

# 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.

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

✓ 0.1s

Python

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

✓ 0.5s

Python

## Model Evaluation:

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

✓ 0.8s

Python

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

✓ 0.5s

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:

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

✓ 0.2s

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
1  import numpy as np
2  from flask import Flask, request, render_template
3  import pickle
4
5  app = Flask(__name__)
6  model = pickle.load(open('model.pkl', 'rb'))
7
8
9  @app.route('/')
10 def home():
11     return render_template('index.html')
12
13
14 @app.route('/predict', methods=['POST'])
15 def predict():
16     '''
17     For rendering results on HTML GUI
18     '''
19     int_features = [int(x) for x in request.form.values()]
20     final_features = [np.array(int_features)]
21     prediction = model.predict(final_features)
22
23     output = round(prediction[0], 2)
24
25     return render_template('index.html', prediction_text='Sales should be {}'.format(output))
26
27
28 if __name__ == "__main__":
29     app.run(debug=True)
30
```

## Index.html

```
index.html x app.py
Week_5 > templates > index.html > html > body > div.login > br
12 </head>
13
14 <body>
15 <div class="login">
16 <h1>Sales Prediction</h1>
17 How much sales can we anticipate generating if we invest a specific amount of money in each adve
18 <br>
19 <br>
20
21 <!-- Main Input For Receiving Query to our ML -->
22 <form action="{{ url_for('predict')}}"method="post">
23 <input type="text" name="TV" placeholder="TV" required="required" />
24 <input type="text" name="Radio" placeholder="Radio" required="required" />
25 <input type="text" name="Newspaper" placeholder="Newspaper" required="required" />
26
27 <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
28 </form>
29
30 <br>
31 <br>
32 {{ prediction_text }}
33
34 </div>
35 
38 </html>
39
```

## 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



# Sales Prediction

How much sales can we anticipate generating if we invest a specific amount of money in each advertising medium?

Sales should be 14.19

## Deploy the Model on Heroku

App name



**advertising-sales-app** is available

Choose a region



United States



Add this app to a pipeline

Pipelines form a continuous deployment workflow and enable additional features. [Learn more](#).



Choose a pipeline



Create app

## Connect to the github

Deploy

Metrics

Activity

Access

Settings

se an existing  
e 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 **apps**, and create apps for r [Learn more](#).



Choose a pipeline



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 [Negar-Github/week\\_5](#) by [Negar-Github](#)

Disconnect...

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

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

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 29d277d8



Release phase



Deploy to Heroku



Your app was successfully deployed.

 View