

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.

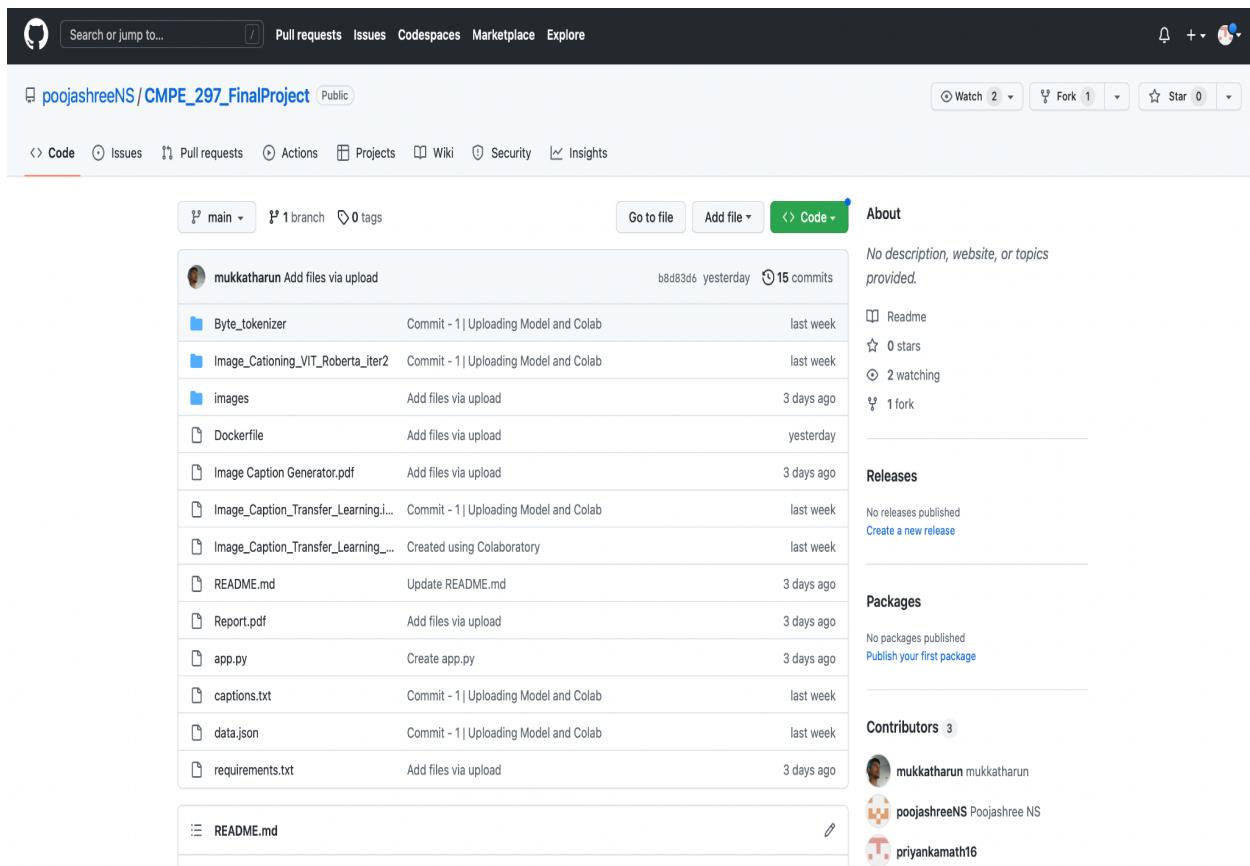


Figure 1: Project GitHub Repository Source: Author

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.

The screenshot shows a DagsHub workspace interface. At the top, there's a navigation bar with links for Issues, Pull Requests, Resources, Explore, and Pricing. Below the navigation bar, the GitHub repository details are shown: **poojashreeNS / CMPE_297_FinalProject**, connected to https://github.com/poojashreeNS/CMPE_297_FinalProject.git, updated 4 hours ago. The workspace has 1 Branch and 0 Releases. A green button labeled "Remote" is visible.

The main area displays a list of files and commits from the GitHub repository. The commits are as follows:

- mukkathan b8d83d66f2 Add files via upload 20 hours ago 15 Commits
- Byte_tokenizer 5bfed81385 Commit - 1 | Uploading Model and Colab 4 days ago
- Image_Captioning_VIT_Roberta_Iter2 5bfed81385 Commit - 1 | Uploading Model and Colab 4 days ago
- images d26dcbae73 Add files via upload 3 days ago
- Dockerfile b8d83d66f2 Add files via upload 20 hours ago
- Image_Caption_Generator.pdf 8819724166 Add files via upload 2 days ago
- Image_Caption_Transfer_Learning.ipynb 5bfed81385 Commit - 1 | Uploading Model and Colab 4 days ago
- Image_Caption_Transfer_Learning.ipynb e31d7c5b83 Created using Colaboratory 3 days ago
- README.md 4d1752f665 Update README.md 2 days ago
- Report.pdf 8819724166 Add files via upload 2 days ago
- app.py 98776539f Create app.py 3 days ago
- captions.txt 5bfed81385 Commit - 1 | Uploading Model and Colab 4 days ago
- data.json 5bfed81385 Commit - 1 | Uploading Model and Colab 4 days ago
- requirements.txt 487877bd7d Add files via upload 3 days ago

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.

The screenshot shows a DagsHub workspace interface. At the top, there's a navigation bar with links for Issues, Pull Requests, Resources, Explore, and Pricing. Below the navigation bar, the GitHub repository details are shown: **poojashreeNS / CMPE_297_FinalProject**, connected to https://github.com/poojashreeNS/CMPE_297_FinalProject.git, updated 4 hours ago. The workspace has 11 Experiments, 0 Issues, 0 Pull Requests, 0 Reports, 0 Discussions, and 0 Annotations.

The main area displays a table titled "Compare" showing the pipeline of model training. The columns are: Code, Name, Created, Labels, Sou..., Group, optimizer, Version, loss_function, Rouge_F1, Rouge_Preci..., and Rouge_Recall. The data rows are:

Code	Name	Created	Labels	Sou...	Group	optimizer	Version	loss_function	Rouge_F1	Rouge_Preci...	Rouge_Recall
	fog yak	2 hours ago			test						
	sunset dove	10 hours ago			test						
	snowflake ostrich	10 hours ago			test						
	paper chicken	10 hours ago			test						
	fire cougar	3 days ago		Imge_capt...	Adam	0.1	CrossEntropyL...	0.6513	0.6559	0.6518	
	sunset gazelle	3 days ago		Imge_capt...	Adam		CrossEntropyL...	0.5999	0.6064	0.5999	
	moon minnow	3 days ago		Imge_capt...	Adam		CrossEntropyL...	0.3783	0.379	0.3875	
	bush quokka	3 days ago		Imge_capt...	Adam		CrossEntropyL...	0.194	0.1953	0.2023	
	haze stingray	3 days ago		Imge_capt...	Adam		CrossEntropyL...		0.6632	0.6644	
	dew turkey	3 days ago		Imge_capt...	Adam		CrossEntropyL...		0.6632	0.6644	
	frost anglerfish	3 days ago		Imge_capt...							

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

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

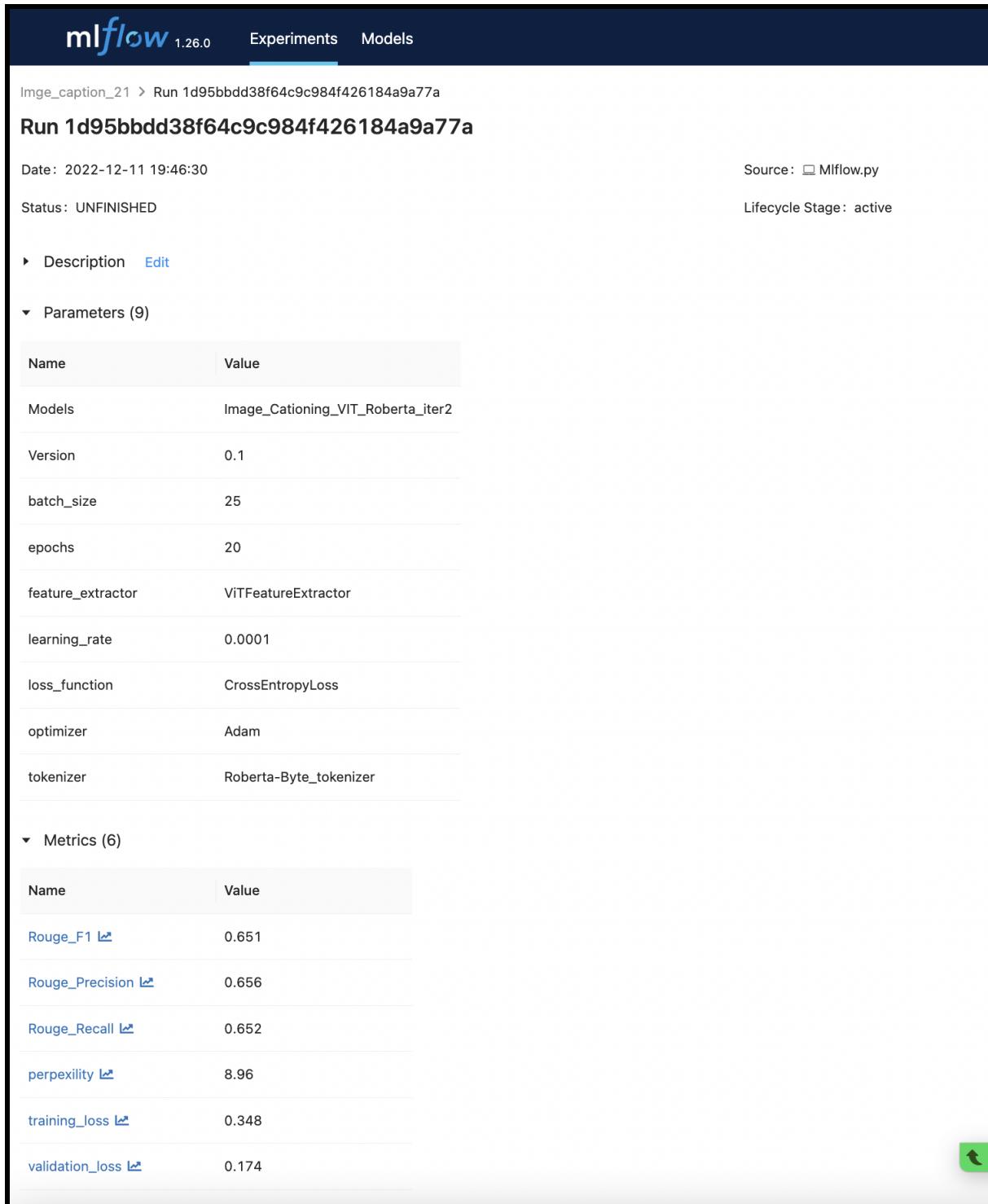
- MLflow Tracking
- MLflow Projects
- MLflow Models
- MLflow Registry

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.

The screenshot shows the MLflow dashboard interface. At the top, there's a navigation bar with the MLflow logo, version 1.26.0, and links for 'Experiments' and 'Models'. On the right side of the header are 'GitHub' and 'Docs' links. Below the header, the main content area is titled 'Experiments' and shows a single experiment named 'Image_caption_21'. A sub-header indicates 'Track machine learning training runs in experiments. Learn more'. The experiment ID is listed as 3. There are buttons for 'Edit' and 'Delete'. Below these are standard dashboard controls: 'Refresh', 'Compare', 'Delete', 'Download CSV', a date range selector ('Start Time' dropdown set to 'All time'), and a search/filter bar with a query 'metrics.rmse < 1 and params.model = "tree"'. The results table is titled 'Showing 7 matching runs' and includes columns for Start Time, Duration, Run Name, User, Source, Version, Models, and various metrics like Rouge_F1, Rouge_Precisit, Rouge_Recall, and Model versions. The last run listed is from 2 hours ago. A 'Load more' button is at the bottom of the table.

Start Time	Duration	Run Name	User	Source	Version	Models	Rouge_F1	Rouge_Precisit	Rouge_Recall	Model	Version
9 minutes ago	-	poojashreens	MLflow.py	-	-	-	0.651	0.656	0.652	Image_Cati...	0.1
31 minutes ago	12.0min	-	poojashreens	MLflow.py	-	-	0.6	0.606	0.6	-	-
1 hour ago	11.0min	-	poojashreens	MLflow.py	-	-	0.378	0.379	0.388	-	-
1 hour ago	10.2min	-	poojashreens	MLflow.py	-	-	0.194	0.195	0.202	-	-
1 hour ago	9.8min	-	poojashreens	MLflow.py	-	-	-	0.663	0.664	-	-
2 hours ago	11.2min	-	poojashreens	MLflow.py	-	-	-	0.663	0.664	-	-
2 hours ago	35.2s	-	poojashreens	MLflow.py	-	-	-	-	-	-	-

Figure 4: MLflow dashboard. Source: Author



The screenshot shows the MLflow interface for a specific experiment run. At the top, the navigation bar includes the MLflow logo (1.26.0), Experiments, and Models. Below the header, the breadcrumb navigation shows 'Image_caption_21 > Run 1d95bbdd38f64c9c984f426184a9a77a'. The main title is 'Run 1d95bbdd38f64c9c984f426184a9a77a'. Key metadata is listed: Date: 2022-12-11 19:46:30, Source: Mlflow.py, Status: UNFINISHED, and Lifecycle Stage: active.

The interface is organized into sections:

- Description**: Editable description.
- Parameters (9)**: A table showing parameter names and values:

Name	Value
Models	Image_Captioning_ViT_Roberta_iter2
Version	0.1
batch_size	25
epochs	20
feature_extractor	ViTFeatureExtractor
learning_rate	0.0001
loss_function	CrossEntropyLoss
optimizer	Adam
tokenizer	Roberta-Byte_tokenizer
- Metrics (6)**: A table showing metric names and values:

Name	Value
Rouge_F1 ↗	0.651
Rouge_Precision ↗	0.656
Rouge_Recall ↗	0.652
perplexity ↗	8.96
training_loss ↗	0.348
validation_loss ↗	0.174

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

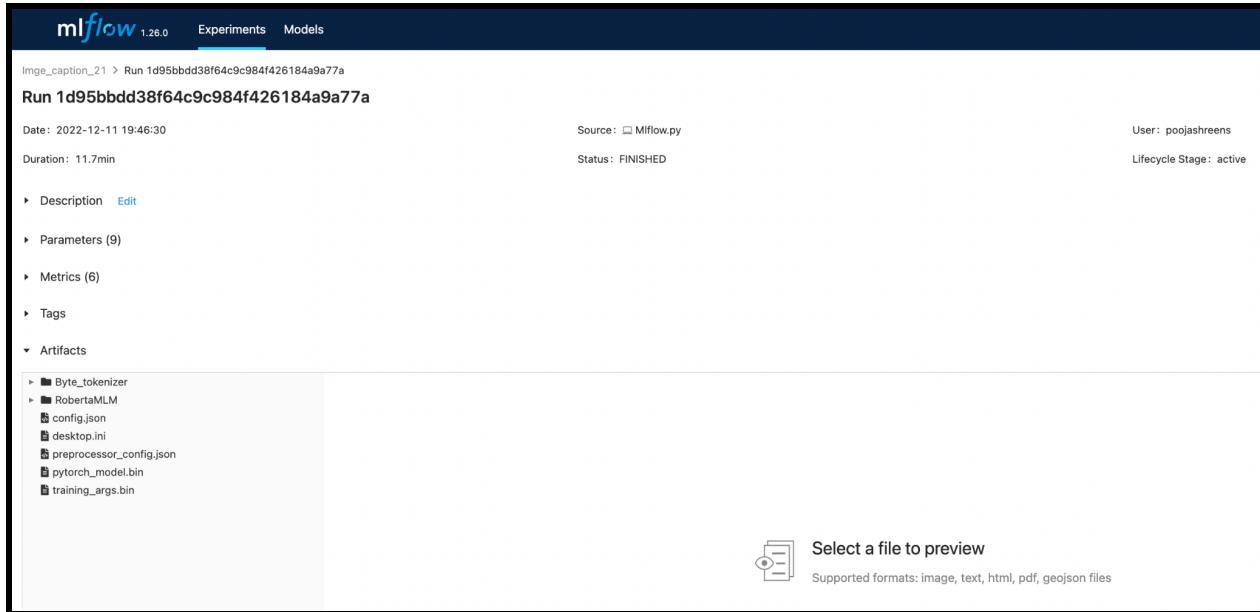


Figure 6: MLflow model artifacts. Source: Author

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

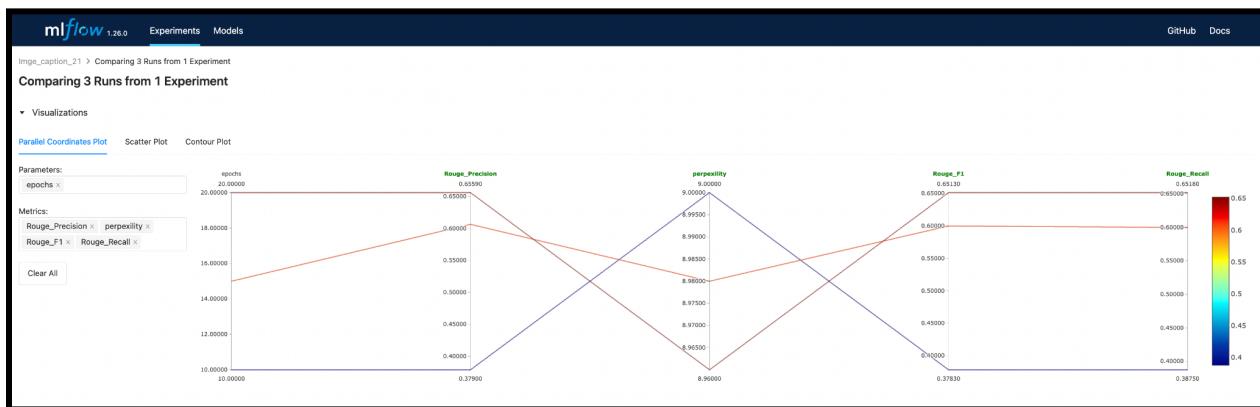


Figure 7: Comparison of experiments on trained models. Source: Author

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.

mlflow 1.26.0 Experiments Models

Registered Models > ImageCaptionRegistry

ImageCaptionRegistry

Created Time: 2022-12-14 08:57:50 Last Modified: 2022-12-14 09:03:35

- Description [Edit](#)
- Tags

[Versions](#) All Active 1 Compare

Version	Registered at	Created by	Stage
Version 1	2022-12-14 08:57:51		Production

Figure 8: MLflow Model registry. Source: Author

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

Hugging Face Search models, datasets, users...

Spaces: tmukka/test

App Files and versions Community Settings

Logs Build Container

```
--> FROM docker.io/library/python:3.8.9@sha256:49d05fff9cb3b185b15ffd92d8e6bd61c20aa916133dca2e3dbe0215270faf53
DONE 0.0s
--> RUN pip install --no-cache-dir pip==22.0.2 && pip install --no-cache-dir datasets huggingface-hub "protobuf<4" "click<8.1"
CACHED
--> WORKDIR /home/user/app
CACHED
--> RUN pip install --no-cache-dir streamlit==1.15.2 gradio==2.9.0.1
CACHED
--> COPY requirements.txt /home/user/app/requirements.txt
CACHED
--> RUN pip install --no-cache-dir -r requirements.txt
CACHED
--> RUN useradd -m -u 1000 user
CACHED
--> COPY packages.txt /root/packages.txt
CACHED
--> RUN apt-get update && xargs -r -a /root/packages.txt apt-get install -y && rm -rf /var/lib/apt/lists/*
CACHED
--> RUN apt-get update && apt-get install -y git git-lfs ffmpeg libsm6 libxext6 cmake libgl1-mesa-glx && rm -rf /var/lib/apt/lists/* && git lfs install
CACHED
--> COPY --chown=user --from=lfs /app /home/user/app
CACHED
--> COPY --chown=user ./ /home/user/app
DONE 0.1s
--> Pushing image
DONE 0.7s
```

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

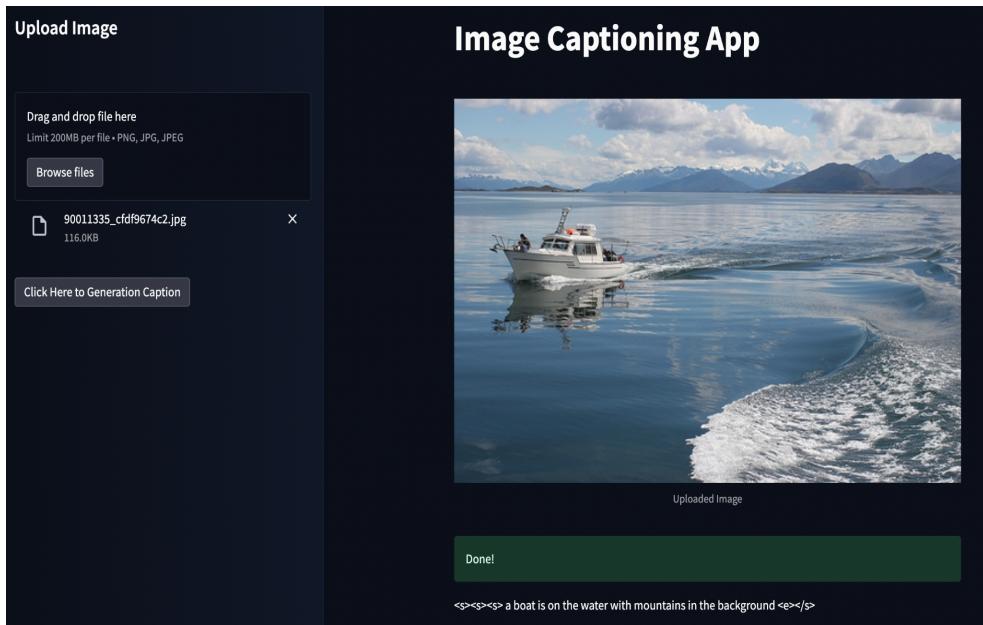


Figure 10: Deployed web application for image caption generator. Source: Author