

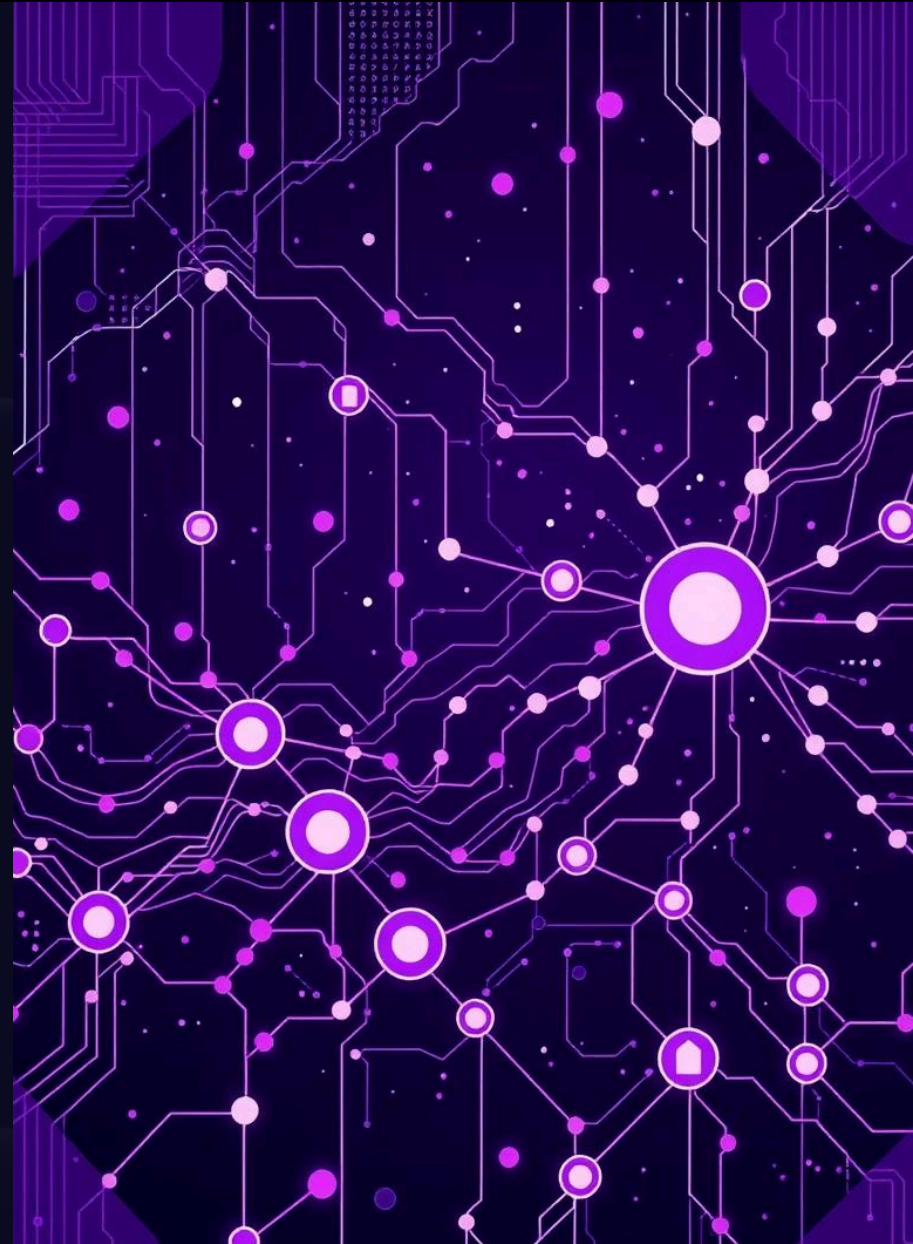
Дәріс 8. Машиналық оқыту негіздері

ML дегеніміз не?

Сәйкестік қателері, дәлдік көрсеткіштері

Feature Engineering

Оқытушы: Сарсембаева Т.С.



Машиналық оқыту

XXI ғасырдағы негізгі тақырыптардың бірі — жасанды интеллект (AI) және деректер ғылымы. Осы салалардың елеулі негізінде машина үйрену жатыр.

Машиналық оқыту (ML): бұл жасанды интеллектің бөлшек-дисциплинасы, ол модельдерді автоматты түрде үйренуге және мәліметтерден болжам жасауға бағытталған. Бұл жүйелерді нақты бағдарламаламай, тәжірибеден (мәліметтерден) үйренуге және жаңа жағдайларда жақсы орындауға мүмкіндік береді.

Data Science ғылымында машиналық оқыту - деректер ғылымы түрінде мәліметтерді талдау, болжам жасау, классификация және шешім қабылдау операциясын басқарудың математикалық жағы. Оның мақсаты - мәліметтер ішіндегі көмескі құрылымдарды анықтап, оларды есептеу модельдерімен сипаттау.

Машина үйрену — бұл деректерден автоматты түрде үйренуге, шешім қабылдауға және өнімділікті жақсартуға мүмкіндік беретін алгоритмдер мен статистика пәні.

Мысалы: "спам" немесе "спам емес" деп е-мейлді бөлуге шешім қабылдайтын алгоритм алғашында адам таңбалаған деректермен үйретіледі, бірақ ақырында жаңа хабарламаларды өзі таңбалайды.

Машина үйренудің негізгі түрлері

Машина үйрену көптеген бағыттарға бөлінеді және олардың әрқайсысының өзіндік ерекшеліктері бар.

1) Қадағаланатын оқыту

Бұл тәсілде кіріс және шығыс мәндері бірдей болады. Мақсат — олардың арасындағы байланыстарды үйрену және жаңа кіріс мәндері үшін дәл болжамдар жасау.

Мысалдар:

- Сызықты регрессия — бағаны болжау;
- Логистикалық регрессия — иә/жоқ шешімдері;
- Шешім ағашы, кездейсоқ орман — жіктеу және регрессия.

2) Қадағалаусыз оқыту

Бұл жерде кірісіміз үйренсек те, не істеу керектігін білмейміз. Деректер арасында өздігінен топтар немесе үлгілерді анықтайды.

Мысалдар:

- K-Means кластерлеу (бірдей объектілер бір топта);
- PCA (негізгі компоненттерді талдау) - өлшемдерді қысқарту;
- T-SNE — көпөлшемдік визуализация.

3) Күшейтілген оқыту

Бұл жерде агент қоршаған ортамен өзара әрекеттеседі және ол дұрыс әрекетіне қарай сыйақы алады. Ол уақыт өте келе өз стратегиясын жетілдіреді.

Мысал: Роботқа жүруді үйрену немесе AlphaGo ойыны.

Машиналық оқыту тәсілдеріне шолу

Тәсіл	Мазмұны	Мысал
Қадағаланатын оқыту (Supervised Learning)	Алгоритмдерге (X) кіріс көрсеткіштері және сол кірістің дұрыс түсіндірілуі (Y) беріледі.	Сызықты регрессия, логистикалық регрессия, Шешім ағашы.
Қадағалаусыз оқыту	Кластерлер/үлгілер ашылады, шығару картасы жоқ.	K-Means, PCA, Иерархиялық кластерлеу
Күшейтілген оқыту (Reinforcement Learning)	Агент қоршаған ортаны бақылап, "ұтыстарға" негізделген дұрыс стратегияны үйренеді.	Q-оқыту, Теңіз үзе, AlphaGo

Қосымша жартылай қадағаланатын және өзін-өзі қадағаланатын үйренуді айтсақ болады.

Бұл тәсілдер таңбаланған және таңбаланбаған деректерді бірге пайдаланады, бұл күрделі деректерде жиі қол жетімді, мысалы, суреттер, мәтін, сөйлеу және т.б.

Сәйкестік қателері, дәлдік көрсеткіштері

Модель құрылған соң, оның қызметінің сапасы қаншалықты екендігін анықтау керек, себебі тек жақсы дайындалған модель ғана сенімді шешімдер қабылдау үшін қолданылуы мүмкін.

Сапа негізінен модельдің болжамдарының нақты нәтижелермен қаншалықты сәйкес келетіндігімен анықталады. Бірақ ML-де екі маңызды мәселе бар:

- **Артық үйрену(Overfitting)** — модель үйретілген деректерге тым бейімделіп, жаңа деректер ұсынылған кезде қателіктер жасайды.
- **Жеткіліксіз үйрену(Underfitting)** – модель жалпы заңдылықтарды үйренуге тым қарапайым.

Сондықтан сапаны өлшеу сатысы ғылыми модельдерді жақсыдан нашарға ажыратудың маңызды фазасы болып табылады.

Негізгі метрикалар

Метрика	Мақсаты	Түсініктеме
Accuracy	Жалпы дәлдік	Дұрыс болжамдардың жалпы үлесі
Precision	Дұрыс табылған позитивтердің үлесі	FP аз болғаны жақсы
Recall	Барлық нақты позитивтердің қаншасы табылды	FN аз болғаны жақсы
F1-score	Дәлдік(Precision) пен Толықтығы(Recall) арасындағы теңгерім	Гармониялық орташа
ROC-AUC	Классификатордың бөлгіш қасиеті	1-ге жақын болғаны жақсы

Қателердің түрлері

Шығын функциялары модельдің сапасын сандық сипаттауға қолданылады.

- Мысалы, регрессияда:

$$L = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Оны орташа квадрат қате (MSE) деп атайды.

Классификацияда — Log Loss, ал нейрондық желілерде — Cross-Entropy Loss жиі қолданылады.

Кросс-валидация(Cross-Validation)

Бұл әдіс деректерді бірнеше сегменттерге бөліп, әр түрлі комбинациялармен модельді тестілеуді мүмкін етеді. Мысалы, 5-Есе Кросс-Валидация қолданылғанда, деректер 5 бөлікке бөлініп, әр бөлікке модельді үйретіп, тестілеміз.

Overfitting және Underfitting мысалдары

Мысалы, егер үй бағасын болжайтын модель өте күрделі болса - ол тек дайындалған мәліметтерге сәйкес нәтиже береді(overfit).

Егер тым қарапайым болса - негізгі модельді түсінбестен барлық үйлердің орташа бағасын қояды(underfit).

Бұл мәселені шешу үшін:

Бұл мәселені шешу үшін:

1. Регуляризация (L1 және L2 регуляризациясы)

Регуляризация - бұл модельдің күрделілігін азайту үшін қолданылатын техника, ол коэффициенттерді шектеу арқылы артық реттеліп кетуін болдырмайды.

L2 регуляризациясы (Ридж регрессиясы)

Олар салмақтардың функциясы ретінде жазланады:

$$L = \text{Loss} + \lambda \sum_{j=1}^p w_j^2$$

Мұнда λ - реттеу коэффициенті. Бұл тәсіл үлкен салмақ мәндерін жазалап, модельді тұрақты етеді.

L1 регуляризациясы (Лассо регрессиясы)

Салмақ шамаларына қарай айыппұл қосылады:

$$L = \text{Loss} + \lambda \sum_{j=1}^p |w_j|$$

Бұл техника параметрлердің кейбіреуін нөлге айналдырады, яғни ол feature selection рөлін атқарады.

1. Early Stopping (Ерте тоқтату)

Ерте тоқтату - оқыту барысында validation loss өссе, процесті ерте тоқтату әдісі. Бұл модельдің артық реттеліп кетпеуін және тиімділігін қамтамасыз ету үшін орындалады.

2. Dropout (нейрондық желілер үшін)

Dropout - жаттығу кезінде нейрондарды ықтимал түрде өшіріп, модельдің жалпылауын арттыратын техника.

Белгілерді алу

Feature Engineering — бұл машиналық оқытудағы ең маңызды және шығармашылық кезеңдердің бірі. Ол модельге сіздің деректер жинағыңызды түсінуге (және одан үйренуге) ең жақсы мүмкіндік беретін жаңа функцияларды жасау, түрлендіру және таңдауды қамтиды.

Машина оқытудың жақсы модельдері көбінесе деректер сапасынан гөрі функциялардың сапасына тәуелді. Сондықтан: функцияларды инженерлеу(Белгілерді алу) «ақпаратты пайдалану өнері» деп аталады.

Функцияларды инженерлеудің басты мақсаты - деректерден тиімді, ақпараттық және болжау функцияларын алу.

Функцияны инженерлеу бірнеше қадамнан тұрады:

01

Функцияларды түсіну

функциялардың қандай екенін және олардың көрінісін білу

02

Функцияларды тазалау

көшірулерді шығару, жетіспеушілік мәндер және қателіктер

03

Функцияларды түрлендіру

кодтау, шкалалау және нормалау

04

Функцияларды жасау

жаңа қосымша функция(лар) шығару немесе есептеу

05

Функцияларды таңдау/қысқарту

ең маңызды функцияларды таңдап немесе қысқарту

Функцияларды инженерлеу қадамдары

1. Функцияларды түсіну

Бірінші кезеңде, біз әрбір бағана немесе айнымалының мәндерін, түрлерін және статистикалық қасиеттерін тексереміз:

- дерек түрі (сандық, мәтін, санаттық);
- диапазон (минимум-максимум);
- таралу сипаттамалары (нормалды немесе жоқ);
- басқа функциялармен корреляция.

Мысал:

```
df.info()  
df.describe()  
df.corr()
```

2. Функцияларды тазалау

Көптеген деректер жиынтықтары «шулы» болып табылады, өйткені оларда қателер пен кемшіліктер бар. Бұл жағдайда:

- NaN мәндері медианалық/орташа мәнімен ауыстырылады;
- қайталанбалы жазбалар алынып тасталады;
- санаттық қателер түзетіледі.

Мысал:

```
df['age'].fillna(df['age'].median(), inplace=True)  
df.drop_duplicates(inplace=True)
```

3. Функцияларды түрлендіру

Бұл кезеңде функциялар сандық түрде түсінілетін пішінге айналған.

Кодтау

Тексттік функцияларды сандық түрге айналдыру.

Мысал:

```
from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()

df['gender_encoded'] = encoder.fit_transform(df['gender'])
```

Шкалалау

Функциялар мәндерін жалпы бір шкалаға қиыстыру. Мысалы, егер "жас" 0-100 аралығында болса және "айлық" 0-100000 аралығында болса, модель алдымен "айлықты" қарастырады. Сондықтан шкалалау қолданылады.

- $z = (x - \mu) / \sigma$.

Нормалау

[0,1] интервалға қайта калибровкау:

- $x' = (x - x_{\min}) / (x_{\max} - x_{\min})$

4. Функцияларды жасау

Функцияларды жасау маңызды қадам болып табылады, өйткені ол модельдің ақылдылық қабілетін арттырады. Басқа жағдайларда, жаңа енгізулер деректер арасындағы жасырын байланыстарды ашуға көмектеседі.

Мысалдар:

- Уақыт мәніне негізделген жаңа мәндер:

```
df['month'] = pd.to_datetime(df['date']).dt.month
```

- Қатынастарды есептеу: BMI (Дене массасының индексі): $BMI \propto \frac{\text{weight}}{\text{height}^2}$
- Интерактивті функциялар:

```
df['income_per_person'] = df['income'] / df['family_size']
```

5. Функцияларды таңдау

Модель барлық ықпал етуші факторларға бірдей сезімтал емес. Сондықтан, оларды ақылды таңдаудың маңызы зор.

Әдістер:

- Корреляцияға негізделген фильтрлеу - тәуелді функцияларды жояды.
- Дерево негізіндегі маңыздылық - қауіптілікке немесе XGBoost көмегімен функцияның маңыздылығын есептейді.
- L1 Регуляризация (Lasso) - айнымалыларға қатысты салмақтарды 0-ге жуық қалдырып аз маңызды функцияларды жояды.
- Оқытушы функцияларды элиминациялау (RFE) - ең байынша ажыратуға қабілетті функцияларды таңдайды.

Мысал:

```
from sklearn.feature_selection import SelectKBest, f_classif

selected = SelectKBest(score_func=f_classif, k=5)

X_new = selected.fit_transform(X, y)
```

6. Өлшемділікті қысқарту

Бұл функцияларды таңдауға байланысты түсінуге, түсінікті және тиімді болу үшін көмектесе алады. Бұл модельді баяу орындап, шамадан тыс ауытқуға әкелуі мүмкін. Сондықтан, PCA (Басты компоненттерді талдау) сияқты әдістер қолданылады.

```
from sklearn.decomposition import PCA
```

```
pca = PCA(n_components=2)
```

```
X_reduced = pca.fit_transform(X)
```

PCA функция корреляциясына негізделіп, оларды дереккөздерден тәуелсіз осьтерге регреске алады. Осылайша, ақпараттардың көбісі әлі де сақталады, және аз ғана өлшемділік жоғалтылады.

Әдебиеттер тізімі:

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. <https://www.deeplearningbook.org>
- Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow. O'Reilly Media.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer. <https://hastie.su.domains/ElemStatLearn/>
- Sarker, I. H. (2021). Data Science and Analytics: An Overview from Data-Driven Smart Computing and Decision Making. SN Computer Science, Springer.
- Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830. <https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>
- Ng, A. (2020). Machine Learning Course. Coursera. <https://www.coursera.org/learn/machine-learning>