

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
!pip install ipyml
# Instead of %matplotlib widget, try using the following:
# %matplotlib inline
# Or this if you're in a Jupyter notebook environment:
# import matplotlib_inline
# matplotlib_inline.backend_inline.set_matplotlib_formats('svg')
import seaborn as sns
sns.set_style('whitegrid')
import folium
import pickle
from prettytable import PrettyTable

```

```
#####
```

```

def save(var):
    outfile = open(var + ".pickle", "wb")
    pickle.dump(eval(var), outfile)
    outfile.close()

```

```

def load(var):
    infile = open(var + ".pickle", "rb")
    globals()[var] = pickle.load(infile)
    infile.close()

```



Collecting ipyml

```

  Downloading ipyml-0.9.7-py3-none-any.whl.metadata (8.7 kB)
  Requirement already satisfied: ipython<10 in /usr/local/lib/python3.11/dist-packages
  Requirement already satisfied: ipywidgets<9,>=7.6.0 in /usr/local/lib/python3.11/dist
  Requirement already satisfied: matplotlib<4,>=3.5.0 in /usr/local/lib/python3.11/dist
  Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from
  Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-packages (fro
  Requirement already satisfied: traitlets<6 in /usr/local/lib/python3.11/dist-packages
  Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.11/dist-pac
  Collecting jedi>=0.16 (from ipython<10->ipyml)

```

```

  Downloading jedi-0.19.2-py2.py3-none-any.whl.metadata (22 kB)
  Requirement already satisfied: decorator in /usr/local/lib/python3.11/dist-packages (
  Requirement already satisfied: pickleshare in /usr/local/lib/python3.11/dist-packages
  Requirement already satisfied: prompt-toolkit!=3.0.0,!3.0.1,<3.1.0,>=2.0.0 in /usr/l
  Requirement already satisfied: pygments in /usr/local/lib/python3.11/dist-packages (f
  Requirement already satisfied: backcall in /usr/local/lib/python3.11/dist-packages (f
  Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.11/dist-pa
  Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.11/dist-packages
  Requirement already satisfied: ipykernel>=4.5.1 in /usr/local/lib/python3.11/dist-pac
  Requirement already satisfied: ipython-genutils~0.2.0 in /usr/local/lib/python3.11/d
  Requirement already satisfied: widgetsnbextension~3.6.0 in /usr/local/lib/python3.11
  Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.11
  Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-pac
  Requirement already satisfied: cycycler>=0.10 in /usr/local/lib/python3.11/dist-package
  Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-pa

```

```

Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist
Requirement already satisfied: debugpy>=1.0 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: jupyter-client>=6.1.12 in /usr/local/lib/python3.11/di
Requirement already satisfied: nest-asyncio in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: psutil in /usr/local/lib/python3.11/dist-packages (fro
Requirement already satisfied: pyzmq>=17 in /usr/local/lib/python3.11/dist-packages (
Requirement already satisfied: tornado>=6.1 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: parso<0.9.0,>=0.8.4 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: wcwidth in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: jupyter-core>=4.6.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (fro
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: nbformat in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: nbconvert>=5 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: Send2Trash>=1.8.0 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: terminado>=0.8.3 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: prometheus-client in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: nbclassic>=0.4.7 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: notebook-shim>=0.2.3 in /usr/local/lib/python3.11/dist
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: bleach!=5.0.0 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: defusedxml in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.11/dist-
Requirement already satisfied: markupsafe>=2.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: mistune<4,>=2.0.3 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.11/dist-pack

```

```

data = pd.read_csv('/content/yellow_tripdata_2015-01.csv')
data.head()

```



	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance
0	2	2015-01-15 19:05:39	2015-01-15 19:23:42	1	
1	1	2015-01-10 20:33:38	2015-01-10 20:53:28	1	
2	1	2015-01-10 20:33:38	2015-01-10 20:43:41	1	
3	1	2015-01-10 20:33:39	2015-01-10 20:35:31	1	
4	1	2015-01-10 20:33:39	2015-01-10 20:52:58	1	

```

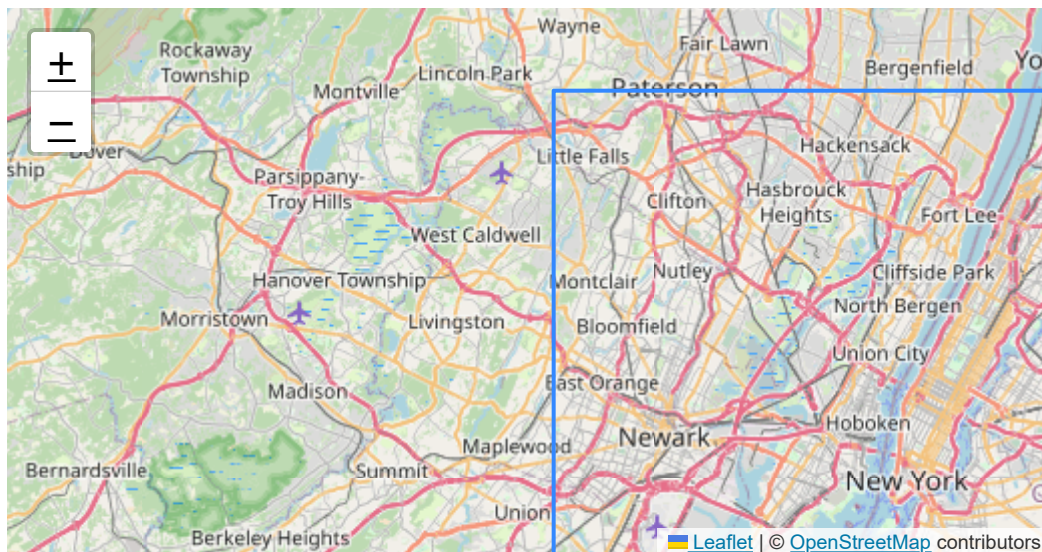
coordinates = [
    [40.915121, -74.247930],
    [40.496226, -74.247930],
    [40.496226, -73.700562].

```

```
[40.915121, -73.700562],  
[40.915121, -74.247930]  
]
```

```
# Create the map and add the line
```

```
map = folium.Map(location=[40.7128, -74.0060], width='80%', height='70%', zoom_start=10)  
polyline = folium.PolyLine(locations=coordinates,weight=2)  
map.add_child(polyline)
```



```
pickup_outliers = data[(data.pickup_latitude < 40.496226) |  
                        (data.pickup_latitude > 40.915121) |  
                        (data.pickup_longitude < -74.247930) |  
                        (data.pickup_longitude > -73.700562) ]
```

```
print("Total pickup outliers :", pickup_outliers.shape[0])

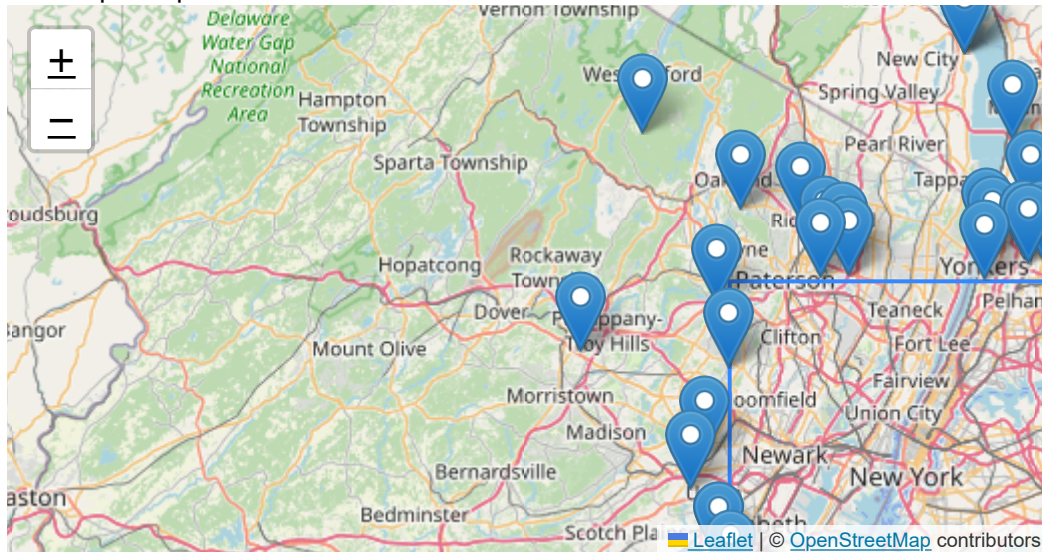
# subset first 100 pickup outliers
sample_locations = pickup_outliers[(pickup_outliers.pickup_latitude != 0.0) &
                                     (pickup_outliers.pickup_longitude != 0.0)].head(100)

# new york city map with bounding box
map = folium.Map(location=[40.7128, -74.0060], width='80%', height='70%', zoom_start=9)
polyline = folium.PolyLine(locations=coordinates, weight=2)
map.add_child(polyline)

# plot outliers markers
for idx, row in sample_locations.iterrows():
    folium.Marker([row.pickup_latitude, row.pickup_longitude]).add_to(map)

map
```

Total pickup outliers : 245762




```

dropoff_outliers = data[(data.dropoff_latitude < 40.496226) |
                        (data.dropoff_latitude > 40.915121) |
                        (data.dropoff_longitude < -74.247930) |
                        (data.dropoff_longitude > -73.700562) ]

print("Total dropoff outliers :", dropoff_outliers.shape[0])

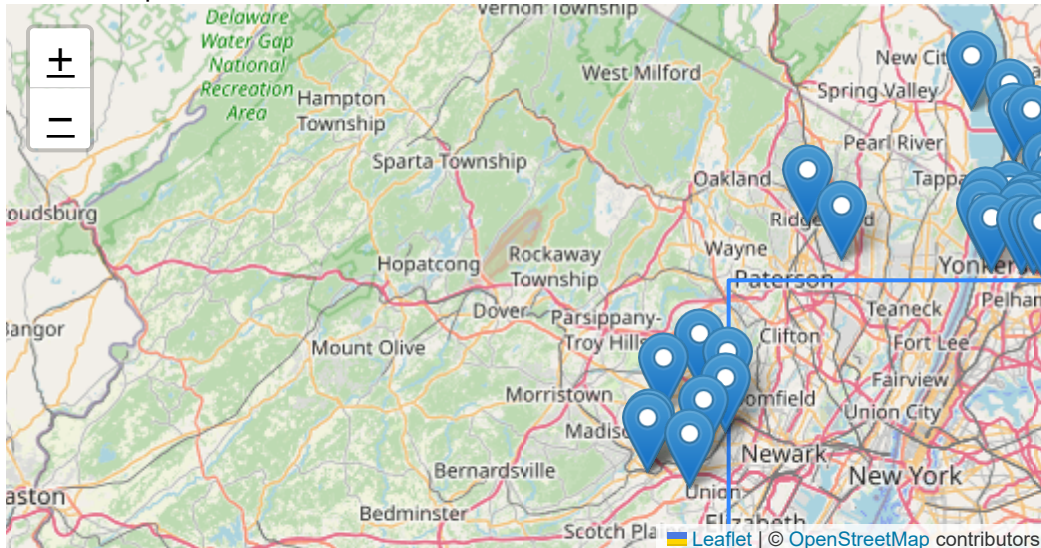
# subset first 100 dropoff outliers
sample_locations = dropoff_outliers[(dropoff_outliers.dropoff_latitude != 0.0) &
                                     (dropoff_outliers.dropoff_longitude != 0.0)].head(100)

# new york city map with bounding box
map = folium.Map(location=[40.7128, -74.0060], width='80%', height='70%', zoom_start=9)
polyline = folium.PolyLine(locations=coordinates,weight=2)
map.add_child(polyline)

# plot outliers markers
for idx, row in sample_locations.iterrows():
    folium.Marker([row.dropoff_latitude, row.dropoff_longitude]).add_to(map)
map

```

Total dropoff outliers : 243843



```
from datetime import datetime

# this function returns the GMT unix timestamp of given datetime str
# it seems that we have been given the datetime str in GMT timezone
def get_unix_timestamp(datetime_str):
    formatted_date = datetime.strptime(datetime_str, "%Y-%m-%d %H:%M:%S")
    return int(datetime.timestamp(formatted_date))

def get_trip_time_with_imp_features(df):
    new_df = df[['passenger_count', 'trip_distance', 'pickup_longitude', 'pickup_latitude',
                 'dropoff_longitude', 'dropoff_latitude', 'total_amount']].copy()

    unix_pickup = np.array([get_unix_timestamp(datetime_str) for datetime_str in df['tpep_pickup_datetime']])
    unix_drop = np.array([get_unix_timestamp(datetime_str) for datetime_str in df['tpep_dropoff_datetime']])
    new_df['trip_time'] = (unix_drop - unix_pickup) / 60.0

    new_df['pickup_time'] = unix_pickup
    new_df['speed'] = (new_df['trip_distance'] / new_df['trip_time']) * 60

    return new_df
```

```
from datetime import datetime

# this function returns the GMT unix timestamp of given datetime str
# it seems that we have been given the datetime str in GMT timezone
def get_unix_timestamp(datetime_str):
    try:
        formatted_date = datetime.strptime(datetime_str, "%Y-%m-%d %H:%M:%S")
    except ValueError:
        # If the format is incorrect, try adding seconds
        try:
            formatted_date = datetime.strptime(datetime_str + ":00", "%Y-%m-%d %H:%M:%S")
        except ValueError:
            # If still incorrect, return None or handle as needed
            return None
    return int(datetime.timestamp(formatted_date))
```

```
def get_trip_time_with_imp_features(df):
    new_df = df[['passenger_count', 'trip_distance', 'pickup_longitude', 'pickup_latitude',
                  'dropoff_longitude', 'dropoff_latitude', 'total_amount']].copy()

    unix_pickup = np.array([get_unix_timestamp(datetime_str) for datetime_str in df['tpep_pickup_datetime']])
    unix_drop = np.array([get_unix_timestamp(datetime_str) for datetime_str in df['tpep_dropoff_datetime']])

    # Remove rows with None values in unix_pickup or unix_drop
    valid_indices = [i for i, (pickup, drop) in enumerate(zip(unix_pickup, unix_drop)) if pickup is not None and drop is not None]
    new_df = new_df.iloc[valid_indices]
    unix_pickup = unix_pickup[valid_indices]
    unix_drop = unix_drop[valid_indices]

    new_df['trip_time'] = (unix_drop - unix_pickup) / 60.0

    new_df['pickup_time'] = unix_pickup
    new_df['speed'] = (new_df['trip_distance'] / new_df['trip_time'].replace(0,0.001)) * 60

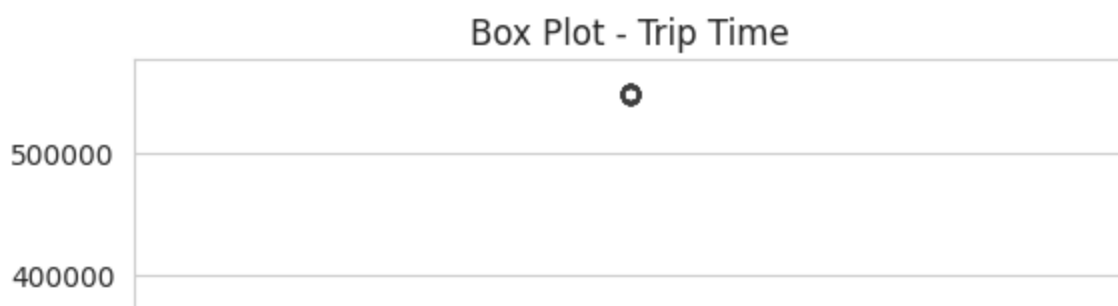
    return new_df

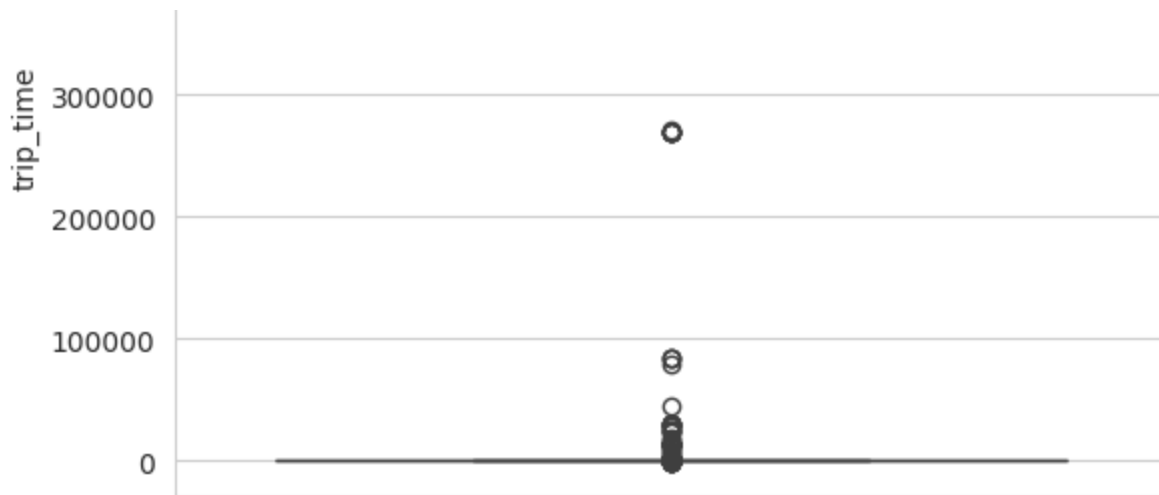
df_imp_feat = get_trip_time_with_imp_features(data)
display(df_imp_feat.head())
df_imp_feat.shape
```

	passenger_count	trip_distance	pickup_longitude	pickup_latitude	dropoff_longitude
0	1	1.59	-73.993896	40.750111	-73.9747
1	1	3.30	-74.001648	40.724243	-73.9944
2	1	1.80	-73.963341	40.802788	-73.9518
3	1	0.50	-74.009087	40.713818	-74.0043
4	1	3.00	-73.971176	40.762428	-74.0041

(12748986, 10)

```
sns.boxplot(y='trip_time', data=df_imp_feat)
plt.title("Box Plot - Trip Time")
plt.show()
```





```
for val in np.arange(0, 110, 10):
    print(val, "percentile : ", np.percentile(df_imp_feat['trip_time'].values, val))
```

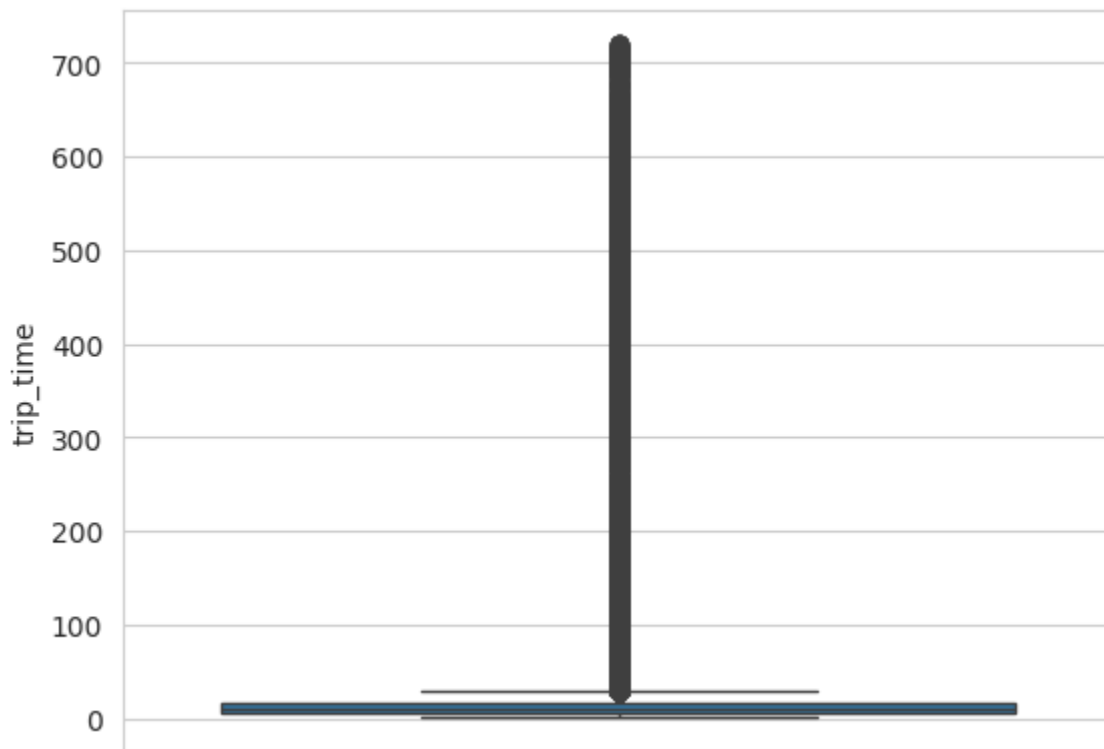
```
0 percentile : -1211.0166666666667
10 percentile : 3.8333333333333335
20 percentile : 5.383333333333334
30 percentile : 6.816666666666666
40 percentile : 8.3
50 percentile : 9.95
60 percentile : 11.866666666666667
70 percentile : 14.283333333333333
80 percentile : 17.633333333333333
90 percentile : 23.45
100 percentile : 548555.6333333333
```

```
for val in np.arange(90, 101, 1):
    print(val, "percentile : ", np.percentile(df_imp_feat['trip_time'].values, val))
```

```
90 percentile : 23.45
91 percentile : 24.35
92 percentile : 25.383333333333333
93 percentile : 26.55
94 percentile : 27.933333333333334
95 percentile : 29.583333333333332
96 percentile : 31.683333333333334
97 percentile : 34.466666666666667
98 percentile : 38.716666666666667
99 percentile : 46.75
100 percentile : 548555.6333333333
```

```
df_imp_feat_mod = df_imp_feat[(df_imp_feat.trip_time > 1) & (df_imp_feat.trip_time < 720)]
```

```
# box plot of modified trip_time
sns.boxplot(y='trip_time', data=df_imp_feat_mod)
plt.show()
```

```
from google.colab import output
output.enable_custom_widget_manager()
```

Support for third party widgets will remain active for the duration of the session. To disable support:

```
from google.colab import output
output.disable_custom_widget_manager()
```

```
df_imp_feat_mod['log_trip_time'] = [np.log(trip_time) for trip_time in df_imp_feat_mod.tr
```

```
<ipython-input-15-ff832546c957>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
df_imp_feat_mod['log_trip_time'] = [np.log(trip_time) for trip_time in df_imp_feat_
```

```
percs = np.arange(0, 101, 10)
values = np.percentile(df_imp_feat_mod.speed.values, percs)
for perc, value in zip(percs, values):
    print(f"{perc} percentile : {value}")
```

```
0 percentile : 0.0
10 percentile : 6.409495548961425
20 percentile : 7.80952380952381
..
..
```

```
30 percentile : 8.929133858267717
40 percentile : 9.98019801980198
50 percentile : 11.06865671641791
60 percentile : 12.286689419795222
70 percentile : 13.796407185628745
80 percentile : 15.963224893917962
90 percentile : 20.186915887850468
100 percentile : 192857142.85714284
```

```
percs = np.arange(90, 101, 1)
values = np.percentile(df_imp_feat_mod.speed.values, percs)
for perc, value in zip(percs, values):
    print(f"{perc} percentile : {value}")
```

```
90 percentile : 20.186915887850468
91 percentile : 20.916454400875093
92 percentile : 21.752988047808763
93 percentile : 22.721893491124263
94 percentile : 23.844155844155843
95 percentile : 25.182552504038775
96 percentile : 26.80851063829787
97 percentile : 28.84304932735426
98 percentile : 31.591128254580514
99 percentile : 35.751350551136035
100 percentile : 192857142.85714284
```

```
df_imp_feat_mod = df_imp_feat_mod[df_imp_feat_mod.speed < 45.32]
```

```
avg_speed = df_imp_feat_mod.speed.sum() / df_imp_feat_mod.speed.count()
print(f"Average speed of cab = {round(avg_speed, 3)} miles/hr")
```

```
Average speed of cab = 12.437 miles/hr
```

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, GRU, Dropout
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

# Load the dataset
df = df_imp_feat_mod.copy() # Use the preprocessed dataset

# Convert pickup_time to datetime and extract useful time-based features
df['pickup_time'] = pd.to_datetime(df['pickup_time'], unit='s')
df['hour'] = df['pickup_time'].dt.hour
df['dayofweek'] = df['pickup_time'].dt.dayofweek
```

```
# Aggregate demand (number of taxi pickups per hour)
df_demand = df.groupby(['hour', 'dayofweek']).size().reset_index(name='demand')

# Feature selection
features = ['hour', 'dayofweek']
target = ['demand']

# Normalize data
scaler_x = MinMaxScaler()
scaler_y = MinMaxScaler()

X = scaler_x.fit_transform(df_demand[features])
y = scaler_y.fit_transform(df_demand[target])

# Reshape for LSTM/GRU input (time-series format)
X = X.reshape((X.shape[0], 1, X.shape[1]))

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

# Define LSTM Model
def build_lstm_model():
    model = Sequential([
        LSTM(64, return_sequences=True, input_shape=(1, X_train.shape[2])),
        Dropout(0.2),
        LSTM(32, return_sequences=False),
        Dense(16, activation='relu'),
        Dense(1)
    ])
    model.compile(optimizer='adam', loss='mse')
    return model

# Define GRU Model
def build_gru_model():
    model = Sequential([
        GRU(64, return_sequences=True, input_shape=(1, X_train.shape[2])),
        Dropout(0.2),
        GRU(32, return_sequences=False),
        Dense(16, activation='relu'),
        Dense(1)
    ])
    model.compile(optimizer='adam', loss='mse')
    return model

# Train LSTM Model
lstm_model = build_lstm_model()
lstm_model.fit(X_train, y_train, epochs=20, batch_size=16, validation_data=(X_test, y_test))

# Train GRU Model
gru_model = build_gru_model()
```

```
gru_model.fit(x_train, y_train, epochs=20, batch_size=16, validation_data=(x_test, y_test)).
```

```
# Predict
```

```
y_pred_lstm = scaler_y.inverse_transform(lstm_model.predict(X_test))
```

```
y_pred_gru = scaler_y.inverse_transform(gru_model.predict(X_test))
```

```
y_actual = scaler_y.inverse_transform(y_test)
```

```
# Plot results
```

```
plt.figure(figsize=(10,5))
```

```
plt.plot(y_actual[:50], label="Actual Demand", color='blue')
```

```
plt.plot(y_pred_lstm[:50], label="LSTM Prediction", color='red', linestyle='dashed')
```

```
plt.plot(y_pred_gru[:50], label="GRU Prediction", color='green', linestyle='dashed')
```

```
plt.legend()
```

```
plt.title("Taxi Demand Prediction using LSTM and GRU")
```

```
plt.show()
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:200: UserWarning:
  super().__init__(**kwargs)
```

```
Epoch 1/20
```

```
9/9 ————— 5s 101ms/step - loss: 0.2171 - val_loss: 0.4167
```

```
Epoch 2/20
```

```
9/9 ————— 2s 13ms/step - loss: 0.1921 - val_loss: 0.3466
```

```
Epoch 3/20
```

```
9/9 ————— 0s 15ms/step - loss: 0.1655 - val_loss: 0.2676
```

```
Epoch 4/20
```

```
9/9 ————— 0s 17ms/step - loss: 0.1147 - val_loss: 0.1796
```

```
Epoch 5/20
```

```
9/9 ————— 0s 21ms/step - loss: 0.0732 - val_loss: 0.0983
```

```
Epoch 6/20
```

```
9/9 ————— 0s 21ms/step - loss: 0.0483 - val_loss: 0.0571
```

```
Epoch 7/20
```

```
9/9 ————— 0s 19ms/step - loss: 0.0458 - val_loss: 0.0517
```

```
Epoch 8/20
```

```
9/9 ————— 0s 25ms/step - loss: 0.0421 - val_loss: 0.0574
```

```
Epoch 9/20
```

```
9/9 ————— 0s 20ms/step - loss: 0.0428 - val_loss: 0.0608
```

```
Epoch 10/20
```

```
9/9 ————— 0s 24ms/step - loss: 0.0403 - val_loss: 0.0565
```

```
Epoch 11/20
```

```
9/9 ————— 0s 23ms/step - loss: 0.0443 - val_loss: 0.0542
```

```
Epoch 12/20
```

```
9/9 ————— 0s 23ms/step - loss: 0.0322 - val_loss: 0.0532
```

```
Epoch 13/20
```

```
9/9 ————— 0s 22ms/step - loss: 0.0388 - val_loss: 0.0527
```

```
Epoch 14/20
```

```
9/9 ————— 0s 12ms/step - loss: 0.0371 - val_loss: 0.0521
```

```
Epoch 15/20
```

```
9/9 ————— 0s 12ms/step - loss: 0.0397 - val_loss: 0.0521
```

```
Epoch 16/20
```

```
9/9 ————— 0s 13ms/step - loss: 0.0320 - val_loss: 0.0528
```

```
Epoch 17/20
```

```
9/9 ————— 0s 12ms/step - loss: 0.0370 - val_loss: 0.0528
```

```
Epoch 18/20
```

```

Epoch 18/20
9/9 ————— 0s 12ms/step - loss: 0.0319 - val_loss: 0.0540
Epoch 19/20
9/9 ————— 0s 17ms/step - loss: 0.0302 - val_loss: 0.0552
Epoch 20/20
9/9 ————— 0s 12ms/step - loss: 0.0304 - val_loss: 0.0556
Epoch 1/20
9/9 ————— 2s 48ms/step - loss: 0.1997 - val_loss: 0.2353
Epoch 2/20
9/9 ————— 0s 12ms/step - loss: 0.0993 - val_loss: 0.0878
Epoch 3/20
9/9 ————— 0s 17ms/step - loss: 0.0469 - val_loss: 0.0502
Epoch 4/20
9/9 ————— 0s 12ms/step - loss: 0.0436 - val_loss: 0.0505
Epoch 5/20
9/9 ————— 0s 12ms/step - loss: 0.0374 - val_loss: 0.0535
Epoch 6/20
9/9 ————— 0s 12ms/step - loss: 0.0311 - val_loss: 0.0521
Epoch 7/20
9/9 ————— 0s 17ms/step - loss: 0.0337 - val_loss: 0.0522
Epoch 8/20
9/9 ————— 0s 12ms/step - loss: 0.0322 - val_loss: 0.0529
Epoch 9/20
9/9 ————— 0s 16ms/step - loss: 0.0286 - val_loss: 0.0549
Epoch 10/20
9/9 ————— 0s 11ms/step - loss: 0.0279 - val_loss: 0.0599
Epoch 11/20
9/9 ————— 0s 12ms/step - loss: 0.0309 - val_loss: 0.0594
Epoch 12/20
9/9 ————— 0s 12ms/step - loss: 0.0328 - val_loss: 0.0568
Epoch 13/20
9/9 ————— 0s 12ms/step - loss: 0.0290 - val_loss: 0.0601
Epoch 14/20
9/9 ————— 0s 18ms/step - loss: 0.0344 - val_loss: 0.0704
Epoch 15/20
9/9 ————— 0s 18ms/step - loss: 0.0340 - val_loss: 0.0637
Epoch 16/20
9/9 ————— 0s 11ms/step - loss: 0.0282 - val_loss: 0.0632
Epoch 17/20
9/9 ————— 0s 17ms/step - loss: 0.0308 - val_loss: 0.0681
Epoch 18/20
9/9 ————— 0s 12ms/step - loss: 0.0294 - val_loss: 0.0661
Epoch 19/20
9/9 ————— 0s 16ms/step - loss: 0.0328 - val_loss: 0.0626
Epoch 20/20
9/9 ————— 0s 11ms/step - loss: 0.0320 - val_loss: 0.0661
2/2 ————— 0s 165ms/step
2/2 ————— 0s 148ms/step

```

Taxi Demand Prediction using LSTM and GRU

