<u>AiEnsured</u>

Ensuring comprehensive validation of AI systems



An article on

FLASK

On Fuel Efficiency Dataset

Submitted by: Rayapureddi Venkata Sri Sai Subhash Intern in Machine Learning May 2023-July 2023

(rayapureddi.subhash@testaing.com)

Flask is a lightweight and popular open-source web application framework written in Python. It is designed to be minimalistic and simple, allowing developers to build web applications quickly and easily. Flask is classified as a microframework, meaning that it provides the basic tools and features needed for web development but leaves the flexibility to add additional libraries and components as needed.

Key features and characteristics of Flask:

- Minimalistic
- Routing
- Lightweight and Extensible
- Built-in Development Server
- RESTful Support
- Active Community and Ecosystem

For the deployment of the model, we need to implement the following files :

- 1. Pickel file of the model.
- 2. Application file(flask).
- 3.Templets(Frond end).

Pickel file of the model:

First, we need to import the pickle library in the model, and dump the model in it and load the model.

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

And download the pickle file into the local system and save it as model.pkl.

Application file(flask):

Develop the flask application in a way that load all the values in the data set according to their data types.

```
if request.method == 'POST':
       MPG = float(request.form['MPG'])
       Acceleration = float(request.form.get('Acceleration'))
       Displacement = float(request.form.get('Displacement'))
       Weight = float(request.form.get('Weight'))
        Cylinders =float( request.form('Cylinders'))
       Model_Year = int(request.form['Model Year'])
       Origin=int(request.form['Origin'])
        # Preprocess numerical features
       MPG encoded = MPG
       Acceleration_encoded = Acceleration
       Displacement_encoded = Displacement
       Weight_encoded=Weight
       Cylinders encoded=Cylinders
       Model Year encoded=Model Year
       Origin encoded=Origin
        data = {
```

```
"MPG": np.array([[MPG_encoded]]),
    "Acceleration": np.array([[Acceleration_encoded]]),
    "Displacement": np.array([[Displacement_encoded]]),
    "Weight": np.array([[Weight_encoded]]),
    "Cylinders": np.array([[Cylinders_encoded]]),
    "Model_Year": np.array([[Model_Year_encoded]]),
    "Origin": np.array([[Origin_encoded]]),
}
```

Also render the html templets into the application and return the predicted value as follows.

```
my_prediction = model.predict([[data]])
return my_prediction
```

Create a HTML file as per your need and save them as templates folder.

Running of flask:

Add the model.pkl, app.py, templates files into one folder and name it as "FLASK".

Now open command prompt and change the directory to where the flask folder is present.

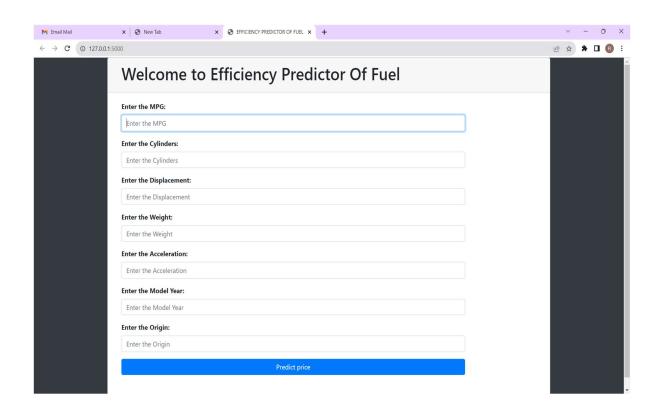
Run the app.py file in the command prompt using the following command:

```
python app.py
```

```
(base) C:\Users\subha>cd C:\Users\subha\OneDrive\Desktop\Flask2

(base) C:\Users\subha\OneDrive\Desktop\Flask2>python app.py
    * Serving Flask app 'app'
    * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
    * Running on http://127.0.0.1:5000
Press CTRL+C to quit
    * Restarting with watchdog (windowsapi)
    * Debugger is active!
    * Debugger PIN: 859-357-585
```

We can see the link below the command, open the link and it will render to your HTML page.



After the entering values we will get the predicted value.