

**Name:** Iris flower classification

**Report date:** 27-10-2023

**Internship Batch:** LISUM26

**Submitted to:** Data Glacier

**Intern:** Murat Kiran

---

## 1. app.py

```
from flask import Flask, request, render_template, redirect, url_for
import joblib
from model import train_and_save_model

app = Flask(__name__)

model, iris_data = train_and_save_model()

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

@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':
        try:
            sepal_length = float(request.form['sepal_length'])
            sepal_width = float(request.form['sepal_width'])
            petal_length = float(request.form['petal_length'])
            petal_width = float(request.form['petal_width'])

            features = [[sepal_length, sepal_width, petal_length, petal_width]]
            prediction = model.predict(features)
            target_name = iris_data.target_names[prediction[0]]

            return render_template('result.html', prediction=f'Predicted Iris
Species: {target_name}')
        except ValueError:
            return render_template('result.html', prediction='Invalid input. Please
enter valid numeric values.')

    return render_template('index.html')

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

## 2. Application

← → ↻ ⓘ 127.0.0.1:5000

### Model Prediction Form

Sepal Length:

Sepal Width:

Petal Length:

Petal Width:

← → ↻ ⓘ 127.0.0.1:5000

### Model Prediction Form

Sepal Length:

Sepal Width:

Petal Length:

Petal Width:

← → ↻ ⓘ 127.0.0.1:5000/predict

### Model Prediction Result

Predicted Iris Species: setosa

[Go Back](#)