



AITOR



AITor

Let's Seek the Success

AITor

EDUCATION PLATFORM

A PERSONALIZED STUDENT
PERFORMANCE ANALYZER
AND RECOMMENDATION SYSTEM

2022-017



2022-017



Meet Our Team



AKILA LIYANAGE

IT19120812



MAHENDRA THAMMITA

IT19120362



NETHSARA LIYANAGE

IT19188546

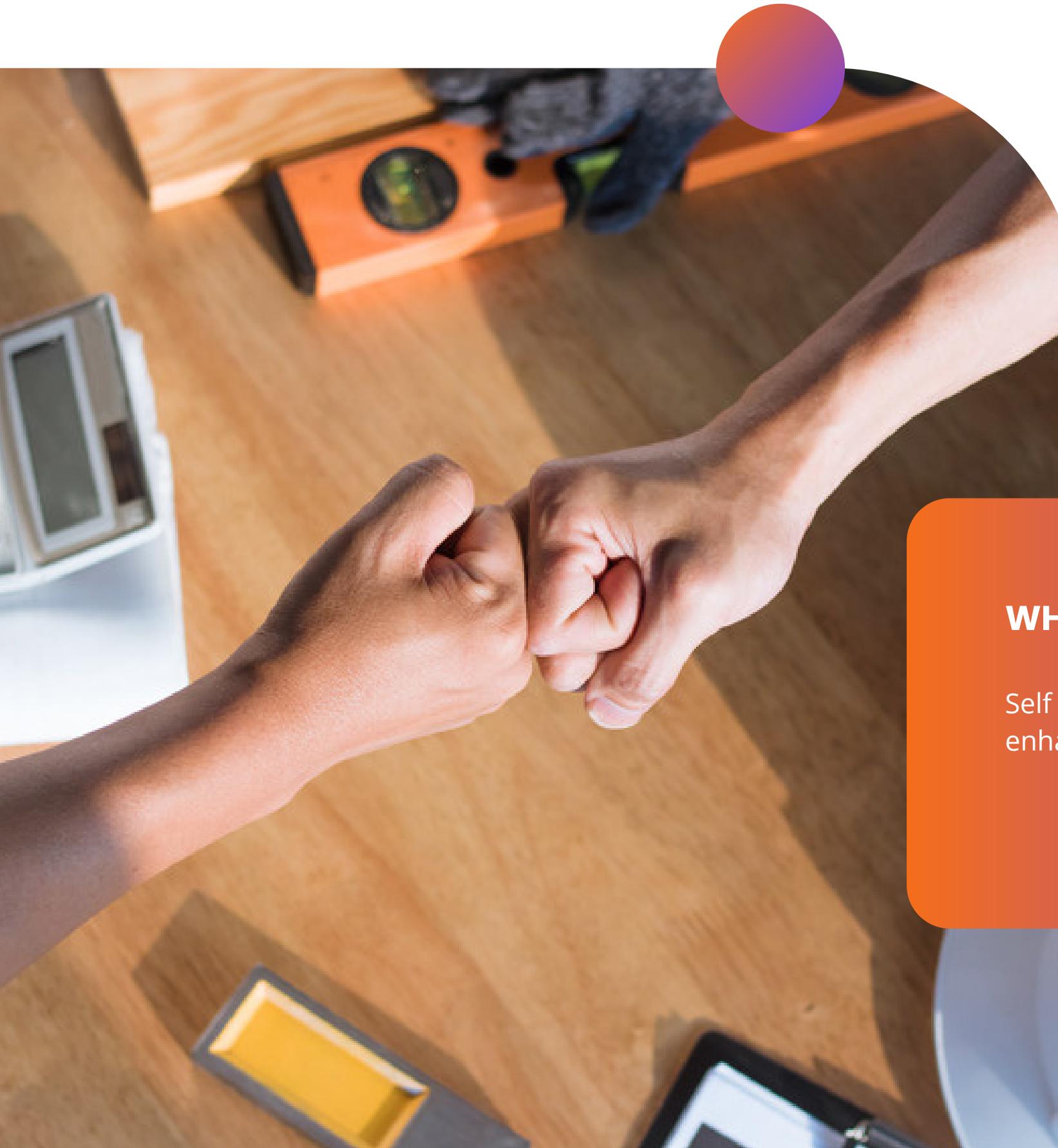


UDITHA JANADARA

IT19138114



2022-017



PROJECT INTRODUCTION

WHAT WE ARE BUILDING

Self learning educational platform enhanced through Artificial Inteligence

WHY WE ARE BUILDING THIS?

- Urged the demand for online education with the rise of COVID 19
- No specific way to statistically analyse the students performance in the distance learning procedure



RESEARCH PROBLEM

01

HOW TO INCREASE THE STUDENT ENGAGEMENT IN DISTANCE LEARNING?

02

HOW TO PROVIDE THE BEST LEARNING MATERIALS TO THE STUDENTS?

03

HOW TO ANALYZE THE STUDENT PERFORMANCE AND GENERATE A FULLY DETAILED REPORT?

04

HOW TO ALIGN STUDENTS WITH WHAT THEY ARE LEARNING AND WHAT THEY CAN BECOME ONCE THEY REACH THE INDUSTRY?



RESEARCH OBJECTIVE

Provide, **personalised** supportive learning environment to students with best **learning materials** and give tutors full **insight of the learning patterns of the students**

01
Identify and assign the students with their learning strategy

02
Recommend the best learning materials to the student

03
Analyze the student performance, forecast and generate a fully detailed report.

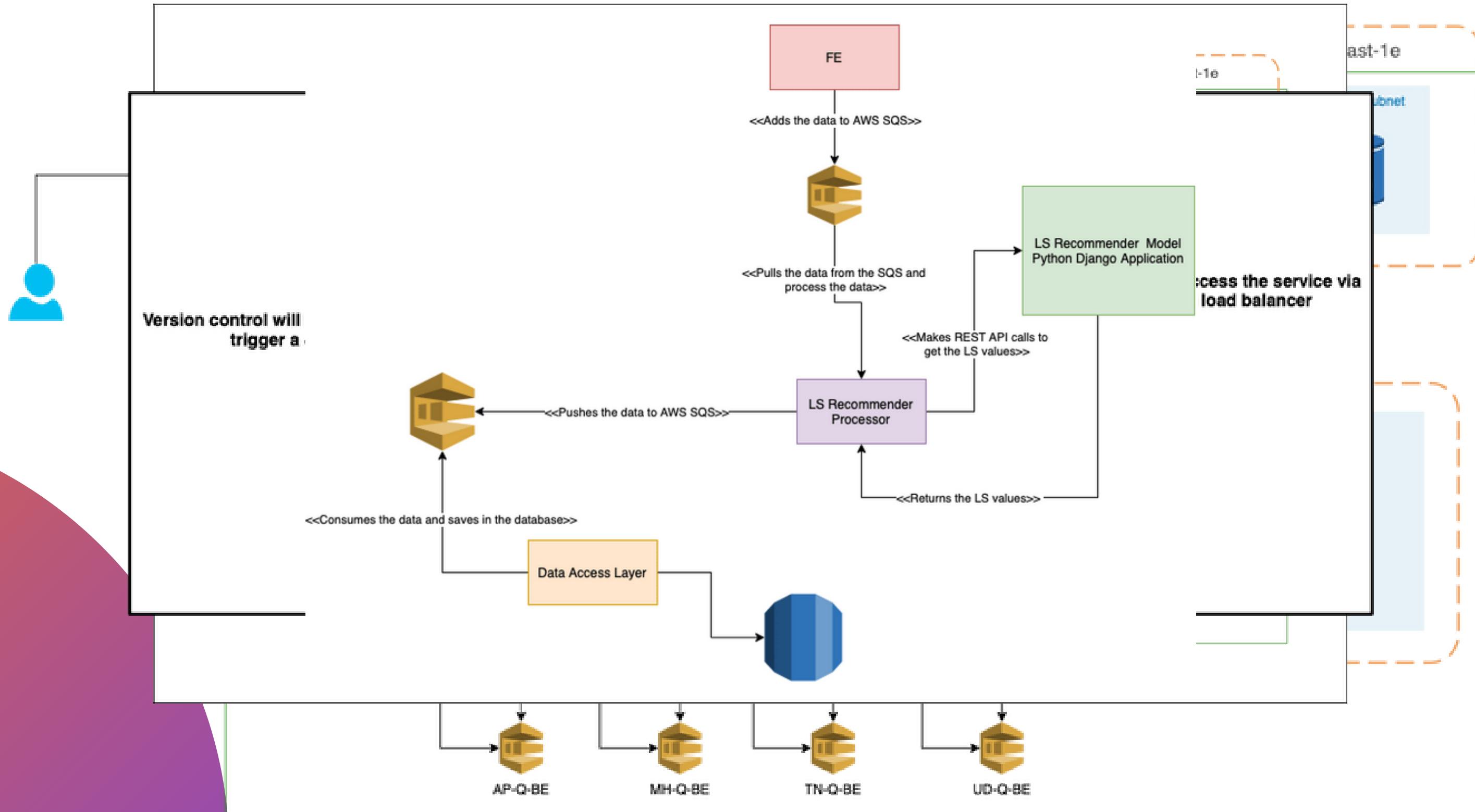
04
Student Skill Identification and Career recommendation





2022-017

SYSTEM ARCHITECTURE





2022-017



LIYANAGE M.L.A.P.

IT19120812

FOCUSED AREA: Identifying and assign learning strategies of the students

SPECIALIZED IN SOFTWARE
ENGINEERING

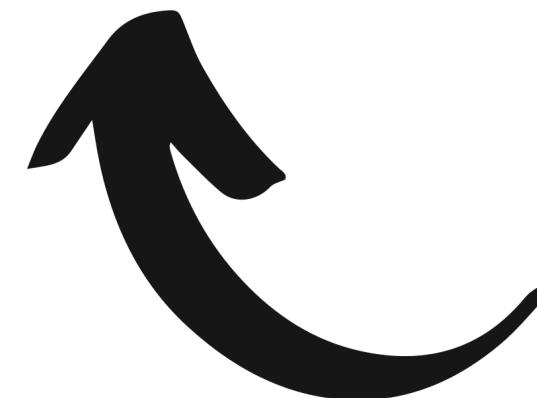




PROBLEM DEFINITION

01

**IDENTIFY THE LEARNING
STRATEGY OF THE STUDENT**



02

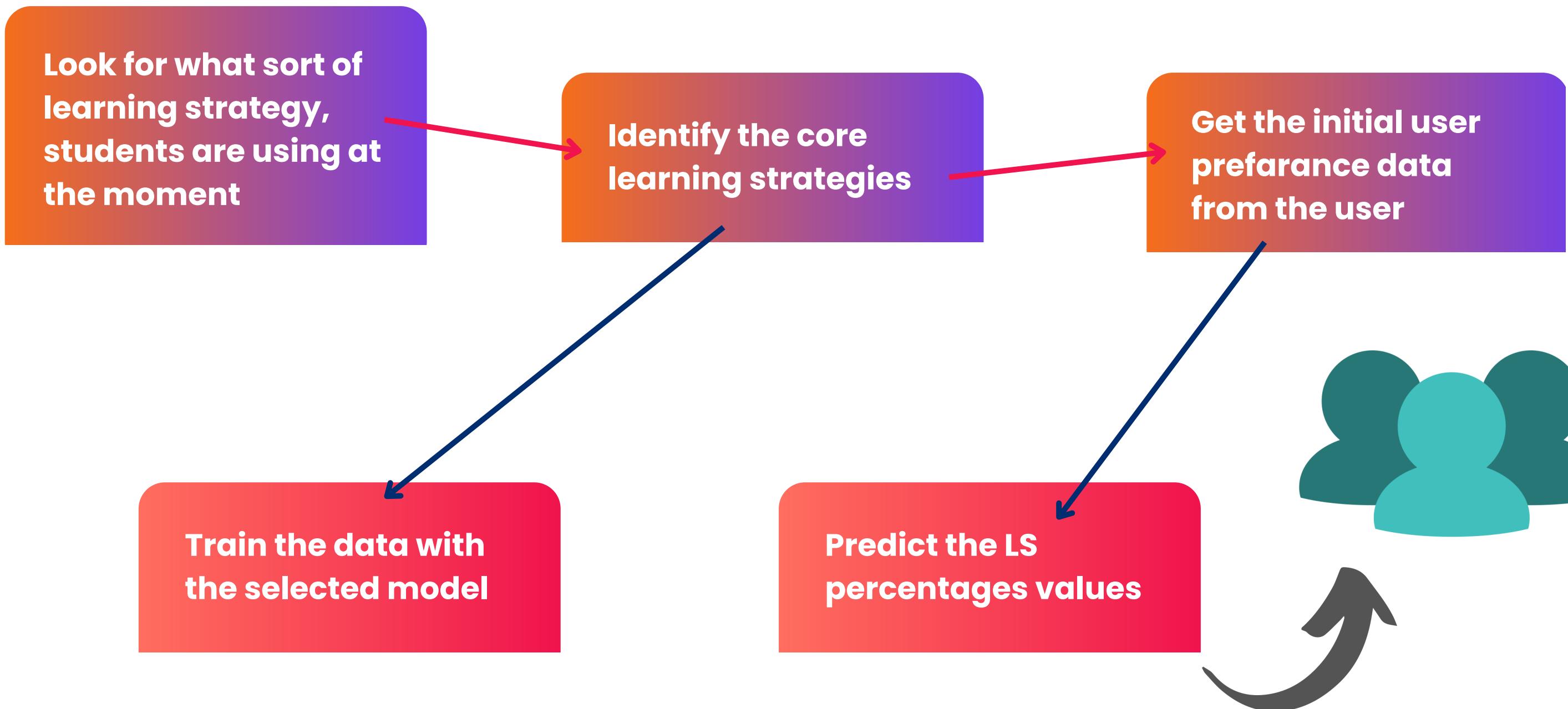
**FINE-TUNE THE IDENTIFIED
LEARNING STRATEGY VARIABLE
VALUES**



2022-017

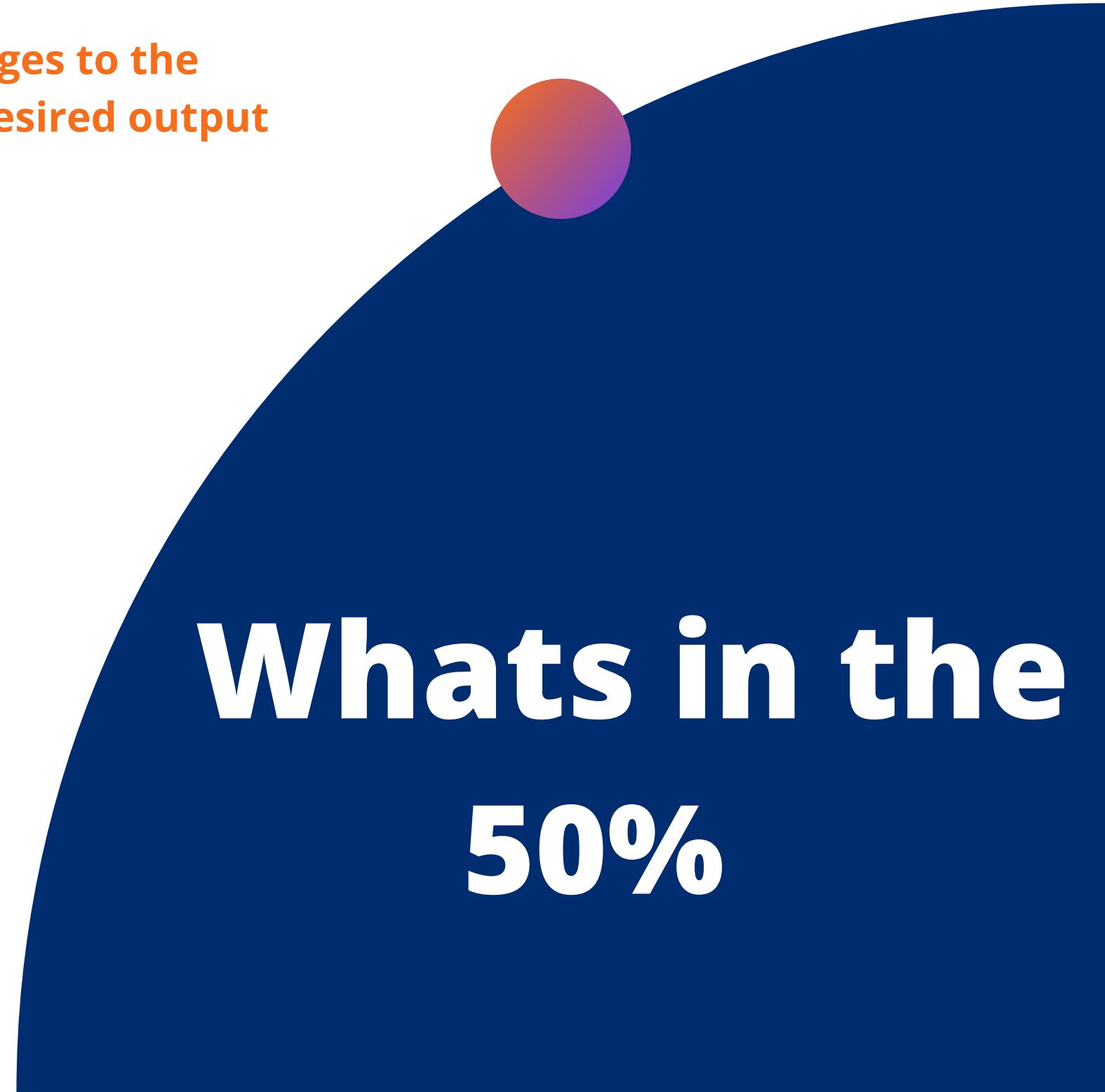


HOW TO IDENTIFY THE LEARNING STRATEGY – PROOF OF CONCEPT





- 01 Ideal Model Selection and Evaluation
- 02 Learning strategy identification algorithm creation
- 03 Initial data collection
- 04 Model training with defined model
- 05 Do necessary changes to the model to get the desired output



**What's in the
50%**



2022-017

APPLICATION OF KEY PILLARS IN THE SPECIALISED AREA



01

Strategies used in selecting the specialised area

02

Why Clustering?

03

Algorithms available

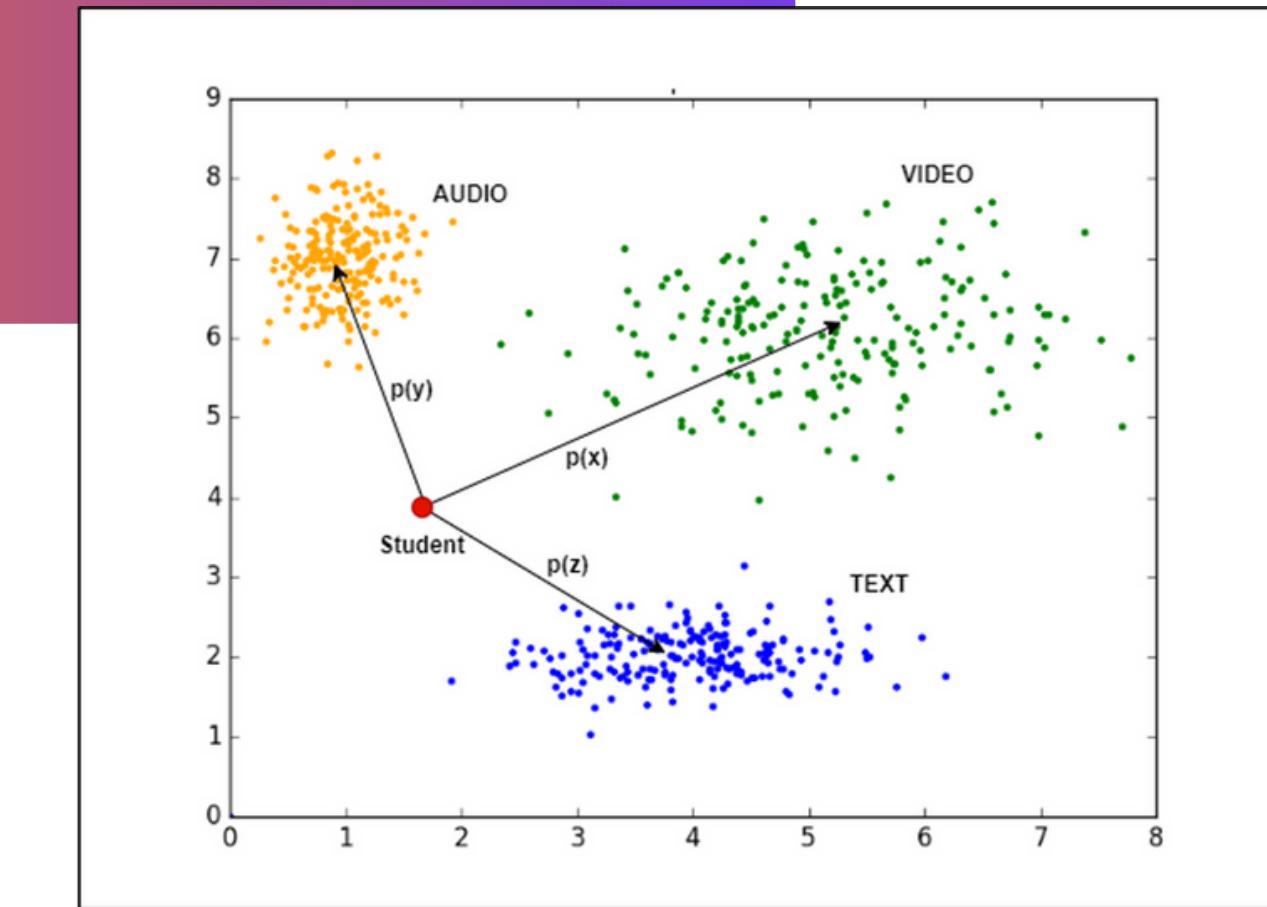
FUZZY-C MEANS CLUSTERING

K-Means Clustering

Mean-Shift Clustering

Expectation–Maximization (EM) Clustering using Gaussian Mixture Models (GMM)

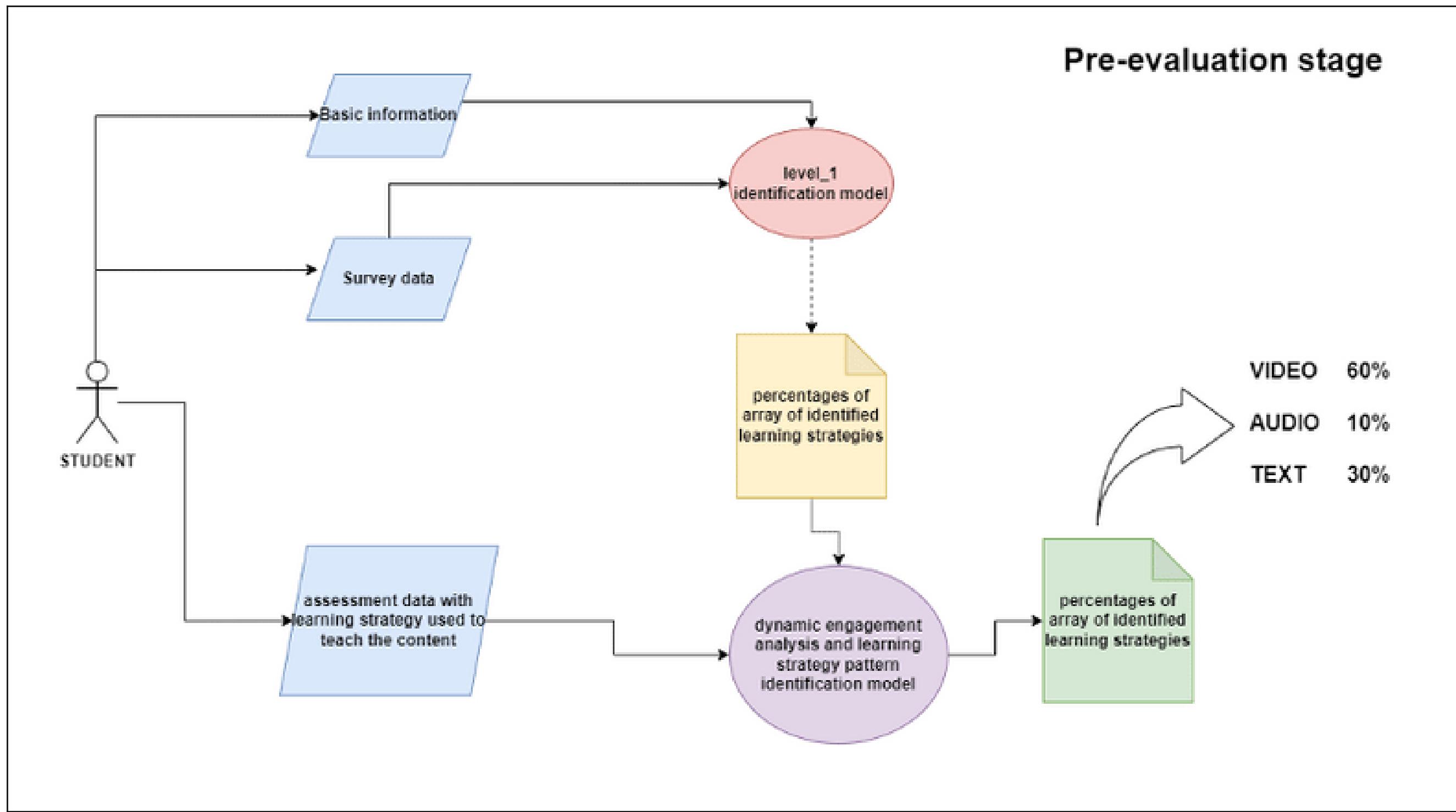
DBSCAN





2022-017

USER FLOW IN HIGH LEVEL





EVIDENCES FOR THE COMPLETION

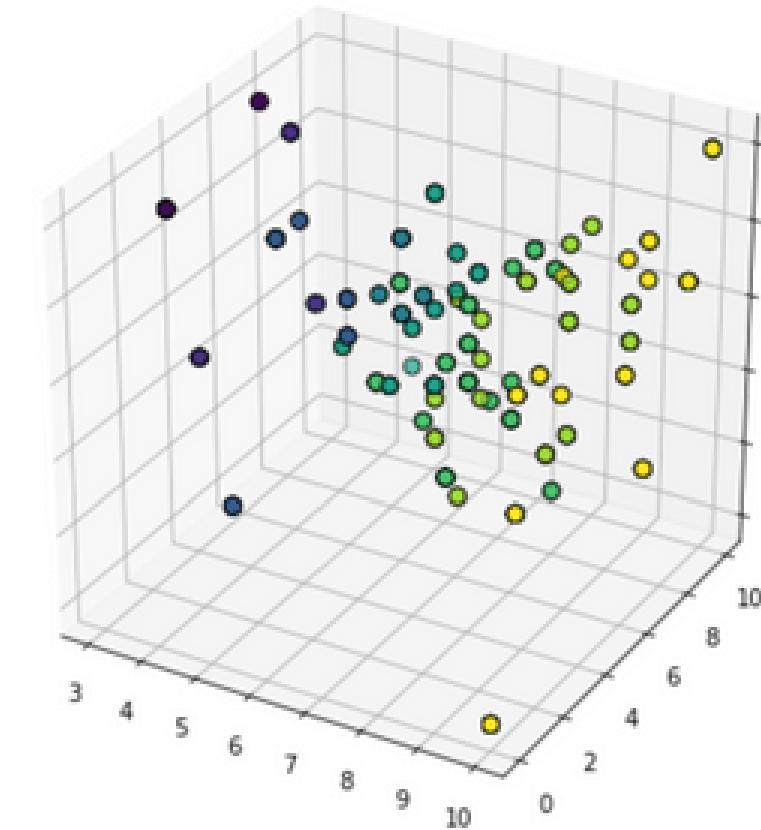
```
[ ] 1 df = pd.read_csv('https://aitor-train-data.s3.amazonaws.com/akila_train_new.csv').fillna(0)

[ ] 1 combined_da`Add code cell`f.T.to_numpy()
2 xpts = [com M/CH+M B _new[0].astype(int)
3 ypts = [combined_data_new[1].astype(int)
4 print(ypts)

[array([ 4,  2,  6,  1,  3,  5,  0,  7,  4,  6,  3,  1, 10,  4,  3,  1,  5,
       3,  6,  0,  9,  6,  4,  1,  5,  7,  7,  8,  1,  2,  2,  4,  6,
       6,  7,  6,  6,  1,  4,  1,  2,  5,  6,  0,  4,  2,  6,  7,  1,  4,
       8,  7,  6,  6,  5,  6,  5,  9,  9,  7,  3,  6,  2,  6,  0,  6,  3,
       3,  7,  0,  4,  2,  6,  7,  1,  4,  8,  7,  6,  6,  5,  6,  5,  9,
       9,  7,  3,  6,  2,  6,  0,  6,  3,  3,  7,  7,  4,  6,  3,  1, 10,
       4,  3,  1,  5,  3,  6,  0,  9,  6,  4,  1,  7,  4,  6,  3,  1, 10,
       4,  3,  1,  5,  3,  6,  0,  9,  6,  4,  1,  2,  6,  1,  3,  5,  0,
       7,  4,  6,  3,  1, 10,  4,  3,  1,  5,  3,  6,  0,  9,  6,  4,  1,
       5,  7,  7,  7,  8,  1,  2,  2,  4,  6,  6,  7,  6,  6,  1,  4,  1,
       2,  5,  6,  0,  4,  2,  6,  7,  1,  4,  8,  7,  6,  6,  5,  6,  5,
       9,  9,  7,  3,  6,  2,  6,  0,  6,  3,  3,  7,  7,  0,  4,  2,  6,  7,
       1,  4,  8,  7,  6,  6,  5,  6,  5,  9,  9,  7,  3,  6,  2,  6,  0,
       6,  3,  3,  7,  7,  4,  6,  3,  1, 10,  4,  3,  1,  5,  3,  6,  0,
       9,  6,  4,  1,  7,  4,  6,  3,  1, 10,  4,  3,  1,  5,  3,  6,  0,
       9,  6,  4,  1,  2,  6,  1,  3,  5,  0,  7,  4,  6,  3,  1, 10,  4,
       3,  1,  5,  3,  6,  0,  9,  6,  4,  1,  5,  7,  7,  7,  8,  1,  2,
       2,  4,  6,  6,  7,  6,  6,  1,  4,  1,  2,  5,  6,  0,  4,  2,  6,
       7,  1,  4,  8,  7,  6,  6,  5,  6,  5,  9,  9,  7,  3,  6,  2,  6,
       0,  6,  3,  3,  7,  0,  4,  2,  6,  7,  1,  4,  8,  7,  6,  6,  5,
       6,  5,  9,  9,  7,  3,  6,  2,  6,  0,  6,  3,  3,  7,  7,  4,  6,
       3,  1, 10,  4,  3,  1,  5,  3,  6,  0,  9,  6,  4,  1,  7,  4,  6,
       3,  1, 10,  4,  3,  1,  5,  3,  6,  0,  9,  6,  4,  1,  1])]
```

```
1 # Visualize the test data
2 colors = ['b', 'orange', 'g', 'r', 'c', 'm', 'y', 'k', 'Brown', 'ForestGreen']
3 labels = np.zeros(1)
4 fig = plt.figure(figsize=(7, 7))
5 ax = fig.add_subplot(111, projection='3d')
6 ax.scatter(combined_data_new[0].astype(int), combined_data_new[1].astype(int), combined_data_new[2].astype(int),
7             linewidths=1, alpha=.7,
8             edgecolor='k',
9             s = 50,
10            c=combined_data_new[0].astype(int))

<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x7fc5c7cbd50>
```

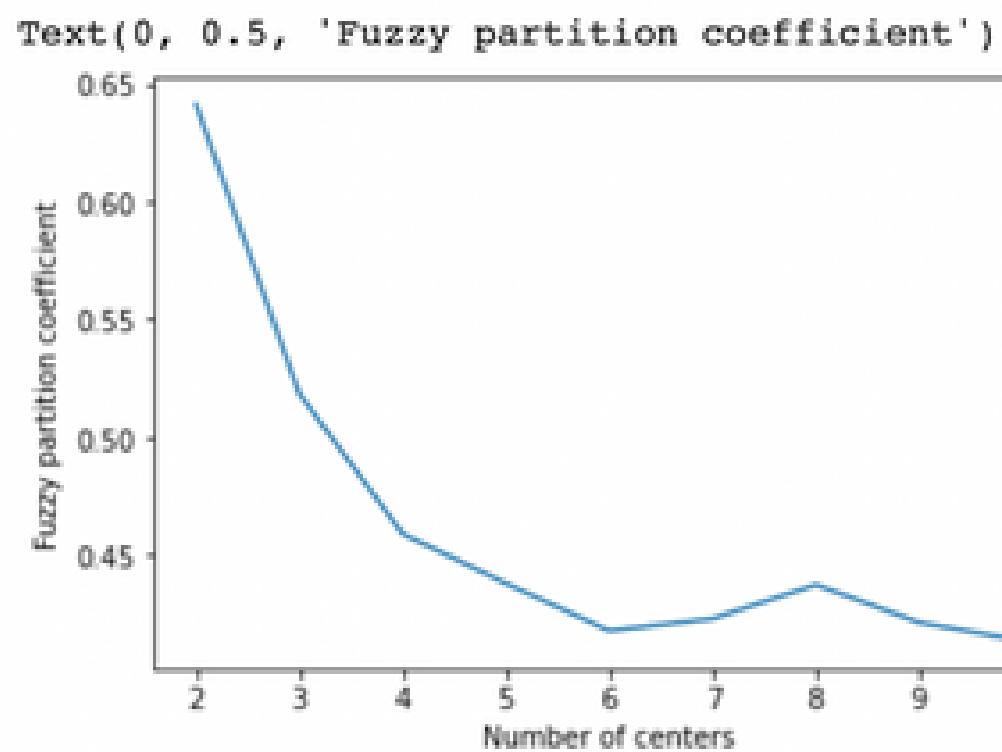




EVIDENCES FOR THE COMPLETION

```
1 fpcs = []
2 for i in range(2,11):
3     cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(combined_data_new, i, 2, error=0.005, maxiter=1000, init=None)
4     fpcs.append(fpc)
```

```
1 fig2, ax2 = plt.subplots()
2 ax2.plot(np.r_[2:11], fpcs)
3 ax2.set_xlabel("Number of centers")
4 ax2.set_ylabel("Fuzzy partition coefficient")
```





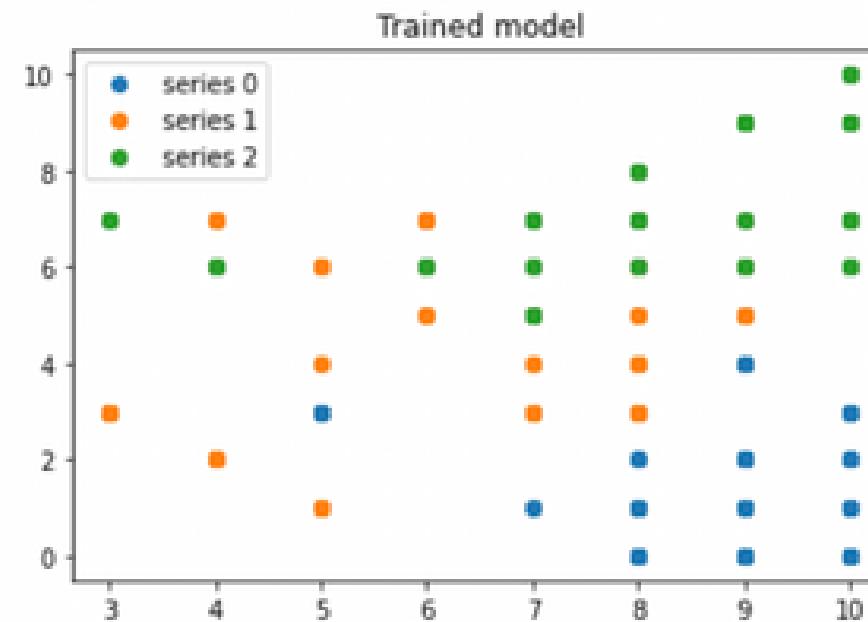
EVIDENCES FOR THE COMPLETION

```
1 cnter , u_orig , _ , _ , _ , _ = fuzz.cluster.cmeans(combined_data_new , 3 , 2 , error=0.005 , maxiter=1000)
2 print(cnter)
```

```
[[8.58033421 1.81229136 7.4920698 ]
 [7.27381637 4.71988479 5.67932412]
 [8.28011788 6.6967631 7.46555968]]
```

```
1 # Show 3-cluster model
2 fig2, ax2 = plt.subplots()
3 ax2.set_title('Trained model')
4 for j in range(3):
5     ax2.plot(combined_data_new[0, u_orig.argmax(axis=0) == j],
6               combined_data_new[1, u_orig.argmax(axis=0) == j], 'o',
7               label='series ' + str(j))
8 ax2.legend()
```

```
<matplotlib.legend.Legend at 0x7fcb57f2da50>
```





Usage of technologies in the relevant key pillar/area

COMPONENT/MODULE	TECHNIQUES	RULES AND ALGORITHMS	TECHNOLOGIES
LS Recommender model	Clustering	Fuzzy-C Means	Python, Scikit-learn, Pandas, Numpy, Django, Pickle, Skfuzzy
User Interfaces	UI/UX		Figma, ReactJs



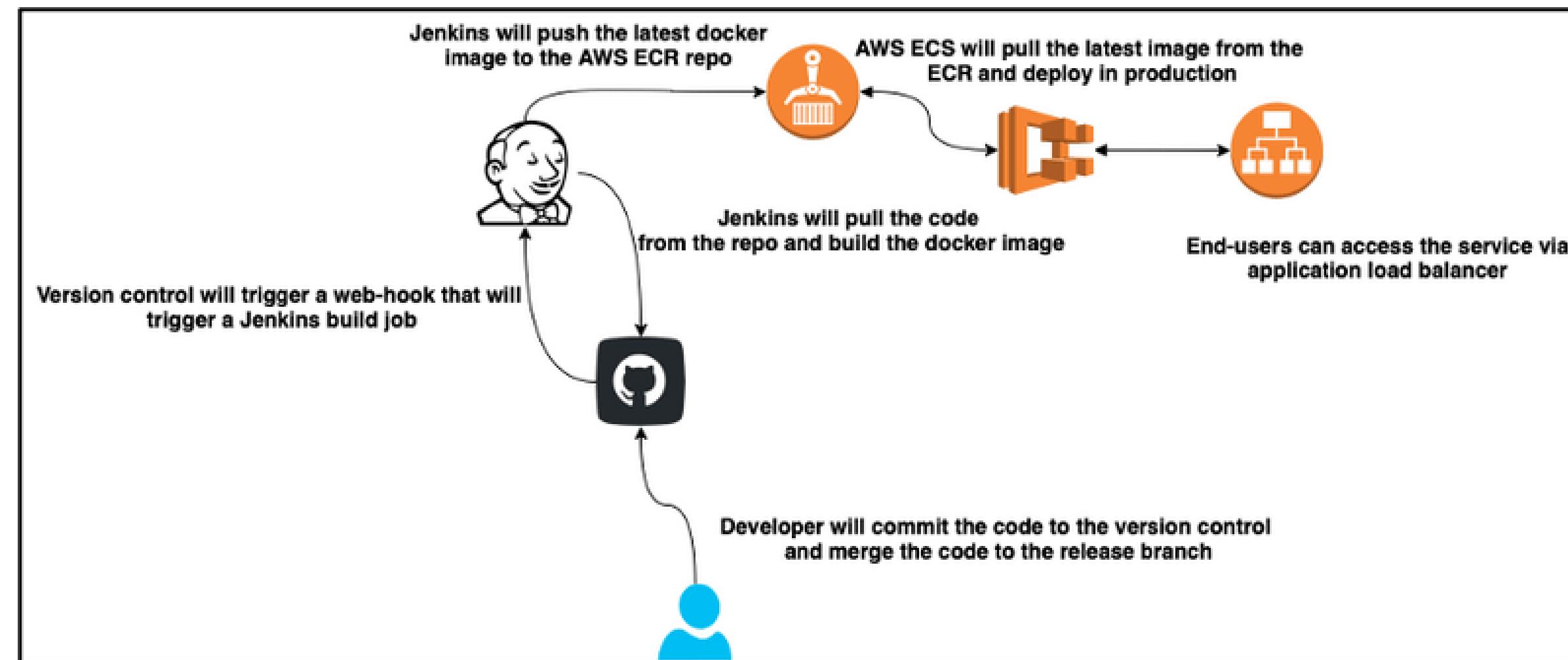


2022-017

DESIGN EXCELLENCE AND BEST PRACTICES FOLLOWED



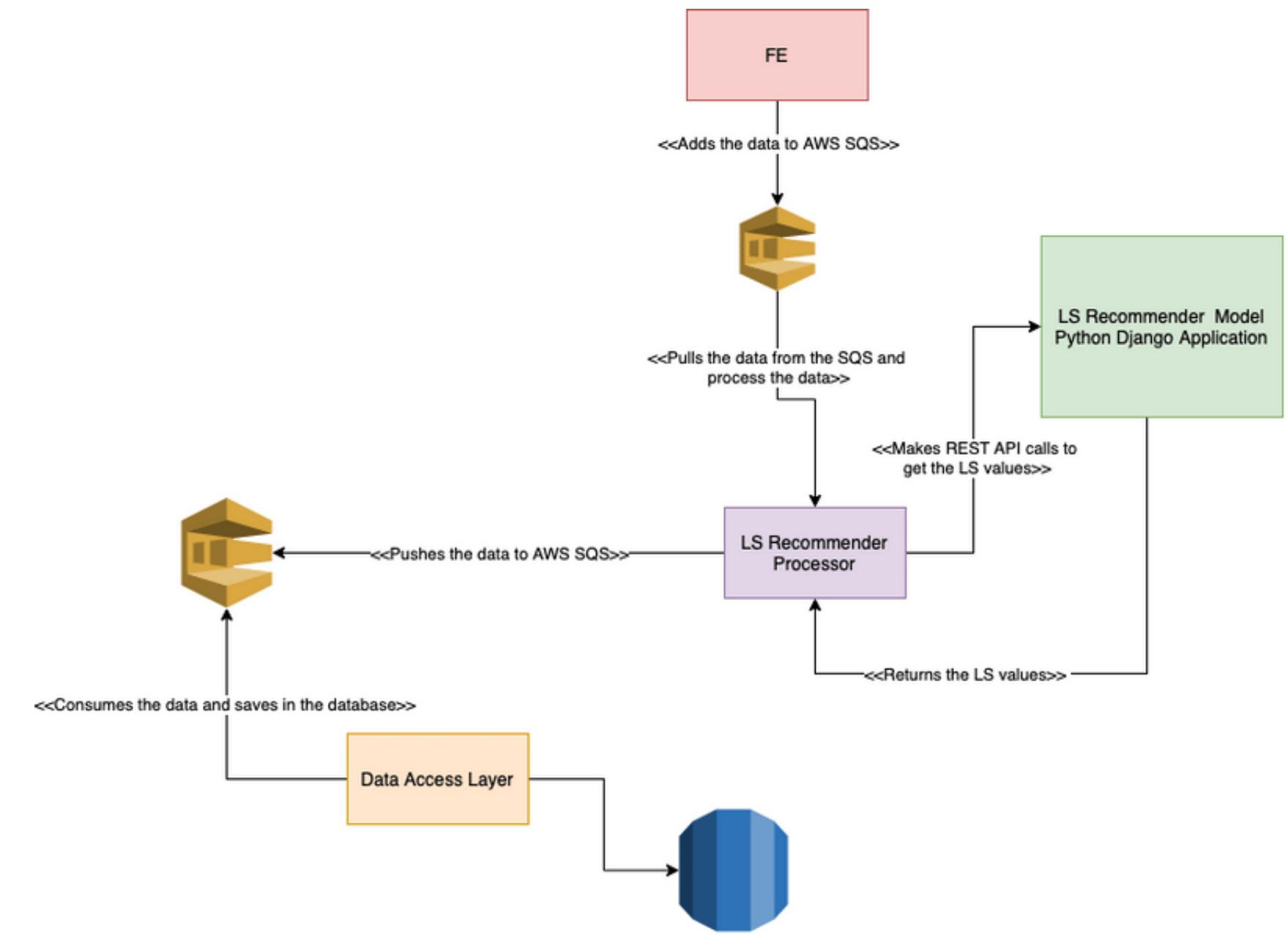
- Entire architecture will follow micro-services topology
- Proposed feature will be developed/ has been developed as fully cloud native manner





STANDERD BEST PRACTICES

- It has been identified that, with the increment of the registrations, the number of users using the LS recommender feature will be in high demand
- Application will be de-coupled by using message queues
- Application will be/has been structured to adopt to full DevSecOps methodologies





USER/FUNCTIONAL REQUIREMENTS

- **Functional Requirements**

- Successful identification of each student's learning strategy
- Allocation/fine tuning of identified LS values of each student

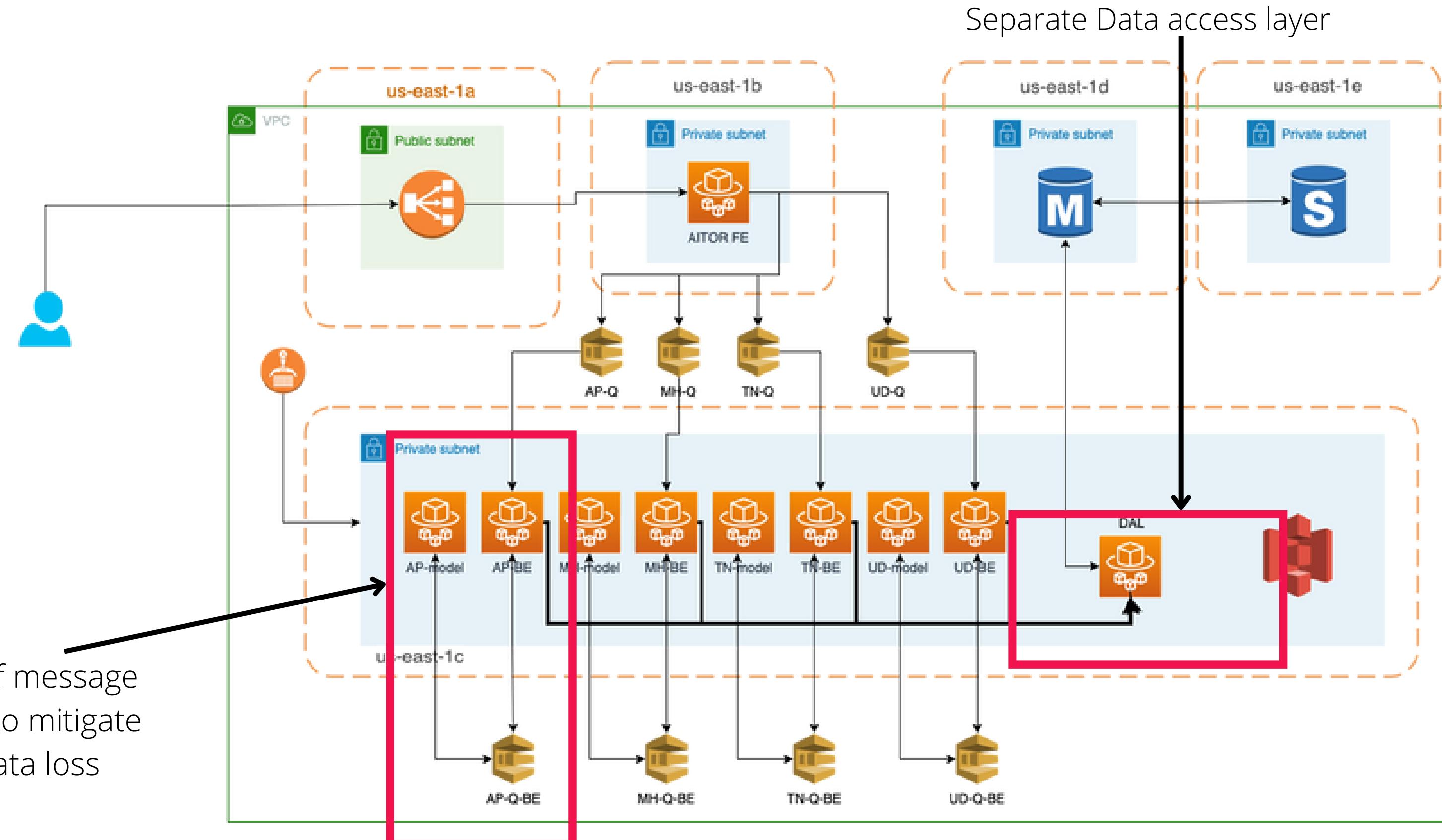
- **Non functional requirements**

- Students should be able to access the application from anywhere in the world
- Students/Tutors should be able to easily afford the application without paying much on features



2022-017

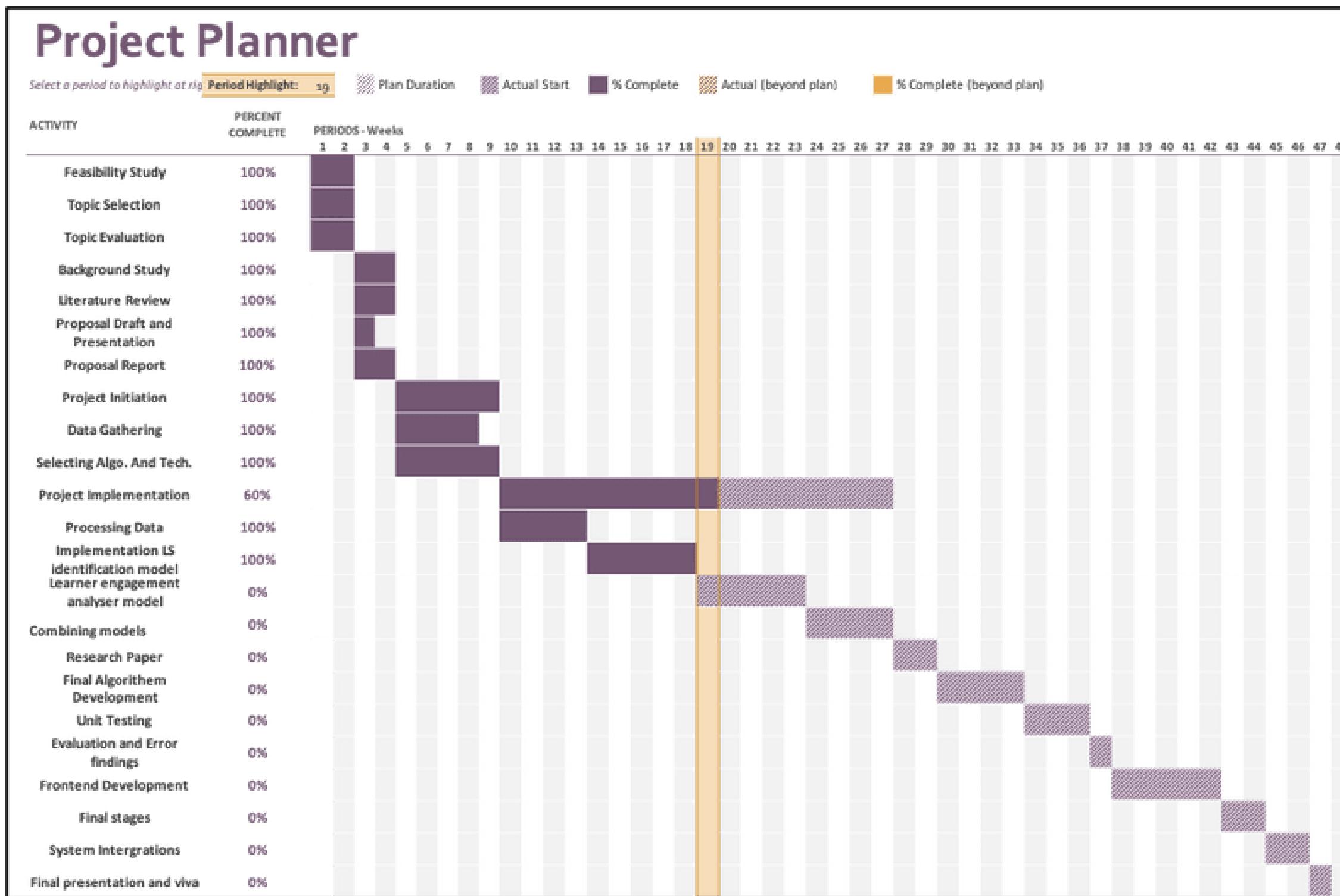
RISK MITIGATION





2022-017

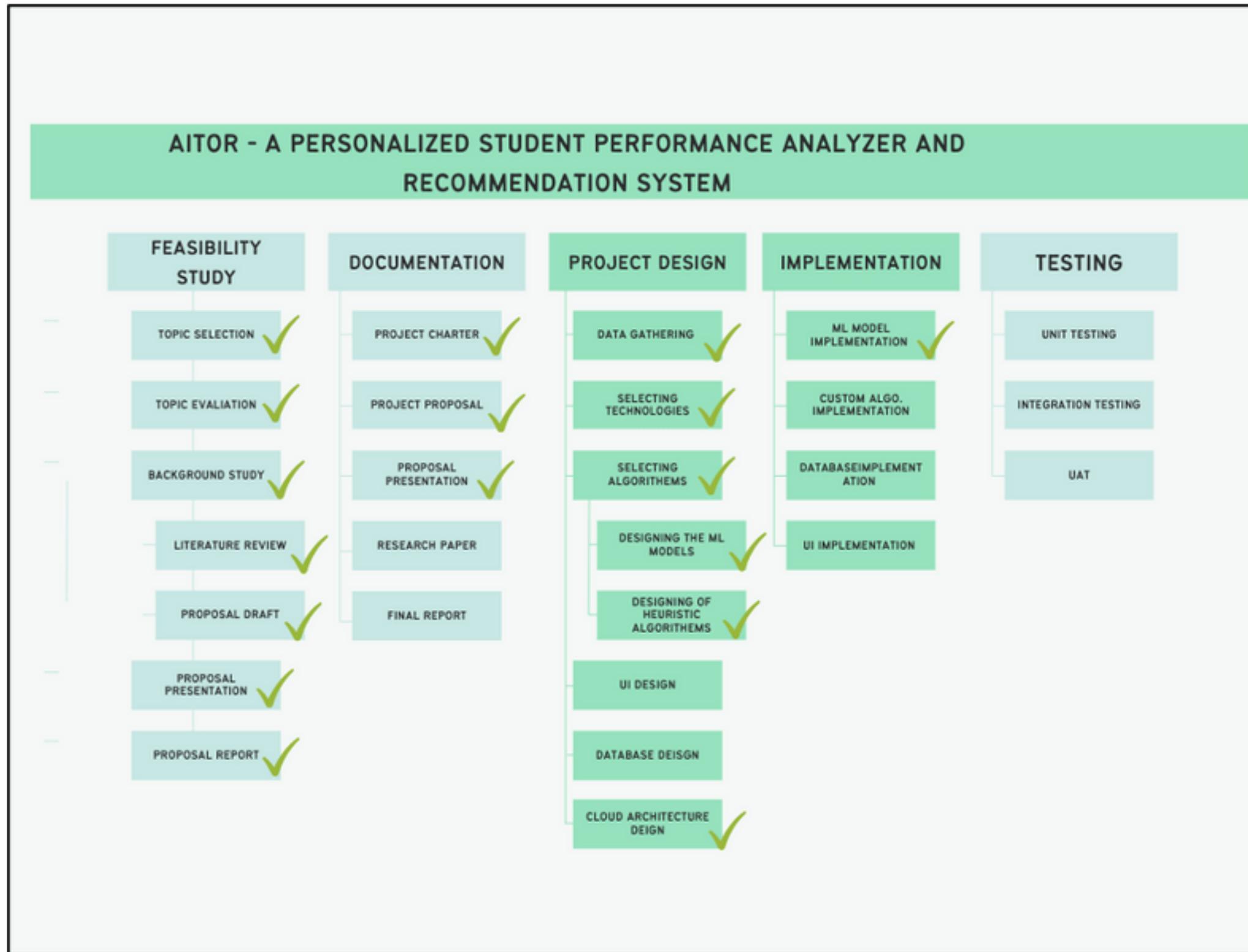
PROJECT GANATT CHART





2022-017

WORK BREAKDOWN CHART



DEMONSTRATION

**Lets
Seek The
SUCCESS.**

• Lorem ipsum is simply dummy text of the printing and typesetting industry. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem ipsum.

< PREVIOUS ● ● ● ● NEXT >

LEARNING STRATEGY IDENTIFICATION



Please fill the following information

On to scale 1 -10,

How comfortable are you learning something by watching a video?

How comfortable are you learning something by reading?

How comfortable are you learning something by listening to an audio?

FINISH



2022-017



Thammita D.H.M.M.P

IT19120362

FOCUSED AREA: Identification and recommendation of best-suited learning materials based on learning style and personal preferences.

SPECIALIZED IN SOFTWARE
ENGINEERING





2022-017



PROBLEM DEFINITION

01

IDENTIFY KEY LEARNING AREAS



02

**LEARNING MATERIAL RECOMMENDATION
BASED ON PERSONAL PREFERENCE AND
LEARNING STYLE**



03

**FINETUNE LEARNING MATERIAL
RECOMMENDATIONS**

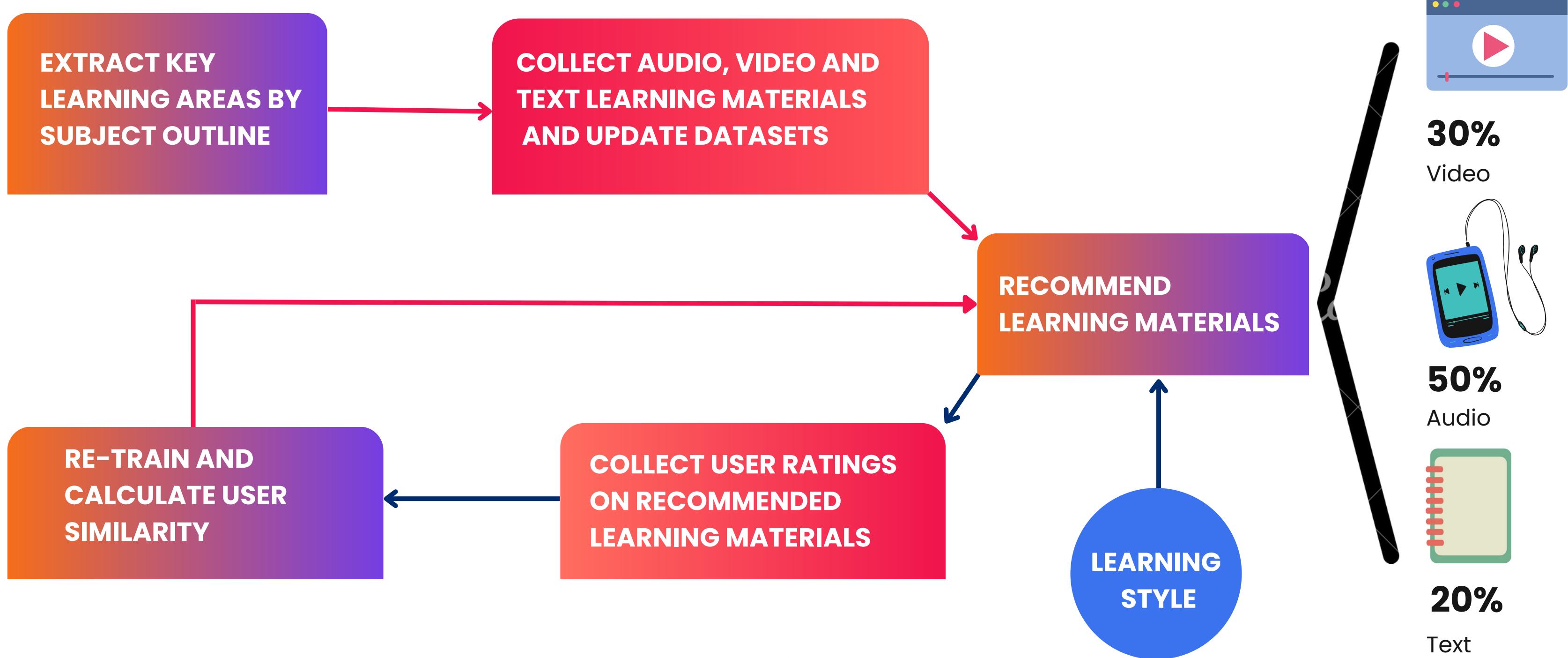




2022-017

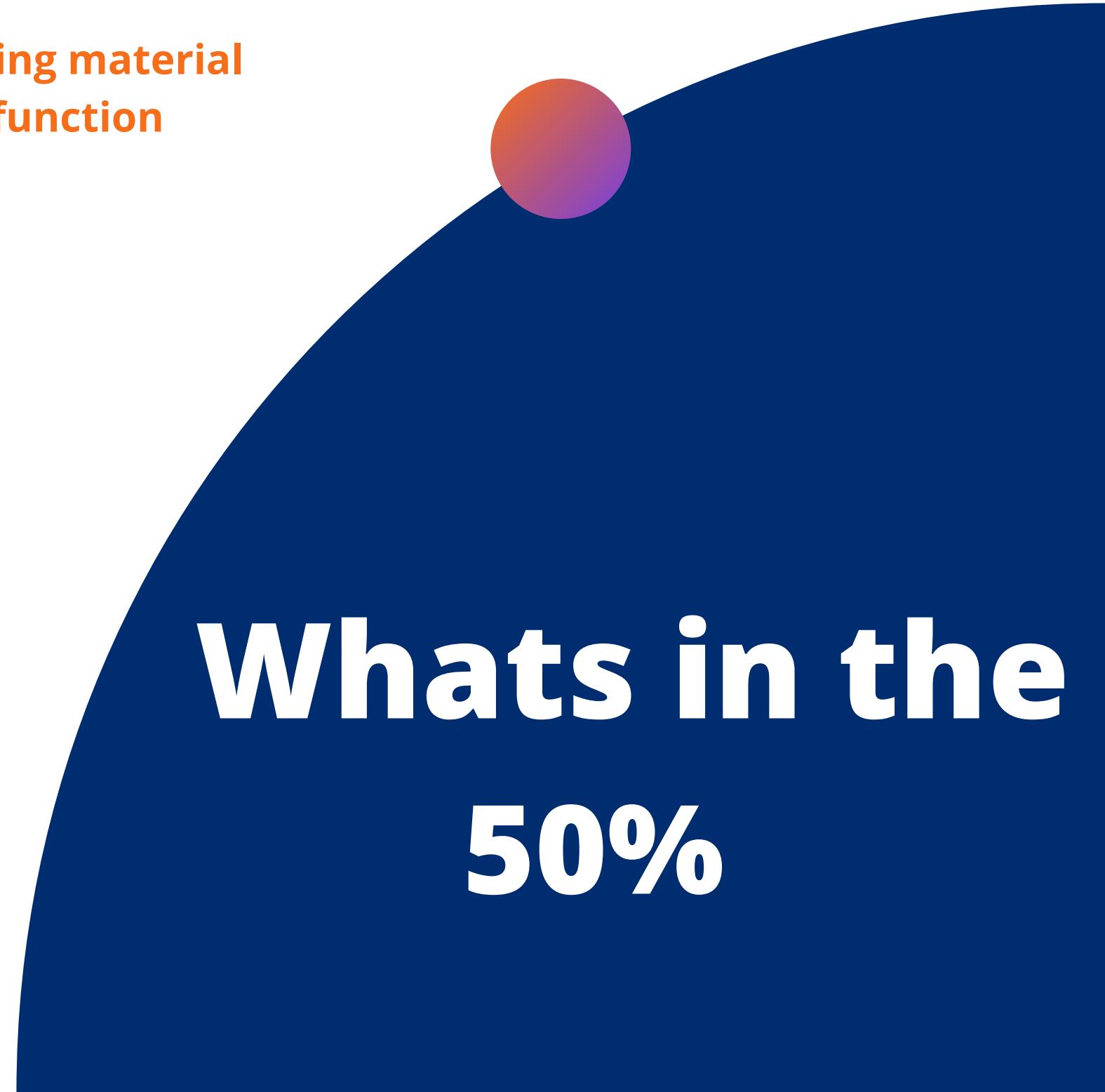


HOW TO RECOMMEND LEARNING MATERIALS- PROOF OF CONCEPT





- 01 Ideal Model Selection and Evaluation
- 02 Key Learning Areas Identification Model Implementation
- 03 Learning Material Recommendation Algorithm Implementation
- 04 Implementation of Connection to the YouTube Data API and initial data collection
- 05 Video-based Learning material Recommendation function implementation



**Whats in the
50%**



APPLICATION OF KEY PILLARS IN THE SPECIALISED AREA

NEAREST NEIGHBOUR COLLABORATIVE FILTERING – USER-BASED COLLABORATIVE FILTERING

01

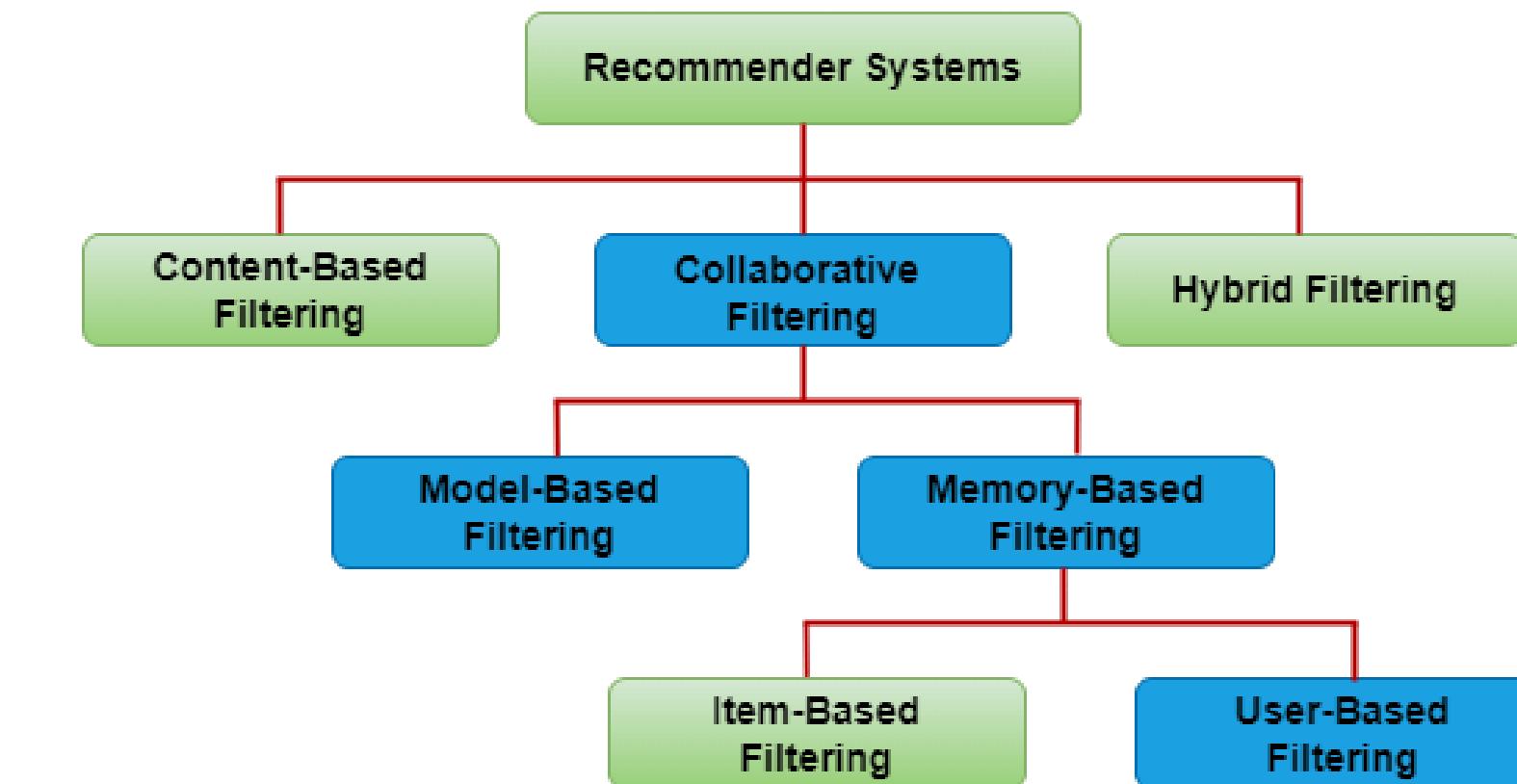
Why Collaborative filtering over content-based filtering?

02

Why User-Based Collaborative filtering over Item-Based Collaborative filtering?

03

What Used in Implementation





2022-017

APPLICATION OF KEY PILLARS IN THE SPECIALISED AREA



SVD – SINGULAR VALUE DECOMPOSITION

01

Different Implementations in Model-Based Filtering

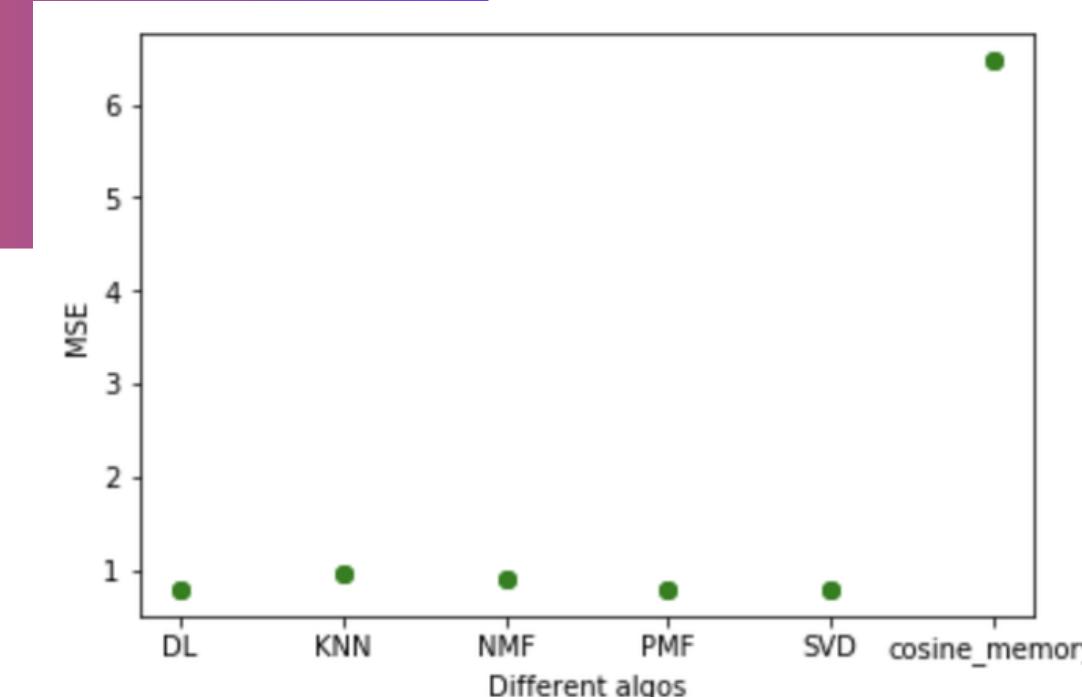
02

What is the best implementation?

02

Why SVD?

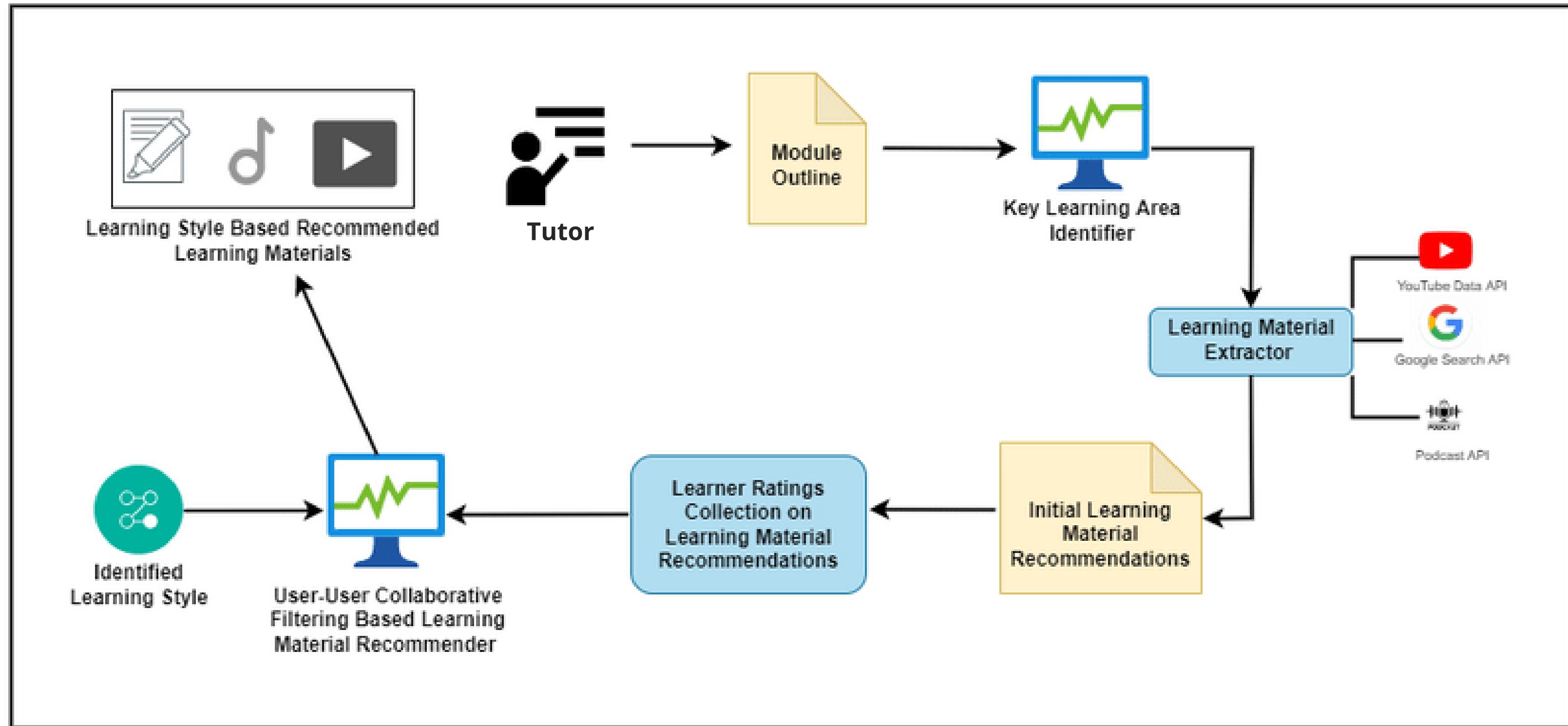
- SVD
- KNN
- Matrix Factorization
- Multi layered neural networks





2022-017

USER FLOW IN HIGH LEVEL





2022-017

EVIDENCES FOR THE COMPLETION



```
# create a df of just the current user
user1 = rating_matrix[rating_matrix.index == current_user]

# and a df of all other users
other_users1 = rating_matrix[rating_matrix.index != current_user]

# calc cosine similarity between user and each other user
similarities = cosine_similarity(user1,other_users1)[0].tolist()

# create list of indices of these users
indices = other_users1.index.tolist()

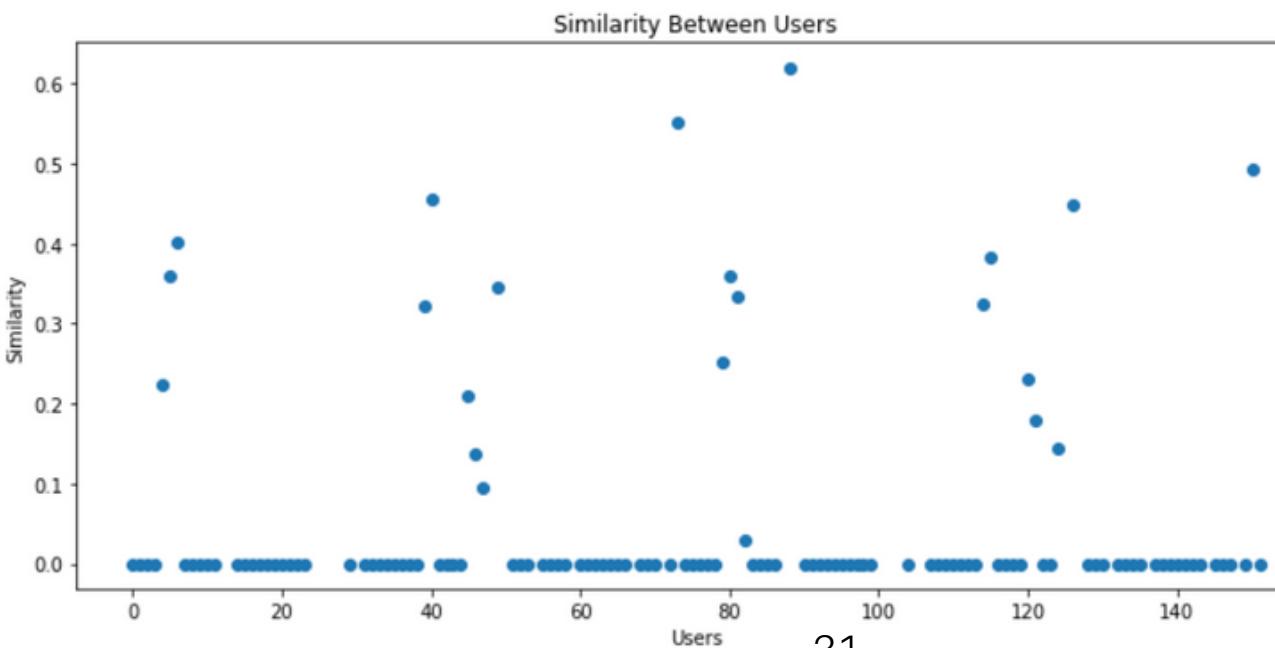
# create key/values pairs of user index and their similarity
index_similarity = dict(zip(indices, similarities))

import matplotlib.pyplot as plt

lists = sorted(index_similarity.items()) # sorted by key, return a list of tuples

x, y = zip(*lists) # unpack a list of pairs into two tuples

plt.figure(figsize=(10,5))
plt.scatter(x, y)
plt.xlabel('Users')
plt.ylabel('Similarity')
plt.title('Similarity Between Users')
plt.tight_layout()
plt.show()
```





EVIDENCES FOR THE COMPLETION

```
# Use the famous SVD algorithm
algo = SVD()

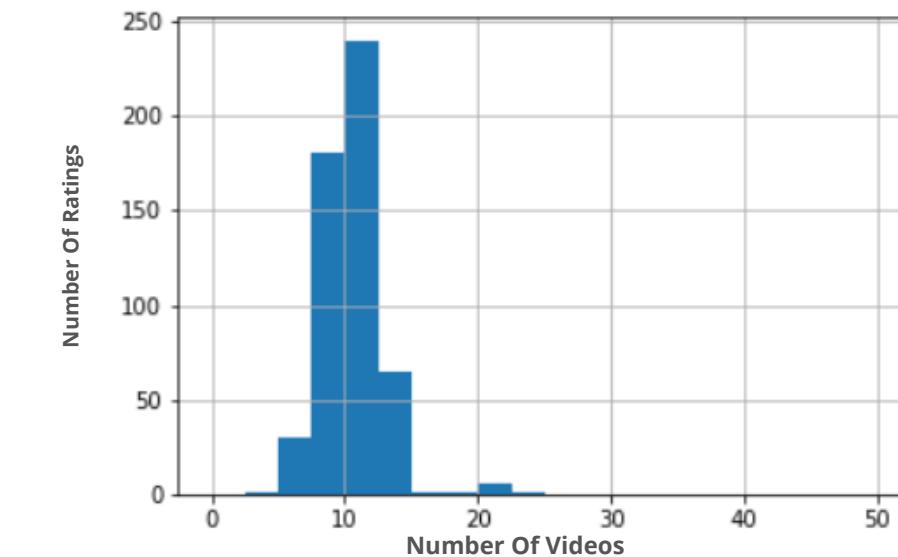
# Run 5-fold cross-validation and then print results
cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=5, verbose=True)
```

Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	2.0349	2.0107	2.0525	2.0766	2.0928	2.0535	0.0292
MAE (testset)	1.7469	1.7099	1.7669	1.7835	1.7984	1.7611	0.0309
Fit time	0.25	0.26	0.25	0.26	0.26	0.26	0.01
Test time	0.01	0.01	0.01	0.01	0.01	0.01	0.00

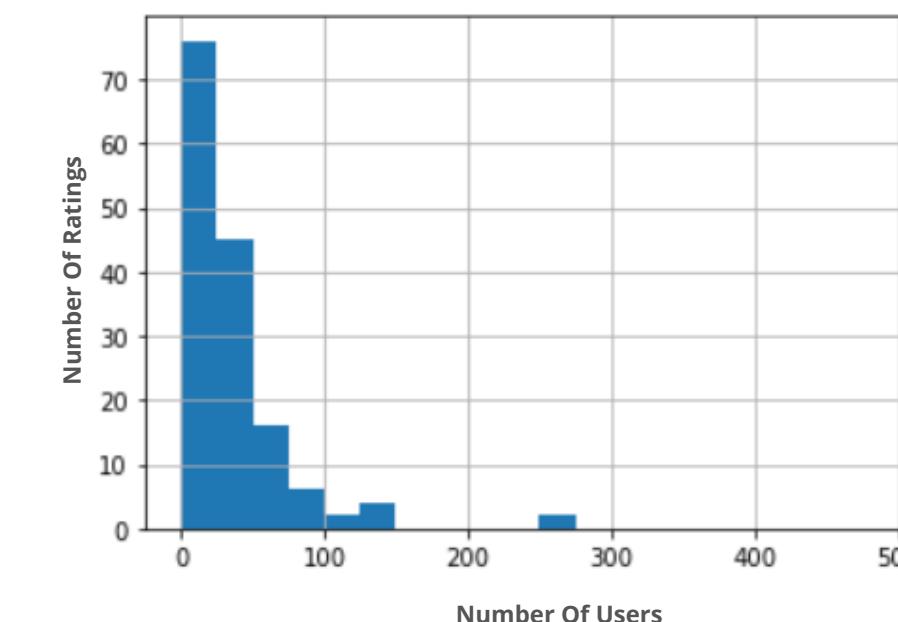
```
# distribution of ratings per anime
ratings_per_video.hist(bins=20, range=(0,50))
```

<AxesSubplot:>



```
# distribution of ratings per user
ratings_per_user.hist(bins=20, range=(0,500))
```

<AxesSubplot:>





USER/FUNCTIONAL REQUIREMENTS

- **Functional Requirements**

- Suggest Learning materials that align with the learning styles of the user.
- Identify key learning areas through learning outlines
- Continuously fine-tune and adjust learning material recommendations according to the learning style.

- **Non-functional requirements**

- Students should be able to freely access the recommended learning materials
- Ensure thehigh availability of recommended items



2022-017



Usage of technologies in the relevant key pillar/area

COMPONENT/MODULE	TECHNIQUES	RULES AND ALGORITHMS	TECHNOLOGIES
Learning Material Recommender Model	Collaborative Filtering	SVD & Cosine Similarity	Python, Scikit-learn, Pandas, Numpy, Django, Surprise
Filtering Keyword Identifier	Information Extraction	RNN	Python, Scikit-learn, Pandas, Numpy, Django, Keras, Tensorflow



surprise

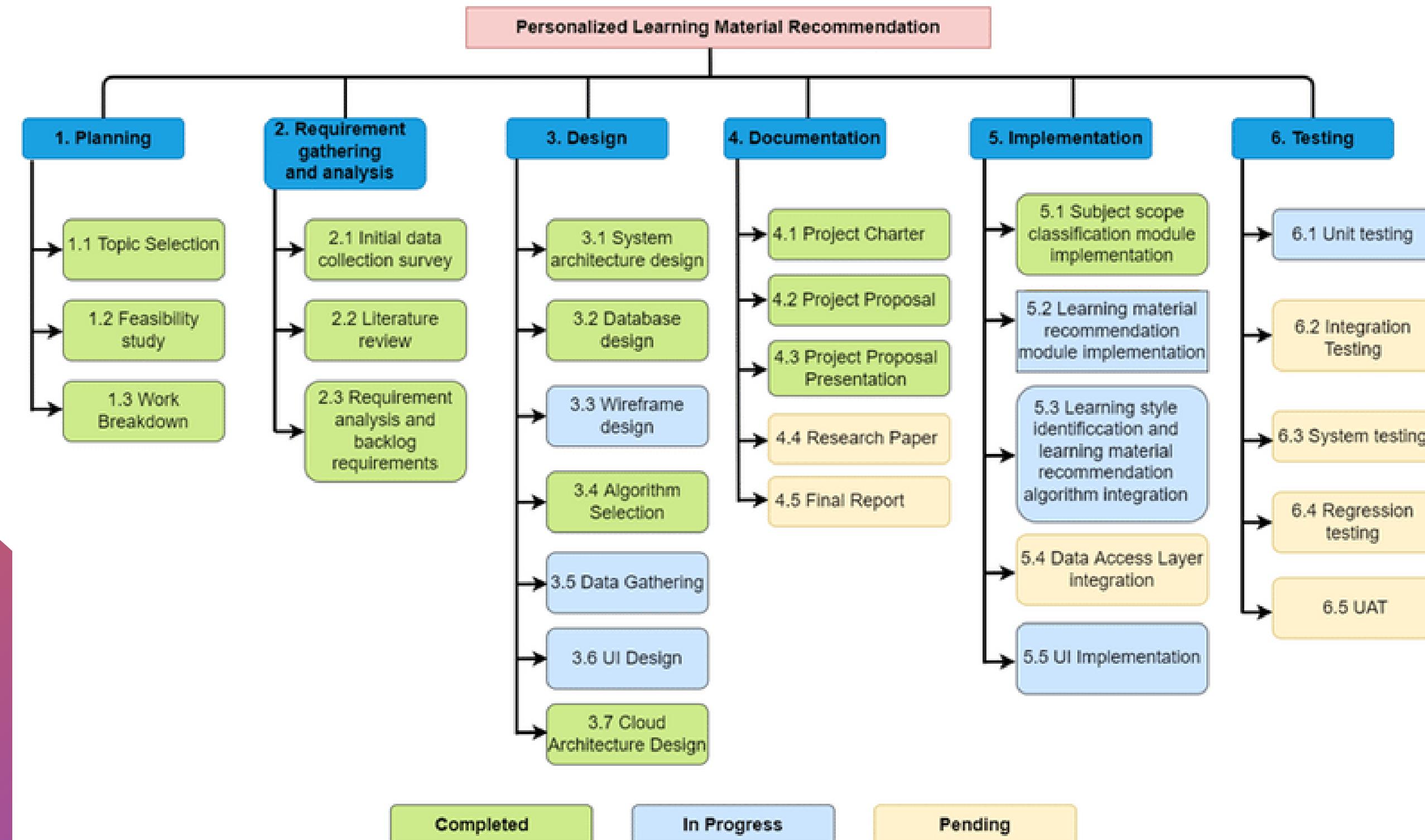




2022-017



Work Breakdown Structure





2022-017

Project Gantt Chart



Task	Semester 1						Semester 2					
	January	February	March	April	May	June	July	August	September	October	November	December
Feasibility study	Completed											
Topic Selection	Completed											
Topic Evaluation												
Background Study	Completed											
Background study and Literature survey	Completed											
Proposal Draft												
Project Proposal	Completed											
Proposal presentation	Completed											
Proposal Report	Completed											
Project Initiation		Completed										
Data gathering		Completed										
Selecting algorithms and technologies		Completed										
Project Implementation			Completed		Completed		Completed		Completed		Completed	
Processing data			Completed		Completed		Completed		Completed		Completed	
Subject scope classification module			Completed		Completed		Completed		Completed		Completed	
Implementation			Completed		Completed		Completed		Completed		Completed	
Learning material classification module			Completed		Completed		Completed		Completed		Completed	
Implementation			Completed		Completed		Completed		Completed		Completed	
Learner classification module Implementation			Completed		Completed		Completed		Completed		Completed	
Combining models			Completed		Completed		Completed		Completed		Completed	
Research paper			Completed		Completed		Completed		Completed		Completed	
Multidimensional Attribute-based Material			Completed		Completed		Completed		Completed		Completed	
Recommender module Implementation			Completed		Completed		Completed		Completed		Completed	
Final Algorithm implementation			Completed		Completed		Completed		Completed		Completed	
Unit Testing			Completed		Completed		Completed		Completed		Completed	
Evaluation and Error fixing			Completed		Completed		Completed		Completed		Completed	
Front end development			Completed		Completed		Completed		Completed		Completed	
Final Stages			Completed		Completed		Completed		Completed		Completed	
System Integration			Completed		Completed		Completed		Completed		Completed	
System Testing			Completed		Completed		Completed		Completed		Completed	
Evaluation and Error fixing			Completed		Completed		Completed		Completed		Completed	
Final Presentation and Viva			Completed		Completed		Completed		Completed		Completed	
Final Report			Completed		Completed		Completed		Completed		Completed	

Completed Tasks

Current Week



2022-017



HIRIMUTHUGODA UJ

IT19138114

FOCUSED AREA: Student skill identification and career recommendation

SPECIALIZED IN SOFTWARE
ENGINEERING



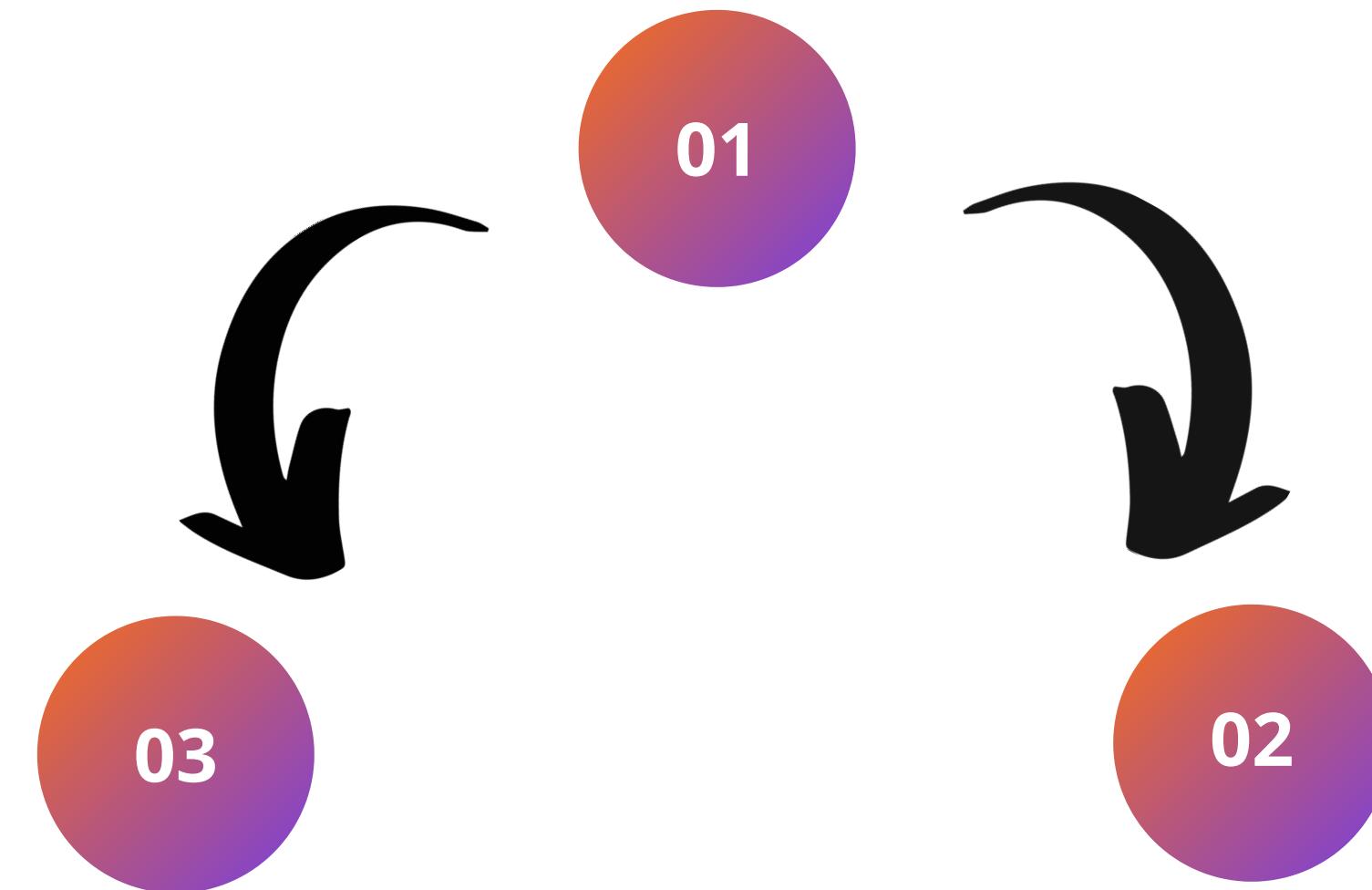


2022-017



PROBLEM DEFINITION

Student skill identification



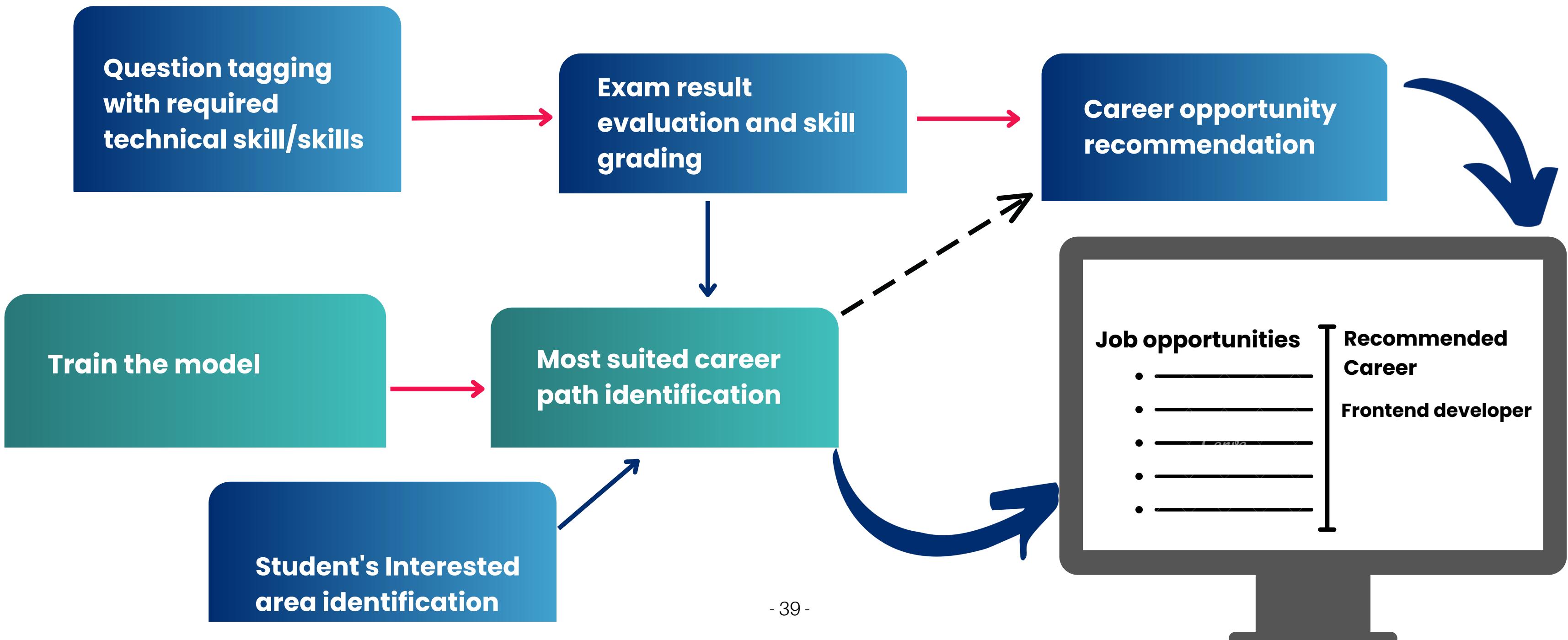
**Career opportunity
recommendation**

**Career path
recommendation**



2022-017

Skill Identification and Career opportunity recommendation – PROOF OF CONCEPT



- 01 **Skill extraction from job postings**
- 02 **Most suitable job opportunities filtering**
- 03 **Career recommendation model**
- 04 **Initial UI implementation**



**Whats in the
50%**



APPLICATION OF KEY PILLARS IN THE SPECIALISED AREA

01

Why Linear SVM over Naive Bayes

02

Text classification

03

Jaccard similarity

Perform well in multi-class classification
Look for interaction between the features

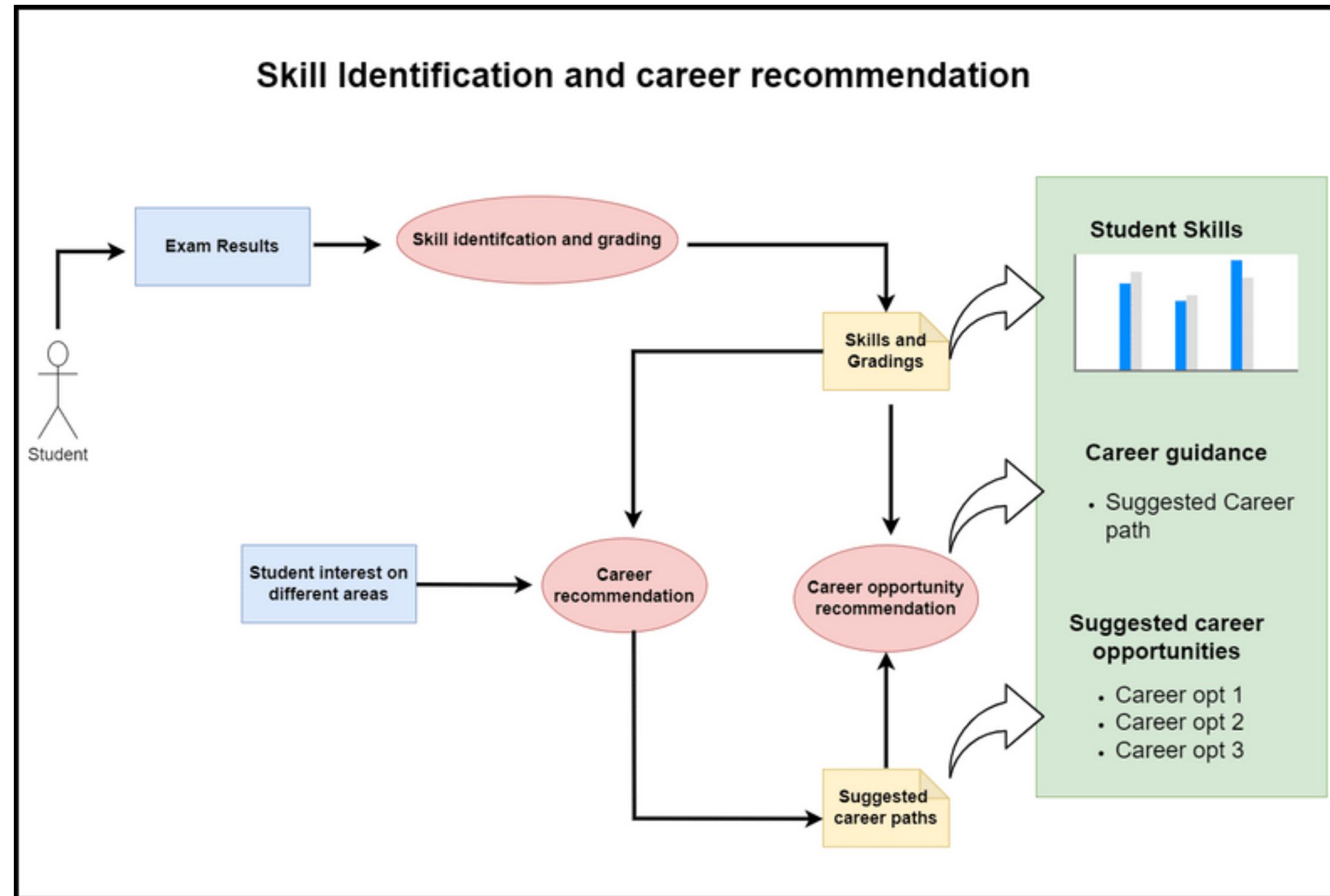
Descriptive input on the student can be used to get the most accurate outcome

Compare similarity between job posting skill and student skills



2022-017

USER FLOW IN HIGH LEVEL



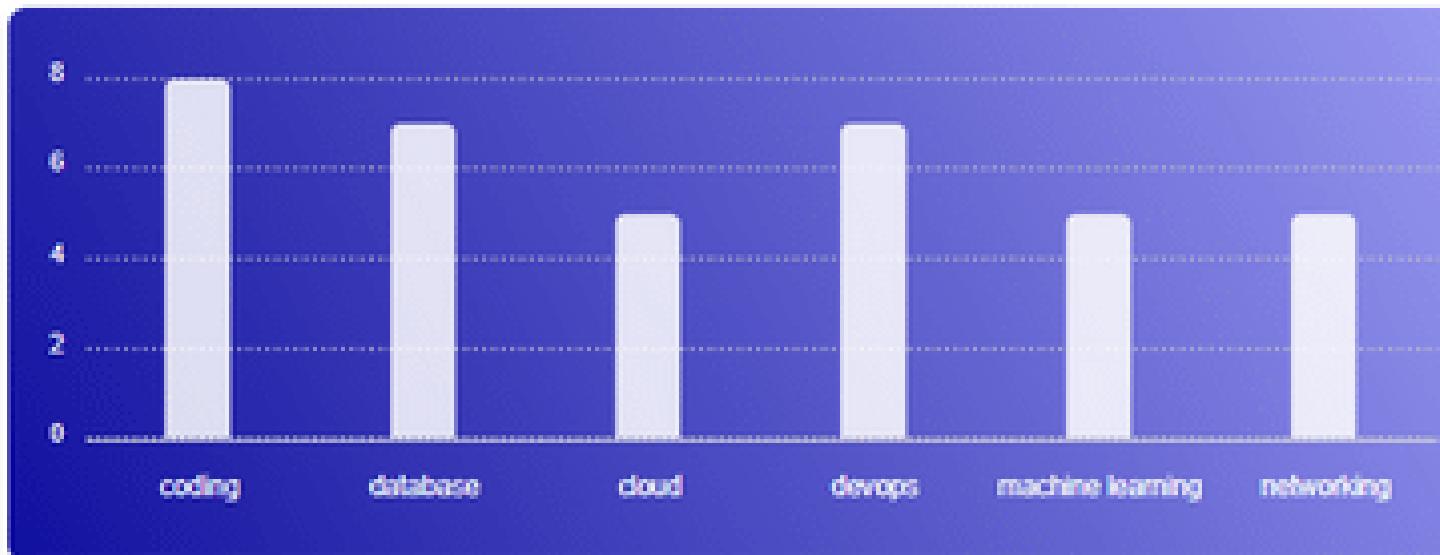


2022-017

EVIDENCES FOR THE COMPLETION



Performance Chart



Suggested career: Web Developer

- Frontend developer
- backend Developer
- Full stack developer

Current job openings

- Senior Software Developer - Data Team
Granify
- Coop Work Experience Student - System Analyst / Data Analyst
Alberta Health Services
- Data Analyst (3 Month Contract)
Robert Half Technology
- Data Engineer
Dotdash
- Data Analyst (3 Month Contract)
Robert Half Technology
- Data Scientist
Granify
- Data Analyst
T.D. Systems Inc.



2022-017

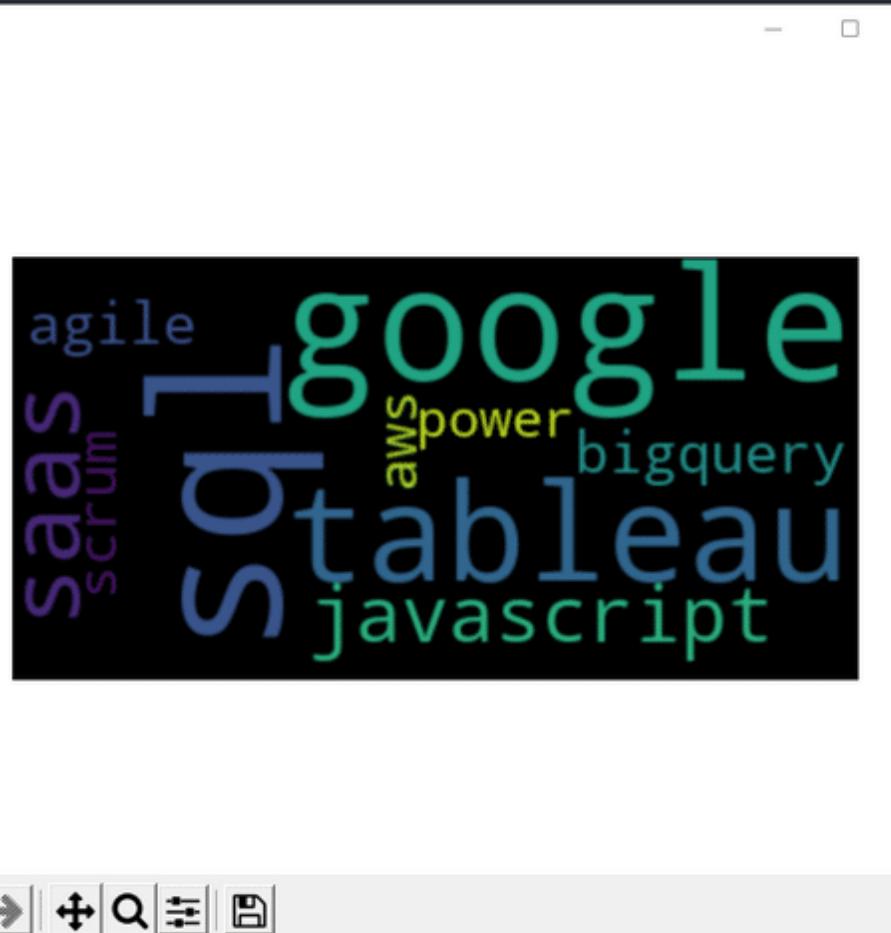
EVIDENCES FOR THE COMPLETION



The screenshot shows a Python code editor with several tabs at the top: skill_identification.py, job_seeker.py, views.py, and result.json. The main code area contains a class definition with methods for calculating Jaccard similarity and comparing job skill sets. A specific method, cal_similarity, includes code to generate a word cloud visualization. A modal window titled 'Figure 1' displays a word cloud with various tech terms like 'google', 'bigquery', 'aws', 'javascript', 'tableau', 'sql', 'sas', 'power', 'scrum', 'agile', and 'sas'. Below the code editor, there are two command-line windows showing directory paths.

```
skill_identification.py x job_seeker.py x views.py x {} result.json x
34     def get_jaccard_sim(self, x_set, y_set):
35         intersection = x_set.intersection(y_set)
36         return float(len(intersection)) / (len(x_set) + len(y_set) - len(intersection))
37
38     def cal_similarity(self, location=None):
39         num_jobs_return = 10
40         similarity = []
41         j_info = self.jobs_info_df.loc[self.jobs_info_df['location'] == location].copy() if len(
42             location) > 0 else self.jobs_info_df.copy()
43         if j_info.shape[0] < num_jobs_return:
44             num_jobs_return = j_info.shape[0]
45         for job_skills in j_info['keywords']:
46             similarity.append(j_info['similarity'])
47         top_match = j_info[similarity].idxmax()
48
49         all_keywords_str =
50         wordcloud = WordCloud()
51         plt.figure()
52         plt.imshow(wordcloud)
53         plt.axis("off")
54         plt.show()
55
56         return top_match
57
58     def extract_inhs_keywords(self):
59         skill_extraction > cal_similarity()
```

object\AITor\2022-017\co_recommender
il.
object\AITor\2022-017\co_recommender
il.



Word cloud of extracted skills from job postings



EVIDENCES FOR THE COMPLETION

```
skill = ["git, html, php, css, java, flutter, docker, react"]

val = nb.predict(skill)
val2 = sgd.predict(skill)
val3 = logreg.predict(skill)

print("Naive Bayes: ", val[0])
print('accuracy %s' % round(accuracy_score(y_pred, y_test),2) + '\n')

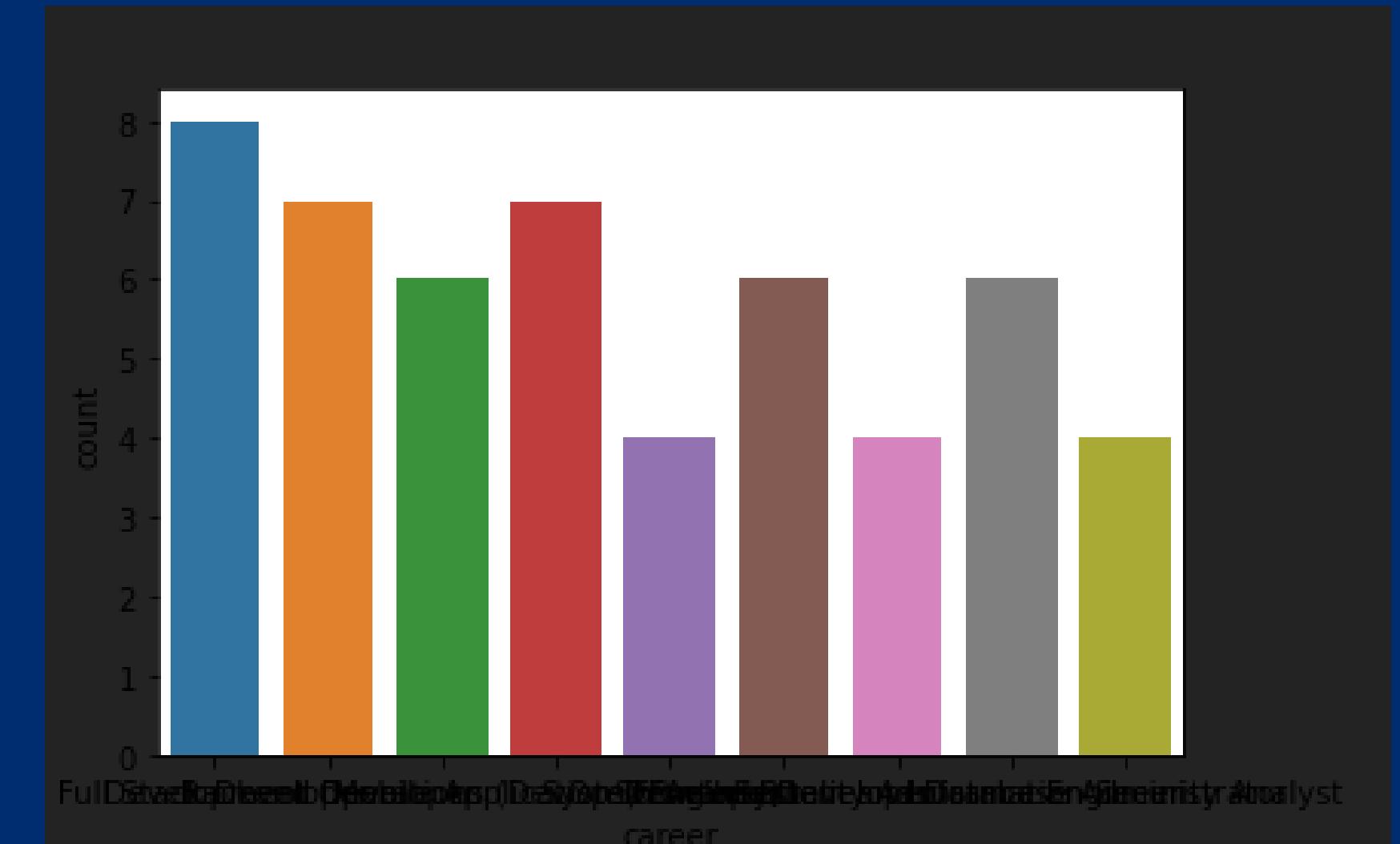
print("Linear Support Vector Machine:", val2[0])
print('accuracy %s' % round(accuracy_score(y_pred, y_test),2) + '\n')

print("Logistic Regression: ", val3[0])
print('accuracy %s' % round(accuracy_score(y_pred, y_test),2))
```

```
Naive Bayes: Full Stack Developer
accuracy 0.91

Linear Support Vector Machine: Full Stack Developer
accuracy 0.91

Logistic Regression: Full Stack Developer
accuracy 0.89
```



Training data count plot

Prediction with different models



USER/FUNCTIONAL REQUIREMENTS

- **Functional Requirements**

- Evaluate and identify student technical skills.
- Recommend most suited job opportunities based on identified skills
- Identify the most suited career path based on the student skills and the interest.

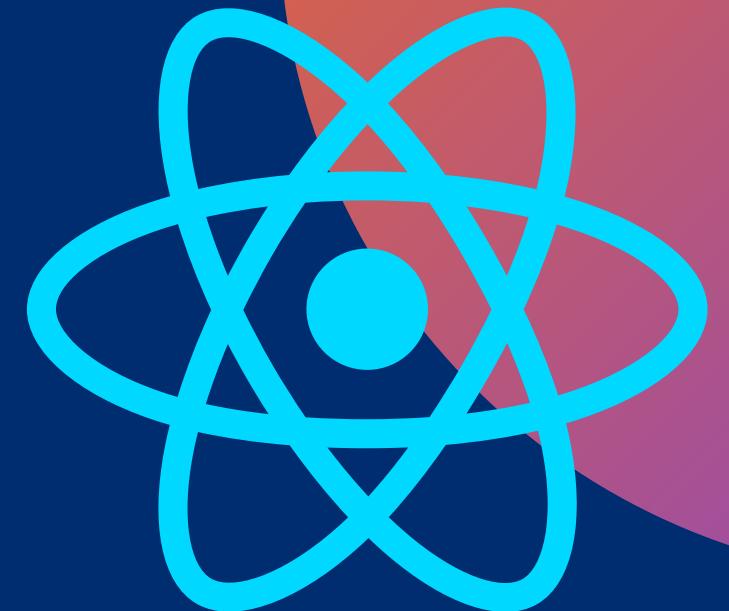
- **Non-functional requirements**

- Students data privacy need to be secure



Usage of technologies in the relevant key pillar/area

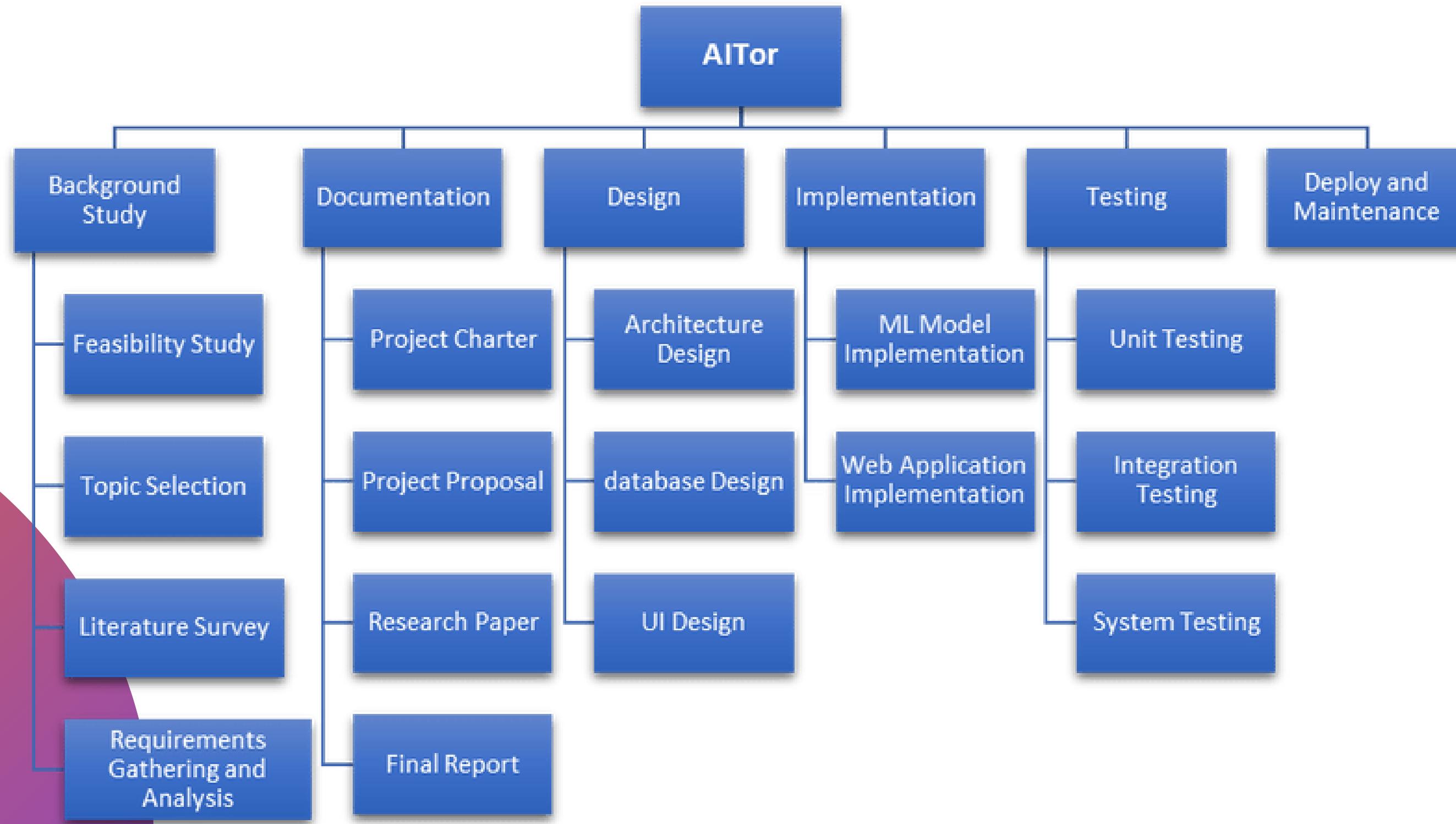
- Python Django will be used in ML model creation
- Scikit learn library will be used to implement ML model
- Initial data preprocessings and Model evaluations is done using Pandas, Numpy and Jupyter notebooks





2022-017

Work Breakdown Structure





2022-017

Project Gantt Chart



Task Name	January	February	March	April	May	June	July	August	September	October	November
Feasibility Study	■										
Topic Selection	■										
Background study and literature survey		■									
Proposal draft		■									
Proposal presentation		■									
Proposal report		■									
Data Gathering			■	■	■						
Design / Planning			■	■	■						
Strategy Selection			■								
ML Model Initialization				■	■						
Model Finalization				■	■						
Job recommendation engine					■	■					
Skill identification module					■	■					
Web API Integration					■	■					
Skill pool creation						■					
Exam module integration						■	■				
UI Implementation							■				
Web Interface Integration							■	■			
Unit Testing							■	■			
Bug Fixing and Improvements							■	■	■		
System Integration							■	■	■		
Integration Testing							■	■	■		
Bug fixing							■	■	■		
Final Presentation							■				
Final Report								■			



2022-017



LIYANAGE N.L.T.N.

IT19188546

FOCUSED AREA: Analyze the student performance, forecast and generate a fully detailed report.

SPECIALIZED IN SOFTWARE
ENGINEERING





PROBLEM DEFINITION

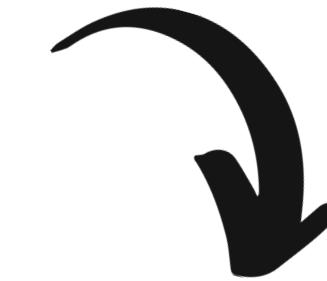
01

**ANALYZE THE PERFORMANCE
OF THE STUDENTS**



02

**CONDUCT A TIME SERIES FORECASTING
FOR THE ANALYZED DATA**



03

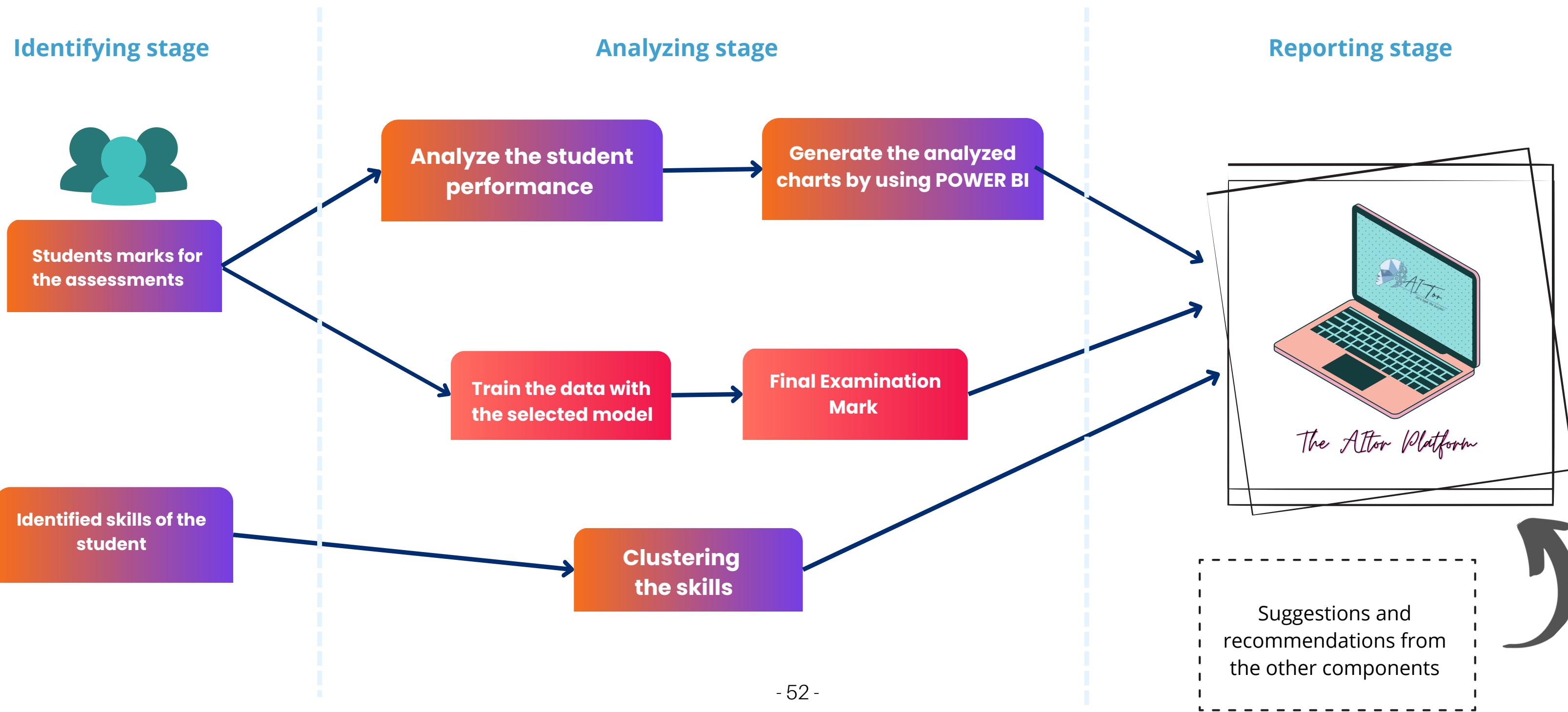
**GENERATE A FULLY DETAILED
REPORT IN THE SENCE OF
INDIVIDUAL AND GROUP**



2022-017



HOW TO ANALYZE, FORECAST AND REPORT THE STUDENT DATA





2022-017

What's in the 50%



- 01 Ideal Model Selection and Evaluation
- 02 Time-series forecasting algorithem creation
- 03 Initial data generate & model training with defined model
- 04 Design of the student profile user interfaces

50%



2022-017

APPLICATION OF KEY PILLARS IN THE SPECIALISED AREA



01

Strategies used to analyze the student

Time-Series analyzing / Performance analyzing by tagging the students

02

Algorithms available for the forecast

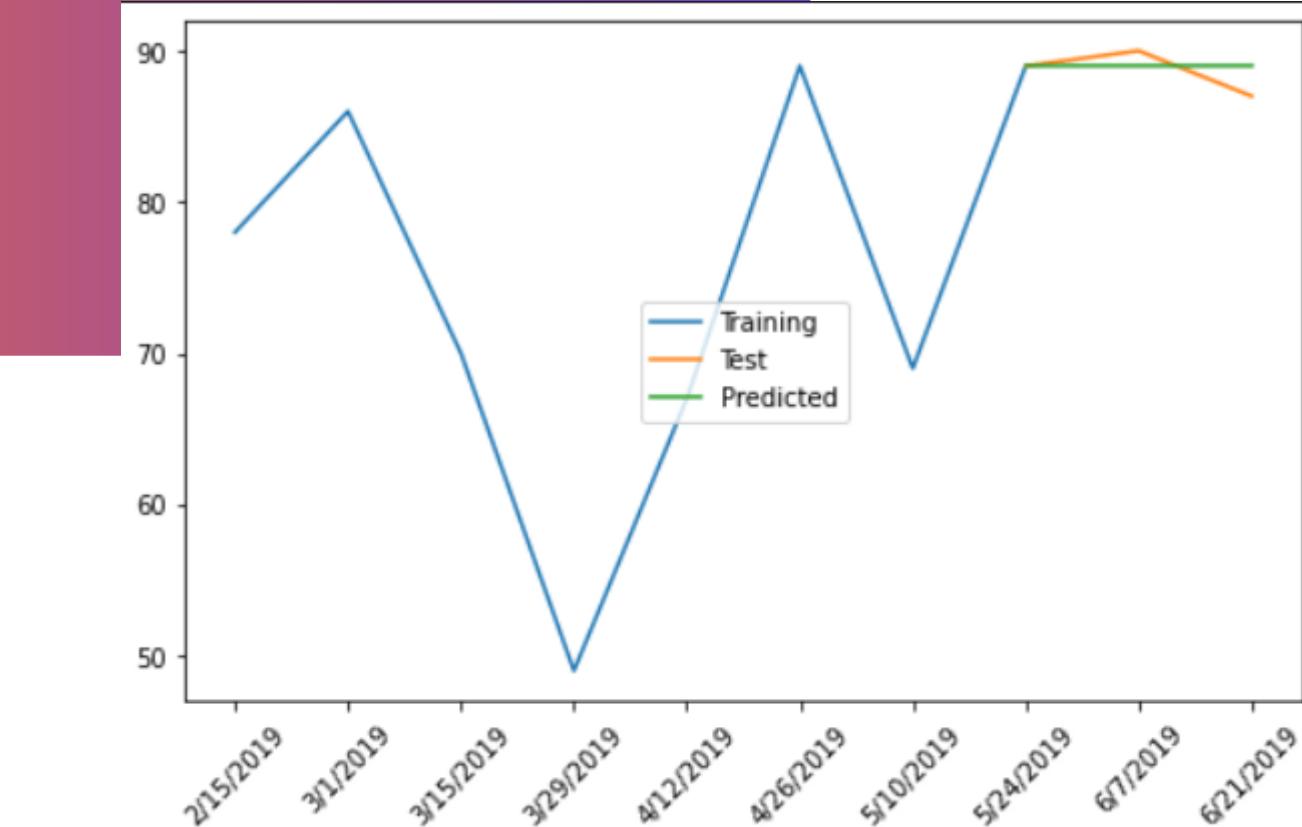
AUTO ARIMA MODEL

LSTM Model

ARIMA and SARIMA models

TBATS Model

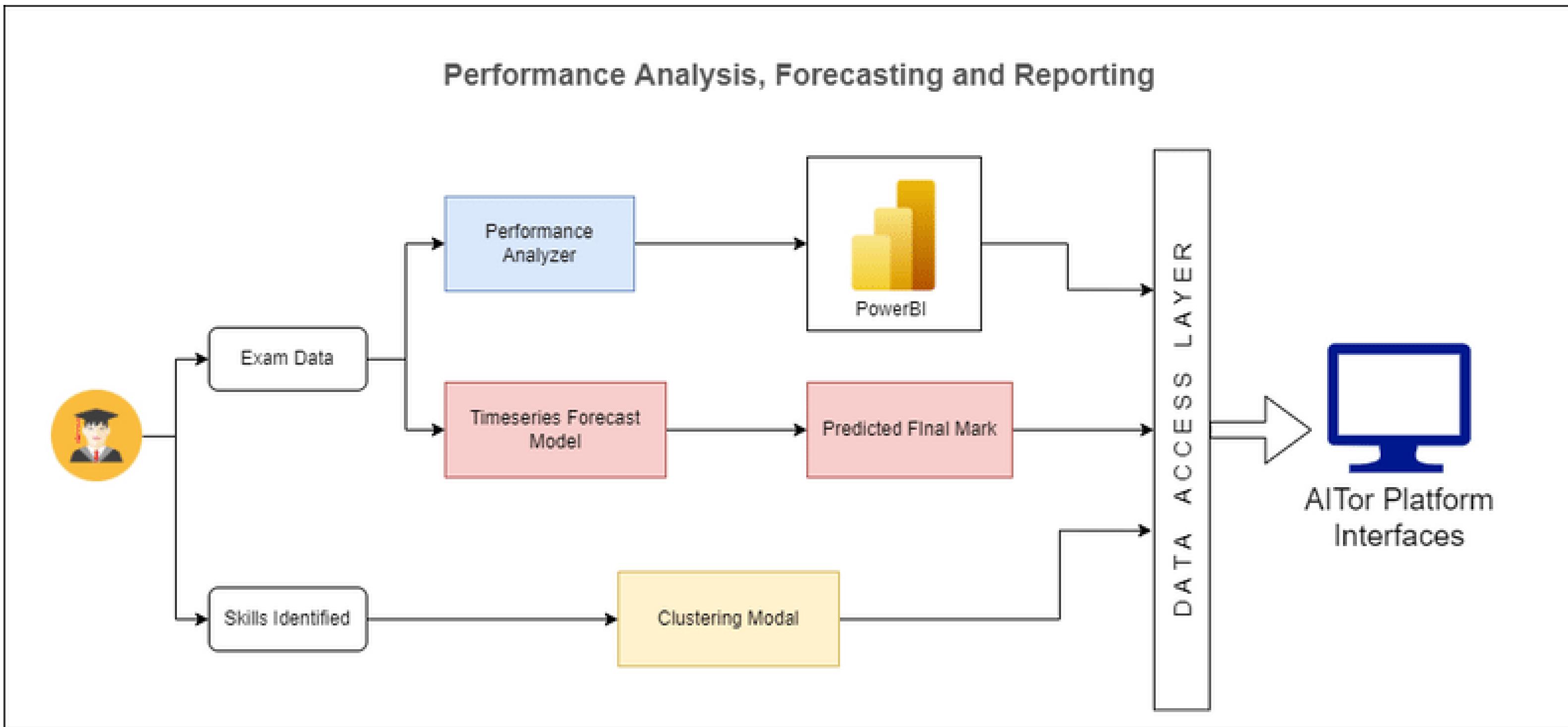
VARMAX Model





2022-017

USER FLOW IN HIGH LEVEL





2022-017

EVIDENCES FOR THE COMPLETION



```
[ ] import pandas as pd  
import numpy as np  
import matplotlib as plt  
%matplotlib inline  
  
[ ] # Load libraries  
import pandas as pd  
# Import Decision Tree Classifier  
from sklearn.tree import DecisionTreeClassifier  
#Import train_test_split function  
from sklearn.model_selection import train_test_split  
#Import scikit-learn metrics module for accuracy calculation  
from sklearn import metrics
```

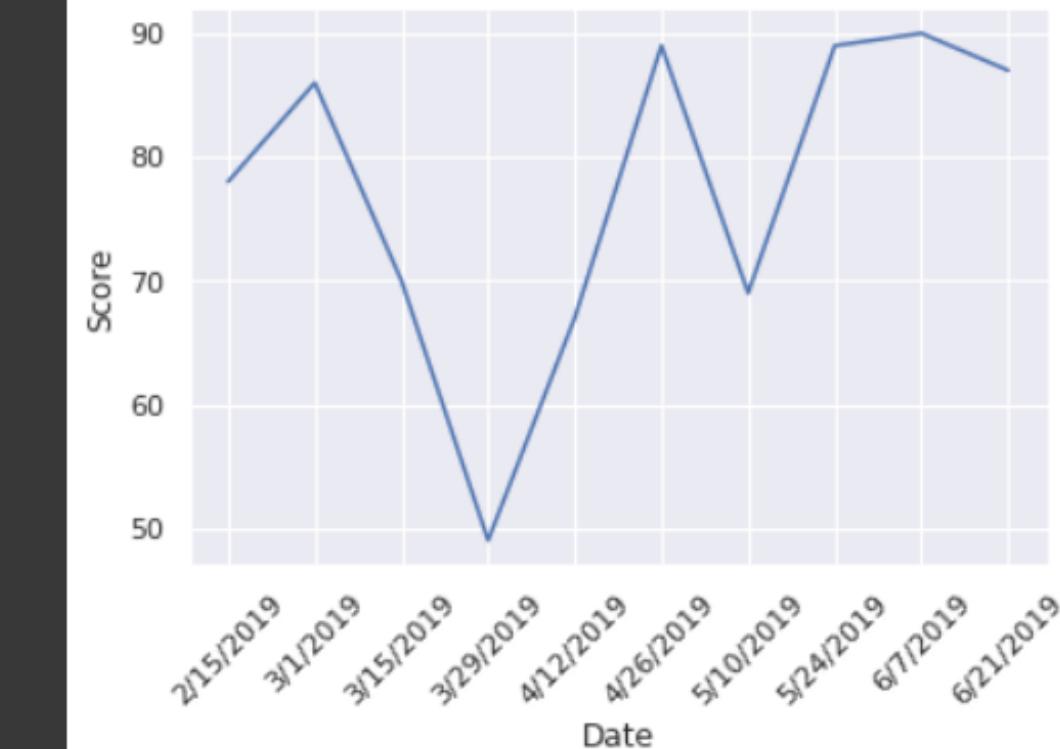
```
[ ] dataset = pd.read_csv('/content/train_data.csv')
```

```
dataset.head()
```

	Date	Score
0	2/15/2019	78
1	3/1/2019	86
2	3/15/2019	70
3	3/29/2019	49
4	4/12/2019	67

```
[ ] import matplotlib.pyplot as plt  
import seaborn as sns  
  
[ ] sns.set()  
  
[ ] plt.ylabel('Score')  
plt.xlabel('Date')  
plt.xticks(rotation=45)  
  
plt.plot(dataset['Date'], dataset['Score'], )
```

```
[<matplotlib.lines.Line2D at 0x7fc80de5cb10>]
```





2022-017

EVIDENCES FOR THE COMPLETION



```
train = dataset[dataset['Date'] < "5/10/2019"]
test = dataset[dataset['Date'] > "5/10/2019"]

print(test)

train_date = train['Date'].to_numpy()
train_score = train['Score'].to_numpy()

test_date = test['Date'].to_numpy()
test_score = test['Score'].to_numpy()

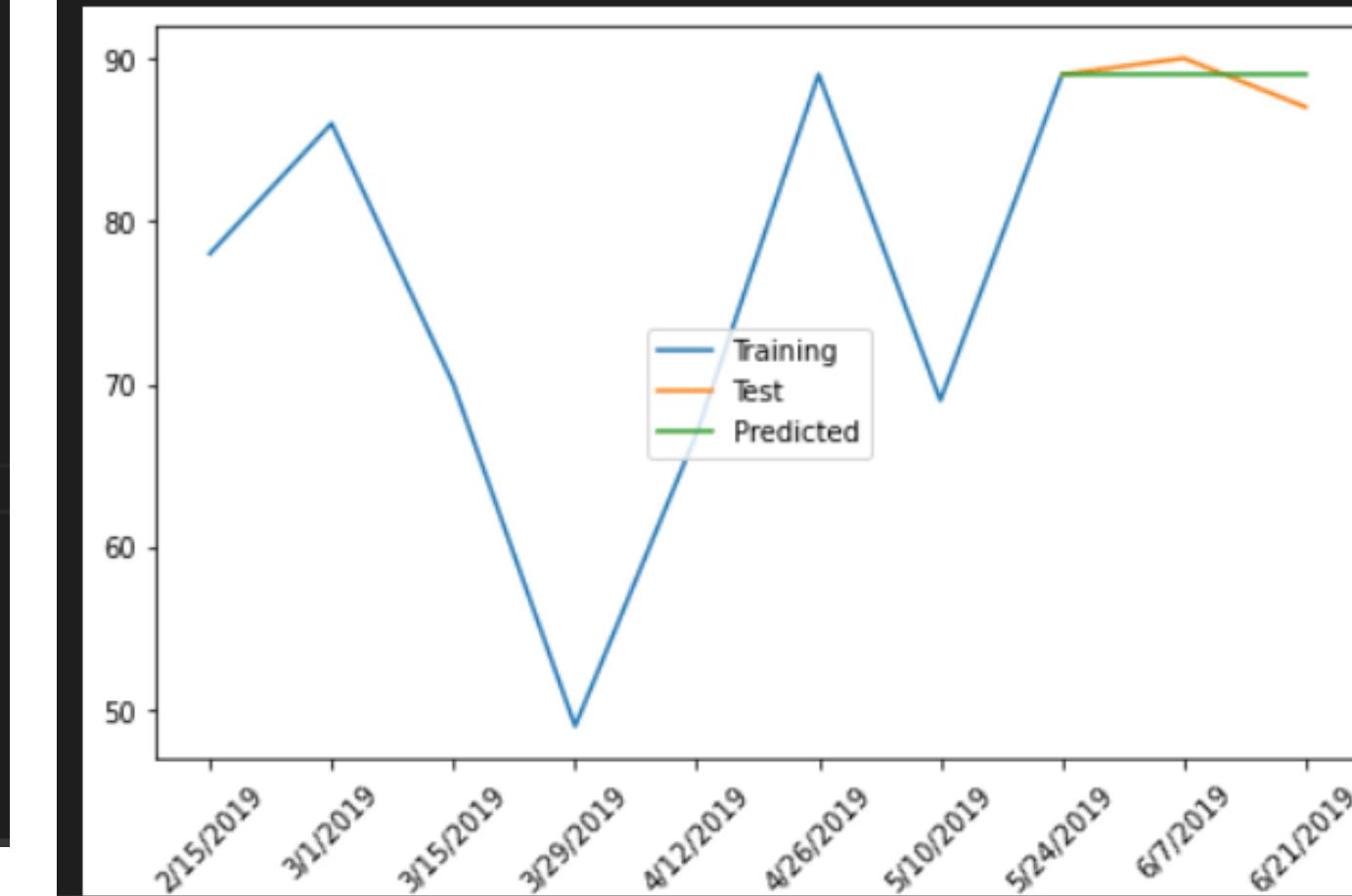
print(test_date.shape)

plt.plot(train_date,train_score, color = "blue")
plt.plot(test_date,test_score, color = "red")
plt.ylabel('Score')
plt.xlabel('Date')
plt.xticks(rotation=45)
plt.title("Train/Test split for Student Marks Data")
plt.show()
```

```
plt.figure(figsize=(8,5))
plt.plot(train,label="Training")
plt.plot(test, label="Test")
plt.plot (prediction, label="Predicted" )
plt.legend( loc = 'center')

plt.xticks(rotation=45)
plt. show()

✓ 0.3s
```





2022-017

EVIDENCES FOR THE COMPLETION



STUDENT PROFILE

CGPA : 3.99

Student Name
Study Area

STUDENT PROFILE
CURRENT SUBJECTS
PERFORMANCE ANALYSIS
FINAL MARK FORECASTING

SKILLS IDENTIFIED

Tag Name Tag Name
Tag Name Tag Name
Tag Name

Copyright © 2022 Team AITor. All rights reserved.

LOGOUT

STUDENT PROFILE

CGPA : 3.99

Student Name
Study Area

Previous Results

Semester	Subject Code	Subject
Year 1 Sem 1	IT1010	Subject Name 1
Year 1 Sem 1	IT1010	Subject Name 1
Year 1 Sem 1	IT1010	Subject Name 1
Year 1 Sem 1	IT1010	Subject Name 1
Year 1 Sem 1	IT1010	Subject Name 1
Year 1 Sem 1	IT1010	Subject Name 1

Copyright © 2022 Team AITor. All rights reserved.

LOGOUT

STUDENT PROFILE
CURRENT SUBJECTS
PERFORMANCE ANALYSIS
FINAL MARK FORECASTING

SKILLS IDENTIFIED

Tag Name Tag Name
Tag Name Tag Name
Tag Name

Mark Deviation Along With The Average

Copyright © 2022 Team AITor. All rights reserved.



USER/FUNCTIONAL REQUIREMENTS

- **Functional Requirements**

- Conduct a performance analysis based on the student exam mark.
- Forecast the final examination marks based on the other examinations.
- Generate a fully detailed report which helps both the tutor and the student.

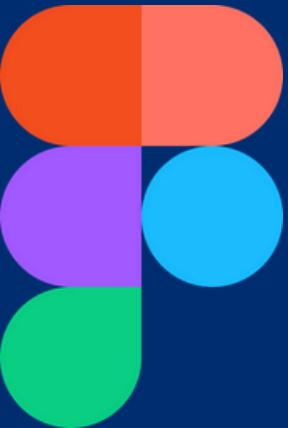
- **Non-functional requirements**

- Ensuring the security of the student's data when reporting the outcome.



Usage of technologies in the relevant key pillar/area

COMPONENT/MODULE	TECHNIQUES	RULES AND ALGORITHMS	TECHNOLOGIES
TS Forecasting model	Timeseries Forecasting	ARIMA	Python, Scikit-learn, Pandas, Numpy, Django
User Interface for Student Profile	ui/ux		Figma





2022-017

PROJECT GANTT CHARTT

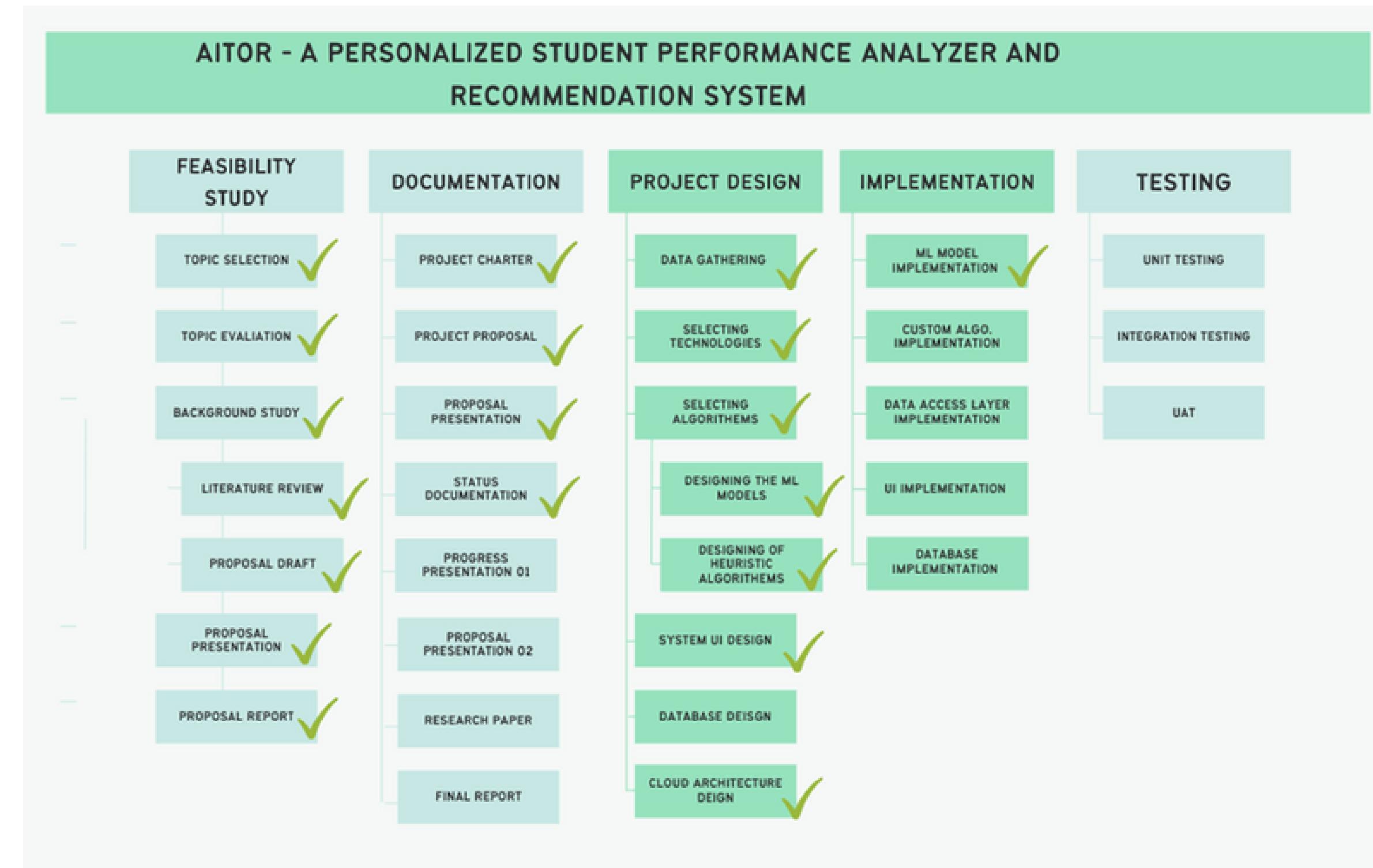


Task	Semester 1						Semester 2					
	January	February	March	April	May	June	July	August	September	October	November	December
Feasibility study	Green											
Topic Selection	Light Green											
Topic Evaluation												
Background Study	Dark Grey	Dark Grey										
Background study and literature survey		Light Green										
Proposal Draft		Light Green										
Project Proposal	Dark Grey	Dark Grey	Dark Grey									
Proposal presentation		Light Green										
Proposal Report												
Project Initiation	Dark Grey	Dark Grey	Dark Grey									
Data gathering			Light Green									
Selecting algorithms and technologies			Light Green									
Project Implementation	Dark Grey	Dark Grey	Dark Grey	Dark Green	Dark Green	Dark Green	Dark Green	Dark Green	Dark Grey	Dark Grey	Dark Grey	Dark Grey
Processing data				Light Green								
Implementation of the Time-series forecasting model				Light Green								
Design the Student Profile					Light Green							
Progress Presentation 01					Light Green	Blue						
Implementing the progress analyzing functions						Blue						
Research paper						Blue						
Data Access Layer Implementation						Blue						
PowerBI Integration						Blue						
Unit Testing						Blue						
Evaluation and Error fixing						Blue						
Front end development						Blue						
Final Stages	Dark Grey	Dark Grey	Dark Green	Dark Green	Dark Green	Dark Grey	Dark Grey	Dark Grey				
Progress Presentation 02							Blue					
System Testing							Blue					
Evaluation and Error fixing							Blue					
Final Presentation and Viva								Blue				
Final Report								Blue				



2022-017

WORK BREAKDOWN CHART



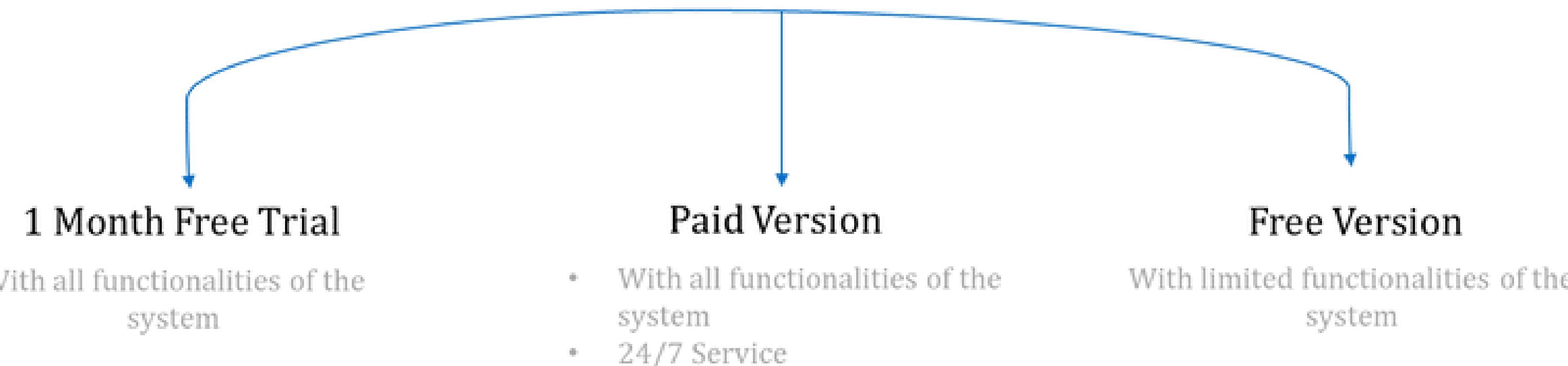


2022-017

Commercialization



What we offer?





2022-017

VALUE PREPOSITION



Package	Features Included	Price
Pro starter	<ul style="list-style-type: none">• Student performance analyser• Target for educational institutes having student count of below 1000	USD 1500 per month 1.5 USD per Student
Prod Ultimate	<ul style="list-style-type: none">• Student performance analyser• Carrier recommendation• Final grade forecasting and Tutor guide• Target for educational institutes having student count of over 1000 ~ 3000	USD 3000 per month ~ 1.5 USD per Student

Thank You!

The screenshot displays the AITor student management system interface. On the left, there's a sidebar with navigation links: STUDENT PROFILE, CURRENT SUBJECTS, PERFORMANCE ANALYSIS, and FINAL MARK FORECASTING. The main area shows a student profile for a user named 'Esther' with a CGPA of 3.99. It includes sections for STUDENT PROFILE, CURRENT SUBJECTS, and PERFORMANCE ANALYSIS. The PERFORMANCE ANALYSIS section contains a chart titled 'Mark Deviation Along With The Average' showing a line graph from May 27 to June 2. The FINAL MARK FORECASTING section shows a table of skills identified for various subjects.

Subject	Grade	Average Mark
Subject Name 1	A	78.5
Subject Name 2	A	78.5
Subject Name 3	A	78.5
Subject Name 4	A	78.5
Subject Name 5	A	78.5

FROM TEAM,



| 2022-017