

Тақырыбы: Шешімдер Ағаштары және Ансамбльдік Әдістер: Random Forest және Gradient Boosting

Шешімдер ағашы (Decision Tree) — бұл классификация және регрессия мәселелерін шешуге арналған интуитивті және интерпретациясы оңай машиналық оқыту моделі. Ағаш бірізді шарттардың жиынтығын білдіреді, ол бақылауларды мәліметтердің ерекшеліктеріне сүйене отырып, ақырғы сыныптарға немесе мәндерге (жапырақ түйіндерге) бөледі.

Ағаштың құрылымы:

- Түбір түйін (Root Node): Ең жоғарғы түйін, бүкіл деректер жиынтығын көрсетеді.
- Ішкі түйіндер (Internal Nodes): Деректерді бөлу үшін қолданылатын шартты (ерекшелікке қойылған сұрақ) қамтиды.
- Тармақтар (Branches): Шарттың нәтижесі (Иә/Жоқ, $A > B$).
- Жапырақ түйіндер (Leaf Nodes): Соңғы шешімді немесе болжанған сыныпты/мәнді көрсетеді.

Түйінді бөлу критерийлері (Splitting Criteria)

Ағаштың тиімділігі әрбір түйінде деректерді қаншалықты "таза" немесе "біртекті" топтарға бөлетініне байланысты. Бұл тазалықты өлшеу үшін екі негізгі метрика қолданылады:

1. Джини индексі (Gini Index): Классификация үшін қолданылады. Бұл топтағы кездейсоқ таңдалған екі элементтің әртүрлі сыныптарға жату ықтималдылығын өлшейді. Төмен Джини индексі – жоғары біртектілікті білдіреді. Ең жақсы бөлу – Джини индексін ең көп азайтатын бөлу.

$$Gini = 1 - \sum_{i=1}^C (p_i)^2$$

мұндағы p_i — таңдалған түйіндегі i -ші сыныптың үлесі, C — сыныптар саны.

2. Ақпараттық өсім (Information Gain/Entropy): Энтропия – топтағы "таза еместік" немесе "тәртіпсіздік" дәрежесін өлшейді. Ақпараттық өсім – бөлу нәтижесіндегі энтропияның төмендеу мөлшері. Ең жоғары ақпараттық өсім беретін ерекшелік таңдалады.

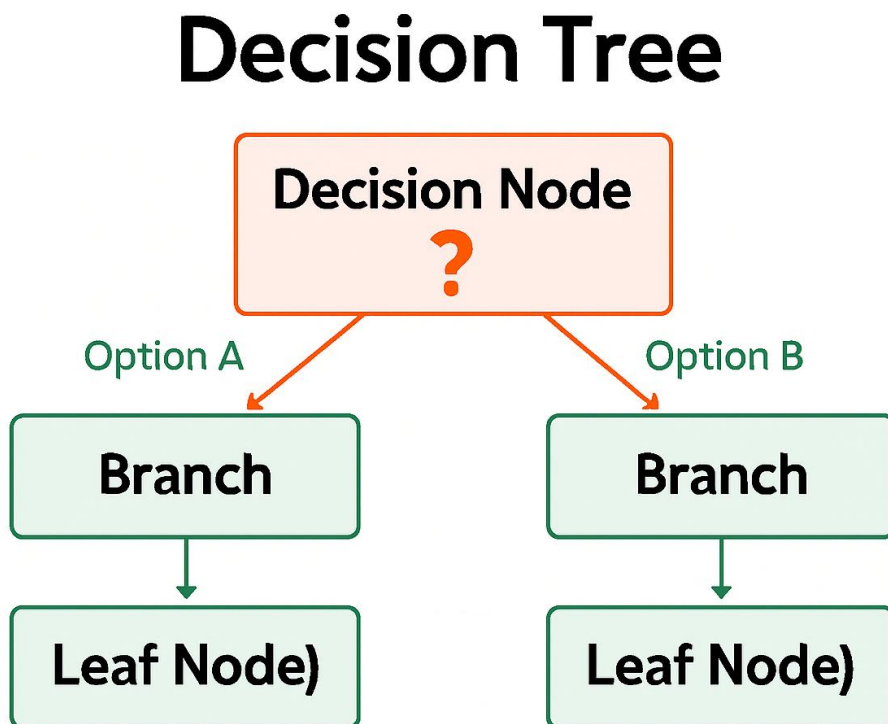
$$Entropy = - \sum_{i=1}^C p_i \log_2(p_i)$$

Шешімдер ағашының кемшіліктері

- Артық оқыту (Overfitting): Егер ағаш өте терең өсірілсе, ол оқыту деректеріндегі шуды және ерекшеліктерді жаттап алады да, жаңа (көрілмеген) деректерде өте нашар жұмыс істейді.
- Тұрақсыздық: Оқыту деректеріндегі кішігірім өзгерістер (бірнеше нүктені алып тастау немесе қосу) ағаштың құрылымын түбегейлі өзгертіп жіберуі мүмкін.

Осы кемшіліктерді жою үшін ансамбльдік әдістер қолданылады.

Сурет 5.1: Шешімдер ағашының құрылымы



Ансамбльдік әдістер: Random Forest және Gradient Boosting

Ансамбльдік әдістер (Ensemble Methods) — бұл бір модельді емес, бірнеше жеке (әлсіз) модельдерді біріктіру арқылы ортақ болжам жасайтын машиналық оқыту стратегиясы. Бұл бірнеше моделдің күшін біріктіріп, бір ғана моделге қарағанда әлдеқайда дәлірек және тұрақты нәтиже алуға мүмкіндік береді.

Ең танымал ансамбльдік әдістер екі топқа бөлінеді:

1. Бэггинг (Bagging): Бірнеше тәуелсіз модельдерді параллель оқыту және олардың нәтижелерін орташалау/дауыс беру арқылы біріктіру (мысалы, Random Forest).
2. Бустинг (Boosting): Бірнеше модельдерді бірізді (кезекпен) оқыту, мұнда келесі модель алдыңғы моделдің қателерін түзетуге бағытталған (мысалы, Gradient Boosting).

Random Forest (Кездейсоқ Орман) Алгоритмі

Random Forest — бұл бэггинг (Bagging) стратегиясына негізделген, көптеген шешімдер ағаштарынан тұратын ансамбльдік алгоритм. Ол шешімдер ағашының артық оқытылу (overfitting) мәселесін шешеді және модельдің тұрақтылығын арттырады.

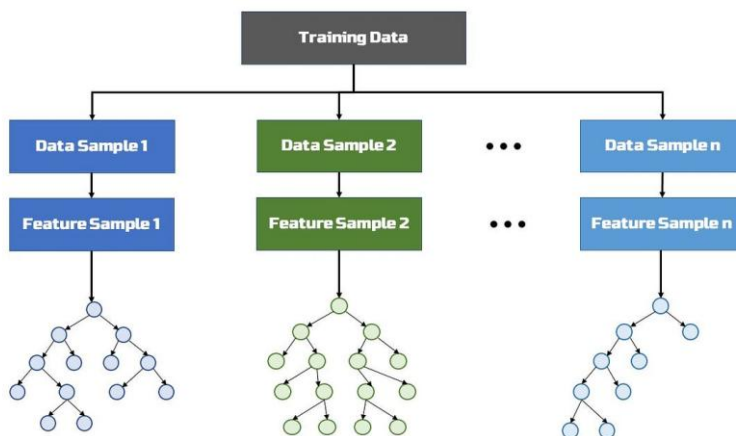
Random Forest-тың негізгі қадамдары:

1. Bootstrap Sampling (Бэггинг): Оқыту деректерінің бастапқы жиынтығынан қайталап іріктеу арқылы бірнеше (N) жаңа ішкі жиынтықтар (bootstrap samples) құрылады. Әрбір ішкі жиынтық бастапқы жиынтықтың шамамен 63%-ын қамтиды.
2. Ерекшеліктерді кездейсоқ таңдау: Әрбір ағаштың түйінін құру кезінде, қолжетімді барлық ерекшеліктерден (атрибуттардан) тек кездейсоқ таңдалған кішігірім ішкі жиынтық ($m \ll M$) ғана бөлуге қолданылады. Бұл әдіс екі еселенген кездейсоқтықты қамтамасыз етеді және әрбір ағаштың бір-біріне тәуелсіз болуына кепілдік береді.
3. Ағашты құру: Әрбір ішкі жиынтық үшін одан бір шешімдер ағашы құрылады. Ағаштар әдетте толығымен өсіріледі (кесусіз).
4. Болжам жасау:
 - Классификация үшін: Барлық ағаштардың болжамдары жиналып, көпшілік дауыс (majority voting) арқылы соңғы сынып анықталады.
 - Регрессия үшін: Барлық ағаштардың болжамдарының орташа мәні алынады.

Артықшылықтары:

- Артық оқытуға өте төзімді.
- Өлшемділіктің жоғарылауына төзімді.
- Жетіспейтін деректермен және аномалиялармен жақсы жұмыс істейді.

Сурет 5.2: Random Forest архитектурасы (Бэггинг)



Gradient Boosting (Градиентті Бустинг) Алгоритмі

Gradient Boosting (Градиентті Бустинг) — бұл бустинг (Boosting) стратегиясына негізделген қуатты ансамбльдік әдіс. Бұл алгоритмде модельдер бірізді түрде оқытылады, және әрбір келесі модель алдыңғы модельдердің жіберген қателіктерін (қалдықтарды) түзетуге бағытталған.

Негізгі қағидасы:

1. Алғашқы модель: Бір қарапайым модельмен (көбінесе бір жапырақты шешімдер ағашы – 'stub') басталады.
2. Қалдықтарды есептеу: Модельдің болжамы мен нақты мән арасындағы қателіктер (қалдықтар) есептеледі.
3. Екінші модельді оқыту: Екінші модель нақты мақсатты мәндерді емес, бірінші модельдің жіберген қателіктерін болжауға үйретіледі.
4. Қателіктерді төмендету: Жаңа модельдің нәтижесі алдыңғы модельдің нәтижесіне қосылады, бірақ оған оқыту жылдамдығын (learning rate) ескере отырып, кішігірім салмақ беріледі. Бұл қателікті градиенттік түсу (Gradient Descent) принципі бойынша біртіндеп азайтуға мүмкіндік береді.

Жаңа болжамның формуласы:

$$F_m(X) = F_{m-1}(X) + \eta \cdot h_m(X)$$

мұндағы:

- $F_m(X)$ — m -ші кезеңдегі ансамбльдің болжамы.
- $F_{m-1}(X)$ — алдыңғы ансамбльдің болжамы.
- $h_m(X)$ — m -ші (жаңа) әлсіз модель (ағаш).
- (η) — оқыту жылдамдығы (Learning Rate), әдетте $0 < \eta \leq 1$.

Мысал алгоритмдері: XGBoost, LightGBM, CatBoost – бұлар Градиентті Бустингтің жоғары тиімді және кеңінен қолданылатын түрлері.

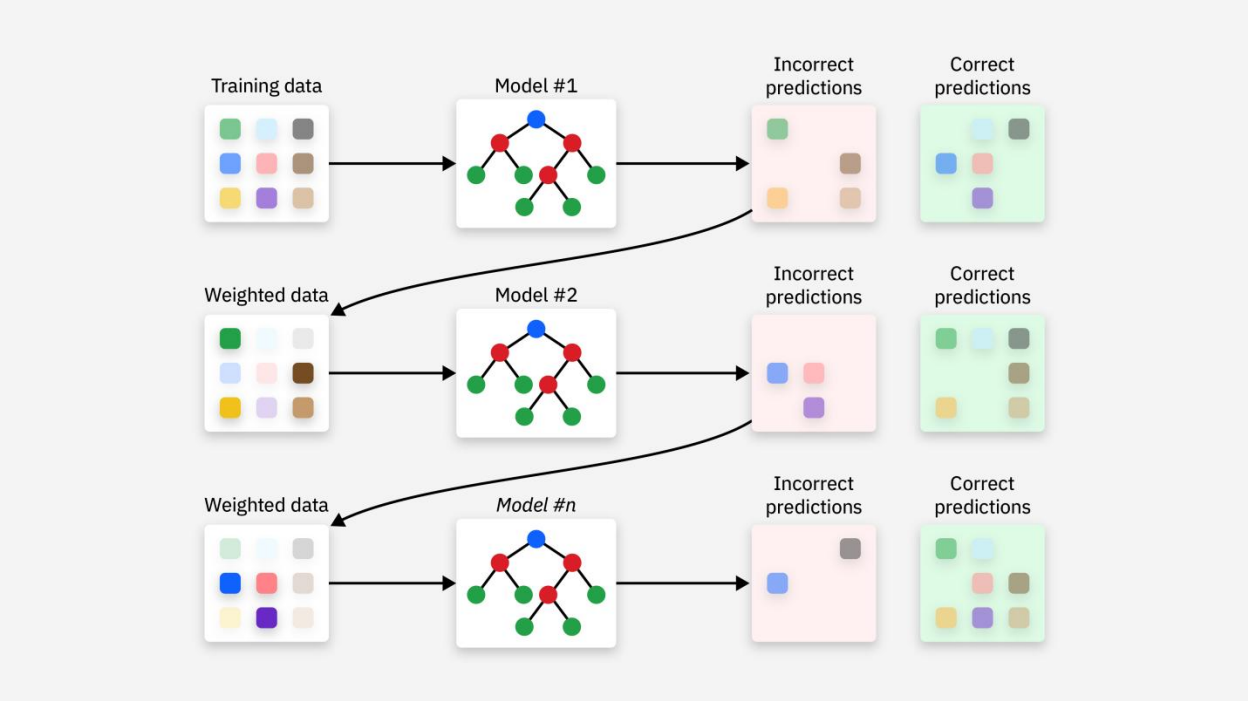
Артықшылықтары:

- Жоғары дәлдік: Көптеген мәселелерде ең жоғары дәлдікке қол жеткізеді.
- Тиімділік: Қателерді біртіндеп, жүйелі түрде азайтады.

Кемшіліктері:

- Артық оқытуға сезімтал: Егер параметрлер (тереңдік, ағаш саны, оқыту жылдамдығы) дұрыс таңдалмаса, Random Forest-қа қарағанда артық оқытуға бейім.
- Параллельдендіру қиын: Модельдер бірізді оқытылатындықтан, есептеуді параллельдендіру Random Forest-қа қарағанда қиынырақ.

Сурет 5.3: Gradient Boosting архитектурасы (Бустинг)



Практикалық есептер

Шешімдер ағаштары мен ансамбльдік әдістер іс жүзіндегі күрделі мәселелерді шешуде шешуші рөл атқарады.

Сала	Мәселе	Қолданылатын әдіс
Қаржы	Несие тәуекелін бағалау	Random Forest (Тұрақтылық үшін), Gradient Boosting (Дәлдік үшін)
Денсаулық сақтау	Аурудың диагностикасын болжау	Шешімдер ағашы (Интерпретациялау үшін)
Маркетинг	Клиенттердің кетуін (Churn) болжау	Random Forest (Ең маңызды факторларды анықтау үшін)
E-commerce	Клиенттің тауар сатып алу ықтималдығын болжау	Gradient Boosting (Жоғары болжамдық дәлдік үшін)

Сала	Мәселе	Қолданылатын әдіс
Өндіріс	Жабдықтардың істен шығуын болжау (Predictive Maintenance)	Ансамбльдік әдістер

Мысал (Feature Importance): Random Forest алгоритмінің маңызды артықшылығы — ол модельдің болжамына әрбір ерекшеліктің (атрибуттың) қосқан салымын (Feature Importance) анықтауға мүмкіндік береді. Бұл бизнес-шешімдер қабылдау үшін өте құнды ақпарат болып табылады. Мысалы, несие тәуекелін бағалауда клиенттің "Айлық табысы" "Несие тарихына" қарағанда маңыздырақ екенін анықтау.

Қорытынды

Шешімдер ағашы — қарапайым және интерпретацияланатын негізгі модель болса, Random Forest пен Gradient Boosting сияқты ансамбльдік әдістер оның кемшіліктерін жойып, модельдің болжамдық қабілетін айтарлықтай арттырады.

- Random Forest — тұрақтылықты, дәлдікті және артық оқытудан қорғауды қажет ететін жағдайларға қолайлы.
- Gradient Boosting — ең жоғары болжамдық дәлдікке қол жеткізу қажет болған жағдайда, ең қуатты әдістердің бірі болып табылады, бірақ оны мұқият баптауды қажет етеді.

Бұл әдістер бүгінгі таңда деректерді талдау және машиналық оқыту саласындағы ең тиімді және кеңінен қолданылатын құралдар болып қала береді.

Қолданылған әдебиеттер тізімі

1. Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques (3rd ed.)*. Elsevier. (Шешімдер ағаштарының негіздері, Gini Index, Entropy)
2. Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5–32. (Random Forest алгоритмін сипаттаған бастапқы ғылыми мақала)
3. Friedman, J. H. (2001). *Greedy function approximation: a gradient boosting machine*. The Annals of Statistics, 29(5), 1189–1232. (Gradient Boosting Machine теориялық негізі)
4. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.)*. Springer. (Ансамбльдік әдістерге арналған толыққанды статистикалық оқулық)
5. Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (Қазіргі ең танымал бустинг алгоритмінің сипаттамасы)

