Simple Imputer in Python

SimpleImputer is a class in the sklearn.impute module that can be used to replace missing values in a dataset, using a variety of input strategies. SimpleImputer is designed to work with numerical data, but can also handle categorical data represented as strings. SimpleImputer can be used as part of a scikit-learn Pipeline. The default strategy is “mean”, which replaces missing values with the mean value of the column. Other options include “most_frequent” (which replaces missing values with the most common value in the column) and “constant” (which replaces missing values with a constant value). SimpleImputer can also be used to impute multiple columns at once by passing in a list of column names. SimpleImputer will then replace missing values in all of the specified columns. When using SimpleImputer, it is important to consider whether or not imputing is the best option for your data. In some cases, it may be better to drop rows or columns with missing values instead of imputing them.

SimpleImputer class is used to impute / replace the numerical or categorical missing data related to one or more features with appropriate values such as following:

1. **Mean:** When SimpleImputer() is invoked without any arguments, it defaults to using the mean strategy. Missing values get replaced with the mean along each column. This strategy can only be used with numeric data.
2. **Median:** Missing values get replaced with the median along each column. This strategy can only be used with numeric data.
3. **Most frequent (mode):** Missing values get replaced with the most frequent value along each column. This strategy can be used with strings or numeric data.
4. **Constant:** Missing values get replaced with the fill_value. This strategy can be used with strings or numeric data.

Each of the above type represents **strategy** when creating an instance of SimpleImputer. Here is the Python code sample representing the usage of SimpleImputor for replacing numerical missing value with the mean:

```
In [188]: students = [[85, 'M', 'verygood'],
                      [95, 'F', 'excellent'],
                      [75, None, 'good'],
                      [np.NaN, 'M', 'average'],
                      [70, 'M', 'good'],
                      [np.NaN, None, 'verygood'],
                      [92, 'F', 'verygood'],
                      [98, 'M', 'excellent']]
```

```
In [189]: dfstd = pd.DataFrame(students)
          dfstd.columns = ['marks', 'gender', 'result']
```

```
In [190]: dfstd
```

Out[190]:

	marks	gender	result
0	85.0	M	verygood
1	95.0	F	excellent
2	75.0	None	good
3	NaN	M	average
4	70.0	M	good
5	NaN	None	verygood
6	92.0	F	verygood
7	98.0	M	excellent

```
In [187]: from sklearn.impute import SimpleImputer

          imputer = SimpleImputer(missing_values=np.NaN, strategy='mean')
          dfstd.marks = imputer.fit_transform(dfstd['marks'].values.reshape(-1,1))[:,0]
          dfstd
```

Out[187]:

	marks	gender	result
0	85.000000	M	verygood
1	95.000000	F	excellent
2	75.000000	F	None
3	85.833333	M	average
4	70.000000	M	good
5	85.833333	M	None
6	92.000000	F	verygood
7	98.000000	M	excellent