Project Name: Cloud and API deployment with Forest fire data

Name: Griffin Palfrey

Batch Code: LISUM35

Submission Date: 08/02/24

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 snipped of the data set:

```
X,Col2: 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'

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

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

# Prepare your feature matrix X and target vector y
X = data[['X', 'Y', 'month', 'day', 'FFMC', 'DMC', 'DC', 'ISI', 'temp', 'RH', 'wind', 'rain']]
y = data['area']

# Split t(variable) X_test: Any test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a model
model = RandomForestRegressor()
model.fit(X_train, y_train)

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

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

```
ires (1) > app.py > ...
from flask import Flask, request, render_template, jsonify
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 an 'index.html' in a 'templates' directory to outline more specifically the inputs on the site.

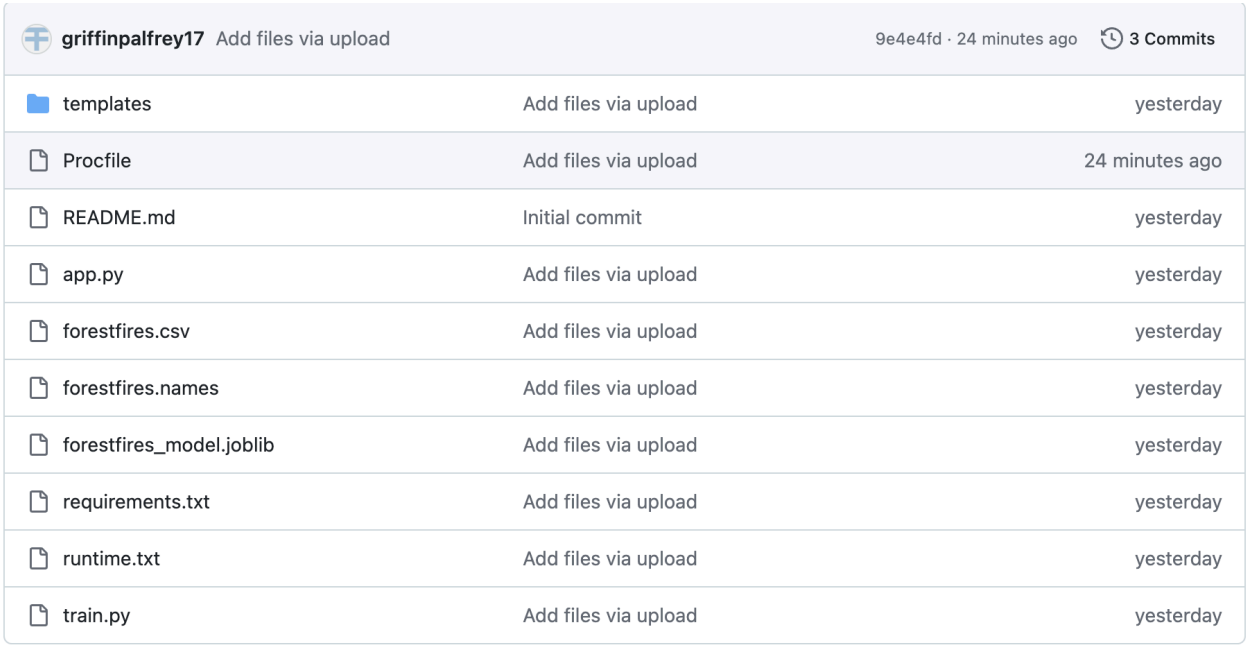
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Forest Fires Area Prediction</title>
    <style>
        label {
            font-weight: bold;
        }
        .description {
            font-size: 0.9em;
            color: gray;
        }
        .container {
            max-width: 600px;
            margin: auto;
            padding: 20px;
            border: 1px solid #ccc;
            border-radius: 5px;
            background-color: #f9f9f9;
        }
        .form-group {
            margin-bottom: 15px;
        }
        .form-group label {
            display: block;
            margin-bottom: 5px;
        }
        .form-group input {
            width: 100%;
            padding: 8px;
            box-sizing: border-box;
        }
        .form-group button {
            padding: 10px 20px;
            background-color: #007bff;
            border: none;
            color: white;
            cursor: pointer;
        }
        .form-group button:hover {
            background-color: #0056b3;
        }
```

```
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Forest Fires Area Prediction</h1>
        <form action="/predict" method="post">
            <div class="form-group">
                <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>
            </div>

            <div class="form-group">
                <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>
            </div>

            <div class="form-group">
                <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>
            </div>

            <div class="form-group">
                <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>
            </div>

            <div class="form-group">
                <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>
            </div>

            <div class="form-group">
                <label for="DMC">DMC (Duff Moisture Code):</label>
                <input type="number" step="0.1" id="DMC" name="DMC" required>
```

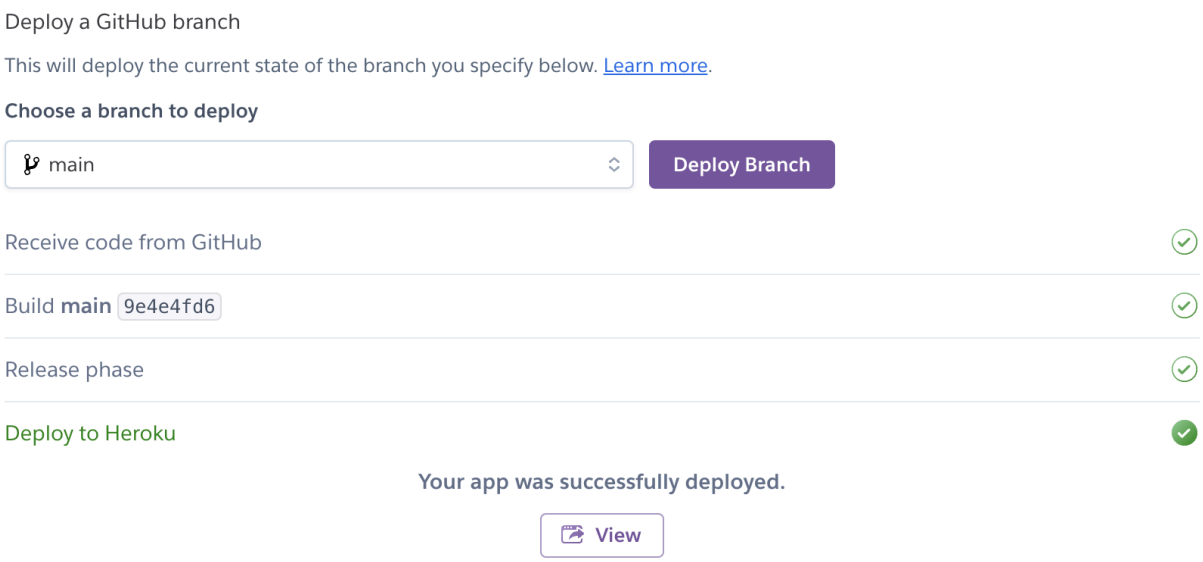
```
    <div class="form-group">
        <label for="DC">DC (Drought Code):</label>
        <input type="number" step="0.1" id="DC" name="DC" required>
        <p class="description">Range: 7.9 to 860.6</p>
    </div>

    <div class="form-group">
        <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>
    </div>

    <div class="form-group">
        <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>
    </div>

    <div class="form-group">
        <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>
    </div>

    <div class="form-group">
        <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>
    </div>

    <div class="form-group">
        <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>
    </div>

    <div class="form-group">
        <button type="submit">Predict</button>
    </div>
```

```
            <div class="form-group">
                <button type="submit">Predict</button>
            </div>
        </form>

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

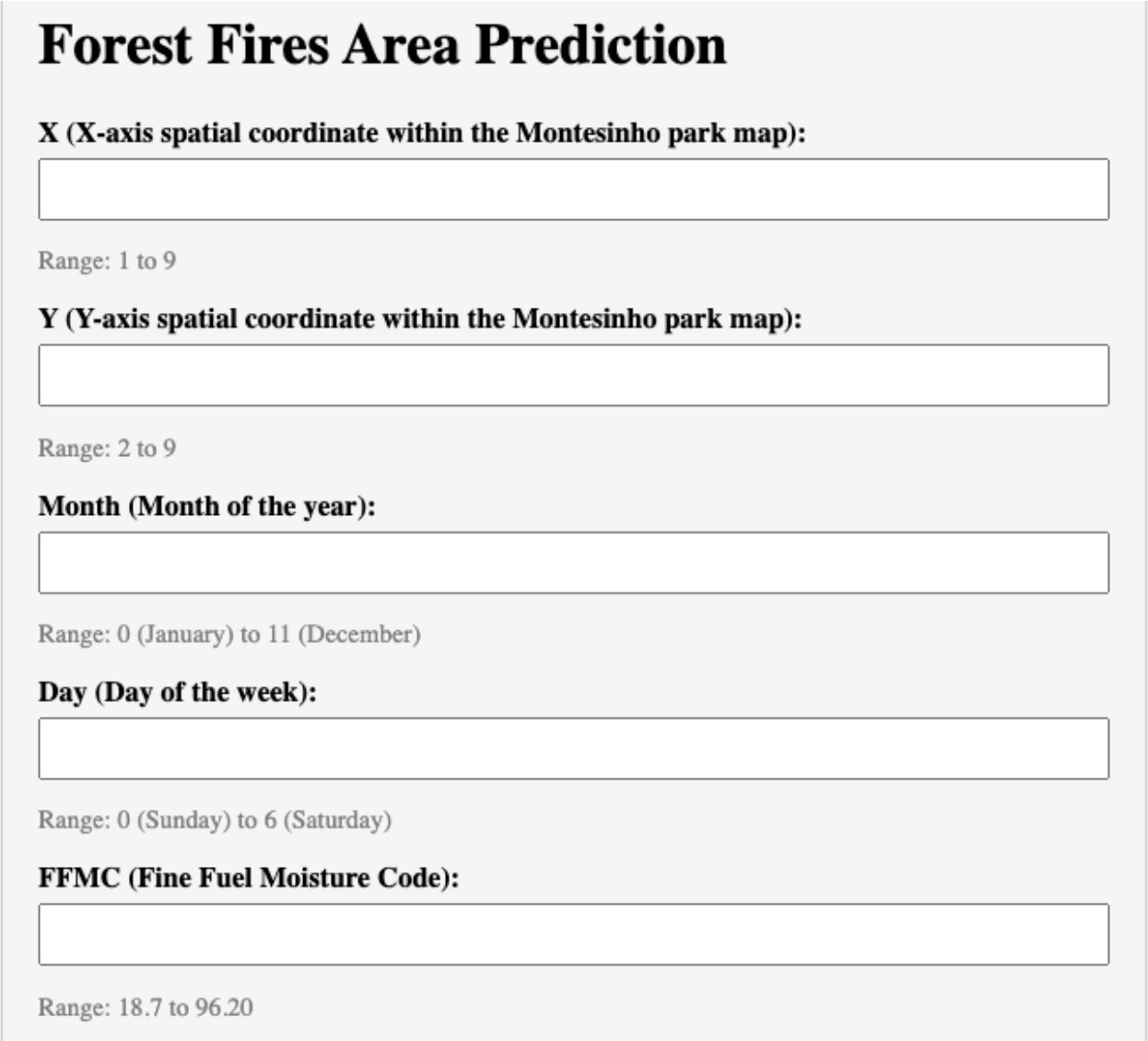
Step 5: Upload files to Git



Step 6: Deploy the model on Heroku



Step 7: Visit the API and input appropriate values



Wind speed (in km/h):

Range: 0.4 to 9.40

Rain (outside rain in mm/m2):

Range: 0.0 to 6.4

Predict