Saif Amer
LISUM43
Submitted April 4th, 2025

Selecting Toy data and Saving Model:

```
# Import necessary libraries
import pandas as pd
import numpy as np
import joblib
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Load the iris dataset
iris = load_iris()
X = iris.data
y = iris.target

# Create a dataframe for better visualization
iris_df = pd.DataFrame(X, columns=iris.feature_names)
iris_df['target'] = y
iris_df['species'] = iris_df['target'].map({
    0: 'setosa',
    1: 'versicolor',
    2: 'virginica'
})

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Create and train a model
```

```
model = RandomForestClassifier(n_estimators=100,
random_state=42)
model.fit(X_train, y_train)

# Check accuracy
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy:.2f}")



# Save the model to disk
joblib.dump(model, 'iris_model.pkl')

# To test that it worked, you can reload it
loaded_model = joblib.load('iris_model.pkl')
test_pred = loaded_model.predict(X_test)
print(f"Loaded Model Accuracy: {accuracy_score(y_test,
test_pred):.2f}")
```

Deploying the Model:

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

app = Flask(__name__)

# Load the model
model = joblib.load('iris_model.pkl')

@app.route('/')
```

```
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    # Get the input values from the form
    features = [float(request.form.get(f)) for f in [
        'sepal_length', 'sepal_width', 'petal_length',
'petal_width'
    ]]

    # Convert to numpy array
    final_features = np.array(features).reshape(1, -1)

    # Make prediction
    prediction = model.predict(final_features)

    # Get the species name
    species_dict = {0: 'Setosa', 1: 'Versicolor', 2: 'Virginica'}
    result = species_dict[prediction[0]]

    return render_template('result.html', prediction=result,
features=features)

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

```
<!DOCTYPE html>
```

```
<html>
<head>
   <title>Iris Flower Prediction</title>
   <link rel="stylesheet" href="{{ url_for('static',
filename='css/style.css') }}">
</head>
<body>
   <div class="container">
       <h1>Iris Flower Species Predictor</h1>
       <form action="{{ url_for('predict') }}" method="post">
           <div class="form-group">
               <label for="sepal_length">Sepal Length
(cm):</label>
               <input type="number" step="0.1"
name="sepal_length" id="sepal_length" required>
           </div>
           <div class="form-group">
               <label for="sepal_width">Sepal Width
(cm):</label>
               <input type="number" step="0.1"
name="sepal_width" id="sepal_width" required>
           </div>
           <div class="form-group">
               <label for="petal_length">Petal Length
(cm):</label>
               <input type="number" step="0.1"
name="petal_length" id="petal_length" required>
           </div>
           <div class="form-group">
               <label for="petal_width">Petal Width
(cm):</label>
```

```
                <input type="number" step="0.1"
name="petal_width" id="petal_width" required>
            </div>
            <button type="submit">Predict Species</button>
        </form>
    </div>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Prediction Result</title>
    <link rel="stylesheet" href="{{ url_for('static',
filename='css/style.css') }}">
</head>
<body>
    <div class="container">
        <h1>Prediction Result</h1>
        <div class="result-box">
            <h2>The Iris flower is predicted to be: <span
class="prediction">{{ prediction }}</span></h2>
            <h3>Input Features:</h3>
            <ul>
                <li>Sepal Length: {{ features[0] }} cm</li>
                <li>Sepal Width: {{ features[1] }} cm</li>
                <li>Petal Length: {{ features[2] }} cm</li>
                <li>Petal Width: {{ features[3] }} cm</li>
            </ul>
```

```
            <a href="{{ url_for('home') }}" class="btn">Try
Again</a>
        </div>
    </div>
</body>
</html>
```