

Week 4 Assignment

Name: **Enias Vontas**

Batch Code: **LISUM01**

Submission Date: **3rd of July, 2021**

Submitted to:

<https://github.com/EniasVontas/Assignments/upload/main/Week4>

We consider the 'insurance.csv' dataset, which was obtained from kaggle (<https://www.kaggle.com/mirichoi0218/insurance/>) and has 1338 rows and 7 features: age, gender, BMI, number of children, whether the person is a smoker or not, the region that they live and the insurance cost.

We will try to predict a person's insurance cost based on 5 of the above features, as the region that someone lives will not be considered.

The first five rows of our dataset and the types of features we have:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

```
age          int64
sex          object
bmi          float64
children     int64
smoker       object
region       object
charges      float64
dtype: object
```

We observe that the 'bmi' features is a float, as well as the 'charges' feature. The 'sex', 'smoker' and 'region' features are objects. As we said, for simplicity, the feature 'region' will not be considered so we create dummy variables for the other two features. Our new dataset with the independent variables and our dependent ('charges') variable are:

	age	bmi	children	sex_male	smoker_yes
0	19	27.900	0	0	1
1	18	33.770	1	1	0
2	28	33.000	3	1	0
3	33	22.705	0	1	0
4	32	28.880	0	1	0

	charges
0	16884.92400
1	1725.55230
2	4449.46200
3	21984.47061
4	3866.85520

We would like to create our linear regression model and then use 'pickle' library to save it on our disk. So we create a .py file where we import the necessary libraries for our regression and the pickling that will follow. We load the data and create the data frame 'X' for our independent variables and the data frame 'Y' for our dependent variable. Then we modify the categorical features of our 'X' data frame to dummy variables. Lastly, we fit the linear regression model.

```
import pandas as pd
import numpy as np
import pickle
from sklearn.linear_model import LinearRegression

dataset = pd.read_csv("C:\\Program Files\\Git\\DataSets\\insurance.csv")

X = dataset[["age", "sex", "bmi", "children", "smoker"]]
Y = dataset[["charges"]]

X = pd.get_dummies(data=X, drop_first=True)

model = LinearRegression()
model.fit(X,Y)
```

We then pickle the model in order to save it to the disk and then load it again when we run the .py script in the command line.

```
pickle.dump(model, open('model.pkl', 'wb'))
model = pickle.load(open('model.pkl', 'rb'))
```

Next we create the app.py file in order to initialize the flask app and to render the template of the html file. We load our 'pickled' model and then create two routes, one for the template, and the other for the prediction, where we superimpose floating point numbers, as we can have BMIs with decimal points. Then we predict the insurance cost based on the numbers that we will enter on the html file on our browser. We round the result, on the second decimal place.

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

app = Flask(__name__) #Initialize the flask App
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 = [float(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = prediction[0].round(decimals=2)

    return render_template('index.html',
                           prediction_text='Insurance Charges should be $ {}'.format(output))

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

Next we create the html file based on the lectures that are provided on the internship dashboard.

```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>Insurance Prediction</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 Insurance Charges</h1>

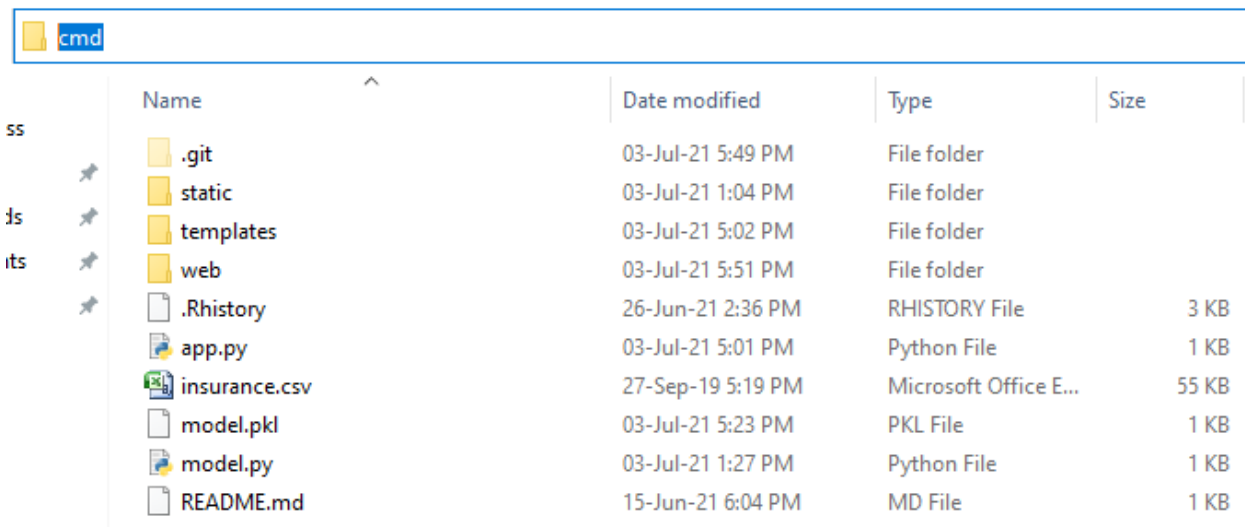
    <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict') }}"method="post">
      <input type="text" name="age" placeholder="Age" required="required" />
      <input type="text" name="bmi" placeholder="BMI" required="required" />
      <input type="text" name="children"
            placeholder="No of Children" required="required" />
      <input type="text" name="sex_male" placeholder="Sex{1: male,0: female}"
            required="required" />
      <input type="text" name="smoker_yes"
            placeholder="Smoker{1: smoker,0: non_smoker}" required="required" />

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

    <br>
    <br>
    {{ prediction_text }}
  </div>
</body>
</html>
```

There is also .css file for styling, which was obtained from the github profile in the url (<https://github.com/MaajidKhan/DeployMLModel-Flask/tree/master/static/css>).

We run the command prompt on the folder where we have saved our model.



	Name	Date modified	Type	Size
ss	.git	03-Jul-21 5:49 PM	File folder	
	static	03-Jul-21 1:04 PM	File folder	
js	templates	03-Jul-21 5:02 PM	File folder	
its	web	03-Jul-21 5:51 PM	File folder	
	.Rhistory	26-Jun-21 2:36 PM	RHISTORY File	3 KB
	app.py	03-Jul-21 5:01 PM	Python File	1 KB
	insurance.csv	27-Sep-19 5:19 PM	Microsoft Office E...	55 KB
	model.pkl	03-Jul-21 5:23 PM	PKL File	1 KB
	model.py	03-Jul-21 1:27 PM	Python File	1 KB
	README.md	15-Jun-21 6:04 PM	MD File	1 KB

Now we run the command 'python model.py' in order to fit our regression model and then the command 'python app.py' to create the flask app. Lastly, we copy the url highlighted and paste it to our browser (in this case, Mozilla Firefox).

```
C:\ Windows [Version 10.0.19041.1052]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\Git\DataSets>python model.py

C:\Program Files\Git\DataSets>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: 136-333-228
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Here we can see the url path, and the boxes that we have to fill in order to predict the insurance cost.




A screenshot of a web browser window. The address bar shows the URL '127.0.0.1:5000'. Below the address bar, there are two tabs: 'Most Visited' and 'Getting Started'. The main content area displays a header with the text: '# -*- coding: utf-8 -*- "" Created on Sat Jul 3 13:05:39 2021 @author: Enias ""'.

Predict Insurance Charges

Age	BMI	No of Children	Sex{1:male,0:female}	Smoker{1:smoker,0:non	Predict
-----	-----	----------------	----------------------	-----------------------	---------

Here we can see the predicted insurance cost of someone who is 30 years old, has a BMI of 23.2, has 1 child, is male and a non smoker. It is predicted to be 3504.21 \$. We can also observe that the url changed and now has '/predict' at the end of it.



A screenshot of a web browser window. The address bar shows the URL '127.0.0.1:5000/predict'. Below the address bar, there are two tabs: 'Most Visited' and 'Getting Started'. The main content area displays a header with the text: '# -*- coding: utf-8 -*- "" Created on Sat Jul 3 13:05:39 2021 @author: Enias ""'.

Predict Insurance Charges

30	23.2	1	1	0	Predict
----	------	---	---	---	---------

Insurance Charges should be \$ [3504.21]