Cloud and API deployment

Name: Nonhlanhla Luphade

Batch Code: LISP01

Submission Date: 3/16/21

Submitted to: Data Glacier

Steps

1. Select any toy data
2. Train and save the model
3. Create a web app using Flask
4. Commit the code to GitHub
5. Create an account in Heroku
6. Link GitHub to Heroku
7. Deploy the model

Step 1: The Data

	total_bill	sex	smoker	day	time	size	tip
0	12.16	1.0	Yes	Friday	Lunch	2	2.20
1	21.50	1.0	No	Sunday	Dinner	4	3.50
2	10.33	0.0	No	Thursday	Lunch	2	2.00
3	14.78	1.0	No	Sunday	Dinner	2	3.23
4	18.04	1.0	No	Sunday	Dinner	2	3.00
..	...	...	...	...	...	...	...
194	18.28	1.0	No	Thursday	Lunch	2	4.00
195	17.29	1.0	No	Thursday	Lunch	2	2.71
196	18.43	1.0	No	Sunday	Dinner	4	3.00
197	18.78	0.0	No	Thursday	Dinner	2	3.00
198	15.98	0.0	No	Friday	Lunch	3	3.00

[199 rows x 7 columns]

- The tip data was used for this process.
- Only 4 columns were considered to keep the model simple (total_bill, sex, size and tip)

Step 1: The Model

- A linear model was used and saved for the deployment purpose.

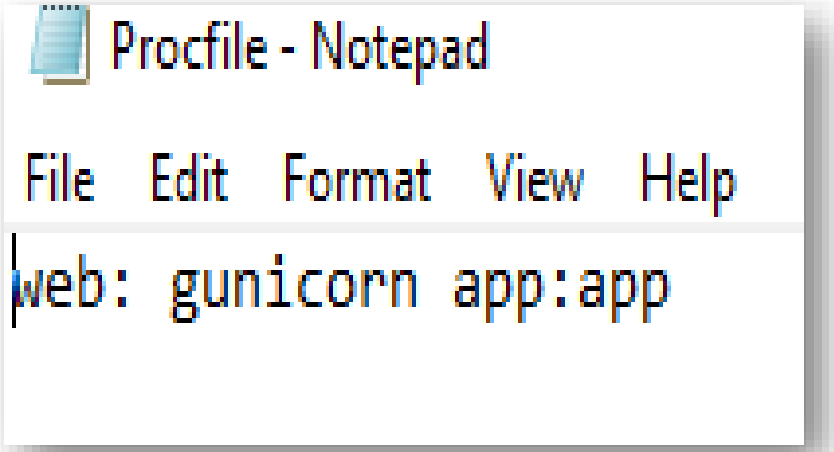
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle

df = pd.read_csv('my_tipdata.csv')
pd.DataFrame(df,columns=['total_bill','sex','smoker','day','time','size','tip'])
df.sex = df.sex.astype('category')

X = df[['total_bill','sex','size']]
Y = df['tip']

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X, Y)
pickle.dump(regressor, open('model.pkl','wb'))
model = pickle.load(open('model.pkl','rb'))
```

Step 2: The App

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

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():

    int_features = [float(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Tip should be $ {}'.format(output))


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

The app.py file is used to create a web app using Flask.

Step 2: The Requirements file and Procfile



The Procfile gives Heroku the instructions on which file should be executed first.

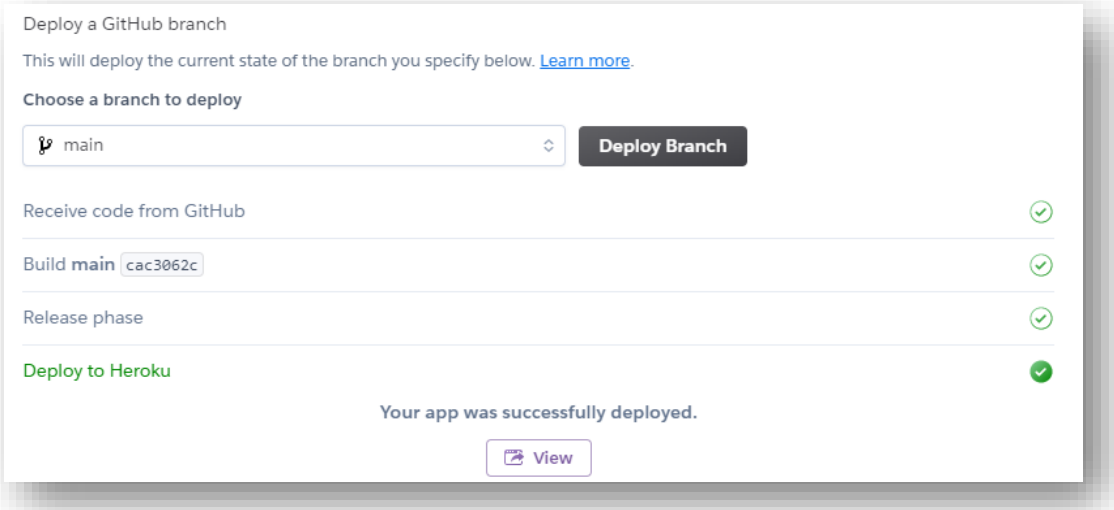
```
Flask==1.1.1
gunicorn==19.9.0
itsdangerous==1.1.0
Jinja2==2.10.1
MarkupSafe==1.1.1
Werkzeug==0.15.5
numpy>=1.9.2
scipy>=0.15.1
scikit-learn>=0.18
matplotlib>=1.4.3
pandas>=0.19
```

The requirements file contains the required libraries that have to be installed in the environment for the deployment in Heroku.

Step 3: Files on GitHub

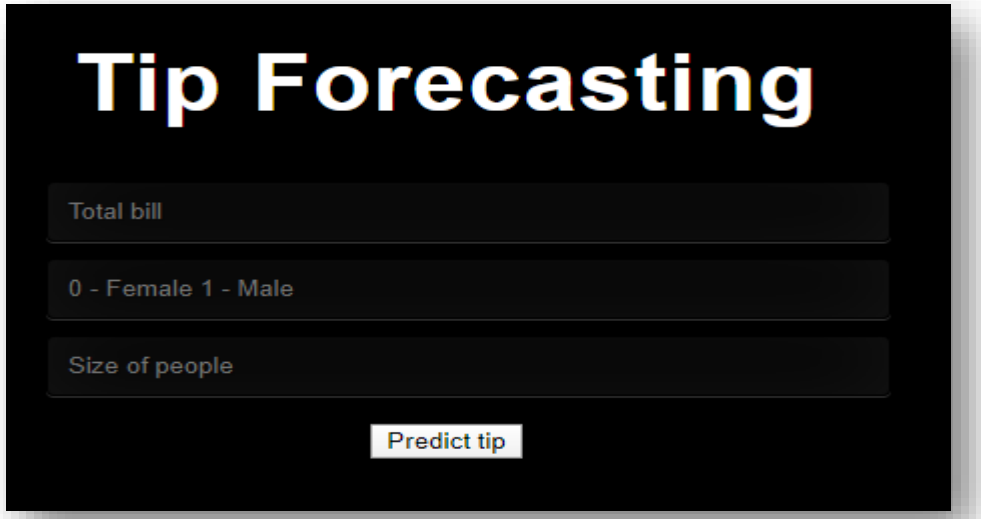
shelovescode000 addfiles	`cac3062` 25 minutes ago	2 commits
static/css	addfiles	25 minutes ago
templates	addfiles	25 minutes ago
Procfile	addfiles	25 minutes ago
README.md	Initial commit	1 hour ago
app.py	addfiles	25 minutes ago
model.pkl	addfiles	25 minutes ago
model.py	addfiles	25 minutes ago
my_tipdata.csv	addfiles	25 minutes ago
requirements.txt	addfiles	25 minutes ago

Step 5-7:Deployment

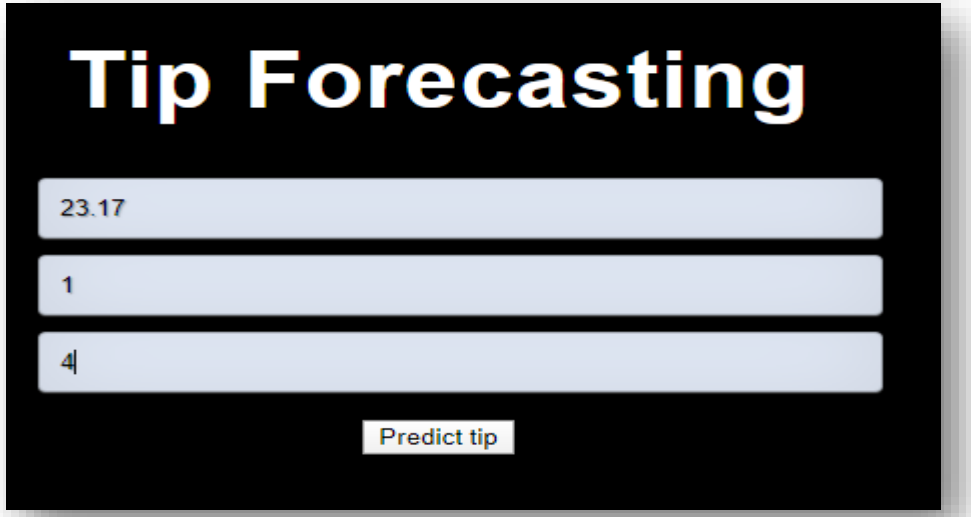


GitHub was linked to Heroku and deployed.

View app



Try examples



Predict