Week 5: Cloud and API deployment

Name: Negar
Batch Code: LISUM14
Submission Date: November 5, 2022
Submitted to: Data Glacier

Dataset:

	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000	36.525000	45.100000	19.050000
max	296.400000	49.600000	114.000000	27.000000

Split the data into 70% train and 30% test

```
# Train Test Split
from sklearn.model_selection import train_test_split
X = df.drop('Sales',axis=1)
y = df['Sales']

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_state=23)
```

✓ 1.3s Python

Building Model

Build a model that forecasts sales based on the money spent on various marketing platforms.

```python
# Building Model
from sklearn.linear_model import LinearRegression
```

✓ 0.1s Python

```python
lin_reg = LinearRegression()
lin_reg.fit(X_train,y_train)
```

✓ 0.5s Python

Model Evaluation:

```python
# prediction
y_pred = lin_reg.predict(X_test)
```

✓ 0.8s Python

```python
#Evaluation
from sklearn.metrics import mean_squared_error
from sklearn.metrics import accuracy_score
from sklearn.metrics import r2_score
```

✓ 0.5s Python

```python
print(lin_reg.__class__.__name__,'MSE:', mean_squared_error(y_test, y_pred))
print(lin_reg.__class__.__name__,'R2 Score:', r2_score(y_test, y_pred))
```

✓ 0.9s Python

```
LinearRegression MSE: 2.806741947581631
LinearRegression R2 Score: 0.9210102489166259
```

Save the Model using Pickle:

```python
#Save the Model
import pickle
```

✓ 0.2s Python

```python
pickle.dump(lin_reg, open("model.pkl", 'wb'))
```

✓ 0.6s Python

Deploy the model on flask (web app):

App.py

The app.py file runs the flask web application

index.html app.py

Week_5 > app.py > predict

```python
import numpy as np
from flask import Flask, request, 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():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [int(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='Sales should be {}'.format(output))


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

```

Index.html

index.html app.py

Week_5 > templates > index.html > html > body > div.login > br

```
</head>

<body>
 <div class="login">
  <h1>Sales Prediction</h1>
  How much sales can we anticipate generating if we invest a specific amount of money in each adve
  <br>
  <br>

    <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
    <input type="text" name="TV" placeholder="TV" required="required" />
        <input type="text" name="Radio" placeholder="Radio" required="required" />
    <input type="text" name="Newspaper" placeholder="Newspaper" 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:
</body>
</html>
```

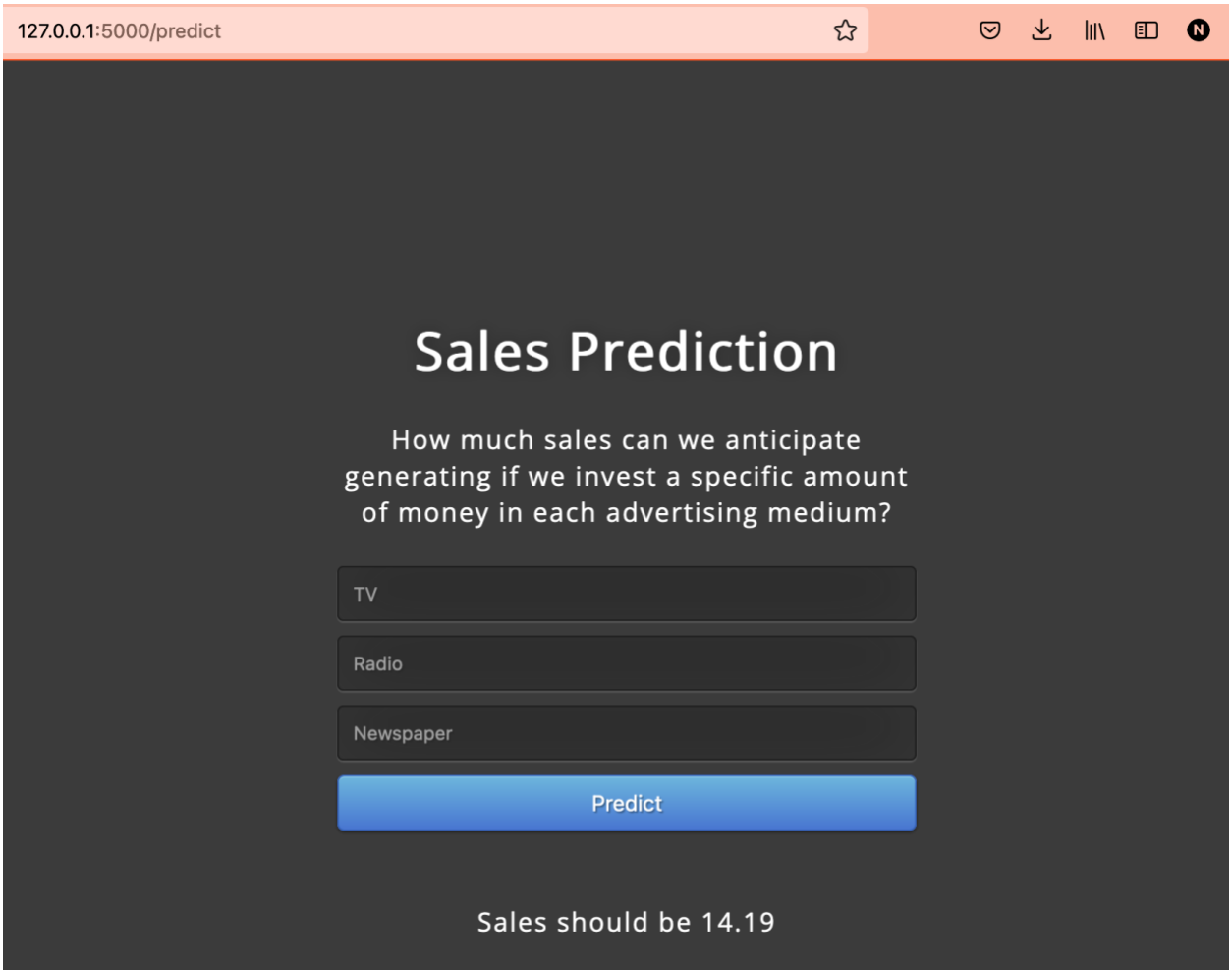
Running the app

-- **>** python app.py

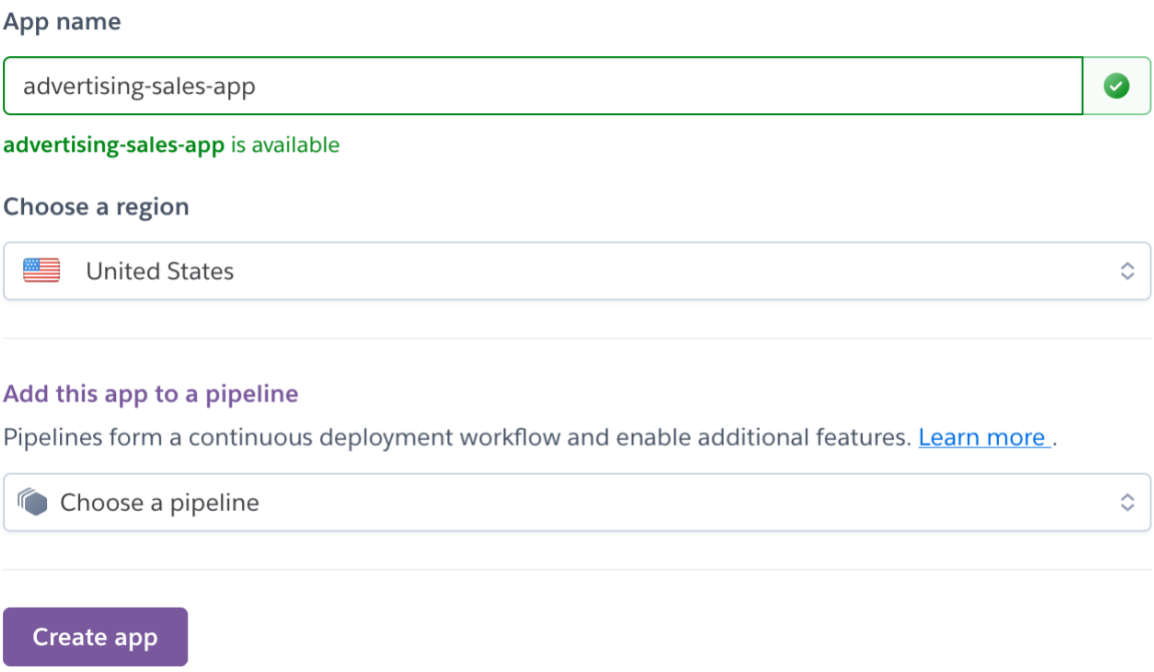
TERMINAL

```
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * 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
```

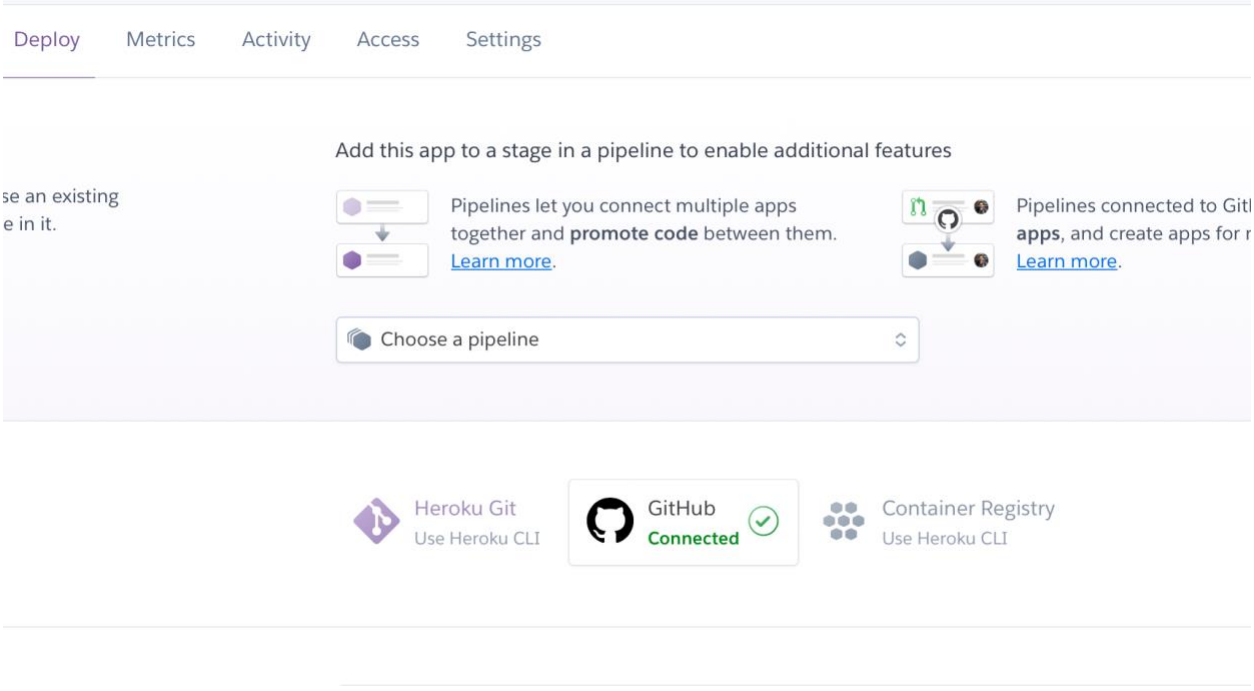
Now we could open http://127.0.0.1:5000/ link to a web browser and see the app

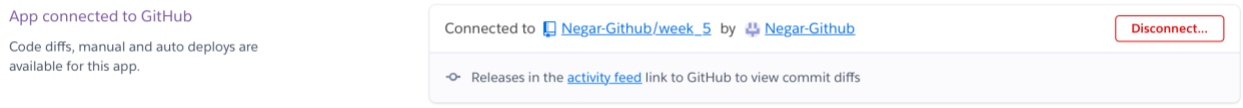

127.0.0.1:5000/predict
Sales Prediction
How much sales can we anticipate generating if we invest a specific amount of money in each advertising medium?
TV
Radio
Newspaper
Predict
Sales should be 14.19

Deploy the Model on Heroku



Connect to the github





Click on the view to open the link to the advertising app:
https://sales-app-advertising.herokuapp.com