ДЕРЕКТЕРДІ ИНТЕЛЛЕКТУАЛДЫ ТАЛДАУ

Дәріс 15

Машиналық оқытуға кіріспе

Оспан Ә.Ғ., PhD, доцент-зерттеуші

ҚазҰУ, Жасанды интеллект және Big Data кафедрасы

2026-2027 оқу жылы

Дәрістің мақсаты мен мазмұны

Мақсат және оқу нәтижелері

- Жіктеуді, регрессияны және кластерлеуді ажырату
- Оқыту мен тексеру арасында деректер ағып кетпейтін Pipeline құру
- Модельді бірнеше метрика бойынша базалық модельмен салыстыру

Дәріс мазмұны

1. **X және y**
2. **Деректерді бөлу**
3. **Базалық модель және Pipeline**
4. **Метрикалар және CV**
5. **Жалғыз қорытынды тексеру**

Үш есеп: санат, сан және топтар құрылымы

Есеп	Жауап	Оқу сұрағы	Бағалау
Жіктеу	Санат	GK әлде басқа позиция ма?	Қателер матрицасы, F1
Регрессия	Сан	Белгілер бойынша сандық өлшем	MAE, RMSE
Кластерлеу	Белгісіз топ	Қандай профильдер ұқсас?	Орнықтылық және пәндік мағына

Болжамдық байланыс пен себепті түсіндіру — әртүрлі нәтижелер.

GK оқу мақсаты және алдын ала рұқсат етілген белгілер

Мақсат және X

- Position==GK болса y=1; әйтпесе y=0.
- Age, SprintSpeed, Stamina, Strength.
- Reactions және ShortPassing алты белгіні толықтырады.

Алып тастау

- Position және барлық GK* X-тен шығарылады.
- Name, Club, Nationality, Wage, Value қолданылмайды.
- Unnamed: 0 — техникалық ескі индекс.

Белгілер сапаны тексеруге дейін анықталып, қолжетімді болу сәтімен бағаланады.

Деректерді оқып, 20%-ын қорытынды тексеруге сақтау

Python мысалы

```
from pathlib import Path
import pandas as pd
from sklearn.model_selection import train_test_split

football = pd.read_csv(Path('data') / 'football.csv')
features = ['Age', 'SprintSpeed', 'Stamina',
            'Strength', 'Reactions', 'ShortPassing']
X = football.loc[:, features].copy()
y = football['Position'].eq('GK').astype('int64')
assert not any(c.startswith('GK') for c in X)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y)
assert X_train.index.intersection(X_test.index).empty
print(len(X_train), len(X_test))
```

Нәтиже: 10317 2580

Әділ бөлу

- Түрлендірулерді баптауға дейін test бөлінеді.
- Стратификация класс үлестерін шамамен сақтайды.

Шекараларды тексеру

- train және test жолдары қиылыспайды.
- Жолдарды кездейсоқ бөлу жаңа клубтарды тексермейді.

Train, ішкі CV және қорытынды Test

1

Test-ті бөлек сақтау

Жолдардың 20%-ы; шешім таңдауға қолданбау.

2

Train ішіндегі CV

Бес бөлік; әр fold-та барлық түрлендіру қайта үйреніледі.

3

Рәсімді бекіту

Белгілер, алгоритмдер, параметрлер, метрикалар және 0.5 шегі.

4

Қорытынды бағалау

Бүкіл train-де оқытып, test-ті бір рет тексеру.

- CV ішіне X_test немесе y_test түспеуі керек.
- Test қаралғаннан кейін C, белгілер не шек таңдалмайды.

Базалық модель және логистикалық регрессия

DummyClassifier

- Әрдайым train-нің ең жиі класын болжайды.
- Белгілерді елемейді: пайданың бастапқы бағдары.
- Жоғары accuracy GK үшін нөлдік recall-ды жасыруы мүмкін.

LogisticRegression

- Сызықтық score сигмоидамен ықтималдыққа түрленеді.
- Әдепкі жіктеу шегі 0.5-ке тең.
- Регуляризация; C=1.0 Test-ке дейін бекітіледі.

Негізгі модель бірдей деректерде қарапайым ережемен салыстырылады.

Pipeline масштабтау мен жіктеуішті біріктіреді

Python мысалы

```
from sklearn.dummy import DummyClassifier
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

baseline = DummyClassifier(strategy='most_frequent')
model = Pipeline([
    ('scale', StandardScaler()),
    ('classifier', LogisticRegression(
        C=1.0, max_iter=1000, random_state=42)),
])
```

Масштаб

- Орташа мәндер мен SD оқыту деректерінде есептеледі.
- Тексеру жолдары сақталған параметрлермен түрленеді.

Бір estimator

- Pipeline-ды cross_validate ішіне қауіпсіз беруге болады.
- CV ішінде дайындау да қайтадан үйреніледі.

CV-дің бір қайталауында не болады?

1

Train-нің төрт бөлігі

Орташа мәндер мен SD-ні тек осы жолдардан бағалау.

2

Жіктеуішті оқыту

Түрленген оқыту белгілерін пайдалану.

3

Train-нің бесінші бөлігі

Дайын scaler-ді қолданып, метрикаларды есептеу.

4

Келесі Fold

Бүкіл Pipeline-ның жаңа көшірмесін жасау.

- cross_validate алдында бүкіл Train-ді масштабтауға болмайды.
- Бөлек сақталған Test ешбір CV қайталауына қатыспайды.

Қателер матрицасы: оң класс — GK



Жолдар: нақты класс. Барлығы 100 бақылау

Жолдар — шынайы, бағандар — болжанған

- Класс реті [0,1]; 1 дегеніміз GK.
- TN=85, FP=5, FN=2, TP=8.
- Accuracy=0.93; GK recall=0.8; GK precision шамамен 0.615.
- Мысал football моделінің сапасын емес, метрикаларды көрсетеді.

Жоғары жалпы дұрыс жауаптар үлесі сирек класс қателерін жасыруы мүмкін.

Метрика қателер туралы нақты сұраққа жауап береді

Метрика	Сұрақ	Сандық мысал
Дұрыс жауаптар үлесі (accuracy)	Барлық жауаптың қандай үлесі дұрыс?	93/100=0.93
GK дәлдігі (precision)	Болжанған GK-дің қандай үлесі дұрыс?	8/13~0.615
GK толықтығы (recall)	Шынайы GK-дің қандай үлесі табылды?	8/10=0.8
F1 GK	precision мен recall қалай біріктіріледі?	16/23~0.696
Теңгерілген дұрыс жауаптар үлесі	Кластардың орташа толықтығы қандай?	(85/90+8/10)/2~0.872

Метрикалар FP және FN құнын ескеріп, қорытынды тексеруге дейін таңдалады.

Тек Train ішіндегі бес CV бөлігі

Python мысалы

```
from sklearn.model_selection import StratifiedKFold, cross_validate

cv = StratifiedKFold(
    n_splits=5, shuffle=True, random_state=42)
for name, est in [('dummy', baseline), ('logistic', model)]:
    scores = cross_validate(
        est, X_train, y_train, cv=cv,
        scoring=['balanced_accuracy', 'f1'])
    for metric in ['balanced_accuracy', 'f1']:
        values = scores['test_' + metric]
        print(name, metric, values.mean(), values.std(ddof=1))
```

Деректер шекарасы

- Шақыруда тек X_train және y_train бар.
- Бір cv екі модельге салыстырмалы бөліктер береді.

Түсіндіру

- test_* мұнда ішкі тексеру бөлігін білдіреді.
- Fold бағаларының орташа мәні мен SD автоматты сенімділік аралығы емес.

Holdout алдында қорытынды шешімді бекіту

Test-ке дейін бекіту

- X тізімі, y анықтамасы және екі алгоритм.
- C=1.0, scaler, split, CV және 0.5 шегі.
- Accuracy, balanced accuracy, precision, recall, F1 және матрица.

Қорытынды нәтижеден кейін

- Метрикаларды сақтап, қателерді сипаттау.
- Test бойынша жаңа белгілер мен баптаулар таңдамау.
- Жақсартуларды келесі бөлек тәжірибе ретінде жоспарлау.

Қорытынды Test Train бойынша барлық шешімнен кейін бір рет қолданылады.

Бір қорытынды кезең: екі модель бір Test ішінде

Python мысалы

```
from sklearn.metrics import (
    balanced_accuracy_score, classification_report,
    confusion_matrix)

for name, est in [('dummy', baseline), ('logistic', model)]:
    est.fit(X_train, y_train)
    prediction = est.predict(X_test)
    print(name, confusion_matrix(
        y_test, prediction, labels=[0, 1]))
    print('Balanced accuracy:',
          balanced_accuracy_score(y_test, prediction))
    print(classification_report(
        y_test, prediction, labels=[0, 1], zero_division=0))
```

Бір қорытынды тексеру

- Екі алгоритм бірдей Test жолдарын алады.
- Оқыту тек бүкіл Train ішінде орындалады.

Бірлескен есеп

- Матрица, balanced accuracy және әр класс бойынша есеп.
- Метрикалар қаралғаннан кейін рәсімді өзгертпеу.

Оқу Holdout нәтижесі нақты қолдану сапасын дәлелдемейді.

Сапаның шектері және ең қауіпті қателер

Деректердің ағып кетуі

- X ішінде Position немесе GK*.
- Split не CV алдында масштабтау.

Тексеру

- Test бойынша C, шек не seed таңдау.
- Сирек класс қателерінсіз тек accuracy беру.

Жалпылау

- Жолдық Split жаңа клубтар мен жылдарды тексермейді.
- Оқу байланысы себептілікті не спорттық жарамдылықты дәлелдемейді.

Дәрістің негізгі нәтижелері

1. X және y тексеруге дейін анықталады; Position және барлық GK* X-тен шығарылады.
2. Базалық модель белгілерсіз қарапайым ереженің сапасын көрсетеді.
3. Pipeline scaler мен модельді тек оқыту бөліктерінде үйретеді.
4. CV тек Train қолданады; Holdout шешімдер бекітілгеннен кейін тексеріледі.
5. Метрикалар мен қателер матрицасы дереккөзге қатысты шектеулі түсіндірмені қажет етеді.

Түсінуді тексеру және өзіндік жұмыс

1. LogisticRegression мұнда неліктен жіктеуді шешеді?
2. Әр fold-та StandardScaler қай жерде және қай жолдарда үйренеді?
3. Базалық модельде accuracy жоғары, GK recall нөл болуы неліктен мүмкін?
4. cross_validate ішіндегі test_f1 Holdout метрикасынан несімен ерекшеленеді?
5. Жолдарды кездейсоқ бөлу жаңа клубтарға жалпылауды неліктен дәлелдемейді?

Өзіндік жұмыс

[[72,8],[6,14]] матрицасы үшін precision, recall, F1 және balanced accuracy есептеңіз. Шынайы және болжанған кластарды белгілеңіз.

Құжаттама және бастапқы материалдар

Құжаттама

1. **scikit-learn: деректердің ағып кетуі және жиі қателер**
   https://scikit-learn.org/stable/common_pitfalls.html
2. **scikit-learn: Pipeline**
   https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html
3. **scikit-learn: жіктеу метрикалары**
   https://scikit-learn.org/stable/modules/model_evaluation.html

Дәрістің негізі

- football.csv тарихи оқу деректер жиыны