samples

March 4, 2024

[ ]:
```
import pandas as pd
import numpy as np
import random
dataset=pd.read_csv("example/census.csv")
datamod=dataset.copy()
```

0.0.1 Simple sampling

[ ]:
```
def simple_sampling(datasset,_,__,):
    return datasset.sample(n=_,random_state=__ )  ##frac or n, whichever is␣
 ↪larrger
data_simple_sample=simple_sampling(datamod,500,2)
data_simple_sample.shape
```

[ ]:
```
(500, 15)
```

[ ]:
```
data_simple_sample.isna().sum()
```

0.0.2 Sistematic sampling

[ ]:
```
def systematic_sampling(datasset, samples_number, seed):
    step= round(len(datasset) / samples_number)
    datasset=datasset.sample(frac=1, random_state=seed).copy  ## better results
    first_step=np.random.randint(0, step)
    index=np.arange(first_step, len(datamod), step)
    return dataset.iloc[index]
data_systematic_sampling=systematic_sampling(datamod,100,1)
data_systematic_sampling.shape
```

[ ]:
```
(100, 15)
```

0.0.3 Grouping_sampling

[ ]:
```
def grounping_sampling(datasset, group_number,seed ):
    id_group, count, groups = 0,0, []
```

```
    group_size = round(len(datasset) / group_number)

    datasset=datasset.sample(frac=1, random_state=seed).copy()

    for _ in datasset.iterrows():
        groups.append(id_group)
        count += 1
        if count == group_size :
            count = 0
            id_group += 1
    datasset['groups'] = groups
    np.random.seed=seed
    selected_group= np.random.randint(0, id_group )
    return datasset[datasset['groups'] == selected_group]

data_grouping_sampling=grounping_sampling(datamod, 10,2)
```

[ ]:
```
data_grouping_sampling.shape
```

[ ]:
```
(3256, 16)
```

0.0.4 Stratified Sampling

[ ]:
```
from sklearn.model_selection import StratifiedShuffleSplit
```

[ ]:
```
def stratified_sampling(datamod, test_siz,seed):
    split=StratifiedShuffleSplit(test_size=test_siz, random_state=seed)

    for x,y in split.split(datamod,datamod ['income']):
        df_x = datamod.iloc[x]   ## dont need, is original data fracional
        df_y = datamod.iloc[y]
    return df_x, df_y
df_x , df_y = stratified_sampling(datamod, 0.2,2 )
data_stratified_sampling=df_y.copy()
```

[ ]:
```
df_x.shape ,  df_y.shape, datamod.shape
```

[ ]:
```
((26048, 15), (6513, 15), (32561, 15))
```

0.0.5 Reservoir Sampling // E-commerce style (no fund)

[ ]:
```
import numpy as np

def resevoir_sampling(dataset, n):
    stream = []
    for i in range(len(dataset)):
```

```
        stream.append(i)

    reservoir = [dataset.iloc[i] for i in range(n)]
    i = n
    while i < len(dataset):
        j = np.random.randint(0, i + 1)
        if j < n:
            reservoir[j] = dataset.iloc[i]
        i += 1
    return pd.DataFrame(reservoir)

data_resevoir_sample = resevoir_sampling(datamod, 10)
```

[ ]:
```
data_resevoir_sample.shape
```

[ ]:
```
(100, 15)
```

0.0.6 Test_samples

[ ]:
```
datamod["age"].mean() , data_simple_sample["age"].mean() ,␣
↪data_systematic_sampling["age"].mean(), data_grouping_sampling["age"].
↪mean(), data_stratified_sampling["age"].mean() ,data_resevoir_sample["age"].
↪mean()
```

[ ]:
```
(38.58164675532078,
 37.172,
 37.9,
 38.083230958230956,
 38.414248426224475,
 37.51)
```