

# Data Intake Report Flask Deployment

Project Name: Flask Deployment

Report Date: 05/22/2024

Internship Batch: LISUM33

Version: 1.0

Data Intake by: Marina Tsvetkova

Data Intake Reviewer: Data Glacier

Data Storage Location:

[https://github.com/Marinatsv07/Data\\_Glacier\\_Internship/tree/main/Week\\_4](https://github.com/Marinatsv07/Data_Glacier_Internship/tree/main/Week_4)

## Model Setup

### Loading data:

train data: df

```
=pd.read_parquet(https://d37ci6vzurychx.cloudfront.net/trip-data/yellow\_tripdata\_2024-01.parquet)
```

```
def load_model(model_path: str) -> LinearRegression:
```

```
    """
```

```
    Load the model from the given file.
```

```
    Parameters:
```

```
    model_path (str): Path to the saved model file (Pickle format).
```

```
    Returns:
```

```
    LinearRegression: Loaded model.
```

```
    """
```

```
    print("Loading model...")
```

```
    with open(model_path, 'rb') as file:
```

```
    model = pickle.load(file)
    print("Model loaded.")
    return model
```

```
model_path = 'linear_regression_model.pkl'
model = load_model(model_path)
```

```
y = df_filtered['duration_min']
print("Making predictions...")
y_pred = model.predict(X)
```

### **Training model:**

```
vectorizer = DictVectorizer(sparse=True)
feature_matrix = vectorizer.fit_transform(data_dicts)
# Target variable
y = df_filtered['duration']
```

```
# Split data into training and test sets
```

```
X_train, X_test, y_train, y_test = train_test_split(feature_matrix, y, test_size=0.2,
random_state=42)
```

```
# Create and train a linear regression model
```

```
model = LinearRegression()
model.fit(X_train, y_train)
```

### **Predicting the data:**

```
PULocationID = request.form['PULocationID']
DOLocationID = request.form['DOLocationID']
print(f"PULocationID %s", PULocationID)
print(f"DOLocationID %s", DOLocationID)
```

```
# Create a dataframe for the input data
```

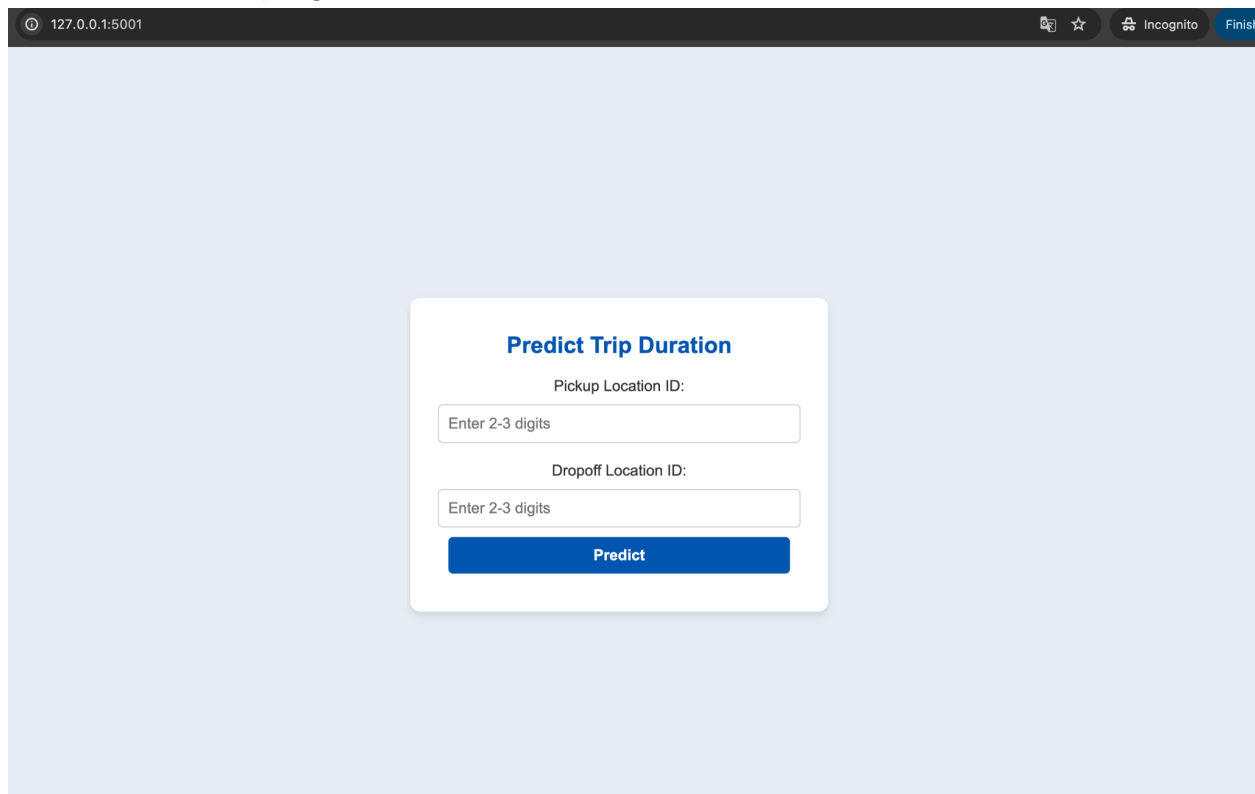
```
input_data = pd.DataFrame({'PULocationID': PULocationID,
'DOLocationID': DOLocationID})

# Convert location IDs to strings
input_data['PULocationID'] = input_data['PULocationID'].astype(str)
input_data['DOLocationID'] = input_data['DOLocationID'].astype(str)

# Transform the input data using the loaded vectorizer
data_dicts = input_data.to_dict(orient='records')
X_input = vectorizer.transform(data_dicts)

# Make predictions using the loaded model
prediction = model.predict(X_input)
# Return the prediction result
prediction_text = f'Predicted Trip Duration: {prediction[0]:.2f} minutes'
```

## HTML & CSS Deployment



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5001'. The browser is in Incognito mode. The main content area has a light blue background. In the center, there is a white card with rounded corners and a subtle shadow. The card is titled 'Predict Trip Duration' in bold blue text. Below the title, there are two input fields. The first is labeled 'Pickup Location ID:' and has a placeholder text 'Enter 2-3 digits'. The second is labeled 'Dropoff Location ID:' and also has a placeholder text 'Enter 2-3 digits'. At the bottom of the card, there is a blue button with the text 'Predict' in white.

127.0.0.1:5001/predict

### Predict Trip Duration

Pickup Location ID:

456

Dropoff Location ID:

125

Predict

**Predicted Trip Duration: 18.04 minutes**

## Deployment

## Predict Trip Duration

Pickup Location ID:

456

Dropoff Location ID:

125

Predict

**Predicted Trip Duration: 18.04  
minutes**

## Predict Trip Duration

Pickup Location ID:

Enter 2-3 digits

Dropoff Location ID:

Enter 2-3 digits

Predict