

## 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 X
model.py > ...
1  import numpy as np
2  import pandas as pd
3  import pickle
4  from sklearn import tree
5
6  # Loading dataset
7  FileCSV="diabetes.csv"
8  df_diabetes = pd.read_csv(FileCSV,sep=",")
9
10 X = df_diabetes.iloc[:, 0:-1].values
11 y = df_diabetes.iloc[:, -1].values
12
13 # Initialization and fitting of the model
14 clf = tree.DecisionTreeClassifier(criterion="entropy", max_depth=3)
15 clf.fit(X, y)
16
17 # Serialization
18 pickle.dump(clf, open('model.pkl','wb'))
19
20 # Deserialization
21 model = pickle.load(open('model.pkl','rb'))
22 print(model.predict([[6, 148, 72, 35, 0, 33.6, 0.627, 50]]))
23
```

## HTML Template:

```
index.html x
templates > index.html > html > body > div.login
1 <!DOCTYPE html>
2 <html >
3 <head>
4   <meta charset="UTF-8">
5   <title>ML API</title>
6   <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
7   <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
8   <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
9   <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
10  <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
11
12 </head>
13
14 <body>
15   <div class="login">
16     <h1>Diabetes predictor</h1>
17
18     <!-- Main Input For Receiving Query to our ML -->
19     <form action="{{ url_for('predict') }}" method="post">
20       <input type="text" name="pregnancies" placeholder="Pregnancies" required="required" />
21       <input type="text" name="glucose" placeholder="Glucose" required="required" />
22       <input type="text" name="blood_pressure" placeholder="Blood Pressure" required="required" />
23       <input type="text" name="skin_thick" placeholder="Skin Thickness" required="required" />
24       <input type="text" name="insulin" placeholder="Insulin" required="required" />
25       <input type="text" name="bmi" placeholder="BMI" required="required" />
26       <input type="text" name="diab_pedigree_func" placeholder="Diabetes Pedigree Function" required="required" />
27       <input type="text" name="age" placeholder="Age" required="required" />
28
29       <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
30     </form>
31
32     <br>
33     <br>
34     {{ prediction_text }}
35
36   </div>
37   
38
39 </body>
40 </html>
41
```

## Flask app

Web app endpoint:

```
Help app.py - week_5 - Visual Studio Code
app.py 1 x model.py 2
app.py > predict_api
1 import numpy as np
2 from flask import Flask, request, render_template, jsonify
3 import pickle
4 import pandas as pd
5
6 app = Flask(__name__)
7 model = pickle.load(open('model.pkl', 'rb'))
8
9 @app.route('/')
10 def home():
11     return render_template('index.html')
12
13 # Web app endpoint
14 @app.route('/predict', methods=['POST'])
15 def predict():
16     ...
17     For rendering results on HTML GUI
18     ...
19     features = [float(x) for x in request.form.values()]
20     features_array = [np.array(features)]
21
22     prediction = model.predict(features_array)[0]
23
24     if prediction == 0:
25         outcome = 'No diabetes'
26     else:
27         outcome = 'Diabetes'
28
29     return render_template('index.html', prediction_text='The predicted diagnosis is: {}'.format(outcome))
30
31 # API endpoint
32 @app.route('/predict_api/')
33 def predict_api():
34     ...
35     For returning results in json format
36     ...
37     model = pickle.load(open('model.pkl', 'rb'))
38     pregnancies = request.args.get('pregnancies')
39     glucose = request.args.get('glucose')
40     blood_press = request.args.get('blood_press')
41     skin_thick = request.args.get('skin_thick')
42     insulin = request.args.get('insulin')
43     bmi = request.args.get('bmi')
44     d_pedigree_f = request.args.get('d_pedigree_f')
45     age = request.args.get('age')
46
```

API endpoint:

```
30
31 # API endpoint
32 @app.route('/predict_api/')
33 def predict_api():
34     """
35     For returning results in json format
36     """
37     model = pickle.load(open('model.pkl', 'rb'))
38     pregnancies = request.args.get('pregnancies')
39     glucose = request.args.get('glucose')
40     blood_press = request.args.get('blood_press')
41     skin_thick = request.args.get('skin_thick')
42     insulin = request.args.get('insulin')
43     bmi = request.args.get('bmi')
44     d_pedigree_f = request.args.get('d_pedigree_f')
45     age = request.args.get('age')
46
47     df = pd.DataFrame({'pregnancies':[pregnancies], 'glucose':[glucose],
48                       'blood_press':[blood_press], 'skin_thick':[skin_thick],
49                       'insulin':[insulin], 'bmi':[bmi],
50                       'd_pedigree_f':[d_pedigree_f], 'age':[age]})
51
52     prediction = model.predict(df)[0]
53
54     if prediction == 0:
55         outcome = 'No diabetes'
56     else:
57         outcome = 'Diabetes'
58
59     return jsonify({'diabetes':outcome})
60
61
62 if __name__ == "__main__":
63     app.run(debug=True)
```

Github repository:

The screenshot shows the GitHub repository page for 'Noe-GH / diabetes\_predictor\_heroku'. The repository is public and has 9 commits. The commit history table lists files and their commit dates:

File	Commit	Time
static	First commit	1 hour ago
templates	First commit	1 hour ago
Procfile	Seventh commit	4 minutes ago
README.md	Initial commit	1 hour ago
app.py	First commit	1 hour ago
diabetes.csv	First commit	1 hour ago
model.pkl	First commit	1 hour ago
model.py	First commit	1 hour ago
requirements.txt	Sixth commit	12 minutes ago
runtime.txt	Second commit	26 minutes ago

The README.md file content is displayed below the commit history:

```
diabetes_predictor_heroku
```

Diabetes predictor deployed on Heroku with web app and API endpoints.

The right sidebar shows repository statistics: 0 stars, 1 watching, 0 forks. It also includes sections for Releases, Packages, Environments (dg-diabetes-predictor is active), and Languages (CSS 52.0%, Python 27.1%, HTML 20.7%, Procfile 0.2%).

## Heroku dashboard:

Heroku

Jump to Favorites, Apps, Pipelines, Spaces...

Personal > dg-diabetes-predictor

☆ Open app More

GitHub Noe-GH/diabetes\_predictor\_heroku

Overview Resources Deploy Metrics Activity Access Settings

Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features

Pipelines let you connect multiple apps together and promote code between them. [Learn more](#)

Pipelines connected to GitHub can enable review apps, and create apps for new pull requests. [Learn more](#)

Choose a pipeline

Deployment method

Heroku Git  
Use Heroku CLI

GitHub  
Connected

Container Registry  
Use Heroku CLI

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to Noe-GH/diabetes\_predictor\_heroku by Noe-GH

Disconnect...

Releases in the [activity feed](#) link to GitHub to view commit diffs

Automatic deploys

Enables a chosen branch to be automatically

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

Heroku

Jump to Favorites, Apps, Pipelines, Spaces...

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

Enable automatic deploys from GitHub

Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more](#).

Choose a branch to deploy

main

☐ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Enable Automatic Deploys

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more](#)

Choose a branch to deploy

main Deploy Branch

Receive code from GitHub

Build main 64d22aa5

Release phase

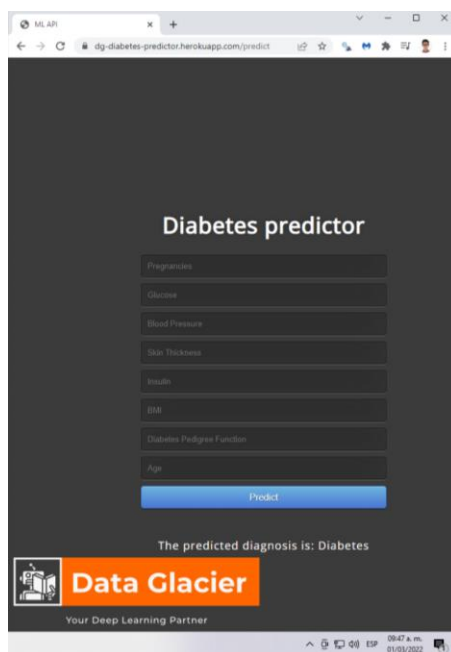
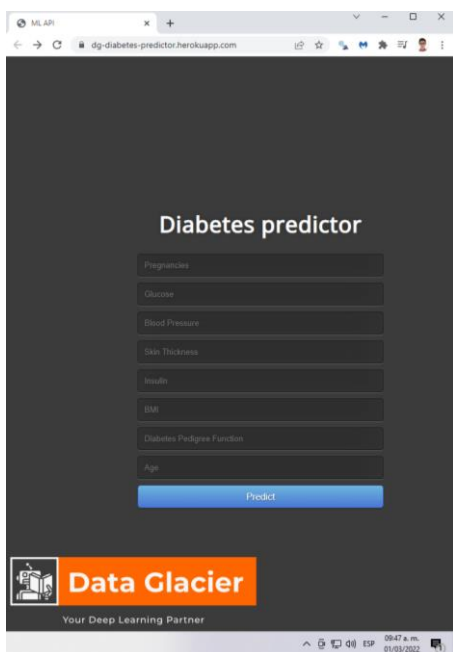
Deploy to Heroku

View

requirements.txt:

```
Help
app.py 1 model.py 2 requirements.txt X
requirements.txt
1 click==8.0.4
2 colorama==0.4.4
3 Flask==2.0.3
4 gunicorn==20.0.4
5 itsdangerous==2.0.1
6 Jinja2==3.0.3
7 joblib==1.1.0
8 MarkupSafe==2.0.1
9 numpy==1.22.2
10 pandas==1.4.1
11 python-dateutil==2.8.2
12 pytz==2021.3
13 scikit-learn==1.0.2
14 scipy==1.8.0
15 six==1.16.0
16 sklearn==0.0
17 threadpoolctl==3.1.0
18 Werkzeug==2.0.3
19
```

Web app on Heroku:



API on Heroku:

