

Qupid

The teams matchmaking
service



Content Table

1

**About
the Project**

About Project

2

**Project
Overview**

How to Use? What is
to Use?

3

**Under the
Hood:
Algorithm**

Under the Hood: On
Paper

4

**Under the
Hood: Tech**

Under the Hood:
How was it made
Practically?

5

**Project
Stages**

How was the Project
Developed ?

6

**Future and
Challenges**

Challenges Faced
During Creation and
Features for Future

About Qupid

Is this Section a Pitch Deck?



1

About the Project

Qupid is a matchmaking service that is responsible for creating efficient teams for workplaces and classes.



Why?

It is always difficult to come up with efficient teams when we are working on projects. These are some problems that we face :

- Teams are not diverse
- Teams are not balanced in skill level
- Teams have higher distribution members belonging to one major skill.

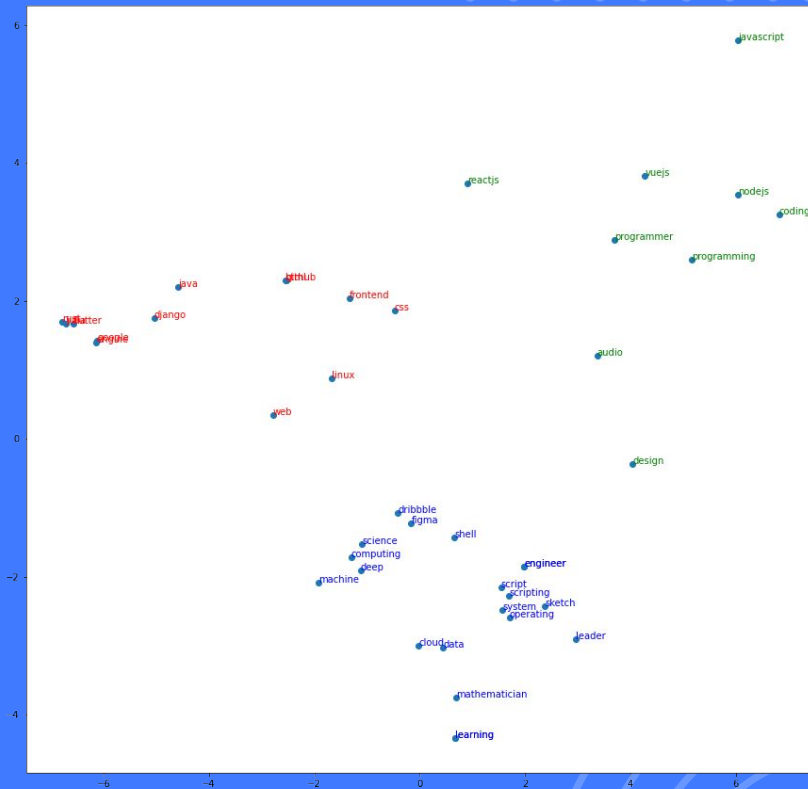
With Qupid our goal is to deal with all these problems and make teams that are diverse and balanced in skill level.



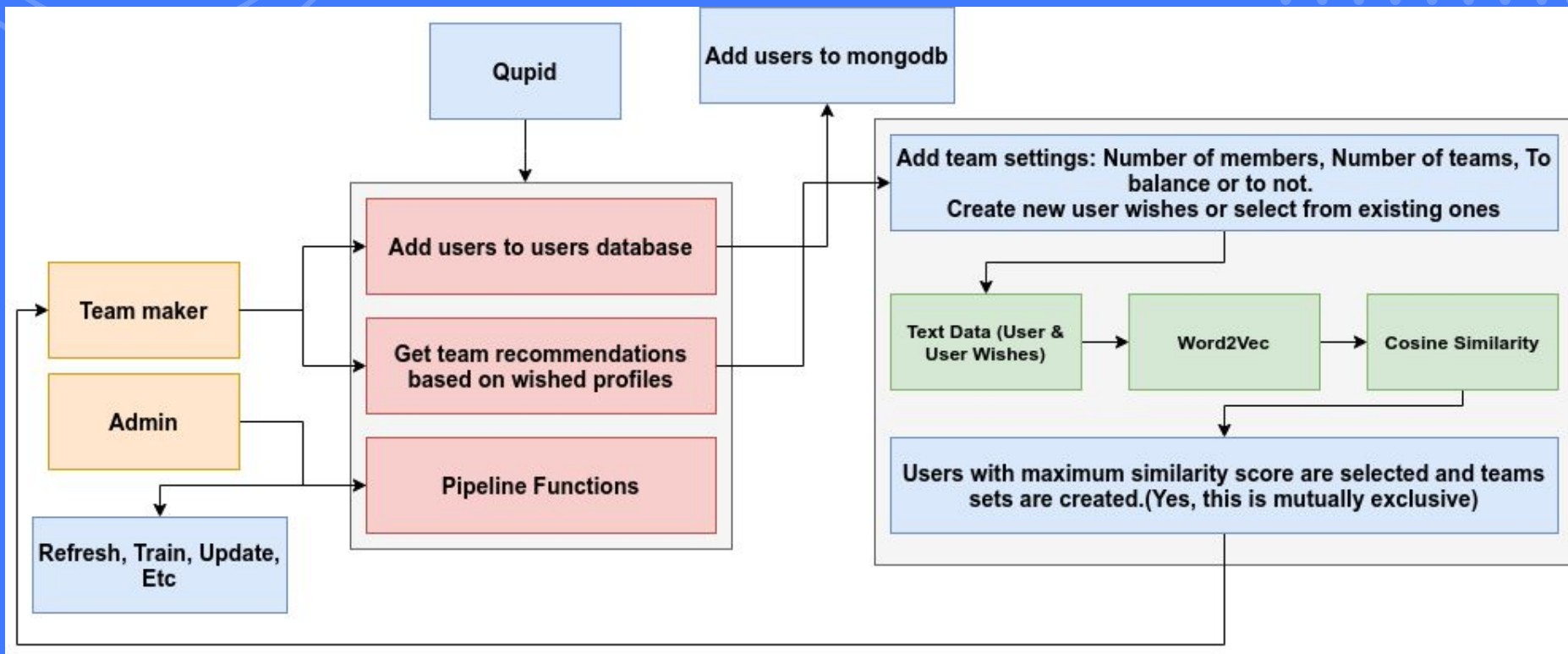
How?

Qupid uses a custom algorithm designed by me. At its core this is based on *word embeddings* created from *scraping tweets* related to data appearing in user profiles and then using *cosine similarity* to make teams that are just like the *team that the team maker has wished*. This is explained in detail in section Under the Hood.

3 cluster Kmeans plot of 2 major dimensions (PCA) of trained Word2Vec Model. Notice how similar words are grouped together



Flowcharts are great!



Project Overview

Let's view the project from
front!



2

4 Main Sections of Qupid



Home



Add Users



Make Teams



Settings

Welcome to Qupid!

The teams matchmaker!

Qupid is a matchmaking service that is responsible for creating efficient teams for workplaces and classes. It is based on a custom made content based recommendation algorithm that uses a self trained word2vec model at its core. This model is trained from scrapping tweet data that has terms related to team users profiles.

Data Pipeline Model

```
graph TD
    subgraph Input
        A1[All real profiles]
        A2[Fake profiles at time of recommendation asking]
        A3[Words for which don't exist in trained model]
        A4[MongoDB with user data in human readable format]
    end
    subgraph PipelineAdmin[Pipeline admin functions]
        A5[Manual Trigger]
        A6[Every time recommendations are asked]
        A7[Manual Trigger to retrain Word2vec model on new tweet data]
    end
    subgraph Core
        A8[Admin triggered dataset update hit(convert mongo data to vector data)]
        A9[Find similarity score for and add in column (memoize solution)]
        A10[Return Most similar profiles]
        A11[based on Word2Vec Model]
        A12[Collect unseen words]
        A13[Similar Words dataset]
        A14[Similarity finding algorithm. Cosine Similarity/Nearest Neighbour]
    end
    subgraph Output
        A15[recommended team]
    end

    A1 --> A8
    A2 --> A9
    A3 --> A12
    A4 --> A12
    A5 --> A8
    A6 --> A9
    A7 --> A8
    A8 --> A9
    A9 --> A10
    A10 --> A11
    A11 --> A12
    A12 --> A13
    A13 --> A14
    A14 --> A15
```

The diagram illustrates the Data Pipeline Model for Qupid. It shows the flow of data and actions from input to output, including database components, automatic triggers, and pipeline actions/algorithms.

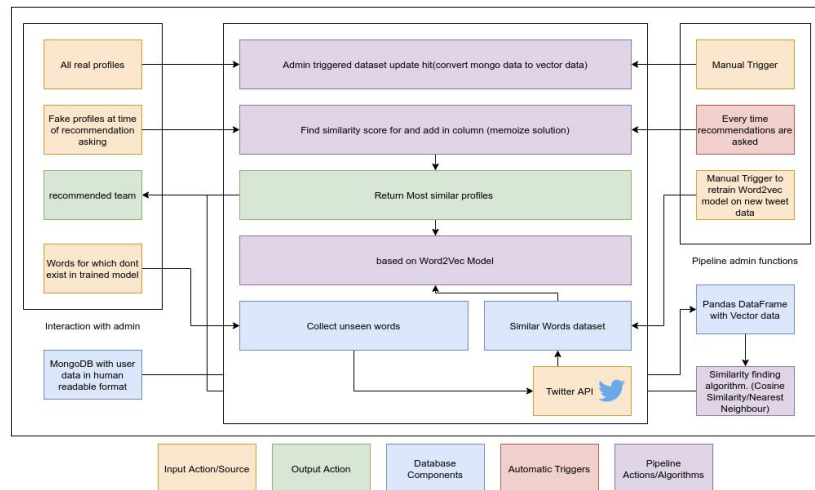
Legend:

- Input Action/Source (Orange)
- Output Action (Green)
- Database Components (Blue)
- Automatic Triggers (Red)
- Pipeline Actions/Algorithms (Purple)

The teams matchmaker!

Qupid is a matchmaking service that is responsible for creating efficient teams for workplaces and classes. It is based on a custom made content based recommendation algorithm that uses a self trained word2vec model at its core. This model is trained from scrapping tweet data that has terms related to team users profiles.

Data Pipeline Model



Add Users

192.168.0.114:3000 says
Created New User

OK

Example

Fullname: Kaushal Patil

email: kaushalpatil10@gmail.com

password

role: Programmer

Specialisation 1: python

Specialisation 2: dl

Specialisation 3: javascript

Organisation: Ahmedabad University

Fullname
Kaush P

Email
kaushalpatil20@gmail.com

Password

Role
dl

Specialisation 1
python

Specialisation 2
tensorflow

Specialisation 3
data

Organisation
au

CREATE USER

Copyright © Qupid 2021.

How it works ?

1. You add details such as fullname, email, password, role, speciality 1, speciality 2, speciality 3 and organisation and click on create button.
2. A new user is created in database.

What to do next?

- To make your user appear in pipeline dataframe you got to settings and click on update users in dataset button. (This is kept this way so multiple users can be added once and the pipeline data can be updated at once(to save time on atomic requests)).



How Team Creation Works.

```

graph TD
    User[User] --> Prompt[Is Prompted to make a dream team with self defined fake user profiles]
    Prompt --> DB[Database: Contains member traits of user that will be used to assign them to a team created by the recommender]
    Prompt --> Features[Features: M  
Specialty: N  
Overall Strength: O  
Member Traits]
    Prompt --> Options[Customization features for teams  
(Options):  
- Balance Team (By strength)  
- Number of Teams to create  
- TBD]
    DB --> DreamTeam[Dream Team To be used to generate recommendation  
(Contains user created ideal member profiles)]
    Features --> DreamTeam
    Options --> DreamTeam
    DreamTeam --> RS[Recommender Service: One by one takes member profiles from Dream Team and find similar options from Database]
    RS --> RealTeam[Real Team: Containing profile of real users]
    RS --> DB
    
```

Team Settings

Number of members in a team

Number of Teams

☒ Balance Teams

BEGIN PROCESS

Number of members in a team

2

Number of Teams

2

- ✓ Balance Teams

RS takes data from DB and options from user

Real Team: Containing profile of real users

How it works ?


1. You add teams details such as number of members in a team and number of teams that you want to create in one call.
2. You select if you want to balance teams or not.

What to do next?

- Wish for certain kind of users and create one wish if the already existing don't match your requirement.



Make Teams: Page 2

 Qupid: The teams matchmaker!

Add A Custom Wish User

Role of wished user

Speciality 1 of wished user

Speciality 2 of wished user

Speciality 3 of wished user

Org of wished user

CREATE WISHED USER


Base Your Team On


Id of wish user 1


603c2d545736b91c75098173


Id of wish user 2


603aa29d635cc066c2f64845

 603c1df36870bd179f6c203c
Role: fullstack
Specialities: python,javascript,dl

 603c1cef3197f6ea28759d14
Role: developer
Specialities: flutter,android,audio

 603c1bcea74551f5858b330c
Role: github
Specialities: vcs,scripting

 603aa29d635cc066c2f64845
Role: frontend
Specialities: frontend,css,html,deeplearning

 603aa29a635cc066c2f64844
Role: frontend
Specialities: frontend,css,html,deeplearning

MAKE MY TEAMS!

Copyright © Qupid 2021.

How it works ?


1. You copy ID of a user wish from right side into the Base your team section ID.
2. If a suitable user wish doesn't exist already you create one using th create from.

What to do next?

- That's it! Your teams will be cooked and readily server in 3..2..1.. Poof!



Make Teams: Here are your built Teams!

 Qupid: The teams matchmaker!

Your created teams

Team 1

Member 1

Fullname: Kaushal Patil

Email: kaushalpatil10@gmail.com

Role: fullstack

Specialities: python,javascript,backend,frontend,deeplearning

Org: Ahmedabad University

Member 2

Fullname: Kaushal Patil

Email: kaushalpatil1@gmail.com

Role: github

Specialities: python,javascript,backend,frontend,deeplearning

Org: Ahmedabad University

Team 2

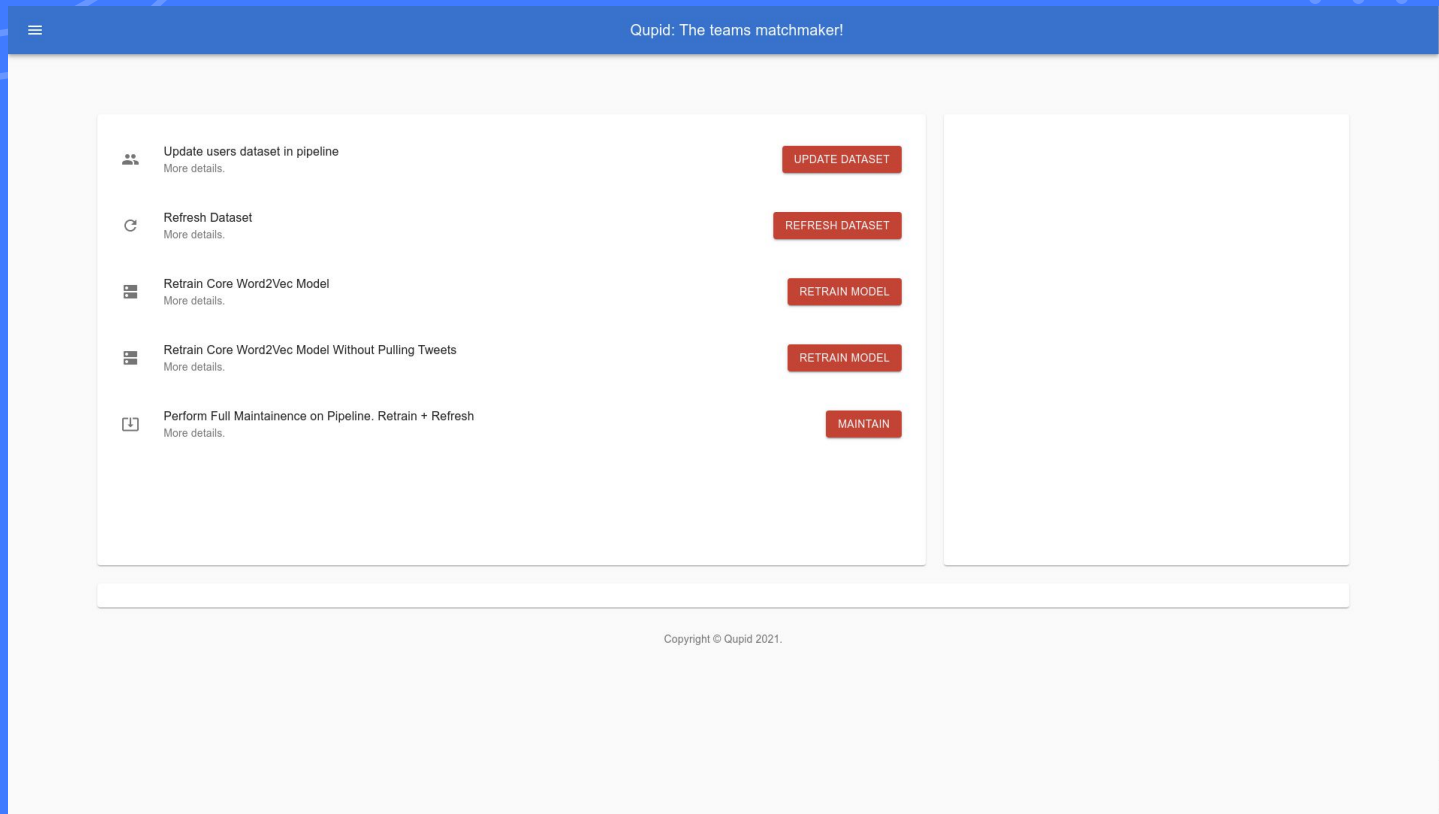
Member 1

Fullname: Arpit Vaghela

Email: arpitvaghela@gmail.com

Copyright © Qupid 2021.

Settings Page



How it works ?

1. This page has some pipeline options. They are to be used carefully since they are data heavy operations. Majorly these are related to:
 - a. Data Source Cleanup
 - b. Retraining Model
 - c. Maintaining Pipeline

What to do next?

- Get a coffee and wait. Some of these operation take a lot of time. Specially the retrain with tweets scrap one. (Fun fact approximately 20,000 tweets are scrapped for training)



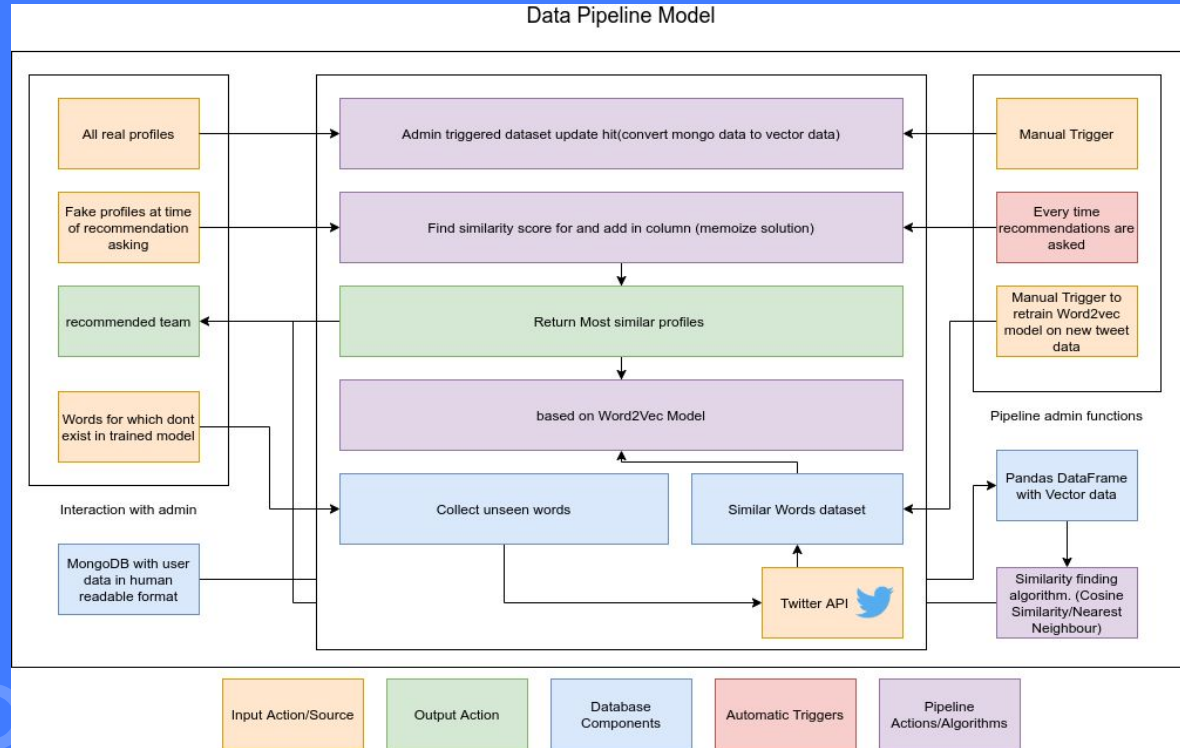
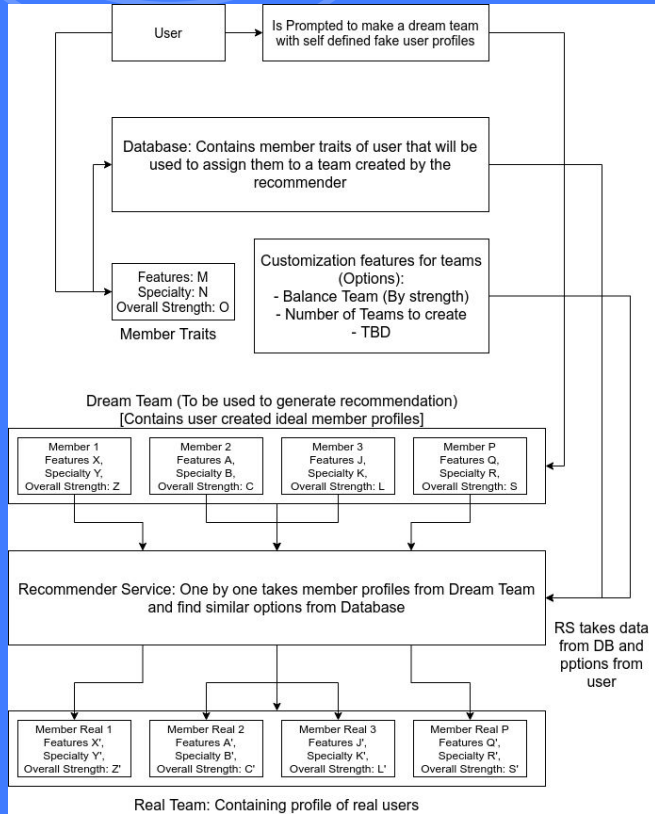
Under the Hood: Algorithm

All hail draw.io

3



Prototype Algorithm & Under the Hood Algorithm



Under the Hood: Tech

Show me the cool code! Major cool code written related to AOB: Train Model, Generate Word to Vec Insights, Cosine Similarity, Teams Creation



4

Train Model and Generate Insights

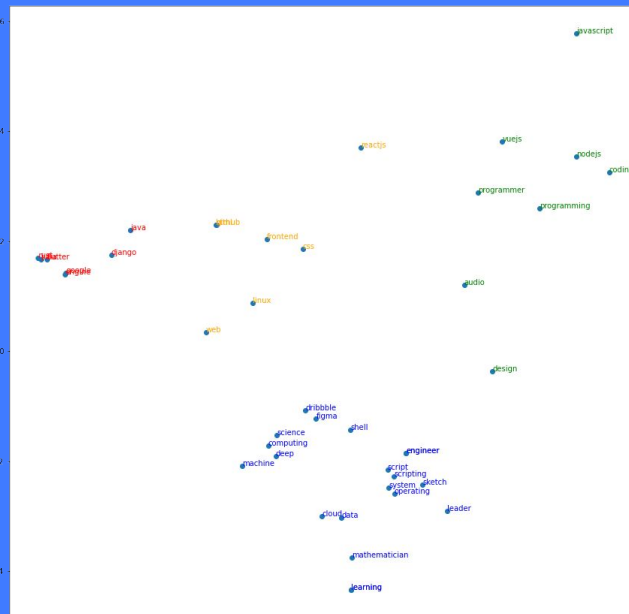
Code Link:

<https://github.com/Kaushal1011/Qupid/blob/main/app/server/routes/model/train.py>

What does it do :

1. Scraps Tweets (based on dynamic oob terms)
2. Clean Tweets (Removes stop words and special chars)
3. Creates a dataframe with all text tweet data.
4. Tokenizes Tweet Data.
5. Trains a gensim Word2Vec model on Tokenized data.
6. Generates Model Insight
 - a. Performs 3 Groups K Means Cluster.
 - b. Performs 2 PC PCA on selected word vectors
 - c. Scatter plots selected word vectors

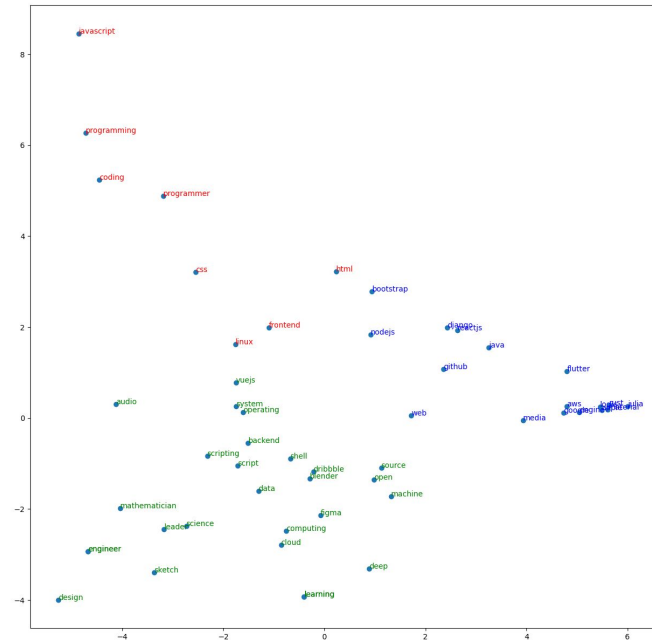
N Clusters = 6



Current Word2Vec Model Insight

This is the model that's currently being used by the application. Some insights:

1. Notice how similar words are closer
 - a. ReactJS, NodeJS, Django
 - b. Frontend, Html, CSS
2. Notice how some words are at same distance from some words:
 - a. Javascript → Node JS ≈ Javascript → ReactJS
 - b. deep → learning ≈ machine → learning



Recommend Code & Cosine Similarity

Code Link:

<https://github.com/Kaushal1011/Qupid/blob/main/app/server/routes/pipeline.py>

What does it do :

1. Takes user wishes' vector data and finds cosine similarity with user data.
 - a. SimScore
$$\text{formula} = (\text{cossim}(\text{role}) * 1.3 + \text{cossim}(\text{sp1}) + \text{cossim}(\text{sp2}) + \text{cossim}(\text{sp3}) * 0.7) / 4$$

(More weight given to role over speciality 3).
2. Finds mutually exclusive sets of users for all teams by maintain a hashtable.
3. Balance teams. (Balancing is basically finding X best users for X teams and then assigning each user to a team randomly, this makes sure that team 1 doesn't always gets best matches)

Project Stages

A whole lot of issues were created and closed on github



5

☐ ⓘ 2 Open ✓ 6 Closed

Author ▾


Label ▾

Projects ▾


Milestones ▾


Assignee ▾


Sort ▾


☐  **return mutually exclusive set of users when recommending teams**


#7 by Kaushal1011 was closed 6 hours ago


☐  **qupid front end**


#6 by Kaushal1011 was closed 13 hours ago  4 of 5


☐  **Add database schema, model schema & pipeline design**


#5 by Kaushal1011 was closed 2 days ago  5 of 5


☐  **Data,Pipeline & Model related stuff**

#3 by Kaushal1011 was closed 16 hours ago  5 of 5

☐  **Backend related stuff**

#2 by Kaushal1011 was closed 16 hours ago  6 of 6

☐  **Get data for word2vec from Linkedin/Twitter scraping or create on your own**

#1 by Kaushal1011 was closed 3 days ago  7 of 7

Future and Challenges

Will cupid really become the angel that makes teams?



6

Challenges

1. Integrating Recommendation algorithm in a pipeline sense.
2. Creating Mutually Exclusive Set of teams required some thinking and use of hashtables
3. Usage of csv to manage vector data. This was later changed to pickle file.
4. Writing code the pandas/python way.



Future

1. A lot of optimisation tasks can be done in the pipeline to make it faster and scale up this project
2. Use of data scraped from something such as linkedin can work better over tweets as linkedin has professional user data and profile.
3. Collaborative filtering can be used to recommend user wishes over showing all user wishes at one place.



Thanks!

Do you have any questions?

Follow the project updates

<https://github.com/Kaushal1011/Qupid>



<https://www.linkedin.com/in/kaushal1011/>

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution