Cloud and API Deployment

Name: SONIYA SUNNY

Batch Code: LISUM10: 30

Submission Date: June 30, 2022

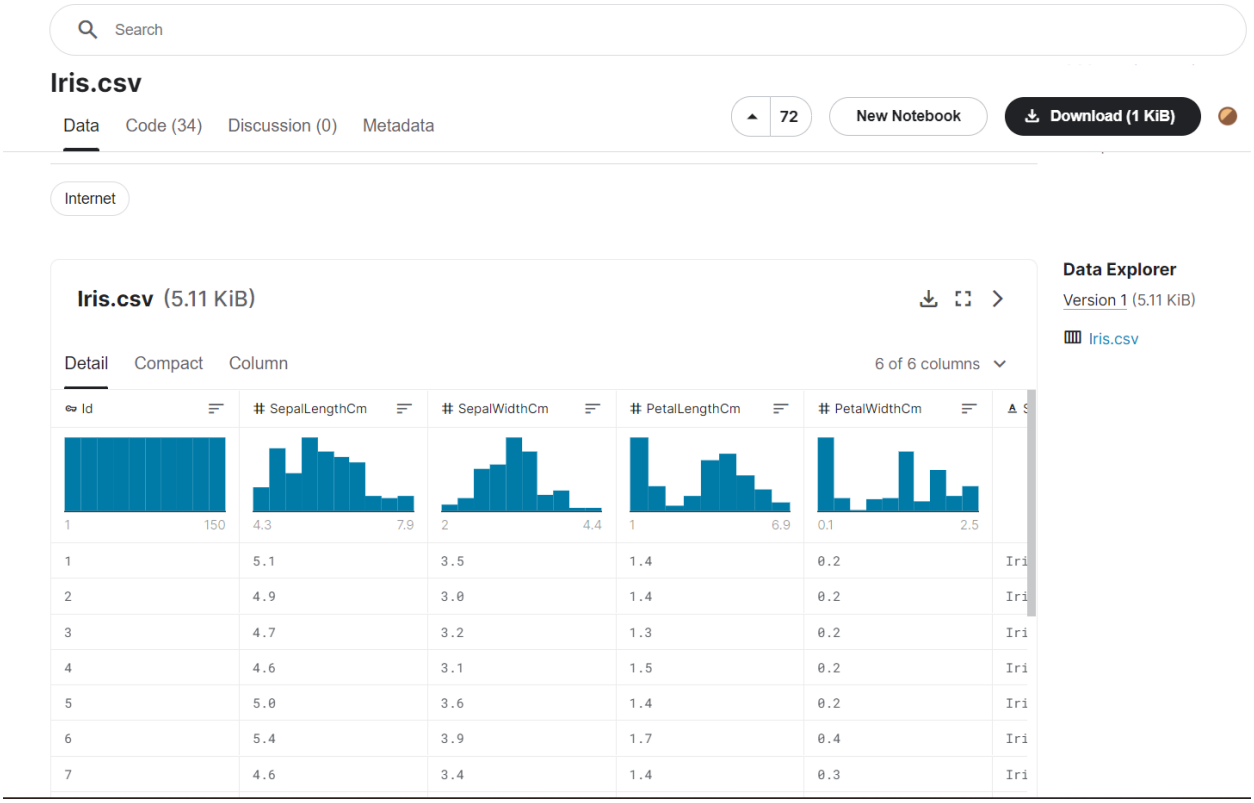
Submitted to: Data Glacier

Steps:

1. ML Model Training: -

Data Collection

Selected the famous Iris Dataset, from Kaggle website.



Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	
1	5.1	3.5	1.4	0.2	Iri
2	4.9	3.0	1.4	0.2	Iri
3	4.7	3.2	1.3	0.2	Iri
4	4.6	3.1	1.5	0.2	Iri
5	5.0	3.6	1.4	0.2	Iri
6	5.4	3.9	1.7	0.4	Iri
7	4.6	3.4	1.4	0.3	Iri

Model Building

```python
# Importing necessary libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import pickle

# Reading the data
iris = pd.read_csv("Iris.csv")
print(iris.head())
iris.drop("Id", axis=1, inplace = True)
y = iris['Species']
iris.drop(columns='Species',inplace=True)
X = iris[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]

# Training the model
x_train,x_test,y_train,y_test = train_test_split(X, y, test_size=0.3)
model = LogisticRegression()
model.fit(x_train,y_train)

#saving model to disk
pickle.dump(model, open('model.pkl','wb'))

#loading model to compare result
model = pickle.load(open('model.pkl','rb'))
print(model.predict([[5.1,3.5,1.4,0.2]]))
```

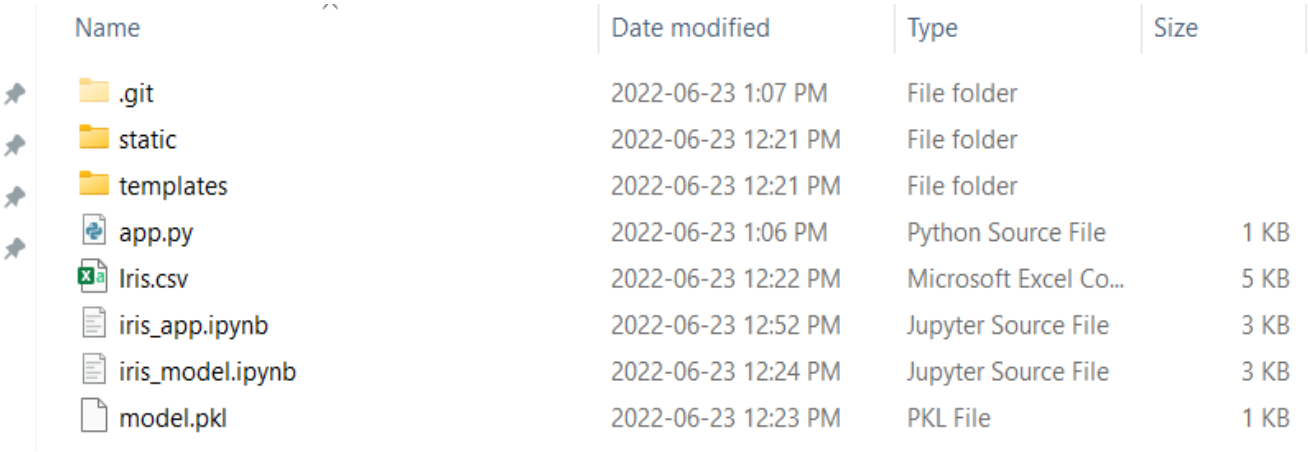
✓ 0.8s

```
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      Species
0   1            5.1           3.5            1.4           0.2  Iris-setosa
1   2            4.9           3.0            1.4           0.2  Iris-setosa
2   3            4.7           3.2            1.3           0.2  Iris-setosa
3   4            4.6           3.1            1.5           0.2  Iris-setosa
4   5            5.0           3.6            1.4           0.2  Iris-setosa
['Iris-setosa']
```

Built a Logistic Regression model and saved the model as a pickle file in the same directory.

2. Creating web app using Flask: -

Added more folders



Name	Date modified	Type	Size
.git	2022-06-23 1:07 PM	File folder	
static	2022-06-23 12:21 PM	File folder	
templates	2022-06-23 12:21 PM	File folder	
app.py	2022-06-23 1:06 PM	Python Source File	1 KB
Iris.csv	2022-06-23 12:22 PM	Microsoft Excel Co...	5 KB
iris_app.ipynb	2022-06-23 12:52 PM	Jupyter Source File	3 KB
iris_model.ipynb	2022-06-23 12:24 PM	Jupyter Source File	3 KB
model.pkl	2022-06-23 12:23 PM	PKL File	1 KB

Added static folder and templates folder.

Added index.html file in templates folder

Added style.css file in static folder

Created app.py file

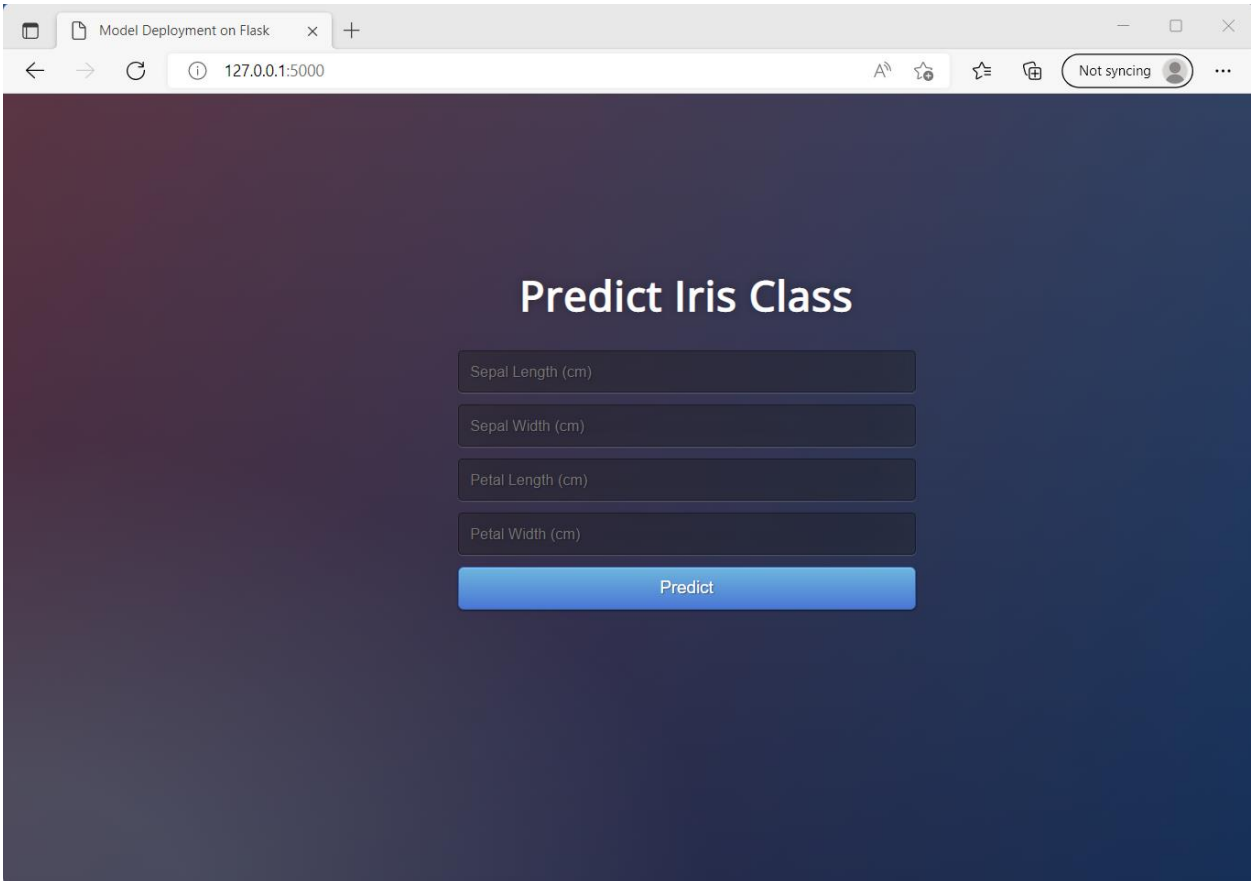
Created iris_app.ipynb file for sample execution

```python
# importing necessary libraries and functions
import numpy as np
from flask import Flask, request, render_template
import pickle

app = Flask(__name__) #Initialize the flask App
model = pickle.load(open('model.pkl', 'rb')) # loading the trained model

@app.route('/') # Homepage
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
    '''For rendering results on HTML GUI'''

    # retrieving values from form
    init_features = [float(x) for x in request.form.values()]
    final_features = [np.array(init_features)]
    prediction = model.predict(final_features) # making prediction

    return render_template('index.html', prediction_text='Predicted Class: {}'.format(prediction))

if __name__ == "__main__":
    app.run()
```

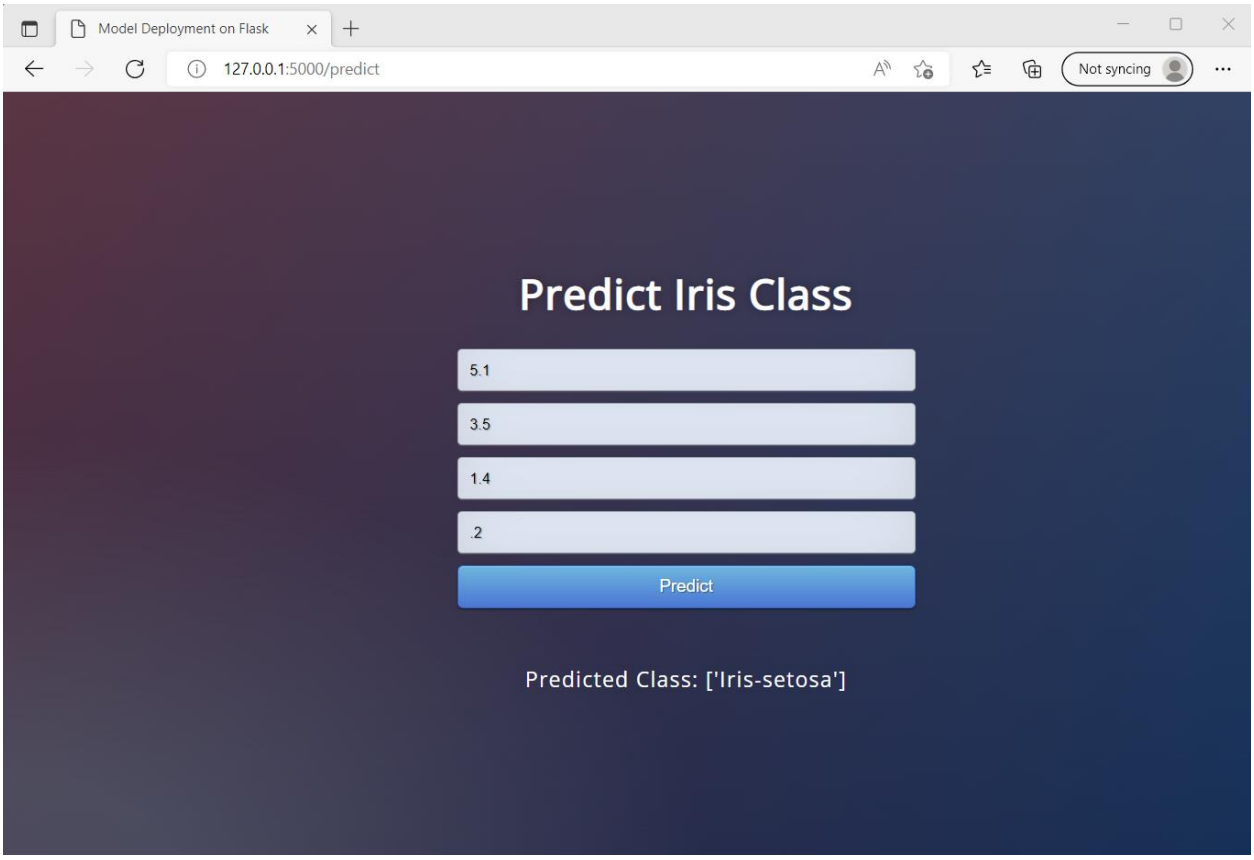
✓ 3.9s

```
* Serving Flask app "__main__" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off

* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

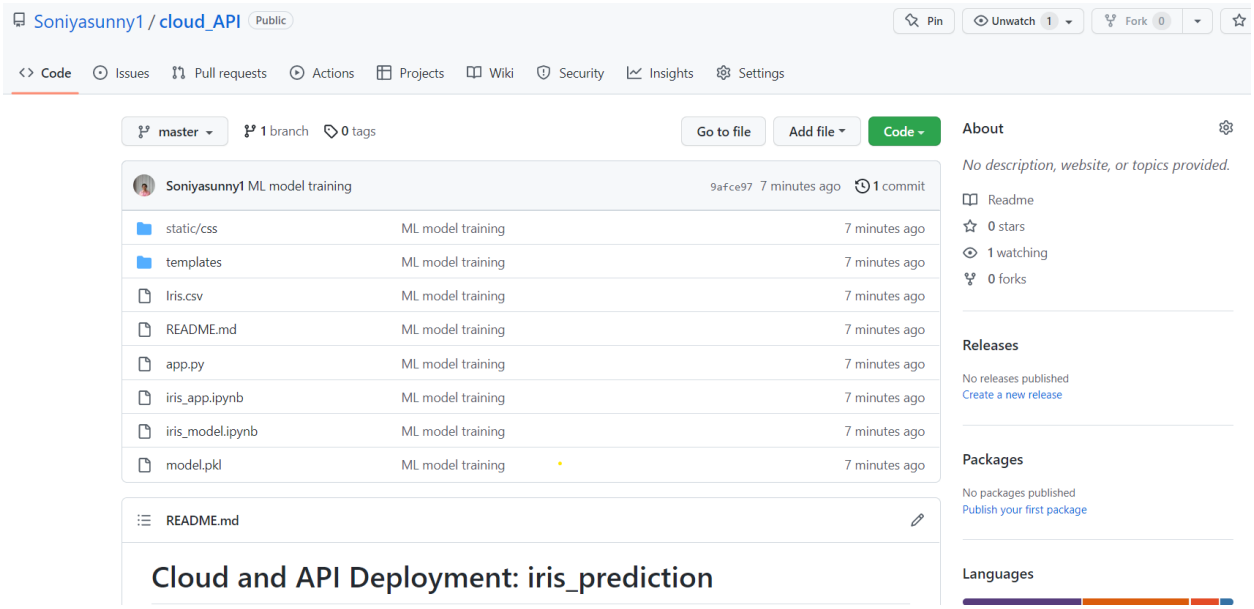
Copied the url and opened through a browser



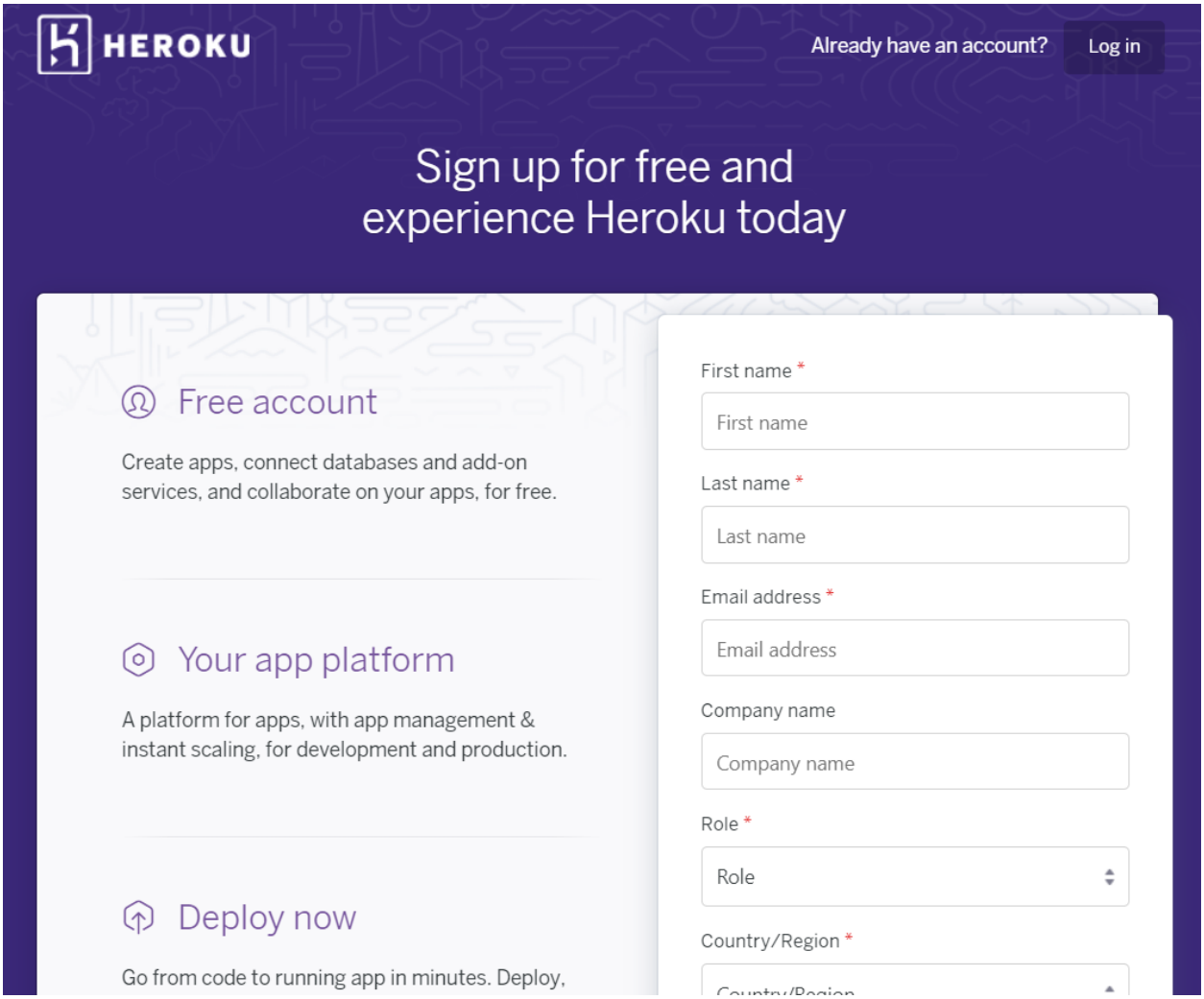
Tested the model

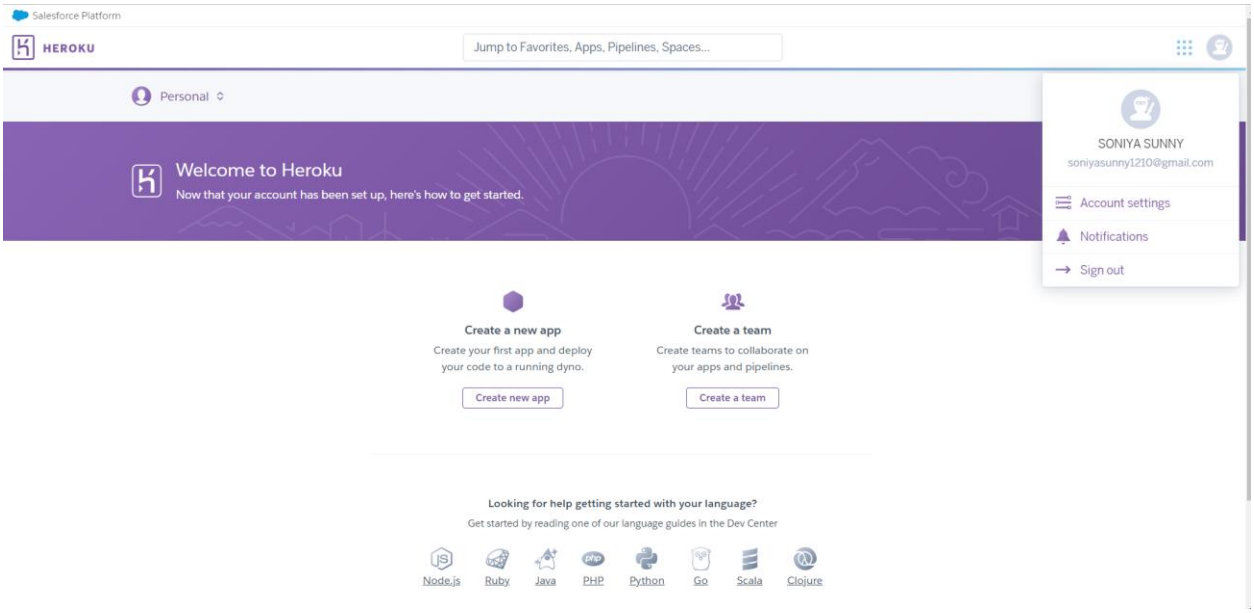


3. Committing Code in online Repo: -

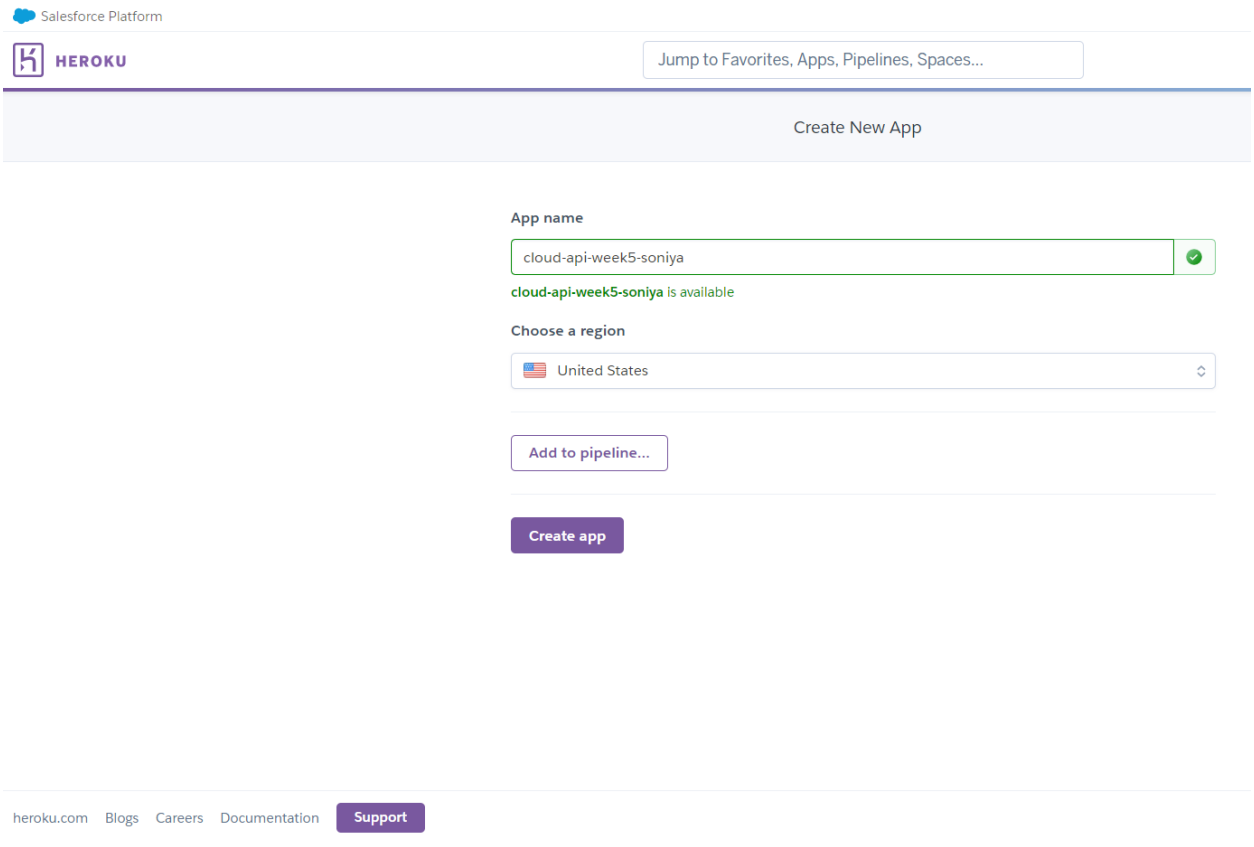


4. Account creation in Heroku: -

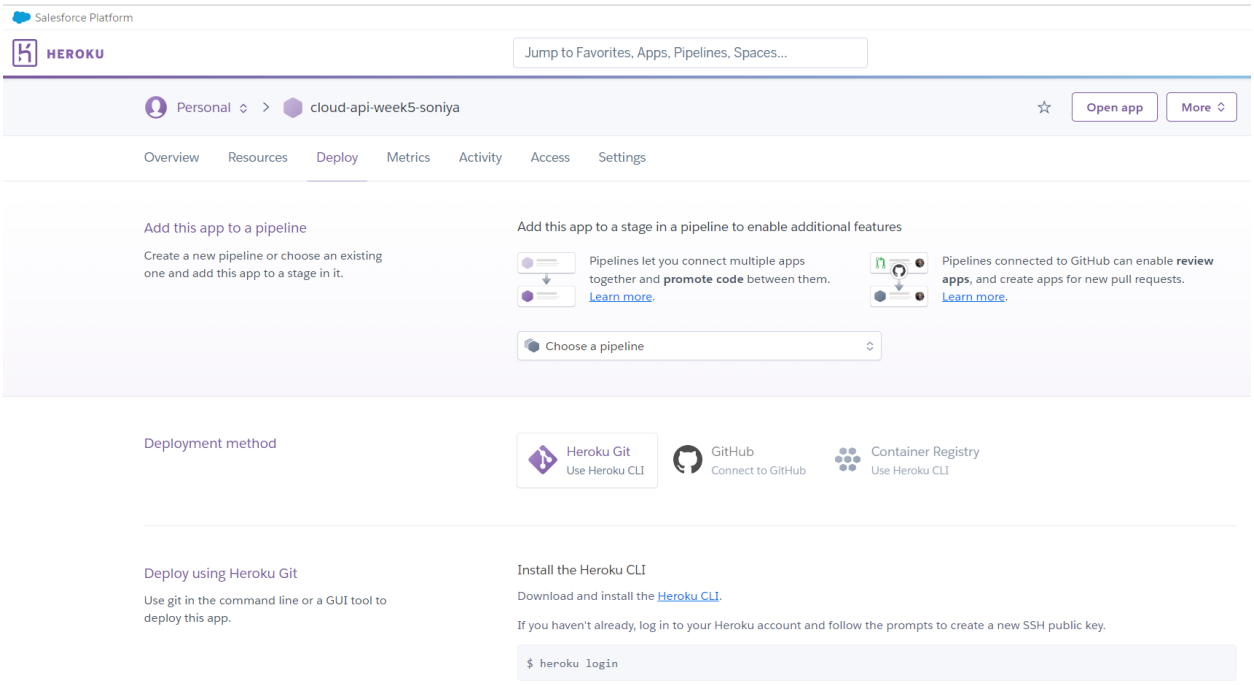




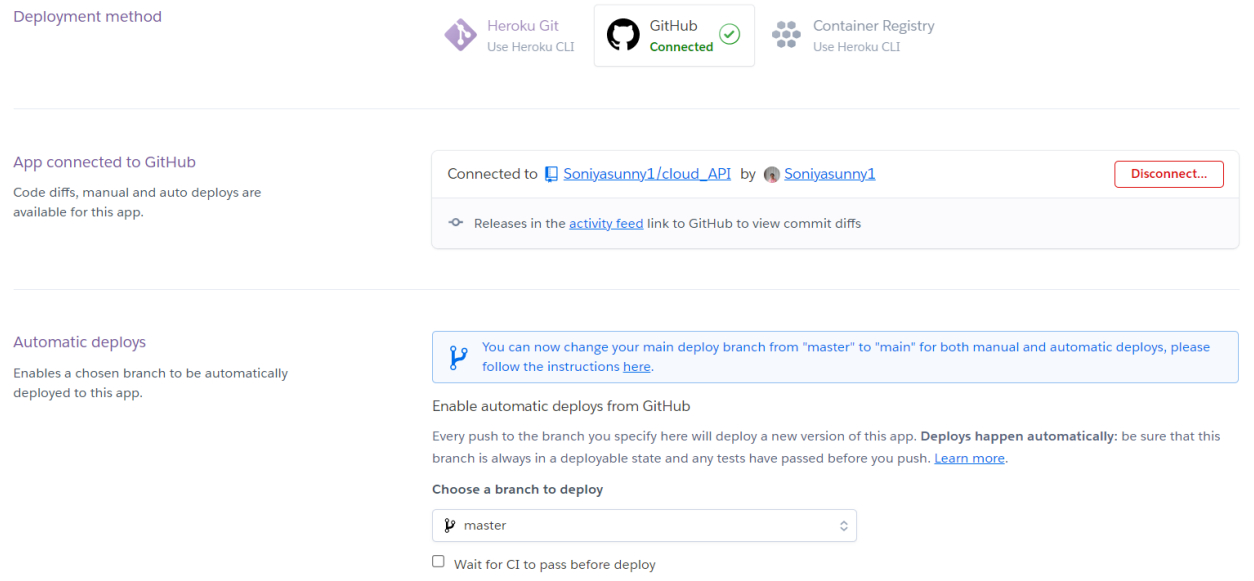
5. Linking of online repo to Heroku: -

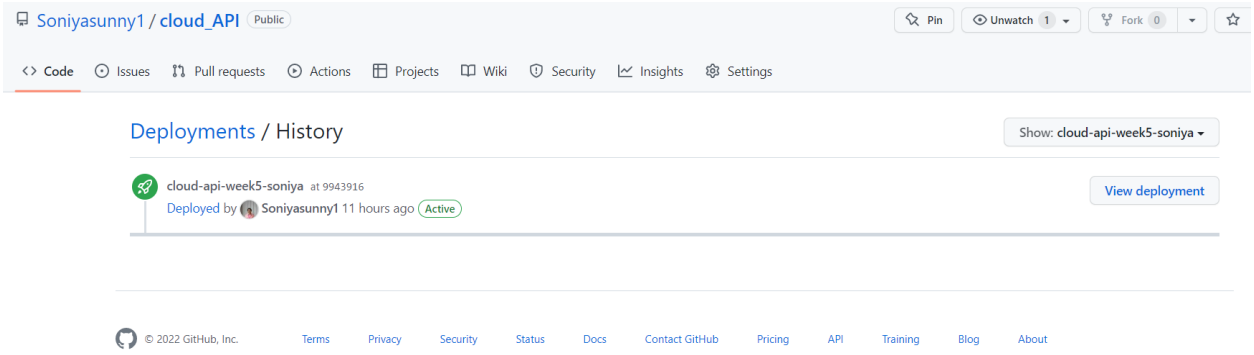


Heroku app created with name ‘cloud-api-week5-soniya’:

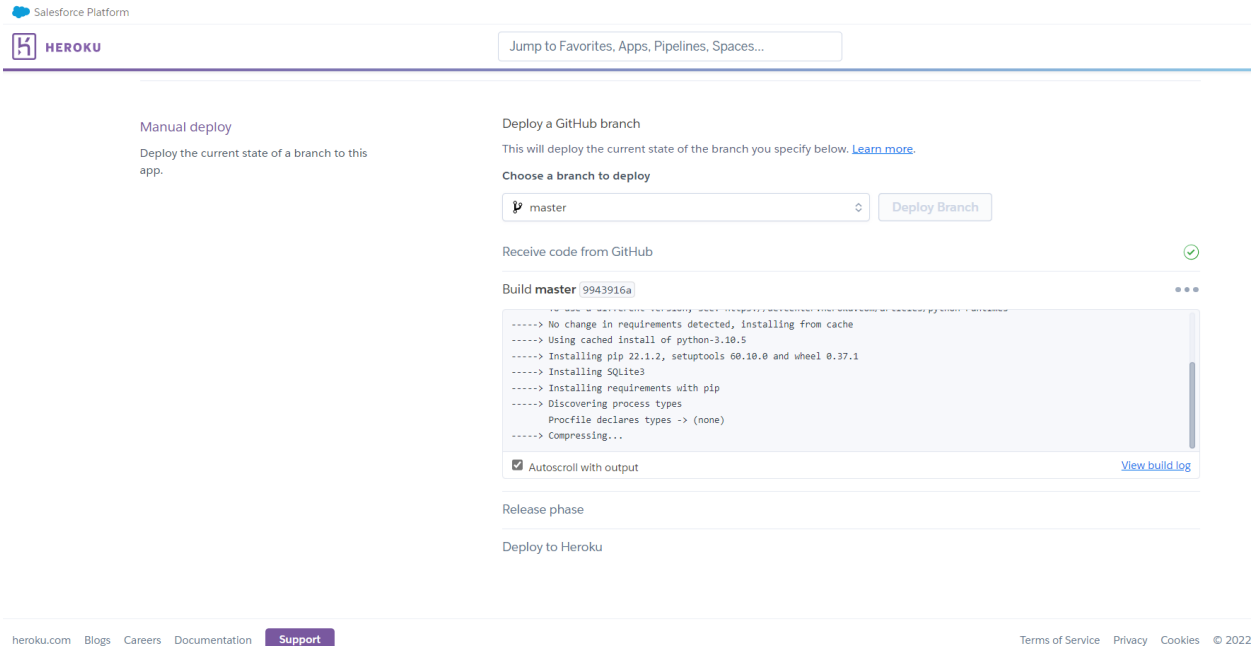


Connected to github:

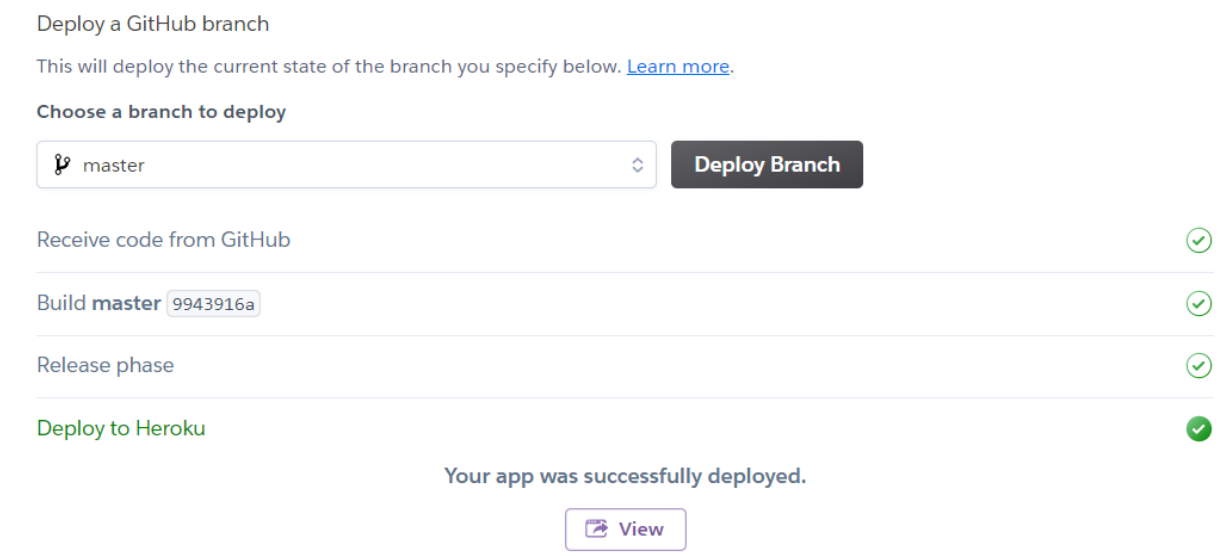




6. Deployment of ML model: -



Deployed branch manually and App successfully deployed:



7. Testing the web app: -

Got the output from the unique url provided by Heroku.