What is stratified sampling in pandas and how is it used?

Authored by

stats writer

April 19, 2024

Stratified sampling in pandas is a data sampling technique that involves dividing a dataset into subgroups or strata based on specific characteristics or attributes. This method is used to ensure that the sample accurately represents the population being studied. It involves randomly selecting data points from each stratum, which helps to reduce bias and provide a more representative sample. Stratified sampling is commonly used in data analysis and machine learning to improve the accuracy and reliability of results. This technique allows for a more comprehensive and in-depth analysis of the data by ensuring that all subgroups within the population are adequately represented in the sample.

Stratified Sampling in Pandas (With Examples)

Researchers often take samples from a population and use the data from the sample to draw conclusions about the population as a whole.

One commonly used sampling method is stratified random sampling, in which a population is split into groups and a certain number of members from each group are randomly selected to be included in the sample.

This tutorial explains two methods for performing stratified random sampling in Python.

Example 1: Stratified Sampling Using Counts

Suppose we have the following pandas DataFrame that contains data about 8 basketball players on 2 different teams:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'team': ,
'position': ,
'assists': ,
'rebounds': })

#view DataFrame
df
```

team position assists rebounds
0 A G 5 11
1 A G 7 8
2 A F 7 10
3 A G 8 6
4 B F 5 6
5 B F 7 9
6 B C 6 6
7 B C 9 10

The following code shows how to perform stratified random sampling by randomly selecting 2 players from each team to be included in the sample:

df.groupby('team', group_keys=False).apply(lambda x: x.sample(2))

team position assists rebounds
0 A G 5 11
3 A G 8 6
6 B C 6 6
5 B F 7 9

Notice that two players from each team are included in the stratified sample.

Example 2: Stratified Sampling Using Proportions

Once again suppose we have the following pandas DataFrame that contains data about 8 basketball players on 2 different teams:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'team': ,
'position': ,
'assists': ,
'rebounds': })
```

#view DataFrame
df

team position assists rebounds
0 A G 5 11
1 A G 7 8
2 A F 7 10
3 A G 8 6
4 B F 5 6
5 B F 7 9
6 B C 6 6
7 B C 9 10

Notice that 6 of the 8 players (75%) in the DataFrame are on team A and 2 out of the 8 players (25%) are on team B.

The following code shows how to perform stratified random sampling such that the proportion of players in the sample from each team matches the proportion of players from each team in the larger DataFrame:

import numpy as np

#define total sample size desired

N = 4

#perform stratified random sampling
df.groupby('team', group_keys=False).apply(lambda x: x.sample(int(np.rint(N*len(x)/len(df))))).sample(frac=1).reset_index(drop=True)

team position assists rebounds
0 B F 7 9
1 B G 8 6
2 B C 6 6
3 A G 7 8

Notice that the proportion of players from team A in the stratified sample (25%) matches the proportion of players from team A in the larger DataFrame.

Similarly, the proportion of players from team B in the stratified sample (75%) matches the proportion of players from team B in the larger DataFrame.

The following tutorials explain how to select other types of samples using pandas:

How to Perform Cluster Sampling in Pandas