

DWDM Tutorial 3

U18CO021: SAHIL BONDRE

3. Analyse the above techniques

1 Import Libraries

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
[2]: from sklearn.datasets import load_iris
```

1.1 Loading Data

```
[3]: data = load_iris()
df = pd.DataFrame(np.array(data.data), columns=data.feature_names)
df["class"] = data.target
# Add random NA Values for analysis
df = df.mask(np.random.random(df.shape) < 0.05)
df.describe()
```

```
[3]:      sepal length (cm)  sepal width (cm)  petal length (cm)  \
count      147.000000      142.000000      142.000000
mean         5.836735         3.052817         3.762676
std          0.827861         0.439002         1.760142
min          4.300000         2.000000         1.000000
25%          5.100000         2.800000         1.600000
50%          5.800000         3.000000         4.350000
75%          6.400000         3.300000         5.100000
max          7.900000         4.400000         6.900000

      petal width (cm)      class
count      141.000000  141.000000
mean         1.180851    1.007092
std          0.761757    0.815006
min          0.100000    0.000000
```

25%	0.300000	0.000000
50%	1.300000	1.000000
75%	1.800000	2.000000
max	2.500000	2.000000

```
[4]: df.head()
```

```
[4]:   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1                3.5                NaN                0.2
1                4.9                3.0                1.4                0.2
2                4.7                3.2                1.3                0.2
3                4.6                3.1                1.5                0.2
4                5.0                3.6                1.4                0.2

   class
0    NaN
1    0.0
2    0.0
3    0.0
4    0.0
```

```
[5]: df.isna().sum()
```

```
[5]: sepal length (cm)    3
     sepal width (cm)    8
     petal length (cm)    8
     petal width (cm)    9
     class              9
     dtype: int64
```

1.2 KNN Imputation

```
[6]: from sklearn.impute import KNNImputer

knn_imputer = KNNImputer(missing_values=np.NaN)
knn_df = pd.DataFrame(knn_imputer.fit_transform(df))
knn_df.columns = df.columns
knn_df.index = df.index
print(knn_df.isna().sum())
knn_df.describe()
```

```
sepal length (cm)    0
sepal width (cm)    0
petal length (cm)    0
petal width (cm)    0
```

```
class          0
dtype: int64
```

```
[6]:      sepal length (cm)  sepal width (cm)  petal length (cm)  \
count          150.000000          150.000000          150.000000
mean             5.842400             3.052133             3.761200
std              0.826114             0.432169             1.765635
min              4.300000             2.000000             1.000000
25%              5.100000             2.800000             1.600000
50%              5.800000             3.000000             4.350000
75%              6.400000             3.300000             5.100000
max              7.900000             4.400000             6.900000
```

```
      petal width (cm)      class
count          150.000000  150.000000
mean             1.202667   0.993333
std              0.767136   0.813119
min              0.100000   0.000000
25%              0.300000   0.000000
50%              1.300000   1.000000
75%              1.800000   2.000000
max              2.500000   2.000000
```

1.3 Single Value Imputation

```
[7]: from sklearn.impute import SimpleImputer

simple_imputer = SimpleImputer(missing_values=np.NAN, strategy="mean")
simple_df = pd.DataFrame(simple_imputer.fit_transform(df))
simple_df.columns = df.columns
simple_df.index = df.index
print(simple_df.isna().sum())
simple_df.describe()
```

```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
class               0
dtype: int64
```

```
[7]:      sepal length (cm)  sepal width (cm)  petal length (cm)  \
count          150.000000          150.000000          150.000000
mean             5.836735             3.052817             3.762676
std              0.819484             0.427054             1.712238
min              4.300000             2.000000             1.000000
```

25%	5.100000	2.800000	1.600000
50%	5.800000	3.000000	4.200000
75%	6.400000	3.300000	5.100000
max	7.900000	4.400000	6.900000

	petal width (cm)	class
count	150.000000	150.000000
mean	1.180851	1.007092
std	0.738392	0.790009
min	0.100000	0.000000
25%	0.300000	0.000000
50%	1.300000	1.000000
75%	1.800000	2.000000
max	2.500000	2.000000

1.4 Most Frequent Value Replacement

```
[8]: mf_imputer = SimpleImputer(missing_values=np.NAN, strategy="most_frequent")
mf_df = pd.DataFrame(mf_imputer.fit_transform(df))
mf_df.columns = df.columns
mf_df.index = df.index
print(mf_df.isna().sum())
mf_df.describe()
```

```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
class                0
dtype: int64
```

```
[8]:      sepal length (cm)  sepal width (cm)  petal length (cm)  \
count      150.000000      150.000000      150.000000
mean         5.82000      3.05000      3.642000
std          0.82787      0.42722      1.786612
min          4.30000      2.00000      1.000000
25%          5.10000      2.80000      1.500000
50%          5.75000      3.00000      4.200000
75%          6.40000      3.30000      5.100000
max          7.90000      4.40000      6.900000
```

	petal width (cm)	class
count	150.000000	150.000000
mean	1.122000	1.006667
std	0.774499	0.790010
min	0.100000	0.000000

25%	0.200000	0.000000
50%	1.300000	1.000000
75%	1.800000	2.000000
max	2.500000	2.000000

1.5 Global Constant Replacement

```
[9]: gcr_imputer = SimpleImputer(missing_values=np.NAN, strategy="constant",
    ↪fill_value=2)
gcr_df = pd.DataFrame(gcr_imputer.fit_transform(pd.DataFrame(df["class"])))
gcr_df.columns = ["class"]
print(gcr_df.isna().sum())
gcr_df.describe()
```

```
class    0
dtype: int64
```

```
[9]:          class
count  150.000000
mean    1.066667
std     0.824675
min     0.000000
25%     0.000000
50%     1.000000
75%     2.000000
max     2.000000
```