University of Central Punjab

Faculty of Information Technology

Applied Machine Learning (Assignment # 2)

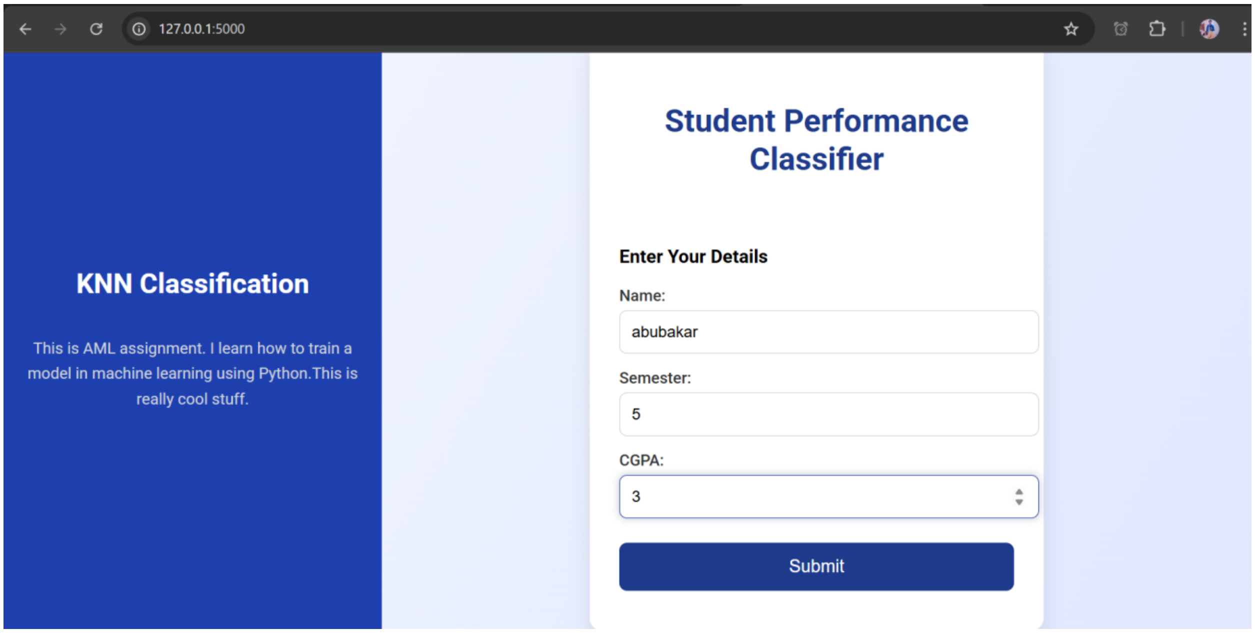
Course Code: SEAD3463	**Marks:** 30	**Semester:** Fall 2025
Submission Date:	**Section:**	**Course Teacher:**

Important Instruction:

1. Please do assignment on your own, copied (partial / full) material will be strictly marked zero. You need to write all details as per your experiment.
2. Please strictly follow the guidelines provided in the questions, otherwise marks will be deducted.
3. Soft copy of the assignment must be uploaded well before the deadline and hard copy (printed) must be submitted in the class.

Name: Abu bakar Rasheed
Roll:L1f23bsse0371

Github_Repo=



127.0.0.1:5000

KNN Classification

This is AML assignment. I learn how to train a model in machine learning using Python.This is really cool stuff.

Student Performance Classifier

Enter Your Details

Name: abubakar

Semester: 5

CGPA: 3

KNN Prediction Result

You entered:

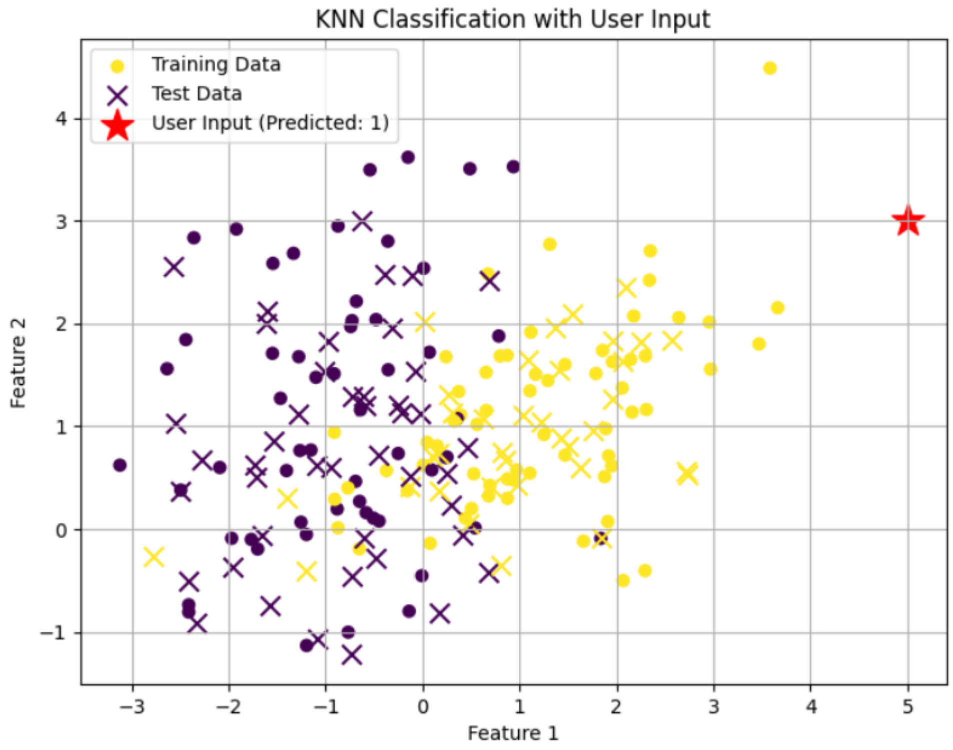
- **Feature 1:** 5.0
- **Feature 2:** 3.0

Predicted Class: 1

Model Evaluation Metrics (on Test Set)

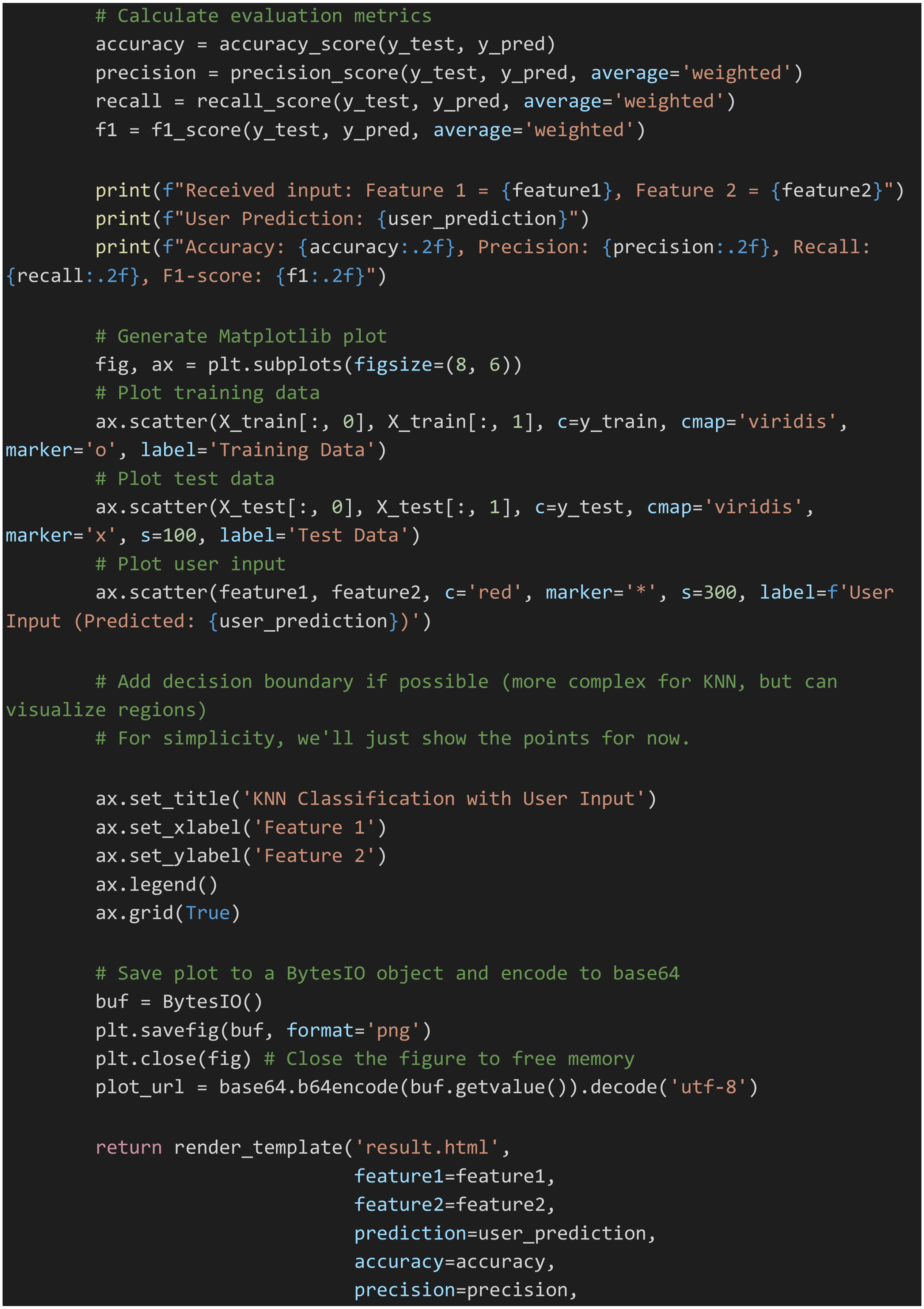
Accuracy: 0.79
Precision (Weighted): 0.79
Recall (Weighted): 0.79
F1-Score (Weighted): 0.79

Classification Plot



CODE:
APP.PY

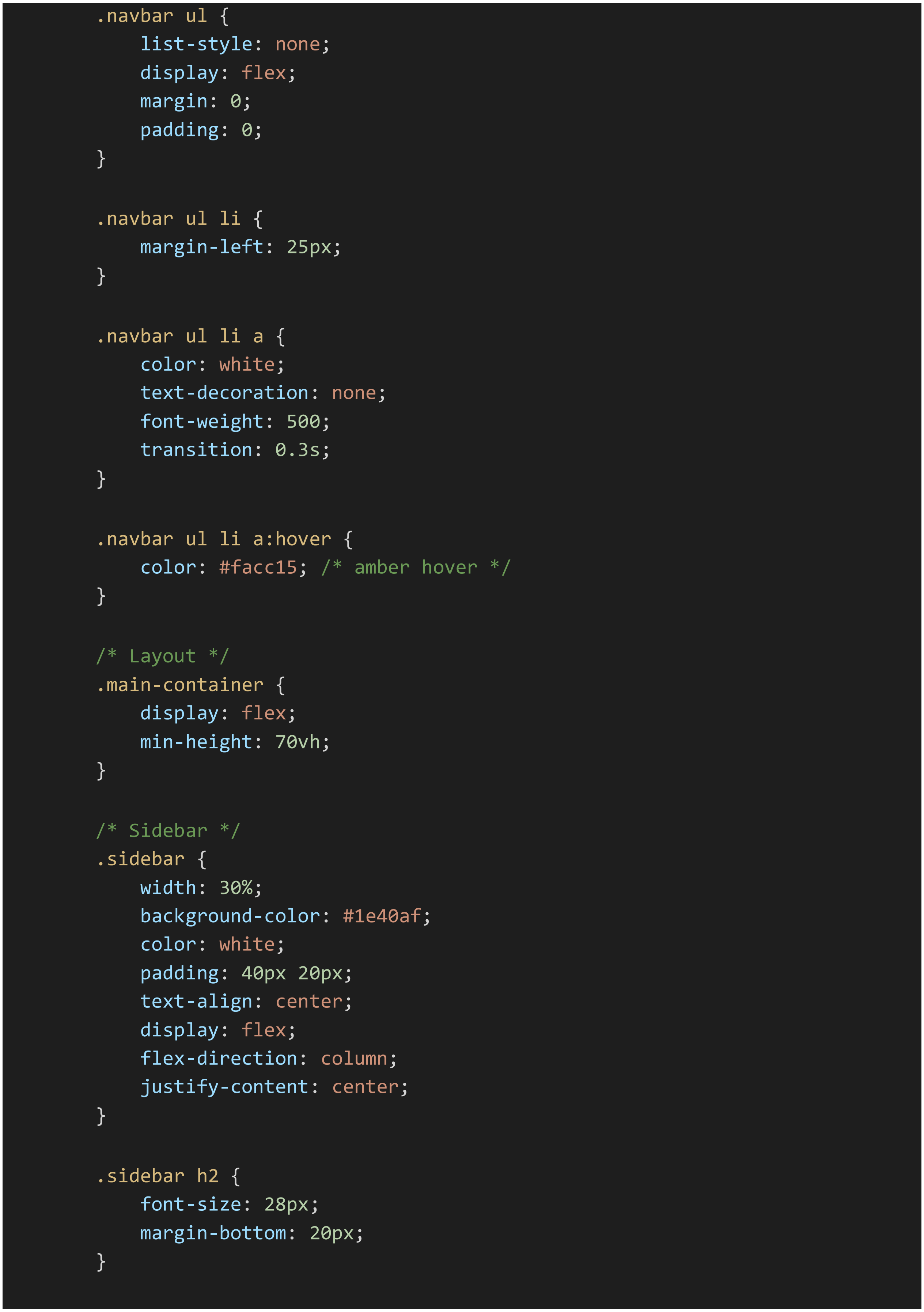
```
from flask import Flask, render_template, request
import joblib
import os
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score
import numpy as np
import matplotlib.pyplot as plt
import base64
from io import BytesIO

# Initialize the Flask app
app = Flask(__name__)

# Load the KNN model and test data
model_dir = 'model_data'
knn_model = joblib.load(os.path.join(model_dir, 'knn_model.joblib'))
X_test = joblib.load(os.path.join(model_dir, 'X_test.joblib'))
y_test = joblib.load(os.path.join(model_dir, 'y_test.joblib'))
X_train = joblib.load(os.path.join(model_dir, 'X_train.joblib')) # Added this
line
y_train = joblib.load(os.path.join(model_dir, 'y_train.joblib')) # Added this
line

# Define the home page route
@app.route('/')
def home():
    return render_template('index.html')

# Define the prediction route to handle form submissions
@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':
        feature1 = float(request.form['feature1'])
        feature2 = float(request.form['feature2'])

        # Make prediction for user input
        user_input = np.array([[feature1, feature2]])
        user_prediction = knn_model.predict(user_input)[0]

        # Make predictions on the test set for evaluation metrics
        y_pred = knn_model.predict(X_test)
```



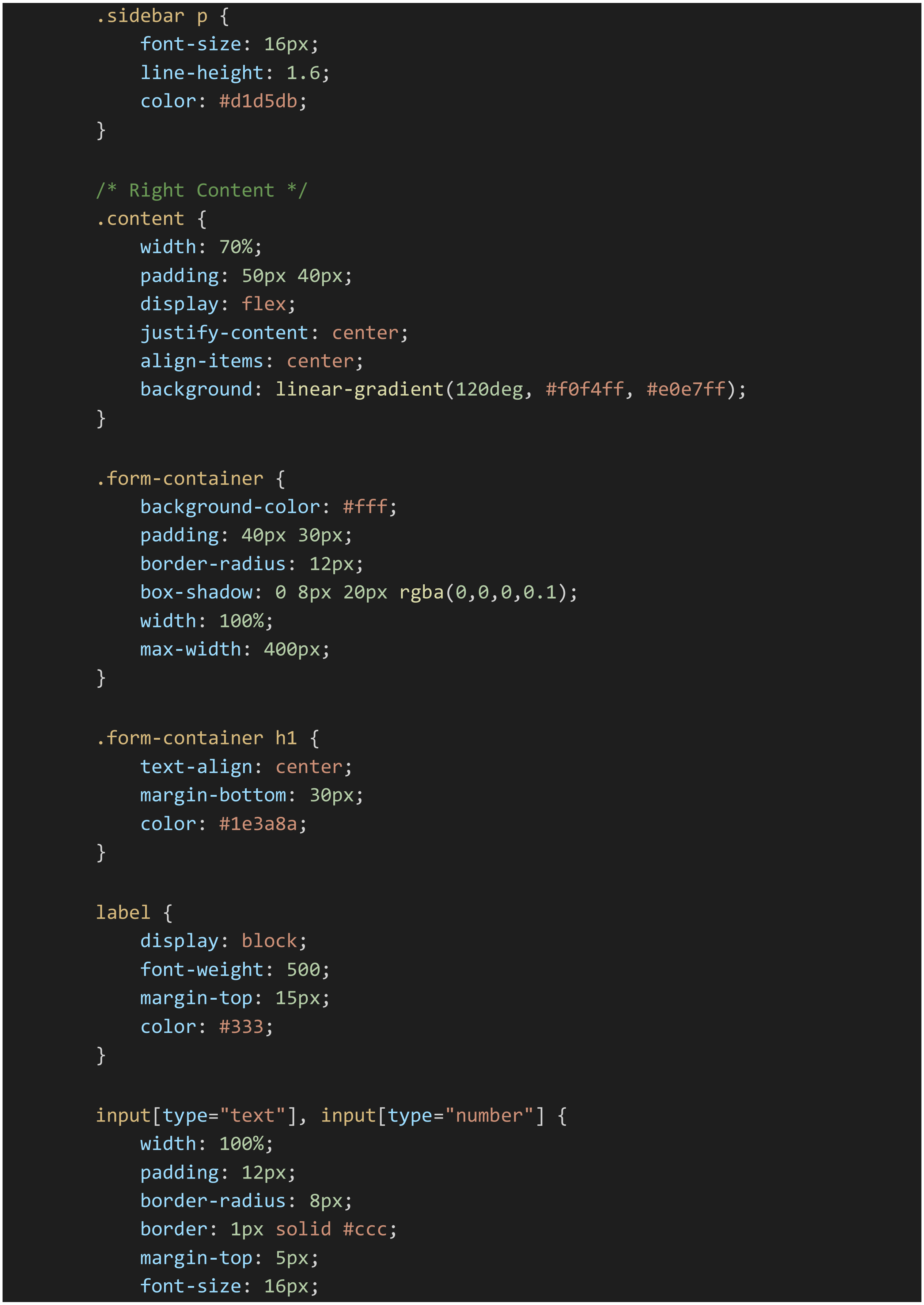
```
        # Calculate evaluation metrics
        accuracy = accuracy_score(y_test, y_pred)
        precision = precision_score(y_test, y_pred, average='weighted')
        recall = recall_score(y_test, y_pred, average='weighted')
        f1 = f1_score(y_test, y_pred, average='weighted')

        print(f"Received input: Feature 1 = {feature1}, Feature 2 = {feature2}")
        print(f"User Prediction: {user_prediction}")
        print(f"Accuracy: {accuracy:.2f}, Precision: {precision:.2f}, Recall:
{recall:.2f}, F1-score: {f1:.2f}")

        # Generate Matplotlib plot
        fig, ax = plt.subplots(figsize=(8, 6))
        # Plot training data
        ax.scatter(X_train[:, 0], X_train[:, 1], c=y_train, cmap='viridis',
marker='o', label='Training Data')
        # Plot test data
        ax.scatter(X_test[:, 0], X_test[:, 1], c=y_test, cmap='viridis',
marker='x', s=100, label='Test Data')
        # Plot user input
        ax.scatter(feature1, feature2, c='red', marker='*', s=300, label=f'User
Input (Predicted: {user_prediction})')

        # Add decision boundary if possible (more complex for KNN, but can
visualize regions)
        # For simplicity, we'll just show the points for now.

        ax.set_title('KNN Classification with User Input')
        ax.set_xlabel('Feature 1')
        ax.set_ylabel('Feature 2')
        ax.legend()
        ax.grid(True)

        # Save plot to a BytesIO object and encode to base64
        buf = BytesIO()
        plt.savefig(buf, format='png')
        plt.close(fig) # Close the figure to free memory
        plot_url = base64.b64encode(buf.getvalue()).decode('utf-8')

        return render_template('result.html',
                               feature1=feature1,
                               feature2=feature2,
                               prediction=user_prediction,
                               accuracy=accuracy,
                               precision=precision,
                               recall=recall,
                               f1=f1,
                               plot_url=plot_url)

# Run the Flask application
if __name__ == '__main__':
    app.run(debug=True)
```

Index.html & CSS :

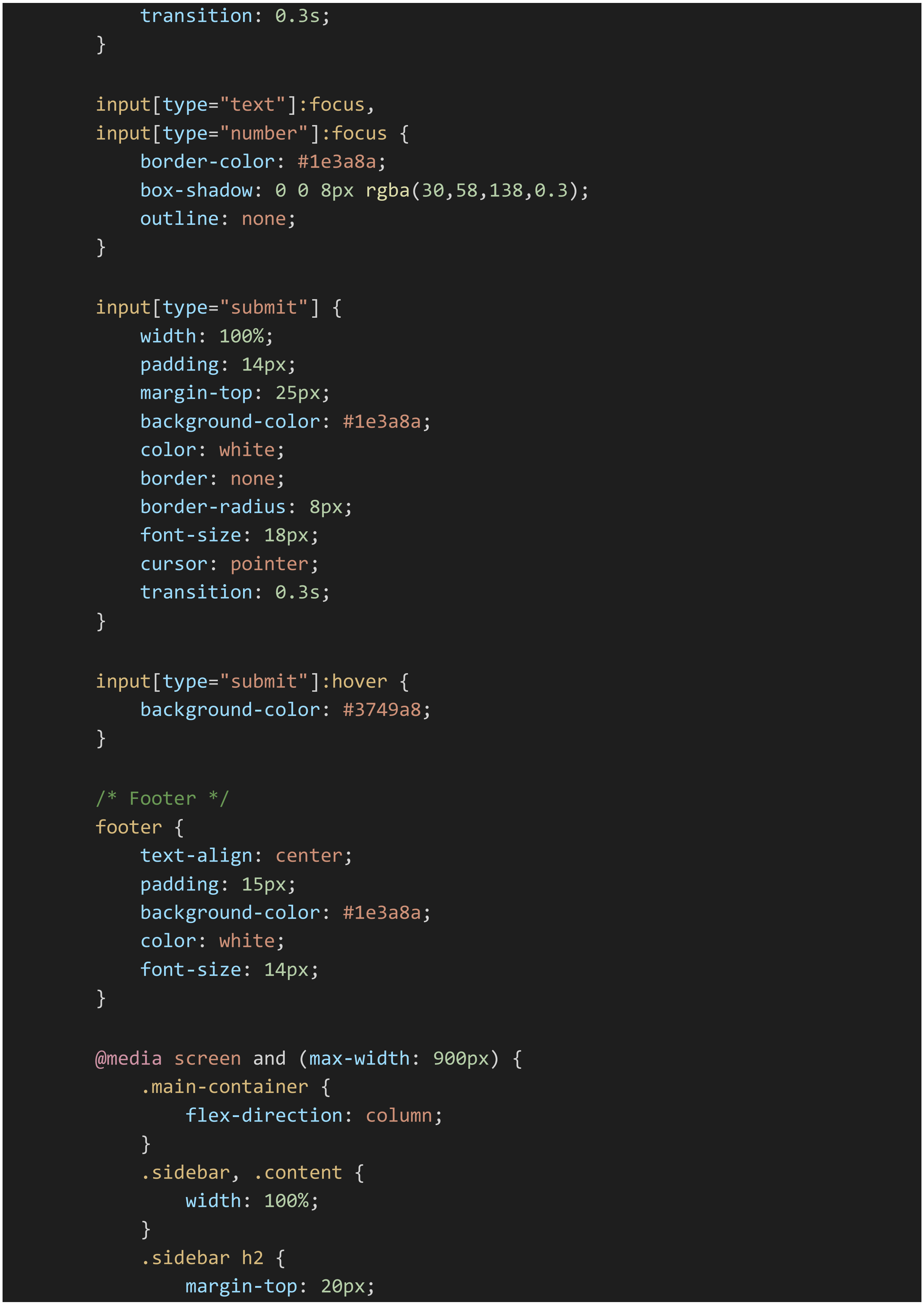
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>KNN Classification</title>
    <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700&display=swap" rel="stylesheet">
    <style>
        body {
            margin: 0;
            font-family: 'Roboto', sans-serif;
            background-color: #f0f2f5;
            height: 70vh;
        }

        /* Navbar */
        .navbar {
            display: flex;
            justify-content: center;
            align-items: center;
            background-color: #1e3a8a;
            color: white;
            padding: 15px 0;
            box-shadow: 0 4px 8px rgba(0,0,0,0.1);
            font-weight: 500;
        }

        .navbar .logo {
            font-weight: 700;
            font-size: 22px;
            margin-right: 40px;
        }
```



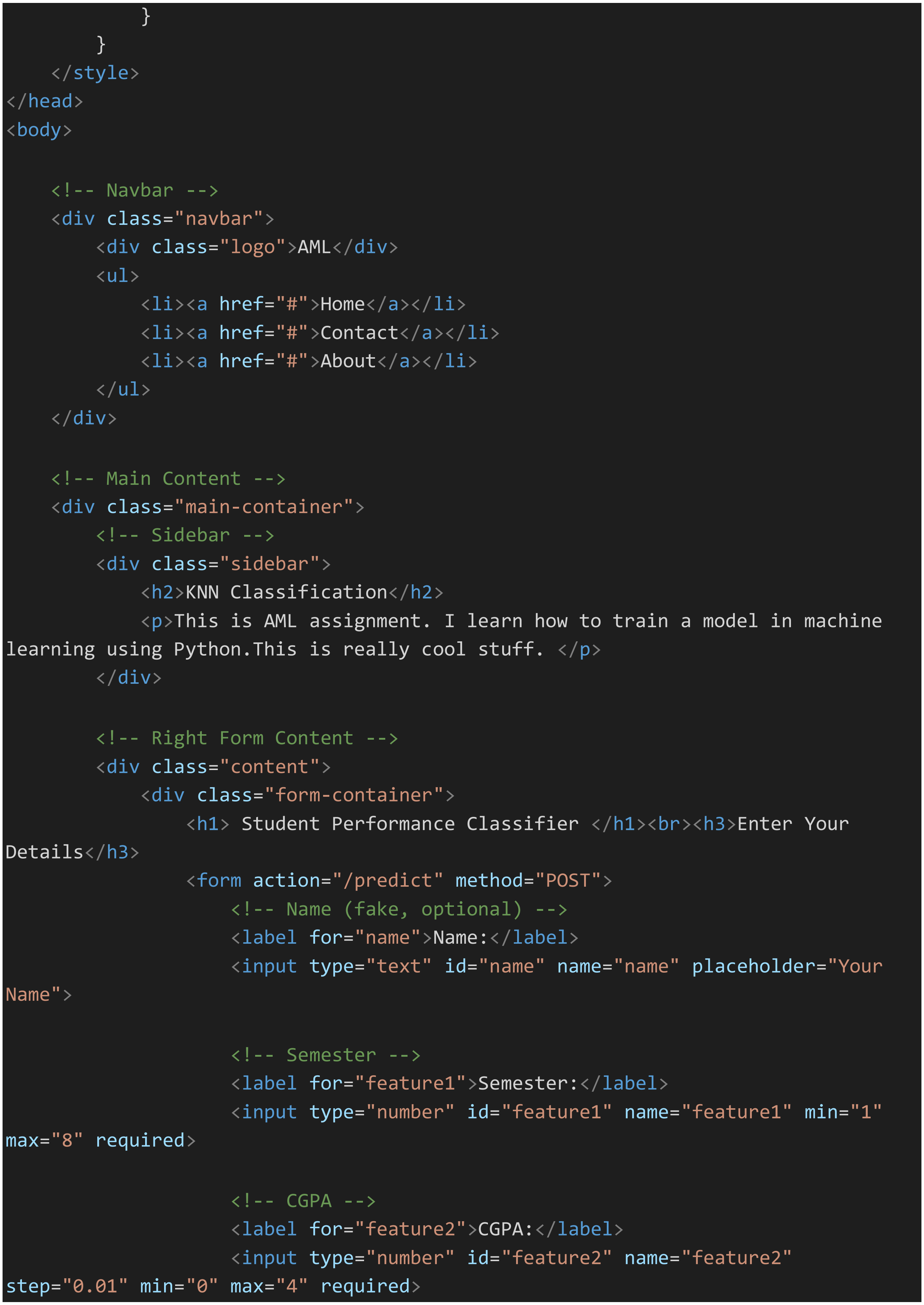
```
        .navbar ul {
            list-style: none;
            display: flex;
            margin: 0;
            padding: 0;
        }

        .navbar ul li {
            margin-left: 25px;
        }

        .navbar ul li a {
            color: white;
            text-decoration: none;
            font-weight: 500;
            transition: 0.3s;
        }

        .navbar ul li a:hover {
            color: #facc15; /* amber hover */
        }

        /* Layout */
        .main-container {
            display: flex;
            min-height: 70vh;
        }

        /* Sidebar */
        .sidebar {
            width: 30%;
            background-color: #1e40af;
            color: white;
            padding: 40px 20px;
            text-align: center;
            display: flex;
            flex-direction: column;
            justify-content: center;
        }

        .sidebar h2 {
            font-size: 28px;
            margin-bottom: 20px;
        }
```



```css
        .sidebar p {
            font-size: 16px;
            line-height: 1.6;
            color: #d1d5db;
        }

        /* Right Content */
        .content {
            width: 70%;
            padding: 50px 40px;
            display: flex;
            justify-content: center;
            align-items: center;
            background: linear-gradient(120deg, #f0f4ff, #e0e7ff);
        }

        .form-container {
            background-color: #fff;
            padding: 40px 30px;
            border-radius: 12px;
            box-shadow: 0 8px 20px rgba(0,0,0,0.1);
            width: 100%;
            max-width: 400px;
        }

        .form-container h1 {
            text-align: center;
            margin-bottom: 30px;
            color: #1e3a8a;
        }

        label {
            display: block;
            font-weight: 500;
            margin-top: 15px;
            color: #333;
        }

        input[type="text"], input[type="number"] {
            width: 100%;
            padding: 12px;
            border-radius: 8px;
            border: 1px solid #ccc;
            margin-top: 5px;
            font-size: 16px;
```



```css
    transition: 0.3s;
}

input[type="text"]:focus,
input[type="number"]:focus {
    border-color: #1e3a8a;
    box-shadow: 0 0 8px rgba(30,58,138,0.3);
    outline: none;
}

input[type="submit"] {
    width: 100%;
    padding: 14px;
    margin-top: 25px;
    background-color: #1e3a8a;
    color: white;
    border: none;
    border-radius: 8px;
    font-size: 18px;
    cursor: pointer;
    transition: 0.3s;
}

input[type="submit"]:hover {
    background-color: #3749a8;
}

/* Footer */
footer {
    text-align: center;
    padding: 15px;
    background-color: #1e3a8a;
    color: white;
    font-size: 14px;
}

@media screen and (max-width: 900px) {
    .main-container {
        flex-direction: column;
    }
    .sidebar, .content {
        width: 100%;
    }
    .sidebar h2 {
        margin-top: 20px;
```

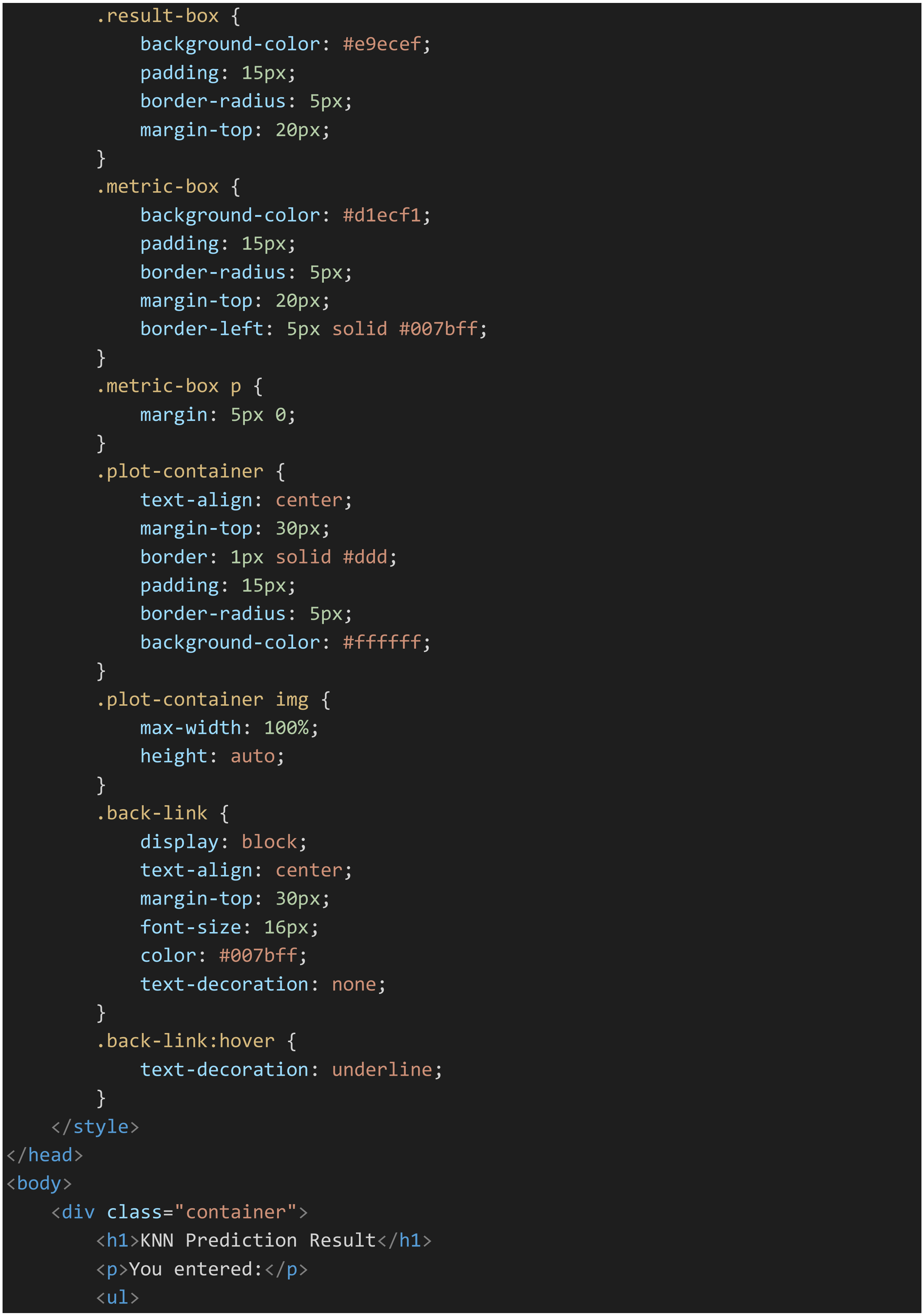


```
            }
        }
    </style>
</head>
<body>

    <!-- Navbar -->
    <div class="navbar">
        <div class="logo">AML</div>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">Contact</a></li>
            <li><a href="#">About</a></li>
        </ul>
    </div>

    <!-- Main Content -->
    <div class="main-container">
        <!-- Sidebar -->
        <div class="sidebar">
            <h2>KNN Classification</h2>
            <p>This is AML assignment. I learn how to train a model in machine
learning using Python.This is really cool stuff. </p>
        </div>

        <!-- Right Form Content -->
        <div class="content">
            <div class="form-container">
                <h1> Student Performance Classifier </h1><br><h3>Enter Your
Details</h3>
                <form action="/predict" method="POST">
                    <!-- Name (fake, optional) -->
                    <label for="name">Name:</label>
                    <input type="text" id="name" name="name" placeholder="Your
Name">

                    <!-- Semester -->
                    <label for="feature1">Semester:</label>
                    <input type="number" id="feature1" name="feature1" min="1"
max="8" required>

                    <!-- CGPA -->
                    <label for="feature2">CGPA:</label>
                    <input type="number" id="feature2" name="feature2"
step="0.01" min="0" max="4" required>
```

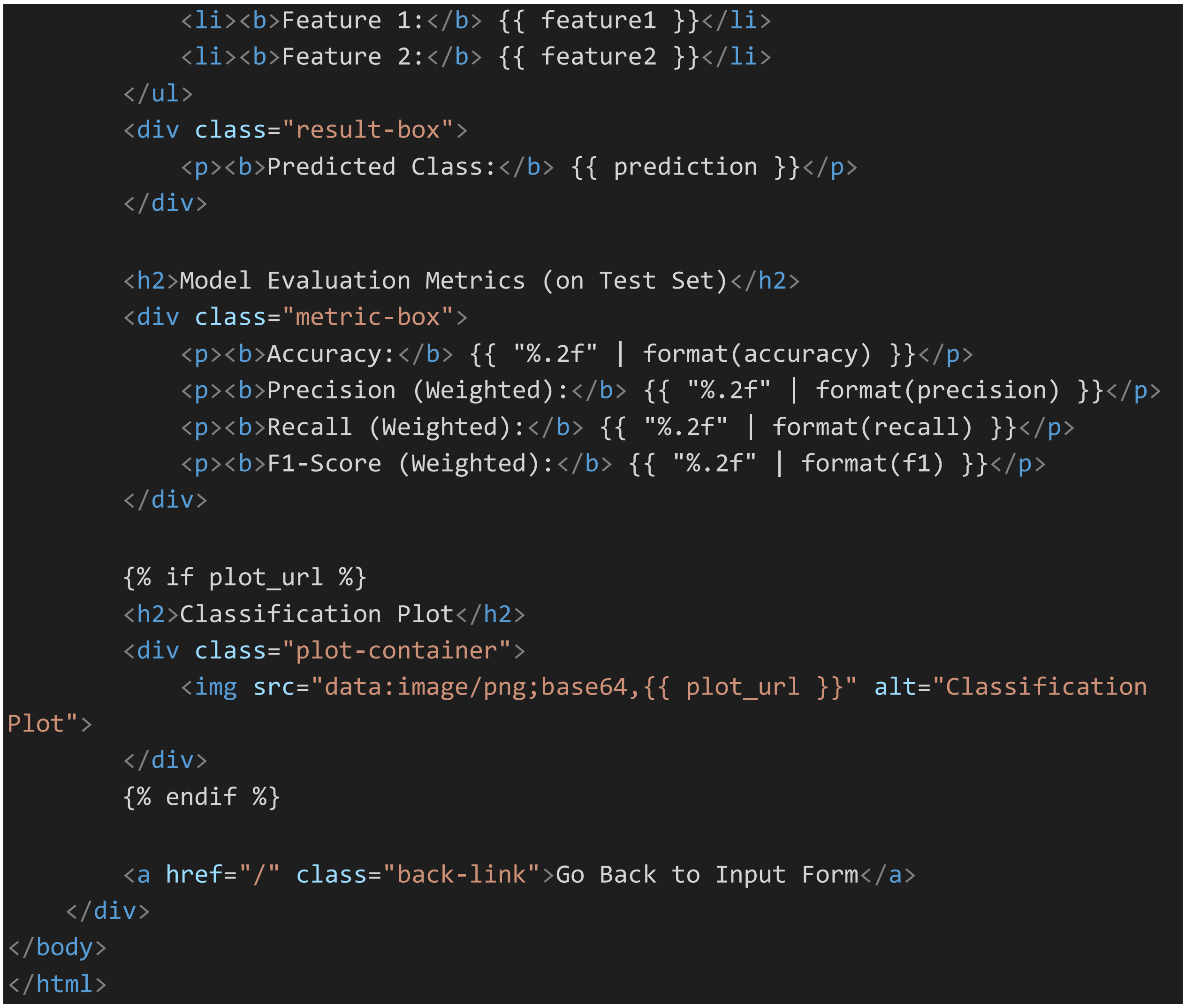
```
                    <input type="submit" value="Submit">
                </form>
            </div>
        </div>
    </div>

    <!-- Footer -->
    <footer>
        Developed by Abubakar
    </footer>

</body>
</html>
```

Reseult.html & CSS

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>KNN Prediction Result</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
            background-color: #f4f4f4;
        }
        .container {
            background-color: #fff;
            padding: 30px;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            max-width: 800px; /* Increased max-width for plot */
            margin: auto;
        }
        h1 {
            text-align: center;
            color: #333;
        }
        p {
            font-size: 18px;
            margin-bottom: 10px;
        }
```



```
        .result-box {
            background-color: #e9ecef;
            padding: 15px;
            border-radius: 5px;
            margin-top: 20px;
        }
        .metric-box {
            background-color: #d1ecf1;
            padding: 15px;
            border-radius: 5px;
            margin-top: 20px;
            border-left: 5px solid #007bff;
        }
        .metric-box p {
            margin: 5px 0;
        }
        .plot-container {
            text-align: center;
            margin-top: 30px;
            border: 1px solid #ddd;
            padding: 15px;
            border-radius: 5px;
            background-color: #ffffff;
        }
        .plot-container img {
            max-width: 100%;
            height: auto;
        }
        .back-link {
            display: block;
            text-align: center;
            margin-top: 30px;
            font-size: 16px;
            color: #007bff;
            text-decoration: none;
        }
        .back-link:hover {
            text-decoration: underline;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>KNN Prediction Result</h1>
        <p>You entered:</p>
        <ul>
```



```html
            <li><b>Feature 1:</b> {{ feature1 }}</li>
            <li><b>Feature 2:</b> {{ feature2 }}</li>
        </ul>
        <div class="result-box">
            <p><b>Predicted Class:</b> {{ prediction }}</p>
        </div>

        <h2>Model Evaluation Metrics (on Test Set)</h2>
        <div class="metric-box">
            <p><b>Accuracy:</b> {{ "%.2f" | format(accuracy) }}</p>
            <p><b>Precision (Weighted):</b> {{ "%.2f" | format(precision) }}</p>
            <p><b>Recall (Weighted):</b> {{ "%.2f" | format(recall) }}</p>
            <p><b>F1-Score (Weighted):</b> {{ "%.2f" | format(f1) }}</p>
        </div>

        {% if plot_url %}
        <h2>Classification Plot</h2>
        <div class="plot-container">
            <img src="data:image/png;base64,{{ plot_url }}" alt="Classification Plot">
        </div>
        {% endif %}

        <a href="/" class="back-link">Go Back to Input Form</a>
    </div>
</body>
</html>
```