8-dars: Model Building-Report

K-cross validation: o'z-o'zidan k-ta degani, son keladi o'rniga. K-cross validation-k bo'lakli o'zaro tekshiruvdir. Modelimizni yaxshi ishlayaptimi yoki yo'qligini bahalaymiz ekan. Ya'ni bu **metriclarga** yaqin ekan.

L-Fold cross validation: **Supervised Machine Learning** model **unseen** dataset bilan qanday ishlayapti (yaxshi yoki yomon) baxolash usulidir.

Masalan, o'qituvchi o'rgatdi: o'quvchi imtixonda qanday javob berayapti: ya'ni o'qituvchi savollarni chalkashtirib va o'quvchi ko'rmagan savollar shaklida beradigan bo'lsa qanday ishlayapti yaxshimi yoki yomon o'shani baxolaymz ekan.

Generalization - umumiylashtirish: bu modelning -yangi, ko'rilmagan ma'lumotlarda ham yaxshi ishalsh qobilyati.
Ya'ni model faqat **train** datasetni yodalb olmasligi kerak, balki real hayotdagi datada ham to'g'ri natija berishi kerak.

Bu nega muhim?
1) Trainda juda yaxshi
2) Testda juda yomon --> Bu esa overfitting bo'ladi.
Misol: Agar men matematikani faqat eski test savollarini yodlab o'rgansam, yangi savollarda qiynalaman. Lekin tushunib o'rgansam-har qanday savolga javob topa olaman --> bu **Generalization.**

Classification --Model ma'lumotni kategoriya (class) ga ajratadi.
Natija odatda:
- **0** yoki **1**
- **Ha** yoki **yo'q**
- **Spam** yoki **Not Spam** --> Misol: **Email spam**

Regression (uzliksiz son bashorat qiladi) **--** misol uyni narxini predict qilishimiz kerak, bizda xonalar soni, etajlar, maktabga yaqinmi yo'qmi, metroga yoki katta yo'lga yaqinmi yoki yo'qmi, shunga o'xshagan featurelarni berayapmizda demak uyni narxini o'z-o'zidan o'sishi kerak. Xonasini o'lchami katta bo'lsa model o'ylaydi ha bo'ldi demak bu xonani o'lcahmidan, qulayligi jihatidan narxi shunga qarab oshayotkaninga mantiqan umumiylashtiradi va bu xolat ayni--**Generalization** deyiladi.

Demak, cross-validation modelni **generalization** qobilyatini o'lchar ekan, umumiylashtirish qobilyatini o'lchar ekan. Chunki o'sha **generalization** bo'lmasa tasavvur qilayapsizlarmi bir 80% train qilganimizda, bizda **accuracy** 90% chiqdik deylik. Keyin esa qolgan 20% ni ko'rmaganku, 20% beradigan bo'lsak **accuracy** 70% chiqdi. Bu huddi yaxshi-yomon, oq-qora kami antominlarga qiyoslasask bo'ladi. **Generalziation (yaxshi, oq)** <--> **Overfitting (yomon, qora).**

Maqsadimiz **modelni generalziation qobilyatini oshirishim** ekan uning qobilyatini oshirish va **yaxshi generalization qilyaptimi**, **yaxshi umumiylashtiryaptimi** yoki yo'qmi o'shani o'lchaydigan usul bizda **cross_validation** ekan.

Cross Validation turlari:
1) Fold cross-validation

2) Stratified K-Fold Cross Validation
3) Leave -One-Out Cross Valdiation (LOOCV)
4) Leave -p-out Cross Validation

Single split : **x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 42)**

k-fold split : Datasetni k-ta qismlarga ajratadi, train k-1 folds va Test (qolganini qiladi) ya'ni datasertni k-ta qismiga bo'lib o'sha k dan -1 ni ayrigandan qolgani trainingga 1 tasi esa testga o'tadi.

from sklearn.model_selection import KFold, cross_val_score
kf = KFold(n_splits = 5, shuffle = True, random_state = 42) --> bu yerda n ta qismlarga bo'l, va ularni randomly aralashtir, va model stabill ishlashi uchun random_statega 42 ber.
scores = cross_val_score(lr, x, y, cv=kf, scoring='r2')

Undan so'ngra biz scoresning o'rtacha qiymatini, masivning standard og'ishi ya'ni ma'lumotlar o'racha qiymatdan qanchalik tarqalganligini (yoyilganligini) ko'rsatadi.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}}$$

Bu yerda:

σ — population standard deviation
N — elementlar soni
x_i — har bir qiymat
μ — population mean
$\sum$ — yig'indi

(Qisqa qilib: Natijalar o'rtachadan qanchalik uzoqlashlashganini ko'rsatadi)

Endi esa biz agar Regressionda ham foydalanamiz desak bizga **make_scorer** classi kerak bo'ladi.
from sklearn.metrics import make_scorer
mae = make_scorer(mean_absolute_error, greater_is_better = False)

Bu yerda **greater_is_better = False** bo'ldi sababi bizga kichigi muhim ya'ni error qanchalik kichik bo'lsa shuncha yaxshi. Buning default qiymatda **True** shuning uchun **False** qildik.

scores = cross_val_score(lr, x, y, cv=kf, scoring=mae) bu yerda biz mexmon va mezbon sifatida eslab qolsak bo'ladi ya'ni **mae** ni chaqirib olganimiz uchun '' siz yozdik. Natijada odatda manfiy ham chiqadi unda **print(-scores)** --> bu esa manfiyni musbat qiladi. **print(-scores.mean()) -->** shu orqali o'rtacha arifmetik qiymat topiladi.