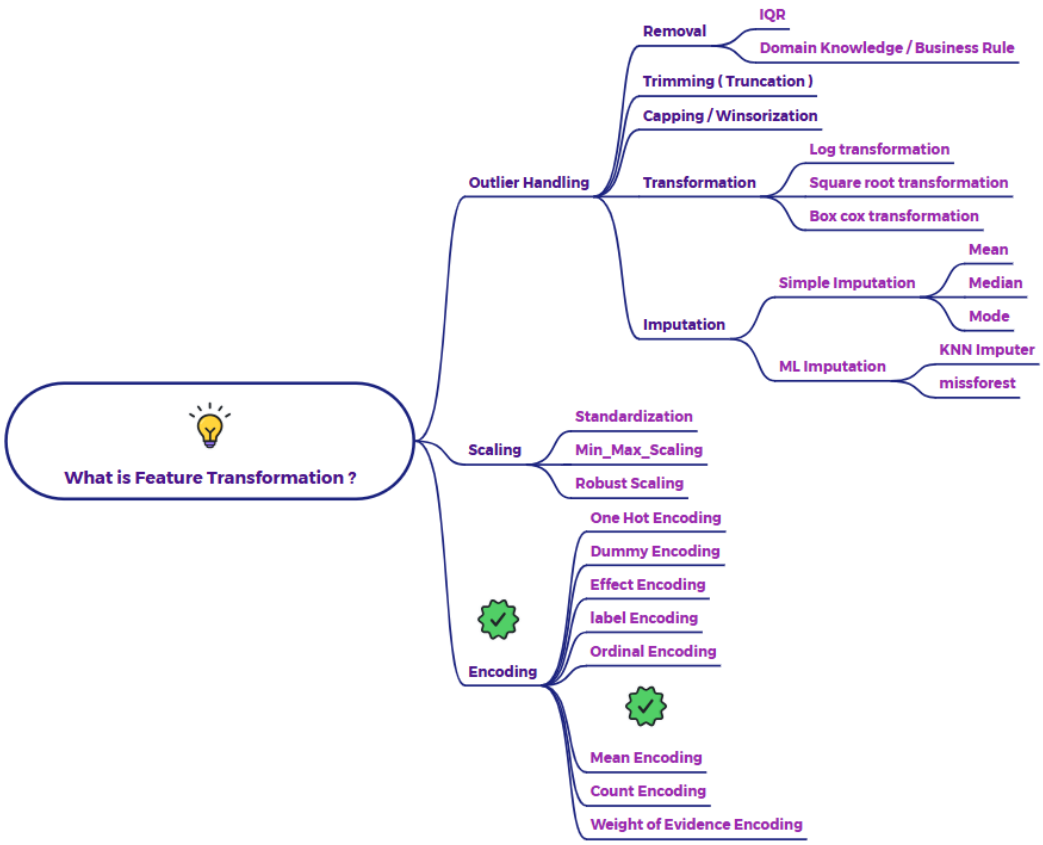
Explain Mean Encoding with an example



1. **Explanation of Mean Encoding**

Mean encoding is a technique used to convert categorical variables into numerical values by replacing each category with the mean of the target variable for that category. In simpler terms, for each category in a categorical column, we calculate the average value of the dependent variable (the variable we're trying to predict) and use that average value to represent the category.

2. **How to Calculate Mean Encoding**

Here's a step-by-step explanation with an example:

Example:

Suppose we have a dataset of customer information:

City	Spend
New York	100
London	150
New York	120
Tokyo	200
London	180
Tokyo	250

We want to encode the "City" column using mean encoding with "Spend" as the target variable.

1. **Group by Category:** Group the data by the categorical column ("City") and calculate the mean of the target variable ("Spend") for each group.
    - New York: (100 + 120) / 2 = 110
    - London: (150 + 180) / 2 = 165
    - Tokyo: (200 + 250) / 2 = 225
2. **Create Mapping:** Create a mapping (e.g., a dictionary) between each unique category and its corresponding mean value.
    - {"New York": 110, "London": 165, "Tokyo": 225}
3. **Replace Categories with Means:** Replace the original categories in the "City" column with their calculated mean values.

The resulting mean-encoded data looks like this:

City	Spend
110	100
165	150
110	120
225	200
165	180
225	250

3. **When to Use Mean Encoding**
    - When you have a categorical variable and a numerical target variable.
    - When you want to capture the relationship between the categories and the target variable in a simple numerical representation.
    - Mean encoding can work well with tree-based models.
4. **Strengths and Weaknesses of Mean Encoding**
    - **Strengths:**
        - Simple to implement.
        - Captures the relationship between the categorical variable and the target variable.
        - Can improve the performance of some models.

- **Weaknesses:**
    - **prone to overfitting**, especially with categories that have few data points.
    - Can lead to data leakage if not implemented carefully (e.g., if the mean for a category is calculated using the data point being encoded).