

Deployment on Flask (Week 4)

Name: Manoj Kumar Thangaraj

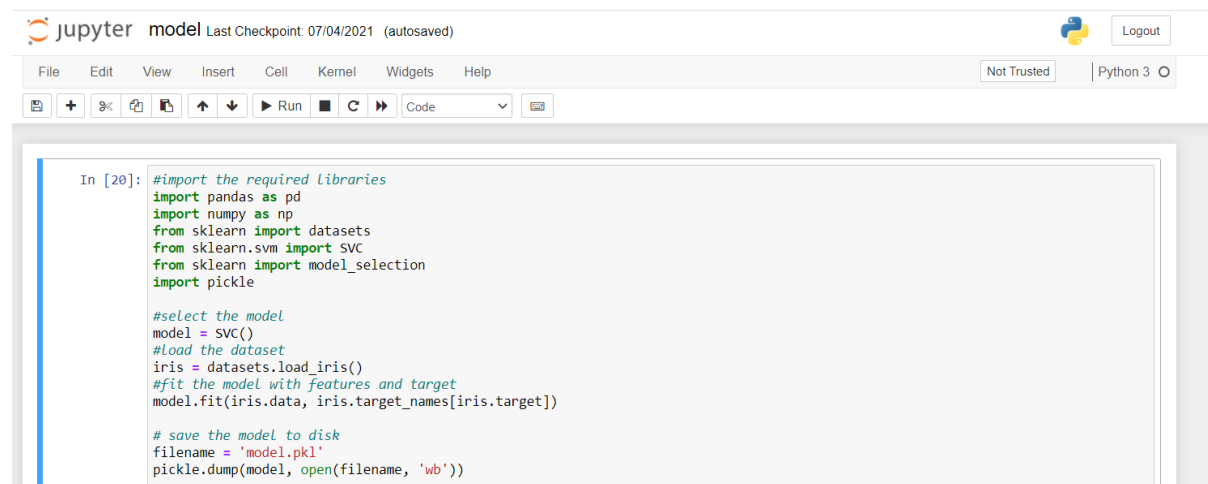
Batch Code: LISUM01

Submission Date: 04/07/2021

Task Objectives:

1. Select any toy data (simple data).
 2. Save the model
 3. Deploy the model on flask (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.
-

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

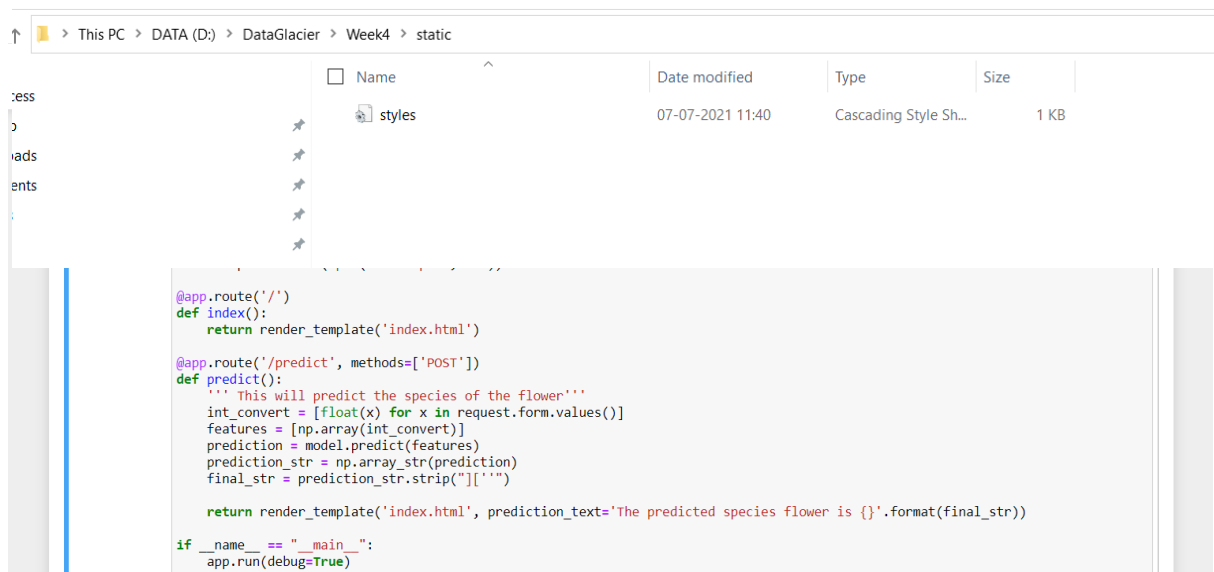


```
In [20]: #import the required libraries
import pandas as pd
import numpy as np
from sklearn import datasets
from sklearn.svm import SVC
from sklearn import model_selection
import pickle

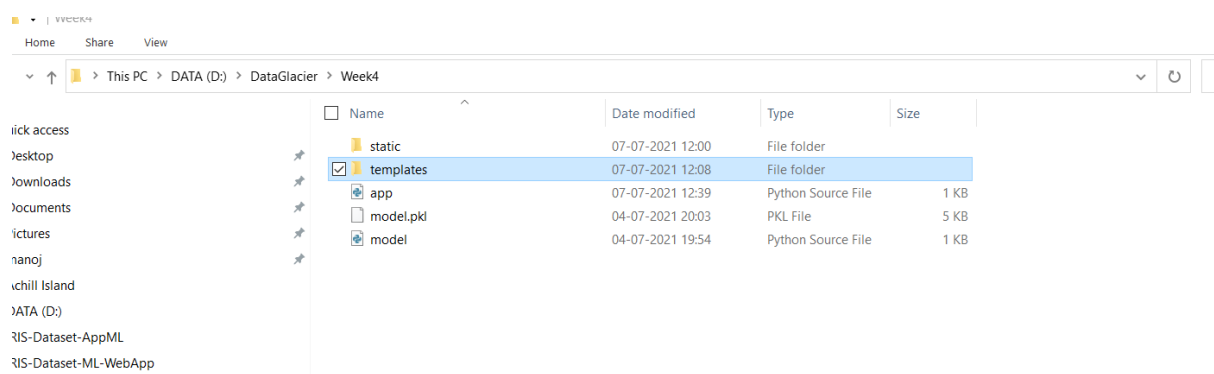
#select the model
model = SVC()
#Load the dataset
iris = datasets.load_iris()
#fit the model with features and target
model.fit(iris.data, iris.target_names[iris.target])

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

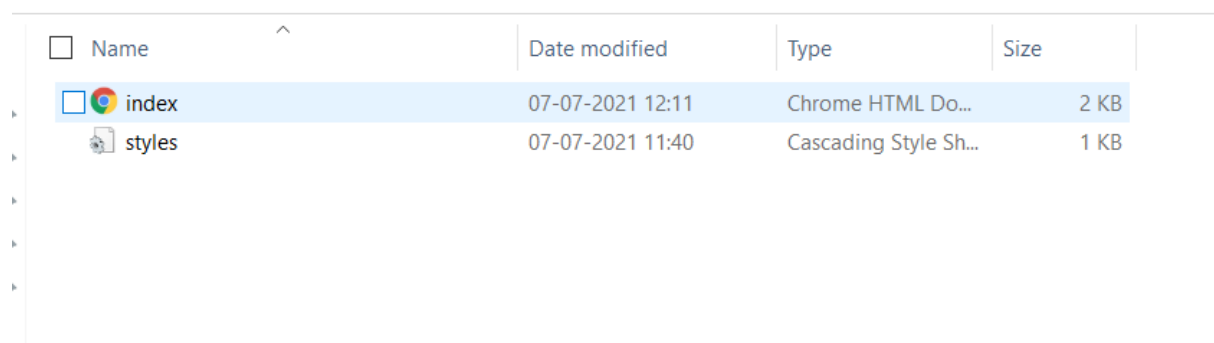
As per the above picture, the dataset is loaded straight from the scikit-learn library and machine learning model used is SVM. Once the model is fitted, the model is saved into a pickle file.



The 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.

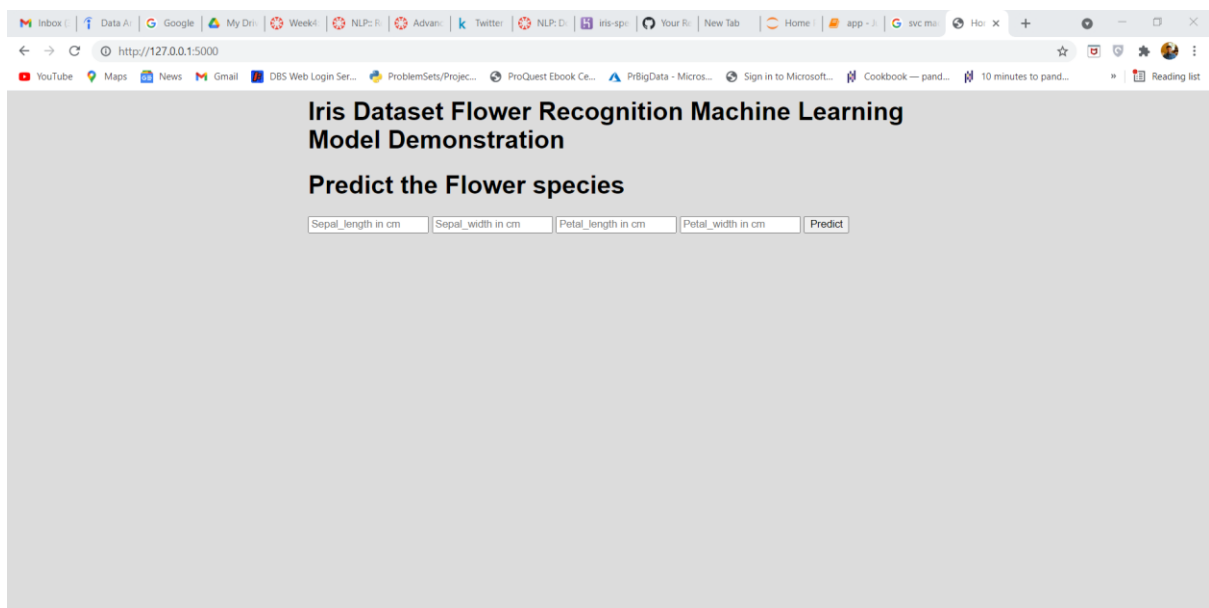


The week 4 folder consists of files app.py, model.pkl, model.py, templates folder contains the index.html file and style file in css format and static folder consist of style css file.

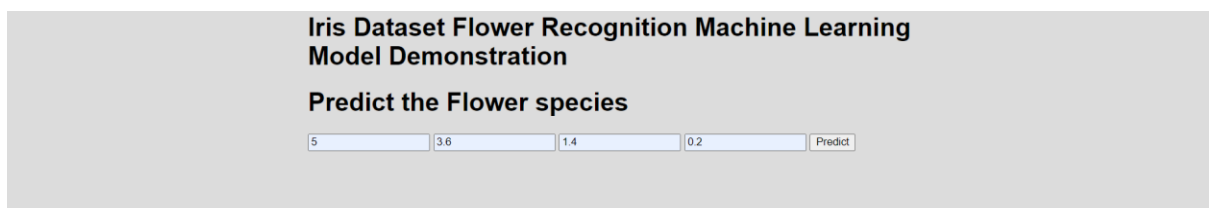


```
MINGW64:/d/DataGlacier/Week4
manoj@DESKTOP-TEE8Q4V MINGW64 /d/DataGlacier/Week4
$ python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with windowsapi reloader
* Debugger is active!
* Debugger PIN: 290-661-837
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

When we run the app.py in the git bash, it will trigger a local server with the http link. Once the link is fed in the web browser. The web app is deployed.



We can feed the values in the required fields for the prediction.



On clicking the predict button,

Iris Dataset Flower Recognition Machine Learning Model Demonstration

Predict the Flower species

The predicted species flower is setosa

Therefore, app.py is run, the web app is deployed on a local server, based on the trained model, app.py gives us prediction.