CI/CD Pipeline

CI/CD pipeline for this project is generated using MLflow, DagsHub and Huggingface Spaces. MLflow is an open source platform for managing the end-to-end machine learning lifecycle. Dagshub integrates with github and MLflow and helps us in versioning datasets & models, track experiments, label data, and visualize results and Huggingface Spaces is used to deploy the application.

As a crucial component of the ML project, the GitHub project repository is linked to DagsHub utilizing GitHub Actions for CI/CD.

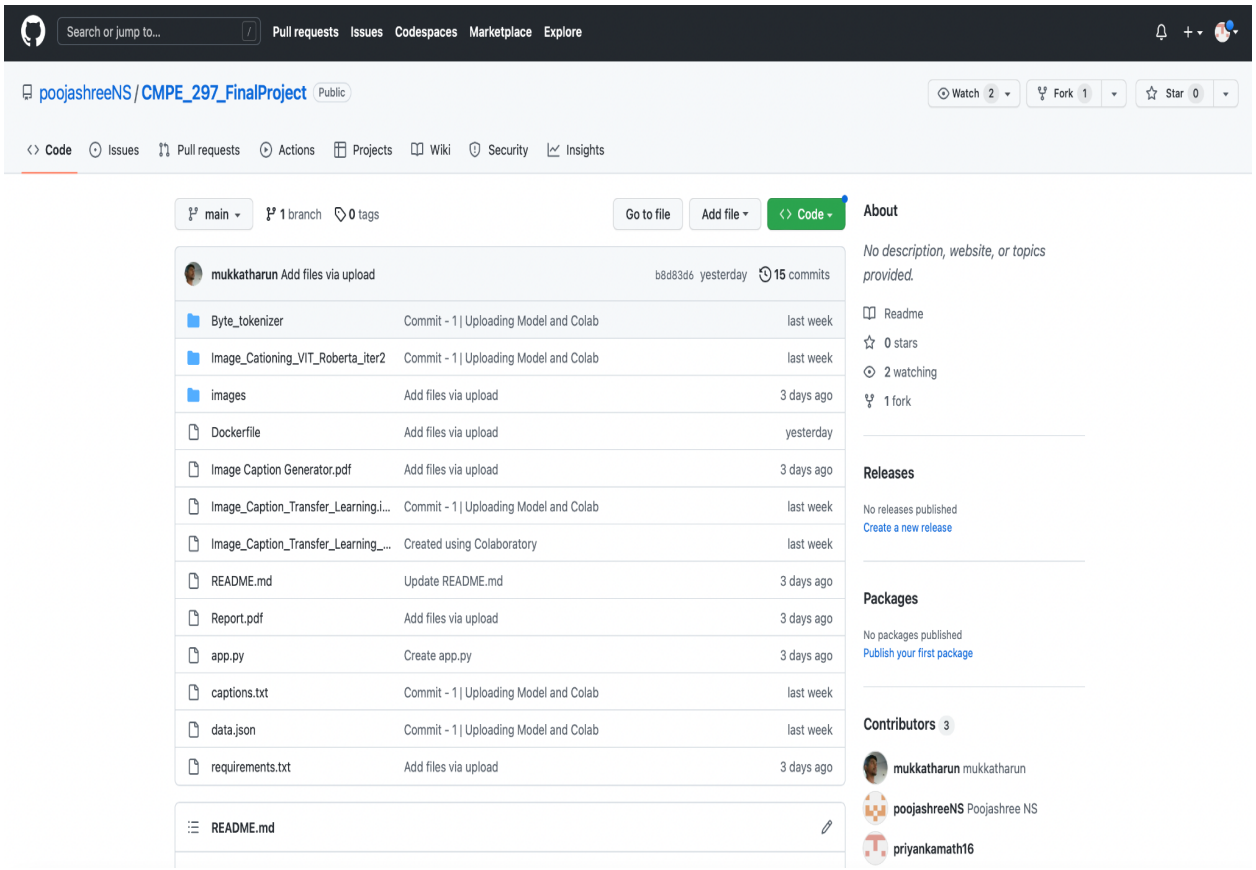

Figure 1: Project GitHub Repository Source: Author

The workflow for DagsHub's GitHub-connected repositories is much more efficient. Additionally to automatically syncing the repository on push and enabling management of PR and Issues from both platforms with the aid of GitHub Actions, DagsHub also syncs the git-tracked files.

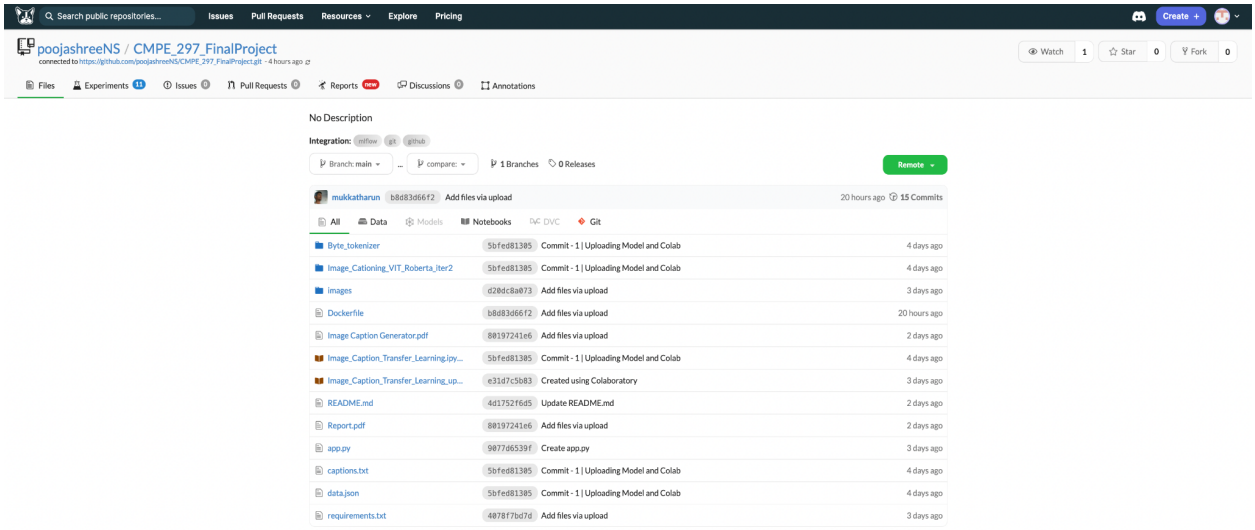

Figure 2: DagsHub workspace that is linked to the project github repository. Source: Author

When code is pushed, the GitHub connected repositories that have subscribed to the GitHub webhooks are automatically synchronized. The project is now completely configured with a remote object storage and experiment tracking server, while git-tracked files continue to reside on GitHub.

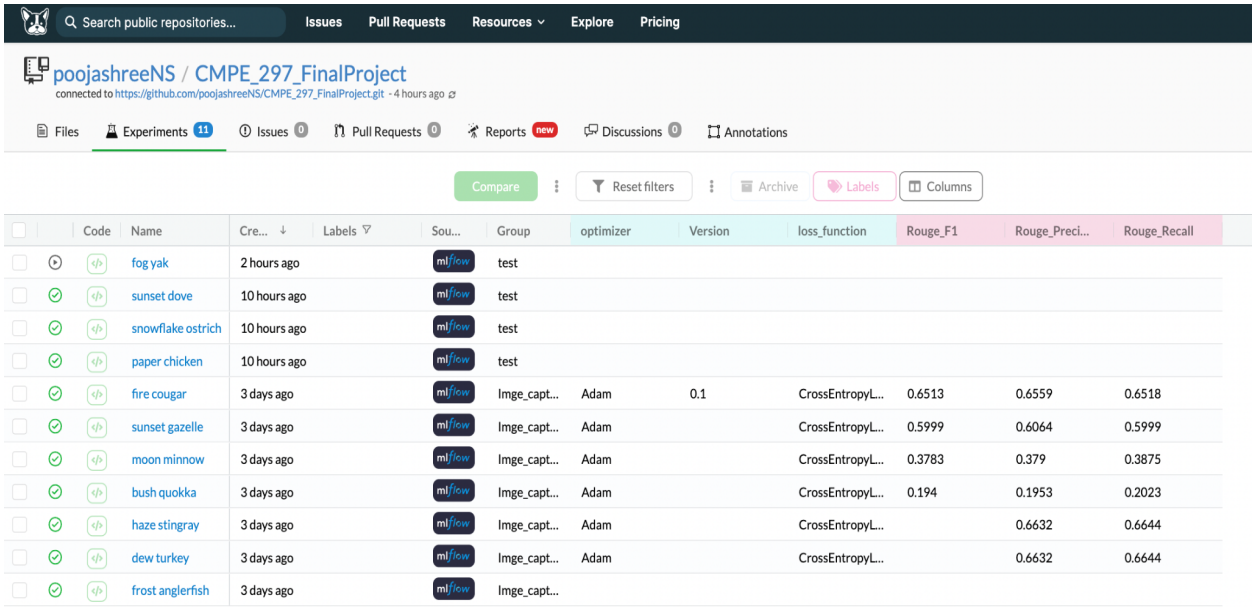

Figure 3: Pipeline of the model training. Source: Author

Four elements are offered by MLflow to assist in managing the ML workflow:

- MLflow Tracking
- MLflow Projects
- MLflow Models
- MLflow Registry

A new experiment is established each time a pipeline is triggered, tracking all the parameters, metrics, datasets, and model artifacts. The figures that depict the experiments and the information they track are provided below.

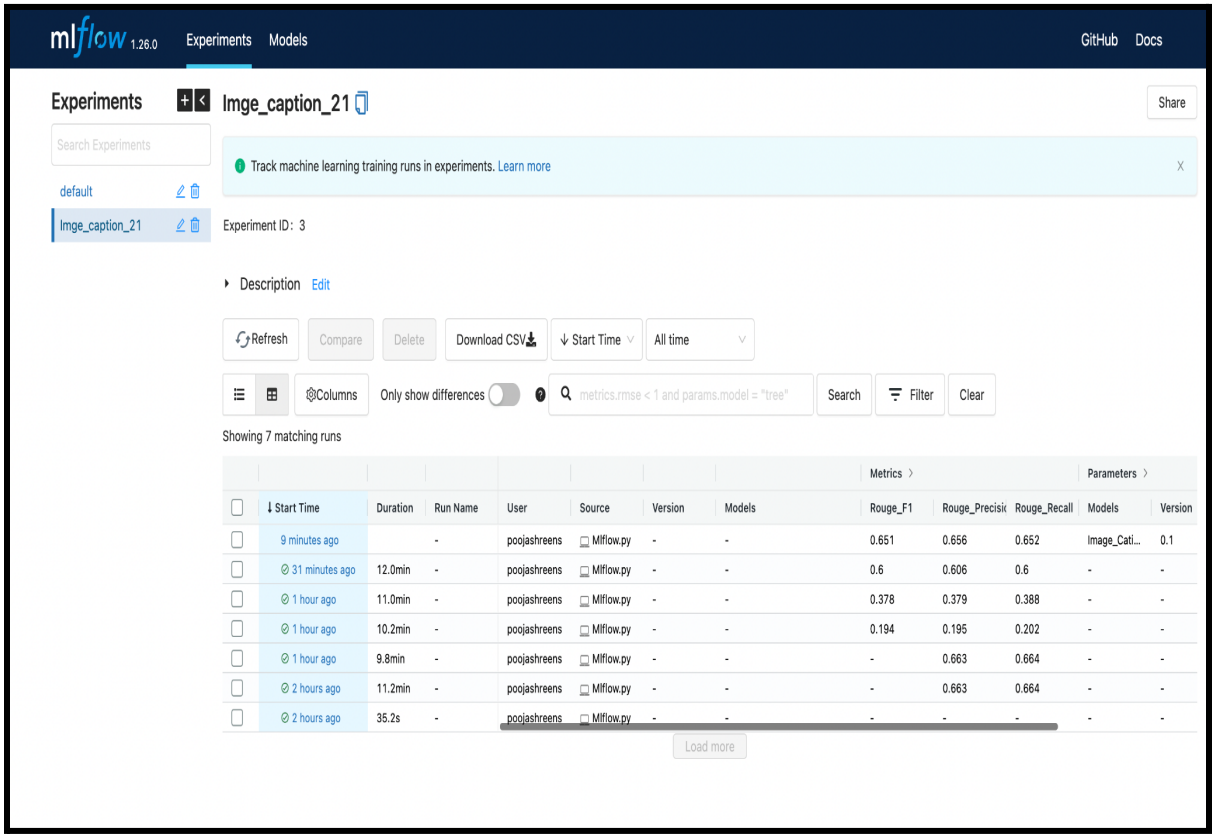

Figure 4: MLflow dashboard. Source: Author

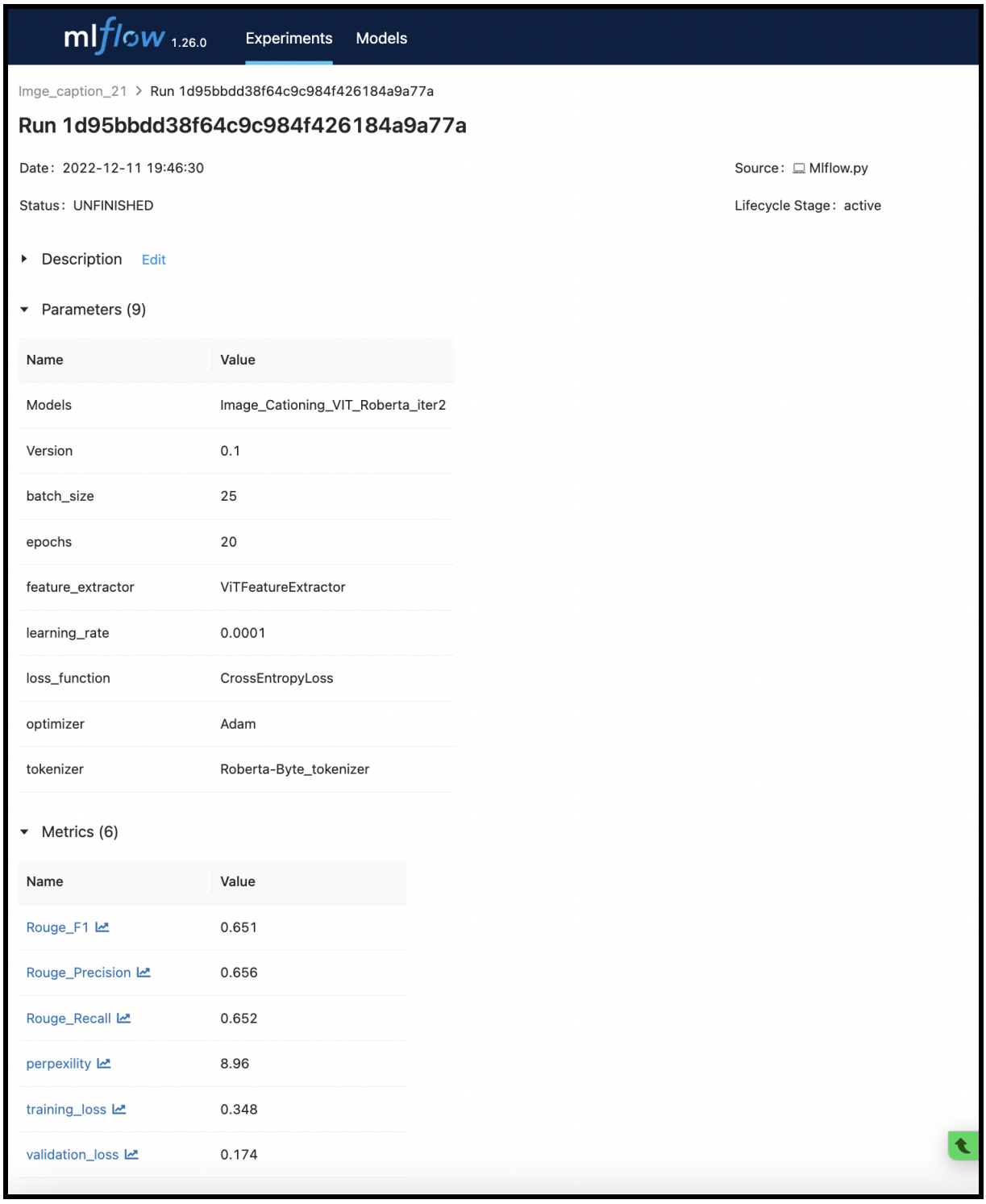

Figure 5: MLflow experimental model parameters and metrics. Source: Author

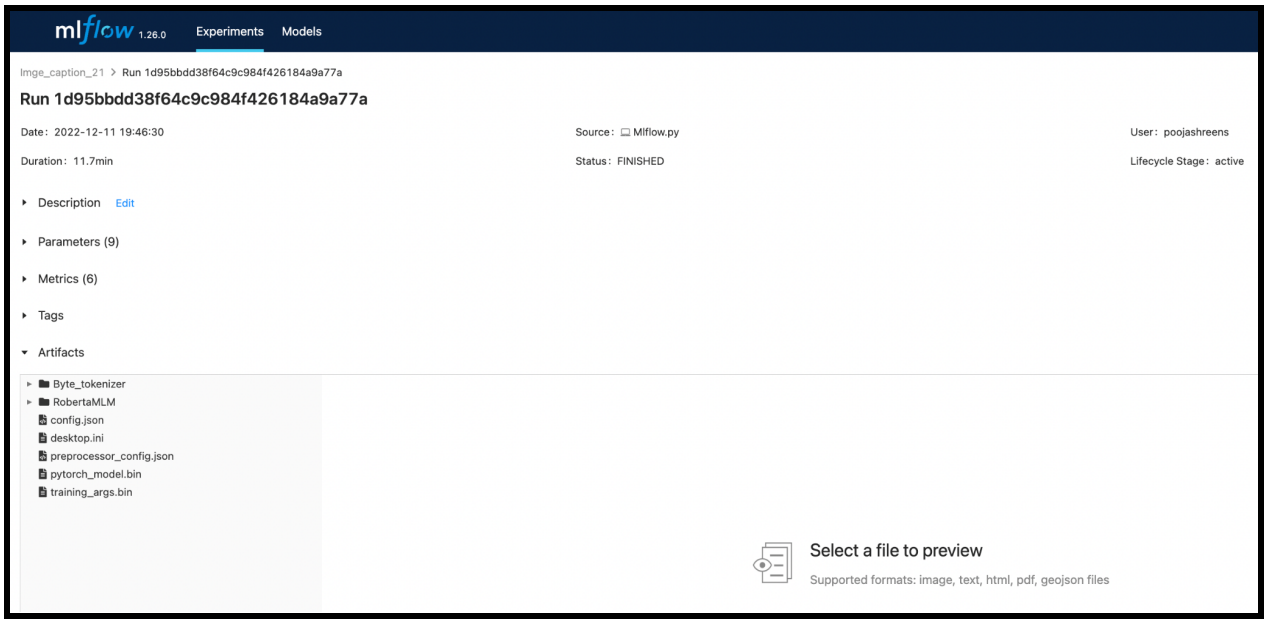

Figure 6: MLflow model artifacts. Source: Author

We conducted experiments with different values for Epoch, batch size, learning rate, etc and the resultant metrics are compared using MLFlow as shown below.

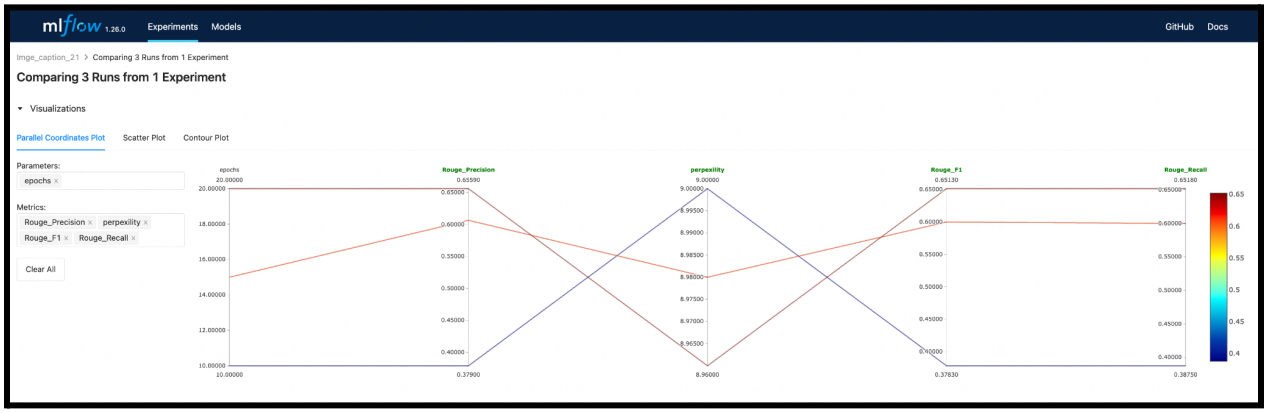

Figure 7: Comparison of experiments on trained models. Source: Author

One stage may at any time be allocated to each unique model version. For typical use-cases like Staging, Production, or Archived, MLflow offers predefined stages. A model version can be moved from one stage to another.

Figure 8: MLflow Model registry. Source: Author

The code is dockerized and deployed on hugging-face spaces once the best model has been chosen.

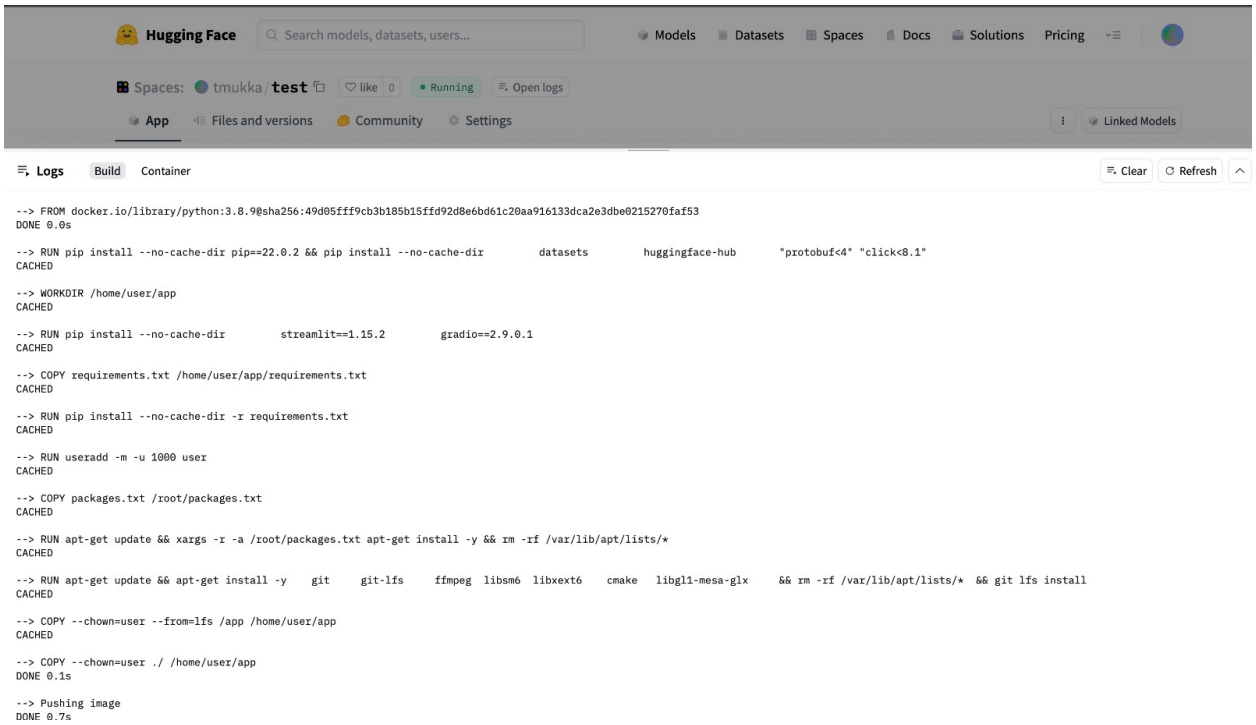

Figure 9: Deployment of dockerized image on hugging face spaces. Source: Author

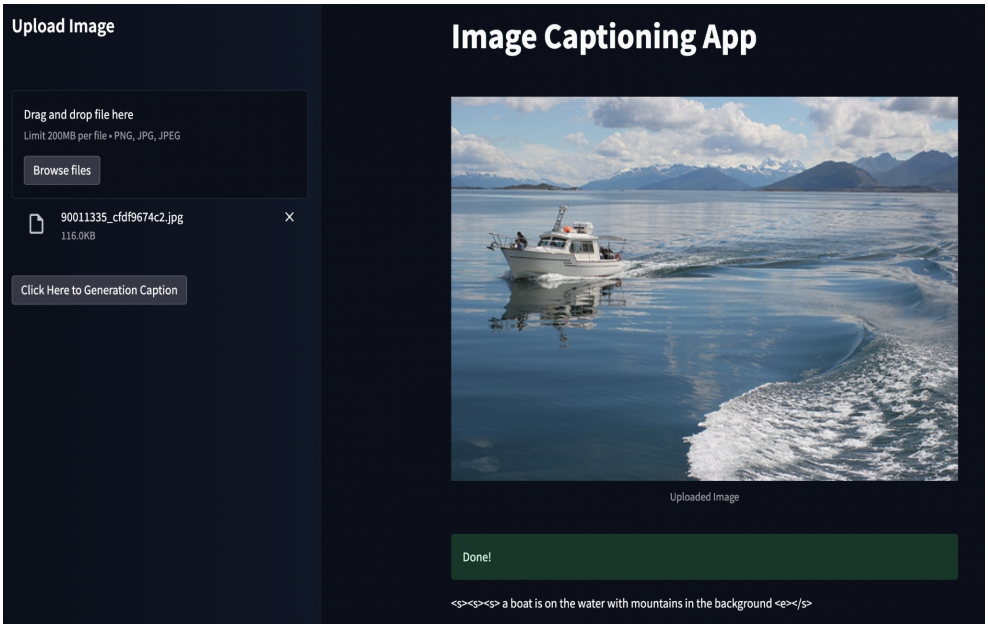

Figure 10: Deployed web application for image caption generator. Source: Author