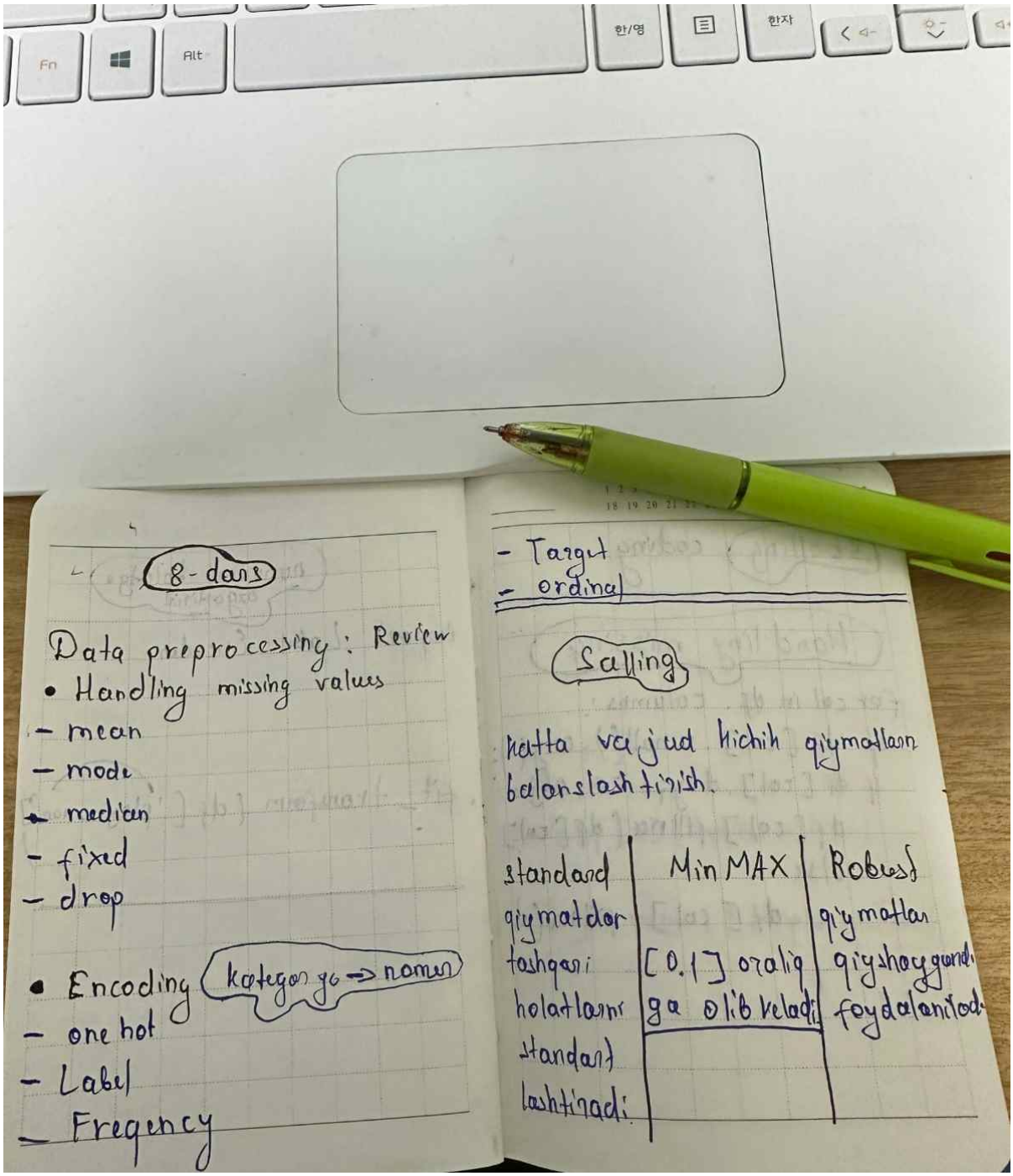


8-dars

Data preprocessing: Review

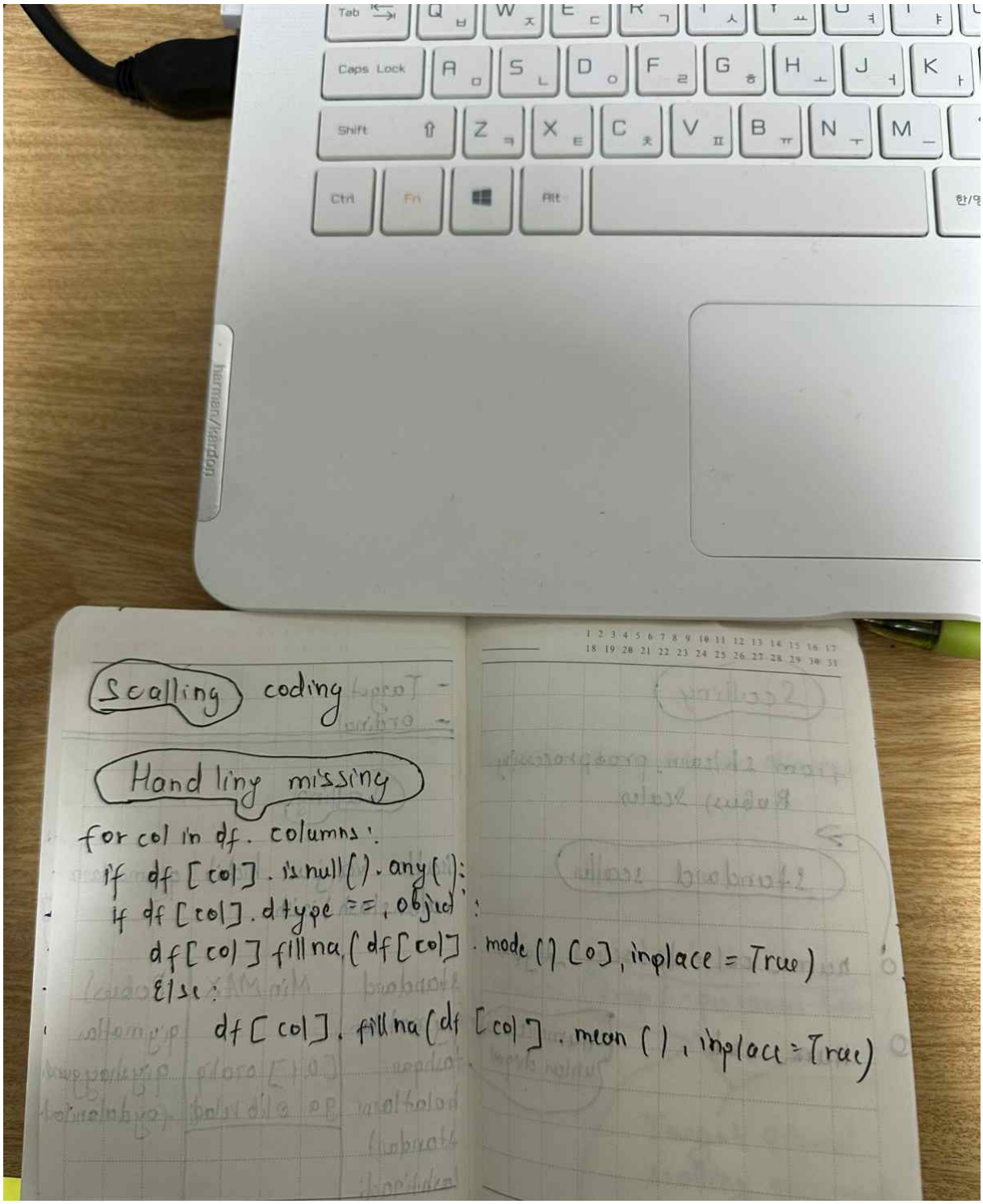
- Handling missing values
  - mean
  - mode
  - median
  - fixed
  - drop
- Encoding (kategoryalarni → nomer)
  - one hot
  - Label
  - Fregency

- Target
- ordinal

Scaling

hatto va juda kichik qiymatlarni balanslashtirish.

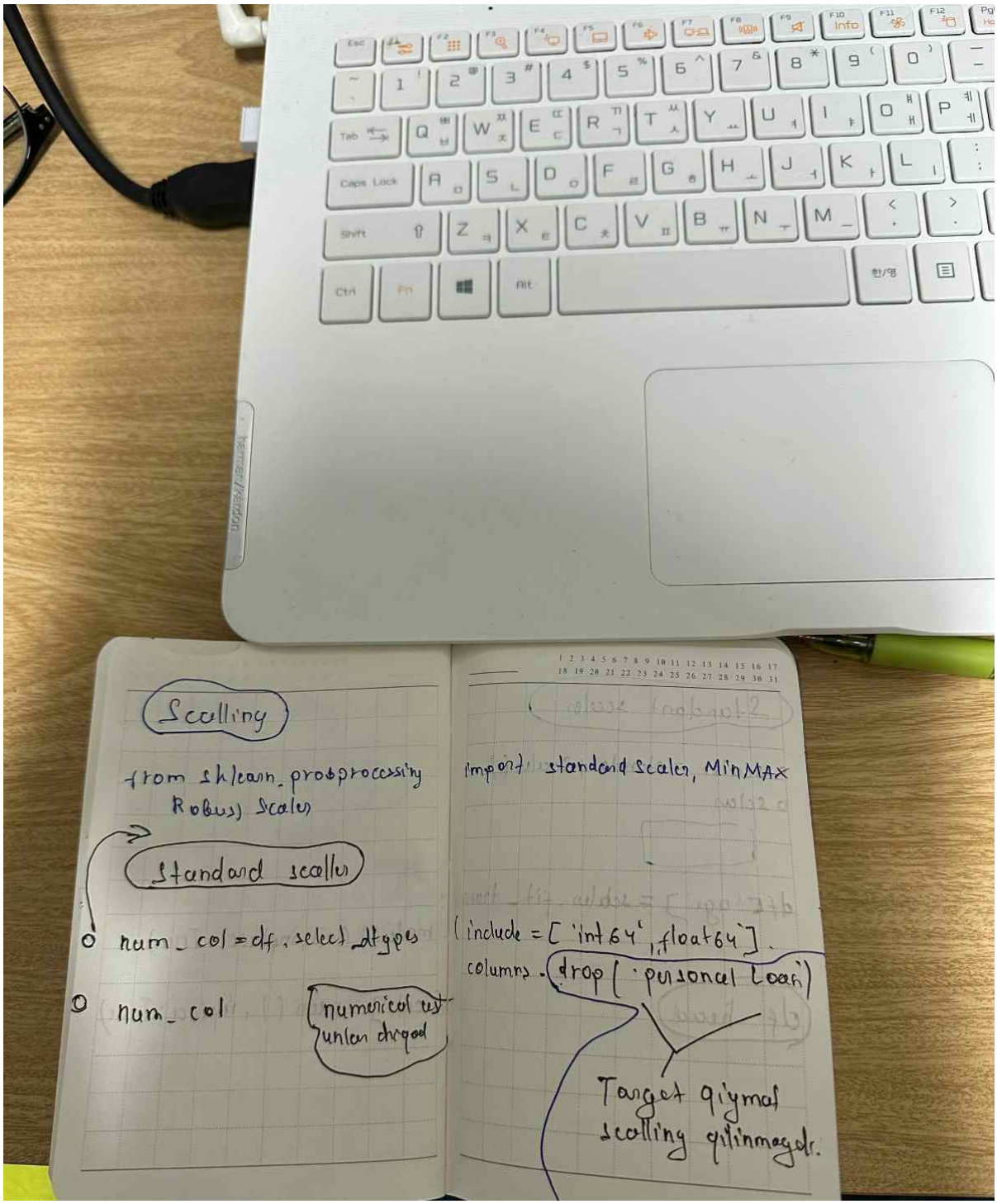
standard	Min MAX	Robust
qiymatdor tashqari holatlarni standart lashtiradi	[0,1] oraliq ga olib keladi	qiymatlar qiyshaygandi foydalaniladi

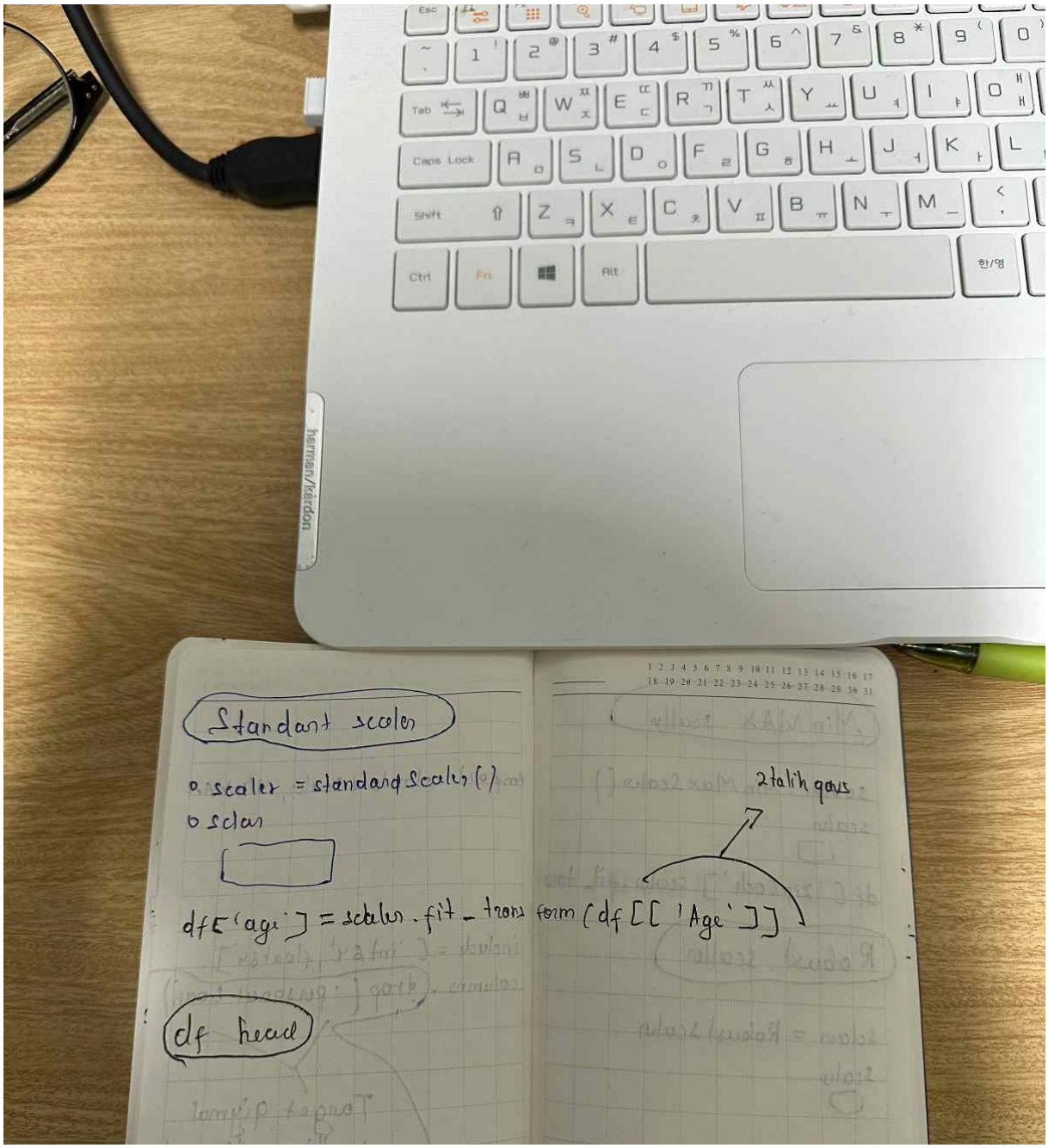


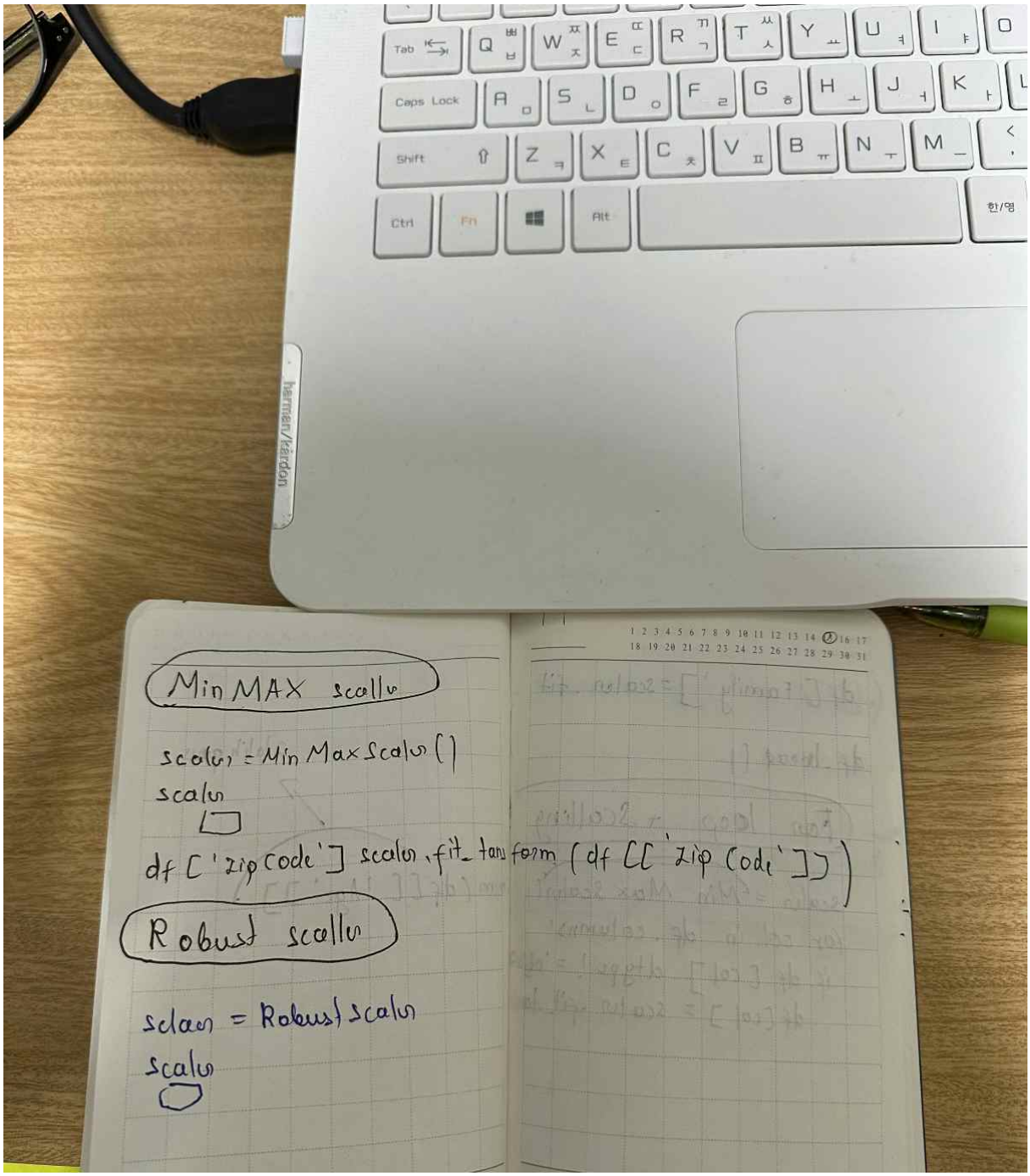
Scalling coding

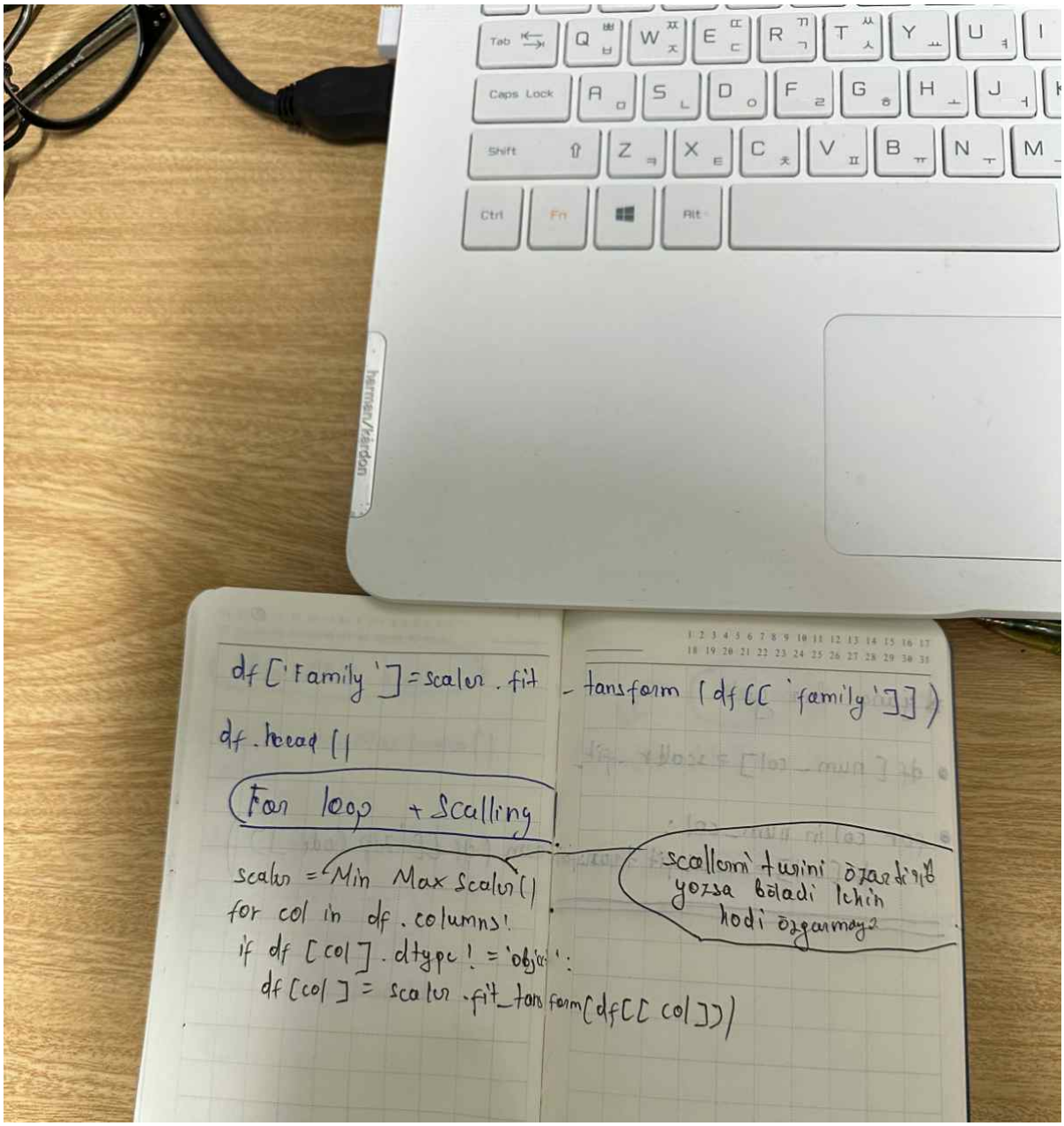
Handling missing

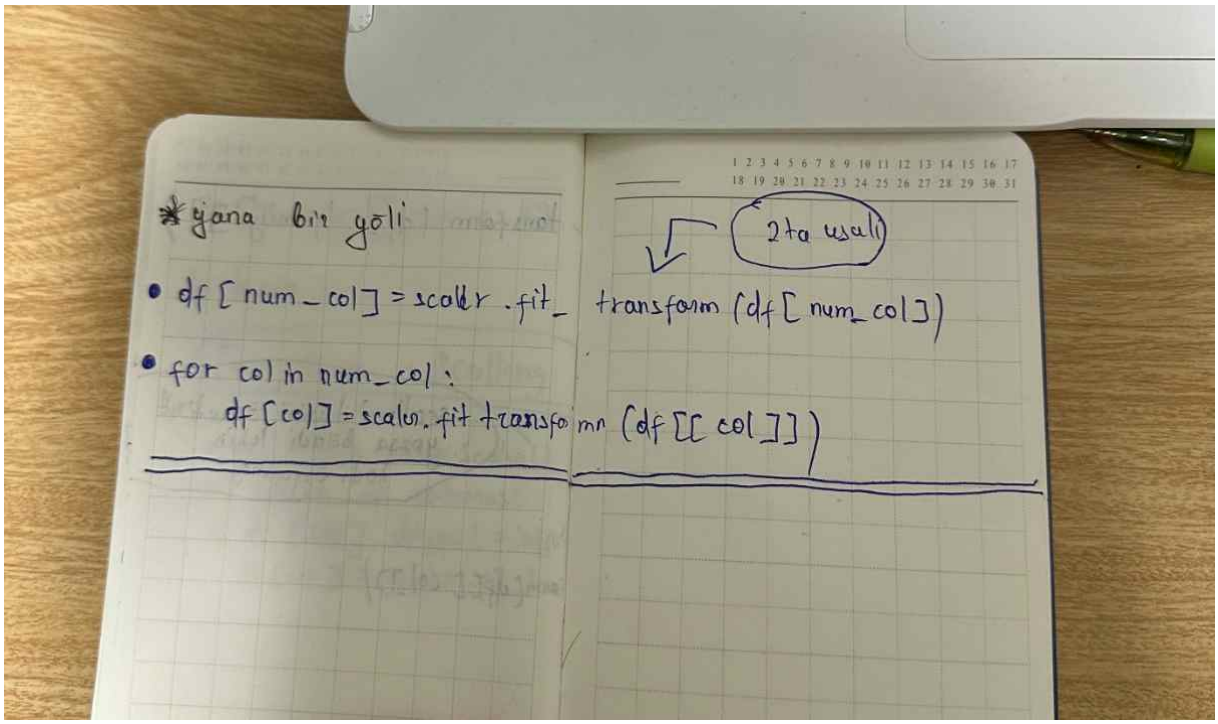
```
for col in df.columns:
  if df[col].isnull().any():
    if df[col].dtype == 'object':
      df[col].fillna(df[col].mode()[0], inplace=True)
    else:
      df[col].fillna(df[col].mean(), inplace=True)
```


Scalling
from sklearn.preprocessing
Robust Scaler
Standard scaller
num_col = df.select_dtypes
num_col
numerical ust
unlar dropod
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31
import standard Scaler, MinMAX
(include = ['int64', 'float64'].
columns .drop('personal Loan')
Target qiymat
scalling qilinmagak.


Standart scaler
scaler = standardScaler()
scalar
df['age'] = scaler.fit_transform(df[['Age']])
2 talih qavs
df head


Min MAX scaller
scaler = MinMaxScaler()
scaler
df['zipcode'] scaler.fit_transform(df[['zipcode']])
Robust scaller
sclaer = RobustScaler
scaler


df['Family'] = scaler.fit - tansform (df[['family']])
df.head()
For loop + Scalling
scaler = MinMaxScaler()
for col in df.columns:
if df[col].dtype != 'obj
df[col] = scaler.fit_tansform(df[[col]])
scalerni tepini ozgartirib yozsa boladi lekin kodi ozgarmaydi


yana bir yoli
2ta usuli
df [num_col] = scaler.fit_ transform (df [num_col])
for col in num_col:
df [col] = scaler.fit transform (df [[col]])