8-dars Report.

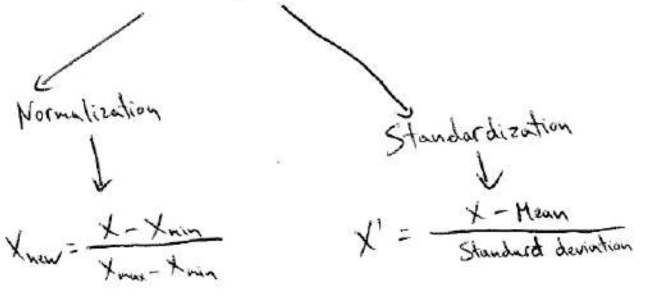
DataPreprocessing → Manual.
DataPreprocessing → Auto.

Encoding:

Label → 1-chi kelgan harf orqali raqamlashtirish.
One-hot → Ustunga ajratib raqamlashtirish.
Ordinary → ketma-ketlik (ősuvchidan → kamayuvchi,
kamayuvchidan - ősuvchi)
Target
Frequency

Data Preprocessing: Review

- Handling missing values:
    - Mean (Ma'lum bir ustunning őrta arifmetik qiymati b/n tőldirish)
    - Mode (Eng kőp takrorlangan bilan tőldiradi)
    - Median (Eng őrtasidagi qiymat bilan tőldiradi)
    - Fixed (őzimiz bilgan holatda raqam/harf/sőz b/n tőldiradi)
    - Drop (Qator yoki Ustun bőyicha tashlab yuboradi)
        - ↳ 50% dan kőp tushirib qoldirilgan qiymat bőlsa
        - ↳ Tushirib qoldirilgan qiymat juda oz bőlsa
- Encoding
    - One-hot (Sun'iy ustun yaratib classlar bőyicha ustunlarni chiqaradi)
    - Label (Alifbo tartibida (ketma-ket) → raqamlashtirib beradi.
    - Frequency (Nechmarta takrorlangan bőlsa, ősha ustun classlar soniga nisbati bilan olinadi)
    - Target (Nechmarta takrorlangan bőlsa, tőgrisida turgan target qiymatni őrta arifmetikasi bilan olinadi)
    - Ordinal (Object bőlishi kerak / ketma-ketlik bőlishi kerak)
- Scaling (Juda katta va juda kichkina qiymatlarni balanslashtirish uchun ishlatilinadigan Data Preprocessing bőlimi)
→ (Numerical bőlishi kerak / [0,1] qiymat bőlsa őzgartirmaymiz /)
    - Standard Scaler $(z = (x - \mu)/\sigma)$
    - MinMaxScaler (X_scaled = (X - X_min)/(X_max - X_min) [0;1] ga olib bora
    - Robust Scaler (X_scaled = (X - median) - IQR


Feature scaling
Normalization
Standardization
$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$
$X' = \frac{X - Mean}{Standard\ deviation}$

☆ Target qiymatni ~~Scaling~~ qilmaymiz!!!

☆ Scaling kutubxonasini chaqirib olish:

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler, RobustScaler (LabelEncoder)
(encoder = LabelEncoder)                print(df.dtypes)
(encoder
num_col = df.select_dtypes(include=['int64', 'float64']).columns.drop('Target values col')
                                          # izoh: Target qiymat scaling qilinmaydi:

num_col
```

Scaling:

1) StandardScaler: (manfiy/musbat)

```
        scaler = StandardScaler()
        scaler
    df['Column'sname'] = scaler.fit_transform(df[['Column's name]])
```

→ Agar ustun nomi (1ta) bo'lsa (2ta) qaus ([[ma]]) ochish kerak.

2) MinMaxScaler: (faqat musbat son chiqadi)

```
        scaler = MinMaxScaler()
        scaler
    df['Column's name'] = scaler.fit_transform(df[['Column's name']])
```

3) RobustScaler

```
        scaler = RobustScaler()
         scaler
    df['Column'sname'] = scaler.fit_transform(df[['Column'sname']])
```

For loop + Scaling

- Jarayonni avtomatlashtirish uchun ishlatiladi.
- Umumiy yoki gruppaviy bolib scaling qilish uchun ishlatiladi.

1) scaler = MinMaxScaler()

2) scaler = StandardScaler()

3) sealer = RobustScaler()

(Scaling I-usul)

```
for col in df.columns:
    if df[col].dtype != 'Object':
        df[col] = scaler.fit_transform(df[[col]])
```

```
~ num_col = df.select_dtypes(include=['int64', 'float64']).columns.drop('Survived')
~ num-col
~ df[num-col] = scaler.fit_transform(df[num-col])
~ for col in num_col:
        df[col] = scaler.fit_transform(df[[col]])
```

(Scaling II-usul)

```
num_col = df.select_dtypes(include=['int64', 'float64']).columns.drop('Column'sname')
for col in num_col:
    df[col] = scaler.fit_transform(df[[col]])
```

(III docs)