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
- Naïve 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 - *Jason*

Brownlee - Link

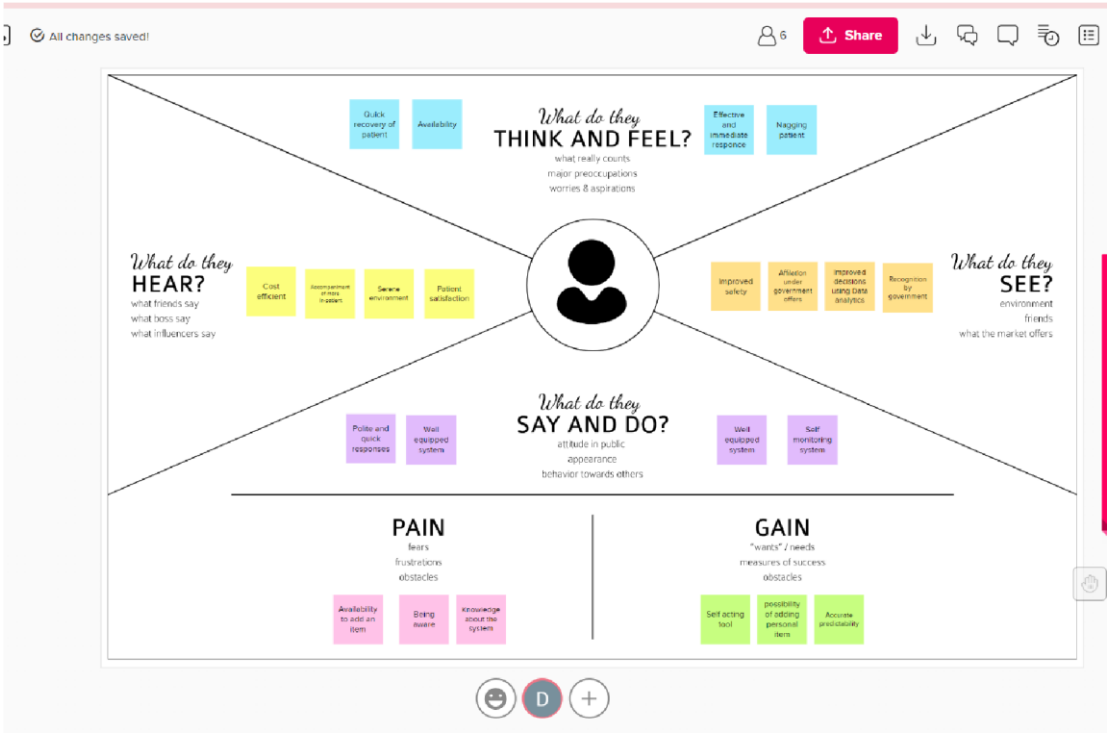
- Big Data Analytics in Healthcare That Can Save People - Sandra Durcevic - Link
- Learning Process of a Neural Network – Jordi Torres - Link

2.3 **Problem statement**

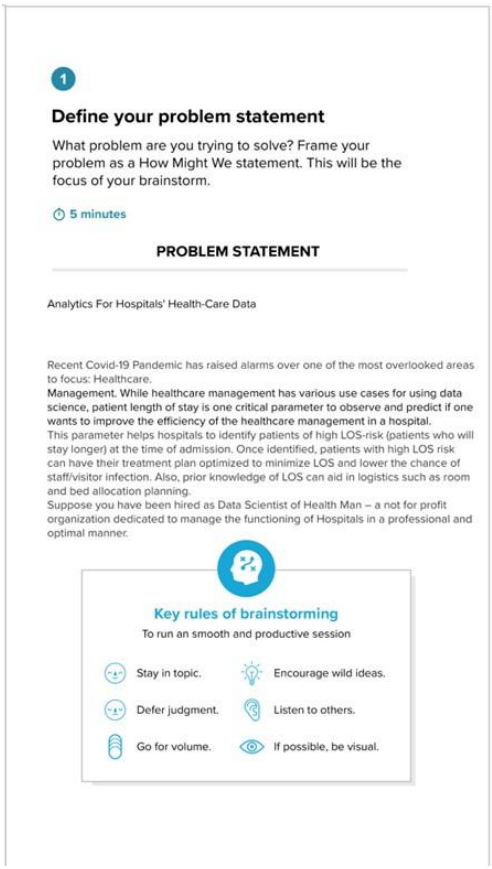
The task is to accurately predict the Length of Stay for each patient on case-by-case basis so that the Hospitals can use this information for optimal resource allocation and better functioning. The length of stay is divided into 11 different classes ranging from 0-10 days to more than 100 days.

3. Ideation & proposed solution

3.1 **Empathy map Canvas**



3.2 **Ideation and Brainstorming**



1

Define your problem statement

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

5 minutes

PROBLEM STATEMENT

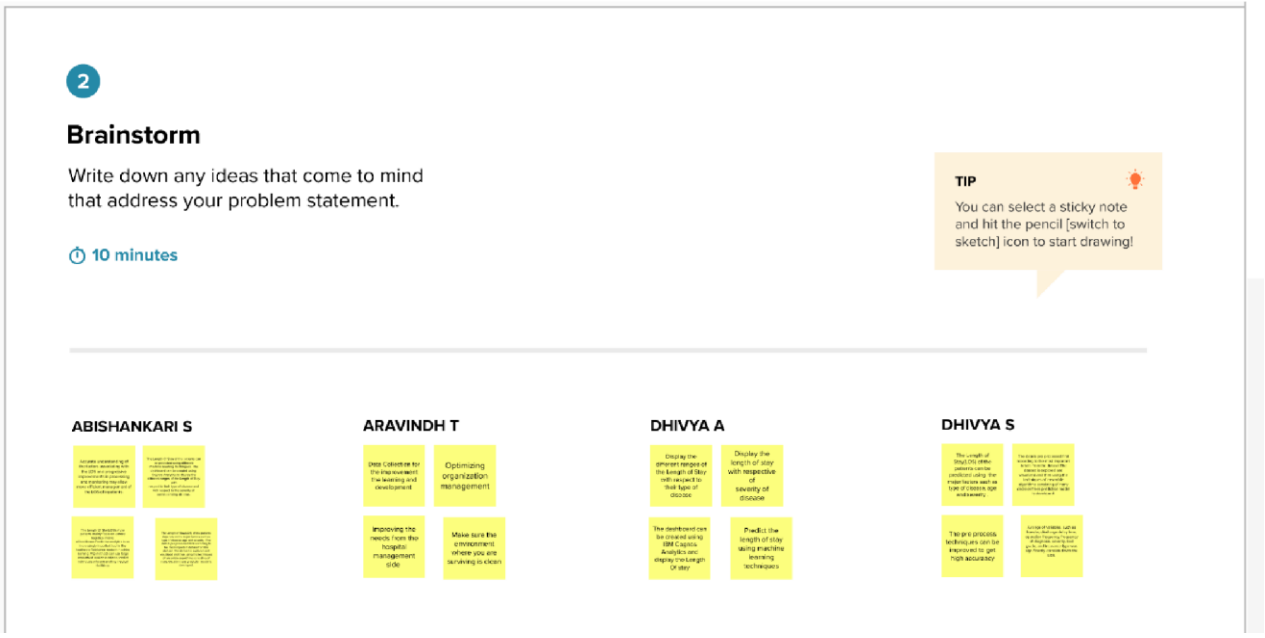
Analytics For Hospitals' Health-Care Data

Recent Covid-19 Pandemic has raised alarms over one of the most overlooked areas to focus: Healthcare.
Management. While healthcare management has various use cases for using data science, patient length of stay is one critical parameter to observe and predict if one wants to improve the efficiency of the healthcare management in a hospital.
This parameter helps hospitals to identify patients of high LOS-risk (patients who will stay longer) at the time of admission. Once identified, patients with high LOS risk can have their treatment plan optimized to minimize LOS and lower the chance of staff/visitor infection. Also, prior knowledge of LOS can aid in logistics such as room and bed allocation planning.
Suppose you have been hired as Data Scientist of Health Man – a not for profit organization dedicated to manage the functioning of Hospitals in a professional and optimal manner.

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.



2

Brainstorm

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

10 minutes

TIP

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

ABISHANKARI S

ARAVINDH T

Data Collection for the improvement the learning and development

Optimizing organization management

Improving the needs from the hospital management side

Make sure the environment where you are surviving is clean

DHIVYA A

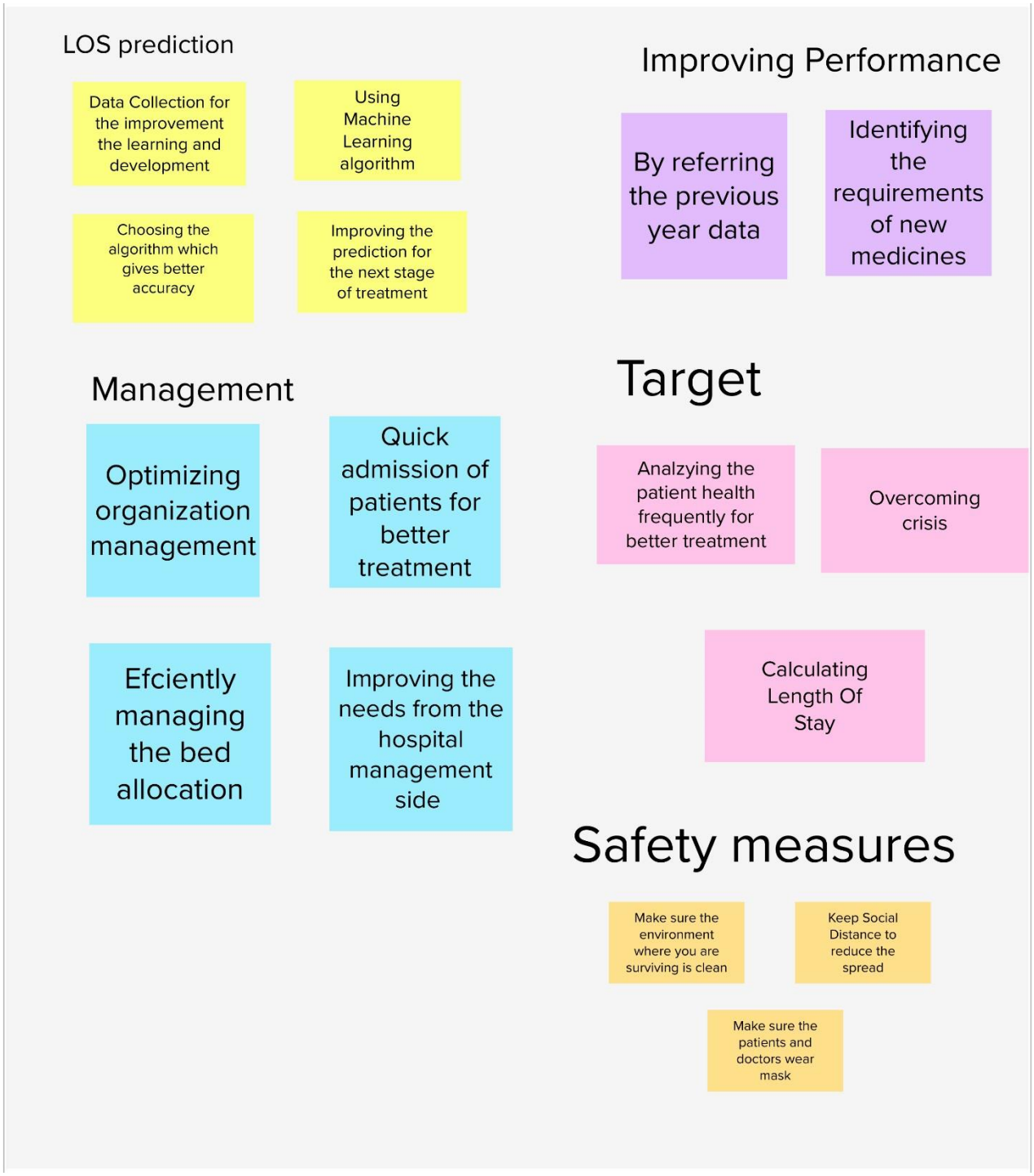
Display the different ranges of the length of Stay with respect to their type of disease

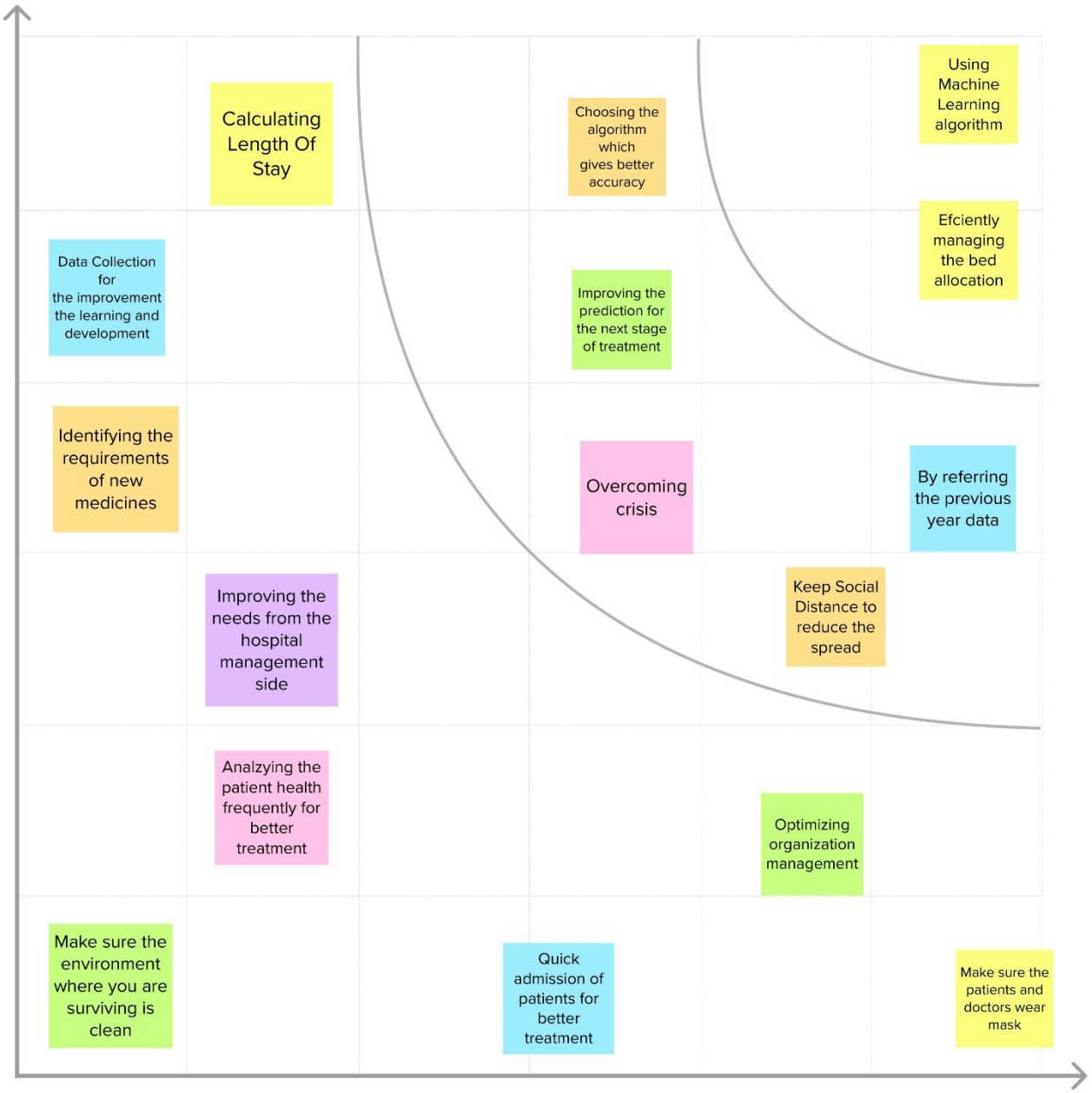
Display the length of stay with respective of severity of disease

The dashboard can be created using IBM Cognos Analytics and display the Length Of stay

Predict the length of stay using machine learning techniques

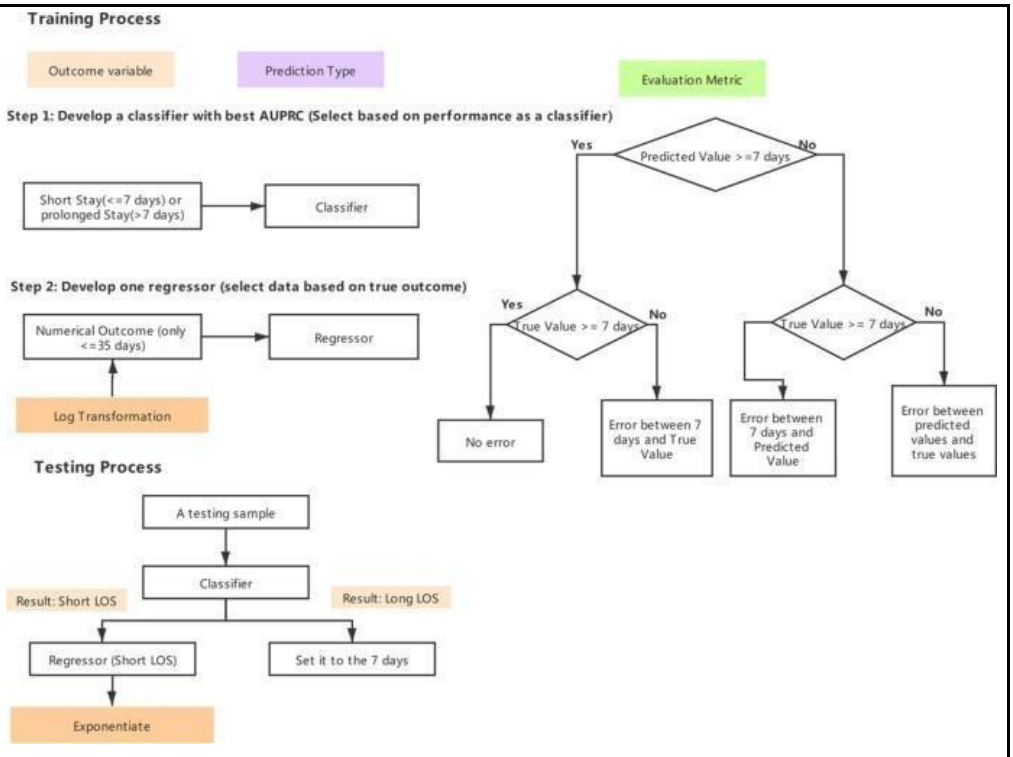
DHIVYA S





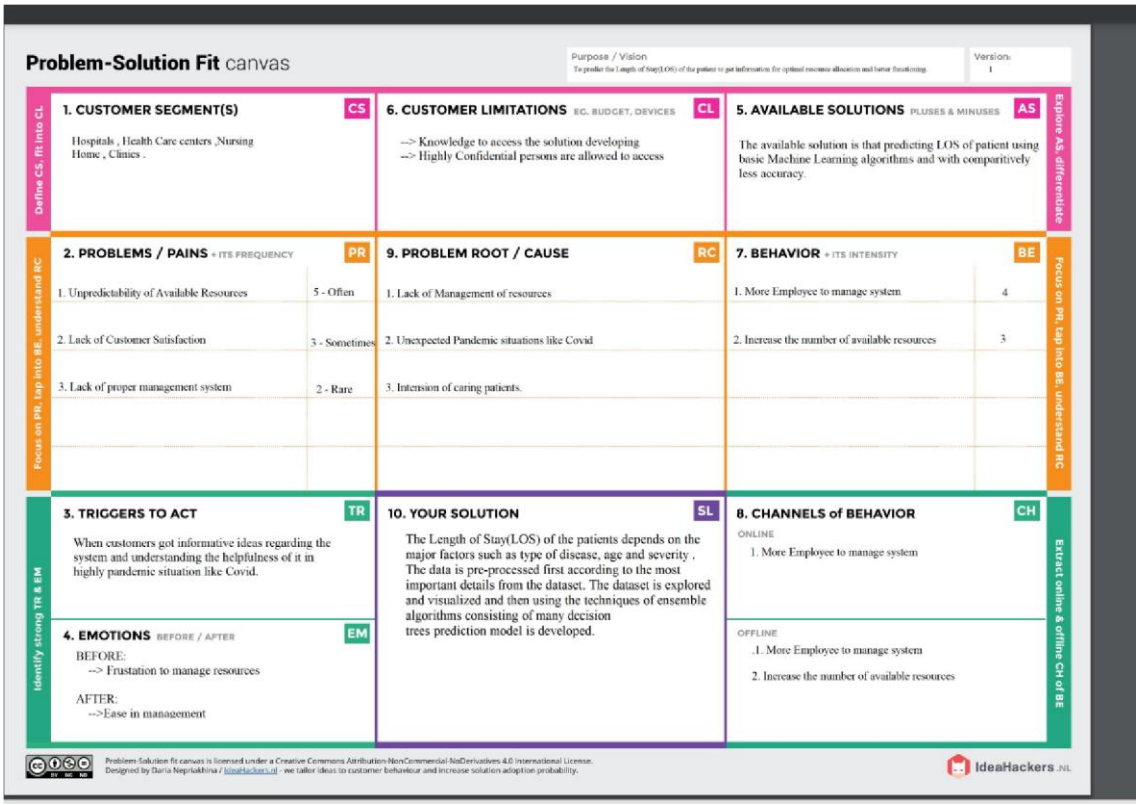
3.3 **Proposed solution**

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To predict the Length of Stay(LOS) of the patient to get information for optimal resource allocation and better functioning.

2.	Idea / Solution description	The Length of Stay(LOS) of the patients depends on the major factors such as type of disease, age and severity . The data is pre-processed first according to the most important details from the dataset. The dataset is explored and visualized and then using the techniques of ensemble algorithms consisting of many decisions trees prediction model is developed.
3.	Novelty / Uniqueness	The problem in real time is to find the availabilities. The uniqueness of our proposal is to convey the availabilities to the consumer with maximal accuracy.
4.	Social Impact / Customer Satisfaction n	It helps to identify patients of high LOS risk (patients who will stay longer) at the time of admission. Once identified, patients with high LOS risk can have their treatment plan optimized to minimize LOS and lower the chance of staff/visitor infection. Also, prior knowledge of LOS can aid in logistics such as room and bed allocation planning. The problem is to manage the functioning of Hospitals in a professional and optimal manner.
5.	Business Model (Revenue Model)	

6.	Scalability of the Solution	The primary model of our solution is to target only less consumers. So, it is sufficient to implement with local servers. In future, it can be extended to the large scale on needs .At that time the usage servers are considerably high. So, it can be extended further to the Cloud services.

3.4 Problem solution fit



4. Requirement analysis

4.1 Functional requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn

FR2	User Confirmation	Confirmation via Email Confirmation via OTP
FR3	Operability	Share patient data and make it interoperable among the management
FR4	Accuracy	The dashboard will be able to predict length of stay based on multiple combinations based on input sources with a n accuracy of up to 85%
FR5	Compliance	The product is to be used within the hospital, so any form of data need not be hidden
FR6	Productivity	The dashboard is believed to improve the predictions of Length of Stay and thereby creating a scenario of providing better solution

4. 2.Nonfunctional requirements

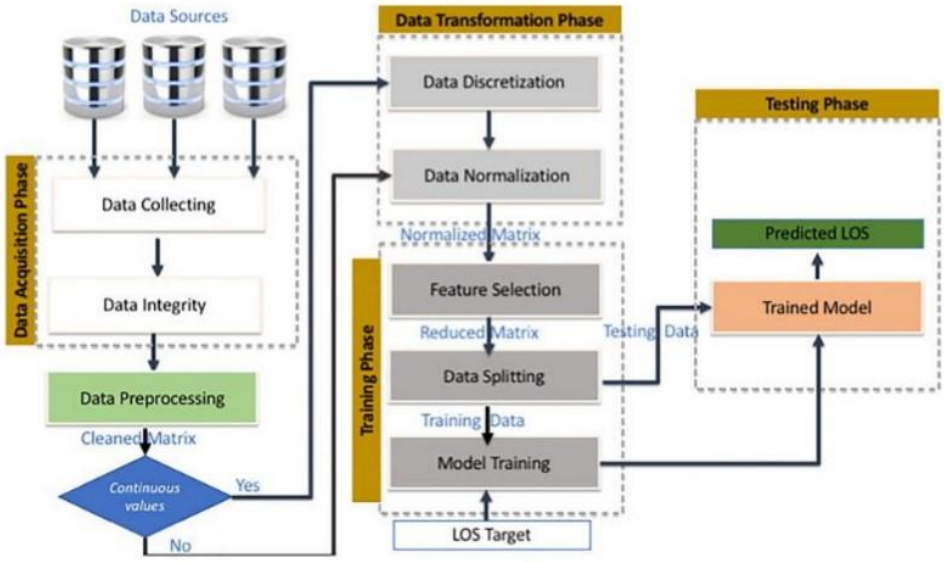
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This Dashboards are designed to offer a comprehensive overview of patient's LOS and do so using data visualization tools like charts and graphs.
NFR-2	**Security**	General industry level security shall be provided
NFR-3	**Reliability**	This dashboard will be consistent and reliable to the users and helps the user to use in effective, efficient and reliable manner.

NF R-4	**Performance**	The dashboard reduces the time needed for analysing data and has an automated system for that which improves the performance
NF R-5	**Availability**	The dashboard can be available to meet user's demand in timely manner and it is also helps to provide necessary information to the user's dataset
NF R-6	**Scalability**	It is a multi-tenant system which is capable of rimming on lower-level systems as well.

5. 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.



5.2 Solution & Technical Architecture

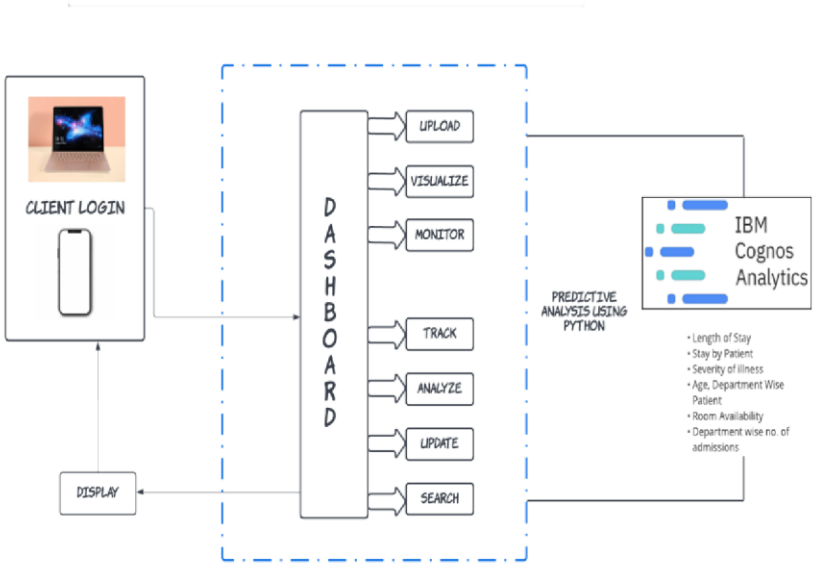


Table1: Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g., Web UI, Mobile App,	HTML, CSS, JavaScript / Angular Js / React Js etc…
2.	Application Logic-1	Chatbot etc. 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	MySQL, NoSQL, etc.
6.	Cloud Database	etc. 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	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	Encryption.
3.	Scalable Architecture	firewalls etc. Justify the scalability of architecture (3 –	Can supports higher workloads
4.	Availability	tier, Micro Justify the availability of application (e.g., use of load balancers,	Highly available

5.	Performance	distributed servers Design consideration for the performance of the application (number of requests per sec, use of Cache)	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 (Mobile user)	Registration	USN-1	As a user, I can register for the dashboard by entering my email, and password, and confirming my password.	I can access my account in the dashboard	High	Sprint-1
		USN-2	As a user, I will receive a confirmation email once I have registered for the dashboard	I can receive a confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the dashboard through social media	I can register & access the dashboard with Social Media Login	Low	Sprint-2
		USN-4	As a user, I can register for the dashboard through Gmail	I can register and access dashboard with Gmail	Medium	Sprint-2

	Login	USN-5	As a user, I can log into the application by entering email & password	I can login to the account in my email login.	High	Sprint-2

	Dashboard	USN-6	As a user, I can use my account in my dashboard for uploading dataset.	I can login to the account for uploading dataset.	Medium	Sprint-3
Customer (Web user)	Website	USN-7	As a user, I can use my dashboard in website	I can login into the dashboard by visiting website.	Medium	Sprint-3
Custom er Care Executive		USN-8	As a user, I can contact Customer care Executive for my login.	I can contact customer executive for my login.	High	Sprint-4
Administrator		USN-9	As a user, I can contact administrator for my queries.	I can contact administrator for solving my queries.	High	Sprint-4
Exploration	Dashboard	USN-10	As a user, I can prepare data by using Exploration Techniques.	I can prepare data by using Exploration Techniques.	High	Sprint-3

Presentation	Dashboard	USN-11	As a user, I can Present data in my dashboard.	I can present data by using my account in dashboard.	High	Sprint-4
Visualization	Dashboard	USN-12	As a user, I can Prepare Data by using Visualization Techniques.	I can prepare data by using Visualization Techniques.	High	Sprint-3

6. Project planning & scheduling

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the dashboard by entering my email, and password, and confirming my password.	10	High	Abishankari S

Sprint-1		USN-2	As a user, I will receive a confirmation email once I have registered for the dashboard	10	High	Aravindh T
Sprint-2		USN-3	As a user, I can register for the dashboard through social media	8	Low	Dhivya A
Sprint-2		USN-4	As a user, I can register for the dashboard through Gmail	8	Medium	Dhivya S

Sprint-2	Login	USN-5	As a user, I can log into the application by entering email & password	4	High	Abishankari S
Sprint-3	Dashboard	USN-6	As a user, I can use my account in my dashboard for uploading dataset.	8	Medium	Aravindh T

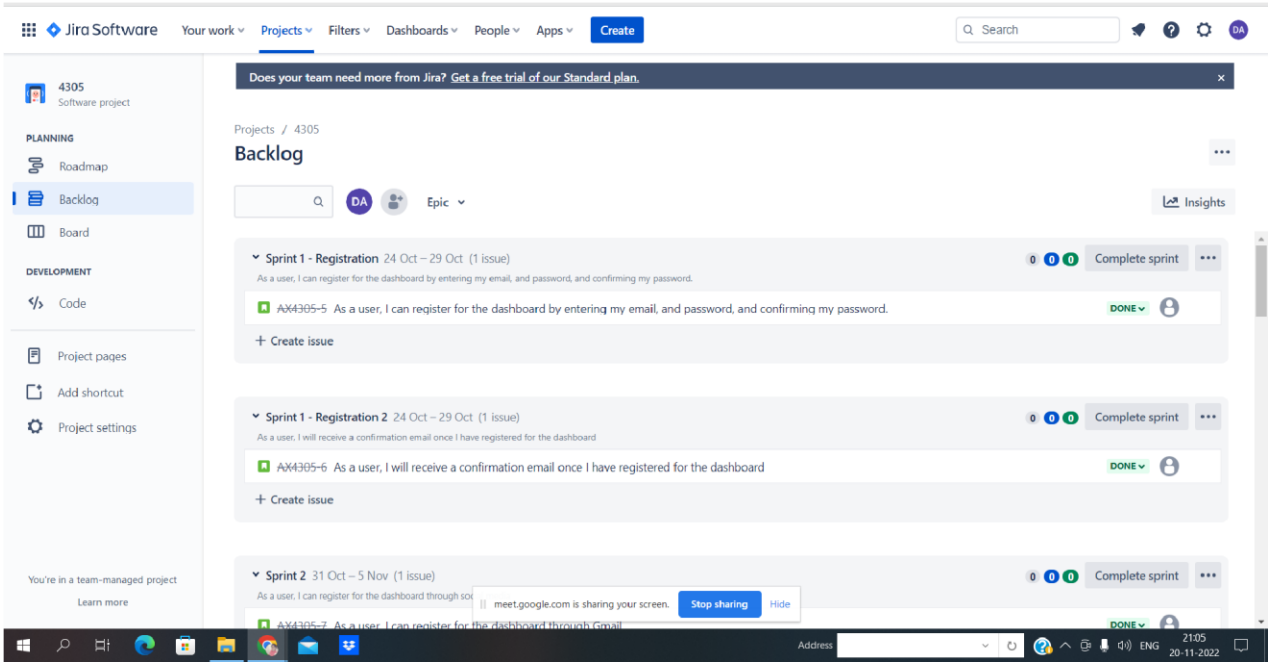
Sprint-3	Website	USN-7	As a user, I can use my dashboard in website	5	Medium	Dhivya A
Sprint-3		USN-8	As a user, I can contact the Customer care Executive for my login.	2	High	Dhivya S
Sprint-3		USN-9	As a user, I can contact the administrator for my queries.	5	High	Abishankari S
Sprint-4	Dashboard	USN-10	As a user, I can prepare data by using Exploration Techniques.	4	High	Aravindh T
Sprint-4	Dashboard	USN-11	As a user, I can Present data in my dashboard.	8	High	Dhivya A
Sprint-4	Dashboard	USN-12	As a user, I can Prepare Data by using Visualization Techniques.	8	High	Dhivya S

6. 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	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3. Reports from JIRA

Jira Sprints



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)

$$P(H \mid E) = [\, \boldsymbol{P}(\boldsymbol{E} \mid \boldsymbol{H}) / \boldsymbol{P}(\boldsymbol{E})] \, P(H)$$

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 the 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.

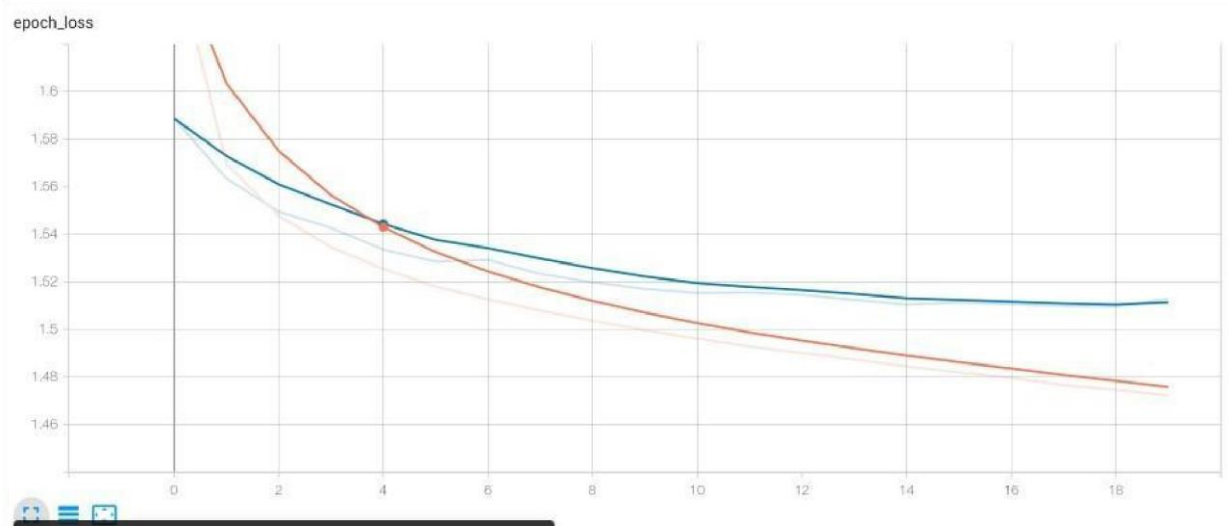
Finally, evaluating the model with a test set yields an accuracy score of **41.79%**. Neural Networks supposedly performs better than any other models. But because of the smaller dataset, it was not able to learn more accurately than the XGBoost model. It nearly took 20 minutes to train the model.

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 from Naïve Bayes	Length of Stay predicted from XGBoost	Length of Stay predicted 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:

1.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

2.It helps hospitals in managing resources and in the development of new treatment plans

3. Effective use of hospital resources and reducing the length of stay can reduce overall national medical expenses.

11) Conclusion

In this project, different variables were analysed 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

analyse the outcomes of check-ups 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 analyses 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:

```
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/IBM-EPBL/IBM-Project-10738-1659200545