

How to Handle Missing Values of Categorical Variables?



CHIRAG GOYAL – April 27, 2021

[Beginner](#) [Data Cleaning](#) [Data Exploration](#) [Python](#) [Technique](#)

This article was published as a part of the [Data Science Blogathon](#).

Introduction

“Data is the fuel for Machine Learning algorithms”.

Real-world data collection has its own set of problems, It is often very messy which includes **missing data, presence of outliers, unstructured manner**, etc. Before looking for any insights from the data, we have to first perform preprocessing tasks which then only allow us to use that data for further observation and train our machine learning model.

Missing value in a dataset is a very common phenomenon in the reality. In this blog, you will see how to handle missing values for categorical variables while we are performing data preprocessing. Missing value correction is required to reduce bias and to produce powerful suitable models. Most of the algorithms can't handle missing data, thus you need to act in some way to simply not let your code crash. So, let's begin with the methods to solve the problem.

Methods for dealing with missing values

Example 1, Let's have a **dummy dataset** in which there are three independent features(predictors) and one dependent feature(response).

Feature-1	Feature-2	Feature-3	Output
Male	23	24	Yes
- - - -	24	25	No
Female	25	26	Yes
Male	26	27	Yes

Here, We have a missing value in row-2 for Feature-1.

The popular methods which are used by the machine learning community to handle the missing value for categorical variables in the dataset are as follows:

1. Delete the observations: If there is a large number of observations in the dataset, where all the classes to be predicted are sufficiently represented in the training data, then try deleting the missing value observations, which would not bring significant change in your feed to your model.

For Example,1, Implement this method in a given dataset, we can delete the entire row which contains missing values(delete row-2).

2. Replace missing values with the most frequent value: You can always impute them based on **Mode** in the case of categorical variables, just make sure you don't have highly skewed class distributions.

NOTE: But in some cases, this strategy can make the data imbalanced wrt classes if there are a huge number of missing values present in our dataset.

– Generally, replacing the missing values with the mean/median/mode is a crude way of treating missing values. Depending on the context, like if the variation is low or if the variable has low leverage over the response, such a rough approximation is acceptable and could give satisfactory results. In this case, since you are saying it is a categorical variable — this step may not be applicable.

For Example, 1, To implement this method, we replace the missing value by the most frequent value for that particular column, here we replace the missing value by Male since the count of Male is more than Female (Male=2 and Female=1).

3. Develop a model to predict missing values: One smart way of doing this could be training a classifier over your columns with missing values as a dependent variable against other features of your data set and trying to impute based on the newly trained classifier.

Here's the algorithm that you can follow:

- Divide the data into two parts. One part will have the present values of the column including the original output column, the other part will have the rows with the missing values.
- Divide the 1st part (present values) into cross-validation set for model selection.
- Train your models and test their metrics against the cross-validated data. You can also perform a grid search or randomized search for the best results.
- Finally, with the model, predict the unknown values which are missing in our problem.

NOTE: Since you are trying to impute missing values, things will be nicer this way as they are not biased and you get the best predictions out of the best model.

For Example, 1, To implement the given strategy, firstly we will consider Feature-2, Feature-3, and Output column for our new classifier means these 3 columns are used as independent features for our new classifier and the Feature-1 considered as a target outcome and note that here we consider only non-missing rows as our train data and observations which is having missing value will become our test data. We have to do the prediction using our model on the test data and after predictions, we have the dataset which is having no missing value.

4. Deleting the variable: If there are an exceptionally larger set of missing values, try excluding the variable itself for further modeling, but you need to make sure that it is not much significant for predicting the target variable i.e, Correlation between dropped variable and target variable is very low or redundant.

For Example, 1, To implement this strategy to handle the missing values, we have to drop the complete column which contains missing values, so for a given dataset we drop the Feature-1 completely and we use only left features to predict

our target variable.

5. Apply unsupervised Machine learning techniques: In this approach, we use unsupervised techniques like **K-Means**, **Hierarchical clustering**, etc. The idea is that you can skip those columns which are having missing values and consider all other columns except the target column and try to create as many clusters as no of independent features(after drop missing value columns), finally find the category in which the missing row falls.

For Example, 1, To implement this strategy, we drop the Feature-1 column and then use Feature-2 and Feature-3 as our features for the new classifier and then finally after cluster formation, try to observe in which cluster the missing record is falling in and we are ready with our final dataset for further analysis.

Implementation in Python

Import necessary dependencies.

```
In [23]: # Import Necessary Dependencies
import numpy as np
import pandas as pd
```

Load and Read the Dataset.

```
In [25]: # Read and Load the Dataset
df=pd.read_csv('toy_dataset.csv')
df.head()

Out[25]:
```

	Feature-1	Feature-2	Feature-3	Feature-4
0	Male	23	24	Yes
1	NaN	24	25	No
2	Female	25	26	Yes
3	Male	26	27	Yes

Find the number of missing values per column.

```
In [26]: # find no of missing values per column
df.isnull().sum()

Out[26]: Feature-1    1
Feature-2    0
Feature-3    0
Feature-4    0
dtype: int64
```

Activate Window
Go to Settings to activate

Apply Strategy-1(Delete the missing observations).

```
In [27]: # Delete the missing observation
df.dropna(axis=0,inplace=True)
df.head()

Out[27]:
```

	Feature-1	Feature-2	Feature-3	Feature-4
0	Male	23	24	Yes
2	Female	25	26	Yes
3	Male	26	27	Yes

Apply Strategy-2(Replace missing values with the most frequent value).

```
In [33]: # Fill missing value with the most frequent value of that column
df = df.fillna(df.mode().iloc[0])
df.head()
```

```
Out[33]:
```

	Feature-1	Feature-2	Feature-3	Feature-4
0	Male	23	24	Yes
1	Male	24	25	No
2	Female	25	26	Yes
3	Male	26	27	Yes

Apply Strategy-3(Delete the variable which is having missing values).

```
In [39]: # Delete the complete column which is having missing values
df.dropna(axis=1,inplace=True)
df.head()

Out[39]:
```

	Feature-2	Feature-3	Feature-4
0	23	24	Yes
1	24	25	No
2	25	26	Yes
3	26	27	Yes

Apply Strategy-4(Develop a model to predict missing values).

For this strategy, we firstly encoded our Independent Categorical Columns using “One Hot Encoder” and Dependent Categorical Columns using “Label Encoder”.

– Read and Load the Encoded Dataset.

– Make missing records as our Testing data.

– Make non-missing records as our Training data.

– Separate Dependent and Independent variables.

```
In [39]: # Make independent and dependent variables from train subset
X_train=train.iloc[:,1:4]
y_train=train.iloc[:,0]
```

– Fit our Logistic Regression model.

```
In [45]: # Fit our Logistic regression model
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(X_train,y_train)

Out[45]: LogisticRegression()
```

– Predict the class for missing records.

```
In [46]: # predict the class for missing record
input=[[24,25,0]]
lr.predict(input)

Out[46]: array([1.])
```

Activate Window
Go to Settings to activate

This completes our implementation part!

End Notes

Thanks for reading!

This article introduces you to different ways to tackle the problem of having missing values for categorical variables.

If you liked this and want to know more, go visit my other articles on Data Science and Machine Learning by clicking on the [Link](#)

Please feel free to contact me on [Linkedin](#), [Email](#).

Something not mentioned or want to share your thoughts? Feel free to comment below And I'll get back to you.

Till then Stay Home, Stay Safe to prevent the spread of **COVID-19**, and Keep Learning!

About the Author

Chirag Goyal

Currently, I pursuing my Bachelor of Technology (B.Tech) in Computer Science and Engineering from **the Indian Institute of Technology Jodhpur(IITJ)**. I am very enthusiastic about Machine learning, Deep Learning, and Artificial Intelligence.

The media shown in this article are not owned by Analytics Vidhya and is used at the Author's discretion.

[Complete Guide to Feature Engineering: Zero to Hero](#)

[Exploratory Data Analysis\(EDA\) in Python!](#)

[Introduction to Feature Engineering - Everything You Need to Know!](#)

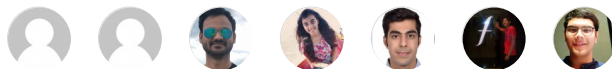
[blogathon](#) [impute missing values](#)

About the Author



[CHIRAG GOYAL](#)

Our Top Authors



Download

Analytics Vidhya App for the Latest blog/Article



Next Post

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name*

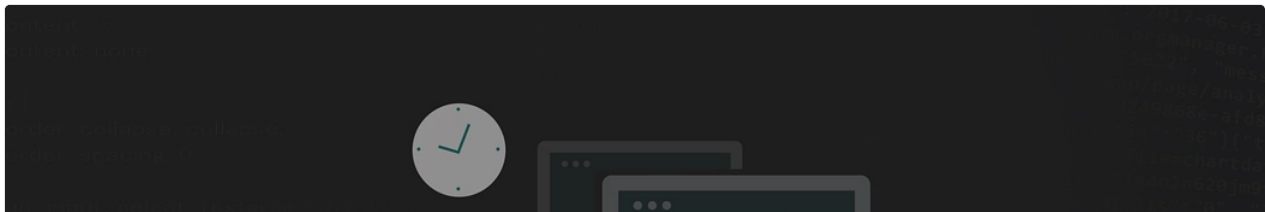
Email*

Website

☐ Notify me of follow-up comments by email. ☐ Notify me of new posts by email.

Submit

Top Resources



[Python Tutorial: Working with CSV file for Data Science](#)

Harika Bonthu - AUG 21, 2021