

Exercise 1: One-Hot Encoding for Categorical Variables

You are analyzing customer data for an online retailer, and the dataset includes the following information about each customer's preferred delivery method:

Customer	Preferred_Delivery
1	Standard
2	Express
3	Standard
4	Same-Day
5	Express

Tasks

1. Use one-hot encoding to convert the **Preferred_Delivery** feature into numerical values.
2. **Explain One-Hot Encoding:**
 - Briefly describe why one-hot encoding is useful for categorical variables.
 - Identify a potential downside of one-hot encoding when there are many categories in the column.

Exercise 2: Label Encoding for Ordinal Variables

A survey is conducted to rate customer satisfaction, with responses categorized as "Very Unsatisfied," "Unsatisfied," "Neutral," "Satisfied," and "Very Satisfied."

Respondent	Satisfaction_Level
1	Neutral
2	Very Satisfied
3	Unsatisfied
4	Satisfied
5	Very Unsatisfied

Tasks

1. Apply Label Encoding to convert the **Satisfaction_Level** into numerical data.
2. Explain the Use of Label Encoding:
 - Describe why label encoding is appropriate for ordered categories like satisfaction levels.
 - Consider what might go wrong if this approach was used on a non-ordinal categorical variable, such as color.

Exercise 3: Target Encoding for High Cardinality Categorical Data

A mobile app company is analyzing user behavior, and one feature of interest is City, which has too many unique values to feasibly use one-hot encoding. You have a table with sample users and their City and the Average_Spending in the app:

User	City	Average_Spending
1	New York	15
2	Los Angeles	10
3	New York	12
4	Chicago	8
5	Chicago	9
6	Los Angeles	11

Tasks

1. Calculate Target Encoding for Each City by considering the **Average_Spending**.
2. **Explain Target Encoding:**
 - Discuss why target encoding is used in cases where one-hot encoding might produce too many columns.
 - Point out that this technique introduces information from the target variable, so it should be used carefully to avoid data leakage.

Exercise 4: Binary Encoding for Nominal Data with Many Categories

A company has a Product_ID feature with many unique values in their sales data. Encoding each product individually with one-hot encoding would result in too many columns. Here is a sample of the data:

Transaction	Product_ID
1	P1
2	P2
3	P3
4	P4
5	P1

Tasks

1. **Apply Binary Encoding:**
 - Convert each Product_ID to its binary equivalent.
 - Create new columns for each binary digit, so each Product_ID is represented by a compact series of binary columns.
2. **Explain Binary Encoding:**
 - Discuss how binary encoding provides a middle ground by reducing the number of columns compared to one-hot encoding while avoiding numeric ordering in the encoding.
 - Point out that it's useful for high-cardinality nominal data where there's no inherent order.