### Model Evaluation nima?

Model Evaluation — bu mashinani o‘rganish (machine learning) jarayonida qurilgan modelning samaradorligini, aniqligini va umumlashuv (generalization) qobiliyatini baholash jarayonidir. Bu bosqich modelning haqiqiy hayotdagi ma’lumotlarga nisbatan qanday ishlashini aniqlab beradi.Model Evaluation nima uchun kerak? 1) Modelni tekshirish O‘rgatilgan model yangi, ko‘rilmagan ma’lumotlarda to‘g‘ri ishlayaptimi yoki yo‘qmi aniqlanadi.2) Overfitting/Underfittingni aniqlash Model juda ko‘p yoki kam o‘rganib qolganini bilish uchun.3)Modellarni taqqoslash: Turli algoritmlar yoki parametrlar natijalarini solishtirish.3)Optimal parametr tanlash: Hyperparameter tuning orqali eng yaxshi natijani olish. Baholash mezonlari Classification uchun --> Accuracy,Precision,Recall,F1-score,

Regression uchun --> Mean Absolute Error(MAE),Mean Squared Error(MSE),Root Mean Squared Error(RMSE),R² Score (Coefficient of Determination)

### Decision Tree qanday ishlaydi? Over/Underfittingni DT da qanday oldini olsak bo’ladi?

Decision Tree (qaror daraxti) — bu mashinani o‘rganishdagi nazoratli o‘rganish (supervised learning) algoritmi bo‘lib, u ma’lumotlarni “if-else” qoidalar asosida sinflarga ajratadi yoki qiymatlarni bashorat qiladi. Daraxt tuzilmasida har bir ichki tugun (node) bitta xususiyat (feature) bo‘yicha qaror chiqaradi, shoxlar esa bu qarorning natijasiga ko‘ra ma’lumotni bo‘lishni ifodalaydi. Yaproq (leaf) tugunlar esa modelning yakuniy javobini, ya’ni klass yoki qiymatni beradi.

Model ishga tushirilganda, u ma’lumotlar ustida entropiya yoki Gini impurity kabi mezonlardan foydalangan holda bo‘linmalar hosil qiladi. Maqsad — har bir bo‘linmadan so‘ng klasslar imkon qadar aniq ajralgan bo‘lishi. Bu jarayon rekursiv tarzda davom etadi, ya’ni har bir kichik bo‘lak yana o‘z navbatida kichikroq bo‘laklarga ajraladi, toki daraxt to‘liq qurilmaguncha.

Biroq, Decision Tree’lar overfitting (ortiqcha moslashish) muammosiga juda moyil. Ya’ni, model trening (o‘rgatuvchi) ma’lumotlarga juda yaxshi moslashadi, lekin yangi ma’lumotlarga nisbatan noto‘g‘ri bashorat qiladi. Shuningdek, underfitting (kam moslashish) ham yuz berishi mumkin, bu holatda model ma’lumotlardagi naqshlarni yetarli darajada o‘rganmaydi.Ushbu muammolarni oldini olish uchun quyidagi usullar qo‘llaniladi:Max depth: Daraxt chuqurligini cheklash orqali ortiqcha tafsilotlardan qochiladi.Min samples split: Har bir tugunni ajratish uchun kerak bo‘lgan minimal namunalar soni belgilanadi.Min samples leaf: Har bir yaproqda qolishi kerak bo‘lgan minimal namunalar soni belgilanadi.Pruning (budash): Daraxt qurilganidan so‘ng, ahamiyatsiz yoki haddan tashqari nozik tugunlar kesib tashlanadi.Ensemble usullari: Bir nechta qaror daraxtlarini birlashtiruvchi Random Forest yoki Gradient Boosting kabi metodlar yordamida umumiylashuv kuchi orttiriladi.Xulosa qilib aytganda, Decision Tree’lar sodda va tushunarli modellardir, ammo ularning haddan tashqari murakkablashuvi yoki soddaligi natijasida overfitting yoki underfitting yuz berishi mumkin. Shu sababli, model qurishda tegishli cheklovlar va nazoratlar orqali balansni ta’minlash muhimdir.

## Linear Modellar qanday ishlaydi?

**Linear model** — bu kompyuterga o‘rgatadigan oddiy formulali model bo‘lib, u raqamli ma’lumotlar orasidagi chiziqli bog‘liqlikni topishga harakat qiladi.

Modelning ishlash mantig‘i shundayko Har bir xususiyat (masalan: yosh, tajriba, bal) muhimlik darajasi bilan ko‘paytiriladi va natijada umumiy javob chiqadi

Formulasi oddiy ko‘rinishda: y^=w1×x1+w2×x2+⋯+wn×xn+b  
x — kirish ma’lumotlari (masalan, inson yoshi, ish tajribasi),  
w — har bir xususiyatning og‘irligi (modelga qanchalik ta’sir qilishini bildiradi),  
b— boshlang‘ich qo‘shiladigan raqam (intercept yoki bias),  
y​ — modelning natijasi, ya’ni bashorati.  
Misol:Aytaylik sizda quyidagi ma’lumot bor:  
"Kv.metr soni" va "xonalar soni" orqali uy narxini topish.  
Model o‘rganadi: "Har 1 kv.m = 1 million so‘m, har 1 xona = 5 million so‘m"  
Formulasi bo‘ladi: Yangi uyni baholamoqchi bo‘lsangiz, shu formulaga qiymatlarni qo‘yib hisoblaydi. Model nimani o‘rganadi? Model shunchaki bu sonlar (og‘irliklar w)ni to‘g‘ri aniqlashni o‘rganadi. Qanchalik yaxshi aniqlasa, shunchalik aniq bashorat qiladi. Xulosa qilib aytsak Linear model — bu juda oddiy va tushunarli model. U biz beradigan raqamli ma’lumotlar orqali qanday bog‘liqlik borligini topadi va shunga qarab yangi natijalarni hisoblaydi. U tez ishlaydi, lekin faqat chiziqli munosabat bo‘lsa yaxshi ishlaydi.  
  
Logistic Regressionda qanday funksiyadan foydalanamiz?  
  
Logistic Regression modelida sigmoid funksiyasi (yoki logistic function) ishlatiladi. Bu funksiya model chiqishini 0 va 1 oralig‘idagi ehtimollik sifatida beradi. Sigmoid funksiyaning formulasi quyidagicha:σ(z)=1+e−z1 Sigmoid funksiyasi klassifikatsiya uchun ishlatiladi: Natija 0.5 dan katta bo‘lsa → 1-klass (ha) ,Natija 0.5 dan kichik bo‘lsa → **0-klass** (yo‘q)

Modelni stabilligini taminlash usullari xaqida gapiring?

Modelning stabilligi — bu model har xil ma’lumot to‘plamlari yoki kichik o‘zgarishlar bilan ishlaganda barqaror va ishonchli natijalar berishini anglatadi. Quyidagi usullar modelning stabilligini oshirishda qo‘llaniladi:  
Cross-Validation Modelni baholash uchun faqat bitta train/test bo‘linmasiga tayanmaslik. Ayniqsa,   
K-Fold Cross Validation orqali turli bo‘linmalarda modelni sinab ko‘rish — stabillikni oshiradi.  
Regularization L1 (Lasso) yoki L2 (Ridge) regularizatsiya yordamida modelning og‘irliklarini nazorat qilish orqali overfittingni kamaytirish va modelni umumiylashtirish qobiliyatini yaxshilash mumkin.  
Ensemble usullari Bir nechta modelni birlashtiradigan usullar (masalan, Bagging, Random Forest, Boosting) stabillikni oshiradi, chunki ularni kombinatsiyasi bitta modeldan yaxshiroq ishlaydi.   
Feature selection va engineeringFazoviy shovqinsiz va mantiqiy xususiyatlarni tanlash modelni noaniqlikdan saqlaydi. Shuningdek, noto‘g‘ri yoki ortiqcha xususiyatlar modelni beqaror qiladi.  
Hyperparameter tuning Modelning parametrlarini Grid Search yoki Random Search orqali ehtiyotkorlik bilan tanlash orqali stabillik va ishlash samaradorligi ortadi.  
Data normalization/standardization Xususiyatlar bir xil miqyosda bo‘lishi uchun ularni normalizatsiya qilish (masalan, MinMaxScaler yoki StandardScaler) modelni yaxshiroq o‘rgatadi va barqarorlashtiradi.  
Outlierlarni aniqlash va tozalash Ekstremal qiymatlar (outliers) modelni chalg‘itishi mumkin. Shuning uchun ularni aniqlab, ehtiyotkorlik bilan olib tashlash modelni stabillashtiradi. Xulosa qilib aytsak Model stabilligi — bu modelni haqiqiy dunyoda ishlatishga tayyor qilishdagi muhim bosqichdir. Barqaror model kichik o‘zgarishlarga nisbatan chidamli bo‘ladi va ishonchli natijalar beradi. Yuqoridagi usullarni qo‘llash orqali modelni yanada kuchli va foydali qilish mumkin.

**Robust va Standart scalingni qanday oxshashlik Tomoni bor va ular qanday ishlaydi?**StandardScaler nima va qanday ishlaydi? StandardScaler bu har bir ustunni o‘rtacha qiymati (mean) 0 ga, standart ogishi 1 ga keltiradi. Yani, barcha xususiyatlar z-score asosida standartlashadi.Formulasi: z=x−μσz

Bu yerda: x- asl qiymat, μ- ustunning o‘rtacha qiymati, σ sigma — ustunning standart og‘ishi.Kamchiligi: Outlierlar (g‘ayrioddiy qiymatlar) mavjud bo‘lsa, o‘rtacha va dispersiya buziladi → noto‘g‘ri scaling bo‘ladi.  
  
RobustScaler nima va qanday ishlaydi?

RobustScaler — bu ustunni medianasi 0 ga va IQR (interquartile range) asosida masshtablaydi. Bu metod outlierlarga sezgir emas, ya’ni mustahkam (robust). Formulasi:z=x−medianIQRz Bu yerda: median — ustundagi o‘rtadagi qiymat, IQR=Q3−Q1 — choraklar oralig‘i (75% - 25%). Afzalligi: Outlierlar ta’sir qilmaydi → stabil masshtablash.oxshashlik tomonlari: bir xil – Har bir xususiyatni bir xil miqyosga (scaling) keltirish.Xususiyatlar bo‘yicha alohida ishlaydi – Har bir ustun mustaqil masshtablanadi.Modelni tezroq va aniqroq orgatishga yordam beradi Ayniqsa, Gradient Descent, KNN, SVM kabi algoritmlarda.Outlierlarni e'tiborga olmaydi Ya'ni, har ikkisi xususiyatlarni normallashtiradi, lekin RobustScaler bu borada kuchliroq.Malumot birligiga ta’sir qilmaydinScaling faqat qiymatlarni masshtablaydi, ma’lumot mazmuni o‘zgarmaydi.

# Target Encoding va Frequency Encoding

**Target Encoding** — bu kategorik ustundagi har bir qiymatni, unga mos keluvchi target ustunning o‘rtacha qiymati bilan almashtirish usuli. Masalan, City ustuni va Income target ustuni bo‘lsa:

City | Income  
Tashkent | 3000  
Tashkent | 3200  
Samarkand| 2000

Target Encoding quyidagicha bo‘ladi:  
- Tashkent → (3000 + 3200) / 2 = 3100  
- Samarkand → 2000

Yangi ustun: City\_encoded = [3100, 3100, 2000]

Afzalliklari: Target bilan bog‘liq real axborotni modelga kiritadi.Regression modellarda ayniqsa samarali. Kamchiliklari:Overfitting xavfi bor, ayniqsa har bir kategoriya kam marta uchrasa.Cross-validation bilan ehtiyotkor ishlatish zarur.  
**Frequency Encoding**

Frequency Encoding — bu har bir kategorik qiymatni, u datasetda nechta marta uchragani asosida son bilan almashtirish.Masalan, Color ustuni bo‘lsa:  
Color |   
-------|---  
Red | 3 marta  
Blue | 2 marta  
Green | 1 marta

Encoding bo‘lib chiqadi: Color\_encoded = [3, 3, 3, 2, 2, 1]

Afzalliklari:Juda sodda va tez. Outlier yo‘q.Barcha modellarda ishlaydi (KNN, Tree, Regression).

Kamchiliklari:Target bilan bevosita bog‘liq emas, shuning uchun unchalik informativ bo‘lmasligi mumkin.Ba'zida kategoriyalar soni kam bo‘lsa, unchalik foydali bo‘lmaydi.

## 