MACHINE LEARNING HANDS-ON GUIDE

Stratified Sampling & Test Set Creation

Ինչպես ճիշտ բաժանել տվյալները և պահպանել ներկայացուցչական բաշխումը

Random vs. Stratified Split

Pure Random

Կարող է խախտել կարևոր հատկանիշների բաշխումը train և test բաժանումներում:

Stratified Sampling

Ապահովում է, որ հիմնական կատեգորիաների հարաբերակցությունը խստորեն պահպանվի:

Pipeline-ի Հիմնական Քայլերը

1. Median Income

Շատ կարևոր անդադար փոփոխական:

2. pd.cut()

Ստեղծում ենք կատեգորիաներ (income_cat):

3. Split

Իրականացնում ենք stratified բաժանում:

4. Cleanup

Հեռացնում ենք ժամանակավոր income_cat-ը:

Ինչպես է աշխատում split()-ը

splitter.split()	Վերադարձնում է index-ների հավաքածու (train_index, test_index)
housing_full.iloc[]	Ստանում ենք իրական DataFrame-ները

Ինչու՞ n_splits = 10

Մի քանի Split-ներ

Ստեղծում է 10 տարբեր stratified train/test տարբերակներ նույն dataset-ից:

Cross-Validation

Նախապատրաստություն է հաջորդ գաղափարներին՝ մոդելի միջին performance-ը գնահատելու համար:

Ավելի Կարճ Ճանապարհ

train_test_split() ֆունկցիան

Կարիք չկա ձեռքով գրել ամբողջ StratifiedShuffleSplit-ը, եթե պետք է միայն մեկ split:

stratify = housing_full["income_cat"] parameter-ը անում է ամբողջ աշխատանքը:

Արդյունքների Ստուգումը

Category 3
~35%

Category 2
~32%

Category 4
~18%

Category 5 & 1
11% & 4%

Ինչու՞ ենք ջնջում income_cat-ը

Ժամանակավոր սյունակ

Այն ստեղծվել էր բացառապես stratified sampling անելու և բաշխումը վերահսկելու համար:

Մաքրում split-ից հետո

train-ից և test-ից հեռացնում ենք այն: Իսկ բնօրինակ median_income-ը մնում է անփոփոխ:

Test Set-ի ստեղծումն ավարտված է ✅

Հաջորդ փուլ՝ Explore and Visualize the Data to Gain Insights (տվյալների վերլուծություն և նոր օրինաչափությունների բացահայտում):