

Дәріс 10. Машиналық оқыту алгоритмдері (2 бөлім)

Бірнеше сызықтық регрессия

Логистикалық регрессия

Шешім ағаштары және кездейсоқ ормандар

Машиналық оқыту (Machine learning) — бұл деректердегі заңдылықтарды автоматты түрде анықтай алатын және болашақ нәтижелерді болжай алатын модельдерді құру процесі.

Алдыңғы дәрістерде біз К-жақын көршілерді (KNN), Аңғал Байесті (Naïve Bayes) және қарапайым сызықтық регрессияны (Simple Linear Regression) талдадық.

Енді келесі деңгейге өтіп, күрделі модельдерді қарастырайық — бірнеше регрессия, логистикалық регрессия, шешім ағаштары және кездейсоқ ормандар. Бұл алгоритмдер Data Science, эконометрика, қаржы, денсаулық сақтау, әлеуметтік талдау сияқты салаларда кеңінен қолданылады.

Олар болжам жасауға ғана емес, сонымен қатар себеп-салдарлық байланыстарды түсінуге және модельді түсіндіруді қамтамасыз етуге көмектеседі (interpretability).

Бірнеше сызықтық регрессия

Бірнеше сызықтық регрессия (Multiple Linear Regression) - бұл бірнеше тәуелсіз айнымалылар (feature) мен бір тәуелді айнымалы (target) арасындағы сызықтық қатынасты модельдейтін әдіс. Бұл модель мәліметтердегі бірнеше айнымалылардың әсерін ескере отырып, тәуелді айнымалыны болжау үшін қолданылады. Классикалық сызықтық регрессия тек бір тәуелсіз айнымалымен жұмыс істейтін болса, бірнеше сызықтық регрессия бір уақытта бірнеше тәуелсіз айнымалыларды ескереді және олардың тәуелді айнымалыға әсерін есептейді.

Математикалық түрде:

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

$\hat{y}$ – модель болжаған мән,

b_0 – еркін мүше(intercept),

b_i – әр белгінің (feature) салмағы,

x_2 – түсіндірмелі айнымалылар.

немесе матрица түрінде:

$$\hat{y} = X\beta$$

X — $n \times (p + 1)$ дизайн-матрица (бірінші баған-бірліктер, содан кейін әрбір баған-айнымалылар), $\beta = (\beta_0, \beta_1, \dots, \beta_p)^T$ -коэффициенттердің векторы.

Модельдің мақсаты - нақты мәндер y мен болжамдар $\hat{y}$ арасындағы айырмашылықты азайту:

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

Бұл функция MSE деп аталады, яғни орташа квадраттық қате.

Есептеу әдісі

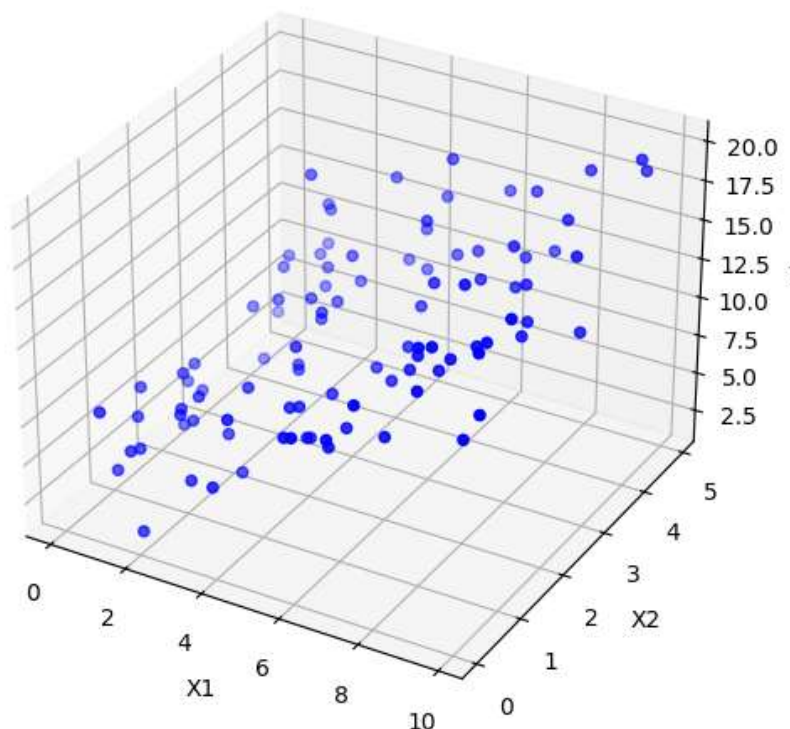
Регрессия коэффициенттері аналитикалық түрде есептеледі:

$$\beta = (X^T X)^{-1} X^T y$$

X – деректер матрицасы, y – мақсаттық вектор.

Модельдің мақсаты тәуелді айнымалыны болжау үшін тәуелсіз айнымалылардың әрқайсысы үшін өз салмағын табу болып табылады. Бұл салмақтар деректерге негізделген (яғни олар максималды ықтималдыққа сәйкес келеді). Модельдің параметрлері (салмақтары) оптимизациялау әдістері арқылы анықталады, мысалы, ең кіші квадраттар әдісі (Ordinary Least Squares, OLS).

Бірнеше сызықтық регрессия



Артықшылықтары:

Түсіндіру оңай (әр белгінің әсерін көруге болады);

Есептеу оңай;

Негізгі модель ретінде пайдаланылады.

Шектеулер:

Тек сызықтық байланыстар үйренеді;

Көпжақтылықпен (белгілер арасындағы корреляция жоғары болған кезде) дәлдік төмендейді;

Тербелістерге сезімтал (шектен тыс).

Логистикалық регрессия

Логистикалық регрессия - тәуелді айнымалының екі немесе одан да көп кластары арасындағы ықтималдықтарды болжауға арналған статистикалық модель. Бұл модельдің негізгі мақсаты әртүрлі деректер негізінде бір классты немесе санатты болжау болып табылады. Логистикалық регрессия көбінесе жіктеу мәселелерін шешу үшін, яғни белгісіз объектілердің белгілі категорияға жататындығын анықтау үшін қолданылады.

Логистикалық регрессияның жұмыс принципі:

Логистикалық регрессия - бұл сызықтық модель, бірақ оның нәтижесі сызықтық емес болады. Болжам жасамас бұрын, модельде тәуелсіз айнымалылардың сызықтық комбинациясы қолданылады, бірақ соңында ол логистикалық функцияның (sigmoida) ықтималдығына айналады.

Логистикалық функция (sigmoida):

Логистикалық регрессияның негізі $[0, 1]$ интервалындағы мәндерді ықтималдықтарға түрлендіретін сигмоидтық функция болып табылады.

Сигмоидтық функция келесідей есептеледі:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

z -сызықтық комбинация (мысалы, $z = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n$).

Шешім қабылдау:

Логистикалық регрессия нәтижесінде шығыс мәні 0 немесе 1 болады, яғни ықтималдылықты есептеп, оны белгілі бір шекпен салыстырыңыз (әдетте 0,5). Егер ықтималдық 0,5-тен жоғары болса, Модель объектіні 1-классқа, ал төменгісі 0-классқа жатқызады.

Қолданылуы: Екілік классификация: логистикалық регрессия тек екі мәнге тәуелді айнымалыларды болжауда қолданылады (мысалы, 0 және 1, Иә және жоқ, ауру және денсаулық).

Модельді оқыту үшін көптеген әртүрлі әдістер қолданылады (мысалы, ықтималдықтың максималды әдісі).

Модельдің артықшылықтары:

1. Логистикалық регрессия есептеу тұрғысынан қарапайым, жылдам және түсінікті.
2. Бұл модельді болжау ықтималды болғандықтан, ықтималдықтарды түсіндіруге мүмкіндік береді.
3. Күрделі емес: тек сызықтық байланыстарды суреттейді, бірақ оның тиімділігін қосымша әдістермен арттыруға болады (мысалы, жүйелеу немесе көпмүшелік кеңею).

Кемшіліктері:

1. Логистикалық регрессия тек сызықтық шешімдерді жасай алады, яғни.әр түрлі кластар арасындағы бөлу сызығы сызықты болуы керек.
2. Күрделі немесе жоғары өлшемді деректер үшін бұл модель дұрыс нәтиже бермеуі мүмкін.

Формуласы:

$$P(y = 1 | x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n)}}$$

Мұнда логистикалық функция (сигмоид) нәтижені 0 — ден 1 аралығында шектейді, яғни ықтималдық береді.

Мақсаты

Модельдің мақсаты-логарифмдік шығын функциясын(log-loss) минимизациялау:

$$L = -\frac{1}{n} \sum_{\{i=1\}}^{\{n\}} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$



Негізгі қасиеттері:

- Нәтиже ықтималдық ретінде түсіндіріледі;
- Decision boundary — сызықтық;
- Регуляризация (L1/L2) арқылы overfitting азайтылады;
- Multiclass классификация үшін One-vs-Rest схемасы қолданылады.

Қолдану мысалдары: Ауруды анықтау (иә/жоқ), несие берілетін клиенттің болжамы, спамды анықтау, клиенттің кету ықтималдығын есептеу және т.б.

Шешім ағаштары (Decision Trees)

Шешім ағашы - бұл көптеген ережелер мен шарттарға негізделген болжамдарды жүзеге асыратын деректерді жіктеу немесе регрессиялау үшін қолданылатын модель түрі. Бұл модельді ойлап табу процесінде негізгі идея - деректерді бірқатар шарттар арқылы оңай және түсінікті түрде бөлу. Шешім ағашы түсінікті және интуитивті, өйткені ол ағаш құрылымы арқылы шешім қабылдайды.

Шешім ағашы келесі элементтерден тұрады:

1. Түбір (Root): шешім ағашының ең бірінші бөлігі. Бұл нүкте барлық деректердің біріктірілген күйін білдіреді. Әдетте, осы түбірде деректерді шығару үшін ең тиімді атрибут таңдалады.

2. Тармақтар (Branches): әрбір ішкі тамырдан шығатын сызықтар. Олар ағаштың бұтақтарын қосымша жағдайлар мен сұрыптаулар арқылы бағыттайды.

3. Түбегейлі түйіндер (Leaf Nodes): белгілі бір шешім немесе болжам жасалатын шешім ағашының соңғы түйіндері. Әрбір радикалды түйіннің белгілі бір класы немесе сандық мәні болады.

4. Шарттар мен бөліну ережелері: әрбір ішкі түйінде деректер белгісіне негізделген бөлу ережесі бар. Мысалы, көрсетілген атрибуттардың нақты мәніне байланысты деректер екі немесе одан да көп бөлікке бөлінеді.

Математикалық негізінде:

Шешім қабылдау ақпараттық ұтыс (Information Gain) принципіне негізделеді:

$$IG = \{Entropy(parent)\} - \sum_i \frac{N_i}{N} Entropy(child_i)$$

мұндағы энтропия:

$$Entropy = - \sum_{k=1}^K p_k \log_2(p_k)$$

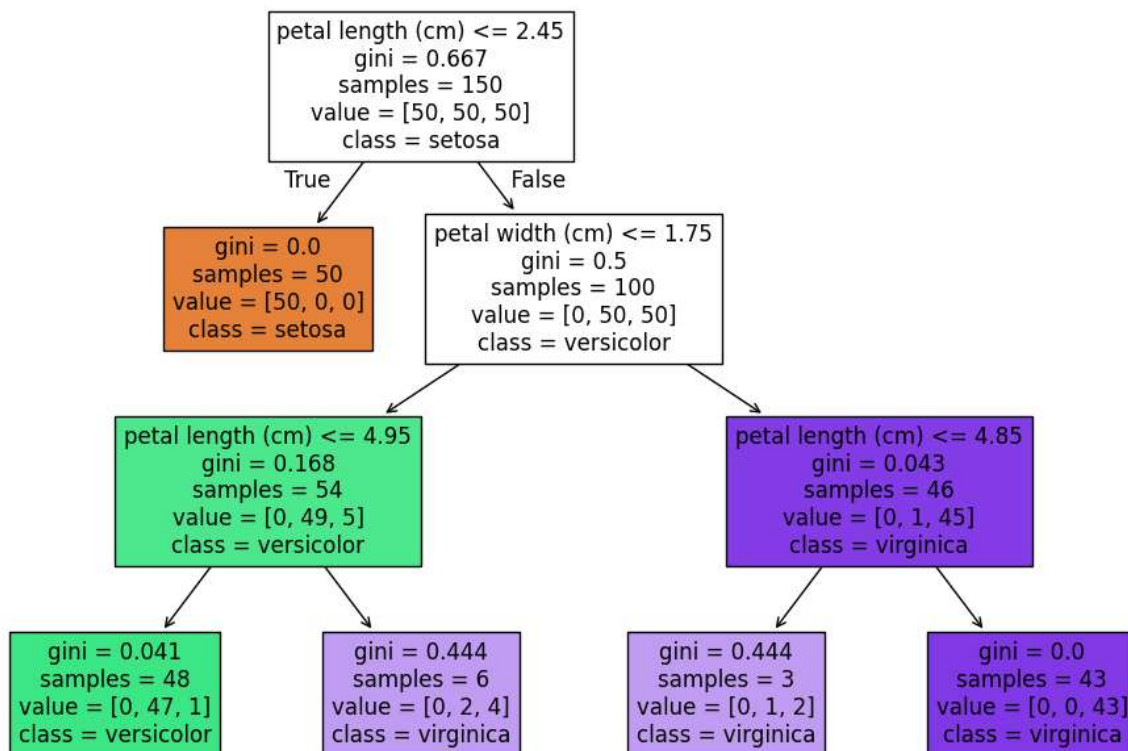
Шешім ағашын классификация (класстарды бөлу) және регрессия (сандық мәндерді болжау) тапсырмаларында да қолдануға болады. Бұл модельдің басты ерекшелігі - ол бірқатар сұрақтар мен ережелер негізінде түйіндер мен тармақтар арқылы деректерді бөледі, нәтижесінде түпкілікті шешім қабылданады.

Мысалға:

Классификация: егер шешім ағашы тұтынушылардың сатып алу әрекетін болжау үшін пайдаланылса, онда ағаштың әрбір түйінінде белгілі бір сұрақтар қойылуы мүмкін және соңында клиенттің болашақ сатып алуын "иә" немесе "жоқ"деп болжауға болады."

Регрессия: егер мақсат үйдің бағасын болжау болса, онда ағаштың әрбір бұтағы әртүрлі атрибуттарға (жер учаскесінің ауданы, бөлмелер саны, орналасқан жері және т.б.) негізделген сұрақтар қояды.), және соңында нақты баға болжанатын болады.

Шешім ағашының құрылымы



Кездейсоқ орман

Кездейсоқ орман (Random Forest) - бұл бірнеше шешім ағаштарының ансамблі болып табылатын статистикалық оқыту әдісі. Оның басты ерекшелігі - бірнеше шешім ағаштарын құру және әр ағаштың нәтижелерін біріктіру арқылы болжам жасау. Бұл модель әртүрлі ағаштардың пікірлерін біріктіріп, ортақ шешім қабылдайды. Әрбір ағаштың өзіндік сипаттамалары болғанымен, оларды біріктірген кезде модельдің болжамы дәлірек және тұрақтылыққа ие болады.

Негізгі идея - көптеген ағаштардың әртүрлілігін пайдалану, осылайша тек бір ағаштың күшті және әлсіз жақтарын теңестіру.

Жұмыс жасау механизміне келсек кездейсоқ орман алгоритмінің жұмысы келесі негізгі қадамдардан тұрады:

1. Кездейсоқ деректерді іріктеу(Bootstrap sampling):

Әрбір шешім ағашы үшін кездейсоқ деректер бөлігі таңдалады. Бұл әдіс Bootstrap деп аталады, онда әрбір ағаш үшін кездейсоқ жасалған деректер үлгісі пайдаланылады. Бұл таңдау әрбір жеке шешім ағашы үшін әртүрлі деректер опцияларын алуға мүмкіндік береді.

2. Кездейсоқ мүмкіндіктерді таңдау:

Әрбір шешім ағашында бөлуді құру кезінде функциялардың кездейсоқ жиынтығы ғана қарастырылады. Яғни, әрбір түйінде барлық белгілердің орнына тек бірнешеуін таңдап, бөлу

керек. Бұл тәсіл модельдің әртүрлілігін арттырып, ағаштар арасындағы корреляцияны азайтуға көмектеседі.

3. Дауыс беру (классификация) немесе ағаштардың орташа есебі (регрессиясы):

Классификация: егер модель жіктеу тапсырмасын орындаса, әр ағаш өз болжамын жасайды және ең көп дауыс жинаған сынып таңдалады (шешім көпшілік дауыспен қабылданады).

Регрессия: егер модель регрессия тапсырмасын орындаса, барлық ағаштардың болжамды мәндерінің орташа мәнін алу керек.

Формуласына келетін болсақ:

$$\hat{y} = \text{mode}(f_1(x), f_2(x), \dots, f_m(x))$$

Бұл формула моданы (mode) қолдануды көрсетеді, яғни жиынның ең көп тараған мәнін алу үшін пайдалануды көрсетеді:

$\hat{y}$ - болжамды немесе алынған нәтиже (болжам),

$(f_1(x), f_2(x), \dots, f_m(x))$ - m түрлі модельдердің немесе функциялардың болжамды мәндері.

Python код:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Random Forest моделін үйрету
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)

print("Нақтылық:", accuracy_score(y_test, y_pred))
```

Әдебиеттер тізімі:

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
<https://www.deeplearningbook.org>
2. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer.
<https://hastie.su.domains/ElemStatLearn/>
3. Sarker, I. H. (2021). Data Science and Analytics: An Overview from Data-Driven Smart Computing and Decision Making. SN Computer Science, Springer.
4. 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>