

8- dars uchun -Report

9- darsda biz Data Preprocessing uchun ta'rif berdik; **Data Preprocessing** bu ma'lumotarni **tayyorlab olish protsesi** ya'ni **kompyuterni ma'lumotlarni tushuna** oladigan qilib olish jarayonidir.

Ma'lum bir bosqichlarni tashlab ketsak ham ma'lum bir algoritmlarda kodlarimiz bexato ishlashi mumkin, ammo ikta xolatda to'xtashi mumkin bu 1) Ko'd o'zi to'xtaydik va ekranga yozuv chiqadi- **Error** degan ya'ni senda tushib qoldirilgan qiymat bor deya aniqrog'i **NaN** bor deydi va to'g'irla deydi.

2) Holatda esa ba'zi algoritmlar-yana ham aniqroq aytganda kuchliroq algoritmlar tushurib qoldirilgan qiymatlarni ham farqi yo'q ishlayveradi. Ammo, tushirib qoldirilgan qiymatlardagi joyi kemtik bo'lib qolaveradi. Ya'ni o'sha joyidan muammo yuza chiqib kelaveradi. Bu esa pirovardida **prediction** qilish qismida yoinki natijinani olish qismida bizga kutilgandek natija qaytarmaydi. Shuning Data Preprocessing qismi juda muhim bosqichlardandir.

Scaling - Juda katta va juda kichik qiymatlar ortasidagi balansga olib kelishdagi Data Preprocessing bo'limidir. Bunda ustunlar numerical bo'lishi va qiymatlarimizni agar [0,1] oraliqda bo'lsa scaling qilish shart emsaligiga alohida e'tibor beramiz. Scaling qilishdan oldin va keyin ham `df.head()` qilishimiz maqsadga muvofiq sababi xato qilmabmizmi yoki yo'qmi, ko'dimiz ishladiimi yoki yo'qmi degan muhim savollarga javob olamiz.

```
from sklearn.preprocessing import StandardScaler,  
MinMaxScaler, RobustScaler
```

Shu orqali 3 xil scaling classlarini import qilib olamiz.

```
num_col = df.select_dtypes(include=['int64',  
'float64']).columns.drop('Personal Loan')
```

Izoh: --> Target qiymat scaling qilinmaydi.

Bu yerdagi asosiy e'tibor berildigan jabhalardan biri bu numerical qiymatlarni o'zingizning datasetingizdan kelib chiqib **dtipelarni** yozasiz **include** ichiga. Masalan, int64, float64 va hokazo. Va har birini bittadan yozsangiz kifoya. Shuningdek, Target qiymat scaling qilinmaydi.

```
scaler = StandardScaler()  
scaler  
df['Age'] = scaler.fit_transform(df[['Age']])
```

df.head() # Bu yerda StandardScaler bo'lgani sabab bizda manfiy tarafga ham o'tkazib yubordi.

Bu yerda iktalik qovus ichiga agar faqat bir ustunni scaling qilmoqchi bo'lsak yozamiz. Agar 2 va undan ortiq ustunlarni scaling qilmoqchi bo'lsak 1 ta to'rtburchak qovus kifoya qiladi.

```
scaler = MinMaxScaler()  
scaler  
df['ZIP Code'] = scaler.fit_transform(df[['ZIP Code']]) #  
Python case sensitive va asosan musbat ko'rinishda [0,1]  
orqaliqqa olib keladi.
```

```
# RobustScaler
```

```

scaler = RobustScaler()
scaler
df['Family'] = scaler.fit_transform(df[['Family']])
df.head() # Bu yerda Familyni ham manfiy ham musbat
tomonga o'tkazib yuboradi.

```

For Loop + Scaling;

Bunda biz jarayonni **avtomatlashtirish** uchun ishlatamiz va **umumiy** yoki **guruhlarga** bo'lib **scaling** qilishimiz uchun ishlatiladi.

```

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

```

Yoki

```

df[num_col] = scaler.fit_transform(df[num_col])
#numerical columns orqali scaling qilish
num_col = df.select_dtypes(include=['int64',
'float64']).columns.drop('Personal Loan')
# Izoh: --> Target qiymat scaling qilinmaydi.
for col in num_col:
    df[col] = scaler.fit_transform(df[[col]]) # bu ham
bir variant scaling qilish uchun

```

Va so'ngida biz data structure haqida gaplashdik bunda data(Raw Data) - ya'ni fan tilida hom ashyo undan so'ng Data bilan tanishuv bunda bazi predictive kuchga ega bo'lmagan namelarni tashlab yuborsak ham bo'ladi. Data Preprocessing (Handling missing values, Encoding va Scaling), undan so'ng Model selection (algorithm -> method) va uning datadan kelib chiqib qaysi biri mos kelishi masalan Linear uchun Linear Algorithm, Non-Linear uchun esa Complex Algorithms ishlatilishi kerak bo'lishligi haqida ma'lumot berildi.