FINAL PROJECT REPORT

Team: *Average and Savage*

Parth Kapadia, Sanjana Parakh, Oscar Mui, Amit Birajdar
{parthjil, svparakh, omui, abirajda}@usc.edu

University of Southern California

Project Title	User Recommendation System for Data Enthusiasts
Date Started	08-22-2022
Date Completed	11-28-2022
Project Sponsor	Dr. Anna Farzindar

Table of Contents

Executive Summary

The advent of the internet boom has led to a hyper-competitive space, with organizations seeking competitive advantage over the others to boost retention and sales. There has been a massive increase in the use of recommender systems to provide a more customized and engaging user experience to clients. One of the benefits of using a recommendation system is its ability to understand and predict user interest and behavior, and then make recommendations based on these insights. In addition, the informed decisions suggested by the recommender would typically reduce the time required to find another user with similar goals and significantly increase the probability of discovering other users that match one's interest, thereby resulting in increased loyalty and satisfaction of users.

The goal of our project is to provide users with a personalized list of other users that they are likely to be interested in for work collaboration. The idea is to create a platform to help users easily find fellow collaborators, removing the hassle and struggle of finding fellow team members that match the required skills and background and are equally enthusiastic about working together on a particular project.

Our Mission:
To create a platform designed to help users easily find fellow collaborators, removing the hassle and struggle of finding fellow team members that match the required skills and background and are equally enthusiastic about working together on a particular project.

Lean Six Sigma Project

We employ the DMAIC methodology for this project, which organizes the sequencing of activities in five categories: Define, Measure, Analyze, Implement, Control. The report delves into each of these categories. Apart from these categories, this report accounts for an additional category to describe the technical details regarding the machine learning approaches utilized as well as system implementation, prototype & demonstration.

1. Define Phase

The define phase emphasizes on the problem description and the process quantitatively to accurately determine the way the performance of the system will be measured.

1.1. Customer Satisfaction

How do you currently find fellow collaborators for a project?

- ➔ As a student I usually end up collaborating with my friends for any project as it's easier to approach them.
- ➔ I have also approached people by leveraging my friends and families network.

What are the issues that you currently face while choosing collaborators for a project?

- ➔ Sometimes, it's difficult to reach out to other people with different interests and skill sets, cold messaging mostly is a shot in the dark as most people don’t reply back.
- ➔ A lot of people are currently not available/interested in working on a project which makes the process more difficult.
- ➔ It’s difficult to approach people who are not in my field for an interdisciplinary project as I don’t have access to them.

What are your requirements for a recommendation platform?

- ➔ The UI should be reliable and easily accessible and should be constantly updated to show which students/collaborators are currently available to join a project.
- ➔ The platform should include a feature for users to showcase some of their work or link it to their github or kaggle pages to lend more authenticity to the person and their skills.
- ➔ It would be nice to include a filter for domain (ex. Finance, Tech, etc.) because domain is generally more important than tools, programming languages, etc.
- ➔ A system that displays a ranked-list of recommendations, so that similarity scores are clearly visible to the user.

- ➔ A chat feature on the website would allow users to interact with people who have been tagged as highly similar by the system.

Based on all the feedback that we have received, we will try to incorporate all suggestions in our system and ensure that we deliver a complete one stop solution for our stakeholders to ease the process of finding fellow collaborators based on their requirements and make the overall process hassle free.

1.2. Tools Application

During the define phase of the project, the project charter turned out to be the most valuable document as a well defined, concise document providing a high-level view of all the elements involved in the project enabled clear communication with the stakeholders and better understanding of their requirements. At the beginning of the project, where the details aren’t concrete and there is uncertainty surrounding the project, such feedback gathering is quite valuable. Additionally, a clear understanding of the requirements, responsibilities, limitations, and scope of the project aides in starting right.

2. Measure Phase

The measure phase involves using different metrics and measures to gauge the performance of the system and understand any improvement opportunities.

2.1. Process Mapping

The workflow of this project has been depicted using various different kinds of process maps. These maps include SIPOC (appendix 1), High Level Process Map (appendix 2), Common Process Map (appendix 3), Detailed Process Map (appendix 4) and a Functional Process Map (appendix 5). The entire process of illustrating these process maps was instrumental in learning more about each step of the workflow including the inputs and outputs outlined at each step.

This was crucial in devising a workflow to integrate the two models that have been developed into a single entity that can be easily used. All the efforts made in creating the SIPOC served as a foundation for creating a list of tasks which have been elaborated in the Detailed Process Map. This facilitated the team to follow a structured plan of continuous development and provided checkpoints for the various tasks.

2.2. The Vital Few

There are various factors that affect the user recommendation, apart from the different quantifiable factors user recommendations tend to be affected by other factors as well. However, in order to constrict the scope of this project we have limited our system to include only quantitative features. Additionally, the scope is further narrowed down to include only the features pertaining to the academic and work career of any individual. We have also decided to gather data only from LinkedIn since gathering user data across various platforms isn’t feasible.

2.3. Data Exploration and Preprocessing

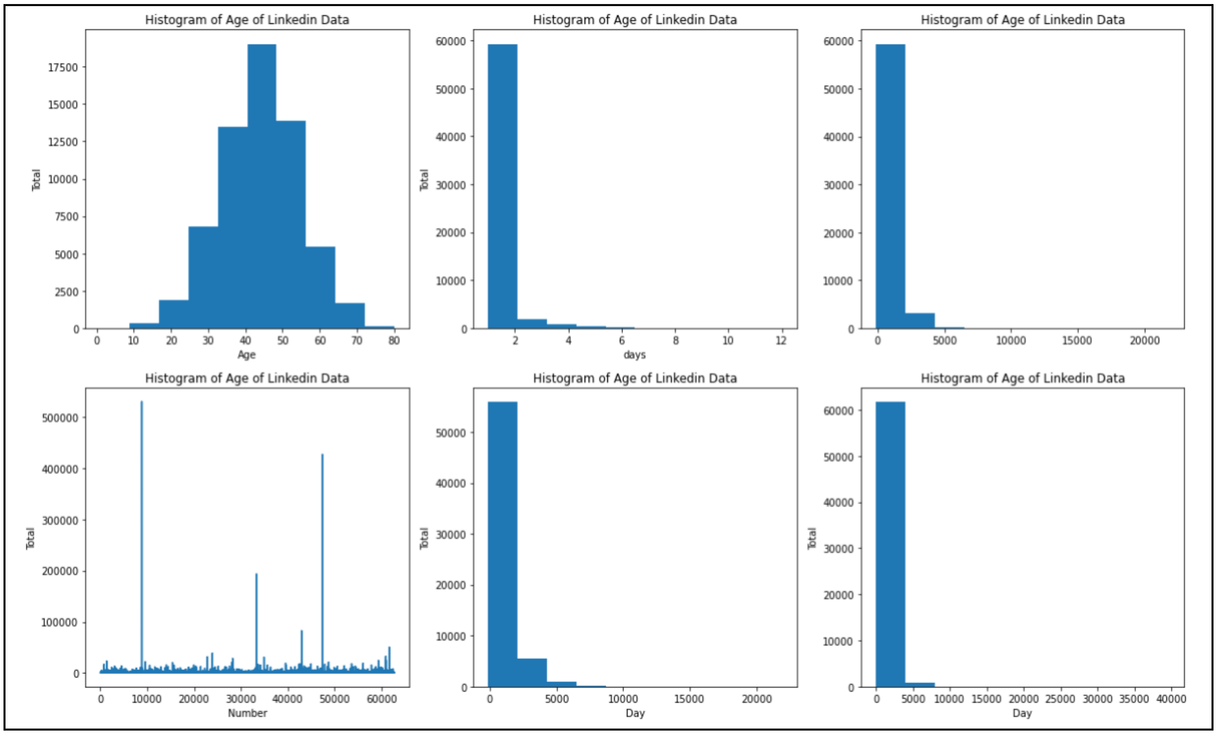
Our dataset for user features comes from a Kaggle dataset of LinkedIn profile data containing information about profile and job data.

For a detailed description of the metadata or to download the dataset, refer to the Kaggle dataset listed below.

Link: https://www.kaggle.com/datasets/killbot/linkedin

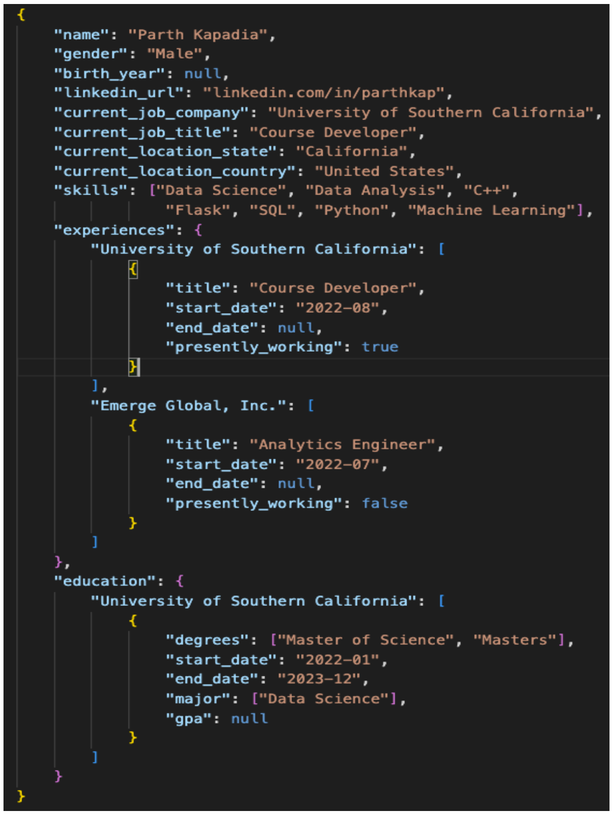
Dataset Size: 62709 people, 52 features

	n_pos	avg_pos_len	tenure_len	age	n_followers
count	62709	62709	62709	62709	62709
mean	1.243505717	765.6571887	962.7278381	44.04871709	1225.838173
std	0.756841051	750.7252101	1088.312164	10.68384209	6406.523908
min	1	-120	-120	1	0
25%	1	274	304	37	405
50%	1	578	640	44	725
75%	1	1035	1217	51	1244
max	12	21884	21884	80	530566



The LinkedIn dataset contains numerous features, including profile data, employment data, race, gender, age, gender, and more. We find that the average user in our dataset is 44 years old, has over 1200 followers, and has an average aggregate tenure of 962 days. Additionally, each user has had an average of 1.24 different company positions with an average tenure of 765 days per company. Therefore, we conclude that the average user in the LinkedIn dataset tends to be a middle-aged employee with a large social network and roughly 2-3 years of experience.

- **LinkedIn profile data fetched using peopledatalabs.com linkedIn API:**



```
{
    "name": "Parth Kapadia",
    "gender": "Male",
    "birth_year": null,
    "linkedin_url": "linkedin.com/in/parthkap",
    "current_job_company": "University of Southern California",
    "current_job_title": "Course Developer",
    "current_location_state": "California",
    "current_location_country": "United States",
    "skills": ["Data Science", "Data Analysis", "C++",
                "Flask", "SQL", "Python", "Machine Learning"],
    "experiences": {
        "University of Southern California": [
            {
                "title": "Course Developer",
                "start_date": "2022-08",
                "end_date": null,
                "presently_working": true
            }
        ],
        "Emerge Global, Inc.": [
            {
                "title": "Analytics Engineer",
                "start_date": "2022-07",
                "end_date": null,
                "presently_working": false
            }
        ]
    },
    "education": {
        "University of Southern California": [
            {
                "degrees": ["Master of Science", "Masters"],
                "start_date": "2022-01",
                "end_date": "2023-12",
                "major": ["Data Science"],
                "gpa": null
            }
        ]
    }
}
```

When a user inputs his name and email address associated with his LinkedIn account on the web application, we would fetch his profile information using the API from peopledatalabs.com

Using this information, we would match or provide recommendations on other users having similar interests, complimenting skills and those fitting the user's preferences.

We match the data in the dataset with this fetched information with features like:

Gender, nationality, age (calculated using *birth_year*), skills, current_job_title, work_experience, avg_no_of_days_per_role, etc.

2.4. Tools Application

The peopledatalabs.com linkedIn API and the Kaggle LinkedIn profile dataset were the most crucial tools in this phase of the project. They provided us with all the data that we needed to train and evaluate our machine learning models. Access to these tools was extremely beneficial as it provided us with the required data and saved us time procuring it from other sources, in turn allowing us to focus on other aspects of the project.

In addition, the process maps served an invaluable role helping us clearly define the inputs and outputs of each stage. The maps helped devise a workflow and break down the entire process into manageable modules.

3. Analysis Phase

3.1. Selecting Charts for Analysis

In accordance with the lean philosophy, we carried out a logical flow of tasks to find the project's fundamental issues. Three charts were chosen for the root cause investigation. The first is a Pareto Chart, which allowed the team to pinpoint and identify the biggest issues that could have contributed to project failure using the 80/20 rule. The Fishbone Diagram then assisted in identifying the precise elements that feed the problem-prone areas. Last but not least, the 5 Whys method illuminates glaring issues and aids in locating the real sources of error so that they can be fixed.

3.2. Value Function

Understanding how the project at hand communicates its value to the clients is essential to examining the underlying causes of any problems that may arise during the assignment. The transformation of a number of data sources into the optimal yield should be observable as the process of creating value. Prior to moving on with the underlying driver examination, it is beneficial to specifically define the task's worth capabilities.

The worth capability for this particular project can be understood as:

Python Package = func (Datafetch from Linkedin API, Preprocess Data, Model Evaluation, User Interface, Server Deployment)

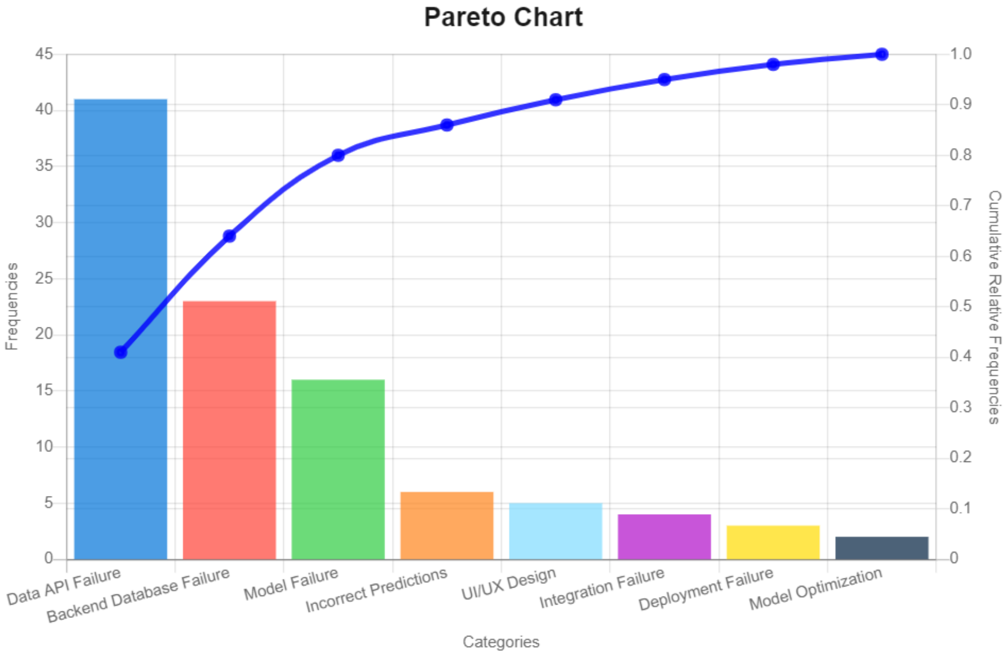
The user enters their Email Address and Name in the UI from which API request is made on the backend where linkedin data for that user is received. The data is preprocessed in the backend and passed through a model which recommends the user and this POST request is sent to the our UI where the user is recommended to other users in the database with whom our user can collaborate.

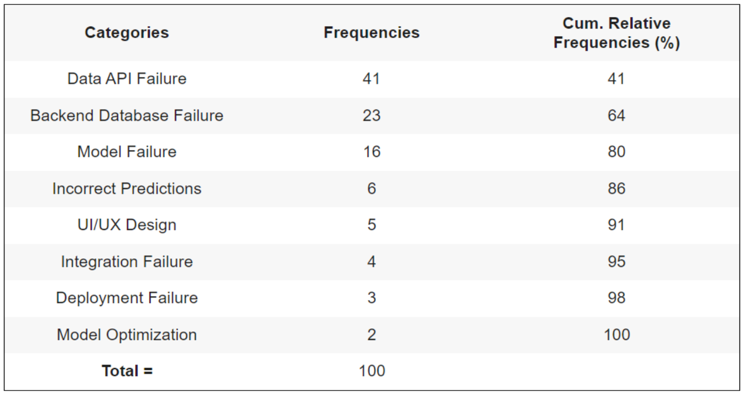
3.3. Root Cause Analysis

We must first be aware of a framework's current state in order to enhance it further. To illustrate the project's current condition, our team made use of the thoughtfully designed Lean Six Sigma tools. As previously mentioned (in 3.1. Choosing Diagrams for Examination), our group combined three different representation diagrams to achieve that goal. We nevertheless produced a Pareto graph. It ranks the venture's flaws by influence and estimates them both directly and indirectly. Then, using the

80/20 rule, we decided which 20% of the reasons could lead to 80% of the troubles. This made it easier to concentrate on the problems that required fundamental management. We identified three main reasons for model failure:

1. Data API Failure
2. Backend Database Failure
3. Model Failure





Categories	Frequencies	Cum. Relative Frequencies (%)
Data API Failure	41	41
Backend Database Failure	23	64
Model Failure	16	80
Incorrect Predictions	6	86
UI/UX Design	5	91
Integration Failure	4	95
Deployment Failure	3	98
Model Optimization	2	100
Total =	100	

Next, we generated a Fishbone Diagram (appendix 6), which looks at the five sources of project defects: Data API failure, Model Failure, Backend Database Failure, Model Evaluation, and People. For each of these categories, the team brainstormed which elements could have potentially driven towards the effect (Y) being analyzed, which was "Failure to deliver the product (i.e. Wrong user recommendation)". The highlighted challenges from this exercise were (1) Wrong evaluation technique, (2)API Request Error, and (3) Less Data to Train.

Finally, we performed a 5 Whys (appendix 7) analysis, which deals with looking at the immediately visible problems that can cause the team to deliver "Wrong User recommendation", and then asking why five times, in order to drive towards the root issue that is behind the superficially observable problems. Some common root causes were identified and are explored in further detail in the section below.

3.4. Sources of Variation

The charts mentioned in 3.3 indicates that the product variations might come from the following three measure:

a. **Linkedin Data API Failure**:
   i. This is the main source of user data and it is like a root to our tree hence its failure in any form cannot be anticipated.
   ii. The main issue occurs when there have been limited API calls and all of them are used up. In this case, the API will not return the data of the requested user.
   iii. To add on that, if the linkedin profile of the user whose data is requested is incomplete, then there will be many empty fields in the data which will need to be handled.

b. **Backend Database Failure:**
   i. If Linkedin API is root then the backend database is the stem of our tree. If under any circumstances, the backend fails then the whole system may collapse.
   ii. The system can fail under a DDoS attack where it gets a lot of requests in a short period of time.
   iii. The server can have a computational bottleneck where it can only handle a certain number of active users and a certain number of requests per minute.

c. Model Failure
   i. Determining which ML models to use is always a tough job, but there are many ways to fail to do so.
   ii. If we have less data than anticipated, the model may underfit resulting in generalizing all the recommendations.
   iii. Even if the data is very complex then our model might face errors.

3.5. Potential Solutions

The potential solution for the mentioned problem can be as follows:

- The data can be collected using different API keys so there is no failed request. Also, the preprocessing code should be written which converts all the data in a single generalized format and also deals with the Null values in the linkedin profile.
- The bandwidth of the server should be set according to the anticipated demand of users. Appropriate measures to keep the servers secure from any sort of Cyber attacks.
- The built model should be evaluated using A/B test and using a metric which is well suited for this User Recommendation approach.

3.6. Tools Application

The main driver examination was the root cause analysis tool used in this stage. This lean six sigma procedure provided important guidance in identifying fundamental issues and resolving those issues. The force of the 80/20 rule and the adequacy of surveying the effect of different separated issues with a bringing together measurement was likewise an important learning step that worked with smooth change to the following phase of the undertaken.

4. Improve Phase

In the Improve phase, we identify improvements for possible gaps and inefficiencies in the system and test these improvements to address the root cause.

4.1. Solution Evaluation

This section analyzes four key aspects of different solution routes along with their advantages and disadvantages.

Data Collection: Being vigilant about the source of data is extremely important to maintain the highest standards of data integrity and security.

Methods	Pros	Cons
Verified API source	Data fetched is trustworthy and data integrity, privacy is taken care of.	Fraudulent API sources may breach privacy and this might cause trouble
Use of relevant attributes	Irrelevant data clutter is avoided resulting in easier handling of fetched data and eliminating duplication of data.	Unnecessary data may be fetched resulting in wastage of resources. Use of irrelevant attributes may also lead to poor results by the machine learning models.
Tracking API Request Limit	Code can be written to accommodate for minimal repeat API requests hence, less requirement of monetary resources.	Redundant API requests may max out request limit and may result in account being blocked or higher cost.
Backup API sources	Decreases dependency on a single source for data.	Maintaining and tracking multiple data sources may be a hassle. Also, higher costs.

Data Preprocessing and Data Analysis: This is a critical step as it makes sure the input to the machine learning model is valid and of the highest quality. This results in a robust model and better performance.

Methods	Pros	Cons
Removing Null values	The most simple way to deal with null value.	If there are too many null values, we may lose a lot of information. Null values may also be meaningful.
Replacing Null values using Mean, Median or Mode	Replacing null values with Mean, Median or Mode will be better than completely removing them or will preserve the sequence of data.	Data may be skewed, leading to bias predictions.
Text-Preprocessing Methods (Removing Common Words and Special Characters)	Removes irrelevant characters and words which may help the model compare important text.	May accidentally delete important words or characters that add context to text.
Text-Preprocessing Methods (Extracting Keywords)	Allows us to filter out words that are important for the model, such as programming languages.	May extract words that are not important and bias the model.
K Best Feature Selection	It allows us to retain the top k features of X with the highest score. Also, helps	An additional regression model needs to be implemented over the entire database each time.

	avoid a situation where our model is overfit.	

Model Selection: Based on our use case, we select appropriate models and make sure they generalize well. It is important to start with a basic model, understand the results and incrementally use complex models.

Methods	Pros	Cons
Clustering (K-means)	Algorithms are not so complex and the results are interpretable. Can act as a baseline for complex models.	Dealing with a large number of dimensions can be problematic because of higher time complexity
Content-Based Recommendation - User & Item Based Recommendation	Can make relevant suggestions to new users based on their similar content to other users.	May be too difficult to implement given that our data does not have labeled targets. Data may be too sparse.
Model Metric Selection: Cosine Similarity	The cosine distance measurement is a basic measurement for distance and does not account for magnitude.	Recommendations can be poor since they are solely based on distance, not context. Magnitude is not accounted for.

Hybrid model: KMeans + Cosine Similarity	Uses the pros of both these models and outputs a more generalized set of results that works well with real (test) data	Sometimes, results can be restricted to a single cluster and lead to a lack of diverse results.

Model Evaluation:

Methods	Pros	Cons
Root Mean Squared Error	The output value you get is in the same unit as the required output variable which makes interpretation of loss easy.	It is not that robust to outliers as compared to MAE.
Raw Accuracy	Simplest evaluation metric to use and interpret.	May be skewed by imbalance datasets. Does not account for magnitude in errors.
nDCG	Best method for ranking recommendations for a recommender system	The NDCG does not penalize for bad documents in the results and it does not penalize missing documents in the results

4.2. Recommended Solution

After evaluating models based on A/B Testing and nDCG results, the hybrid model which uses KMeans and Cosine Similarity in conjunction, gives the best results among all the other models that were experimented with.

Data Collection and Preprocessing:

Action Items	People Responsible	Deadlines
Data Collection	Sanjana Parakh, Oscar Mui	9/15/2022
Data Preprocessing	Amit Birajdar, Parth Kapadia	9/25/2022
Fetching data from API	Amit Birajdar, Parth Kapadia	9/25/2022

Machine Learning Model Development:

Action Items	People Responsible	Deadlines
KMeans	Amit Birajdar	10/05/2022
Cosine Similarity	Parth Kapadia	10/12/2022
Hybrid Model	Amit Birajdar, Parth Kapadia	10/19/2022

UI development and model integration:

Action Items	People Responsible	Deadlines
UI development	Sanjana Parakh, Oscar Mui	10/25/2022
Model Integration	Sanjana Parakh, Oscar Mui	11/8/2022

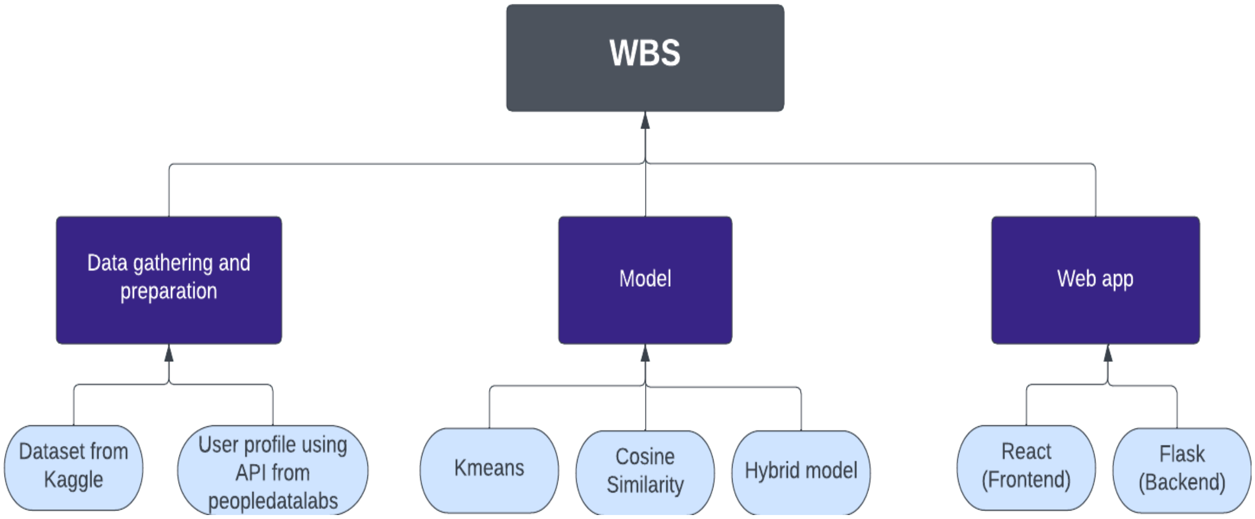
4.3. Pilot Design

Our pilot design included data gathering, data exploration, preprocessing model selection, model building and evaluation and an initial iteration of the user interface. We started with defining a problem statement. We understood the stakeholders requirements and defined a problem statement that aligned with what the stakeholders were looking for. This was a value addition to their already existing solution. We prepared an initial problem scoping document and collected data for the same. This was then presented and approved by the professor. Upon gathering the required data, we proceeded with cleaning and formatting it. For getting the user profile, we decided to make use of a third party API by peopledatalabs. This data was also cleaned and formatted to match the database format. With the data prepared and ready to be used, we implemented three machine learning models viz, KMeans, Cosine Similarity and a third one which was a hybrid of these two. With the results from these models, to evaluate them, we performed A/B testing. Talking to potential users also helped us get more insights into what the user is looking for in the system. We analyzed their feedback and included their suggestions into the system. The hybrid model was the one which gave a comparably better performance and hence, we decided to proceed with it. Finally, our pilot design included an initial iteration of the user interface which provided a look and feel of the system.

4.4. Work Breakdown Structure

The tasks involved in improving the project's deliverable can be divided into three main sections:

1. Data gathering and preparation: This involves getting the LinkedIn user dataset from Kaggle and getting the user profile from LinkedIn using a third party API from peopledatalabs.
2. Model: This involved implementing and testing three models - KMeans, Cosine Similarity and a Hybrid model that uses both KMeans and Cosine Similarity.
3. Web App: This includes creating the front end using React and backend using Flask.



5. Control Phase

5.1. Control Solutions Considered

1. Mistake Proofing (Poka-Yoke): User input may be varied since it is subjected to different factors like the case used, synonyms used etc. Hence, we started with providing a tutorial to the user on how the system can be used and reduced the number of user inputs. In case user input was necessary, we tried to have a selection of choice rather than a text input to reduce the variability. Text input cases were handled in the backend to make all inputs uniform and pass it as variables to the backend.

2. Issue tracking: Implemented model code in Google Colab which has a brief history stored and allows rollback.

3. Project schedule coordination: Kept track of assigned task schedules to make sure deadlines were met with satisfactory work.

5.2. Control Solution Implementation

1. Github version control: Published and maintained code on Github. This provided us with the ability to keep a record of previous versions of code while working on the latest version. In case something goes wrong, it is easier to rollback to the latest stable commit version.

2. Issue Tracking: Using Github, Google Colab and Google Drive helped keep track of changes by enabling the option to view and compare different versions.

3. Mistake Proofing: Provided user with a tutorial on how to use the system. Eliminated unnecessary user input. Whenever input was necessary, we tried not to use text input and rather used selection methods using dropdowns, radio buttons, etc.

6. Result and System Implementation

In this section, we provide a brief description about the system implementation and the results obtained.

6.1. Machine Learning Approaches

Our team considered the following machine learning approaches to recommend relevant users to customers.

1. K-Means Clustering

a. We use a kaggle dataset of over 15,000 scraped, anonymized LinkedIn profiles containing various features regarding profile data, job data, and background information.

i. https://www.kaggle.com/datasets/killbot/linkedin

b. After cleaning the data for null values and irrelevant data, we run K-Means clustering to determine the clusters of users who are similar to each other and are assumed to be valid recommendations. To determine the optimal number of clusters, we use the highest Silhouette Elbow Score during hyperparameter optimization.

c. To determine if a recommendation is accurate, we conduct A/B testing by assigning the ten closest recommendations from the cluster to a real potential user. Users rate the ten recommendations in their preferred order, and the nDCG value is calculated. The nDCG score serves as our primary metric for model validation.

2. Cosine Similarity

a. We use a kaggle dataset of over 15,000 scraped, anonymized LinkedIn profiles containing various features regarding profile data, job data, and background information.

i. https://www.kaggle.com/datasets/killbot/linkedin

b. After cleaning the data for null values and irrelevant data, we calculate the cosine similarity score between a given user and all other users in the dataset. Scores are then ranked in descending order, with a higher score being a better match. The idea here is that a strong recommendation will be similar to a given user in terms of the selected features.

c. To determine if a recommendation is accurate, we conduct A/B testing by assigning the ten highest cosine similarity scores to a real potential user. Users rate the ten recommendations in their preferred order, and the nDCG value is calculated. The nDCG score serves as our primary metric for model validation.

3. **Hybrid Approach - Combining K-Means Clustering and Cosine Similarity**

a. We use a kaggle dataset of over 15,000 scraped, anonymized LinkedIn profiles containing various features regarding profile data, job data, and background information.

i. https://www.kaggle.com/datasets/killbot/linkedin

b. After cleaning the data for null values and irrelevant data, we run K-Means clustering to determine the clusters of users who are similar to each other. To determine the optimal number of clusters, we use the highest Silhoutte Elbow Score during hyperparameter optimization.

c. When a real potential user inputs their information, we calculate the cosine similarity between the given user and all of the other users in the cluster, adding weights to the calculation based on the highest weighted features in the cluster.

d. To determine if a recommendation is accurate, we conduct A/B testing by assigning the ten highest cosine similarity scores to a real potential user. Users rate the ten recommendations in their preferred order, and the nDCG value is calculated. The nDCG score serves as our primary metric for model validation.

6.2. System Implementation

Our product is designed to help users easily find fellow collaborators, removing the hassle and struggle of finding fellow team members that match the required skills and background through a robustly designed system.

The process begins when a user enters the full name and email address associated with their LinkedIn profile. With an active authentication key in the background, these parameters are used to find the associated LinkedIn profile using the API provided by LinkedIn Developer Tools. Additionally, users have the option to select a series of filters such as minimum experience, skills, and age, that will narrow results as desired. These filter inputs are recognized and used later in the final recommendation results.

Next, our backend system retrieves the profile data as a JSON object and transforms the data via feature engineering into the appropriate set of features that match the initial model inputs. These are fed into the model, which utilizes a hybrid approach between K-Means Clustering and Cosine Similarity as described in section 6.1, to compute similarity scores between the user and our database of LinkedIn profiles.

The set of similarity scores are then filtered such that computed scores whose features do not fit the criteria specified by the user are removed. The top ten filtered scores are then retrieved along with the associated LinkedIn profile details and returned to the frontend via an API.

Finally, the frontend retrieves the results, renders the components via React, and returns a stylized list of recommendations to the end user.

6.3. Prototype & Demonstration

The user functionalities on every page is detailed below using a prototyped sequence of user interaction steps:

1. User opens the website and enters the *Home* page as shown in appendix 6. The *Home* page provides a brief overview of the product as well as a link to get started.

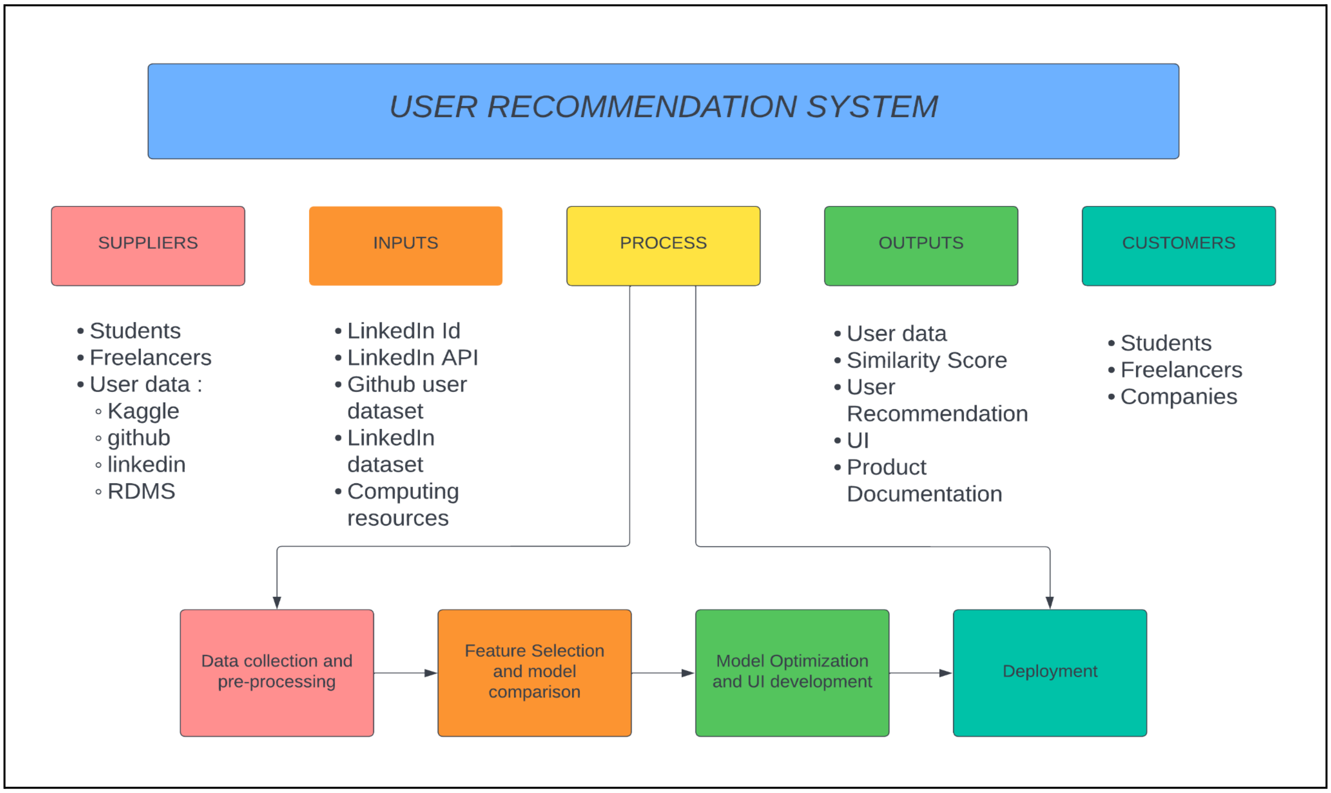
2. User clicks on the link to get started or clicks the **Recommendation** hyperlink at the top of the page to enter the *Recommendation* page as shown in appendix 7. In the *Recommendation* page, users are able to input their LinkedIn profile and filter parameters in order to retrieve a set of ten recommendations.

3. User clicks on the **Projects** hyperlink at the top of the page to enter the *Projects* page as shown in appendix 8. In the *Projects* page, users are given a list of various project collaboration resources along with hyperlinks to their website and brief summaries.

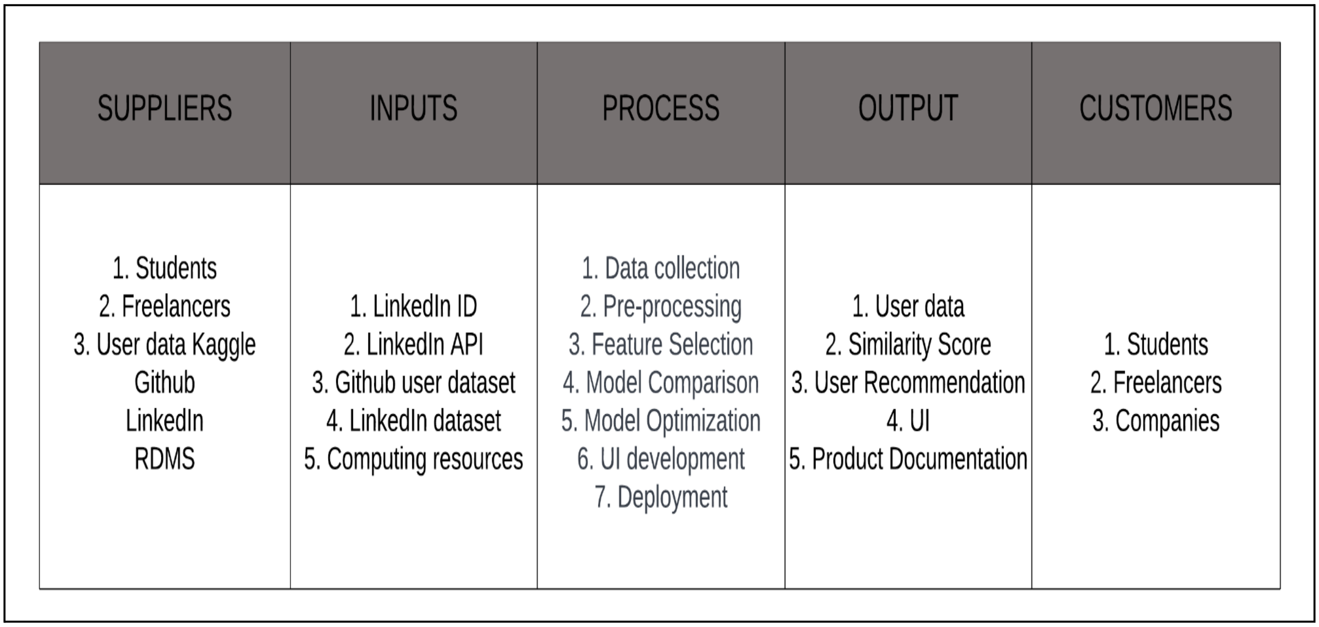
4. User clicks on the **About** hyperlink at the top of the page to enter the *About* page as shown in appendix 9. The *About* page introduces the USC-based team behind our product along with brief, individual bios for each of the contributing members.

Appendix

Appendix 1: SIPOC

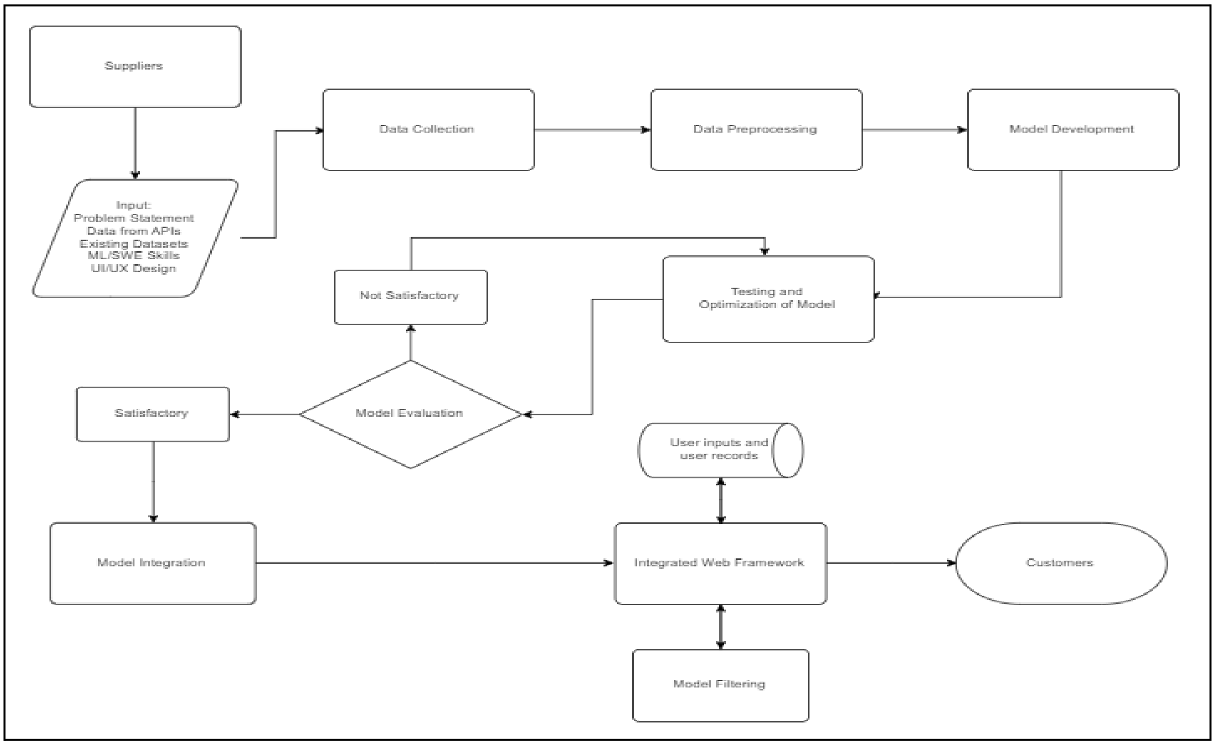


Appendix 2: High Level Process Map

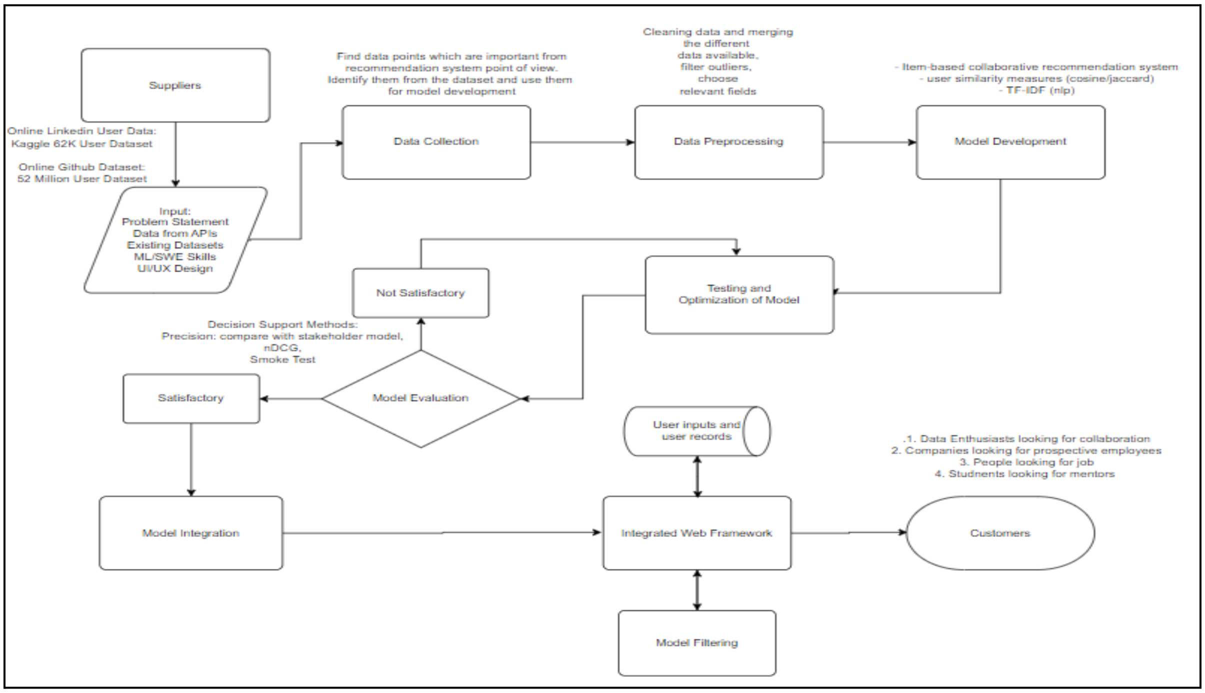


SUPPLIERS	INPUTS	PROCESS	OUTPUT	CUSTOMERS
1. Students 2. Freelancers 3. User data Kaggle Github LinkedIn RDMS	1. LinkedIn ID 2. LinkedIn API 3. Github user dataset 4. LinkedIn dataset 5. Computing resources	1. Data collection 2. Pre-processing 3. Feature Selection 4. Model Comparison 5. Model Optimization 6. UI development 7. Deployment	1. User data 2. Similarity Score 3. User Recommendation 4. UI 5. Product Documentation	1. Students 2. Freelancers 3. Companies

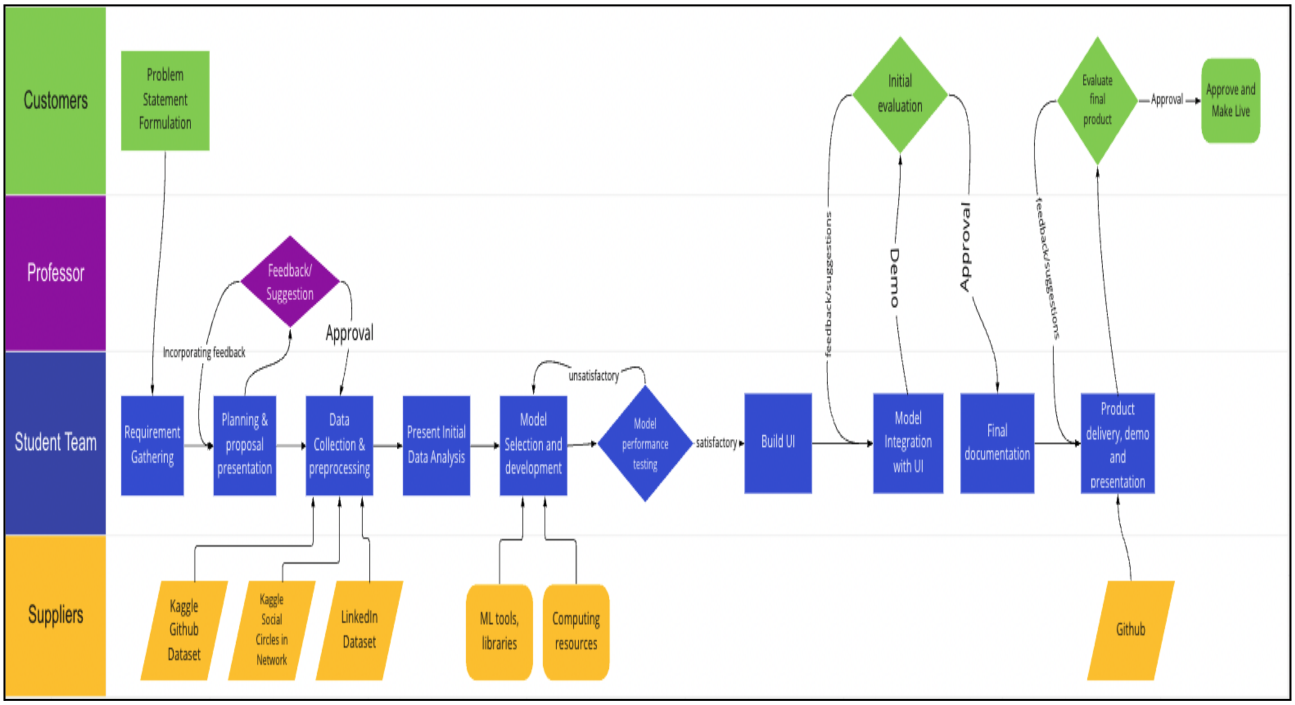
Appendix 3: Common Process Map



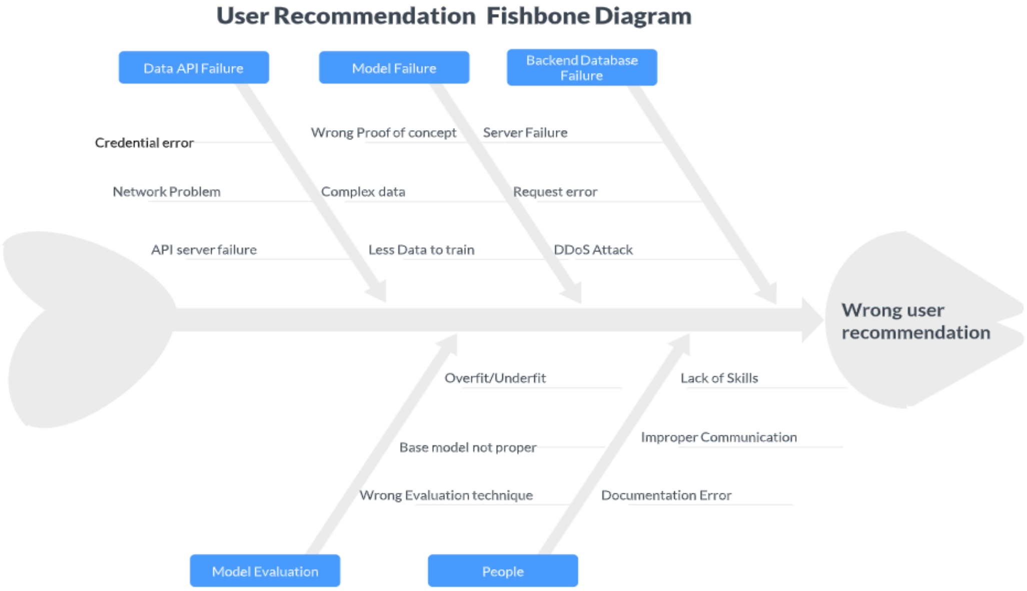
Appendix 4: Detailed Process Map



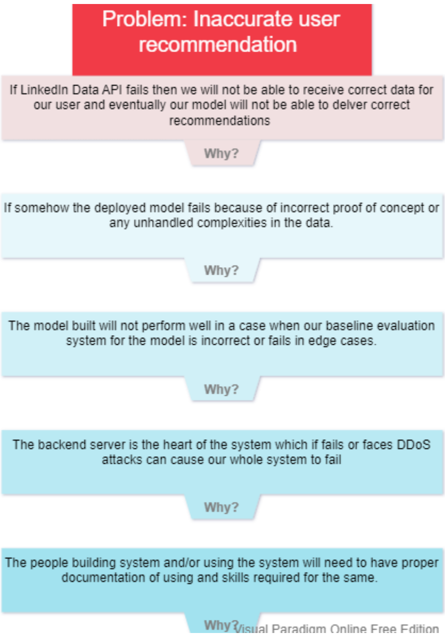
Appendix 5: Functional Process Map



Appendix 6: Fishbone Diagram



Appendix 7: 5 WHYs



Problem: Inaccurate user recommendation

If LinkedIn Data API fails then we will not be able to receive correct data for our user and eventually our model will not be able to delver correct recommendations

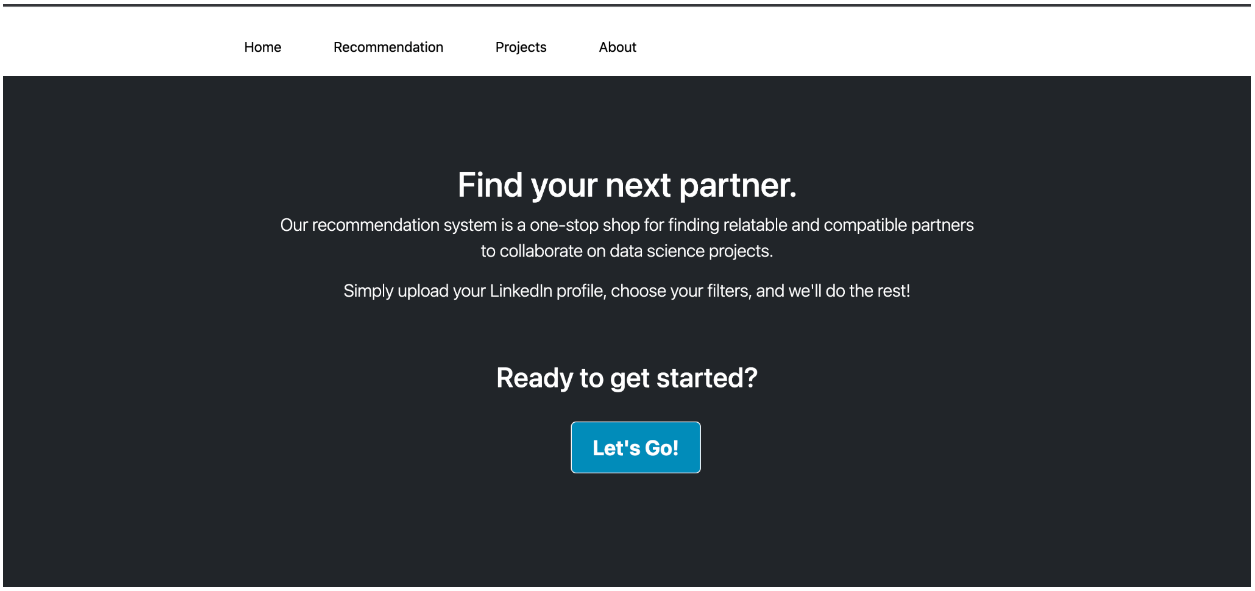
If somehow the deployed model fails because of incorrect proof of concept or any unhandled complexities in the data.

The model built will not perform well in a case when our baseline evaluation system for the model is incorrect or fails in edge cases.

The backend server is the heart of the system which if fails or faces DDoS attacks can cause our whole system to fail

The people building system and/or using the system will need to have proper documentation of using and skills required for the same.

Appendix 8: Home Page



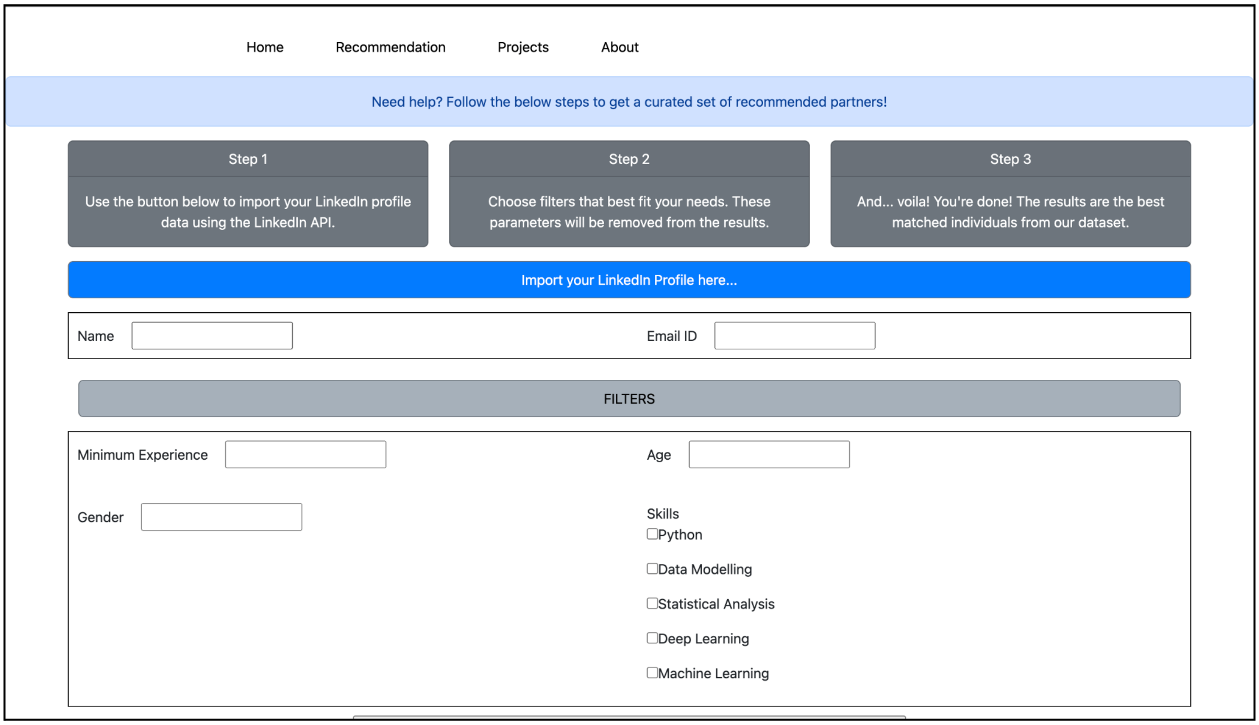
Find your next partner.

Our recommendation system is a one-stop shop for finding relatable and compatible partners to collaborate on data science projects.

Simply upload your LinkedIn profile, choose your filters, and we'll do the rest!

Ready to get started?

Appendix 9: Recommendations Page



Need help? Follow the below steps to get a curated set of recommended partners!

Step 1

Use the button below to import your LinkedIn profile data using the LinkedIn API.

Step 2

Choose filters that best fit your needs. These parameters will be removed from the results.

Step 3

And... voila! You're done! The results are the best matched individuals from our dataset.

Name

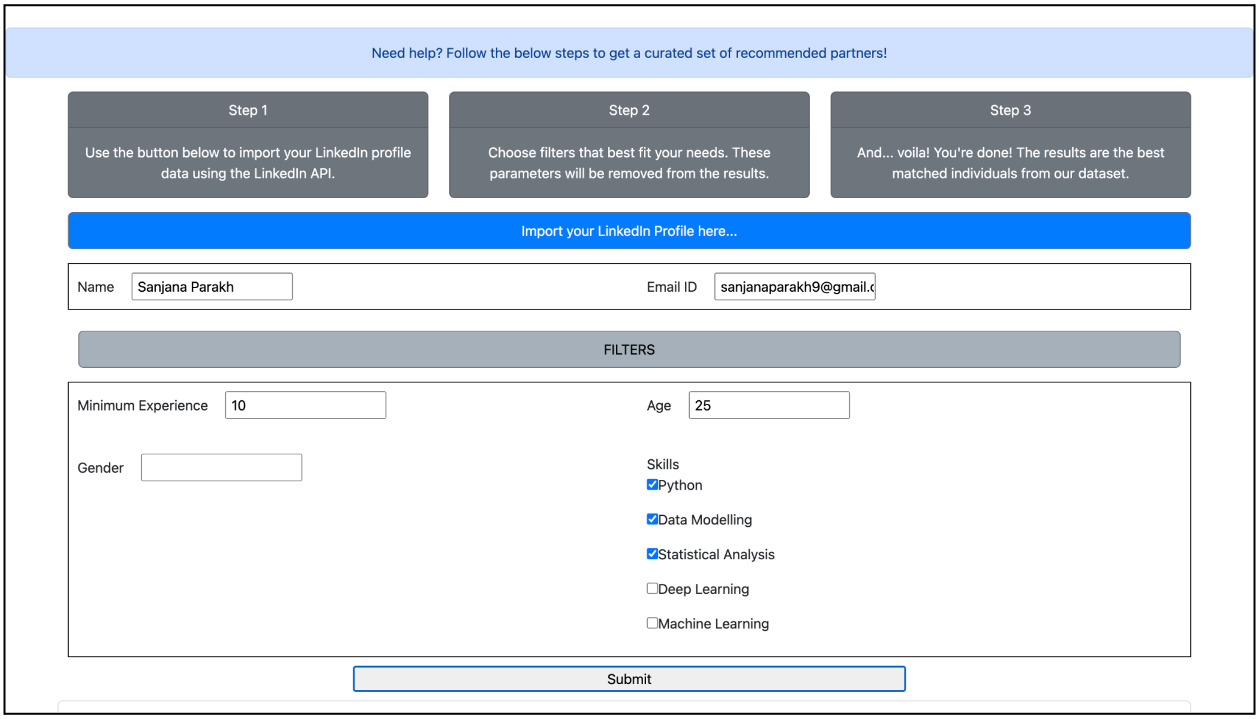
Email ID

Minimum Experience

Age

Gender

Skills
□Python
□Data Modelling
□Statistical Analysis
□Deep Learning
□Machine Learning



Need help? Follow the below steps to get a curated set of recommended partners!

Step 1

Use the button below to import your LinkedIn profile data using the LinkedIn API.

Step 2

Choose filters that best fit your needs. These parameters will be removed from the results.

Step 3

And... voila! You're done! The results are the best matched individuals from our dataset.

Name Sanjana Parakh

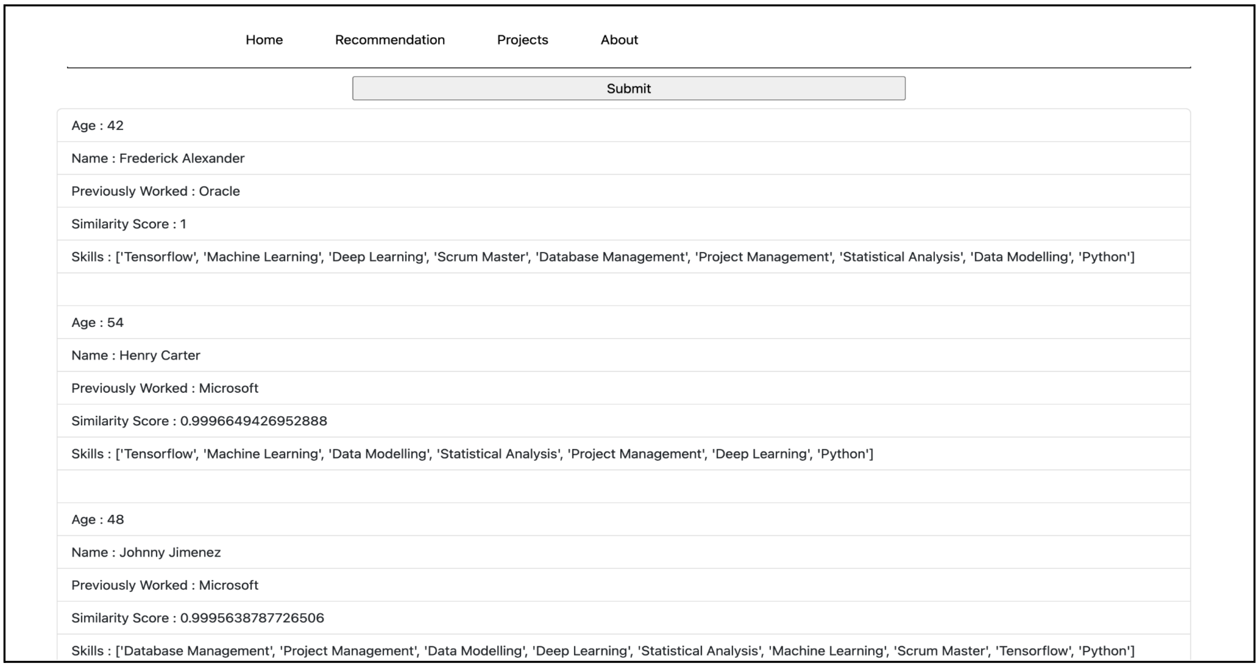
Email ID sanjanaparakh9@gmail.c

Minimum Experience 10

Age 25

Gender

Skills
☑Python
☑Data Modelling
☑Statistical Analysis
□Deep Learning
□Machine Learning



Age : 42

Name : Frederick Alexander

Previously Worked : Oracle

Similarity Score : 1

Skills : ['Tensorflow', 'Machine Learning', 'Deep Learning', 'Scrum Master', 'Database Management', 'Project Management', 'Statistical Analysis', 'Data Modelling', 'Python']

Age : 54

Name : Henry Carter

Previously Worked : Microsoft

Similarity Score : 0.9996649426952888

Skills : ['Tensorflow', 'Machine Learning', 'Data Modelling', 'Statistical Analysis', 'Project Management', 'Deep Learning', 'Python']

Age : 48

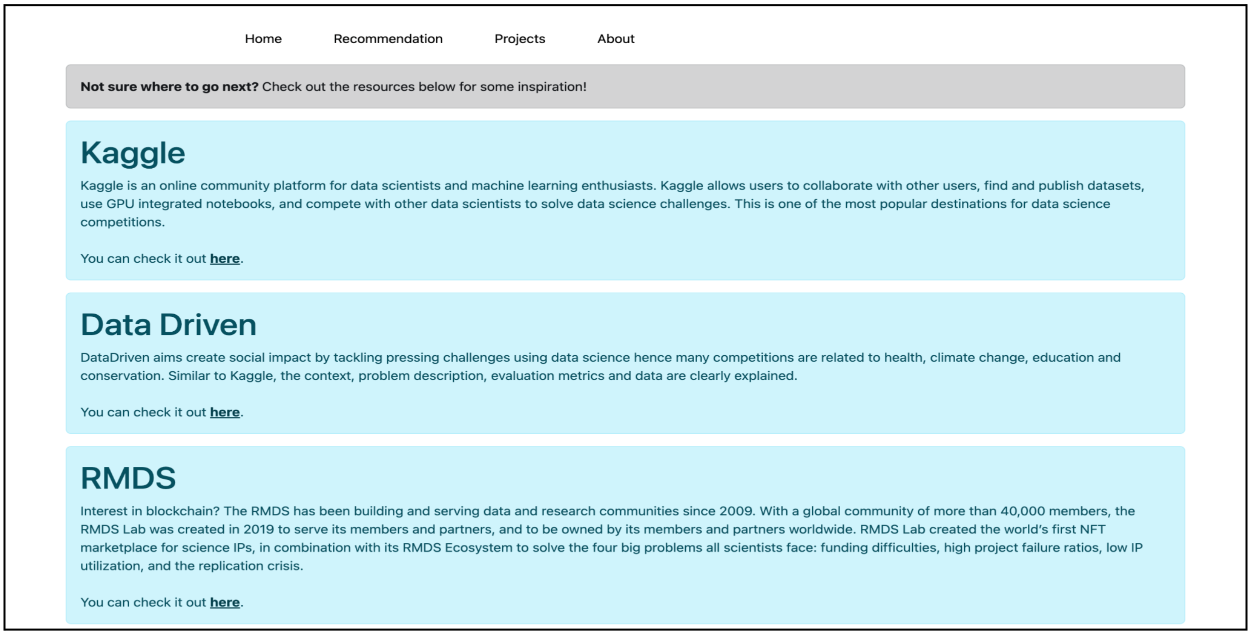
Name : Johnny Jimenez

Previously Worked : Microsoft

Similarity Score : 0.9995638787726506

Skills : ['Database Management', 'Project Management', 'Data Modelling', 'Deep Learning', 'Statistical Analysis', 'Machine Learning', 'Scrum Master', 'Tensorflow', 'Python']

Appendix 10: Project Ideas Page



Not sure where to go next? Check out the resources below for some inspiration!

Kaggle

Kaggle is an online community platform for data scientists and machine learning enthusiasts. Kaggle allows users to collaborate with other users, find and publish datasets, use GPU integrated notebooks, and compete with other data scientists to solve data science challenges. This is one of the most popular destinations for data science competitions.

You can check it out **here**.

Data Driven

DataDriven aims create social impact by tackling pressing challenges using data science hence many competitions are related to health, climate change, education and conservation. Similar to Kaggle, the context, problem description, evaluation metrics and data are clearly explained.

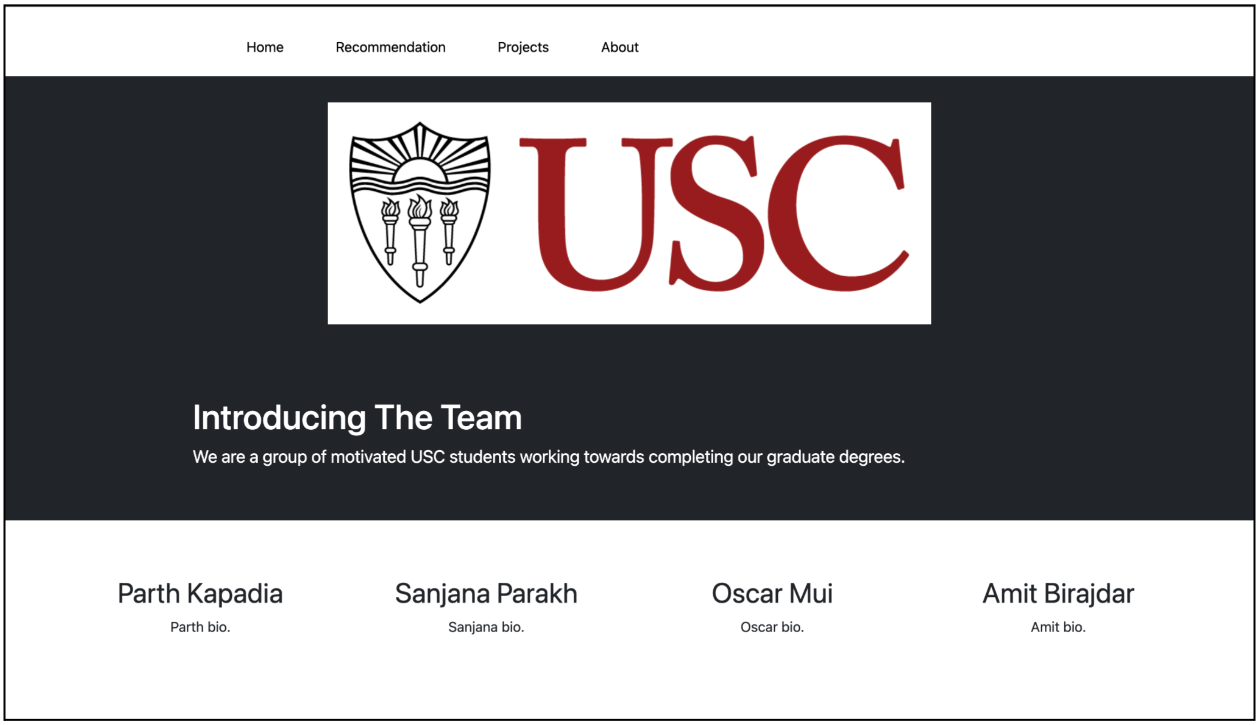
You can check it out **here**.

RMDS

Interest in blockchain? The RMDS has been building and serving data and research communities since 2009. With a global community of more than 40,000 members, the RMDS Lab was created in 2019 to serve its members and partners, and to be owned by its members and partners worldwide. RMDS Lab created the world's first NFT marketplace for science IPs, in combination with its RMDS Ecosystem to solve the four big problems all scientists face: funding difficulties, high project failure ratios, low IP utilization, and the replication crisis.

You can check it out **here**.

Appendix 11: About Page



Introducing The Team

We are a group of motivated USC students working towards completing our graduate degrees.

Parth Kapadia

Parth bio.

Sanjana Parakh

Sanjana bio.

Oscar Mui

Oscar bio.

Amit Birajdar

Amit bio.