Deployment on Flask &
Cloud and API deployment

Name: Yihsuan Sun
Email: yihsuan.sun88@gmail.com
Internship Batch: LISUM09
Submission date: 06/03/2022
Submitted to: Data Glacier Internship

Data:

toy_datset.csv

Goal:

- Deploy the model on flask
- To predict a person is ill or not
- Deploy the model on any open source cloud eg Heroku

Step 1: Data preprocessing

- Load the dataset
- Replace city name with number
- Replace gender with number

Desktop > FlaskPratice > app.py > ...

```python
from flask import Flask, render_template, request, jsonify
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import pickle
import numpy as np


# Load dataset
df = pd.read_csv("toy_dataset.csv")
# read first few rows
print(df.head())

# replace
df = df.replace(
    to_replace=['Austin', "Boston", "Dallas", "Los Angeles", "Mountain View", "New York City", "San Diego", "Washington D.C."],
    value=[1,2,3,4,5,6,7,8,])
#print(df.head())
#print(df.tail())

df = df.replace(
    to_replace=['Male', "Female"],
    value=[0,1])

```

Step 2:

- Select independent and dependent variables, and then split the data into training and testing
- Feature scaling
- Fit model
- Pickle

```python
# Select independent and dependent vairables\
X = df[["City", "Gender", "Age", "Income"]]
y = df["Illness"]

# Splting the data into traing and testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature scaling
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# model
classifier = RandomForestClassifier()

# fit model
classifier.fit(X_train, y_train)

# pickle
pickle.dump(classifier, open("model.pkl", "wb"))

```

Step 3:

- Deploy the model on flask

```python
app = Flask(__name__)

model = pickle.load(open("model.pkl", "rb"))

@app.route("/")
def home():
    return render_template("index.html")

@app.route("/predict", methods = ['POST'])
def predict():
    float_features = [float(x) for x in request.form.values()]
    features = [np.array(float_features)]
    prediction = model.predict(features)

    return render_template("index.html", prediction_text = "Is the preson Ill? (Yes or No)  {}".format(prediction))


if __name__ == '__main__':
    app.run(port = 3000, debug=True)
```

Step 4:

- Create documents designed to be displayed in a web browser (html)

```html
<!DOCTYPE html>
<html>
    <head>
        <title>
            Tutorial
        </title>
    </head>

    <body>
        <div class = "login">
            <h1 class="text-center">
                Illness prediction
            </h1>

            <form action="{{ url_for('predict')}}"method="post">
                <input type="text" name="City" placeholder="Austin(1)/Boston(2)/Dallas(3)/Los Angeles(4)/Mountain View(5)/New York City(6)/San Diego(7)/Washingtor
                <input type="text" name="Gender" placeholder="Male(0)/Female(1)" required="required" />
                <input type="text" name="Age" placeholder="age" required="required" />
                <input type="text" name="Income" placeholder="income" required="required" />

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

            <br>
            <br>
            {{prediction_text}}

        </div>
    </body>
</html>
```

Web:

Enter the city, gender, age, and income to get the prediction result, the result will show whether a person is ill or not. (Yes or No)

This is an example.

Before adding:

Illness prediction

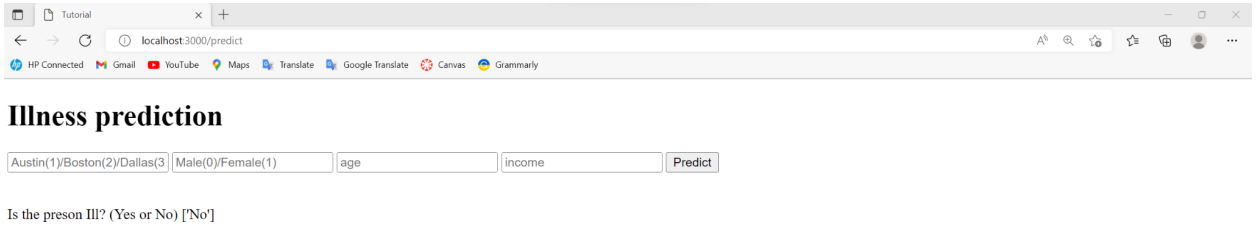
Austin(1)/Boston(2)/Dallas(3 | Male(0)/Female(1) | age | income | Predict

Enter the city, gender, age, and income

Illness prediction

8 | 0 | 66 | 3000 | Predict

Get the prediction



Heroku:

- Create an account and login
- Create new app and enter app name and the region
- install gunicorn

```
sunni@MSI MINGW64 /c/Week5/DataGlacierWeek5 (main)
$ pip install gunicorn
Collecting gunicorn
  Downloading gunicorn-20.1.0-py3-none-any.whl (79 kB)
     ---------------------------------------- 79.5/79.5 kB 2.2 MB/s eta 0:00:00
Requirement already satisfied: setuptools>=3.0 in c:\program files\windowsapps\p
ythonsoftwarefoundation.python.3.10_3.10.1264.0_x64__qbz5n2kfra8p0\lib\site-pack
ages (from gunicorn) (58.1.0)
Installing collected packages: gunicorn
  WARNING: The script gunicorn.exe is installed in 'C:\Users\sunni\AppData\Local
\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-pa
ckages\Python310\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warn
ing, use --no-warn-script-location.
Successfully installed gunicorn-20.1.0
WARNING: There was an error checking the latest version of pip.
```

- Add Procfile and requirements.txt

```
sunni@MSI MINGW64 /c/Week5/DataGlacierWeek5 (main)
$ touch Procfile

sunni@MSI MINGW64 /c/Week5/DataGlacierWeek5 (main)
$ pip freeze > requirements.txt
```

- Add in some code to help connection between heroku and github

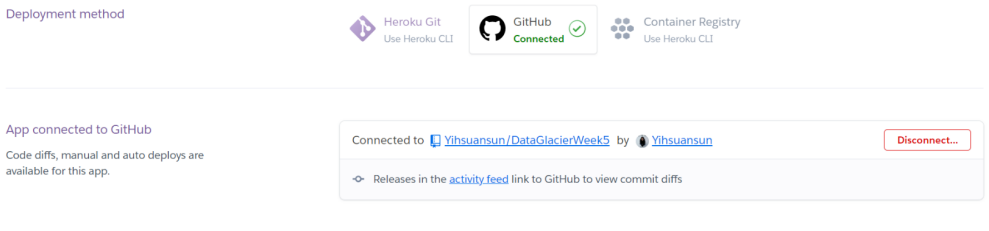
```
web: gunicorn app:app
```

- Results got from requirements.txt file

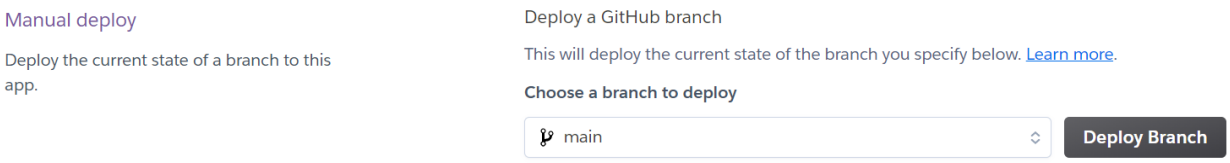
C: > Week5 > DataGlacierWeek5 > requirements.txt

```
click==8.1.3
colorama==0.4.4
cycler==0.11.0
Flask==2.1.2
fonttools==4.33.3
gunicorn==20.1.0
itsdangerous==2.1.2
Jinja2==3.1.2
joblib==1.1.0
kiwisolver==1.4.2
MarkupSafe==2.1.1
matplotlib==3.5.2
numpy==1.22.4
packaging==21.3
pandas==1.4.2
Pillow==9.1.1
pyparsing==3.0.9
python-dateutil==2.8.2
pytz==2022.1
scikit-learn==1.1.1
scipy==1.8.1
six==1.16.0
threadpoolctl==3.1.0
Werkzeug==2.1.2
```

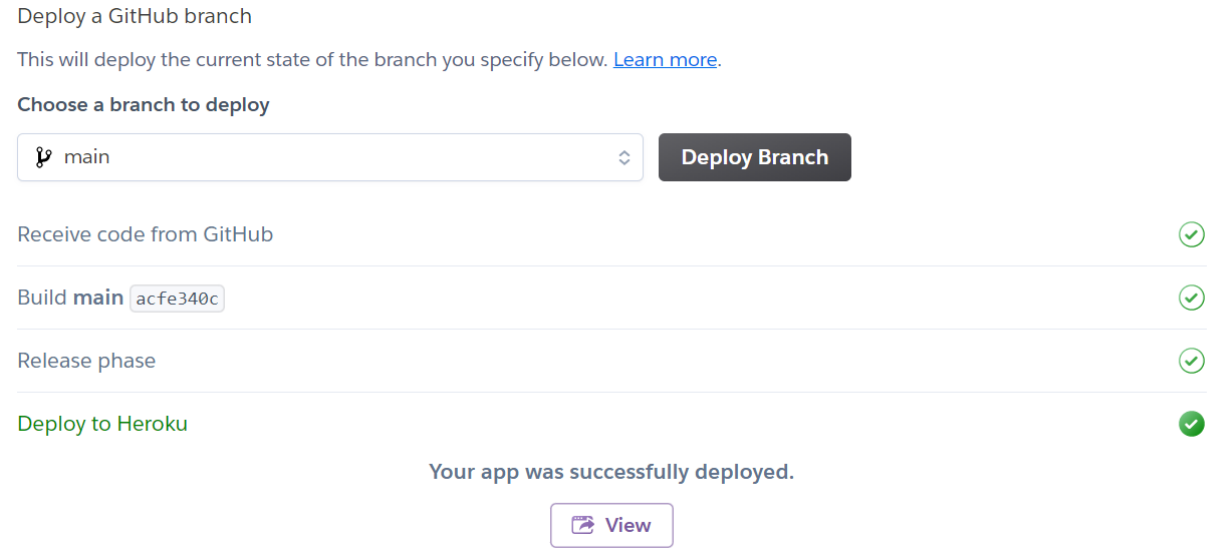
- Connect to github repo (DataGlacierWeek5)



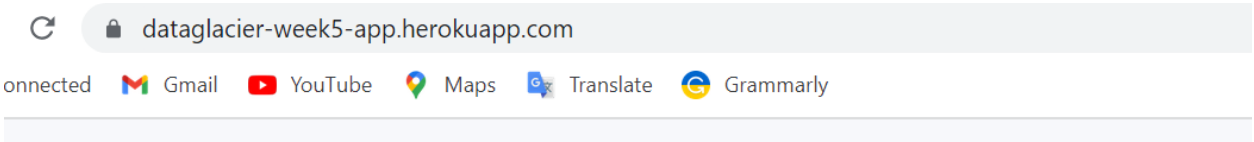
- Deploy branch



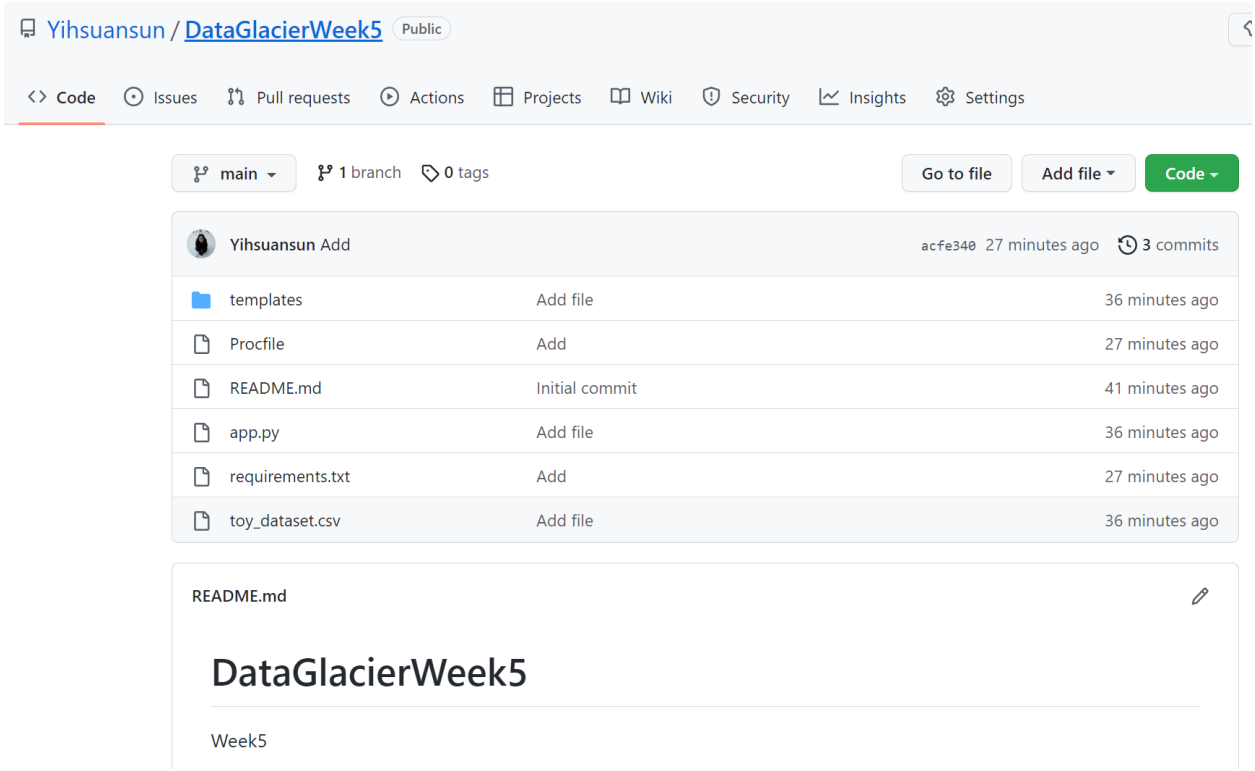
- Successfully deployed



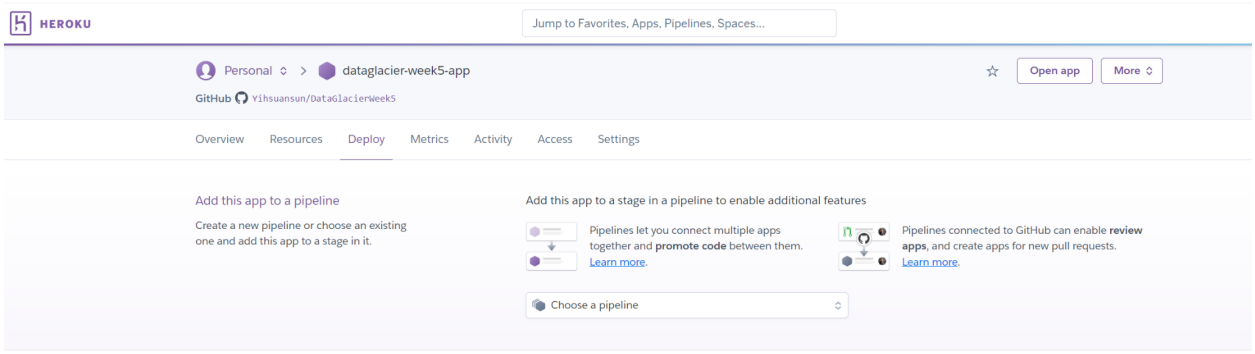
- finalize everything

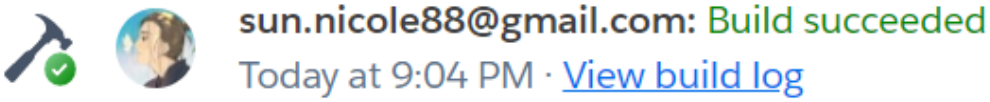


github:



Heroku:


HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
Personal
dataglacier-week5-app
GitHub Yihsuansun/DataGlacierWeek5
Open app
More
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 Yihsuansun/DataGlacierWeek5 by Yihsuansun
Disconnect...
Releases in the activity feed link to GitHub to view commit diffs


sun.nicole88@gmail.com: Build succeeded
Today at 9:04 PM · View build log