**Pre-processing**

|  |  |  |  |
| --- | --- | --- | --- |
| Country | Age | Salary | Purchased |
| India | 49 | 62000 | No |
| Pakistan | 32 | 38000 | Yes |
| Bhutan | 35 | 44000 | No |
| Bangladesh | NaN | 51000 | No |
| Nepal | 45 | NaN | Yes |
| Srilanka | 40 | 48000 | Yes |
| Burma | NaN | 42000 | No |
| China | 53 | 69000 | Yes |
| Afganistan | 55 | 73000 | No |

**Concept:**

* ***1. missing\_values***: This option is used to pass which value should be treated as the missing value. In our case, a string NaN is a missing value, which will be visible from the screenshot to come.
* ***2. strategy***: We are passing ‘mean’ value for this option. It takes three values in Python which are as follows.
  + *2.1 mean*: This is the default value. The mean of the column will replace the missing value (NaN in our case).
  + *2.2 median*: The median of the column will replace the missing value.
  + *2.3 most\_frequent*: The most frequent value will replace the missing value for the respective column.
* ***3. axis***: We are passing 0 for this option. It takes two values, either 0 or 1, where 0 indicates that we want to impute the missing data in columns, whereas 1 indicates the missing data is imputed in rows.

import pandas as pd

dataset = pd.read\_csv('E:\\Latha\\LathaSKPIMCS\\Machine Learning\\Class\\Practical\\Preprocessing\\Data1.csv')

X = dataset.iloc[:,:-1].values #Takes all rows of all columns except the last column

Y = dataset.iloc[:,-1].values # Takes all rows of the last column

X

Y

print(dataset.columns)

dataset

dataset.info()

dataset.head()

dataset.tail()

#Row and column count

dataset.shape

#Count missing values

dataset.isnull().sum().sort\_values(ascending=False)

dataset.isnull().sum()

#Removing insufficient column

dataset\_new = dataset.drop(['Age',], axis = 1)

dataset\_new

#To measure the central tendency of variables

dataset\_new.describe()

#To change column name

dataset.rename(index=str, columns={'Country' : 'Countries','Age' : 'age', 'Salary' : 'Sal','Purchased' : 'Purchased'}, inplace = True)

dataset

#Count missing values

dataset.isnull().sum().sort\_values(ascending=False)

#Print the missing value column

dataset[dataset.isnull().any(axis=1)].head()

#Remove missing value rows

ds\_new = dataset.dropna()

ds\_new

ds\_new.shape

ds\_new.isnull().sum()

#To check datatype

ds\_new.dtypes

#To convert as integer

ds\_new['age'] = ds\_new['Age'].astype('int64')

ds\_new.dtypes

ds\_new

**Imputing Mean, Median and Most\_frequent**

from sklearn.impute import SimpleImputer

import numpy as np

imputer = SimpleImputer(missing\_values=np.nan, strategy='mean')

imputer = imputer.fit(X[:, 1:3])

X[:, 1:3] = imputer.transform(X[:, 1:3])

X

from sklearn.impute import SimpleImputer

imputer = SimpleImputer(missing\_values = 'NaN', strategy = 'median', axis=0)

imputer = imputer.fit(X[:, 1:3])

X[:, 1:3] = imputer.transform(X[:, 1:3])

X

from sklearn.impute import SimpleImputer

imputer = SimpleImputer(missing\_values = 'NaN', strategy = 'most\_frequent', axis=0)

imputer = imputer.fit(X[:, 1:3])

X[:, 1:3] = imputer.transform(X[:, 1:3])

X