

# **ANALYTICS FOR HOSPITAL'S HEALTHCARE DATA PROJECT REPORT**

**SUBMITTED BY**

**SUBALAKSHMI G-737819CSR197**

**SUBHIKSHA S-737819CSR198**

**SUJA S-737819CSR201**

**THAARANI S-737819CSR207**

**TEAM ID: PNT2022TMID04490**

# 1. INTRODUCTION

## 1.1 Project overview:

The pressure on healthcare institutions to enhance patient outcomes and provide better care is expanding. Even while this situation is difficult, it also gives enterprises a chance to significantly raise the standard of care by utilizing additional information and insights from their data. Health care analytics is the term for the efficient analysis of data to discover patterns and trends in the collected data. The average duration of stay for a patient is one of many performance measures used in healthcare management. With the help of the project Hospitals can tailor their treatment programmes to minimize length of stay (LOS) and cut down on infection rates among patients, workers, and all the people in the hospital.

## 1.2. Purpose

The project objective is to precisely estimate each patient's length of stay, in order to effectively utilize hospital resources.

# 2. LITERATURE SURVEY

## 2.1 Existing problem

Covid-19 recently One of the most neglected areas to concentrate on has come under scrutiny due to the pandemic: healthcare management. Patient duration of stay is a crucial statistic to monitor and forecast if one wishes to increase the effectiveness of healthcare management in a hospital, even if there are many use cases for data science in healthcare management.

## 2.2. References

- <https://www.sciencedirect.com/science/article/pii/S2352914822000855>
- <https://www.sciencedirect.com/science/article/pii/S2666827022000603>
- [https://www.researchgate.net/publication/355174497\\_Robust\\_Length\\_of\\_Stay\\_Prediction\\_Model\\_for\\_Indoor\\_Patients](https://www.researchgate.net/publication/355174497_Robust_Length_of_Stay_Prediction_Model_for_Indoor_Patients)
- <https://www.sciencedirect.com/science/article/pii/S2090447921001349>
- <https://towardsdatascience.com/predicting-inpatient-length-of-stay-at-hospitals-using-python-big-data-304e79d8c008>

## 2.3. Problem statement

The goal is to correctly anticipate the length of stay for each patient on a case-by-case basis so that hospitals may utilize this data to better allocate resources and operate. The length of stay is divided into 11 different classes ranging from 0-10 days to more than 100 days.

S.NO	PAPER	AUTHOR	YEAR	METHOD AND ALGORITHM	ACCURACY
1	Machine Learning model for predicting the length of stay in the intensive care unit for covid – 19 patients in the eastern province of Saudi Arabia	Dina A. Alabbad, Abdullah M. Almuhaideb, Shikah J. Alsunaidi, Kawther Alqudhaihai, Fatimah A. Alamoudi, Maha K. Alhobaishi, Naimah A. Alaqeel, Mohammed S. Alshahrani	2022	To predict the length of stay of a patient. Here we employed four algorithm Random Forest(RF), Gradient Boosting(GB), Extreme Gradient Boosting (XG Boost), Ensemble Models. Through this experiments the prediction is done in this algorithm Random Forest gives the highest accuracy when compared to other methods.	94.16%
2	Time - to – event modelling for hospital length of stay prediction for covid – 19 patients	Y. Wen, M.F. Rahman, Y.Zhuang et al, Michael Pokojovy, Honglun Xu, Peter McCaffrey, Alexander Vo, Eric Walser, Scott Moen, Tzu-Liang (Bill) Tseng	2022	This study uses a technique called time - to – event modelling which is also known as survival analysis. It uses algorithm like Logistic regression, Random forests, Support Vector Machine, Decision free – based methods. The survival analysis is a branch of statistics concerned with analysing time - to - event data and predicting the probability of occurrence of an event. The event could be any format	70%

3	Robust length of stay prediction model for indoor patients	Ayesha Siddiqi, Syed Abbas Zilqurnian Naqvi, Ahsan Naeem, Allah Ditta, Hani Alquahayz, Muhammad Adnan Khan	2021	The length of stay of patient of different disease is identified. So that the hospital can manage the available resources and new patient getting entries for their prompt treatment. Here they use algorithms such as Ridge Regression(RR), Decision Tree Regression Extreme Gradient Boosting Regression (XGBR), Random Forest Regression (RFR). Process like Raw dataset are processed then exploring the data, Machine learning modelling, performance measuring, selection of robust model based on the performance.	92%
4	Predicting length of stay in hospitals intensive care unit using general admission features	Merhan A. Abd-Elrazek, Ahmed A. Eltahawi, Mohamed H. Abd Elaziz, Mohamed N. Abd-Elwhab	2021	This paper is based on length of stay of patient in the ICU. Here the data is pre-processed and the dataset is divided into K fold cross validation. ML techniques used are Neural Networks(NN), Classification Tree(CT), Tree Bagges(TB), Random Forest(RF), Fuzzy Logic(FL), Support Vector Machine(SVM), KNN, Regression Tree(RT) and Navie Bayes(NB). Proposed techniques are data acquisition, data pre-processing, data transformation, training and testing	92%

5	Predicting inpatient length of stay at hospitals using python + bigdata	Vishal Tien	2020	In this study the paper describes to create a model that can predict length of stay for patients upon admission to a hospital. The algorithms used is Logistic Regression, Boosted Decision Tree, Random forest. In this the APR DRGcode, a classification system that classify patients according to the reason of admission, severity of illness and a risk of mortality and the APR severity of illness score are the most important feature In predicting the patients length of stay.	70%
---	-------------------------------------------------------------------------	-------------	------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----

### 3. IDEATION & PROPOSED SOLUTION

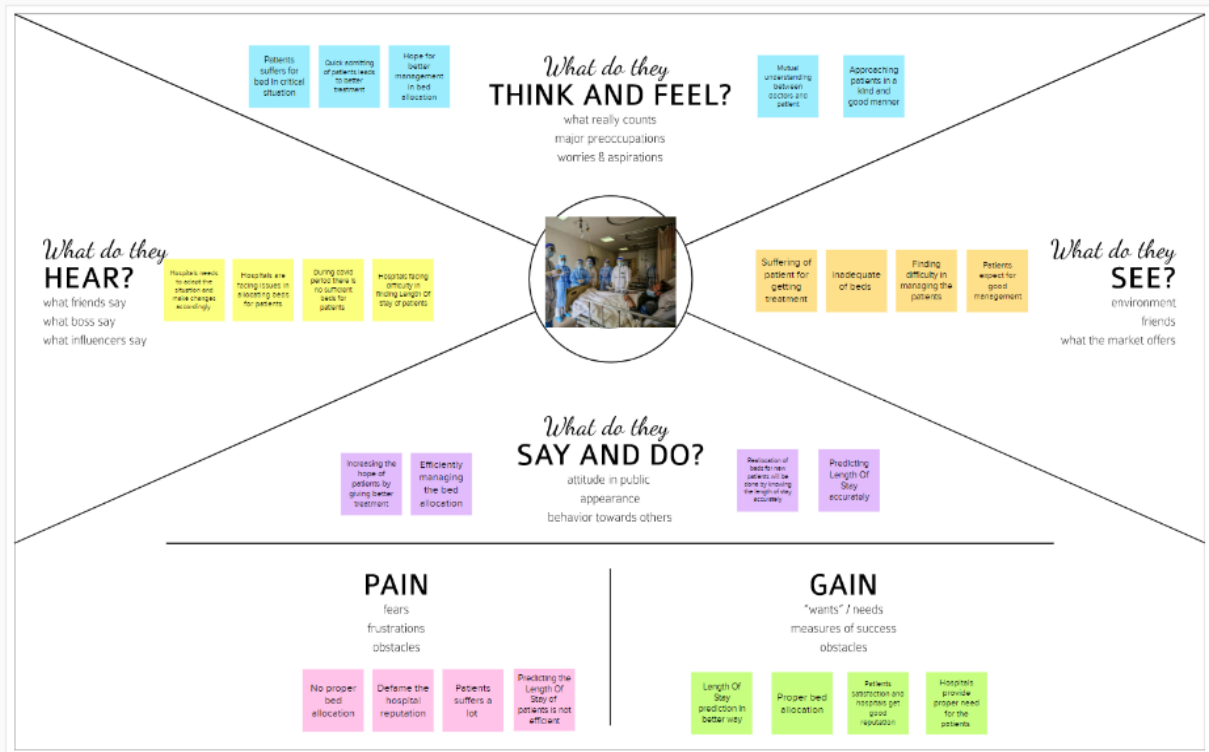
#### 3.1 Empathy map canvas

# Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Analytics For Hospitals' Health-Care Data Empathy map



## 3.2 Ideation and Brainstorming

1

### Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕒 5 minutes

#### Problem Statement

Analytics For Hospitals' Health-Care Data

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

## Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

### TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

#### Suja S

- Data Collection for the improvement the testing and development
- Using Machine Learning algorithm
- By referring the previous year data
- Choosing the algorithm which gives better accuracy

#### Subalakshmi G

- Make sure the environment where you are surviving is clean
- Improving the prediction for the next stage of treatment
- Identifying the requirements of new medicines
- Analyzing the patient health frequently for better treatment

#### Thaarani S

- Calculating Length Of Stay
- Keep Social Distance to reduce the spread
- Overcoming crisis
- Optimizing organization management

#### Subhiksha S

- Quick admission of patients for better treatment
- Improving the results from the hospital management side
- Efficiently managing the bed allocation
- Make sure the patients and doctors wear mask

3

## Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

🕒 20 minutes

#### LOS prediction

- Data Collection for the improvement the testing and development
- Using Machine Learning algorithm
- Choosing the algorithm which gives better accuracy
- Improving the prediction for the next stage of treatment

#### Improving performance

- By referring the previous year data
- Identifying the requirements of new medicines

#### Safety measures

- Make sure the environment where you are surviving is clean
- Keep Social Distance to reduce the spread
- Make sure the patients and doctors wear mask

### TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

#### Management

- Optimizing organization management
- Quick admission of patients for better treatment
- Efficiently managing the bed allocation
- Improving the results from the hospital management side

#### Target

- Analyzing the patient health frequently for better treatment
- Overcoming crisis
- Calculating Length Of Stay

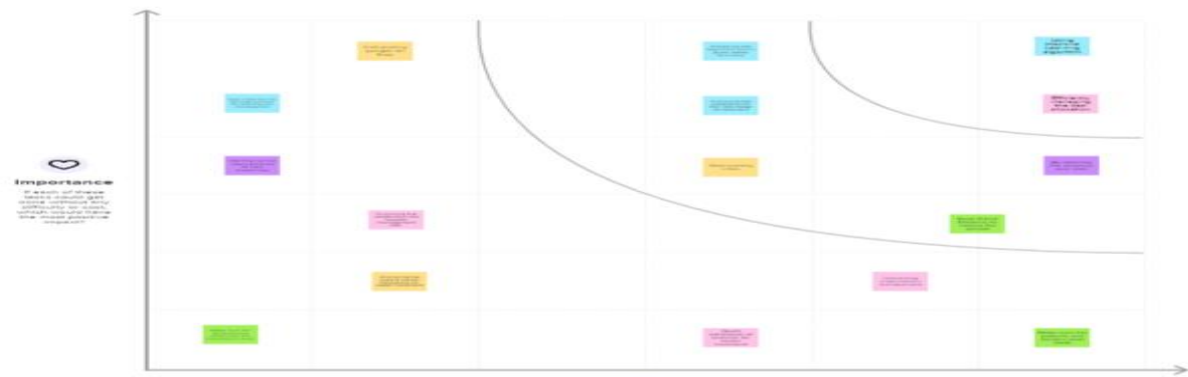




#### Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⌚ 20 minutes

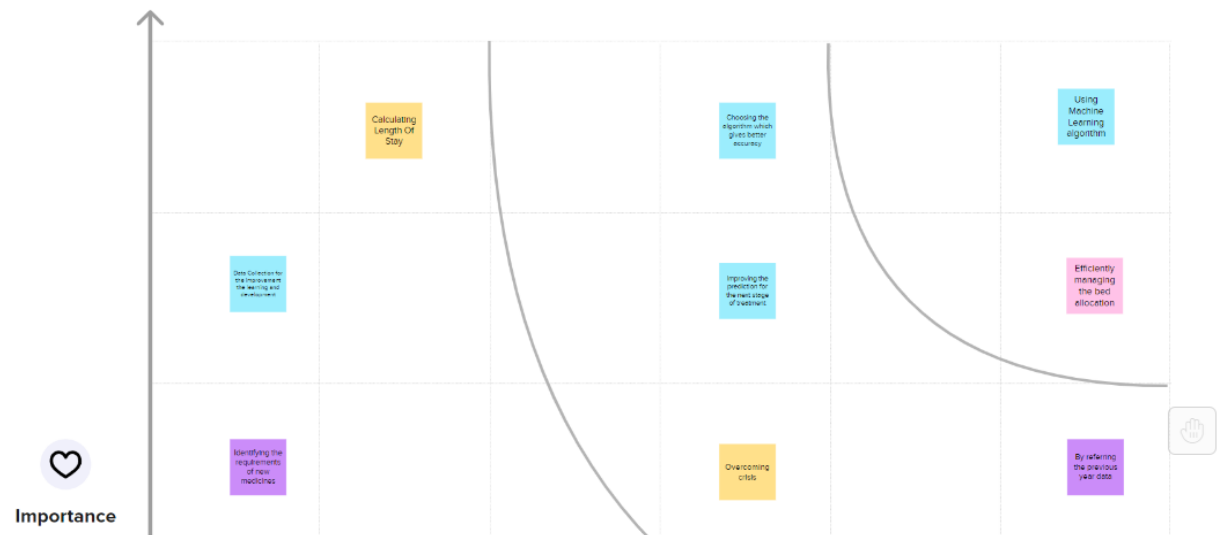


4

#### Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⌚ 20 minutes





### 3.3 Proposed solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Analytics for Hospitals' Health Care Data. To analyse the Length Of Stay (LOS) of patient. To have a better allocation of beds for patients.
2.	Idea / Solution description	Developing a project for Hospitals to analyse the Length Of Stay for the current patients. And allocate the bed for further patients. By using the machine learning techniques, the Length Of Stay can be predicted accurately
3.	Novelty / Uniqueness	There are many applications that analyse the Length Of Stay of a patient. Our focus is to propose the machine learning technique. And use the recent algorithms which predict accurately. Analyze based on up to date, data of the patient.
4.	Social Impact / Customer Satisfaction	For example, in critical situation like covid, it is useful for hospitals to analyze the Length Of Stay and allocate beds for the patients. It will be useful to overcome the difficulty faced by the patients and the hospitals.
5.	Business Model (Revenue Model)	Right now the application is profitless but in future we might add an option of premium plans for advanced learning.
6.	Scalability of the Solution	Based on the situation the patients visiting the hospitals may change. This project is scalable for all hospitals in any kind of situation.

### 3.4 Problem solution fit

Define CS, fit into CC	<div>1. CUSTOMER SEGMENT(S)<div>CS</div></div> <div>Here, the term client refers to all individuals, including children, adults, and seniors, who are suffering from ailments, or patients.</div>	<div>6. CUSTOMER CONSTRAINTS<div>C</div></div> <div>This work is very simple and it will be easy to use by the customer and easily understandable. It is user friendly. Here the data exploration and viewing the data makes them clear. Through the diagrammatic representations, the customer can easily understand</div>	<div>5. AVAILABLE SOLUTIONS<div>AS</div></div> <div>We can use human calculation. Instead of utilizing a human to calculate, a machine learning algorithm is used, which improves computation accuracy and efficiency.</div>	Explore AS, differentiate
	<div>2. JOBS-TO-BE-DONE / PROBLEMS<div>J&amp;P</div></div> <div>Solving the problem of allocating beds for patients.</div> <div>JOBS-TO-BE_DONE:</div> <div><div>Upload the dataset of patient</div><div>Prepare the data</div><div>explore the data</div><div>According to the rules the necessary operations are done</div><div>Visualize the data</div><div>Using the algorithms, the allocation of bed for patients can be predicted accurately</div></div>	<div>9. PROBLEM ROOT CAUSE<div>RC</div></div> <div>Diseases affect a large number of the population. The majority of people who reside in rural locations remote from cities experience issues.</div> <div>Due to unhygienic environment and Less awareness on health among the people it may cause problems</div>	<div>7. BEHAVIOUR<div>BE</div></div> <div>If the patients see in online, they can use if they find difficult to admit in a hospital they may move to other hospital. Through the resources they can get solution for their problems</div>	
<div>3.TRIGGERS<div>TR</div></div> <div>The process may take more time</div>	<div>10. YOUR SOLUTION<div>SL</div></div> <div>We will anticipate the length of stay in this case using the machine learning technique. utilizing an algorithm that provides more accuracy. so that the sufferers might receive care swiftly and become healthy.</div>	<div>8. CHANNELS of BEHAVIOUR<div>CH</div></div> <div>8.1 ONLINE</div> <div>Booking the beds in online which save the time for patients</div> <div>8.2 OFFLINE</div> <div>Based on the hospitals database they can allocate the beds for patient</div>		
<div>4. EMOTIONS: BEFORE / AFTER<div>EM</div></div> <div><div>Disappointment</div><div>Angry</div><div>Frustrated</div><div>Neglected</div><div>Mental pressure</div><div>Emotional</div><div>Unhealthy</div></div>				

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	<b>User Registration</b>	Registration through Form
FR-2	<b>User Confirmation</b>	Confirmation via EmailConfirmation via OTP
FR-3	<b>Patient Report</b>	The patient Report is made up of the patient's database, which includes their personal information and the name of the doctor they are seeing. Ward information and, if available, medical information.
FR-4	<b>Discharge management</b>	Update the information about patients leaving the hospital after receiving care, then including that bed in the list of available beds.
FR-5	<b>Operability</b>	Collect patient data and make it operable among the management.
FR-6	<b>Ward conformation</b>	Confirmation of bed for the patient if the bed is available. They can check the availability of bed through the information provided in the dashboard.

## 4.2. Non-functional requirements

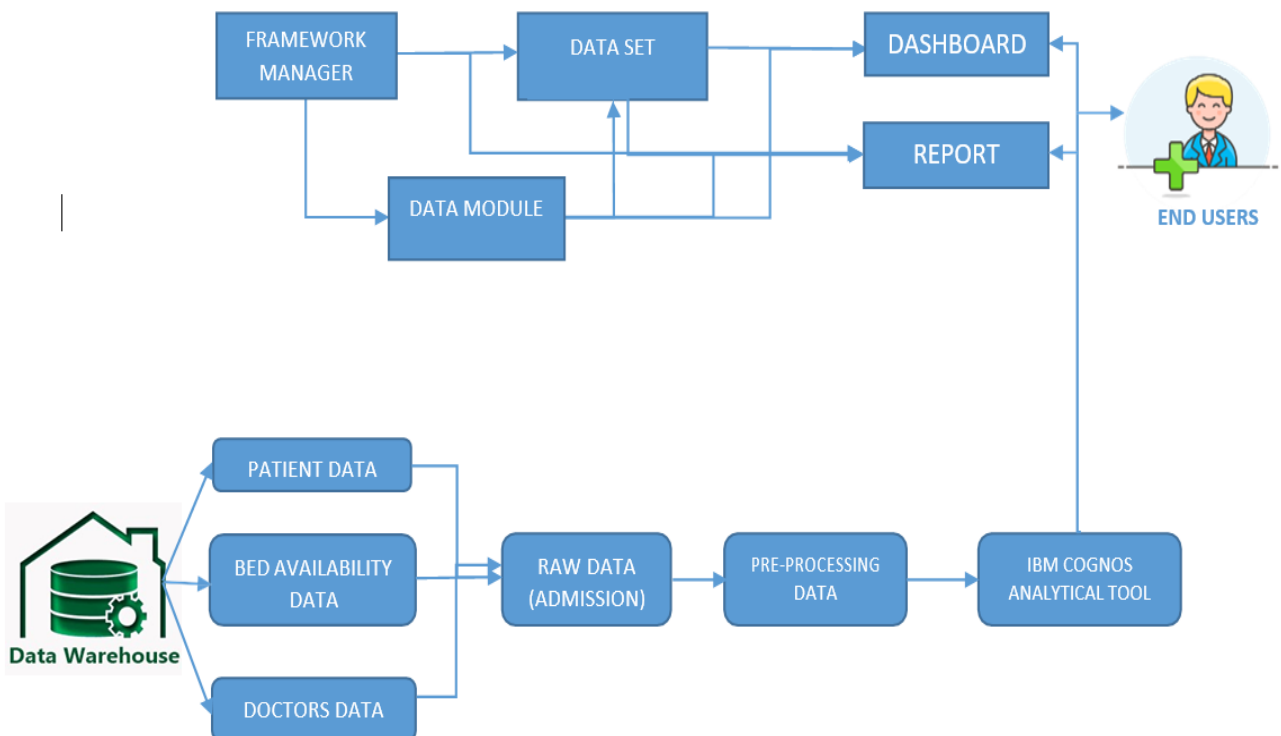
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	Patients can examine the available beds and get an overview of the length of stay for each patient via the dashboard. And through the data visualization they can understand clearly as it is represented through graphs and charts.
NFR-2	<b>Security</b>	Confidence Security should be provided like industry level security.
NFR-3	<b>Reliability</b>	The dashboard will be more effective for customers to utilize, because it will be dependable and operate well regularly. Additionally, it delivers a precise and efficient outcome. It is user friendly.
NFR-4	<b>Performance</b>	It swiftly analyses a patient's length of stay. It saves time because finding it will take longer for a human and The automated system improves the performance. Measuring the performance based on its efficiency and how quickly it response to the patients query.
NFR-5	<b>Availability</b>	The dashboard will be available all the time as demand of the patient will be at any time. It will be available for any kind of an emergency level. And provide necessary information according to the users demand.
NFR-6	<b>Scalability</b>	It is scalable as it can run even in the lower level machine also. Which will be more efficient for the users.

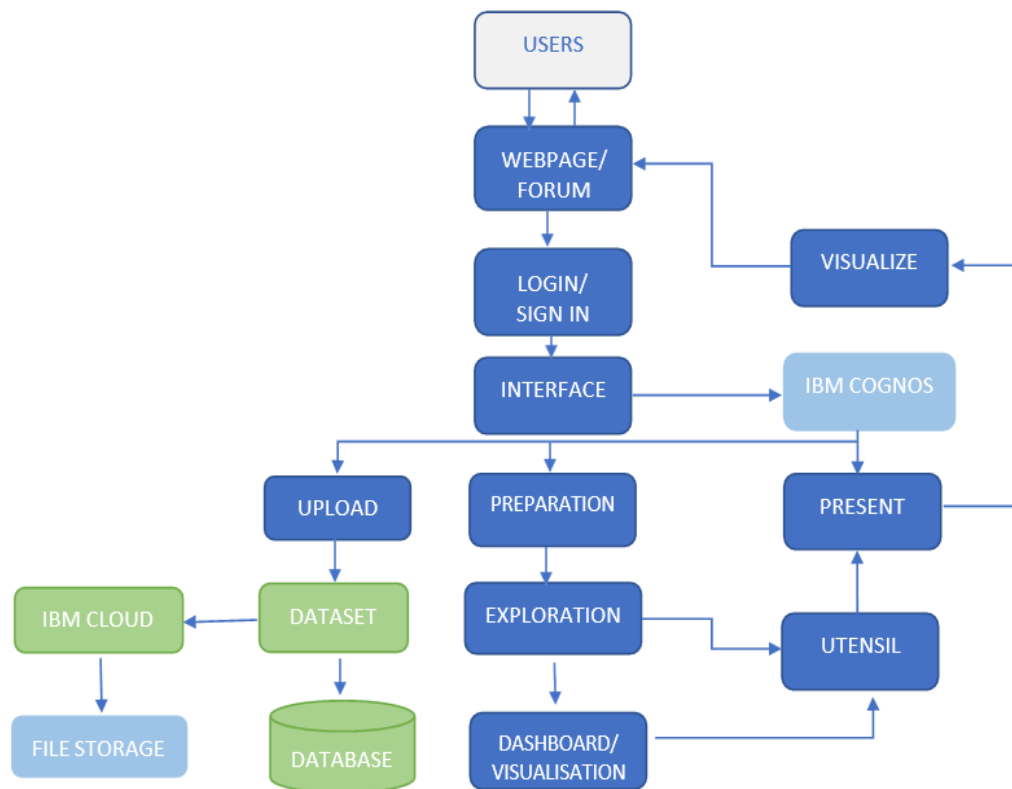
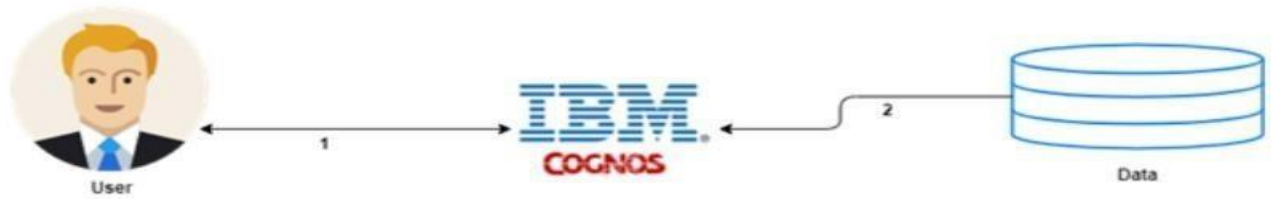
## 5.PROJECT DESIGN

### 5.1 Data Flow Diagrams

The classic visual representation of how information moves through a system is a data flow diagram (DFD). The appropriate amount of the system need can be graphically represented by a clean and unambiguous DFD. It demonstrates how information enters and exits the system, what modifies the data, and where information is kept.



## 5.2 Solution & Technical Architecture



**Table-1 : Components & Technologies**

S.No	Component	Description	Technology
1.	Dashboard	Logic for a process in the dashboard	IBM Cognos Analytics
2.	Cloud Database	Database Service on Cloud	IBM Cloudant
3.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
4.	Uploading and visualization data	Using exploration and visualization	IBM Cognos Analytics

**Table-2: Application Characteristics**

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The dashboard Framework is used to see the Length Of Stay of patient and all the hospital details and through the visualization of data the user can easily understand and it is user friendly	IBM Cognos
2.	Security Implementations	Industry level security will be provided	IAM Controls
3	Scalable Architecture	The workload may change and the user requirement may change and the architecture of the dashboard is designed in such a way that it can even handle the more workload	IBM Cognos
4	Availability	The user can view the upto date information and also the dashboard available all the time . It will beuseful for the user to get the information.	IBM Cognos
5	Performance	The dashboard quickly response to the user commands and it is user friendly	IBM Cognos



### 5.3 User Stories :

User Type	Functional Requirement(Epic)	User Story Number	User Story /Task	Acceptance criteria	Priority	Release
Managing patient details & Ward availability	USN-1	The official application enablesthe technical staffs in the hospital management to access patient data from anywhere in theworld and execute any operation on it.	Data is easilyaccessible wherever andwhenever.	Medium	Sprint-1	
Data Visualization	USN-2	Visualization techniques are used to determine the frequency of occurrence and recovery time ofa disease, and they can be usedto identify data trends.	Make easy andbetter understanding while visualizing data.	High	Sprint-2	
Dashboard	USN-3	The dashboard provides the information about the severity of the disease, recoveryperiod based on previous data and the current details of the data. This will help to find Length Of Stay.	Give quick access for datain need of patient details.	Medium	Sprint-3	
LOS predicted data	USN-4	Keeps track ofthe patient details about length of stay	Provides Better Management	High	Sprint-4	

## 6.PROJECT PLANNING & SCHEDULING

### 6.1. Sprint planning & Estimation

Sprint	Functiona l Requireme nt (Epic)	User Stor y Num ber	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Analyze	USN-1	As an admin, I will analyze the given dataset (Data pre-processing)	20	High	Suja S
Sprint-2	Visualization	USN-2	As a user, I can select the visualization type (Creating visualization)	20	Medium	Subalakshmi G
Sprint-3	Dashboard	USN-3	As a user, I can upload the datasets to the dashboard and view visualizations (Creating dashboard)	20	Medium	Subhiksha S
Sprint-4	Predict	USN-4	As an admin, I will predict the length of stay (Prediction)	20	High	Thaarani S

### 6.2 Sprint Delivery Schedule

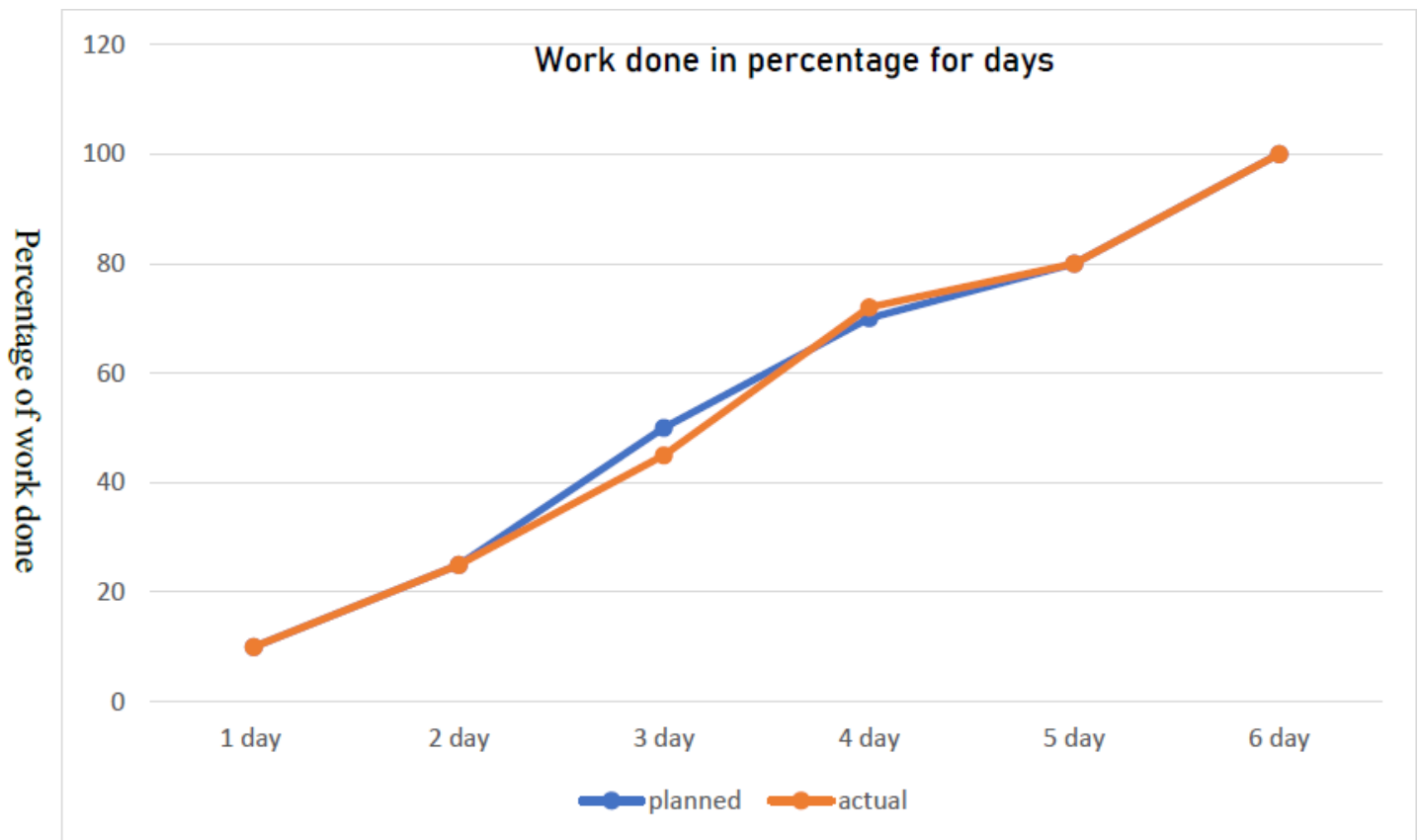
Sprint	Tota l Stor y Poin ts	Durati on	Sprint Start Date	Sprint End Date (Planne d)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

**Velocity:**

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

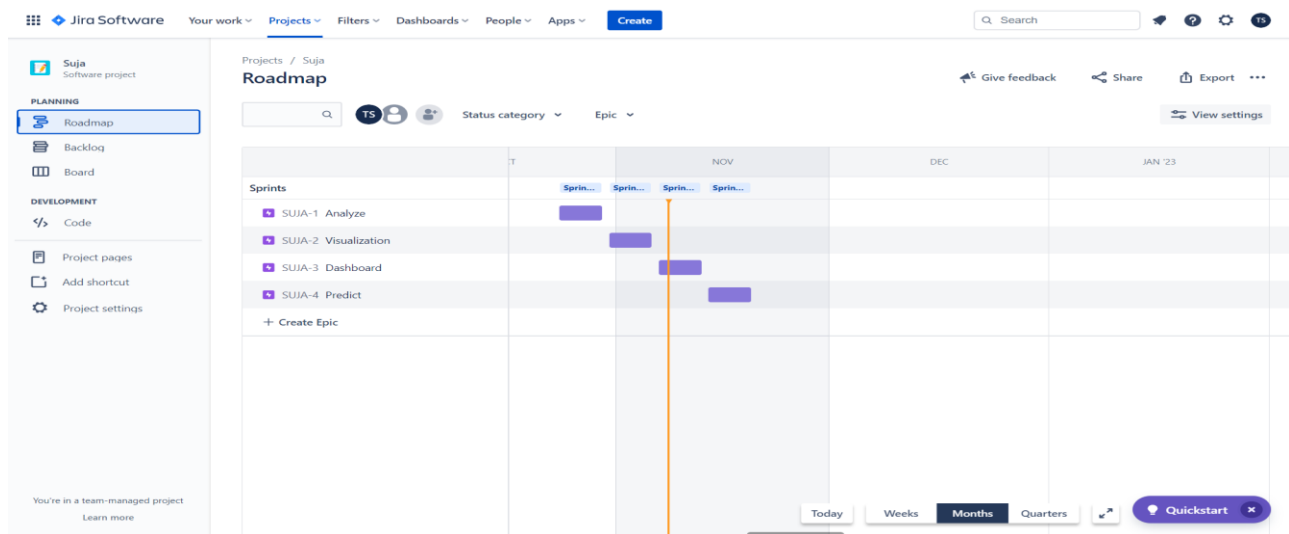
**Burndown Chart:**

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

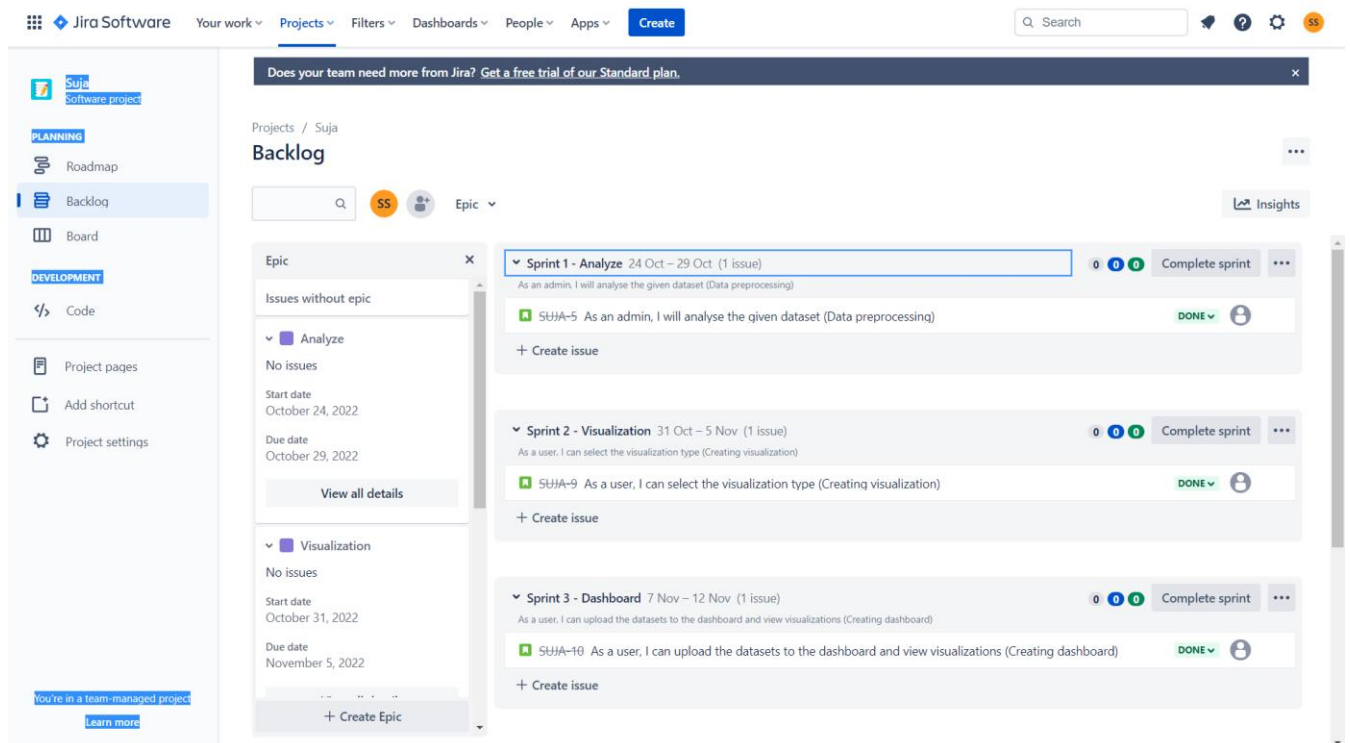


## 6.3 Reports from JIRA

### Roadmap



### Backlog



Jira Software Your work Projects Filters Dashboards People Apps Create

Search

Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / Suja

## Backlog

SS Epic

Insights

**Epic**

View all details

**Dashboard**

No issues

Start date: November 7, 2022

Due date: November 12, 2022

View all details

**Predict**

No issues

Start date: November 14, 2022

Due date: November 19, 2022

+ Create Epic

**Sprint 3 - Dashboard** 7 Nov – 12 Nov (1 issue)

As a user, I can upload the datasets to the dashboard and view visualizations (Creating dashboard)

SUJA-10 As a user, I can upload the datasets to the dashboard and view visualizations (Creating das... DONE

+ Create issue

**Sprint 4 - Predict** 14 Nov – 19 Nov (1 issue)

As an admin, I will predict the length of stay (Prediction)

SUJA-8 As an admin, I will predict the length of stay (Prediction) + Epic DONE

+ Create issue

**Backlog** (0 issues)

Your backlog is empty.

+ Create issue

## Dashboard

Jira Software Your work Projects Filters Dashboards People Apps Create

Search

Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / Suja

## All sprints

SS Epic Sprint

GROUP BY: None Insights

0 days remaining Complete sprint

**TO DO**

**IN PROGRESS**

**DONE 4 ISSUES**

As an admin, I will analyse the given dataset (Data preprocessing)

SUJA-5

As a user, I can select the visualization type (Creating visualization)

SUJA-9

As a user, I can upload the datasets to the dashboard and view visualizations (Creating dashboard)

SUJA-10

As an admin, I will predict the length of stay (Prediction)

SUJA-8

## 7. CODING AND SOLUTIONING

### 7.1 Feature 1

**Data exploration** is the first step of data analysis used to explore and visualize data to uncover insights from the start or identify areas or patterns to dig into more. Using interactive dashboards and point-and-click data exploration, users can better understand the bigger picture and get to insights faster.

Steps:

1. Variable Identification
2. Univariate Analysis
3. Bi-Variable Analysis
4. Detecting / Treating missing values
5. Detecting / Treating outliers
6. Feature Engineering

### CODE:

#### Data Exploration on Healthcare dataset

# Import packages

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv("/content/train_data.csv")

data.head()
```

Out[4]:

	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
0	1	8	c	3	Z	3	radiotherapy	R	F	2.0	31397.0	7.
1	2	2	c	5	Z	2	radiotherapy	S	F	2.0	31397.0	7.
2	3	10	e	1	X	2	anesthesia	S	E	2.0	31397.0	7.
3	4	26	b	2	Y	2	radiotherapy	R	D	2.0	31397.0	7.
4	5	26	b	2	Y	2	radiotherapy	S	D	2.0	31397.0	7.

```
data.tail()
```

```
Out[5]:
```

	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_P
	25077	25078	21	c	3	Z	2	gynecology	S	A	3.0	12058.0
	25078	25079	3	c	3	Z	5	gynecology	Q	A	3.0	12058.0
	25079	25080	12	a	9	Y	4	gynecology	R	B	4.0	12058.0
	25080	25081	15	c	5	Z	6	gynecology	P	F	4.0	12058.0
	25081	25082	2	c	5	Z	5	gynecology	Q	F	4.0	NaN

```
data.info()
```

```
RangeIndex: 25082 entries, 0 to 25081
Data columns (total 18 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   case_id                              25082 non-null  int64
1   Hospital_code                        25082 non-null  int64
2   Hospital_type_code                  25082 non-null  object
3   City_Code_Hospital                  25082 non-null  int64
4   Hospital_region_code                25082 non-null  object
5   Available Extra Rooms in Hospital    25082 non-null  int64
6   Department                          25082 non-null  object
7   Ward_Type                          25082 non-null  object
8   Ward_Facility_Code                  25082 non-null  object
9   Bed Grade                           25078 non-null  float64
10  patientid                           25081 non-null  float64
11  City_Code_Patient                   24831 non-null  float64
12  Type of Admission                   25081 non-null  object
13  Severity of Illness                 25081 non-null  object
14  Visitors with Patient               25081 non-null  float64
15  Age                                 25081 non-null  object
16  Admission_Deposit                   25081 non-null  float64
17  Stay                                25081 non-null  object
dtypes: float64(5), int64(4), object(9)
memory usage: 3.4+ MB
```

```
data.nunique()
```

```
Out[7]:
```

case_id	25082
Hospital_code	32
Hospital_type_code	7
City_Code_Hospital	11
Hospital_region_code	3
Available Extra Rooms in Hospital	12
Department	5
Ward_Type	5
Ward_Facility_Code	6
Bed Grade	4
patientid	4922
City_Code_Patient	32
Type of Admission	3
Severity of Illness	3
Visitors with Patient	21
Age	10
Admission_Deposit	4825
Stay	11
dtype: int64	

```
data.isnull().sum()
```

```
Out[8]:
```

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	4
patientid	1
City_Code_Patient	251
Type of Admission	1
Severity of Illness	1
Visitors with Patient	1
Age	1
Admission_Deposit	1
Stay	1
dtype: int64	

```
(data.isnull()).sum()/(len(data))* 100
```

```
Out[9]: case_id          0.000000
        Hospital_code    0.000000
        Hospital_type_code 0.000000
        City_Code_Hospital 0.000000
        Hospital_region_code 0.000000
        Available Extra Rooms in Hospital 0.000000
        Department      0.000000
        Ward_Type        0.000000
        Ward_Facility_Code 0.000000
        Bed_Grade         0.015948
        patientId         0.003987
        City_Code_Patient 1.000718
        Type of Admission 0.003987
        Severity of Illness 0.003987
        Visitors with Patient 0.003987
        Age               0.003987
        Admission_Deposit 0.003987
        Stay              0.003987
        dtype: float64
```

```
data.drop(columns=['City_Code_Patient'], inplace = True)
data.info()
```

```
RangeIndex: 25082 entries, 0 to 25081
Data columns (total 17 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   case_id                              25082 non-null  int64
1   Hospital_code                        25082 non-null  int64
2   Hospital_type_code                  25082 non-null  object
3   City_Code_Hospital                  25082 non-null  int64
4   Hospital_region_code                25082 non-null  object
5   Available Extra Rooms in Hospital    25082 non-null  int64
6   Department                          25082 non-null  object
7   Ward_Type                           25082 non-null  object
8   Ward_Facility_Code                  25082 non-null  object
9   Bed_Grade                           25078 non-null  float64
10  patientId                            25081 non-null  float64
11  Type of Admission                    25081 non-null  object
12  Severity of Illness                  25081 non-null  object
13  Visitors with Patient                25081 non-null  float64
14  Age                                  25081 non-null  object
15  Admission_Deposit                    25081 non-null  float64
16  Stay                                25081 non-null  object
dtypes: float64(4), int64(4), object(9)
memory usage: 3.3+ MB
```

```
data.describe()
```

```
In [12]: data.describe()
```

```
Out[12]:
```

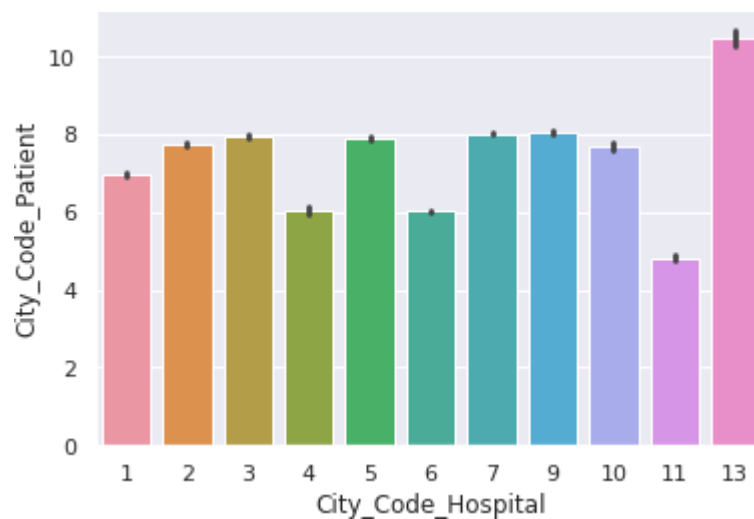
	case_id	Hospital_code	City_Code_Hospital	Available Extra Rooms in Hospital	Bed_Grade	patientId	Visitors with Patient	Admission_Deposit
count	25082.000000	25082.000000	25082.000000	25082.000000	25078.000000	25081.000000	25081.000000	25081.000000
mean	12541.500000	18.810980	4.769715	3.104258	2.663649	65344.442765	3.248395	4981.075515
std	7240.694062	8.632074	3.162466	1.141522	0.857561	37908.702777	1.762000	1053.491789
min	1.000000	1.000000	1.000000	0.000000	1.000000	70.000000	0.000000	1820.000000
25%	6271.250000	11.000000	2.000000	2.000000	2.000000	32622.000000	2.000000	4307.000000
50%	12541.500000	21.000000	5.000000	3.000000	3.000000	64413.000000	3.000000	4855.000000
75%	18811.750000	26.000000	7.000000	4.000000	3.000000	98066.000000	4.000000	5496.000000
max	25082.000000	32.000000	13.000000	12.000000	4.000000	131595.000000	24.000000	10999.000000



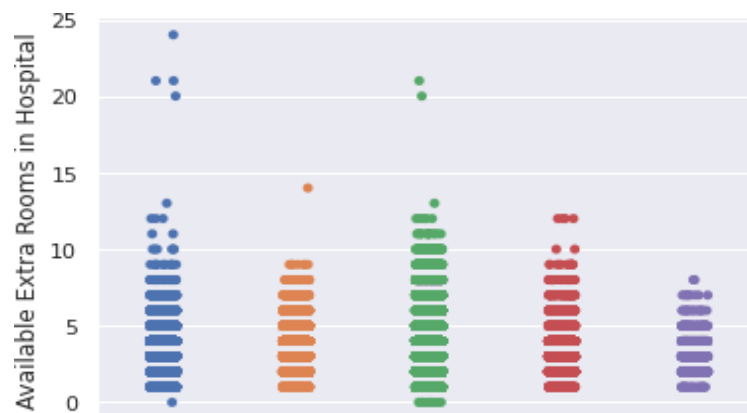
**Data visualization** is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. Additionally, it provides an excellent way for employees or business owners to present data to non-technical audiences without confusion.

### CODE :

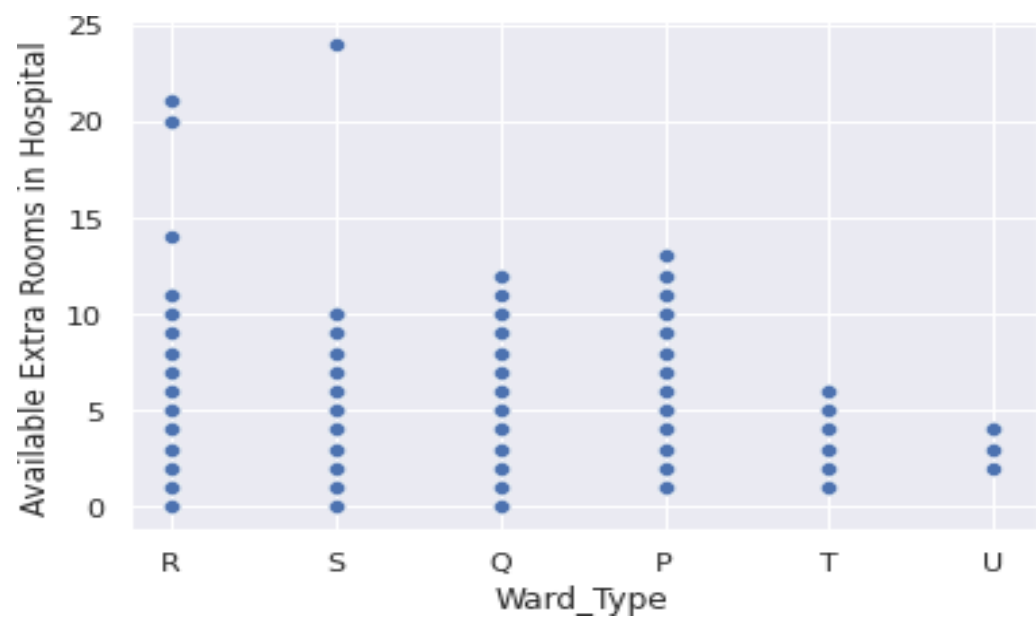
```
sns.barplot(data['City_Code_Hospital'], data['City_Code_Patient'])
```



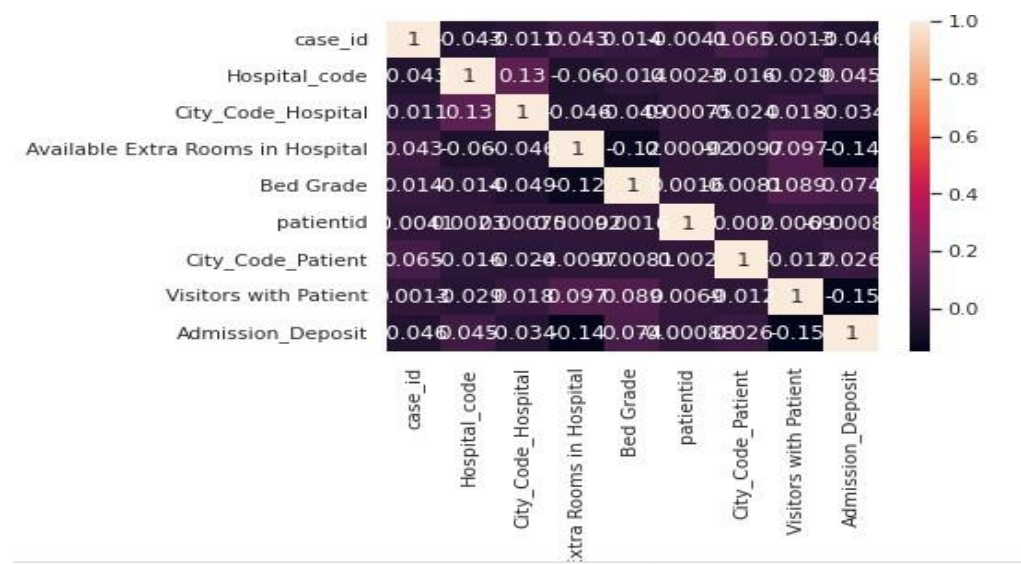
```
sns.stripplot(data['Department'], data['Available Extra Rooms in Hospital'])
```



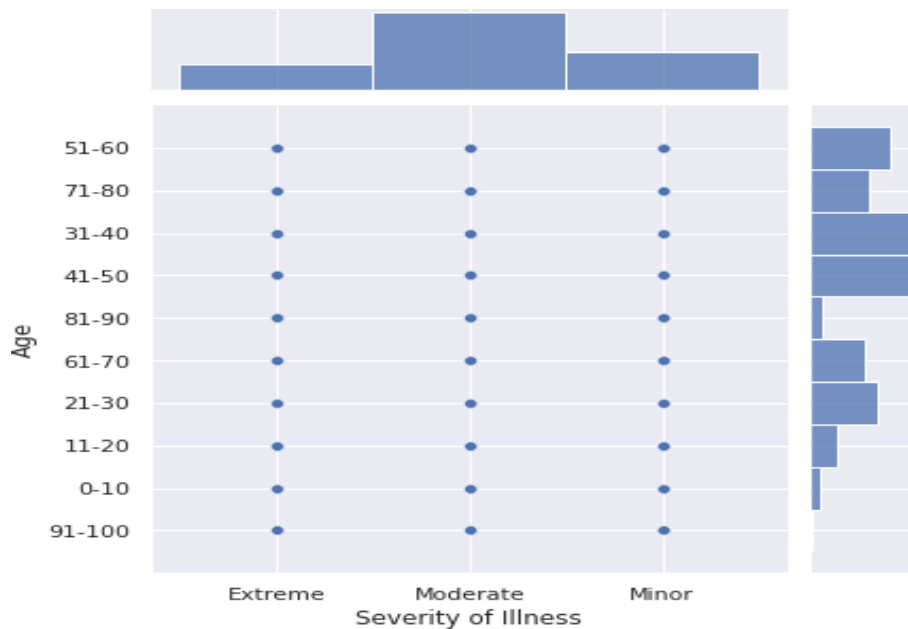
```
sns.scatterplot(data['Ward_Type'], data['Available Extra Rooms in Hospital'])
```



```
sns.heatmap(data.corr(), annot=True)
```



```
sns.jointplot(data['Severity of Illness'], data['Age'])
```



```
sns.barplot(data['Department'], data['Bed Grade'])
```

```
sns.pairplot(data, hue="Admission_Deposit")
```

## 7.2 Feature 2

### Predictive Model

Predictive analytics is a branch of advanced analytics that makes predictions about future outcomes using historical data combined with statistical modeling, data mining techniques and machine learning. Companies employ predictive analytics to find patterns in this data to identify risks and opportunities.

#### 1) KNN

Algorithm of K Nearest Neighbors. The k-nearest neighbor's algorithm, sometimes referred to as KNN or k-NN, is a supervised learning classifier that employs proximity to produce classifications.

**CODE:**

```
# Accuracy while using KNN
knn = KNeighborsClassifier(n_neighbors = 3)
knn.fit(X_train, Y_train)
Y_pred = knn.predict(X_test)
knn_accuracy = round(knn.score(X_train, Y_train) * 100, 2)
print("Accuracy of KNN ")
knn_accuracy
```

**This model gives an accuracy score of 54.92% after validating.**

**2) DECISION TREE**

The non-parametric supervised learning approach used for classification and regression applications is the decision tree. It is organised hierarchically and has a root node, branches, internal nodes, and leaf nodes.

**CODE:**

```
# Accuracy while using Decision Tree
decision_tree = DecisionTreeClassifier()
decision_tree.fit(X_train, Y_train)
Y_pred = decision_tree.predict(X_test)
decision_tree_accuracy = round(decision_tree.score(X_train, Y_train) * 100, 2)
print("Accuracy of Decision Tree ")
decision_tree_accuracy
```

**This model gives an accuracy score of 99.64% after validating. While training this model with decision tree algorithm it gives more accuracy than KNN.**

**3) RANDOM FOREST**

Popular machine learning algorithm Random Forest is a part of the supervised learning methodology. It can be applied to ML issues involving both classification and regression. It is built on the idea of ensemble learning, which is a method of integrating various classifiers to address difficult issues and enhance model performance.

Random Forest, as the name implies, is a classifier that uses a number of decision trees on different subsets of the provided dataset and averages them to increase the dataset's predictive accuracy. Instead than depending on a single decision tree, the random forest uses forecasts from each tree and predicts

the result based on the votes of the majority of predictions. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

### CODE:

```
# Accuracy while using 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)
print("Accuracy of Random Forest ")
acc_random_forest
```

**This model gives an accuracy score of 99.64% after validating. Hence satisfied with the accuracy.**

### 7.3 Database Schema:

1. Case\_id
2. Hospital\_code
3. Hospital\_type\_code
4. City\_code\_Hospital
5. Available Extra Rooms in Hospital
6. Department
7. Ward\_type
8. Ward\_Facility\_code
9. Bed Grade
- 10.10.Patient\_id
- 11.City\_code\_Patient
- 12.Type of Admission
- 13.Severity of Illness
- 14.Visitors with Patient
- 15.Age
- 16.Admission\_Deposit
- 17.Stay

## 8. TESTING

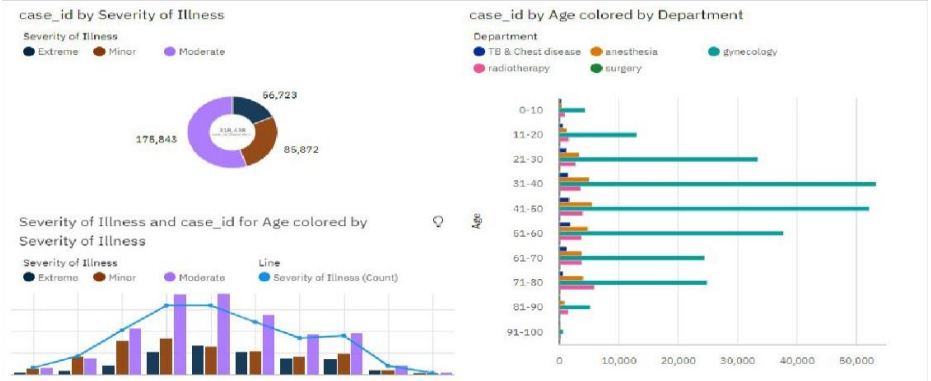
## 8.1 TEST CASES

## Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	<pre> In [44]: LOS_predicted['Stay'] = LOS_predicted['Stay'].replace(stay_dict.values(), stay_dict.keys())  In [45]: LOS_predicted.to_csv('LOS.csv', index = False)  In [46]: LOS = pd.read_csv('/content/LOS.csv') LOS.info()  RangeIndex: 134865 entries, 0 to 134864 Data columns (total 2 columns):  #   Column  Non-Null Count  Dtype ---  --  0   case_id 134865 non-null    int64  1   Stay    134865 non-null    object dtypes: int64(1), object(1) memory usage: 2.1+ MB  In [47]: LOS.head(10)  Out[47]:    case_id  Stay 0  310439  21-30 1  310440  31-40 2  310441  21-30 3  310442  51-60 4  310443  21-30 5  310444  21-30 6  310445  21-30 7  310453  31-40 8  310454  21-30 9  310455  41-50  In [48]: plt.figure(figsize=(10,5)) LOS.Stay.value_counts().plot(kind="bar", color = ['green'])  Out[48]: </pre>

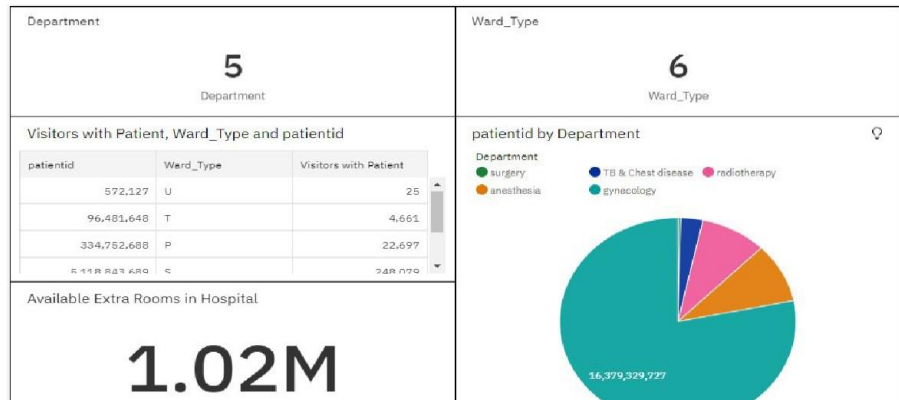
2.	Accuracy	<p>Training Accuracy</p> <p>Validation Accuracy</p> <p><b>99.64%</b></p>	<pre> In [39]: # Accuracy while using Decision Tree decision_tree = DecisionTreeClassifier() decision_tree.fit(X_train, Y_train) Y_pred = decision_tree.predict(X_test) decision_tree_accuracy = round(decision_tree.score(X_train, Y_train) * 100, 2) print("Accuracy of Decision Tree ") decision_tree_accuracy  Accuracy of Decision Tree Out[39]: 99.64  In [40]: # Accuracy while using 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) print("Accuracy of Random Forest ") acc_random_forest  Accuracy of Random Forest Out[40]: 99.64 </pre>
----	----------	--------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Model Performance Testing:

S.N o.	Parameter	Screenshot / Values
1.	Dashboard design	<p><b>No of Visualizations / Graphs : 17</b></p> 

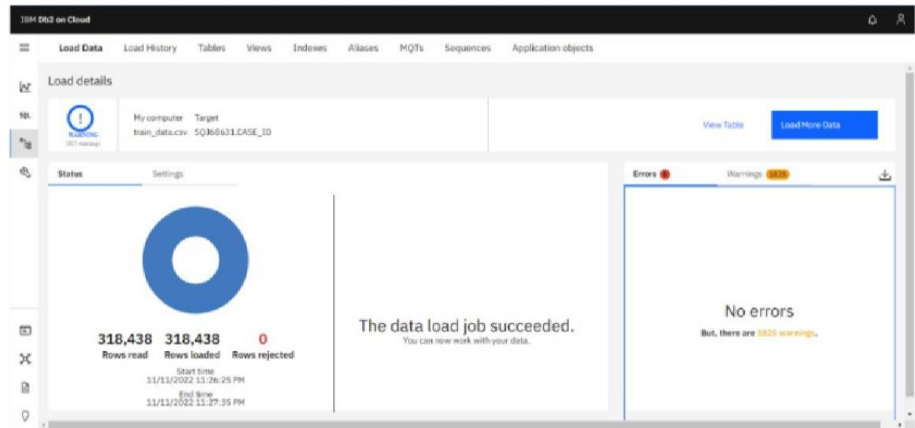
## 2. Data Responsiveness

The visualizations are responsive enough to view the data and fit the screen



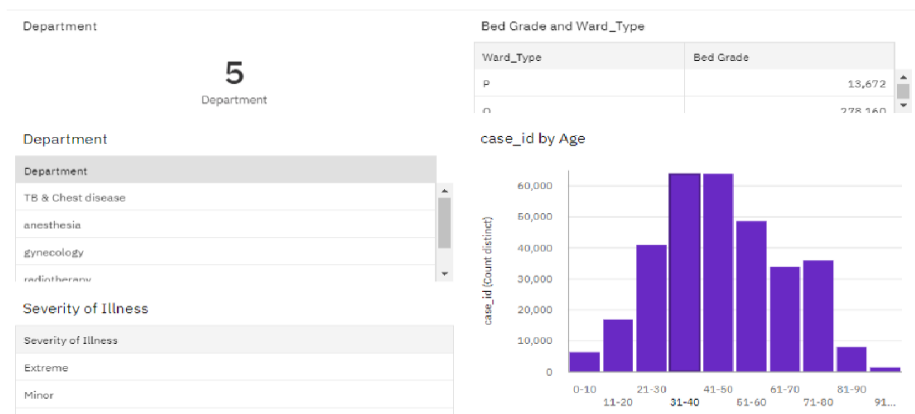
## 3. Amount Data to Rendered (DB2 Metrics)

**No.of Rows read: 318438 No.of Rows loaded: 318438**



## 4. Utilization of Data Filters

The filters are used to see only the relevant data about the usecase





5.	Effective User Story	No of Scene Added – 5
6.	Descriptive Reports	No of Visualizations / Graphs – 6

## 8.2 User Acceptance Testing

### Purpose of the Document:

The purpose of this document is to briefly explain the test coverage and open issues of the Analytics for Hospital's Healthcare Data project at the time of the release to User Acceptance Testing (UAT).

### Defect Analysis:

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	8	4	0	2	14
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	13	4	3	16	36
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	1	4	2	1	8
Totals	23	18	12	22	76

**Test case analysis:**

This report shows the number of test cases that have passed, failed and untested.

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	9	0	0	9
Client Application	43	0	0	43
Security	1	0	0	1
Outsource Shipping	1	0	0	1
Exception Reporting	9	0	0	9
Final Report Output	10	0	0	10
Version Control	1	0	0	1

## 9) RESULTS

### 9.1 Performance Metrics

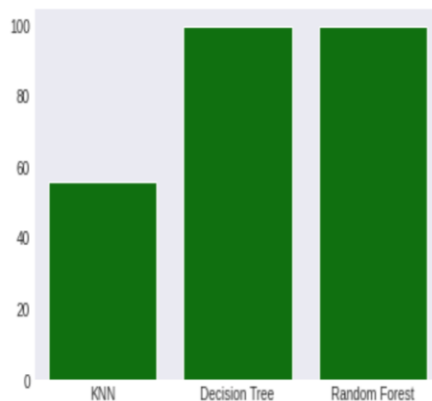
#### Accuracy :

The easiest performance metric to understand is accuracy, which is just the proportion of properly predicted observations to all observations. One would believe that if our model is accurate, it is the best. Yes, accuracy is an excellent indicator, but only when the values of the false positive and false negative rates are nearly equal in the datasets.

$$\text{Accuracy} = \frac{\text{number of correct predictions}}{\text{total number of predictions}}$$

```
✓ 0s sns.barplot(x= ['KNN','Decision Tree','Random Forest'],y= [knn_accuracy, decision_tree_accuracy, acc_random_forest],color = 'green')
```

```
[ ] <matplotlib.axes._subplots.AxesSubplot at 0x7fd8df9ee710>
```



### Length of Stay prediction Dataset:

1	case_id	Stay	1	case_id	Stay	1	case_id	Stay
2	318439	71-80	2	318439	21-30	2	318439	31-40
3	318440	21-30	3	318440	31-40	3	318440	31-40
4	318441	21-30	4	318441	21-30	4	318441	21-30
5	318442	21-30	5	318442	21-30	5	318442	21-30
6	318443	71-80	6	318443	21-30	6	318443	21-30
7	318444	71-80	7	318444	21-30	7	318444	21-30
8	318445	21-30	8	318445	31-40	8	318445	31-40
9	318453	21-30	9	318453	31-40	9	318453	81-90
10	318454	21-30	10	318454	0-10	10	318454	31-40
11	318455	21-30	11	318455	31-40	11	318455	31-40
12	318456	21-30	12	318456	21-30	12	318456	21-30
						13	318457	21-30

LOS using KNN

LOS using Decision Tree

LOS using Random Forest

We are using the algorithms - KNN, Decision Tree and Random Forest classifications for Length Of Stay prediction. While analyzing Decision Tree and Random Forest gives similar results and with high accuracy **99.64%**.

## **10.ADVANTAGES AND DISADVANTAGES**

### **Advantages:**

- Hospitals can better manage their patients and allocate resources by forecasting a patient's length of stay at the time of admission.
- It aids medical facilities in resource management and the creation of innovative treatment strategies.
- Reducing the length of stay and making efficient use of hospital resources can lower overall medical spending nationally.
- It is simple to outline the staffing process.
- The presence of visualizations helps users comprehend the current situation and take the appropriate action
- It is easier to forecast how long a patient will stay in the hospital.
- Better Health Results
- It is simple to define the staffing process.

### **Disadvantages:**

- Shortage in the staffing
- Cybersecurity risk

## **11.CONCLUSION**

Effective LOS prediction, particularly at the admission stage, provides crucial information that helps the hospital administration and medical staff make crucial decisions. While preventing a significant loss of resources, hospital administration can distribute the appropriate and necessary resources and the best medical team for treating the patient. The medical staff can create a suitable medical plan using this forecast as well. Finally, this projection can be used by insurance companies and patient family to plan and manage their budgets. It is suggested to use a comprehensive general framework to forecast patient LOS in the ICU. This framework trained the real gathered data set using a variety of ML approaches to forecast the patients' length of stay.

## **12. FUTURE SCOPE**

The analysis strategy will be equipped and used with an improved decision-making process in the future, allowing for the proper activity planning and processes to be selected. The stakeholders will have no trouble understanding the visualization techniques. More specialized techniques will be added. Expecting better hospital management.

## 13.APPENDIX

### SOURCE CODE:

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

### DATA PREPARATION

```
import os
for dirname, _, filenames in os.walk('/content/Healthcare_Data'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

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

dictionary
```

### DATA EXPLORATION

```
train.info()

train.tail(5)

plt.figure(figsize=(10,7))
train.Stay.value_counts().plot(kind="bar", color = ['blue'])

train.isnull().sum()
```

## DATA PREPROCESSING

```

train.dropna(inplace=True)
test.dropna(inplace=True)

# Combining test and train dataset for processing
new_set = [train, test]

from sklearn.preprocessing import LabelEncoder
for data in new_set:
    label = LabelEncoder()
    data['Department'] = label.fit_transform(data['Department'])

for dataset in new_set:
    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 Admissi
on'])
    dataset['Severity of Illness'] = label.fit_transform(dataset['Severity of I
llness'])

new_set[0]

new_set[1]

new_set[0].Age.hist()

new_set[0].Age.hist()

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}

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

new_set[0].Age.hist()

new_set[0].Stay.unique()

```



```

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': 9, 'More than 100 Days': 10}

new_set[0]['Stay'] = new_set[0]['Stay'].replace(stay_dict.keys(), stay_dict.values())

new_set[0].Stay.hist()

for data in new_set:
    print(data.shape)

new_set[0].info()

new_set[1].info()

columns_list = ['Type of Admission', 'Available Extra Rooms in Hospital', 'Visitors with Patient', 'Admission_Deposit']
len(columns_list)

from sklearn.preprocessing import StandardScaler
s1= StandardScaler()

for dataset in new_set:
    dataset[columns_list]= s1.fit_transform(dataset[columns_list].values)

plt.figure(figsize=(17,17))
sns.heatmap(new_set[0].corr(), annot=True, cmap='BuPu')

```

## MODELLING THE DATA

```

from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier

train = new_set[0]
test = new_set[1]

sample

X_train = train.drop(['case_id', 'Stay', 'Hospital_region_code'], axis=1)

```

```

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

X_train = X_train.astype(int)
Y_train = Y_train.astype(int)
X_test = X_test.astype(int)

sample.shape

X_test.columns

Y_train

# Accuracy while using KNN
knn = KNeighborsClassifier(n_neighbors = 3)
knn.fit(X_train, Y_train)
Y_pred = knn.predict(X_test)
knn_accuracy = round(knn.score(X_train, Y_train) * 100, 2)
print("Accuracy of KNN ")
knn_accuracy

LOS_predicted = pd.DataFrame({
    "case_id": test["case_id"],
    "Stay": Y_pred
})
LOS_predicted['Stay'] = LOS_predicted['Stay'].replace(stay_dict.values(), stay_
dict.keys())
LOS_predicted.to_csv('KNN.csv', index = False)

# Accuracy while using Decision Tree
decision_tree = DecisionTreeClassifier()
decision_tree.fit(X_train, Y_train)
Y_pred = decision_tree.predict(X_test)
decision_tree_accuracy = round(decision_tree.score(X_train, Y_train) * 100, 2)
print("Accuracy of Decision Tree ")
decision_tree_accuracy

LOS_predicted = pd.DataFrame({
    "case_id": test["case_id"],
    "Stay": Y_pred
})
LOS_predicted['Stay'] = LOS_predicted['Stay'].replace(stay_dict.values(), stay_
dict.keys())
LOS_predicted.to_csv('dec.csv', index = False)

```

```

# Accuracy while using 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)
print("Accuracy of Random Forest ")
acc_random_forest

sns.barplot(x= ['KNN', 'Decision Tree', 'Random Forest'], y= [knn_accuracy, decision_tree_accuracy, acc_random_forest], color = 'green')

```

## RESULT - LOS Predicted Data

```

sample

LOS_predicted = pd.DataFrame({
    "case_id": test["case_id"],
    "Stay": Y_pred
})

LOS_predicted['Stay'] = LOS_predicted['Stay'].replace(stay_dict.values(), stay_dict.keys())

LOS_predicted.to_csv('LOS.csv', index = False)

LOS = pd.read_csv('/content/LOS.csv')
LOS.info()

LOS.head(10)

plt.figure(figsize=(10, 5))
LOS.Stay.value_counts().plot(kind="bar", color = ['green'])

```

**CODE LINK :**

<https://colab.research.google.com/drive/1OQSB0wjYaKYCxRIHHMpwfLslmXsAf?usp=sharing>

**GITHUB LINK:**

<https://github.com/IBM-EPBL/IBM-Project-17076-1659627567>