Breast Cancer

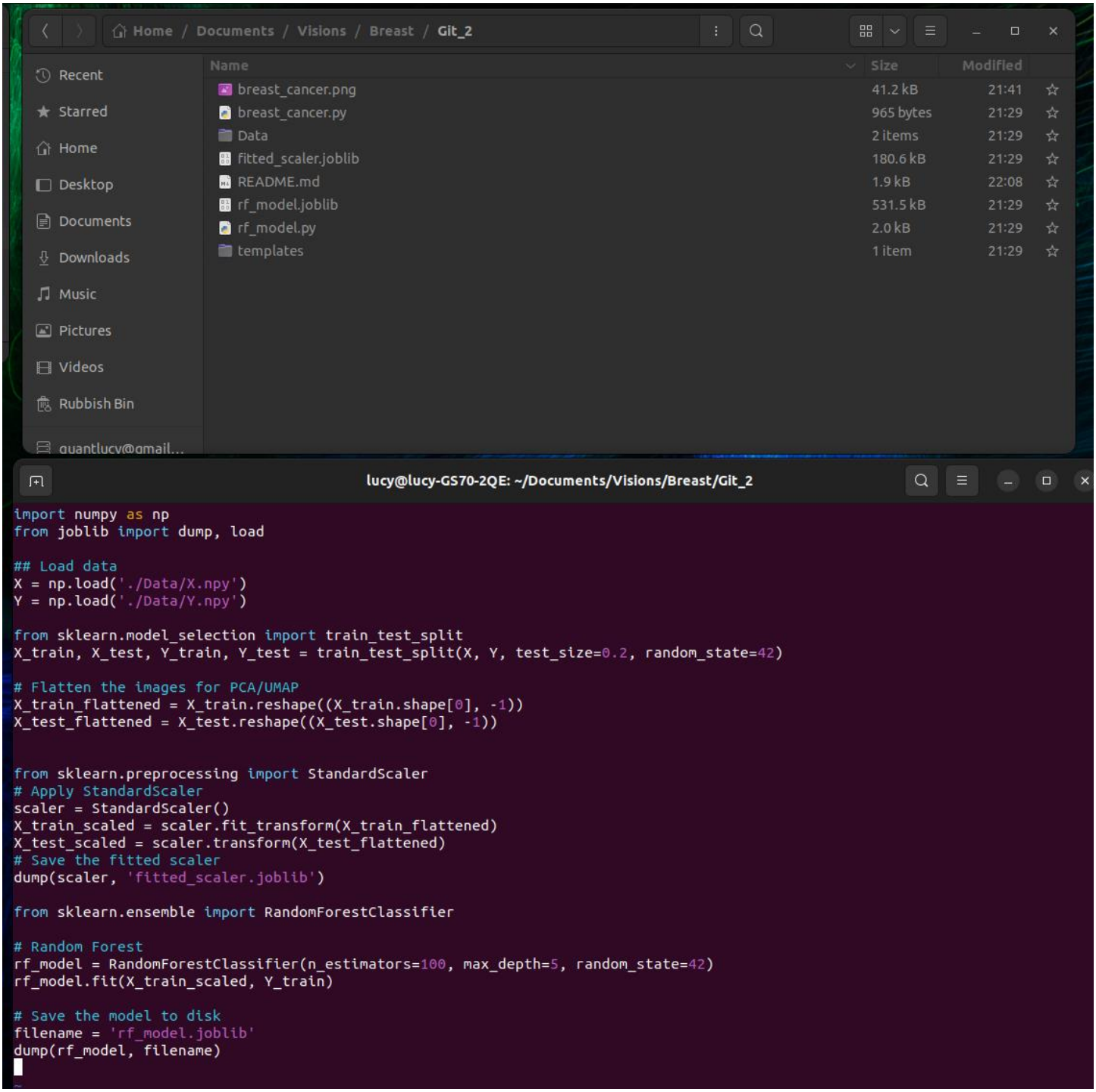
Name :Lucy Nowacki

Batch code: LISUM32

Submission date: 28/04/2024

Submitted to: Data Glacier

FILES STRUCTURE & ML MODEL

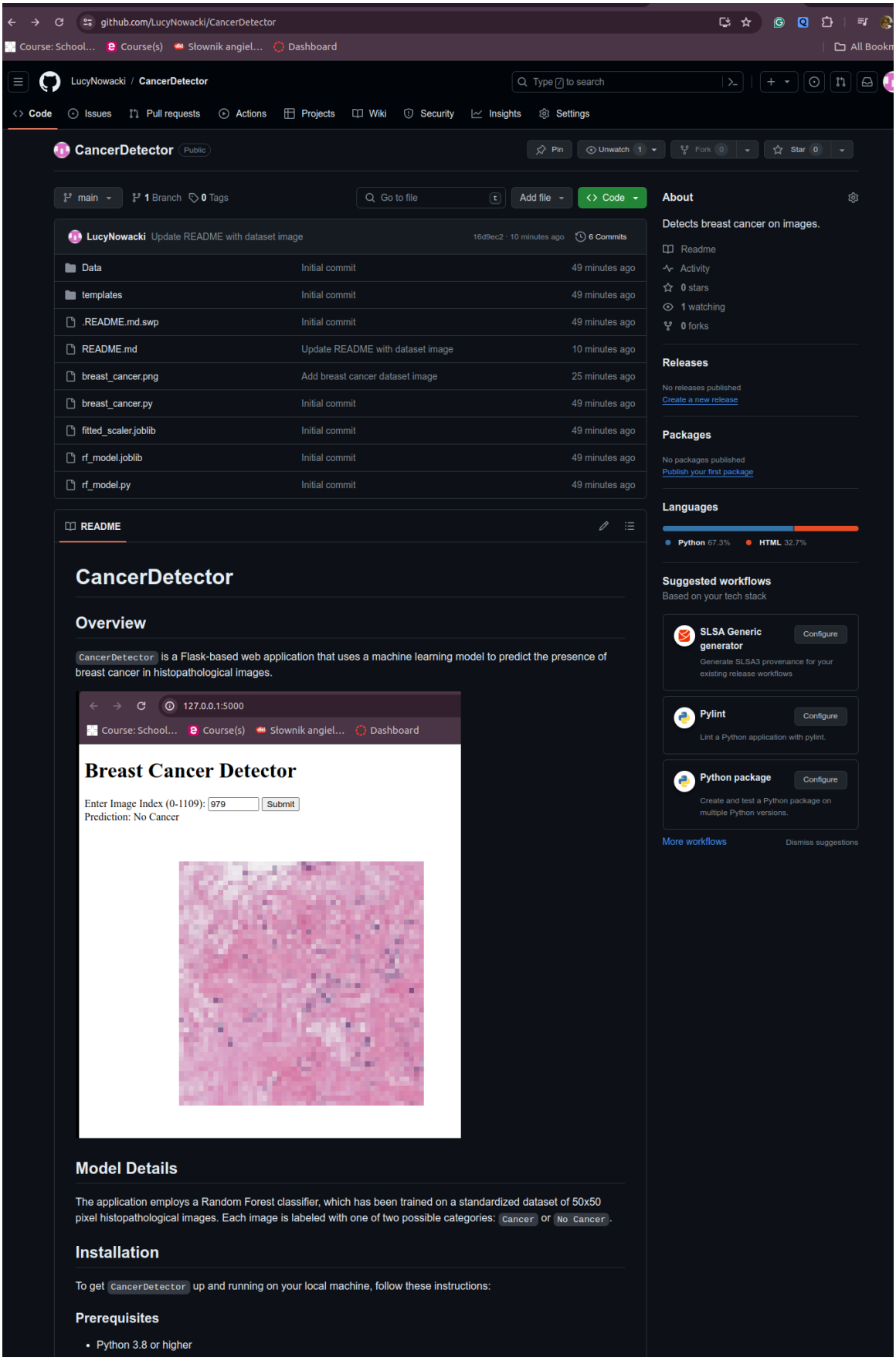


FLASK IMPLEMENTATION

lucy@lucy-GS70-2QE: ~/Documents/Visions/Breast/Git_2

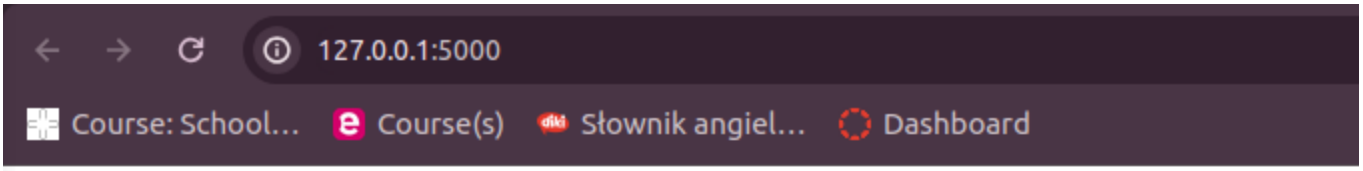
```python
from flask import Flask, request, jsonify, render_template
from joblib import load
from sklearn.preprocessing import StandardScaler
import numpy as np
import os

import base64
from io import BytesIO
import matplotlib.pyplot as plt
import matplotlib
matplotlib.use('Agg')

app = Flask(__name__)

# Load the model and scaler at app startup
model_path = os.path.join('/home/lucy/Documents/Visions/Breast/Git_1', 'rf_model.joblib')
model = load(model_path)
scaler_path = '/home/lucy/Documents/Visions/Breast/Git_1/fitted_scaler.joblib'
scaler = load(scaler_path)

@app.route('/')
def index():
    # Display the form for user input
    return render_template('index.html')


@app.route('/predict', methods=['POST'])
def predict():
    try:
        image_index = int(request.form['image_index'])
        X = np.load('./Data/X.npy')  # Load data

        print("Selected image shape:", X[image_index].shape)  # Should print (50, 50, 3)

        # Flatten the image for prediction
        image_flat = X[image_index].reshape(-1)  # Flatten the image
        print("Flattened image shape:", image_flat.shape)  # Should print (7500,)

        # Transform the data using the pre-fitted scaler
        image_scaled = scaler.transform([image_flat])  # Make sure it's a 2D array for scaling

        # Predict using the loaded model
        prediction = model.predict(image_scaled)
        result = 'No Cancer' if prediction[0] == 0 else 'Cancer'

        # Convert image to Base64 for displaying
        fig, ax = plt.subplots()
        ax.imshow(X[image_index])  # Use original shape for displaying
        ax.axis('off')
        buf = BytesIO()
        plt.savefig(buf, format='png')
        plt.close(fig)
        image_base64 = base64.b64encode(buf.getvalue()).decode('utf-8')
        buf.close()

        return jsonify({'prediction': result, 'image': image_base64})
    except Exception as e:
        return jsonify({'error': str(e)}), 500

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

GITHUB DEPLOYEMENT

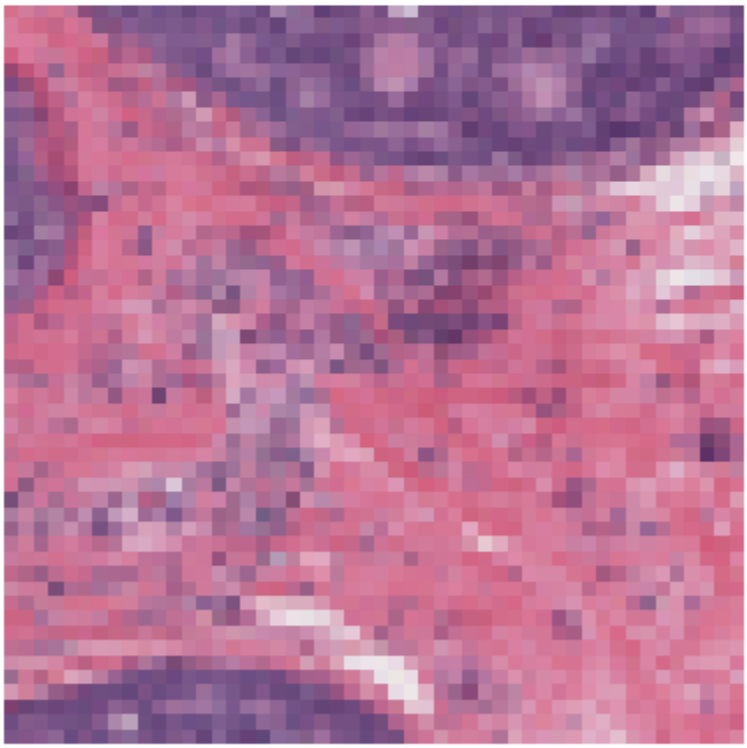


START APP

```
lucy@lucy-GS70-2QE: ~/Documents/Visions/Breast/Git_2
(lucy) lucy@lucy-GS70-2QE:~/Documents/Visions/Breast/Git_2$ export FLASK_APP=rf_model.py
(lucy) lucy@lucy-GS70-2QE:~/Documents/Visions/Breast/Git_2$ flask run
 * Serving Flask app 'rf_model.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [28/Apr/2024 22:24:08] "GET / HTTP/1.1" 200 -
Selected image shape: (50, 50, 3)
Flattened image shape: (7500,)
127.0.0.1 - - [28/Apr/2024 22:24:15] "POST /predict HTTP/1.1" 200 -
```



Breast Cancer Detector

Enter Image Index (0-1109): 420 Submit

Prediction: Cancer