dataset:

	LocationID	busyness	temp	precip	snow	vis	wind_spd	uv	day_of_week	is_holiday	day	month	hour	timeslot
0	4	16.5	10.6	0.0	0.0	13	1.6	0.0	5	False	1	1	0	Late Night
1	12	1.0	10.6	0.0	0.0	13	1.6	0.0	5	False	1	1	0	Late Night
2	13	17.5	10.6	0.0	0.0	13	1.6	0.0	5	False	1	1	0	Late Night
3	24	12.5	10.6	0.0	0.0	13	1.6	0.0	5	False	1	1	0	Late Night
4	41	13.5	10.6	0.0	0.0	13	1.6	0.0	5	False	1	1	0	Late Night
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
495376	163	0.5	2.2	0.0	0.0	16	2.1	0.0	3	False	1	12	22	Late Night
495377	50	0.5	2.2	0.0	0.0	11	1.5	0.0	3	False	1	12	23	Late Night
495378	68	0.5	2.2	0.0	0.0	11	1.5	0.0	3	False	1	12	23	Late Night
495379	79	0.5	2.2	0.0	0.0	11	1.5	0.0	3	False	1	12	23	Late Night
495380	148	0.5	2.2	0.0	0.0	11	1.5	0.0	3	False	1	12	23	Late Night

495381 rows × 14 columns

Input: LocationID, temp, precip, snow, vis, wind_spd, uv, day_of_week, is_holiday, day, month, hour, timeslot

Output: busyness

LocationID: ID of each taxi zone

precip: Accumulated liquid equivalent precipitation (default mm)

wind_spd: Wind speed (Default m/s)

vis: Visibility (default KM)

uv: UV Index (0-11+)

snow: Accumulated snowfall (default mm)

day: 1-31

Month: 1-12

Hour:

applied with the nearest hour

```
         VendorID tpep_pickup_datetime tpep_dropoff_datetime
1809900         1  2022-01-24 15:23:01   2022-01-22 06:00:37  \
2392571         6  2022-01-01 01:01:54   2022-01-01 01:01:36
2392642         6  2022-01-01 01:01:44   2022-01-01 01:01:20
2393302         6  2022-01-01 04:01:37   2022-01-01 04:01:19
2393327         6  2022-01-01 04:01:23   2022-01-01 04:01:12
...           ...                  ...                   ...
39549415        6  2022-12-05 13:12:17   2022-12-05 13:12:08
39549767        6  2022-12-05 15:12:40   2022-12-05 15:12:26
39549861        6  2022-12-05 15:12:59   2022-12-05 15:12:35
39550168        6  2022-12-05 17:12:54   2022-12-05 17:12:17
39550880        6  2022-12-05 20:12:14   2022-12-05 20:12:01
```

```
# Split 'tpep_pickup_datetime' into date and time columns
df['pickup_date'] = df['tpep_pickup_datetime'].dt.date
df['pickup_time'] = df['tpep_pickup_datetime'].dt.time

# Split 'tpep_dropoff_datetime' into date and time columns
df['dropoff_date'] = df['tpep_dropoff_datetime'].dt.date
df['dropoff_time'] = df['tpep_dropoff_datetime'].dt.time

# Remove the original 'tpep_pickup_datetime' and 'tpep_dropoff_datetime' columns
# df = df.drop(['tpep_pickup_datetime', 'tpep_dropoff_datetime'], axis=1)
```

In [36]:
```
def round_to_nearest_hour(time):
    rounded_time = time.replace(minute=0, second=0)
    return rounded_time

df['dropoff_time'] = df['dropoff_time'].apply(round_to_nearest_hour)
df['pickup_time'] = df['pickup_time'].apply(round_to_nearest_hour)
```

```
def round_to_nearest_hour(time):
    rounded_time = time.replace(minute=0, second=0)
    return rounded_time

df['dropoff_time'] = df['dropoff_time'].apply(round_to_nearest_hour)
df['pickup_time'] = df['pickup_time'].apply(round_to_nearest_hour)
```

timeslot:

```python
# Define a function to assign time slots
def time_slots(hour):
    if hour in range(6, 12):
        return 'Morning'
    elif hour in range(12, 17):
        return 'Afternoon'
    elif hour in range(17, 22):
        return 'Evening'
    else:
        return 'Late Night'


merged_data['timeslot'] = merged_data['hour'].apply(time_slots)
```


```
# Define a function to assign time slots
def time_slots(hour):
    if hour in range(6, 12):
        return 'Morning'
    elif hour in range(12, 17):
        return 'Afternoon'
    elif hour in range(17, 22):
        return 'Evening'
    else:
        return 'Late Night'


merged_data['timeslot'] = merged_data['hour'].apply(time_slots)
```


day_of_week:

In [74]:
```
merged_data['date'] = pd.to_datetime(merged_data['date'])
merged_data['day_of_week'] = merged_data['date'].dt.day_name()
merged_data
```

Out[74]:

	LocationID	time	date	busyness	temp	precip	snow	vis	wind_spd	uv	day_of_week
0	4	00:00:00	2022-01-01	16.5	10.6	0.0	0.0	13	1.6	0.0	Saturday
1	12	00:00:00	2022-01-01	1.0	10.6	0.0	0.0	13	1.6	0.0	Saturday
2	13	00:00:00	2022-01-01	17.5	10.6	0.0	0.0	13	1.6	0.0	Saturday
3	24	00:00:00	2022-01-01	12.5	10.6	0.0	0.0	13	1.6	0.0	Saturday
4	41	00:00:00	2022-01-01	13.5	10.6	0.0	0.0	13	1.6	0.0	Saturday
...	...	...	...	...	...	...	...	...	...	...	...
495376	163	22:00:00	2022-12-01	0.5	2.2	0.0	0.0	16	2.1	0.0	Thursday
495377	50	23:00:00	2022-12-01	0.5	2.2	0.0	0.0	11	1.5	0.0	Thursday
495378	68	23:00:00	2022-12-01	0.5	2.2	0.0	0.0	11	1.5	0.0	Thursday
495379	79	23:00:00	2022-12-01	0.5	2.2	0.0	0.0	11	1.5	0.0	Thursday
495380	148	23:00:00	2022-12-01	0.5	2.2	0.0	0.0	11	1.5	0.0	Thursday

495381 rows × 11 columns

In [75]:
```
day_to_number = {
    'Monday': 0,
    'Tuesday': 1,
    'Wednesday': 2,
    'Thursday': 3,
    'Friday': 4,
    'Saturday': 5,
    'Sunday': 6
}

merged_data['day_of_week'] = merged_data['day_of_week'].map(day_to_number)
merged_data
```

Out[75]:

	LocationID	time	date	busyness	temp	precip	snow	vis	wind_spd	uv	day_of_week
0	4	00:00:00	2022-01-01	16.5	10.6	0.0	0.0	13	1.6	0.0	5
1	12	00:00:00	2022-01-01	1.0	10.6	0.0	0.0	13	1.6	0.0	5
2	13	00:00:00	2022-01-01	17.5	10.6	0.0	0.0	13	1.6	0.0	5
3	24	00:00:00	2022-01-01	12.5	10.6	0.0	0.0	13	1.6	0.0	5
4	41	00:00:00	2022-01-01	13.5	10.6	0.0	0.0	13	1.6	0.0	5
...	...	...	...	...	...	...	...	...	...	...	...
495376	163	22:00:00	2022-12-01	0.5	2.2	0.0	0.0	16	2.1	0.0	3
495377	50	23:00:00	2022-12-01	0.5	2.2	0.0	0.0	11	1.5	0.0	3
495378	68	23:00:00	2022-12-01	0.5	2.2	0.0	0.0	11	1.5	0.0	3


```
merged_data['date'] = pd.to_datetime(merged_data['date'])
merged_data['day_of_week'] = merged_data['date'].dt.day_name()
merged_data

day_to_number = {
    'Monday': 0,
    'Tuesday': 1,
    'Wednesday': 2,
    'Thursday': 3,
    'Friday': 4,
    'Saturday': 5,
```

```
    'Sunday': 6
}

merged_data['day_of_week'] = merged_data['day_of_week'].map(day_to_number)
merged_data
```

is_holiday:

In [76]:
```
from pandas.tseries.holiday import USFederalHolidayCalendar as calendar

cal = calendar()
holidays = cal.holidays(start='2022-01-01', end='2022-12-31')

# Check if the pickup date is a holiday and assign day types
merged_data['is_holiday'] = merged_data['date'].isin(holidays)
merged_data
```

Out[76]:

	LocationID	time	date	busyness	temp	precip	snow	vis	wind_spd	uv	day_of_week	is_holiday
0	4	00:00:00	2022-01-01	16.5	10.6	0.0	0.0	13	1.6	0.0	5	False
1	12	00:00:00	2022-01-01	1.0	10.6	0.0	0.0	13	1.6	0.0	5	False
2	13	00:00:00	2022-01-01	17.5	10.6	0.0	0.0	13	1.6	0.0	5	False
3	24	00:00:00	2022-01-01	12.5	10.6	0.0	0.0	13	1.6	0.0	5	False
4	41	00:00:00	2022-01-01	13.5	10.6	0.0	0.0	13	1.6	0.0	5	False
...	...	...	...	...	...	...	...	...	...	...	...	...
495376	163	22:00:00	2022-12-01	0.5	2.2	0.0	0.0	16	2.1	0.0	3	False
495377	50	23:00:00	2022-12-01	0.5	2.2	0.0	0.0	11	1.5	0.0	3	False
495378	68	23:00:00	2022-12-01	0.5	2.2	0.0	0.0	11	1.5	0.0	3	False
495379	79	23:00:00	2022-12-01	0.5	2.2	0.0	0.0	11	1.5	0.0	3	False
495380	148	23:00:00	2022-12-01	0.5	2.2	0.0	0.0	11	1.5	0.0	3	False

495381 rows × 12 columns

```
from pandas.tseries.holiday import USFederalHolidayCalendar as calendar

cal = calendar()
holidays = cal.holidays(start='2022-01-01', end='2022-12-31')

# Check if the pickup date is a holiday and assign day types
merged_data['is_holiday'] = merged_data['date'].isin(holidays)
merged_data
```