

# Data Encoding

**Goal:** Convert categorical variables into numeric form for use in ML models.

## 1. Label Encoding

Python Example:

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['encoded_label'] = le.fit_transform(df['category'])
```

## 2. One-Hot Encoding

Python Example:

```
import pandas as pd

df_encoded = pd.get_dummies(df, columns=['category'], drop_first=
    True)
```

## 3. Ordinal Encoding

Python Example:

```
from sklearn.preprocessing import OrdinalEncoder

edu_levels = [['High School', 'Bachelors', 'Masters', 'PhD']]
oe = OrdinalEncoder(categories=edu_levels)
df['education_encoded'] = oe.fit_transform(df[['education']])
```

## 4. Binary Encoding

Python Example:

```
# pip install category_encoders
import category_encoders as ce

encoder = ce.BinaryEncoder(cols=['city'])
df_encoded = encoder.fit_transform(df)
```

## 5. Custom Encoding (Using Mapping)

**Purpose:** When you want full control over the numeric values assigned to categories.

**Use case:** You know the meaning/order of categories or want compact/meaningful values.

**Python Example:**

```
# Suppose we have this feature:  
# df['weather'] = ['sunny', 'rainy', 'cloudy', 'sunny']  
  
weather_map = {'sunny': 2, 'cloudy': 1, 'rainy': 0}  
df['weather_encoded'] = df['weather'].map(weather_map)
```

**Tip:** You can also use `.replace()` similarly:

```
df['weather_encoded'] = df['weather'].replace(weather_map)
```