


Impute missing data values in Python – 3 Easy Ways!

By Safa Mulani / October 7, 2020



Hello, folks! In this article, we will be focusing on **3 important techniques to Impute missing data values** in Python.

So, let us begin.

Why

Before

So, a e,
maybe

Having a missing value in a machine learning model is considered very inefficient and hazardous because of the following reasons:

- Reduces the efficiency** of the ML model.

- Affects the overall distribution** of data values.

- It leads to a **biased effect** in the estimation of the ML model.

This is when imputation comes into picture.

By imputation, we mean to replace the missing or null values with a particular value in the entire dataset.

Impute by mean

Impute by median

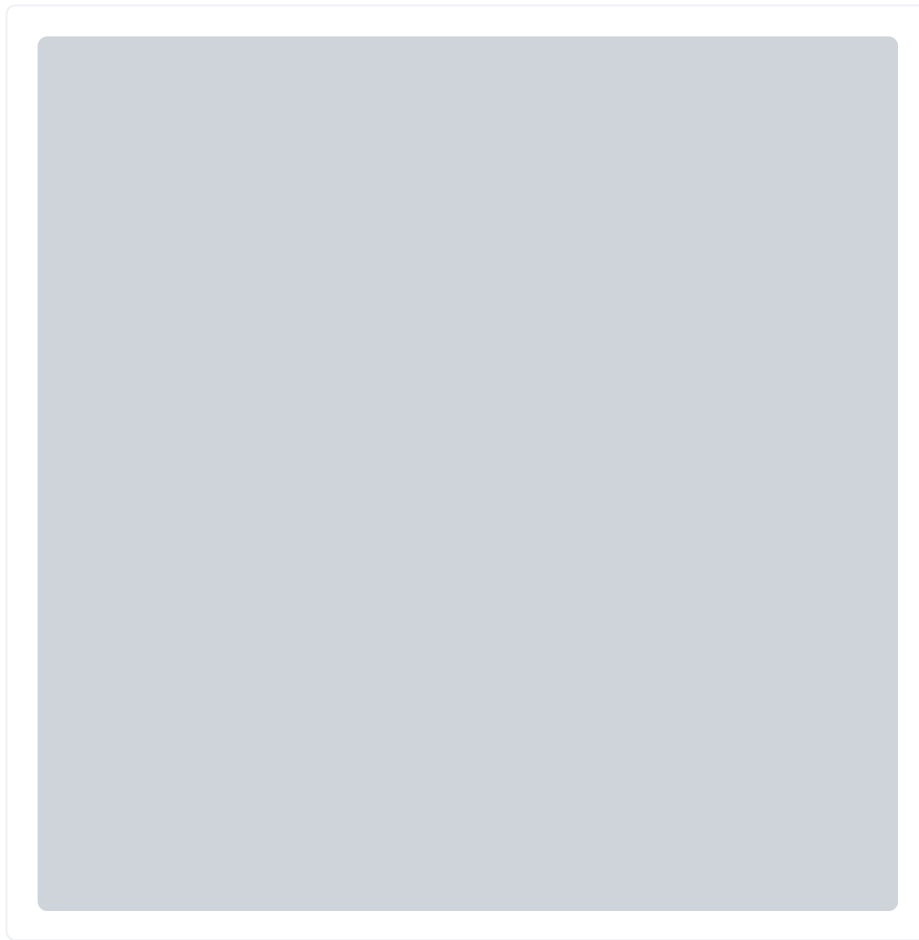
Knn Imputation

Let us now understand and implement each of the techniques in the upcoming section.

1. Impute missing data values by MEAN

The missing values can be imputed with the mean of that particular feature/data variable. That is, the null or missing values can be replaced by the mean of the data values of that particular data column or dataset.

Let us have a look at the below dataset which we will be using throughout the article.



Dataset For Imputation

As clearly seen, the above dataset contains NULL values. Let us now try to impute them with the mean of the feature.

Import the required libraries

Here, at first, let us load the necessary datasets into the working environment.

```
#Load libraries
import os
import pandas as pd
import numpy as np
```

We have used [pandas.read_csv\(\) function](#) to load the dataset into the environment.

```
marketing_train = pd.read_csv("C:/marketing_tr.csv")
```

Verify missing values in the database

Before we imputing missing data values, it is necessary to check and detect the presence of missing values using `isnull()` function as shown below–

```
marketing_train.isnull().sum()
```

After executing the above line of code, we get the following count of missing values as output:

```
custAge      1804
profession    0
marital       0
responded     0
dtype: int64
```

As clearly seen, the data variable 'custAge' contains 1804 missing values out of 7414 records.

Use the mean() method on all the null values

Further, we have used `mean()` function to impute all the null values with the mean of the column 'custAge'.

```
missing_col = ['custAge']
#Technique 1: Using mean to impute the missing values
```

```
for i in missing_col:
    marketing_train.loc[marketing_train.loc[:,i].isnull(),i]=marketing_train.
```

Verify the changes

After performing the imputation with mean, let us check whether all the values have been imputed or not.

```
marketing_train.isnull().sum()
```

As seen below, all the missing values have been imputed and thus, we see no more missing values present.

```
custAge      0
profession   0
marital      0
responded    0
dtype: int64
```

2. Imputation with median

In this technique, we impute the missing values with the median of the data values
or the data set

Example:

```
#Load libraries
import os
import pandas as pd
import numpy as np

marketing_train = pd.read_csv("C:/marketing_tr.csv")
print("count of NULL values before imputation\n")
marketing_train.isnull().sum()

missing_col = ['custAge']

#Technique 2: Using median to impute the missing values
for i in missing_col:
    marketing_train.loc[marketing_train.loc[:,i].isnull(),i]=marketing_train.

print("count of NULL values after imputation\n")
marketing_train.isnull().sum()
```

Here, we have imputed the missing values with median using `median()` function.

Output:

```
responded      0
dtype: int64
count of NULL values after imputation
custAge        0
profession      0
marital         0
responded      0
dtype: int64
```

3. KNN Imputation

In this technique, the missing values get imputed based on the KNN algorithm i.e. **K-nearest-neighbour algorithm**.

In this algorithm, the missing values get replaced by the nearest neighbor estimated values.

Let us understand the implementation using the below example:

KNN Imputation:

```
#Load libraries
import os
import pandas as pd
import numpy as np
```

```
print("count of NULL values before imputation\n")
marketing_train.isnull().sum()
```

Here, is the count of missing values:

```
count of NULL values before imputation
custAge      1804
profession    0
marital       0
responded     0
dtype: int64
```

In the below piece of code, we have converted the data types of the data variables to object type with categorical codes assigned to them.

```
lis = []
for i in range(0, marketing_train.shape[1]):

    if(marketing_train.iloc[:,i].dtypes == 'object'):
        marketing_train.iloc[:,i] = pd.Categorical(marketing_train.iloc[:,i],
        #print(marketing_train[[i]])
        marketing_train.iloc[:,i] = marketing_train.iloc[:,i].cat.codes
        marketing_train.iloc[:,i] = marketing_train.iloc[:,i].astype('object')

    lis.append(marketing_train.columns[i])
```

The `KNN()` function is used to impute the missing values with the nearest neighbour possible.

```
#Apply KNN imputation algorithm
marketing_train = pd.DataFrame(KNN(k = 3).fit_transform(marketing_train),
```

Output of imputation:

```
Imputing row 1/7414 with 0 missing, elapsed time: 13.293
Imputing row 101/7414 with 1 missing, elapsed time: 13.311
Imputing row 201/7414 with 0 missing, elapsed time: 13.319
Imputing row 301/7414 with 0 missing, elapsed time: 13.319
Imputing row 401/7414 with 0 missing, elapsed time: 13.329
.
.
.
.
.
Imputing row 7101/7414 with 1 missing, elapsed time: 13.610
Imputing row 7201/7414 with 0 missing, elapsed time: 13.610
Imputing row 7301/7414 with 0 missing, elapsed time: 13.618
Imputing row 7401/7414 with 0 missing, elapsed time: 13.618
```

Output:

```
count of NULL values before imputation
custAge          0
profession       0
marital          0
responded        0
dtype: int64
```

Conclusion

By this, we have come to the end of this topic. In this article, we have implemented 3 different techniques of imputation.

Feel free to comment below, in case you come across any question.

For more such posts related to Python, Stay tuned @ [Python with AskPython](#) and Keep Learning!

[← Previous Post](#)

[Next Post →](#)

Search ...



[How to Write a Styler to a file, buffer or string in LaTeX?](#)

[Appending Dataframes in Pandas with For Loops](#)

[Vanishing Gradient Problem With Solution](#)

[Converting String to Numpy Datetime64 in a Dataframe](#)

[How to Change Datetime Format in Pandas](#)

[Activating a Virtual Environment in Windows 10 Command Prompt](#)

[How to Convert a Datetime to Date](#)

[Determine if Two Lists Have Same Elements, Regardless of Order](#)

[Adding Tuples to Lists in Python](#)

[How to Get Week Numbers in Python](#)

Favorite Sites

[GoLang Tutorials](#)

[VM-Help](#)

[Linux Tutorials](#)

[MySQL Tutorials](#)

[CodeForGeek](#)

[Mkyong](#)

Copyright © 2023 AskPython · All Rights Reserved

[Privacy Policy](#) · [Terms and Conditions](#) · [Contact](#) · [About](#) · [Team](#)

AskPython is part of JournalDev IT Services Private Limited
