
(318438, 14)
(137057, 13)
```

In [27]:

```
1 combined[1].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 137057 entries, 0 to 137056
Data columns (total 13 columns):
 #   Column                             Non-Null Count   Dtype  
---  ------                             --------------   -----  
 0   case_id                            137057 non-null  int64  
 1   Hospital_code                      137057 non-null  int64  
 2   Hospital_type_code                 137057 non-null  int32  
 3   City_Code_Hospital                 137057 non-null  int64  
 4   Available Extra Rooms in Hospital  137057 non-null  int64  
 5   Department                         137057 non-null  int32  
 6   Ward_Type                          137057 non-null  int32  
 7   Ward_Facility_Code                 137057 non-null  int32  
 8   Type of Admission                  137057 non-null  int32  
 9   Severity of Illness                137057 non-null  int32  
 10  Visitors with Patient              137057 non-null  int64  
 11  Age                                137057 non-null  int64  
 12  Admission_Deposit                  137057 non-null  float64
dtypes: float64(1), int32(6), int64(6)
memory usage: 10.5 MB
```

Scaling numerical data

In [28]:

```
1 columns_list = ['Type of Admission', 'Available Extra Rooms in Hospital', 'Visitors with Patient', 'Admission_Deposit']
```

In [29]:

```
1 len(columns_list)
```

Out[29]:

```
4
```

In [30]:

```
1 from sklearn.preprocessing import StandardScaler
2
3 ss= StandardScaler()
4
5 for dataset in combined:
6     dataset[columns_list]= ss.fit_transform(dataset[columns_list].values)
```

In [31]:

```
1 combined[0]
```

Out[31]:

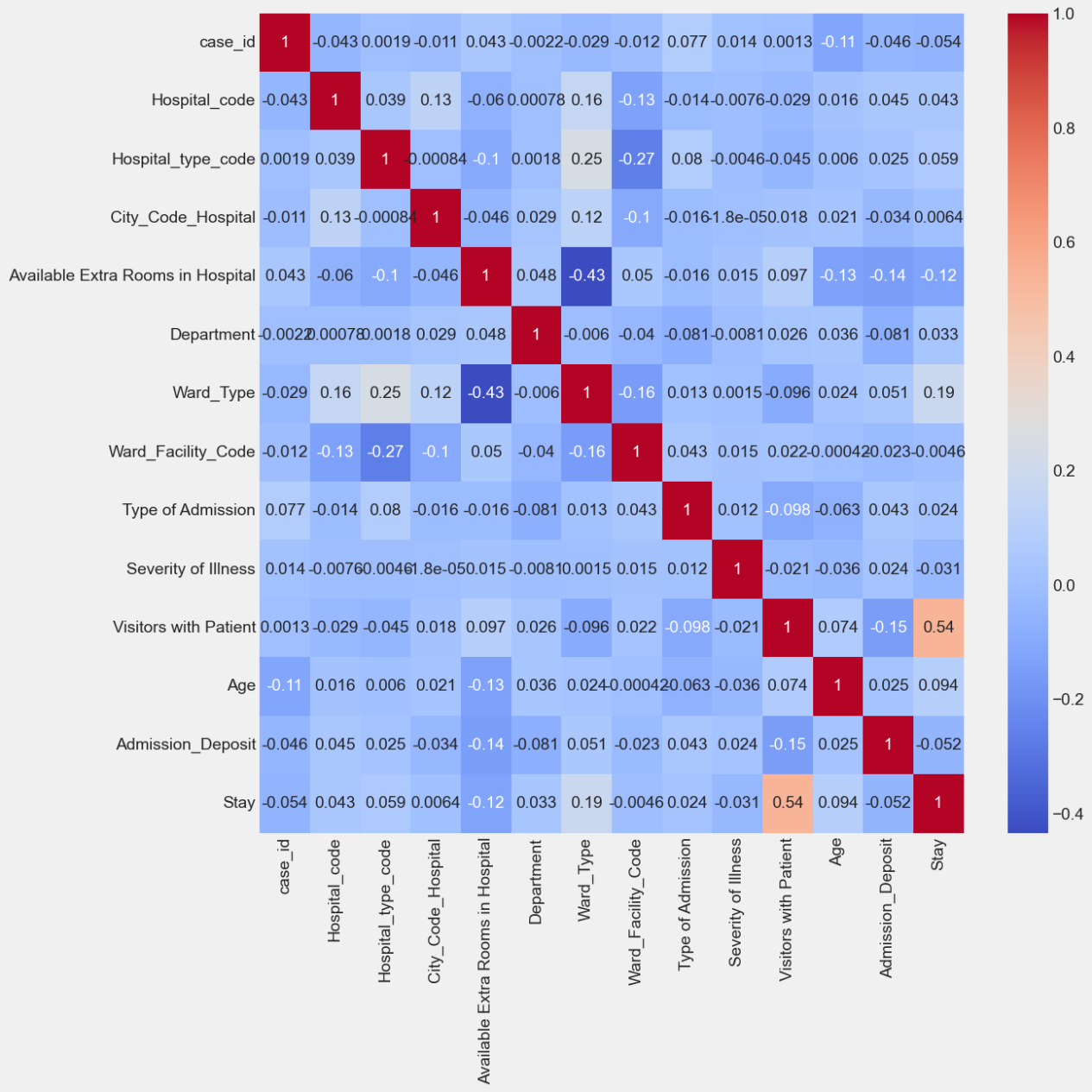
	Hospital_type_code	City_Code_Hospital	Available Extra Rooms in Hospital	Department	Ward_Type	Ward_Facility_Code	Type of Admission	Severity of Illness	Visitors with Patient	Age	Admission_Depo
	2	3	-0.169177	3	2	5	-1.136165	0	-0.727923	5	0.0278
	2	5	-1.025217	3	3	5	0.315306	0	-0.727923	5	0.9875
	4	1	-1.025217	1	3	4	0.315306	0	-0.727923	5	-0.1249
	1	2	-1.025217	3	2	3	0.315306	0	-0.727923	5	2.2003
	1	2	-1.025217	3	3	3	0.315306	0	-0.727923	5	0.6231
	...	...	...	...	...	...	...	...	...	...	
	0	6	-0.169177	3	1	5	-1.136165	2	-0.161049	4	-0.6779
	0	1	-1.025217	1	1	4	1.766778	2	0.405826	8	1.6730
	0	4	-0.169177	2	2	5	-1.136165	1	-0.161049	7	-0.5941
	1	2	-0.169177	1	1	3	0.315306	1	0.972701	1	-1.0303
	0	7	1.542903	2	1	2	-1.136165	1	-0.727923	1	-0.1184

In [32]:

```
plt.figure(figsize=(12,12))
sns.heatmap(combined[0].corr(), annot=True, cmap='coolwarm')
```

Out[32]:

<Axes: >



data modeling

In [33]:

```
# machine learning
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import Perceptron
from sklearn.linear_model import SGDClassifier
from sklearn.tree import DecisionTreeClassifier
```

In [34]:

```
train = combined[0]
test = combined[1]
```

In [35]:

```
X_train = train.drop(['case_id', 'Stay'], axis=1)
Y_train = train["Stay"]
X_test  = test.drop("case_id", axis=1).copy()
X_train.shape, Y_train.shape, X_test.shape
```

Out[35]:

```
((318438, 12), (318438,), (137057, 12))
```

In [36]:

```
X_test.columns
```

Out[36]:

```
Index(['Hospital_code', 'Hospital_type_code', 'City_Code_Hospital',
       'Available Extra Rooms in Hospital', 'Department', 'Ward_Type',
       'Ward_Facility_Code', 'Type of Admission', 'Severity of Illness',
       'Visitors with Patient', 'Age', 'Admission_Deposit'],
      dtype='object')
```

In [37]:

```
Y_train
```

Out[37]:

```
0         0
1         4
2         3
3         4
4         4
         ..
318433    1
318434    3
318435    1
318436    1
318437    0
Name: Stay, Length: 318438, dtype: int64
```

In [38]:

```
# KNN
knn = KNeighborsClassifier(n_neighbors = 3)
knn.fit(X_train, Y_train)
Y_pred = knn.predict(X_test)
acc_knn = round(knn.score(X_train, Y_train) * 100, 2)
acc_knn
```

Out[38]:

```
58.09
```

In [39]:

```
# Decision Tree
decision_tree = DecisionTreeClassifier()
decision_tree.fit(X_train, Y_train)
Y_pred = decision_tree.predict(X_test)
acc_decision_tree = round(decision_tree.score(X_train, Y_train) * 100, 2)
acc_decision_tree
```

Out[39]:

```
99.71
```

In [40]:

```
# Random Forest

random_forest = RandomForestClassifier(n_estimators=100)
random_forest.fit(X_train, Y_train)
Y_pred = random_forest.predict(X_test)
random_forest.score(X_train, Y_train)
acc_random_forest = round(random_forest.score(X_train, Y_train) * 100, 2)
acc_random_forest
```

Out[40]:

```
99.71
```

Submission

In [41]:

```
submission = pd.DataFrame({
        "case_id": test["case_id"],
        "Stay": Y_pred
})
submission['Stay'] = submission['Stay'].replace(stay_dict.values(), stay_dict.keys())
```

In [42]:

```
submission.to_csv('submission.csv', index = False)
```

In [ ]:

```

```