Q1. What is data encoding? How is it useful in data science?

Data encoding is the process of converting categorical data into numerical form so that it can be used for machine learning algorithms. It is useful in data science because many machine learning algorithms work with numerical data, and encoding allows us to represent categorical information in a way that these algorithms can process.

Q2. What is nominal encoding? Provide an example of how you would use it in a real-world scenario.

Nominal encoding involves assigning unique integers to categorical values. For example, if you have categories like "red," "blue," and "green," nominal encoding would replace them with integers like 1, 2, and 3, respectively. This encoding doesn't impose any ordinal relationship between the categories.

Real-world scenario: In a survey dataset, you have a column "City" with categorical values like "New York," "London," and "Paris." You can use nominal encoding to convert these city names into numerical values (e.g., 1 for New York, 2 for London, and 3 for Paris).

Q3. In what situations is nominal encoding preferred over one-hot encoding? Provide a practical example.

Nominal encoding is preferred over one-hot encoding when the categorical variable has many unique categories. One-hot encoding creates a binary column for each unique category, leading to a high number of columns, which can increase the dimensionality of the dataset significantly.

Practical example: Consider a dataset with a "Country" column containing 100 different country names. One-hot encoding would create 100 new columns, which might cause issues with computational resources and model performance. In such cases, nominal encoding might be preferred.

Q4. Suppose you have a dataset containing categorical data with 5 unique values. Which encoding technique would you use to transform this data into a format suitable for machine learning algorithms? Explain why you made this choice.

For a dataset with 5 unique values, nominal encoding would be a suitable choice. Since the number of unique values is relatively small, nominal encoding can efficiently represent the categorical data as numerical values without significantly increasing dimensionality.

Q5. In a machine learning project, you have a dataset with 1000 rows and 5 columns. Two of the columns are categorical, and the remaining three columns are numerical. If you were to use nominal encoding to transform the categorical data, how many new columns would be created? Show your calculations.

If you're using nominal encoding for categorical data, the number of new columns created would depend on the number of unique categories in each categorical column. Assuming the two categorical columns have 4 and 6 unique categories, respectively:

Column 1: 4 unique categories -> 1 column (Nominal encoding) Column 2: 6 unique categories -> 1 column (Nominal encoding)

Total new columns created for categorical data = 1 + 1 = 2 new columns.

Q6. You are working with a dataset containing information about different types of animals, including their species, habitat, and diet. Which encoding technique would you use to transform the categorical data into a format suitable for machine learning algorithms? Justify your answer.

For categorical variables like species, habitat, and diet, where there's no inherent order or ranking between categories, nominal encoding would be suitable. Nominal encoding allows representing these categorical variables with unique numerical values without implying any ordinal relationship between them.

Q7.You are working on a project that involves predicting customer churn for a telecommunications company. You have a dataset with 5 features, including the customer's gender, age, contract type, monthly charges, and tenure. Which encoding technique(s) would you use to transform the categorical data into numerical data? Provide a step-by-step explanation of how you would implement the encoding.

For this scenario:

Gender: As it has only two categories (male, female), you can use nominal encoding (e.g., 0 for male, 1 for female). Contract type: If it has multiple categories (e.g., month-to-month, one-year, two-year), nominal encoding would work well to assign numerical values to each category (e.g., 1, 2, 3). The steps involve identifying categorical columns, mapping each unique category to a numerical value using techniques like label encoding or one-hot encoding based on the nature of the categorical data. For binary categories like gender, nominal encoding suffices, while for multi-category columns like contract type, nominal encoding could be used unless there's an inherent order, in which case ordinal encoding might be appropriate.