

SPRINT 2

Date	07-11-2022
Team ID	PNT2022TMID53243
Project Title	Analytics for Hospitals' Health-Care Data
Team Members	Varsha K,Pranav G,Pranav G,Hirthik prasaad R

Data Cleaning and Preparation

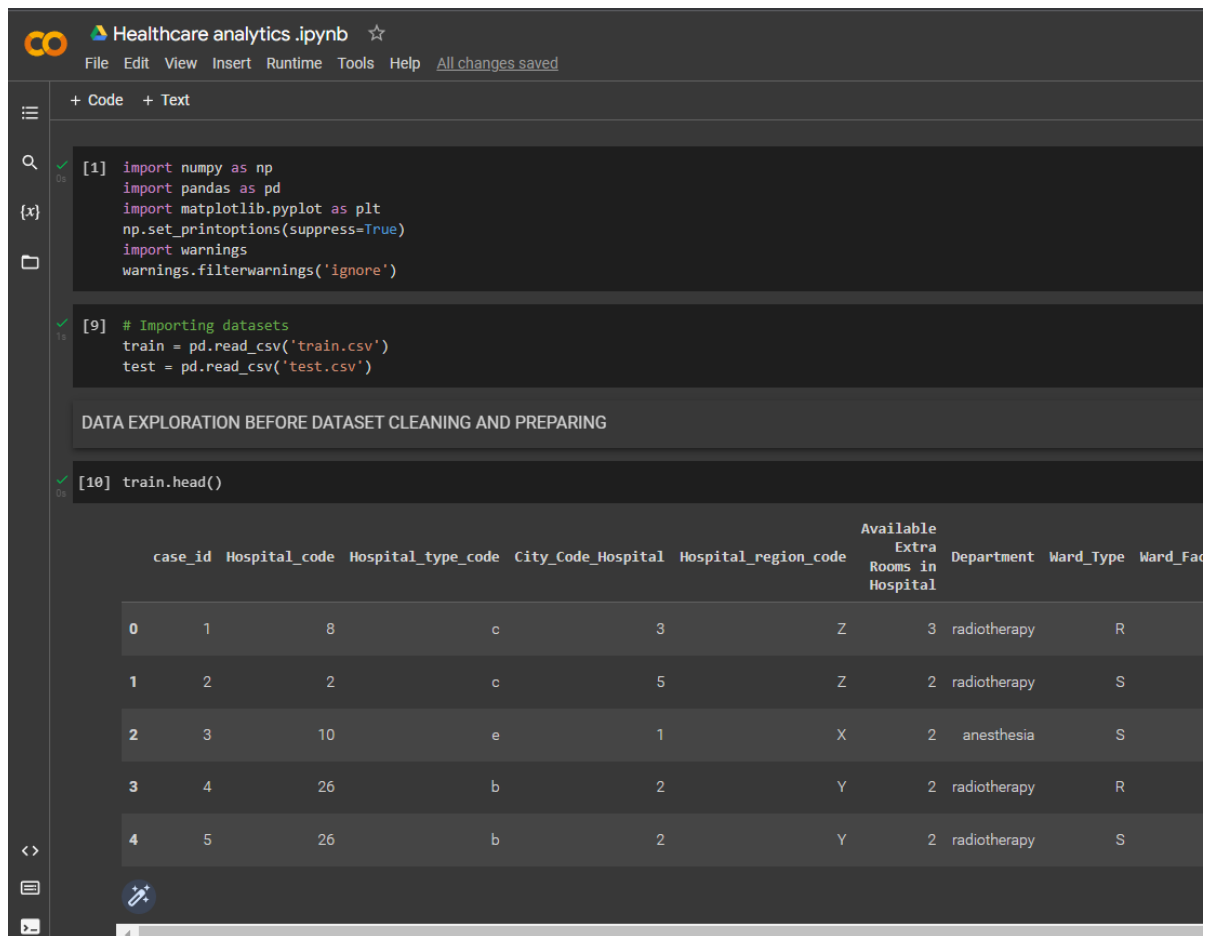
In this data set, variables “City_code_patient” and “Bed Grade” have missing values. These missing values must be treated before feeding to the algorithm as they distort the model performance.

So, the missing values are replaced using the “mode” of the column. Since most of the variables in the dataset have ordinal data, we transformed them into levels by using a label encoder to perform further analysis on the data.

Distinct Observations of Ordinal Data

Variables	Number of distinct observations
Hospital_type_code	7
Hospital_region_code	3
Department	5
Ward_Type	6
Ward_Facility_Code	6
Type of Admission	3
Severity of Illness	3
Age	10
Stay	11

Data Exploration in python



The screenshot shows a Jupyter Notebook interface with the title "Healthcare analytics.ipynb". The notebook contains the following code cells:

```
[1] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
np.set_printoptions(suppress=True)
import warnings
warnings.filterwarnings('ignore')
```

```
[9] # Importing datasets
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
```

Below the code cells, there is a section titled "DATA EXPLORATION BEFORE DATASET CLEANING AND PREPARING".

```
[10] train.head()
```

The output of the `train.head()` command is a DataFrame with the following columns: `case_id`, `Hospital_code`, `Hospital_type_code`, `City_Code_Hospital`, `Hospital_region_code`, `Available Extra Rooms in Hospital`, `Department`, `Ward_Type`, and `Ward_Facility`. The first five rows of data are displayed:

	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Available Extra Rooms in Hospital	Department	Ward_Type	Ward_Facility
0	1	8	c	3	Z	3	radiotherapy	R	
1	2	2	c	5	Z	2	radiotherapy	S	
2	3	10	e	1	X	2	anesthesia	S	
3	4	26	b	2	Y	2	radiotherapy	R	
4	5	26	b	2	Y	2	radiotherapy	S	



+ Code + Text



```
[11] train.info()
      train.Stay.unique()
```

```
<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
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)
```



+ Code + Text



```
[12] # NA values in train dataset
train.isnull().sum().sort_values(ascending = False)
```

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

```
[13] # NA values in test dataset
test.isnull().sum().sort_values(ascending = False)
```

City_Code_Patient	2157
Bed Grade	35
case_id	0



+ Code + Text

```
[13] # NA values in test dataset
test.isnull().sum().sort_values(ascending = False)
```

```
City_Code_Patient      2157
Bed_Grade              35
case_id                0
Age                   0
Visitors with Patient  0
Severity of Illness    0
Type of Admission      0
patientid             0
Ward_Facility_Code     0
Hospital_code          0
Ward_Type              0
Department             0
Available Extra Rooms in Hospital  0
Hospital_region_code   0
City_Code_Hospital     0
Hospital_type_code     0
Admission_Deposit      0
dtype: int64
```

```
[14] # Dimension of train dataset
train.shape
```

```
(318438, 18)
```

```
[15] # Dimension of test dataset
```



+ Code + Text



```
✓ [15] # Dimension of test dataset  
0s test.shape
```

```
(137057, 17)
```

```
✓ [16] # Number of distinct observations in train dataset  
0s for i in train.columns:  
    print(i, ': ', train[i].nunique())
```

```
case_id : 318438  
Hospital_code : 32  
Hospital_type_code : 7  
City_Code_Hospital : 11  
Hospital_region_code : 3  
Available Extra Rooms in Hospital : 18  
Department : 5  
Ward_Type : 6  
Ward_Facility_Code : 6  
Bed Grade : 4  
patientid : 92017  
City_Code_Patient : 37  
Type of Admission : 3  
Severity of Illness : 3  
Visitors with Patient : 28  
Age : 10  
Admission_Deposit : 7300  
Stay : 11
```



+ Code + Text



```
[17] # Number of distinct observations in test dataset
      for i in test.columns:
          print(i, ': ', test[i].nunique())
```

```
case_id : 137057
Hospital_code : 32
Hospital_type_code : 7
City_Code_Hospital : 11
Hospital_region_code : 3
Available Extra Rooms in Hospital : 15
Department : 5
Ward_Type : 6
Ward_Facility_Code : 6
Bed Grade : 4
patientid : 39607
City_Code_Patient : 37
Type of Admission : 3
Severity of Illness : 3
Visitors with Patient : 27
Age : 10
Admission_Deposit : 6609
```

DATA PREPARATION

```
Healthcare analytics .ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

DATA PREPARATION

[18] #Replacing NA values in Bed Grade Column for both Train and Test datasets
train['Bed Grade'].fillna(train['Bed Grade'].mode()[0], inplace = True)
test['Bed Grade'].fillna(test['Bed Grade'].mode()[0], inplace = True)

[19] #Replacing NA values in City_Code_Patient Column for both Train and Test datasets
train['City_Code_Patient'].fillna(train['City_Code_Patient'].mode()[0], inplace = True)
test['City_Code_Patient'].fillna(test['City_Code_Patient'].mode()[0], inplace = True)

[20] # Label Encoding Stay column in train dataset
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
train['Stay'] = le.fit_transform(train['Stay'].astype('str'))

[21] #Imputing dummy Stay column in test dataset to concatenate with train dataset
test['Stay'] = -1
df = pd.concat([train, test])
df.shape

(455495, 18)

[22] #Label Encoding all the columns in Train and test datasets
for i in ['Hospital_type_code', 'Hospital_region_code', 'Department',
          'Ward_Type', 'Ward_Facility_Code', 'Type of Admission', 'Severity of Illness', 'Age']:
    le = LabelEncoder()
    df[i] = le.fit_transform(df[i].astype(str))

[23] #Separating Train and Test Datasets
train = df[df['Stay']!= -1]
test = df[df['Stay']== -1]
```

Data exploration after preparing:

```
Healthcare analytics .ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

DATA EXPLORATION AFTER DATASET PREPARATION

[24] train.head()

case_id  Hospital_code  Hospital_type_code  City_Code_Hospital  Hospital_region_code  Available
Extra Rooms in Hospital  Department  Ward_Type  Ward_Facility_Code  Bed
Grade  patientid  City_Code_Patient  Type of Admission  Severity
of Illness  Visitors
with Patient  Age  Admission_Deposit  $
0      1      8      2      3      2      3      3      2      5  2.0  31397      7.0  0  0  2  5      4911.0
1      2      2      2      5      2      2      3      3      5  2.0  31397      7.0  1  0  2  5      5954.0
2      3      10     4      1      0      2      1      3      4  2.0  31397      7.0  1  0  2  5      4745.0
3      4      26     1      2      1      2      3      2      3  2.0  31397      7.0  1  0  2  5      7272.0
4      5      26     1      2      1      2      3      3      3  2.0  31397      7.0  1  0  2  5      5558.0

[26] test.head()

case_id  Hospital_code  Hospital_type_code  City_Code_Hospital  Hospital_region_code  Available
Extra Rooms in Hospital  Department  Ward_Type  Ward_Facility_Code  Bed
Grade  patientid  City_Code_Patient  Type of Admission  Severity
of Illness  Visitors
with Patient  Age  Admission_Deposit  $
0  318439      21      2      3      2      3      2      3      0  2.0  17006      2.0  0  2  2  7      3095.0
1  318440      29     0      4      0      2      2      3      5  2.0  17006      2.0  1  2  4  7      4018.0
2  318441      26     1      2      1      3      2      1      3  4.0  17006      2.0  0  2  3  7      4492.0
3  318442      6      0      6      0      3      2      1      5  2.0  17006      2.0  1  2  3  7      4173.0
4  318443      28     1      11     0      2      2      2      5  2.0  17006      2.0  1  2  4  7      4161.0
```




+ Code + Text



```
✓ [30] 16 Admission_Deposit      318438 non-null float64
0s      17 Stay                  318438 non-null int64
dtypes: float64(3), int64(15)
memory usage: 46.2 MB
```

```
✓ [31] test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 137057 entries, 0 to 137056
Data columns (total 18 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 int64
3   City_Code_Hospital                       137057 non-null int64
4   Hospital_region_code                     137057 non-null int64
5   Available Extra Rooms in Hospital        137057 non-null int64
6   Department                               137057 non-null int64
7   Ward_Type                                137057 non-null int64
8   Ward_Facility_Code                       137057 non-null int64
9   Bed Grade                                137057 non-null float64
10  patientid                                137057 non-null int64
11  City_Code_Patient                         137057 non-null float64
12  Type of Admission                         137057 non-null int64
13  Severity of Illness                       137057 non-null int64
14  Visitors with Patient                     137057 non-null int64
15  Age                                        137057 non-null int64
16  Admission_Deposit                         137057 non-null float64
17  Stay                                      137057 non-null int64
dtypes: float64(3), int64(15)
memory usage: 19.9 MB
```

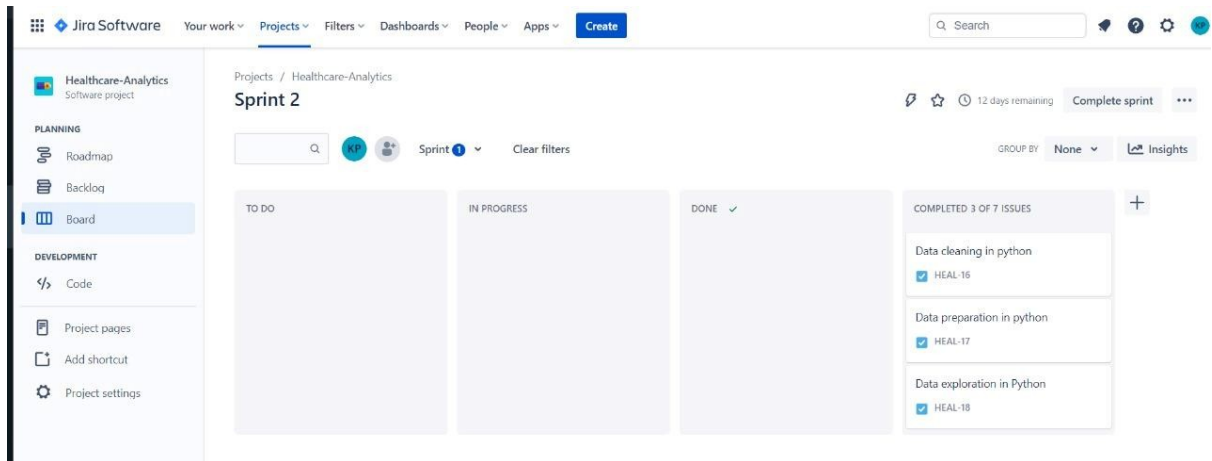
Jira Sprint 2 Tracking:

This screenshot shows the Jira Sprint 2 Tracking interface for the 'Healthcare-Analytics' project. The interface includes a top navigation bar with 'Jira Software', 'Your work', 'Projects', 'Filters', 'Dashboards', 'People', 'Apps', and a 'Create' button. A search bar is located on the right. The left sidebar contains a 'Healthcare-Analytics' project card and a navigation menu with 'PLANNING' (Roadmap, Backlog, Board) and 'DEVELOPMENT' (Code, Project pages, Add shortcut, Project settings). The main content area displays 'Sprint 2' with a search bar, a filter icon, a 'Sprint 1' dropdown, and a 'Clear filters' button. The sprint progress bar shows '12 days remaining' and a 'Complete sprint' button. The 'GROUP BY' dropdown is set to 'None', and an 'Insights' button is visible. The sprint board is divided into four columns: 'TO DO 3 OF 3 ISSUES', 'IN PROGRESS', 'DONE ✓', and 'COMPLETED'. The 'TO DO' column contains three issues: 'Data cleaning in python' (HEAL-16), 'Data preparation in python' (HEAL-17), and 'Data exploration in Python' (HEAL-18). The 'IN PROGRESS', 'DONE', and 'COMPLETED' columns are currently empty.

TO DO 3 OF 3 ISSUES	IN PROGRESS	DONE ✓	COMPLETED
Data cleaning in python ✓ HEAL-16			
Data preparation in python ✓ HEAL-17			
Data exploration in Python ✓ HEAL-18			

This screenshot shows the Jira Sprint 2 Tracking interface after some progress has been made. The sprint progress bar still shows '12 days remaining' and a 'Complete sprint' button. The 'GROUP BY' dropdown is still set to 'None', and the 'Insights' button is visible. The sprint board is divided into four columns: 'TO DO 1 OF 1 ISSUE', 'IN PROGRESS 1 OF 1 ISSUE', 'DONE 1 OF 1 ISSUE ✓', and 'COMPLETED'. The 'TO DO' column contains one issue: 'Data exploration in Python' (HEAL-18). The 'IN PROGRESS' column contains one issue: 'Data preparation in python' (HEAL-17). The 'DONE' column contains one issue: 'Data cleaning in python' (HEAL-16). The 'COMPLETED' column is empty.

TO DO 1 OF 1 ISSUE	IN PROGRESS 1 OF 1 ISSUE	DONE 1 OF 1 ISSUE ✓	COMPLETED
Data exploration in Python ✓ HEAL-18	Data preparation in python ✓ HEAL-17	Data cleaning in python ✓ HEAL-16	



“Uploaded ipynb file in the sprint 2 folder in github.”