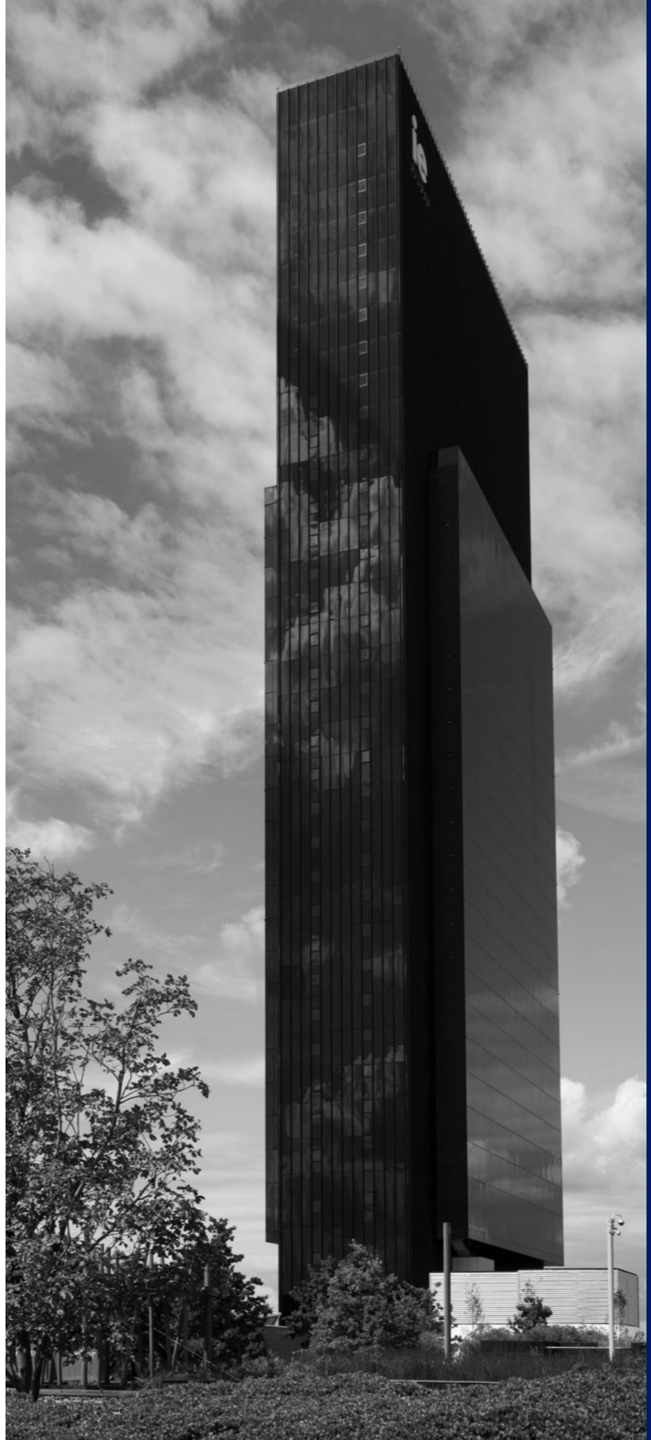


Code Compass

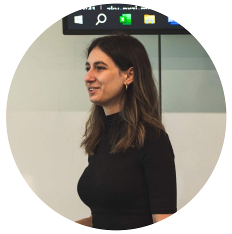
Midterm Review

Agile Team

Product Manager + Data Engineer
Gabriel de Olaguibel

Data Scientist + ML Ops
Miranda Drummond

Data Engineer + Data Scientist + ML Engineer
Luca Cuneo



Data Scientist + MLOps
Keti Sulamanidze

Data Engineer + Data Scientist
Maud Hovland

Data Scientist + ML Engineer
Félix Gómez-Guillamón

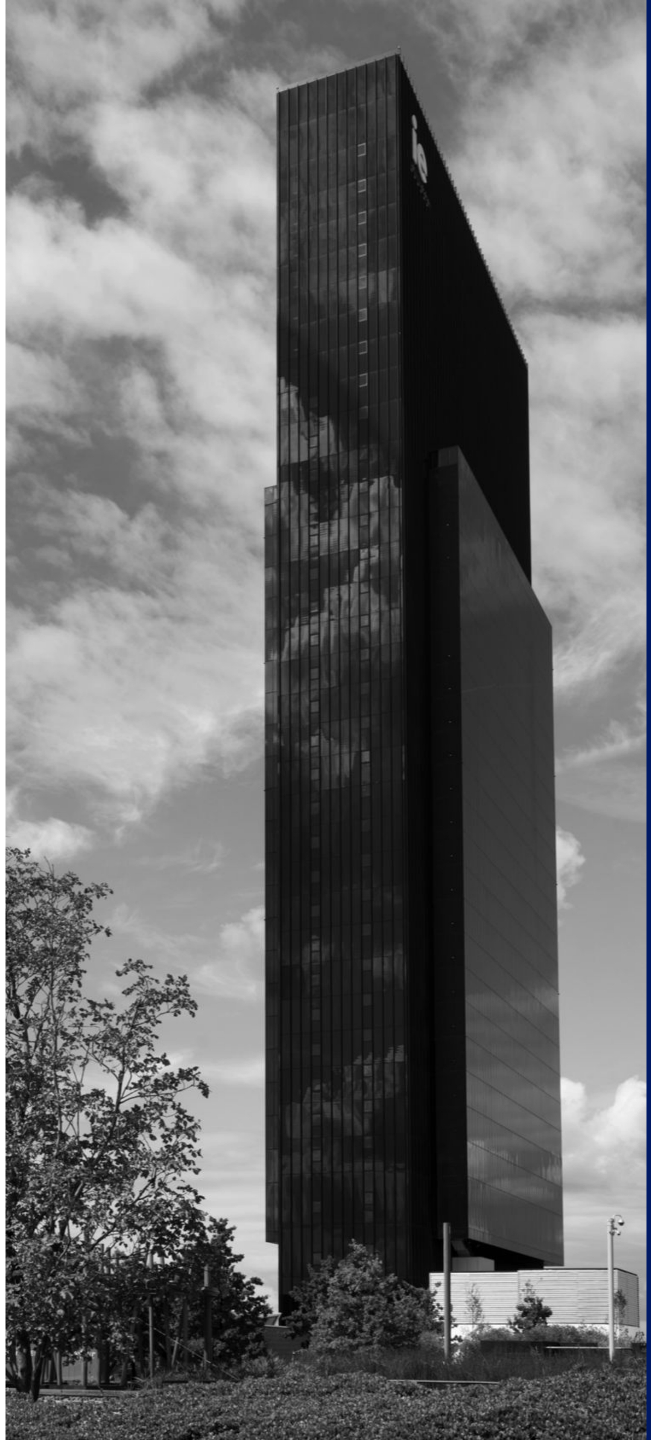
Code Compass

Navigating Developers to Personalized GitHub exploration and Insights

Agenda

- Product Goals
- Demonstrate Progress
- Development Methodology
- Future Roadmap
- Collect Feedback

UNIVERSITY



Product Goals

UNIVERSITY

Vision

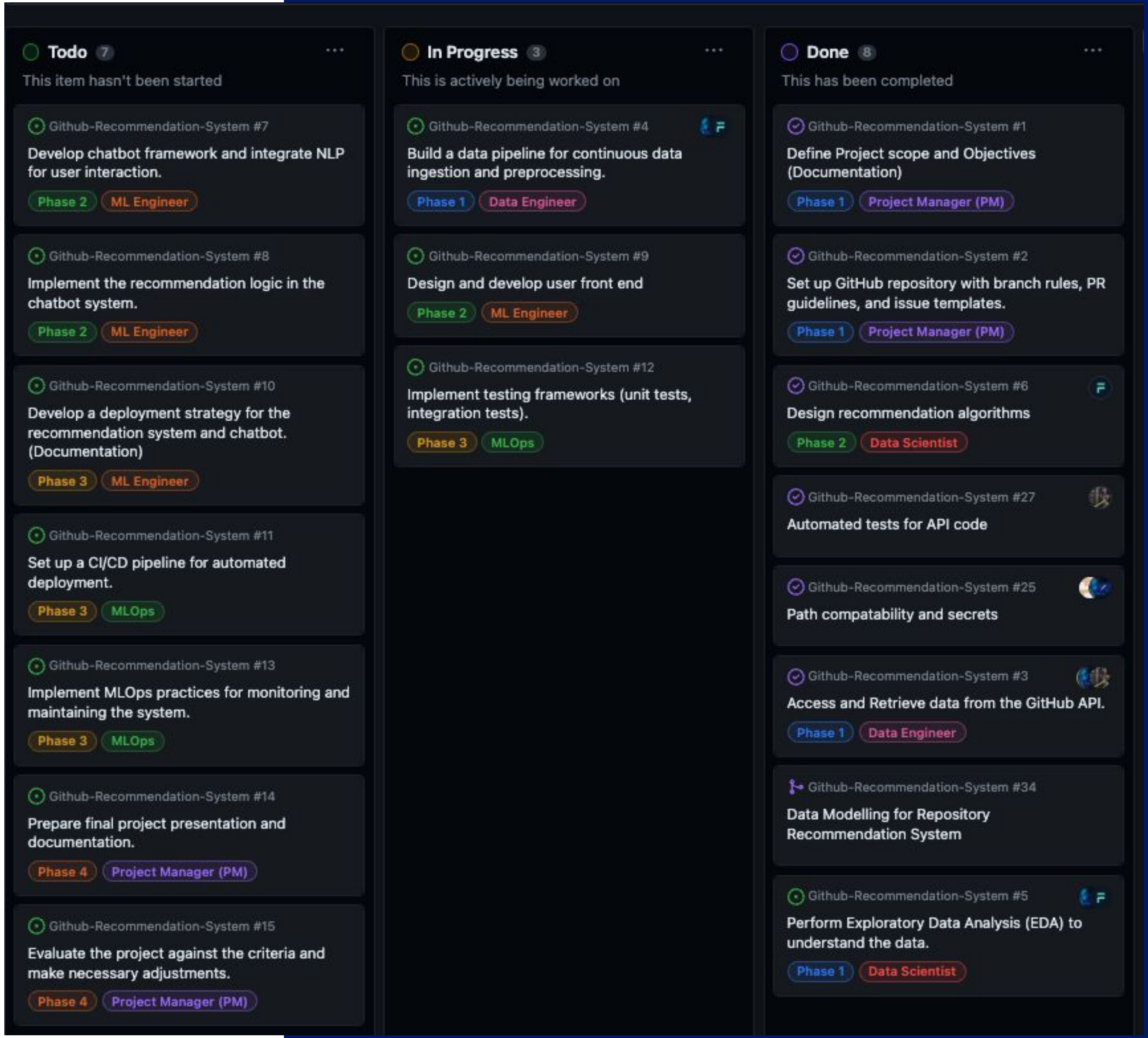
Enhance the GitHub experience for developers and learners by providing personalized guidance and resources. This system aims to simplify the process of finding relevant projects and learning materials, making the path to knowledge and collaboration more accessible and efficient.

Solution

Addressing these challenges, Code Compass proposes a novel solution: a combination of a recommendation system and a chatbot interface. This dual approach aims to:

- ***Curate Personalized Recommendations**: Utilizing user-specific data such as project involvement, coding languages, and collaboration networks, Code Compass will recommend relevant GitHub projects, technologies, and connections.*
- ***Interactive Learning and Exploration**: The integrated chatbot will serve as an interactive guide, helping users to delve deeper into recommendations, understand technology applications, and navigate GitHub more effectively.*

Backlog



Todo 7
This item hasn't been started

Github-Recommendation-System #7
Develop chatbot framework and integrate NLP for user interaction.
Phase 2 ML Engineer

Github-Recommendation-System #8
Implement the recommendation logic in the chatbot system.
Phase 2 ML Engineer

Github-Recommendation-System #10
Develop a deployment strategy for the recommendation system and chatbot. (Documentation)
Phase 3 ML Engineer

Github-Recommendation-System #11
Set up a CI/CD pipeline for automated deployment.
Phase 3 MLOps

Github-Recommendation-System #13
Implement MLOps practices for monitoring and maintaining the system.
Phase 3 MLOps

Github-Recommendation-System #14
Prepare final project presentation and documentation.
Phase 4 Project Manager (PM)

Github-Recommendation-System #15
Evaluate the project against the criteria and make necessary adjustments.
Phase 4 Project Manager (PM)

In Progress 3
This is actively being worked on

Github-Recommendation-System #4
Build a data pipeline for continuous data ingestion and preprocessing.
Phase 1 Data Engineer

Github-Recommendation-System #9
Design and develop user front end
Phase 2 ML Engineer

Github-Recommendation-System #12
Implement testing frameworks (unit tests, integration tests).
Phase 3 MLOps

Done 8
This has been completed

Github-Recommendation-System #1
Define Project scope and Objectives (Documentation)
Phase 1 Project Manager (PM)

Github-Recommendation-System #2
Set up GitHub repository with branch rules, PR guidelines, and issue templates.
Phase 1 Project Manager (PM)

Github-Recommendation-System #6
Design recommendation algorithms
Phase 2 Data Scientist

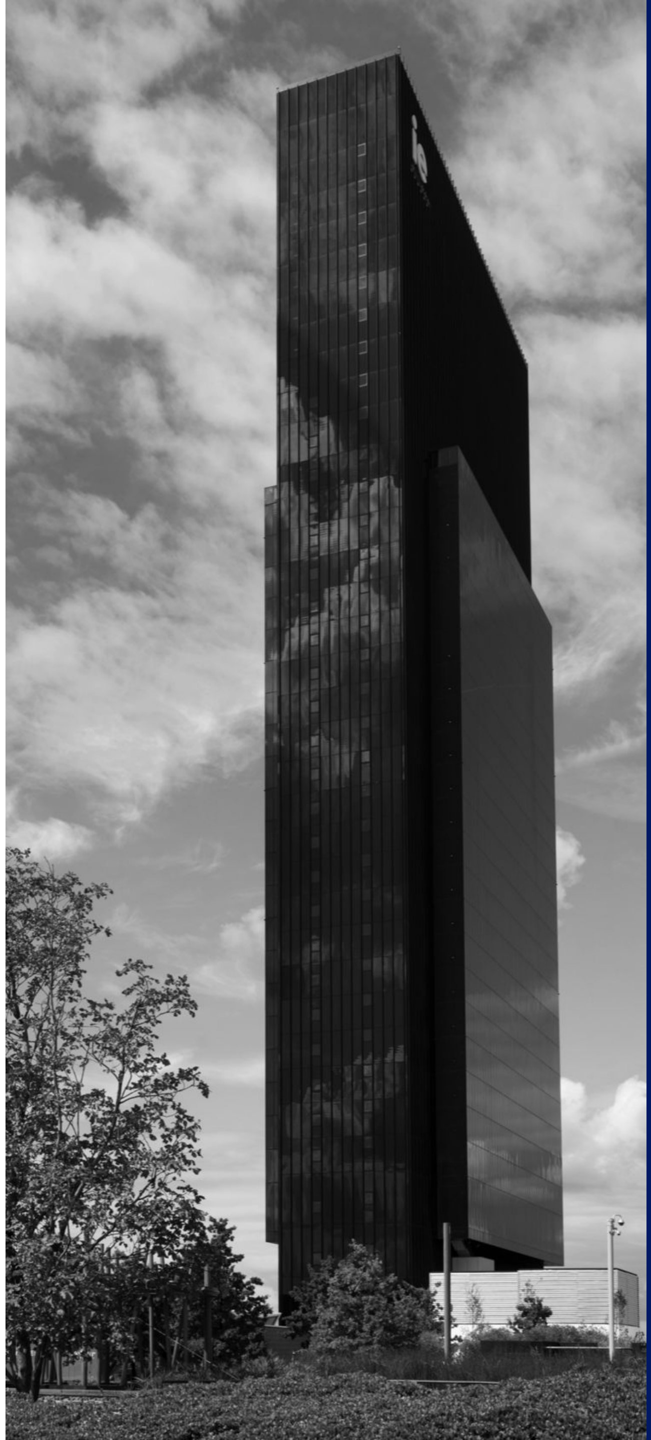
Github-Recommendation-System #27
Automated tests for API code

Github-Recommendation-System #25
Path compatability and secrets

Github-Recommendation-System #3
Access and Retrieve data from the GitHub API.
Phase 1 Data Engineer

Github-Recommendation-System #34
Data Modelling for Repository Recommendation System

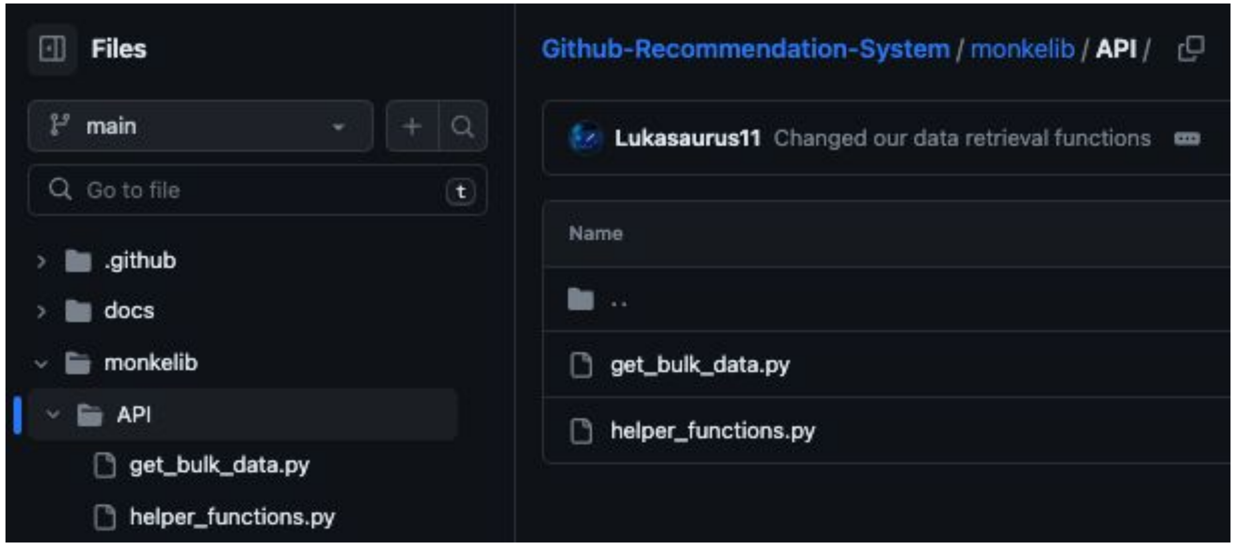
Github-Recommendation-System #5
Perform Exploratory Data Analysis (EDA) to understand the data.
Phase 1 Data Scientist



Progress

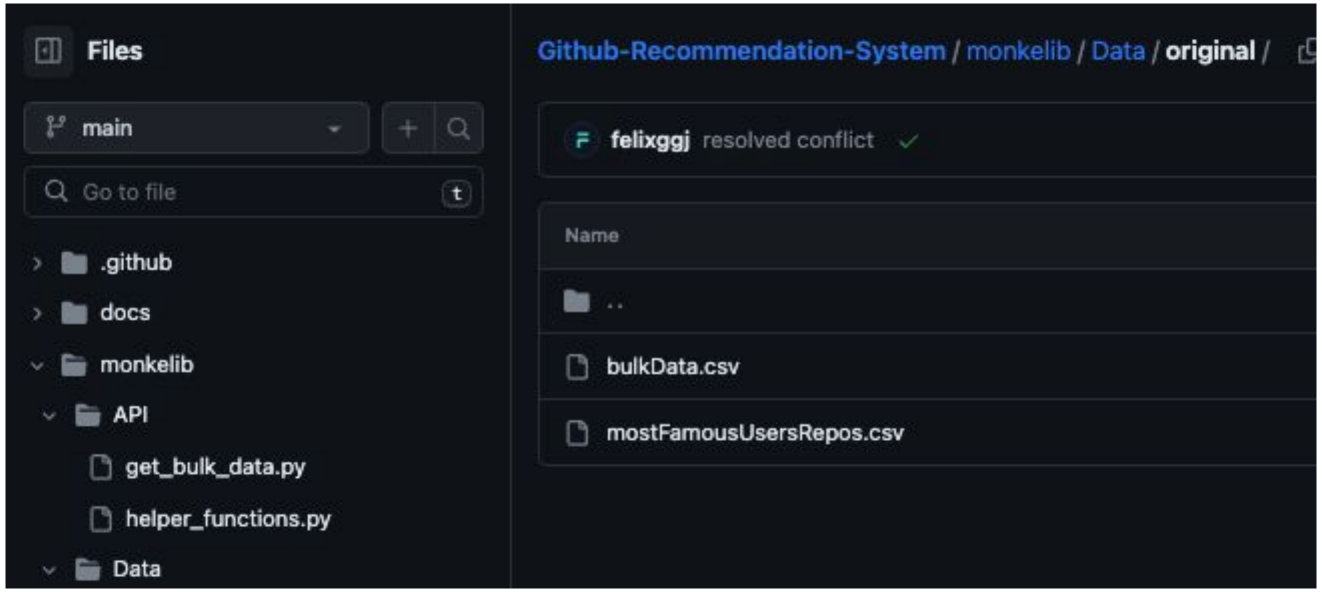
UNIVERSITY

Data Engineering / Prep



- *Utilizing Github API to retrieve repository and profile information*
- *All code files contained within monkelib/API folder*
  - *get_bulk_data.py:*
    - *Loads Github API token*
    - *Extracts Repository data*
    - *Saves data to CSV*
  - *Helper_functions.py: Contains utility functiontions for tasks like loading the authentication token, saving data, and structuring repository information.*

Data Engineering / Prep



Files

main

Go to file

- .github
- docs
- monkelib
  - API
    - get_bulk_data.py
    - helper_functions.py
  - Data

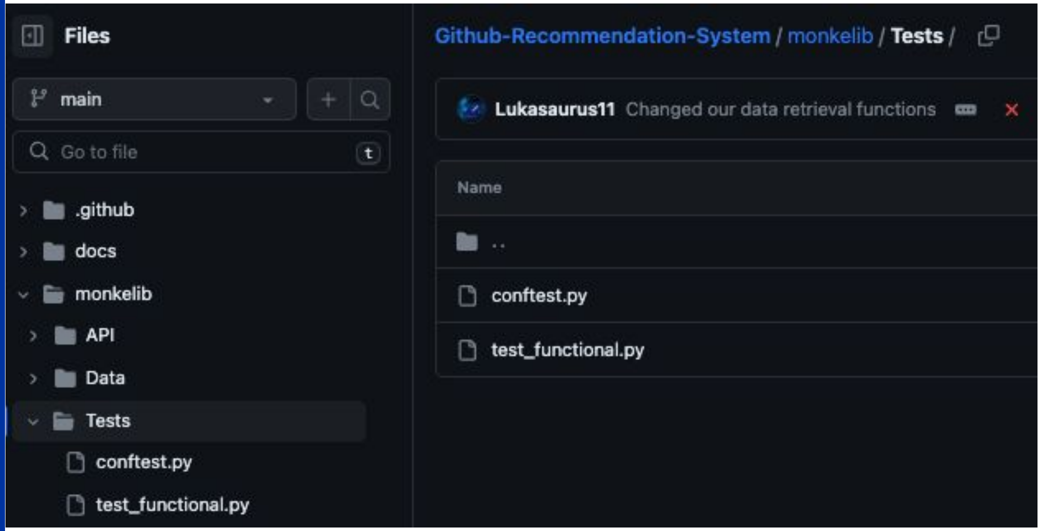
Github-Recommendation-System / monkelib / Data / original /

felixggj resolved conflict ✓

Name
..
bulkData.csv
mostFamousUsersRepos.csv

Outputting results of data extraction into /Data/original folder for the Data Scientists to use for modeling

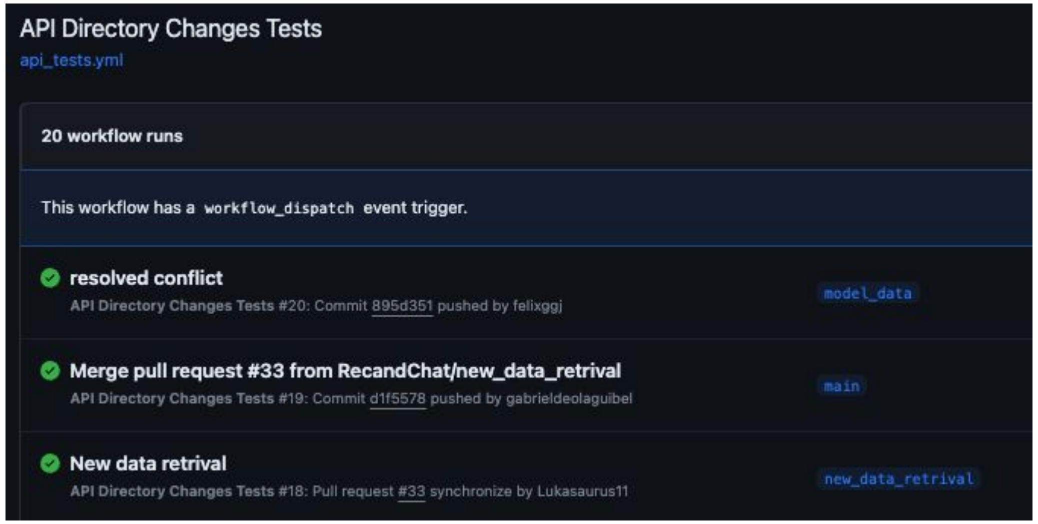
Data Engineering Tests


Files
main
Go to file
.github
docs
monkelib
API
Data
Tests
conftest.py
test_functional.py

Github-Recommendation-System / monkelib / Tests /

Lukasaurus11 Changed our data retrieval functions

Name
..
conftest.py
test_functional.py


API Directory Changes Tests
api_tests.yml

20 workflow runs

This workflow has a `workflow_dispatch` event trigger.

resolved conflict
API Directory Changes Tests #20: Commit 895d351 pushed by felixggj — model_data

Merge pull request #33 from RecandChat/new_data_retrival
API Directory Changes Tests #19: Commit d1f5578 pushed by gabrieldeolaguibel — main

New data retrival
API Directory Changes Tests #18: Pull request #33 synchronize by Lukasaurus11 — new_data_retrival

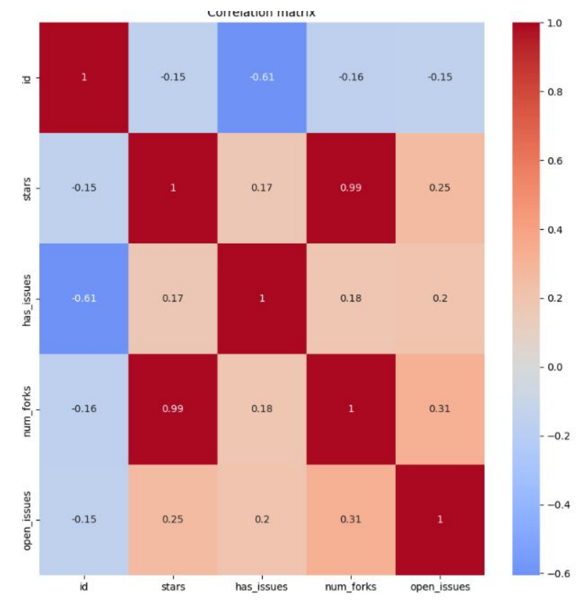
***Tests (monkelib/Tests)**: Ensures the reliability and functionality of data retrieval from the GitHub API.*

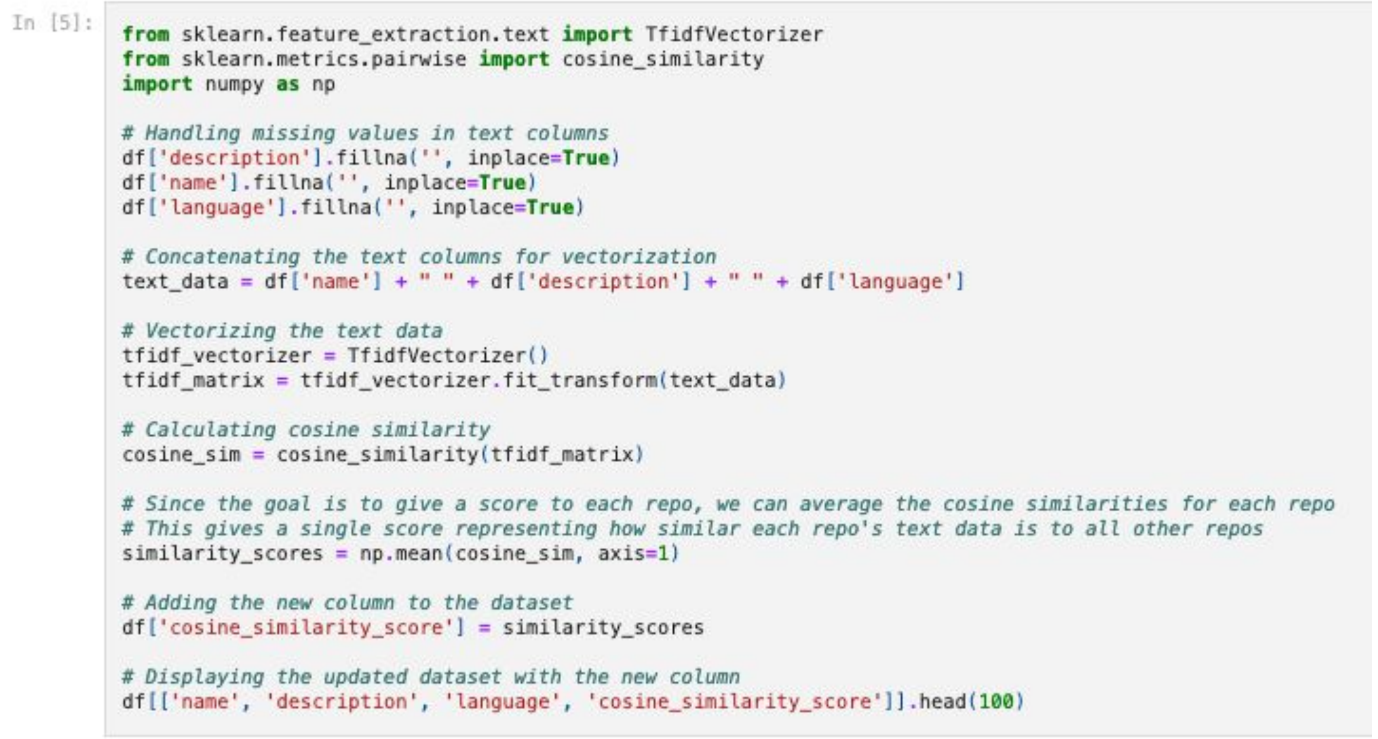
- *conftest.py*
  - *defines fixtures for standardized test data.*
  - *Provides sample users and configurations for other test scripts.*
- *Test_functional.py*
  - *Contains functional tests for data retrieval functions.*
  - *Validates user, follower, and repository data fetching.*
  - *Tests both success and failure scenarios to ensure robustness.*

***Workflow Triggers (.github/workflows/api_tests.yml)**: Automated testing in response to changes within the project, specifically for code in the monkelib/API directory.*

- *On push, PR, Manual*

EDA / Feature Engineering



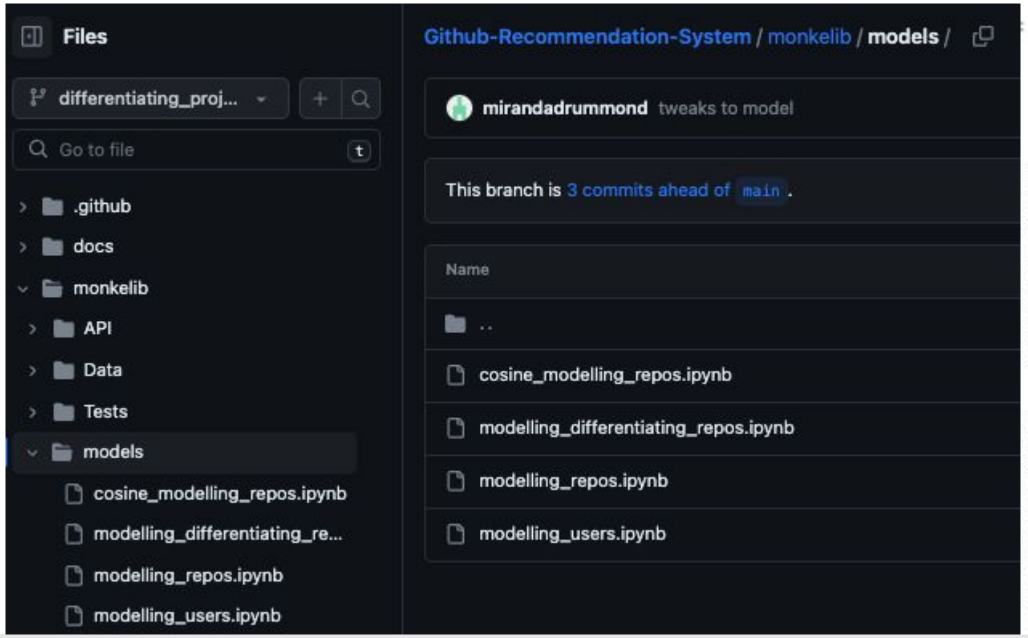


In [5]:
```python
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np

# Handling missing values in text columns
df['description'].fillna('', inplace=True)
df['name'].fillna('', inplace=True)
df['language'].fillna('', inplace=True)

# Concatenating the text columns for vectorization
text_data = df['name'] + " " + df['description'] + " " + df['language']

# Vectorizing the text data
tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix = tfidf_vectorizer.fit_transform(text_data)

# Calculating cosine similarity
cosine_sim = cosine_similarity(tfidf_matrix)

# Since the goal is to give a score to each repo, we can average the cosine similarities for each repo
# This gives a single score representing how similar each repo's text data is to all other repos
similarity_scores = np.mean(cosine_sim, axis=1)

# Adding the new column to the dataset
df['cosine_similarity_score'] = similarity_scores

# Displaying the updated dataset with the new column
df[['name', 'description', 'language', 'cosine_similarity_score']].head(100)
```

EDA and Feature Engineering:

- ***Data Preparation***: *Loads and cleans data, removing unnecessary columns and handling missing values.*
- ***Visualization and Correlation***: *Utilizes plots to visualize data and performs correlation analysis to identify key relationships.*
- ***Encoding and Vectorization***: *Applies one-hot encoding to categorical data and uses TfidfVectorizer for textual data to prepare it for machine learning.*
- ***Normalization***: *Scales numerical features to a uniform range using techniques like MinMaxScaler.*
- ***Data Preparation for ML***: *Converts processed data into tensors, making it suitable for training with PyTorch models.*

Recommendation Models



Files

differentiating_proj...

Go to file

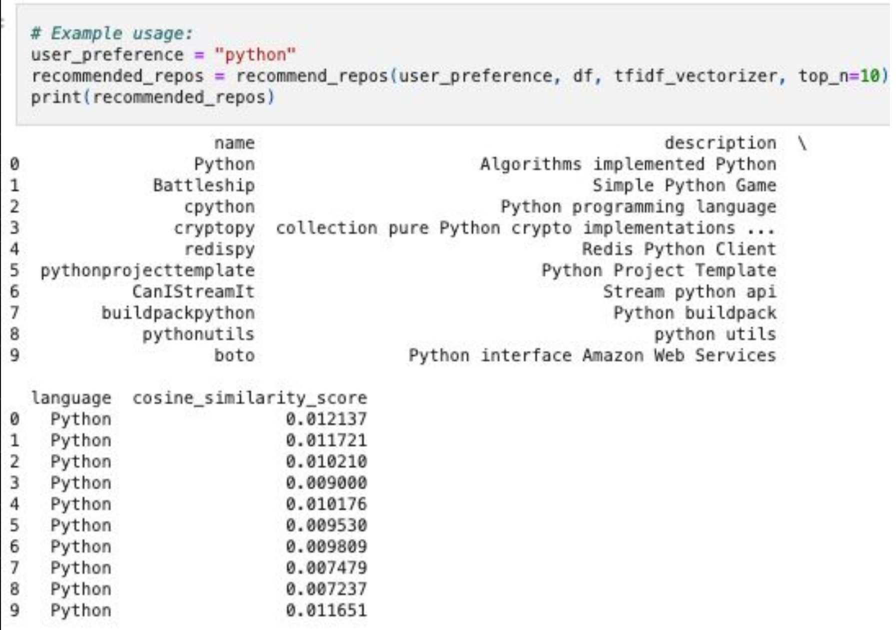
- .github
- docs
- monkelib
  - API
  - Data
  - Tests
  - models
    - cosine_modelling_repos.ipynb
    - modelling_differentiating_re...
    - modelling_repos.ipynb
    - modelling_users.ipynb

Github-Recommendation-System / monkelib / models /

mirandadrummond tweaks to model

This branch is 3 commits ahead of `main` .

Name
..
cosine_modelling_repos.ipynb
modelling_differentiating_repos.ipynb
modelling_repos.ipynb
modelling_users.ipynb



```python
# Example usage:
user_preference = "python"
recommended_repos = recommend_repos(user_preference, df, tfidf_vectorizer, top_n=10)
print(recommended_repos)
```

```
                 name                                  description  \
0              Python                 Algorithms implemented Python
1          Battleship                            Simple Python Game
2             cpython                   Python programming language
3            cryptopy  collection pure Python crypto implementations ...
4             redispy                           Redis Python Client
5 pythonprojecttemplate                      Python Project Template
6          CanIStreamIt                            Stream python api
7       buildpackpython                              Python buildpack
8          pythonutils                                  python utils
9                boto           Python interface Amazon Web Services

  language  cosine_similarity_score
0   Python                 0.012137
1   Python                 0.011721
2   Python                 0.010210
3   Python                 0.009000
4   Python                 0.010176
5   Python                 0.009530
6   Python                 0.009809
7   Python                 0.007479
8   Python                 0.007237
9   Python                 0.011651
```

Models

cosine_modelling_repos.ipynb:

- *Uses cosine similarity to recommend repositories based on language preferences.*
- *Prepares text data for similarity scoring and vectorization.*

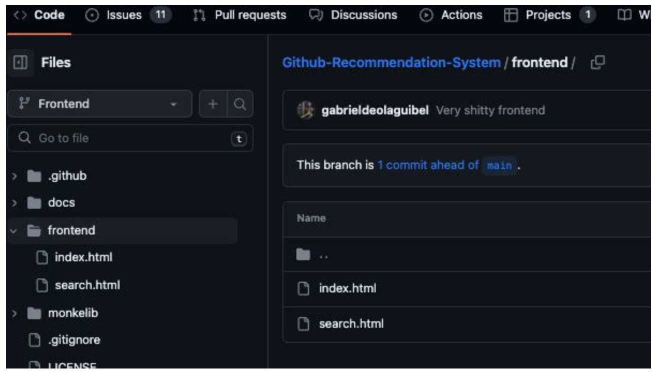
modelling_repos.ipynb:

- *Explores combining repository data with user interactions for recommendations.*
- *Implements feature engineering, including encoding and vectorization, for model input preparation.*

modelling_differentiating_repos.ipynb

- *Utilized domain specific embedding method to vectorize users.*
- *Finds the most differentiating project among the most similar users of the target user.*

Simple Front end



Code · Issues 11 · Pull requests · Discussions · Actions · Projects 1 · W

Files

Frontend

Go to file t

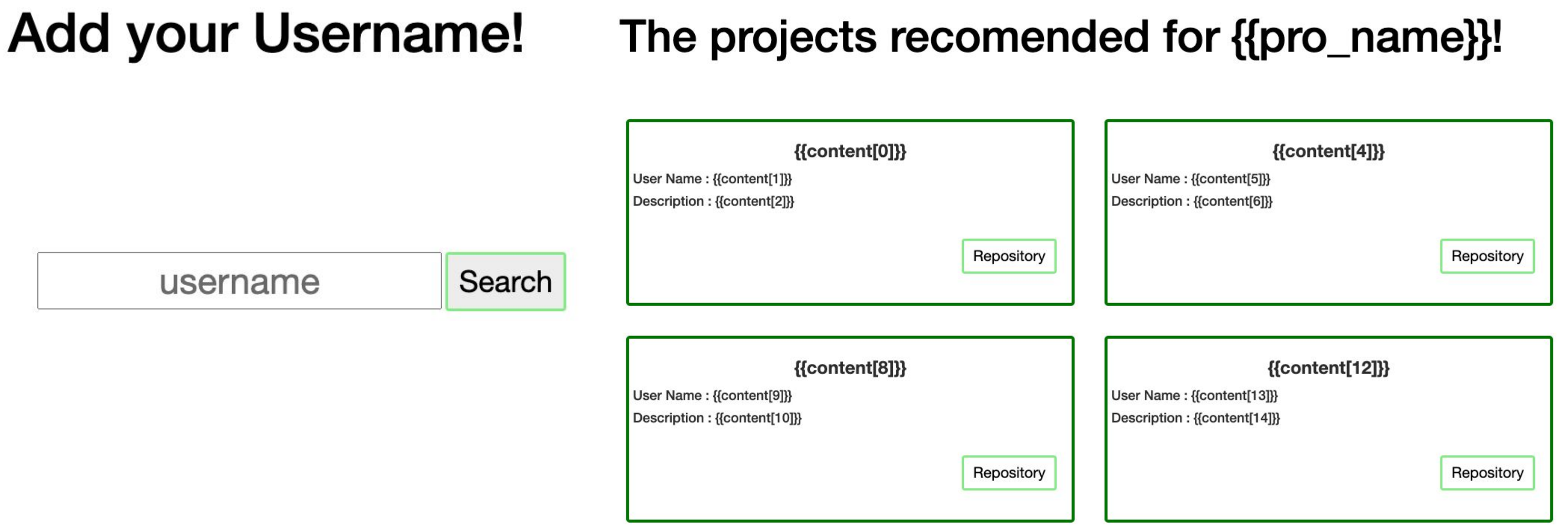
- .github
- docs
- frontend
  - index.html
  - search.html
- monkelib
- .gitignore
- LICENSE

Github-Recommendation-System / frontend /

gabrieldeolaguibel Very shitty frontend

This branch is 1 commit ahead of `main` .

Name
- ..
- index.html
- search.html



Add your Username!

username | Search

The projects recomended for {{pro_name}}!

{{content[0]}}
User Name : {{content[1]}}
Description : {{content[2]}}
Repository

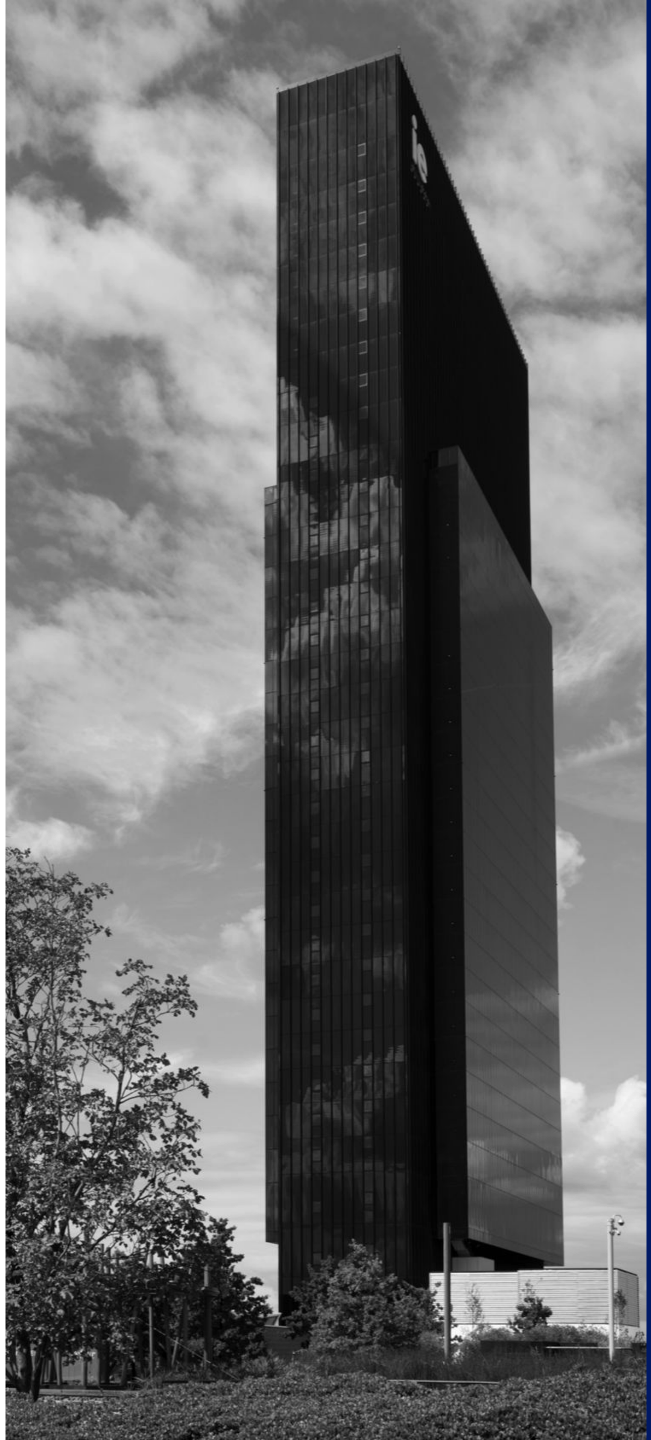
{{content[4]}}
User Name : {{content[5]}}
Description : {{content[6]}}
Repository

{{content[8]}}
User Name : {{content[9]}}
Description : {{content[10]}}
Repository

{{content[12]}}
User Name : {{content[13]}}
Description : {{content[14]}}
Repository

Under development within ‘frontend’ Branch

- *Users will provide their github Username and receive a list of the most recommended repositories they may be interested in based on their profile*
- *Requires flask connection to the model*

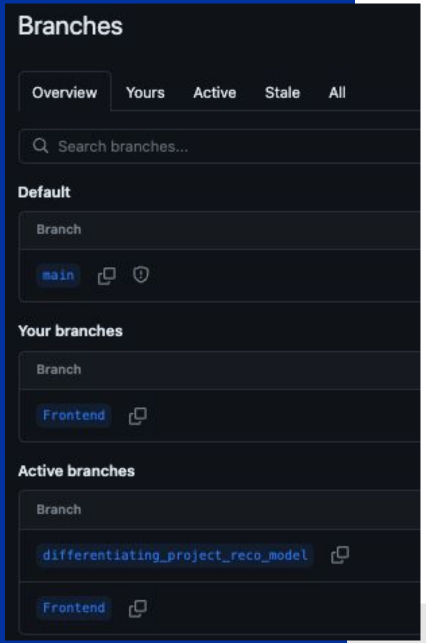


Development Methodology

UNIVERSITY

Development Methodology

Good Coding Practices


Branches

Overview Yours Active Stale All

Search branches...

Default

Branch

main

Your branches

Branch

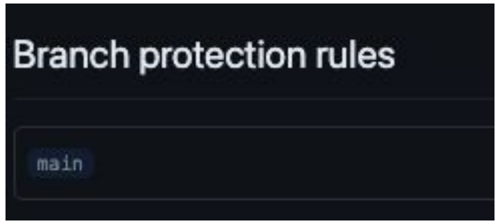
Frontend

Active branches

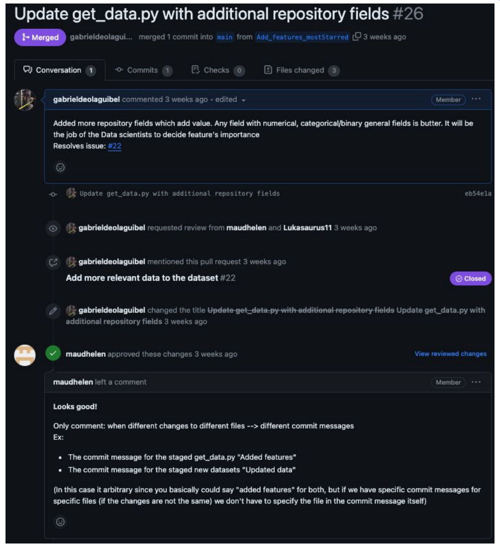
Branch

differentiating_project_reco_model

Frontend


Branch protection rules

main


Update get_data.py with additional repository fields #26

Merged gabrieldeolagui... merged 1 commit into main from Add_features_mostStarred 3 weeks ago

Conversation 1 Commits 1 Checks 0 Files changed 3

gabrieldeolaguibel commented 3 weeks ago • edited Member

Added more repository fields which add value. Any field with numerical, categorical/binary general fields is butter. It will be the job of the Data scientists to decide feature's importance
Resolves issue: #22

Update get_data.py with additional repository fields eb54e1a

gabrieldeolaguibel requested review from maudhelen and Lukasaurus11 3 weeks ago

gabrieldeolaguibel mentioned this pull request 3 weeks ago
Add more relevant data to the dataset #22 Closed

gabrieldeolaguibel changed the title ~~Update get_data.py with additional repository fields~~ Update get_data.py with additional repository fields 3 weeks ago

maudhelen approved these changes 3 weeks ago View reviewed changes

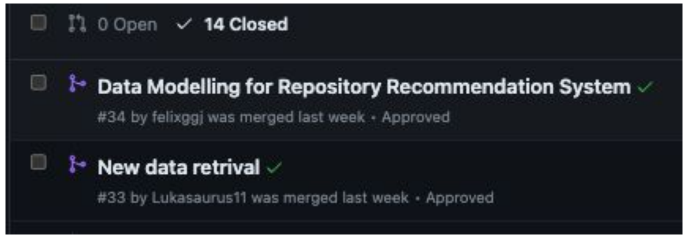
maudhelen left a comment Member

Looks good!

Only comment: when different changes to different files --> different commit messages
Ex:

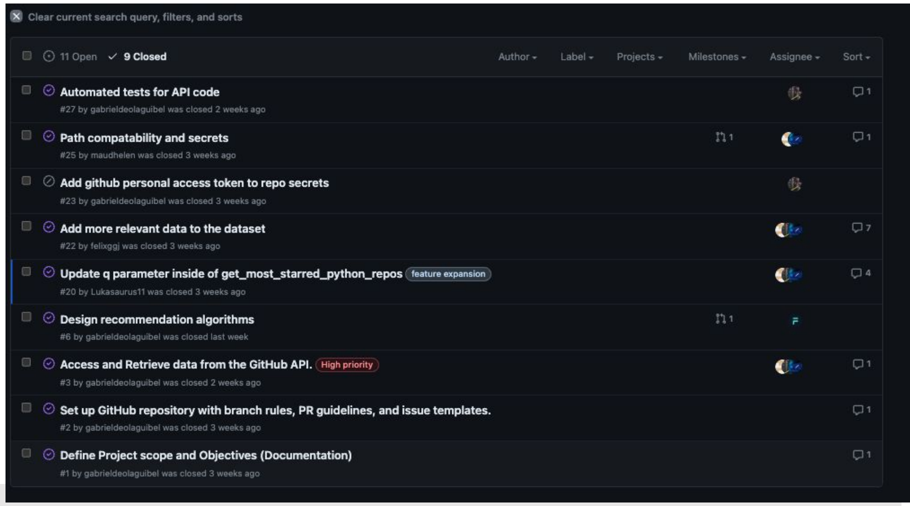
- The commit message for the staged get_data.py "Added features"
- The commit message for the staged new datasets "Updated data"

(In this case it arbitrary since you basically could say "added features" for both, but if we have specific commit messages for specific files (if the changes are not the same) we don't have to specify the file in the commit message itself)


0 Open ✓ 14 Closed

Data Modelling for Repository Recommendation System ✓
#34 by felixggj was merged last week • Approved

New data retrival ✓
#33 by Lukasaurus11 was merged last week • Approved


Clear current search query, filters, and sorts

11 Open ✓ 9 Closed Author Label Projects Milestones Assignee Sort

Automated tests for API code 1
#27 by gabrieldeolaguibel was closed 2 weeks ago

Path compatability and secrets 1 1
#25 by maudhelen was closed 3 weeks ago

Add github personal access token to repo secrets
#23 by gabrieldeolaguibel was closed 3 weeks ago

Add more relevant data to the dataset 7
#22 by felixggj was closed 3 weeks ago

Update q parameter inside of get_most_starred_python_repos feature expansion 4
#20 by Lukasaurus11 was closed 3 weeks ago

Design recommendation algorithms 1
#6 by gabrieldeolaguibel was closed last week

Access and Retrieve data from the GitHub API. High priority 1
#3 by gabrieldeolaguibel was closed 2 weeks ago

Set up GitHub repository with branch rules, PR guidelines, and issue templates. 1
#2 by gabrieldeolaguibel was closed 3 weeks ago

Define Project scope and Objectives (Documentation) 1
#1 by gabrieldeolaguibel was closed 3 weeks ago

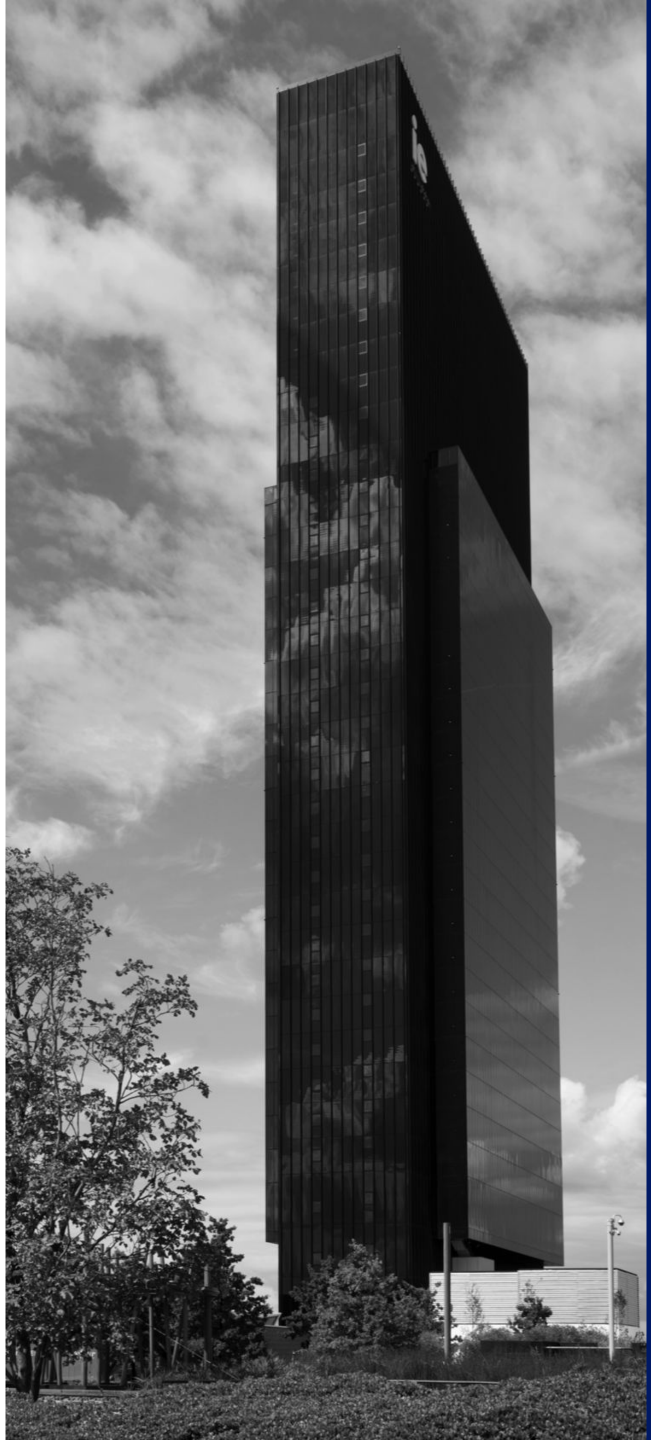
Required development in short-lived branches:
- *No Member is allowed to push to main*
- *Branches are deleted after merges to main*

Add code via PRs:
- *Requires another member to review and comment.*
- *Requires automated workflow tests to pass*

Issues:
- *Issues are part of the github project backlog*
- *Members submit issue with what need to be fixed/done and get feedback from others on the ideas*
- *PRs directly address issues*

Libraries and Repository design:
- *Notebooks and scripts call on monkelib library or external libraries*
- *Code is neatly organized in its respective folders following “Project template” design by Miguel.*

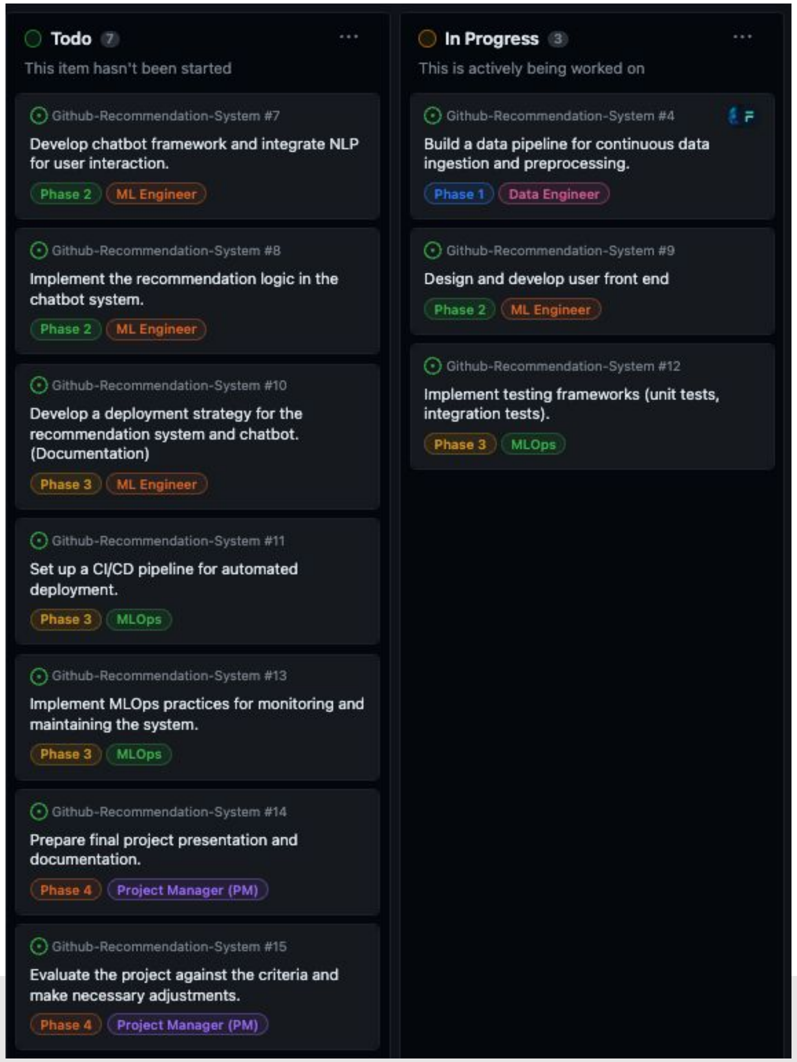
ie UNIVERSITY



Future Roadmap

UNIVERSITY

Roadmap



Todo 7
This item hasn't been started

Github-Recommendation-System #7
Develop chatbot framework and integrate NLP for user interaction.
Phase 2 · ML Engineer

Github-Recommendation-System #8
Implement the recommendation logic in the chatbot system.
Phase 2 · ML Engineer

Github-Recommendation-System #10
Develop a deployment strategy for the recommendation system and chatbot. (Documentation)
Phase 3 · ML Engineer

Github-Recommendation-System #11
Set up a CI/CD pipeline for automated deployment.
Phase 3 · MLOps

Github-Recommendation-System #13
Implement MLOps practices for monitoring and maintaining the system.
Phase 3 · MLOps

Github-Recommendation-System #14
Prepare final project presentation and documentation.
Phase 4 · Project Manager (PM)

Github-Recommendation-System #15
Evaluate the project against the criteria and make necessary adjustments.
Phase 4 · Project Manager (PM)

In Progress 3
This is actively being worked on

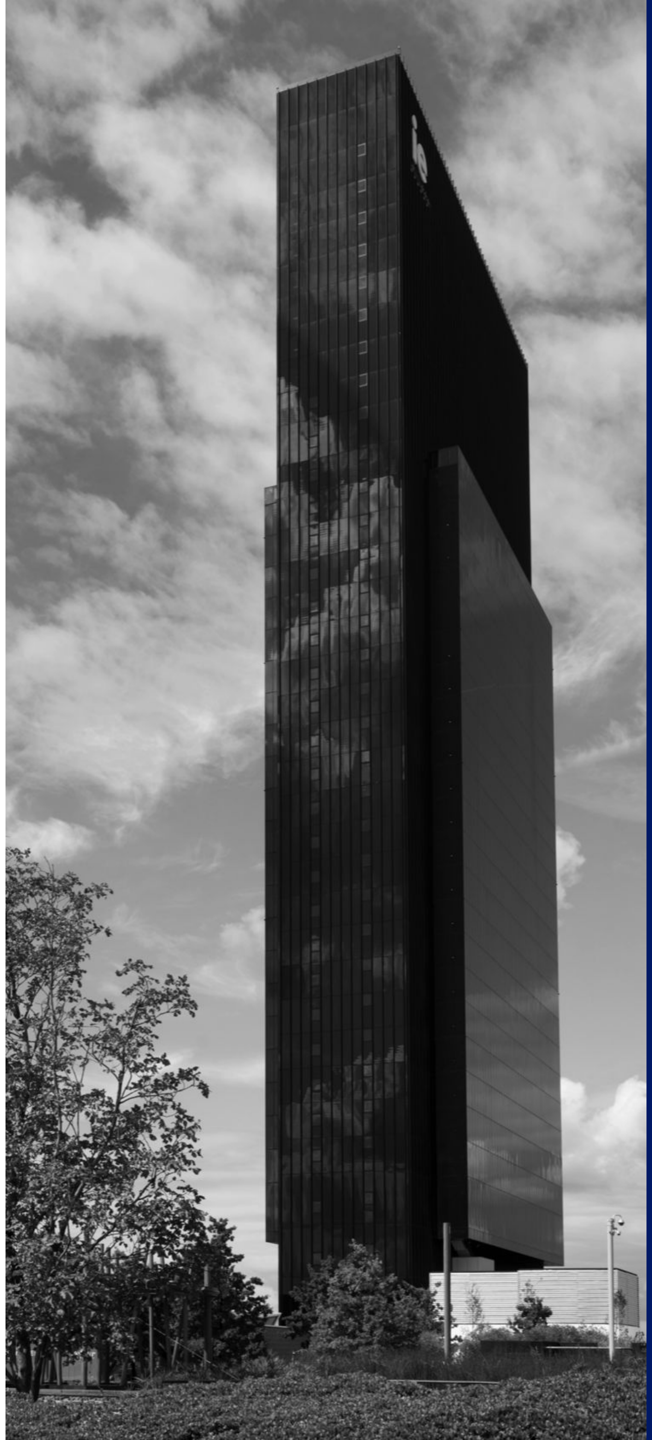
Github-Recommendation-System #4
Build a data pipeline for continuous data ingestion and preprocessing.
Phase 1 · Data Engineer

Github-Recommendation-System #9
Design and develop user front end
Phase 2 · ML Engineer

Github-Recommendation-System #12
Implement testing frameworks (unit tests, integration tests).
Phase 3 · MLOps

Future progress focus:
Deployment to frontend
MLOps
Chatbot

ie UNIVERSITY



Feedback / Questions

UNIVERSITY