Model Deployment Report

Name: Manil Shangle
Batch Code: LISUM43
Submission Date: 03/31/2025
Submitted To: Data Glacier

1. Introduction

This report details the process of deploying a machine learning model using Flask. The project includes data preprocessing, model training, deployment via Flask, and a web-based user interface for making predictions. The application allows users to input features and receive predictions with visual insights.

2. Model Training

2.1 Data Preprocessing

- The dataset was cleaned and preprocessed using Pandas and NumPy.
- Features were selected based on their relevance to classification.

2.2 Model Selection and Training

- The model was trained using **XGBoost** due to its efficiency and accuracy.
- Hyperparameters were tuned for optimal performance.
- Below is a snapshot of the training process:

```
import pandas as pd
import numpy as np
from xgboost import XGBClassifier
import pickle

# Load dataset
df = pd.read_csv("dataset.csv")
```

```
# Define features and target
X = df[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']]
y = df['species']

# Train model
model = XGBClassifier()
model.fit(X, y)

# Save the model
pickle.dump(model, open("model.pkl", "wb"))
```

---

3. Flask Application Deployment

3.1 Setting Up Flask App

A Flask application was developed to serve the trained model and provide a web-based UI for predictions.

3.2 Flask App Code

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

app = Flask(__name__)
model = pickle.load(open("model.pkl", "rb"))

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

@app.route('/predict', methods=['POST'])
def predict():
    features = [float(x) for x in request.form.values()]
    prediction = model.predict([features])[0]
    return render_template('index.html', prediction_text=f'Predicted species: {prediction}')

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

---

4. Web Interface Implementation

A web interface was designed to allow users to input feature values and receive predictions. The UI also includes a visualization of input values.

Features:

- User-friendly form inputs for entering sepal and petal dimensions.
- Prediction display with a **clear species name** instead of a numerical class.
- A **bar chart** displaying input feature values.

---

5. Testing and Validation

The application was tested for:

- **Correct Predictions:** Ensured output matches expected results from test data.
- **UI Functionality:** Verified input handling and prediction display.
- **Performance:** Evaluated response time and efficiency.

---

6. Conclusion

This project successfully deployed an ML model using Flask, allowing users to predict flower species based on sepal and petal measurements. The application provides **accurate predictions**, a **user-friendly interface**, and **visual insights** for a better experience.

Future Improvements:

- Improve UI aesthetics for better user engagement.
- Implement a REST API for broader usability.
- Deploy the model on a cloud platform for scalability.

End of Report