Team Id	**PNT2022TMID30840**
Project Name	**Analytics for hospitals health care data**
Team Members	**620119104118 - Vishnu Prasath R** **620119104102 - Thamizharasu P** **620119104074 - Prem Kumar M** **620119104119 - Viswanath M**

Project Report

Analytics for Hospitals Health-care data

CHAPTER-1
INTRODUCTION

1.1 Project Overview

In this project, we will work closely with hospital health care data, and to do that, we will examine the health care dataset. From that dataset, we will derive various insights that help us understand the weighting of each feature and how they relate to one another, but this time, our only goal is to determine the likelihood that a person will be seen and treated or not.

A significant amount of data generated by the health care sector can be used to make predictions and judgments with the help of machine learning. By analysing patient data that uses a machine-learning algorithm to categorise whether a patient has an illness or not, this study hopes to anticipate future health care. Machine learning methods can be extremely helpful in this situation. There is a common set of basic risk variables that determine whether someone will ultimately be at risk for severe disease or not, despite the fact that disease might manifest itself in various ways. We may state that this technique can be very well adapted to perform the analysis by gathering the data from many sources, classifying them under appropriate headings, and then analysing to extract the needed data.

1.2 Purpose

Healthcare data is used to save lives and enhance quality of life, therefore businesses and governments are working hard to develop fresh approaches. Large amounts of data may be processed, stored, and analysed using artificial intelligence.

By gathering accurate patient data, you can identify comparable potential clients and adjust your marketing strategies and medical practises accordingly. The secret to success for healthcare facilities is better consumer relations—in this case, better patient interactions. As a result, the use of electronic technologies in the

healthcare setting guarantees secure and effective data administration. For effective health care delivery, it is crucial to set up suitable medical data management systems. Medical data processing, health care data, and electronic medical data are some related terms.

Clinical and patient data are used in data analytics in healthcare to better patient outcomes, enhance care, and improve company management. To maximise your potential non-labor cost savings, you should choose a healthcare consulting company that specialises in health data analytics.

CHAPTER-2
LITERATURE SURVEY

2.1 Existing problem

The globe is currently surrounded with data, much like oxygen. In the digitised age, the amount of data that we gather and consume is thriving aggressively. Social media and the increased usage of new technologies generate enormous amounts of data that, when correctly examined, can yield wonderful information. Due to its vastness, this massive dataset, also known as big data, cannot be stored in conventional databases. In order to make better decisions and get better results, organisations must manage and analyse massive data. As a result, big data analytics is currently getting a lot of attention. Big data analytics in healthcare has the potential to increase patient care and provide clinical decision support. In this essay, we examine the history and many approaches to big data analytics in healthcare.

2.2 References

[1] Alexandros Labrinidis and H. V. Jagadish, “Challenges and opportunities with big data,” Proc. VLDB Endow. 5, pp. 2032-2033, August 2012.

[2] Aneeshkumar, A.S. and C.J. Venkateswaran, “Estimating the surveillance of liver disorder using classification algorithms”. Int. J. Comput. Applic., 57: pp. 39-42, 2012.

[3] Amir Gandomi, Murtaza Haider, “Beyond the hype: Big data concepts, methods, and analytics,” International Journal of Information Management 35, pp. 137-144, 2015.

[4] Chaitrali, S., D. Sulabha and S. Apte, “Improved study of heart disease prediction system using data mining classification techniques," Int. J. Comput. Applic. 47: 44-48, 2012.

[5] Doug Beaver, Sanjeev Kumar, Harry C. Li, Jason Sobel, Peter Vajgel, Facebook

Inc, “Finding a Needle in Haystack: Facebook’s Photo Storage” 2010.

[6] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber, “Bigtable: A Distributed Storage System for Structured Data,” ACM Trans. Comput. Syst. 26, 2, Article 4, June 2008.

[7] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels, “Dynamo: amazon's highly available key-value store,” In Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles (SOSP '07). ACM, New York, NY, USA, 205-220.

[8] Hsi-Jen et al. “A retrospective analysis of prognostic indicators in dental implant therapy using the C5.0 decision tree algorithm”, Journal of Dental Sciences, Volume 8, Issue 3 , 248-255, 2013.

[9] I.A.T. Hashem, et al, “The rise of “big data” on cloud computing: Review and open research issues,” Information Systems, 2014.

[10]Jakrarin Therdphapiyanak, Krerk Piromsopa, “An analysis of suitable parameters for efficiently applying K-means clustering to large TCPdump data set using Hadoop framework,” In Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference, pp. 1-6, May 2013.

[11]Jason Brownlee, “Machine Learning Foundations, Master the definitions and concepts”, Machine Learning Mastery, 2011.

[12]Jawei Han and Micheline Kamber, “Data Mining: Concepts and Techniques,” Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.
[13]L. Hall, N. Chawla, and K. Bowyer, "Decision tree learning on very large data sets," in International Conference on Systems, Man and Cybernetics, pp. 2579-2584, IEEE Oct 1998.

[14]Mark A. Beyer, Douglas Laney, “The importance of ‘Big Data’: A Definition,” Gartner, retrieved on 21 June 2012.

[15]Nilima Patil and Rekha Lathi, “Comparison of C5.0 and CART Classification algorithms use pruning technique”, 2012.

[16]Patil D.V, Prof. Dr. R. S. Bichkar, "A Hybrid Evolutionary Approach To Construct Optimal Decision Trees with Large Data Sets", IEEE, 2006.

[17]Rajesh, K. and S. Anand, “Analysis of SEER dataset for breast cancer diagnosis using C4.5 classification algorithm,” Int. J. Adv. Res. Comput. Commun. Eng., 1: 72-77, 2012.

[18]Ramalingam, V.V., S.G. Kumar and V. Sugumaran, “Analysis of EEG signals using data mining approach,” Int. J. Comput. Eng. Technol., 3: 206- 212, 2012.

2.3 Problem Statement Definition

The analysis of health care presents the biggest problem. There are equipment that can analyse health care data, but they are either expensive or ineffective at calculating the likelihood of a disease in a human. The mortality rate and total consequences can be reduced by early detection of health care data. Since it takes more intelligence, time, and knowledge, it is not always possible to accurately monitor patients every day, and a doctor cannot consult with a patient for a whole 24 hours. In the modern world, we have a lot of data, so we can use a variety of machine learning algorithms to analyse the data and look for hidden patterns. In medical data, the hidden patterns might be used for health diagnosis.

Hospitals are in dire need of assistance due to the staggering hospitalisation rates, escalating cybersecurity threats, and an increase in mental illnesses brought on by stringent lockdown procedures. Healthcare big data appears to be a workable solution.

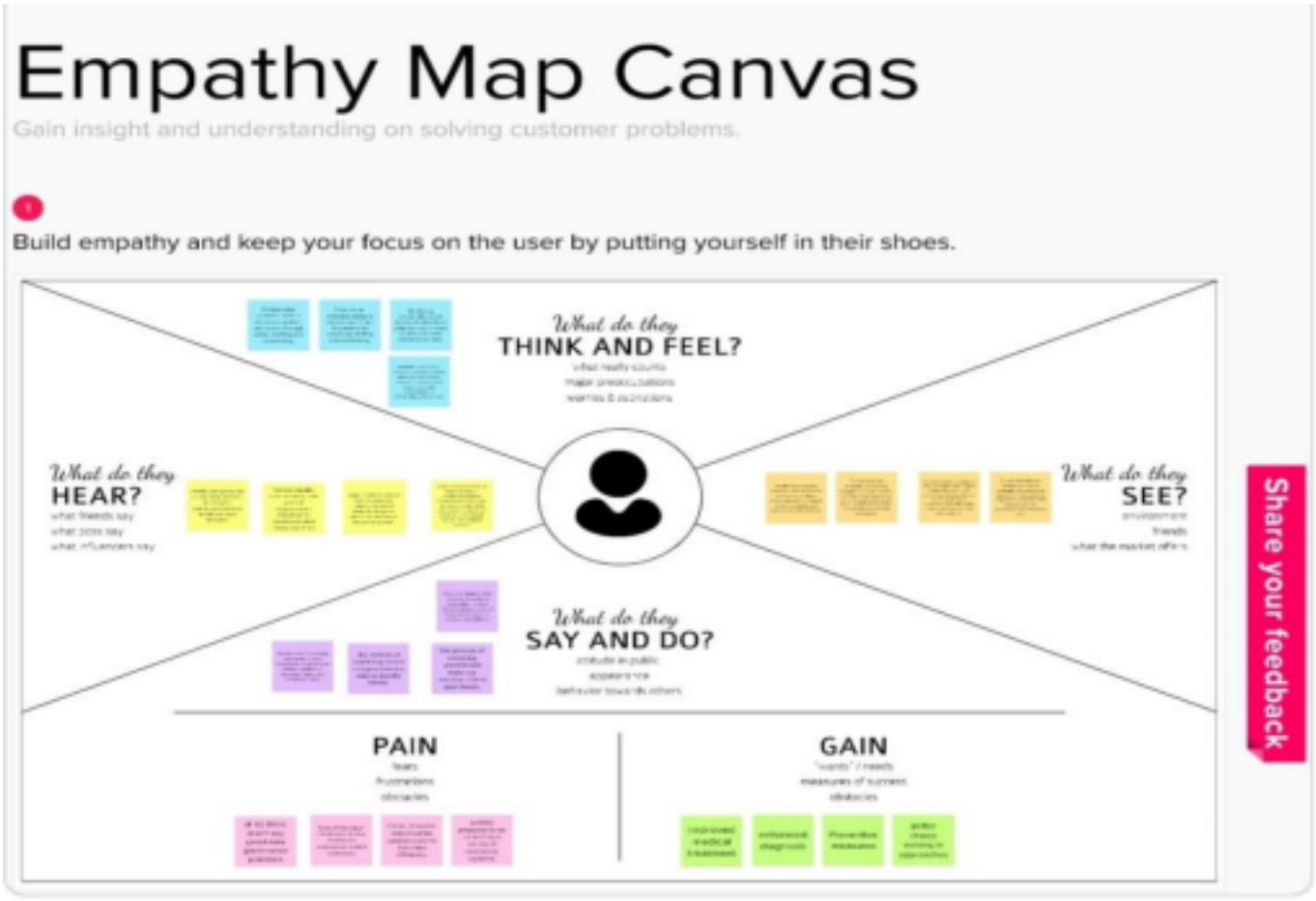
CHAPTER-3
IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

The basic empathy map, which aids in identifying and describing the user's wants and pain locations, is expanded upon in an empathy map canvas. Additionally, this data is useful for enhancing user experience. Teams use user insights to map out what matters to, influences, and how their target audience presents themselves.

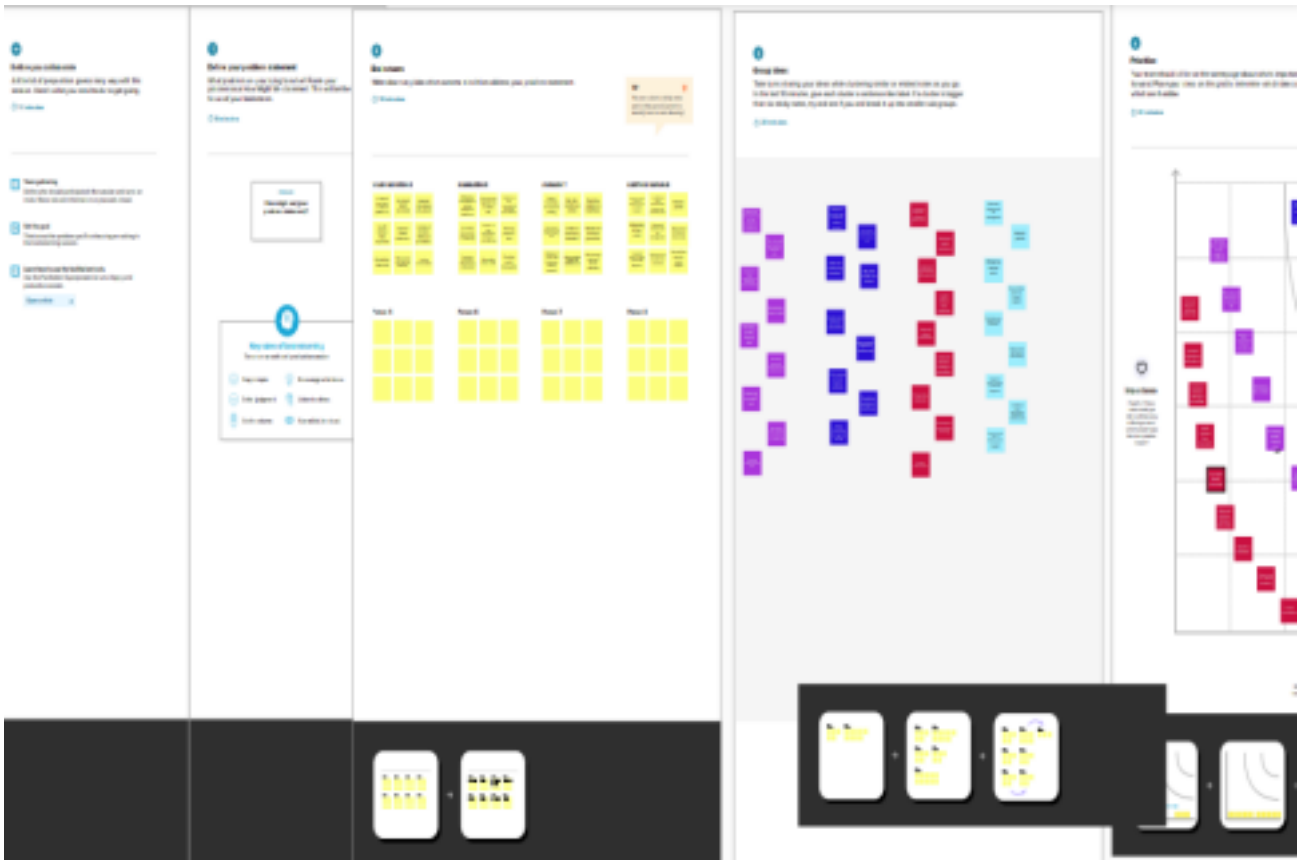
Using this data, personas are then developed to assist teams in visualising and empathising with users as people rather than just as a general marketing demographic or account number. By assisting teams in comprehending the perspectives and attitude of their customers, brands are better able to present users with an engaging experience. By using a template, you may speed up the process of creating empathy map canvases and ensure that they are all of a similar calibre.

Empathy Map Canvas Visualizing and Predicting Heart Diseases with an Interactive Dashboard:



3.2 Ideation & Brainstorming

During a brainstorming session, everyone on a team is encouraged to engage in the process of original thought that results in problem solving. Volume over quality is prioritised, unconventional ideas are welcomed and developed upon, and everyone is urged to participate so that they can all contribute to the development of a wide range of innovative solutions.



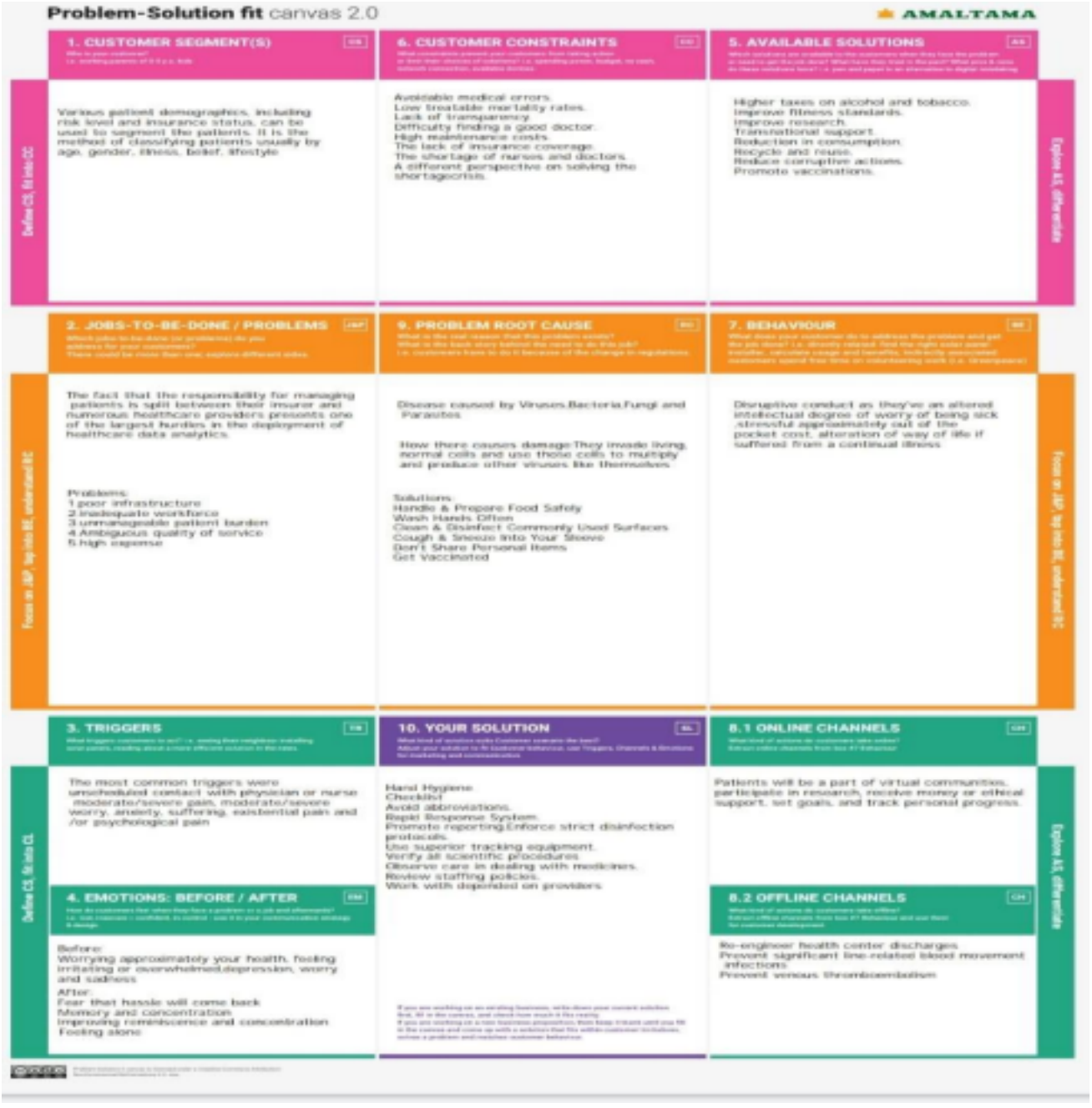
3.3 Proposed Solution

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

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	EHR data matched patient-reported data in 23.5 percent of records in a study at an ophthalmology practise. Patients' EHR data did not agree in any way when they reported having three or more eye health complaints.
2.	Idea / Solution description	Predictive analytics can create patient journey dashboards and disease trajectories that can lead to effective, and result driven healthcare. It improves treatment delivery, cuts costs, improves efficiencies, and so on.
3.	Novelty / Uniqueness	Healthcare data frequently resides in several locations. from various departments, such as radiology or pharmacy, to various source systems, such as EMRs or HR software. The organisation as a whole contributes to the data. This data becomes accessible and usable when it is combined into a single, central system, such as an enterprise data warehouse (EDW).
4.	Social Impact / Customer Satisfaction	Enhanced diagnosis Improved medical treatment Improved health results Improved relationships with patients More positive health indicators
5.	Business Model (Revenue Model)	The two factors that have the biggest negative effects on hospital income are claim denials and patient incapacity to pay their part. 90% more uncollectible claim denials were written off by hospitals and healthcare systems in 2017 compared to the preceding six years.
6.	Scalability of the Solution	A variety of institutions must store, evaluate, and take action on the massive amounts of data being produced by the health care sector as it expands quickly. India is a vast, culturally varied nation with a sizable population that is increasingly able to access centralised healthcare services.

3.4 Problem Solution fit

The term "problem-solution fit" simply refers to the fact that you have identified a problem with a client and that the solution you have developed to address it truly resolves the client's issue.



CHAPTER-4

REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User SignUp	SignUp through Form SignUp through Gmail
FR-2	Credential Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login through Form
FR-4	Forgot password	OTP via email
FR-5	Data collection	The majority of hospitals in the United States today have electronic health records, which are the focus of hospital data analytics. A comprehensive record that contains all relevant information about the patient's health is known as a digital health record.

4.2 Non-Functional requirements

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Reliability**	A random sample of 10% of the medical records was examined independently by two reviewers to ascertain inter-rater reliability. We applied a straightforward computer-based random sample technique to choose these medical records.
NFR-2	**Maintainability**	Based on four layers, the Maintainability Information Database (MID) is organised. The database has all the project data pertaining to maintenance planning, and this data is integrated with the BIM models of the project for improved project aspect integration.
NFR-3	**Performance**	People value their health more than the majority of other products and services. Both governments and people spend a lot of money on healthcare. People want to choose their healthcare with knowledge.

CHAPTER-5

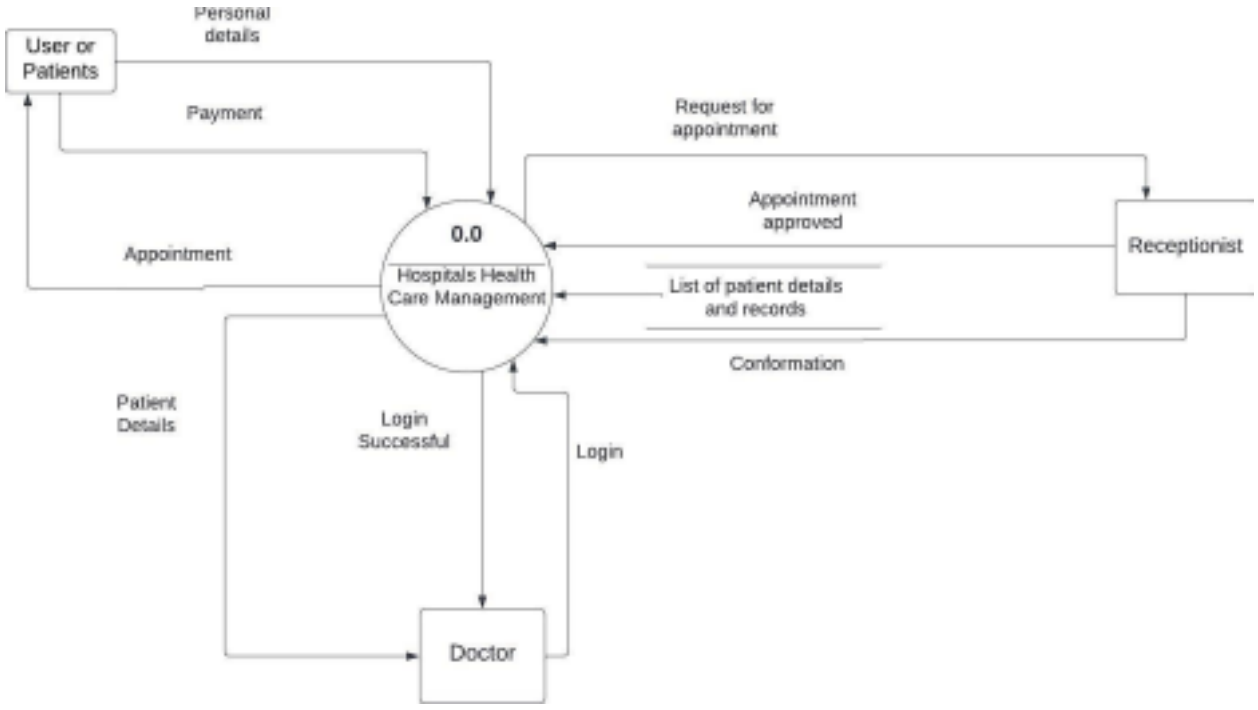
PROJECT DESIGN

5.1 Data Flow Diagrams

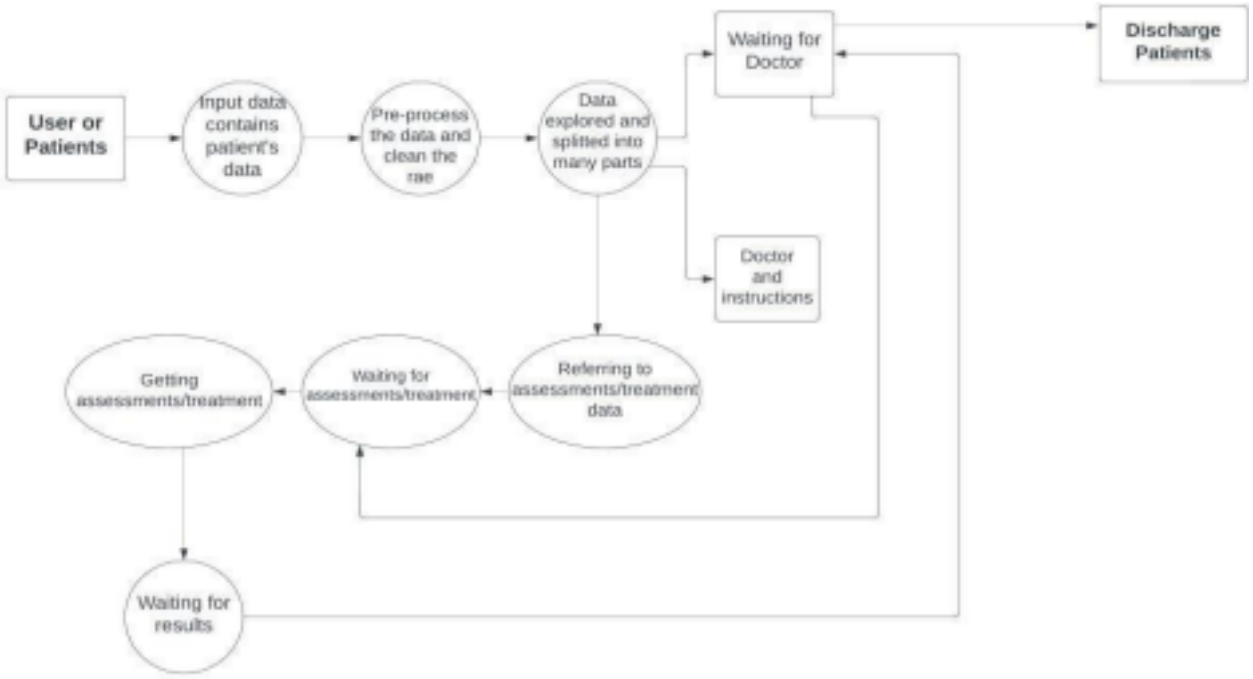
The classic visual representation of how information moves through a system is a data flow diagram (DFD). A tidy and understandable DFD can graphically represent the appropriate quantity of the system demand. It demonstrates how information enters and exits the system, what modifies the data, and where information is kept.

Data Flow Diagram for Heart Disease Prediction Dashboard:

DFD LEVEL 0:



Flow:



1) User creates an account in the application.

2) User enters the medical records in the dashboard.

3) User can view the visualizations of trends in the form of graphs and charts for his/her medical records with the trained dataset.

4) User can view the accuracy of probability of occurrence of heart disease in the dashboard.

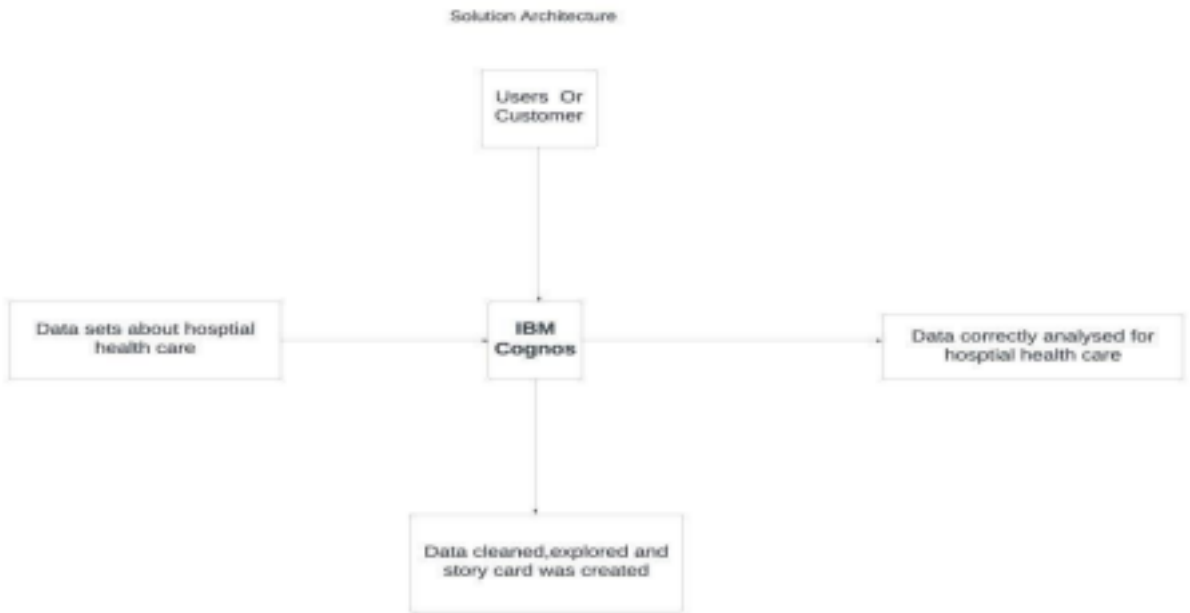
5.2 Solution & Technical Architecture

A complicated process with numerous sub-processes, solution architecture connects business issues with technological solutions. Its objectives 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.

Solution Architecture Diagram:



Technology Stack (Architecture & Stack):

Technical Architecture:

Analysis of Hospitals Health-Care data:

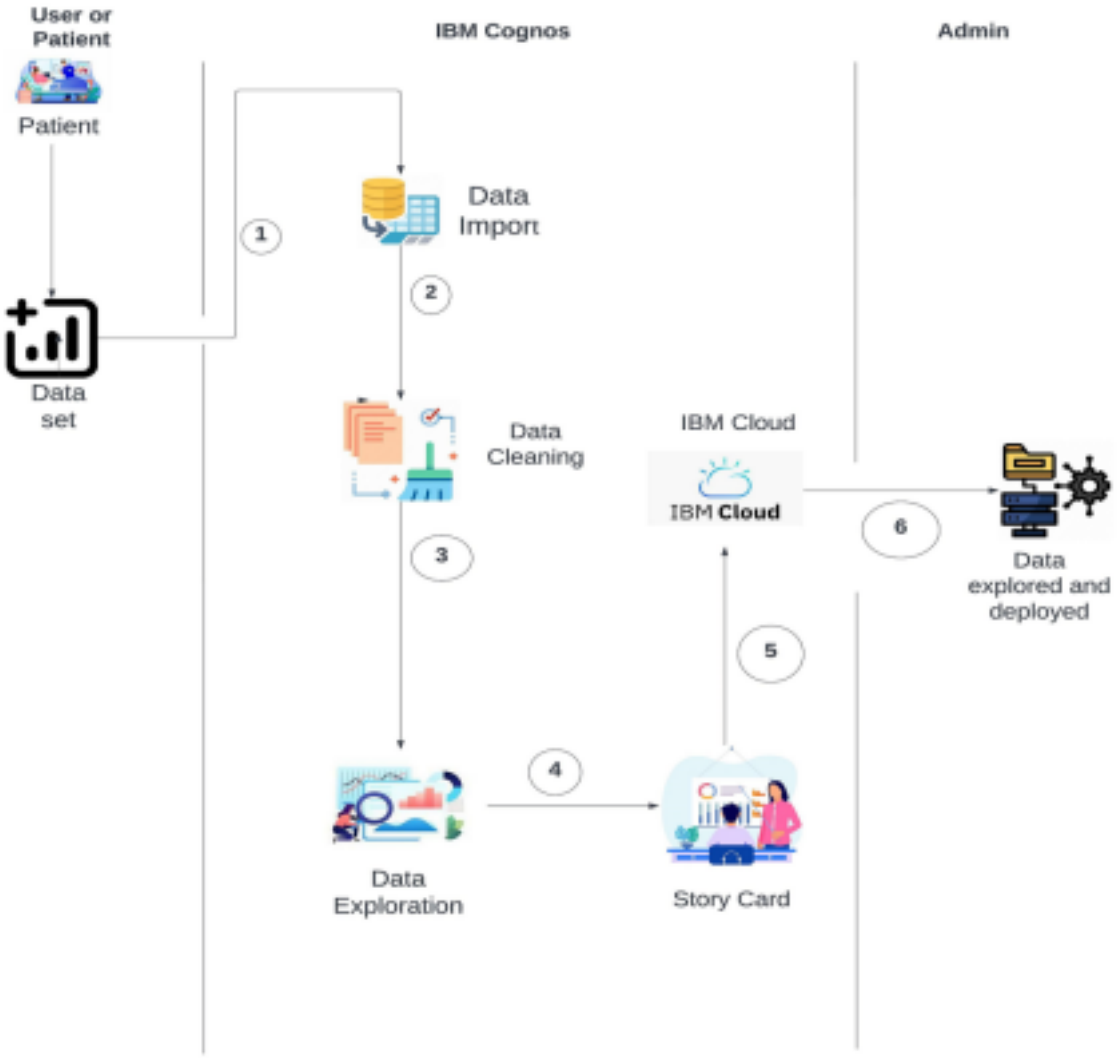


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	IBM Cognos / Python .
2.	Data Set	The data set prepared for hospitals health care	Python .
3.	IBM Cognos	Data analytics platform	IBM Watson service
4.	Data Import	Data set is imported in IBM cognos	IBM Watson Assistant
5.	Data Cleaning	Data is cleaned by using some mathematical techniques such as mean,mode etc.to clean the null and missing data.	IBM Assistant
6.	Data Exploration	Cleaned data can be explored.	IBM Cognos
7.	Story Card	Data is explored and story card was prepared for visual representation	IBM Cognos
8.	IBM Cloud	Storage of data	IBM DB2
9.	Data Explored and Deployed	Purpose of External API to explored and deployed	Data deployed to user by UI
10.	Admin	Purpose of Data set model	Recognition of data set model etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source	Open source model is used for the data set	Python
2.	Security Implementations	Security for our data set	SHA 256, SHA 1
3.	Scalable Architecture	health care service utilizes the relational patient data and big data analytics to tailor the medication recommendations	Python
4.	Availability	The availability of technology used in data analytics	Python Anaconda distribution and jupyter notebook is available and open source application
5.	Performance	The performance of the application and its efficiency	Python and other languages is that Python is usually interpreted. Interpreted languages tend to perform worse than compiled languages, each command takes up a greater number of machine instructions .

5.3 User Stories

User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Patient)	Registration	USN-1	As a user, I can gather the details of the patients.	I can access datasets my account Kaggle	High	Sprint 1
			As an Analyst, I will check the data set andclean the dataset to create an efficient model.	I can clean the datasets using Cognos Analytics	High	Sprint-1
		USN-3	As a user, I can register through Gmail		Medium	Sprint-1
	Login	USN-4	As a user, I can log in by entering email & password		High	Sprint-1
	Forgot Password	USN-5	As a user, if i forgot my password, by clicking forgot	By entering the OTP sent Ia email.	High	Sprint-1
			OTP is sent to email,			

	Data collection	USN-6	As a user, I can upload the input data set in IBM Cognos		High	Sprint-1

CHAPTER-6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation Product Backlog, Sprint Schedule, and Estimation:

Use the below template to create product backlog and sprint schedule

User Type	Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Customer (Patient)	Sprint 1	Datasets	USN-1	As a user, I can enter the details of the patients working in our organisation of the detail	2	High	R. Vishnu Prasath
Analyst	Sprint 1		USN-2	As a Analyst, I will check the data set and clean the dataset to create an efficient model .	1	High	P. Thamizharasu
	Sprint 1		USN-3	As an Analyst I will also correct the raw data and create a data module	2	Medium	M. Premkumar
	Sprint 2	Cleaning, Exploring data and creating model	USN-4	As an Analyst I can create a Exploratory data analysis to identify the important factors of	2	High	M. Viswanath

				patient data set			
	Sprint 2		USN-5	As a Data analyst, I create a predicted model by also preparing story	1	High	R. Vishnu Prasath
	Sprint 3	Data Prediction	USN-6	As a Data analyst, I will create different types of models in explored data to identify suitable	5	High	R. Vishnu Prasath
				model with effectively and efficiently			
Admin	Sprint 4	Creation of deployed data UI	USN-7	As an Analyst, I will import my analysed model into suitable framework	2	High	M. Prem kumar
	Sprint 4		USN-8		5	High	P. Thamizharasu

6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	15	5 Days	24 Oct 2022	29 Oct 2022	15	29 Oct 2022
Sprint-2	15	5 Days	31 Oct 2022	05 Nov 2022	15	05 Nov 2022
Sprint-3	15	5 Days	07 Nov 2022	12 Nov 2022	15	12 Nov 2022
Sprint-4	15	5 Days	14 Nov 2022	19 Nov 2022	15	19 Nov 2022

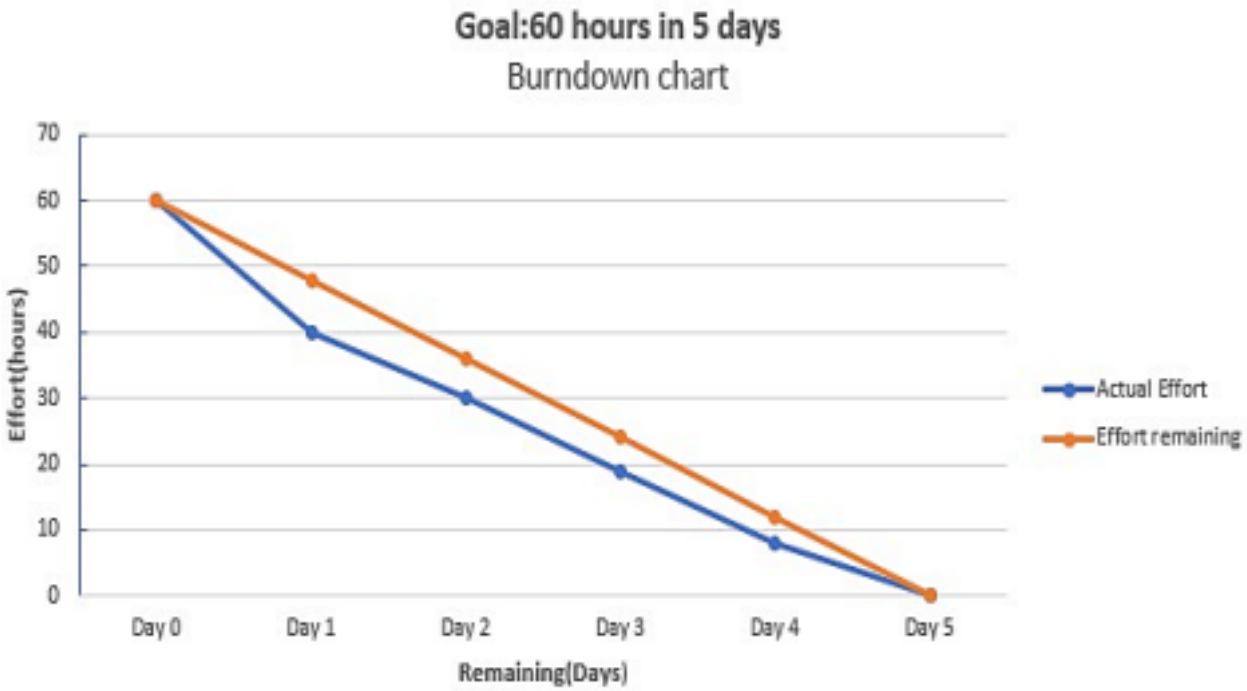
Velocity:

We have a 5-day sprint duration, and the velocity of the team is 15 (points per sprint). The team's average velocity (AV) per iteration unit (story points per day)

Actual Velocity = Sprint Duration/Velocity = 15/5 = 3

Burndown Chart:

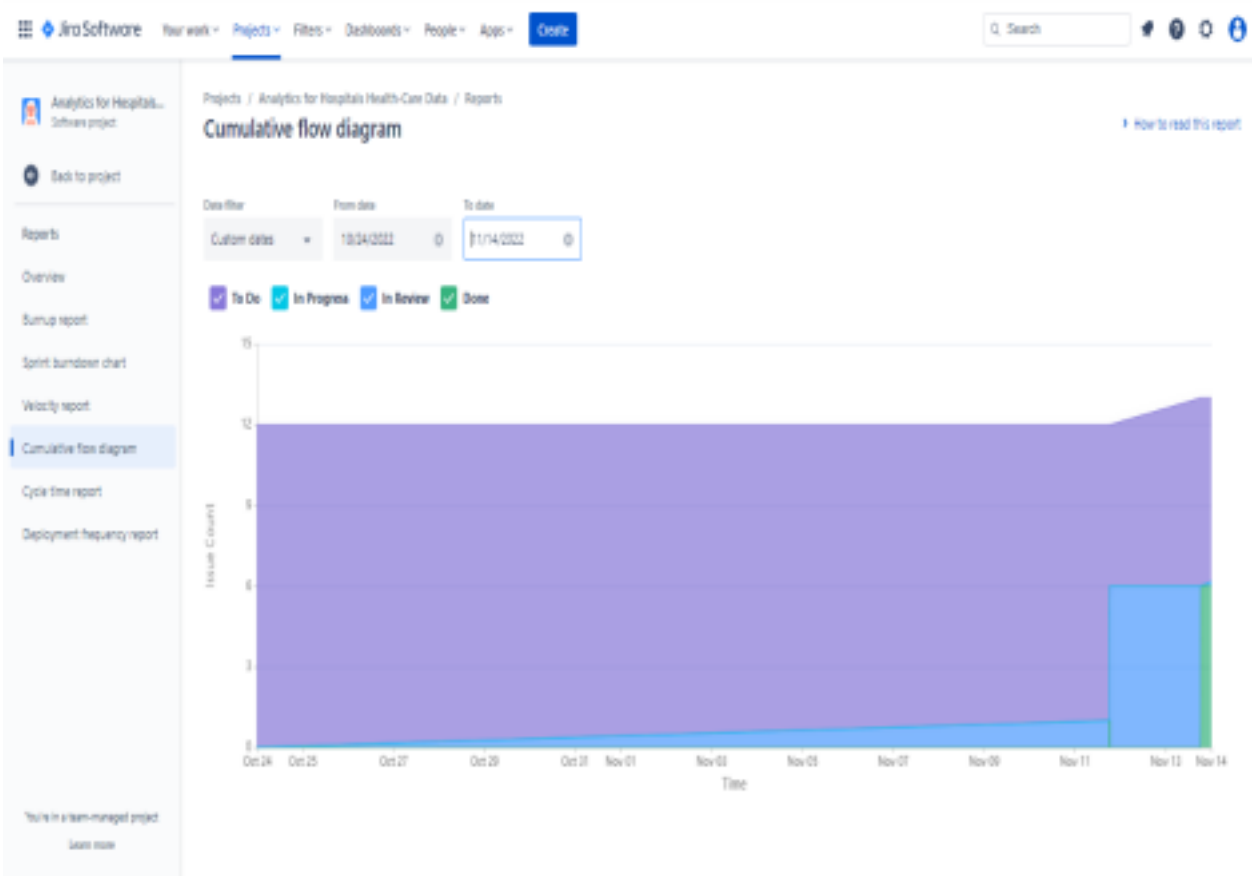
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



6.3 Reports from JIRA

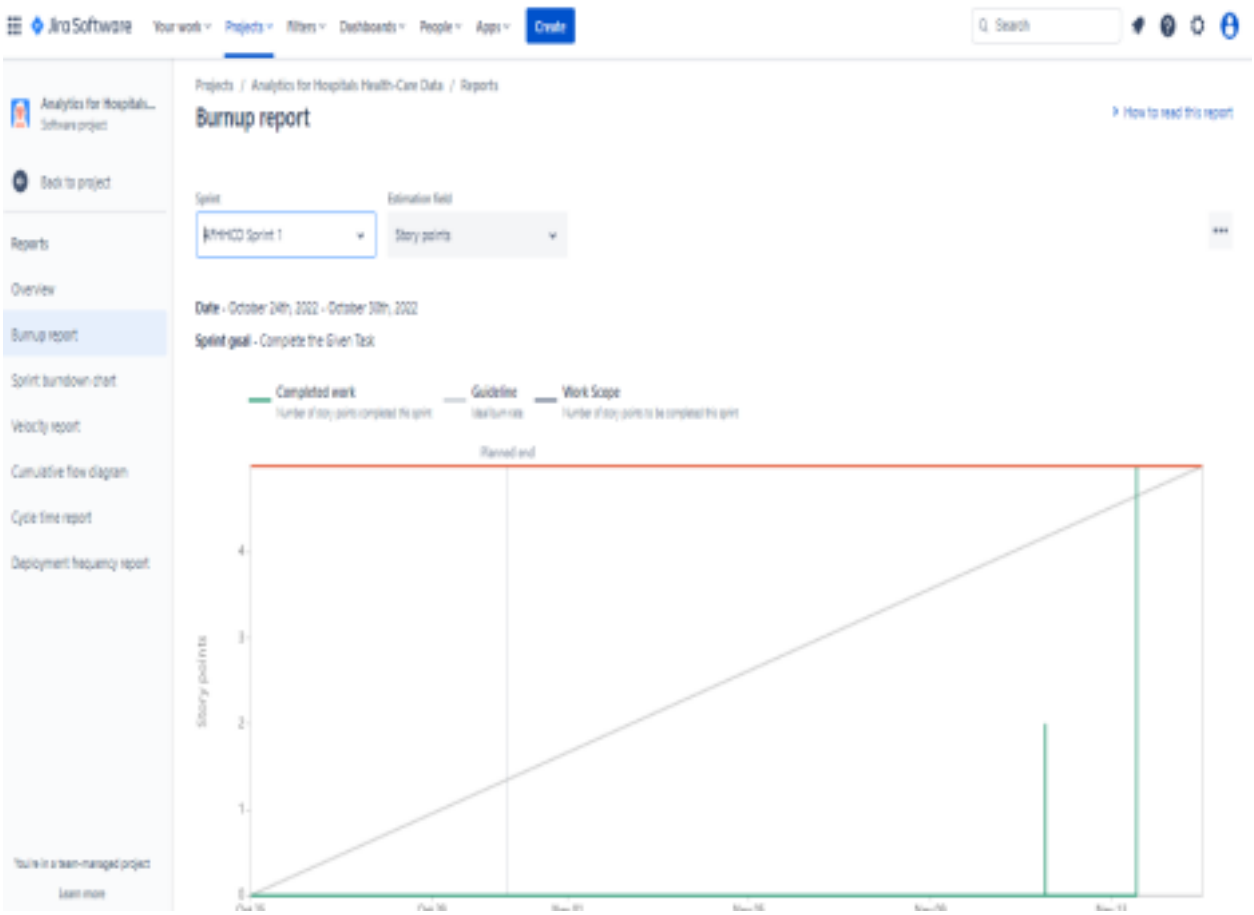
Jira brings teams together for everything from agile software development and customer service to start-ups and companies. Jira assists teams in planning, assigning, tracking, reporting, and managing work. Jira Software, the top solution for agile teams, helps software teams build better. You can define project categories as a Jira administrator so that your team can view work from related projects in a single location. Your team has access to filters, reports, advanced search categories, and more.

CUMULATIVE JIRA :

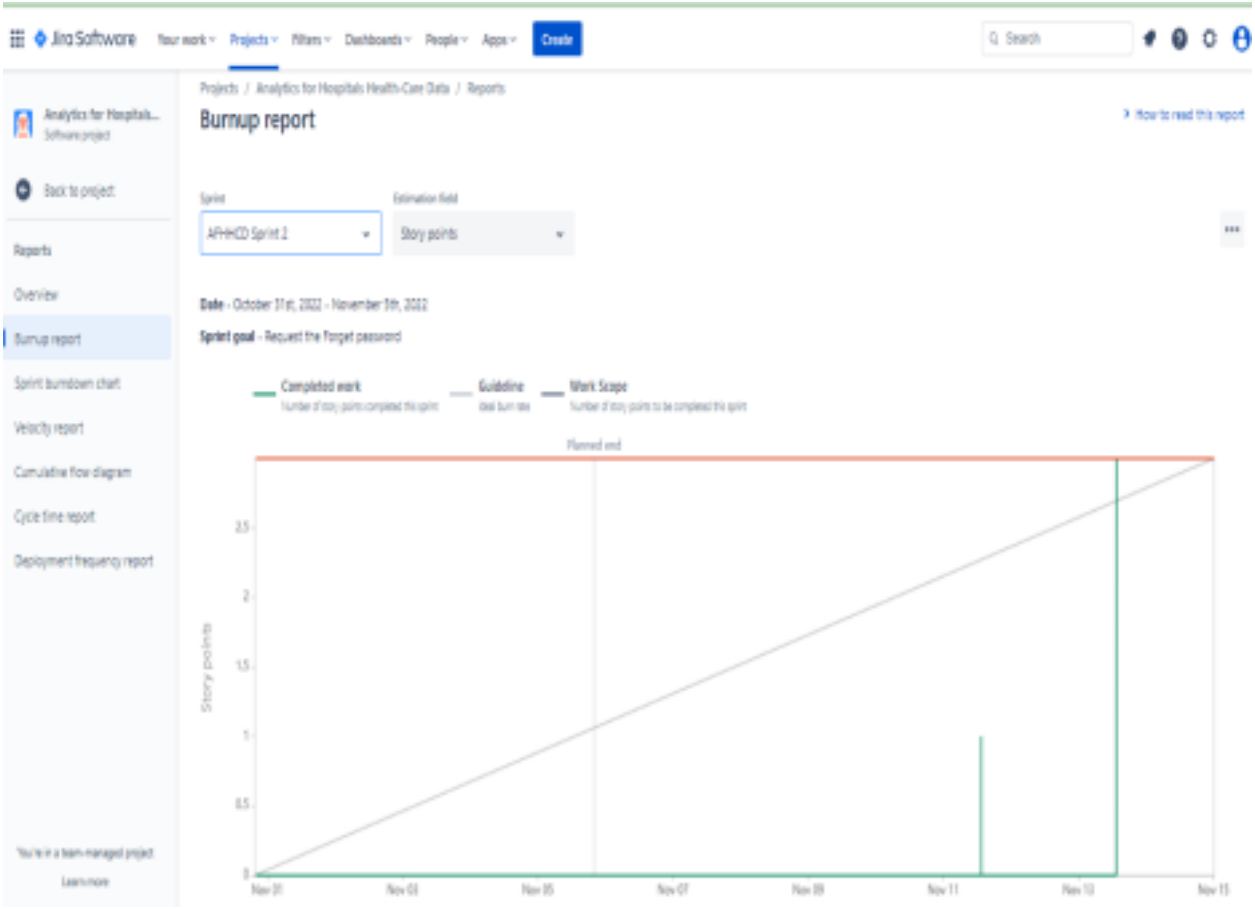


BURNUP REPORT:

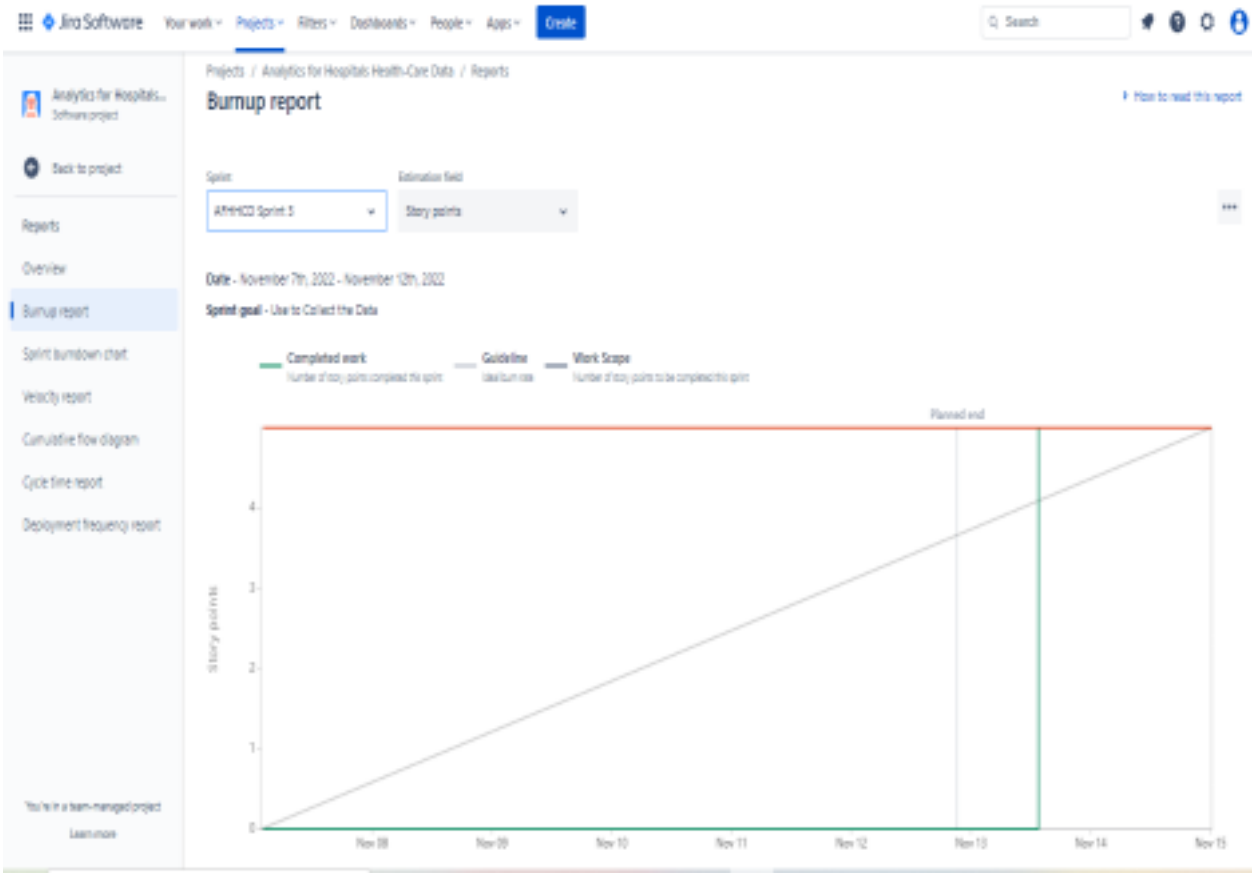
SPRINT-1:


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Analytics for Hospitals...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Projects / Analytics for Hospitals Health-Care Data / Reports
Burnup report
How to read this report
Sprint
Estimation field
Story points
Date - October 24th, 2022 - October 30th, 2022
Sprint goal - Complete the Given Task
Completed work
Guideline
Work Scope
Planned end
Story points
0
1
2
3
4

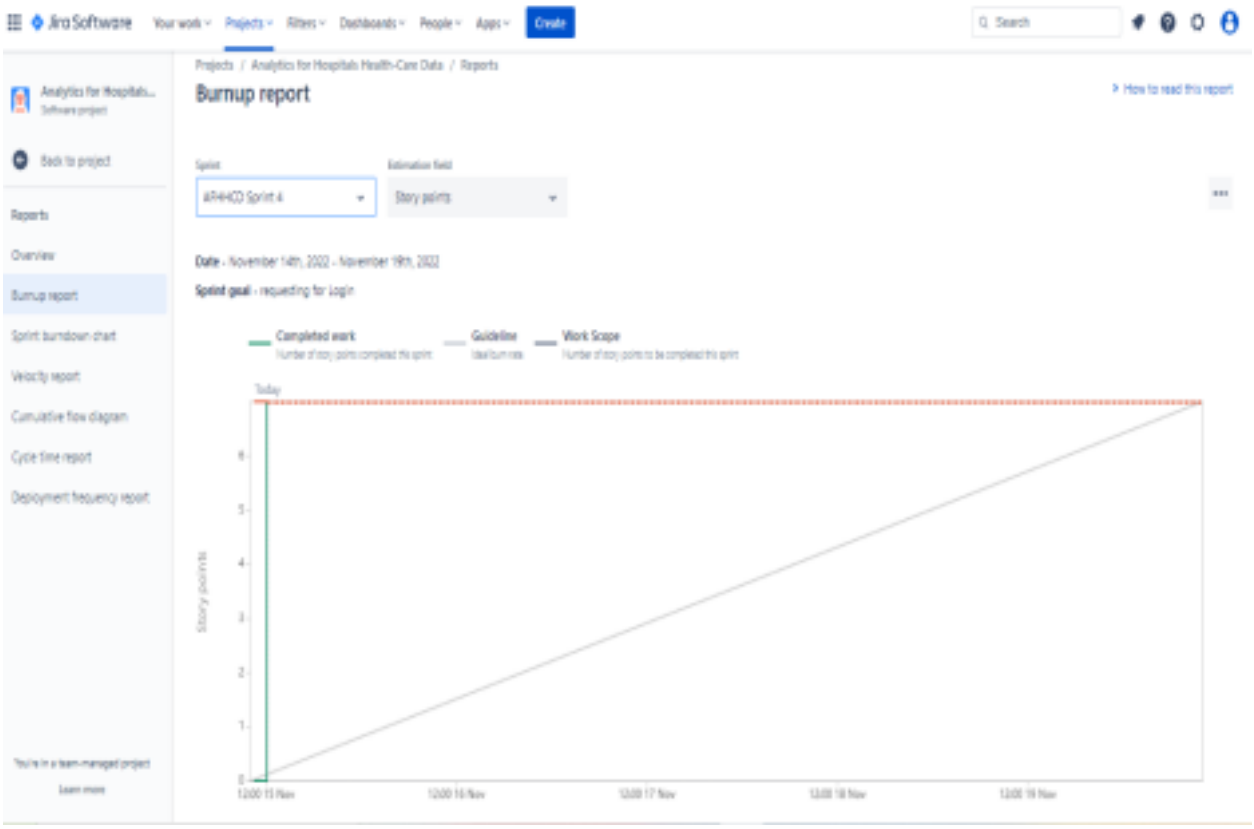
SPRINT-2:


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Analytics for Hospitals...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Projects / Analytics for Hospitals Health-Care Data / Reports
Burnup report
How to read this report
Sprint
APHHCD Sprint 1
Estimation field
Story points
Date - October 31st, 2022 - November 9th, 2022
Sprint goal - Request the forget password
Completed work
Guideline
Work Scope
Planned end
Story points
2.5
2
1.5
1
0.5
0
Nov 01
Nov 03
Nov 05
Nov 07
Nov 09
Nov 11
Nov 13
Nov 15

SPRINT-3:

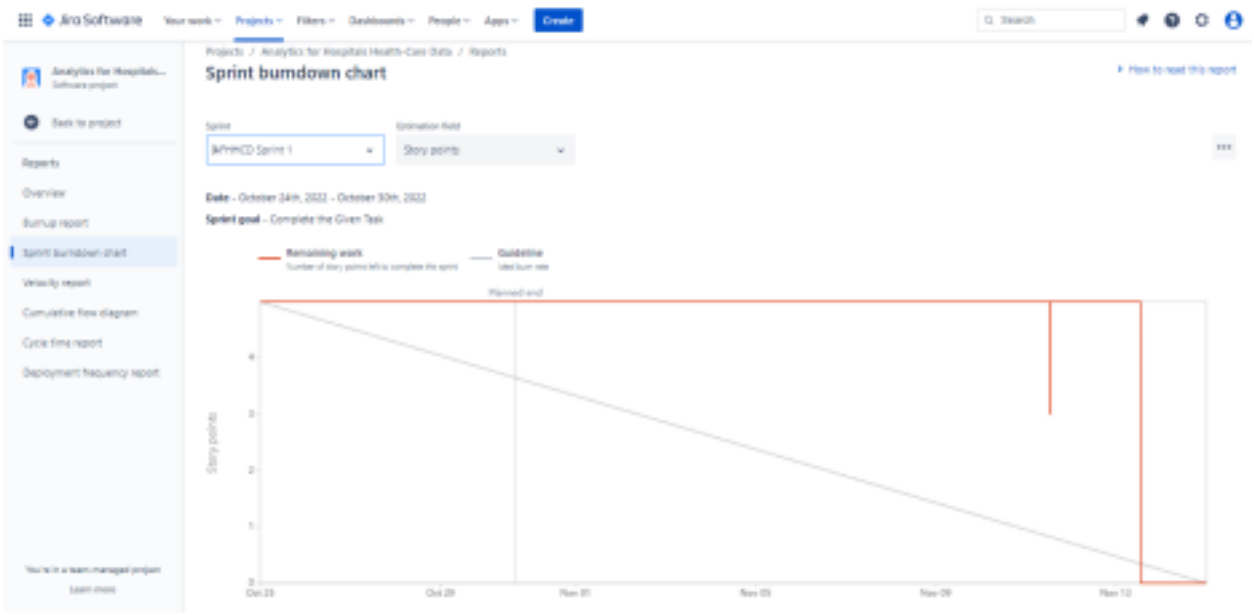

Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Analytics for Hospitals...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Projects / Analytics for Hospitals Health-Care Data / Reports
Burnup report
How to read this report
Sprint
AHHCD Sprint 5
Estimation field
Story points
Date - November 7th, 2022 - November 12th, 2022
Sprint goal - Use to Collect the Data
Completed work
Guideline
Work Scope
Planned end
Story points
0
1
2
3
4
Nov 08
Nov 09
Nov 10
Nov 11
Nov 12
Nov 13
Nov 14
Nov 15

SPRINT-4:

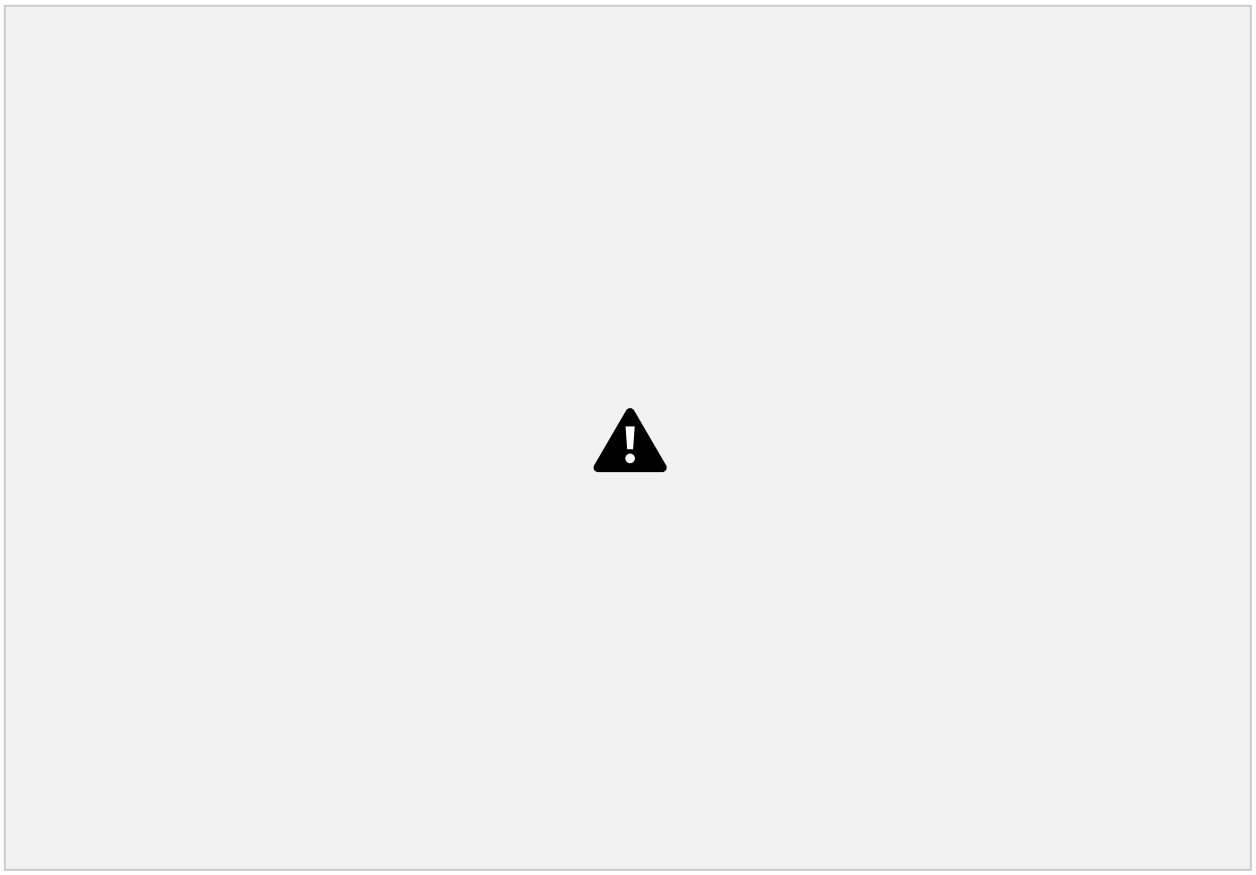


Burndown Chart:

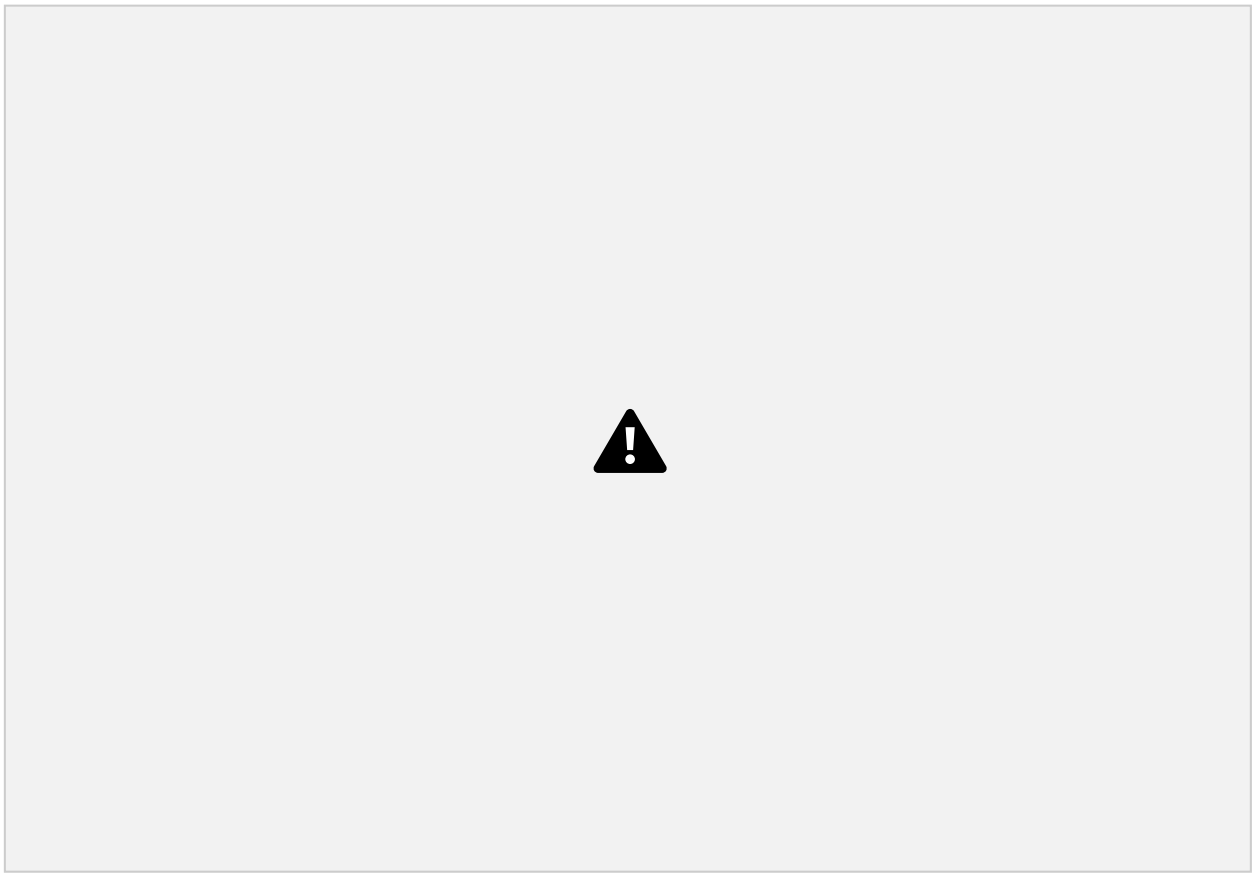
Burndown Chart Sprint-1:



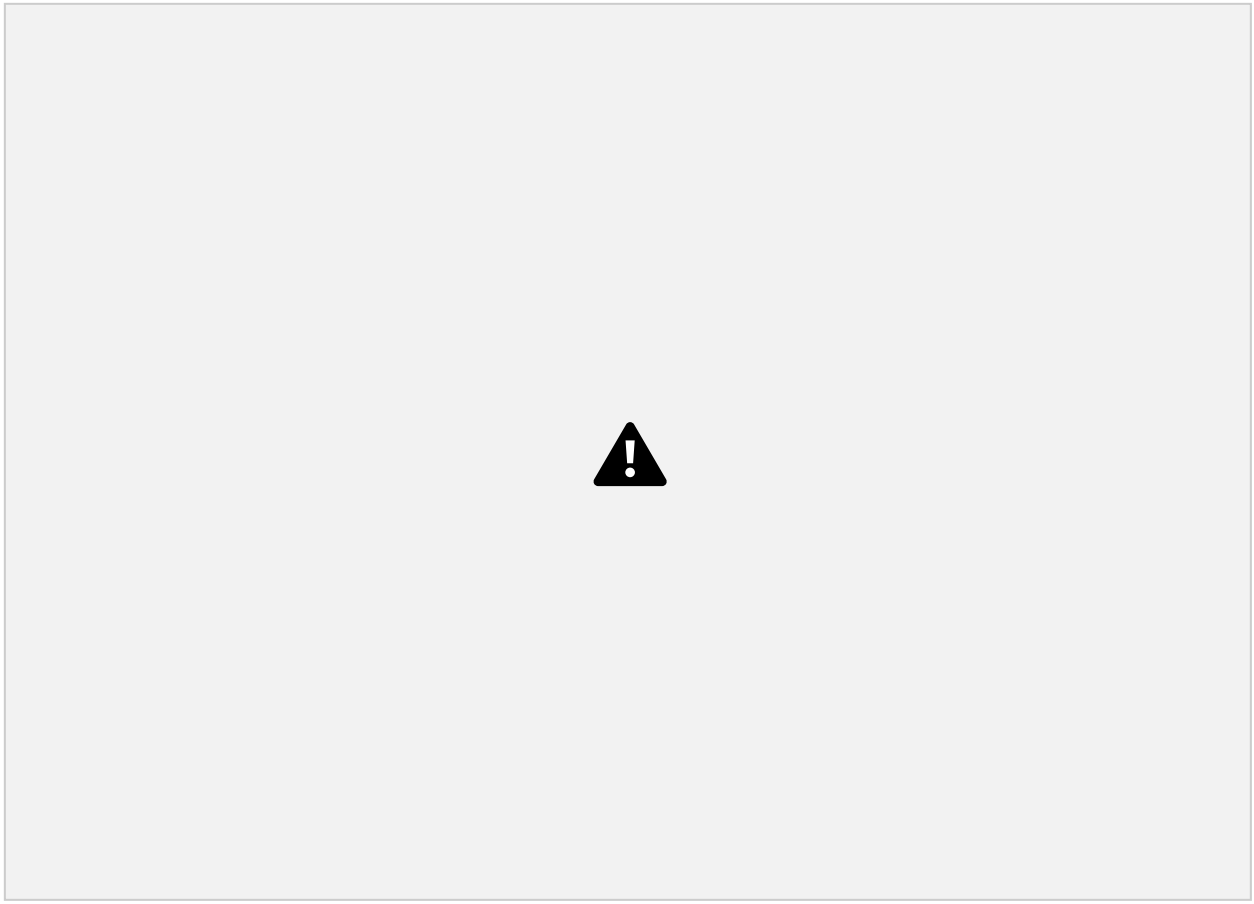
Burndown Chart Sprint-2:



Burndown Chart Sprint-3:



Burndown Chart Sprint-4:



CHAPTER-7
CODING & SOLUTIONING

7.1 Random Forest

Random Forest is a supervised learning algorithm. Random forest can be used for both classification and regression problems, by using random forest regressor we can use random forest on regression problems. But we have used random forest on classification in this internship project so we will only consider the classification part.

7.1.1 Random Forest pseudocode

• Randomly select “k” features from total “m” features. Where k << m • Among the “k” features, calculate the node “d” using the best split point. • Split the node into daughter nodes using the best split.
• Repeat 1 to 3 steps until the “l” number of nodes has been reached. • Build forest by repeating steps 1 to 4 for “n” number times to create “n” number of trees.

7.1.2 Random Forest prediction pseudocode

✓ Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target).
✓ Calculate the votes for each predicted target.
✓ Consider the highly voted predicted target as the final prediction from the random forest algorithm. Code: max_accuracy = 0 for x in range(500): rf_classifier = RandomForestClassifier(random_state=x)

Code:

```
max_accuracy = 0

for x in range(500):

rf_classifier = RandomForestClassifier(random_state=x)
rf_classifier.fit(X_train,Y_train)

Y_pred_rf = rf_classifier.predict(X_test)
```

```
current_accuracy = round(accuracy_score(Y_pred_rf,Y_test)*100,2)

if(current_accuracy>max_accuracy):

max_accuracy = current_accuracy

best_x = x

print(max_accuracy)

print(best_x)

rf_classifier = RandomForestClassifier(random_state=best_x)

rf_classifier.fit(X_train,Y_train)

Y_pred_rf = rf_classifier.predict(X_test)

Y_pred_rf.shape score_rf = round(accuracy_score(Y_pred_rf,Y_test)*100,2) score_rf
```

7.2. K-Nearest Neighbors

We can implement a KNN model by following the below steps:

- Load the data
- Initialize the value of k

- For getting the predicted class, iterate from 1 to total number of training data points

Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it's the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.

○ Sort the calculated distances in ascending order based on distance

values ○ Get top k rows from the sorted array

○ Get the most frequent class of these rows

○Return the predicted class

Code:

```
knn_classifier=

KNeighborsClassifier(n_neighbors=31,leaf_size=30)

knn_classifier.fit(X_train,Y_train)

Y_pred_knn = knn_classifier.predict(X_test)

score_knn = round(accuracy_score(Y_pred_knn,Y_test)*100,2)

score_knn
```

7.3 Decision Tree Pseudocode:

● Place the best attribute of the dataset at the root of the tree.

● Split the training set into subsets. Subsets should be made in such a way that each subset contains data with the same value for an attribute

● Repeat step 1 and step 2 on each subset until you find leaf nodes in all the branches of the tree.

Assumptions while creating a Decision Tree- At the beginning, the whole training set is considered as the root. Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model. Records are distributed recursively on the basis of attribute values. Order to place attributes as root or internal node of the tree is done by using some statistical approach.

The popular attribute selection measures:

- Information gain
- Gini index

Attribute selection method- Choosing which attribute to place at the root of the tree or at various levels as internal nodes is a difficult step when a dataset has "n" attributes. The problem cannot be resolved by simply choosing any node at random to be the root. A random technique could produce subpar outcomes with little accuracy. Researchers researched on this attribute selection issue and came up with some solutions. They recommended employing criteria such as information gain, Gini index, etc. Every attribute's value will be determined using these criteria. The attributes are arranged in the tree according to the order in which the values are sorted, with the attribute with the highest value (in the case of information gain) being placed at the root. When utilising information,

Gini Index -The frequency with which a randomly selected element will be erroneously detected is gauged by the Gini Index. It implies that an attribute with a lower Gini index ought to be chosen.

Code: dt_classifier = DecisionTreeClassifier(

max_depth=20,

min_samples_split=2,

min_samples_leaf=1,

min_weight_fraction_leaf=0.00001, max_features='auto',

random_state=46)

dt_classifier.fit(X_train, Y_train)

Y_pred_dt=dt_classifier.predict(X_test) score_dt =
round(accuracy_score(Y_pred_dt,Y_test)*100,2)

score_dt

7.4 Naïve Bayes

Bayes'Theorem is stated as:

$$P(h|d) = (P(d|h) * P(h)) / P(d)$$

P(h|d) is the probability of hypothesis h given the data d. This is called the posterior probability.
P(d|h) is the probability of data d given that the hypothesis h was true. **P(h)** is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.
P(d) is the probability of the data (regardless of the hypothesis).

We are interested in calculating the posterior probability of P(h|d) from the prior probability p(h) with P(D) and P(d|h). After calculating the posterior probability for a number of different hypotheses, we will select the hypothesis with the highest probability. This is the maximum probable hypothesis and may formally be called the (MAP) hypothesis.

This can be written as:

MAP(h) = max(P(h|d)) or
MAP(h) = max((P(d|h) * P(h)) / P(d)) or
MAP(h) = max(P(d|h) * P(h))

The P(d) is a normalizing term which allows us to calculate the probability. We can drop it when we are interested in the most probable hypothesis as it is constant and only used to normalize. Back to classification, if we have an even number of instances in each class in our training data, then the probability of each class (e.g. P(h)) will be equal. Again, this would be a constant term in our equation, and we could drop it so that we end up with:

MAP(h) = max(P(d|h))

Naive Bayes is a classification algorithm for binary (two-class) and multi class classification problems. The technique is easiest to understand when described using binary or categorical input values. It is called Naive Bayes or Idiot Bayes because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable. Rather than attempting to

calculate the values of each attribute value P (d1, d2, d3|h), they are assumed to be conditionally independent given the target value and calculated as P(d1|h) * P(d2|H) and so on. This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not hold.

MAP(h) = max(P(d|h) * P(h))

Gaussian Naïve Bayes:

mean(x) = 1/n * sum(x)

Where n is the number of instances and x are the values for an input variable in your training data. We can calculate the standard deviation using the following equation:

standard deviation(x) = sqrt (1/n * sum(xi-mean(x)^2))

This is the square root of the average squared difference of each value of x from the mean value of x, where n is the number of instances, sqrt() is the square root function, sum() is the sum function, xi is a specific value of the x variable for the i'th instance and mean(x) is described above, and ^2 is the square. Gaussian PDF with a new input for the variable, and in return the Gaussian PDF will provide an estimate of the probability of that new input value for that class.

pdf (x, mean, sd) = (1 / (sqrt (2 * PI) * sd)) * exp (-((x-mean^2)/(2*sd^2)))

Where pdf(x) is the Gaussian Probability Density Function (PDF), sqrt () is the square root, mean and sd are the mean and standard deviation calculated above, Pi is the numerical constant, exp () is the numerical constant e or Euler's number raised to power and x is the input value for the input variable.

Code:

```
 nb_classifier = GaussianNB( var_smoothing=1e-50)

nb_classifier.fit(X_train,Y_train)
nb_classifier.predict(X_test)

Y_pred_nb = nb_classifier.predict(X_test)

score_nb = round(accuracy_score(Y_pred_nb,Y_test)*100,2)

score_nb
```

7.6 Libraries used

The vast array of standard libraries that come with Python cover topics like web services tools, string manipulation, data analysis, and machine learning, among others. These built-in libraries make it easier to handle difficult programming tasks by reducing the amount of code needed to accomplish each task and providing the user with a number of useful functions.

7.6.1 Data Visualization

● Matplotlib: Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical mathematics extension NumPy, a big data numerical handling resource.

- ○ pyplot
- ○ rcParams
- ○ rainbow

● Seaborn: Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with dataframes and the Pandas library. The graphs created can also be customized easily.

7.6.2 Data Manipulation

● NumPy: The NumPy library in python is used for scientific computing and array manipulation. It can perform different operations such as indexing of an array etc.

● Pandas: The Pandas library in python is used for structuring, manipulating, and organizing data in a tabular structure called the data frame which is further used for data analysis.

● Scikit-learn: ○ sklearn.model_selection ■ train_test_split ○ sklearn.preprocessing ■ StandardScaler ■ LabelEncoder

7.6.3 Data Modeling

● Scikit-learn: Scikit-learn is one of the most useful libraries that python offers. It has various statistical learning algorithms such as regression models (linear regression, logistic regression), SVM's, random forest for classification tasks and k-means for clustering, etc.

○ sklearn.ensemble.RandomForestClassifier

○ sklearn.neighbors.KNeighborsClassifier

○ sklearn.tree.DecisionTreeClassifier

○ sklearn.naive_bayes.GaussianNB

7.6.4 Data Validation

● **Scikit-learn-metrics:** The sklearn.metrics module implements several loss, score, and utility functions to measure classification performance. sklearn.metrics - log_loss, roc_auc_score, precision_score, f1_score, recall_score, roc_curve, auc, plot_roc_curve, classification_report, confusion_matrix, accuracy_score, fbeta_score, matthews_corrcoef

● **Mlxtend**: Mlxtend (machine learning extensions) is a Python library of useful tools for day-to-day data science tasks. mlxtend.plotting - plot_confusion_matric

CHAPTER-8

TESTING

8.1 Testing and Validations

Validation is a complex process with many possible variations and options, so specifics vary from database to database, but the general outline is:

• **Requirement Gathering**

o The Sponsor decides what the database is required to do based on regulations, company needs, and any other important factors.

o The requirements are documented and approved.

• **System Testing**

o Procedures to test the requirements are created and documented. o The version of

the database that will be used for validation is set up. o The Sponsor approves the

test procedures.

o The tests are performed and documented.

o Any needed changes are made. This may require another, shorter round of testing and documentation.

• **System Release**

o The validation documentation is finalized.

o The database is put into production. .

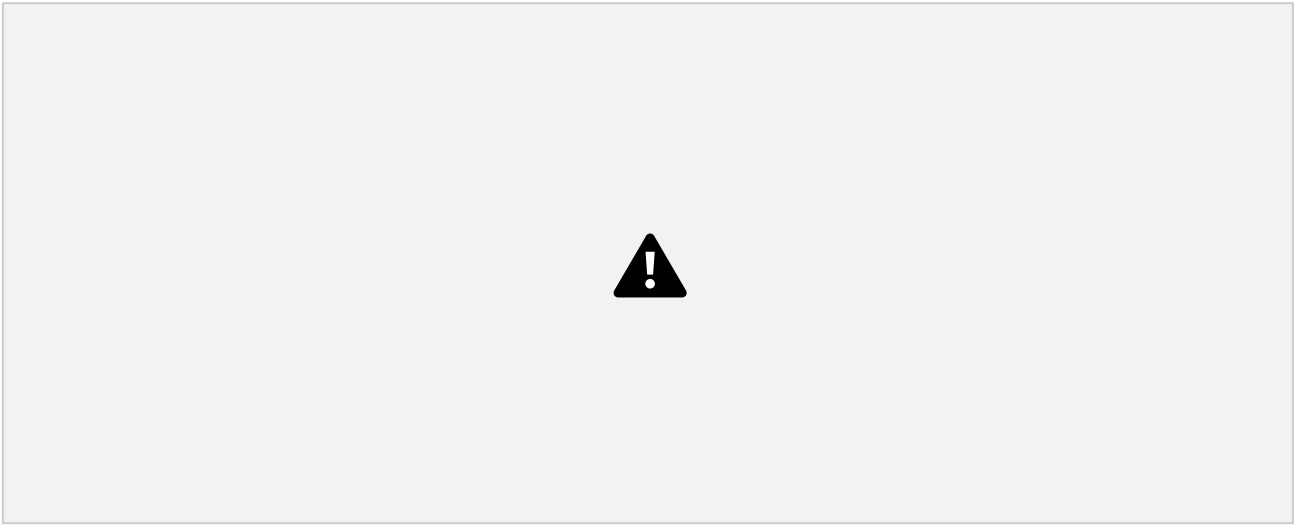
8.1.1 User Acceptance Testing

1.Purpose of Document

The purpose oft his documentation to briefly explain the test coverage and open issues of the[Visualizing and Predicting Heart Diseases]project at the time of the release to UserAcceptanceTesting(UAT).

2.DefectAnalysis

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



3.TestCaseAnalysis

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

Section	Total Cases	Not Tested	Fail	Pass
City_code	22	0	0	22
Available ExtraRooms in Hospital	31	0	0	31
Department	4	0	0	4
BedGrade	9	0	0	9

TypeofAdmission	2	0	0	2

8.1.2 Test Cases Report

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
City_Code	Dashboard/ report, report	Cognos Analytics	Verify the dataset for accurate performance	A quality dataset	1.Upload the dataset 2.Explore the data 3.Create dashboard/ Report,Story	Accurate Prediction	Working as expected	Pass	Cognos analytics to accurate predict of patients City_Code	yes	high	R. Vishnu Prasath
Avaliable Extra Room in Hospital	Dashboard/ report, report	Cognos Analytics	Verify the dataset for accurate performance	A quality dataset	1.Upload the dataset 2.Explore the data 3.Create dashboard/ Report,Story	Accurate Prediction	Working as expected	fail	Cognos analytics to accurate predict of patients Avslisble Extra Room in Hospital	no	low	M. Prem Kumar
Department	Dashboard/ report, report	Cognos Analytics	Verify the dataset for accurate performance	A quality dataset	1.Upload the dataset 2.Explore the data 3.Create dashboard/ Report,Story	Accurate Prediction	Working as expected	Pass	some data not accuracy	yes	high	P. Thamizharasu

Ward_ Type	Dashboard/ report, report	Cognos Analytics	Verify the dataset for accurate performance	A quality dataset	1.Upload the dataset 2.Explore the data 3.Create dashboard/ Report ,Stor	Accurate Prediction	Working as expected	Pass	Cognos analytics to accurate predict of patients Ward_ Type	yes	high	R. Vishnu Prasath

Bed Grade	Dashboard/ report, report	Cognos Analytics	Verify the dataset for accurate performance	A quality dataset	1.Upload the dataset 2.Explore the data 3.Create dashboard/ Report ,Story	Accurate Prediction	Working as expected	fail	Cognos analytics to accurate predict of patients Bed Grade	no	low	R. Vishnu Prasath
Type of Admission	Dashboard/ report, report	Cognos Analytics	Verify the dataset for accurate performance	A quality dataset	1.Upload the dataset 2.Explore the data 3.Create dashboard/ Report ,Story	Accurate Prediction	Working as expected	Pass	Cognos analytics to accurate predict of Type of Admission	yes	high	M. Viswanath

8.1.3 Performance Testing

Project team shall fill the following information in model performance testing template.

S.No	Parameter	Screenshot / Values
1.	Dashboard designs	No of Visualizations / Graphs –11 dashboard tabs with 1-2 visualizations in each dashboard
2.	Data Responsiveness	It hides certain aspects of the visualization if the size is limited, to maximize the space that is available to display data. • There was two different datasets with the common column and full outer join was done by that common column. • There was another dataset with various continuous values , those values was grouped as common.
3.	Amount Data to Rendered (DB2 Metrics)	There are some relevant datasets are uploaded in the IBM DB2.
4.	Utilization of Data Filters	If any similar datasets in the provided area this is reduced by joining the common column of those two dataset. i.e. Cardinality is used.
5.	Effective User Story	No of Scene Added – 10 stories with 1-2 visualizations in each story
6.	Descriptive Reports	No of Visualizations / Graphs – 2 reports with 3 – 5 visualization in each report

8.2 Testing Levels

8.2.1 Functional Testing:

This type of testing is done against the functional requirements of the project
. Types:

Unit testing: Each unit /module of the project is individually tested to check for bugs. If any bugs found by the testing team, it is reported to the developer for fixing.

Integration testing: All the units are now integrated as one single unit and checked for bugs. This also checks if all the modules are working properly with each other.

System testing: This testing checks for operating system compatibility. It includes both functional and non functional requirements.

Sanity testing: It ensures change in the code doesn't affect the working of the project.

Smoke testing: this type of testing is a set of small tests designed for each build.
Interface testing: Testing of the interface and its proper functioning.

Regression testing: Testing the software repetitively when a new requirement is added, when bug fixed etc.

Beta/Acceptance testing: User level testing to obtain user feedback on the product.

8.2.2 Non-Functional Testing:

This type of testing is mainly concerned with the non-functional requirements such as performance of the system under various scenarios. Performance testing: Checks for speed, stability and reliability of the software, hardware or even the network of the system under test.

Compatibility testing: This type of testing checks for compatibility of the system with different operating systems, different networks etc. Localization testing: This checks for the localized version of the product mainly concerned with UI.

Security testing: Checks if the software has vulnerabilities and if any, fix them.

Reliability testing: Checks for the reliability of the software

Stress testing: This testing checks the performance of the system when it is exposed to different stress levels.

Usability testing: Type of testing checks the easily the software is being used by the customers.

Compliance testing: Type of testing to determine the compliance of a system with internal or external standards

Reliability

The structure must be reliable and strong in giving the functionalities. The movements must be made unmistakable by the structure when a customer has revealed a couple of enhancements. The progressions made by the Programmer must be Project pioneer and in addition the Test designer.

• **Maintainability**

The system watching and upkeep should be fundamental and focus in its approach. There should not be an excess of occupations running on diverse machines such that it gets hard to screen whether the employments are running without lapses.

• **Performance**

The framework will be utilized by numerous representatives all the while. Since the system will be encouraged on a single web server with a lone database server outside of anyone's ability to see, execution transforms into a significant concern. The structure should not capitulate when various customers would use everything the while. It should allow brisk accessibility to each and every piece of its customers.

• **Portability**

The framework should to be effectively versatile to another framework. This is obliged when the web server, which s facilitating the framework gets adhered because of a few issues, which requires the framework to be taken to another

framework.

• **Scalability**

The framework should be sufficiently adaptable to include new functionalities at a later stage. There should be a run of the mill channel, which can oblige the new functionalities.

• **Flexibility**

Flexibility is the capacity of a framework to adjust to changing situations and circumstances, and to adapt to changes to business approaches and rules. An adaptable framework is one that is anything but difficult to reconfigure.

8.3 White Box Testing

White Box Testing is defined as the testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing. It is usually performed by developers.

It is one of two parts of the "Box Testing" approach to software testing. Its counterpart, Blackbox testing, involves testing from an external or enduser type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing.

The term "WhiteBox" was used because of the see-through box concept. The clear box or WhiteBox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "Black Box Testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.

8.4 Different Stages of Testing

8.4.1 Unit Testing

A level of software testing known as "unit testing" involves testing specific programme units or components. The goal is to confirm that each piece of software operates as intended. The smallest testable component of any software is called a unit. It typically has one or more inputs and one output. An individual programme, function, process, etc. can all be considered units in procedural programming. The smallest unit in object-oriented programming is a method, which can be a part of a base/super class, abstract class, or derived/child class. (Some treat an application module as a unit. This should be avoided because that module presumably contains a lot of different units.) Frameworks for unit testing, drivers, stubs, and mock/fake objects

Benefits

Conviction in updating/maintaining code is increased by unit testing. We will be able to quickly identify any flaws introduced by the update if effective unit tests are written and run each time any code is modified. Additionally, the unexpected consequences of modifications to any code are reduced if the interdependencies between the codes have previously been reduced to enable unit testing. Codes can be reused more often. Codes must be modular in order to support unit testing. Code reuse is facilitated by this. Growth happens more quickly. How? If unit testing is not in place, you create your code and run a haphazard "developer test" (you set some breakpoints, launch the GUI, and supply a few inputs that should hopefully trigger your function).

8.4.2 Integration Testing

INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing

Tasks
Integration Test Plan

- o Prepare
- o Review
- o Rework
- o Baseline Integration Test Cases/Scripts
- o Prepare
- o Review
- o Rework
- o Baseline Integration Test

8.4.3 System Testing

SYSTEM TESTING is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

system testing: The process of testing an integrated system to verify that it meets specified requirements

8.4.4 Acceptance Testing

ACCEPTANCE TESTING is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

TESTING HEALTH CARE DATA:

Testing is a method used to determine whether generated computer software is accurate, complete, secure, and of high quality. Testing is the process of conducting a technical examination, which includes running a software or application with the goal of identifying flaws. Based on the test samples used for training, our model develops the ability to link a certain input (i.e., set of characteristics) to the associated output (i.e., tag). The machine learning algorithm receives input features and tags (such as 1-normal, 2-healthy, and illness), which are used to create a model. To accurately categorise and predict Health disease cases with the fewest possible variables, a comparison analysis of various classifiers was conducted for the classification of the Health dataset.

Input	**Expected Output**	**Actual Output**
Data Visualization	Various visual representations of the data to understand more about the relationship between various features.	Pass
Data Processing	Convert some categorical variables into dummy variables and scale all the values before training the Machine Learning models.	Pass
Dataset	Split the dataset into training and testing datasets.	Pass
Training dataset	Train the model using the training dataset.	Pass
Testing dataset	Tests if the model is accurate based on the output of the testing dataset.	Pass

Training and Subsequent testing

Input	Expected Output	Actual Output
No Health cate data failure	Should be labeled as 1 (no health care data failure) and should show output as “The patient is not likely to have health disease”.	Pass
Health care data analysis failure	Should be labeled as 2 (health care data failure) and should show output as “The patient is likely to have health care data failure”.	Pass

8.5 Model Evaluation

The most important evaluation metrics for this problem domain are Accuracy, Sensitivity, Specificity, Precision, F1-measure, Log Loss, ROC and Mathew correlation coefficient.

• Accuracy: which refers to how close a measurement is to the true value and can be calculated using the following formula:

• Precision: which is how consistent results are when measurements are repeated and can be calculated using the following formula: Precision = True Positive / (True Positive + False Positive)

• Sensitivity: Sensitivity is a measure of the proportion of actual positive cases that gotpredicted as positive (or true positive). Sensitivity is also termed as Recall. Sensitivity = True Positive / (True Positive + False Negative)

• Specificity: Specificity is defined as the proportion of actual negatives, which got predicted asthe negative (or true negative). Specificity = True Negative / (True Negative + False Positive) • Mathew Correlation coefficient
(MCC): The Matthews correlation coefficient (MCC), instead, is a more reliable

statisticalrate which produces a high score only if the prediction obtained good results in all ofthe four confusion matrix categories (true positives, false negatives, true negatives, and false positives), proportionally both to the size of positive elements and the size of negative elements in the dataset.

• Logic loss Logarithmic loss measures the performance of a classification model where the prediction input is a probability value between 0 and 1. The goal of our machine learning models is to minimize this value. A perfect model would have a log loss of 0. Log loss increases as the predicted probability diverges from the actual label. So, predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high log loss.

• F1 Score F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 score is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost.

F1 Score = 2(Recall Precision) / (Recall + Precision)

• **ROC Curve**

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate & False Positive Rate.

8.5.1 Random Forest Classifier

Code:

```
y_pred_rfe = rf_classifier.predict(X_test)


plt.figure(figsize=(10, 8))
```

```
 CM=confusion_matrix(Y_test,y_pred_rfe)

sns.heatmap(CM, annot=True)

TN = CM[0][0] FN = CM[1][0] TP = CM[1][1]

FP = CM[0][1] specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_rfe)

acc= accuracy_score(Y_test, y_pred_rfe)

 roc=roc_auc_score(Y_test, y_pred_rfe)

prec = precision_score(Y_test, y_pred_rfe)

rec = recall_score(Y_test, y_pred_rfe)

f1 = f1_score(Y_test, y_pred_rfe)

 mathew = matthews_corrcoef(Y_test, y_pred_rfe)

model_results =pd.DataFrame([['Random Forest',acc, prec,rec,specificity, f1,roc,
loss_log,mathew]], columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])

model_results
```

```
Y_pred_rf = np.around(Y_pred_rf)

print(metrics.classification_report(Y_test,Y_pred_rf))


plot_roc_curve(rf_classifier,X_test,Y_test)

plt.xlabel('False Positive Rate') plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic Curve')
```

8.5.2 K-Nearest Neighbors Classifier

```
y_pred_knne = knn_classifier.predict(X_test)

plt.figure(figsize=(10, 8))

CM=confusion_matrix(Y_test,y_pred_knne) sns.heatmap(CM, annot=True)  TN = CM[0][0]

FN = CM[1][0]

TP = CM[1][1]

FP = CM[0][1]

specificity = TN/(TN+FP)

loss_log = log_loss(Y_test, y_pred_knne)

acc= accuracy_score(Y_test, y_pred_knne)
roc=roc_auc_score(Y_test, y_pred_knne)

prec = precision_score(Y_test, y_pred_knne)

 rec = recall_score(Y_test, y_pred_knne)

f1 = f1_score(Y_test, y_pred_knne)

mathew = matthews_corrcoef(Y_test, y_pred_knne)

model_results =pd.DataFrame([['K-Nearest Neighbors ',acc, prec,rec,specificity, f1,roc,
```

```
loss_log,mathew]], columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])

model_results


Y_pred_knn = np.around(Y_pred_knn)

print(metrics.classification_report(Y_test,Y_pred_knn))

plot_roc_curve(knn_classifier,X_test,Y_test)

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic Curve')
```

8.5.3 Decision Tree Classifier

```
 y_pred_dte = dt_classifier.predict(X_test)

plt.figure(figsize=(10, 8))

CM=confusion_matrix(Y_test,y_pred_dte)

sns.heatmap(CM, annot=True)

TN = CM[0][0] FN = CM[1][0] TP = CM[1][1]

FP = CM[0][1]

specificity = TN/(TN+FP)

 loss_log = log_loss(Y_test, y_pred_dte)

acc= accuracy_score(Y_test, y_pred_dte)

 roc=roc_auc_score(Y_test, y_pred_dte)

 prec = precision_score(Y_test, y_pred_dte)
rec = recall_score(Y_test, y_pred_dte)

f1 = f1_score(Y_test, y_pred_dte)

mathew = matthews_corrcoef(Y_test, y_pred_dte)
```

```
model_results =pd.DataFrame([['Decision Tree',acc, prec,rec,specificity, f1,roc,  loss_log,mathew]],
columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])

model_results


Y_pred_dt = np.around(Y_pred_dt)

print(metrics.classification_report(Y_test,Y_pred_dt))

plot_roc_curve(dt_classifier,X_test,Y_test)

plt.xlabel('False Positive Rate')

 plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic Curve')
```

8.5.4 Naive Bayes Classifier

```
y_pred_nbe = nb_classifier.predict(X_test)

plt.figure(figsize=(10, 8))

CM=confusion_matrix(Y_test,y_pred_nbe)

sns.heatmap(CM, annot=True)

TN = CM[0][0] FN = CM[1][0]

TP = CM[1][1] FP = CM[0][1]

specificity = TN/(TN+FP) l

oss_log = log_loss(Y_test, y_pred_nbe)

acc= accuracy_score(Y_test, y_pred_nbe) roc=roc_auc_score(Y_test, y_pred_nbe) prec =

precision_score(Y_test, y_pred_nbe)

 rec = recall_score(Y_test, y_pred_nbe)
```

```
f1 = f1_score(Y_test, y_pred_nbe)

 mathew = matthews_corrcoef(Y_test, y_pred_nbe)

model_results =pd.DataFrame([['Naive Bayes ',acc, prec,rec,specificity, f1,roc, loss_log,mathew]],
columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])

model_results


Y_pred_nb = np.around(Y_pred_nb)

print(metrics.classification_report(Y_test,Y_pred_nb))


plot_roc_curve(nb_classifier,X_test,Y_test)

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic Curve')
```

CHAPTER 9
RESULTS

9.1 Performance metrics:

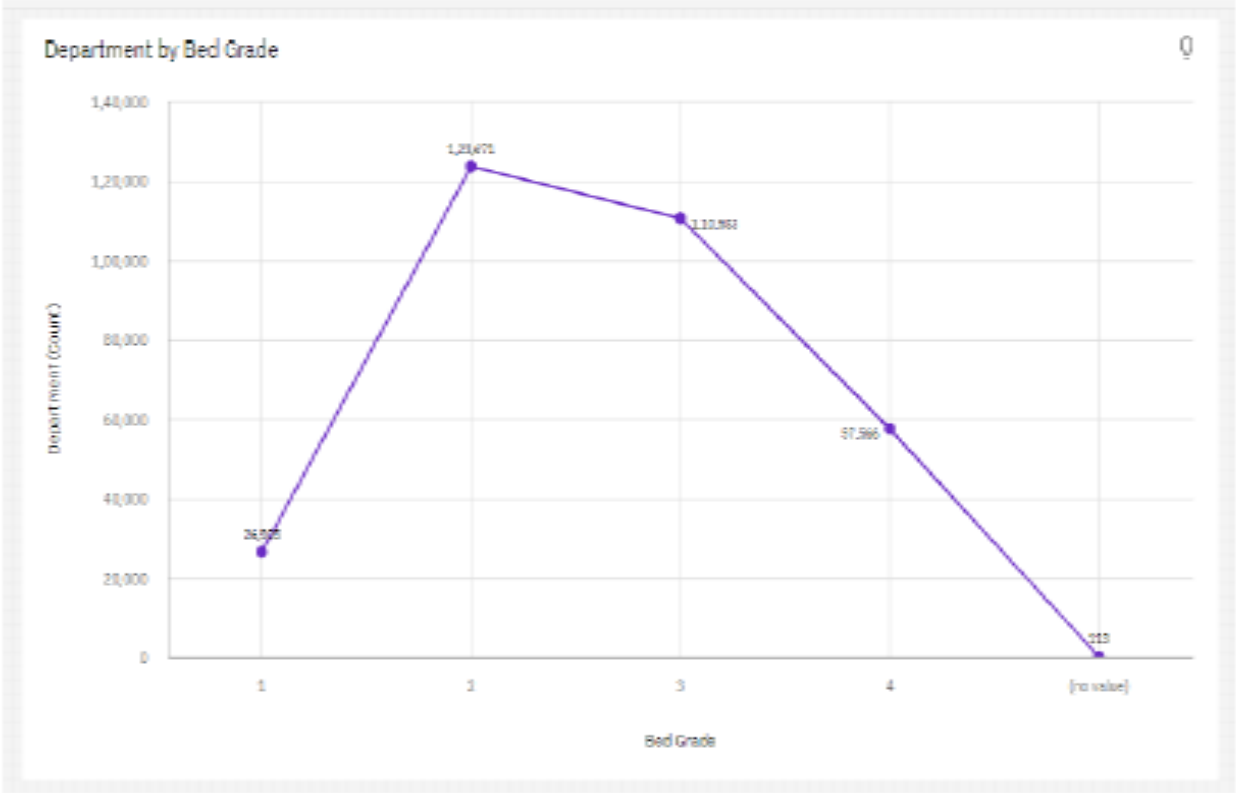
Performance Metrics shows the level of the work that we have done to the datasets.

STARTING :

DashBoard-1:

ENDING :

DashBoard-11:



CHAPTER 10
ADVANTAGES AND DISADVANTAGES

1. Easy Access To Patient Data

A well-designed hospital information system makes patient data easily accessible to healthcare professionals. All the necessary information on a patient from multiple hospital departments can be accessed on the screen with just a few clicks. By logging into the HIS, the treating physician will have immediate access to the patient's test results, allowing her to make timely treatment decisions without having to search for the IPD file.

2. Cost Effective

When HIS is properly implemented, it eliminates a lot of the manual tasks that are typically performed in hospitals, especially where paperwork and record keeping are necessary. Because a lot of labour is automated and does not require manual involvement to store or analyse the information, it aids in the reduction of manpower. Additionally, it significantly reduces storage costs.

3. Improved Efficiency

When processes are automated using software, they are handled mechanically without human intervention, which immediately ensures increased efficiency. The software won't experience human issues like exhaustion, misunderstandings, or lack of concentration; instead, it will consistently complete every task given to it with accuracy.

4. Reduces Scope of Error

The scope of mistake is drastically decreased on HIS since procedures are automated and a lot of duties are given to the software to complete with the highest level of accuracy and with the least amount of human involvement. When paying an IDP patient for pharmaceuticals used with HIS, for example,

there is almost little chance that the bill would be incorrect because the drug that the nurse indents is the drug that is billed—unless there is a stock shortage or a change in the prescription order after the indent has been issued. As part of automated regular operating procedure, the software saves the per unit rate of the medicine. The software will precisely calculate the amount due once the medicine name and quantity are entered.

5. Increased Data Security & Retrieve-ability

Hospitals are required to preserve records, which is a pain with two challenges: keeping the data secure with only authorised people having access to it and retrieving it as quickly as feasible. The enduring issues of space constraint, protection from the elements, protection from pest damage, etc., should be added.

The best answer to these issues is HIS. On the server or in the cloud, all the data is safely kept. Since HIS relies on logins, data security is no longer an issue because it allows access to data based on the user's role, such as receptionist, doctor, nurse, or radiologist. Data that is kept on a server or in the cloud can be retrieved with only a few clicks and is then shown on the screen shortly after.

6. Improved Patient Care

Improved access to patient data and improved work efficiency means better and faster clinical decisions. In this age of evidence based medicine, the faster the clinician gets the diagnostic reports and the quicker her orders are implemented the faster is the patient recovery and the better it is on the patient care index. With automation, all departments in the hospitals are inter connected and the faster information access further improves the quality of patient care and the resultant bed turnover in the hospital.

DISADVANTAGES

1.Lack of Empathy in Patient and Doctor Interaction

Even when they are not physically present to one another, technology can help keep healthcare personnel and patients linked. Instead of relying on sporadic consultations, it is conceivable, for instance, to create and update a treatment plan on an ongoing basis by utilising data and technology.As we proceed through the COVID-19 pandemic, clinicians are using telehealth more and more as a critical tool. In these hard times, the use of such instruments has maintained the healthcare system and ensured that patients receive continuity of treatment.

2.Frustration with poor implementation

Healthcare professionals that spend more time struggling with technology as opposed to patient care, are likely to disregard the use of technology and future iterations.But even for healthcare professionals that are in favor of using and implementing technology, ensuring that the technology-assisted outcomes are more accurate, or better at diagnosing is crucial.

3.Cybersecurity risks

Risk related to breach of protected health information.Risk related to alteration of data that may be used to make wrong healthcare decisions.Risk related to alteration of device functionality that results in adverse events

CHAPTER 11
CONCLUSION

Data analytics has the potential to transform the way healthcare providers use sophisticated technologies to gain insight from their clinical and other data repositories and make informed decisions. In the future we'll see the rapid, widespread implementation and use of big data analytics across the healthcare organization and the healthcare industry.

To that end, the several challenges highlighted above, must be addressed. As big data analytics becomes more mainstream, issues such as guaranteeing privacy, safeguarding security, establishing standards and governance, and continually improving the tools and technologies will garner attention. Big data analytics and applications in healthcare are at a nascent stage of development, but rapid advances in platforms and tools can accelerate their maturing process.

In addition, while most platforms currently available are open source, the typical advantages and limitations of open source platforms apply. To succeed, big data analytics in healthcare needs to be packaged so it is menu-driven, user-friendly and transparent. Real-time big data analytics is a key requirement in healthcare.

The lag between data collection and processing has to be addressed. The dynamic availability of numerous analytics algorithms, models and methods in a pull-down type of menu is also necessary for large-scale adoption. The important managerial issues of ownership, governance and standards have to be considered. And woven through these issues are those of continuous data acquisition and data cleansing.

CHAPTER 12
FUTURE SCOPE

1. Improved Decision Making:

Data Analytics eliminates guesswork and manual tasks. Be it choosing the right content, planning marketing campaigns, or developing products. Organizations can use the insights they gain from data analytics to make informed decisions.

2. Better Customer Service:

Data analytics allows you to tailor customer service according to their needs. It also provides personalization and builds stronger relationships with customers. Analyzed data can reveal information about customers' interests, concerns, and more.

3. Efficient Operations:

With the help of data analytics, you can streamline your processes, save money, and boost production. With an improved understanding of what your audience wants, you spend lesser time creating ads and content that aren't in line with your audience's interests.

4. Effective Marketing:

Data analytics gives you valuable insights into how your campaigns are performing. This helps in fine-tuning them for optimal outcomes. Additionally, you can also find potential customers.

APPENDIX

Source code:


```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 1,
   "id": "58826ff6",
   "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": 2,
   "id": "edddd370",
   "metadata": {},
   "outputs": [],
   "source": [
    "#load data\n",
    "d1 = pd.read_csv('C:/Users/vishnu/OneDrive/Desktop/Data Sets/Healthcare_Data/sample_sub.csv')\n",
    "d2 = pd.read_csv('C:/Users/vishnu/OneDrive/Desktop/Data Sets/Healthcare_Data/train_data_dictionary.csv')\n",
```

```
          "test = pd.read_csv('C:/Users/vishnu/OneDrive/Desktop/Data Sets/Healthcare_Data/test_data.csv')\n",
          "train = pd.read_csv('C:/Users/vishnu/OneDrive/Desktop/Data Sets/Healthcare_Data/train_data.csv')"
         ]
        },
        {
         "cell_type": "code",
         "execution_count": 3,
         "id": "e1b9b524",
         "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": 3,
        "metadata": {},
        "output_type": "execute_result"
       }
      ],
      "source": [
       "train.head()"
      ]
     },
     {
      "cell_type": "code",
      "execution_count": 4,
      "id": "2d62e39e",
```

```
"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"
         ]
        },
        {
         "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": 4,
         "metadata": {},
         "output_type": "execute_result"
        }
       ],
       "source": [
        "train.info()\n",
        "train.Stay.unique()"
       ]
      },
      {
       "cell_type": "code",
       "execution_count": 5,
       "id": "1fa41bd3",
       "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": 5,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "# NA values in train dataset :\n",
 "train.isnull().sum().sort_values(ascending = False)"
]
},
```

```
{
 "cell_type": "code",
 "execution_count": 6,
 "id": "41d106b9",
 "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": 6,
   "metadata": {},
```

```
  "output_type": "execute_result"
 }
],
"source": [
 "# NA values in test dataset :\n",
 "test.isnull().sum().sort_values(ascending = False)"
]
},
{
"cell_type": "code",
"execution_count": 7,
"id": "be8ba886",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "(318438, 18)"
   ]
  },
  "execution_count": 7,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "# Dimension of train dataset\n",
 "train.shape"
]
},
{
```

```
"cell_type": "code",
"execution_count": 8,
"id": "090485c7",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "(137057, 17)"
   ]
  },
  "execution_count": 8,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "# Dimension of test dataset\n",
 "test.shape"
]
},
{
"cell_type": "code",
"execution_count": 9,
"id": "e51bda8e",
"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": [
 "# Number of distinct observations in train dataset \n",
 "for i in train.columns:\n",
 "    print(i, ':', train[i].nunique())"
]
},
{
"cell_type": "code",
"execution_count": 10,
"id": "0044c5e6",
```

```
        "metadata": {},
        "outputs": [],
        "source": [
         "#Replacing NA values in Bed Grade Column for both Train and
Test datssets\n",
         "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": 11,
        "id": "c63adfbf",
        "metadata": {},
        "outputs": [],
        "source": [
         "#Replacing NA values in  Column for both Train and Test
datssets\n",

"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": 12,
```

```
 "id": "2a9edfb1",
 "metadata": {},
 "outputs": [],
 "source": [
  "# Label Encoding Stay column in train dataset\n",
  "from sklearn.preprocessing import LabelEncoder\n",
  "le = LabelEncoder()\n",
  "train['Stay'] = le.fit_transform(train['Stay'].astype('str'))"
 ]
},
{
 "cell_type": "code",
 "execution_count": 13,
 "id": "3599a357",
 "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</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>4</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>3</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>4</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>4</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  \n",
          "1          5954.0    4  \n",
          "2          4745.0    3  \n",
          "3          7272.0    4  \n",
          "4          5558.0    4  "
         ]
        },
        "execution_count": 13,
        "metadata": {},
        "output_type": "execute_result"
       }
      ],
      "source": [
       "train.head()"
      ]
     },
```

```
{
 "cell_type": "code",
 "execution_count": 14,
 "id": "42d5dad1",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "(455495, 18)"
    ]
   },
   "execution_count": 14,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "#Imputing dummy Stay column in test datset to concatenate with train dataset\n",
  "test['Stay'] = -1\n",
  "df = pd.concat([train, test])\n",
  "df.shape"
 ]
},
{
 "cell_type": "code",
 "execution_count": 15,
 "id": "dca36f50",
 "metadata": {},
 "outputs": [],
```

```
"source": [
 "#Label Encoding all the columns in Train and test datasets\n",
 "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": 16,
"id": "d7ba36ee",
"metadata": {},
"outputs": [],
"source": [
 "#Spearating Train and Test Datasets\n",
 "train = df[df['Stay']!=-1]\n",
 "test = df[df['Stay']==-1]"
]
},
{
"cell_type": "code",
"execution_count": 17,
"id": "f2b44162",
"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": 18,
          "id": "5ff32237",
          "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": 19,
 "id": "47f16ffb",
 "metadata": {},
 "outputs": [],
 "source": [
  "# Droping duplicate columns\n",
  "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": 20,
 "id": "d707cb55",
 "metadata": {},
 "outputs": [],
 "source": [
  "# Splitting train data for Naive Bayes and XGBoost\n",
  "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": 21,
 "id": "cf5157c0",
 "metadata": {},
 "outputs": [],
 "source": [
  "from sklearn.naive_bayes import GaussianNB\n",
  "target = y_train.values\n",
  "features = X_train.values\n",
  "classifier_nb = GaussianNB()\n",
  "model_nb = classifier_nb.fit(features, target)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 22,
 "id": "1de566dd",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "Acurracy: 34.55439015199096\n"
   ]
  }
 ],
 "source": [
  "prediction_nb = model_nb.predict(X_test)\n",
  "from sklearn.metrics import accuracy_score\n",
  "acc_score_nb = accuracy_score(prediction_nb,y_test)\n",
  "print(\"Acurracy:\", acc_score_nb*100)"
```

```
        ]
       },
       {
        "cell_type": "code",
        "execution_count": 23,
        "id": "4546ebc5",
        "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"
         ]
        }
       ],
       "source": [
        "# Segregation of features and target variable\n",
        "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)"
       ]
      },
      {
       "cell_type": "code",
       "execution_count": 24,
       "id": "0b9ddd20",
       "metadata": {},
       "outputs": [
        {
         "data": {
          "text/plain": [
           "(318438, 20)"
          ]
         },
         "execution_count": 24,
         "metadata": {},
```

```
   "output_type": "execute_result"
   }
  ],
  "source": [
   "# Data Scaling\n",
   "from sklearn import preprocessing\n",
   "X_scale = preprocessing.scale(X)\n",
   "X_scale.shape"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 25,
  "id": "3d9fedcd",
  "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": 31,
  "id": "af56e1ca",
  "metadata": {},
  "outputs": [],
  "source": [
   "# Naive Bayes\n",
   "pred_nb = classifier_nb.predict(test1.iloc[:,1:])\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": 32,
"id": "1b4e8e33",
"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>Stay</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>318439</td>\n",
"      <td>21-30</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>318440</td>\n",
"      <td>51-60</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>318441</td>\n",
"      <td>21-30</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>318442</td>\n",
"      <td>21-30</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>318443</td>\n",
"      <td>31-40</td>\n",
"    </tr>\n",
"  </tbody>\n",
```

```
"</table>\n",
"</div>"
],
"text/plain": [
"   case_id   Stay\n",
"0   318439  21-30\n",
"1   318440  51-60\n",
"2   318441  21-30\n",
"3   318442  21-30\n",
"4   318443  31-40"
]
},
"execution_count": 32,
"metadata": {},
"output_type": "execute_result"
}
],
"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": 33,
"id": "ce6673de",
"metadata": {},
"outputs": [
{
```

```
  "data": {
   "text/plain": [
    "(137057, 20)"
   ]
  },
  "execution_count": 33,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "# Neural Network\n",
 "test_scale = preprocessing.scale(z)\n",
 "test_scale.shape"
]
},
{
"cell_type": "code",
"execution_count": 34,
"id": "c6c5fbc5",
"metadata": {},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "Stay\n",
   "0-10                2598\n",
   "11-20               26827\n",
   "21-30               72206\n",
   "31-40               15639\n",
```

```
    "41-50                469\n",
    "51-60              13651\n",
    "61-70                 92\n",
    "71-80                955\n",
    "81-90                296\n",
    "91-100                 2\n",
    "More than 100 Days    4322\n",
    "Name: case_id, dtype: int64\n"
   ]
  }
 ],
 "source": [
  "# Naive Bayes\n",
  "print(result_nb.groupby('Stay')['case_id'].nunique())"
 ]
}
],
"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
}
```


Github link:-

https://github.com/IBM-EPBL/IBM-Project-27906-1660099668