



Effective Categorical Variable Encoding for Machine Learning



Filipe Filardi · [Follow](#)

Published in Towards AI · 5 min read · Jan 8, 2023





Image by [DCStudio](#) on [Freepik](#)

A categorical variable is a common type of data found in many machine learning datasets. Effective handling of categorical variables can be crucial for building successful models since it contains rich information that can be used to predict outcomes in Machine Learning.

However, working with categorical variables can be challenging, as many models are designed to handle numerical data. As a result, some people may

need clarification about correctly processing categorical data, leading to confusion and potentially suboptimal model performance.

This article aims to provide a clear and comprehensive overview of the most popular approaches to handling categorical data in Machine Learning. By understanding the different options available and their implications, I hope to provide readers the knowledge and tools they need to handle categorical data in their Machine Learning projects.

Categorical Data in Machine Learning

Categorical data consists of data that can be classified into categories. In machine learning, it is common to encounter categorical data from variables such as *gender*, *race*, *nationality*, *genre*, or *occupation*. Categorical data is often present in real-world datasets, and it is vital to handle it properly.

One of the main challenges of working with categorical data is that most machine learning algorithms are designed to work with numerical data. This means that categorical data must be transformed into a numerical format to be used as input to the model.

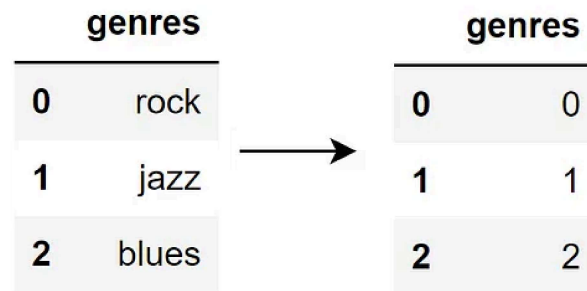
Dealing with categorical data

This section will explore some popular methods for dealing with categorical data in machine learning.

What is “Replacing for Numbers”?

Replacing for numbers refers to the process of replacing a categorical variable with a numerical value.

For example, continuing with the example above, if we replaced the categorical variable with numerical values, we would get the following:



The diagram illustrates the process of replacing categorical data with numerical values. It consists of two tables, both titled 'genres', connected by a right-pointing arrow. The first table on the left has two columns: an index column with values 0, 1, and 2, and a genre column with values 'rock', 'jazz', and 'blues'. The second table on the right has two columns: an index column with values 0, 1, and 2, and a numerical column with values 0, 1, and 2. This shows that 'rock' is replaced by 0, 'jazz' by 1, and 'blues' by 2.

genres	
0	rock
1	jazz
2	blues

genres	
0	0
1	1
2	2

Example of Replacing | Image by Author

Here's the python code using replace in a Pandas data frame as a reference:

```
df.replace({'rock': 0, 'jazz': 1, 'blues': 2})
```

What is a “Label Encoder”?

Label Encoder is another method for encoding categorical variables. It assigns a unique numerical value to each category in the categorical variable.

Using Label Encoder on the previous example would result in the same values as replacing. While replace might be a suitable approach for a small number of categories, it can become impractical when dealing with many categories.

genres			
0	rock	0	0
1	jazz	1	1
2	blues	2	2

Example of Label Encoder | Image by Author

Here's the Python code using the Label Encoder:

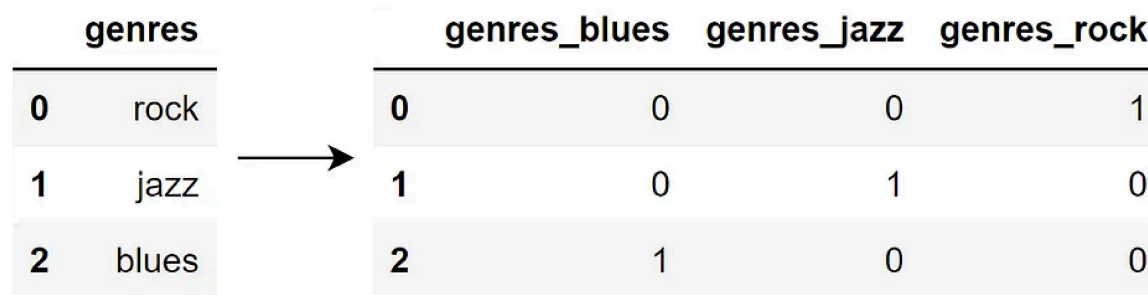
```
from sklearn import preprocessing

le = preprocessing.LabelEncoder()
le.fit(df['genres'])

df['genres'] = le.transform(df['genres'])
```

What is converting to a “dummy variable“?

It is the process of creating a new binary column for each category in a categorical variable, with a 0 or 1 indicating the presence or absence of that category, such as:



genres			genres_blues	genres_jazz	genres_rock
0	rock	→	0	0	1
1	jazz		0	1	0
2	blues		1	0	0

Example of Dummy | Image by Author

There are two ways of doing that. The first is using `get_dummies()` of Pandas library:

```
import pandas as pd

X_encoded = pd.get_dummies(df, columns=['genres'])
```

The other is using OneHotEncoder() of Scikit-learn (sklearn):

```
from sklearn.preprocessing import OneHotEncoder

enc = OneHotEncoder()
enc.fit(df)

X_encoded = enc.transform(df).toarray()
```

Dummifying and One Hot Encoding are essentially the same things. The main difference is that “dummify” is a more colloquial term, and “One Hot encoding” is the technical term used in the machine learning literature.

Why are Dummies Preferred Over the other solutions?

There are several reasons why Dummies are generally preferred over other encoding methods:

Avoiding implied ordinal relationships and preventing bias

Dummies create separate columns for each category, allowing the model to learn the relationships between the individual categories and the target variable. Replacing for numbers and label encoder, on the other hand, imply

an ordinal relationship between the categories and does not create separate columns for each category, which can lead to misleading results if the categories do not have an inherent order.

For example, suppose you replace “rock” with 1, “jazz” with 2, and “blues” with 3 in your dataset. In that case, your model may assume that “jazz” is twice as important as “rock” and “blues” is three times as important as “rock”. This can introduce bias into the model, as it makes assumptions about the order in which you assign the numbers.

Dummies allow the model to learn more complex relationships

Because it creates separate columns for each category, the model can learn more complex relationships between the categories and the target variable.

On the other hand, the other mentioned encoders only allow the model to learn the overall relationship between the numerical value and the target variable, which may not capture the full complexity of the data.

When to Avoid Dummies

There are certain situations in which Dummies may not be the best approach. Here are the most important ones:

- **High cardinality:** One Hot Encoding creates a separate column for each category in the categorical variable. This can lead to a high number of columns, especially if the categorical variable has many unique values. In such cases, One Hot Encoding may result in a sparse and unwieldy data set, which can be challenging to work with.
- **Memory constraints:** One Hot Encoding can also be problematic if the data set is large and requires a lot of memory to store. The resulting data set can take up a lot of space, which may not be feasible if memory is limited.
- **Multicollinearity:** Occurs when there is a high correlation between the dummy variables, which can cause the coefficients in the model to be unstable and difficult to interpret. Dummy variables are naturally correlated because they are created from the same categorical variable.

In these situations, alternative encoding methods, such as label encoder or target encoding, may be more appropriate, which can handle high cardinality more efficiently.

If you are interested in learning more about **multicollinearity** and **target encoding**, there are many other resources available. You might want to check out the following articles:

Beware of the Dummy variable trap in pandas

Important caveats to be kept in mind when encoding data with `pandas.get_dummies()`

towardsdatascience.com

This article discusses the issue of multicollinearity in detail and provides tips on how to deal with it going further into the parameters of `OneHotEncoder()` and `to_dummy()` functions.

Target-encoding Categorical Variables

One nice alternative to One-hot encoding your categories

towardsdatascience.com

This article comprehensively analyzes **target encoding** to solve the dimensionality problem.

I hope this article has helped you to build confidence when deciding how to handle categorical variables in your dataset and when to consider getting out of your one-hot encoding comfort zone.

It is essential to carefully consider the data's characteristics and the model's requirements when deciding which encoding method to use. The two articles referenced in this post are excellent references. Check them out!

If you're interested in reading other articles written by me. Check out my [repo](#) with all articles I've written so far, separated by categories.

Thanks for reading

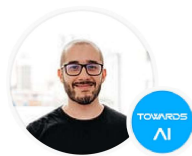
Machine Learning

Categorical Data

Python

Guides And Tutorials

Data Science



Written by Filipe Filardi

117 Followers · Writer for Towards AI



Data Scientist with a passion for making Development and Data Science more accessible

More from Filipe Filardi and Towards AI



 Filipe Filardi in Level Up Coding

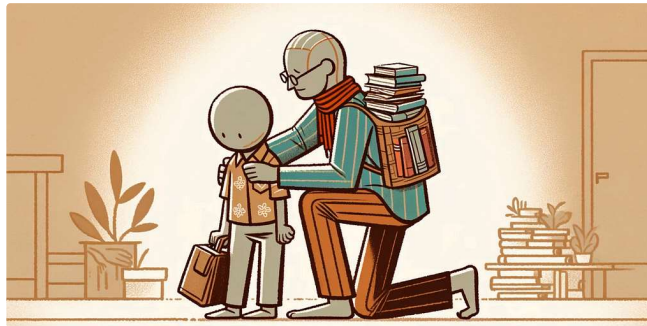
7 reasons you struggle to learn to code and how to fix them

There are common reasons why people struggle to learn how to code. Don't let them...

4 min read · Jan 17, 2023

 265

 4



 Ignacio de Gregorio in Towards AI

RAG 2.0, Finally Getting RAG Right!

The Creators of RAG Present its Successor

★ · 9 min read · 6 days ago

 1.1K

 9



 Boris Meinardus in Towards AI

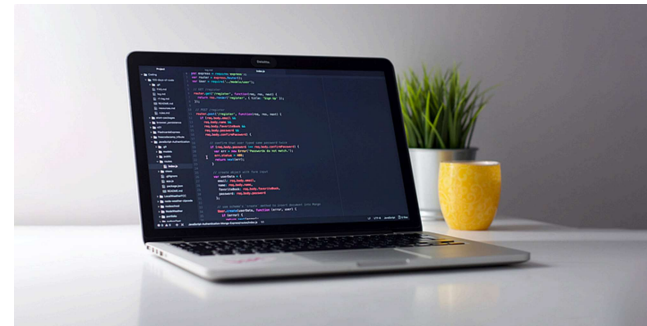
Machine Learning Was Hard Until I Learned These 5 Secrets!

The secrets no one tells you but make learning ML a lot easier and enjoyable.

★ · 10 min read · Mar 28, 2024

 1.7K

 16



 Filipe Filardi

Strings in Python: From Basics to Advanced Techniques

In this article, we cover how to create and manipulate strings, special characters, built-...

10 min read · Jan 2, 2023

 164

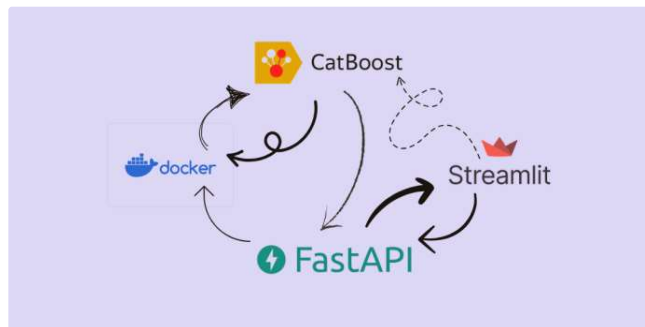
 2



See all from Filipe Filardi

See all from Towards AI

Recommended from Medium

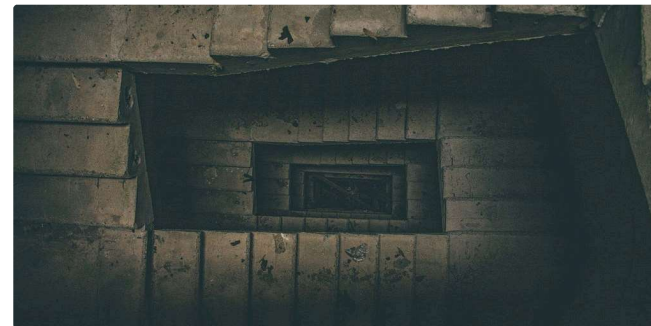


Ramazan Olmez

End-to-End Machine Learning Project: Churn Prediction

The main objective of this article is to develop an end-to-end machine learning project. For...

18 min read · Feb 22, 2024



Niranjan Appaji

A Guide to Handling High Cardinality in Categorical Variables

High cardinality refers to a situation in a dataset where a particular feature has a larg...

5 min read · Dec 28, 2023

 32 

  5 



Lists



Predictive Modeling w/ Python

20 stories · 1103 saves



Practical Guides to Machine Learning

10 stories · 1317 saves



Coding & Development

11 stories · 564 saves



Natural Language Processing

1377 stories · 870 saves



 Sze Zhong LIM in Data And Beyond

Mastering Exploratory Data Analysis (EDA): Everything You...

A systematic approach to EDA your data and prep it for machine learning.



 Subha

Handling missing values in dataset —7 methods that you need to know

While working with data it is a common scenario for the data scientists to deal with...

18 min read · Apr 6, 2024

 326  3



 Dr. Ernesto Lee 

Advanced Stock Pattern Prediction using LSTM with the Attention...

Introduction

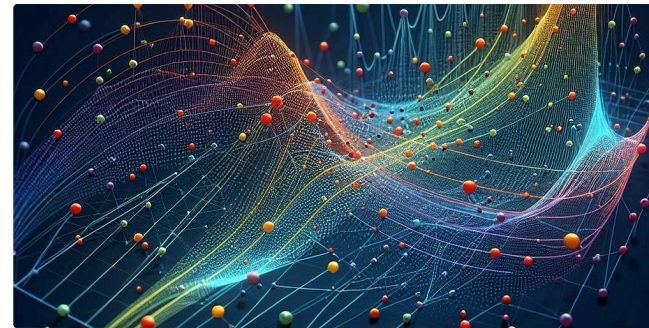
15 min read · Apr 8, 2024

 406  7



9 min read · Feb 13, 2024

 114 



 Tim Sumner in Towards Data Science

A New Coefficient of Correlation

What if you were told there exists a new way to measure the relationship between two...

10 min read · Mar 31, 2024

 2.5K  35



See more recommendations

