import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder, OrdinalEncoder, OneHotEncoder

# Sample DataFrame

data = {

'color': ['red', 'blue', 'green', 'red', 'blue'],

'size': ['small', 'medium', 'large', 'medium', 'small'],

'grade': ['A', 'B', 'C', 'B', 'A'],

'city': ['Bengaluru', 'Mysuru', 'Hubli', 'Bengaluru', 'Mysuru']

}

df = pd.DataFrame(data)

df

#encoded\_df = df.copy()

encoded\_df = df

encoded\_df

# 1. Label Encoding

encoded\_df = df.copy()

#encoded\_df = df

label\_encoder = LabelEncoder()

encoded\_df['color\_label'] = label\_encoder.fit\_transform(encoded\_df['color'])

encoded\_df

# 2. Ordinal Encoding

encoded\_df = df.copy()

ordinal\_encoder = OrdinalEncoder(categories=[['small', 'medium', 'large']])

encoded\_df['size\_ordinal'] = ordinal\_encoder.fit\_transform(encoded\_df[['size']])

encoded\_df

# One-Hot Encoding

encoded\_df = pd.get\_dummies(df, columns=['color'], drop\_first=False)

encoded\_df

# Dummy Encoding

encoded\_df = pd.get\_dummies(df, columns=['color'], drop\_first=True)

encoded\_df

from category\_encoders import BinaryEncoder

# Create Binary Encoder

binary\_encoder = BinaryEncoder(cols=['grade'])

# Apply Binary Encoding to 'grade'

encoded\_df = binary\_encoder.fit\_transform(df)

encoded\_df

# 5. Frequency Encoding

encoded\_df = df.copy()

encoded\_df['city\_encoded'] = df['city'].map(df['city'].value\_counts())

encoded\_df