



Data Glacier

Your Deep Learning Partner

Deployment on Flask

Batch code: LISP01

Submission date: 21 March, 2021

Submission to: Data Glacier

Agenda

Model

Deploy on flask

Web App Test

Model

Training and saving model

```
In [31]: train_x = train[['KM Travelled', 'month', 'day', 'Company_Pink Cab', 'City']]
train_y = train['Profit']
test_x = test[['KM Travelled', 'month', 'day', 'Company_Pink Cab', 'City']]
test_y = test['Profit']
```

```
In [32]: from sklearn.ensemble import RandomForestRegressor as RF
logit=RF(n_jobs=2, n_estimators=50, max_depth=15, min_samples_split=12)
logit.fit(train_x, train_y)
```

```
Out[32]: RandomForestRegressor(max_depth=15, min_samples_split=12, n_estimators=50,
                                n_jobs=2)
```

```
In [33]: logit.score(train_x, train_y)
```

```
Out[33]: 0.8491818608083906
```

```
In [34]: logit.score(test_x, test_y)
```

```
Out[34]: 0.8066990043933405
```

Trained model

```
In [39]: filename = 'finalized model.sav'
pickle.dump(logit, open(filename, 'wb'))
loaded_model = pickle.load(open(filename, 'rb'))
result = loaded_model.score(train_x, train_y)
print(result)
```

```
0.8491818608083906
```

Saved model

Deploy on Flask

Importing libraries, loading model and creating main view

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

app = Flask(__name__)

filename = '../w4 deliverables/finalized_model.sav'
model = pickle.load(open(filename, 'rb'))

@app.route('/')
def registrarse():
    result={'profit':'',
           'descr':'' }
    return render_template('registrarse.html', result=result)
```

Loaded model



Deploy on Flask

Creating model view

Input data

Prediction

```
@app.route("/upload", methods=['GET','POST'])
def uploader():
    if request.method == 'POST':
        # obtenemos el archivo del input "archivo"

        km=request.form['km']
        company=request.form['company']
        city=request.form['city']
        day=request.form['day']
        month=request.form['month']

        print(type(km),type(company),type(city),type(day),type(month))

        int_features = [float(km),int(month),int(day),int(company),int(city)]
        final_features = [np.array(int_features)]
        prediction =model.predict(final_features)

        output = round(prediction[0],2)
        print(output)

        result={
            'profit':output,
            'descr':'The profit should be: $' }
        return render_template('registrarse.html',result=result)

if __name__ == '__main__':
    app.run(host='0.0.0.0' , port = 8080, debug=True)
```

Deploy on Flask

Creating template

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

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

<body>

  <div id='pric'>
    <h1>Predict rides profit</h1>
    <form action="/upload" method="POST" class="form-inline tm-search-form" enctype="multipart/form-data">

      <label for="mapel">Company</label>
      <select name="company" id="cars">
        <option value="1">Pink</option>
        <option value="0">Yellow</option>
      </select>

      <label for="mapel">Day</label>
      <select name="day" id="cars">
        <option value="1">1</option>
        <option value="2">2</option>
        <option value="3">3</option>
        <option value="4">4</option>
        <option value="5">5</option>
        <option value="6">6</option>
        <option value="7">7</option>
        <option value="8">8</option>
        <option value="9">9</option>
        <option value="10">10</option>
```

```
      <label for="mapel">City</label>
      <select name="city" id="cars">
        <option value="0">ATLANTA GA</option>
        <option value="1">AUSTIN TX</option>
        <option value="2">BOSTON MA</option>
        <option value="3">CHICAGO IL</option>
        <option value="4">DALLAS TX</option>
        <option value="5">DENVER CO</option>
        <option value="6">LOS ANGELES CA</option>
        <option value="7">MIAMI FL</option>
        <option value="8">NASHVILLE TN</option>
        <option value="9">NEW YORK NY</option>
        <option value="10">ORANGE COUNTY</option>
        <option value="11">PHOENIX AZ</option>
        <option value="12">PITTSBURGH PA</option>
        <option value="13">SACRAMENTO CA</option>
        <option value="14">SAN DIEGO CA</option>
        <option value="15">SEATTLE WA</option>
        <option value="16">SILICON VALLEY</option>
        <option value="17">TUCSON AZ</option>
        <option value="18">WASHINGTON DC</option>
      </select>

      <label for="fdni">KM</label>
      <input type="number" id="fdni" name="km" placeholder="Put KM" minlength="8" maxlength="8" required>

      <input type="submit" value="Send">
    </form>
    <strong>{{result.descr}}<strong>
  </div>
</body>
</html>
```

Deploy on Flask

Creating styles for template

```
input[type=text], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}

input[type=number], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}

input[type=submit] {
  width: 100%;
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}
```

```
input[type=submit]:hover {
  background-color: #45a049;
}

#pric {
  width: 30%;
  border-radius: 5px;
  background-color: #f2f2f2;
  padding: 10px;
  margin: 0 auto;
}

body {
  background-image: url("123.jpg");
}

h1 {
  text-align: center;
}
```

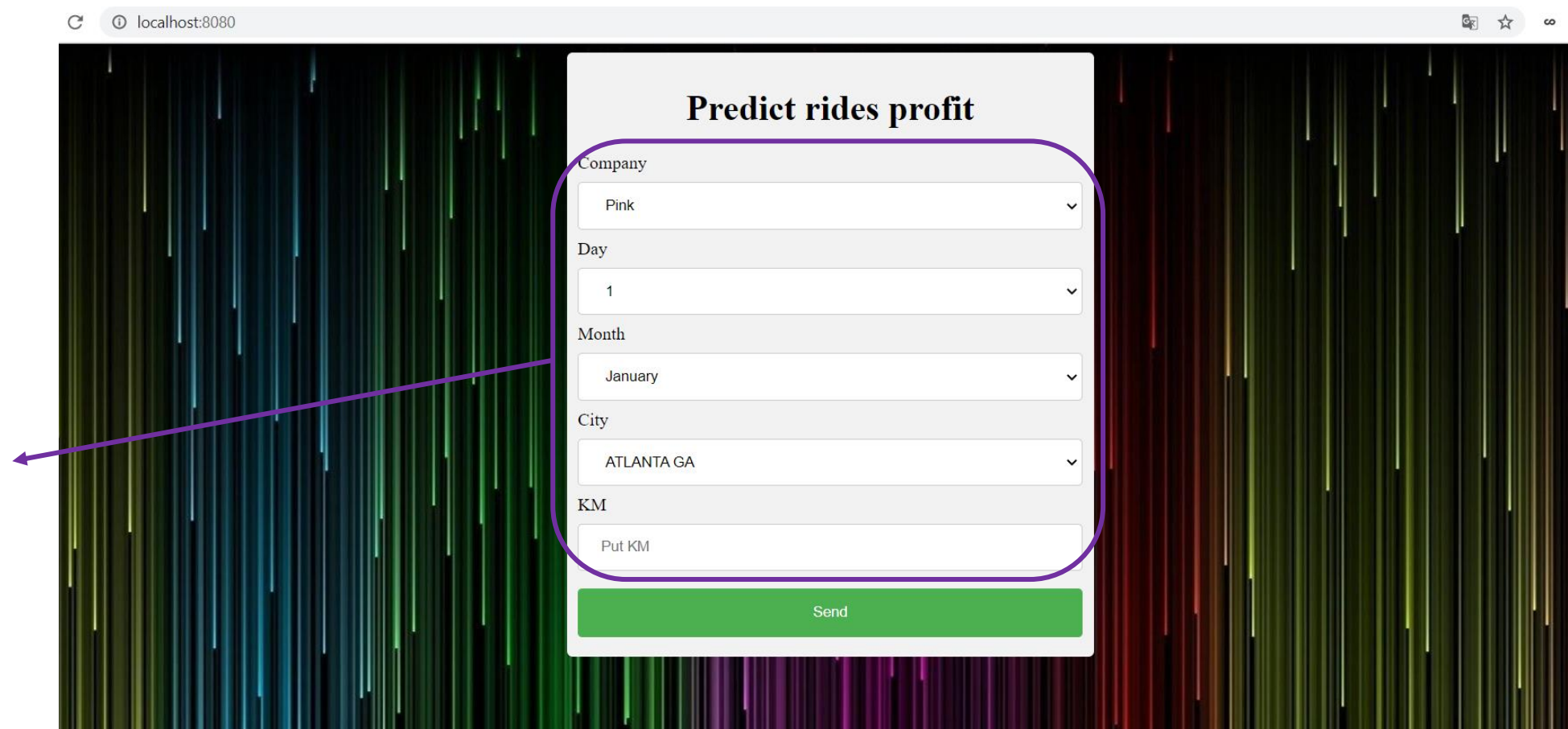
Deploy on Flask

Running App

```
(flask) D:\repos\DG\W4\flask>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: 266-435-914
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```


Web App Test

Opening URL: localhost:8080

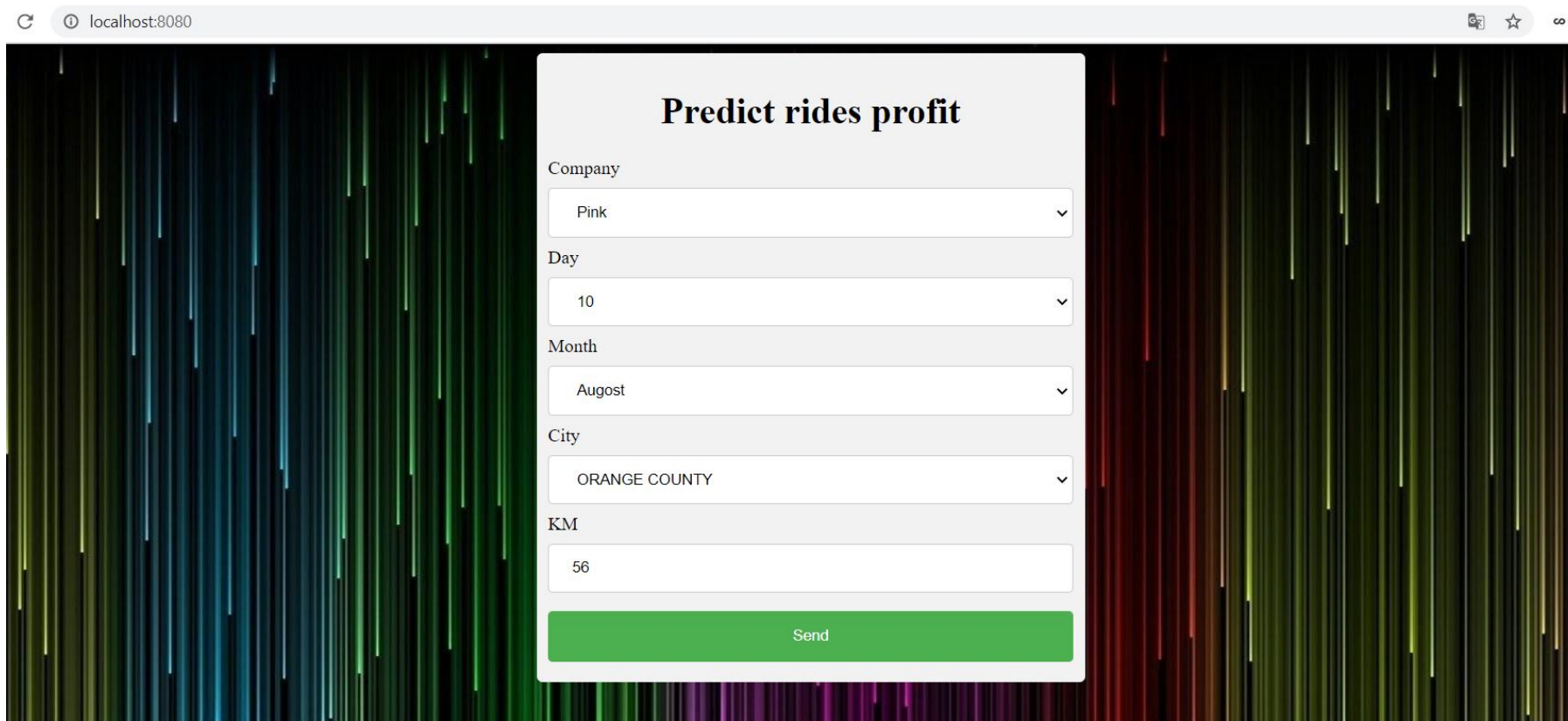


The screenshot shows a web browser window with the address bar displaying 'localhost:8080'. The main content area features a form titled 'Predict rides profit'. The form contains five input fields, each with a label and a dropdown arrow: 'Company' (selected: Pink), 'Day' (selected: 1), 'Month' (selected: January), 'City' (selected: ATLANTA GA), and 'KM' (selected: Put KM). A green 'Send' button is located at the bottom of the form. A purple arrow points from the text 'Input data' to the form, indicating the input data fields.

Input data

Web App Test

Putting data



A screenshot of a web browser window displaying a web application. The browser's address bar shows 'localhost:8080'. The web page has a dark background with a colorful, abstract pattern of vertical lines. In the center, there is a light gray rectangular box containing the title 'Predict rides profit' in bold black text. Below the title, there are five input fields, each with a label above it: 'Company' (dropdown menu showing 'Pink'), 'Day' (dropdown menu showing '10'), 'Month' (dropdown menu showing 'August'), 'City' (dropdown menu showing 'ORANGE COUNTY'), and 'KM' (text input field showing '56'). At the bottom of the box is a green button with the text 'Send' in white.

Web App Test

Prediction of profit

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/upload'. The page content is a form titled 'Predict rides profit' set against a background of colorful vertical streaks. The form contains the following elements:

- Company:** A dropdown menu with 'Pink' selected.
- Day:** A dropdown menu with '1' selected.
- Month:** A dropdown menu with 'January' selected.
- City:** A dropdown menu with 'ATLANTA GA' selected.
- KM:** A text input field containing 'Put KM'.
- Send:** A green button labeled 'Send'.
- Prediction:** A message box at the bottom stating 'The profit should be: \$175.67'.

A purple arrow points from the word 'Prediction' on the left to the prediction message box.

Thank You