

Breast Tissue Prediction

Intern name: **Seyed Mojtaba Hejazi**

Submission date: 29/07/2022

Data Glacier Company

Here are my steps to do this deployment.

1- Extract the dataset from UCI. That is about breast tissue.

2- Build our model:

```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
import pickle
```

```
dataset = pd.read_csv('BreastTissue.csv')
```

```
x= dataset.iloc[:, :9]
```

```
y= dataset.iloc[:, -1]
```

```
KNC=KNeighborsClassifier()
```

```
KNC.fit(x,y)
```

```
pickle.dump(KNC, open('model.pkl','wb'))
```

```
model=pickle.load(open('model.pkl','rb'))print(model.predict([290.4551412,0.1
44164196,0.053058009,74.63506664,1189.545213,15.93815436,35.70333099,
65.54132446,330.2672929]))
```

3- Coding our web application using flask:

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

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

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

@Myapp.route('/predict',methods=['POST'])
def predict():

    float_features = [float(x) for x in request.form.values()]
    final_features = [np.array(float_features)]
    prediction = model.predict(final_features)

    output =str(prediction)

    return render_template('index.html', prediction_text='Breast Tissue would be $
{}'.format(output))

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

4- Coding HTML file for using as a template for our application

```
<!DOCTYPE html>

<html >

<head>

  <meta charset="UTF-8">

  <title>ML API</title>

  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet'
type='text/css'>

  <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet'
type='text/css'>

  <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet'
type='text/css'>

  <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>

  <link rel="stylesheet" href="{ { url_for('static', filename='css/style.css') } }">


</head>


<body>

  <div class="login">

    <h1>Predict Breat Tissue</h1>


    <!-- Main Input For Receiving Query to our ML -->

    <form action="{ { url_for('predict') } }" method="post">

      <input type="text" name="I0" placeholder="Impedivity (ohm) at zero
frequency" required="required" />

      <input type="text" name="PA500" placeholder="phase angle at 500 KHz"
required="required" />

      <input type="text" name="HFS" placeholder="high-frequency slope of
phase angle" required="required" />
```

```
<input type="text" name="DA" placeholder="impedance distance
between spectral ends" required="required" />
```

```
<input type="text" name="Area" placeholder="area under spectrum"
required="required" />
```

```
<input type="text" name="A/DA" placeholder="area normalized by DA"
required="required" />
```

```
<input type="text" name="Max IP" placeholder="maximum of the
spectrum" required="required" />
```

```
<input type="text" name="DR" placeholder="distance between I0 and real
part of the maximum frequency point" required="required" />
```

```
<input type="text" name="P" placeholder="length of the spectral curve"
required="required" />
```

```
<button type="submit" class="btn btn-primary btn-block btn-
large">Predict</button>
```

```
</form>
```

```
<br>
```

```
<br>
```

```
{{ prediction_text }}
```

```
</div>
```

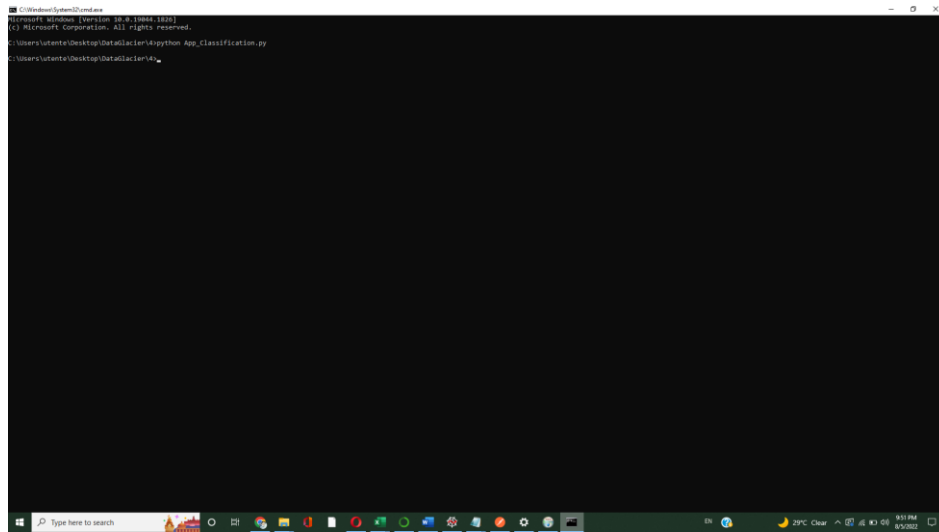
```

```

```
</body>
```

```
</html>
```

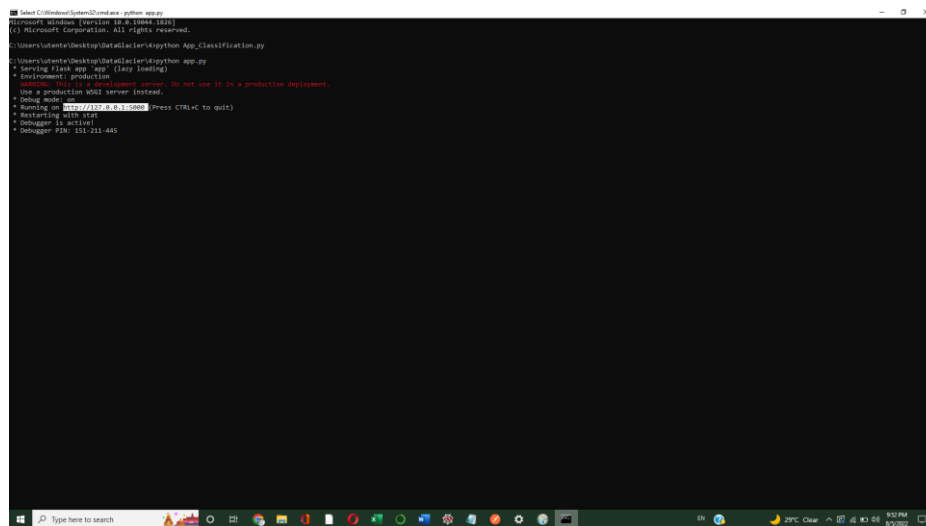
5- Now from CMD, we need to run first our model and then our web application.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.1023]
(c) Microsoft Corporation. All rights reserved.

C:\Users\utente\Desktop\Datascience>python App_Classification.py

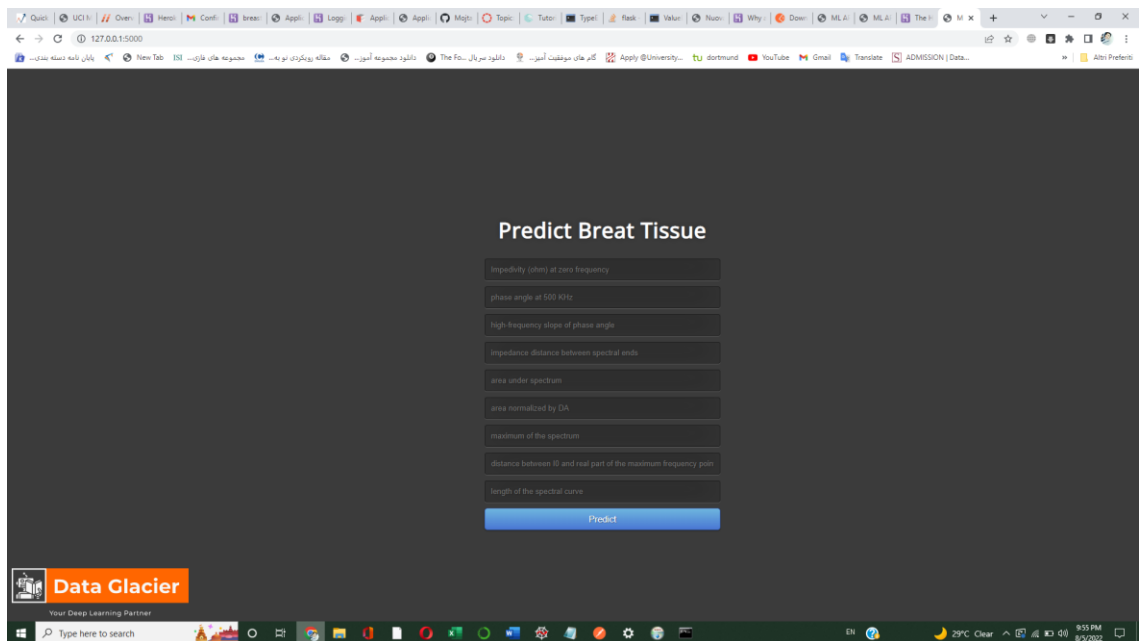
C:\Users\utente\Desktop\Datascience>
```



```
C:\Users\utente\Desktop\Datascience>python app.py
Microsoft Windows [Version 10.0.19041.1023]
(c) Microsoft Corporation. All rights reserved.

C:\Users\utente\Desktop\Datascience>python App_Classification.py
C:\Users\utente\Desktop\Datascience>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
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 151-211-045
```

As you can see, we have a local host: <http://127.0.0.1:5000> and we need to open this in Chrome.



And Finally we can do the prediction.

