

Handling Date and Time Variables in ML

Date and time values carry **rich information** — if used correctly, they can significantly improve your model's performance.

➤ What Are Date-Time Variables?

Examples include:

- 2024-03-10
- 10/03/2024 14:35
- 2023-12-25 09:00:00

These can represent:

- Dates (calendar info)
 - Times (clock info)
 - Timestamps (both date and time)
-

➤ How to Handle Them?

1 Parsing the Date-Time

- Convert the text-formatted date into an actual **datetime object** so it can be processed.
-

2 Extracting Features

You can break the datetime into useful parts:

Extracted Feature	Meaning
Year	Long-term trends (e.g., sales over years)
Month	Seasonal effects (e.g., winter sales)
Day	Daily trends or anomalies
Hour	Time of day patterns (e.g., peak traffic)
Dayofweek	Weekday vs weekend behaviors
is_weekend	Binary feature: weekend or not
Quarter	Quarterly cycles (business or finance)

These new columns can act as powerful features.

3 Time Differences / Duration

- Calculate how much time has passed between two events.

E.g., days_since_last_purchase, time_between_visits

4 Handling Missing Values

- **Imputation:** Fill missing dates using a strategy (e.g., previous value, mean date).
- **Interpolation:** For time series, estimate missing points using trends in the data.

5 Cyclic Encoding

- For features like hour, month, or day-of-week (which wrap around), consider:
 1. Representing them using sine/cosine to preserve their cyclic nature.

➤ Common Pitfalls

- Using raw datetime strings in models (not meaningful)
- Ignoring timezone or daylight saving
- Missing values not handled properly

Why It Matters

- Models can detect **patterns over time** (e.g., weekly sales, hourly traffic)
 - Improves predictions in time-related tasks (forecasting, churn prediction, etc.)
-