

Name: Shreya Ramachandra

Batch No: LISUM16

Submission Date: 3rd January 2023

Submitted to: Data Glacier

Title: Flask Deployment on AWS – EC2 Instance (Prediction Salary Analysis)

Salary Prediction Model Deployment on Flask

```
D:\Internship\Data Glacier\Week 5\app.py

model.py X app.py X index.html X request.py X

1 import numpy as np
2 from flask import Flask, request, jsonify, render_template
3 import pickle
4
5 app = Flask(__name__)
6 model = pickle.load(open('model.pkl', 'rb'))
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12 @app.route('/predict', methods=['POST'])
13 def predict():
14     '''
15     For rendering results on HTML GUI
16     '''
17     int_features = [int(x) for x in request.form.values()]
18     final_features = [np.array(int_features)]
19     prediction = model.predict(final_features)
20
21     output = round(prediction[0], 2)
22
23     return render_template('index.html', prediction_text='Employee Salary should be $ {}'.format(output))
24
25
26 if __name__ == "__main__":
27     app.run(host='0.0.0.0', port=8080)
```

D:\Internship\Data Glacier\Week 5\model.py



model.py X

app.py X

index.html X

request.py X

```
1 # Importing the libraries
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import pandas as pd
5 import pickle
6
7 dataset = pd.read_csv('https://raw.githubusercontent.com/ShreyaRamachandra/Deploy-machine-Learning-model-using-flask/main/hiring.csv')
8
9 dataset['experience'].fillna(0, inplace=True)
10
11 dataset['test_score'].fillna(dataset['test_score'].mean(), inplace=True)
12
13 X = dataset.iloc[:, :3]
14
15 #Converting words to integer values
16 def convert_to_int(word):
17     word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
18                 'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
19     return word_dict[word]
20
21 X['experience'] = X['experience'].apply(lambda x : convert_to_int(x))
22
23 y = dataset.iloc[:, -1]
24
25 #Splitting Training and Test Set
26 #Since we have a very small dataset, we will train our model with all available data.
27
28 from sklearn.linear_model import LinearRegression
29 regressor = LinearRegression()
30
31 #Fitting model with training data
32 regressor.fit(X, y)
33
34 # Saving model to disk
35 pickle.dump(regressor, open('model.pkl', 'wb'))
36
37 # Loading model to compare the results
38 model = pickle.load(open('model.pkl', 'rb'))
39 print(model.predict([[2, 9, 6]]))
```

Requirements for Model Deployment



main ▾

[Deployment-of-ML-model-in-AWS-EC2-Instance](#) / requirements.txt



ShreyaRamachandra Create requirements.txt



1 contributor

11 lines (11 sloc) | 177 Bytes

```
1 Flask==1.1.1
2 gunicorn==19.9.0
3 itsdangerous==1.1.0
4 Jinja2==2.10.1
5 MarkupSafe==1.1.1
6 Werkzeug==0.15.5
7 numpy>=1.9.2
8 scipy>=0.15.1
9 scikit-learn>=0.18
10 matplotlib>=1.4.3
11 pandas>=0.19
```

Create AWS account and access EC2 Instance

The screenshot shows the AWS Management Console interface. On the left is a navigation menu with options like EC2 Dashboard, Events, Tags, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, and Volumes. The main content area is titled 'Instances (1/1) Info'. It features a search bar, a filter for 'Instance state = running', and a table of instances. The instance 'MLDeployTest' (i-0e4ecec61e57aac2b) is listed as 'Running' with a 't2.micro' instance type. Below the table, the details for this instance are shown, including its inbound and outbound security group rules. The inbound rules table has two entries: one for port 22 (TCP) and one for all ports (All). The outbound rules table has one entry for all ports (All). At the bottom of the console, there is a footer with 'Feedback', a language selection prompt, and copyright information for Amazon Web Services, Inc.

Name	Security group rule ID	Port range	Protocol	Source	Security groups
-	sgr-0a990020d5f5face6	22	TCP	0.0.0.0/0	launch-wizard-2
-	sgr-033bb6c64a604318d	All	All	0.0.0.0/0	FullAccess

Name	Security group rule ID	Port range	Protocol	Destination	Security groups
-	2 IDs	All	All	0.0.0.0/0	launch-wizard-2 Full

The screenshot shows the 'Connect to instance' page in the AWS Management Console. The page title is 'Connect to instance' with an 'Info' link. Below the title, it says 'Connect to your instance i-0e4ecec61e57aac2b (MLDeployTest) using any of these options'. There are four tabs: 'EC2 Instance Connect', 'Session Manager', 'SSH client', and 'EC2 serial console'. The 'SSH client' tab is selected. The page displays the instance ID 'i-0e4ecec61e57aac2b (MLDeployTest)' and provides a list of steps to connect to the instance using an SSH client. The steps are: 1. Open an SSH client. 2. Locate your private key file. The key used to launch this instance is MLDeployTest.pem. 3. Run this command, if necessary, to ensure your key is not publicly viewable. 4. Connect to your instance using its Public DNS: ec2-18-217-140-114.us-east-2.compute.amazonaws.com. An example command is provided: ssh -i "MLDeployTest.pem" ubuntu@ec2-18-217-140-114.us-east-2.compute.amazonaws.com. A note at the bottom states: 'Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.'

Instance ID
i-0e4ecec61e57aac2b (MLDeployTest)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is MLDeployTest.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 MLDeployTest.pem
4. Connect to your instance using its Public DNS:
ec2-18-217-140-114.us-east-2.compute.amazonaws.com

Example:
ssh -i "MLDeployTest.pem" ubuntu@ec2-18-217-140-114.us-east-2.compute.amazonaws.com

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Access the files and update the flask code in WinSCP

Week 5 - ubuntu@ec2-18-217-140-114.us-east-2.compute.amazonaws.com - WinSCP

Local Mark Files Commands Session Options Remote Help

Synchronize Queue Transfer Settings Default

ubuntu@ec2-18-217-140-114.us-east-2.compute.amazonaws.com X New Session

Dr: My Files Upload Edit Properties New

D:\Internship\Data Glacier\Week 5\

Name	Size	Type	Changed
..		Parent directory	2022-12-25 7:07:41 PM
templates		File folder	2022-12-25 2:36:43 PM
.gitignore.txt	1 KB	Text Document	2022-12-25 2:57:07 PM
.py	1 KB	PY File	2022-12-25 8:51:33 PM
1.png	17 KB	PNG File	2022-12-25 2:51:45 PM
2.png	111 KB	PNG File	2022-12-25 11:03:49 PM
3.png	131 KB	PNG File	2022-12-25 11:04:18 PM
4.png	23 KB	PNG File	2022-12-25 11:04:46 PM
5.png	114 KB	PNG File	2022-12-25 11:05:18 PM
6.png	210 KB	PNG File	2022-12-25 11:06:26 PM
app.py	1 KB	PY File	2022-12-25 8:02:30 PM
MLDeployTest.pem	2 KB	PEM File	2022-12-25 6:55:48 PM
MLDeployTest.ppk	2 KB	PPK File	2022-12-25 7:07:41 PM
model.pkl	1 KB	PKL File	2022-12-25 8:21:33 PM
model.py	2 KB	PY File	2022-12-25 2:49:55 PM
request.py	1 KB	PY File	2022-12-25 2:42:04 PM
requirement.txt	1 KB	Text Document	2022-12-25 2:56:04 PM
tests.py	5 KB	PY File	2022-12-25 10:00:44 PM

0 B of 614 KB in 0 of 17

ubuntu /home/ubuntu/ Find Files Download Edit Properties New

Name	Size	Changed	Rights	Owner
..		2022-12-25 6:56:42 PM	rxr-xr-x	root
templates		2022-12-25 7:13:32 PM	rxrwxr-x	ubuntu
app.py	1 KB	2022-12-25 8:02:30 PM	rw-rw-r--	ubuntu
model.pkl	1 KB	2022-12-25 2:49:58 PM	rw-rw-r--	ubuntu
model.py	2 KB	2022-12-25 2:49:55 PM	rw-rw-r--	ubuntu
request.py	1 KB	2022-12-25 2:42:04 PM	rw-rw-r--	ubuntu
requirement.txt	1 KB	2022-12-25 2:56:04 PM	rw-rw-r--	ubuntu

0 B of 2.82 KB in 0 of 6

8 hidden

Run python app.py in Ubuntu

```
ubuntu@ip-172-31-43-215: ~  
Command 'cd-' not found, did you mean:  
  command 'cd5' from deb cd5 (0.1-4)  
  command 'cde' from deb cde (0.1+git9-g551e54d-1.2)  
  command 'cdw' from deb cdw (0.8.1-2)  
  command 'cdi' from deb cdi (2.0.4-1)  
  command 'cdb' from deb tinycdb (0.78build3)  
  command 'cdp' from deb irpas (0.10-9)  
  command 'cdo' from deb cdo (2.0.4-1)  
Try: sudo apt install <deb name>  
ubuntu@ip-172-31-43-215: ~/.local/lib/python3.10/site-packages/jinja2$ cd -  
/home/ubuntu/.local/lib/python3.10/site-packages  
ubuntu@ip-172-31-43-215: ~/.local/lib/python3.10/site-packages$ cd -  
/home/ubuntu/.local/lib/python3.10/site-packages/jinja2  
ubuntu@ip-172-31-43-215: ~/.local/lib/python3.10/site-packages/jinja2$ cd -  
/home/ubuntu/.local/lib/python3.10/site-packages  
ubuntu@ip-172-31-43-215: ~/.local/lib/python3.10/site-packages$ cd -  
/home/ubuntu/.local/lib/python3.10/site-packages/jinja2  
ubuntu@ip-172-31-43-215: ~/.local/lib/python3.10/site-packages/jinja2$ cd -  
/home/ubuntu/.local/lib/python3.10/site-packages  
ubuntu@ip-172-31-43-215: ~/.local/lib/python3.10/site-packages$ cd..  
cd.: command not found  
ubuntu@ip-172-31-43-215: ~/.local/lib/python3.10/site-packages$ cd  
ubuntu@ip-172-31-43-215:~$ python3 app.py  
* Serving Flask app "app" (lazy loading)  
* Environment: production  
  WARNING: This is a development server. Do not use it in a production deployment.  
  Use a production WSGI server instead.  
* Debug mode: off  
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)  
^Cubuntu@ip-172-31-43-215:~$ python3 app.py  
* Serving Flask app "app" (lazy loading)  
* Environment: production  
  WARNING: This is a development server. Do not use it in a production deployment.  
  Use a production WSGI server instead.  
* Debug mode: off  
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)  
184.148.57.127 - - [26/Dec/2022 03:35:59] "GET / HTTP/1.1" 200 -  
184.148.57.127 - - [26/Dec/2022 03:35:59] "GET /favicon.ico HTTP/1.1" 404 -  
/home/ubuntu/.local/lib/python3.10/site-packages/sklearn/base.py:409: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names  
warnings.warn(  
184.148.57.127 - - [26/Dec/2022 03:36:07] "POST /predict HTTP/1.1" 200 -  
^Cubuntu@ip-172-31-43-215:~$ python3 app.py  
* Serving Flask app "app" (lazy loading)  
* Environment: production  
  WARNING: This is a development server. Do not use it in a production deployment.  
  Use a production WSGI server instead.  
* Debug mode: off  
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

Result: Predict Salary Analysis Model deployed on Flask code

←

→

↻

🔒 127.0.0.1:5000

Predict Salary Analysis

Experience

Test Score

Interview Score

Predict

Predict Salary Analysis Model deployed on AWS cloud

←

→

↻

⚠ Not secure | ec2-18-217-140-114.us-east-2.compute.amazonaws.com:8080/predict

Predict Salary Analysis

2

9

9

Predict

Employee Salary should be \$ 59881.82