Numeric Metamorphosis: Converting Categorical Features With Python

***I.V. Dwaraka Srihith*[1]*, A. David Donald*[2]*, T. Aditya Sai Srinivas*[3]*, G. Thippanna*[4]*, P. Vijaya Lakshmi*[5]**

[1]Student, Alliance University, Bangalore

[2], Assistant Professor, [3]Associate Professor, [4]Professor, [5]Student, Ashoka Women's Engineering College, Kurnool

****Corresponding Author***

***E-Mail Id**: - dwarakanani525@gmail.com*

ABSTRACT

In data preprocessing for machine learning, converting categorical features to numerical values is a crucial step. Python offers various techniques to achieve this transformation. One common approach is Label Encoding, where each category is assigned a unique integer. This method is suitable when there's a meaningful ordinal relationship between categories. Alternatively, One-Hot Encoding can be used to create binary columns for each category, which is ideal when there's no inherent order among them. These conversions enable machine learning algorithms to work with categorical data efficiently, making them an essential part of the data preparation process, ultimately leading to more accurate and effective predictive models.

***Keywords**: Categorical features, Numerical values, Data preprocessing, Machine learning(ML), Python.*

INTRODUCTION TO CONVERTING CATEGORICAL FEATURES TO NUMERICAL WITH PYTHON

In the realm of data preprocessing for machine learning, one of the essential tasks is converting categorical features into numerical values. Categorical features represent attributes or characteristics that don't have a natural numerical representation, such as colors, types of cars, or product categories. Machine learning algorithms typically require numerical inputs, so converting these categorical features is crucial for making the data usable in models.

Python, with its rich ecosystem of libraries, offers several methods to accomplish this conversion. In this guide, we'll explore two popular techniques for converting categorical features to numerical values: Label Encoding and One-Hot Encoding. These methods serve different purposes and are chosen based on the nature of your categorical data.

1. Label Encoding: Label encoding assigns a unique integer to each category within a categorical feature. This method is suitable when there's an inherent ordinal relationship between the categories. For example, if you're encoding "low," "medium," and "high" labels for temperature, where "high" is greater than "medium" and "medium" is greater than "low," label encoding is appropriate.
2. One-Hot Encoding: One-hot encoding creates binary columns for each category within a categorical feature. Each column represents the presence or absence of a specific category, using 1s and 0s. This method is preferred when there's no natural order among the categories, and each category is equally important. For

instance, when encoding different car makes like "Toyota," "Ford," and "Honda," one-hot encoding is the way to go.
In this guide, we'll walk through practical examples of both label encoding and one-hot encoding using Python and popular libraries like pandas and scikit-learn. By the end, you'll have a clear understanding of how to preprocess your categorical data, making it suitable for various machine learning algorithms. Whether you're working on classification, regression, or any other machine learning task, this skill will be invaluable in your data preprocessing toolbox.

RELATED WORK

The related work in the field of converting categorical features to numerical values encompasses a range of online resources and tutorials that provide insights and practical guidance for data preprocessing in Python. Several notable sources include:
1. ProjectPro.io (https://www.projectpro.io/recipes/convert-categorical-variables-into-numerical-variables-in-python): This resource offers practical recipes and examples for converting categorical variables into numerical ones using Python, making it a valuable reference for data analysts and machine learning practitioners.
2. GeeksforGeeks (https://www.geeksforgeeks.org/how-to-convert-categorical-string-data-into-numeric-in-python/): GeeksforGeeks provides a detailed tutorial on converting categorical string data into numeric format in Python, catering to both beginners and experienced programmers seeking clarity on the topic.
3. The Clever Programmer (https://thecleverprogrammer.com/2020/11/22/convert-categorical-features-to-numerical-with-python/): This blog post delves into the conversion of categorical features to numerical ones in Python, offering practical insights and code examples to assist data scientists and analysts.
4. PB Python (https://pbpython.com/categorical-encoding.html): PB Python explores various categorical encoding techniques, including one-hot encoding and label encoding, providing a comprehensive understanding of their applications and trade-offs in data preprocessing.
5. CatBoost Documentation (https://catboost.ai/docs/concepts/algorithm-main-stages_cat-to-numberic): CatBoost's official documentation discusses its approach to handling categorical features and converting them into numerical representations, making it valuable for users of the CatBoost machine learning library.
6. Analytics Vidhya (https://www.analyticsvidhya.com/blog/2015/11/easy-methods-deal-categorical-variables-predictive-modeling/): Analytics Vidhya's blog post explores practical methods for dealing with categorical variables in predictive modeling, providing insights into encoding strategies and their impact on model performance.
These resources collectively contribute to the body of knowledge surrounding the conversion of categorical features into numerical values, offering a variety of perspectives, techniques, and code examples to support researchers, data scientists, and analysts in their data preprocessing endeavors.

WHAT ARE CATEGORICAL FEATURES?

Categorical features, also known as categorical variables or qualitative variables, are a type of data variable used in statistics and machine learning to represent non-numeric data. These features represent categories or groups and can take on values from a limited, fixed set of distinct categories. Categorical features are often used to describe attributes or characteristics that are not inherently numerical.

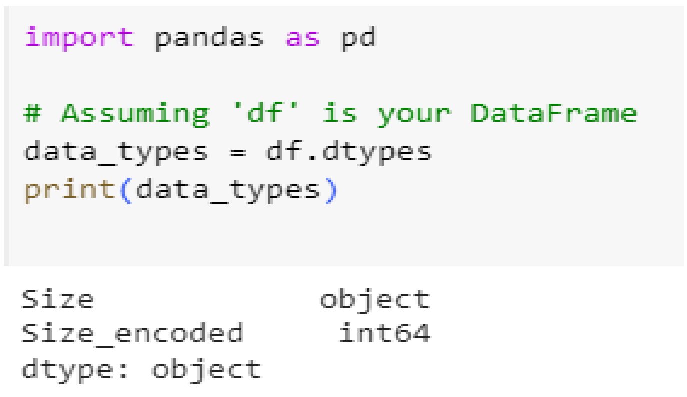
There are two primary types of categorical features:
1. Nominal Categorical Features: Nominal features represent categories with no inherent order or ranking between them. These categories are purely distinct, and you can't perform mathematical operations like addition or subtraction on them. Examples of nominal categorical features include:
- Colors (e.g., red, blue, green)
- Animal types (e.g., cat, dog, bird)
- Countries (e.g., USA, Canada, Australia)

2. Ordinal Categorical Features: Ordinal features represent categories with a specific order or ranking between them. Unlike nominal features, ordinal features have a meaningful sequence or hierarchy. You can perform basic mathematical operations like comparison (greater than, less than) on ordinal data. Examples of ordinal categorical features include:
- Education levels (e.g., high school diploma, bachelor's degree, master's degree)
- Customer satisfaction ratings (e.g., very dissatisfied, dissatisfied, neutral, satisfied, very satisfied)
- Income levels (e.g., low income, medium income, high income)

Categorical features are common in real-world datasets, and handling them properly is crucial for machine learning tasks. Depending on the machine learning algorithm you plan to use, you may need to convert categorical features into a numerical format. This is typically done through techniques like label encoding or one-hot encoding, as mentioned in the previous response.
Properly encoding categorical features is important because most machine learning algorithms require numerical input data. Misinterpreting categorical data as numerical can lead to incorrect results, so it's essential to use appropriate encoding methods based on the nature of the categorical feature.

1. Identifying categorical features

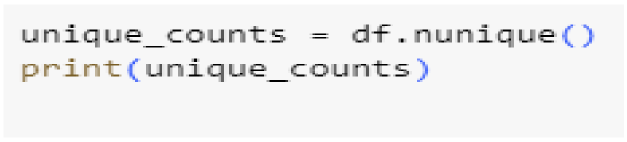
Identifying categorical features in a dataset is an important step in data preprocessing for machine learning and statistical analysis. Distinguishing between categorical and numerical features helps you apply appropriate data processing techniques. Here are several methods to identify categorical features in a dataset:
Data Inspection: Begin by visually inspecting your dataset. Look at the data and column headers. If you see columns that contain text, labels, or discrete categories rather than continuous numbers, they are likely to be categorical.
Data Types: Check the data types of each column in your dataset. In Python's pandas library, you can use the dtypes attribute to see the data types of all columns:



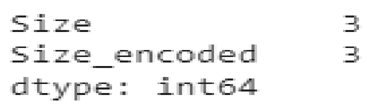
```
import pandas as pd

# Assuming 'df' is your DataFrame
data_types = df.dtypes
print(data_types)
```

```
Size              object
Size_encoded       int64
dtype: object
```

Columns with data types like object, str, category, or bool are often indicative of categorical features.
Unique Value Count: Calculate the number of unique values in each column
.



```
unique_counts = df.nunique()
print(unique_counts)
```

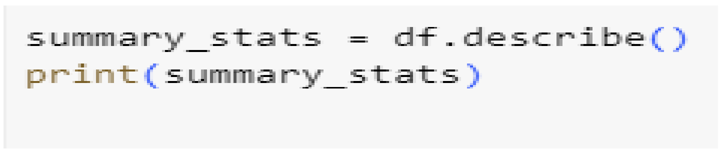


```
Size            3
Size_encoded    3
dtype: int64
```

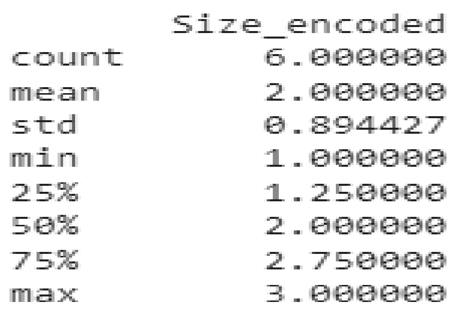
Categorical features typically have a limited number of unique values, while numerical features have a wider range. You can use the nunique() method in pandas to count unique values:

Columns with a small number of unique values are more likely to be categorical.
Domain Knowledge: Sometimes, your domain knowledge about the data can be invaluable. You may already know which columns represent categorical information based on your understanding of the dataset.

Statistical Summary: Use descriptive statistics to summarize the data in each column. Tools like describe() in pandas can help you see if the values are discrete and whether they have a natural order. Numerical features typically have a wider range and can have a broader distribution, whereas categorical features have a limited set of values.



```
summary_stats = df.describe()
print(summary_stats)
```



```
       Size_encoded
count      6.000000
mean       2.000000
std        0.894427
min        1.000000
25%        1.250000
50%        2.000000
75%        2.750000
max        3.000000
```

Visualization: Create visualizations of your data to get a better sense of the nature of each feature. For example, bar plots or histograms can help visualize categorical and numerical features differently.

Once you've identified your categorical features, you can apply appropriate encoding techniques like one-hot encoding or label encoding if necessary, depending on the machine learning algorithm you plan to use. Properly distinguishing between categorical and numerical features is essential for building accurate predictive models.

Transforming Categorical Data into Numerical Values Using Python

Converting categorical features to numeric values is a crucial preprocessing step in machine learning. The method you choose depends on the nature of your data. Two common techniques are Label Encoding and One-Hot Encoding.

Label Encoding:
Label Encoding is suitable when there is an ordinal relationship between the categories, meaning that one category is "greater" or "lesser" than another.

```python
import pandas as pd
from sklearn.preprocessing import LabelEncoder

# Sample data with a categorical feature
data = {'Category': ['Low', 'Medium', 'High', 'Low', 'Medium']}
df = pd.DataFrame(data)

# Initialize the LabelEncoder
label_encoder = LabelEncoder()

# Fit and transform the categorical feature
df['Category_encoded'] = label_encoder.fit_transform(df['Category'])

# Display the result
print(df)
```

```
  Category  Category_encoded
0      Low                 1
1   Medium                 2
2     High                 0
3      Low                 1
4   Medium                 2
```

In this example, the "Low" category is assigned 0, "Medium" is assigned 1, and "High" is assigned 2.

One-Hot Encoding: One-Hot Encoding is suitable when there is no inherent order between the categories, and each category is independent..

```python
import pandas as pd

# Sample data with a categorical feature
data = {'Category': ['A', 'B', 'A', 'C', 'B']}
df = pd.DataFrame(data)

# Perform one-hot encoding using pandas' get_dummies function
df_encoded = pd.get_dummies(df, columns=['Category'], prefix=['Category'])

# Display the result
print(df_encoded)
```

```
   Category_A  Category_B  Category_C
0           1           0           0
1           0           1           0
2           1           0           0
3           0           0           1
4           0           1           0
```

This will create binary columns for each category, where a 1 indicates the presence of that category and 0 indicates its absence.
Choose the encoding method based on the characteristics of your data and the requirements of your machine learning model. Label encoding can be useful when there's an ordinal relationship, but be cautious, as some algorithms may misinterpret the encoded values as having a meaningful order. One-hot encoding is safer when there's no such order, but it can lead to a higher-dimensional dataset..

CONCLUSION

Converting categorical features to numeric values is a fundamental data preprocessing step in machine learning. Properly handling these features is essential for building accurate predictive models. Label encoding is suitable when categorical data has an inherent order, facilitating the representation of ordinal relationships.

On the other hand, one-hot encoding is ideal for nominal categorical data without a natural order, creating binary columns to represent each category independently. The choice between these techniques should consider the nature of the data and the specific requirements of the machine learning algorithm being employed.

Understanding and appropriately encoding categorical features contribute significantly to the success of predictive modeling, ensuring that valuable information from non-numeric attributes is effectively incorporated into the analysis.

REFERENCES

1. https://www.projectpro.io/recipes/convert-categorical-variables-into-numerical-variables-in-python
2. https://www.geeksforgeeks.org/how-to-convert-categorical-string-data-into-numeric-in-python/
3. https://thecleverprogrammer.com/2020/11/22/convert-categorical-features-to-numerical-with-python/
4. https://pbpython.com/categorical-encoding.html
5. https://catboost.ai/docs/concepts/algorithm-main-stages_cat-to-numberic
6. https://www.analyticsvidhya.com/blog/2015/11/easy-methods-deal-categorical-variables-predictive-modeling/