



**Data Glacier**

Your Deep Learning Partner

# Week-4 Assignment

## Deployment on Flask

Name : Dhvanilkumar Prajapati

Batch code : LISUM12

Submission date : 27<sup>th</sup> August 2022

Submitted to : Data Glacier

# Deployment

- Toy data: load\_iris
- Machine Learning Model Preparation
- HTML file for web application
- CSS file for Web Layout
- Model deployment using Flask



**Data Glacier**

Your Deep Learning Partner

# Machine Learning Model Preparation

- Iris dataset is selected as toy data to play with.
- This dataset is used to predict species of Iris flower.
- There are three species mentioned in the dataset as follows:
  1. Setosa
  2. Versicolor
  3. Verginica
- As it is a classification problem, K nearest neighbors algorithm is used to predict the species based on inputs.
- Inputs are as follows:

Sepal length, Sepal width, Petal length and Petal Width

# Machine Learning Model Preparation

---

```
# Loading data into the dataframe dataset
dataset = pd.DataFrame(load_iris().data, columns=load_iris().feature_names)
target_mapping={0:'setosa', 1:'versicolor', 2:'virginica'}
dataset['Target'] = load_iris()['target']
dataset['Target_name'] = dataset['Target'].map(target_mapping)

# Defining independent and dependent data
X = dataset.drop(columns=['Target', 'Target_name'])
Y = dataset['Target_name']

# Splitting data into test and train data
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=42)

# Standardize independent features
scaler = StandardScaler()
scaled = pd.DataFrame((scaler.fit_transform(X)), columns=X.columns)

# Implementing KNN algorithm
model = KNeighborsClassifier(n_neighbors=1)
model.fit(X_train, Y_train)
predicted = pd.DataFrame(model.predict(X_test))
category=['setosa', 'versicolor', 'virginica']
confusion_matrix = pd.DataFrame(confusion_matrix(Y_test, predicted), columns=category, index=category)
print(confusion_matrix)

# Saving the python code as Pickle File
pickle.dump(model, open('model.sav', 'wb'))
```

# HTML File for Web Application

---

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Week-4 Assignment</title>
    <link rel="stylesheet" type="text/css" href="static\predict.css">
  </head>
  <body>
    <h1><u>IRIS Species Prediction Model</u></h1>
    <br>
    <h3>Please enter below required values: </h3>
    <form action="{{ url_for('predict')}}" method="post">
      Sepal length (cm): <input type="text" name="sepal_length" required="required"/><br><br>
      Sepal width (cm): <input type="text" name="sepal_width" required="required" /><br><br>
      Petal length (cm): <input type="text" name="petal_length" required="required" /><br><br>
      Petal width (cm): <input type="text" name="petal_width" required="required" /><br><br>
      <input type="submit" value="Submit" />
    </form>
    <br>
    <h3>{{ prediction }}</h3>
  </body>
</html>
```



# IRIS Species Prediction Model

**Please enter below required values:**

Sepal length (cm):

Sepal width (cm):

Petal length (cm):

Petal width (cm):

**{{ prediction }}**

HTML  
Webpage

# CSS File for Web Layout

---

```
html{  
  background-image: linear-gradient(to left top, #051937, #004d7a, #008793, #00bf72, #a8eb12);  
  height:100%;  
}  
body{  
  color:white;  
  margin-top: 200px;  
  background-image:  
  url('https://www.freepik.com/premium-photo/texture-fabric-are-two-tone-color-background_2159300.htm');  
  text-align: center}
```

## IRIS Species Prediction Model

Please enter below required values:

Sepal length (cm):

Sepal width (cm):

Petal length (cm):

Petal width (cm):

# HTML & CSS



```
from flask import Flask, request, render_template
import pickle
import numpy as np

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

@app.route('/')
def home():
    print('start')
    return render_template('predict.html')

@app.route('/predict', methods=['POST'])
def predict():
    input_values = [float(x) for x in request.form.values()]
    features = [np.array(input_values)]
    output = model.predict(features)
    print('predict')
    return render_template('predict.html', prediction='Species should be {}'.format(output[0].upper()))

if __name__ == '__main__':
    app.run(port=5000, debug=True)
```

## Model Deployment using Flask

# Submitting Inputs

## IRIS Species Prediction Model

Please enter below required values:

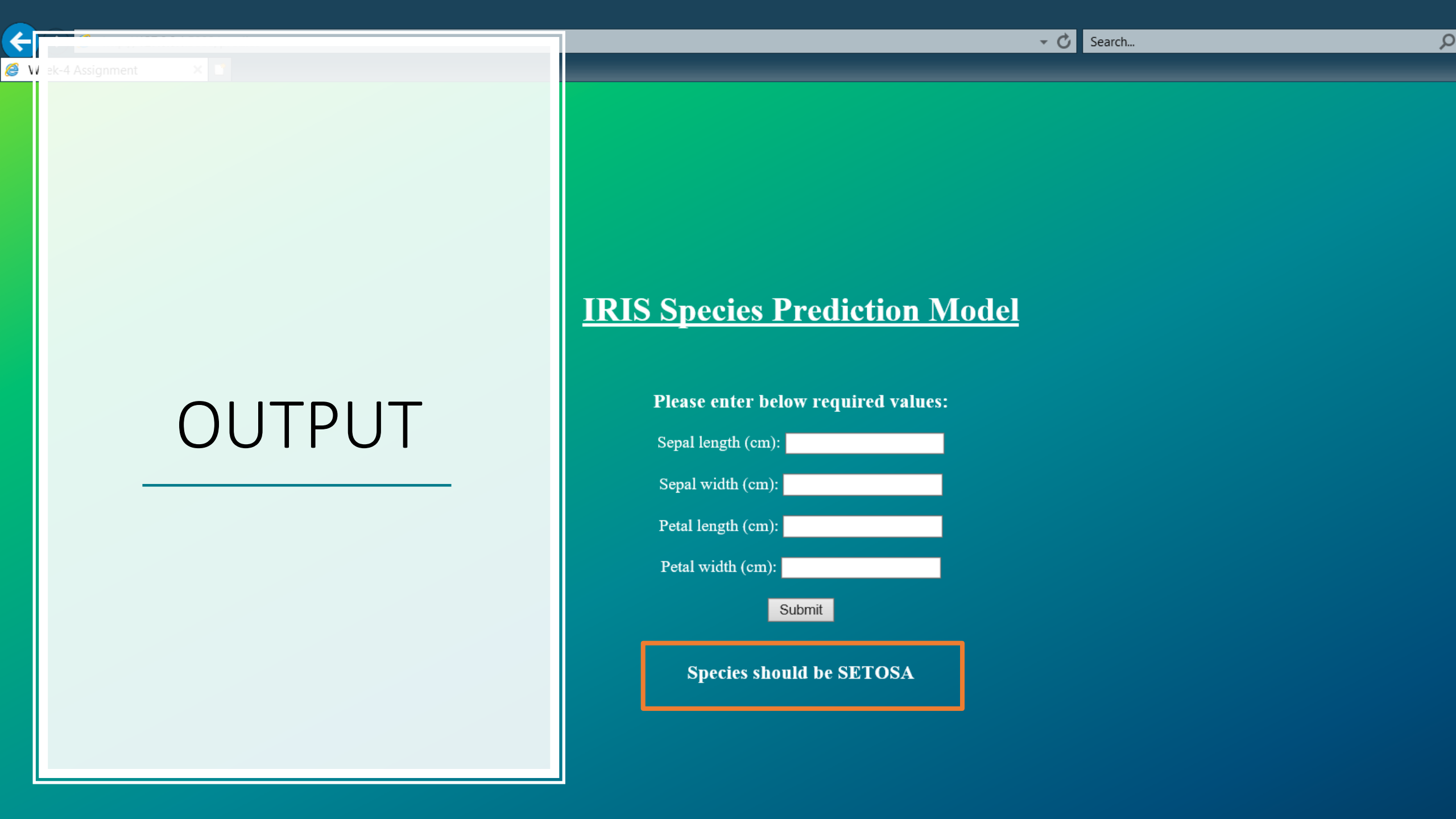
Sepal length (cm):

Sepal width (cm):

Petal length (cm):

Petal width (cm):  x

Submit



Week-4 Assignment



Search...



OUTPUT

## IRIS Species Prediction Model

Please enter below required values:

Sepal length (cm):

Sepal width (cm):

Petal length (cm):

Petal width (cm):

Submit

Species should be SETOSA

# Thank You



**Data Glacier**

Your Deep Learning Partner