from flask import Flask, render\_template, request

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import GaussianNB

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

import matplotlib.pyplot as plt

import seaborn as sns

import os

app = Flask(\_\_name\_\_)

# Step 1: Load dataset (hardcoded in the code)

def load\_data():

    filepath = r"C:\Users\omk57\Downloads\olympics\_dataset.csv"

    return pd.read\_csv(filepath)

# Step 2: Preprocessing - Create binary 'Medal\_Won' column and aggregate data

def preprocess\_data(data):

    data['Medal\_Won'] = data['Medals'].apply(lambda x: 0 if x == 'No medal' else 1)

    team\_year\_data = data.groupby(['Team', 'Year', 'Sport']).agg(

        total\_participation=('ID', 'count'),

        medals\_won=('Medal\_Won', 'sum')

    ).reset\_index()

    # Calculate win percentage

    team\_year\_data['win\_percentage'] = (team\_year\_data['medals\_won'] / team\_year\_data['total\_participation']) \* 100

    return team\_year\_data

# Function to plot bar chart with country names and win percentages

def plot\_histogram(sport\_data, sport):

    plt.figure(figsize=(12, 6))

    # Filter data for only those countries with some winning chances

    filtered\_data = sport\_data[sport\_data['win\_percentage'] > 0]

    # Sort data by win percentage

    sorted\_data = filtered\_data.sort\_values(by='win\_percentage', ascending=False)

    # Plot bar chart

    sns.barplot(x='Team', y='win\_percentage', data=sorted\_data, palette='coolwarm')

    # Add titles and labels

    plt.title(f'Win Percentage by Country for {sport}', fontsize=16)

    plt.xlabel('Country (Team)', fontsize=12)

    plt.ylabel('Win Percentage', fontsize=12)

    # Rotate the x-axis labels for better readability

    plt.xticks(rotation=45, ha='right')

    # Save the chart as an image to display in the HTML

    image\_path = os.path.join('static', 'histogram.png')

    plt.tight\_layout()

    plt.savefig(image\_path)

    plt.close()

@app.route("/", methods=["GET", "POST"])

def index():

    if request.method == "POST":

        sport = request.form.get("sport")

        data = load\_data()

        team\_year\_data = preprocess\_data(data)

        sport\_data = team\_year\_data[team\_year\_data['Sport'] == sport]

        if sport\_data.empty:

            return render\_template('index.html', error=f"No data available for {sport}.")

        # Features and labels for both models

        X = sport\_data[['total\_participation', 'medals\_won']]

        # Binarize win\_percentage (e.g., 50% threshold)

        sport\_data['win\_success'] = sport\_data['win\_percentage'].apply(lambda x: 1 if x > 50 else 0)

        y = sport\_data['win\_success']

        # Split data for training/testing

        X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

        # Naive Bayes Model

        naive\_bayes = GaussianNB()

        naive\_bayes.fit(X\_train, y\_train)

        y\_pred\_naive = naive\_bayes.predict(X\_test)

        naive\_accuracy = accuracy\_score(y\_test, y\_pred\_naive)

        # Logistic Regression Model

        logistic\_reg = LogisticRegression(max\_iter=1000)

        logistic\_reg.fit(X\_train, y\_train)

        y\_pred\_logistic = logistic\_reg.predict(X\_test)

        logistic\_accuracy = accuracy\_score(y\_test, y\_pred\_logistic)

        # Calculate predicted chances of winning for both models

        sport\_data['winning\_chance\_naive\_bayes'] = naive\_bayes.predict\_proba(X)[:, 1] \* 100

        sport\_data['winning\_chance\_logistic'] = logistic\_reg.predict\_proba(X)[:, 1] \* 100

        # Filter only countries with winning chances > 0

        sport\_data = sport\_data[(sport\_data['winning\_chance\_naive\_bayes'] > 0) |

                                (sport\_data['winning\_chance\_logistic'] > 0)]

        if sport\_data.empty:

            return render\_template('index.html', error=f"No countries have a chance of winning in {sport}.")

        results\_df = pd.DataFrame({

            'Team': sport\_data['Team'],

        'Winning\_Probability\_NB': sport\_data['winning\_chance\_naive\_bayes'],

        'Winning\_Probability\_LR': sport\_data['winning\_chance\_logistic']

        })

        results\_df.to\_csv('winning\_probabilities.csv', index=False)

        # Generate histogram and save it

        plot\_histogram(sport\_data, sport)

        # Decide which model is more accurate

        more\_accurate = "Logistic Regression" if logistic\_accuracy > naive\_accuracy else "Naive Bayes"

        # Send data to the HTML page

        results = sport\_data[['Team', 'total\_participation', 'medals\_won', 'win\_percentage']].to\_dict(orient='records')

        for i, result in enumerate(results):

            result['winning\_chance\_logistic'] = round(sport\_data['winning\_chance\_logistic'].iloc[i], 2)

            result['winning\_chance\_naive\_bayes'] = round(sport\_data['winning\_chance\_naive\_bayes'].iloc[i], 2)

        return render\_template('index.html', sport=sport, results=results, logistic\_accuracy=logistic\_accuracy \* 100,

                               naive\_accuracy=naive\_accuracy \* 100, more\_accurate=more\_accurate)

    return render\_template('index.html')

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

/\* General Styles \*/

body {

    font-family: 'Helvetica Neue', Arial, sans-serif;

    background-color: #f9fafc;

    color: #333;

    margin: 0;

    padding: 20px;

}

/\* Container \*/

.container {

    max-width: 900px;

    margin: auto;

    background: #ffffff;

    padding: 40px;

    border-radius: 12px;

    box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1);

    border: 2px solid #e3e7ed;

}

/\* Headings \*/

h1 {

    text-align: center;

    font-size: 2.5rem;

    color: #4682b4;

    margin-bottom: 30px;

}

h2, h3 {

    text-align: center;

    color: #4682b4;

    margin: 20px 0;

}

/\* Form Styling \*/

form {

    text-align: center;

    margin-bottom: 40px;

}

label {

    display: block;

    font-size: 1.2rem;

    color: #4682b4;

    margin-bottom: 10px;

}

input[type="text"] {

    width: 80%;

    max-width: 400px;

    padding: 12px;

    border: 2px solid #b2c8df;

    border-radius: 6px;

    font-size: 1rem;

    margin-bottom: 20px;

}

button {

    padding: 12px 30px;

    background-color: #5f9ea0;

    color: white;

    border: none;

    border-radius: 8px;

    font-size: 1.2rem;

    cursor: pointer;

    transition: background-color 0.3s ease;

}

button:hover {

    background-color: #4682b4;

}

/\* Output Container \*/

.output-container {

    display: flex;

    flex-direction: column; /\* Change to column layout \*/

    align-items: center; /\* Center items \*/

    margin-top: 20px;

}

/\* Results Section \*/

.results {

    width: 100%; /\* Ensure it uses full width \*/

}

table {

    width: 100%; /\* Use full width of the container \*/

    border-collapse: collapse;

    margin: 5% 0;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.05);

    background-color: #fafafa;

}

th, td {

    padding: 12px; /\* Reduced padding for smaller table size \*/

    text-align: center;

    font-size: 0.9rem; /\* Smaller font size \*/

}

th {

    background-color: #f0f8ff;

    color: #333;

    font-weight: 600;

    border-bottom: 3px solid #b2c8df;

}

td {

    border-bottom: 1px solid #e3e7ed;

    color: #555;

}

tr:nth-child(even) {

    background-color: #f7fafd;

}

tr:hover {

    background-color: #f0faff;

}

/\* Power BI Report Section \*/

.powerbi-report-container {

    display: flex;

    flex-direction: column;

    align-items: center; /\* Center contents horizontally \*/

    margin-top: 20px; /\* Add some spacing from the histogram \*/

}

iframe {

    width: 100%; /\* Make the iframe responsive \*/

    height: 600px; /\* Set a height for better visibility \*/

    max-width: 900px; /\* Optional: limit the width of the iframe \*/

    border: none; /\* Optional: Remove the border \*/

}

/\* Error Message \*/

.error {

    color: #ff6b6b;

    text-align: center;

    font-size: 1.2rem;

    margin-top: 20px;

    font-weight: bold;

}

/\* Image Styling \*/

img {

    display: block;

    max-width: 100%;

    margin: 30px auto;

    border-radius: 12px;

    border: 3px solid #b2c8df;

    box-shadow: 0 6px 12px rgba(0, 0, 0, 0.1);

}

/\* Footer Styling \*/

footer {

    text-align: center;

    margin-top: 40px;

    font-size: 0.9rem;

    color: #888;

}

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Olympic Medal Prediction</title>

    <link rel="stylesheet" href="{{ url\_for('static', filename='styles.css') }}">

</head>

<body>

    <div class="container">

        <h1>Olympic Medal Prediction</h1>

        <form method="POST">

            <label for="sport">Enter the sport name:</label>

            <input type="text" id="sport" name="sport" required>

            <button type="submit">Predict</button>

        </form>

        <div class="output-container">

            {% if sport %}

                <div class="results">

                    <h2>Results for {{ sport }}</h2>

                    <table>

                        <thead>

                            <tr>

                                <th>Country</th>

                                <th>Total Participation</th>

                                <th>Medals Won</th>

                                <th>Win Percentage</th>

                                <th>Naive Bayes Winning Chances</th>

                                <th>Logistic Regression Winning Chances</th>

                            </tr>

                        </thead>

                        <tbody>

                            {% for result in results %}

                            <tr>

                                <td>{{ result['Team'] }}</td>

                                <td>{{ result['total\_participation'] }}</td>

                                <td>{{ result['medals\_won'] }}</td>

                                <td>{{ result['win\_percentage'] }}</td>

                                <td>{{ result['winning\_chance\_naive\_bayes'] }}%</td>

                                <td>{{ result['winning\_chance\_logistic'] }}%</td>

                            </tr>

                            {% endfor %}

                        </tbody>

                    </table>

                    <h3>Model Accuracy</h3>

                    <p>Naive Bayes Accuracy: {{ naive\_accuracy }}%</p>

                    <p>Logistic Regression Accuracy: {{ logistic\_accuracy }}%</p>

                    <p>The more accurate model is: <strong>{{ more\_accurate }}</strong></p>

                    <h3>Win Percentage Histogram</h3>

                    <img src="{{ url\_for('static', filename='histogram.png') }}" alt="Win Percentage Histogram">

                    <!-- Move the Power BI report below the histogram -->

                    <div class="powerbi-report-container">

                        <h2>Power BI Report</h2>

                        <iframe src="https://app.powerbi.com/reportEmbed?reportId=a94f7be8-7199-4461-b72a-55430628dd61&autoAuth=true&ctid=c7b00d7f-ad99-442a-b12f-c2c912044fdc" frameborder="0" allowFullScreen="true"></iframe>

                    </div>

                </div>

            {% elif error %}

                <p class="error">{{ error }}</p>

            {% endif %}

        </div>

    </div>

</body>

</html>