

# EDA-Lag Features

---

## Lag Features

In Exploratory Data Analysis (EDA), **lag features** are a type of feature engineering technique used to extract information from time series data. A lag feature is essentially a shifted version of the original variable, where each value is replaced by its corresponding value at some previous time point.

## Why Lag Features?

Lag features can help reveal patterns and relationships in time series data that may not be immediately apparent when looking at the raw data. They can also be used to:

1. **Account for temporal dependencies:** By incorporating lagged values, you can capture effects that occur after a certain period.
2. **Improve model performance:** Lag features can provide additional context to your models, leading to better predictions and forecasts.

## Types of Lag Features

There are several types of lag features, including:

1. **Lag [n]:** This is the most basic type of lag feature, where each value is replaced by its corresponding value at time  $t-n$ .
2. **Lead [n]:** Similar to lags, but looking forward in time instead of backward.
3. **Difference (e.g., `diff(1)`, `diff(12)`):** This calculates the difference between consecutive values or values separated by some fixed period.

## Example

Suppose we have a dataset with daily sales data for an e-commerce company. We want to analyze the effect of last week's sales on this week's sales.

```
import pandas as pd

# Sample data
data = {'Date': ['2022-01-01', '2022-01-02', ..., '2022-12-31'],
        'Sales': [100, 120, 110, ..., 500]}
df = pd.DataFrame(data)

# Create lag feature (last week's sales)
df['Lag_Sales'] = df.groupby('Date')['Sales'].transform(lambda x: x.shift(7))

# Create lead feature (next week's sales)
df['Lead_Sales'] = df.groupby('Date')['Sales'].transform(lambda x: x.shift(-7))
```

In this example, **Lag\_Sales** represents the sales from last week for each corresponding day, while **Lead\_Sales** represents the sales for next week.

## Advice

When working with lag features:

- Be mindful of seasonality and trends in your data when selecting lag periods.
- Experiment with different types of lag features (e.g., simple lags vs. exponential smoothing) to find what works best for your problem.
- Consider normalizing or scaling your lag features to prevent feature dominance.

By incorporating lag features into your EDA workflow, you can gain a deeper understanding of the underlying dynamics in your time series data and make more informed decisions about modeling and forecasting.