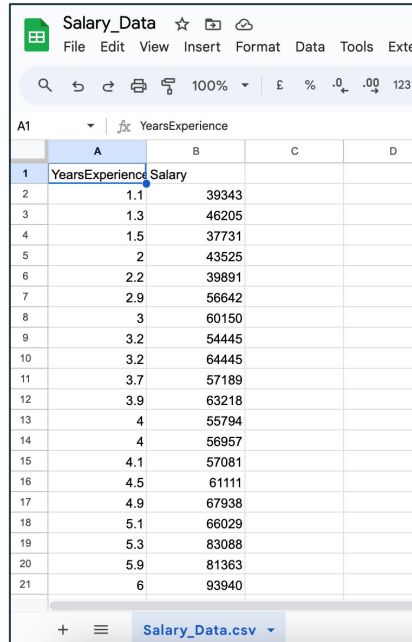


Week 4 - Deployment on Flask

Fabio Pontecchiani - LISUM31 - Submitted to Data Glacier Team on the
27/03/24

Data Selection



Salary_Data

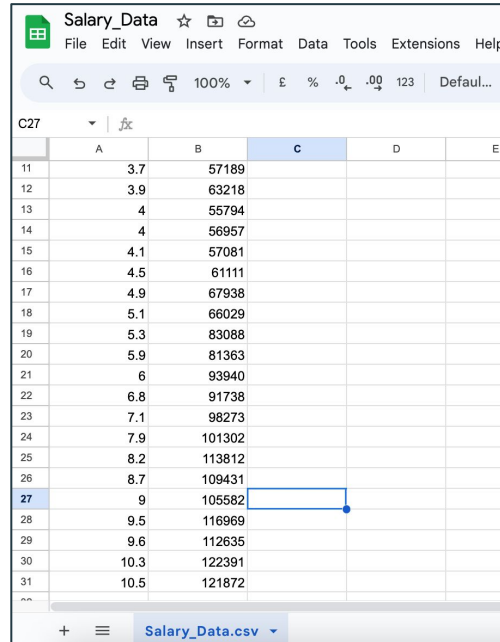
File Edit View Insert Format Data Tools Extensions Help

100% 123

A1 =YearsExperience

	A	B	C	D
1	YearsExperience	Salary		
2	1.1	39343		
3	1.3	46205		
4	1.5	37731		
5	2	43525		
6	2.2	39891		
7	2.9	56642		
8	3	60150		
9	3.2	54445		
10	3.2	64445		
11	3.7	57189		
12	3.9	63218		
13	4	55794		
14	4	56957		
15	4.1	57081		
16	4.5	61111		
17	4.9	67938		
18	5.1	66029		
19	5.3	83088		
20	5.9	81363		
21	6	93940		

Salary_Data.csv



Salary_Data

File Edit View Insert Format Data Tools Extensions Help

100% 123 Default...

C27

	A	B	C	D	E
11	3.7	57189			
12	3.9	63218			
13	4	55794			
14	4	56957			
15	4.1	57081			
16	4.5	61111			
17	4.9	67938			
18	5.1	66029			
19	5.3	83088			
20	5.9	81363			
21	6	93940			
22	6.8	91738			
23	7.1	98273			
24	7.9	101302			
25	8.2	113812			
26	8.7	109431			
27	9	105582			
28	9.5	116969			
29	9.6	112635			
30	10.3	122391			
31	10.5	121872			

Salary_Data.csv

Creation of the Model

```
model.py - /Users/ponte/Salary_predictor/myenv/model.py (3.10.5)
# Simple Linear Regression
'''
This model predicts the salary of the employ based on experience using simple linear regression model.
'''

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle
import requests
import json

# Importing the dataset
dataset = pd.read_csv('Salary_Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state = 0)

# Fitting Simple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predicting the Test set results
y_pred = regressor.predict(X_test)

# Saving model using pickle
pickle.dump(regressor, open('model.pkl', 'wb'))

# Loading model to compare the results
model = pickle.load( open('model.pkl', 'rb'))
print(model.predict([[1.8]]))
```

Creation of Server.py

```
server.py - /Users/ponte/Salary_predictor/myenv/server.py (3.10.5)
# Create API of ML model using flask
'''
This code takes the JSON data while POST request and performs the prediction using loaded model and returns
the results in JSON format.
'''

# Import libraries
import numpy as np
from flask import Flask, request, jsonify
import pickle

app = Flask(__name__)

# Load the model
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/api/', methods=['POST'])
def predict():
    # Get the data from the POST request.
    data = request.get_json(force=True)

    # Make prediction using model loaded from disk as per the data.
    prediction = model.predict([np.array(data['exp'])])

    # Take the first value of prediction
    output = prediction[0]

    return jsonify(output)

if __name__ == '__main__':
    try:
        app.run(port=5000, debug=True)
    except:
        print("Server is exited unexpectedly. Please contact server admin.")
```

Creation of index.html

```
/Users/ponte/Salary_predictor/myenv/templates/index.html
1 <!DOCTYPE html>
2 <html>
3 <!-- From https://codepen.io/frytyler/pen/EGdtg-->
4 <head>
5   <meta charset="UTF-8">
6   <title>ML API</title>
7   <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
8   <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
9   <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
10  <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
11  <link rel="stylesheet" href="{% url_for('static', filename='css/style.css') %}">
12
13 </head>
14
15 <body>
16   <div class="login">
17     <div>
18
19     </div>
20
21     <!-- Main Input For Receiving Query to our ML -->
22     <form action="{% url_for('predict') %}" method="post">
23       <input type="text" name="YearsExperience" placeholder="YearsExperience" required="required" />
24
25       <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
26     </form>
27
28     <br>
29     <br>
30     {{ prediction_text }}
31
32   </div>
33
34 </body>
35 </html>
37
```

Creation of style.css

```
/Users/ponte/Salary_predictor/myenv/static/css/style.css

1 @import url(https://fonts.googleapis.com/css?family=Open+Sans);
2 .btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px; margin-bottom: 0; font-size: 1
3 .btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] { background-color: #e6e6e6; }
4 .btn-large { padding: 9px 14px; font-size: 15px; line-height: normal; -webkit-border-radius: 5px; -moz-border-r
5 .btn:hover { color: #333333; text-decoration: none; background-color: #e6e6e6; background-position: 0 -15px; -v
6 .btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25); color: #ffffff; }
7 .btn-primary.active { color: rgba(255, 255, 255, 0.75); }
8 .btn-primary { background-color: #4a77d4; background-image: -moz-linear-gradient(top, #6eb6de, #4a77d4); backgr
9 .btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-primary.disabled, .btn-primary[disabled] {
10 .btn-block { width: 100%; display: block; }
11
12 * { -webkit-box-sizing: border-box; -moz-box-sizing: border-box; -ms-box-sizing: border-box; -o-box-sizing: border-
13
14 html { width: 100%; height: 100%; }
15
16 body {
17     width: 100%;
18     height: 1000px;
19     font-family: 'Open Sans', sans-serif;
20     background: #092756;
21     color: #fff;
22     font-size: 18px;
23     text-align: center;
24     letter-spacing: 1.2px;
25     overflow-y: scroll !important;
26     overflow-x: hidden;
27 }
28
29 .login {
30     position: absolute;
31     top: 5%;
32     left: 38%;
33
34     width: 400px;
35     height: 800px;
36 }
37
38 .login h1 { color: #fff; text-shadow: 0 0 10px rgba(0, 0, 0, 0.3); letter-spacing: 1px; text-align: center; }
39
40 input {
```

Creation of request.py

● ● ● request.py - /Users/ponte/Salary_predictor/myenv/request.py (3.10.5)

```
import requests

# URL
url = 'http://localhost:5000/api/'

# Change the value of experience that you want to test
payload = {
    'exp': 1.8
}

r = requests.post(url, json=payload)

print(r.json())
```

Creation of the app.py

```
app.py - /Users/ponte/Salary_predictor/myenv/app.py (3.10.5)
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():
    """
    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='Salary is {}'.format(output))

@app.route('/predict_api', methods=['POST'])
def predict_api():
    """
    For direct API calls through request
    """
    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)

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


Enter the virtual environment from terminal (MacOS)

```
(base) fabios-MacBook-Pro:Salary_Predictor ponte$ source myenv/bin/activate
(myenv) (base) fabios-MacBook-Pro:Salary_Predictor ponte$ python server.py
Traceback (most recent call last):
  File "/Users/ponte/Salary_predictor/server.py", line 9, in <module>
    import numpy as np
ModuleNotFoundError: No module named 'numpy'
```

Deployment of the Model on Flask (Web App)

1. Install necessary packages & run Model.py

```
(myenv) (base) fabios-MacBook-Pro:myenv ponte$ pip install scikit-learn
Collecting scikit-learn
  Downloading scikit_learn-1.4.1.post1-cp311-cp311-macosx_10_9_x86_64.whl.metadata (11 kB)
Requirement already satisfied: numpy<2.0, >=1.19.5 in ./lib/python3.11/site-packages (from scikit-learn) (1.26.4)
Collecting scipy>=1.6.0 (from scikit-learn)
  Downloading scipy-1.12.0-cp311-cp311-macosx_10_9_x86_64.whl.metadata (60 kB)
    _____ 60.4/60.4 kB 1.0 MB/s eta 0:00:00
Collecting joblib>=1.2.0 (from scikit-learn)
  Downloading joblib-1.3.2-py3-none-any.whl.metadata (5.4 kB)
Collecting threadpoolctl>=2.0.0 (from scikit-learn)
  Downloading threadpoolctl-3.4.0-py3-none-any.whl.metadata (13 kB)
Downloading scikit_learn-1.4.1.post1-cp311-cp311-macosx_10_9_x86_64.whl (11.6 MB)
    _____ 11.6/11.6 MB 1.8 MB/s eta 0:00:00
Downloading joblib-1.3.2-py3-none-any.whl (302 kB)
    _____ 302.2/302.2 kB 2.1 MB/s eta 0:00:00
Downloading scipy-1.12.0-cp311-cp311-macosx_10_9_x86_64.whl (38.9 MB)
    _____ 38.9/38.9 MB 3.6 MB/s eta 0:00:00
Downloading threadpoolctl-3.4.0-py3-none-any.whl (17 kB)
Installing collected packages: threadpoolctl, scipy, joblib, scikit-learn
Successfully installed joblib-1.3.2 scikit-learn-1.4.1.post1 scipy-1.12.0 threadpoolctl-3.4.0
(myenv) (base) fabios-MacBook-Pro:myenv ponte$ python model.py
[43638.88864165]
```

Deployment of the Model on Flask (Web App)

2. Install other necessary packages & run Server.py

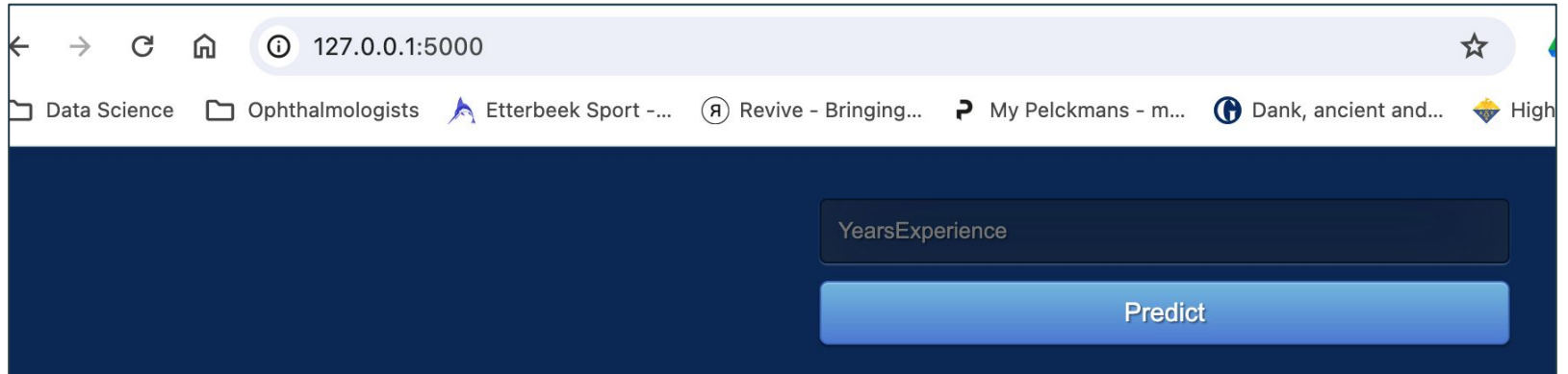
```
(myenv) (base) fabios-MacBook-Pro:myenv ponte$ python server.py
Traceback (most recent call last):
  File "/Users/ponte/Salary_predictor/myenv/server.py", line 10, in <module>
    from flask import Flask, request, jsonify
ModuleNotFoundError: No module named 'flask'
(myenv) (base) fabios-MacBook-Pro:myenv ponte$ pip install flask
Collecting flask
  Downloading flask-3.0.2-py3-none-any.whl.metadata (3.6 kB)
Collecting Werkzeug>=3.0.0 (from flask)
  Downloading werkzeug-3.0.1-py3-none-any.whl.metadata (4.1 kB)
Collecting Jinja2>=3.1.2 (from flask)
  Downloading Jinja2-3.1.3-py3-none-any.whl.metadata (3.3 kB)
Collecting itsdangerous>=2.1.2 (from flask)
  Downloading itsdangerous-2.1.2-py3-none-any.whl.metadata (2.9 kB)
Collecting click>=8.1.3 (from flask)
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker>=1.6.2 (from flask)
  Downloading blinker-1.7.0-py3-none-any.whl.metadata (1.9 kB)
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask)
  Downloading MarkupSafe-2.1.5-cp311-cp311-macosx_10_9_x86_64.whl.metadata (3.0 kB)
Downloading flask-3.0.2-py3-none-any.whl (101 kB)
 101.3/101.3 kB 2.0 MB/s eta 0:00:00
Downloading blinker-1.7.0-py3-none-any.whl (13 kB)
Downloading click-8.1.7-py3-none-any.whl (97 kB)
 97.9/97.9 kB 2.2 MB/s eta 0:00:00
Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Downloading Jinja2-3.1.3-py3-none-any.whl (133 kB)
 133.2/133.2 kB 5.2 MB/s eta 0:00:00
Downloading werkzeug-3.0.1-py3-none-any.whl (226 kB)
 226.7/226.7 kB 4.1 MB/s eta 0:00:00
Downloading MarkupSafe-2.1.5-cp311-cp311-macosx_10_9_x86_64.whl (14 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, blinker, Werkzeug, Jinja2, flask
Successfully installed Jinja2-3.1.3 MarkupSafe-2.1.5 Werkzeug-3.0.1 blinker-1.7.0 click-8.1.7 flask-3.0.2 itsdangerous-2.1.2
(myenv) (base) fabios-MacBook-Pro:myenv ponte$ python server.py
* Serving Flask app 'server'
* 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
* Debugger is active!
* Debugger PIN: 866-673-361
```

Deployment of the Model on Flask (Web App)

3. Run App.py

```
(myenv) (base) fabios-MacBook-Pro:myenv ponte$ python app.py
* Serving Flask app 'app'
* 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
* Debugger is active!
* Debugger PIN: 866-673-361
```

Copy-Paste the URL to the Browser



Try out the Model!

The image displays two separate user interfaces for testing a model, both set against a dark blue background. Each interface consists of a dark input field, a light blue 'Predict' button, and a text label showing the result.

The top interface has an input field containing the number '12', a 'Predict' button, and a result label that reads 'Salary is 138967.5'.

The bottom interface has an input field containing the number '6', a 'Predict' button, and a result label that reads 'Salary is 82891.85'.