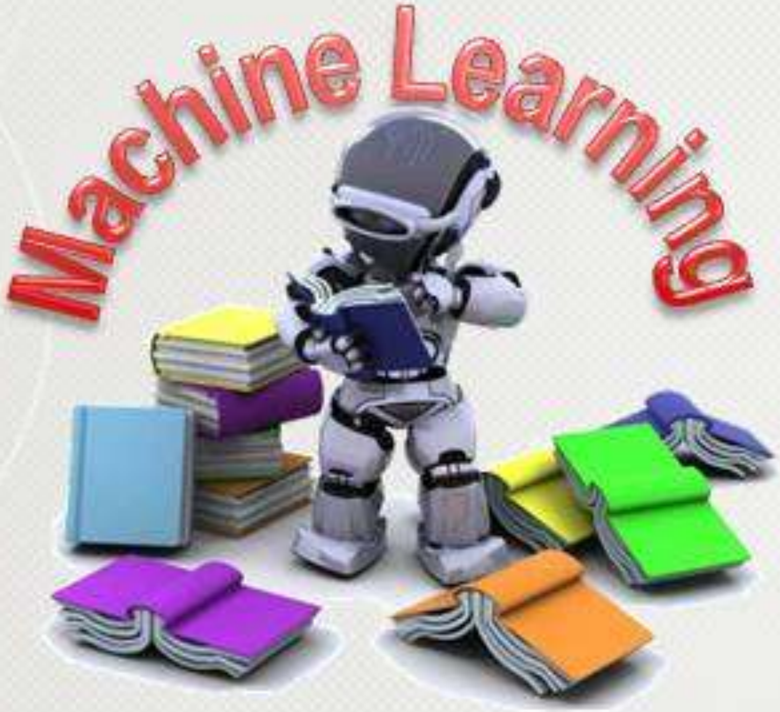
teleuniv
Innovative Interactive Immersive



Categorical Data Preprocessing

Introduction

Why preprocessing ?

Real world data are generally

Incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data

Noisy: containing errors or outliers

Inconsistent: containing discrepancies in codes or names Tasks in data preprocessing

Introduction

- Data pre-processing is an important step of solving every machine learning problem.
- Most of the datasets used with Machine Learning problems need to be processed / cleaned / transformed so that a Machine Learning algorithm can be trained on it.
- Most commonly used pre-processing techniques are very few like - missing value imputation, encoding categorical variables, scaling, etc.

Categorical data

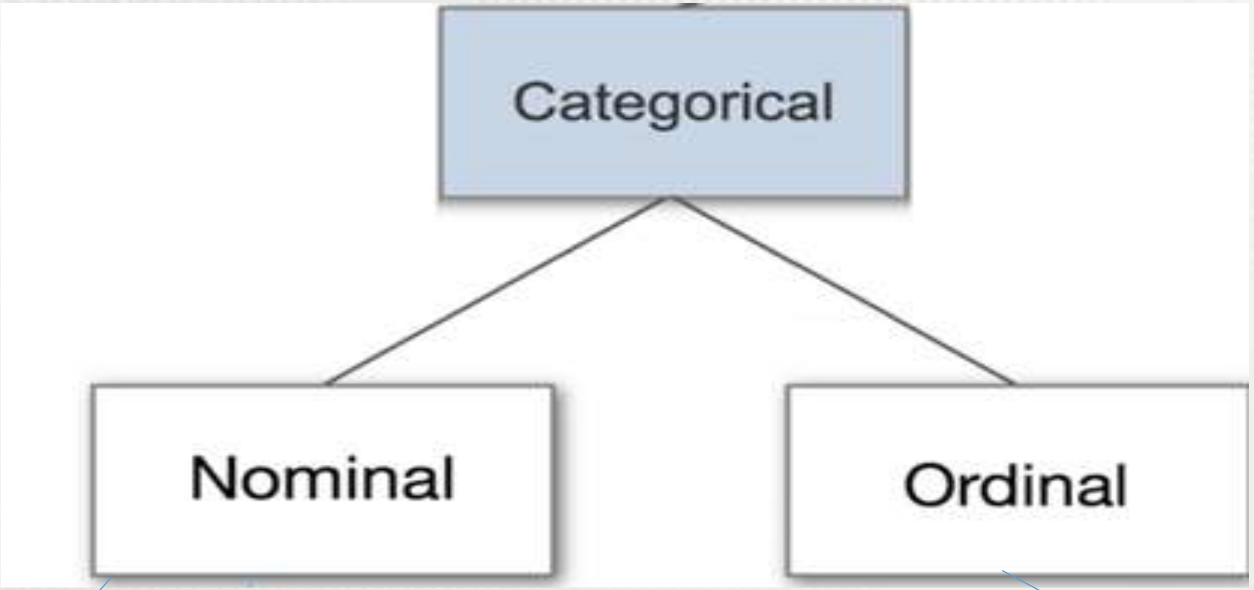
Categorical Attributes –

- When the number unique values in a categorical column are too high, check the value counts of each of those values. Replace rarely occurring values together into a single value like 'Other' before encoding.
- When number of unique values is huge and even the values are equally distributed, try to find some related values and see if the multiple categorical values can be clubbed into single (grouping), thereby reducing the count of categorical values.

Related Attributes –

- If there multiple attributes with same information with different granularity, like city and state, it's better to keep columns like state and delete city column. Additionally, keeping both columns and assessing feature importance might help in eliminating one column.

Types of Categorical Data



Gender : Male, Female
Car Color: Brown, red,blue,orange,white

Railway Reservation Tickets : 1 class,2 class,3 class
Feedback on machine learning: average, good,very good,excellent
Education : Kindergarden, School, Undergraduate, bachelor, master,doctoral

Types of Data

Categorical data:

- It represents characteristics.
- Therefore it can represent things like a person's gender, language etc.

1. **Nominal Data**

Nominal values represent discrete units and are used to label variables, that have no quantitative value.

- ➢ nominal data that has no order.
- ➢ Used to "name," or label a series of values.

EX: what's your favourite movie

- o Spider man
- o Ant man
- o Iron man

Types of Data

2.Ordinal Data

Ordinal values represent discrete and ordered units. It is therefore nearly the same as nominal data, “except that it’s ordering matters”.

➡ Ordinal scales provide good information about the order of choices.

Ex1:

What’s your rating for Avengers Infinity War?

○ *

○ **

○ ***

○ ****

○ *****

- Perform the following steps to identify categorical data

- Load the data
- Describe the data set columns
- Identify the categorical data

- Example:

```
import pandas as pd
import numpy as np
df_loan=pd.read_csv("D:/Narsimlu/Courses/DataScience/datasets/loan.csv")
#display the data set
print(df_loan)
#find the type of data
obj_df = df_loan.select_dtypes(include=['object']).copy()
print(obj_df.head())
#Nominal data
print("Nominal Data")
print(obj_df["Loan Status"].value_counts())
#ordinal data
print("Ordinal Data")
print(obj_df["Term"].value_counts())
print(obj_df["Years in current job"].value_counts())
```

- Output:

```
Nominal Data
Fully Paid      77361
Charged Off     22639
Name: Loan Status, dtype: int64
```

```
Ordinal Data
Short Term     72208
Long Term      27792
Name: Term, dtype: int64
10+ years     31121
2 years        9134
3 years        8169
< 1 year       8164
5 years        6787
1 year         6460
4 years        6143
6 years        5686
7 years        5577
8 years        4582
9 years        3955
Name: Years in current job, dtype: int64
```

Data Encoding

- Many machine learning algorithms cannot operate on label data directly. They require all input variables and output variables to be numeric.

- In general, this is mostly a constraint of the efficient implementation of machine learning algorithms rather than hard limitations on the algorithms themselves.

- This means that categorical data must be converted to a numerical form.

Label Encoding

- Numerical variable will to assign a unique number to each possible outcome of the variable and replace the variables values with its corresponding number.
- Ex:

Purpose	loan_purpose_cat
Business loan	0
Buy house	1
Buy a car	2
Debt Consolidation	3
Educational Expenses	4

One hot Encoding

- it works very well unless your categorical variable takes on a large number of values.
- One hot encoding creates new (binary) columns, indicating the presence of each possible value from the original data.
- Ex:

Hyderabad
Guntur
Hyderabad
Vizag
Vizag

city	New_city_hyd	New_city_gnt	New_city_viz
Hyderabad	1	0	0
Guntur	0	1	0
Hyderabad	1	0	0
Vizag	0	0	1
Vizag	0	0	1

Data Encoding

Label Encoding

Ex: Business loan->0,Buy house->1,Buy a car->2 etc..

One-hot Encoding

Home Ownership	H_Rent	H_H_Mort	H_Own Home
Rent	1	0	0
Rent	0	1	0
Own Home	1	0	0
Home Mortgage	0	0	1
Home Mortgage	0	0	1

Categorical data

Label Encoder	One Hot Encoder
Numeric representation, ordinals	Binary representation
Loses uniqueness of values, single dimension in vector space	Individual values expressed as a different dimension in orthogonal vector space
Suitable with categorical values that are ordinal in nature, like – fog_level (low, medium, high)	Suitable with non-ordinal types of categorical attributes, like – car_type (hatchback, sedan, SUV, etc.)
Label encoded categorical attributes don't pose any further challenges	One hot encoded categorical attributes might dramatically increase the feature space (curse of dimensionality). When One hot encoding is used, it's often followed by PCA to tackle high-dimensionality

Label Encoding Example

```
print("label encoding")
print(obj_df["Purpose"].dtype)
print(obj_df["Purpose"].head(10))
obj_df["Purpose"] = obj_df["Purpose"].astype('category')
print(obj_df["Purpose"].dtype)
obj_df["loan_purpose_cat"] = obj_df["Purpose"].cat.codes
print(obj_df["loan_purpose_cat"].head(10))
```

Output

```
label encoding
object
0      Home Improvements
1     Debt Consolidation
2     Debt Consolidation
3     Debt Consolidation
4     Debt Consolidation
5     Debt Consolidation
6     Debt Consolidation
7              Buy House
8     Debt Consolidation
9     Debt Consolidation
Name: Purpose, dtype: object
```

```
category
0    5
1    3
2    3
3    3
4    3
5    3
6    3
7    1
8    3
9    3
Name: loan_purpose_cat, dtype: int8
```

One hot Encoding Example

```
import pandas as pd
import numpy as np
df_loan=pd.read_csv("D:/Narsimlu/Courses/DataScience/datasets/loan.csv")
#one hot encoding
print(df_loan["Home Ownership"].head())
df_loan["Home Ownership"] = df_loan["Home Ownership"].astype('category')
df_one_hot=pd.get_dummies(df_loan["Home Ownership"],prefix=["Home"])
print(df_one_hot.head())
#apply the data preprocessing
print(df_one_hot.isnull().sum())
```

Output

```
0     Home Mortgage
1     Home Mortgage
2          Own Home
3          Own Home
4              Rent
Name: Home Ownership, dtype: object
   ['Home']_HaveMortgage      ...         ['Home']_Rent
0                      0      ...                     0
1                      0      ...                     0
2                      0      ...                     0
3                      0      ...                     0
4                      0      ...                     1

[5 rows x 4 columns]
['Home']_HaveMortgage     0
['Home']_Home Mortgage    0
```

THANK YOU