MA'LUMOTLAR TO'PLAMINI O'QITISH, BAHOLASH VA TEST TO'PLAMLARIGA AJRATISH USULLARI

O'tkir Hamdamov[1], Botir Elov[2], Ruhillo Alayev[3]

[1]*texnika fanlari doktori. Menejment va zamonaviy texnologiyalar universiteti.*
E-pochta: utkir.hamdamov@mail.ru
[2]*texnika fanlari falsafa doktori, dotsent. Alisher Navoiy nomidagi Toshkent davlat o'zbek tili va adabiyoti universiteti.*
E-pochta: elov@navoiy-uni.uz
[3]*texnika fanlari falsafa doktori. Mirzo Ulug'bek nomidagi O'zbekiston Milliy universiteti*
E-pochta: mr.ruhillo@gmail.com

KEYWORDS

ma'lumotlar to'plamini o'qitish, training set, o'quv to'plami, validation set, baholash to'plami, test set, test to'plami, Machine Learning, ML, samaradorlik, accuracy, aniqlik, precision, eslab qolish, recall, F1-ball, F1 score.

ABSTRACT

Mashinali o'qitish algoritmlari ma'lumotlardagi shablon(qolip)larni o'rganadi va ulardan yangi ma'lumotlar bo'yicha bashorat qilish uchun foydalanadi. Ishlab chiqilgan sun`iy intellekt modellari ish faoliyatini ularning o'qitish jarayonida ko'rmagan ma'lumotlarga qarab baholash muhim. Bu ishni ma'lumotlarni taqsimlash yoki ajratish usuli orqali hal qilish mumkin. Ushbu maqolada mashinali o'qitishda ma'lumotlar to'plamini o'qitish, baholash va test to'plamlariga ajratish hamda modelni samaradorlik (accuracy), aniqlik (precision), eslab qolish(recall) va F1-ball (F1 score) orqali baholash usullari keltiriladi.

Kirish

Mashinali o'qitish (Machine Learning, ML) modelini ishlab chiqilganidan so'ng, u ilgari ko'rmagan yangi ma'lumotlarga qanchalik yaxshi umumlasha olishini baholash lozim. Mashinali o'qitshning birinchi qoidasi: modelni o'qitish va baholash uchun bir xil ma'lumotlar to'plamidan foydalanish mumkin emas. Agar ishonchli ML modelini yaratmoqchi bo'lsangiz, ma'lumotlar to'plamini **o'quv, baholash** va **test** to'plamlariga ajratishingiz kerak [1,2]. Xato ajratilgan o'quv, baholash va test ma'lumotlar to'plami asosida ishlab chiqilghan ML modeli natijalari noto'g'ri bo'ladi va siz modelning aniqligi haqida noto'g'ri tasavvurga ega bo'lasiz. Ushbu maqolada ML modeli uchun ma'lumotlarni ajratish jarayoni muhimligi va model aniqligini oshirishninh samarali usullari haqida qisqacha tushuntirish beriladi.

1.1. Modelni sinovdan o'tkazish

Nazorat qilinadigan mashinali o'qitish algoritmlari (Supervised machine learning algorithms) – sun`iy intellektning bashorat qilish va tasniflash qobiliyatiga ega vositalaridir [3,4]. Biroq, bu bashoratlar qanchalik to'g'ri ekanligini aniqlash kerak. Chunki, ishlab chiqilgan tasniflagich amalga oshirgan har bir bashorat haqiqatda noto'g'ri bo'lishi mumkin! Nazorat ostidagi mashinali o'qitish algoritmlari, ta'rifiga ko'ra, oldindan teglangan/belgilangan ma'lumotlar to'plamiga ega ekanligidan foydalanib, ushbu vazifani hal qilish mumkin. Ishlab chiqilgan algoritm samaradorligini tekshirish uchun ushbu ma'lumotlar quyidagi turlarga ajratiladi [2,5,6]:

- *o'qitish to'plami (training set);*
- *baholash to'plami (validation set);*
- *test to'plami (test set).*

1.2. O'qitish, baholash va test to'plamlari

Ma'lumotlarni o'qitish to'plami (training data) – bu algoritm o'rganadigan ma'lumotlar bo'lib, qo'llaniliadigan algoritmdan foydalanayotganingizga qarab o'qitish jarayoni tashkil etiladi [1,7]. Odatda, o'qitish to'plami ML modelini o'qitish uchun foydalaniladigan ma'lumotlarning eng katta qismidir. Ushbu ma'lumotlar kuzatish xususiyatlari va ularning tegishli chiqish belgilaridan iborat. Model ushbu ma'lumotlardan qoliplar(shablon)ni o'rganadi va

ulardan yangi, ko‘rinmaydigan ma’lumotlar bo‘yicha bashorat qilish uchun foydalanadi.

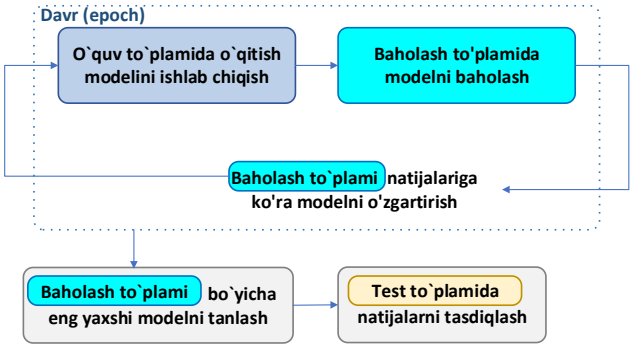
Modelni o‘qitishdagi har bir **davr(epoch)da** bir xil o‘quv ma’lumotlari neyron tarmoq arxitekturasiga qayta-qayta beriladi va model ma’lumotlarning xususiyatlarini o‘rganishda davom etadi [1,2,7]. O‘qitish to‘plami barcha holatlarni yetarli darajada qamrab oladigan *turli xildagi ma’lumotlar to‘plamiga ega bo‘lishi kerak*. Faqat shu holda model **barcha stsenariylarda o‘qitiladi** va kelajakda paydo bo‘lishi mumkin bo‘lgan har qanday ko‘rinmas ma’lumotlar namunasini bashorat qila oladi.

Ma’lumotlarni baholash to‘plami (validation data) – ma’lumotlarni o‘qitish to‘plamidan foydalangan holda mashg‘ulotdan so‘ng, validatsiya/baholash to‘plamidagi tanlamalar tasniflagichning aniqligi yoki xatosini hisoblash uchun ishlatiladi [8]. Biz ushbu ma’lumotlardan modelning giperparametrlarini sozlash va ortiqcha o‘rnatishning oldini olish uchun foydalanamiz. Baholash ma’lumotlari model real dunyoda duch keladigan ko‘rinmas ma’lumotlardan tashkil topishi kerak. Odatda, baholash to‘plami – bu o‘quv jarayonida ML modelini ishlashini baholash uchun foydalanadigan ma’lumotlarning kichikroq qismidan iborat.

Baholash to‘plami – bu o‘qitish to‘plamidan alohida ma’lumotlar to‘plami bo‘lib, u mashg‘ulot davomida bizning modelimiz ishlashini tasdiqlash uchun ishlatiladi. Bu jarayon xuddi tanqidchining mashg‘ulotlar to‘g‘ri yo‘nalishda ketyaptimi yoki yo‘qligini aytishiga o‘xshaydi. Model o‘qitish to‘plamida o‘qitiladi va bir vaqtning o‘zida modelni baholash har bir davrdan keyin baholash to‘plamida amalga oshiriladi. Ma’lumotlar to‘plamini baholash to‘plamiga ajratishning asosiy g‘oyasi bizning modelimiz haddan tashqari mos kelishining oldini olishdir. Ya’ni model o‘qitish to‘plamidagi namunalarni tasniflashda juda yaxshi bo‘ladi, lekin u ilgari ko‘rmagan ma’lumotlarni umumlashtira olmaydi va aniq tasniflay olmaydi.

Modelni baholash jarayonida, biz baholash to‘plamidagi har bir tanlanmaning haqiqiy belgilarini bilamiz. Baholash to‘plamidagi har bir tanlanmadan tasniflagichimizga kuzatish sifatida foydalanib, ushbu tanlanma uchun tasnifni aniqlaymiz. Olingan natijaga ko‘ra, baholash tanlanmasining haqiqiy yorlig‘ini ko‘rib chiqishimiz va uni to‘g‘ri aniqlanganligini tekshiramiz. Agar buni baholash to‘plamining har bir tanlanmasi uchun amalga oshirsak, baholash jarayoni xatosini hisoblashimiz mumkin! Baholash jarayoni xatosi bizni qiziqtirgan yagona ko‘rsatkich bo‘lmasligi mumkin. Mashinali o‘qitish algoritmi samaradorligini baholashning eng yaxshi usuli uning **aniqligi (precision)**, **eslab qolish (recall)** va **F1 bali (F1 score)**ni hisoblashdir [9,10].

Ma’lumotlarni sinovdan o‘tkazish to‘plami (test data) – o‘qitish va sozlashdan keyin modelning yakuniy ishlashini baholash uchun foydalanadigan ma’lumotlardir. Test to‘plamidagi ma’lumotlar o‘quv va baholash to‘plamidagi ma’lumotlaridan butunlay farq qilish kerak. U samaradorlik, aniqlik va boshqa ko‘rsatkichlar bo‘yicha yakuniy model ishlashini ifodalaydi. Oddiy qilib aytganda, u "Model qanchalik yaxshi ishlaydi?" degan savolga javob beradi.



1-rasm. *O‘qitish, tekshirish va baholash to‘plamlari*

2. Ma’lumotlar to‘plamini ajratish usullari

Ma’lumotlar to‘plamida turli xil namunalar va bo‘linishlarni yaratish bizga haqiqiy model ishlashini baholashga yordam beradi. Ma’lumotlar to‘plamini ajratish nisbati ma’lumotlar to‘plami va modeldagi namunalar soniga bog‘liq. Ma’lumotlar to‘plamini qancha qismini baholash to‘plamiga ajratish kerakligini aniqlash juda qiyin masala. Agar o‘quv ma’lumotlar to‘plami juda kichik hajmda bo‘lsa, algoritm samarali o‘qitish uchun yetarli ma’lumotlarga ega bo‘lmasligi mumkin. Boshqa tomondan, agar baholash to‘plami juda kichik bo‘lsa, modelning *samaradorlik, aniqlik, eslab qolish* va *F1 bali* ko‘rsatkichlari katta farqga

ega bo‘lishi mumkin. Umuman olganda, bugungi kunda amalga oshirilgan tadqiqotlarda ma’lumotlarni 80 foizini o‘qitish to‘plamiga va 20 foizini baholash to‘plamiga ajratish tavsiya qilinadi.

Ma’lumotlar to‘plamini ajratish bo‘yicha olinishi mumkin bo‘lgan ba’zi umumiy xulosalar quyidagilar:

–Agar sozlash uchun bir nechta giperparametrlar mavjud bo‘lsa, ML modeli ishlashini optimallashtirish uchun kattaroq hajmdagi baholash to‘plamini talab qilinadi. Xuddi shunday, agar modelda giperparametrlar kam bo‘lsa yoki umuman bo‘lmasa, kichik hajmdagi ma’lumotlar to‘plami yordamida modelni baholash oson bo‘ladi.

–Model turli stsenariylarni o‘qitilishi uchun har bir davrdan keyin modelni baholash lozim.

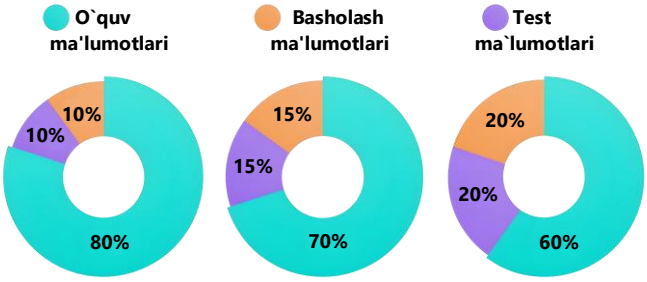
–Ma’lumotlarning o‘lchami/xususiyatlarining oshishi bilan neyron tarmoq funktsiyalarining giperparametrlari ham modelni murakkablashtiradi. Bunday stsenariylarda ma’lumotlarning katta qismi baholash to‘plami bilan o‘quv majmuasida saqlanishi kerak.

Bugungi kunda ma’lumotlar to‘plamini optimal bo‘linish foizi mavjud emas. Qo‘yilgan masala talablarga mos keladigan va modelning ehtiyojlariga javob beradigan bo‘linish foizini aniqlash kerak. Biroq, optimal bo‘linish haqida qaror qabul qilishda ikkita asosiy muammo mavjud:

–Agar o‘quv ma’lumotlari kamroq bo‘lsa, mashinali o‘qitish modeli mashg‘ulotlarda katta farqlarni ko‘rsatadi.

–Kamroq hajmdagi test yoki baholash ma’lumotlari bilan ML modeli samaradorlik statistikasi katta farqlarga ega bo‘ladi.

Shu sababli, ma’lumotlar to‘plami model ehtiyojlariga mos keladigan optimal bo‘linishni topish kerak. Quyida 2-rasmda ma’lumotlar to‘plamini ajratishda duch kelishi mumkin bo‘lgan qo‘pol standart bo‘linish usullari keltirilgan.



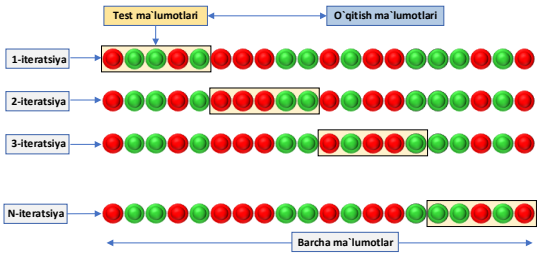
2-rasm. *O‘qitish, tekshirish va baholash to‘plamlariga ajratishga namunalar*

2.1. Ma’lumotlarni ajratishning ilg‘or usullari

Mashinali o‘qitish modellarini sinab ko‘rishning ishonchli va samarali usulini ta’minlash uchun limiy va amaliy tadqiqotlarda ma’lumotlarni ajratishning turli usullari keltirilgan. Ushbu usullarning eng mashhurlaridan ba’zilari quyida tavsiflanadi.

2.2. N-Fold Cross-Validation

Ba’zan ma’lumotlar to‘plami shunchalik kichikki, uni 80/20 ga bo‘lish katta farq (dispersiya)ga olib keladi. Bu muammoning bir yechimi N-Fold Cross-Validation usulini amalga oshirishdir [11]. Ushbu usulning asosiy g‘oyasi shundan iboratki, biz jarayonni N marta bajaramiz va aniqlikni o‘rtacha hisoblaymiz. Misol uchun, 10 marta o‘zaro tekshirishda baholash ma’lumotlarning dastlabki 10% ni *accuracy*, *aniqlik*, *eslab qolish* va *F1 balini* hisoblaymiz. So‘ngra, ma’lumotlarning ikkinchi 10 foizini baholashga o‘rnatamiz va bu statistikani qayta hisoblaymiz. Bu jarayonni 10 marta bajariladi (1-rasm). Har safar baholash to‘plami boshlang‘ich ma’lumotlarning boshqa qismini tashkil qiladi. Ketma-ket amalga oshirilgan baholash qiymatlarining o‘rtacha hisoblash orqali modelimizning o‘rtacha bahosini aniqlash mumkin. N-Fold Cross-Validation usuli bizning ma’lumotlar to‘plamimiz kichik bo‘lsa yoki biz modelimizning giperparametrlarini sozlashni xohlaganimizda foydalidir.



3-rasm. *N-Fold Cross-Validation usuli yordamida ma'lumotlar to'plamini qismlarga ajratish*

N-Fold Cross-Validation usulida model turli namunalarda "N" marta o'qitiladi va baholanadi. Bu jarayonni bir misol bilan tushuntiramiz.

Aytaylik, bizda 1000 ta quyoshli va yomg'rli havo tasvirlaridan iborat muvozanatli, 2-sinf ma'lumotlar to'plami mavjud bo'lsin. Ushbu ma'lumotlar to'plami faqat o'qitish va tekshirish uchun ishlatiladi. Ma'lumotlar to'plami ustida 5 (N=5) marta o'zaro tekshirishni amalga oshirmoqchimiz. Biz birinchi navbatda ma'lumotlar to'plamini teng va farqli 5 ta qismlarga ajratamiz: har biri 200 ta tasvirdan iborat; ularni 1,2,3,4 va 5-qismlar deb belgilaymiz. 200 ta tasvirdan iborat ushbu kichik to'plamlarning har biri bir-biridan farq qiluvchi namunalardan iborat.

So'ngra, 1-5-qismlardan foydalangan holda 5 ta to'liq ma'lumotlar to'plamini (1-5-ma'lumotlar to'plami sifatida belgilangan) quyidagi tarzda shakllantiramiz:

–ma'lumotlar to'plami uchun baholash to'plami sifatida 1-qismdan foydalanib va o'qitish to'plamini yaratish uchun 2-5-qismlarni birlashtiramiz;

–ma'lumotlar to'plami uchun baholash to'plami sifatida 2-qismdan foydalanib va o'qitish to'plamini yaratish uchun 1-,3-5-qismlarni birlashtiramiz;

–Xuddi shunday 3-,4-,5- qismlar uchun yuqoridagi yondashuv asosida baholash va o'quv to'plarini hosil qilamiz;

E'tibor bering, har bir qism asl ma'lumotlar to'plamining 20% ma'lumotlaridan iborat bo'lganligi sababli, 1-5 ma'lumotlar to'plamining har biri 80%-20% oq`uv baholash nisbatiga ega.

Umumlashtirib aytadigan bo'lsak, har bir N-Fold o'zaro tekshirish ma'lumotlar to'plami o'zining baholash to'plamida (100/N)% ma'lumotlarga ega (bu erda 100/5 = 20% baholash to'plamida edi).

N-Fold o'zaro tekshiruvidan foydalanish ML modelini (bir-biridan mustaqil ravishda o'qitilgan) ma'lumotlarning turli taqsimotlariga (aniqrog'i, N marta) ta'sir qiladi. Bu o'quv va baholash to'plamlarida ma'lumotlarni tanlashda yuzaga kelishi mumkin bo'lgan har qanday noto'g'rilikni engillashtiradi. O'rtacha va standart og'ish qiymatlari odatda N-Fold o'zaro tekshirish sxemalaridan foydalanganda aniqlanadi.

Amalga oshirilgan tahlil natijalari shuni ko'rsatadiki, N-Fold o'zaro tekshirish usuli ham tasodifiy tanlab olish bilan bir xil muammoga duch keladi. Ma'lumotlar to'plamining "N" qismlari yoki kichik to'plamlari yaratilganda ma'lumotlarni taqsimlash nomutanosib bo'lishi mumkin. Yuqoridagi misolda shunday bo'lishi mumkinki, ma'lumotlar to'plamining 2-qismi 200 ta quyoshli havo tasvirlaridan iborat bo'lib, yomg'irli havolar tasviri mavjud bo'lmasligi mumkin.

Shunday qilib, "Stratified N-Fold Cross-Validation" usuli bunday nomuvofiqliklarni bartaraf qiladi. Stratified sampling usuli kabi ma'lumotlarning sinf nisbati "N" kichik to'plamlari yoki ma'lumotlar qismlarini yaratishda saqlanadi. Shunday qilib, bir xil sinf taqsimoti ushbu "N" qismlari yakuniy to'liq ma'lumotlar to'plamini yaratish uchun birlashtirilganda amalga oshiriladi.

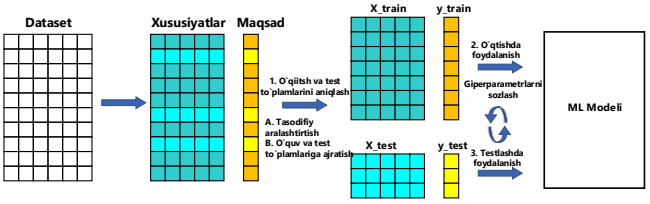
Scikit-learn paketi yordamida ma'lumotlar to'plamida Cross-Validation qanday amalga oshirishga misol:

```
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import make_pipeline
from uznlpdatasets import load_corpus
data = load_corpus()
X, y = data.data, data.target
pipeline = make_pipeline(TfidfVectorizer(), LogisticRegression(max_iter=1000))
scores = cross_val_score(pipeline, X, y, cv=5)
print(f"Accuracy: {scores.mean():.2f}")
```

Accuracy: 0.93

2.3. Random Train-Test Split (Tasodifiy ravishda o‘quv-test to‘plamlariga ajratish)

Tasodifiy tanlab olish ma’lumotlar to‘plamini bo‘lishning eng qadimgi va eng mashhur usulidir [12]. Ushbu usulda ma’lumotlar to‘plami aralashtiriladi, namunalar tasodifiy tanlanadi va foydalanuvchi tomonidan qancha foiz nisbati berilganiga qarab o‘quv, baholash yoki test to‘plamiga joylashtiriladi (4-rasm).



4-rasm. *Tasodifiy ravishda o‘quv-test to‘plamlariga ajratish usuli*

Ma’lumotlarni *o‘quv, baholash* va *test to‘plamiga* ajratishning eng oddiy usuli bu tasodifiy bo‘linish bo‘lib, ma’lumotlar to‘plami tasodifiy ravishda o‘quv va test to‘plamiga ajratiladi. Agar ma’lumotlar to‘plami yetarlicha katta hajmli bo‘lganda ushbu usulni amalga oshirish oson bo‘lib, yaxshi natija beradi. Biroq, agar ma’lumotlar to‘plami kichik hajmda bo‘lsa yoki test to‘plami biz umumlashtirmoqchi bo‘lgan ma’lumotlarga mos kelmasa, *o‘quv ma’lumotlariga haddan tashqari moslashish (risk of overfitting)* muammosi yuzaga keladi. Tasodifiy bo‘linish, agar ma’lumotlar to‘plami nomutanosib bo‘lsa, noto‘g‘ri ma’lumotlar to‘plamiga olib kelishi mumkin, ya’ni u bir sinfdan boshqalarga qaraganda ko‘proq namunalarni o‘z ichiga oladi.

Misol uchun, agar ma’lumotlar to‘plamida 1000 ta tasvir bo‘lsa, ulardan 800 tasi "erkak" toifasiga va 200 tasi "ayol" toifasiga tegishli bo‘lsa, ma’lumotlarni tasodifiy tanlab olish uchun 80%-20% nisbat qo‘llaniladi. Biroq, ushbu usuldan foydalanishda o‘qitish to‘plami faqat erkak jinsiga mansub insonlar tasvirlaridan, baholash to‘plami esa faqat ayol jinsiga mansub insonlar tasvirlaridan iborat bo‘lishi mumkin. Bu juda ekstremal bo‘lmasa ham, o‘quv va baholash to‘plamlari o‘rtasida nomutanosiblik bo‘lishi mumkin.

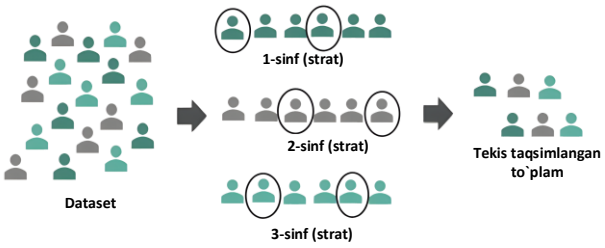
Bunday hollarda tabaqalashtirilgan tasodifiy bo‘linish strategiyasi tavsiya etiladi. Bunda, har bir sinfning ulushi o‘quv va test majmualarida bir xil bo‘lishi ta’minlanadi. Scikit-learn yordamida O‘zbek tili korpusi ma’lumotlar to‘plamida tasodifiy bo‘linishni qanday amalga oshirishga misol:

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
from uznlpdatasets import load_corpus
from sklearn.metrics import accuracy_score
# datasetni yuklash
data = load_corpus()
X, y = data.data, data.target
# o‘qitish va test to‘plamlarga ajratish
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
pipeline = make_pipeline(TfidfVectorizer(), LogisticRegression(max_iter=1000))
# o‘qitish
pipeline.fit(X_train, y_train)
# testlash
y_pred = pipeline.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

Accuracy: 0.91

Yuqoridagi misolda UzbDataset ma’lumotlar to‘plamini **scikit-learn** paketidagi **train_test_split()** metodi yordamida 80% o‘quv ma’lumotlariga va 20% test ma’lumotlariga ajratildi. Natijalar takrorlanmasligini ta’minlash uchun **random_state** parametriga qat'iy belgilangan qiymat o‘rnatildi. So‘ngra, o‘quv ma’lumotlari to‘plami asosida logistik regressiya modeli o‘qitiladi va test to‘plamida uning to‘g‘riligi baholanadi.

2.4. Stratified Sampling (Tekis taqsimot asosida namunalar to‘plamini shakllantirish)



5-rasm. *Stratified Sampling usuli asosida to'plamlarni shakllantirish*

Stratified Sampling – bu o'quv va test to'plamlaridagi har bir sinfning ulushi asl ma'lumotlar to'plamidagi bilan bir xil bo'lishini ta'minlaydigan ma'lumotlarni ajratish strategiyasidir [13]. Ushbu yondashuv, ayniqsa, bir sinf boshqalarga qaraganda ancha keng tarqalgan nomutanosib ma'lumotlar to'plamlari bilan ishlashda foydalidir. **Stratified Sampling** model reprezentativ ma'lumotlar asosida o'qitilganligini va baholanganligini ta'minlashga yordam beradi va bu modelning umumiy ish faoliyatini yaxshilashi mumkin.

Aytaylik, ma'lumotlar to'plami 1000 ta tasvirdan iborat bo'lib, ulardan 600 tasi "erkak" va 400 tasi "ayol" tasviri. Bunday holda, tabaqalashtirilgan namuna olish o'quv va baholash to'plamlarida tasvirlarning 60% "erkak" toifasiga va 40% "ayol" toifasiga tegishli ekanligini ta'minlaydi. Ya'ni, agar ma'lumotlarni ajratishda 80%-20% bo'linish kerak bo'lsa, o'qitish to'plamidagi 800 ta rasmdan 480 tasi (60%) erkaklarga, qolgan 320 tasi (40%) esa ayollarga tegishli bo'ladi.

Keling, yana bir misolni ko'rib chiqamiz. Obyektni aniqlash vazifalarida muayyan tasvirlar turli toifalarga mansub bir nechta turli obyektlarni o'z ichiga olishi mumkin. Ma'lumotlar to'plamidagi ba'zi rasmlarda erkakning 10 ta namunasi va yosh bolaning atigi 1 namunasi bo'lsin. Boshqalarida 10 ta yosh bola va 2 ta erkak, boshqasida esa yosh bolalarsiz, 5 ayol va 5 erkak bo'lishi mumkin. Bunday hollarda tasvirlarning tasodifiy bo'linishi obyekt toifalari bo'yicha taqsimlanishini buzishi mumkin. Boshqa tomondan, tabaqalashtirilgan tanlama ma'lumotlarni obyekt toifalari taqsimoti muvozanatli bo'ladigan tarzda taqsimlashi mumkin. Tabaqalashtirilgan namuna olish usuli ma'lumotlarni taqsimlashning yanada adolatli usuli bo'lib, ML modeli bir xil ma'lumotlarni taqsimlashda o'qitiladi va baholanadi.

Quyidagi dastur kodida **scikit-learn** paketi yordamida UzbDataset ma'lumotlar to'plamidan **Stratified Sampling** usuli namuna olishga misol:

```
from sklearn.model_selection import StratifiedShuffleSplit
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import make_pipeline
from uznlpdatasets import import load_corpus
from sklearn.metrics import accuracy_score
# datasetni yuklash
data = load_corpus()
X, y = data.data, data.target
pipeline = make_pipeline(TfidfVectorizer(), LogisticRegression(max_iter=1000))
# StratifiedShuffleSplit ni e'lon qilish
sss = StratifiedShuffleSplit(n_splits=5, test_size=0.2, random_state=42)
scores = []
for train_index, test_index in sss.split(X, y):
    # to'plamlarga ajratish
    X_train, X_test = [X[i] for i in train_index], [X[i] for i in test_index]
    y_train, y_test = y[train_index], y[test_index]
    # o'qitish
    pipeline.fit(X_train, y_train)
    # testlash
    y_pred = pipeline.predict(X_test)
    score = accuracy_score(y_test, y_pred)
    scores.append(score)
accuracy = sum(scores) / len(scores)
print(f"Accuracy: {accuracy:.2f}")
```

Accuracy: 0.97

Baholash to'plami vositasida Holdout Validation (Holdout Validation with Validation Set)

Baholash to'plami vositasida Holdout Validation – bu ma'lumotlar to'plamini o'qitish to'plamiga, baholash to'plamiga va test to'plamiga ajratishni o'z ichiga olgan ma'lumotlarni ajratish strategiyasi [14,15]. Biz modelga **moslashish uchun** o'qitish to'plamidan, modelning **giperparametrlarini sozlash uchun** baholash to'plamidan va **yakuniy modelni baholash uchun** test to'plamidan foydalanamiz. Odatda,

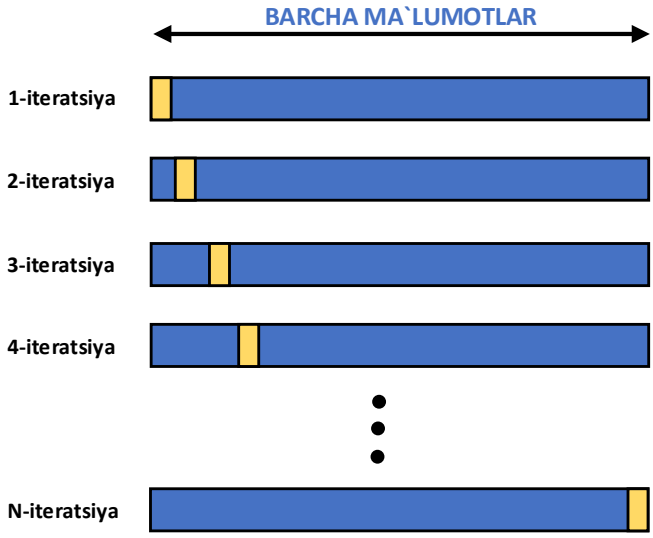
bunday yondashuv bizda *katta hajmdagi ma'lumotlar to'plami mavjud bo'lganida* va *cross-validationni hisoblash*ni amalga oshirmaslik uchun foydali bo'ladi. Quyida scikit-learn paketi yordamida UzbDataset ma'lumotlar to'plamida baholash to'plami vositasida **Holdout Validation** tekshiruvini qanday amalga oshirishga misol keltirilgan:

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
from sklearn.datasets import fetch_20newsgroups
from sklearn.metrics import accuracy_score
# datasetni yuklash
data = load_corpus()
X, y = data.data, data.target
# 1-qadam: o'qitish hamda baholash-testlash to'plamlariga ajratish (60%,40%)
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.4, random_state=42)
# 2-qadam: baholash-testlash to'plammini 2 ta baholash (50%) va testlash (50%) to'plamlariga ajratish
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)
pipeline = make_pipeline(TfidfVectorizer(), LogisticRegression(max_iter=1000))
# o'qitish
pipeline.fit(X_train, y_train)
# baholash to'plam
val_predictions = pipeline.predict(X_val)
val_accuracy = accuracy_score(y_val, val_predictions)
print(f"Validation Accuracy: {val_accuracy:.2f}")
# test to'plam
test_predictions = pipeline.predict(X_test)
test_accuracy = accuracy_score(y_test, test_predictions)
print(f"Test Accuracy: {test_accuracy:.2f}")
```

Validation Accuracy: 0.97
Test Accuracy: 0.96

Ushbu misolda biz UzbDataset ma'lumotlar to'plamini scikit-learn paketidan **train_test_split** yordamida **60% o'qitish to'plamiga**, **20% baholash to'plamiga** va **20% test to'plamiga** ajratdik.

2.5. Leave-One-Out Cross-Validation



6-rasm. *Leave-One-Out Cross-Validation*

Leave-one-out cross-validation (LOOCV) – bu k-marta o'zaro baholashning maxsus yondashuvi bo'lib, unda **k** qiymati ma'lumotlar to'plamidagi namunalar soniga mos tarzda o'rnatiladi [16]. Boshqacha qilib aytganda, ma'lumotlar to'plamidagi har bir namuna bir marta baholash to'plami sifatida ishlatiladi, qolgan namunalar esa o'qitish uchun ishlatiladi. LOOCV – bu modelning ishlashini baholashning juda ishonchli usuli bo'lsa-da, katta hajmdagi ma'lumotlar to'plami uchun hisoblash juda murakkab bo'lishi mumkin. Quyida scikit-learn paketi yordamida UzbDataset ma'lumotlar to'plamida LOOCV ususlining qanday amalga oshirishga misol keltirilgan:

```
from sklearn.model_selection import LeaveOneOut
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
from uznlpdatasets import load_corpus
from sklearn.metrics import accuracy_score
# datasetni yuklash
data = load_corpus()
```

```
    X, y = data.data, data.target
    # Leave-one-out cross-validation obyectini shakllantirish
    loo = LeaveOneOut()
    pipeline = make_pipeline(TfidfVectorizer(), LogisticRegression(max_iter=1000))
    predictions = []
actual_labels = []

    # Logistik regressiya modelini LOOCV usuli yordamida o‘qitish
    for train_index, test_index in loo.split(X):
        # Split the data
            X_train, X_test = [X[i] for i in train_index], [X[i] for i in test_index]
            y_train, y_test = y[train_index], y[test_index]
        # o‘qitish
        pipeline.fit(X_train, y_train)
        # testlash
        y_pred = pipeline.predict(X_test)
        predictions.append(y_pred[0])
        actual_labels.append(y_test[0])
    accuracy = accuracy_score(actual_labels, predictions)
    print(f"Accuracy: { accuracy:.2f}")
```

_____

Accuracy: 0.97

3. Xatoliklar va ularni bartaraf qilish

ML modellarni ishlab chiqishda ma'lumotlarni o‘qitishda yo‘l qo‘yadigan keng tarqalgan xatolarni qisqacha muhokama qilamiz.

3.1. Past sifatli o‘quv ma'lumotlari

O‘qitiladigan ma'lumotlarining sifati model samaradorligini oshirish uchun juda muhimdir. Agar o‘quv ma'lumotlari "axlat" bo‘lsa, modelning yaxshi ishlashini kutish mumkin emas. Shuningdek, ML algoritmlari o‘quv ma'lumotlariga sezgir bo‘lganligi sababli, hatto o‘quv majmuasidagi kichik o‘zgarishlar/xatolar ham model ishlashida sezilarli xatolarga olib kelishi mumkin. Odatda, mashinali o‘qitishda ma'lumotlarni o‘qitish, baholash va test to‘plamlariga ajratishdan avval boshlang‘ich qayta ishlash bosqichni amalga oshiriladi. Bunda, ma'lumotlar to‘plidagi ortiqcha, keraksiz elementlar olib tashlanadi va ma'lumotlar model uchun zarur formatga keltiriladi.

3.2. Inadequate Sample Size (kam hajmdagi ma'lumotlar to‘plami)

O‘qitish, baholash yoki test to‘plamlarida namuna hajmining yetarli emasligi ML modelining ishonchsiz ishlash ko‘rsatkichlariga olib kelishi mumkin. Agar o‘quv to‘plami juda kichik bo‘lsa, model yetarlicha shablonlarni aniqlamasligi yoki yaxshi umumlashtirmasligi mumkin. Xuddi shunday, agar baholash to‘plami yoki test to‘plami juda kichik bo‘lsa, model samaradorligini baholash bo‘lmasligi mumkin.

3.3. Data Leakage (Ma'lumotlarning "oqib" chiqishi)

Ma'lumotlarning oqib chiqishi baholash yoki test to‘plamidagi ma'lumotlar o‘quv to‘plamiga tasodifan sizib ketganda sodir bo‘ladi. Bu haddan tashqari optimistik ishlash ko‘rsatkichlariga va yakuniy model aniqligini oshirib yuborishga olib kelishi mumkin. Ma'lumotlarning sizib chiqishini oldini olish uchun o‘quv, tekshirish va test to‘plami o‘rtasida qat'iy ajratishni ta'minlash, modelni o‘qitish jarayonida baholash to‘plamlaridan hech qanday ma'lumot ishlatilmasligiga ishonch hosil qilish juda muhimdir.

3.4. Improper Shuffle or Sorting (Noto‘g‘ri aralashtirish yoki saralash)

Ma'lumotlarni qismlarga ajratishdan oldin ularni noto‘g‘ri aralashtirish yoki saralash yakuniy modelni umumlashtirishga ta'sir qilishi mumkin. Misol uchun, agar ma'lumotlar to‘plami o‘qitish va tekshirish to‘plamiga bo‘linishdan oldin tasodifiy aralashtirilmasa, u model o‘qitish paytida foydalanishi mumkin bo‘lgan noaniqliklar yoki shablonlarni kiritishi mumkin. Natijada, o‘qitilgan model o‘ziga xos shablonlarga mos kelishi va yangi, ko‘rinmaydigan ma'lumotlarga yaxshi umumlashtira olmasligi mumkin.

3.5. Overfitting (Haddan tashqari moslashish)

ML modeli ko‘rinmas ma'lumotlarni tasniflay olmaydigan darajada o‘quv ma'lumotlaridagi shablonlarni eslab qolsa, *haddan tashqari moslashish* sodir bo‘ladi. O‘quv ma'lumotlaridagi **shovqin (noise)** yoki **tebranishlar (fluctuations)** xususiyatlar sifatida

ko‘riladi va model tomonidan o‘qitiladi. Bu o‘quv majmuasida modelning yaxshi ishlashiga olib keladi, lekin baholash va test to‘plamlarida past samaradorlikka olib keladi.

3.6. Overemphasis on Validation and Test Set metrics (Baholash va test to‘plami ko‘rsatkichlariga ortiqcha e'tibor berish)

Baholash to‘plami ko‘rsatkichi modelni o‘qitish yo‘lini belgilaydigan ko‘rsatkichdir. Modelni o‘qitishdagi har bir davrdan keyin ML modeli baholash to‘plamida baholanadi. Baholash to‘plami ko‘rsatkichlari asosida mos keladigan yo‘qotish shartlari hisoblab chiqiladi va giperparametrlar o‘zgartiriladi. Ko‘rsatkichlar model ishlashining umumiy traektoriyasiga ijobiy ta’sir ko‘rsatishi uchun tanlanishi kerak.

3.7. Modelni / test to‘plamini o‘zgartirish

Ishlab chiqilgan modelning aniqligini tushunish muhim. Chunki model samaradorligini oshirish uchun model parametrlarini sozlashni amalga oshirish mumkin. Masalan, K-Nearest Neighbors algoritmida **K** parametrni oshirib yoki kamaytirganda aniqlikka qanday ta’sir ko‘rsatishini kuzatish mumkin.

Model samaradorligi yetarli bo‘lgach, test to‘plamini aniqlash mumkin. Bu tajriba boshida bo‘lingan ma’lumotlarning bir qismidir. Bu siz tasniflamoqchi bo‘lgan real ma’lumotlarning o‘rnini bosish uchun mo‘ljallangan. U baholash to‘plamiga juda o‘xshash tarzda ishlaydi. Faqat, modelni ishlab chiqish yoki sozlash paytida bu ma’lumotlardan foydalanilmaydi. Test to‘plamidagi *accuracy, aniqlik, eslab qolish* va *F1 balni* hisoblash orqali ishlab chiqilgan algoritm real ma’lumotlar bilan qanday ishlashi aniqlanadi.

Xulosa

Ma’lumotlar to‘plamini *o‘quv, baholash* va *test* to‘plamiga ajratishni o‘z ichiga olgan ma’lumotlarni ajratish strategiyasi mashinali o‘qitishda hal qiluvchi qadamdir. ML modeli uchun mos keladigan ma’lumotlarni ajratish strategiyasini tanlash modelning ishlashiga sezilarli darajada ta’sir ko‘rsatadi. ML modeli o‘quv majmuasidagi kirish xususiyatlari va maqsadli teglar bilan o‘qitiladi va xato yoki yo‘qotish funksiyasini minimallashtirish uchun model parametrlari sozlanadi. Baholash to‘plami ML modelini turli xil giperparametrlar sozlamalarini solishtirishga va baholash xatosi yoki yo‘qotishlarni eng kam bo‘lganini tanlashga imkon beradi. Baholash to‘plami, shuningdek, ML modelini o‘quv to‘plamida yaxshi, lekin yangi ma’lumotlarda yomon ishlayotganida, ortiqcha moslashishni oldini olishga yordam beradi. Test to‘plami – bu o‘quv jarayonidan keyin ML modeli ishlashini baholash uchun foydalanadigan ma’lumotlarning bir qismi. Test to‘plami ML modelini o‘qitish yoki baholash uchun foydalanilmaydi. alki, faqat ko‘rinmas ma’lumotlarda sinab ko‘rish uchun foydalaniladi. Test to‘plami ML modelini haqiqiy ma’lumotlarda qanchalik yaxshi ishlashini baholashga va uning umumlashtirish qobiliyatini o‘lchashga yordam beradi.

Bugungi kunda, ma’lumotlar to‘plamini o‘qitish, baholash va test to‘plamlariga qanday ajratish bo‘yicha aniq qoida yo‘q, ammo amaliyotda 60-20-20 nisbatidan foydalaniladi. Biroq, bu nisbat ma’lumotlar to‘plamining hajmi va xususiyatlariga, ML modeli turi va murakkabligiga qarab farq qilishi mumkin. Ushbu maqolada biz mashinali o‘qitishda eng ko‘p qo‘llaniladigan ma’lumotlarni ajratish/taqsimlash strategiyalarini, jumladan Random Splitting, Stratified Sampling, Holdout Validation with Validation Set, Cross-Validation va Leave-One-Out Cross-Validation usullarini ko‘rib chiqdik va UzbDataset ma’lumotlar to‘plamiga qo‘lladik.

Model ishlashini baholash uchun o‘qitish va baholash to‘plamlaridan foydalanish uchun model va giperparametrlar to‘plamini tanlash, o‘quv to‘plamida modelni o‘qitish, baholash to‘plamida sinab ko‘rishingiz va xatolar hamda yo‘qotishlarni solishtirish kerak. Agar ML modeli juda moslangan (overfitting) bo‘lsa, murakkablikni kamaytirish yoki tartibga solishni qo‘shish lozim. Agar u mos (underfitting) bo‘lmasa, murakkablikni oshirish yoki tartibga solishni olib tashlash zarur. Agar model yaxshi moslangan bo‘lsa, giperparametrlarni turli qiymatlarini yoki turli modellarni sinab ko‘rish kerak. Baholash xatosi yoki yo‘qotilishini minimallashtiradigan eng yaxshi model va giperparametrlarni topmaguningizcha ushbu jarayonni takrorlash lozim. Va nihoyat, yangi ma’lumotlar bo‘yicha

uning ishlashini xolis baholash uchun modelni test to'plamida sinab ko'rish kerak.

Foydalanilgan adabiyotlar

1. Dobbin, K. K., & Simon, R. M. (2011). Optimally splitting cases for training and testing high dimensional classifiers. *BMC medical genomics*, *4*, 1-8.
2. Bai, Y., Chen, M., Zhou, P., Zhao, T., Lee, J., Kakade, S., ... & Xiong, C. (2021, July). How important is the train-validation split in meta-learning?. In *International Conference on Machine Learning* (pp. 543-553). PMLR.
3. Singh, A., Thakur, N., & Sharma, A. (2016, March). A review of supervised machine learning algorithms. In *2016 3rd international conference on computing for sustainable global development (INDIACom)* (pp. 1310-1315). Ieee.
4. Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, *48*(3), 128-138.
5. Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.
6. Rácz, A., Bajusz, D., & Héberger, K. (2015). Consistency of QSAR models: Correct split of training and test sets, ranking of models and performance parameters. *SAR and QSAR in Environmental Research*, *26*(7-9), 683-700.
7. Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, *6*(1), 20-29.
8. Polyzotis, N., Zinkevich, M., Roy, S., Breck, E., & Whang, S. (2019). Data validation for machine learning. *Proceedings of machine learning and systems*, *1*, 334-347.
9. Yacouby, R., & Axman, D. (2020, November). Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In *Proceedings of the first workshop on evaluation and comparison of NLP systems* (pp. 79-91).
10. Naidu, G., Zuva, T., & Sibanda, E. M. (2023, April). A review of evaluation metrics in machine learning algorithms. In *Computer Science On-line Conference* (pp. 15-25). Cham: Springer International Publishing.
11. Zhang, X., & Liu, C. A. (2023). Model averaging prediction by K-fold cross-validation. *Journal of Econometrics*, *235*(1), 280-301.
12. Tan, J., Yang, J., Wu, S., Chen, G., & Zhao, J. (2021). A critical look at the current train/test split in machine learning. *arXiv preprint arXiv:2106.04525*.
13. Meng, X. (2013, May). Scalable simple random sampling and stratified sampling. In *International conference on machine learning* (pp. 531-539). PMLR.
14. Yadav, S., & Shukla, S. (2016, February). Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In *2016 IEEE 6th International conference on advanced computing (IACC)* (pp. 78-83). IEEE.
15. Kim, J. H. (2009). Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational statistics & data analysis*, *53*(11), 3735-3745.
16. Wong, T. T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern recognition*, *48*(9), 2839-2846.