1-dars: Model Building

1-darsda biz asosan kirish qismi o'laroq yangi oy uchun reja bularga; --**ML-Python,** --**Model Evaluation, Linear Model (Funksiyalar va List)**, --**Cross Validation, Decision Tree, Random Forest (Tree va Dictionary)**, --KNN, SVM (Set), --**Overfitting, Underfitting, Analysis va Final Exam(Tuple)lar bilan qasqacha keltirildi.**

--Model Evaluation: Classification: accuracy_score, classification_report
--Regression: r2_score, mean_squared_error, root_mean_squared_error, va **mean_absolute_error** haqida gaplashdik.

accuracy_score o'z-o'zidan modelimiz yoki o'sha algoritmimiz ishlayotkan bo'lsak demak o'sha algoritm natijani accurately(tekis) predict qilishligi. O'z-o'zidan qiymat qancha yuqori bo'lsa shuncha yaxshi bo'ladi. Misol uchun 60-70% ni hisoblaydigan bo'lsak 60-70% aniqlikda ishlayotkan bo'ladi. Aytaylik bizda classificationda qanday holda qiymatimiz javobi chiqadi. Bizda predict degan button bor javobi: yes, accuracy_score = 90% ya'ni bemor jarohat olishi 90% ga aniq bo'ldi degani.

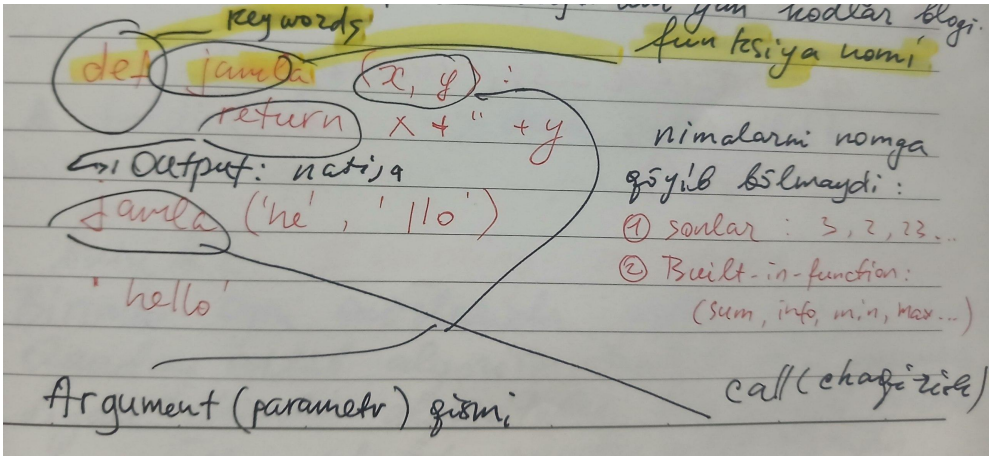
--classification_report: bu bizning modelimiz haqida umumiy hisobot bo'ladi. classification_report: accuracy_scoredan kengroqdir. Uning tarkibida: precision, recall, f1-score, sample yoki supportlar bor, ya'n bizda **yes** va **no** lar borki bu o'z-o'zidan hulosa beradiki manabu 90% ni yana ham tekshirib olish imkonini beradi. Haqiqatda ham 90% to'g'ri chiqdimi yoki noto'g'ri chiqdimi? chunki har doim ham manabu 90% to'g'ri chiqavermaydi. Masalan datasetimizning target qiymarlari **yes** va **no** lardan iborat bo'lsin classification_report bizga nechatsi **yes** va nechtasi **no** ligi haqida umumiy ma'lumot beradi.

--**Regression**: **r2_score** algoritmimiz datasetimiz qiymatlarini qau darajada yaxshi o'rgana olganiligini ko'rsatuvchi raqam hisoblanadi. Biz **training** va **predict** qilamiz, undan so'ng esa **r2_score**ni topamiz. **r2_score** huddi accuracy_score kabi qancahlik yuqorik natija bersa 0%-100% oralig'ida biz shunchalik ishonchimiz komil bo'ladiki bizning algoritmimiz yaxshi o'rgangaligi haqida. O'z-o'zidan yaxshi o'rgansa bizga **testing** qilganimizda yaxshi qiymat beradi.

--**Overfitting**: model training datani juda yaxshi yodlab oladi, ammo yangi datada yomon ishlaydi. Bunda training_accuacy: yuqori, test_accuracy: past, sabab model juda murakkab (Masalan talaba savolni o'zinigina yodlab olishi ammo yangi savolni yecha olmasligi kabi).
--**Underfitting**: Model datani yetarlicha o'rganmaydi, training va testda ham yomon ishlaydi. Bunda training_accuacy: past, test_accuracy: past (Masalan talaba umuman o'qimagan).

mse, rmse, mae -- bular barcahsi Error ya'ni xatolikni qanchalik ekanligini o'lchaydi y'ani bizda xatoli qanchalik past chqisa shunchalik natija yaxshiligini bilamiz. Linear Regressionda biz kamida bittasidan foydalanib errorlarni tekshirishimiz kerak ekan. **r2_score** bizda datasetimiz qanchalik yaxshi o'rganganligini ko'rsatadigan bo'lsa **mae, mse, rmse** lar esa bizga qanchalik xatoli bo'lganligi yoki qiymatlar qanchalik bir-biriga yaqin bo'lganligi haqida habar beradi. Masalan biz 15,000 bilan 16,000 orasidagi xatolik 1000 desak bizda yana bitta xolat bor 2 bilan 10 orasida 8 farq bor qaysi xolatda modelimiz yaxshi bo'lgan degan savolga albatta 1000 deya javob beramiz sababi 15,000 bilan 16,000 orasida 1000 farq orasi yaqinlikni beradi ammo 2 bilan 10 orasidagi 8 farq bu deyarli 5 borabar tafovvutni beradi.

--Funksiya bu qayta foydalana oladigan va biror vazifani bajara oladigan kodlar logiga aytiladi.



Biz Problem-solvingada:
1) def + boshqa
2) **def + for**
3) def + if
4) def + for + if
5) **def + data structure (lsit, dict, tuple, set, va tree)** lar kirshligi haqida ham gaplashdik.

--Function + ML qismida esa biz birinchi for loop bilan va keyin esa uni funksiya ichiga olgan holda missing valuelarni to'ldirdik.