

机器学习中缺失值处理方法大全（附代码）



21 人赞同了该文章

欢迎关注 @机器学习社区，专注学术论文、机器学习、人工智能、Python技巧

今天我们来看看数据预处理中一个有趣的问题：如何处理缺失值。在我们探讨问题之前，我们一起回顾一些基本术语，帮助我们了解为什么需要关注缺失值。**本文内容较长，建议收藏后学习，喜欢点赞支持一下。**

数据清洗简介

数据预处理中的数据清洗与机器学习方法、深度学习架构或数据科学领域的任何其他复杂方法无关。我们有数据收集、数据预处理、建模（机器学习、计算机视觉、深度学习或任何其他复杂方法）、评估，以及最后的模型部署等等。因此数据处理建模技术是一个非常热门话题，但数据预处理有很多工作等着我们去完成。

在数据分析与挖掘过程中，会熟悉这个比例：**60:40**，这意味着 60% 的工作与数据预处理有关，有时这个比例会高至80%以上。

在这篇文章中，我们将一起学习数据预处理模块中的数据清洗。即从数据集中纠正或消除不准确、损坏、格式错误、重复或不完整的数据的做法称为数据清理。

技术交流群

建了机器学习交流群！想要交流群的同学，可以直接加微信号：**mlc2060**。加的时候备注一下：**研究方向 + 学校/公司 + 知乎**，即可。然后就可以拉你进群了。

填补缺失值的重要

为了有效地管理数据，理解缺失值的概念很重要。如果数据工作者没有正确处理缺失的数字，他或她可能会对数据得出错误的结论，这将对建模阶段产生重大影响。这是数据分析中的一个重要问题，因为它会影响结果。在分析数据过程，当我们发现有一个或多个特征数据缺失时，此时就很难完全理解或相信由此所得到的结论或建立的模型。数据中的缺失值可力，甚至由于估计的偏差而导致错误的结果。

缺失值导致的问题

1. 在缺乏证据的情况下，统计能力，即检验在零假设错误时拒绝该零
2. 数据的丢失可能导致参数估计出现偏差。
3. 具有降低样本代表性的能力。

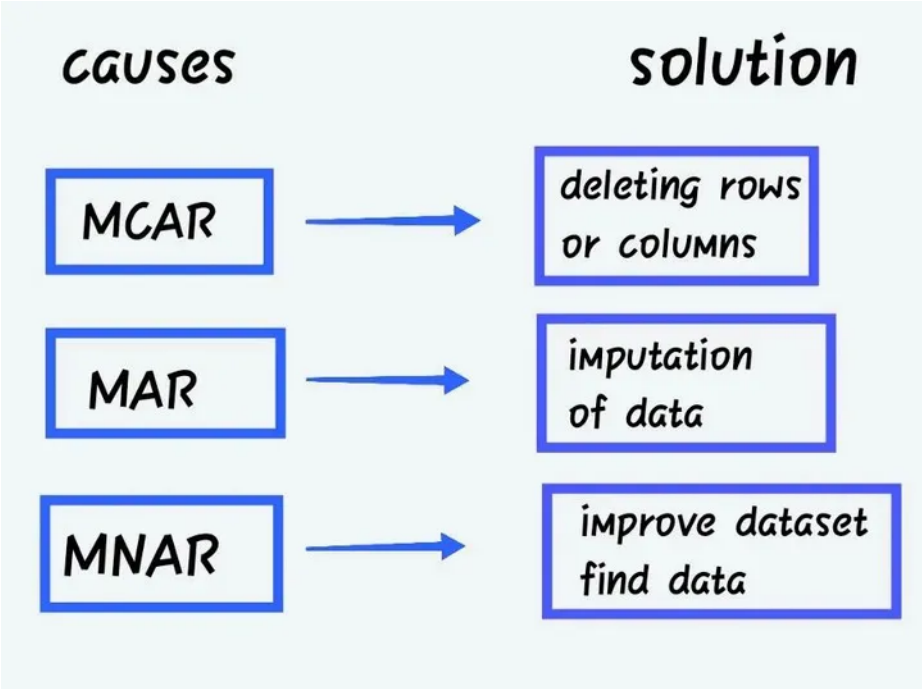
登录即可查看 **超5亿** 专业优质内容

超 5 千万创作者的优质提问、专业回答、深度文章和精彩视频尽在知乎。

立即登录/注册

根据数据集或数据中不存在的模式或数据，可以将其分类。

1. **完全随机缺失(MCAR)**当丢失数据的概率与要获得的精确值或观察到的答案的集合无关时。
2. **随机缺失(MAR)**当丢失响应的概率由观察到的响应的集合而不是预期达到的精确缺失值决定时。
3. **非随机缺失(MNAR)**



除了上述类别之外，MNAR 是缺失数据。MNAR 数据案例很难处理。在这种情况下，对缺失数据进行建模是获得参数的公平近似值的唯一方法。

缺失值的类别

具有缺失值的列分为以下几类：

1. **连续变量或特征** —— 数值数据集，即数字可以是任何类型
2. **分类变量或特征** —— 它可以是数值的或客观的类型。
例如：客户评分 -- 差、满意、好、更好、最好或性别 -- 男性或女性。

缺失值插补类型

插补有多种大小和形式。这是在为我们的应用程序建模以提高精度之前解决数据集中缺失数据问题的方法之一。

1. **单变量插补或均值插补**是指仅使用目标变量对值进行插补。
2. **多元插补**：根据其他因素**插补**值，例如使用线性回归根据其他变量估计缺失值。
3. **单一插补**：要构建单个插补数据集，只需在数据集中**插补**一次缺失值。
4. **大量插补**：在数据集中多次插补相同的缺失值。这本质上需要重复单个插补以获得大量插补数据集。

如何处理数据集中缺失的数据

有很多方法可以处理缺失的数据。首先导入我们需要的库。

导入库

登录即可查看 超5亿 专业优质内容

超 5 千万创作者的优质提问、专业回答、深度文章和精彩视频尽在知乎。



```
# 然后我们需要导入数据集，  
dataset.head()
```

	Salary	Gender	Age	PhD
0	140.0	1	47	1
1	30.0	0	65	1
2	35.1	0	56	0
3	30.0	1	23	0
4	80.0	0	53	1

检查数据集的维度

```
dataset.shape
```

检查缺失值

```
print(dataset.isnull().sum())  
  
Salary      0  
Gender      0  
Age         0  
PhD         0  
dtype: int64
```

01 不作任何处理

不对丢失的数据做任何事情。一方面，有一些算法有处理缺失值的能力，此时我们可以将完全控制权交给算法来控制它如何响应数据。另一方面，各种算法对缺失数据的反应不同。例如，一些算法基于训练损失减少来确定缺失数据的最佳插补值。以 XGBoost 为例。但在某些情况下算法也会出现错误，例如线性回归，此时意味着我们必须在数据预处理阶段或模型失败时处理数据缺失值，我们必须弄清楚出了什么问题。


实际工作中，我们需要根据实际情况具体分析，这里为了演示缺失值的处理方法，我们运用试错法，根据结果反推缺失值的处理方法。

```
# 带有缺失值的旧数据集  
dataset["Age"][:10]  
  
0    47  
1    65  
2    56  
3    23  
4    53  
5    27  
6    53  
7    30  
8    44  
9    63
```

×

登录即可查看 超5亿 专业优质内容

超 5 千万创作者的优质提问、专业回答、深度文章和精彩视频尽在知乎。



排除具有缺失数据的记录是一个最简单的方法。但可能会因此而丢失一些关键数据点。我们可以通过使用 Python pandas 包的 `dropna()` 函数删除所有缺失值的列来完成此操作。与其消除所有列中的所有缺失值，不如利用领域知识或寻求领域专家的帮助来有选择地删除具有与机器学习问题无关的缺失值的行/列。

- **优点：** 删除丢失的数据后，模型的鲁棒性将会变得更好。
- **缺点：** 有用的数据丢失，不能小看了这点，这也可能很重要。但如果数据集中缺失值很多，将会严重影响建模效率。

```
#deleting 行 - 错过的值
dataset.dropna(inplace=True)
print(dataset.isnull().sum())

Salary      0
Gender      0
Age         0
PhD         0
dtype: int64
```

03 均值插补

使用这种方法，可以先计算列的非缺失值的均值，然后分别替换每列中的缺失值，并独立于其他列。最大的缺点是它只能用于数值数据。这是一种简单快速的方法，适用于小型数值数据集。但是，存在例如忽略特征相关性的事实的限制等。每次填补仅适用于其中某一独立的列。

此外，如果跳过离群值处理，几乎肯定会替换一个倾斜的平均值，从而降低模型的整体质量。

- **缺点：** 只适用于数值数据集，不能在独立变量之间的协方差

```
#Mean - 缺失值
dataset["Age"] = dataset["Age"].replace(np.NaN, dataset["Age"].mean())
print(dataset["Age"][:10])

0      47
1      65
2      56
3      23
4      53
5      27
6      53
7      30
8      44
9      63
Name: Age, dtype: int64
```

04 中位数插补

解决上述方法中的异常值问题的另一种插补技术是利用中值。排序时，它会忽略异常值的影响并更新该列中出现的中间值。

- **缺点：** 只适用于数值数据集，不能在独立变量之间的协方差


```
#Median - 缺失值
dataset["Age"] = dataset["Age"].replace(np.NaN, dataset["Age"].median())
print(dataset["Age"][:10])
```

05 众数插补

×

登录即可查看 超5亿 专业优质内容

超 5 千万创作者的优质提问、专业回答、深度文章和精彩视频尽在知乎。



值来填补缺失值。

不幸的是，由于这种方法忽略了特征连接，存在数据偏差的危险。如果类别值不平衡，则更有可能在数据中引入偏差（类别不平衡问题）。

- **优点：** 适用于所有格式的数据。
- **缺点：** 无法预测独立特征之间的协方差值。

```
#Mode - 缺失值
import statistics
dataset["Age"] = dataset["Age"].replace(np.NaN, statistics.mode(dataset["Age"]))
print(dataset["Age"][:10])
```

06 分类值的插补

当分类列有缺失值时，可以使用最常用的类别来填补空白。如果有很多缺失值，可以创建一个新类别来替换它们。

- **优点：** 适用于小数据集。通过插入新类别来弥补损失
- **缺点：** 不能用于除分类数据之外的其他数据，额外的编码特征可能会导致精度下降

```
dataset.isnull().sum()

# 确实值 - 分类 - 解决方案
dataset["PhD"] = dataset["PhD"].fillna('U')

# 检查分类中的缺失值 - 机舱
dataset.isnull().sum()
```

07 前一次观测结果(LOCF)

这是一种常见的统计方法，用于分析纵向重复测量数据时，一些后续观察缺失。

```
#LOCF - 前一次观测结果
dataset["Age"] = dataset["Age"].fillna(method = 'ffill')
dataset.isnull().sum()
```

08 线性插值

这是一种近似于缺失值的方法，沿着直线将点按递增顺序连接起来。简而言之，它以与在它之前出现的值相同的升序计算未知值。因为线性插值是默认的方法，我们不需要在使用它的时候指定它。这种方法常用于时间序列数据集。

```
#interpolation - 线性
dataset["Age"] = dataset["Age"].interpolate(method='linear', limit_direction='forward')

dataset.isnull().sum()
```

09 KNN 插补

一种基本的分类方法是 k 最近邻 (kNN) 算法。类成员是 k-NN 分类的。项目的分类取决于它与训练集中的点的相似程度，该对象将进入其 k 最近邻的类。如果 k = 1，则该项目被简单地分配给该项目最近邻居的类。使用缺失邻域，然后根据邻域中的非缺失值对它们进行插补可能有助于生成关

```
# for knn imputation - 我们需要移除归一化数据和我们需要转换的分类
```

收起


简介

群 # for knn imputation - 我们需要移除归一化数据和我们需要转换的分类

×

登录即可查看 超5亿 专业优质内容

超 5 千万创作者的优质提问、专业回答、深度文章和精彩视频尽在知乎。



类别	<code>dataset = pd.concat([dataset, cat_dummies], axis=1)</code> <code>dataset.head()</code>
补类型	<code># 删除不需要的功能</code>
数据集中缺失的数据	<code>dataset = dataset.drop(['Gender'], axis=1)</code> <code>dataset.head()</code>
任何处理	
用时将其删除（主要...	<code># scaling 在 knn 之前是强制性的</code>
插补	<code>from sklearn.preprocessing import MinMaxScaler</code> <code>scaler = MinMaxScaler()</code>
数据插补	<code>dataset = pd.DataFrame(scaler.fit_transform(dataset), columns = dataset.columns)</code> <code>dataset.head()</code>
插补	
值的插补	<code># knn 插值</code>
次观测结果(LOCF)	<code>from sklearn.impute import KNNImputer</code> <code>imputer = KNNImputer(n_neighbors=3)</code> <code>dataset = pd.DataFrame(imputer.fit_transform(dataset), columns = dataset.columns)</code>
插值	
N 插补	<code>#检查是否丢失</code> <code>dataset.isnull().sum()</code>
式方程 (MICE) 进行...	
.	
.	

10 由链式方程 (MICE) 进行多元插补的插补

MICE 是一种通过多重插补替换数据集中缺失数据值的方法。可以首先制作一个或多个变量中缺失值的数据集的重复副本。

```
#MICE
import numpy as np
import pandas as pd
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
df = pd.read_csv('https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv')
df = df.drop(['PassengerId', 'Name'],axis=1)
df = df[['Survived', 'Pclass', 'Sex', 'SibSp', 'Parch', 'Fare', 'Age']]
df["Sex"] = [1 if x=="male" else 0 for x in df["Sex"]]

df.isnull().sum()
imputer=IterativeImputer(imputation_order='ascending',max_iter=10,random_state=42,n_ne
imputed_dataset = imputer.fit_transform(df)
```

写作最后

对于我们的数据集，我们可以使用上述想法来解决缺失值。处理缺失值的方法取决于我们的特征中的缺失值和我们需要应用的模型。

推荐文章


- 真香！机器学习中这3种交叉验证方法要掌握！
- 150亿参数，谷歌开源了史上最大视觉模型V-MoE的全部代码
- Facebook 推出多模态通用模型 FLAVA，吊打 CLIP 平均十个点！
- GitHub 7.5k star量，各种视觉Transformer的PyTorch实现合集整理
- 赶快收藏，PyTorch 常用代码段合集真香！

清华南开开发attention 7年全回顾：注意力机制还有7大问题要研究！

×

登录即可查看 超5亿 专业优质内容

超 5 千万创作者的优质提问、专业回答、深度文章和精彩视频尽在知乎。



何凯明团队又出新论文！北大、上交校友教你用ViT做迁移学习

GAN “家族” 又添新成员——EditGAN，不但能自己修图，还修得比你我都好

大道至简，何恺明新论文火了：Masked Autoencoders让计算机视觉通向大模型

kaggle、TDS、arXiv....., 我最喜欢的10个顶级数据科学资源

当Transformer又遇见U-Net！Transformer-Unet：医学图像分割新工作

有了这个机器学习画图神器，论文、博客都可以事半功倍了！

谷歌打怪升级之路：从EfficientNet到EfficientNetV2

不用1750亿！OpenAI CEO放话：GPT-4参数量不增反减

编辑于 2022-01-16 12:44

机器学习 深度学习（Deep Learning） 缺失值处理

写下你的评论...

1 条评论

默认 最新



QuanterLi

缺失值处理很重要。根据不同业务场景，选择不同的缺失值更重要。

2022-01-15

回复 1

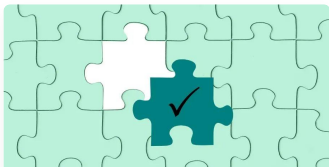
文章被以下专栏收录



机器学习社区

专注分享学术论文、机器学习、人工智能、Python

推荐阅读



机器学习中缺失值处理方法大全
(附代码)

沪漂城哥

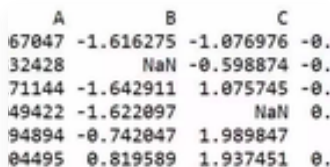
发表于学习笔记



机器学习中处理缺失值的9种方法

deeph...

发表于deeph...



机器学习tips（一）：缺失值的处理

疯狂的ma...

发表于数据分析和...

机器学习
和分类

本文为
老师系
公式未
Course
Top Ur
不定期
关右

登录即可查看 超5亿 专业优质内容

超 5 千万创作者的优质提问、专业回
答、深度文章和精彩视频尽在知乎。

