ANAlYTICS FOR HOSPITALS HEALTH-CARE DATA

REPORT

TEAM ID PNT2022TMID35084

SUBMITTED BY,

JENI ALPHONSA A(963319106036)

JENILA J(963319106039)

JESNEY R(963319106041)

LISHA S(963319106053)

TABLE OF CONTENTS

ABSTRACT

The current study performs a systematic literature review (SLR) to synthesise prior research on the applicability of big data analytics (BDA) in healthcare. The SLR examines the outcomes of 41 studies, and presents them in a comprehensive framework. The findings from this study suggest that applications of BDA in healthcare can be observed from five perspectives, namely, health awareness among the general public, interactions among stakeholders in the healthcare ecosystem, hospital management practices, treatment of specific medical conditions, and technology in healthcare service delivery. This SLR recommends actionable future research agendas for scholars and valuable implications for theory and practice.

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 analyticsrefers to the analysis of data using quantitative and qualitative techniques to explore trends and patterns in the acquired data. While healthcare management uses various metricsfor performance, a patient's lengthof stay is an importantone.

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

1.2. Purpose

Thegoal of this project is to accurately predict the Length of Stay for each patient so thatthe 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 toimprove the efficiency of thehealthcare management in a hospital.

2.2 References

i. Janatahack: Healthcare AnalyticsII - *Analytics Vidhya* - Link

ii. What Is Naive Bayes Algorithm in Machine Learning? - *Rohit Dwivedi*- Link

iii. Naïve Bayes for Machine Learning– From Zero to Hero - *Anand Venkataraman* - Link

iv. XGBoost Parameters - *XGBoost Documentation* - Link

v. Predicting Heart Failure Using Machine Learning, Part 2- *Andrew A Borkowski* - Link

vi. How to Tune the Number and Size of DecisionTrees with XGBoostin Python -*JasonBrownlee* - Link

vii. Big Data Analytics inHealthcare That Can Save People - *Sandra Durcevic*- Link

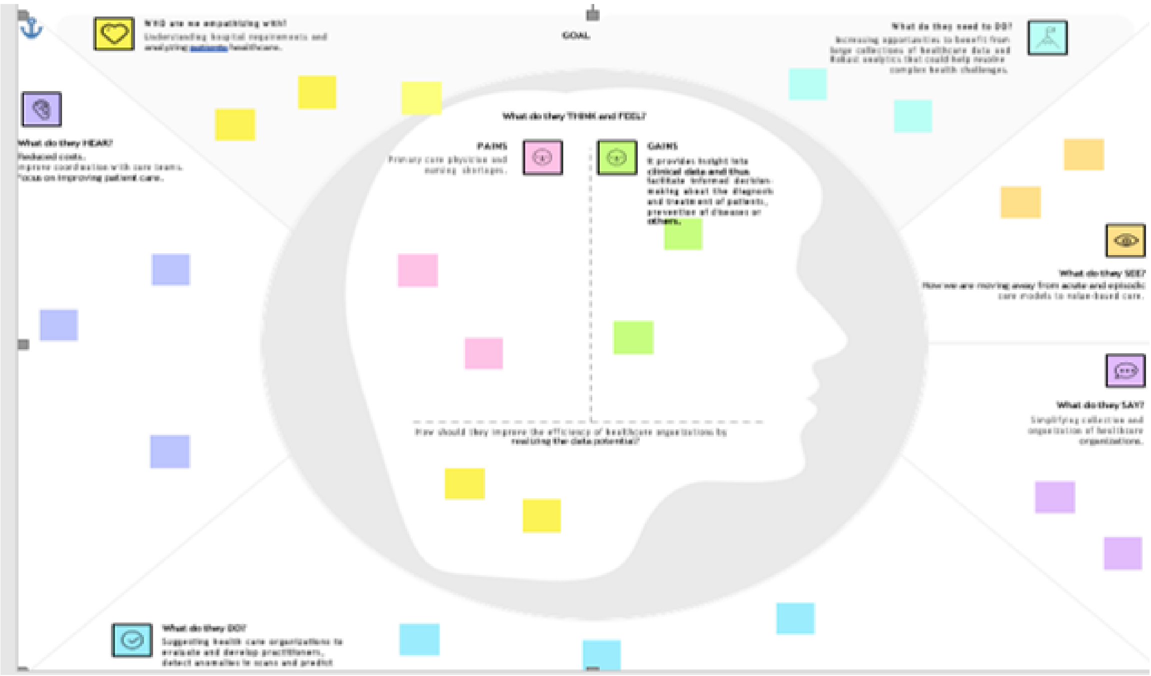
viii. Learning Process of a Neural Network – *Jordi Torres* - Link

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 differentclasses ranging from 0-10 days tomore than 100 days.

3. Ideation & proposed solution

3.1 Empathy map canvas



WHO are we empathizing with?
Understanding hospital requirements and analyzing patients healthcare.

What do they need to DO?
Increasing opportunities to benefit from large collections of healthcare data and Robust analytics that could help resolve complex health challenges.

What do they HEAR?
Reduced costs.
Improve coordination with care teams.
Focus on improving patient care.

What do they THINK and FEEL?

PAINS
Primary care physician and nursing shortages.

GAINS
It provides insight into clinical data and thus facilitate informed decision-making about the diagnosis and treatment of patients, prevention of diseases or others.

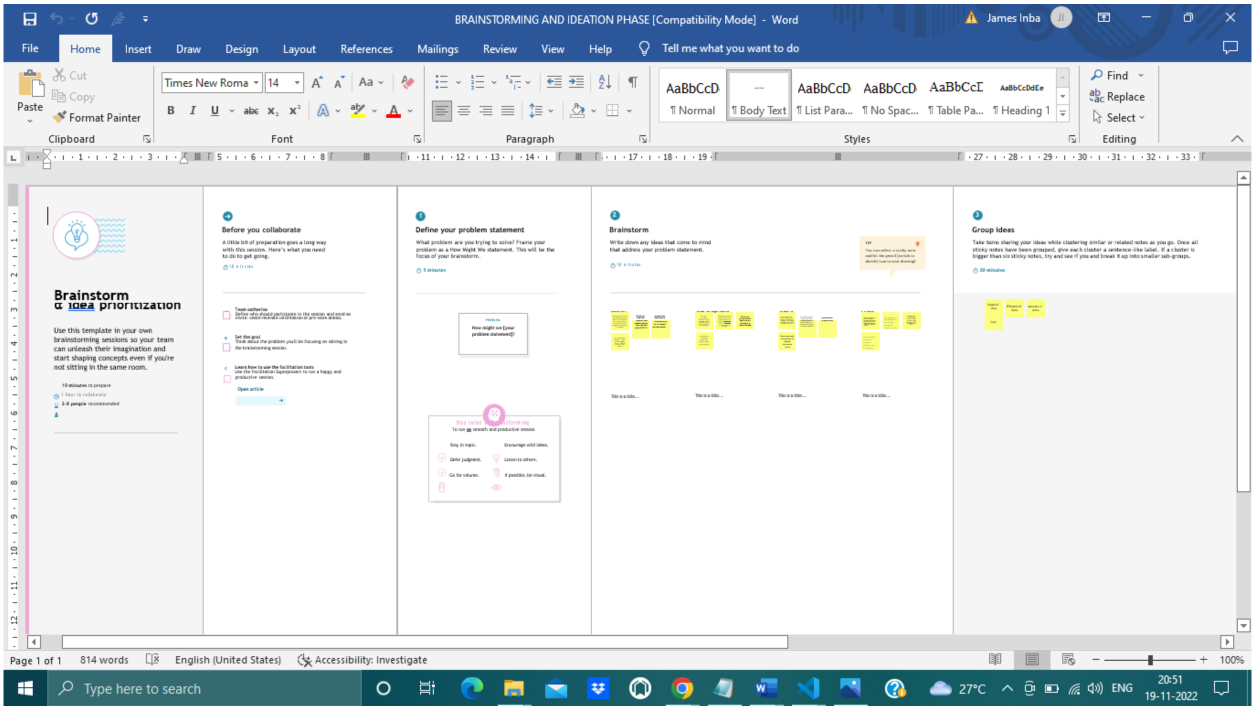
How should they improve the efficiency of healthcare organizations by realizing the data potential?

What do they SEE?
How we are moving away from acute and episodic care models to value-based care.

What do they SAY?
Simplifying collection and organization of healthcare organizations.

What do they DO?
Suggesting health care organizations to evaluate and develop practitioners, detect anomalies in scans and predict

3.2 Ideation and Brainstorming



Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article

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

How might we [your problem statement]?

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!

This is a title...
This is a title...
This is a title...
This is a title...

3 Group ideas

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

20 minutes

3.3 Proposed solution

S.No.	Parameter	Description
1.	Problem Statement (Problem tobesolved)	The task is to accurately predict the Length of Stay for each patient on case-by- case basisso that the Hospitals can use this informationfor optimal resource allocation and betterfunctioning. The lengthof stayisdivided into 11 different classes ranging from 0-10daysto more than 100 days.

2.	Idea / Solution description	Naïve Bayes is a classification technique that works on the principle of Bayes theorem with an assumption of independence among the variables. Here the goal is to predict Length of Stay i.e., “Stay” column (Target Variable) and it is classified into 11 levels. We must find the probabilityof each patient’s length of stay using feature variables, which contain the patient’s condition and hospital-level information. These feature variables are ordinal and naïve Bayesis a perfect multilevel classifier.
3.	Novelty / Uniqueness	Accurate understanding of the factorsassociating with the LOS and progressive improvements in processing and monitoring may allow more efficient management of theLOS of inpatients
4.	Social Impact / CustomerSatisfaction	A shorterLOS reduces the risk of acquiring staphinfections and otherhealthcare-relatedconditions, frees up vital bed spaces, and cuts overallmedical expenses
5.	Business Model (Revenue Model)	The length of stay (LOS) is an important indicator of the efficiency of hospital management. Reduction in the number of inpatient days results in decreased risk of infection and medication side effects, improvement in the quality of treatment, and increased hospital profitwith more efficient bed management
6.	Scalability of the Solution	Remote patient monitoring systems enabling effective distance treatment. Patient portals that allow people to better

		manage their health themselves;

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 Registration through LinkedIN
FR-2	User Confirmation	Confirmation via EmailConfirmation via OTP
FR-3	Operability	Share patient data and make it interoperable among themanagement
FR-4	Accuracy	The dashboard will be able to predict length of staybased on multiple combinations based on input sourceswith a n accuracy of upto 85%
FR-5	Compliance	The product is to be used withinthe hospital so any formofdata need not be hidden
FR-6	Productivity	The dashboard is believed to improve the predictions ofLength of Stay and thereby creating a scenario of providing bettersolution

4.2 Nonfunctional requirements

FR No.	Non-Functional Requirement	Description

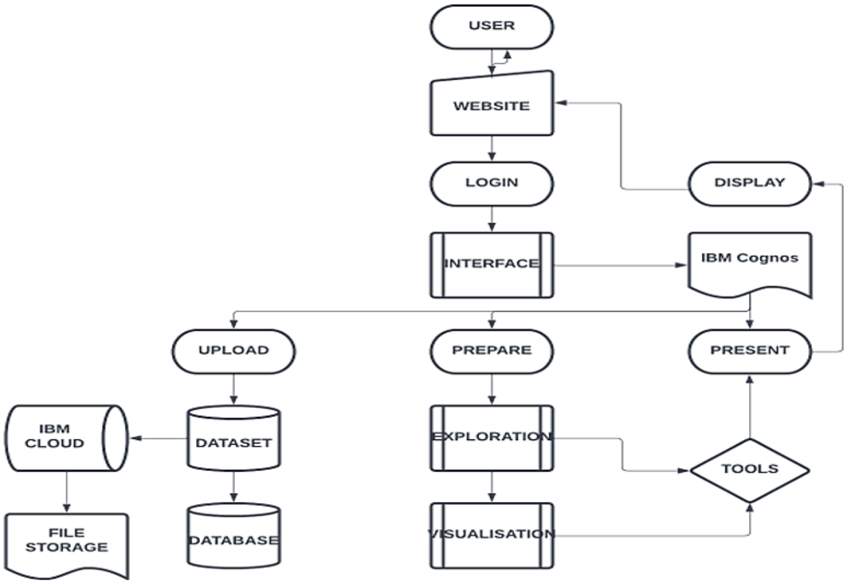
NFR-1	**Usability**	This Dashboards are designed to offer a comprehensive overview of patient’s LOS, anddosothrough the use of data visualization toolslike charts andgraphs.
NFR-2	**Security**	General industry level security shallbe provided
NFR-3	**Reliability**	This dashboard will be consistent and reliable to theusers and helps the user to use in effective, efficientand reliable manner.
NFR-4	**Performance**	The dashboard reduces the time needed for analysingdata and has an automated system forthatwhich improves the performance
NFR-5	**Availability**	The dashboard can available to meet user’sdemand in timely manner and it is also helps to providenecessary information to the user’s dataset
NFR-6	**Scalability**	It is a multi-tenant system which is capable ofrimming 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 informationflows within a system. A neat and clear DFD can depict the rightamount of the system requirement graphically. It shows how data enters and leaves the system, what changes theinformation, and where data is stored.

5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptancecriteria	Priority	Release
Customer	Dashboard	USN 1	As a user,I can uploadthe datasets to thedashboar d	I can access various operations	Medium	Sprint-4

	View	USN 2	As a user,I can view the patient details	I can view the visual data and the result after the prediction	Medium	Sprint-3
Admin	Analyse	USN 3	As an admin, I will analyse the given dataset	I can analyse thedataset	High	Sprint-2

6. Project planning& scheduling

6.1 Sprint Planning & Estimation

SPRINT 1

- Collection of data
- Data preprocessing
- Upload the dataset

SPRINT 2

DATA EXPLORATION:

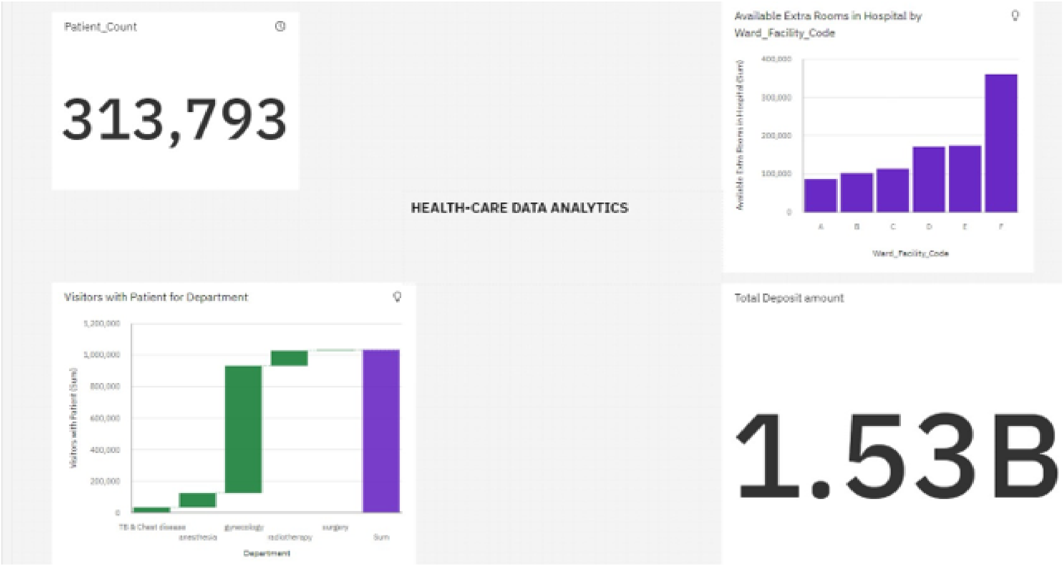
- Patient id by stay
- Patient id for department
- Severity of Illness by Age colored by City Code Hospital
- Case id by Ward Type

- Case id by Department
- Bed Grade by Department
- Case id by Severity of Illness
- Patient by Ward Type
- Available Extra Rooms in Hospital by Ward type
- Stay by Department
- Admission count for Department

SPRINT 3

CREATION OF DASHBOARD

DASHBOARD



HEALTH-CARE DATA ANALYTICS

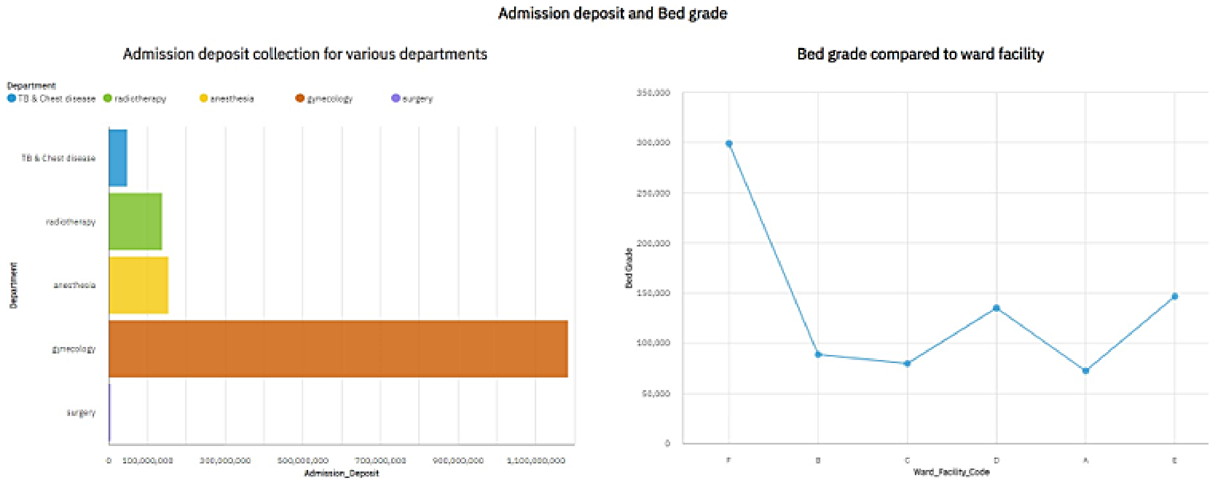
Patient_Count

313,793

Total Deposit amount

1.53B

SPRINT 4

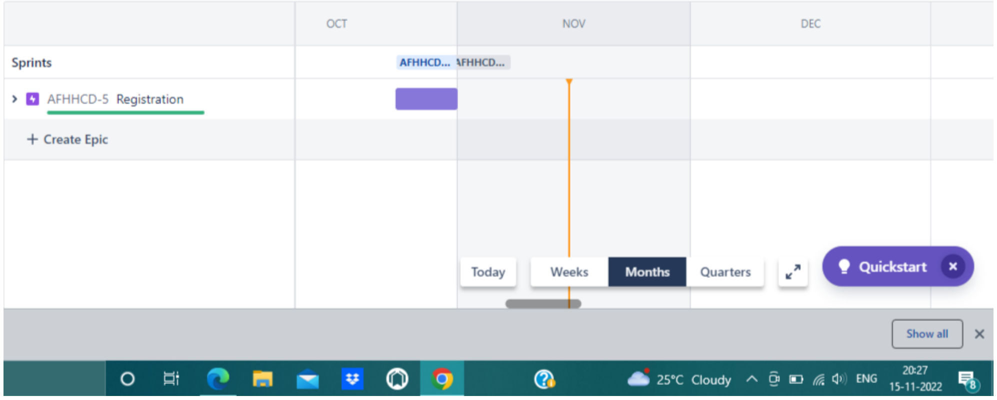


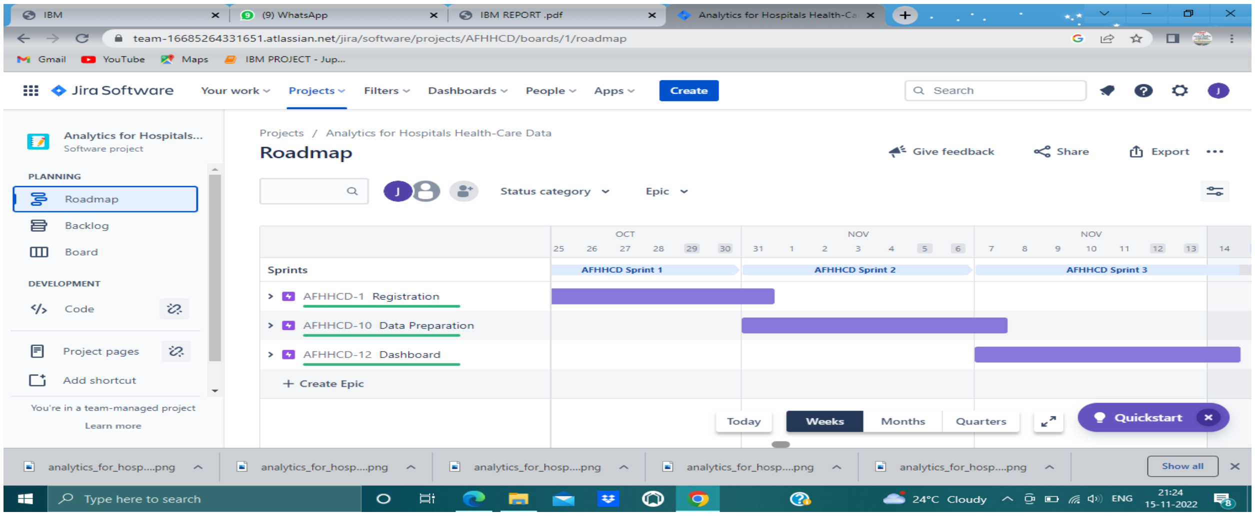
6.2 Sprint DeliverySchedule

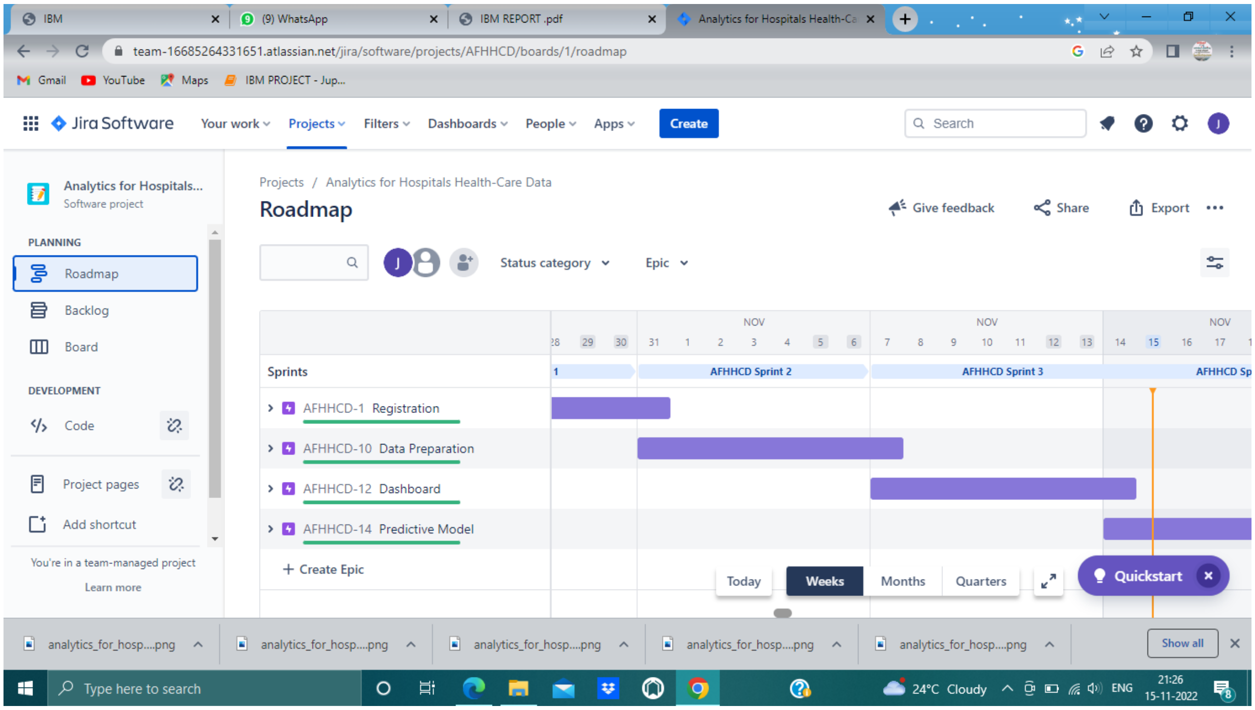
Sprint	Total StoryPoints	Duration	Sprint Start Date	Sprint End Date (Planne	Story Points Completed (as on Planned	Sprint Release Date (Actual)
Sprint -1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint -2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint -3	20	6 Days	07 Nov 2022	12 Nov 2022	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







CODING AND SOLUTIONING

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.

XGBoost Model

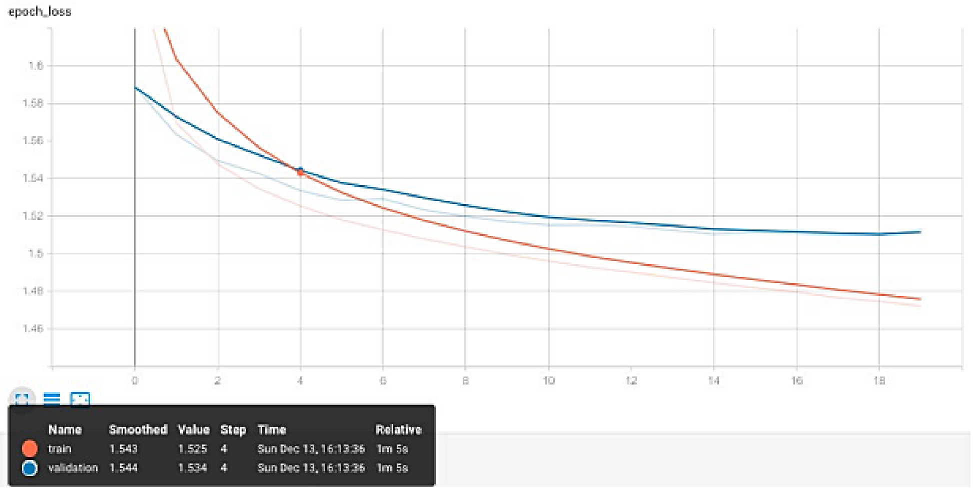
Boosting is a sequential technique that works on the principle of an ensemble. At any instant T, the model outcomes are weighed based on the outcomes of the previous instant (T -1). It combines the set of weak learners and improves prediction accuracy. Tree ensemble is a set of classification and regression 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.

9)RESULTS

9.1performance metrices

9) Results

9.1 Performance metrics



	Name	Smoothed	Value	Step	Time	Relative
●	train	1.543	1.525	4	Sun Dec 13, 16:13:36	1m 5s
●	validation	1.544	1.534	4	Sun Dec 13, 16:13:36	1m 5s

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.

CODE

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 24

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)
```