Team TRISHUL

Project Plan

Project Objectives:

The objective of our project is to develop a robust sentiment analysis system for fake news detection. We aim to leverage natural language processing (NLP) techniques and machine learning algorithms to analyze the emotional tone of news articles and identify potential instances of fake news. By achieving this, we strive to address the growing issue of misinformation and its detrimental impact on society.

Our project seeks to achieve the following key objectives:

- Accurate Fake News Detection: We aim to develop a system that can effectively differentiate between credible and fake news articles. By analyzing the sentiment expressed in the text, our system will identify subjective biases, misleading information, and potential manipulation, enabling users to make informed judgments about the authenticity of news content.
- User-friendly Interface: We will design and develop a user-friendly interface that allows users to easily interact with the system. The interface will provide a seamless experience for users to input news articles, receive sentiment analysis results, and obtain insights into the credibility of the news content.
- Real-time Analysis: Our system will be designed to perform sentiment analysis and fake news detection in real-time. This capability will enable timely identification of potential fake news, empowering users to verify the accuracy and credibility of news articles as they consume information.
- Robust Performance: We aim to optimize the performance of our sentiment analysis system to handle large volumes of news articles efficiently. Through careful implementation and algorithmic enhancements, we will strive to achieve high accuracy, precision, and recall rates in fake news detection.
- Integration and Scalability: Our system will be designed to integrate with various data sources and platforms. We will ensure that the system can handle diverse data formats and seamlessly communicate with other modules, making it scalable and adaptable to different environments.
- Ethical Considerations: Throughout the development process, we will prioritize ethical considerations such as user privacy, data security, and fairness. We will adhere to best practices and guidelines to ensure that our system operates with transparency, accountability, and respect for user rights.

By achieving these objectives, our project aims to contribute to the fight against fake news and promote a more informed and trustworthy information ecosystem. We believe that our sentiment analysis system will provide valuable insights for individuals, journalists, and organizations to make informed decisions, foster critical thinking, and combat the spread of misinformation.

Sprint Plan:

Sprint 1:

Duration: Week 1 - Week 2

Tasks:

- Conduct research on sentiment analysis techniques and fake news detection algorithms.
- Define project requirements and specifications.
- Set up the development environment and establish version control.
- Assign roles and responsibilities to team members.
- Deadline: Complete research and finalize project requirements by the end of Week 2.

Sprint 2:

Duration: Week 3 - Week 4

Tasks:

- Design the user interface and create wireframes.
- Develop the front-end of the system using HTML, CSS, and JavaScript.
- Implement basic server-side functionality for data storage and retrieval.
- Begin working on the sentiment analysis algorithm.
- Deadline: Complete the user interface design, front-end development, and initial server-side functionality by the end of Week 4.

Sprint 3:

Duration: Week 5 - Week 6

Tasks:

- Refine and optimize the sentiment analysis algorithm.
- Train machine learning models using labeled datasets.
- Implement the back-end functionality for sentiment analysis and fake news detection.
- Conduct comprehensive testing to ensure accuracy and reliability.
- Deadline: Complete the sentiment analysis algorithm, machine learning models, and back-end functionality by the end of Week 6.

Sprint 4:

Duration: Week 7 - Week 8

Tasks:

- Integrate the front-end and back-end components of the system.
- Implement real-time analysis capabilities.
- Perform system testing and debugging.
- Enhance the user interface and improve user experience.

- Deadline: Complete system integration, real-time analysis, and user interface enhancements by the end of Week 8.

Sprint 5:

Duration: Week 9 - Week 10

Tasks:

- Conduct thorough performance testing and optimization.
- Address any remaining bugs or issues.
- Finalize documentation, including user guides and technical documentation.
- Prepare for the final presentation and demo.
- Deadline: Complete performance testing, bug fixing, documentation, and preparation for the final presentation by the end of Week 10.

Roles:

- Cloud Engineer: Harnoor Singh Chawla, Vaibhav Ranka

Role: As a Cloud Engineer, we will be to design, deploy, and manage the cloud infrastructure required to support the generative AI and large language model solution. We will ensure scalability, reliability, and security in the cloud environment. Additionally, we will optimize the system's performance by leveraging cloud-native technologies and services, such as autoscaling, load balancing, and containerization.

- Front-end Developer: Vikram Kumar

Role: As a Front-end Developer, I will focus on designing and developing the user interface (UI) for the generative AI and large language model solution. I will collaborate with UX Lead to translate user requirements into intuitive and visually appealing interfaces. My expertise in front-end technologies, such as HTML, CSS, and JavaScript, will be crucial in creating interactive and user-friendly experiences.

- Back-end Developer(s): Vaibhav Ranka, Saurabh Anand, Harnoor Singh Chawla

Role: As a Back-end Developer, we will be responsible for developing and maintaining the server-side components of the generative AI and large language model solution. We will work closely with Data Science/Machine Learning team to implement the algorithms and models, handle data storage and retrieval, and manage the system's logic and functionality. Proficiency in programming languages like Python, Java, or C#, and experience with web frameworks (e.g., Django, Flask) will be essential for this role.

- Data Science / Machine Learning: Vikram Kumar, Arushi Khera

Role: As a Data Science/Machine Learning expert, we will play a crucial role in developing machine learning algorithms, natural language processing, and deep learning techniques to train and optimize the models. Our responsibilities will include data preprocessing, feature engineering, model selection, evaluation, and ongoing research to improve the models' performance and generative capabilities.

- Testing: Harnoor Singh Chawla, Saurabh Anand, Harshkumar Mehta

Role: As a Testing professional, our role will be to ensure the quality, reliability, and accuracy of the generative AI and large language model solution. We will develop and execute comprehensive testing strategies, including unit testing, integration testing, and regression testing, to identify and address any issues or bugs. We will work closely with the development team to conduct rigorous testing throughout the development lifecycle, ensuring a robust and stable solution.

- Integration and Middleware: Saurabh Anand, Harshkumar Mehta

Role: As an Integration and Middleware specialist, we will focus on seamlessly integrating various components and services within the generative AI and large language model solution. We will design and implement middleware solutions to facilitate communication and data flow between different modules and systems. Our expertise in middleware technologies, such as RESTful APIs and message queues, will be critical in establishing efficient and reliable integration.

- UX Lead: Arushi Khera, Vikram Kumar

Role: As a UX Lead, our role will be to lead the user experience design efforts for the generative AI and large language model solution. We will conduct user research, define user personas, and create wireframes and prototypes to ensure a user-centric approach. Collaborating with Front-end Developers and other team members, we will translate user requirements into intuitive and engaging interfaces, ensuring a seamless and enjoyable user experience.

Architecture setup:

Architecture Setup and Cloud Environment:

For our project, we have chosen to utilize GitHub as our cloud environment and version control system. GitHub provides a robust platform for collaboration, code management, and version tracking, making it an ideal choice for our team's development needs.

Our architecture setup involves the following components:

- GitHub Repository: We have created a dedicated repository on GitHub to store and manage our project codebase. This repository serves as a centralized location where

team members can collaborate, contribute, and track changes to the code throughout the project's lifecycle.

- Branching Strategy: We follow a branching strategy on GitHub, utilizing branches to manage different development stages and features. This allows team members to work on separate branches, making it easier to review, merge, and manage code changes without interfering with the main codebase.
- Pull Requests: We use pull requests on GitHub to facilitate code reviews and ensure quality control. Team members submit their changes as pull requests, enabling other members to review the code, provide feedback, and suggest improvements before merging the changes into the main branch.
- Issue Tracking: GitHub's issue tracking system helps us manage and track project tasks, bugs, and enhancements. We create issues for specific features or tasks, assign them to team members, and track their progress throughout the project.
- Collaboration and Documentation: GitHub provides various collaboration features such as commenting, discussions, and documentation through Markdown files. These features enable effective communication, documentation of important decisions, and sharing of knowledge among team members.

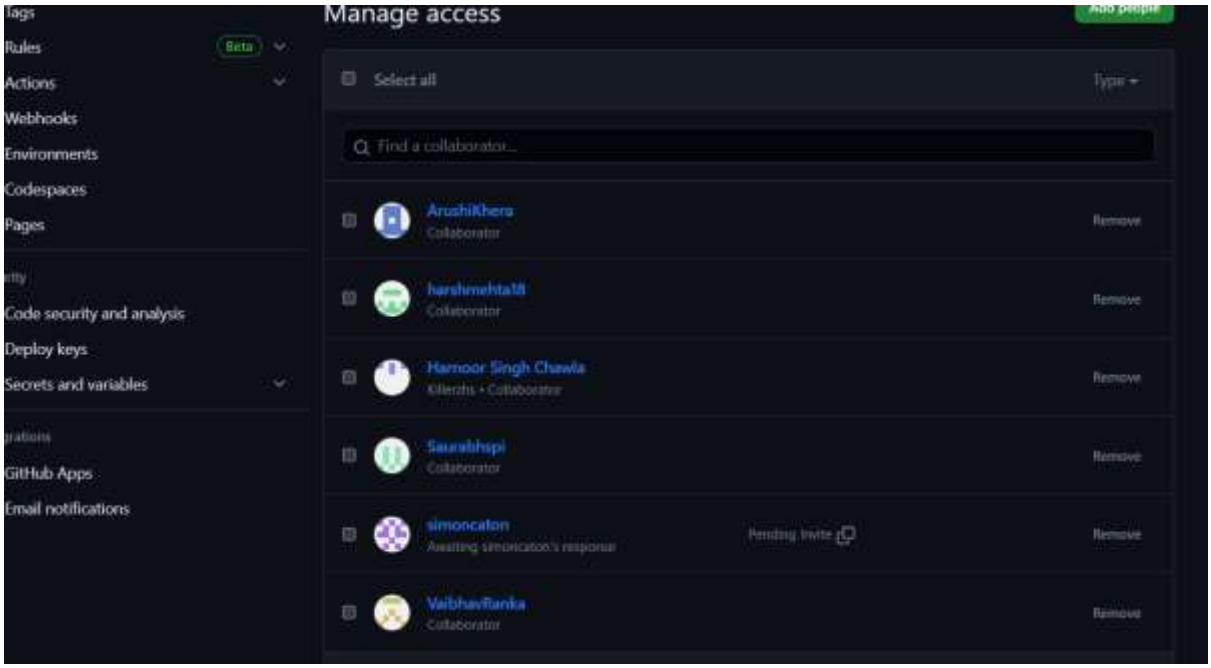
Evidence of our Cloud Environment:

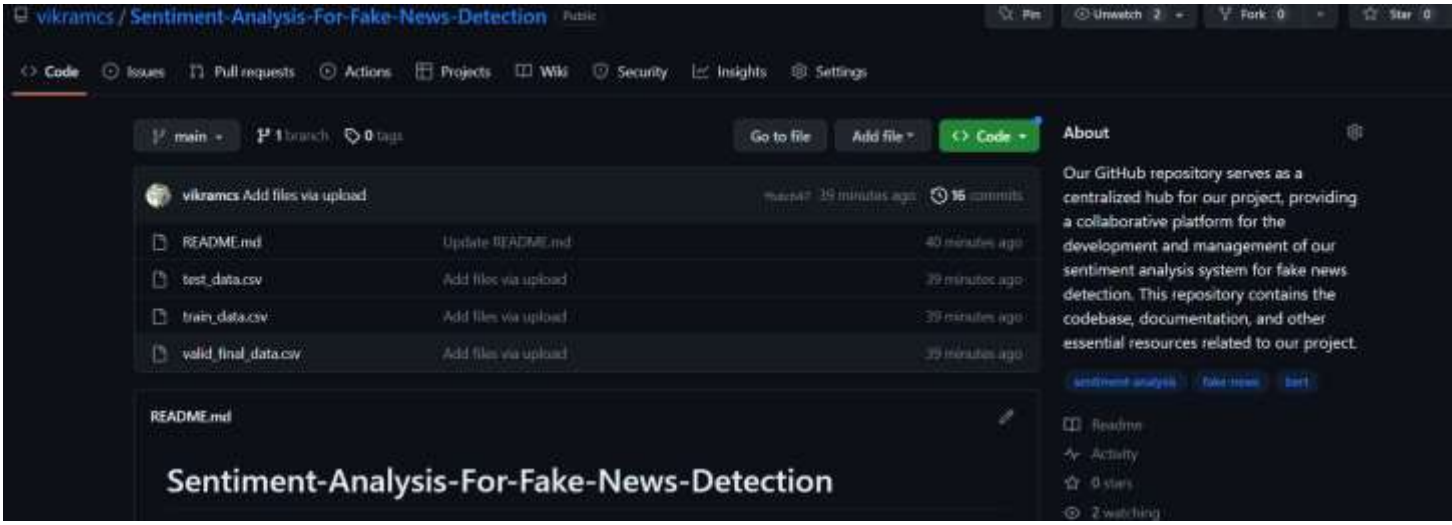
To provide evidence of our GitHub cloud environment setup, we can showcase the following:

Provide the link to our GitHub repository, which contains the project codebase, commit history, and branch structure.

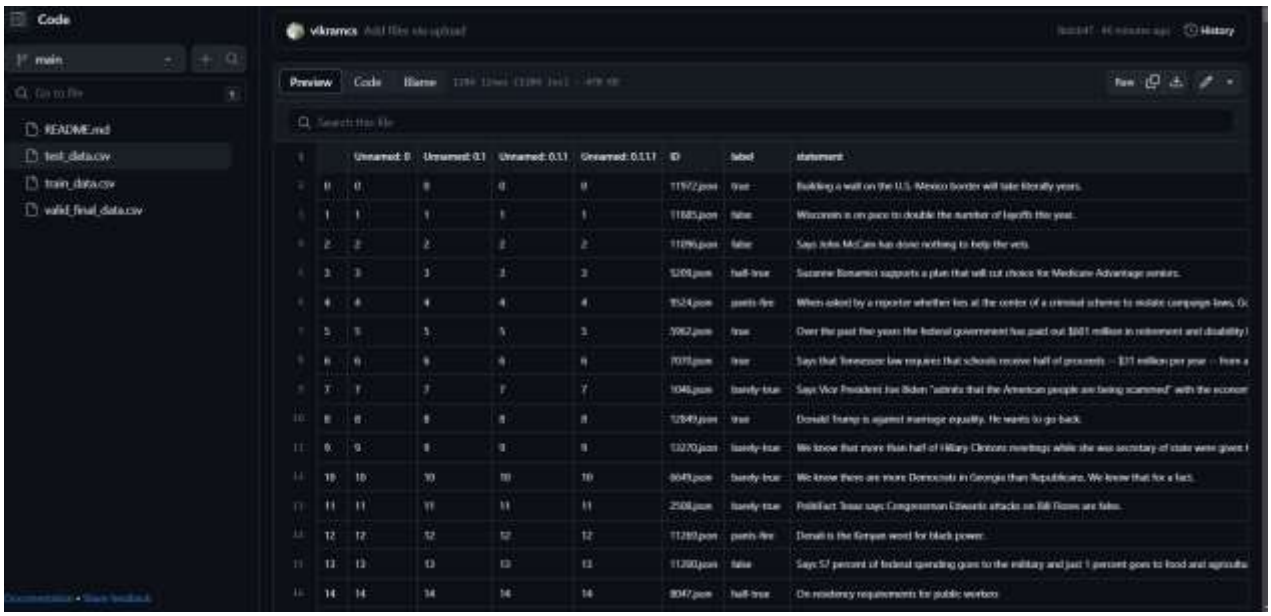
https://github.com/vikramcs/Sentiment-Analysis-For-Fake-News-Detection/tree/main

Share screenshots or examples of pull requests and code reviews conducted on GitHub, demonstrating our team's collaboration and quality control processes.




vikramcs / Sentiment-Analysis-For-Fake-News-Detection
Public
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
vikramcs Add files via upload
16 commits
README.md
Update README.md
test_data.csv
Add files via upload
train_data.csv
Add files via upload
valid_final_data.csv
Add files via upload
README.md
Sentiment-Analysis-For-Fake-News-Detection
About
Our GitHub repository serves as a centralized hub for our project, providing a collaborative platform for the development and management of our sentiment analysis system for fake news detection. This repository contains the codebase, documentation, and other essential resources related to our project.
Readme
Activity
0 stars
2 watching


Community
Community Standards
Traffic
Commits
Code frequency
Dependency graph
Network
Forks
Contributions to main, excluding merge commits and bot accounts
15
10
5
0
Jun 04
vikramcs
#1
Killerzhs
#2
5/29/2023
6/1/2023
6/4/2023
6/7/2023


Code
vikramcs
History
main
Preview
Code
README.md
test_data.csv
train_data.csv
valid_final_data.csv
Unnamed: 0
Unnamed: 0.1
Unnamed: 0.1.1
ID
label
statement

By leveraging GitHub as our cloud environment, we ensure a streamlined and efficient development process, effective collaboration among team members, and seamless version control for our project.

Data Plan:

In our project on sentiment analysis for fake news detection, we will utilize a combination of existing datasets and generate additional data to train and evaluate our models. The data we will use includes:

- News Article Dataset: We will acquire a diverse collection of news articles from reliable sources, including both credible and fake news articles. This dataset will serve as the foundation for training and testing our sentiment analysis models. We will ensure that the dataset covers a wide range of topics and includes various sentiments expressed in the articles.
- Labeled Dataset for Training: To train our sentiment analysis models, we will create a labeled dataset where each news article is annotated with sentiment labels, indicating whether the article expresses positive, negative, or neutral sentiment. This dataset will be generated by manually labeling a subset of the acquired news articles.
- Evaluation Dataset: We will reserve a separate portion of the acquired news articles for evaluation purposes. This dataset will be used to assess the performance of our trained models on unseen data and provide insights into the accuracy and effectiveness of our system in detecting fake news sentiment.
- Data Generation: In addition to the existing datasets, we will generate synthetic data to augment our training set. This may include creating variations of existing news articles by altering certain features, such as the headlines, content structure, or sentiment polarity. This data generation process will help increase the diversity and quantity of training data, improving the robustness of our sentiment analysis models.

Evidence of Data Sufficiency:

To demonstrate that the data plan is sufficient for the scope of the project and the addressed challenge, we can provide the following evidence:

To fulfill the data plan requirements and provide evidence of the dataset's sufficiency, we conducted an analysis of the news article dataset. The dataset includes various attributes such as ID, label, statement, subject, speaker, speaker job, state info, party affiliation, counts for different truth labels, context, sentiment, sentiment score, sentiment magnitude, and emotional indicators (anger, fear, joy, disgust, sad).

Summary of the Dataset:

- The dataset contains a diverse range of news articles covering topics such as immigration, jobs, military, campaign advertising, economy, education, healthcare, and more.
- It includes statements made by different speakers, including politicians, representatives, and public figures, along with their party affiliations and job positions.

- The dataset comprises articles from various states across the United States, providing a broad perspective on regional issues.
- Each statement is labeled with a truth rating, ranging from "pants on fire" (false) to "true," allowing for evaluation and analysis of the statements' accuracy.
- The sentiment attribute captures the sentiment expressed in the statements, such as positive, negative, or neutral, along with sentiment scores and magnitudes.
- Emotional indicators (anger, fear, joy, disgust, sad) provide additional insights into the emotional tone of the statements.

The dataset's size and the variety of topics covered make it suitable for addressing the challenge of sentiment analysis and fake news detection. The inclusion of truth labels allows us to evaluate the performance of our sentiment analysis system accurately. The sentiment attributes and emotional indicators provide valuable information for sentiment analysis and understanding the underlying emotions in the statements.

Evaluation Data: For the evaluation of our sentiment analysis system, we will select a subset of the dataset and manually annotate it with ground truth sentiment labels. This subset will be used as the evaluation data to assess the system's performance and compare it against the labeled data.

By utilizing this diverse news article dataset and incorporating manual annotations for evaluation, we ensure that our project's scope is adequately supported, and we have sufficient data to develop and evaluate our sentiment analysis system for fake news detection.

By presenting this evidence, we can establish that our data plan encompasses a comprehensive collection of news articles, a labeled dataset for training, an evaluation dataset, and additional synthetic data generation, ensuring that we have sufficient data to tackle the challenge of sentiment analysis for fake news detection and evaluate the performance of our system effectively.

Plan for Providing GitHub Evidence:

To demonstrate that all members of the team have made regular non-arbitrary commits to the team GitHub repository, we can follow the following plan:

- Establish GitHub Guidelines: At the beginning of the project, establish clear guidelines for team members to regularly contribute to the GitHub repository. Emphasize the importance of committing and pushing updates regularly, and outline the types of contributions that should be made, such as meeting minutes, design diagrams, project plan sections, code snippets, documentation, etc.
- Communication and Collaboration: Encourage team members to communicate and collaborate effectively using GitHub. Encourage them to create and participate in discussions related to the project, share ideas, ask questions, and provide feedback on each other's work.

- Task Tracking: Utilize GitHub's issue tracking system to assign tasks and track progress. Each task can be assigned to a team member, and once completed, it should be committed to the repository with appropriate comments or descriptions.
- Regular Check-ins: Set up regular check-in meetings or milestones where team members provide updates on their progress and commit their work to the GitHub repository. This can be done on a weekly or bi-weekly basis, depending on the project timeline.
- Code Reviews: Encourage team members to conduct code reviews using pull requests on GitHub. This will not only ensure code quality but also provide evidence of active contributions. Require that each pull request contains a clear description of the changes made and any relevant documentation.
- Documentation and Reporting: Ensure that all project-related documents, such as meeting minutes, design diagrams, project plan sections, and figures, are committed to the repository regularly. Encourage team members to document their work and progress throughout the project.
- Review and Verification: Regularly review the GitHub repository to ensure that all team members are actively contributing and making regular commits. Verify that the commits are meaningful and align with the project goals and tasks.

By following this plan, we can provide clear evidence on GitHub that all team members have actively contributed to the project by making regular non-arbitrary commits. This will showcase the team's collaboration, progress, and individual contributions throughout the project's development.

Team Management Evidence:

- Regular Team Meetings:

Our team conducts regular meetings to ensure effective communication, collaboration, and progress tracking. These meetings take place using **Google Meet** and we are using team management tool called **ClickUp** , providing a virtual platform for face-to-face discussions and updates.

- Meeting Format:

Our team meetings are scheduled on a weekly basis, usually on Wednesday and Saturday. The meetings typically last for 40-45 minutes to allow sufficient time for all members to provide updates, discuss challenges, and plan for the upcoming tasks.

- Meeting Minutes:

The minutes capture the key points discussed during the meetings, including progress updates, decisions made, action items, and any other important information. These minutes serve as a record of our discussions and ensure that all team members are well-informed about the project's progress and next steps.

Team: TRISHUL

Project Discussion:

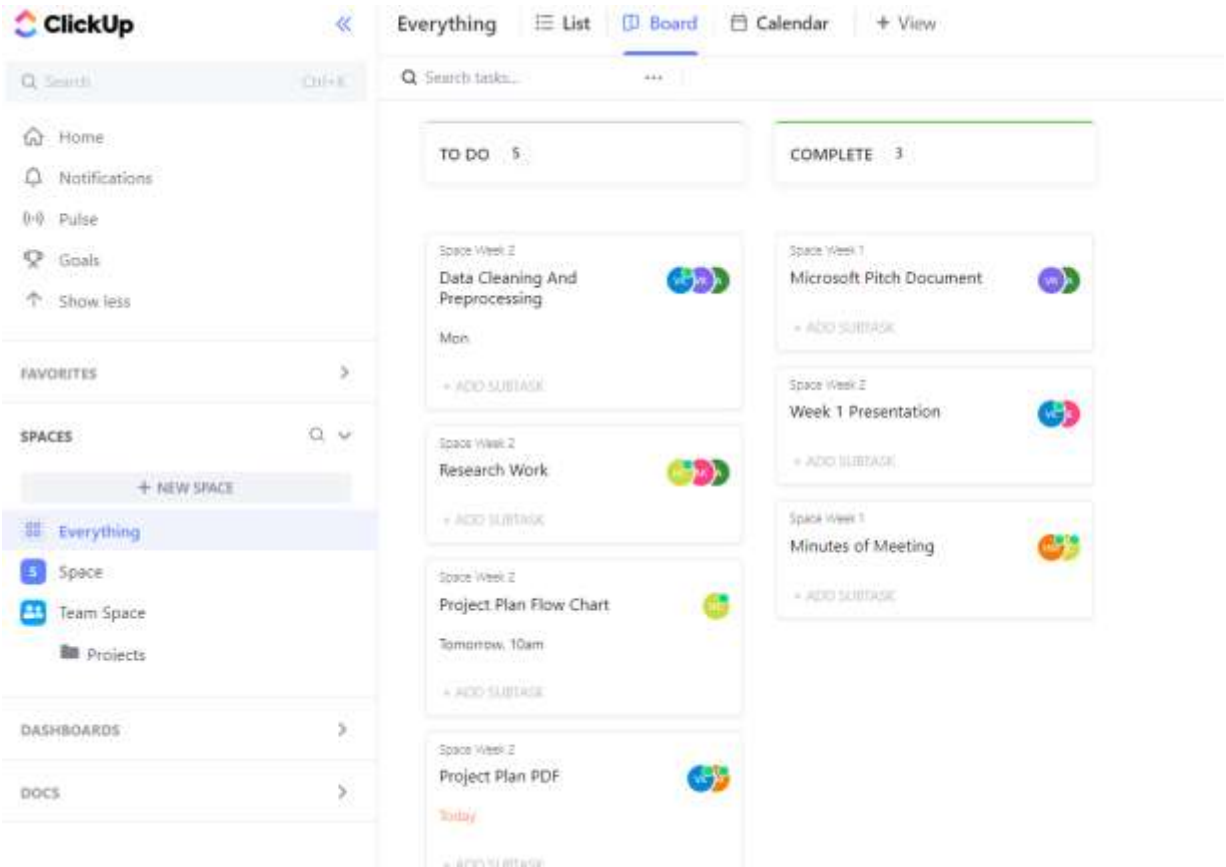
Online Google Meet Discussions:

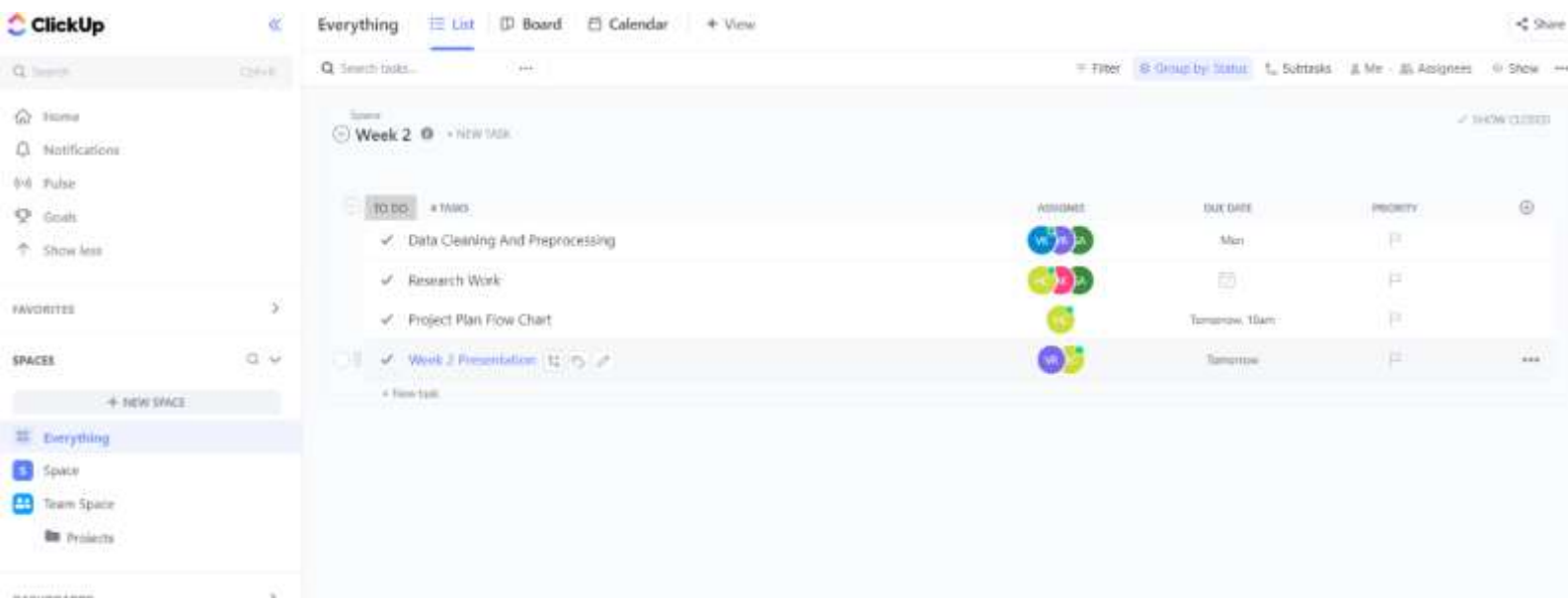
SR No	DATE	Discussion
1)	02/06/23	• Established GitHub repository for the project: The team agreed to create a GitHub repository to store all project documents and code in a centralized location. This will facilitate collaboration and version control. • Identified initial data sources and research: The team discussed potential data sources and research papers relevant to the project. Action item: Team members to share 1-2 data sources or research papers that could inform the project.

Tool for Task Management and Progress Tracking:

To manage tasks, track progress, and define sprints, we utilize tool called **ClickUp**. This tool provides a centralized platform where we can assign tasks, set deadlines, and monitor the status of each task. It enables us to track individual and team progress, ensuring that we stay on schedule and meet project milestones.

Attached is a snapshot of our task management tool:





It showcases the sprint board, task assignments, and the progress of each task. The tool allows us to easily visualize the distribution of tasks, identify bottlenecks, and ensure that everyone is accountable for their assigned responsibilities.

By leveraging Google Meet for team meetings and utilizing a dedicated task management tool, we ensure regular communication, efficient task management, and transparent progress tracking within our team. These practices enable us to stay aligned, address any challenges promptly, and maintain a structured approach towards achieving our project goals.