

دوره دیتا ساینس کاربردی

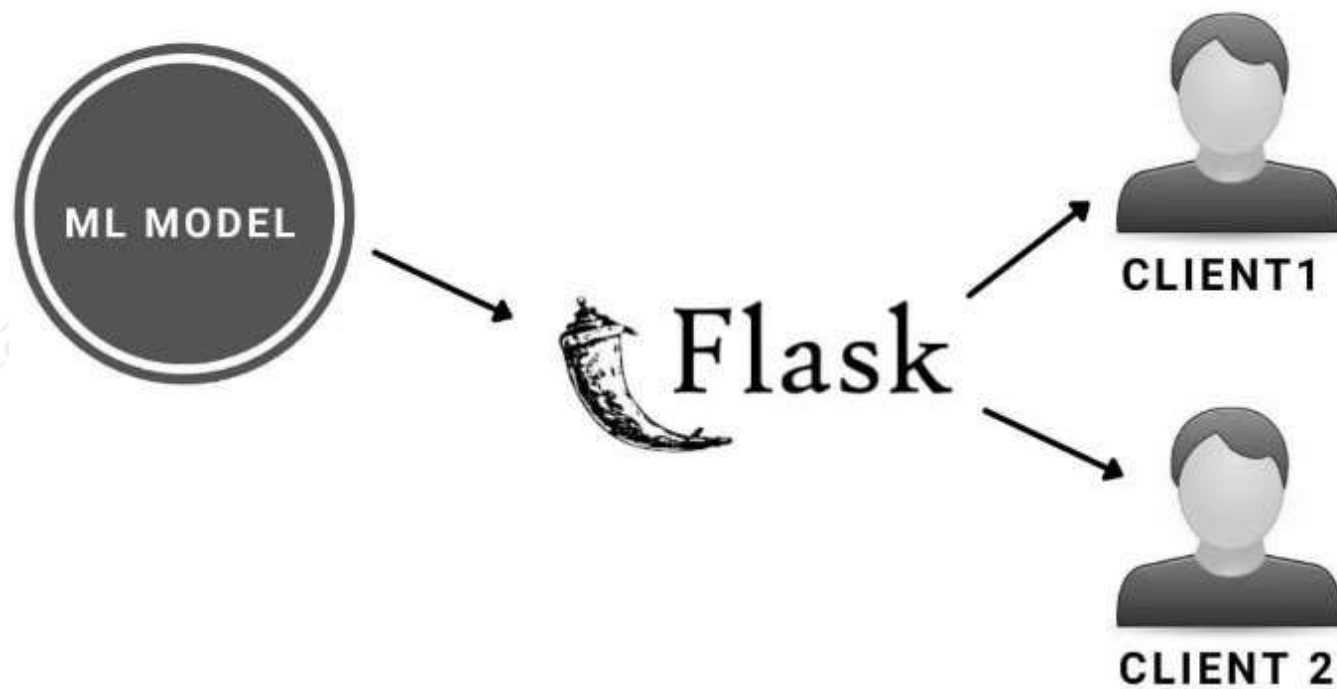
ML Model Deployment

—● dataroadmap ●—

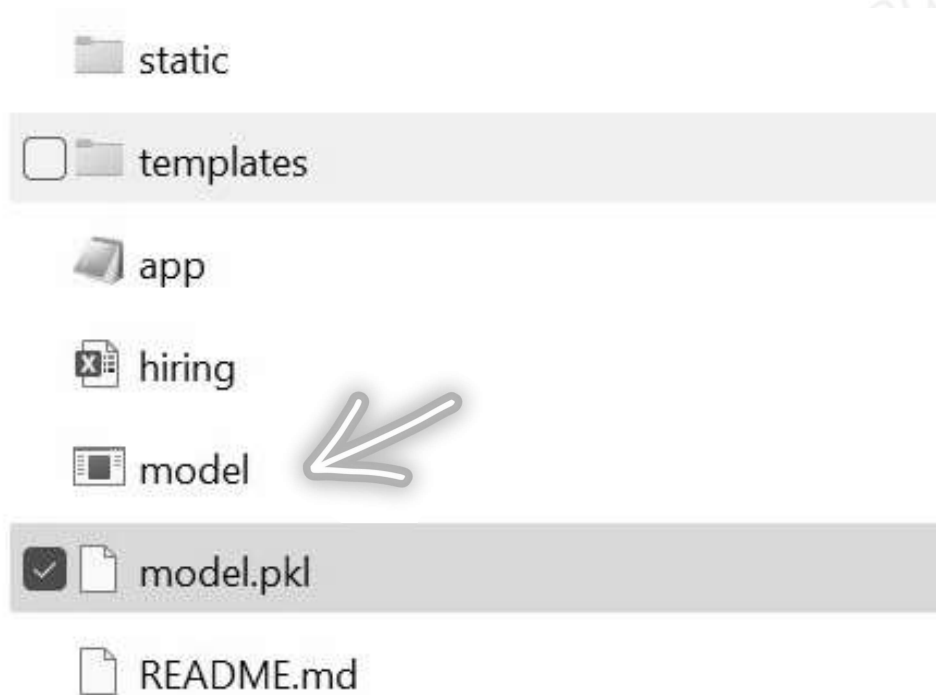
مدرس: مونا حاتمی

جلسه چهاردهم

ML Model Deployment



Open File



Open model.ipynb

```
] In [1]: # Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle
```

```
] In [2]: df = pd.read_excel('hiring.xlsx')
```

```
] In [3]: df
```

```
Out[3]:
```

	experience	test_score	interview_score	salary
0	NaN	8.0	9.0	50000.0
1	NaN	8.0	6.0	45000.0
2	five	6.0	7.0	60000.0
3	five	10.0	10.0	65000.0

Fill missing values

```
4]: ▶ df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 96 entries, 0 to 95  
Data columns (total 4 columns):  
#   Column          Non-Null Count  Dtype    
---  ---            -  
0   experience       72 non-null    object   
1   test_score       84 non-null    float64  
2   interview_score  96 non-null    float64  
3   salary           96 non-null    float64  
dtypes: float64(3), object(1)  
memory usage: 3.1+ KB
```

```
5]: ▶ df['experience'].fillna(0, inplace=True)
```

```
df['test_score'].fillna(df['test_score'].mean(), inplace=True)
```

Define X,y

```
X=df.iloc[:, :3]  
y = df.iloc[:, -1]
```

```
X
```

```
[7]:
```

	experience	test_score	interview_score
0	0	8.0	9.0
1	0	8.0	6.0
2	five	6.0	7.0
3	two	10.0	10.0

Feature Engineering

```
▶ #Converting words to integer values  
def convert_to_int(word):  
    word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,  
                 'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}  
    return word_dict[word]  
  
▶ X['experience'] = X['experience'].apply(lambda x : convert_to_int(x))
```

Train- Test- Split

```
▶ #Splitting Training 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=0.1, random_state = 101)
```


Train the Model

```
# Creating and Training the Model  
from sklearn.linear_model import LinearRegression  
lm = LinearRegression()  
  
#Fitting model with training data  
lm.fit(X_train, y_train)
```

```
[15]:  
LinearRegression
```

Test the Model

```
]:  ► #test the model  
    predictions = lm.predict(X_test)
```

```
]:  ► from sklearn import metrics
```

```
]:  ► print('MAE:', metrics.mean_absolute_error(y_test, predictions))  
    print('MSE:', metrics.mean_squared_error(y_test, predictions))  
    print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
MAE: 2165.3395707998825
```

```
MSE: 8112909.576770613
```

```
RMSE: 2848.3169726648425
```

Save the Model

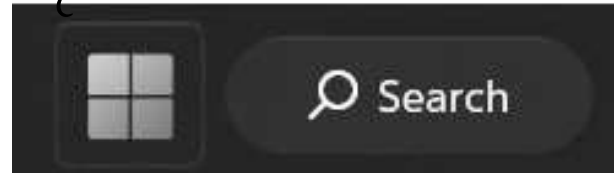
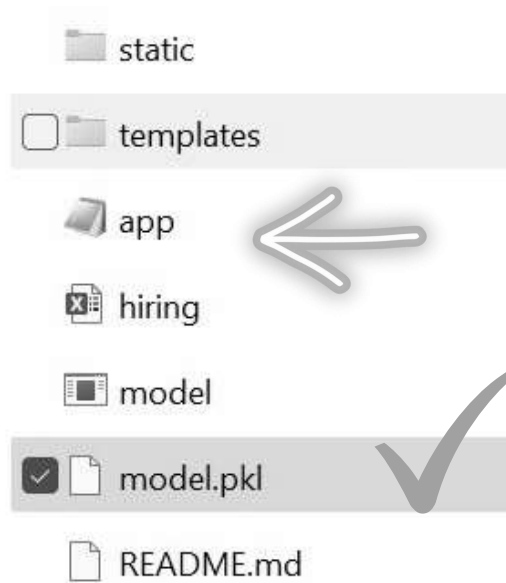
```
▶ # Saving model to disk  
pickle.dump(lm, open('model.pkl', 'wb'))
```

Load the Model

```
▶ # Loading model to compare the results  
model = pickle.load(open('model.pkl','rb'))  
print(model.predict([[2, 9, 6]]))
```

```
[57354.85194693]
```

Command Prompt



Command Prompt

Run app.py

```
Command Prompt
Microsoft Windows [Version 10.0.22000.1335]
(c) Microsoft Corporation. All rights reserved.

C:\Users\monah>cd c:\ 1
c:\>cd dataroadmap 2
c:\dataroadmap>cd monogram2 3
c:\dataroadmap\Monogram2>cd week-14
c:\dataroadmap\Monogram2\week-14>cd Deployment-flask-master
c:\dataroadmap\Monogram2\week-14\Deployment-flask-master>python app.py
```

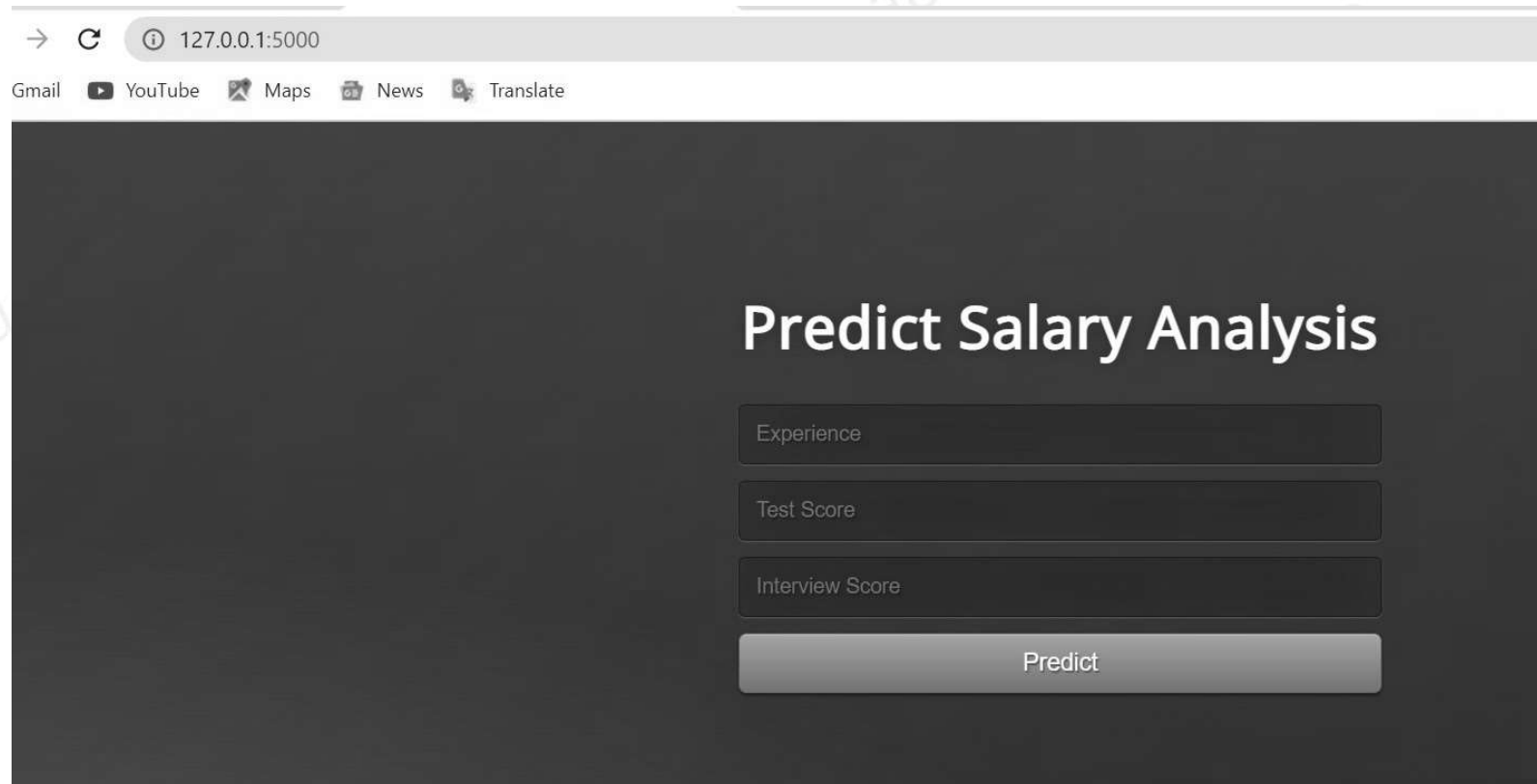
This PC > Windows (C:) > dataroadmap > Monogram2 > week-14 > Deployment-flask-master



Copy URL

```
C:\dataroadmap\Monogram2\week-14\Deployment-flask-master>python 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: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 104-371-979
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Enter inputs



The image shows a web browser window with the address bar displaying '127.0.0.1:5000'. Below the address bar, there are links for Gmail, YouTube, Maps, News, and Translate. The main content area has a dark background and features the title 'Predict Salary Analysis' in white. Below the title, there are three input fields labeled 'Experience', 'Test Score', and 'Interview Score'. At the bottom of the form is a 'Predict' button.

→ ↻ ⓘ 127.0.0.1:5000

Gmail YouTube Maps News Translate

Predict Salary Analysis

Experience

Test Score

Interview Score

Predict

Result

Predict Salary Analysis

Experience

Test Score

Interview Score

Predict

Employee Salary should be \$ 76901.79

Web App

- static
- templates
- app
- hiring
- model
- model.pkl
- README.md

Predict Salary Analysis

Experience Test Score Interview Score Predict

{{ prediction_text }}

```
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/c:
```

```
style - Notepad
File Edit View

@import url(https://fonts.googleapis.com/css?family=Open+Sa
.btn { display: inline-block; *display: inline; *zoom: 1; p
13px; line-height: 18px; color: #333333; text-align: center
vertical-align: middle; background-color: #f5f5f5; backgrou
```

1

2

3

app.py

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

```
app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))
```

1

```
@app.route('/')
def home():
    return render_template('index.html')
```

2

```
@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='Employee Salary should be $ {}'.format(output))
```

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



Assignment:

تمرین:

کدهای ارائه شده در درس را بررسی و اجرا کنید.