Name: Cloud and API deployment
Report date: 03/06/2024
Internship Batch: LISUM30
Version: 1.0
Data intake by: FNU Mrinal Shasta Rajput

Select data (simple data)

Tabular data details:

Total number of observations	1436
Total number of files	1
Total number of features	5
Base format of the file	.csv
Size of the data	33 KB

The data contains information on the prices of used cars between the years 1998-2004. The price becomes our predicted value and we use distance covered (KM), Horse power (HP), Gear type (Automatic or Manual) and CC of the car to predict the price.

The model

I used a linear regression model to predict car price and saved the model by serializing using Pickle.

```python
# Importing the libraries
import numpy as np
import pandas as pd
import pickle
from sklearn.linear_model import LinearRegression
```
```python
dataset = pd.read_csv('D:/DataGlacier/Used car data(1998-2004).csv')
```
```python
dataset.head
```
```python
X = dataset.drop(columns=['Price'])
y = dataset[['Price']]
```
```python
X.head
```
```python
y.head
```
```python
regressor = LinearRegression()
```
```python
#Fitting model with trainig data
regressor.fit(X, y)
```
```python
# Saving model to disk
pickle.dump(regressor, open('model.pkl','wb'))
```
```python
# Loading model to compare the results
model = pickle.load(open('model.pkl','rb'))
print(model.predict([[72000, 100, 1, 1000]]))
```
```
    [[9233.90967125]]
```

Deploy the model on flask (web app)

Below is the index file that I used:

```
<!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 Used Car Price</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
                <input type="text" name="KM" placeholder="Distance KM" required="required" />
                <input type="text" name="HP" placeholder="Horsepower" required="required" />
                <input type="text" name="Automatic" placeholder="Automatic 1(Yes)/0(No)" required="required" />
                <input type="text" name="CC" placeholder="CCs" required="required" />

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

   <br>
   <br>
   {{ prediction_text }}

 </div>
 <img src="/static/images/Original.svg" style="width: 400px;position: absolute;bottom: 10px;left: 10px;" alt="Company Logo"/>

</body>
</html>
```

App.py file:

```
```python
import numpy as np
from flask import Flask, request,render_template
import pickle
```
```python
app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))
```
```python
@app.route('/')
def home():
    return render_template('index.html')
```
```python
@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering 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)

    output = np.round(prediction[0], 2)

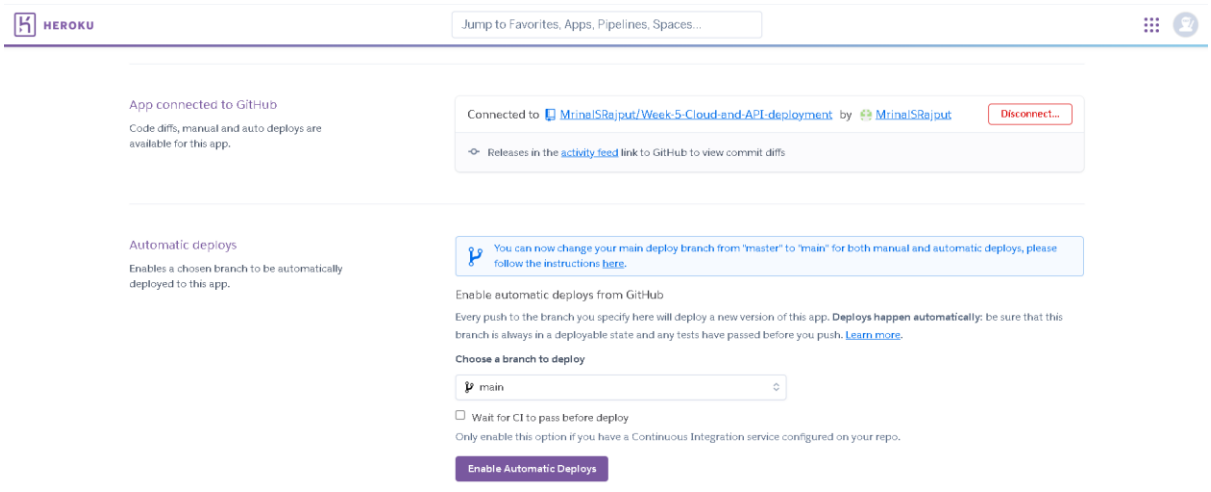
    return render_template('index.html', prediction_text='Price of used car $ {}'.format(output))

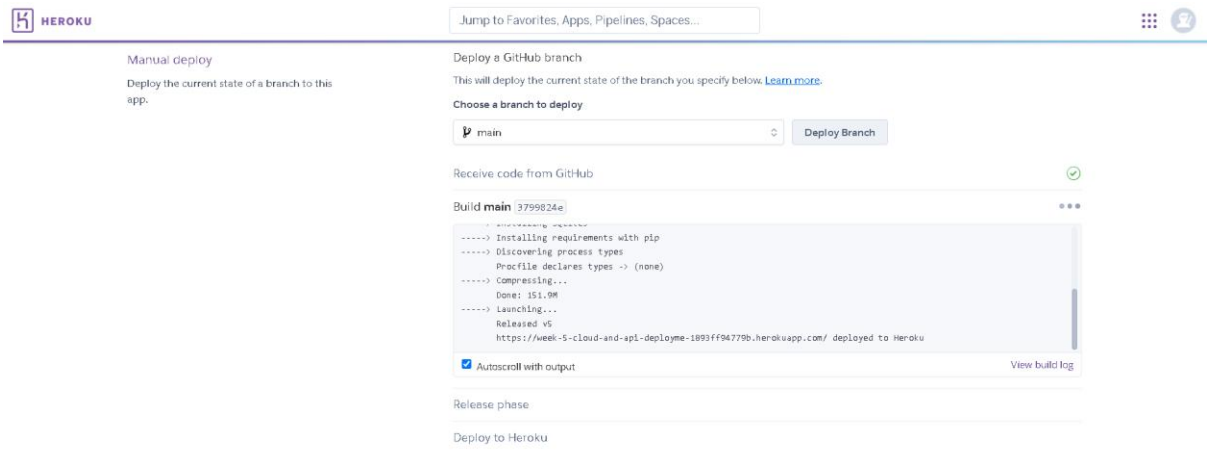
if __name__ == "__main__":
    app.run(debug=True)
```
```

Below in cmd app.py is ran and there we get the url (http://127.0.0.1:5000) to use in chrome:

```
D:\DataGlacier>python app.py
C:\Users\Mrinal Rajput\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklea
on 1.4.1.post1. This might lead to breaking code or invalid results. Use at your own r
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limita
  warnings.warn(
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
C:\Users\Mrinal Rajput\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklea
on 1.4.1.post1. This might lead to breaking code or invalid results. Use at your own r
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limita
  warnings.warn(
 * Debugger is active!
 * Debugger PIN: 940-799-733
127.0.0.1 - - [28/Feb/2024 21:26:44] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Feb/2024 21:26:44] "GET /static/css/style.css HTTP/1.1" 304 -
127.0.0.1 - - [28/Feb/2024 21:26:44] "GET /static/images/Original.svg HTTP/1.1" 304 -
C:\Users\Mrinal Rajput\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklea
  warnings.warn(
127.0.0.1 - - [28/Feb/2024 21:26:52] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [28/Feb/2024 21:26:52] "GET /static/css/style.css HTTP/1.1" 304 -
127.0.0.1 - - [28/Feb/2024 21:26:52] "GET /static/images/Original.svg HTTP/1.1" 304 -
```

Heroku:




HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
Manual deploy
Deploy the current state of a branch to this app.
Deploy a GitHub branch
This will deploy the current state of the branch you specify below. Learn more.
Choose a branch to deploy
main
Deploy Branch
Receive code from GitHub
Build main 3799824e
-----> Installing requirements with pip
-----> Discovering process types
Procfile declares types -> (none)
-----> Compressing...
Done: 151.9M
-----> Launching...
Released v5
https://week-5-cloud-and-api-deployme-1893ff94779b.herokuapp.com/ deployed to Heroku
Autoscroll with output
View build log
Release phase
Deploy to Heroku