

## Final Project Setup Guide - Group 28

This README details the setup and deployment of our chatbot movie recommender.

### Download, Extract, and Install Prerequisite Files:

**Ngrok:** <https://ngrok.com/download>

**Final\_Project\_28.zip**

**Please download the final file “tfidf\_matrix\_array.npy” from this link, it was too big to add to Brightspace:**  
<https://drive.google.com/file/d/1NMcQYtGKmUaqWdfdDIiZykyUpvvl0Twb/view?usp=sharing>

IDE Used: Visual Studio (<https://visualstudio.microsoft.com/>)

- Open Pyth\_flask.ipynb file and Run it on the Visual Studio - The file contains python flask integration with the Dialog Flow.
- You might have downloaded
- CombineDataSetLargeGenreIntegrated.csv
- tfidfmodel.pkl
- tfidf\_gen.pkl
- tfidf\_matrix\_array.npy
- tfidf\_matrix\_array\_gen.npy
- We request you keep the files in one folder and provide the path for the following file in the code.

```
def getDataframe():  
    path = 'D:/Movie recommendation/MovieSummaries/CombineDataSetLargeGenreIntegrated.csv'
```

```
def gettfidf():  
    with open('D:/Movie recommendation/MovieSummaries/tfidfmodel.pkl', 'rb') as f:
```

```
def gettfidf_gen():  
    with open('D:/Movie recommendation/tfidf_gen.pkl', 'rb') as f:
```

```
def getTfidf_matrix():  
    tfidf_matrix_array = np.load('D:/Movie recommendation/MovieSummaries/tfidf_matrix_array.npy')
```

```
def getTfidf_matrix_gen():  
    tfidf_matrix_array = np.load('D:/Movie recommendation/tfidf_matrix_array_gen.npy')
```

- After running all the cells, the file will display an IP link in this manner: <http://127.0.0.1:5000/> (don't use this link, just an example)
- After you get the link, click on the link and confirm if a message is being displayed as 'Hello World. I am finally connected to the web.' will be displayed on the browser.
- If you get this message displayed it means the flask is connected with the local server.
- After this you need to download the ngrok and run the ngrok.exe file
- In the ngrok terminal write the following command: ngrok.exe http 5000
- This will deploy it to online server
- After running the above command you will see the status as online as shown below in the image

```
ngrok (Ctrl+C to quit)
Announcing ngrok-rs: The ngrok agent as a Rust crate: https://ngrok.com/rust

Session Status      online
Account             Harshit (Plan: Free)
Version             3.2.2
Region              United States (us)
Latency              41ms
Web Interface        http://127.0.0.1:4040
Forwarding           https://7d37-2607-fea8-bda4-d800-818-9b7f-61c3-a0b2.ngrok.io -> http://localhost:5000

Connections          ttl    opn    rt1    rt5    p50    p90
                    27     0      0.00   0.00   3.84   4.85

HTTP Requests
-----
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK
```

- In the above Image it is evident that a forwarding link will generate, copy the https forwarding link.
- To check if it is being deployed online, paste the forwarding link to the browser.
- **\*IMPORTANT\*** You need to authenticate your ngrok token after signing up in the ngrok server else it will display an error.
- Login into your ngrok account and confirm your auth token by allowing ngrok to send you a mail.
- Open the Dialogflow with the link provided below and in the fulfillment section enable the webhook and paste the forwarding link you copied earlier under the URL section.
- **\*IMPORTANT\*** After pasting the forwarding link add ‘ /webhook’ in front of it as it is necessary to detect the webhook integration in our flask code of Pyth\_flask.ipynb file.
- SAVE IT

## Fulfillment

### Webhook

ENABLED 

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

URL\* <https://7d37-2607-fea8-bda4-d800-818-9b7f-61c3-a0b2.ngrok.io/webhook>

- 
- Finally our Dialogflow is now integrated with the local server.
- Make sure the last cell in the Pyth\_flask.ipynb is running, making the server active.
- The model is already being deployed in the code of Pyth\_flask.ipynb file through ‘model.pkl’ file
- **\*IMPORTANT\*** Make sure you have all the files in one folder including the ‘model.pkl’ file and the Pyth\_flask.ipynb file before running it.
- Go into the chat section of DialogFlow and provide the summary/ plot of the movie you might want to watch in the Dialogflow.
- The dialog flow will provide the recommended movie and the suggested genre.

**Group\_28\_Final\_project.ipynb**- The file consisting the code of the models

