Team id: PNT2022TMID48593

Project Report
Analytics for Hospitals' Health-Care Data

1. Introduction

1.1 Project overview:

Healthcare organizations are under increasing pressure to improve patient care outcomes and achieve better care. While this situation represents a challenge, it also offers organizations an opportunity to dramatically improve the quality of care by leveraging more value and insights from their data. Health care analytics refers to the analysis of data using quantitative and qualitative techniques to explore trends and patterns in the acquired data. While healthcare management uses various metrics for performance, a patient's length of stay is an important one.

Being able to predict the length of stay (LOS) allows hospitals to optimize their treatment plans to reduce LOS, to reduce infection rates among patients, staff, and visitors.

1.2. Purpose

The goal of this project is to accurately predict the Length of Stay for each patient so that the hospitals can optimize resources and function better.

2. Literature survey

2.1 Existing problem

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

2.2. References

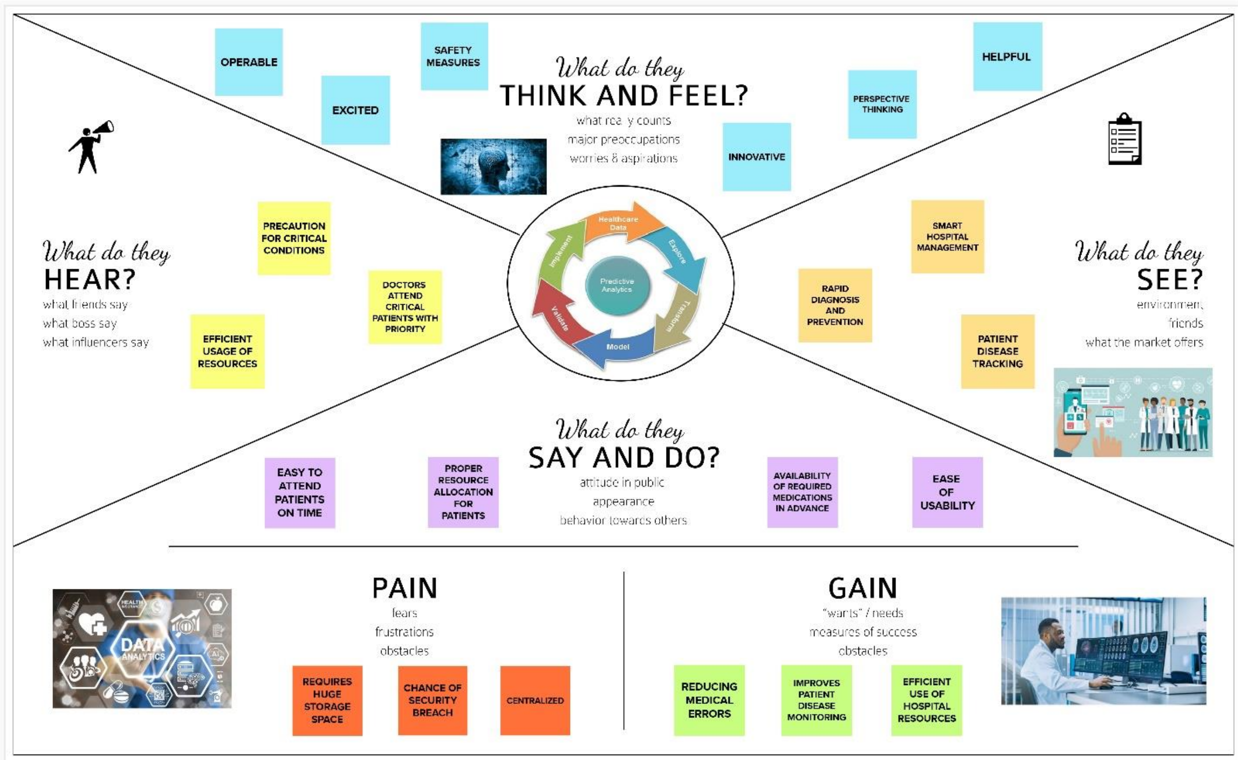
- Janatahack: Healthcare Analytics II - Analytics Vidhya - Link
- What Is Naive Bayes Algorithm in Machine Learning? - Rohit Dwivedi - Link
- Naive Bayes for Machine Learning – From Zero to Hero - Anand Venkataraman - Link
- XGBoost Parameters - XGBoost Documentation - Link
- Predicting Heart Failure Using Machine Learning, Part 2- Andrew A Borkowski - Link
- How to Tune the Number and Size of Decision Trees with XGBoost in Python-JasonBrownlee - Link
- Big Data Analytics in Healthcare That Can Save People - Sandra Durcevic - Link
- Learning Process of a Neural Network – Jordi Torres – Link

2.3. Problem statement

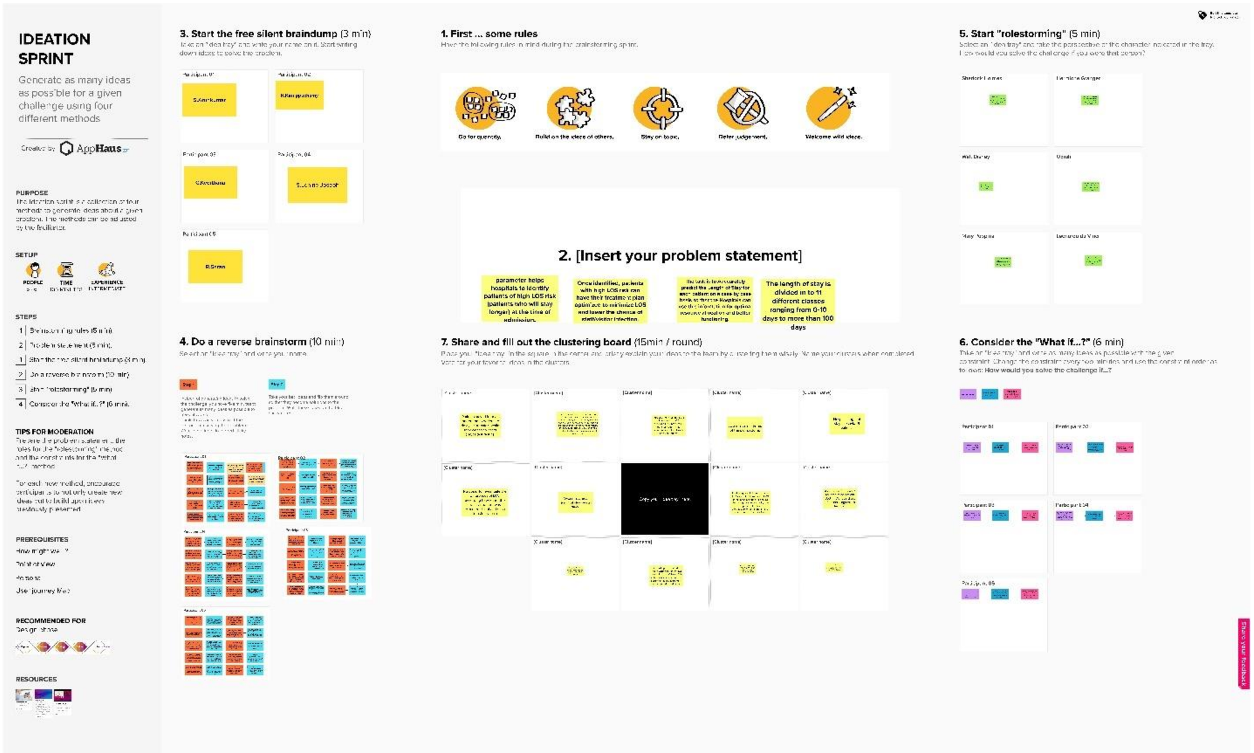
The task is to accurately predict the Length of Stay for each patient on case-by-case basis so that the Hospitals can use this information for optimal resource allocation and better functioning. The length of stay is divided into 11 different classes ranging from 0-10 days to more than 100 days.

3. Ideation & proposed solution

3.1 Empathy map Canvas



3.2 Brainstorming

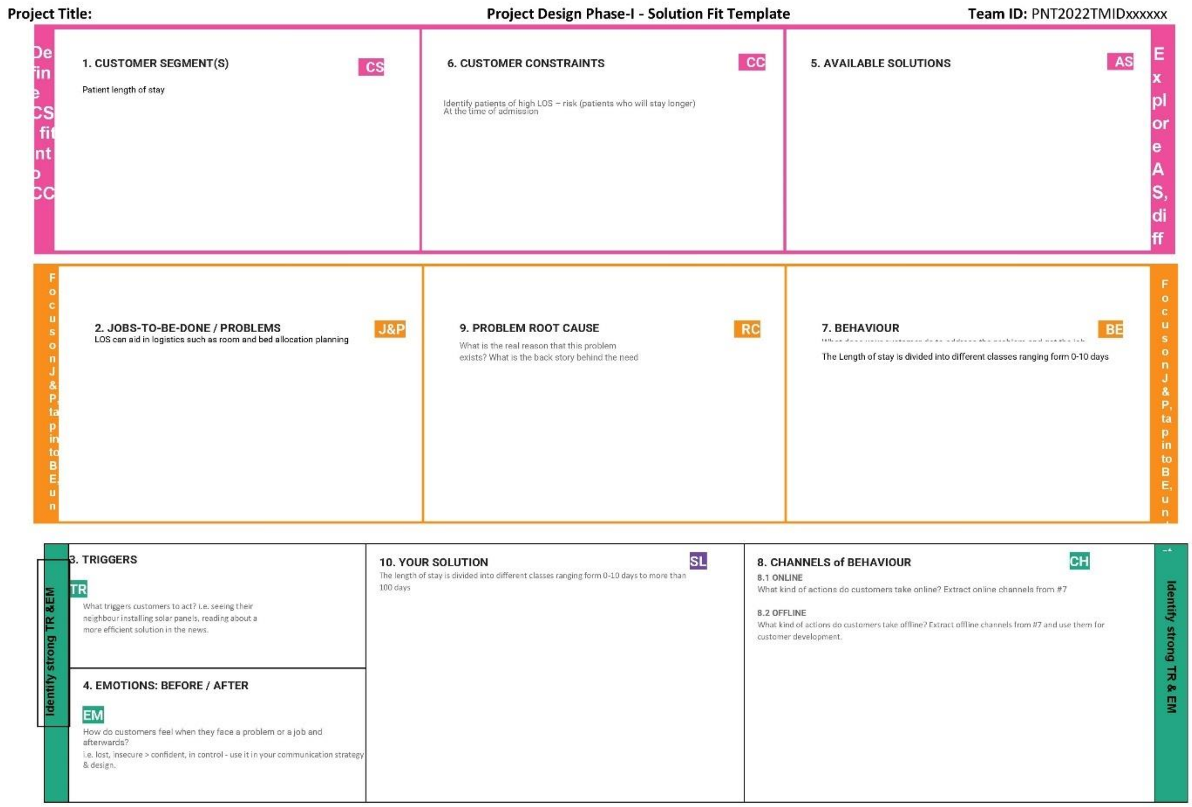

IDEATION SPRINT
Generate as many ideas as possible for a given challenge using four different methods
Created by AppHaus
PURPOSE
SETUP
PEOPLE
TIME
EXPERIENCE
STEPS
TIPS FOR MODERATION
For each new method, encourage participants to not only create new ideas but to build upon ideas previously presented
PREREQUISITES
RECOMMENDED FOR
RESOURCES
3. Start the free silent braindump (3 min)
Participant 01
Participant 02
Participant 03
Participant 04
Participant 05
4. Do a reverse brainstorm (10 min)
1. First ... some rules
Have the following rules in mind during the brainstorming sprint.
Go for quantity.
Build on the ideas of others.
Stay on topic.
Defer judgement.
Welcome wild ideas.
2. [Insert your problem statement]
parameter helps hospitals to identify patients of high LOS risk (patients who will stay longer) at the time of admission.
The length of stay is divided in to 11 different classes ranging from 0-10 days to more than 100 days
7. Share and fill out the clustering board (15min / round)
5. Start "rolestorming" (5 min)
Sherlock Holmes
Walt Disney
Oprah
6. Consider the "What if...?" (6 min)
How would you solve the challenge if...?

3.3 Proposed solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	parameter helps hospitals to identify patients of high LOS risk (patients who will stay longer) at the time of admission. Once identified, patients with high LOS risk can have their treatment plan optimized to minimize LOS and lower the chance of staff/visitor infection. The task is to Accurately predict the Length of Stay for each patient on a case by case basis so that the Hospitals can use this information for optimal resource allocation and better functioning. The length of stay is divided into 11 different classes ranging from 0-10 days to more than 100 days.
2.	Idea / Solution description	Reduce patient Length of hospital stay: Implement Process Changes.A Critical part of improving LOS is using data to understand and improve processes that directly affect a patients LOS. Remove Discharge Barriers. Improve Care Transitions

3.	Novelty / Uniqueness	Understanding of the factors associated with LOS of the COVID-19 patients may help the care providers and the patients to better anticipate the LOS,optimize the resources and processes,and prevent protracted stays.
4.	Social Impact / Customer Satisfaction	Satisfaction can be improved through variables such as reliability,empathy and responsiveness,and the loyalty of patient.
5.	Business Model (Revenue Model)	(I)It can be collaborated with diagnosis centers and hospitals. (ii)It can be collaborated with government for health awareness camps.
6.	Scalability of the Solution	Optimal resources utilization. Predicting hospital length of stay(LOS) for patients with COVID-19 infection is essential to ensure that adequate bed capacity can be provided without unnecessarily restricting care for patients with other conditions.

3.4 Problem solution fit



Project Title:

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMIDxxxxxx

1. CUSTOMER SEGMENT(S) CS

Patient length of stay

6. CUSTOMER CONSTRAINTS CC

Identify patients of high LOS – risk (patients who will stay longer)
At the time of admission

5. AVAILABLE SOLUTIONS AS

2. JOBS-TO-BE-DONE / PROBLEMS J&P

LOS can aid in logistics such as room and bed allocation planning

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need

7. BEHAVIOUR BE

The Length of stay is divided into different classes ranging form 0-10 days

Identify strong TR &EM

3. TRIGGERS

TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

4. EMOTIONS: BEFORE / AFTER

EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

10. YOUR SOLUTION SL

The length of stay is divided into different classes ranging form 0-10 days to more than 100 days

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Identify strong TR & EM

4. Requirements analysis

4.1 Functional requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via Message
FR-3	Interoperability	Dashboard helps to share the patient’s information interoperable to the hospitals in timely manner.
FR-4	Accuracy	Dashboard helps predict the patient’s Health risks accurately based on LOS (Length of Stay).
FR-5	Compliance	The compliance of a dashboard is like to use very interactively in real time by the hospitals.
FR-6	Concise	These dashboards are clear, intuitive, and customizable and interactive in manner.

1. **Nonfunctional requirements**

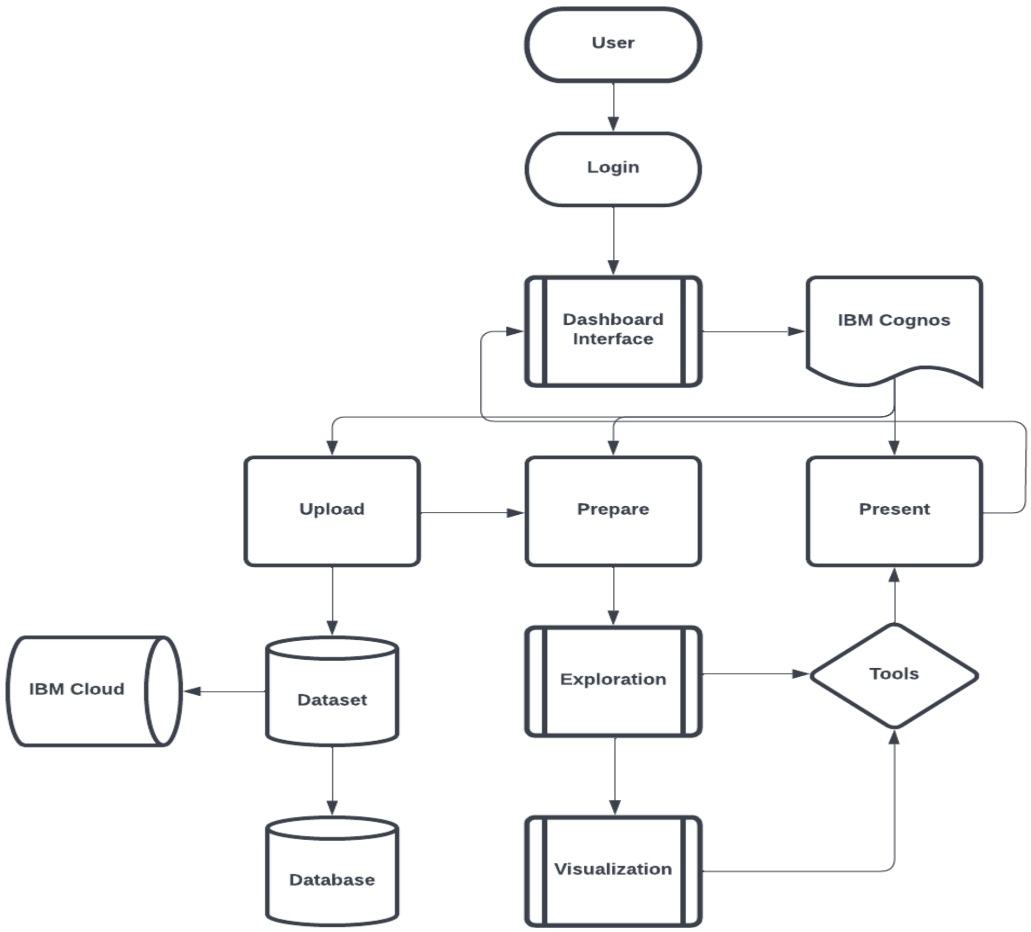
FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	This Dashboards are designed to offer a comprehensive overview of patient's LOS, and do so through the use of data visualization tools like charts and graphs.
NFR-2	**Security**	The Dashboard helps to indicate the current threat level to the Hospitals; an indication of events and incidents that have occurred; a record of authentication errors; unauthorized access
NFR-3	**Reliability**	This dashboard will be consistent and reliable to the users and helps the user to use in effective, efficient and reliable manner.
NFR-4	**Performance**	The dashboard reduces the time needed for analysing data and has an automated system for that which improves the performance

NFR-5	**Availability**	The dashboard can available to meet user's demand in timely manner and it is also helps to provide necessary information to the user's dataset
NFR-6	**Scalability**	It is a multi-tenant system which is capable of rimming on lower-level systems as well.

4. PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.


User
Login
Dashboard Interface
IBM Cognos
Upload
Prepare
Present
IBM Cloud
Dataset
Exploration
Tools
Database
Visualization

5.2 Solution & Technical Architecture

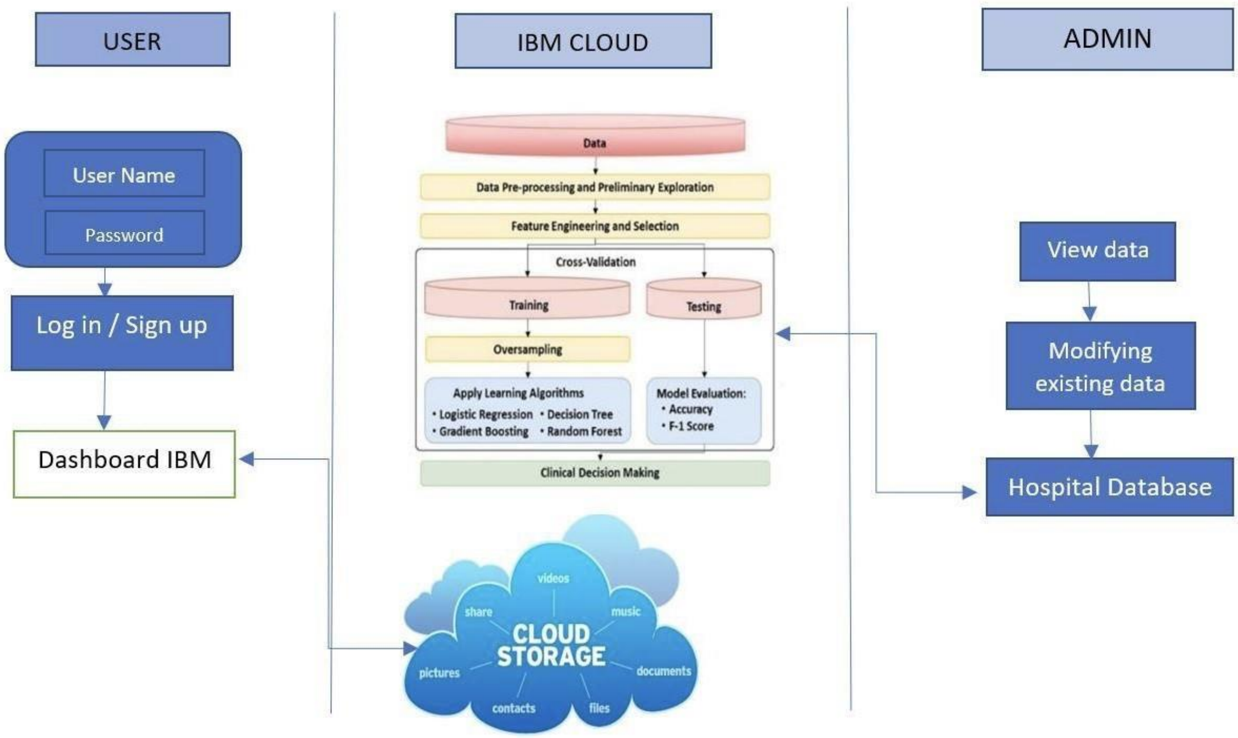


Table1: Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g., Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc…
2.	Application Logic-1	Logging in as a patient / user in the application	Python
3.	Application Logic-2	Logging in as an admin in the application	IBM Watson Assistant

5.	Database	All the data about patients such as disease, address and etc.	MySQL, NoSQL, etc.
6.	Cloud Database	IBM Watson cloud is used for storage, Cloud	IBM DB2, IBM Cloud ant etc.
7.	External API-1	Purpose of External API used in the application	Aadhar API, etc.
8.	Machine Learning Model	Purpose of Machine Learning Model	Regression Model, etc.
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration, Cloud Server Configuration	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S. No	**Characteristics**	**Description**	**Technology**
1.	Open-Source Frameworks	List the open-source frameworks used	Python
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Encryption.

3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Can supports higher workloads
4.	Availability	Justify the availability of application (e.g.	Highly available
		use of load balancers, distributed servers etc.)	
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	It performs good uses various tools and ideas in a scientific manner to meet the desired outcomes

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release

Customer	Dashboard	USN 1	As a user, I can upload the datasets to the dashboard	I can access various operations	Medium	Sprint-4
	View	USN 2	As a user, I can view the	I can view the visual data	Medium	Sprint-3
			patient details	and the result after the prediction		
Admin	Analyse	USN 3	As an admin, I will analyse the given dataset	I can analyse the dataset	High	Sprint-2

	Predict	USN 4	As an admin, I will predict the length of stay	I can predict the length of stay	High	Sprint-1

6 Project planning & scheduling

6.1 Sprint Planning & Estimation

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-1	Data Collection	USN-1	The User needs a complete data about the patients admitted in the hospital and a dataset should be prepared.	2	Medium	Arun Kumar

Sprint-1	Data Exploration	USN-2	As a user, I need nicely visualized dashboard of number of beds occupied and number of free beds in hospital.	4	High	Karuppuchamy
Sprint-2	Track of patient visit of Hospital	USN-3	Tracking a patient Health care over years of visit and Screening of data they have in hospital.	2	Medium	Lenine Joseph
Sprint -2	Dashboard	USN - 4	As a user, I want the interactive dashboard to analyse the data. Have the data in terms of Graph.	4	High	Keerthana

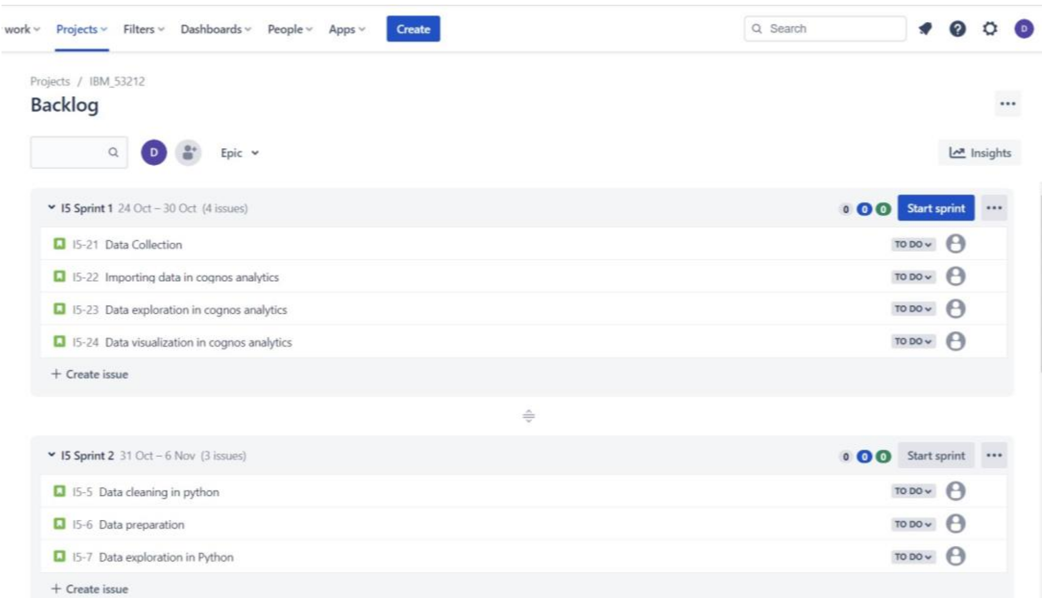
Sprint-3	Detailed EHR’s of patient	USN-5	Provided greater details in the EHR’s of individual patient with clear idea of what to do.	2	Medium	Saran
Sprint- 3	Story Creation	USN-6	As a user, I need the story animation of the data set with insights	4	High	Keerthana Arun Kumar
Sprint-4	Predict LOS	USN-7	As a user, I want the flawless system to predict the length of stay of the patients	4	High	Karuppuchamy Lenine Joseph
Sprint-4	Using ML algorithm for Prediction	USN-8	As a user, I need prior knowledge of LOS can aid in logistics such as room and	4	High	Saran Arun Kumar
			bed allocation planning.			

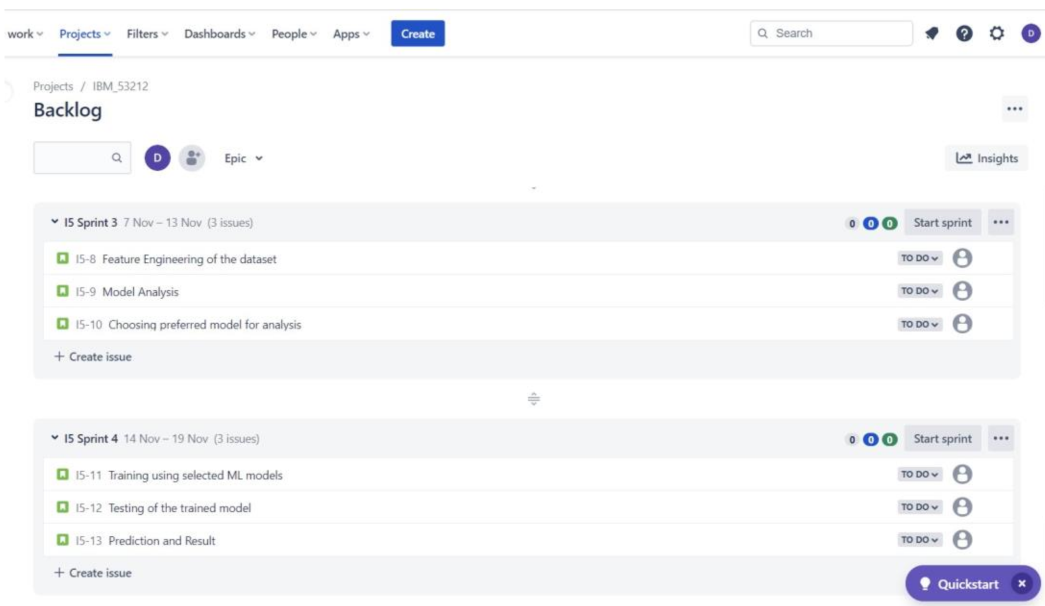
5.2 Sprint Delivery Schedule

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	20	6 Days	24 Oct 2022	30Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	06 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	13 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

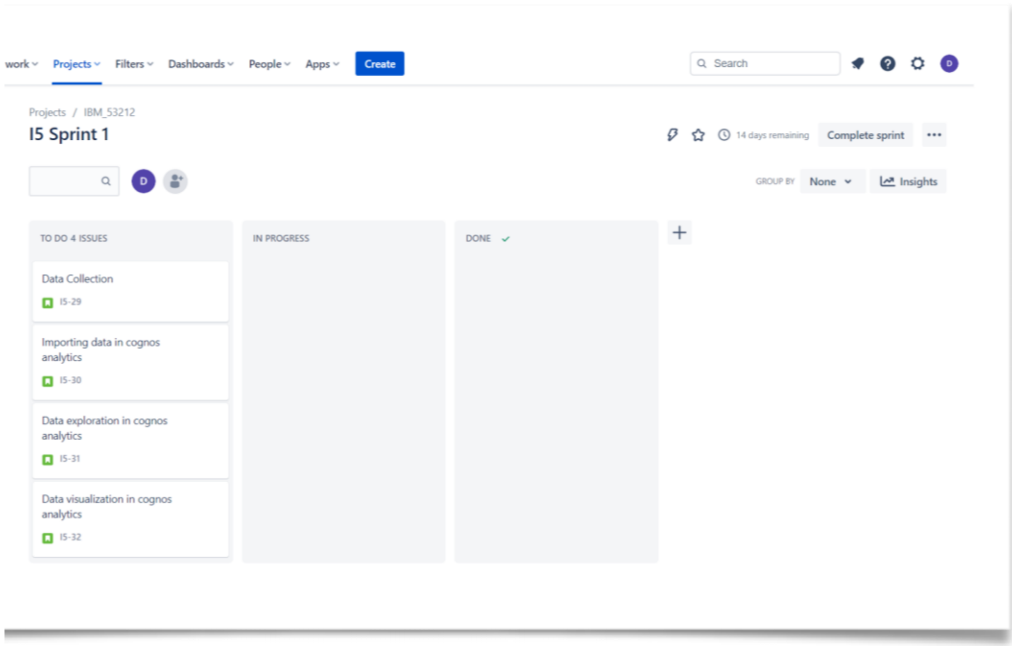
1. **Reports from JIRA**

Jira Sprints

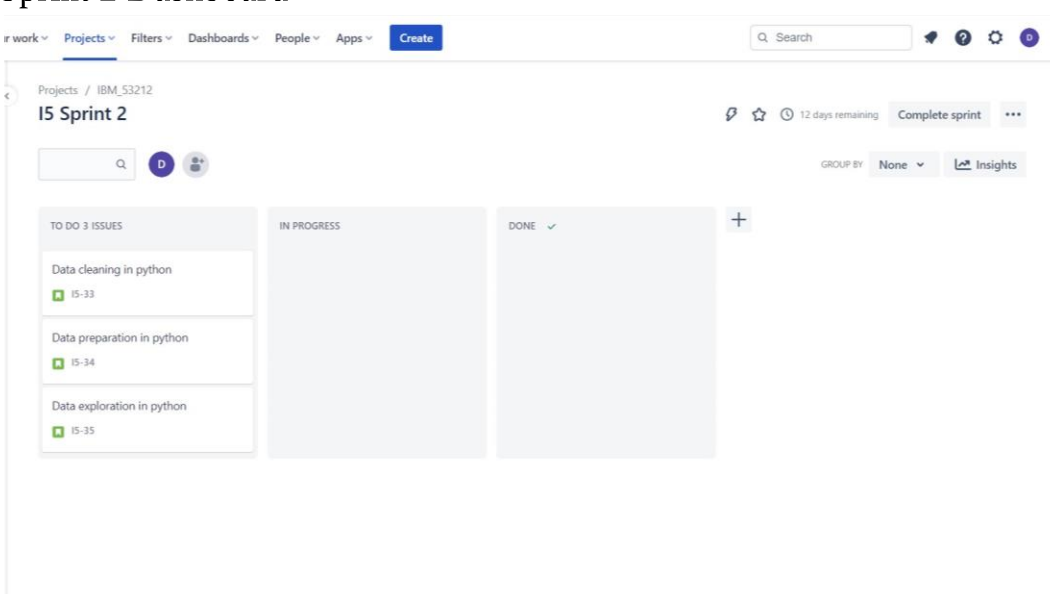

work
Projects
Filters
Dashboards
People
Apps
Create
Search
Projects / IBM_53212
Backlog
D
Epic
Insights
I5 Sprint 1 24 Oct – 30 Oct (4 issues)
0 0 0
Start sprint
I5-21 Data Collection
TO DO
I5-22 Importing data in cognos analytics
TO DO
I5-23 Data exploration in cognos analytics
TO DO
I5-24 Data visualization in cognos analytics
TO DO
+ Create issue
I5 Sprint 2 31 Oct – 6 Nov (3 issues)
0 0 0
Start sprint
I5-5 Data cleaning in python
TO DO
I5-6 Data preparation
TO DO
I5-7 Data exploration in Python
TO DO
+ Create issue


work
Projects
Filters
Dashboards
People
Apps
Create
Search
Projects / IBM_53212
Backlog
D
Epic
Insights
I5 Sprint 3 7 Nov – 13 Nov (3 issues)
0 0 0
Start sprint
I5-8 Feature Engineering of the dataset
TO DO
I5-9 Model Analysis
TO DO
I5-10 Choosing preferred model for analysis
TO DO
+ Create issue
I5 Sprint 4 14 Nov – 19 Nov (3 issues)
0 0 0
Start sprint
I5-11 Training using selected ML models
TO DO
I5-12 Testing of the trained model
TO DO
I5-13 Prediction and Result
TO DO
+ Create issue
Quickstart

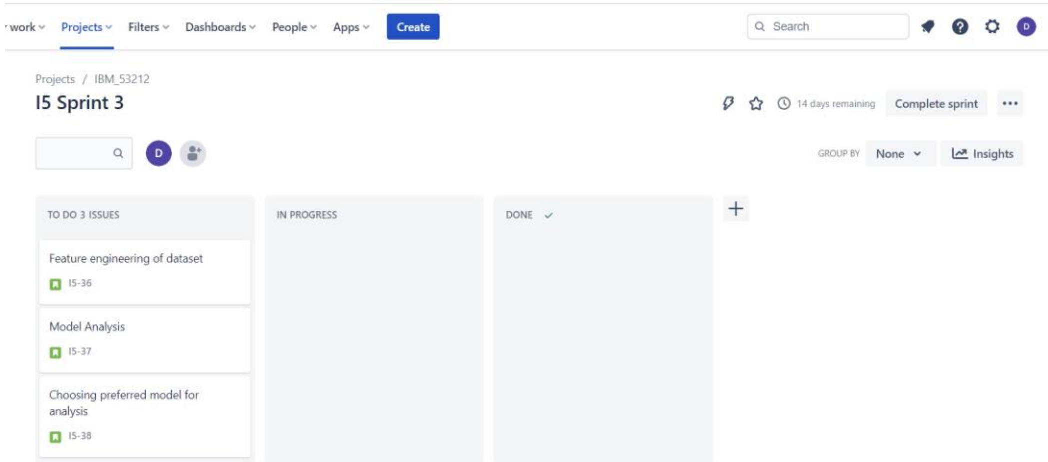
Sprint 1 Dashboard



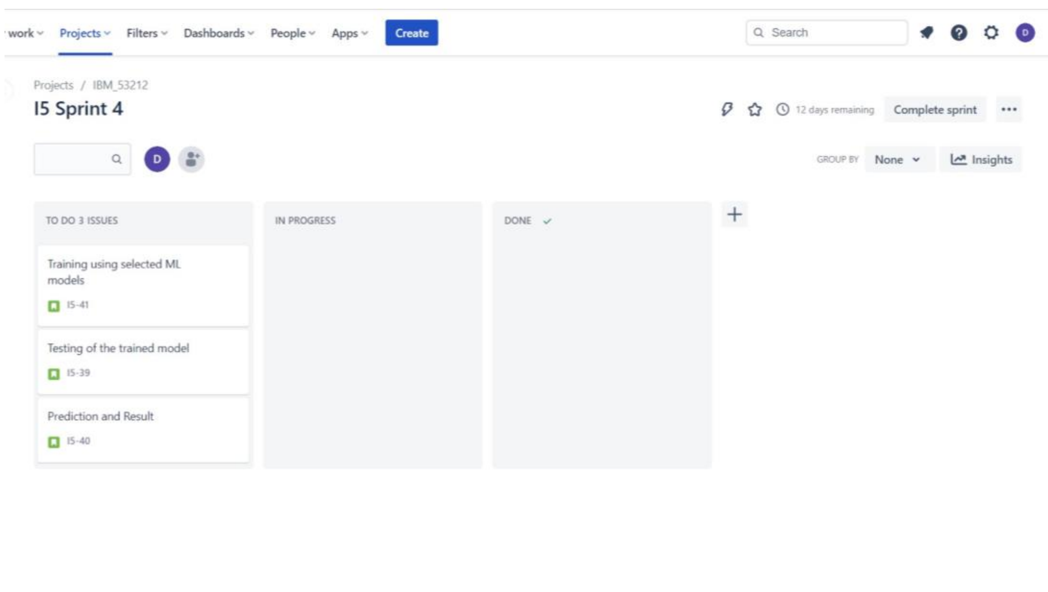
Sprint 2 Dashboard



Sprint 3 Dashboard



Sprint 4 Dashboard



7 . Coding & solutioning

ML Models

Naive Bayes Model

In Bayes theorem, given a Hypothesis H and Evidence E, it states that the relation between the probability of Hypothesis P(H) before getting Evidence and

probability of hypothesis after getting Evidence P(H|E) P (H | E) = [$\boldsymbol{P}$ ($\boldsymbol{E}$ | $\boldsymbol{H}$) / $\boldsymbol{P}$($\boldsymbol{E}$)] P(H)
When we apply Bayes Theorem to our data it represents as follows.

- P(H) is the prior probability of a patient's length of stay (LOS).
- P(E) is the probability of a feature variable.
- P(E|H) is the probability of a patient's LOS given that the features are true. • P(H|E) is the probability of the features given that patient's LOS is true.

Model is trained using Gaussian Naïve Bayes classifier, partitioned train data is fed to the model in array format then the trained model is validated using validation data.

This model gives an accuracy score of 34.55% after validating.

2) XGBoost Model
Boosting is a sequential technique that works on the principle of an ensemble. At any instant T, the model outcomes are weighed based on the outcomes of the previous instant (T -1). It combines the set of weak learners and improves prediction accuracy. Tree ensemble is a set of classification and regression trees. Trees are grown one after another, and they try to reduce the misclassification rate. The final prediction score of the model is calculated by summing up each and individual score.
Before feeding train data to the XGB Classifier model, booster parameters must be tuned. Tunning the model can prevent overfitting and can yield higher accuracy. In this XGBoost model, we have used the following parameters for tunning,

- learning_rate = 0.1 - step size shrinkage used to prevent overfitting. After each boosting step, we can directly get the weights of new features, and eta shrinks the feature weights to make the boosting process more conservative.
- max_depth = 4 – Maximum depth of the tree. This value describes the complexity of the model. Increasing its value results in overfitting.
- n_estimators = 800 – Number of gradient boosting trees or rounds. Each new
  tree attempts to model and correct for the errors made by the sequence of

previous trees. Increasing the number of trees can yield higher accuracy but the model reaches a point of diminishing returns quickly.

- objective = ‘multi:softmax’ – this parameter sets XGBoost to do multiclass classification using the softmax objective because the target variable has 11 Levels.
- reg_alpha = 0.5 - L1 regularization term on weights. Increasing this value will make the model more conservative.
- reg_lambda = 1.5 - L2 regularization term on weights and is smoother than L1 regularization. Increasing this value will model more conservative.
- min_child_weight = 2 - Minimum sum of instance weight needed in a child.

Once the model was trained and validated, it yields an accuracy score of 43.04%. This model nearly took 25 minutes to get trained but when compared to the Naïve Bayes model it gave an 8.5% improvement.

3) Neural Network Model

Neural Networks are built of simple elements called neurons, which take in a real value, multiply it by weight, and run it through a non-linear activation function. The process records one at a time and learns by comparing their classification of the record with the known actual classification of the record. The errors from the initial classification of the first record are fed back into the network and used to modify the network's algorithm for further iterations.In this neural network model, there are **six** dense layers, the final layer is an output layer with an activation function “**SoftMax**”. SoftMax is used here because each patient must be classified in one of the 11 levels in the Stay variable.

In this model, increasing the number of neurons from each layer to the other layer, will increase the hypothetical space of the model and try to learn more patterns from the data. There are a total of **442,571** trainable parameters. Every layer is activated using “**relu**” activation function because it overcomes the vanishing gradient problem, allowing models to learn faster and perform better.

Finally, evaluating the model with a test set yields an accuracy score of **41.79%**.

Neural Networks supposedly performs better than any other models. But because of the smaller dataset, it was not able to learn more accurately than the XGBoost model. It nearly took 20 minutes to train the model.

In the Naive Bayes model, patients are more likely to be misclassified. This model is biased towards the duration of 21-30 days, it has classified 72,206 patients for this level.
Whereas the other two models XGBoost and Neural Networks are predicting mostly similar Length of Stay for the patient
Examining these predictions, many of the patients are staying in the hospital for 21-30 days and very few people are staying for 61-70 days. As far as the distribution of Length of Stay is concerned, 13% of the patients are discharged from the hospital within 20 days and 1% of the overall patients are staying in the hospital for more than 60 days.

9) Results

9.1 Performance metrics

Finally, evaluating the model with a test set yields an accuracy score of **42.05%**. Neural Networks supposedly performs better than any other models. But because of the smaller dataset, it was not able to learn more accurately than the XGBoost model.
In the Naïve Bayes model, patients are more likely to be misclassified. This model is biased towards the duration of 21-30 days, it has classified 72,206 patients for this level

Length of Stay	Predicted Observations from Naïve Bayes	Predicted Observations from XGBoost	Predicted Observations from Neural Network
0-10 Days	2598	4373	4517
11-20 Days	26827	39337	35982
21-30 Days	**72206**	58261	61911
31-40 Days	15639	12100	8678
41-50 Days	469	61	26
51-60 Days	13651	19217	21709
61-70 Days	92	16	1
71-80 Days	955	302	248
81-90 Days	296	1099	1165
91-100 Days	2	78	21
More than 100 Days	4322	2213	2799

Whereas the other two models XGBoost and Neural Networks are predicting mostly similar Length of Stay for the patient, we can see this similarity for the first five cases. In we can see that the observations classified by both these models are marginally similar.

case_id	Length of Stay predicted	Length of Stay predicted	Length of Stay predicted
	from Naïve Bayes	from XGBoost	from Neural Networks
318439	21-30	0-10	0-10
318440	51-60	51-60	51-60
318441	21-30	21-30	21-30
318442	21-30	21-30	21-30
318443	31-40	51-60	51-60

Examining these predictions, many of the patients are staying in the hospital for 21-30 days and very few people are staying for 61-70 days. As far as the distribution of Length of Stay is concerned, 13% of the patients are discharged from the hospital within 20 days and 1% of the overall patients are staying in the hospital for more than 60 days.

10) Advantages:

11) Conclusion

In this project, different variables were analyzed that correlate with Length of Stay by using patient-level and hospital-level data.

By predicting a patient's length of stay at the time of admission helps hospitals to allocate resources more efficiently and manage their patients more effectively. Identifying factors that associate with LOS to predict and manage the number of days patients stay, could help hospitals in managing resources and in the development of new treatment plans. Effective use of hospital resources and reducing the length of stay can reduce overall national medical expenses.

12) Future insights

- **Smart Staffing & Personnel Management:** having a large volume of quality data helps health care professionals in allocating resources efficiently. Healthcare professionals can analyze the outcomes of checkups among individuals in various demographic groups and determine what factors prevent individuals from seeking treatment.
- **Advanced Risk & Disease Management:** Healthcare institutions can offer accurate, preventive care. Effectively decreasing hospital admissions by digging into insights such as drug type, conditions, and the duration of patient visits, among many others.
- **Real-time Alerting: Clinical Decision Support (CDS):** applications in hospitals analyzes patient evidence on the spot, delivering recommendations to health professionals when they make prescriptive choices. However, to prevent unnecessary in-house procedures, physicians prefer people to stay away from hospitals

- **Enhancing Patient Engagement:** Every step they take, heart rates, sleeping habits, can be tracked for potential patients (who use smart wearables). All this information can be correlated with other trackable data to identify potential health risks.

Appendix:
Code:
Feature engineering:

```
def get_countid_enocde(train, test, cols, name):
  temp = train.groupby(cols)['case_id'].count().reset_index().rename(columns =
{'case_id': name}) temp2 =
  test.groupby(cols)['case_id'].count().reset_index().rename(columns =
{'case_id': name}) train = pd.merge(train, temp, how='left',
  on= cols) test = pd.merge(test,temp2, how='left', on= cols)
  train[name] = train[name].astype('float') test[name] =
  test[name].astype('float')
  train[name].fillna(np.median(temp[name]), inplace = True)
  test[name].fillna(np.median(temp2[name]), inplace =
  True) return train, test

train, test = get_countid_enocde(train, test, ['patientid'], name = 'count_id_patient')
train, test = get_countid_enocde(train, test,
                                 ['patientid', 'Hospital_region_code'], name =
'count_id_patient_hospitalCode') train, test
= get_countid_enocde(train, test,
                                 ['patientid', 'Ward_Facility_Code'], name =
'count_id_patient_wardfacilityCode')


# Droping duplicate columns test1 = test.drop(['Stay', 'patientid',
'Hospital_region_code', 'Ward_Facility_Code'], axis =1) train1 =
train.drop(['case_id', 'patientid', 'Hospital_region_code', 'Ward_Facility_Code'],
axis =1)
```

# Splitting train data for Naive Bayes and XGBoost
X1 = train1.drop('Stay', axis =1) y1 = train1['Stay']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X1, y1, test_size =0.20, random_state =100)

Models Naïve bayes Model

from sklearn.naive_bayes import GaussianNB
target = y_train.values features =
X_train.values classifier_nb = GaussianNB()
model_nb = classifier_nb.fit(features, target)

prediction_nb = model_nb.predict(X_test) from
sklearn.metrics import accuracy_score acc_score_nb
= accuracy_score(prediction_nb,y_test)
print("Acurracy:", acc_score_nb*100) **XGBoost model**

import xgboost classifier_xgb = xgboost.XGBClassifier(max_depth=4, learning_rate=0.1, n_estimators=800, objective='multi:softmax', reg_alpha=0.5, reg_lambda=1.5, booster='gbtree', n_jobs=4, min_child_weight=2, base_score= 0.75) model_xgb = classifier_xgb.fit(X_train,

y_train)

prediction_xgb = model_xgb.predict(X_test)
acc_score_xgb = accuracy_score(prediction_xgb,y_test)
print("Accuracy:", acc_score_xgb*100)

Neural Network
X = train.drop('Stay', axis =1)
y = train['Stay']
print(X.columns) z =

```
test.drop('Stay', axis = 1)
print(z.columns)

# Data Scaling
from sklearn import preprocessing
X_scale = preprocessing.scale(X)
X_scale.shape

X_train, X_test, y_train, y_test = train_test_split(X_scale, y, test_size =0.20,
random_state =100)

import keras from keras.models
import Sequential from keras.layers
import Dense import tensorflow as tf

from keras.utils import to_categorical
#Sparse Matrix a =
to_categorical(y_train) b
= to_categorical(y_test)

model = Sequential() model.add(Dense(64, activation='relu',
input_shape = (254750, 20))) model.add(Dense(128,
activation='relu')) model.add(Dense(256, activation='relu'))
model.add(Dense(512, activation='relu')) model.add(Dense(512,
activation='relu')) model.add(Dense(11, activation='softmax'))

model.compile(optimizer= 'SGD',
        loss='categorical_crossentropy',
        metrics=['accuracy'])
callbacks = [tf.keras.callbacks.TensorBoard("logs_keras")]
model.fit(X_train, a, epochs=20, callbacks=callbacks, validation_split = 0.2)
```

GitHub link: https://github.com/Arunkumar9221/IBM-Project-41960-1660646548