

## Lesson-2 Encoding Methods

May 23, 2024

```
[1]: # creating a random dataset
import pandas as pd
data = {"Grade": ["A", 'A', 'B', 'C', 'A', 'D', 'D']}

df = pd.DataFrame(data)
```

```
[3]: # check if the dataset had non-numeric values
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  -
 0   Grade    7 non-null      object
dtypes: object(1)
memory usage: 184.0+ bytes
```

```
[5]: # copying my original dataset
df1 = df.copy()

# importing the sklearn module
from sklearn.preprocessing import LabelEncoder

# initialize the model
label = LabelEncoder()

# fit the dataset
df1['Grade'] = label.fit_transform(df1['Grade'])

df1
```

```
[5]:   Grade
0      0
1      0
2      1
3      2
4      0
```

```
5     3
6     3
```

```
[6]: # copy the dataset
df2 = df.copy()

# apply the dummy variables
df2 = pd.get_dummies(df2, columns=['Grade'])

df2
```

```
[6]:   Grade_A  Grade_B  Grade_C  Grade_D
0         1         0         0         0
1         1         0         0         0
2         0         1         0         0
3         0         0         1         0
4         1         0         0         0
5         0         0         0         1
6         0         0         0         1
```

```
[ ]: # apply the dummy variables on multiple columns
df2 = pd.get_dummies(df2, columns=['Grade', 'Grade2', 'Grade3'])
```

```
[7]: # copy dataset
df3 = df.copy()
# create a nested dictionary
encoded = {'Grade' : {"A": 90, "B": 80, "C": 70, "D": 60}}
# apply replace method
df3.replace(encoded)
df3
```

```
[7]:   Grade
0     A
1     A
2     B
3     C
4     A
5     D
6     D
```

```
[ ]:
```