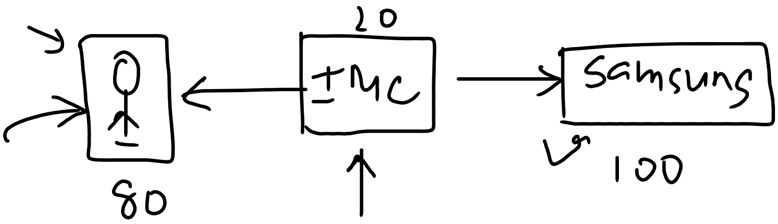
Problem Statement

26 September 2024 16:30



Business Context:

We are "Influence Boost Inc.," an influencer management company seeking to expand our network by attracting more influencers to join our platform. Due to a limited marketing budget, traditional advertising channels are not viable for us. To overcome this, we aim to offer a solution that addresses a significant pain point for influencers, thereby encouraging them to engage with our company.

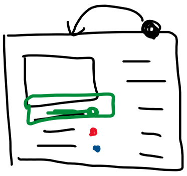
Business Problem:

1. Need to Attract More Influencers:
   - Objective: Increase our influencer clientele to enhance our service offerings to brands and stay competitive.
   - Challenge: Limited marketing budget restricts our ability to reach and engage potential influencer clients through conventional means.
2. Identifying the Influencer Pain Point:
   - Understanding Influencer Challenges: To effectively attract influencers, we need to understand and address the key challenges they face.
   - Research Insight: Influencers, especially those with large followings, struggle with managing and interpreting the vast amount of feedback they receive via comments on their content.
3. Big Influencers Face Issues with Comment Analysis:
   - Volume of Comments: High-profile influencers receive thousands of comments on their videos, making manual analysis impractical.
   - Time Constraints: Influencers often lack the time to sift through comments to extract meaningful insights.
   - Impact on Content Strategy: Without efficient comment analysis, influencers miss opportunities to understand audience sentiment, address concerns, and tailor their content effectively.

Our Solution

26 September 2024 16:30

To directly address the significant pain point faced by big influencers—managing and interpreting vast amounts of comment data—we present the "Influencer Insights" Chrome plugin. This tool is designed to empower influencers by providing in-depth analysis of their YouTube video comments, helping them make data-driven decisions to enhance their content and engagement strategies.
Key Features of the Plugin:



1. Sentiment Analysis of Comments

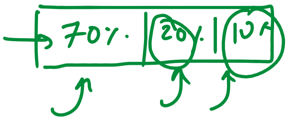
100 comment

70 → +ve

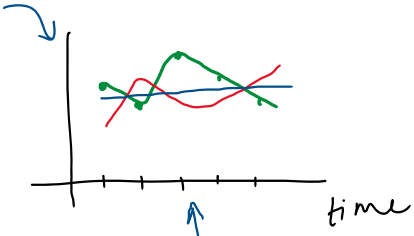
10 → -ve

20 → neutral

- Real-Time Sentiment Classification:
  - The plugin performs real-time analysis of all comments on a YouTube video, classifying each as positive, neutral, or negative.
- Sentiment Distribution Visualization:
  - Displays the overall sentiment distribution with intuitive graphs or charts (e.g., pie charts or bar graphs showing percentages like 70% positive, 20% neutral, 10% negative).



- Detailed Sentiment Insights:
  - Allows users to drill down into each sentiment category to read specific comments classified under it.



- Trend Tracking: X
  - Monitors how sentiment changes over time, helping influencers identify how different content affects audience perception.

Advanced (paid)

2. Summary of Comments (X → OpenAI API

spam

- Automated Comment Summarization:

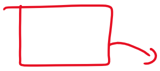
feedbc

  - Utilizes natural language processing algorithms to generate concise summaries of the most discussed topics within the comments. Deep Learning
- Highlight Key Themes:

concern

  - Identifies and summarizes common feedback, suggestions, or concerns raised by the audience.

3. Additional Comment Analysis Features



Avg line

- Word Cloud Visualization:
  - Generates a word cloud showcasing the most frequently used words and phrases in the comments.
  - Helps quickly identify trending topics, keywords, or recurring themes.

python

image

- Average Comment Length:
  - Calculates and displays the average length of comments, indicating the depth of audience engagement.
- Spam and Troll Detection: X → ml model →
  - Filters out spam, bot-generated comments, or potentially harmful content to streamline the analysis.
- Export Data Functionality: → comment export
  - Enables users to export analysis reports and visualizations in various formats (e.g., PDF, CSV) for further use or sharing with team members.

Challenges

26 September 2024 16:31

reddit

supervised ml
↓
classification

input o/p

comment +ve
-ve
neutral

37K comments
labelled

1. Data
    a. Availability ✓ → reddit
    b. Lack of general kind of dataset ✓
    c. Multi-language comments ✓
    d. Spam and bot comments ✓
    e. Slang emoji and informal comments ✓
    f. Sarcastic comments ✓
    g. Evolving language usage (e.g. sick) -> drift ✓
    h. Privacy and data compliance

2. Building a efficient model
    a. Data noise, variability, class imbalance

3. Latency

4. User Experience

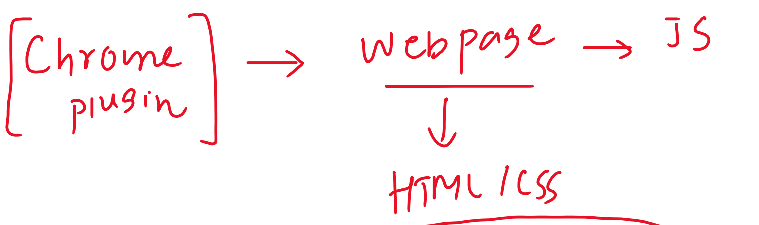
API →

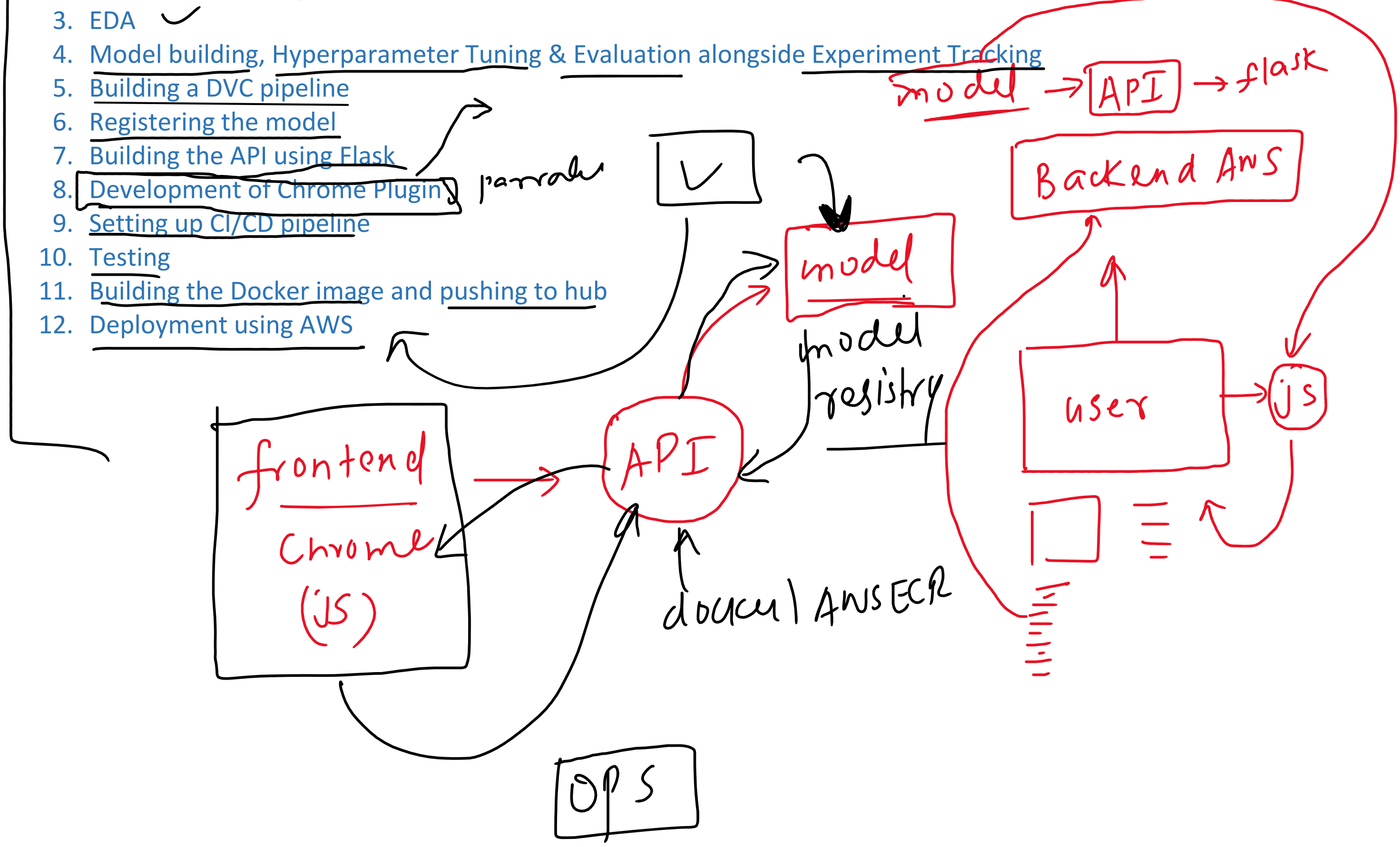
universal

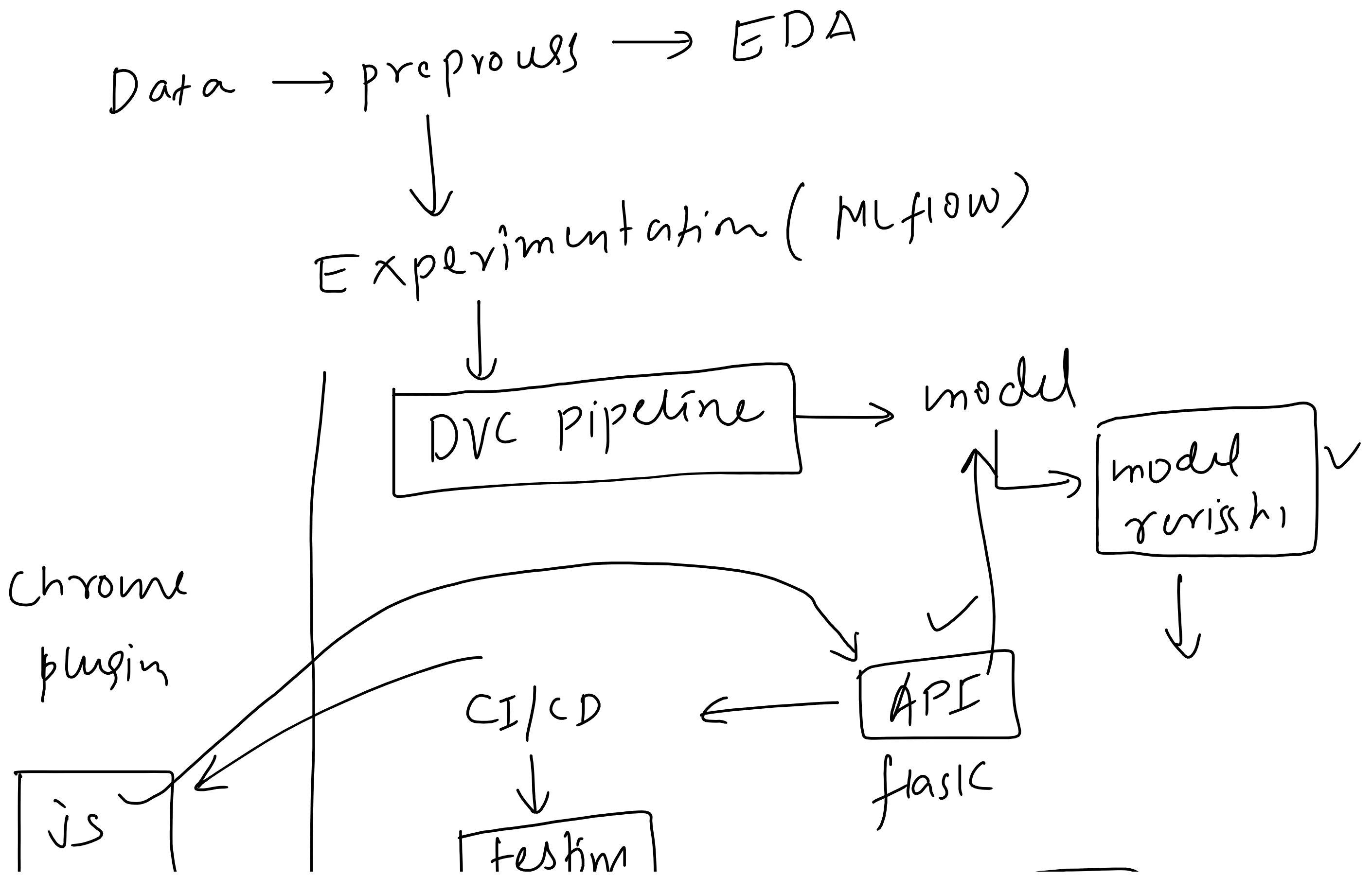
Workflow

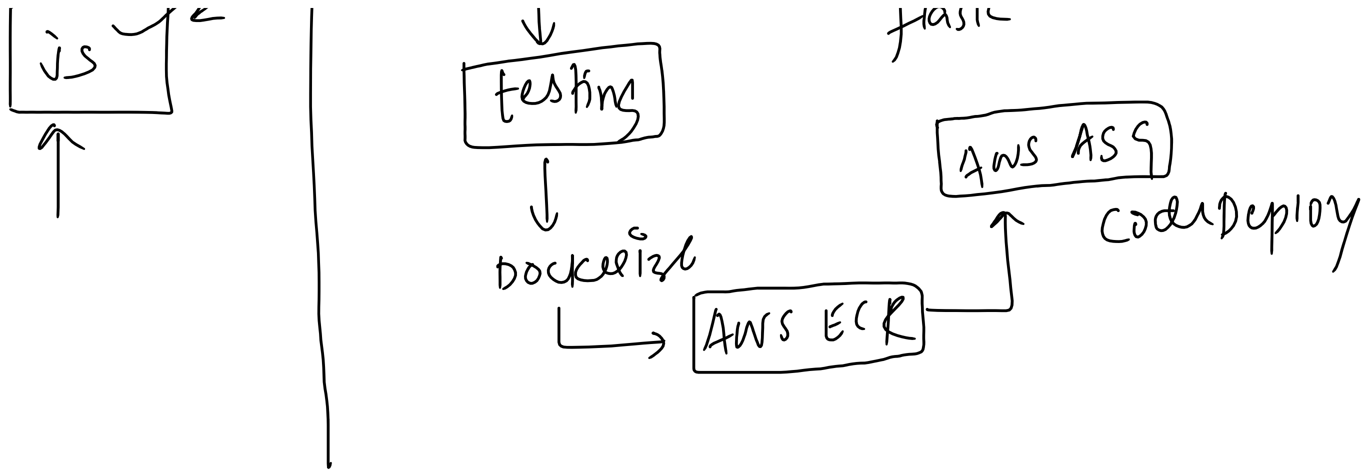
26 September 2024 16:31

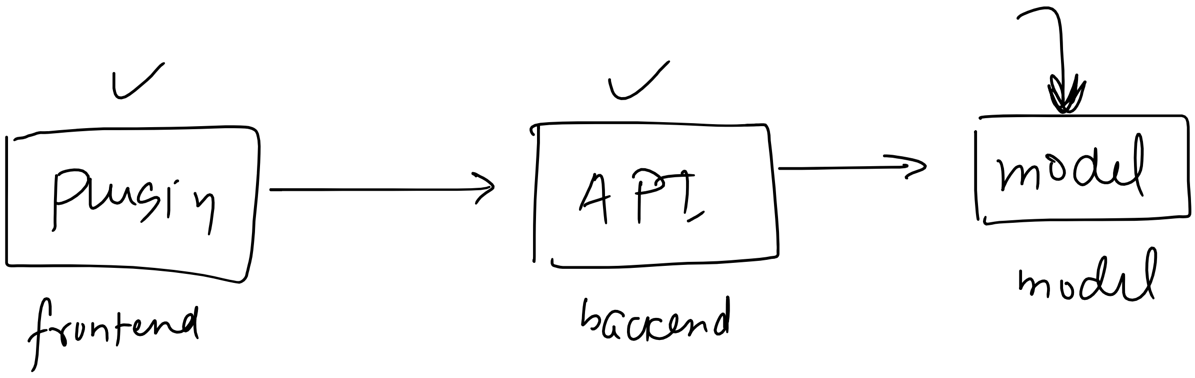
1. Data collection
2. Data Preprocessing
3. EDA
4. Model building, Hyperparameter Tuning & Evaluation alongside Experiment Tracking
5. Building a DVC pipeline
6. Registering the model
7. Building the API using Flask
8. Development of Chrome Plugin
9. Setting up CI/CD pipeline
10. Testing
11. Building the Docker image and pushing to hub
12. Deployment using AWS








js
testing
Dockerize
AWS ECR
AWS ASG
CodeDeploy
flask


✓
✓
Plugin
API
model
frontend
backend
model

Technologies

26 September 2024 16:31

[Data → prepro
↳ eda]

1. Version Control and Collaboration

- Git ✓
  - Purpose: Distributed version control system for tracking changes in source code.
  - Usage: Manage codebase, track changes, and collaborate with team members.
- GitHub ✓
  - Purpose: Hosting service for Git repositories with collaboration features.
  - Usage: Store repositories, manage issues, pull requests, and facilitate team collaboration.

2. Data Management and Versioning

- DVC (Data Version Control) ✓
  - Purpose: Version control system for tracking large datasets and machine learning models.
  - Usage: Version datasets and machine learning pipelines, enabling reproducibility and collaboration.
- AWS S3 (Simple Storage Service) ✓
  - Purpose: Scalable cloud storage service.
  - Usage: Store datasets, pre-processed data, and model artifacts tracked by DVC.

3. Machine Learning and Experiment Tracking

- Python ✓
  - Purpose: Programming language for backend development and machine learning.
  - Usage: Implement data processing scripts, machine learning models, and backend services.
- Machine Learning Libraries: ✓
  - scikit-learn ✓
    - Purpose: Library for classical machine learning algorithms.
    - Usage: Implement baseline models and preprocessing techniques.
- NLP Libraries:
  - NLTK (Natural Language Toolkit) ✓
    - Purpose: Platform for building Python programs to work with human language

data.

    - Usage: Tokenization, stemming, and other basic NLP tasks.

  - spaCy ✓
    - Purpose: Industrial-strength NLP library.
    - Usage: Advanced NLP tasks like named entity recognition, part-of-speech tagging.

- Mlflow ✓
  - Purpose: Platform for managing the ML lifecycle, including experimentation, reproducibility, deployment, and a central model registry.
  - Usage: Track experiments, log parameters, metrics, and artifacts; manage model versions.

- MLflow Model Registry ✓
  - Purpose: Component of MLflow for managing the full lifecycle of ML models.
  - Usage: Register models, manage model stages (e.g., staging, production), and collaborate on model development.

- Optuna ✓
  - For Hyperparameter tuning

4. Continuous Integration/Continuous Deployment (CI/CD)

- GitHub Actions ✓
  - Purpose: Automation platform that enables CI/CD directly from GitHub repositories.
  - Usage:
    - Automate testing, building, and deployment pipelines.
    - Trigger workflows on events like code commits or pull requests.

5. Cloud Services and Infrastructure

- AWS (Amazon Web Services)
  - AWS EC2 (Elastic Compute Cloud) ✓
    - Purpose: Scalable virtual servers in the cloud.
    - Usage: Host backend services, APIs, and model servers.
  - AWS Auto Scaling Groups ✓
    - Purpose: Automatically adjust the number of EC2 instances to handle load changes.
    - Usage:

            - Ensure that the application scales out during demand spikes to maintain performance.
            - Scale in during low demand periods to reduce costs.
            - Maintain application availability by automatically adding or replacing instances as needed.

    - AWS CodeDeploy ✓
        - Purpose: Deployment service that automates application deployments to various compute services like EC2, Lambda, and on-premises servers.
        - Usage:
            - Automate the deployment process of backend services and machine learning models to AWS EC2 instances or AWS Lambda.
            - Integrate with GitHub Actions to create a seamless CI/CD pipeline that deploys code changes automatically upon successful testing.

    - AWS CloudWatch ✓
        - Purpose: Monitoring and observability service.
        - Usage: Monitor application logs, set up alerts, and track performance metrics.

    - AWS IAM (Identity and Access Management) ✓
        - Purpose: Securely manage access to AWS services.
        - Usage: Control access permissions for users and services.

6. Programming Languages and Libraries

- Python ✓
    - Purpose: Backend development, data processing, machine learning.
    - Usage: Implement APIs, machine learning models, data pipelines.

- JavaScript ✓
    - Purpose: Frontend development, especially for web applications and browser extensions.
    - Usage: Develop the Chrome extension's user interface and functionality.

- HTML and CSS ✓
    - Purpose: Markup and styling languages for web content.
    - Usage: Structure and style the Chrome extension's interface.

- Data Processing Libraries:
    - Pandas ✓

        - Purpose: Data manipulation and analysis.
        - Usage: Handle tabular data, preprocess datasets.
    - NumPy ✓
        - Purpose: Fundamental package for scientific computing with Python.
        - Usage: Perform numerical operations, handle arrays.

7. Frontend Development Tools

- Chrome Extension APIs ✓
    - Purpose: APIs provided by Chrome for building extensions.
    - Usage: Interact with browser features, modify web page content, manage extension behaviour.
- Browser Developer Tools ✓
    - Purpose: Built-in tools for debugging and testing web applications.
    - Usage: Inspect elements, debug JavaScript, monitor network activity.
- Code Editors and IDEs:
    - Visual Studio Code ✓
        - Purpose: Source code editor.
        - Usage: Write and edit code for both frontend and backend development.

8. Testing and Quality Assurance Tools

- Testing Frameworks:
    - Pytest ✓
        - Purpose: Testing framework for Python.
        - Usage: Write and run unit tests for backend code and data processing scripts.
    - Unittest ✓
        - Purpose: Built-in Python testing framework.
        - Usage: Write unit tests for Python code.
    - Jest ✓
        - Purpose: JavaScript testing framework.
        - Usage: Write and run tests for JavaScript code in the Chrome extension.

9. Project Management and Communication

- Project Management Tools:

9. Project Management and Communication

- Project Management Tools:
  - Jira
    - Purpose: Issue and project tracking software.
    - Usage: Manage tasks, track progress, and coordinate team activities.
- Communication Tools:
  - Slack
    - Purpose: Team communication platform.
    - Usage: Facilitate real-time communication among team members.
  - Microsoft Teams
    - Purpose: Collaboration and communication platform.
    - Usage: Chat, meet, call, and collaborate in one place.

10. DevOps and MLOps Tools

- Docker
  - Purpose: Containerization platform.
  - Usage: Package applications and dependencies into containers for consistent deployment.

11. Security and Compliance

- SSL/TLS Certificates
  - Purpose: Secure communications over a computer network.
  - Usage: Encrypt data between users and backend services.

12. Monitoring and Logging

- Logging Tools:
  - AWS CloudWatch Logs
    - Purpose: Monitor, store, and access log files.
    - Usage: Collect and monitor logs from AWS resources.
- Monitoring Tools:
  - Prometheus (Optional)
    - Purpose: Open-source monitoring system.
    - Usage: Collect and store metrics, generate alerts.

- Grafana ✓
  - Purpose: Visualization and analytics software.
  - Usage: Create dashboards to visualize metrics.

13. API Development and Testing

- Frameworks:
  - Flask ✓
    - Purpose: Lightweight WSGI web application framework.
    - Usage: Build RESTful APIs for backend services.
  - FastAPI ✗
    - Purpose: Modern, fast web framework for building APIs with Python.
    - Usage: Develop high-performance APIs efficiently.
- API Testing Tools:
  - Postman ✓
    - Purpose: API development environment.
    - Usage: Design, test, and document APIs.

14. Code Quality and Documentation

- Code Linters and Formatters:
  - Pylint ✓ — ci/cd
    - Purpose: Code analysis for Python.
    - Usage: Enforce coding standards, detect code smells.
- Documentation Generation:
  - Sphinx
    - Purpose: Generate documentation from source code.
    - Usage: Create project documentation automatically.

15. Additional Tools and Libraries

- Visualization Libraries:
  - Matplotlib

        - Purpose: Plotting library for Python.
        - Usage: Create static, animated, and interactive visualizations.
    - Seaborn
        - Purpose: Statistical data visualization.
        - Usage: Generate high-level interface for drawing attractive graphics.
    - D3.js
        - Purpose: JavaScript library for producing dynamic, interactive data visualizations.
        - Usage: Create word clouds and other visual elements in the Chrome extension.
- Data Serialization Formats:
    - JSON
        - Purpose: Lightweight data interchange format.
        - Usage: Transfer data between frontend and backend services.

Plan of Action

26 September 2024 16:31