Analytics for Hospitals' Health-Care Data

- Submitted by

Varsha K(195002122)
Hirthik Prasaad(195002050)
Gadam Pranav(195002084)
Pranav G(195002083)

Project Report

CHAPTER 1 INTRODUCTION

1.1 : PROJECT OVERVIEW

The terms "heart disease" and "cardiovascular disease" are frequently used interchangeably. Heart disease is a general term that covers a wide range of heart related medical conditions. The irregular health state that directly affects the heart and all of its components is characterized by these medical conditions.

In order to forecast cardiac disease, this study discusses various data mining, big data, and machine learning techniques. Building an important model for the medical system to forecast heart disease or cardiovascular illness requires the use of data mining and machine learning. Our application helps the user in finding out if they have heart disease or not.

They can find out by entering details such as their heart rate, cholesterol, blood pressure etc. A dashboard is also attached along with the results for better understanding where they can compare their blood pressure and similar metrics with other users. This project focuses on Random Forest Classifier. The accuracy of our project is 87% for which is better than most other systems in terms of achieving accuracy quickly.

1.2 : PURPOSE

This project's goal is to determine, depending on the patient's medical characteristics—such as gender, age, chest pain, fasting blood sugar level, etc.—whether they are likely to be diagnosed with any cardiovascular heart illnesses. The leading cause of death in the developed world is heart disease. Heart disease cases are rising quickly every day, thus it's crucial and worrisome to predict any potential illnesses in advance. This diagnosis is a challenging task thatrequires accuracy and efficiency.

Therefore, there needs to be work done to help prevent the risks of having a heart attack or stroke. It is the main factor in adult deaths. By using a person's medical history, our initiative can identify those who are most likely to be diagnosed with a cardiac condition. It can assist in identifying disease with less medical tests and effective therapies, so that patients can be treatedappropriately. It can identify anyone who is experiencing any heart disease symptoms, such as chest pain or high blood pressure.

Around the world, machine learning is applied in many different fields. There is no exception in the healthcare sector. Machine learning may be crucial in determining whether locomotor disorders, heart illnesses, and other conditions are present or absent. If foreseen well in advance, such information can offer valuable insights to doctors, who can then customise their diagnosis and course of care for each patient.

CHAPTER 2 LITERATURE SURVEY

2.1 EXISTING PROBLEM

Recent Covid-19 Pandemic has raised alarms over one of the most overlooked areas to focus: Healthcare Management. While healthcare management has various use cases for using data science, patient length of stay is one critical parameter to observe and predict if one wants to improve the efficiency of the healthcare management in a hospital.

This parameter helps hospitals to identify patients of high LOS-risk (patients who will stay longer) at the time of admission. Once identified, patients with high LOS risk can have their treatment plan optimized to minimize LOS and lower the chance of staff/visitor infection. Also, prior knowledge of LOS can aid in logistics such as room and bed allocation planning.

Suppose you have been hired as Data Scientist of Health Man – a not for profit organization dedicated to manage the functioning of Hospitals in a professional and optimal manner

2.2 REFERENCES

Kenneth Tan et al [1] suggested to forecast the duration of stay using heart-rate readings and physiological ratings. They suggested a model for the heart rate based on the Markov chain model and estimated transition probabilities using the maximum likelihood estimator and the patient population from McMaster Hospital's Neonatal Intensive Care Unit. Using physiological measures and transition probabilities, they then developed maximum likelihood estimators for LOS. Training and test data sets were utilised to validate the linear and nonlinear estimators that were present.

Keijiro Nakamura et al[2] a deep learning model with weighted predictors is created to predict hospitalisation cost and duration of stay using electronic health records. Through 5-fold cross validation, the suggested model outperforms linear regression in terms of prediction accuracy.

Sneha Grampurohit et al[3] proposed effort to develop a Decision Support System to aid clinicians in predicting inpatient hospital length of stay utilising regression models Linear, Ridge, Lasso, and ElasticNet. Mean Absolute Error was used as the evaluative tool for comparing the outcomes of various regressors. The suggested work has produced a graphical user interface as its application.

P.H. Millard et al [4]presented a model-based technique to extracting high-level length-of-stay trends of long-term care residents from a regularly collected administrative social care dataset. It is an expansion of prior work by the authors to include characteristics of inhabitants. Two applications using data given by an English local government were shown to illustrate the potential use of this method. Stephane Sanchez et al [5]contrasted machine learning-based hospital length of stay (LOS) predictions with and without clinical indicators given in text. Two random forests predicted LOS. The first was unstructured EHR text (EHRs). EHR data was assessed using a UMLS-based word-embedding method with precise matching confined to patient-centric affirmation phrases. The second model used structured data from ICD-10 diagnoses and triage codes (CCMU/GEMSA classifications). The model using unstructured data had a 75.0% accuracy compared to 74.1% for the model containing structured data. The two models produced a similar prediction in 86.6% of cases.developed a unique methodological framework based on predictive data mining to estimate LOS (Length Of Stay) in an emergency department (ED). Sondès Chaabane et al [6] Utilized supervised learning, compact models were constructed in terms of predictor characteristics. The objective was to determine the elements (variables) influencing LOS in EDs in order to develop models for LOS prediction. We discovered two linear regression-based models. Validated models were effectively applied to the categorization and prediction of LOS in the paediatric emergency department (PED) at the regional medical centre in Lille, France.

CHAPTER 3 IDEATION & PROPOSED SOLUTION

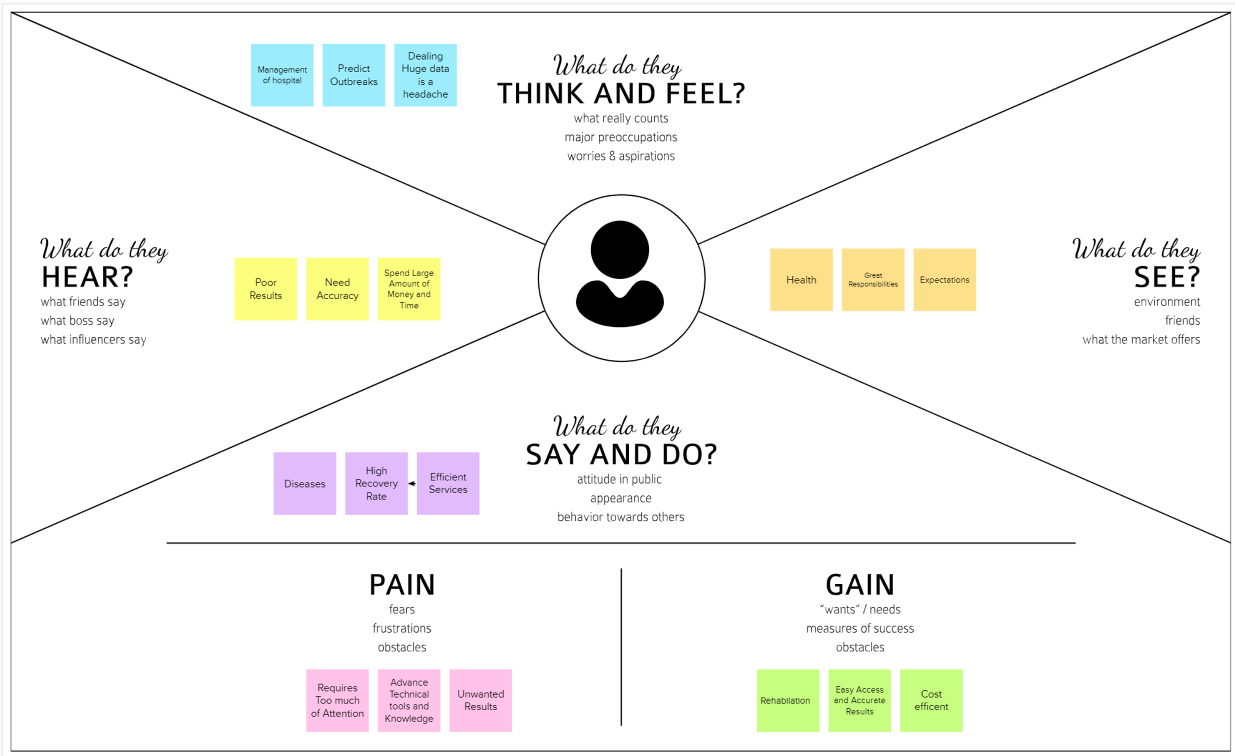
3.1 EMPATHY MAP CANVAS

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.



3.2 IDEATION AND BRAINSTORMING

PRANAV G -TEAM LEADER

1. IDENTIFY PATIENTS OF HIGH LOR RISK
2. OPTIMAL RESOURCE ALLOCATION CAN BE FACILITATED
3. MAKE TIE UPS WITH NGOS
4. PROPOSE A PLAN THAT ATTRACTS MANY HOSPITALS
5. DEVELOP A WEBSITE THAT RUNS 24*7

VARSHA K

ANALYSE VARIOUS HOSPITAL'S IMPACT BECAUSE OF LENGTH OF STAY

ANALYSE HOW COMMON PEOPLE'S COST OF LIVING GETS AFFECTED BECAUSE OF HIGH LOS

INSUFFICENT BEDS AND SUPPLIES LEAD TO MORE DEATH RATES - VISUALISATION

HIRTHIK

PREDICT A PRE DEFINED LOS BASED ON DISEASE

SORT HOSPITALS BASED ON LOS

DASHBOARD FOR AGE WISE AND DEPARTMENT WISE PATIENTS

GADAM PRANAV

- The model is packaged using a container for use in different target environments.

Duties typically include using analytics to determine if resources are being allocated appropriately in a hospital network

PRICING FOR PROJECT=AZURE CALCULATOR
Azure Pricing Calculator saved estimate. This estimate is configured to show the estimated upfront and monthly costs, for a basic implementation that runs 9am-5pm Monday through Friday.

3.3 PROPOSED SOLUTION

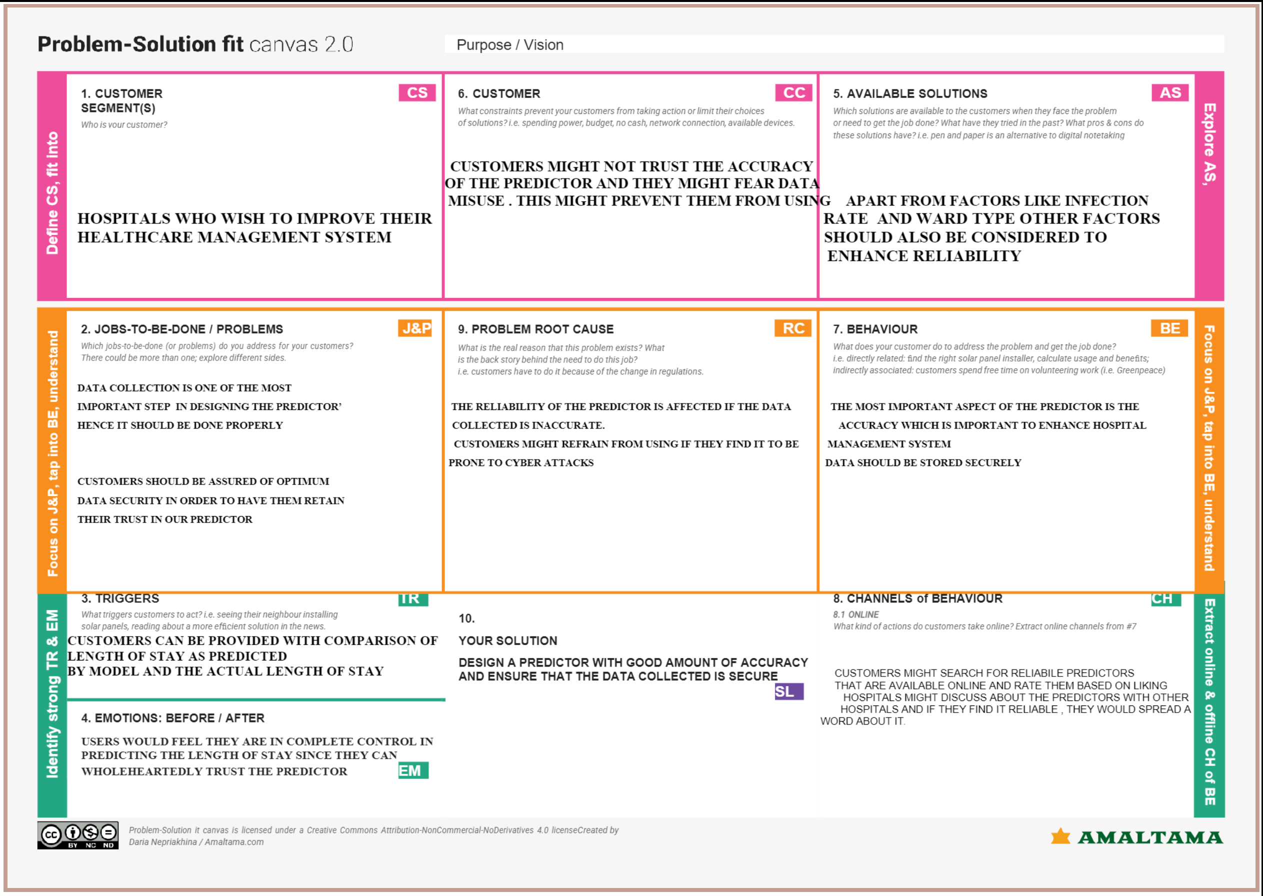
Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Latest Covid-19 Pandemic has raised worries about one of the most neglected areas of focus: Healthcare Administration. While data science has a variety of applications in healthcare management, patient length of stay is one of the most important parameters to watch and anticipate in order to enhance hospital healthcare management efficiency.
2.	Idea / Solution description	This project intends to calculate the length of stay in a particular hospital The key attributes that will be considered for making the decisions are Department ,Ward Type , Severity of Illness etc For determining the % of acceptance, we will be using various ML models such as Logistic Regression, Multiple Linear Regression, Decision Tree & Random Forest and assess which model gives the highest accuracy with the help of performance metrics like accuracy score, precision and recall.
3.	Novelty / Uniqueness	The web-app will predict the length of stay in a particular hospital . It is beneficial for patients, beneficial for the quality of care they get, and beneficial for the financial health of the institution.

4.	Social Impact / Customer Satisfaction	Extended periods of stay have a detrimental effect on hospitals as well. They contribute to a rise in expenses and are often associated with inefficiency, which is an indication that some procedures may need reevaluation. In a similar vein, the duration of stay has a direct influence on bed management, which in turn reduces turnover and affects income. When a patient is held in a bed for a longer period of time than they need, it may mean that the bed is not accessible for another patient who requires it more urgently. This implies that hospitals may not be able to satisfy the demands of their patients. A shorter amount of time spent receiving therapy directly correlates to a shorter overall duration of stay. It is beneficial for patients, beneficial for the quality of care they get, and beneficial for the financial health of the institution.
5.	Business Model (Revenue Model)	1.Advertisements of different hospitals could be placed in the web-app to generate revenue through ads. 2.A separate premium plan could be created where the patients and doctors can input certain details and get information about the length of the stay
6.	Scalability of the Solution	A future update could have chat space where Patients , doctors , administration officials of the hospital can interact .

3.4 PROBLEM SOLUTION FIT



f

CHAPTER 4 REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Details	Submit the documents • Ward type • Severity of illness • Department
FR-4	User Requirements	• Upload all the relevant details in the appropriate location in the website • Based on the uploads, the system would scrape all the necessary information • The length of stay will be predicted based on the scraped information

4.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional Requirements:

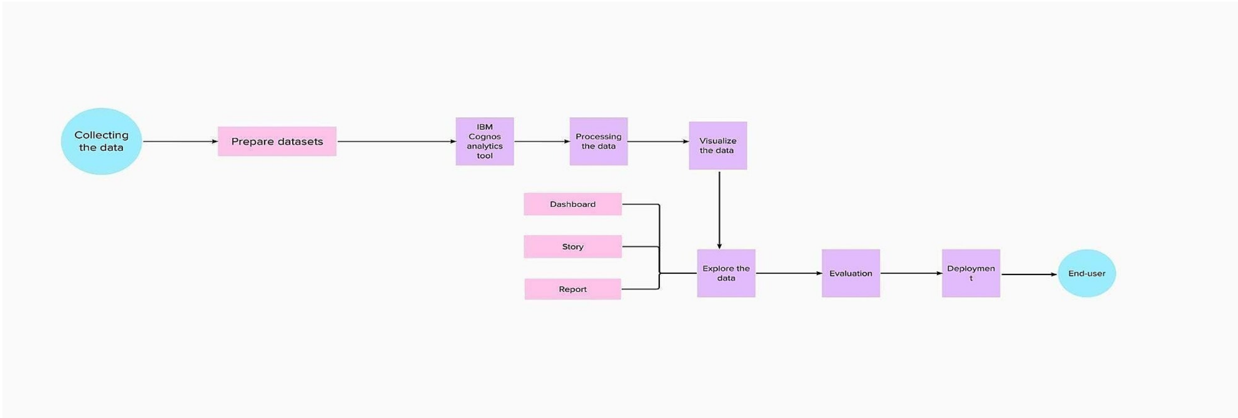
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• The system doesn't expect any technical pre-requisite from the user i.e.; even the naïve user can access it. • User friendly. • Reduced focus on Short Term memory load Focus on Internal Locus of Control. • The page would not take a lot of time to load the content and display them (< 30 seconds).
NFR-2	**Security**	• Only the authenticated user would be able to utilize the services of the site. • Database should be backed up every hour
NFR-3	**Reliability**	• The system would always strive for maximum reliability due to the importance of data and damages that could be cause by

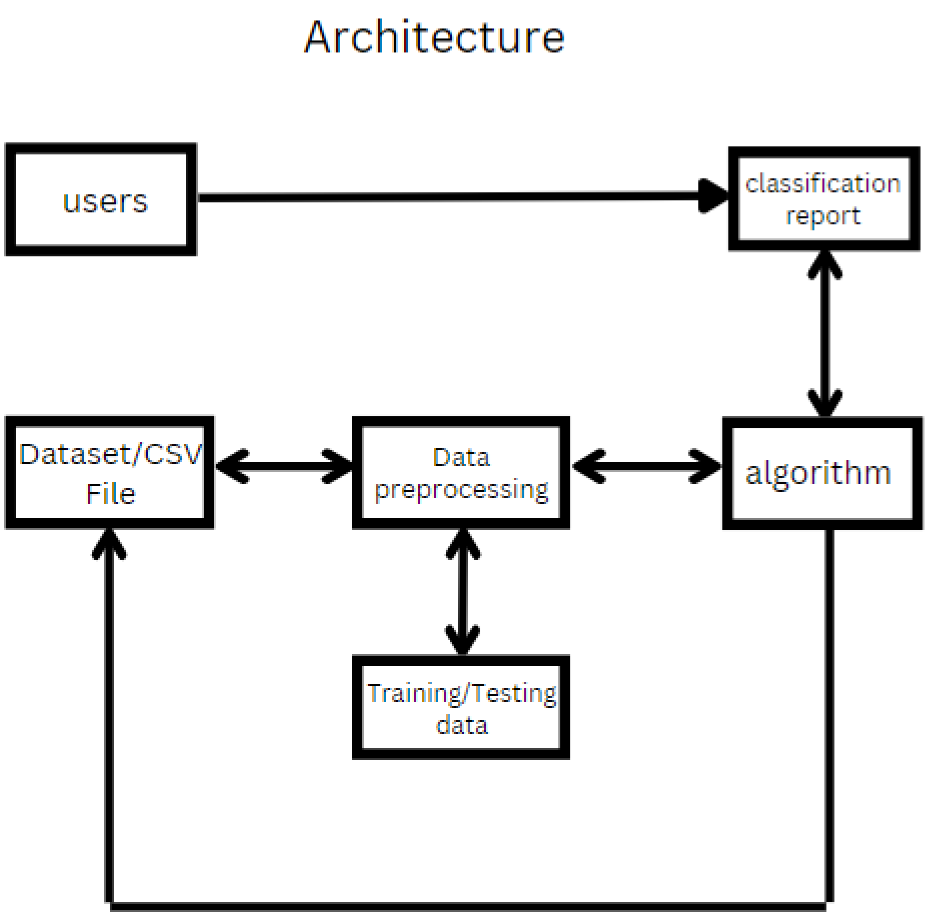
CHAPTER 5 PROJECT DESIGN

Data Flow Diagrams ,Solution & Technical Architecture

5.1 Data Flow Diagrams: A Data Flow Diagram (DFD) is a graphical representation of the flow of data
in a business information system. It describes the processes that are involved in a system to transfer data
from the input to the file storage and reports generation. It shows how data enters and leaves the system,
what changes the information, and where data is stored.



5.2 Solution and Architecture diagram:



Solution Architecture: Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

• Find the best tech solution to solve existing business problems.

• Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.

• Define features, development phases, and solution requirements.

• Provide specifications according to which the solution is defined, managed, and delivered.

5.3 User Stories:

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Hospitals		USN-1	By providing my email address, a password,and a password confirmation, I can register for the program as a user.	I can access my account / dashboard	High	Sprint-1
		USN-2	When I register for the application as auser, I will get a confirmation email.	I can receive confirmation email &click confirm	High	Sprint-1
		USN-3	Using this algorithm both cost and time can be estimated by providing user details	Can be compared with previous length of stay	Medium	Sprint-1

CHAPTER 6 PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION:

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a health care provider I can create account inIBM cloud and the dataarecollected.	20	High	2 Members

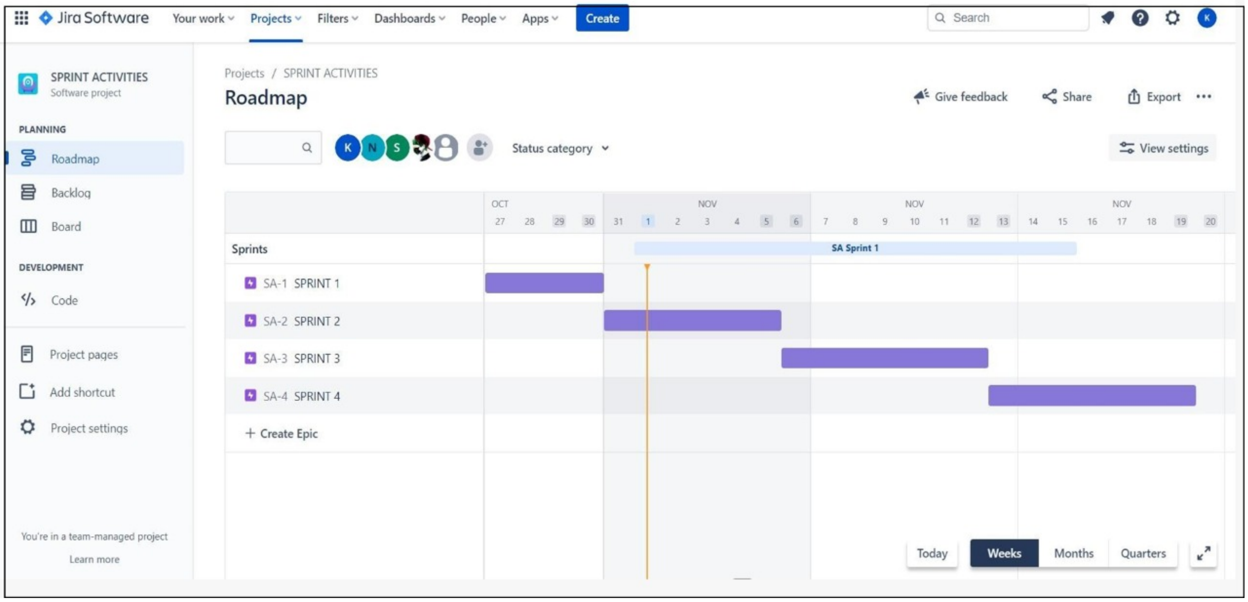
Sprint-2	Analyze	USN-2	As a health care provider all the data thatare collected is cleaned and uploaded inthe database or IBM cloud.	20	Medium	2 Members
Sprint-3	Dashboard	USN-3	As a health care provider I can use my account in my dashboard for uploading dataset.	10	Medium	2 Members
Sprint-3	Visualization	USN-4	As a health care provider I can prepare data for Visualization.	10	High	2 Members
Sprint-4	Visualization	USN-5	As a health care provider I canpresent data in my dashboard.	10	High	2 Members
Sprint-4	Prediction	USN-6	As a health care provider Ican predict the length ofstay	10	High	2 Members

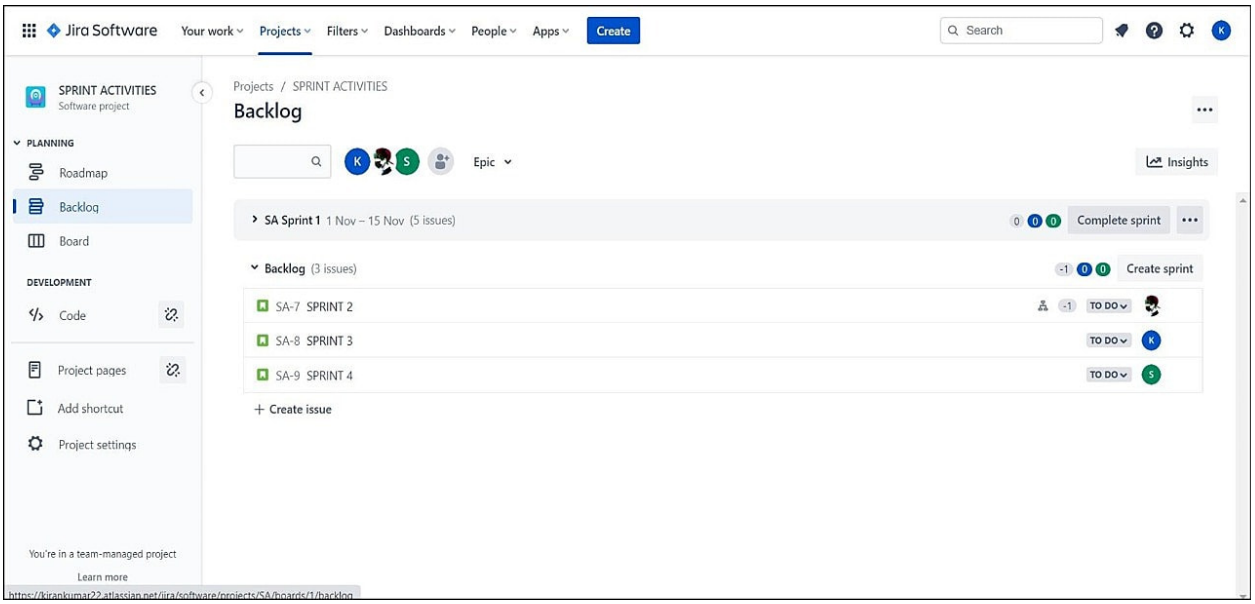
6.2 SPRINT DELIVERY SCHEDULE

Sprint	**Total Story Points**	**Duration**	**Sprint StartDate**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date(Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

:

REPORTS FROM JIRA:





Jira Software

CHAPTER 7 CODING & SOLUTIONING

```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 161,
   "id": "d886d19e",
   "metadata": {},
   "outputs": [],
```

```
   "source": [
    "import numpy as np\n",
    "import pandas as pd\n",
    "import matplotlib.pyplot as plt\n",
    "np.set_printoptions(suppress=True)\n",
    "import warnings\n",
    "warnings.filterwarnings('ignore')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 162,
   "id": "0e06a585",
   "metadata": {},
   "outputs": [],
   "source": [
    "train = pd.read_csv('/Users/hirthikprasaad/Desktop/train.csv')\n",
    "test = pd.read_csv('/Users/hirthikprasaad/Desktop/test.csv')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 163,
   "id": "274ec4b0",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
       "    }\n",
```

```
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>case_id</th>\n",
"      <th>Hospital_code</th>\n",
"      <th>Hospital_type_code</th>\n",
"      <th>City_Code_Hospital</th>\n",
"      <th>Hospital_region_code</th>\n",
"      <th>Available Extra Rooms in Hospital</th>\n",
"      <th>Department</th>\n",
"      <th>Ward_Type</th>\n",
"      <th>Ward_Facility_Code</th>\n",
"      <th>Bed Grade</th>\n",
"      <th>patientid</th>\n",
"      <th>City_Code_Patient</th>\n",
"      <th>Type of Admission</th>\n",
"      <th>Severity of Illness</th>\n",
"      <th>Visitors with Patient</th>\n",
"      <th>Age</th>\n",
"      <th>Admission_Deposit</th>\n",
"      <th>Stay</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>1</td>\n",
"      <td>8</td>\n",
"      <td>c</td>\n",
"      <td>3</td>\n",
"      <td>Z</td>\n",
"      <td>3</td>\n",
"      <td>radiotherapy</td>\n",
"      <td>R</td>\n",
"      <td>F</td>\n",
"      <td>2.0</td>\n",
"      <td>31397</td>\n",
"      <td>7.0</td>\n",
"      <td>Emergency</td>\n",
"      <td>Extreme</td>\n",
```

```
"      <td>2</td>\n",
"      <td>51-60</td>\n",
"      <td>4911.0</td>\n",
"      <td>0-10</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>2</td>\n",
"      <td>2</td>\n",
"      <td>c</td>\n",
"      <td>5</td>\n",
"      <td>Z</td>\n",
"      <td>2</td>\n",
"      <td>radiotherapy</td>\n",
"      <td>S</td>\n",
"      <td>F</td>\n",
"      <td>2.0</td>\n",
"      <td>31397</td>\n",
"      <td>7.0</td>\n",
"      <td>Trauma</td>\n",
"      <td>Extreme</td>\n",
"      <td>2</td>\n",
"      <td>51-60</td>\n",
"      <td>5954.0</td>\n",
"      <td>41-50</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>3</td>\n",
"      <td>10</td>\n",
"      <td>e</td>\n",
"      <td>1</td>\n",
"      <td>X</td>\n",
"      <td>2</td>\n",
"      <td>anesthesia</td>\n",
"      <td>S</td>\n",
"      <td>E</td>\n",
"      <td>2.0</td>\n",
"      <td>31397</td>\n",
"      <td>7.0</td>\n",
"      <td>Trauma</td>\n",
"      <td>Extreme</td>\n",
```

```
"     <td>2</td>\n",
"     <td>51-60</td>\n",
"     <td>4745.0</td>\n",
"     <td>31-40</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>3</th>\n",
"     <td>4</td>\n",
"     <td>26</td>\n",
"     <td>b</td>\n",
"     <td>2</td>\n",
"     <td>Y</td>\n",
"     <td>2</td>\n",
"     <td>radiotherapy</td>\n",
"     <td>R</td>\n",
"     <td>D</td>\n",
"     <td>2.0</td>\n",
"     <td>31397</td>\n",
"     <td>7.0</td>\n",
"     <td>Trauma</td>\n",
"     <td>Extreme</td>\n",
"     <td>2</td>\n",
"     <td>51-60</td>\n",
"     <td>7272.0</td>\n",
"     <td>41-50</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>4</th>\n",
"     <td>5</td>\n",
"     <td>26</td>\n",
"     <td>b</td>\n",
"     <td>2</td>\n",
"     <td>Y</td>\n",
"     <td>2</td>\n",
"     <td>radiotherapy</td>\n",
"     <td>S</td>\n",
"     <td>D</td>\n",
"     <td>2.0</td>\n",
"     <td>31397</td>\n",
"     <td>7.0</td>\n",
"     <td>Trauma</td>\n",
"     <td>Extreme</td>\n",
```

```
"     <td>2</td>\n",
"     <td>51-60</td>\n",
"     <td>5558.0</td>\n",
"     <td>41-50</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
"   case_id  Hospital_code Hospital_type_code  City_Code_Hospital  \\\n",
"0        1              8                  c                   3   \n",
"1        2              2                  c                   5   \n",
"2        3             10                  e                   1   \n",
"3        4             26                  b                   2   \n",
"4        5             26                  b                   2   \n",
"\n",
"  Hospital_region_code  Available Extra Rooms in Hospital    Department  \\\n",
"0                    Z                                  3  radiotherapy   \n",
"1                    Z                                  2  radiotherapy   \n",
"2                    X                                  2    anesthesia   \n",
"3                    Y                                  2  radiotherapy   \n",
"4                    Y                                  2  radiotherapy   \n",
"\n",
"  Ward_Type Ward_Facility_Code  Bed Grade  patientid  City_Code_Patient  \\\n",
"0         R                  F        2.0      31397                7.0   \n",
"1         S                  F        2.0      31397                7.0   \n",
"2         S                  E        2.0      31397                7.0   \n",
"3         R                  D        2.0      31397                7.0   \n",
"4         S                  D        2.0      31397                7.0   \n",
"\n",
"  Type of Admission Severity of Illness  Visitors with Patient    Age  \\\n",
"0         Emergency             Extreme                      2  51-60   \n",
"1            Trauma             Extreme                      2  51-60   \n",
"2            Trauma             Extreme                      2  51-60   \n",
"3            Trauma             Extreme                      2  51-60   \n",
"4            Trauma             Extreme                      2  51-60   \n",
"\n",
"   Admission_Deposit   Stay  \n",
"0             4911.0   0-10  \n",
"1             5954.0  41-50  \n",
"2             4745.0  31-40  \n",
```

```
    "3            7272.0  41-50  \n",
    "4            5558.0  41-50  "
   ]
  },
  "execution_count": 163,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "train.head()"
]
},
{
"cell_type": "code",
"execution_count": 164,
"id": "c096c760",
"metadata": {},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "<class 'pandas.core.frame.DataFrame'>\n",
   "RangeIndex: 318438 entries, 0 to 318437\n",
   "Data columns (total 18 columns):\n",
   " #   Column                             Non-Null Count   Dtype  \n",
   "---  ------                             --------------   -----  \n",
   " 0   case_id                            318438 non-null  int64  \n",
   " 1   Hospital_code                      318438 non-null  int64  \n",
   " 2   Hospital_type_code                 318438 non-null  object \n",
   " 3   City_Code_Hospital                 318438 non-null  int64  \n",
   " 4   Hospital_region_code               318438 non-null  object \n",
   " 5   Available Extra Rooms in Hospital  318438 non-null  int64  \n",
   " 6   Department                         318438 non-null  object \n",
   " 7   Ward_Type                          318438 non-null  object \n",
   " 8   Ward_Facility_Code                 318438 non-null  object \n",
   " 9   Bed Grade                          318325 non-null  float64\n",
   " 10  patientid                          318438 non-null  int64  \n",
   " 11  City_Code_Patient                  313906 non-null  float64\n",
   " 12  Type of Admission                  318438 non-null  object \n",
   " 13  Severity of Illness                318438 non-null  object \n",
```

```
    " 14  Visitors with Patient             318438 non-null  int64  \n",
    " 15  Age                               318438 non-null  object \n",
    " 16  Admission_Deposit                 318438 non-null  float64\n",
    " 17  Stay                              318438 non-null  object \n",
    "dtypes: float64(3), int64(6), object(9)\n",
    "memory usage: 43.7+ MB\n"
   ]
  }
 ],
 "source": [
  "train.info()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 165,
 "id": "1483bd67",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array(['0-10', '41-50', '31-40', '11-20', '51-60', '21-30', '71-80',\n",
     "       'More than 100 Days', '81-90', '61-70', '91-100'], dtype=object)"
    ]
   },
   "execution_count": 165,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "train.Stay.unique()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 166,
 "id": "a7617b2f",
 "metadata": {},
 "outputs": [
  {
```

```
   "data": {
    "text/plain": [
     "City_Code_Patient                  4532\n",
     "Bed Grade                           113\n",
     "Hospital_code                         0\n",
     "Admission_Deposit                     0\n",
     "Age                                   0\n",
     "Visitors with Patient                 0\n",
     "Severity of Illness                   0\n",
     "Type of Admission                     0\n",
     "patientid                             0\n",
     "case_id                               0\n",
     "Ward_Facility_Code                    0\n",
     "Ward_Type                             0\n",
     "Department                            0\n",
     "Available Extra Rooms in Hospital     0\n",
     "Hospital_region_code                  0\n",
     "City_Code_Hospital                    0\n",
     "Hospital_type_code                    0\n",
     "Stay                                  0\n",
     "dtype: int64"
    ]
   },
   "execution_count": 166,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "train.isnull().sum().sort_values(ascending = False)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 167,
 "id": "e4321f9d",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "City_Code_Patient                  2157\n",
```

```
    "Bed Grade                          35\n",
    "case_id                            0\n",
    "Age                                0\n",
    "Visitors with Patient              0\n",
    "Severity of Illness                0\n",
    "Type of Admission                  0\n",
    "patientid                          0\n",
    "Ward_Facility_Code                 0\n",
    "Hospital_code                      0\n",
    "Ward_Type                          0\n",
    "Department                         0\n",
    "Available Extra Rooms in Hospital  0\n",
    "Hospital_region_code               0\n",
    "City_Code_Hospital                 0\n",
    "Hospital_type_code                 0\n",
    "Admission_Deposit                  0\n",
    "dtype: int64"
   ]
  },
  "execution_count": 167,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "test.isnull().sum().sort_values(ascending = False)"
]
},
{
"cell_type": "code",
"execution_count": 168,
"id": "7d9a95fc",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "(318438, 18)"
   ]
  },
  "execution_count": 168,
  "metadata": {},
```

```
   "output_type": "execute_result"
  }
 ],
 "source": [
  "train.shape"
 ]
},
{
 "cell_type": "code",
 "execution_count": 169,
 "id": "d8044bf9",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "(137057, 17)"
    ]
   },
   "execution_count": 169,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "test.shape"
 ]
},
{
 "cell_type": "code",
 "execution_count": 170,
 "id": "99650865",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "case_id : 318438\n",
    "Hospital_code : 32\n",
    "Hospital_type_code : 7\n",
    "City_Code_Hospital : 11\n",
```

"Hospital_region_code : 3\n",
"Available Extra Rooms in Hospital : 18\n",
"Department : 5\n",
"Ward_Type : 6\n",
"Ward_Facility_Code : 6\n",
"Bed Grade : 4\n",
"patientid : 92017\n",
"City_Code_Patient : 37\n",
"Type of Admission : 3\n",
"Severity of Illness : 3\n",
"Visitors with Patient : 28\n",
"Age : 10\n",
"Admission_Deposit : 7300\n",
"Stay : 11\n"
]
}
],
"source": [
"for i in train.columns:\n",
"   print(i, ':', train[i].nunique())"
]
},
{
"cell_type": "code",
"execution_count": 171,
"id": "9369c3d3",
"metadata": {},
"outputs": [],
"source": [
"train['Bed Grade'].fillna(train['Bed Grade'].mode()[0], inplace = True)\n",
"test['Bed Grade'].fillna(test['Bed Grade'].mode()[0], inplace = True)"
]
},
{
"cell_type": "code",
"execution_count": 172,
"id": "de00e3ef",
"metadata": {},
"outputs": [],
"source": [
"train['City_Code_Patient'].fillna(train['City_Code_Patient'].mode()[0], inplace = True)\n",
"test['City_Code_Patient'].fillna(test['City_Code_Patient'].mode()[0], inplace = True)"

```
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 173,
  "id": "e155efa5",
  "metadata": {},
  "outputs": [],
  "source": [
   "from sklearn.preprocessing import LabelEncoder\n",
   "le = LabelEncoder()\n",
   "train['Stay'] = le.fit_transform(train['Stay'].astype('str'))"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 174,
  "id": "2d7b0d90",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "(455495, 18)"
     ]
    },
    "execution_count": 174,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "test['Stay'] = -1\n",
   "df = pd.concat([train, test])\n",
   "df.shape"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 175,
  "id": "0b66cd25",
  "metadata": {},
```

```
   "outputs": [],
   "source": [
    "for i in ['Hospital_type_code', 'Hospital_region_code', 'Department',\n",
    "          'Ward_Type', 'Ward_Facility_Code', 'Type of Admission', 'Severity of Illness', 'Age']:\n",
    "    le = LabelEncoder()\n",
    "    df[i] = le.fit_transform(df[i].astype(str))"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 176,
   "id": "59971518",
   "metadata": {},
   "outputs": [],
   "source": [
    "train = df[df['Stay']!=-1]\n",
    "test = df[df['Stay']==-1]"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 177,
   "id": "02e21fda",
   "metadata": {},
   "outputs": [],
   "source": [
    "def get_countid_enocde(train, test, cols, name):\n",
    "  temp = train.groupby(cols)['case_id'].count().reset_index().rename(columns = {'case_id':
name})\n",
    "  temp2 = test.groupby(cols)['case_id'].count().reset_index().rename(columns = {'case_id':
name})\n",
    "  train = pd.merge(train, temp, how='left', on= cols)\n",
    "  test = pd.merge(test,temp2, how='left', on= cols)\n",
    "  train[name] = train[name].astype('float')\n",
    "  test[name] = test[name].astype('float')\n",
    "  train[name].fillna(np.median(temp[name]), inplace = True)\n",
    "  test[name].fillna(np.median(temp2[name]), inplace = True)\n",
    "  return train, test"
   ]
  },
  {
   "cell_type": "code",
```

```
   "execution_count": 178,
   "id": "6060fc4d",
   "metadata": {},
   "outputs": [],
   "source": [
    "train, test = get_countid_enocde(train, test, ['patientid'], name = 'count_id_patient')\n",
    "train, test = get_countid_enocde(train, test, \n",
    "                                 ['patientid', 'Hospital_region_code'], name =
'count_id_patient_hospitalCode')\n",
    "train, test = get_countid_enocde(train, test, \n",
    "                                 ['patientid', 'Ward_Facility_Code'], name =
'count_id_patient_wardfacilityCode')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 179,
   "id": "94ae91ea",
   "metadata": {},
   "outputs": [],
   "source": [
    "test1 = test.drop(['Stay', 'patientid', 'Hospital_region_code', 'Ward_Facility_Code'], axis =1)\n",
    "train1 = train.drop(['case_id', 'patientid', 'Hospital_region_code', 'Ward_Facility_Code'], axis =1)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 180,
   "id": "79bffe9b",
   "metadata": {},
   "outputs": [],
   "source": [
    "X1 = train1.drop('Stay', axis =1)\n",
    "y1 = train1['Stay']\n",
    "from sklearn.model_selection import train_test_split\n",
    "X_train, X_test, y_train, y_test = train_test_split(X1, y1, test_size =0.20, random_state =100)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 181,
   "id": "df59b059",
```

```
  "metadata": {},
  "outputs": [],
  "source": [
   "from sklearn.neighbors import KNeighborsClassifier\n",
   "from sklearn.linear_model import LogisticRegression\n",
   "from sklearn.tree import DecisionTreeClassifier"
  ]
 },
 {
  "cell_type": "markdown",
  "id": "3c0e8956",
  "metadata": {},
  "source": [
   "# logistic"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 182,
  "id": "fca9facd",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "0.35359764474975464\n",
     "0.35279801532470795\n"
    ]
   }
  ],
  "source": [
   "clf = LogisticRegression(max_iter=1000)\n",
   "clf.fit(X_train, y_train)\n",
   "print(clf.score(X_train, y_train))\n",
   "print(clf.score(X_test, y_test))"
  ]
 },
 {
  "cell_type": "markdown",
  "id": "0bba38a4",
  "metadata": {},
```

```
  "source": [
   "# ANN\n"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 183,
  "id": "f194a381",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "Index(['case_id', 'Hospital_code', 'Hospital_type_code', 'City_Code_Hospital',\n",
     "       'Hospital_region_code', 'Available Extra Rooms in Hospital',\n",
     "       'Department', 'Ward_Type', 'Ward_Facility_Code', 'Bed Grade',\n",
     "       'patientid', 'City_Code_Patient', 'Type of Admission',\n",
     "       'Severity of Illness', 'Visitors with Patient', 'Age',\n",
     "       'Admission_Deposit', 'count_id_patient',\n",
     "       'count_id_patient_hospitalCode', 'count_id_patient_wardfacilityCode'],\n",
     "      dtype='object')\n",
     "Index(['case_id', 'Hospital_code', 'Hospital_type_code', 'City_Code_Hospital',\n",
     "       'Hospital_region_code', 'Available Extra Rooms in Hospital',\n",
     "       'Department', 'Ward_Type', 'Ward_Facility_Code', 'Bed Grade',\n",
     "       'patientid', 'City_Code_Patient', 'Type of Admission',\n",
     "       'Severity of Illness', 'Visitors with Patient', 'Age',\n",
     "       'Admission_Deposit', 'count_id_patient',\n",
     "       'count_id_patient_hospitalCode', 'count_id_patient_wardfacilityCode'],\n",
     "      dtype='object')\n"
    ]
   },
   {
    "data": {
     "text/plain": [
      "(318438, 20)"
     ]
    },
    "execution_count": 183,
    "metadata": {},
    "output_type": "execute_result"
   }
```

],
"source": [
"X = train.drop('Stay', axis =1)\n",
"y = train['Stay']\n",
"print(X.columns)\n",
"z = test.drop('Stay', axis = 1)\n",
"print(z.columns)\n",
"\n",
"# Data Scaling\n",
"from sklearn import preprocessing\n",
"X_scale = preprocessing.scale(X)\n",
"X_scale.shape"
]
},
{
"cell_type": "code",
"execution_count": 184,
"id": "6643b9a9",
"metadata": {},
"outputs": [],
"source": [
"X_train, X_test, y_train, y_test = train_test_split(X_scale, y, test_size =0.20, random_state =100)"
]
},
{
"cell_type": "code",
"execution_count": null,
"id": "416e5289",
"metadata": {},
"outputs": [],
"source": [
"import keras\n",
"from keras.models import Sequential\n",
"from keras.layers import Dense\n",
"import tensorflow as tf"
]
},
{
"cell_type": "code",
"execution_count": null,
"id": "99d79add",

```
  "metadata": {},
  "outputs": [],
  "source": [
   "from keras.utils import to_categorical\n",
   "#Sparse Matrix\n",
   "a = to_categorical(y_train)\n",
   "b = to_categorical(y_test)\n",
   "print(scores)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "id": "72f0e86b",
  "metadata": {},
  "outputs": [],
  "source": [
   "model = Sequential()\n",
   "model.add(Dense(64, activation='relu', input_shape = (254750, 20))) \n",
   "model.add(Dense(128, activation='relu'))\n",
   "model.add(Dense(256, activation='relu'))\n",
   "model.add(Dense(512, activation='relu'))\n",
   "model.add(Dense(512, activation='relu'))\n",
   "model.add(Dense(11, activation='softmax'))"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "id": "704ad983",
  "metadata": {},
  "outputs": [],
  "source": [
   "model.summary()"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "id": "4fbd2fa0",
  "metadata": {},
  "outputs": [],
```

```
 "source": [
  "model.compile(optimizer= 'SGD', \n",
  "              loss='categorical_crossentropy', \n",
  "              metrics=['accuracy'])"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "id": "42b29be4",
 "metadata": {},
 "outputs": [],
 "source": [
  "callbacks = [tf.keras.callbacks.TensorBoard(\"logs_keras\")]\n",
  "model.fit(X_train, a, epochs=20, callbacks=callbacks, validation_split = 0.2)"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "id": "c529e1c0",
 "metadata": {},
 "outputs": [],
 "source": [
  "!tensorboard --logdir logs_keras"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "id": "7129c66b",
 "metadata": {},
 "outputs": [],
 "source": [
  "model.fit(X_train, a, epochs=4, validation_split = 0.2)\n",
  "print(\"\\n Model Evaluation\")\n",
  "model.evaluate(X_test,b)"
 ]
},
{
 "cell_type": "markdown",
 "id": "c2efb55d",
```

```
  "metadata": {},
  "source": [
   "# XgBoost\n"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "id": "6e9161fc",
  "metadata": {},
  "outputs": [],
  "source": [
   "import xgboost\n",
   "classifier_xgb = xgboost.XGBClassifier(max_depth=4, learning_rate=0.1,
n_estimators=800,\n",
   "                                objective='multi:softmax', reg_alpha=0.5, reg_lambda=1.5,\n",
   "                                booster='gbtree', n_jobs=4, min_child_weight=2, base_score= 0.75)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "id": "dd5917ee",
  "metadata": {},
  "outputs": [],
  "source": [
   "model_xgb = classifier_xgb.fit(X_train, y_train)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "id": "aa94a83b",
  "metadata": {},
  "outputs": [],
  "source": [
   "prediction_xgb = model_xgb.predict(X_test)\n",
   "acc_score_xgb = accuracy_score(prediction_xgb,y_test)\n",
   "print(\"Accuracy:\", acc_score_xgb*100)"
  ]
 },
 {
```

```
  "cell_type": "markdown",
  "id": "b3679dbc",
  "metadata": {},
  "source": [
   "# predictions"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "id": "16a18feb",
  "metadata": {},
  "outputs": [],
  "source": [
   "pred_nb = clf.predict(X_test)\n",
   "result_nb = pd.DataFrame(pred_nb, columns=['Stay'])\n",
   "result_nb['case_id'] = test1['case_id']\n",
   "result_nb = result_nb[['case_id', 'Stay']]"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "id": "5a9976db",
  "metadata": {},
  "outputs": [],
  "source": [
   "result_nb['Stay'] = result_nb['Stay'].replace({0:'0-10', 1: '11-20', 2: '21-30', 3:'31-40', 4: '41-50', 5:
'51-60', 6: '61-70', 7: '71-80', 8: '81-90', 9: '91-100', 10: 'More than 100 Days'})\n",
   "result_nb.head()"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "id": "37ea741c",
  "metadata": {},
  "outputs": [],
  "source": [
   "pred_xgb = classifier_xgb.predict(test1.iloc[:,1:])\n",
   "result_xgb = pd.DataFrame(pred_xgb, columns=['Stay'])\n",
   "result_xgb['case_id'] = test1['case_id']\n",
```

```
    "result_xgb = result_xgb[['case_id', 'Stay']]"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "e0a40d4e",
   "metadata": {},
   "outputs": [],
   "source": [
    "result_xgb['Stay'] = result_xgb['Stay'].replace({0:'0-10', 1: '11-20', 2: '21-30', 3:'31-40', 4: '41-50', 5: '51-60', 6: '61-70', 7: '71-80', 8: '81-90', 9: '91-100', 10: 'More than 100 Days'})\n",
    "result_xgb.head()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "885a0652",
   "metadata": {},
   "outputs": [],
   "source": [
    "test_scale = preprocessing.scale(z)\n",
    "test_scale.shape"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "2851016b",
   "metadata": {},
   "outputs": [],
   "source": [
    "pred = model.predict_classes(test_scale)\n",
    "pred"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "af40cfd9",
   "metadata": {},
```

"outputs": [],
"source": [
 "result_nn = pd.DataFrame(pred, columns=['Stay'])\n",
 "result_nn['case_id'] = test['case_id']\n",
 "result_nn = result_nn[['case_id', 'Stay']]"
]
},
{
"cell_type": "code",
"execution_count": null,
"id": "4376fc2b",
"metadata": {},
"outputs": [],
"source": [
 "result_nn['Stay'] = result_nn['Stay'].replace({0:'0-10', 1: '11-20', 2: '21-30', 3:'31-40', 4: '41-50', 5: '51-60', 6: '61-70', 7: '71-80', 8: '81-90', 9: '91-100', 10: 'More than 100 Days'})\n",
 "result_nn.head()"
]
},
{
"cell_type": "markdown",
"id": "c4445b7a",
"metadata": {},
"source": [
 "# Results"
]
},
{
"cell_type": "code",
"execution_count": null,
"id": "354ad227",
"metadata": {},
"outputs": [],
"source": [
 "print(result_nb.groupby('Stay')['case_id'].nunique())"
]
},
{
"cell_type": "code",
"execution_count": null,
"id": "95c26fea",
"metadata": {},

```
   "outputs": [],
   "source": [
    "print(result_xgb.groupby('Stay')['case_id'].nunique())"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "3f71b038",
   "metadata": {},
   "outputs": [],
   "source": [
    "print(result_nn.groupby('Stay')['case_id'].nunique())"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "dc954013",
   "metadata": {},
   "outputs": [],
   "source": []
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.9.12"
  }
```

```
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```