데이터 전처리

Scaling

변수의 크기가 너무작거나 클경우 해당 변수가 target에 미치는 영향력이 제대로 표현 X

Min-Max Scaling

값의 범위를 0~1사이로 변경

```
# num_columns => 숫자형 변수
from sklearn.preprocessing import MinMAxScaler
MS = MinMaxScaler()
numeric_data = data[num_columns].values
MS.fit(numeric_data)

MS_data= MS.transform(numeric_data)
MS_data = pd.DataFrame(MS_data, columns=num_columns)
```

Standard Scaling

표준정규분포화를 시킴 평균이 0, 표준편차가 1이 되도록 스케일링

```
from sklearn.preprocessing import StandardScaler
Sd = StandardScaler()
Sd.fit(numeric_data)
Sd_data = Sd.transform(numeric_data)
Sd_data = pd.DataFrame(Sd_data, columns=num_columns)
```

Inputation(결측값 대체)

```
pd.inna(data).sum() # 결측값의 개수

# 평균으로 대체
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
imputer.fit(mean_df[num_columns])
mean_df[num_columns] = imputer.transform(mean_df[num_columns])
# 중앙값으로 대체
imputer = SimpleImputer(strategy='median')
```

```
imputer.fit(median_df[num_columns])
median_df[num_columns] = imputer.transform(median_df[num_columns])
# iterative impute
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
imp_mean = IterativeImputer(random_state=0)
impute_df[num_columns] = imp_mean.fit_transform(impute_df[num_columns])
# 최빈값
imputer = SimpleImputer(strategy='most_frequent')
imputer.fit(mode_df[cat_columns])
mode_df[cat_columns] = imputer.transform(mode_df[cat_columns])
```

Categorical Variable to Numeric Variable

범주형 변수를 수치형 변수로 변환

Label Encoding

n개의 범주형 데이터를 0~n-1의 연속적인 수치 데이터로 표현

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
le.fit(label)
le.classes_
label_encoded = le.transform(label)
le_df = pd.DataFrame(label_encoded, columns = ['label_encoded'])

result = pd.concat([label, le_df], axis=1)
result.sort_values('label_encoded', inplace=True)
```

One-hot Encoding

n개의 범주형 데이터를 n개의 비트(0,1) 벡터로 표현

```
from sklearn.preprocessing import OneHotEncoder
ohe = OneHotEncoder(sparse=False)
ohe.fit(label)
one_hot_encoded = ohe.transform(label)
ohe_df = pd.DataFrame(one_hot_encoded, columns = ohe.categories_[0])
result = pd.concat([label, ohe_df], axis=1)
```