

Aryaman Srivastava  
LISUM34  
7/5/2024  
Submitted to Canvas

```
1 from flask import Flask, request, render_template
2 import numpy as np
3 import pickle
4
5 app = Flask(__name__)
6 model = pickle.load(open("model.pkl", "rb"))
7
8
9 @app.route("/") # home root
10 def home():
11     return render_template("index.html")
12
13
14 @app.route("/predict", methods=["POST"]) # prediction post method for submission
15 def predict():
16     # predict price charged based on user inputted distance travelled and cost of trip
17     args = [int(x) for x in request.form.values()]
18     prediction = model.predict([np.array(args)])
19     output = round(prediction[0], 2)
20     return render_template(
21         "index.html", prediction="You can expect to charge ${}".format(output)
22     )
23
24
25 if __name__ == "__main__":
26     app.run(debug=True)
```

```
5 Procfile
1 web: gunicorn app:app
```

= requirements.txt

```
1  Flask==3.0.3
2  gunicorn==22.0.0
3  itsdangerous==2.2.0
4  Jinja2==3.1.4
5  MarkupSafe==2.1.5
6  Werkzeug==3.0.3
7  numpy==1.26.4
8  scipy==1.14.0
9  scikit-learn==1.5.0
10 matplotlib==3.9.0
11 pandas==2.2.2
```

```
> v
from sklearn import linear_model
import pandas as pd
import numpy as np
import pickle
# build model for predicting price charged by Pink Cab in Boston
df = pd.read_csv("finalCabData.csv")
filteredDf = df[(df["City"] == "BOSTON MA")]
filteredDf = filteredDf[filteredDf["Company"] == "Pink Cab"]
x = np.array(filteredDf[["KM Travelled", "Cost of Trip"]])
y = np.array(filteredDf["Price Charged"])
model = linear_model.LinearRegression()
model.fit(x, y)
#save model to pickle file
pickle.dump(model, open("model.pkl", "wb"))
```

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Predictor</title>
  </head>
  <body>
    <div>
      <form action="/predict" method="post">
        <input
          type="text"
          name="Distance"
          placeholder="Distance Travelled in KM"
        />
        <input type="text" name="Cost" placeholder="Cost of Trip" />
        <button type="submit">Submit</button>
      </form>
      {{prediction}}
    </div>
    <link
      rel="stylesheet"
      href="{{url_for('static', filename='css/styles.css')}}"
    />
  </body>
</html>

```

```

html {
  background-color: red;
  color: white;
}

div {
  width: 50%;
  margin: auto;
  align-items: center;
}

form {
  width: fit-content;
  margin: auto;
}

```


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 dfde800d✓
- Release phase✓
- Deploy to Heroku✓

Your app was successfully deployed.

 View

