Data Glacier

Your Deep Learning Partner

Model Deployment with Flask

LISUM02

19th January 2022

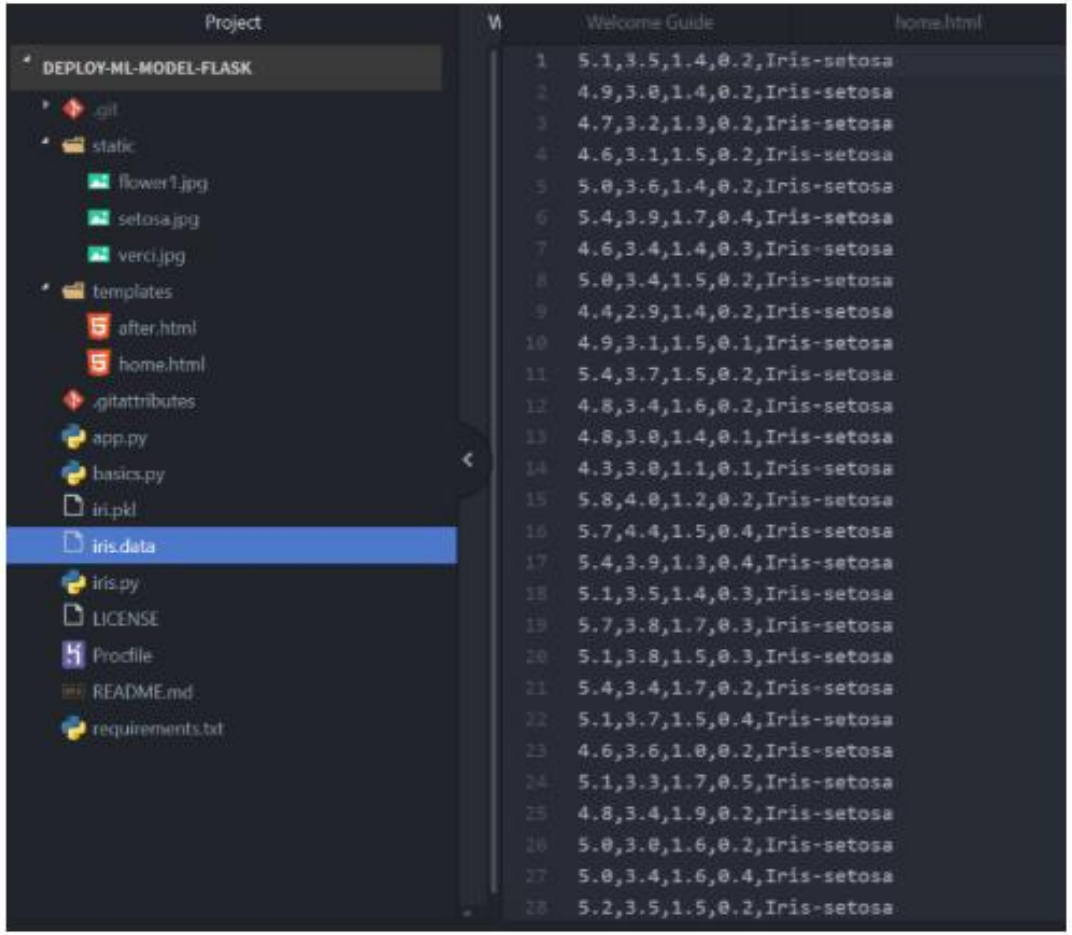
By,

Kelvin Mpofu

Submitted to:

Tasks Involved

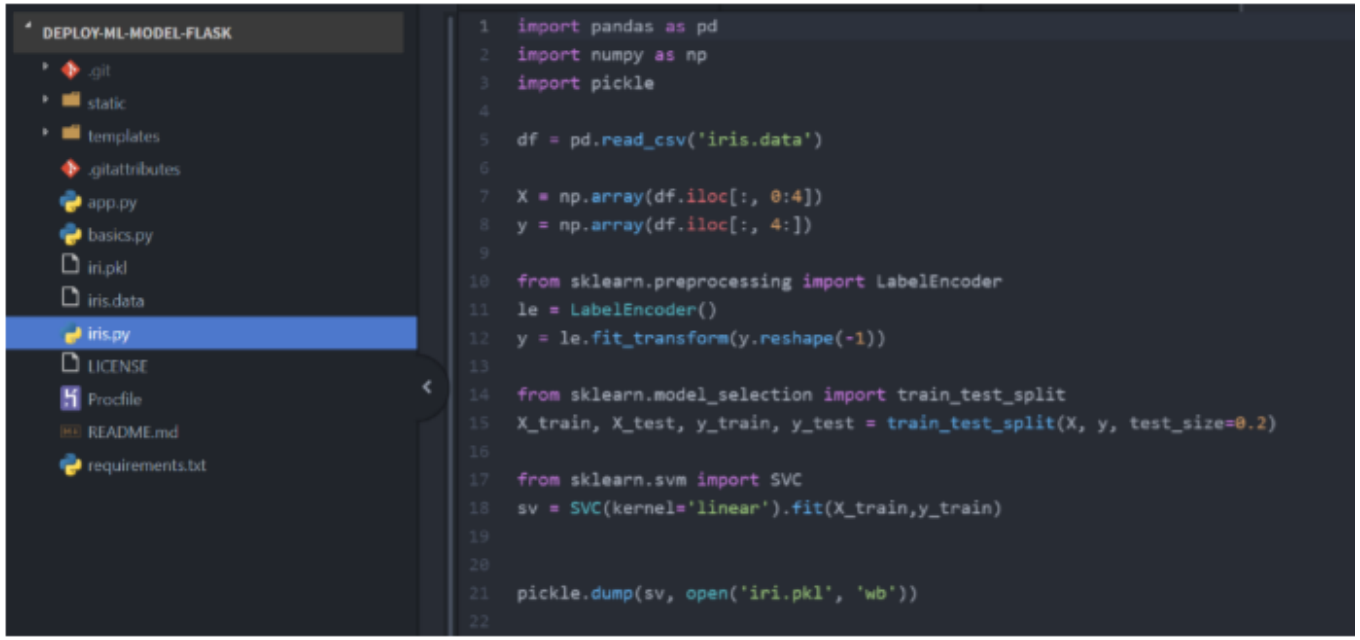
- The data set consists of 50 samples from each of three species of *Iris* (*Iris setosa*, *Iris virginica* and *Iris versicolor*). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.
- Trained a **support vector machine model** to classify the different types of flowers based on their features
- Finally, deployed Support vector machine model to a web-app using **Flask API**.
- This way, we can predict potential customers using the web-app.



Model Validation

- After splitting the data into train and test set, Support vector machine model was used for predicting on test set.

- The model achieved a high **accuracy of 82.5%.**

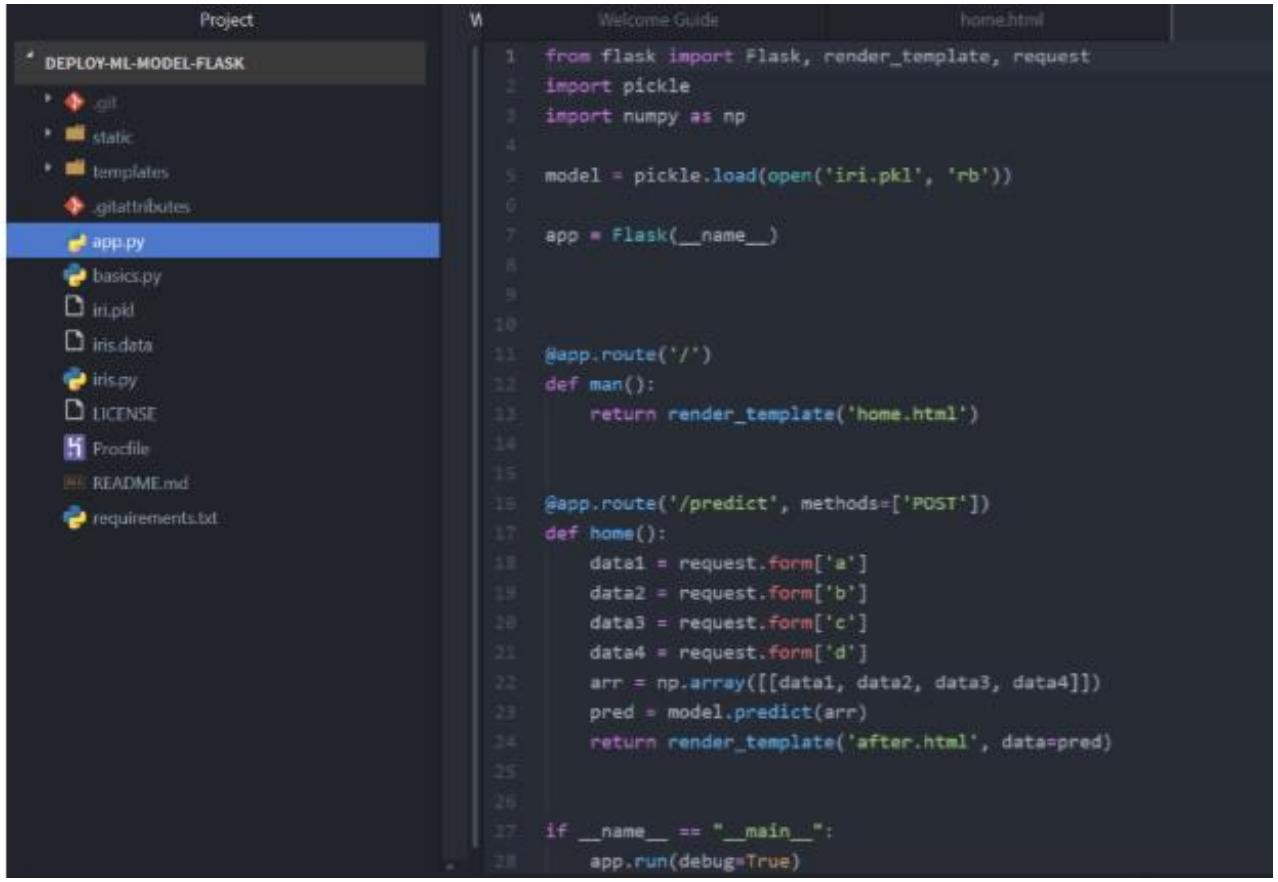


Saving Model & Creating Web-App using Flask API

- The model was then trained on the whole dataset before saving the model to **pickle format.**

- Pickling is done to **convert python object to character object.**

- Next, created a python file to create the web app using **Flask API** module.

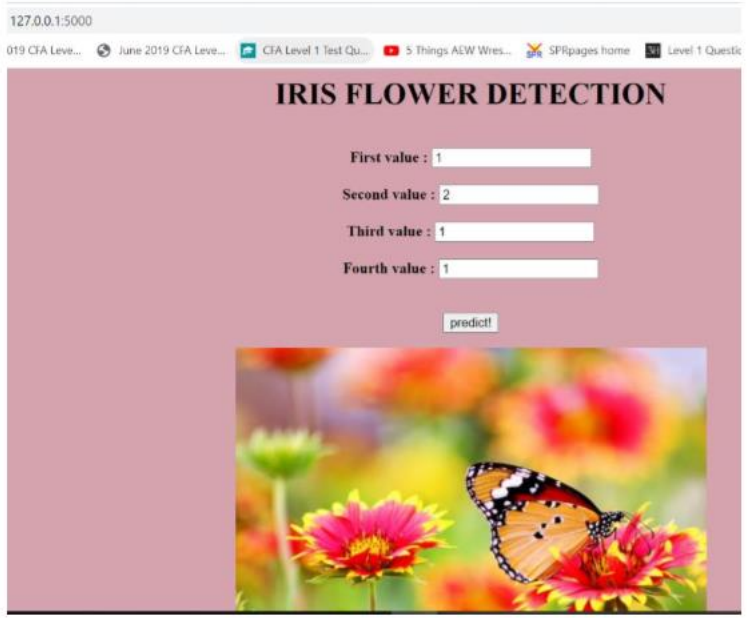


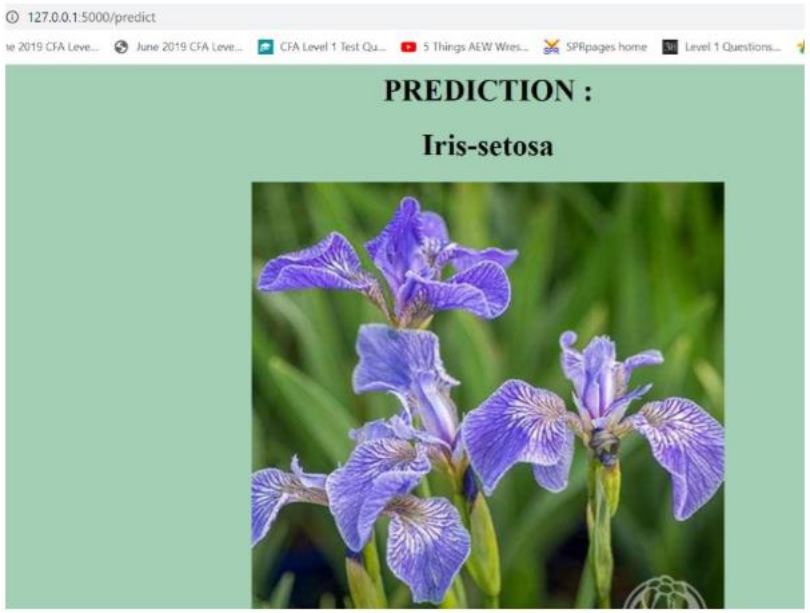
HTML

```html
<html>
    <body bgcolor=#d4a3ae>

        <center>

        <h1> IRIS FLOWER DETECTION </h1><br>

        <form method="POST", action="{{url_for('home')}}">
           <b> First value :  <input type="text", name='a', placeholder="enter 1"> <br><br>
            Second value :  <input type="text", name='b', placeholder="enter 2"> <br><br>
            Third value :  <input type="text", name='c', placeholder="enter 3"> <br><br>
            Fourth value : <input type="text", name='d', placeholder="enter 4"> <br><br><br></b>
            <input type="submit" , value='predict!' >
        </form>

        <img src='static\flower1.jpg' alt="flower">

    </center>

    </body>
</html>
```

Data Glacier

Model Deployment





- Finally, created the web-app and deployed the model into the web-app.

- Based on the model, we can now use the web application to predict the type of flower based on features.

The End