WeekFour

September 22, 2023

1 Week 4: Model Deployment on Flask by Connor Bryson

1.1 Here is the code for my python model with Flask Set up

1.2 I chose a Random Forest Regressor for the data and the data came from the Kaggle Titanic Machine Learning Competition

1.2.1 The model predicts the probability of survivability based on number of siblings and parch.

[ ]:
```
# # -*- coding: utf-8 -*-
# """
# Created on Fri Sep 22 14:32:59 2023

# @author: cdbry
# """

# import numpy as np
# import pandas as pd
# from flask import Flask, request, render_template
# from sklearn.ensemble import RandomForestRegressor

# train_data = pd.read_csv(r"C:\Users\cdbry\Desktop\Data Glacier␣
 ↪Internship\Repositories\Data-Glacier-Internship-Fall-2023\Week␣
 ↪4\Deployment_Flask\data/train.csv")
# test_data = pd.read_csv(r"C:\Users\cdbry\Desktop\Data Glacier␣
 ↪Internship\Repositories\Data-Glacier-Internship-Fall-2023\Week␣
 ↪4\Deployment_Flask\data/test.csv")


# app = Flask(__name__)


# y = train_data["Survived"]
# features = ["SibSp", "Parch"]
# X = pd.get_dummies(train_data[features].dropna(how ='any'))
```

```
# model = RandomForestRegressor(n_estimators=100, max_depth=5, random_state=1)
# model.fit(X, y)


# @app.route('/')

# def home():
#     return render_template("template.html")


# @app.route('/predict', methods = ['POST'])

# def predict():
#     """
#     For rending 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)

#     return render_template('template.html', prediction_text = f"Probability␣
 ↪they survived: {prediction[0]}")


# if __name__ == "__main__":
#     app.run(port = 5000, debug = True)
```

1.3 Next I made the templates folder for the html template and put it with the model

1.4 Then I made the html template which is shown below

[ ]:
```
# <!DOCTYPE html>
# <html >
# <!--From https://codepen.io/frytyler/pen/EGdtg-->
# <head>
#     <meta charset="UTF-8">
#     <title>My Machine Learning 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 Survivability</h1>

#     <!-- Main Input For Recieving Query to our ML -->
#     <form action="{{ url_for('predict')}}" method="post">
#         <input type="text" name="SibSp" placeholder="Siblings"␣
 ↪required="required" />
#         <input type="text" name="Parch" placeholder="Parch"␣
 ↪required="required" />

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

#     <br>
#     <br>
#     {{ prediction_text }}

# </div>

# </body>
# </html>
```

1.5 Below are screenshots of the end product

1.5.1 Home Page