Data Glacier

Data Science Intern at Data Glacier

Week 5: Cloud API Development

Name: Anitha Venkatachalam

Batch Code: LISUM14

Date: 21 OCT 2022

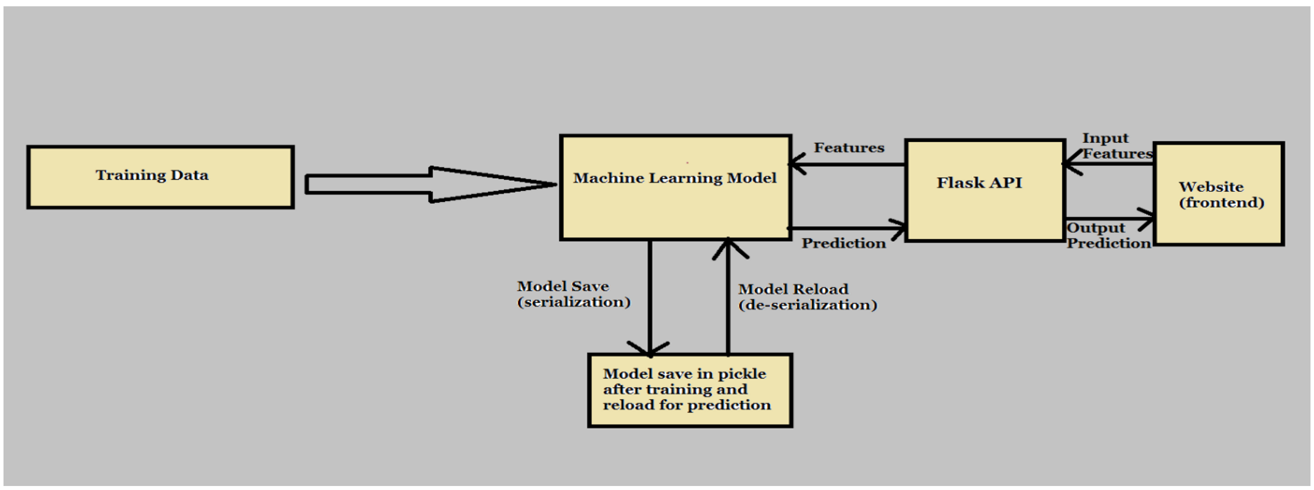
Submitted to: Data Glacier

Table of Contents:

1. Introduction ……….……………………………………………………….………2
2. Build Machine Learning Model ……………………………………….…………3
3. Turning Model into Flask Framework …………………………………...….6
   3.1. App.py …………………………………………………………………...7
   3.2. Index.html ………………………………………..……………...8
4. Running Procedure …………………………………………………..…9
5. Model Development using Heroku ………………………………………...12
   5.1. Steps for Model Development using Heroku

1. Introduction

In this project, we are going to deploying machine learning model using the Flask Framework. As a demonstration, our model helps to predict the values based on the input given in Iris Dataset.

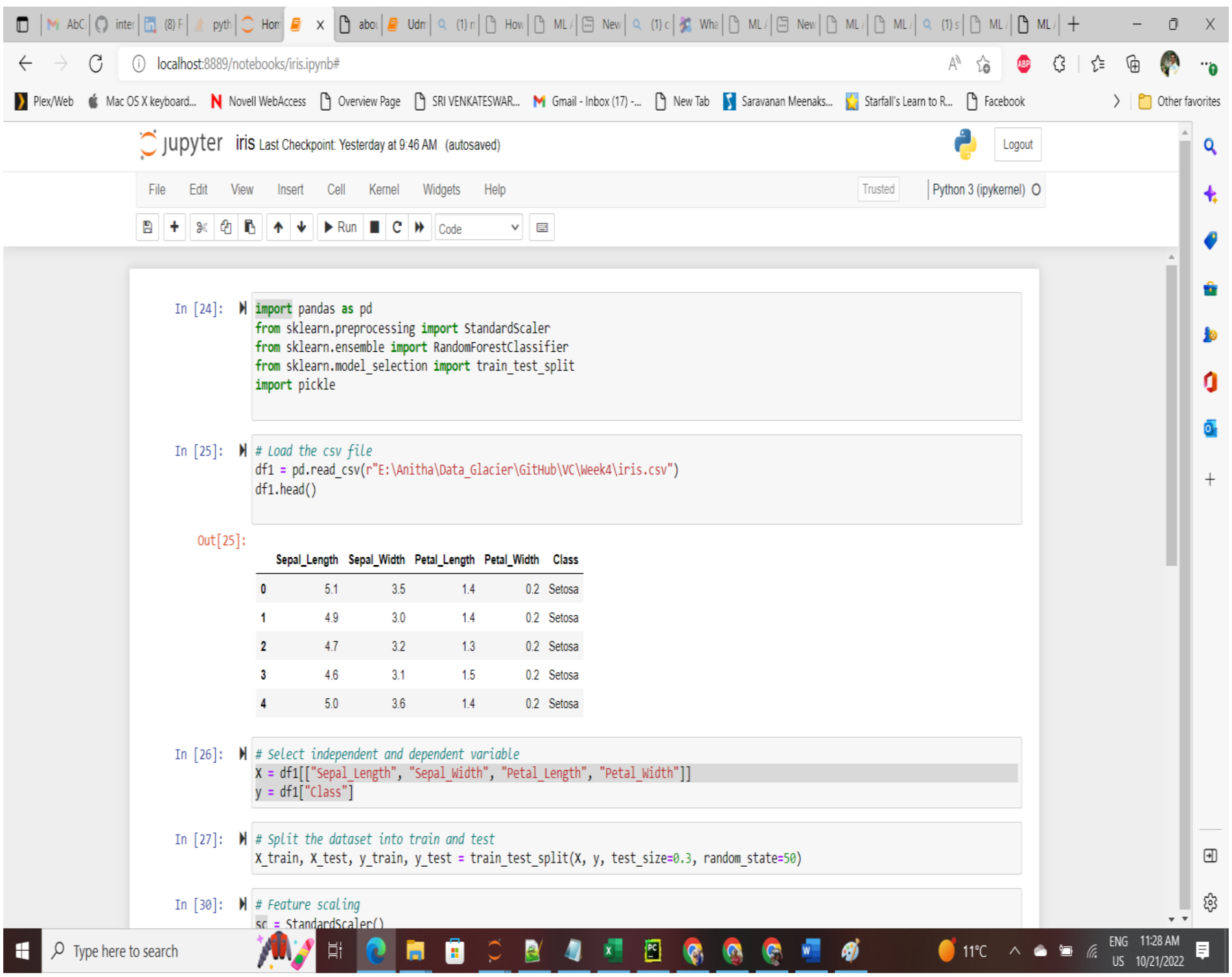


we will focus on both: building a machine learning model for Iris Dataset, then create an API for the model, using Flask, the Python micro-framework for building web applications. This API allows us to utilize predictive capabilities through HTTP requests.

2. Building a Model

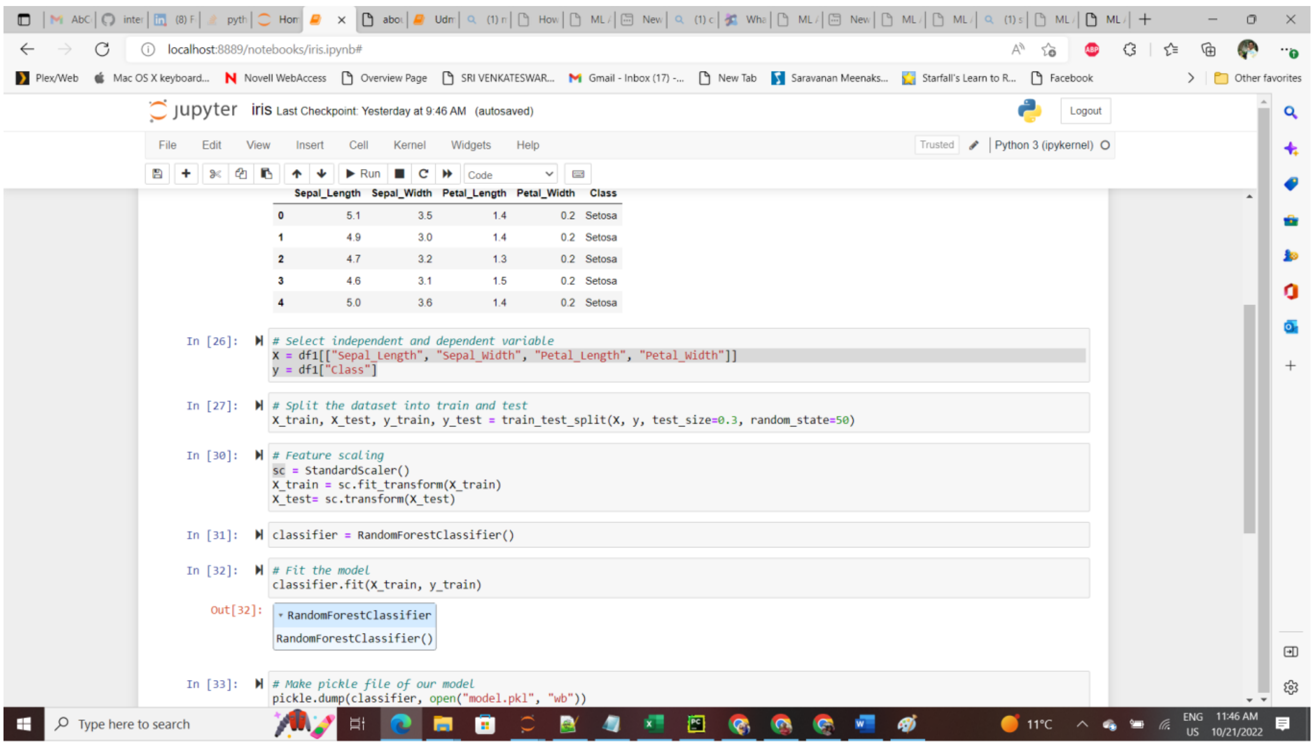
2.1.1 Import Required Libraries and Dataset

In this part, we import libraires and dataset which contain the information of Iris dataset.



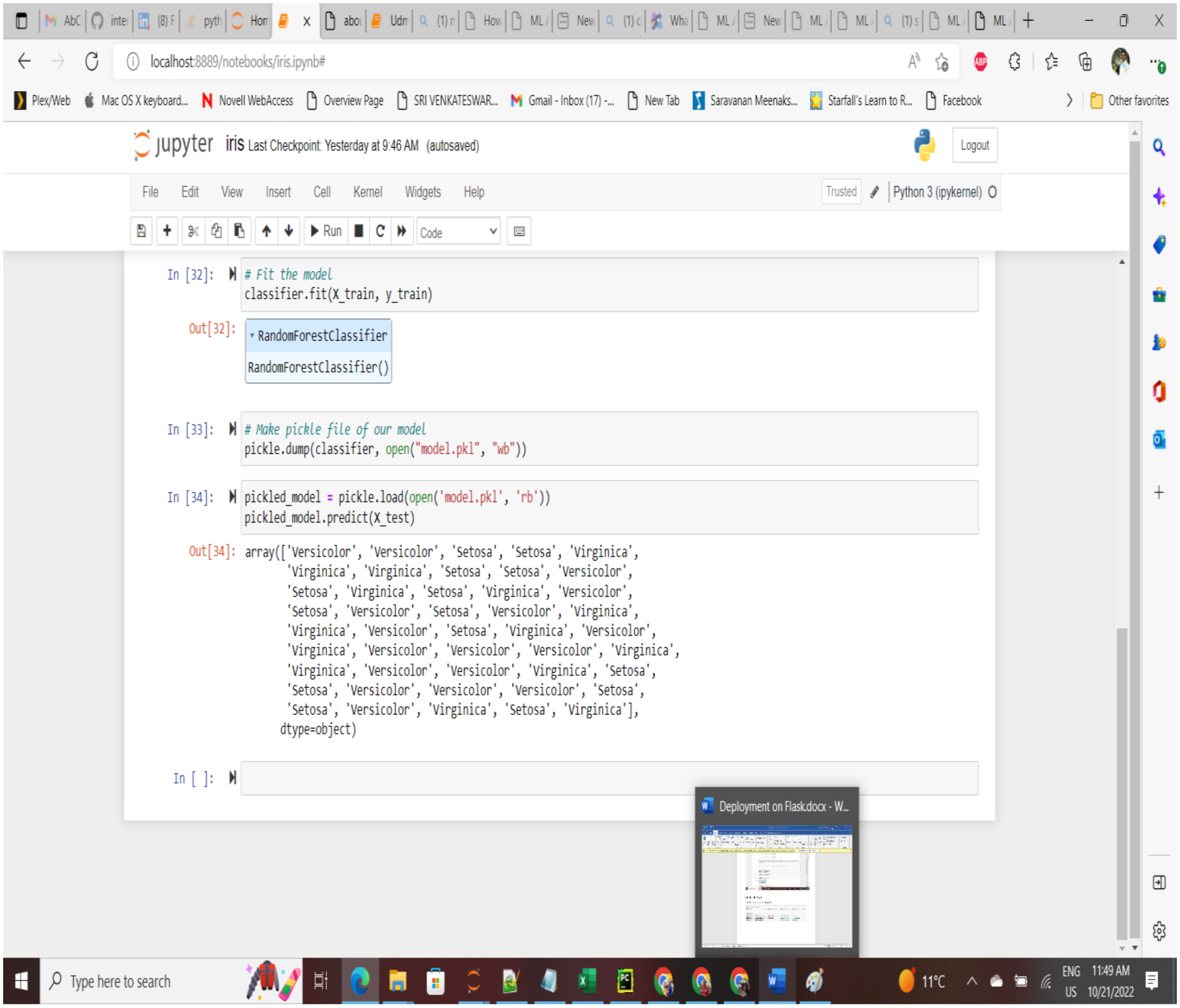
3.1.1 Build Model

After data preprocessing, we implement machine learning model to predict the iris value in the dataset. For this purpose, we implement Standard scalar using sklearn. After importing and initialize Standard scalar model we fit into training dataset.



3.1.3 Save the Model

After that we save our model using pickle



4.Turning Model into Web Application

We develop a web application that consists of a simple web page with a form field with entering the values and predict. After submitting the message to the web application, it will render it on a new page which gives us a result of Versicolor, Virginica.First, we create a folder for this project called Iris, this is the directory tree inside the folder. We will explain each file.

Table 3.1: Application Folder File Directory

app.py	
Templates	**Index.html**
Model	**model.pkl**
Dataset/	**Iris.csv**

3.1 App.py

The *app.py* file contains the main code that will be executed by the Python interpreter to run the Flask web application, it included the ML code for classifying SD.

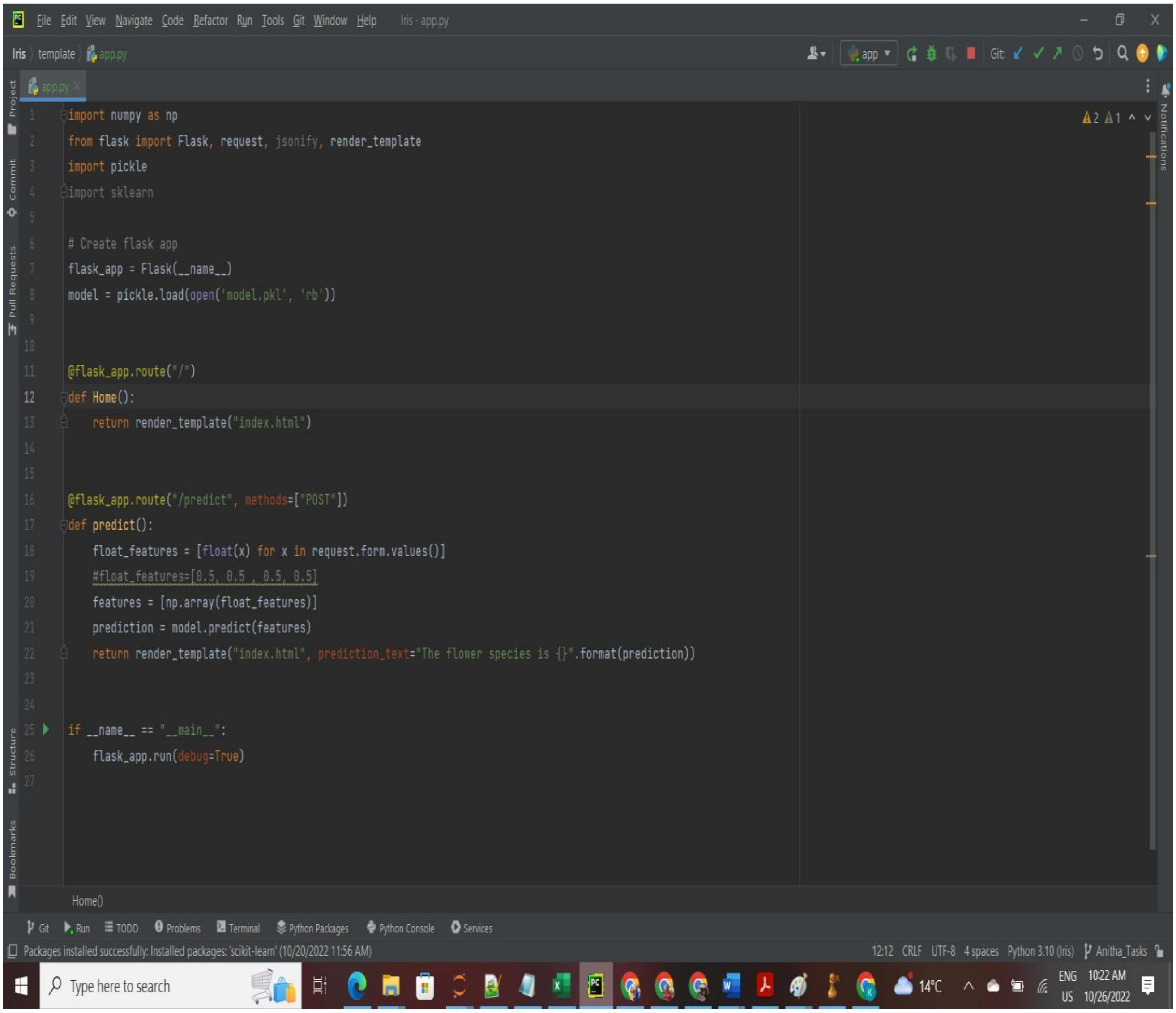


Figure 3.1: App.py

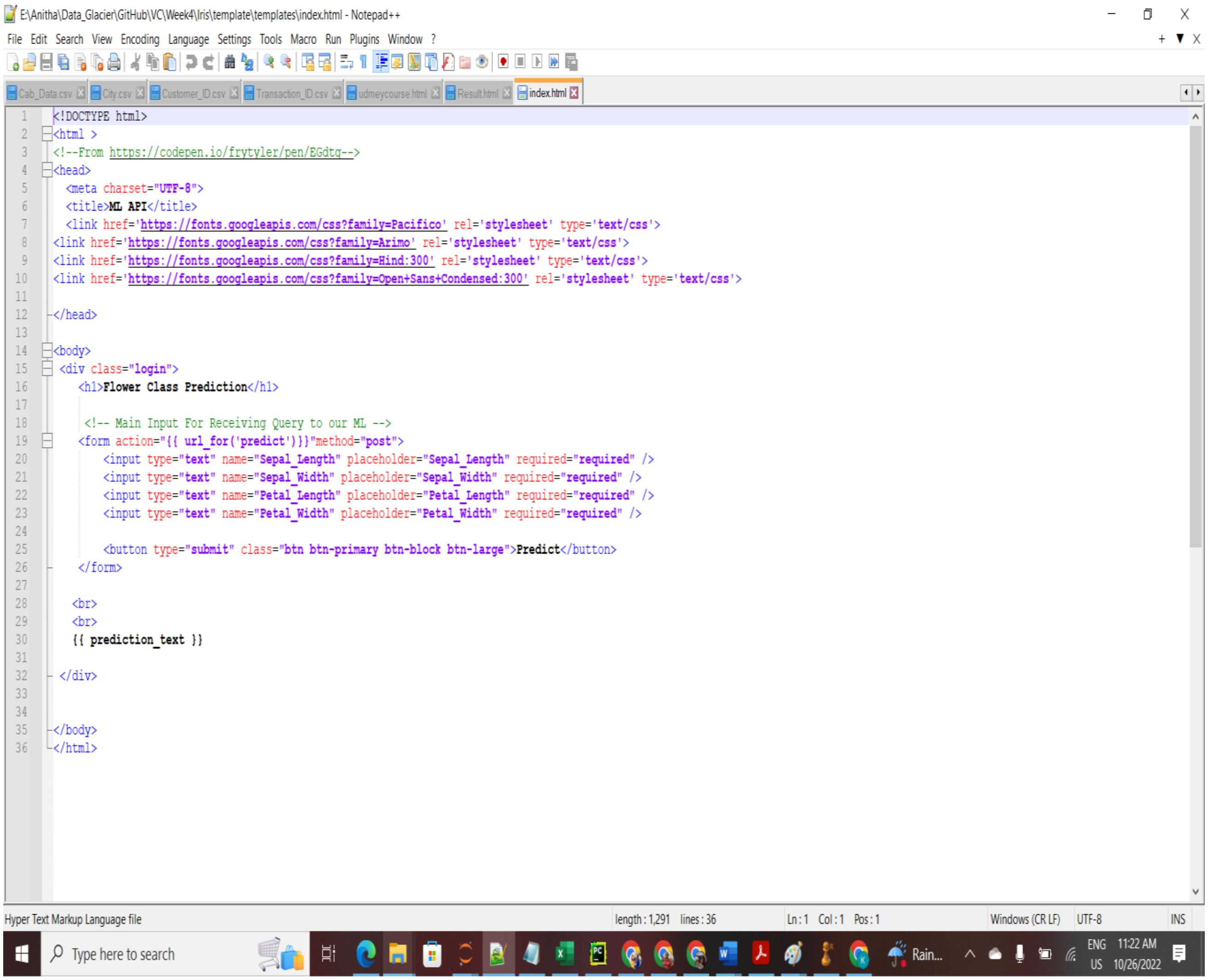
- We ran our application as a single module; thus, we initialized a new Flask instance with the argument *_name* to let Flask know that it can find the HTML template folder (*templates*) in the same directory where it is located.

• Our *home* function simply rendered the *Index.html* HTML file, which is located in the *templates* folder.

- Inside the *predict* function, we access the Iris data set, pre-process the values, and make predictions, then store the model. We access the new message entered by the user and use our model to make a prediction for its label.
- we used the *POST* method to transport the form data to the server in the message body. Finally, by setting the *debug=True* argument inside the app.run method, we further activated Flask's debugger.
- Lastly, we used the *run* function to only run the application on the server when this script is directly executed by the Python interpreter, which we ensured using the *if* statement with *__name__ == '__main__'*.

- **3.2 Index.html**

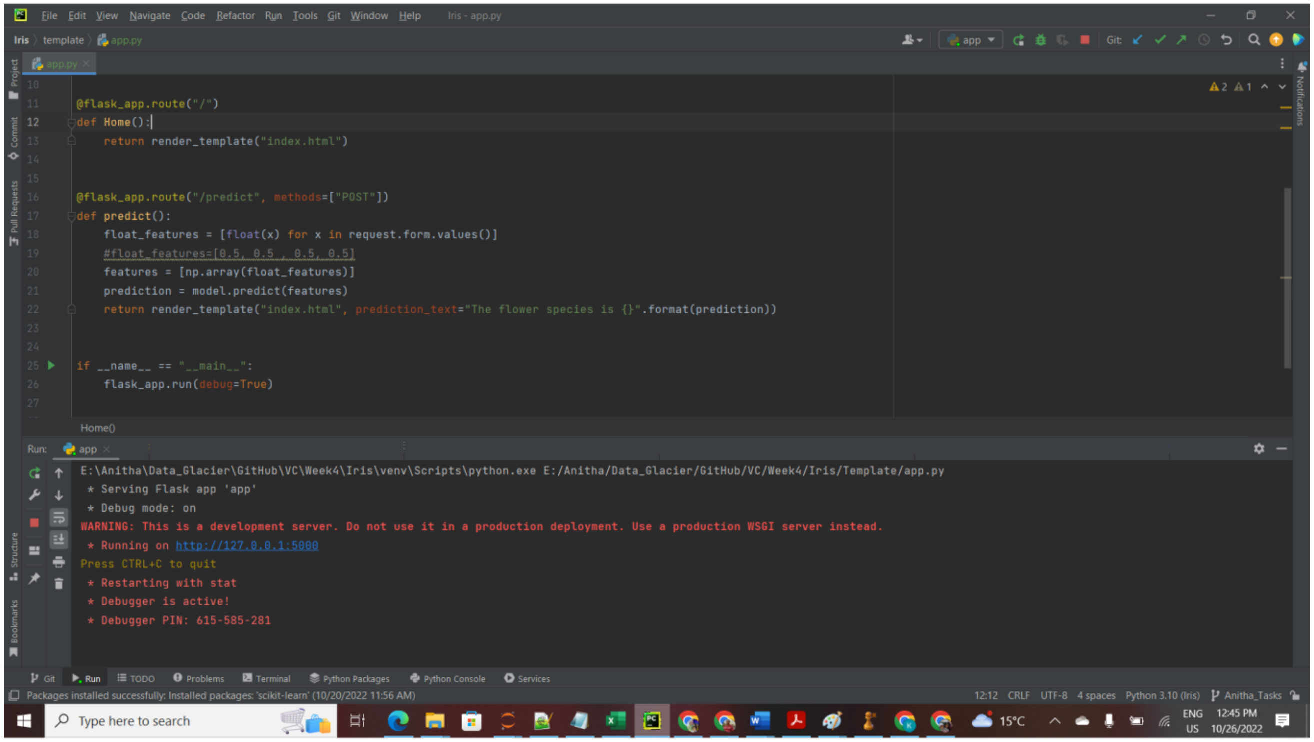
The following are the contents of the *Index.html* file that will enter the values to predict the flowers.



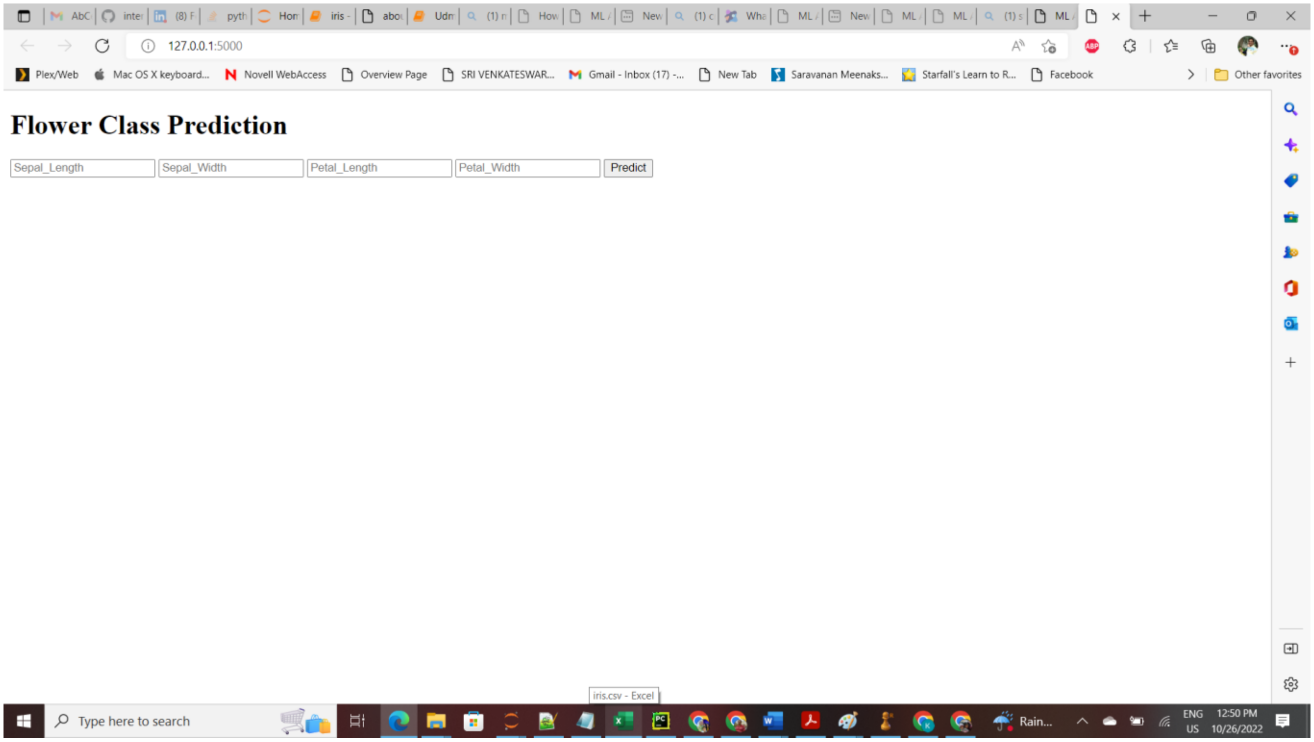
```html
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

</head>

<body>
 <div class="login">
    <h1>Flower Class Prediction</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
        <input type="text" name="Sepal_Length" placeholder="Sepal_Length" required="required" />
        <input type="text" name="Sepal_Width" placeholder="Sepal_Width" required="required" />
        <input type="text" name="Petal_Length" placeholder="Petal_Length" required="required" />
        <input type="text" name="Petal_Width" placeholder="Petal_Width" required="required" />

        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>

   <br>
   <br>
   {{ prediction_text }}

 </div>


</body>
</html>
```

- ### 4.1.2 Running Procedure

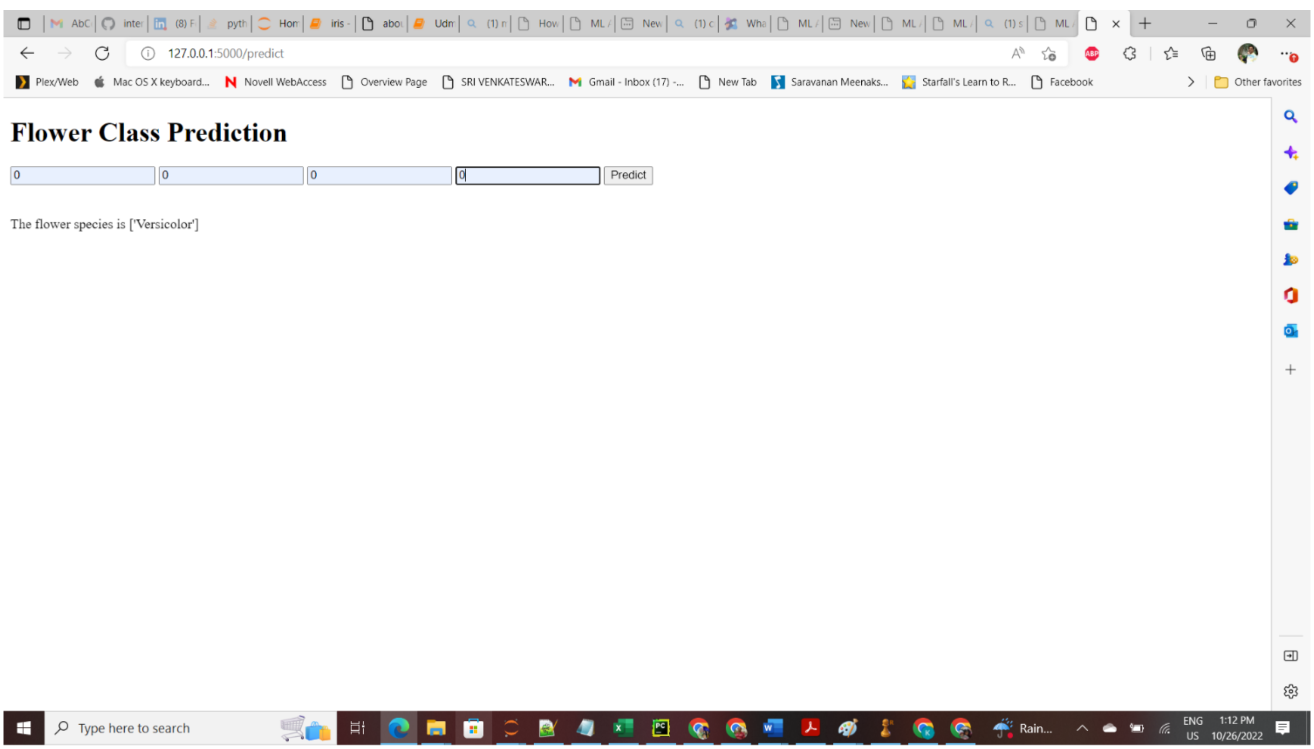
Once we have done all of the above, we can start running the API by clicking run button in *app.py* screen:

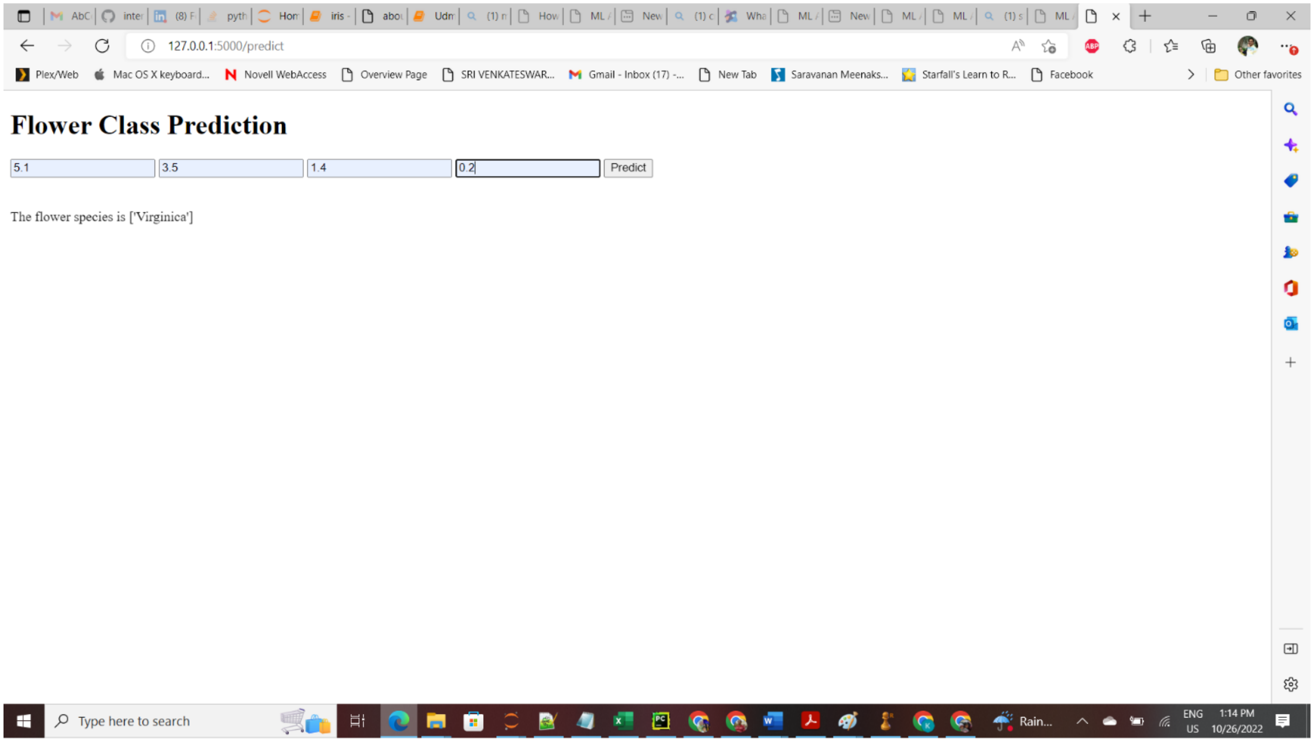


Now we could open a web browser and navigate to http://127.0.0.1:5000/, we should see a simple website with the content like so



Now we enter input in the values in textbox and the value is predicted.





5. Model deployment using Heroku

We're ready to start our Heroku deployment now that our model has been trained, the machine learning pipeline has been set up, and the application has been tested locally. There are a few ways to upload the application source code onto Heroku. The easiest way is to link a GitHub repository to your Heroku account.

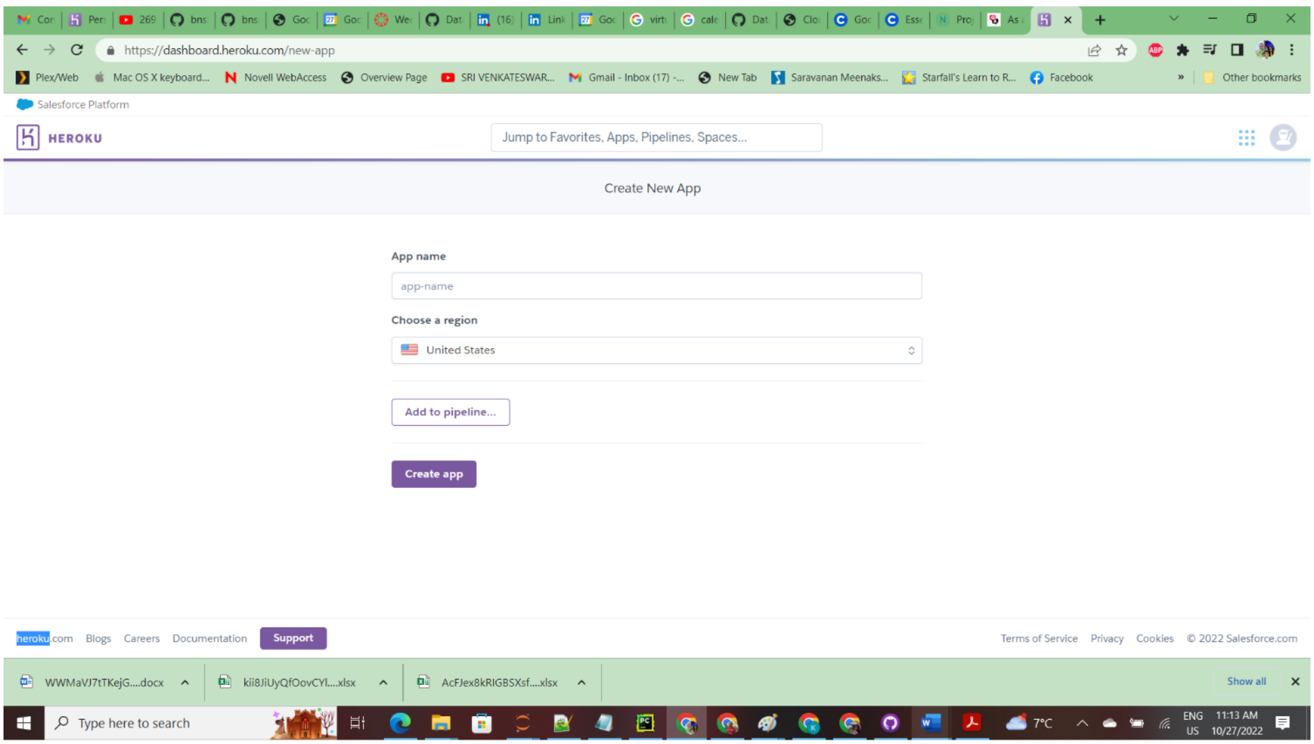
Requirement.txt

It is a text file containing the python packages required to execute the application.

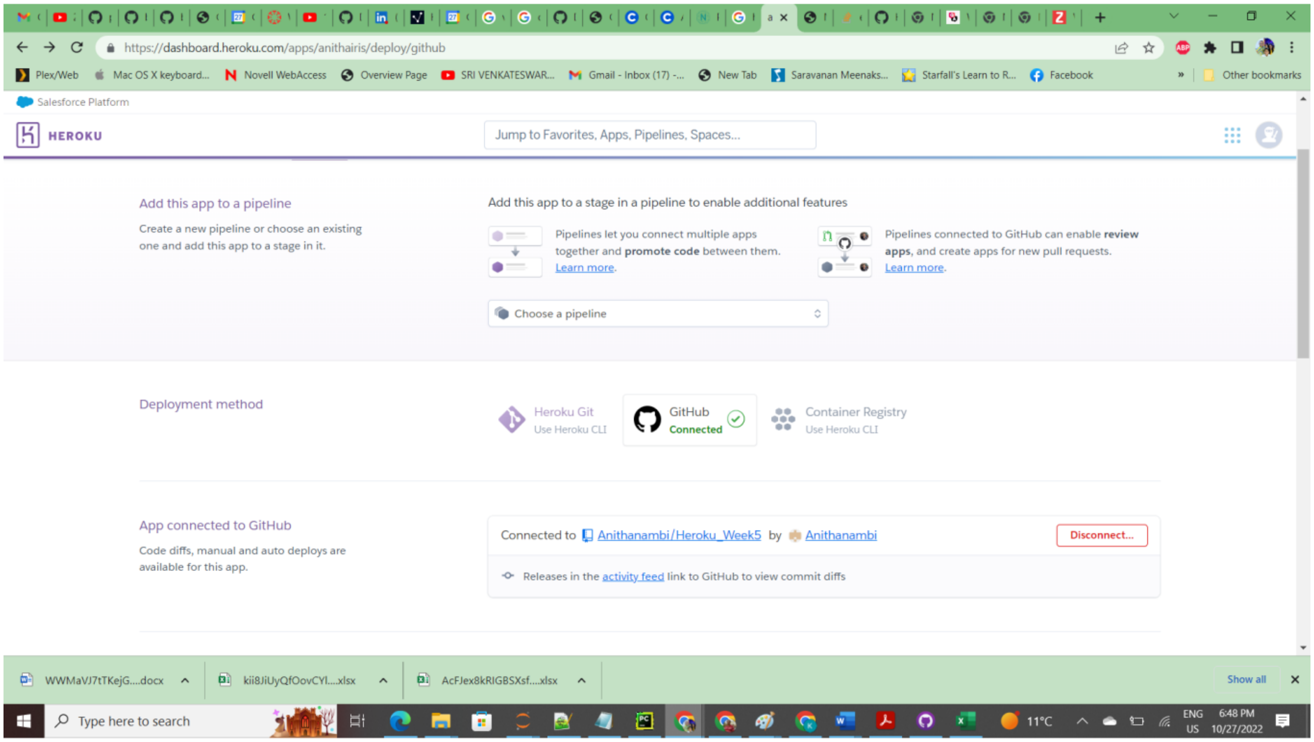
5.1 Steps for Model Deployment Using Heroku

Once we uploaded files to the GitHub repository, we are now ready to start deployment on Heroku. Follow the steps below:

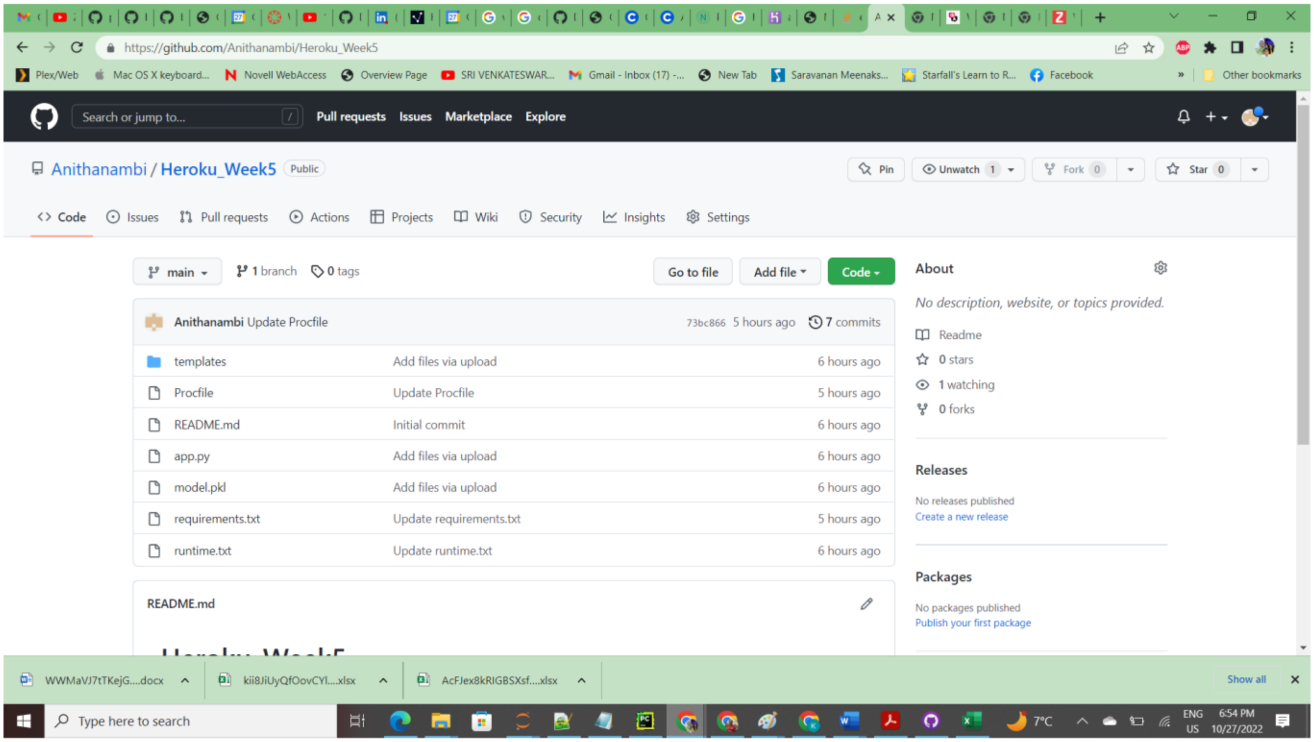
After sign up on **heroku.com** then click on **Create new app**



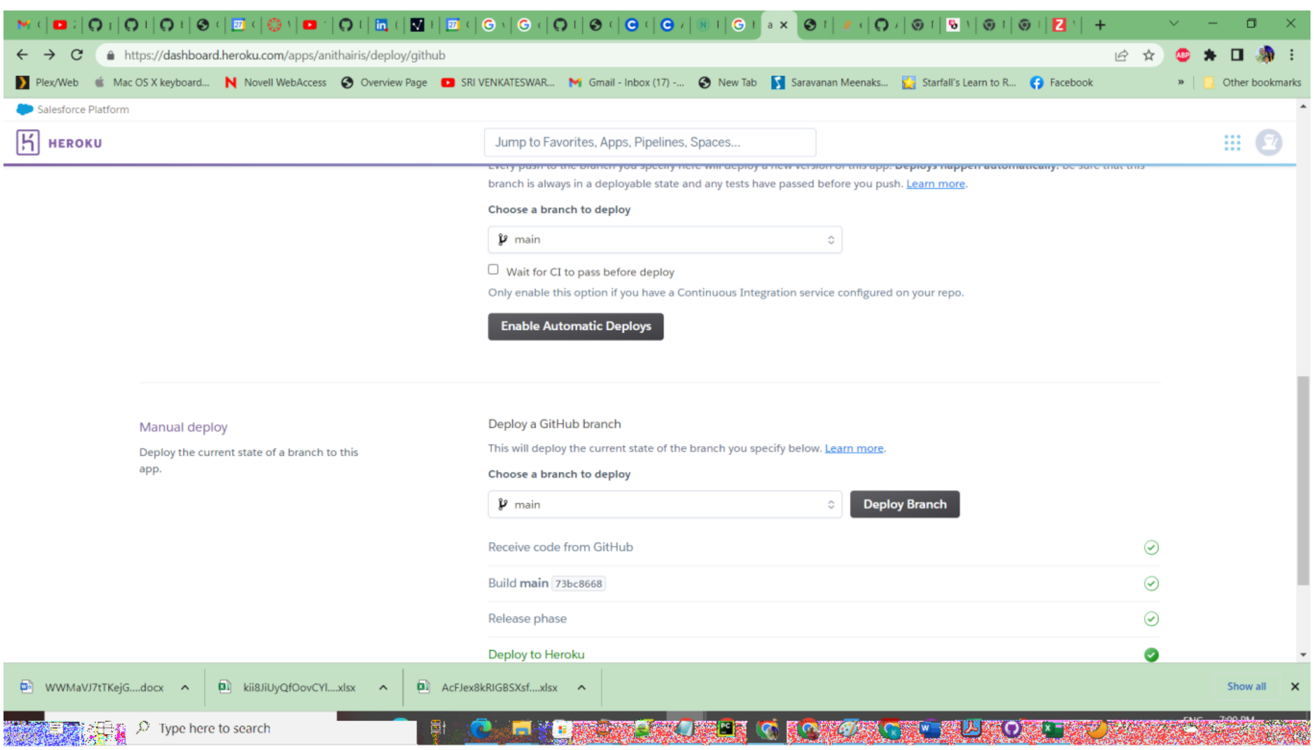
3. Connect to GitHub repository where code is I uploaded.



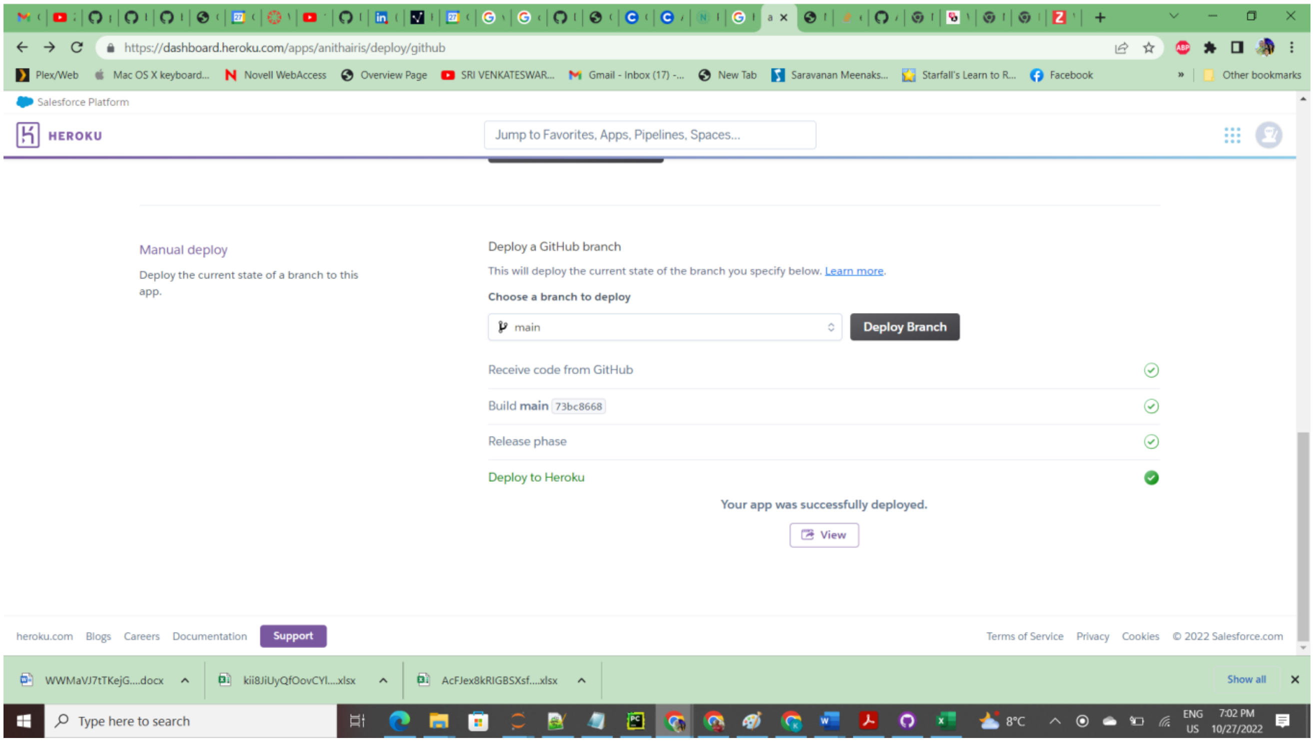
After that I choose the repository where I upload the code.



4. Deploy branch



After 5 minutes our application is Ready



The app is Successfully Published at

https://anithairis.herokuapp.com/