CLOUD AND API DEPLOYMENT

Name: Noé Gracida Hernández

Batch code: LISUM06

Submission date: 01-Mar-22

Submitted to: Data Glacier (Canvas)

Head of the data:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Model:

index.html model.py 2

model.py > ...

```python
import numpy as np
import pandas as pd
import pickle
from sklearn import tree

# Loading dataset
FileCSV="diabetes.csv"
df_diabetes = pd.read_csv(FileCSV,sep=",")

X = df_diabetes.iloc[:, 0:-1].values
y = df_diabetes.iloc[:, -1].values

# Initialization and fitting of the model
clf = tree.DecisionTreeClassifier(criterion="entropy", max_depth=3)
clf.fit(X, y)

# Serialization
pickle.dump(clf, open('model.pkl','wb'))

# Deserialization
model = pickle.load(open('model.pkl','rb'))
print(model.predict([[6, 148, 72, 35, 0, 33.6, 0.627, 50]]))
```

HTML Template:

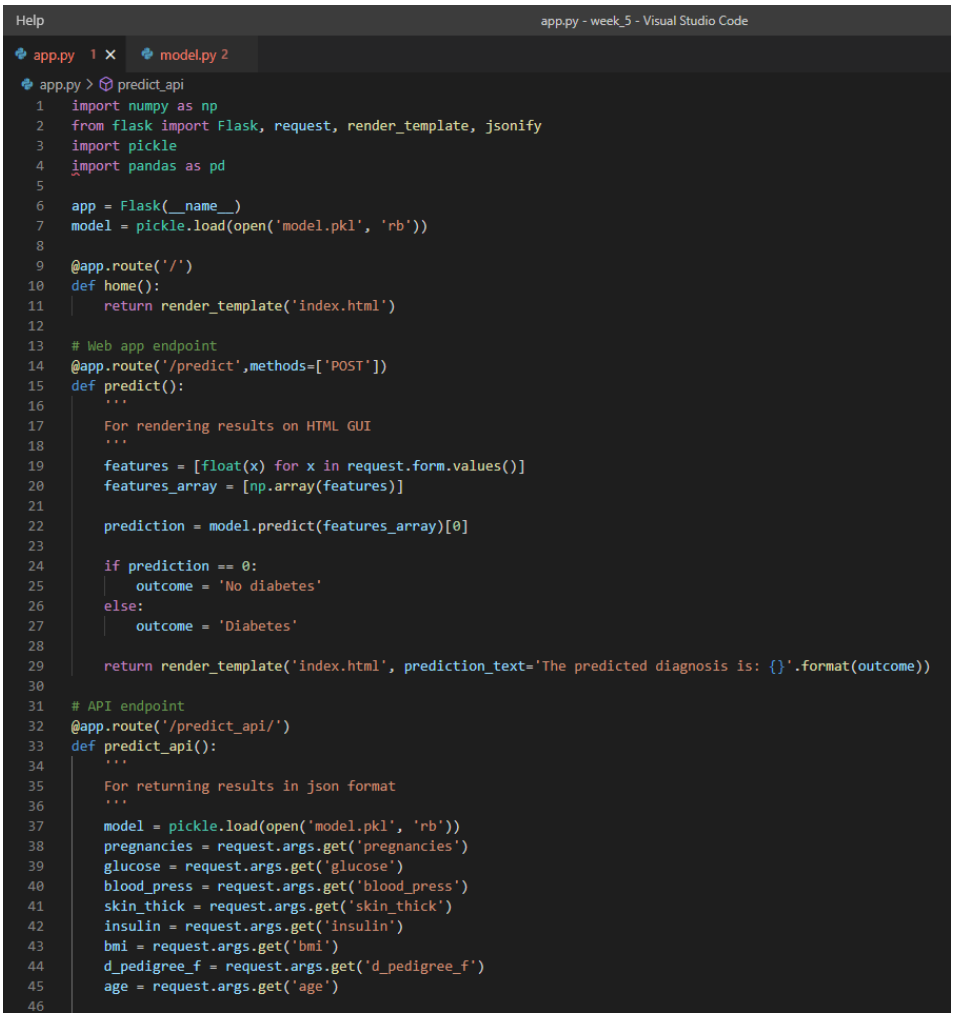
index.html

templates > index.html > html > body > div.login

```
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

</head>

<body>
 <div class="login">
  <h1>Diabetes predictor</h1>

    <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
      <input type="text" name="pregnancies" placeholder="Pregnancies" required="required" />
      <input type="text" name="glucose" placeholder="Glucose" required="required" />
      <input type="text" name="blood_pressure" placeholder="Blood Pressure" required="required" />
      <input type="text" name="skin_thick" placeholder="Skin Thickness" required="required" />
      <input type="text" name="insulin" placeholder="Insulin" required="required" />
      <input type="text" name="bmi" placeholder="BMI" required="required" />
      <input type="text" name="diab_pedigree_func" placeholder="Diabetes Pedigree Function" required="required" />
      <input type="text" name="age" placeholder="Age" required="required" />

      <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>

    <br>
    <br>
    {{ prediction_text }}

 </div>
 <img src="/static/images/Original.svg" style="width: 400px;position: absolute;bottom: 10px;left: 10px;" alt="Company Logo"/>

</body>
</html>

```

Flask app

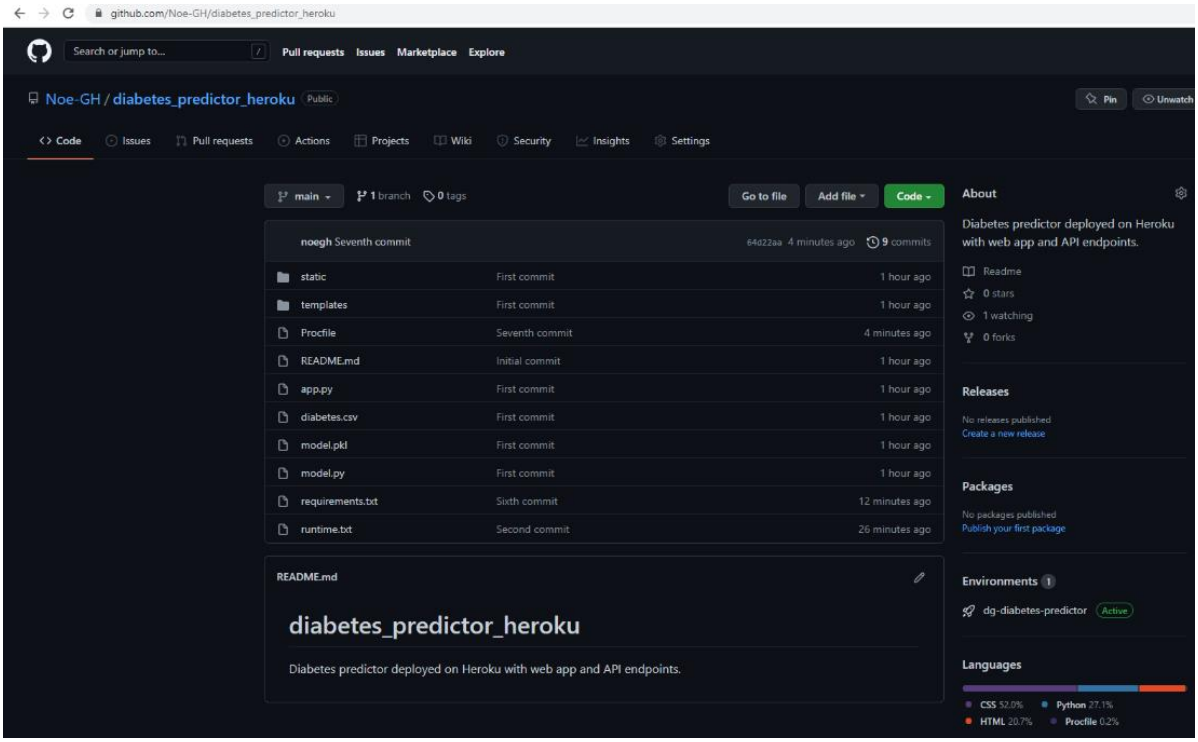
Web app endpoint:



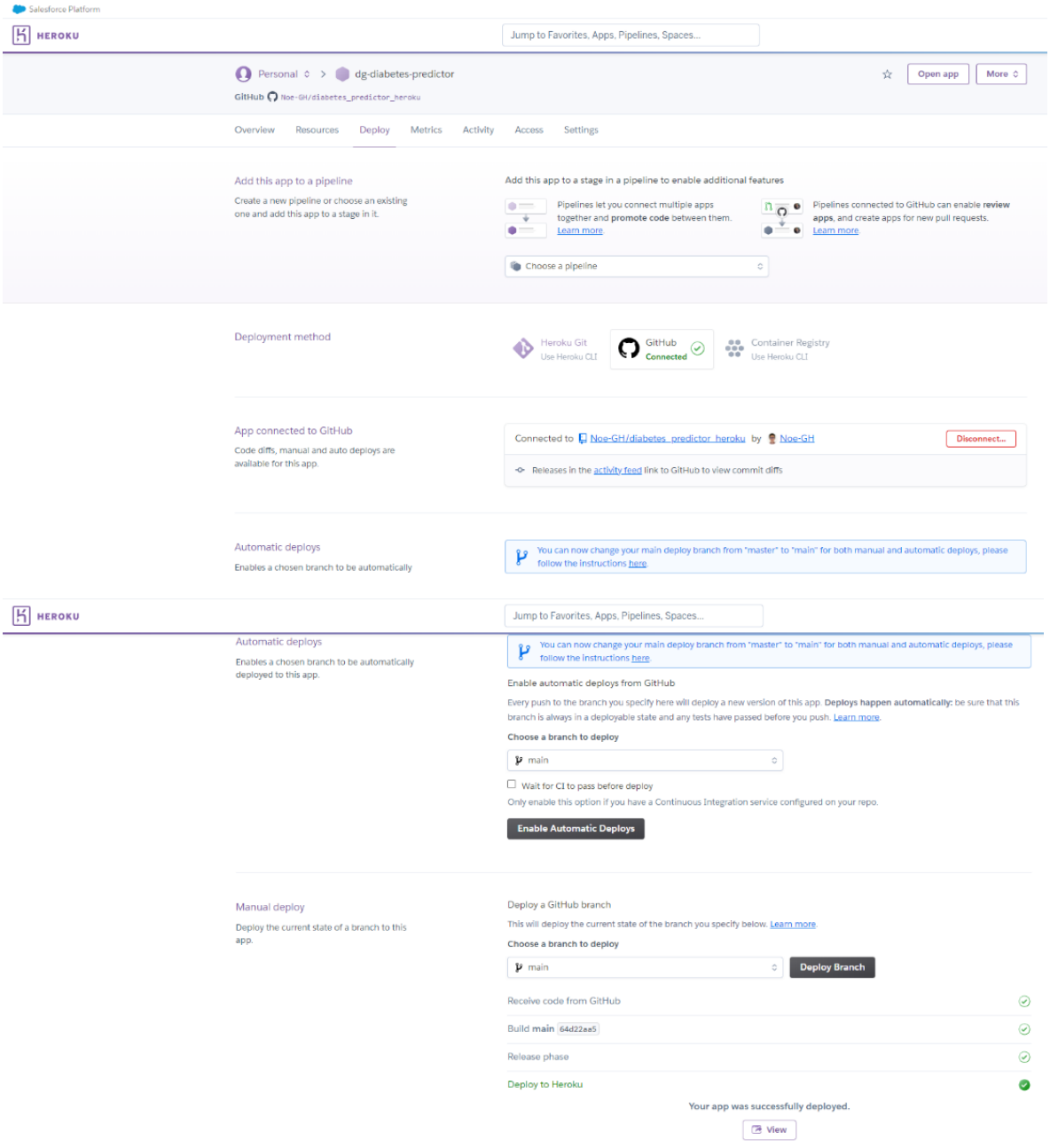
API endpoint:

```python
# API endpoint
@app.route('/predict_api/')
def predict_api():
    '''
    For returning results in json format
    '''
    model = pickle.load(open('model.pkl', 'rb'))
    pregnancies = request.args.get('pregnancies')
    glucose = request.args.get('glucose')
    blood_press = request.args.get('blood_press')
    skin_thick = request.args.get('skin_thick')
    insulin = request.args.get('insulin')
    bmi = request.args.get('bmi')
    d_pedigree_f = request.args.get('d_pedigree_f')
    age = request.args.get('age')

    df = pd.DataFrame({'pregnancies':[pregnancies], 'glucose':[glucose],
                       'blood_press':[blood_press], 'skin_thick':[skin_thick],
                       'insulin':[insulin], 'bmi':[bmi],
                       'd_pedigree_f':[d_pedigree_f], 'age':[age]})

    prediction = model.predict(df)[0]

    if prediction == 0:
        outcome = 'No diabetes'
    else:
        outcome = 'Diabetes'

    return jsonify({'diabetes':outcome})


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

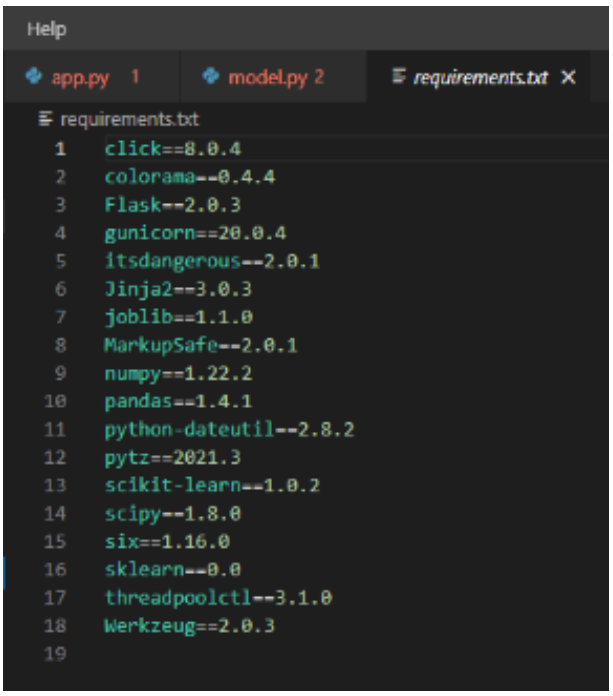
Github repository:



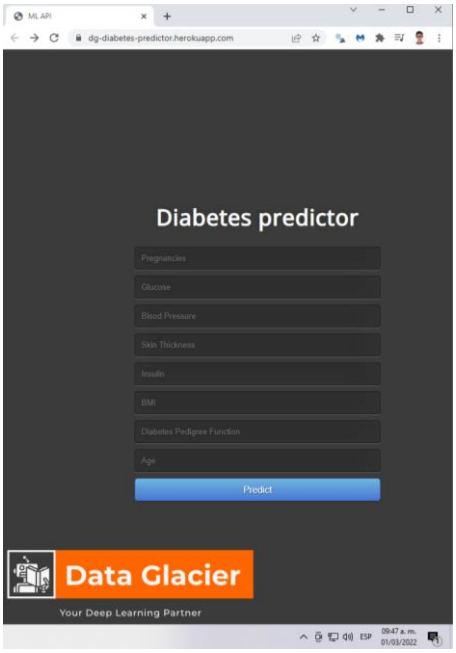
Heroku dashboard:

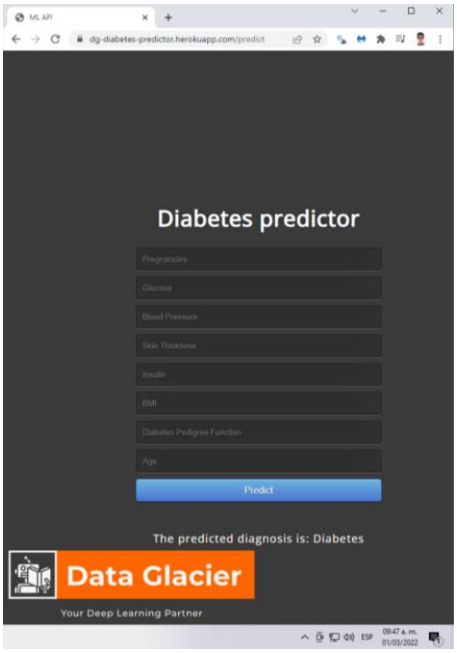


requirements.txt:



Web app on Heroku:





API on Heroku: