FLASK DEPLOYMENT ASSIGNMENT

App Name: Predictor Loan API
Version: 2.0
Creator: Sebastian Bucheli
Submission date: 03/11/2022
Submitted to: Data Glacier

1. Creating API Endpoint (app.py)

```python
import pandas as pd
import pickle
from flask import Flask, jsonify, request


app = Flask(__name__)


@app.route('/', methods=['GET', 'POST'])
def home():
    if request.method == 'GET':
        home = 'Hello world!!'
        return jsonify({'home': home})

@app.route('/predict', methods=['POST'])
def predict():
    """
    Predict with the data entered in the html page.
    """
    with open('model.pkl', 'rb') as f:
        model = pickle.load(f)
    credit_policy = request.args.get('credit_policy')
    int_rate = request.args.get('int_rate')
    installment = request.args.get('installment')
    log_annual_inc = request.args.get('log_annual_inc')
    dti = request.args.get('dti')
    fico = request.args.get('fico')
    days_with_cr_line = request.args.get('days_with_cr_line')
    revol_bal = request.args.get('revol_bal')
    revol_util = request.args.get('revol_util')
    inq_last_6mths = request.args.get('inq_last_6mths')
    delinq_2yrs = request.args.get('delinq_2yrs')
    pub_rec = request.args.get('pub_rec')
```

```
34
35          test_df = pd.DataFrame({
36              'Credit Policy': [credit_policy],
37              'Interest rate': [int_rate],
38              'Installment': [installment],
39              'Log Annual Income': [log_annual_inc],
40              'Debt-to-Income': [dti],
41              'FICO': [fico],
42              'Days with Credit Line': [days_with_cr_line],
43              'Revolving Balance': [revol_bal],
44              'Revolving Line Utilization': [revol_util],
45              'Inquiries in the last 6 months': [inq_last_6mths],
46              'Days past due on a payment': [delinq_2yrs],
47              'Derogatory Public Records': [pub_rec]
48          })
49          prediction = model.predict(test_df)
50          if int(round(prediction[0])) == 1:
51              return jsonify({
52                  "Loan Payment Prediction": "The user won't fully pay the loan"
53              })
54          else:
55              return jsonify({
56                  "Loan Payment Prediction": "The user will fully pay the loan"
57              })
58
59      if __name__ == "__main__":
60          app.run(port=5000, debug=True)
```

2. Running the app in the localhost

```
PS ████████████████████████\assignments\week5> python .\app.py
 * Serving Flask app 'app'
 * Debug mode: on
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
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 122-661-508
```
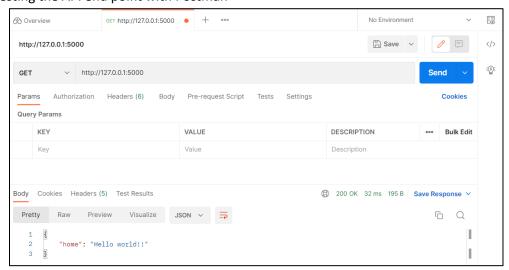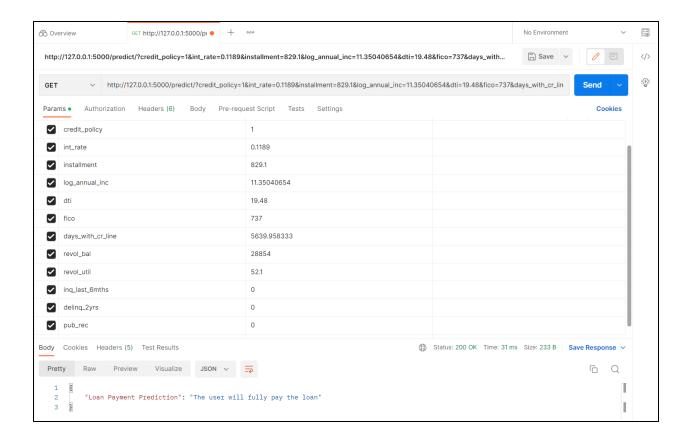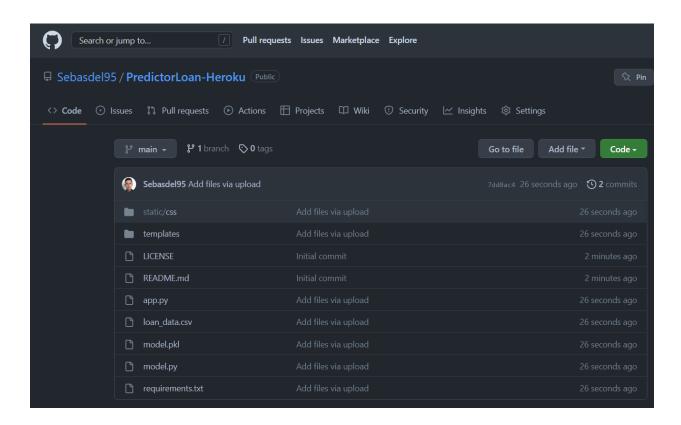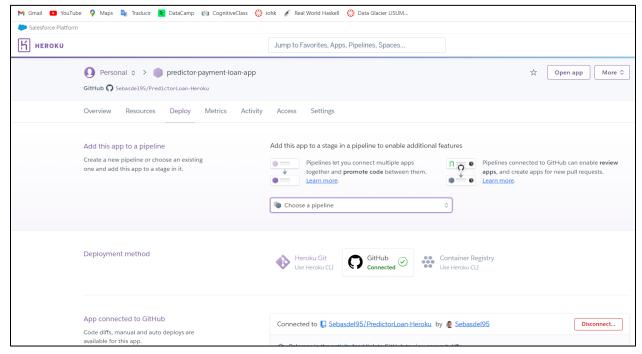
3. Testing the API end point with Postman
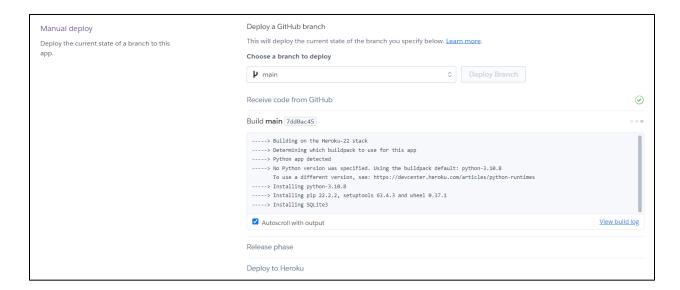
HEROKU DEPLOYMENT

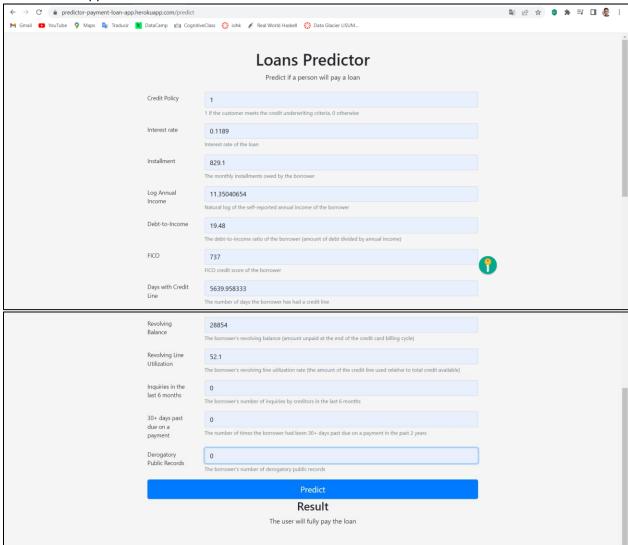4. Adding Predictor Loan Payment app to GitHub

5. Creating web app in Heroku and connecting with GitHub repository.

6. Manual deployment.

7. Web app in Heroku Cloud Server

8. You can enter to the web app with the following link https://predictor-payment-loan-app.herokuapp.com/predict