

WEEK4- DEPLOYMENT ON FLASK

Name: Canmert Demir – Deployment on Flask

Submission date: 25.10.2023

Internship Batch: 30 Sept to 30 Dec 2023

Version: 3.11.4.final.0

Data intake by: Data Glacier

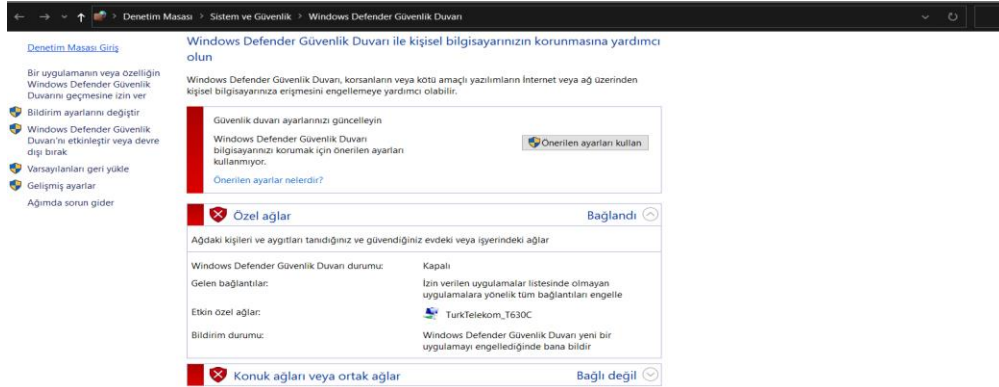
Data intake reviewer: Canmert Demir

Submitted location: <https://github.com/Canmertdemir/Canmertdemir-DataGlacierWeek2>

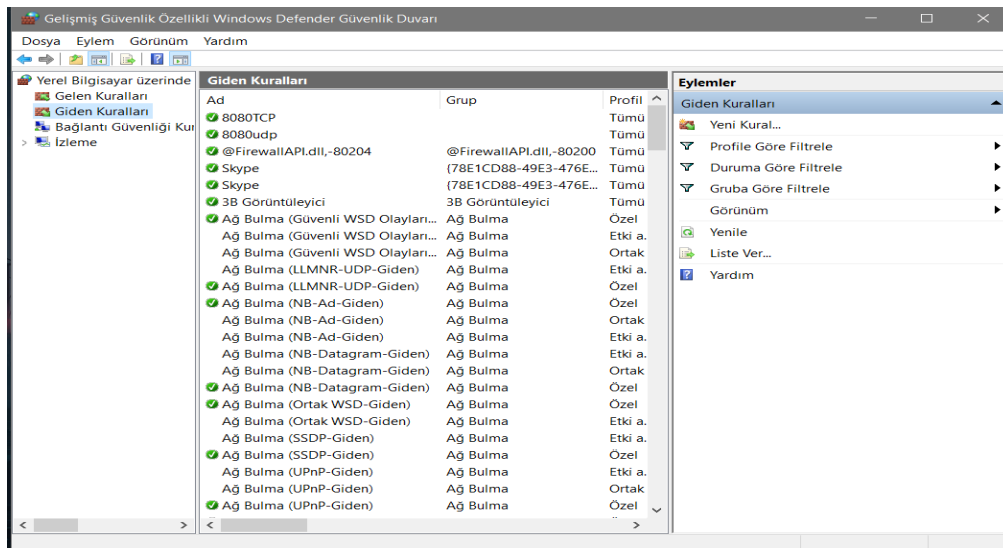
Work Stages are given below.

1) Activation of Localhost:

Sometimes, firewall creates problem for localhost to guaranteed this we close firewall.



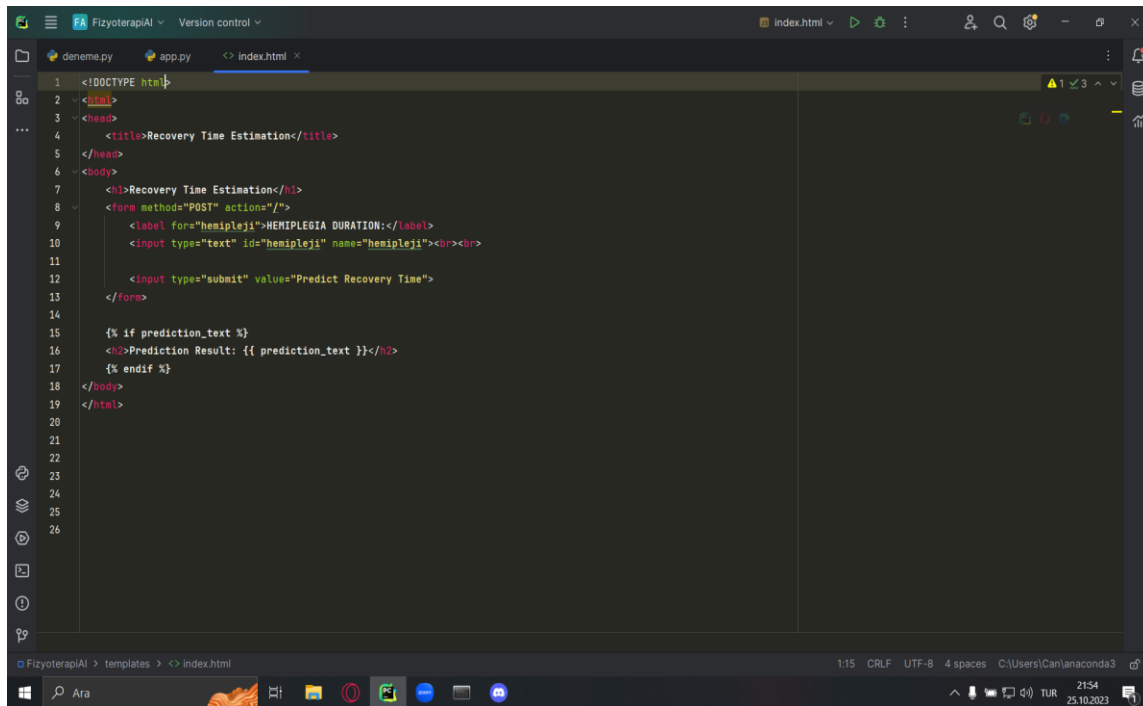
Then, we active local ports TCP and UDP. You can see port first two row of list.



2) Machine Learning Model, Flask Code, HTML Code in Python:

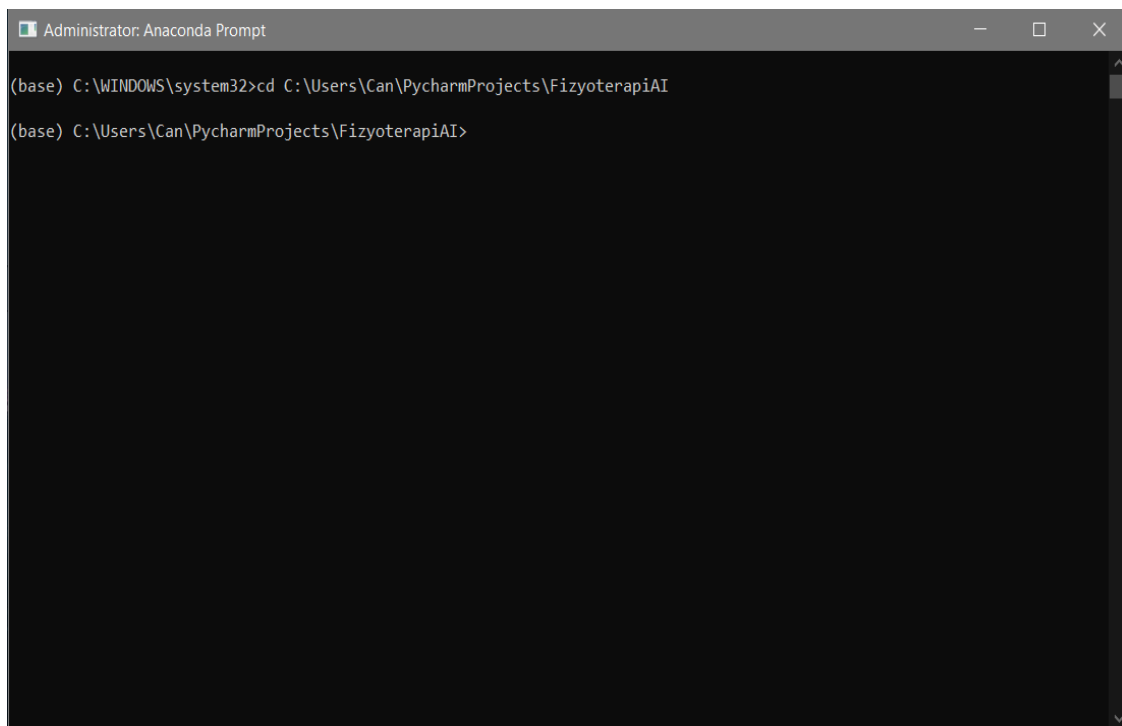
```
FizyoterapiAI Version control
deneme.py app.py index.html
1 import pandas as pd
2 import sklearn.linear_model as lm
3 import sklearn.model_selection as ms
4 import sklearn.metrics as mt
5 import pickle
6
7 data = pd.read_excel("CANSU DEMİR TEZ İSTATİSTİK - SON.xlsx")
8
9 hemipleji = data.loc[:, ["HEMİPLEJİ SÜRESİ"]].values.reshape(-1, 1)
10
11 tinetindenge_son = data["TİNETİNGE DENGİ 2"].values.reshape(-1, 1)
12
13 reg = lm.LinearRegression()
14
15 xtrain, xtest, ytrain, ytest = ms.train_test_split(hemipleji, tinetindenge_son, test_size=0.3, random_state=50)
16
17 # Modelin fitlenmesi aşaması
18 reg.fit(xtrain, ytrain)
19
20 # Modelin tahmin kısmı
21
22 ypredict = reg.predict(xtest)
23
24 print(f"Vücut Kitle Endeksi: {xtest}, Tahmin edilen Dözelme Süresi: {ypredict}")
25
26 pickle.dump(reg, open("denemes.pkl", "wb"))
```

```
FizyoterapiAI Version control
deneme.py app.py index.html
1 from flask import Flask, request, render_template
2 import pickle
3 import numpy as np
4
5 app = Flask(__name__)
6
7 # Edilmiş modeli yükle
8 model = pickle.load(open('denemes.pkl', 'rb'))
9
10 @app.route('/', methods=['GET', 'POST'])
11 def home():
12     if request.method == 'GET':
13         return render_template('index.html')
14     if request.method == 'POST':
15         hemipleji = float(request.form['hemipleji']) # Sadece bir özellik kullan
16         fea = np.array([[hemipleji]])
17         prediction = model.predict(fea)
18         return render_template(template_name_or_list="index.html", prediction_text="Estimated Recovery Time: {}".format(prediction[0]))
19
20 if __name__ == '__main__':
21     app.run(port=5000, debug=True)
```



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Recovery Time Estimation</title>
5 </head>
6 <body>
7 <h1>Recovery Time Estimation</h1>
8 <form method="POST" action="/">
9 <label for="hemiplegia">HEMIPLEGIA DURATION:</label>
10 <input type="text" id="hemiplegia" name="hemiplegia"><br><br>
11
12 <input type="submit" value="Predict Recovery Time">
13 </form>
14
15 {% if prediction_text %}
16 <h2>Prediction Result: {{ prediction_text }}</h2>
17 {% endif %}
18 </body>
19 </html>
20
21
22
23
24
25
26
```

3) Run flask code by using Anaconda Prompt



```
Administrator: Anaconda Prompt

(base) C:\WINDOWS\system32>cd C:\Users\Can\PycharmProjects\FizyoterapiAI

(base) C:\Users\Can\PycharmProjects\FizyoterapiAI>
```

```
Anaconda Prompt

(base) C:\Users\Can>C:\Users\Can\PycharmProjects\FizyoterapiAI
```

```
Administrator: Anaconda Prompt - flask run --host=0.0.0.0

(base) C:\WINDOWS\system32>cd C:\Users\Can\PycharmProjects\FizyoterapiAI

(base) C:\Users\Can\PycharmProjects\FizyoterapiAI>flask run --host=0.0.0.0
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.1.103:5000
Press CTRL+C to quit
```

4) Machine Learning Result in Localhost:5000 & some sample predictions

Recovery Time Estimation

HEMIPLEGIA DURATION:

Recovery Time Estimation

HEMIPLEGIA DURATION:

Prediction Result: Estimated Recovery Time: [16.39911759]

