Assignment – 20 March

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

Data Encoding is the process of converting categorical data into a numerical format that machine learning algorithms can interpret. Since most ML models work with numerical data, encoding categorical data ensures that the information is represented effectively without introducing bias.

Uses in Data Science:

- **Compatibility**: Makes categorical data usable by ML algorithms.
- **Performance**: Improves model training and prediction by providing numeric representations of categories.
- **Interpretability**: Allows models to understand the relationships between categories numerically.

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

Nominal Encoding assigns numeric values to categories without assuming any order or hierarchy. Each category is mapped to a unique integer.

Example: In a dataset for a car rental service:

- Column: **Car Type**
- Categories: Sedan, SUV, Hatchback
- Nominal Encoding: Sedan = 0, SUV = 1, Hatchback = 2

Use Case: If the car type affects rental prices, nominal encoding translates these categories into numbers to enable predictive modeling.

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

Nominal Encoding is preferred when:

1. There are many categories in the data.
2. The dataset is large, and memory optimization is critical.
3. No inherent order exists between categories.

Example: A dataset with 1000 rows and a "City" column containing 50 unique city names:

- Using **one-hot encoding** would create 50 new columns.
- Using **nominal encoding** would use a single column with values [0-49], saving memory and computation time.

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.

Choice: Use **one-hot encoding**.

- If the categories are nominal and unordered, one-hot encoding avoids implying an incorrect order or hierarchy between categories.
- **Example**:
    - Categories: A, B, C, D, E
    - One-hot Encoding:

A: [1, 0, 0, 0, 0]

B: [0, 1, 0, 0, 0]

C: [0, 0, 1, 0, 0]

D: [0, 0, 0, 1, 0]

E: [0, 0, 0, 0, 1]

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.

Assume:

- Column 1 (Categorical): 4 unique values
- Column 2 (Categorical): 5 unique values

With **nominal encoding**, each column is replaced by a single numeric column:

- Total new columns = 2 (one for each categorical column).

Final dataset:

- 3 numerical columns + 2 nominally encoded columns = **5 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.

Choice: Use a combination of **one-hot encoding** and **label encoding**:

- **Species**: One-hot encoding is better as species categories often lack order, and using one-hot prevents bias.
- **Habitat**: One-hot encoding ensures no hierarchy is assumed.
- **Diet**: Use label encoding if diets have a logical order (e.g., herbivore < omnivore < carnivore).

Justification:

- One-hot encoding captures relationships without imposing order.
- Label encoding reduces complexity for ordinal categories.

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.

Solution:

1. **Features**:
   - **Gender**: Binary categorical (Male/Female).
   - **Contract Type**: Multiple categories (Monthly, Yearly, Bi-Annual).
2. **Encoding Process**:
   - **Step 1**: Use **binary encoding** for Gender:
     - Male = 0, Female = 1
   - **Step 2**: Use **one-hot encoding** for Contract Type:

        - Monthly = [1, 0, 0], Yearly = [0, 1, 0], Bi-Annual = [0, 0, 1]

    - **Step 3**: Retain **age**, **monthly charges**, and **tenure** as they are numerical features.

3. **Implementation**:
    - Import necessary libraries: pandas and sklearn.
    - Apply LabelEncoder for binary features and OneHotEncoder for multi-category features.
    - Concatenate the transformed categorical columns with the numerical columns.

Example Table:

Gender	**Age**	**Contract Type**	**Monthly Charges**	**Tenure**
Male	25	Monthly	30	12
Female	40	Yearly	50	24

After encoding:

Gender	**Age**	**Monthly**	**Yearly**	**Bi-Annual**	**Monthly Charges**	**Tenure**
0	25	1	0	0	30	12
1	40	0	1	0	50	24