KNN Imputer

In [1]:
```
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.impute import KNNImputer,SimpleImputer
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score
```

In [2]:
```
df = pd.read_csv('train.csv')[['Age','Pclass','Fare','Survived']]
```

In [3]:
```
df.head()
```

Out[3]:

	Age	Pclass	Fare	Survived
0	22.0	3	7.2500	0
1	38.0	1	71.2833	1
2	26.0	3	7.9250	1
3	35.0	1	53.1000	1
4	35.0	3	8.0500	0

In [4]:
```
df.isnull().mean() * 100
```

Out[4]:
```
Age         19.86532
Pclass       0.00000
Fare         0.00000
Survived     0.00000
dtype: float64
```

In [5]:
```
X = df.drop(columns=['Survived'])
y = df['Survived']
```

In [6]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=2)
```

In [7]:
```
X_train.head()
```

Out[7]:

	Age	Pclass	Fare
30	40.0	1	27.7208
10	4.0	3	16.7000
873	47.0	3	9.0000
182	9.0	3	31.3875
876	20.0	3	9.8458

In [8]:
```
knn = KNNImputer(n_neighbors=3,weights='distance')

X_train_trf = knn.fit_transform(X_train)
X_test_trf = knn.transform(X_test)
```

In [9]:
```
lr = LogisticRegression()

lr.fit(X_train_trf,y_train)

y_pred = lr.predict(X_test_trf)

accuracy_score(y_test,y_pred)
```

Out[9]:
```
0.7150837988826816
```

In [10]:
```
# Comparision with Simple Imputer --> mean

si = SimpleImputer()

X_train_trf2 = si.fit_transform(X_train)
X_test_trf2 = si.transform(X_test)
```

In [11]:
```
lr = LogisticRegression()

lr.fit(X_train_trf2,y_train)

y_pred2 = lr.predict(X_test_trf2)

accuracy_score(y_test,y_pred2)
```

Out[11]:
```
0.6927374301675978
```

In [ ]:

In [ ]: