

Handling_Date_&_Time.ipynb

```
In [2]: #Using datetime Module

from datetime import datetime

# Get current date and time
now = datetime.now()
print(now)

# Create a datetime object
date_obj = datetime(2024, 9, 27, 14, 30) # year, month, day, hour, minute
print(date_obj)

# Convert string to datetime
date_str = '2024-09-27 14:30:00'
date_format = datetime.strptime(date_str, '%Y-%m-%d %H:%M:%S')
print(date_format)

# Convert datetime to string
formatted_date = date_obj.strftime('%Y-%m-%d %H:%M:%S')
print(formatted_date)

# Difference between dates
time_diff = datetime.now() - date_obj
print(time_diff)

2024-10-14 04:47:36.577421
2024-09-27 14:30:00
2024-09-27 14:30:00
2024-09-27 14:30:00
16 days, 14:17:36.578419
```

In [4]: *#Using Pandas for Time Series Analysis*

```
import pandas as pd

# Create a time series
date_range = pd.date_range(start='2024-01-01', periods=5, freq='Y')
print(date_range)

# Create a DataFrame with a DatetimeIndex
data = {'values': [10, 20, 30, 40, 50]}
df = pd.DataFrame(data, index=date_range)
print(df)

# Convert a column to datetime
df['dates'] = ['2024-01-01 10:00:00', '2024-01-02 11:00:00', '2024-01-03 12:00:00']
df['dates'] = pd.to_datetime(df['dates'])
print(df)

# Extract date/time features
df['year'] = df.index.year
df['month'] = df.index.month
df['day'] = df.index.day
df['hour'] = df.index.hour
df['minute'] = df.index.minute

print(df)
```

```
DatetimeIndex(['2024-12-31', '2025-12-31', '2026-12-31', '2027-12-31',
               '2028-12-31'],
              dtype='datetime64[ns]', freq='A-DEC')
values 2024-12-31      10
2025-12-31      20
2026-12-31      30
2027-12-31      40
2028-12-31      50
dates 2024-12-31      10 2024-01-01
10:00:00
2025-12-31      20 2024-01-02 11:00:00
2026-12-31      30 2024-01-03 12:00:00
2027-12-31      40 2024-01-04 13:00:00
2028-12-31      50 2024-01-05 14:00:00
values
dates year month day hour minute 2024-12-31      10 2024-01-01
10:00:00 2024      12 31      0      0
2025-12-31      20 2024-01-02 11:00:00 2025      12 31      0      0
2026-12-31      30 2024-01-03 12:00:00 2026      12 31      0      0
2027-12-31      40 2024-01-04 13:00:00 2027      12 31      0      0
2028-12-31      50 2024-01-05 14:00:00 2028      12 31      0      0
```

In [5]: *#Handling Time-Zone*

```
import pytz

# Set the timezone to UTC
df['timestamp_utc'] = df['dates'].dt.tz_localize('UTC')

# Convert to another timezone (e.g., 'Asia/Kolkata')
df['timestamp_ist'] = df['timestamp_utc'].dt.tz_convert('Asia/Kolkata')

print(df[['dates', 'timestamp_utc', 'timestamp_ist']])
```

	dates	timestamp_utc	\ 2024-12-
31	2024-01-01 10:00:00	2024-01-01 10:00:00+00:00	
2025-12-31	2024-01-02 11:00:00	2024-01-02 11:00:00+00:00	
2026-12-31	2024-01-03 12:00:00	2024-01-03 12:00:00+00:00	
2027-12-31	2024-01-04 13:00:00	2024-01-04 13:00:00+00:00	
2028-12-31	2024-01-05 14:00:00	2024-01-05 14:00:00+00:00	

	timestamp_ist	2024-12-31
2024-01-01	15:30:00+05:30	
2025-12-31	2024-01-02 16:30:00+05:30	
2026-12-31	2024-01-03 17:30:00+05:30	
2027-12-31	2024-01-04 18:30:00+05:30	
2028-12-31	2024-01-05 19:30:00+05:30	

In [8]: *#Lag Features for Time Series Forecasting*

```
# Create lagged features
df['lagged_values'] = df['values'].shift(1)
df['lagged_values_2'] = df['values'].shift(2)

print(df)
```

	values	dates	year	month	day	hour	minute	\
2024-12-31	10	2024-01-01 10:00:00	2024	12	31	0	0	
2025-12-31	20	2024-01-02 11:00:00	2025	12	31	0	0	
2026-12-31	30	2024-01-03 12:00:00	2026	12	31	0	0	
2027-12-31	40	2024-01-04 13:00:00	2027	12	31	0	0	
2028-12-31	50	2024-01-05 14:00:00	2028	12	31	0	0	

	timestamp_utc	timestamp_ist	lagged_values
\			
2024-12-31	2024-01-01 10:00:00+00:00	2024-01-01 15:30:00+05:30	NaN
2025-12-31	2024-01-02 11:00:00+00:00	2024-01-02 16:30:00+05:30	10.0
2026-12-31	2024-01-03 12:00:00+00:00	2024-01-03 17:30:00+05:30	20.0
2027-12-31	2024-01-04 13:00:00+00:00	2024-01-04 18:30:00+05:30	30.0
2028-12-31	2024-01-05 14:00:00+00:00	2024-01-05 19:30:00+05:30	40.0

	lagged_values_2	rolling_mean	2024-12-31
NaN	NaN		
2025-12-31	NaN	NaN	
2026-12-31	10.0	20.0	
2027-12-31	20.0	30.0	

2028-12-31	30.0	40.0
------------	------	------