

Cloud and API Deployment (Week 5)

Name: Manoj Kumar Thangaraj

Batch Code: LISUM01

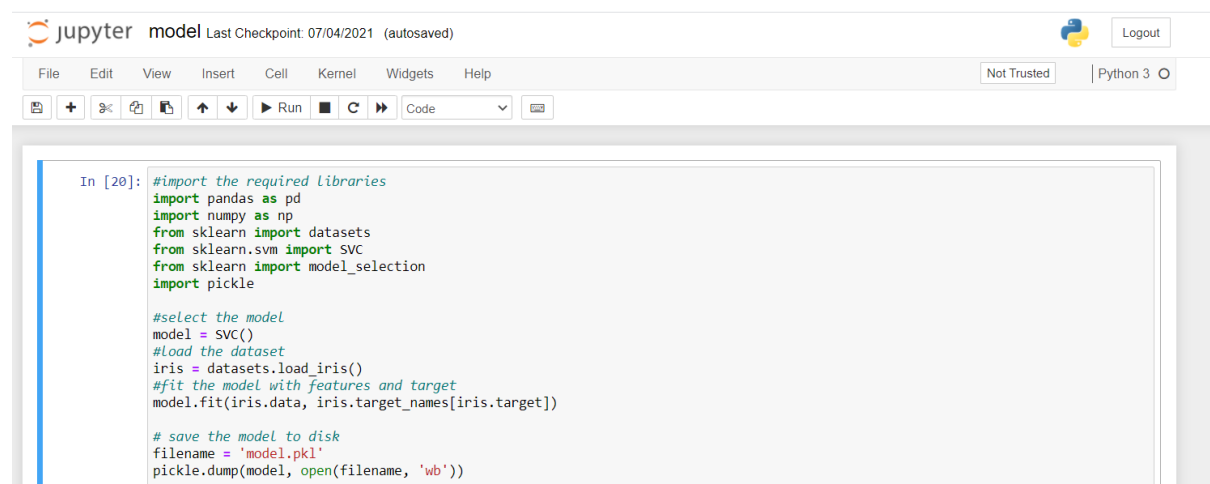
Submission Date: 13/07/2021

Task Objectives:

1. Select any toy data (simple data) (You are allowed to use data set of week 4)
 2. Save the model (You are allowed to use model of week 4)
 3. Deploy the model on any open-source cloud e.g., Heroku (Deployment should be API based as well as web app)
 4. Create pdf document (Name, Batch code, Submission date, submitted to) which should contain snapshot of each step of deployment)
 5. Upload the document to Github
 6. Submit the URL of the uploaded document.
-

For this task, I am choosing Heroku to deploy this model.

As per the task, the selected toy data was IRIS dataset from the Scikit-Learn library.

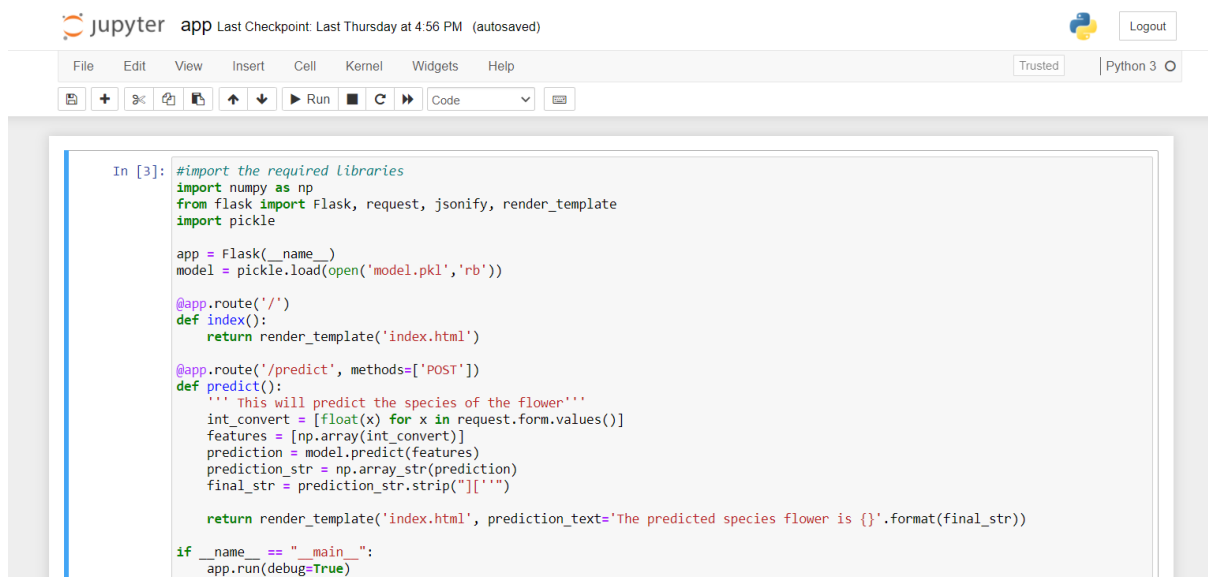


```
In [20]: #import the required libraries
import pandas as pd
import numpy as np
from sklearn import datasets
from sklearn.svm import SVC
from sklearn import model_selection
import pickle

#select the model
model = SVC()
#Load the dataset
iris = datasets.load_iris()
#fit the model with features and target
model.fit(iris.data, iris.target_names[iris.target])

# save the model to disk
filename = 'model.pkl'
pickle.dump(model, open(filename, 'wb'))
```

As per the above picture, the dataset is loaded straight from the scikit-learn library and machine learning model used is SVM. Once the model is fitted, the model is saved into a pickle file.

The image shows a JupyterLab interface. At the top, there's a header with the Jupyter logo, the text "app Last Checkpoint: Last Thursday at 4:56 PM (autosaved)", and a "Logout" button. Below the header is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu bar are "Trusted" and "Python 3" indicators. Below the menu bar is a toolbar with icons for file operations and a "Run" button. The main area is a code editor showing the following Python code:

```
In [3]: #import the required libraries
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

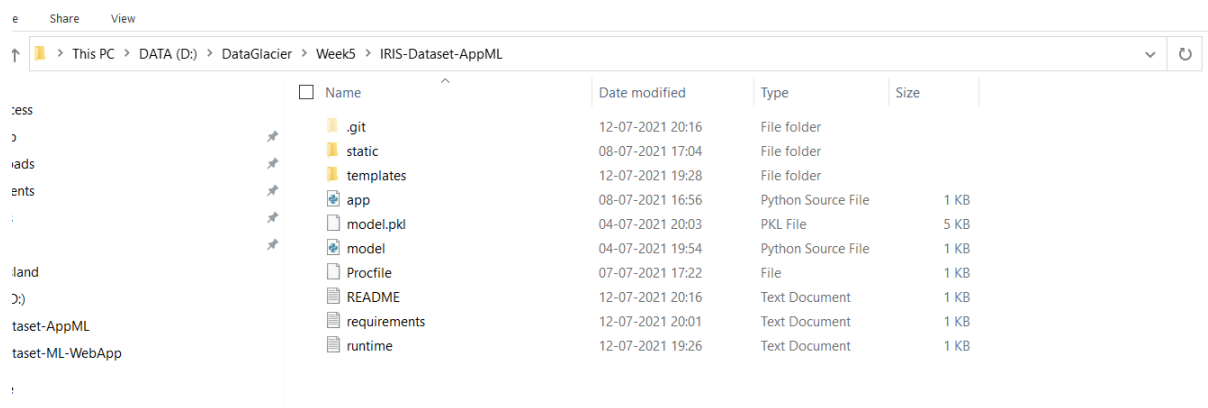
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    ''' This will predict the species of the flower'''
    int_convert = [float(x) for x in request.form.values()]
    features = [np.array(int_convert)]
    prediction = model.predict(features)
    prediction_str = np.array_str(prediction)
    final_str = prediction_str.strip('"')

    return render_template('index.html', prediction_text='The predicted species flower is {}'.format(final_str))

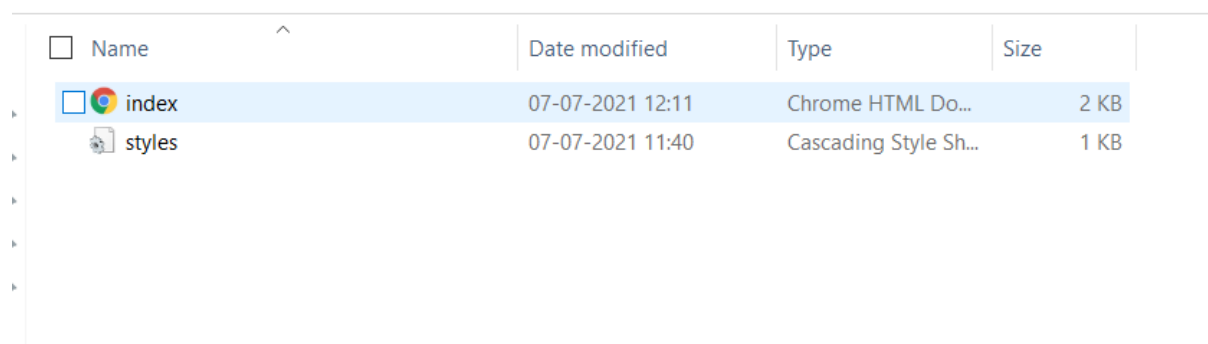
if __name__ == "__main__":
    app.run(debug=True)
```

The app.py which is the main file for the web app to run is in the picture above. These files are downloaded in the python file format.

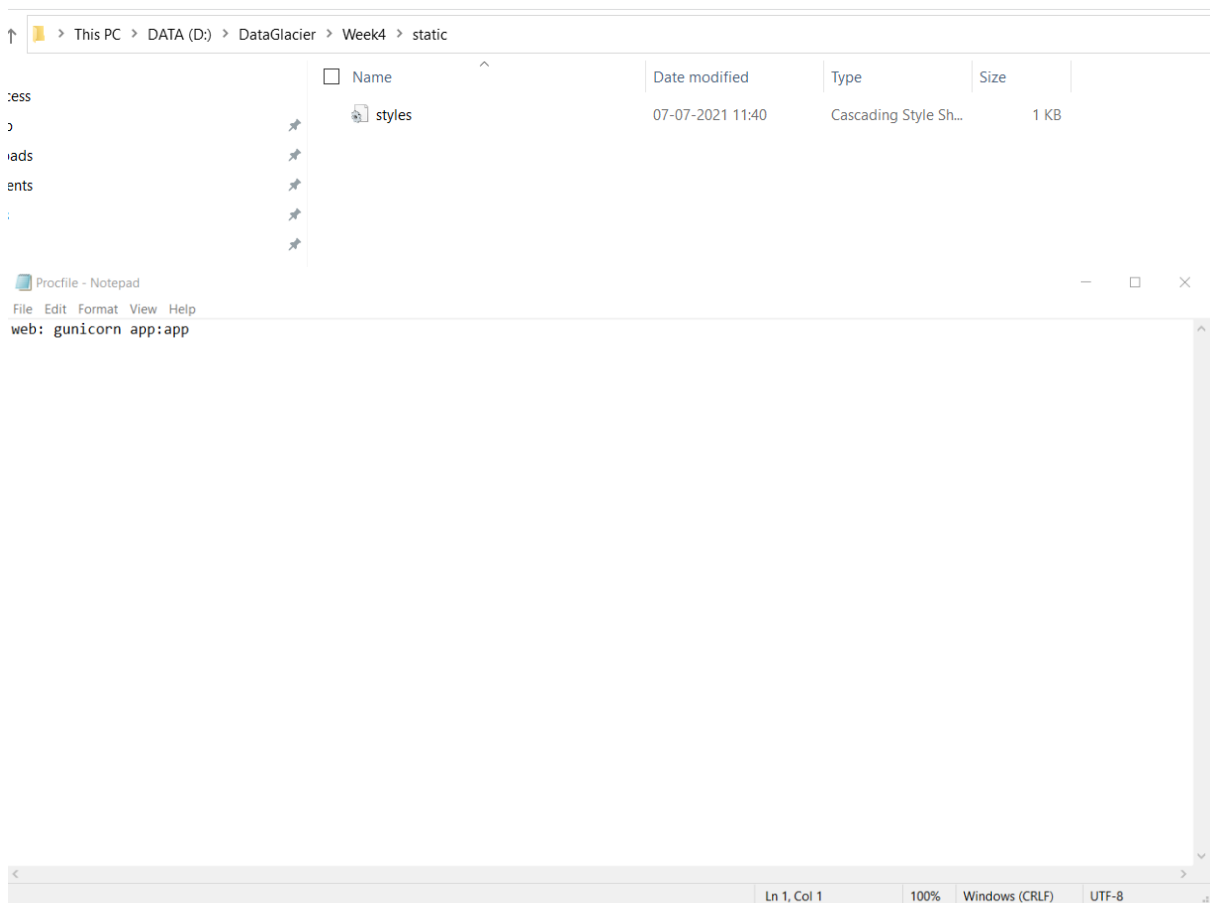
The image shows a Windows File Explorer window. The address bar shows the path: "This PC > DATA (D:) > DataGlacier > Week5 > IRIS-Dataset-AppML". The left sidebar shows a tree view with folders like "ess", "ads", "ents", "land", "taset-AppML", and "taset-ML-WebApp". The main pane shows a table of files and folders in the "Week5" directory.

Name	Date modified	Type	Size
.git	12-07-2021 20:16	File folder	
static	08-07-2021 17:04	File folder	
templates	12-07-2021 19:28	File folder	
app	08-07-2021 16:56	Python Source File	1 KB
model.pkl	04-07-2021 20:03	PKL File	5 KB
model	04-07-2021 19:54	Python Source File	1 KB
Procfille	07-07-2021 17:22	File	1 KB
README	12-07-2021 20:16	Text Document	1 KB
requirements	12-07-2021 20:01	Text Document	1 KB
runtime	12-07-2021 19:26	Text Document	1 KB

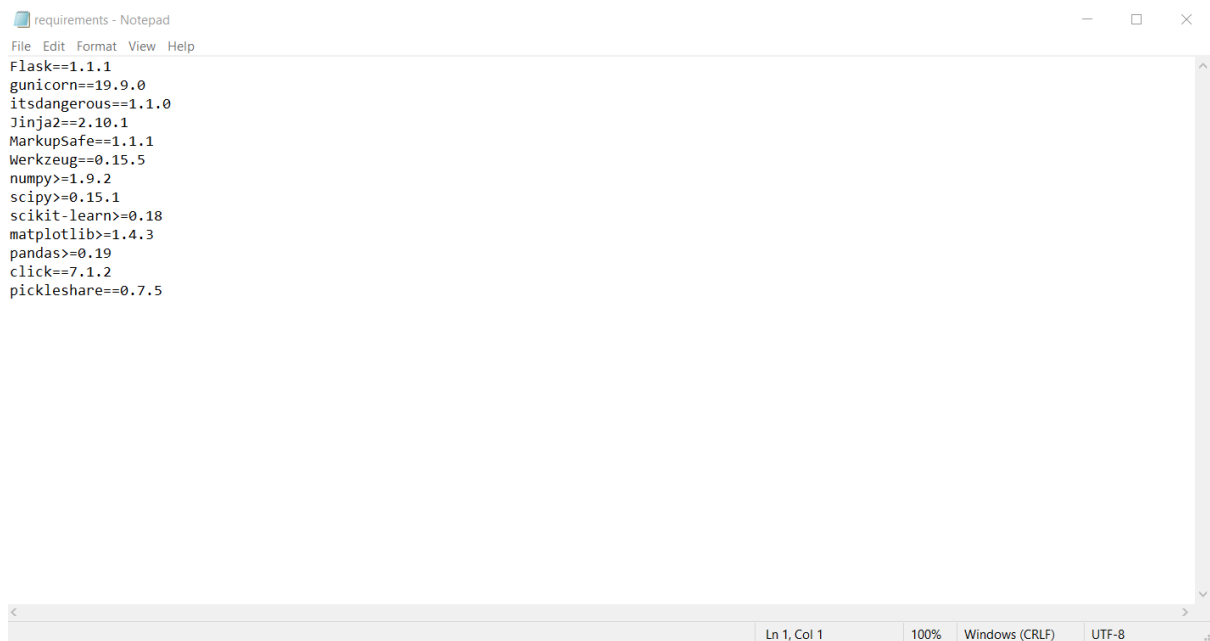
The week 5 folder consists of files app.py, model.pkl, model.py, procfile, requirements.txt, runtime.txt, templates folder contains the index.html file and style file in css format and static folder consist of style css file.

The image shows a Windows File Explorer window displaying the contents of the "templates" folder. The table shows two files: "index" and "styles".

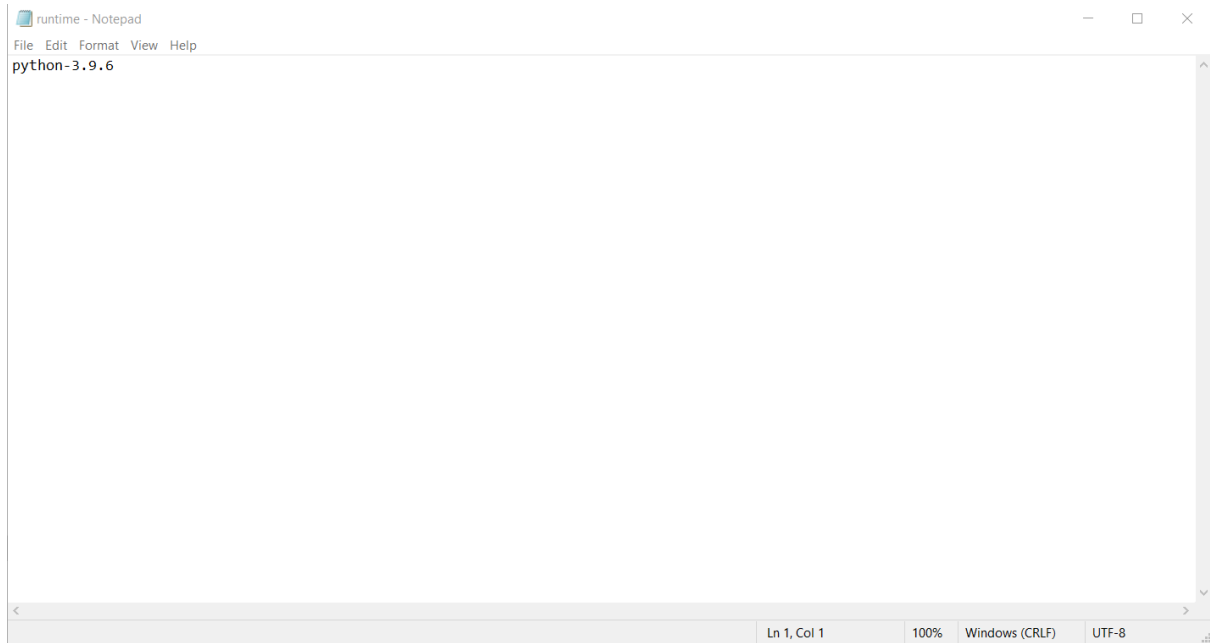
Name	Date modified	Type	Size
index	07-07-2021 12:11	Chrome HTML Do...	2 KB
styles	07-07-2021 11:40	Cascading Style Sh...	1 KB



The procfile consists of the web app that is going to be used, the app that is needed to run from the file.

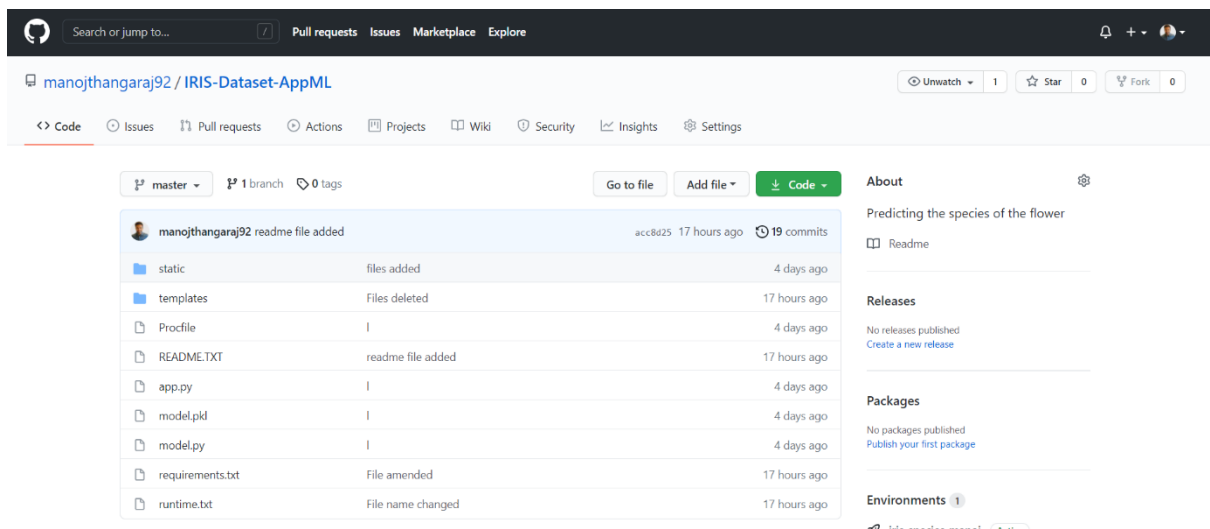


The requirements.txt file should contain in the root of the file for the Heroku to understand and download the required libraries for deploying.



This runtime.txt tells Heroku to use the specified python version to upload.

After this, the whole folder is file is uploaded in the github and connected to Heroku to deploy our model.



This is how the repository in the github looks like. It contains all the required files.

The new app is created on the Heroku,

Salesforce Platform

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Personal > iris-species-manoj

GitHub manojthangaraj92/IRIS-Dataset-AppML

Overview Resources **Deploy** Metrics Activity Access Settings

Add this app to a pipeline
Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features
Pipelines let you connect multiple apps together and **promote code** between them. [Learn more](#)
Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. [Learn more](#)

Choose a pipeline

Deployment method

Heroku Git Use Heroku CLI

GitHub **Connected**

Container Registry Use Heroku CLI

Deployment method

Heroku Git Use Heroku CLI

GitHub **Connected**

Container Registry Use Heroku CLI

App connected to GitHub
Code diffs, manual and auto deploys are available for this app.

Connected to [manojthangaraj92/IRIS-Dataset-AppML](#) by [manojthangaraj92](#) [Disconnect...](#)

Releases in the [activity feed](#) link to GitHub to view commit diffs

Automatic deploys
Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

Enable automatic deploys from GitHub
Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically**: be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more](#)

Choose a branch to deploy

master

☐ Wait for CI to pass before deploy

For the deployment method, github repository is connected and we will choose manual deployment.

Salesforce Platform

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

master

☐ Wait for CI to pass before deploy
Only enable this option if you have a Continuous Integration service configured on your repo.

Enable Automatic Deploys

Manual deploy
Deploy the current state of a branch to this app.

Deploy a GitHub branch
This will deploy the current state of the branch you specify below. [Learn more](#)

Choose a branch to deploy

master **Deploy Branch**

Receive code from GitHub ☒

Build master acc8d252 ☒

Release phase ☒

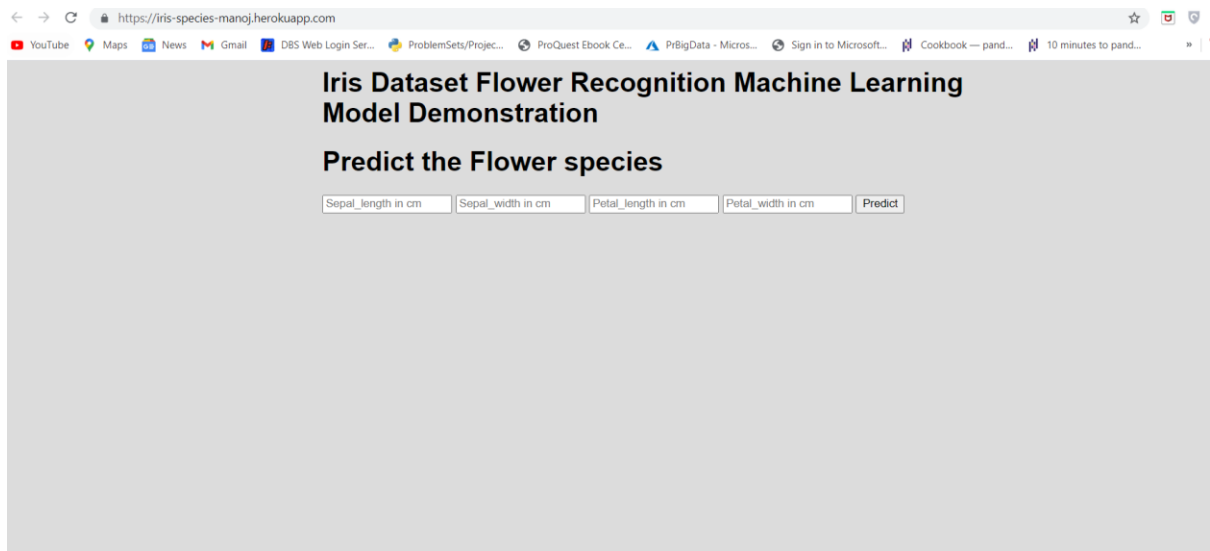
Deploy to Heroku ☒

Your app was successfully deployed.

[View](#)

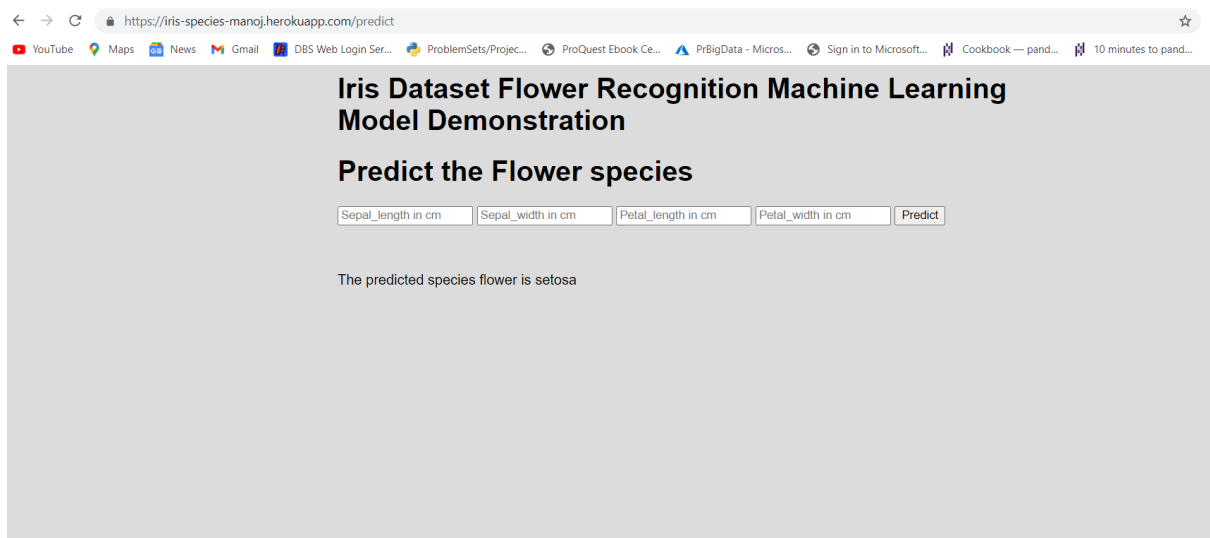
Once the view button is clicked on the above image, we can see our deployed web app.

The web app is deployed in the above link, and it is accessible globally.



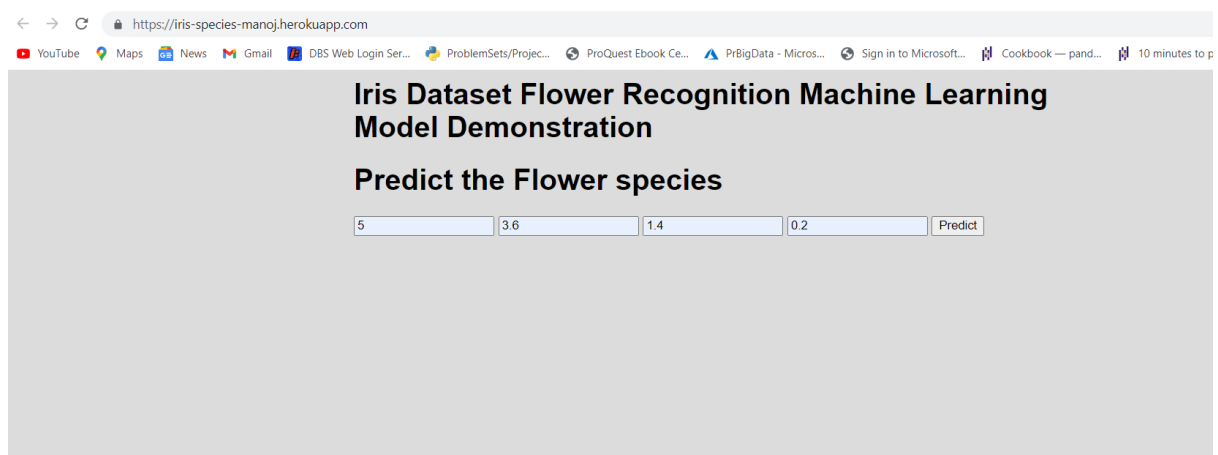
The screenshot shows a web browser window with the URL <https://iris-species-manoj.herokuapp.com>. The page title is "Iris Dataset Flower Recognition Machine Learning Model Demonstration". Below the title, the heading "Predict the Flower species" is displayed. There are four input fields labeled "Sepal_length in cm", "Sepal_width in cm", "Petal_length in cm", and "Petal_width in cm", followed by a "Predict" button.

Once the values are fed in and on clicking the predict button,



The screenshot shows the same web browser window, but now the "Predict" button has been clicked. Below the input fields, the text "The predicted species flower is setosa" is displayed.

We will get the prediction.



The screenshot shows the web browser window with the URL <https://iris-species-manoj.herokuapp.com>. The page title is "Iris Dataset Flower Recognition Machine Learning Model Demonstration". Below the title, the heading "Predict the Flower species" is displayed. The input fields are now filled with values: "5" for "Sepal_length in cm", "3.6" for "Sepal_width in cm", "1.4" for "Petal_length in cm", and "0.2" for "Petal_width in cm". The "Predict" button is still visible.