라벨인코딩 & 원핫인코딩

* 인코딩이란?

인코딩은 사용자가 입력한 문자나 기호들을 컴퓨터가 이용할 수 있는 신호로 만드는 것을 말한다. 넓은 의미의 컴퓨터는 이러한 신호를 입력받고 처리하는 기계를 뜻하며, 신호 처리 시스템을 통해 이렇게 처리된 정보를 사용자가 이해할 수 있게 된다.[2] 이 신호를 입력하는 인코딩과 문자

어떤 정보를 정해진 규칙(Code[1])에 따라 변환하는 것(en-code-ing)을 일컫는다.

ASCII 인코딩

라벨인코딩

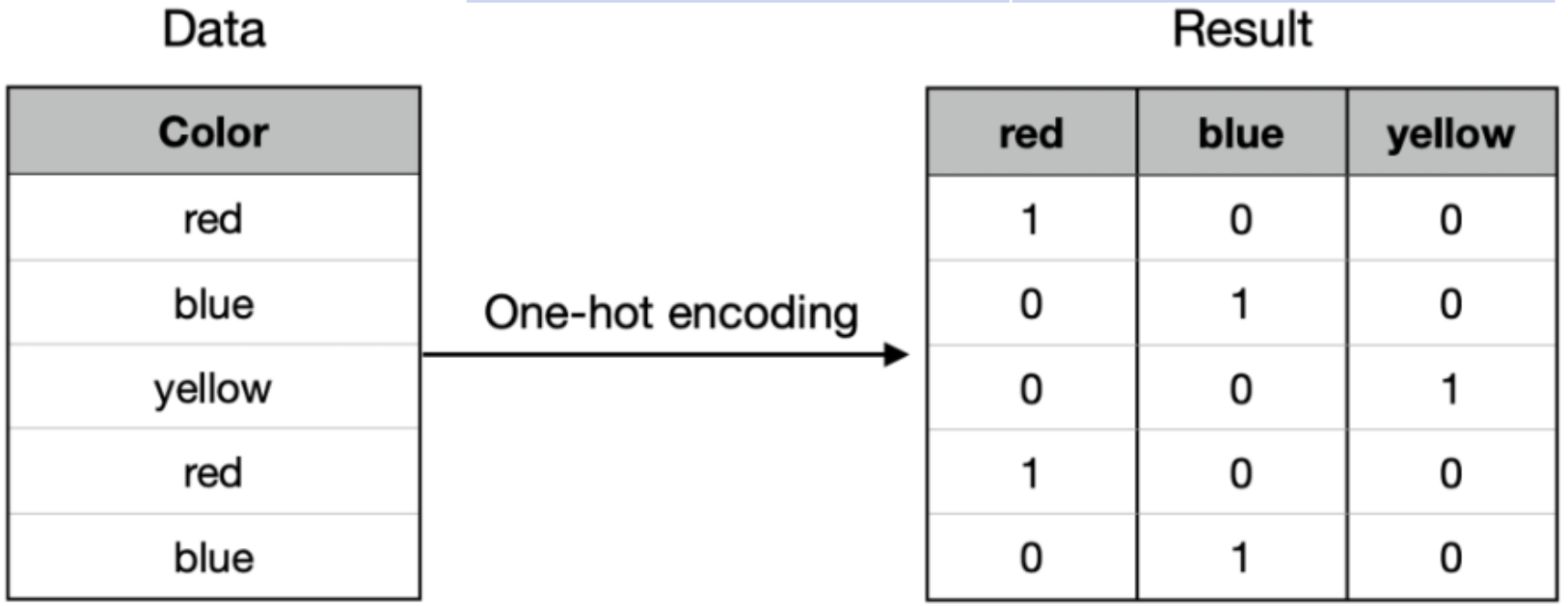
- 각 데이터에 일련번호 할당
- 데이터 수 변화 x
- 가까울수록 높은 상관성
  결정트리는 괜찮음

Data	Label-Encoding
Apple	0
Banana	1
Grape	2
Strawberry	3

원핫인코딩

- 각 데이터에 벡터 할당
- 1 한 개와 0 여러 개인 벡터
- Nunique에 따라 차원 수 증가

Data	One-Hot-Encoding
Red	[1,0,0]
Blue	[0,1,0]
Yellow	[0,0,1]



데이터 인코딩, 왜 사용하는지?

* 인코딩이란?

인코딩은 사용자가 입력한 문자나 기호들을 컴퓨터가 이용할 수 있는 신호로 만드는 것을 말한다. 넓은 의미의 컴퓨터는 이러한 신호를 입력받고 처리하는 기계를 뜻하며, 신호 처리 시스템을 통해 이렇게 처리된 정보를 사용자가 이해할 수 있게 된다.[2] 이 신호를 입력하는 인코딩과 문자

어떤 정보를 정해진 규칙(Code[1])에 따라 변환하는 것(en-code-ing)을 일컫는다.

ASCII 인코딩

데이터 인코딩, 어떻게 사용하는지?

| Label Encoding |

1. **사이킷런 preprocessing**
2. **판다스 factorize**

```
from sklearn.preprocessing import LabelEncoder

df['column_1'] = LabelEncoder().fit_transform(df.column_1)
```

```
for c in col_obj:
  df[c], y_class = pd.factorize(df[c])
```

fit(): 데이터 변환을 위한 기준 정보 설정을 적용
transform(): 데이터 변환

pd.factorize()는 인코딩한 값과 인코딩된 값 반환

데이터 인코딩, 어떻게 사용하는지?

| One-Hot-Encoding |

1. **사이킷런 preprocessing**
2. **판다스 get_dummies**

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder

labels = LabelEncoder().fit_transform(df.column_2).reshape(-1,1)
ohe = OneHotEncoder().fit_transform(labels)
```

```
import pandas as pd

ohe = pd.DataFrame({'column_2':column_2})
pd.get_dummies(ohe)
```

또 다른 인코딩

- **Ordinal Encoding**
- Binary Encoding
- Frequency Encoding
- Etc.

LabelEncoding: 알파벳 순서대로, 0부터

Never	Sometimes	Often	Usually	Always
1	3	2	4	0

OrdinalEncoding: 변수 순서를 유지, 1부터

Never	Sometimes	Often	Usually	Always
1	2	3	4	5

```
Freq = {'Never':1,'Sometimes':2,'Often':3,'Usually':4,'Always':5}
df['freq'] =  df.Frequency.map(Freq)
```

또 다른 인코딩

- Ordinal Encoding
- **Binary Encoding**
- Frequency Encoding
- Etc.

OneHotEncoding: 벡터 사용

Never	Sometimes	Often	Usually	Always
[1,0,0,0,0]	[0,1,0,0,0]	[0,0,1,0,0]	[0,0,0,1,0]	[0,0,0,0,1]

BinaryEncoding: 이진수 사용

Never	Sometimes	Often	Usually	Always
0000	0001	0010	0011	0100

```
Import category_encoders as ce
Bi = ce.BinaryEncoder(cols=['Freq']).fit_transform(df['Freq'])
df = pd.concat([df,Bi],axis=1)
```

또 다른 인코딩

- Ordinal Encoding
- Binary Encoding
- **Frequency Encoding**
- Etc.

LabelEncoding: 정수, 데이터 수와 무관

Apple	Banana	Cat	Cat	Frog
0	1	2	2	3

FrequencyEncoding: 실수, 데이터 수와 연관

Apple	Banana	Cat	Cat	Frog
0.2	0.2	0.4	0.4	0.2

```
Fe = df.groupby('Alphabet').size()/len(df)
df.loc[:,'Alphabet'] = df['Alphabet'].map(Fe)
```

감사합니다.