UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

PROJECT REPORT

1. INTRODUCTION:

1.1.PROJECT OVERVIEW:

A Student education plays a vital role in their life. While planning for education students often have several questions regarding the courses, universities, job opportunities, expenses involved, etc. Securing admission in their dream university is one of their main concerns. It is seen that often students prefer to pursue their education from universities which have global recognition. With the majority of worlds highly reputed universities, wide range of courses offered in every sector, highly accredited education system and teaching, scholarships provided to students, best job market and many more advantages make it the dream destination for the international students.

1.2.PURPOSE:

This is a Requirements Specification Document for a new web-based University Admissions Predictor. This Prediction System is an AI based application that asks for the users to input their academic transcripts data and calculates their chances of admission into the University Tier that they selected. It also provides an analysis of the data and shows how chances of admissions can depend on various factors. This document describes the scope, objectives and goals of the system. In addition to describing the non-functional requirements, this document models the functional requirements with use cases, interaction diagrams and class models. This document is intended to direct the design and implementation of the target system in an object-oriented language.

2. LITERATURE SURVEY:

2.1.EXISTING PROBLEM:

Today in college's student details are entered manually. The student details in separate records are tedious task. Referring to all these records updating is needed. There is a chance for more manual errors.

1.When the student comes in college.

2.First of all, he/she takes admission form from reception.

3.Fills it and submits it into office.

4.Filled form is first checked with documents like merit list an details came from university and verified by an official person, if there is any mistake then it is corrected.

5.Atthetimeofsubmissionofitthefeesisdepositedbythecandidate.

6. At the time of submission of admission form admission number is assigned to the candidate by the institute.

7. Candidate gets the receipt of fees deposition.

DISADVANTAGES OF EXISTING SYSTEM

1. Require much man power i.e. much efforts, much cost and hard to operate and maintain.

2. Since, all the work is done in papers so it is very hard to locate a particular student record when it is required.

2.2.REFERENCES:

1. J. Han, and M. Kamber, "Data Mining: Concepts and Techniques, 2nd edition", Morgan Kaufmann Publishers, 2006

2. J. W. Seifert, Data Mining: An Overview, C-RS Report for Congress, Dec. 16, 2004, www.fas.org/irp/crs/RL31798.pdf.

3. G. Ganapathy, and K. Arunesh, "Models for Recommender Systems in Web Usage Mining Based on User Ratings" Proceedings of the World Congress on Engineering, Vol. I WCE 2011.

4. D. Mican, and N. Tomai, "Association-Rules-Based Recommender System for Personalization in Adaptive Web-Based Applications" http://gplsi.dlsi.ua.es/congresos/qwe10/fitxers/QWE10-Mican.pdf.

5. S. Liao, T. Zou, and H. Chang,"An Association Rules and Sequential Rules Based Recommendation System", Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference, 12-14 Oct. 2008.

6. Q. Li, and B. M. Kim, "Clustering Approach for Hybrid Recommender System" Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI'03), 2003.

7. S. Nadi, M.H. Saraee, and A. Bagheri, "Hybrid Recommender System for Dynamic Web Users", International Journal Multimedia and Image Processing (IJMIP), Vol. 1, Issue 1, March 2011.

8. S. Tiwari, "A Web Usage Mining Framework for Business Intelligence", International Journal of Electronics Communication and Computer Technology (IJECCT) Vol. 1 Issue 1 Sep.2011.

9. X. Su, and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques" Advances in Artificial Intelligence Volume 2009, Article ID 421425.

10. J. A. Freeman, and D. M. Skapura, "Neural Networks: Algorithms. Applications. And Programming", Addison-Wesley Pub (Sd), June 1991.

11. E. Gottlieb, "Using integer programming to guide college admissions decisions: a preliminary report", Journal of Computing Sciences in Colleges, Volume 17, Issue 2, Pages: 271-279, 2001.

12. I. Hatzilygeroudis, A. Karatrantou, and C. Pierrakeas, "PASS: An Expert System with Certainty Factors for Predicting Student Success" Knowledge-Based Intelligent Information & Engineering Systems 2004.
www.informatik.uni?trier.de/~leydb/conf/kes/kes2004-1.htm

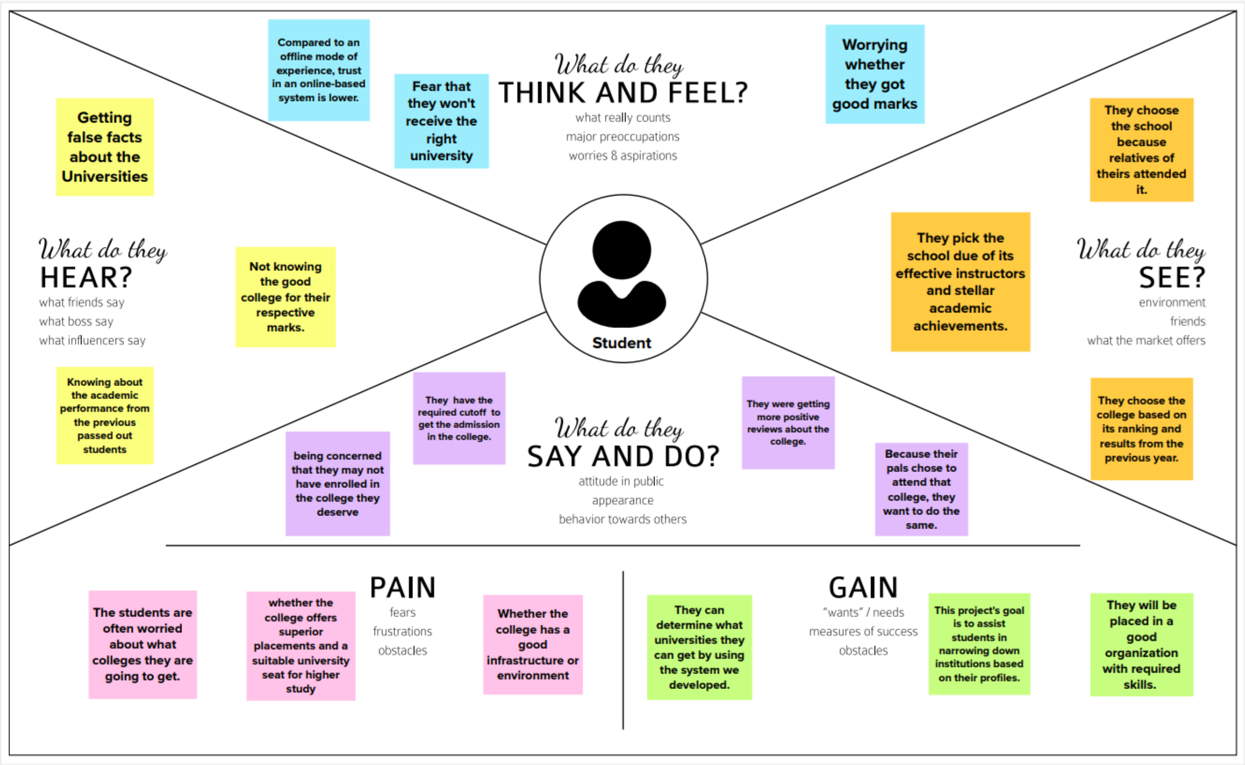
2.3.PROBLEM STATEMENT DEFINITION:

Educational organizations have always played an important and vital role in society for development and growth of any individual. There are different college prediction apps and websites being maintained contemporarily, but using themis tedious tosome extent,due to the lack of articulate information regarding colleges, and the time consumed in searching the best deserving college.

Who is the issue affecting?	Person who decides to choose university.
What are the boundaries of the problem?	Individuals who need better universities for their children.
What is the problem?	If a student received a low cutoff in the university admissions process, he would only have opportunities to attend few low?ranking institutions. Students typically showed interest in particular disciplines.
When does the problem start?	There are many well-known universities that are well?equipped. This influenced the parents to select a more comfortable university for their child.
Why is it important that we fix the problem?	It is necessary for the future of the child. Admitting his child to his comfy university is crucial.
What remedy will address this problem?	By examining the standards and amenities of universities, an automated technique is presented to help a parent find a better university for his child.
What approach was used to address the problem?	Machine learning algorithms are used to identify the university and provide advice on how to ensure that his child gets a seat there

3. IDEATION & PROPOSED SOLUTION:

3.1.EMPATHY MAP CANVAS:



What do they THINK AND FEEL?

what really counts
major preoccupations
worries & aspirations

Compared to an offline mode of experience, trust in an online-based system is lower.

Fear that they won't receive the right university

Worrying whether they got good marks

What do they HEAR?

what friends say
what boss say
what influencers say

Getting false facts about the Universities

Not knowing the good college for their respective marks.

Knowing about the academic performance from the previous passed out students

What do they SEE?

environment
friends
what the market offers

They choose the school because relatives of theirs attended it.

They pick the school due of its effective instructors and stellar academic achievements.

They choose the college based on its ranking and results from the previous year.

Student

What do they SAY AND DO?

attitude in public
appearance
behavior towards others

They have the required cutoff to get the admission in the college.

being concerned that they may not have enrolled in the college they deserve

They were getting more positive reviews about the college.

Because their pals chose to attend that college, they want to do the same.

PAIN

fears
frustrations
obstacles

The students are often worried about what colleges they are going to get.

whether the college offers superior placements and a suitable university seat for higher study

Whether the college has a good infrastructure or environment

GAIN

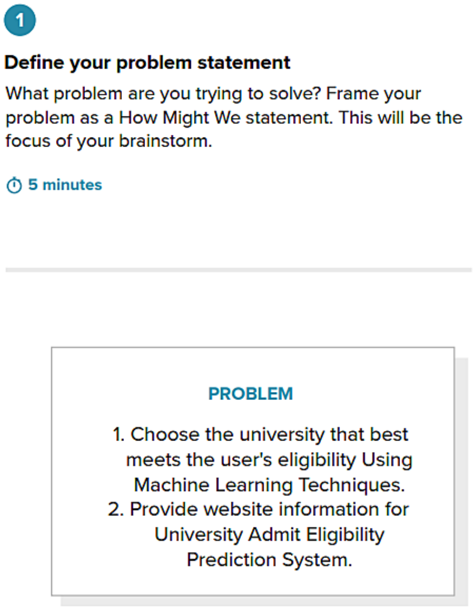
"wants" / needs
measures of success
obstacles

They can determine what universities they can get by using the system we developed.

This project's goal is to assist students in narrowing down institutions based on their profiles.

They will be placed in a good organization with required skills.

3.2.IDEATION & 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

1. Choose the university that best meets the user's eligibility Using Machine Learning Techniques.
2. Provide website information for University Admit Eligibility Prediction System.

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!

PRASANTH P

Collect Student Cutoff	Analyze the Student Cutoff	Reducing the fear of Students
Choosing best method	Enhance User Experience	Cost Efficient method
Show Accurate Results	Verifies the exact problem	Shows the list of University on the basis of student cutoff

RAMKUMAR N G

Ensure user got the correct colllege	Will available as decision making tool	Verify User and college
Make sure user provide the correct data	Shows only reputed and authenticated college	Save user time
Make sure the user got the available college	Recommending respective colleges	Provide best results based on user

NAVEEN B

gathering student information	Confirm that the user provides the correct information.	Test the Student Cutoff Data
selecting the appropriate approach	improved user experience	showcase a variety of possibilities
Try and ensure the user receives the best possible university	Endeavor to achieve more suitable predictions.	provides the list of academic institutions predicted on a student cut - off score

LOGESHWARAN S

Collecting student details	checks the availability	choosing the best college
user friendly application	show various choices	analyze the student cutoff
choosing college from nearest location	makes clarity in choosing college	show accurate results

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, 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

Category 1

- Collect Student Cutoff
- Reducing the fear of Students
- Enhance User Experience
- Shows the list of University on the basis of student cutoff
- Show Accurate Results
- Verifies the exact problem

Category 2

- Confirm that the user provides the correct information.
- Test the Student Cutoff Data
- showcase a variety of possibilities
- improved user experience
- Endeavor to achieve more suitable predictions.
- provides the list of academic institutions predicted on a student cut - off score

Category 3

- Ensure got the correct college
- Decision making tool
- Recommending respective college
- Make sure user provide the correct data
- Verify user and college
- Provide best results based on user

Category 4

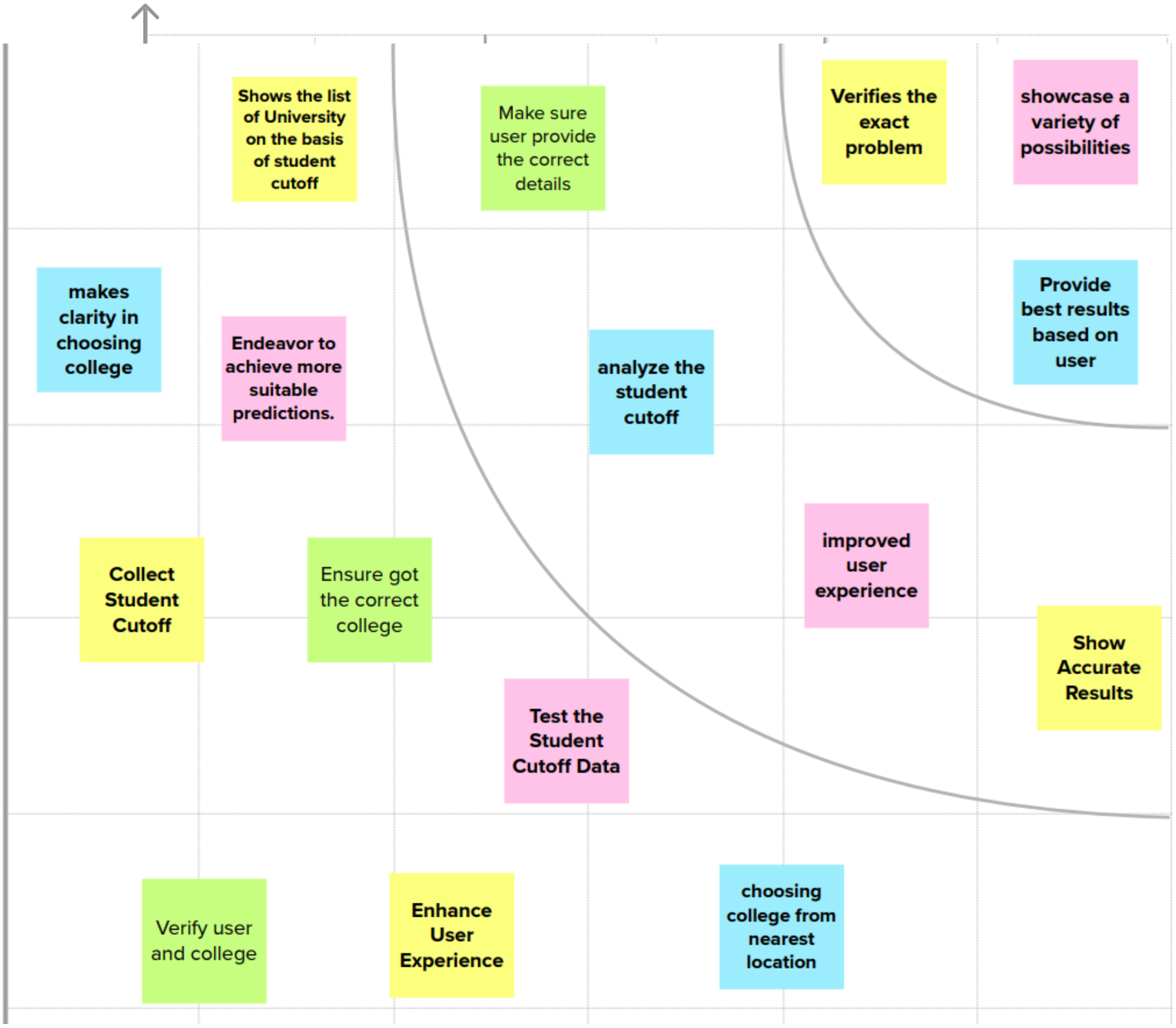
- analyze the student cutoff
- Collecting student details
- show accurate results
- makes clarity in choosing college
- show various choices
- checks the availability

4

Prioritize

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

⏱ 20 minutes



Shows the list of University on the basis of student cutoff

Make sure user provide the correct details

Verifies the exact problem

showcase a variety of possibilities

makes clarity in choosing college

Endeavor to achieve more suitable predictions.

analyze the student cutoff

Provide best results based on user

Collect Student Cutoff

Ensure got the correct college

improved user experience

Show Accurate Results

Test the Student Cutoff Data

Verify user and college

Enhance User Experience

choosing college from nearest location

3.3.PROPOSED SOLUTION:

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

1. Problem Statement (Problem to be solved) To develop a reliable University Eligibility Admit Prediction System that successfully addresses the following constraints:

✓ To gather the student's grades and interests.

✓ To share more information about the universities that the student's interests agreed with.

2. Idea / Solution description Our Project will assist UG graduates in getting into shortlisted colleges for master's programmes based on their GRE, CGPA and TOEFL scores. If the expected prediction gives them a good picture of their prospects of admission to the university. This study will also assist students who are presently preparing to have a better understanding. It will also provide students with information on the university's research prospects, admissions procedure, courses offered, and noteworthy alumni.

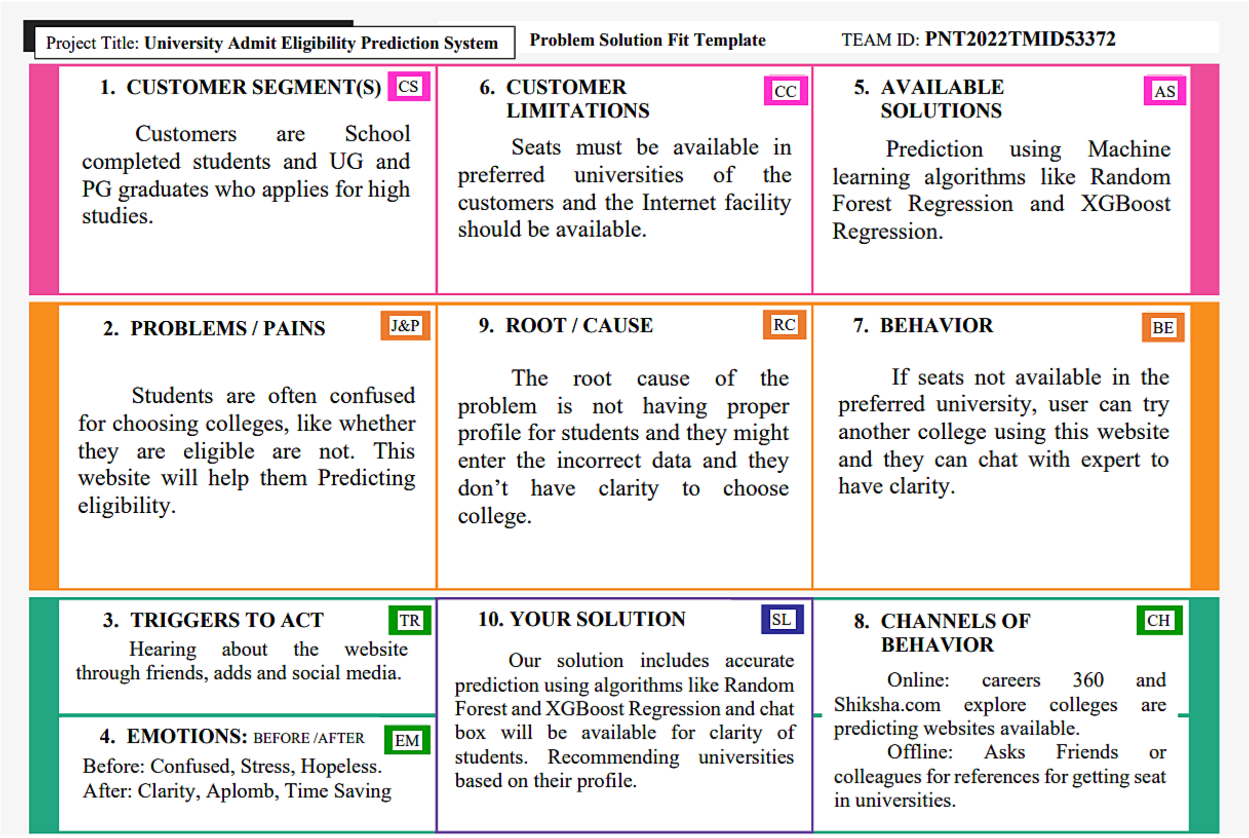
3. Novelty / Uniqueness The project website can identify numerous amenities available at universities and provide directions to the university where it is located. You can also apply for scholarships and financial aid. By using Machine learning models like Regression models, the probability of a student getting admission at a desired university is predicted.

4. Social Impact / Customer Satisfaction This solution will ease their stress about being admitted to their preferred university as well as minimize student anxiety. And this solution will deliver better outcomes for students who are deciding whether or not to attend university.

5. Business Model (Revenue Model) In addition, revenue can be generated by advertising the GRE/TOEFL coaching centres. And the University shall fund the website in order to maintain and progress it. The universities can also find a way to advertise in the website in order to increase the admissions.

6. Scalability of the Solution The solution proposed will be deployed as web-application. So, it is easily accessible by anyone who has internet services and has no specific software and hardware specifications. The dataset used for model training can be scaled according to the available universities' admission data.

3.4.PROBLEM SOLUTION FIT:



Project Title: **University Admit Eligibility Prediction System**

Problem Solution Fit Template

TEAM ID: **PNT2022TMID53372**

1. CUSTOMER SEGMENT(S) CS

Customers are School completed students and UG and PG graduates who applies for high studies.

6. CUSTOMER LIMITATIONS CC

Seats must be available in preferred universities of the customers and the Internet facility should be available.

5. AVAILABLE SOLUTIONS AS

Prediction using Machine learning algorithms like Random Forest Regression and XGBoost Regression.

2. PROBLEMS / PAINS J&P

Students are often confused for choosing colleges, like whether they are eligible are not. This website will help them Predicting eligibility.

9. ROOT / CAUSE RC

The root cause of the problem is not having proper profile for students and they might enter the incorrect data and they don't have clarity to choose college.

7. BEHAVIOR BE

If seats not available in the preferred university, user can try another college using this website and they can chat with expert to have clarity.

3. TRIGGERS TO ACT TR

Hearing about the website through friends, adds and social media.

4. EMOTIONS: BEFORE /AFTER EM

Before: Confused, Stress, Hopeless.
After: Clarity, Aplomb, Time Saving

10. YOUR SOLUTION SL

Our solution includes accurate prediction using algorithms like Random Forest and XGBoost Regression and chat box will be available for clarity of students. Recommending universities based on their profile.

8. CHANNELS OF BEHAVIOR CH

Online: careers 360 and Shiksha.com explore colleges are predicting websites available.

Offline: Asks Friends or colleagues for references for getting seat in universities.

4. REQUIREMENT ANALYSIS:

4.1.FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Interaction	User Interact with the Web page.
FR-2	User Details	Submit the documents • GRE or/and TOEFL Score Sheet • Curriculum Vitae (CV) • Statement of Purpose (SOP) • Letter of Recommendation
FR-3	User Requirements	• Upload all the relevant documents in the appropriate location in the website • Based on the uploads, the system would scrape all the necessary information

		• The list of all possible university for the candidate would be displayed based on the scraped information

4.2.NON-FUNCTIONAL REQUIREMENTS:

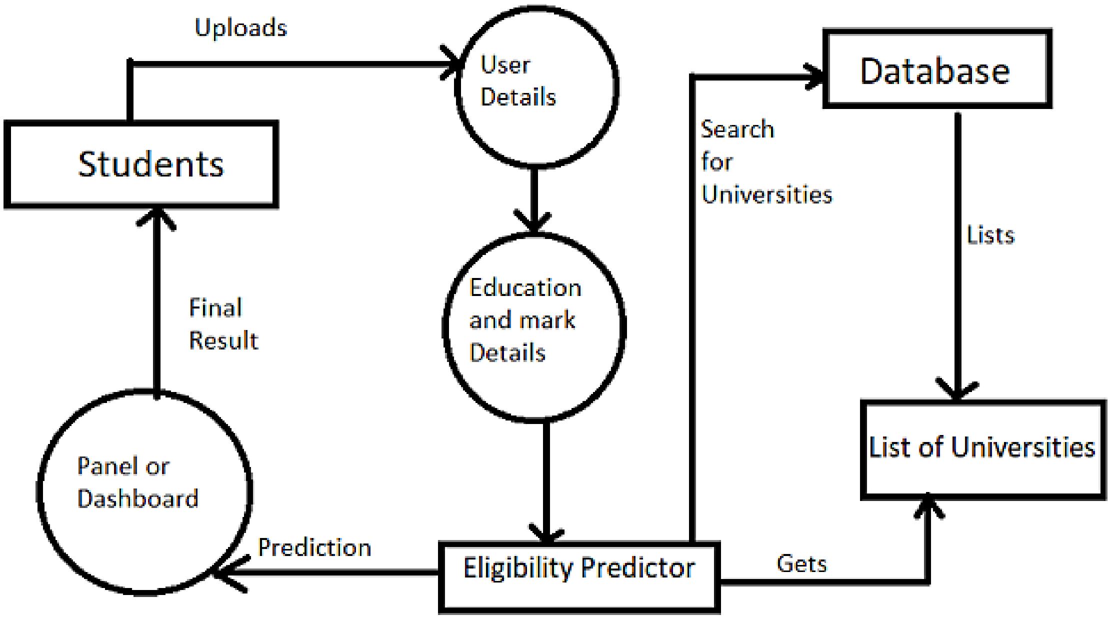
Following are the non-functional requirements of the proposed solution.

NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• The system doesn't expect any technical pre-requisite from the user i.e.; even the naïve user can access it. • User friendly. • Reduced focus on Short Term memory load Focus on Internal Locus of Control. • The page would not take a lot of time to load the content and display them (< 30 seconds).
NFR-2	**Reliability**	The system would always strive for maximum reliability due to the importance of data and damages that could be cause by incomplete and incorrect data.
NFR-3	**Performance**	• The website can efficiently handle the traffic by service the request as soon as possible. • Viewing this webpage using a 56 -kbps modem connection would not exceed 30 seconds (quantitatively, the mean time).
NFR-4	**Availability**	• Minimal data redundancy • Less prone to errors • Fast and efficient
NFR-5	**Scalability**	• Since an academic portal is crucial to the courses that use it, it is crucial that a sizable number of users be able to access the system at the same time. • The admission season is probably when the system will be under the most strain. • It must therefore be able to manage numerous concurrent users

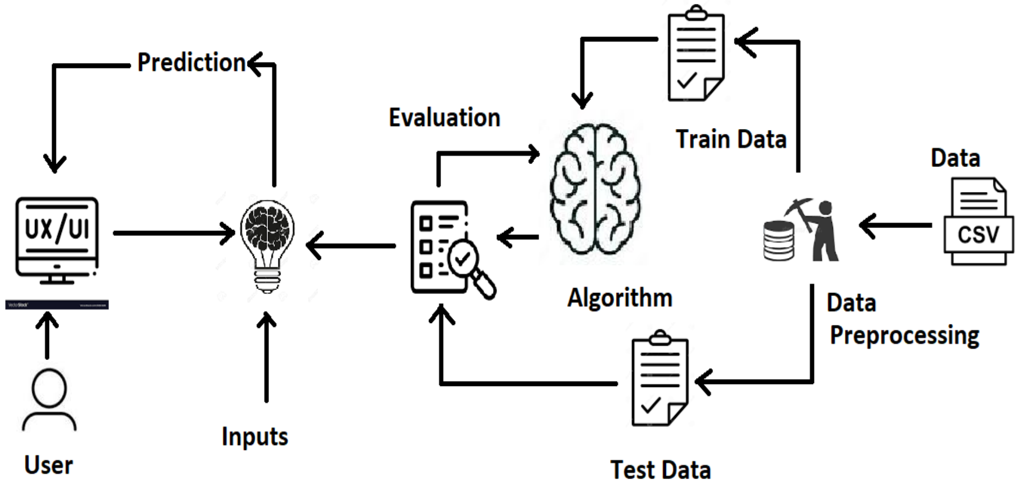
5. PROJECT DESIGN:

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:



5.3.USER STORIES

User type	Functional Requirement (Epic)	User Story No.	User Story / Task	Acceptance criteria	Priority	Release
Customer (Student)	Dashboard	USN-1	As a user, I can view the cut off marks of previous years in my dashboard.	I can access and download the files	High	Sprint?-1
		USN-2	As a user, I can view university details and their rankings.	I can only view(read-only)	Medium	Sprint?-1
		USN-3	As a user, I can review the experience of the students in the university.	I have access the review sections	Medium	Sprint?-2
		USN-4	As a user, I ca upload my documents.	I have read and write access to upoad files	High	Sprint?-1
		USN-5	As a user, I can fill out the general and education details in the form provided.	I have read and write access to the forms filled	High	Sprint?-2
	Predictor	USN-6	I can view the list of universities I am eligible to get an admission.	I can receive the final result as whether eligible or not	High	Sprint?-2
		USN-7	I can view the list of universities I am eligible	I can access the files with read-only permission	Medium	Sprint?-2

			with the same cut-off but in previous years.			
Administrator	Dashboard	USN-8	As an administrator, I can have access to update the latest updates of the universities.	I can have access to read and write the university information in the dashboard	High	Sprint?-3
		USN-9	As an Administrator, I can access any resources available in the page.	I can access the resources that are available	Medium	Sprint?-3
		USN-10	As an Administrator, I can have a track on the universities the student is eligible to get admission.	I can access the list of the universities obtained as final result	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	Interaction	USN-1	As a user, I can interact with the application by entering the Web site link.	3	High	2
Sprint-2	Choose university	USN-1	As a user, I will be able to view the list of University that the	4	Medium	4

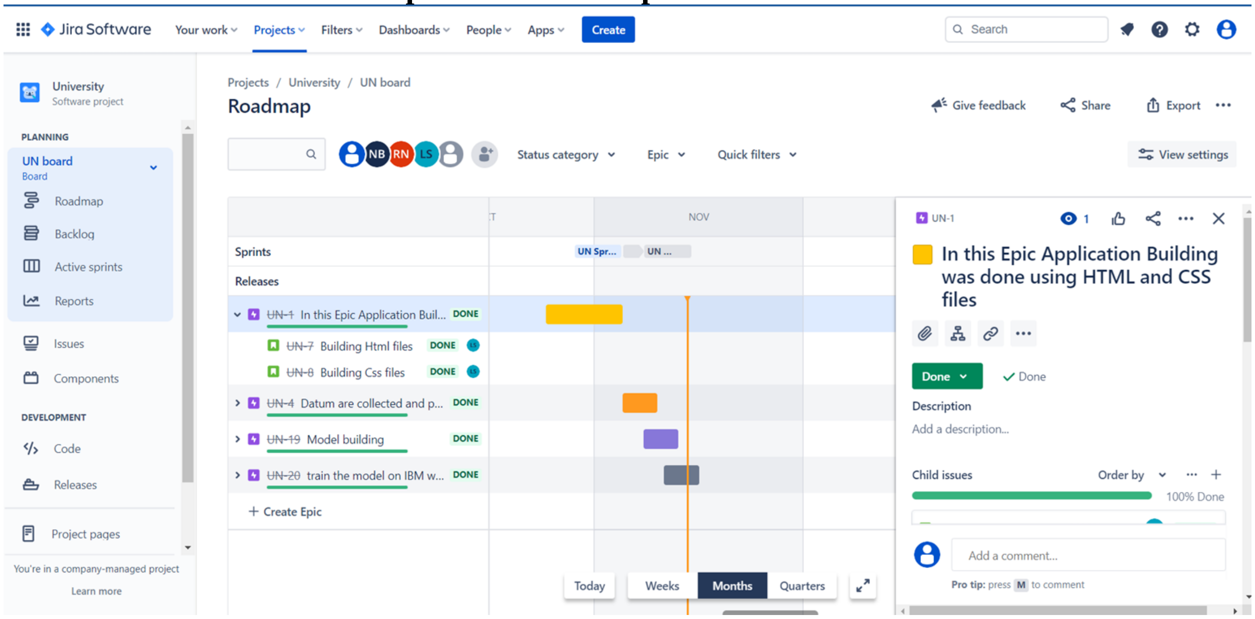
			students are eligible to apply.			
Sprint-2	Choose university	USN-1	As a user, I will be able to view the details of Admission process like date and venue of certification verification.	2	Medium	1
Sprint-3	Admission process	USN-1	As a user, I will be able to view the list of courses that the students are eligible to apply.	3	High	3
Sprint-3	Prediction	USN-1	As a admin, I can test the trained machine learning model by analyzing the user details by machine learning Algorithms.	3	High	3
Sprint-4	Output	USN-1	As a admin, I can upload the confirmation of user for the prediction into the database.	2	High	4

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	5 Days	29 Oct 2022	04 Nov 2022	20	03 Nov 2022
Sprint-2	20	4 Days	04 Nov 2022	08 Nov 2022	20	07 Nov 2022
Sprint-3	20	4 Days	08 Nov 2022	11 Nov 2022	20	10 Nov 2022
Sprint-4	20	4 Days	11 Nov 2022	14 Nov 2022	20	13 Nov 2022

6.3.REPORTS FROM JIRA

Sprint 1 - Roadmap



Roadmap

~~UN-1~~ In this Epic Application Buil... DONE
~~UN-7~~ Building Html files DONE
~~UN-8~~ Building Css files DONE
~~UN-4~~ Datum are collected and p... DONE
~~UN-19~~ Model building DONE
~~UN-20~~ train the model on IBM w... DONE
+ Create Epic

UN-1

In this Epic Application Building was done using HTML and CSS files

Done ✓ Done

Description

Add a description...

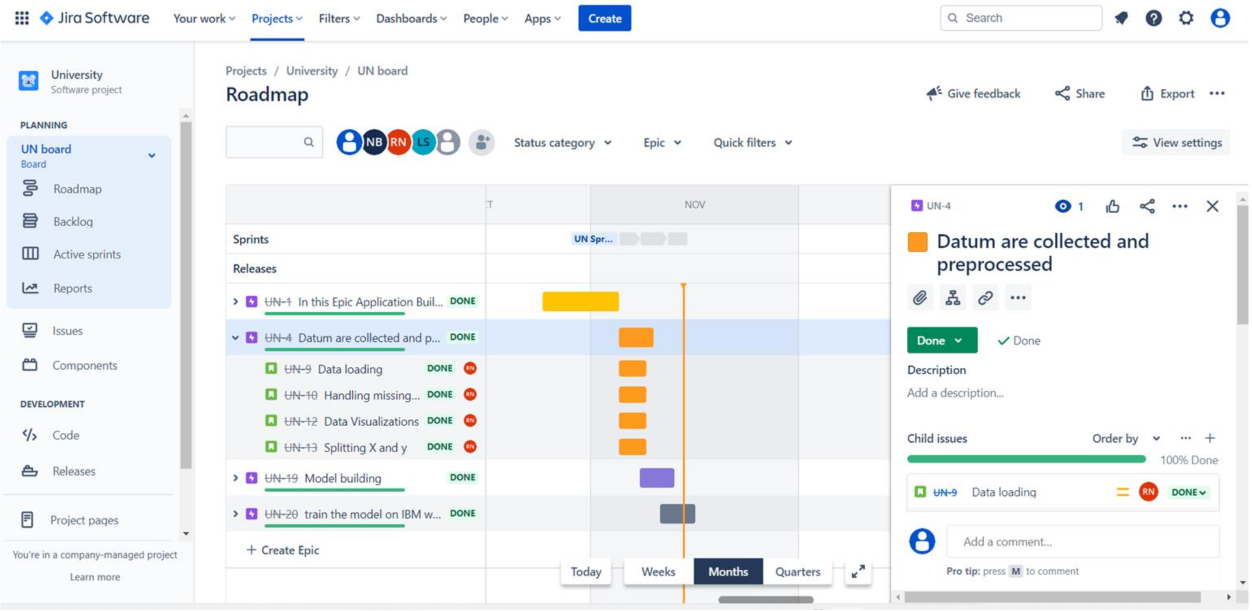
Child issues Order by

100% Done

Add a comment...

Pro tip: press M to comment

Sprint 2 - Roadmap



Roadmap

~~UN-1~~ In this Epic Application Buil... DONE
~~UN-4~~ Datum are collected and p... DONE
~~UN-9~~ Data loading DONE
~~UN-10~~ Handling missing... DONE
~~UN-12~~ Data Visualizations DONE
~~UN-13~~ Splitting X and y DONE
~~UN-19~~ Model building DONE
~~UN-20~~ train the model on IBM w... DONE
+ Create Epic

UN-4

Datum are collected and preprocessed

Done ✓ Done

Description

Add a description...

Child issues Order by

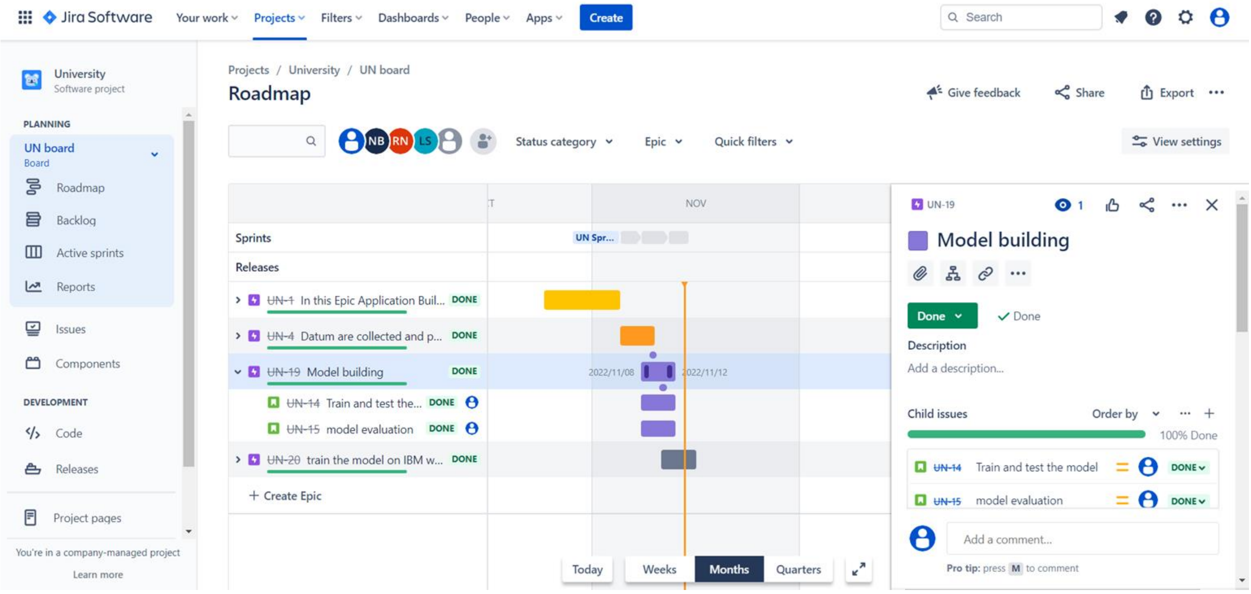
100% Done

~~UN-9~~ Data loading RN DONE

Add a comment...

Pro tip: press M to comment

Sprint 3 - Roadmap



~~UN-1~~ In this Epic Application Buil...	DONE
~~UN-4~~ Datum are collected and p...	DONE
~~UN-19~~ Model building	DONE
~~UN-14~~ Train and test the...	DONE
~~UN-15~~ model evaluation	DONE
~~UN-20~~ train the model on IBM w...	DONE

+ Create Epic

2022/11/08 2022/11/12

UN-19

Model building

Done ✓ Done

Description

Add a description...

Child issues Order by

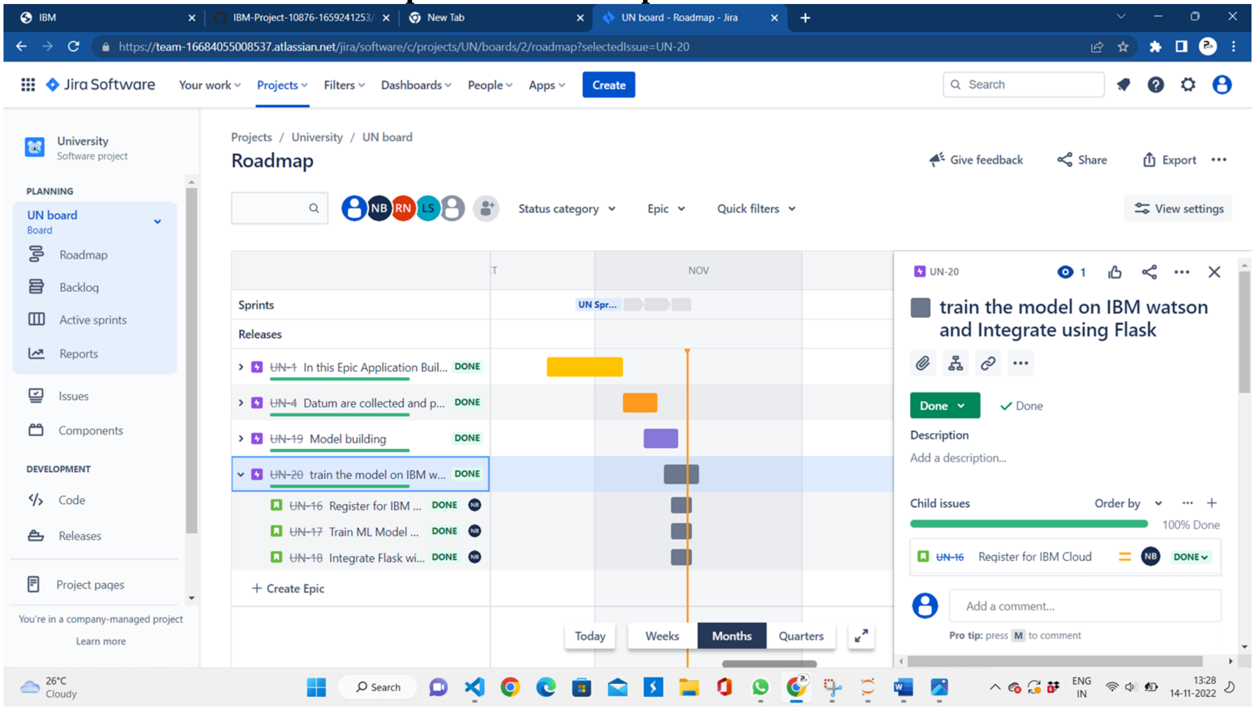
100% Done

~~UN-14~~	Train and test the model	DONE
~~UN-15~~	model evaluation	DONE

Add a comment...

Pro tip: press M to comment

Sprint 3 - Roadmap



~~UN-1~~ In this Epic Application Buil...	DONE
~~UN-4~~ Datum are collected and p...	DONE
~~UN-19~~ Model building	DONE
~~UN-20~~ train the model on IBM w...	DONE
~~UN-16~~ Register for IBM ...	DONE
~~UN-17~~ Train ML Model ...	DONE
~~UN-18~~ Integrate Flask wi...	DONE

+ Create Epic

UN-20

train the model on IBM watson and Integrate using Flask

Done ✓ Done

Description

Add a description...

Child issues Order by

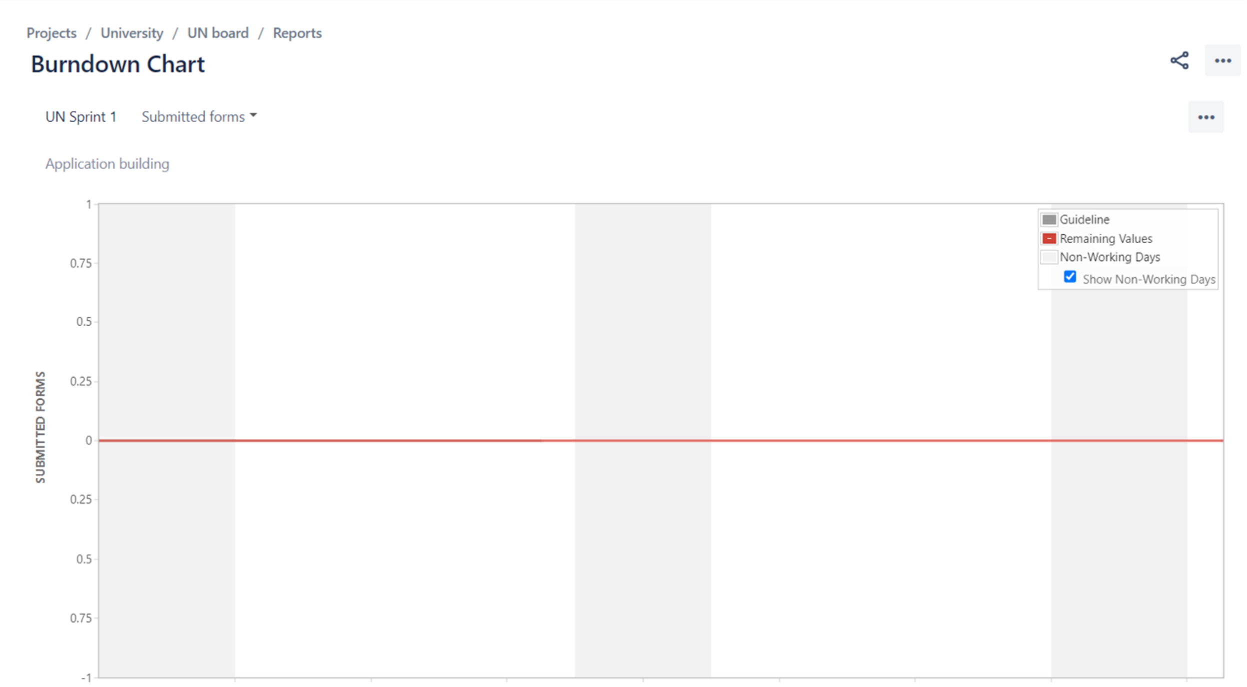
100% Done

~~UN-16~~	Register for IBM Cloud	DONE

Add a comment...

Pro tip: press M to comment

Sprint 4 - Roadmap



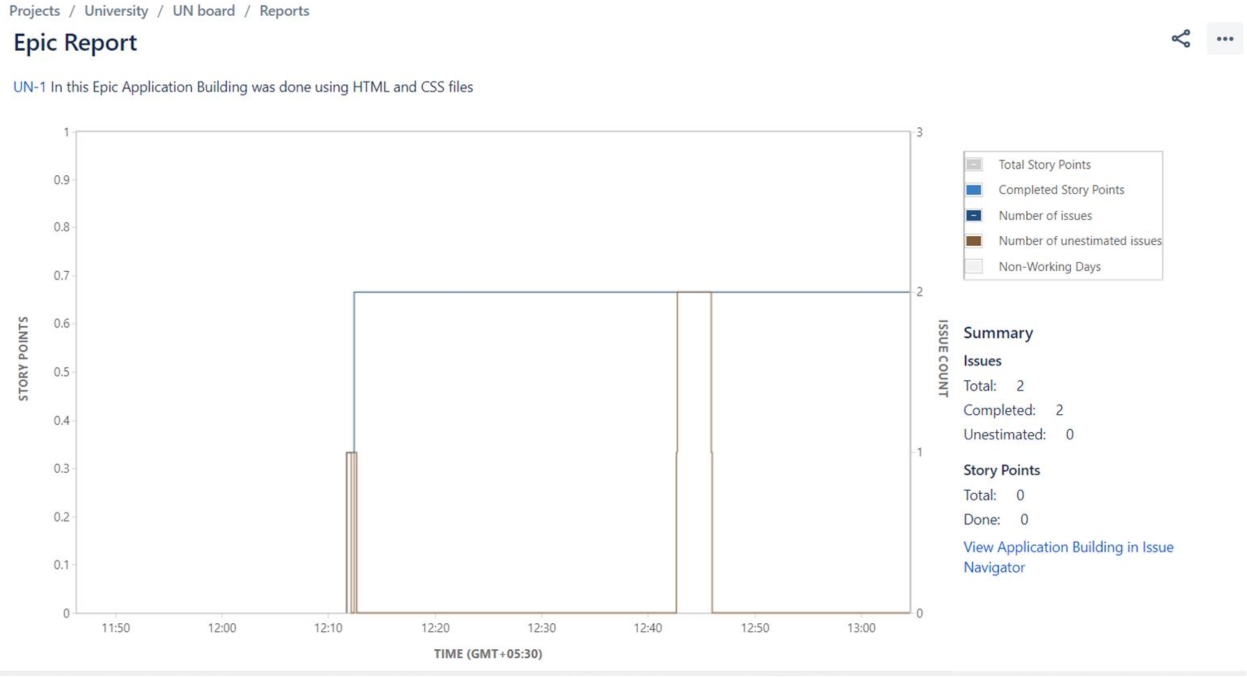
Projects / University / UN board / Reports

Burndown Chart

UN Sprint 1   Submitted forms ▾

Application building

EPIC Report:



Projects / University / UN board / Reports

Epic Report

UN-1 In this Epic Application Building was done using HTML and CSS files

Summary

Issues

Total: 2

Completed: 2

Unestimated: 0

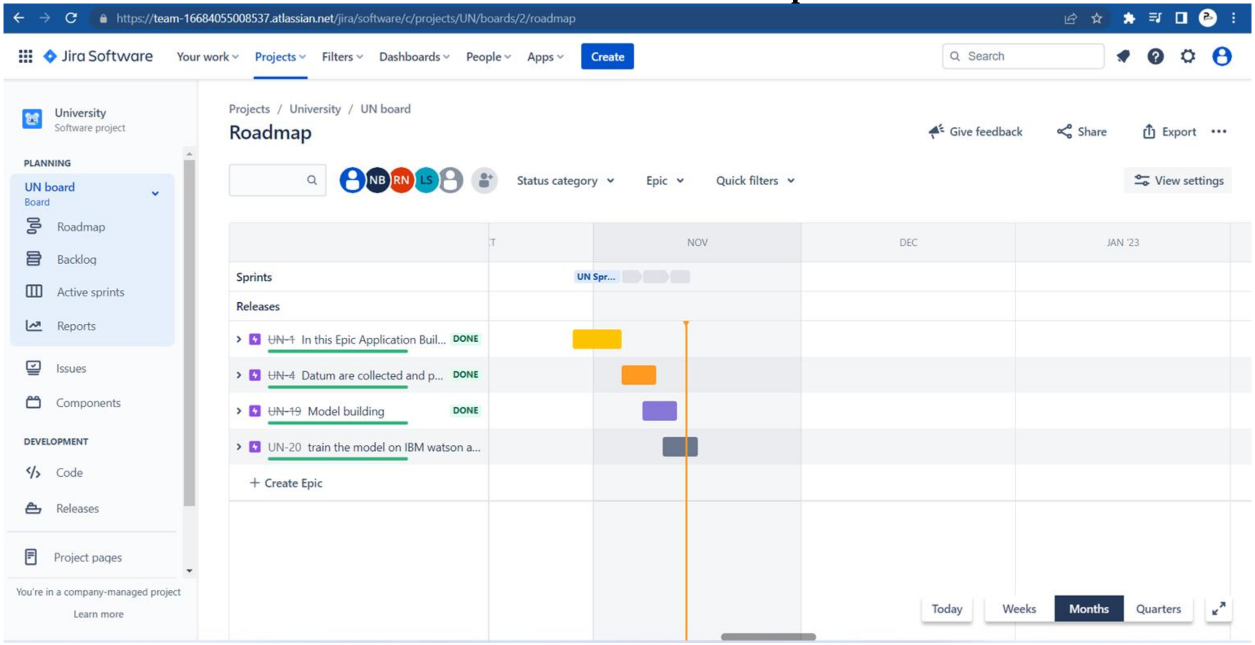
Story Points

Total: 0

Done: 0

View Application Building in Issue Navigator

After Release Roadmap:



~~UN-1~~ In this Epic Application Buil... DONE

~~UN-4~~ Datum are collected and p... DONE

~~UN-19~~ Model building DONE

UN-20 train the model on IBM watson a...

7. CODING & SOLUTIONING:

7.1.RANDOM FOREST REGRESSOR

Coding:

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
X = df[["GRE Score","TOEFL Score","University Rating","SOP","LOR ","CGPA"]]
y = df["Chance of Admit "]
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=42)
ran_for_reg = RandomForestRegressor(n_estimators=100,random_state=42)
ran_for_reg.fit(X_train,y_train)
y_pred_rfr = ran_for_reg.predict(X_test)
r2_score_rfr = r2_score(y_test,y_pred_rfr)
print("Random Forest Regression's Score = {:.3f}".format(r2_score_rfr))
```

Solutioning:

```
Random Forest Regression's Score = 0.804
```

7.2.KNN REGRESSOR

Coding:

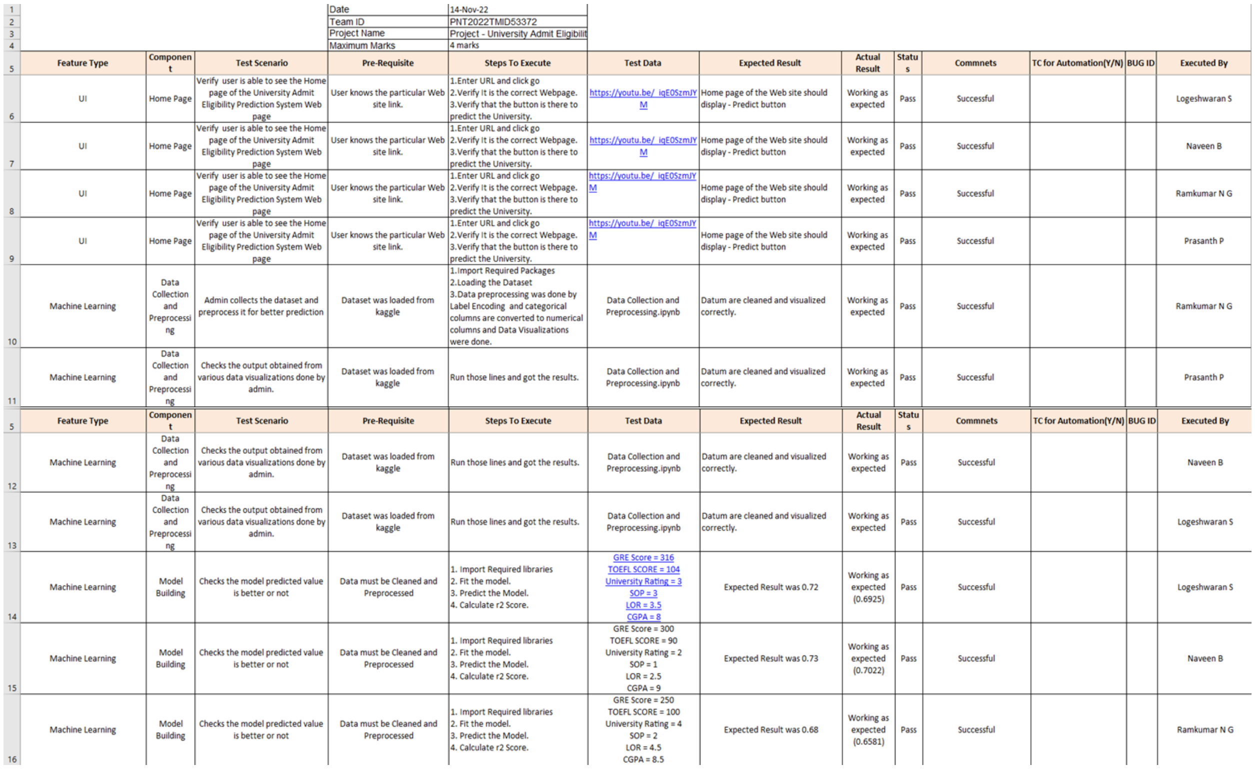
```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import r2_score
X = df[["GRE Score","TOEFL Score","University Rating","SOP","LOR ","CGPA"]]
y = df["Chance of Admit "]
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=42)
knn_model = KNeighborsRegressor(n_neighbors=3)
knn_model.fit(X_train,y_train)
y_pred_knn = knn_model.predict(X_test)
r2_score_knn = r2_score(y_test,y_pred_knn)
print("Random Forest Regression's Score = {:.3f}".format(r2_score_knn))
```

Solutioning:

```
KNeighbors Regressor's Score = 0.642
```

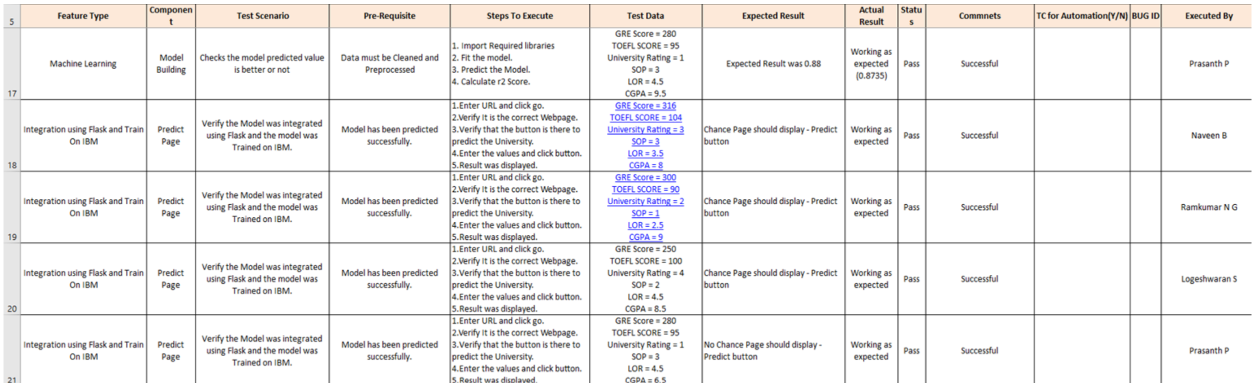
8. TESTING:

8.1.TEST CASES:



Date	14-Nov-22
Team ID	PNT2022TMID53372
Project Name	Project - University Admit Eligibilit
Maximum Marks	4 marks

Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
UI	Home Page	Verify user is able to see the Home page of the University Admit Eligibility Prediction System Web page	User knows the particular Web site link.	1.Enter URL and click go 2.Verify It is the correct Webpage. 3.Verify that the button is there to predict the University.	https://youtu.be/_iqE0SzmJYM	Home page of the Web site should display - Predict button	Working as expected	Pass	Successful			Logeshwaran S
UI	Home Page	Verify user is able to see the Home page of the University Admit Eligibility Prediction System Web page	User knows the particular Web site link.	1.Enter URL and click go 2.Verify It is the correct Webpage. 3.Verify that the button is there to predict the University.	https://youtu.be/_iqE0SzmJYM	Home page of the Web site should display - Predict button	Working as expected	Pass	Successful			Naveen B
UI	Home Page	Verify user is able to see the Home page of the University Admit Eligibility Prediction System Web page	User knows the particular Web site link.	1.Enter URL and click go 2.Verify It is the correct Webpage. 3.Verify that the button is there to predict the University.	https://youtu.be/_iqE0SzmJYM	Home page of the Web site should display - Predict button	Working as expected	Pass	Successful			Ramkumar N G
UI	Home Page	Verify user is able to see the Home page of the University Admit Eligibility Prediction System Web page	User knows the particular Web site link.	1.Enter URL and click go 2.Verify It is the correct Webpage. 3.Verify that the button is there to predict the University.	https://youtu.be/_iqE0SzmJYM	Home page of the Web site should display - Predict button	Working as expected	Pass	Successful			Prasanth P
Machine Learning	Data Collection and Preprocessing	Admin collects the dataset and preprocess it for better prediction	Dataset was loaded from kaggle	1.Import Required Packages 2.Loading the Dataset 3.Data preprocessing was done by Label Encoding and categorical columns are converted to numerical columns and Data Visualizations were done.	Data Collection and Preprocessing.ipynb	Datum are cleaned and visualized correctly.	Working as expected	Pass	Successful			Ramkumar N G
Machine Learning	Data Collection and Preprocessing	Checks the output obtained from various data visualizations done by admin.	Dataset was loaded from kaggle	Run those lines and got the results.	Data Collection and Preprocessing.ipynb	Datum are cleaned and visualized correctly.	Working as expected	Pass	Successful			Prasanth P
Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
Machine Learning	Data Collection and Preprocessing	Checks the output obtained from various data visualizations done by admin.	Dataset was loaded from kaggle	Run those lines and got the results.	Data Collection and Preprocessing.ipynb	Datum are cleaned and visualized correctly.	Working as expected	Pass	Successful			Naveen B
Machine Learning	Data Collection and Preprocessing	Checks the output obtained from various data visualizations done by admin.	Dataset was loaded from kaggle	Run those lines and got the results.	Data Collection and Preprocessing.ipynb	Datum are cleaned and visualized correctly.	Working as expected	Pass	Successful			Logeshwaran S
Machine Learning	Model Building	Checks the model predicted value is better or not	Data must be Cleaned and Preprocessed	1. Import Required libraries 2. Fit the model. 3. Predict the Model. 4. Calculate r2 Score.	GRE Score = 316 TOEFL SCORE = 104 University Rating = 3 SOP = 3 LOR = 3.5 CGPA = 8	Expected Result was 0.72	Working as expected (0.6925)	Pass	Successful			Logeshwaran S
Machine Learning	Model Building	Checks the model predicted value is better or not	Data must be Cleaned and Preprocessed	1. Import Required libraries 2. Fit the model. 3. Predict the Model. 4. Calculate r2 Score.	GRE Score = 300 TOEFL SCORE = 90 University Rating = 2 SOP = 1 LOR = 2.5 CGPA = 9	Expected Result was 0.73	Working as expected (0.7022)	Pass	Successful			Naveen B
Machine Learning	Model Building	Checks the model predicted value is better or not	Data must be Cleaned and Preprocessed	1. Import Required libraries 2. Fit the model. 3. Predict the Model. 4. Calculate r2 Score.	GRE Score = 250 TOEFL SCORE = 100 University Rating = 4 SOP = 2 LOR = 4.5 CGPA = 8.5	Expected Result was 0.68	Working as expected (0.6581)	Pass	Successful			Ramkumar N G



5	Feature Type	Componen t	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Statu s	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
17	Machine Learning	Model Building	Checks the model predicted value is better or not	Data must be Cleaned and Preprocessed	1. Import Required libraries 2. Fit the model. 3. Predict the Model. 4. Calculate r2 Score.	GRE Score = 280 TOEFL SCORE = 95 University Rating = 1 SOP = 3 LOR = 4.5 CGPA = 9.5	Expected Result was 0.88	Working as expected (0.8735)	Pass	Successful			Prasanth P
18	Integration using Flask and Train On IBM	Predict Page	Verify the Model was integrated using Flask and the model was Trained on IBM.	Model has been predicted successfully.	1.Enter URL and click go. 2.Verify It is the correct Webpage. 3.Verify that the button is there to predict the University. 4.Enter the values and click button. 5.Result was displayed.	GRE Score = 316 TOEFL SCORE = 104 University Rating = 3 SOP = 3 LOR = 3.5 CGPA = 8	Chance Page should display - Predict button	Working as expected	Pass	Successful			Naveen B
19	Integration using Flask and Train On IBM	Predict Page	Verify the Model was integrated using Flask and the model was Trained on IBM.	Model has been predicted successfully.	1.Enter URL and click go. 2.Verify It is the correct Webpage. 3.Verify that the button is there to predict the University. 4.Enter the values and click button. 5.Result was displayed.	GRE Score = 300 TOEFL SCORE = 90 University Rating = 2 SOP = 1 LOR = 2.5 CGPA = 9	Chance Page should display - Predict button	Working as expected	Pass	Successful			Ramkumar N G
20	Integration using Flask and Train On IBM	Predict Page	Verify the Model was integrated using Flask and the model was Trained on IBM.	Model has been predicted successfully.	1.Enter URL and click go. 2.Verify It is the correct Webpage. 3.Verify that the button is there to predict the University. 4.Enter the values and click button. 5.Result was displayed.	GRE Score = 250 TOEFL SCORE = 100 University Rating = 4 SOP = 2 LOR = 4.5 CGPA = 8.5	Chance Page should display - Predict button	Working as expected	Pass	Successful			Logeshwaran S
21	Integration using Flask and Train On IBM	Predict Page	Verify the Model was integrated using Flask and the model was Trained on IBM.	Model has been predicted successfully.	1.Enter URL and click go. 2.Verify It is the correct Webpage. 3.Verify that the button is there to predict the University. 4.Enter the values and click button. 5.Result was displayed.	GRE Score = 280 TOEFL SCORE = 95 University Rating = 1 SOP = 3 LOR = 4.5 CGPA = 6.5	No Chance Page should display - Predict button	Working as expected	Pass	Successful			Prasanth P

8.2.USER ACCEPTANCE TESTING:

Purpose of Document

The purposeof this documentis to briefly explain the test coverageand open issues of the [University Admit Eligibility Predictor] project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

This report shows the number of resolvedor closed bugs at each severitylevel, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	3	1	2	17
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	40
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	**24**	**13**	**12**	**25**	**78**

Test Case Analysis

This reportshows the numberof test cases that have passed, failed,and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	50	0	0	50
Outsource Shipping	3	0	0	3
Exception Reporting	8	0	0	8
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS:

9.1.PERFORMANCE METRICS:

R2 score is an indicator of accuracy of Regression Models, and the accuracy is measured as clost to 1 of its value. Therefore, as seen, Random Forest Regression Model is better than KNN Regression on this dataset when comparing their R2 scores.

Coding:

```
r2_score_rfr = r2_score(y_test,y_pred_rfr)
print("Random Forest Regression's Score = {:.3f}".format(r2_score_rfr))
r2_score_knn = r2_score(y_test,y_pred_knn)
print("Random Forest Regression's Score = {:.3f}".format(r2_score_knn))
```

Solutioning:

```
Random Forest Regression's Score = 0.804
KNeighbors Regressor's Score = 0.642
```

10. ADVANTAGES & DISADVANTAGES:

10.1. Advantages:

✓ It helps student for making decision for choosing a right college.

- ✓ Here the chance of occurrence of error is less when compared with the existing system.
- ✓ It is fast, efficient and reliable.
- ✓ Avoids data redundancy and inconsistency.
- ✓ Very user-friendly.
- ✓ Easy accessibility of data.

10.2. Disadvantages:

- Required active internet connection.
- System will provide inaccurate results if data entered incorrectly.

11. CONCLUSION:

In this project, machine learning models were performed to predict the opportunity of a student to get admitted to a master's program. The machine learning models included are K-Nearest Neighbor and Random Forest. Experiments show that the Random Forest Regression surpasses K-Nearest Neighbor.

12. FUTURE SCOPE:

As for the future work, more models can be conducted on more datasets to learn the model that gives the best performance.

The future scope of this project is very broad.

Few of them are:

- This can be implemented in less time for proper admission process.
- This can be accessed anytime anywhere, since it is a web application provided only an internet connection.
- The user had not need to travel a long distance for the admission and his/her time is also saved as a result of this automated system.

13. APPENDIX

13.1. SOURCE CODE:

Model Prediction.py:

## 1. Importing the required libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

## 1.1 Data Loading

```
df = pd.read_csv("Dataset/Admission_Predict.csv")
df.head()
df.shape
df.info()
df.isnull().any()
```

# 2. Data Visualizations

## 2.1 Univariate Analysis

### 2.1.1 Distribution Plot

# DISTRIBUTION OF GRE SCORE:

```
sns.displot(df['GRE Score'])
sns.distplot(df['GRE Score'])
```

# DISTRIBUTION OF TOEFL SCORE

```
sns.displot(df['TOEFL Score'])
```

# DISTRIBUTION OF TOEFL SCORE

```
sns.distplot(df['TOEFL Score'])
```

# DISTRIBUTION OF SOP

```
sns.distplot(df['SOP'])
```

# DISTRIBUTION OF UNIVERSITY RATING

```
sns.displot(df['University Rating'])
```

# DISTRIBUTION OF UNIVERSITY RATING

```
sns.distplot(df['University Rating'])
```

# DISTRIBUTION OF LOR

```
sns.distplot(df['LOR '])
```

# DISTRIBUTION OF CGPA

```
sns.distplot(df['CGPA'])
```

```
# DISTRIBUTION OF RESEARCH
    sns.distplot(df['Research'])

# DISTRIBUTION OF CHANCE OF ADMIT
    sns.distplot(df["Chance of Admit "])

### 2.1.3 Pie Plot
# Pie plot for UNIVERSITY RATING
    plt.pie(df['University Rating'].value_counts(),[0,0,0,0,0.2],labels=[1,2,3,4,5],
autopct="%1.1f%%", colors=["red", 'orange',"yellow",'blue','pink'])
    plt.title("University Rating")

# Pie plot for Research
    plt.pie(df['Research'].value_counts(),[0,0],labels=[0,1],autopct="%1.1f%%",color
s=["pink",'blue'])
    plt.title("Research")

### 2.1.4 Bar Plot
# Bar plot for LOR
    sns.barplot(df['LOR '].value_counts().index,df['LOR '].value_counts())

# Bar plot for SOP
    sns.barplot(df['SOP'].value_counts().index,df['SOP'].value_counts())



## 2.2 Bi-Variate Analysis
### 2.2.1 LINE PLOT
    sns.lineplot(df['GRE Score'],df['Chance of Admit '])
    sns.lineplot(df['TOEFL Score'],df['Chance of Admit '])

### 2.2.2 Scatter Plot
# Scatterplot for CGPA and Chance of Admit
    sns.scatterplot(df['CGPA'],df['Chance of Admit '])

### 2.2.3 relplot
    sns.relplot(data=df,x="GRE Score",y="Chance of Admit ",hue="Research")
    plt.title("GRE Score vs Chance of Admit")
    plt.show()

    sns.relplot(data=df,x="TOEFL Score",y="Chance of Admit ",hue="Research",
kind="line")
    plt.title("TOEFL vs Chance of Admit")
    plt.show()

    sns.relplot(data=df,x="CGPA",y="Chance of Admit ",hue="Research")
```

```
plt.title("GRE Score vs Chance of Admit")
plt.show()

sns.relplot(data=df,x="SOP",y="Chance of Admit ",hue="Research", kind="line")
plt.title("GRE Score vs Chance of Admit")
plt.show()

sns.relplot(data=df,x="LOR ",y="Chance of Admit ",hue="Research",kind="line")
plt.title("GRE Score vs Chance of Admit")
plt.show()
```

### 2.2.4 Bar Plot

```
sns.barplot(data=df,x="University Rating",y="Chance of Admit ")
plt.title("University Rating vs Chance of Admit")
plt.show()
```

## 2.3 Multi-Variate Analysis

### 2.3.1 Histogram

```
df.hist(bins = 30, figsize = (20,20), color = 'orange')
```

### 2.3.2 Pair Plot

```
sns.pairplot(df)
```

# 3 Data Analysis

```
df.head()
df.tail()
```

## 3.1 Drop the Serial No Column

```
df.drop("Serial No.",axis=1,inplace=True)
df.head()
```

## 3.2 Checking for Null values

```
df.isnull().sum()
```

## 3.3 Gettig Information about the dataframe

```
df.info()
```

## 3.4 Statistical Summary of the dataframe

```
df.describe()
```

# 3.5 To find the correlation of columns

```
corr_matrix=df.corr()
corr_matrix
```

#plotting the correlation matrix as a heatmap

```
    fig = plt.figure(figsize=(12,8))
    sns.heatmap(corr_matrix,annot=True)
    plt.show()
```

# 4. Model Building

## 4.1 Importing the required libraries for Regression Model

```
    from sklearn.model_selection import train_test_split
    from sklearn.neighbors import KNeighborsRegressor
    from sklearn.tree import DecisionTreeRegressor
    from sklearn.ensemble import RandomForestRegressor
    from sklearn.metrics import r2_score
```

## 4.2 Split the dataset into dependent column and independent column

```
    X = df[["GRE Score","TOEFL Score","University Rating","SOP","LOR ","CGPA"]]
    y = df["Chance of Admit "]
    X.head()
    y.head()
```

## 4.3 Spliting the dataset into training and testing data

```
    X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2, random_state=42)
```

## 4.4 Regression Models

### 4.4.1 Random Forest Regression

```
    ran_for_reg = RandomForestRegressor(n_estimators=100,random_state=42)
    ran_for_reg.fit(X_train,y_train)
    y_pred_rfr = ran_for_reg.predict(X_test)
    r2_score_rfr = r2_score(y_test,y_pred_rfr)
    print("Random Forest Regression's Score = {:.3f}".format(r2_score_rfr))
```

### 4.4.2 KNN Regression

```
    knn_model = KNeighborsRegressor(n_neighbors=3)
    knn_model.fit(X_train,y_train)
    y_pred_knn = knn_model.predict(X_test)
    r2_score_knn = r2_score(y_test,y_pred_knn)
    print("KNeighbors Regressor's Score = {:.3f}".format(r2_score_knn))
```

# 5. Conclusion

### R2 score is an indicator of accuracy of Regression Models, and the accuracy is measured as clost to 1 of its value. Therefore, as seen, Random Forest Regression Model is better than KNN Regression on this dataset when comparing their R2 scores.

```
import pickle
pickle.dump(ran_for_reg,open('university.pkl','wb'))
```

```
ran_for_reg.predict([[280,95,1,3,4.5,6.5]])
```

app.py

```
import flask
from flask import request, render_template
from flask_cors import CORS
import joblib
import requests
import json

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "dbmlwXit_00dVgPiTfK0wIFqoa5WntN5P62VAiloe-81"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
app= flask.Flask(__name__,
    static_url_path='',
    static_folder='static',
    template_folder='templates'
 )
CORS(app)

@app.route('/',methods=['GET', 'POST'])
def getHomePage():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
    # X = df[["GRE Score","TOEFL Score","University Rating","SOP","LOR ","CGPA"]]
    GRE_score=float(request.form['gre'])
    TOEFL_score=float(request.form['toefl_score'])
    University_rating=float(request.form['university'])
    sop =float(request.form['sop'])
    lor =float(request.form['lor'])
    cgpa =float(request.form['cgpa'])
    X=[[GRE_score,TOEFL_score,University_rating,sop,cgpa]]
    print(X)
    # model=joblib.load('university.pkl')
    # result=model.predict(X)[0]
    payload_scoring = {"input_data": [{"fields": [  "GRE Score",
                            "TOEFL Score",
                            "University Rating",
                            "SOP",
                                "LOR ",
                                "CGPA"
                            ],
                    "values": [
                        [
                            GRE_score,
                            TOEFL_score,
                            University_rating,
                            sop,
                            lor,
                            cgpa
                        ]
                    ]
                    }]
                }
    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/bccfd93c-32ce-4045-b3db-eb65586ecfe0/predictions?version=2022-11-08', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    result=response_scoring.json()
    print(result['predictions'][0]['values'][0][0])
    result=result['predictions'][0]['values'][0][0]
    result=int(result*100)
    print(result)
    if result>50:
        return render_template('chance.html',result=result)
    else:
        return render_template('nochance.html',result=result)
if __name__ == '__main__':
    app.run(debug=True)
```

13.2. GITHUB & PROJECT DEMO LINK:

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-10876-1659241253

PROJECT DEMO LINK:

https://youtu.be/_iqE0SzmJYM

13.3. PROJECT OVERVIEW:

Project Name: University Admit Eligibility Predictor.
Team ID: PNT2022TMID53372
Project type: Web Application
Developers: Naveen B, Logeshwaran S, Prasanth P, Ramkumar NG
Languages used: Python, HTML, CSS
Development Platform: IBM WATSON STUDIO.
Data Set Used: Admission_Predict Dataset
(https://www.kaggle.com/rishal005/admission-predict)

13.4. PURPOSE:

University and College research being one part of the university application process is itself an arduous and lengthy task. This issue being a big problem for students have not been solved till now. There are recognized sites which filters the best universities and colleges based on the location, tuition fees, major and degree but none of them have use machine learning algorithm to solve the issue. Hence, we have done this research project to solve that issue to some extent with the use of data mining techniques.