

Project Name: Flask Deployment on Forest Fire data

Name: Griffin Palfrey

Batch Code: LISUM35

Submission Date: 07/27/24

Submitted to: DataGlacier Reviewers

Step 1: Found and saved a simple online dataset about calculating the probability of a forest fire occurring given a variety of different metrics and attributes.

Small snippet of the data set:

```
X, y = load_data(path='../data/forestfires.csv',
                data_loader=lambda path: pd.read_csv(path),
                column_names=['month', 'day', 'FFMC', 'DMC', 'DC', 'ISI', 'temp', 'RH', 'wind', 'rain', 'area'])
```

month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area		
7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0	0
7	4	oct	tue	90.6	35.4	669.1	6.7	18	33	0.9	0	0
7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0	0
8	6	mar	fri	91.7	33.3	77.5	9	8.3	97	4	0.2	0
8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0	0
8	6	aug	sun	92.3	85.3	488	14.7	22.2	29	5.4	0	0
8	6	aug	mon	92.3	88.9	495.6	8.5	24.1	27	3.1	0	0
8	6	aug	mon	91.5	145.4	608.2	10.7	8	86	2.2	0	0
8	6	sep	tue	91	129.5	692.6	7	13.1	63	5.4	0	0
7	5	sep	sat	92.5	88	698.6	7.1	22.8	40	4	0	0
7	5	sep	sat	92.5	88	698.6	7.1	17.8	51	7.2	0	0
7	5	sep	sat	92.8	73.2	713	22.6	19.3	38	4	0	0
6	5	aug	fri	63.5	70.8	665.3	0.8	17	72	6.7	0	0
6	5	sep	mon	90.9	126.5	686.5	7	21.3	42	2.2	0	0

<https://archive.ics.uci.edu/dataset/162/forest+fires>

Step 2: Trained and saved the model using a python file called 'train.py'

```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
import joblib

# Load the dataset
data = pd.read_csv('forestfires.csv')

# Preprocess the data
data['month'] = data['month'].astype('category').cat.codes
data['day'] = data['day'].astype('category').cat.codes

X = data.drop('area', axis=1)
y = data['area']

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

# Train the model
model = RandomForestRegressor(random_state=42)
model.fit(X_train, y_train)

# Save the model
joblib.dump(model, 'forestfires_model.joblib')
```

Step 3: Create a Flask web app called 'app.py'

```
from flask import Flask, request, render_template
import joblib
import pandas as pd

app = Flask(__name__)

# Load the model
model = joblib.load('forestfires_model.joblib')

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

@app.route('/predict', methods=['POST'])
def predict():
    input_data = {
        'X': [float(request.form['X'])],
        'Y': [float(request.form['Y'])],
        'month': [int(request.form['month'])],
        'day': [int(request.form['day'])],
        'FFMC': [float(request.form['FFMC'])],
        'DMC': [float(request.form['DMC'])],
        'DC': [float(request.form['DC'])],
        'ISI': [float(request.form['ISI'])],
        'temp': [float(request.form['temp'])],
        'RH': [float(request.form['RH'])],
        'wind': [float(request.form['wind'])],
        'rain': [float(request.form['rain'])]
    }

    df = pd.DataFrame(input_data)
    prediction = model.predict(df)

    return render_template('index.html', prediction=prediction[0])

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

Step 4: Add a GUI for predictions

To do this, I created a new directory called 'templates' within which I created a template named 'index.html'. In creating this predictive program, I made sure to mitigate any potential confusion in the column titles by writing out each acronym as well as providing a range for each input.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Forest Fires Area Prediction</title>
  <style>
    label {
      font-weight: bold;
    }
    .description {
      font-size: 0.9em;
      color: gray;
    }
  </style>
</head>
<body>
  <h1>Forest Fires Area Prediction</h1>
  <form action="/predict" method="post">
    <label for="X">X (X-axis spatial coordinate within the Montesinho park map):</label>
    <input type="number" step="1" id="X" name="X" required>
    <p class="description">Range: 1 to 9</p><br>

    <label for="Y">Y (Y-axis spatial coordinate within the Montesinho park map):</label>
    <input type="number" step="1" id="Y" name="Y" required>
    <p class="description">Range: 2 to 9</p><br>

    <label for="month">Month (Month of the year):</label>
    <input type="number" step="1" id="month" name="month" required>
    <p class="description">Range: 0 (January) to 11 (December)</p><br>

    <label for="day">Day (Day of the week):</label>
    <input type="number" step="1" id="day" name="day" required>
    <p class="description">Range: 0 (Sunday) to 6 (Saturday)</p><br>
  </form>

```

```

    <label for="FFMC">FFMC (Fine Fuel Moisture Code):</label>
    <input type="number" step="0.1" id="FFMC" name="FFMC" required>
    <p class="description">Range: 18.7 to 96.20</p><br>

    <label for="DMC">DMC (Duff Moisture Code):</label>
    <input type="number" step="0.1" id="DMC" name="DMC" required>
    <p class="description">Range: 1.1 to 291.3</p><br>

    <label for="DC">DC (Drought Code):</label>
    <input type="number" step="0.1" id="DC" name="DC" required>
    <p class="description">Range: 7.9 to 860.6</p><br>

    <label for="ISI">ISI (Initial Spread Index):</label>
    <input type="number" step="0.1" id="ISI" name="ISI" required>
    <p class="description">Range: 0.0 to 56.10</p><br>

    <label for="temp">Temperature (in Celsius degrees):</label>
    <input type="number" step="0.1" id="temp" name="temp" required>
    <p class="description">Range: 2.2 to 33.30</p><br>

    <label for="RH">Relative Humidity (in %):</label>
    <input type="number" step="1" id="RH" name="RH" required>
    <p class="description">Range: 15 to 100</p><br>

    <label for="wind">Wind speed (in km/h):</label>
    <input type="number" step="0.1" id="wind" name="wind" required>
    <p class="description">Range: 0.4 to 9.40</p><br>

    <label for="rain">Rain (outside rain in mm/m2):</label>
    <input type="number" step="0.1" id="rain" name="rain" required>
    <p class="description">Range: 0.0 to 6.4</p><br>

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

  {% if prediction %}
    <h2>Prediction: {{ prediction }}</h2>
  {% endif %}
</body>
</html>

```

Step 5: Input numbers and receive an output, which is the predictive likelihood of a forest fire occurring giving the metrics taken into consideration.

Forest Fires Area Prediction

X (X-axis spatial coordinate within the Montesinho park map):

Range: 1 to 9

Y (Y-axis spatial coordinate within the Montesinho park map):

Range: 2 to 9

Month (Month of the year):

Range: 0 (January) to 11 (December)

Day (Day of the week):

Range: 0 (Sunday) to 6 (Saturday)

FFMC (Fine Fuel Moisture Code):

Range: 18.7 to 96.20

DMC (Duff Moisture Code):

Range: 1.1 to 291.3

DC (Drought Code):

Range: 7.9 to 860.6

ISI (Initial Spread Index):

Range: 0.0 to 56.10

Temperature (in Celsius degrees):

Range: 2.2 to 33.30

Relative Humidity (in %):

Range: 15 to 100

Wind speed (in km/h):

Range: 0.4 to 9.40

Rain (outside rain in mm/m2):

Range: 0.0 to 6.4

Predict

Prediction: 74.37590000000004