بنام خدا

جلسه دوازدهم داده کاوی –

همانطور که مشاهده کردید اکثر الگوریتم های داده کاوی صرفا قادر به پردازش مجموعه های داده ای عددی هستند. حال فرض کنید یک مجموعه داده ای مانند یک گونه های یک گل خاص علاوه بر سایز طول و عرض گلبرگ و کاسبرگ ویژگی دیگری بنام رنگ دارد که می تواند قرمز، زرد یا نارنجی باشد. این نوع مجموعه های داده ای مجموعه های داده ای کتگوریال (Categorial Datasets) نامیده می شود.

| Sepal length | Sepal width | Petal length | Petal width | Colour |
|---|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 | Red |
| 3 | 4.9 | 3 | 1.4 | Red |
| 4 | 4.7 | 3.2 | 1.3 | Orange |
| 5 | 4.6 | 3.1 | 1.5 | Yellow |
| . . . | | | | |

برای این ستون در این مجموعه داده ای روش one-hot ( One-Hot Encoding Transformation) مناسب است:

| Sepal length | Sepal width | Petal length | Petal width | Red | Orange | Yellow |
|---|---|---|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 | 1 | 0 | 0 |
| 3 | 4.9 | 3 | 1.4 | 1 | 0 | 0 |
| 4 | 4.7 | 3.2 | 1.3 | 0 | 1 | 0 |
| 5 | 4.6 | 3.1 | 1.5 | 0 | 0 | 1 |
| . . . | | | | | | |

برای ستونِ رنگِ مجموعه فوق، روش One-Hot مناسب بود چراکه تعداد رنگ کم بود و هیچ رنگی اولویت خاصی بر دیگری نداشت.

اما حال حساب کنید که برای مجموعه داده ای در مورد خانه سایز آن که به صورت کتگوری (کوچک، متوسط و بزرگ) ثبت شده است در قیمت تاثیر دارد.

| … | #beds | Size-Type | Price |
|---|---|---|---|
| | | Small | 1.2 |
| | | medium | 2 |
| | | Small | 1.5 |
| | | Large | 3 |
| . . . | | | |

در این صورت می توان برای سایز از روش Label Encoding استفاده کرد. در این روش به هر نوع از ستون مربوطه یک عدد اختصاص داده می شود مثلا ۰ برای small، ۱ برای medium و ۲ برای large .

| … | #beds | Size-Type | Price |
|---|---|---|---|
| | | 0 | 1.2 |
| | | 1 | 2 |
| | | 0 | 1.5 |
| | | 2 | 3 |
| . . . | | | |

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import OneHotEncoder

# Sample dataset
data = pd.DataFrame({'Color': ['Red', 'Blue', 'Green', 'Red',
'Blue']})

# Initialize OneHotEncoder
encoder = OneHotEncoder(sparse=False)  # Set
sparse=False to get a NumPy array

# Transform the data
encoded_data = encoder.fit_transform(data[['Color']])

# Convert to DataFrame for better readability
encoded_df = pd.DataFrame(encoded_data,
columns=encoder.get_feature_names_out(['Color']))
```

برای label-encoding

```
from sklearn.preprocessing import LabelEncoder

# Sample dataset
data = pd.DataFrame({'Size': ['Small', 'Medium', 'Large',
'Medium', 'Small']})

# Initialize LabelEncoder
label_encoder = LabelEncoder()

# Transform the data
data['Size_Encoded'] =
label_encoder.fit_transform(data['Size'])
```

شما تا به حال تفاوت یادگیری supervised و unsupervised را آموخته اید. همچنین می دانید که خوشه بندی (clustering) یک مسئله unsupervised learning است و برخی از الگوریتم های خوشه بندی مانند k-means، DBSCAN و Hierarchical را آموخته اید. این الگوریتم ها نساز دارند تا مجموعه داده ای عددی باشند.

برای supervised learning و مسئله classification الگوریتم هایی هستند که نیازی به تبدیل ندارند، مثلا: Decision Tree.