Day 14 | Start Coding | Missing Indicator Imputation

Import Libraries

In [ ]:

```
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.impute import MissingIndicator,SimpleImputer
```

Import Dataset

In [2]:

```
df = pd.read_csv('train.csv',usecols=['Age','Fare','Survived'])
```

In [3]:

```
df.head()
```

Out[3]:

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

Create X & Y

In [4]:

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

Train & Test Split

In [5]:

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=2)
```

In [6]:

```
X_train.head()
```

Out[6]:

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

Review Without “Missing Indicator Method” Technique

In [7]:

```
si = SimpleImputer()
X_train_trf = si.fit_transform(X_train)
X_test_trf = si.transform(X_test)
```

In [8]:

```
X_train_trf
```

Out[8]:

```
array([[ 40.        ,  27.7208    ],
       [  4.        ,  16.7       ],
       [ 47.        ,   9.        ],
       ...,
       [ 71.        ,  49.5042    ],
       [ 29.78590426, 221.7792    ],
       [ 29.78590426,  25.925     ]])
```

Call Logistic Regression

In [9]:

```
from sklearn.linear_model import LogisticRegression

clf = LogisticRegression()

clf.fit(X_train_trf,y_train)

y_pred = clf.predict(X_test_trf)

from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[9]:

```
0.6145251396648045
```

Define Missing Indicator

In [10]:

```
mi = MissingIndicator()

mi.fit(X_train)
```

Out[10]:

```
▾ MissingIndicator
MissingIndicator()
```

In [11]:

```
MissingIndicator()
```

Out[11]:

```
▾ MissingIndicator
MissingIndicator()
```

In [12]:

```
mi.features_
```

Out[12]:

```
array([0], dtype=int64)
```

```
In [13]: X_train_missing = mi.transform(X_train)
```

```
In [14]: X_train_missing
```

```
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
```

Transform: Train Missing

```
In [15]: X_test_missing = mi.transform(X_test)
```

```
In [16]: X_test_missing
```

```
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [ True],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
```

Transform: Test Missing

```
In [17]: X_train['Age_NA'] = X_train_missing
In [18]: X_test
```

Out[18]:

	Age	Fare
707	42.0	26.2875
37	21.0	8.0500
615	24.0	65.0000
169	28.0	56.4958
68	17.0	7.9250
...	...	...
89	24.0	8.0500
80	22.0	9.0000
846	NaN	69.5500
870	26.0	7.8958
251	29.0	10.4625

179 rows × 2 columns

Create New Column

```
In [19]: X_test['Age_NA'] = X_test_missing
```

```
In [20]: X_train
```

Out[20]:

	Age	Fare	Age_NA
30	40.0	27.7208	False
10	4.0	16.7000	False
873	47.0	9.0000	False
182	9.0	31.3875	False
876	20.0	9.8458	False
...	...	...	...
534	30.0	8.6625	False
584	NaN	8.7125	True
493	71.0	49.5042	False
527	NaN	221.7792	True
168	NaN	25.9250	True

712 rows × 3 columns

Writing Code again and Check Accuracy

In [23]:

```
si = SimpleImputer()

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

In [24]:

```
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression()
clf.fit(X_train_trf2,y_train)
y_pred = clf.predict(X_test_trf2)
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[24]:

```
0.6312849162011173
```

In [ ]: