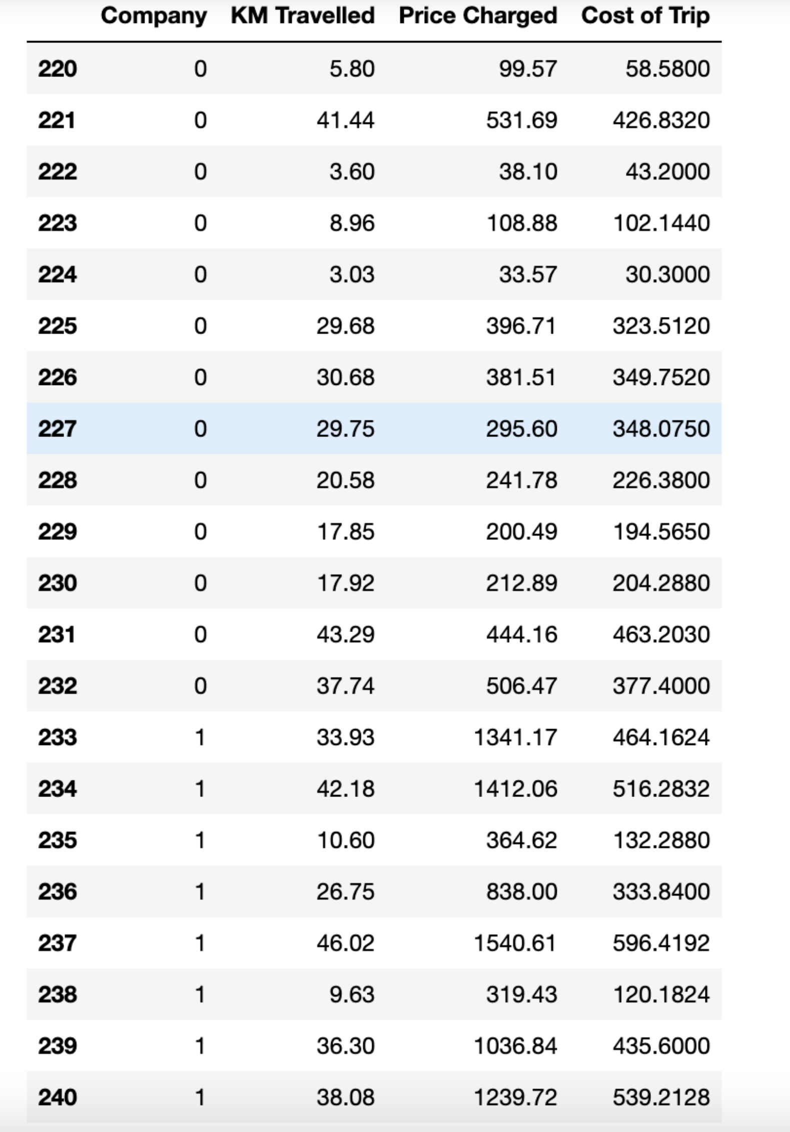
Name: Kefan Yu

Submission Date: Mar 28, 2023

Submitted to: Data Glacier

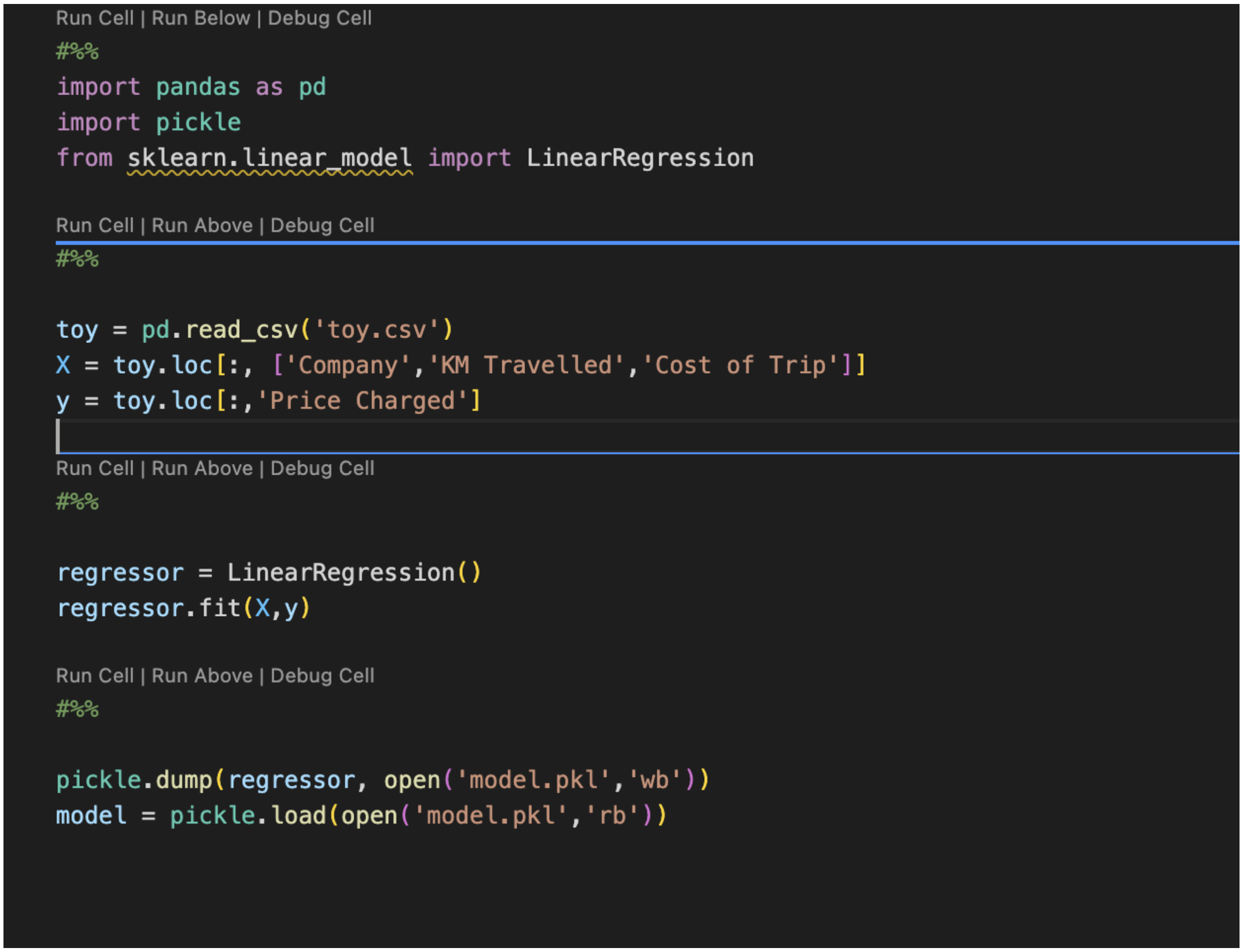
Step 1: Find the Dataset



	Company	KM Travelled	Price Charged	Cost of Trip
220	0	5.80	99.57	58.5800
221	0	41.44	531.69	426.8320
222	0	3.60	38.10	43.2000
223	0	8.96	108.88	102.1440
224	0	3.03	33.57	30.3000
225	0	29.68	396.71	323.5120
226	0	30.68	381.51	349.7520
227	0	29.75	295.60	348.0750
228	0	20.58	241.78	226.3800
229	0	17.85	200.49	194.5650
230	0	17.92	212.89	204.2880
231	0	43.29	444.16	463.2030
232	0	37.74	506.47	377.4000
233	1	33.93	1341.17	464.1624
234	1	42.18	1412.06	516.2832
235	1	10.60	364.62	132.2880
236	1	26.75	838.00	333.8400
237	1	46.02	1540.61	596.4192
238	1	9.63	319.43	120.1824
239	1	36.30	1036.84	435.6000
240	1	38.08	1239.72	539.2128

Step 2: Write model.py

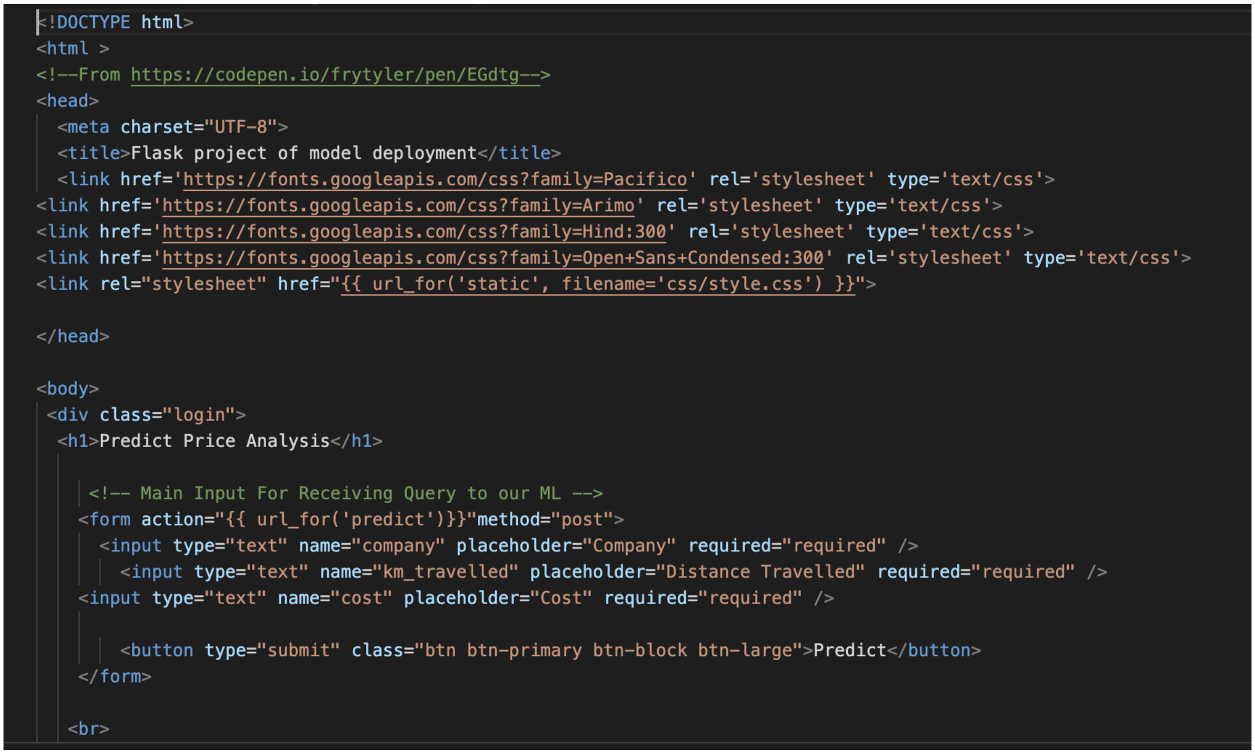
In this step, I subsetted the dataset into X and y (response) in order to make predictions of the columns including company, distance travelled, and cost. After it, I will save the fitted model into model.pkl using pickle.


Run Cell | Run Below | Debug Cell
Run Cell | Run Above | Debug Cell
Run Cell | Run Above | Debug Cell
Run Cell | Run Above | Debug Cell

```
#%%
import pandas as pd
import pickle
from sklearn.linear_model import LinearRegression

#%%

toy = pd.read_csv('toy.csv')
X = toy.loc[:, ['Company','KM Travelled','Cost of Trip']]
y = toy.loc[:,'Price Charged']

#%%

regressor = LinearRegression()
regressor.fit(X,y)

#%%

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

Step 3: Write html webpage

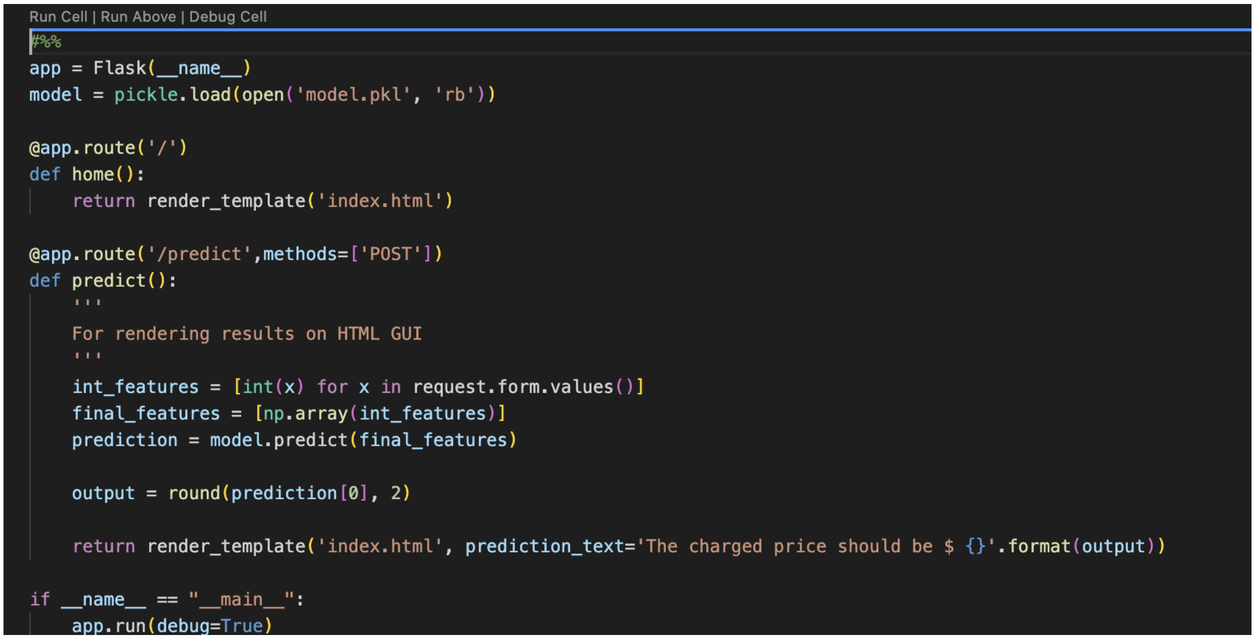
Next step is to implement the html page with css style sheet and input bar.



```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>Flask project of model deployment</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 Price Analysis</h1>

      <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
      <input type="text" name="company" placeholder="Company" required="required" />
        <input type="text" name="km_travelled" placeholder="Distance Travelled" required="required" />
    <input type="text" name="cost" placeholder="Cost" required="required" />

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

   <br>
```

Step 4: Write app.py

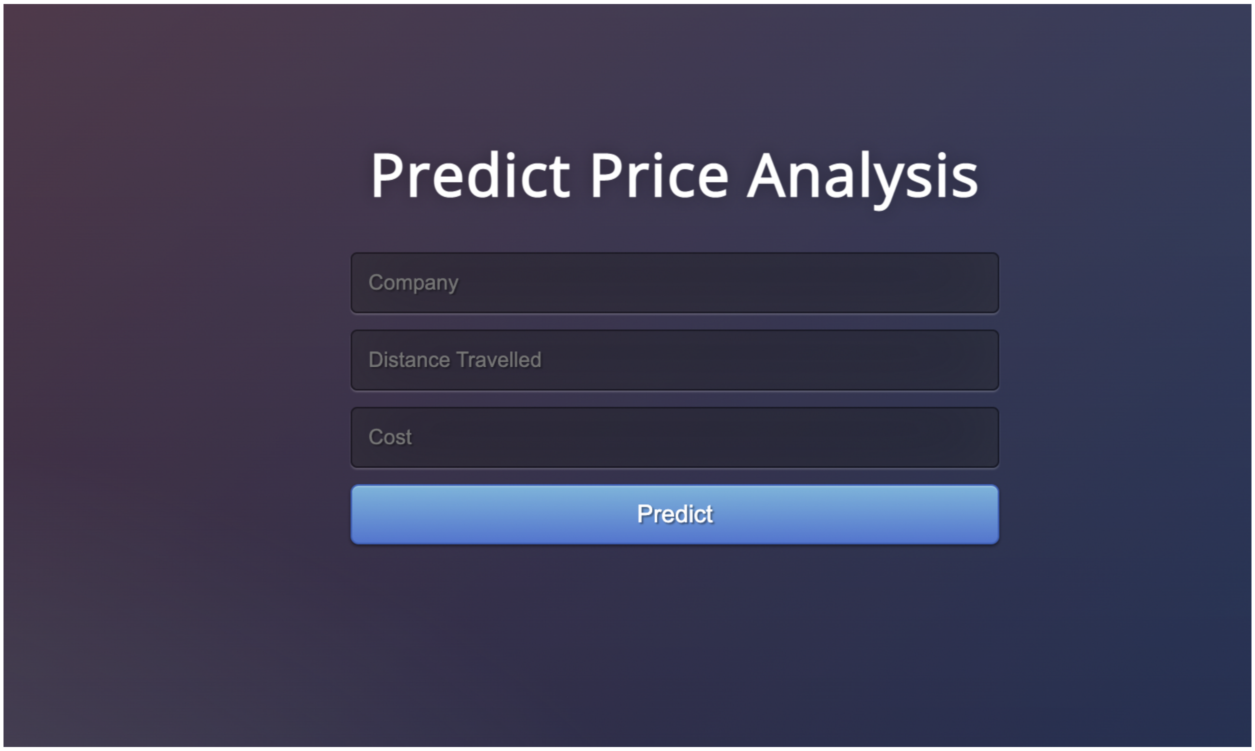
Next is to write app.py with model.pkl and render the html with Flask.



```
#%%
app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@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 = round(prediction[0], 2)

    return render_template('index.html', prediction_text='The charged price should be $ {}'.format(output))

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

Step 5: Run the file in the terminal

Next we run the file in the terminal and get the webpage


Predict Price Analysis
Company
Distance Travelled
Cost
Predict