Week 4: Deployment on Flask

Name: Kseniya Kudzelich

Batch Code: LISUM34

Submission Date: 27 June 2024

Submitted to: Data Glacier

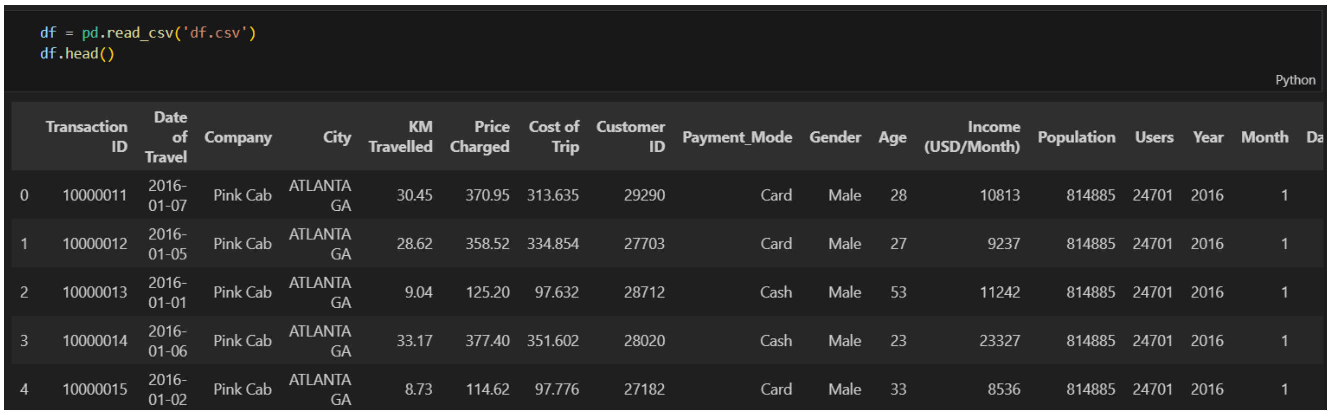
Introduction:

I would like to remind you that during the second and third weeks of the "G2M insight for Cab Investment firm" project, I conducted an in-depth Exploratory Data Analysis (EDA) of the data, tested hypotheses, developed recommendations on choosing a company to invest in, and then presented the results.

This week, based on this data, I trained a Machine Learning model (XGBoost) to predict Profit per Trip, taking into account various features (like "City", "Payment Mode", "Gender", "KM Travelled", "Date", etc.), and then locally deployed this model using the Flask framework.

Modeling

1.1. Load the data



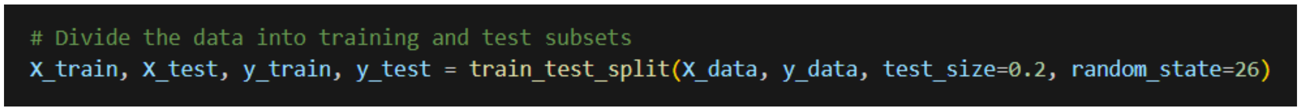
```
df = pd.read_csv('df.csv')
df.head()
```

Python

	Transaction ID	Date of Travel	Company	City	KM Travelled	Price Charged	Cost of Trip	Customer ID	Payment_Mode	Gender	Age	Income (USD/Month)	Population	Users	Year	Month	Da
0	10000011	2016-01-07	Pink Cab	ATLANTA GA	30.45	370.95	313.635	29290	Card	Male	28	10813	814885	24701	2016	1	
1	10000012	2016-01-05	Pink Cab	ATLANTA GA	28.62	358.52	334.854	27703	Card	Male	27	9237	814885	24701	2016	1	
2	10000013	2016-01-01	Pink Cab	ATLANTA GA	9.04	125.20	97.632	28712	Cash	Male	53	11242	814885	24701	2016	1	
3	10000014	2016-01-06	Pink Cab	ATLANTA GA	33.17	377.40	351.602	28020	Cash	Male	23	23327	814885	24701	2016	1	
4	10000015	2016-01-02	Pink Cab	ATLANTA GA	8.73	114.62	97.776	27182	Card	Male	33	8536	814885	24701	2016	1	

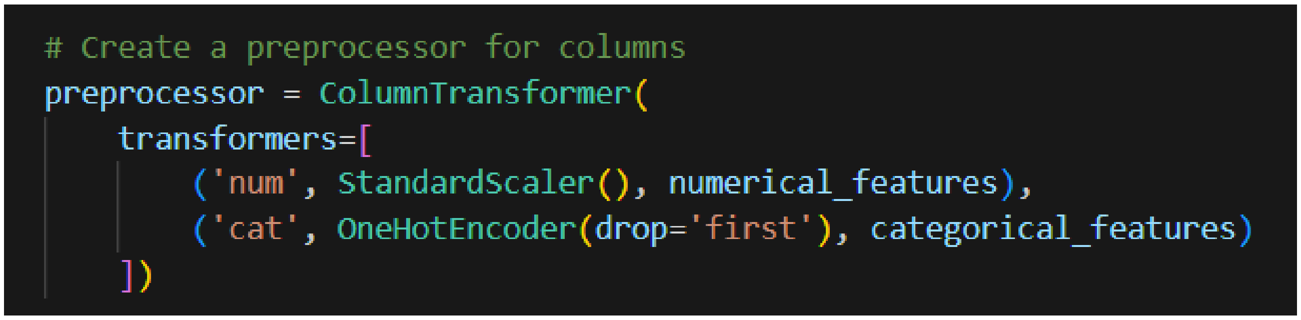
1.2. Data preprocessing

- Train and test split

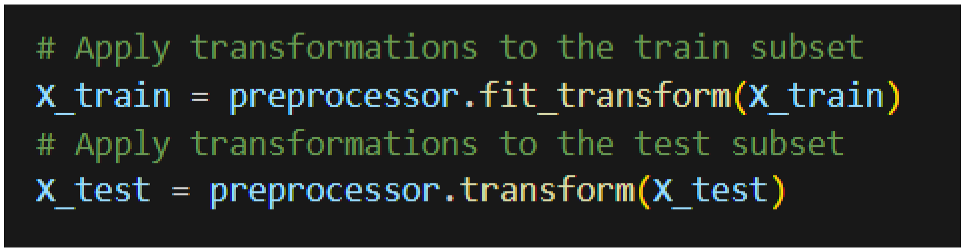


```
# Divide the data into training and test subsets
X_train, X_test, y_train, y_test = train_test_split(X_data, y_data, test_size=0.2, random_state=26)
```

- Scale and encode



```
# Create a preprocessor for columns
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_features),
        ('cat', OneHotEncoder(drop='first'), categorical_features)
    ])
```

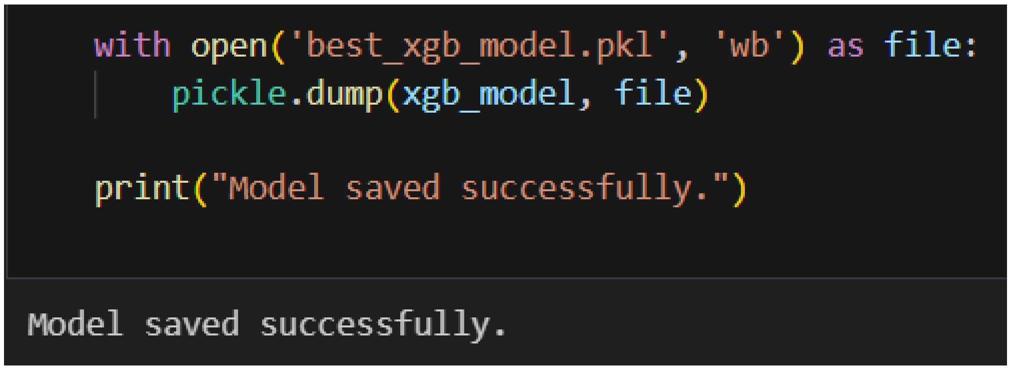


```
# Apply transformations to the train subset
X_train = preprocessor.fit_transform(X_train)
# Apply transformations to the test subset
X_test = preprocessor.transform(X_test)
```

1.3. Train the models

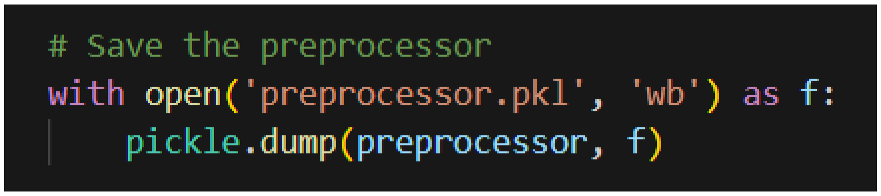
- Linear Regression, Ridge, Lasso, Random Forest, XGBoost, XGBoost with GridSearchCV

1.4. Choose the best model and save it together with the preprocessor using *pickle*



```
with open('best_xgb_model.pkl', 'wb') as file:
    pickle.dump(xgb_model, file)

print("Model saved successfully.")
```

```
Model saved successfully.
```

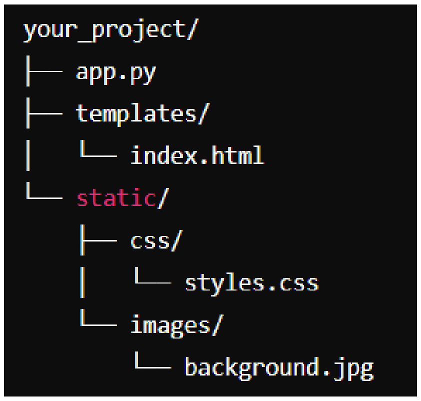


```
# Save the preprocessor
with open('preprocessor.pkl', 'wb') as f:
    pickle.dump(preprocessor, f)
```

Local Deployment on Flask

2.1. Create a directory structure for the project:

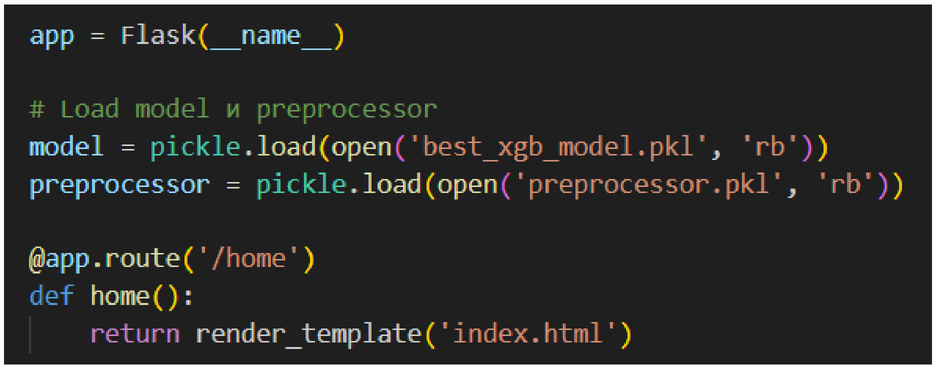
- Organize the project files into a directory structure that includes a Flask script, HTML template, static resources (CSS, images), a model, and a preprocessor.



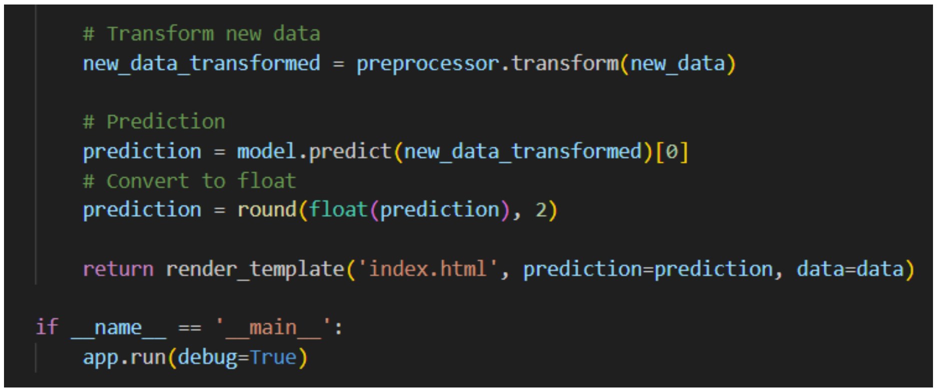
```
your_project/
├── app.py
├── templates/
│   └── index.html
└── static/
    ├── css/
    │   └── styles.css
    └── images/
        └── background.jpg
```

2.2. Create a Flask app:

- Write a Flask script ('*app.py*') that loads the model and the preprocessor, extracts data from the from, processes the input data from the web form, makes predictions and displays the results on the web page.



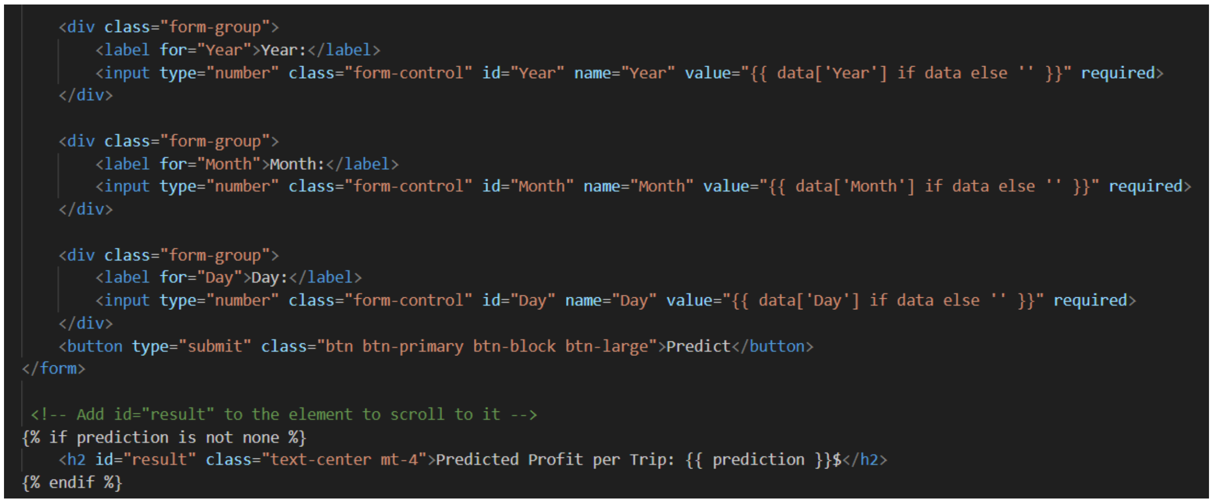
```python
app = Flask(__name__)

# Load model и preprocessor
model = pickle.load(open('best_xgb_model.pkl', 'rb'))
preprocessor = pickle.load(open('preprocessor.pkl', 'rb'))

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



```python
        # Transform new data
        new_data_transformed = preprocessor.transform(new_data)

        # Prediction
        prediction = model.predict(new_data_transformed)[0]
        # Convert to float
        prediction = round(float(prediction), 2)

        return render_template('index.html', prediction=prediction, data=data)

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

2.3. Create an HTML template:

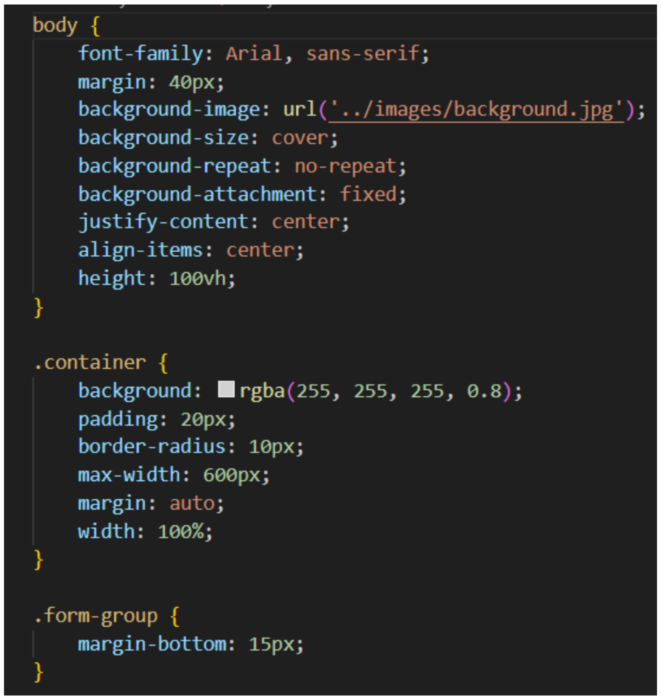
- Create an template ('*index.html*') for a web form that allows users to enter data for prediction. You need to select an option from the list for the "City", "Payment Method", "Gender" and "Company". All other elements of the form should be entered manually. After submitting the form, the template displays the predicted result.



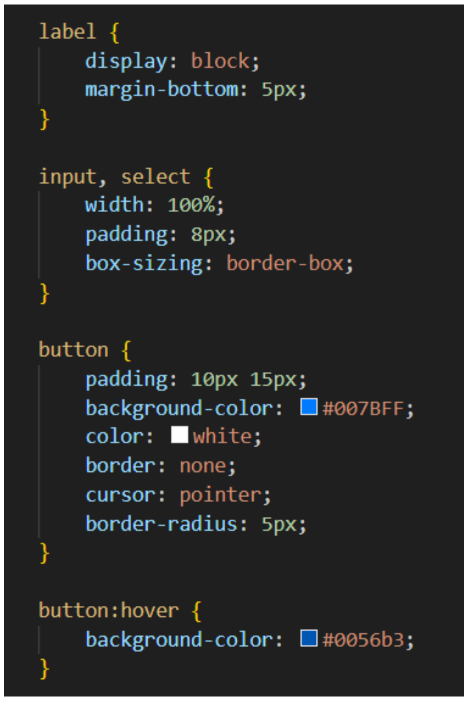
```html
    <div class="form-group">
        <label for="Year">Year:</label>
        <input type="number" class="form-control" id="Year" name="Year" value="{{ data['Year'] if data else '' }}" required>
    </div>

    <div class="form-group">
        <label for="Month">Month:</label>
        <input type="number" class="form-control" id="Month" name="Month" value="{{ data['Month'] if data else '' }}" required>
    </div>

    <div class="form-group">
        <label for="Day">Day:</label>
        <input type="number" class="form-control" id="Day" name="Day" value="{{ data['Day'] if data else '' }}" required>
    </div>
    <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
</form>

 <!-- Add id="result" to the element to scroll to it -->
{% if prediction is not none %}
    <h2 id="result" class="text-center mt-4">Predicted Profit per Trip: {{ prediction }}$</h2>
{% endif %}
```

2.4. Create CSS styles:

- Add styles to enhance the appearance of the web page, including the design of the form and background, to create a more user-friendly and visually appealing interface.



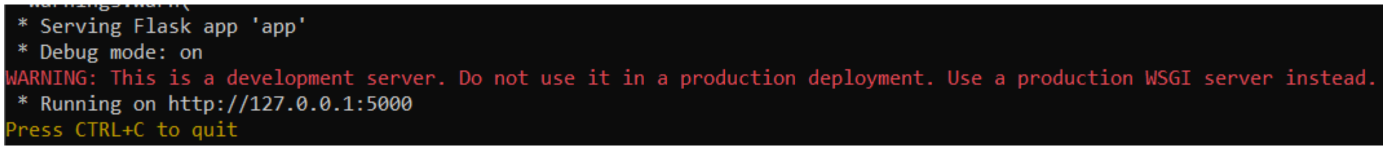
```css
body {
    font-family: Arial, sans-serif;
    margin: 40px;
    background-image: url('../images/background.jpg');
    background-size: cover;
    background-repeat: no-repeat;
    background-attachment: fixed;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

.container {
    background: rgba(255, 255, 255, 0.8);
    padding: 20px;
    border-radius: 10px;
    max-width: 600px;
    margin: auto;
    width: 100%;
}

.form-group {
    margin-bottom: 15px;
}
```



```css
label {
    display: block;
    margin-bottom: 5px;
}

input, select {
    width: 100%;
    padding: 8px;
    box-sizing: border-box;
}

button {
    padding: 10px 15px;
    background-color: #007BFF;
    color: white;
    border: none;
    cursor: pointer;
    border-radius: 5px;
}

button:hover {
    background-color: #0056b3;
}
```

2.5. Run app:

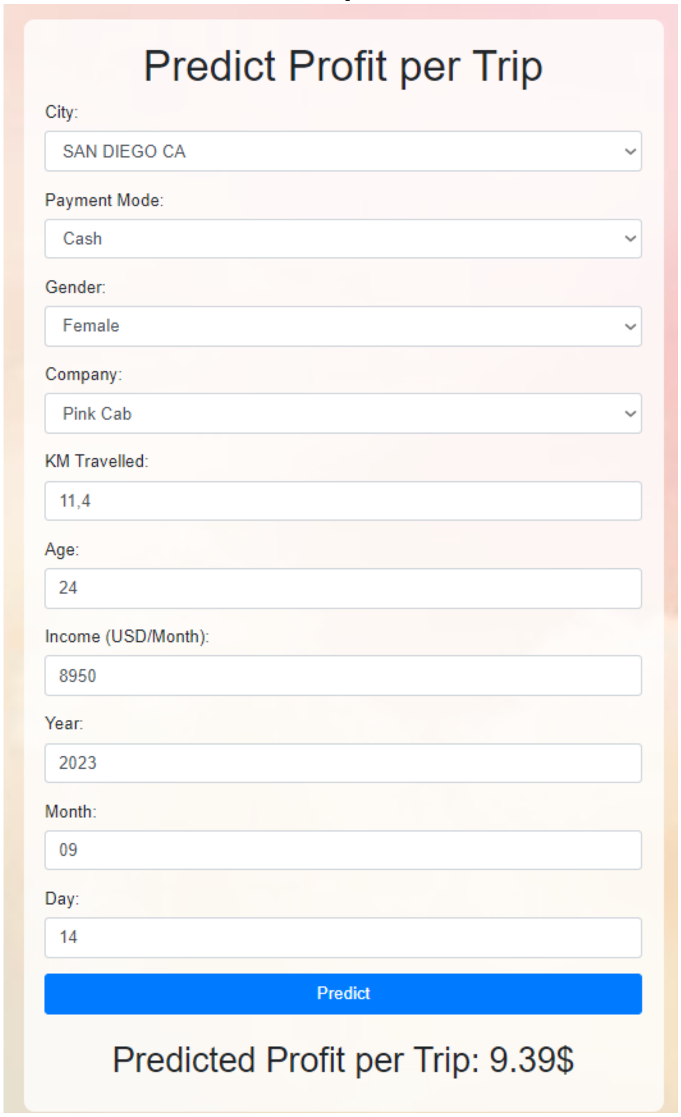
- Launch the Flask app and open it in a web browser on a local server http://127.0.0.1:5000 to test the data entry and predictions, ensuring that all elements are working correctly.



```
 * 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
```

Below you can see 6 examples of using locally deployed Flask app:

Example 1:



Predict Profit per Trip

City: SAN DIEGO CA

Payment Mode: Cash

Gender: Female

Company: Pink Cab

KM Travelled: 11,4

Age: 24

Income (USD/Month): 8950

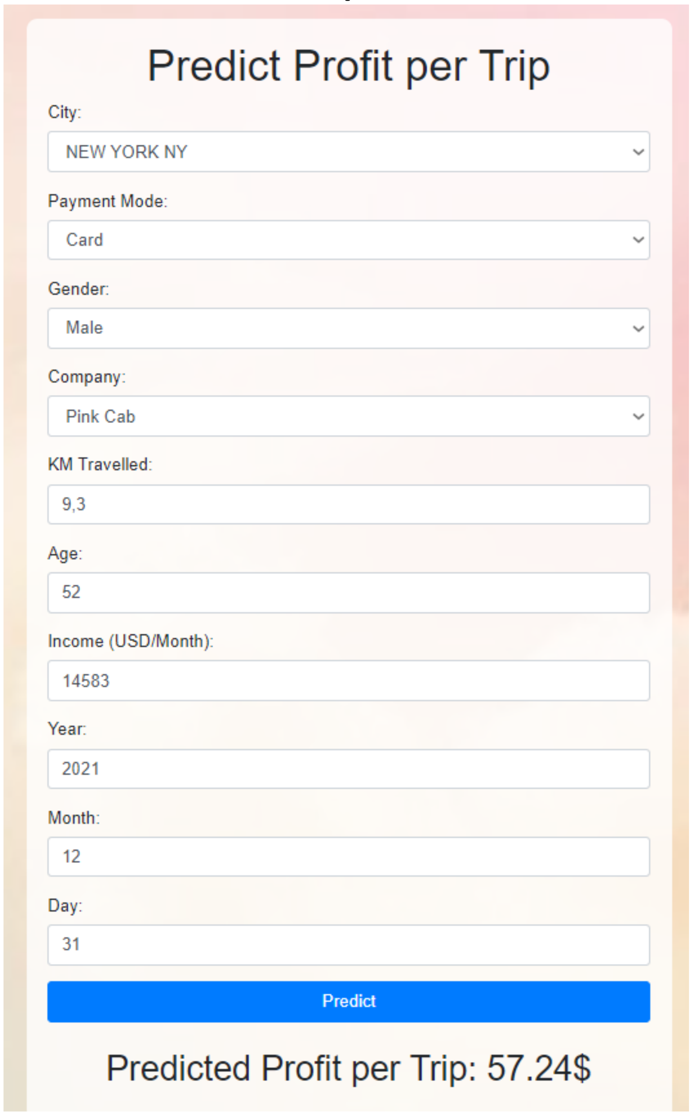
Year: 2023

Month: 09

Day: 14

Predicted Profit per Trip: 9.39$

Example 2:



Predict Profit per Trip

City: NEW YORK NY

Payment Mode: Card

Gender: Male

Company: Pink Cab

KM Travelled: 9,3

Age: 52

Income (USD/Month): 14583

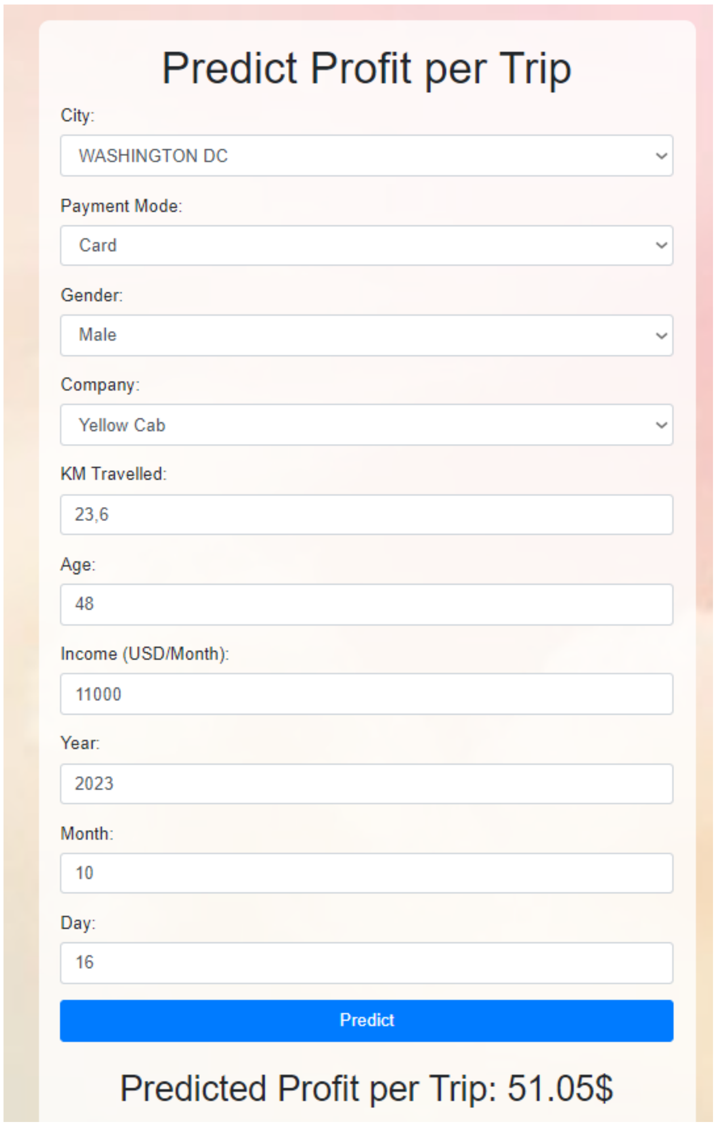
Year: 2021

Month: 12

Day: 31

Predicted Profit per Trip: 57.24$

Example 3:

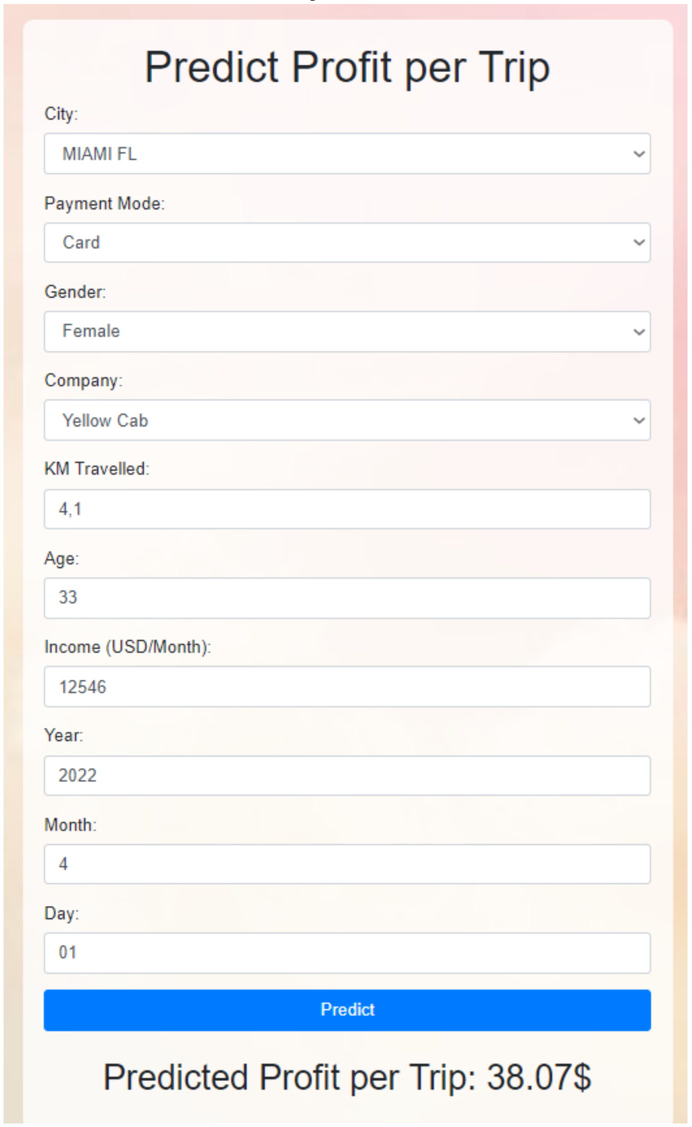


Predict Profit per Trip

City:	WASHINGTON DC
Payment Mode:	Card
Gender:	Male
Company:	Yellow Cab
KM Travelled:	23,6
Age:	48
Income (USD/Month):	11000
Year:	2023
Month:	10
Day:	16

Predicted Profit per Trip: 51.05$

Example 4:



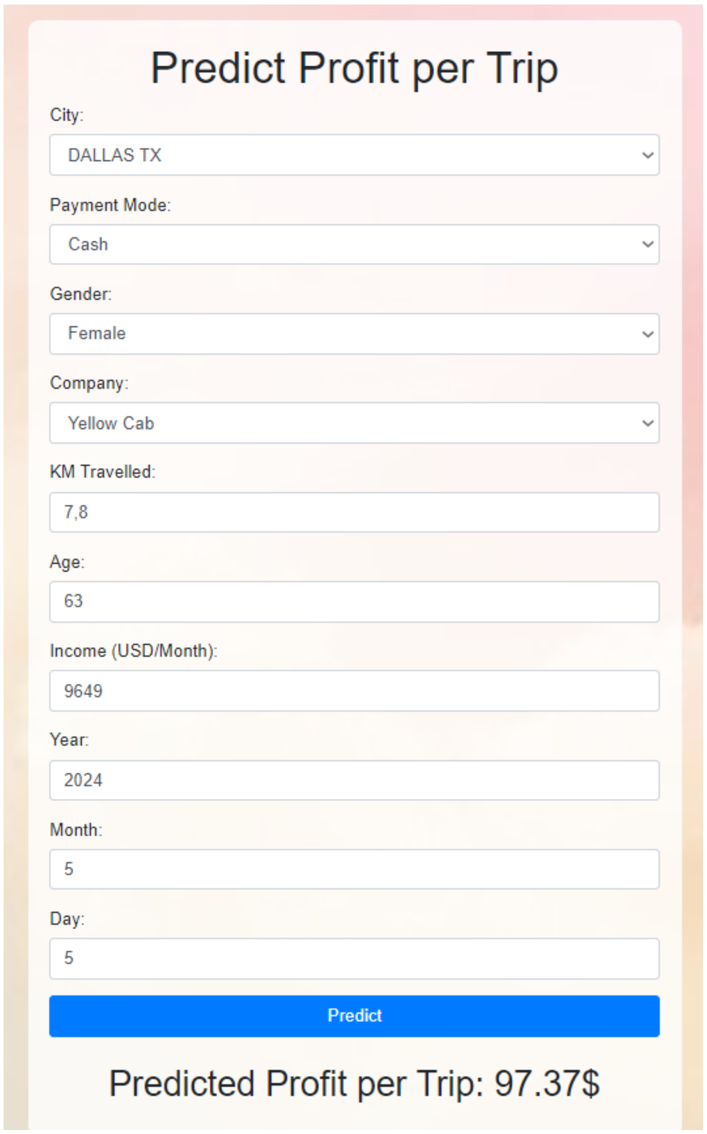
Predict Profit per Trip

City:	MIAMI FL
Payment Mode:	Card
Gender:	Female
Company:	Yellow Cab
KM Travelled:	4,1
Age:	33
Income (USD/Month):	12546
Year:	2022
Month:	4
Day:	01

Predicted Profit per Trip: 38.07$

Example 5:

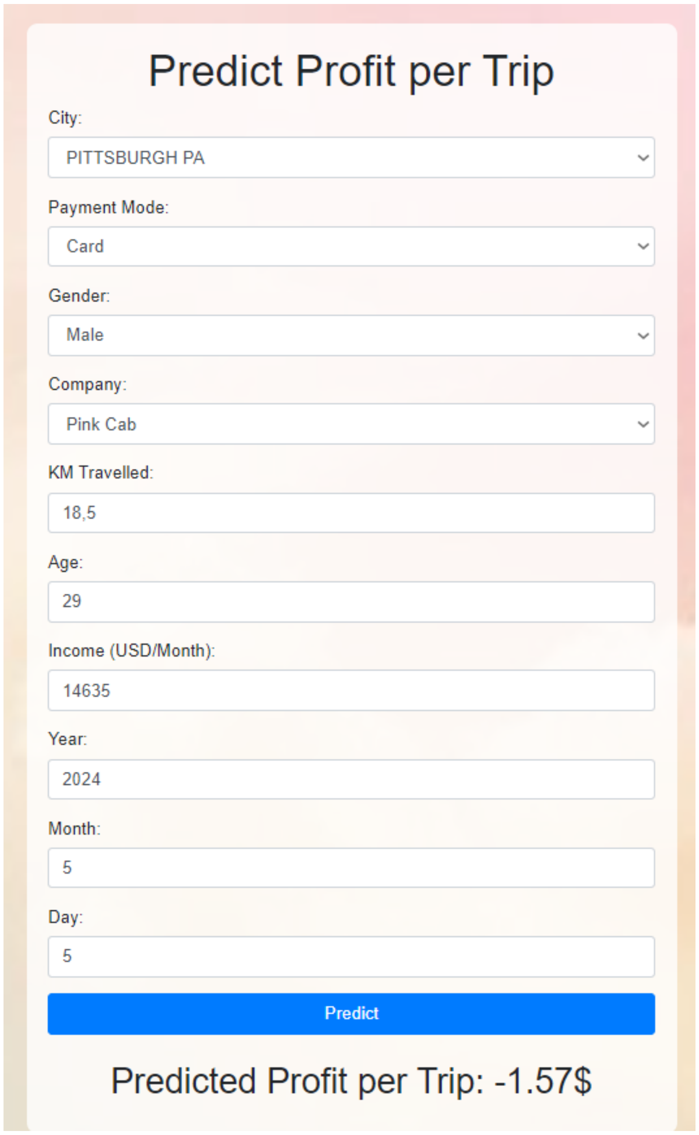
Example 6:



Predict Profit per Trip

City:	DALLAS TX
Payment Mode:	Cash
Gender:	Female
Company:	Yellow Cab
KM Travelled:	7,8
Age:	63
Income (USD/Month):	9649
Year:	2024
Month:	5
Day:	5

Predicted Profit per Trip: 97.37$

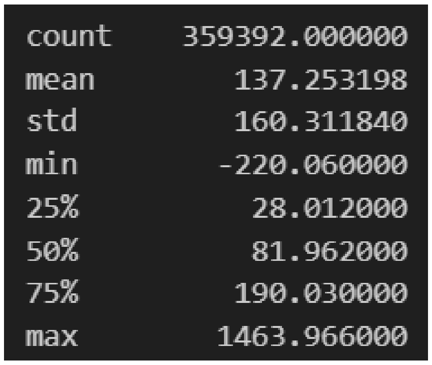


Predict Profit per Trip

City:	PITTSBURGH PA
Payment Mode:	Card
Gender:	Male
Company:	Pink Cab
KM Travelled:	18,5
Age:	29
Income (USD/Month):	14635
Year:	2024
Month:	5
Day:	5

Predicted Profit per Trip: -1.57$

The results are acceptable, as the actual Profit column in the training data ranges from -220.06 to +1463.96.



```
count    359392.000000
mean        137.253198
std         160.311840
min        -220.060000
25%          28.012000
50%          81.962000
75%         190.030000
max        1463.966000
```