6-дәріс.
Сараптамалық жүйелер
Пән: Интеллектуалды ақпараттық жүйелер
Алдыңғы тақырып: Өнімдік және фреймдік модельдер

Дәрістің мақсаты
Бүгінгі біз интеллектуалды ақпараттық жүйелердің маңызды түрлерінің бірі — **сараптамалық жүйелерді** қарастырамыз.
Алдыңғы дәрісте біз білімді компьютерге қалай ұсынуға болатынын зерттедік.
Өнімдік модельде білімді **«ЕГЕР – ОНДА»** ережелері арқылы сипаттадық.
Фреймдік модельде объектілерді слоттар, мәндер, фасеттер және мұрагерлік арқылы құрылымдадық.
Бірақ осы жерде маңызды сұрақ пайда болады:
Егер бізде білім базасы дайын болса, компьютер оны нақты мәселені шешу үшін қалай пайдаланады?
Мысалы, университетте студент туралы көптеген мәлімет бар делік:

- GPA;
- сабаққа қатысуы;
- академиялық қарызы;
- меңгерген пәндері;
- оқу мәртебесі.

Сонымен қатар тәжірибелі эдвайзерлердің студенттің академиялық жағдайын бағалау ережелері бар.
Осы деректер мен ережелерді біріктіріп:
«Студентке қандай ұсыныс беру керек?»
деген сұраққа жауап беретін жүйе құруға болады.
Осындай жүйе **сараптамалық жүйе** деп аталады.

1. Сараптамалық жүйеге не үшін қажеттілік пайда болды?
Алдымен қарапайым жағдайдан бастайық.
Студент **Нұрсұлтан** екінші курста оқиды делік.
Ол туралы жүйеде келесі ақпарат бар:

Көрсеткіш	**Мән**
GPA	1.85
Сабаққа қатысу	68%
Академиялық қарыз	2 пән
Мәртебе	Белсенді

Егер осы кестені қарапайым ақпараттық жүйеге енгізсек, компьютер оны сақтай алады.
Бірақ біз компьютерден:
«Нұрсұлтанның академиялық жағдайы қандай және оған қандай әрекет ұсыну керек?»
деп сұрасақ, жай деректер базасы бірден жауап бере алмайды.
Неге?
Өйткені деректер базасында **мәлімет бар**, бірақ сол мәліметті қалай интерпретациялау керектігі туралы **сараптамалық білім жоқ**.
Ал тәжірибелі эдвайзер бұл жағдайды көріп:
«GPA төмен, екі академиялық қарызы бар және сабаққа қатысуы да төмен. Бұл студент академиялық тәуекел тобында. Онымен жеке консультация өткізу қажет»,
деп қорытынды жасай алады.
Бұл жерде эдвайзер тек мәліметті оқып отырған жоқ.
Ол өзінің тәжірибесін, білімін және белгілі ережелерді қолданып отыр.

Сондықтан интеллектуалды жүйеге тек:
ДЕРЕКТЕР жеткіліксіз.
Оған:
ДЕРЕКТЕР + БІЛІМ + ҚОРЫТЫНДЫ ШЫҒАРУ МЕХАНИЗМІ қажет.
Осы **үш элемент** бізді сараптамалық жүйеге алып келеді.

2. Сараптамалық жүйе дегеніміз не?
Енді негізгі анықтаманы берейік.
Сараптамалық жүйе — белгілі бір пәндік саладағы сарапшының білімі мен тәжірибесін компьютерлік түрде сақтап, сол білім негізінде мәселелерді шешуге, қорытынды жасауға және ұсыныс беруге арналған интеллектуалды ақпараттық жүйе.
Қарапайым тілмен айтсақ:
Сараптамалық жүйе — белгілі бір сала маманының шешім қабылдау логикасын компьютерлік жүйеде жүзеге асыру әрекеті.
Мұнда «**сарапшы**» деген сөз маңызды.
Сарапшы — белгілі бір салада терең білімі мен тәжірибесі бар маман.
Мысалы:

- дәрігер;
- инженер;
- қаржы сарапшысы;
- агроном;
- заңгер;
- техникалық маман;
- университет эдвайзері.

Сараптамалық жүйенің мақсаты сарапшыны толығымен алмастыру деп түсінбеуіміз керек. Көп жағдайда оның міндеті — маманға **шешім қабылдауға көмектесу**.
Сондықтан жалпы схеманы былай көрсетуге болады:
ПАЙДАЛАНУШЫНЫҢ МӘСЕЛЕСІ
САРАПТАМАЛЫҚ ЖҮЙЕ
БІЛІМДІ ҚОЛДАНУ
ҚОРЫТЫНДЫ / ҰСЫНЫС
ТҮСІНДІРУ

Осы жерде сараптамалық жүйенің қарапайым ақпараттық жүйеден маңызды айырмашылығы көрінеді.

3. Ақпараттық жүйе мен сараптамалық жүйенің айырмашылығы
Екі жүйеде де ақпарат сақталуы мүмкін.
Бірақ олардың негізгі міндеттері әртүрлі.

Критерий	**Қарапайым ақпараттық жүйе**	**Сараптамалық жүйе**
Негізгі міндет	Деректерді сақтау және өңдеу	Білім негізінде шешім шығару
Негізгі ресурс	Деректер	Деректер + сараптамалық білім
Ережелер	Міндетті емес	Негізгі элементтердің бірі
Қорытынды шығару	Шектеулі	Арнайы механизм қолданылады
Ұсыныс беру	Әдетте жоқ	Негізгі функциялардың бірі
Шешімді түсіндіру	Міндетті емес	Маңызды функция
Маман тәжірибесін сақтау	Негізгі мақсат емес	Негізгі мақсаттардың бірі

Мысалы, университеттің ақпараттық жүйесі:

«**Айдананың GPA көрсеткіші — 2.1**» деп көрсетеді.
Ал сараптамалық жүйе:
«**Айдананың GPA көрсеткіші 2.1, сабаққа қатысуы 65% және екі пән бойынша академиялық қарызы бар. Сондықтан академиялық тәуекел деңгейі жоғары. Эдвайзер консультациясы ұсынылады**» деген қорытынды жасай алады.
Демек, бірінші жүйе:
«Не белгілі?» деген сұраққа жауап береді.
Ал сараптамалық жүйе:
«Белгілі мәліметтерден қандай қорытынды шығаруға болады?»
деген сұраққа жауап беруге тырысады.
Енді мұндай жүйенің ішінде қандай компоненттер бар екенін қарастырайық.

4. Сараптамалық жүйенің негізгі архитектурасы
Классикалық сараптамалық жүйеде бірнеше негізгі компонент болады.

Компонент	**Қызметі**
Білім базасы	Сарапшының білімін сақтайды
Фактілер немесе жұмыс жадысы	Нақты есепке қатысты ағымдағы деректерді сақтайды
Логикалық шығару механизмі	Білім мен фактілерді қолданып қорытынды шығарады
Пайдаланушы интерфейсі	Пайдаланушымен әрекеттеседі
Түсіндіру модулі	Неліктен осындай шешім шыққанын түсіндіреді
Білімді алу модулі	Жаңа білімді енгізуге және жаңартуға көмектеседі

Оларды бір-бірінен бөлек қарастырмай, тұтас процесс ретінде елестетейік.
ПАЙДАЛАНУШЫ
↓
ПАЙДАЛАНУШЫ ИНТЕРФЕЙСІ
↓
ФАКТІЛЕР / ЖҰМЫС ЖАДЫСЫ
↕
ЛОГИКАЛЫҚ ШЫҒАРУ МЕХАНИЗМІ
↕
БІЛІМ БАЗАСЫ
↓
ҚОРЫТЫНДЫ
↓
ТҮСІНДІРУ МОДУЛІ
↓
ПАЙДАЛАНУШЫ

Енді осы компоненттердің әрқайсысын жеке қарастырайық.

5. Білім базасы
Сараптамалық жүйенің ең маңызды бөліктерінің бірі — **білім базасы**.
Білім базасы — белгілі бір пәндік салаға қатысты фактілерді, ұғымдарды, байланыстарды, ережелерді және сараптамалық тәжірибені сақтайтын құрылым.
Алдыңғы дәрісте қарастырған өнімдік және фреймдік модельдер дәл осы жерде қолданылады.
Мысалы, университеттік академиялық кеңесші жүйесінде мынадай ережелер болуы мүмкін:
R1
ЕГЕР GPA < 2.0 ЖӘНЕ академиялық қарыз саны ≥ 2 БОЛСА, ОНДА AcademicRisk = High.

R2
ЕГЕР Attendance < 70% ЖӘНЕ AcademicRisk = High БОЛСА, ОНДА AdvisorConsultation = Required.
R3
ЕГЕР AdvisorConsultation = Required БОЛСА, ОНДА студентке эдвайзермен кездесу туралы хабарлама жібер.
Бұл ережелер нақты бір студентке ғана арналмаған.
Олар университет мамандарының жалпы тәжірибесін сипаттайды.
Ертең жүйеге басқа студент енгізілсе, сол ережелер оған да қолданылуы мүмкін.
Осы жерде маңызды айырмашылық бар:
білім базасы — жалпы білімді сақтайды;
жұмыс жадысы — қазіргі нақты жағдайды сақтайды.

6. Фактілер базасы және жұмыс жадысы
Енді студент **Дәуренді** қарастырайық.
Жүйеге келесі фактілер енгізілді:
Student = Дәурен
GPA = 1.92
AcademicDebt = 2
Attendance = 64%
Бұл ақпарат жалпы ереже емес.
Бұл — нақты бір студент туралы ағымдағы мәлімет.
Сондықтан ол жұмыс жадысында сақталады.
Білім базасында:
ЕГЕР GPA < 2.0 ЖӘНЕ AcademicDebt ≥ 2 БОЛСА, ОНДА AcademicRisk = High
деген ереже болса, жүйе Дәурен туралы фактілерді осы ережемен салыстырады.
Екі шарт та орындалды:
GPA = 1.92 < 2.0;
AcademicDebt = 2.
Сондықтан жүйе жаңа қорытынды алады:
AcademicRisk = High
Бұл қорытынды да жұмыс жадысына **жаңа факт** ретінде қосылуы мүмкін.
Яғни:
БАСТАПҚЫ ФАКТІЛЕР
ЕРЕЖЕ
ЖАҢА ФАКТ
Осы процесті кім басқарады деген сұрақ туындайды
Оны **логикалық шығару механизмі** басқарады.

7. Логикалық шығару механизмі
Логикалық шығару механизмі немесе **инференция механизмі** — білім базасындағы ережелер мен жұмыс жадысындағы фактілерді пайдаланып жаңа қорытынды шығаратын сараптамалық жүйенің компоненті.
Оны жүйенің «ой қорыту механизмі» деп қарастыруға болады.
Бірақ компьютер адам сияқты интуитивті ойламайды.
Ол белгілі алгоритм бойынша жұмыс істейді:
1. Фактілерді алады.
2. Сәйкес ережелерді іздейді.
3. Ережелердің шарттарын тексереді.
4. Қолданылатын ережені таңдайды.
5. Қорытынды шығарады.
6. Қажет болса жаңа фактіні жұмыс жадысына қосады.

Бұл біз өткен дәрісте қарастырған:
MATCH → SELECT → ACT (сәйкестендіру-таңдау- әрекет)
циклімен тікелей байланысты.
Сондықтан өнімдік модельді қарастырғанымыз кездейсоқ емес.
Ол сараптамалық жүйенің жұмысын түсінуге негіз дайындады.

8. Тура шығару — Forward Chaining
Сараптамалық жүйеде қорытынды шығарудың екі негізгі бағытын қарастыруға болады.
Біріншісі — **тура шығару**.
Forward Chaining кезінде жүйе қолда бар фактілерден бастайды.
Мысал
Студент **Аружан** туралы:
Attendance = 58%
AcademicDebt = 1
деген фактілер бар.
Білім базасында:
R1
ЕГЕР Attendance < 70% БОЛСА,ОНДА AttendanceRisk = High.
R2
ЕГЕР AttendanceRisk = High БОЛСА,ОНДА Warning = Required.
Алдымен R1 орындалады:
AttendanceRisk = High
деген жаңа факт пайда болады.
Одан кейін осы факт R2 ережесін іске қосады:
Warning = Required
Нәтижесінде:
Attendance = 58%
↓
AttendanceRisk = High
↓
Warning = Required
Бұл — Forward Chaining.
Жүйе:
«Менде осындай фактілер бар. Осы фактілерден қандай қорытынды шығады?»
деген бағытта жұмыс істейді.

9. Кері шығару — Backward Chaining
Екінші тәсіл — **Backward Chaining**.
Бұл жерде жүйе фактілерден емес, **мақсаттан немесе гипотезадан** бастайды.
Мысалы, біз:
«**Мирас академиялық тәуекел тобында ма?**»
деген нақты сұраққа жауап іздейміз.
Жүйе алдымен:
AcademicRisk = High
қорытындысын қандай ереже бере алатынын іздейді.
Мысалы:
ЕГЕР GPA < 2.0ЖӘНЕ AcademicDebt ≥ 2 БОЛСА,ОНДА AcademicRisk = High.
Енді жүйеге екі нәрсені тексеру қажет:
Мирастың GPA көрсеткіші 2.0-ден төмен бе?
және
оның кемінде екі академиялық қарызы бар ма?
Егер екі факт те расталса, гипотеза дәлелденеді.

Схема:
МАҚСАТ
↓
ҚАНДАЙ ЕРЕЖЕ ОНЫ ДӘЛЕЛДЕЙДІ?
↓
ҚАНДАЙ ФАКТІЛЕР ҚАЖЕТ?
↓
ФАКТІЛЕРДІ ТЕКСЕРУ
↓
МАҚСАТТЫ РАСТАУ НЕМЕСЕ РАСТАМАУ
Сонымен:
Forward Chaining:фактілер → қорытынды.
Backward Chaining:мақсат → қажетті фактілер.

10. Қай тәсілді қашан қолданамыз?
Екі тәсілдің біреуін барлық есепке қолдану міндетті емес.
Олардың тиімділігі есептің сипатына байланысты.

Сипаттама	**Forward Chaining**	**Backward Chaining**
Басталу нүктесі	Фактілер	Мақсат
Бағыты	Деректен қорытындыға	Мақсаттан дерекке
Негізгі сұрақ	«Осы фактілерден не шығады?»	«Осы гипотеза дұрыс па?»
Қолайлы жағдай	Көп нәтиже іздеу	Бір нақты мақсатты тексеру
Мысал	Мониторинг	Диагностика

Мысалы, университет жүйесі барлық студенттерді автоматты бақылап:
«**Қандай студенттерге ескерту керек?**»
деп іздесе, **тура шығару** ыңғайлы болуы мүмкін.
Ал эдвайзер:
«**Осы нақты студентті академиялық тәуекел тобына жатқызуға бола ма?**»
деп сұраса, **кері шығару** тиімді болуы мүмкін.

11. Пайдаланушы интерфейсі
Сараптамалық жүйенің тағы бір маңызды бөлігі — **пайдаланушы интерфейсі**.
Өте жақсы білім базасы мен өте жақсы инференция механизмі болғанымен, егер пайдаланушы жүйемен жұмыс істей алмаса, оның практикалық пайдасы азаяды.
Пайдаланушы интерфейсі арқылы адам:

- мәлімет енгізеді;
- сұрақ қояды;
- жүйенің қорытындысын көреді;
- қосымша ақпарат береді;
- түсіндіруді сұрай алады.

Мысалы, медициналық сараптамалық жүйе:
«**Науқастың дене температурасы қандай?**»
деп сұрауы мүмкін.
Пайдаланушы:
«**38.7°C**»
деп жауап береді.
Жүйе:
«**Жөтел бар ма?**»
деп келесі сұрақты береді.
Осылайша жүйе қажетті фактілерді біртіндеп жинайды.
Демек, интерфейс тек әдемі экран емес.

Ол пайдаланушы мен білімге негізделген шешім қабылдау механизмі арасындағы **байланыс арнасы**.

12. Түсіндіру модулі

Сараптамалық жүйелердің маңызды ерекшеліктерінің бірі — өз шешімін белгілі бір деңгейде **түсіндіру мүмкіндігі**.

Мысалы, жүйе:

«Студент академиялық тәуекел тобында»

деді делік.

Пайдаланушы:

«**Неліктен?**»

деп сұрауы мүмкін.

Жүйе:

«GPA = 1.87 және академиялық қарыз саны = 3 болғандықтан R4 ережесі іске қосылды. R4 ережесі бойынша GPA < 2.0 және қарыз саны ≥ 2 болса, академиялық тәуекел деңгейі жоғары деп анықталады»

деп түсіндіре алады.

Бұл өте маңызды.

Өйткені пайдаланушы тек:

«Жүйе осылай айтты»

деген жауапты емес:

«Жүйе неге осындай қорытындыға келді?»

деген жауапты да білгісі келеді.

Сондықтан **түсіндіру модулі** әдетте екі сұраққа жауап беруге көмектеседі:

«Неге?»

Неліктен жүйе осы сұрақты қойды немесе осы фактіні қажет етеді?

«Қалай?»

Жүйе осы қорытындыға қалай келді?

Мысалы:

Қорытынды:

AcademicRisk = High

Түсіндіру:

GPA < 2.0AcademicDebt ≥ 2R4 ережесі іске қосылдыAcademicRisk = High

Бұл сараптамалық жүйені жай ғана «**қара жәшік**» болудан сақтайды.

13. Білімді алу мәселесі

Енді біз өте маңызды мәселеге келеміз.

Сараптамалық жүйенің білім базасы өздігінен пайда болмайды.

Білімді бір жерден алу қажет.

Негізгі ақпарат көзі — **сарапшы**.

Мысалы, өсімдік ауруларын анықтайтын жүйе жасасақ, тәжірибелі агрономдардың білімі қажет.

Университеттің академиялық кеңесші жүйесін жасасақ:

- эдвайзерлер;
- оқытушылар;
- оқу үдерісін ұйымдастыратын мамандар;
- университеттің нормативтік құжаттары

білім көзі болуы мүмкін.

Бірақ сарапшының басындағы тәжірибені компьютерлік ережеге айналдыру оңай емес.

Маман:

«**Мен мұндай жағдайда әдетте студентке қосымша кеңес беремін**»

деуі мүмкін.

Біз оған:
«Қандай жағдайда нақты?»
деп сұрауымыз керек.
Мысалы:

- GPA қандай болғанда?
- академиялық қарыз саны қанша?
- сабаққа қатысу маңызды ма?
- қандай ерекше жағдайлар бар?

Осылайша жасырын тәжірибені нақты ережелерге айналдырамыз.
Бұл процесс **білімді алу** деп аталады.

14. Сарапшы және білім инженері

Сараптамалық жүйені жасау кезінде екі рөлді ажыратқан дұрыс.
Сарапшы
Сарапшы — пәндік саланы терең білетін маман.
Мысалы, агроном:

- өсімдік ауруларын;
- олардың белгілерін;
- топырақ ерекшеліктерін;
- ауа райының әсерін;
- емдеу тәсілдерін

жақсы біледі.
Бірақ ол міндетті түрде программалауды білмеуі мүмкін.
Білім инженері
Білім инженері — сарапшының білімін анықтап, құрылымдап, компьютер өңдей алатын формаға келтіретін маман.
Оның жұмысы:
САРАПШЫНЫҢ ТӘЖІРИБЕСІ
↓
БІЛІМДІ АНЫҚТАУ
↓
ФОРМАЛИЗАЦИЯ
↓
ЕРЕЖЕЛЕР / ФРЕЙМДЕР / БАСҚА МОДЕЛЬ
↓
БІЛІМ БАЗАСЫ
Мысалы, агроном:
«Жапырақ сарғайып, топырақ өте ылғалды болса, мен алдымен тамыр жүйесін тексеремін»
дейді.
Білім инженері оны:
ЕГЕР LeafColor = Yellow ЖӘНЕ SoilMoisture = High ОНДА CheckRootSystem = Yes
сияқты құрылымға айналдыра алады.

15. Білімді алу әдістері

Сарапшының білімін алудың бірнеше әдісі бар.

Әдіс	**Сипаттамасы**
Сұхбат	Сарапшыға арнайы сұрақтар қою
Бақылау	Сарапшының нақты жұмысына бақылау жасау
Құжаттарды талдау	Нұсқаулықтар, стандарттар, есептерді зерттеу
Нақты жағдайларды талдау	Бұрын шешілген мысалдарды қарастыру
«Дауыстап ойлау»	Сарапшы шешім қабылдау кезінде ойын түсіндіреді

Мысалы, инженер жабдықтың ақауын анықтап жатыр делік.
Біз одан тек:
«**Ақаудың себебі қандай?**»
деп сұрамаймыз.
Одан:
«**Бірінші нені тексердіңіз? Неге? Осы көрсеткіш қалыпты болса, келесі қадам қандай? Қандай жағдайда басқа шешім қабылдар едіңіз?**»
деп сұраймыз.
Сонда біз дайын жауапты ғана емес, **шешім қабылдау логикасын** аламыз.

16. Білімді формализациялау
Сарапшыдан алынған білім көбінесе табиғи тілде болады.
Мысалы:
«**Егер студенттің көрсеткіштері төмендеп, сабаққа жиі келмей жүрсе, онымен сөйлесу керек**».
Адам үшін бұл түсінікті.
Бірақ компьютер үшін бірнеше ұғым нақты емес:
«көрсеткіштері төмендеді» деген не?
«жиі келмейді» деген қанша пайыз?
«сөйлесу керек» деген қандай әрекет?
Сондықтан оны нақтылау қажет.
Мысалы:
ЕГЕР GPA < 2.0 ЖӘНЕ Attendance < 70% БОЛСА, ОНДА AdvisorConsultation = Required.
Енді компьютер шарттарды тексере алады.
Осы процесс **білімді формализациялау** деп аталады.

17. Белгісіздік мәселесі
Нақты өмірдегі сараптамалық шешімдер әрдайым:
иә / жоқ
түрінде бола бермейді.
Мысалы, дәрігер:
«**Бұл аурудың болуы мүмкін**»
деуі мүмкін.
Инженер:
«**Ақаудың ең ықтимал себебі — сенсор**»
деуі мүмкін.
Агроном:
«**Белгілерге қарағанда өсімдікте ылғал тапшылығы болуы ықтимал**»
деуі мүмкін.
Яғни сарапшы кейде абсолютті сенімді емес.
Сондықтан кейбір сараптамалық жүйелерде:
- сенімділік коэффициенттері;
- ықтималдықтар;
- анық емес логика;
- басқа белгісіздікті өңдеу тәсілдері

қолданылады.
Мысалы:
Hypothesis A — confidence 0.8
дегеніміз жүйе бұл гипотезаны белгілі бір сенімділік дәрежесімен қолдайтынын көрсетуі мүмкін.
Бірақ мұнда маңызды нәрсе бар:

сенімділік көрсеткіші нақты нені білдіретіні жүйенің моделіне байланысты.
Оны автоматты түрде статистикалық ықтималдық деп қабылдауға болмайды.
Белгісіз біліммен жұмыс істеудің арнайы әдістерін кейін жеке тақырып ретінде тереңірек қарастыруға болады.

18. Қайшылықты білім
Тағы бір проблема — әртүрлі сарапшылардың пікірлері бірдей болмауы мүмкін.
Мысалы:
Сарапшы 1:
Егер GPA < 2.0 болса, студентке міндетті түрде консультация тағайындау керек.
Сарапшы 2:
GPA < 2.0 болуы жеткіліксіз. Алдымен академиялық қарыз санын да тексеру қажет.
Екі сарапшының тәжірибесі әртүрлі болуы мүмкін.
Сонда білім инженері:

- ережелерді салыстыруы;
- қайшылықты анықтауы;
- қосымша шарттарды нақтылауы;
- басымдықтарды анықтауы;
- қажет болса бірнеше сарапшының пікірін біріктіруі

керек.
Демек:
көп ереже = міндетті түрде жақсы жүйе
деген сөз емес.
Білім базасының **сапасы, келісімділігі және өзектілігі** маңызды.

19. Сараптамалық жүйе қалай жұмыс істейді?
Енді барлық компонентті бір мысалда біріктірейік.
Университеттің интеллектуалды академиялық кеңесші жүйесін қарастырайық.
Студент — **Мөлдір**.
1-қадам. Пайдаланушы мәлімет енгізеді
GPA = 1.95 Attendance = 66% AcademicDebt = 2
2-қадам. Мәлімет жұмыс жадысына түседі
Жүйе бұл фактілерді ағымдағы есепке қатысты ақпарат ретінде сақтайды.
3-қадам. Инференция механизмі білім базасын тексереді
R1:
ЕГЕР GPA < 2.0 ЖӘНЕ AcademicDebt ≥ 2 БОЛСА, ОНДА AcademicRisk = High.
Шарттар орындалды.
Жаңа факт:
AcademicRisk = High
4-қадам. Жаңа факт келесі ережені іске қосады
R2:
ЕГЕР AcademicRisk = High ЖӘНЕ Attendance < 70% БОЛСА, ОНДА AdvisorConsultation = Required.
Екі шарт та орындалды.
Сондықтан:
AdvisorConsultation = Required
5-қадам. Жүйе қорытынды береді
«Студенттің академиялық тәуекел деңгейі жоғары. Эдвайзер консультациясы қажет».
6-қадам. Түсіндіру береді
«**Бұл шешім GPA = 1.95, академиялық қарыз саны = 2 және сабаққа қатысу = 66% фактілеріне сүйеніп шығарылды. R1 және R2 ережелері іске қосылды**».
Осы бір мысалда сараптамалық жүйенің негізгі архитектурасын көрдік:

ДЕРЕКТЕР
↓
ЖҰМЫС ЖАДЫСЫ
↓
БІЛІМ БАЗАСЫ
↓
ИНФЕРЕНЦИЯ
↓
ҚОРЫТЫНДЫ
↓
ТҮСІНДІРУ

20. Сараптамалық жүйелердің негізгі түрлері

Сараптамалық жүйелерді атқаратын міндетіне қарай бірнеше топқа бөлуге болады.

Түрі	**Негізгі міндеті**	**Қарапайым мысал**
Диагностикалық	Мәселенің себебін анықтау	Құрылғы ақауын анықтау
Кеңес беруші	Ұсыныс беру	Академиялық кеңесші
Интерпретациялық	Деректерді түсіндіру	Өлшеу нәтижесін бағалау
Жоспарлаушы	Әрекеттер жоспарын құру	Жұмыс ретін жоспарлау
Мониторингтік	Жағдайды бақылау	Жабдық көрсеткіштерін бақылау
Болжаушы	Болашақ жағдайды бағалау	Тәуекелді бағалау
Конфигурациялық	Компоненттерді таңдау	Техникалық жүйені жинақтау

Бір жүйе міндетті түрде тек бір категорияға жатпауы мүмкін.
Мысалы, аграрлық сараптамалық жүйе бір уақытта:

- өсімдік жағдайын бақылай алады;
- ауруды диагностикалай алады;
- өңдеу әдісін ұсына алады.

Сондықтан жіктеу жүйенің **негізгі функциясын түсіну үшін** қолданылады.

21. Диагностикалық сараптамалық жүйе

Диагностика — сараптамалық жүйелердің классикалық міндеттерінің бірі.
Мысалы, компьютер қосылмай қалды делік.
Белгілер:

- power indicator = off;
- fan = off;
- monitor = no signal.

Жүйе бірден:
«**Аналық плата бұзылған**»
деп қорытынды жасамауы керек.
Алдымен бірнеше гипотезаны тексереді.
Мысалы:
R1
ЕГЕР PowerIndicator = Off ЖӘНЕ Fan = Off БОЛСА, ОНДА CheckPowerSupply = Yes.
R2
ЕГЕР PowerSupply = Normal ЖӘНЕ SystemDoesNotStart = Yes БОЛСА, ОНДА CheckMotherboard = Yes.
Яғни диагностика:
БЕЛГІЛЕР
ГИПОТЕЗАЛАР
ТЕКСЕРУ
ЫҚТИМАЛ СЕБЕП
түрінде жүреді.

Осыған байланысты кері тізбектеу диагностикалық есептерде пайдалы болуы мүмкін.

22. Сараптамалық жүйені әзірлеу кезеңдері

Енді осындай жүйені нөлден жасаймыз деп елестетейік.
Бірден программалаудан бастау дұрыс емес.
Жұмысты кезең-кезеңмен жүргіземіз.

1-кезең. Мәселені анықтау

Қандай есепті шешеміз?
Мысалы:
«Студенттің академиялық тәуекел деңгейін анықтау».

2-кезең. Пәндік саланы шектеу

Жүйе нақты нені білуі керек?
Барлық университет жұмысын бір жүйеге енгізуге тырыспаймыз.

3-кезең. Сарапшыларды анықтау

Қажетті білім кімде бар?

4-кезең. Білімді алу

Сұхбат, құжаттарды талдау, бақылау және нақты жағдайларды зерттеу.

5-кезең. Білімді формализациялау

Алынған білімді:

- ережелерге;
- фреймдерге;
- логикалық модельдерге

айналдыру.

6-кезең. Инференция тәсілін таңдау

Forward Chaining пе?
Backward Chaining пе?
Әлде аралас тәсіл қажет пе?

7-кезең. Прототип жасау

Алдымен шағын жұмыс істейтін нұсқа жасалады.

8-кезең. Тестілеу

Жүйенің қорытындысын сарапшының шешімімен салыстырамыз.

9-кезең. Түзету

Қате немесе әлсіз ережелерді өзгертеміз.

10-кезең. Қолдану және білімді жаңарту

Пәндік сала өзгерген сайын білім базасы да жаңартылуы тиіс.
Жалпы схема:

МӘСЕЛЕ
САРАПШЫ
БІЛІМДІ АЛУ
ФОРМАЛИЗАЦИЯ
БІЛІМ БАЗАСЫ
ИНФЕРЕНЦИЯ МЕХАНИЗМІ
ПРОТОТИП
ТЕСТІЛЕУ
ДАЙЫН ЖҮЙЕ
ЖАҢАРТУ

23. Сараптамалық жүйені қалай тексереміз?

Жүйе техникалық тұрғыдан жұмыс істегенімен, оның шешімдері дұрыс болмауы мүмкін.
Сондықтан екі маңызды ұғымды ажыратамыз.

Верификация
«Жүйені дұрыс құрдық па?»

Мысалы:

- ережелер дұрыс жазылған ба;
- қайшылықтар бар ма;
- программа ережелерді дұрыс орындай ма;
- қажетті компоненттер байланысқан ба.

Валидация

«Дұрыс жүйені құрдық па?»

Яғни жүйенің нақты шешімдері сарапшының тәжірибесіне және пәндік саланың талаптарына сәйкес келе ме?

Мысалы, 100 бұрынғы академиялық жағдайды аламыз.

Олардың дұрыс шешімдері тәжірибелі эдвайзерлерге белгілі болсын.

Содан кейін сол 100 жағдайды сараптамалық жүйеге береміз.

Егер жүйе мамандардың шешімдерінен жиі ауытқыса, білім базасын қайта қарастыру қажет.

Сондықтан:

жүйенің жұмыс істеуі ≠ жүйенің дұрыс шешім беруі.

24. Сараптамалық жүйелердің артықшылықтары

Енді олардың не үшін пайдалы екенін жинақтайық.

Артықшылық	**Түсіндірме**
Білімді сақтау	Тәжірибелі маманның білімін жүйеде сақтауға болады
Қолжетімділік	Білімді көптеген пайдаланушы қолдана алады
Жылдамдық	Көп ережені жылдам тексере алады
Тұрақтылық	Бірдей фактілерге бірдей логика қолданады
Түсіндіру	Шешімге қандай ереже арқылы келгенін көрсетуге болады
Оқыту	Жас мамандар жүйенің логикасын зерттей алады
Шешімді қолдау	Сарапшыға қосымша құрал ретінде қызмет етеді

Мысалы, тәжірибелі агроном зейнетке шыққан кезде оның көп жылдық тәжірибесінің бір бөлігі ұйымнан бірге кетуі мүмкін.

Егер сол тәжірибенің маңызды бөлігі формализацияланып, білім базасына енгізілсе, ұйым үшін пайдалы білім сақталады.

Бұл сараптамалық жүйелердің **тарихи маңызды** идеяларының бірі.

25. Сараптамалық жүйелердің шектеулері

Бірақ сараптамалық жүйені «**бәрін білетін электрондық маман**» деп түсінуге болмайды.

Оның шектеулері бар.

Бірінші шектеу — білім базасына тәуелділік

Жүйе өзіне берілген білім шегінде жұмыс істейді.

Егер маңызды ереже жоқ болса, дұрыс қорытынды шықпауы мүмкін.

Екінші шектеу — білімді алу қиын

Тәжірибелі маман өзінің шешім қабылдау логикасын әрдайым нақты түсіндіре алмауы мүмкін.

Үшінші шектеу — білім ескіруі мүмкін

Ережелер, технологиялар, нормативтер өзгеріп отырады.

Төртінші шектеу — ерекше жағдайлар

Нақты өмірде стандартты ережеге сәйкес келмейтін жағдайлар көп.

Бесінші шектеу — «ақылға қонымды» жалпы түсініктің шектеулілігі

Классикалық сараптамалық жүйе нақты пәндік салада жақсы жұмыс істегенімен, одан тыс жағдайларда әлсіз болуы мүмкін.

Алтыншы шектеу — қате білім қате шешім береді

Егер білім базасына дұрыс емес ереже енгізілсе, жүйе сол қате білімді өте жылдам және тұрақты қолдануы мүмкін.

Сондықтан:
Сараптамалық жүйенің сапасы оның программалық кодына ғана емес, білім базасының сапасына да тікелей байланысты.

26. Сараптамалық жүйе және машиналық оқыту
Бұл жерде студенттер жиі қоятын сұрақ бар:
«Егер қазір машиналық оқыту және нейрондық желілер бар болса, сараптамалық жүйелердің айырмашылығы қандай?»
Қарапайым салыстырайық.

Сараптамалық жүйе	**Машиналық оқыту жүйесі**
Білімді көбіне адам береді	Үлгіні көбіне деректерден үйренеді
Ережелер айқын болуы мүмкін	Модель ішкі заңдылықтарды үйренеді
«ЕГЕР–ОНДА» ережелері жиі қолданылады	Статистикалық модель қолданылады
Шешімді түсіндіру салыстырмалы түрде оңай болуы мүмкін	Түсіндіру кейде күрделі
Білім инженері маңызды	Деректер мен модельді дайындау маңызды
Ережені қолмен жаңартуға болады	Модельді қайта оқыту қажет болуы мүмкін

Мысалы, біз:
«Егер GPA < 2.0 және академиялық қарыз ≥ 2 болса, тәуекел жоғары»
деген ережені өзіміз жазсақ, бұл — ережеге негізделген тәсіл.
Ал егер бірнеше жылдағы мыңдаған студенттің деректерін беріп, алгоритмге академиялық тәуекелдің заңдылықтарын деректерден үйретсек, бұл машиналық оқытуға жақын тәсіл.
Бірақ олар бір-біріне қарсы емес.
Қазіргі интеллектуалды жүйелерде:
ЕРЕЖЕЛЕР + ДЕРЕКТЕР + МАШИНАЛЫҚ ОҚЫТУ
бірге қолданылуы мүмкін.
Мұндай жүйелерді **гибридтік тәсілдерге** жатқызуға болады.

27. Сараптамалық жүйе мен үлкен тілдік модельдің айырмашылығы
Қазіргі уақытта ChatGPT сияқты үлкен тілдік модельдер кең таралғандықтан, тағы бір айырмашылықты түсіну пайдалы.
Классикалық сараптамалық жүйеде шешімнің негізгі логикасы көбіне нақты білім базасы мен ережелерге сүйенеді.
Мысалы:
R17 іске қосылды → R24 іске қосылды → қорытынды алынды.
Сондықтан шешім жолын бақылау салыстырмалы түрде ыңғайлы.
Ал үлкен тілдік модель табиғи тілдегі өте үлкен көлемдегі заңдылықтарды үйреніп, жауапты сол үйренген модель негізінде құрады.
Сондықтан екеуінің жұмыс принципі бірдей емес.
Қарапайым түрде:
Сараптамалық жүйе:
«Маған берілген нақты білім мен ережелерді қолданамын».
Үлкен тілдік модель:
«Үлкен көлемдегі үйренілген тілдік және мазмұндық заңдылықтар негізінде жауап құрамын».
Практикада оларды біріктіруге де болады.
Мысалы, тілдік модель пайдаланушымен табиғи тілде сөйлеседі, ал маңызды шешімді арнайы тексерілген ережелер базасы шығарады.
Бұл қазіргі интеллектуалды жүйелер үшін қызықты бағыттардың бірі.

28. Сараптамалық жүйелерді қолдану салалары

Сараптамалық жүйелер әртүрлі салаларда қолданылуы мүмкін.

Медицина

- симптомдарды талдау;
- диагностикалық гипотезалар;
- емдеу бойынша шешімді қолдау.

Ауыл шаруашылығы

- өсімдік ауруларын анықтау;
- топырақ жағдайын бағалау;
- суару немесе тыңайтқыш бойынша ұсыныс.

Техника

- жабдық ақауын диагностикалау;
- техникалық қызмет көрсету;
- конфигурация таңдау.

Қаржы

- тәуекелді бағалау;
- несие шешімдерін қолдау;
- күмәнді операцияларды талдау.

Білім беру

- академиялық тәуекелді анықтау;
- пән таңдау бойынша ұсыныс;
- білім деңгейін бағалау;
- жеке оқу траекториясын қолдау.

Яғни сараптамалық жүйе әсіресе:

көп тәжірибелік ереже бар,
шешім қабылдау белгілі бір білімді қажет етеді,
шешімді түсіндіру маңызды

болған салаларда пайдалы.

29. Толық практикалық мысал: университеттік сараптамалық жүйе

Енді дәрістегі барлық негізгі ұғымдарды бір мысалға біріктірейік.

Студент — **Әсел**.

Фактілер

Көрсеткіш	**Мән**
GPA	2.35
Attendance	61%
AcademicDebt	2
FailedMandatoryCourse	Yes
PreviousWarning	Yes

Бір қарағанда GPA 2.0-ден жоғары.

Сондықтан тек GPA-ға қарасақ:

«Үлкен мәселе жоқ»

деп қате ойлауымыз мүмкін.

Бірақ сараптамалық жүйе бірнеше факторды бірге қарастырады.

Білім базасы

R1

ЕГЕР Attendance < 70% ОНДА AttendanceRisk = High.

R2

ЕГЕР AcademicDebt ≥ 2 ЖӘНЕ FailedMandatoryCourse = Yes БОЛСА, ОНДА StudyProgressRisk = High.

R3

ЕГЕР AttendanceRisk = High ЖӘНЕ StudyProgressRisk = High БОЛСА, ОНДА AcademicSupport = Required.

R4

ЕГЕР AcademicSupport = Required ЖӘНЕ PreviousWarning = Yes БОЛСА, ОНДА PriorityConsultation = Required.

Енді жүйенің жұмысын бақылайық.

Attendance = 61%.

Сондықтан R1 іске қосылады:

AttendanceRisk = High.

AcademicDebt = 2 және FailedMandatoryCourse = Yes.

R2 іске қосылады:

StudyProgressRisk = High.

Енді R3 үшін екі факт те бар:

AttendanceRisk = High;

StudyProgressRisk = High.

Сондықтан:

AcademicSupport = Required.

Ал **PreviousWarning = Yes** болғандықтан, **R4** те орындалады:

PriorityConsultation = Required.

Соңғы ұсыныс:

Әселге басым тәртіппен академиялық консультация тағайындау ұсынылады.

Бұл мысалда маңызды нәрсеге назар аударыңыз.

Шешім тек бір көрсеткіштен шыққан жоқ.

Бірнеше бастапқы факт бірнеше ережеден өтті және **аралық қорытындылар** келесі ережелер үшін жаңа фактілерге айналды.

Жүйенің ой қорыту тізбегі:

Attendance = 61%

→ AttendanceRisk = High

және

AcademicDebt = 2 + FailedMandatoryCourse = Yes

→ StudyProgressRisk = High

↓

AcademicSupport = Required

↓

PreviousWarning = Yes

↓

PriorityConsultation = Required

Міне, бұл сараптамалық жүйенің жұмыс логикасын жақсы көрсететін мысал.

30. Сараптамалық жүйені жобалауда жиі кездесетін қателер

Енді студенттер жиі жіберетін бірнеше қатені қарастырайық.

1. Сараптамалық жүйені жай деректер базасы деп қабылдау

Деректерді сақтау жеткіліксіз.

Жүйеде білімді қолданатын механизм болуы керек.

2. Барлық білімді бір үлкен ережеге жазу

Мысалы:

«Егер GPA төмен, сабаққа қатысу төмен, қарыз бар, ескерту бар, студент екінші курста болса...»

деп өте ұзын ереже жазу жүйені басқаруды қиындатады.

Білімді логикалық шағын ережелерге бөлу тиімдірек.

3. Нақты емес ұғымдарды пайдалану

Студент нашар оқыса..

Компьютер «**нашар**» дегенді түсінбейді.
Оны нақты критериймен көрсету қажет.
4. Қайшылықтарды тексермеу
Бір ереже:
«Консультация қажет»
десе, екіншісі сол жағдайда:
«Консультация қажет емес»
деп тұрмауы керек немесе мұндай қайшылықты шешетін басымдық болуы тиіс.
5. Түсіндіруді ұмыту
Пайдаланушыға тек қорытынды беру жеткіліксіз болуы мүмкін.
Неге?
деген сұраққа жауап беру маңызды.
6. Білім базасын жаңартпау
Ескі ережелер уақыт өте келе дұрыс емес шешім бере бастайды.

31. Дәрістің негізгі схемасы
Бүгінгі тақырыпты бір толық схемаға жинақтайық.
ПӘНДІК САЛА
↓
САРАПШЫ
↓
БІЛІМ ИНЖЕНЕРІ
↓
БІЛІМДІ АЛУ
↓
БІЛІМДІ ФОРМАЛИЗАЦИЯЛАУ
↓
БІЛІМ БАЗАСЫ
↕
ИНФЕРЕНЦИЯ МЕХАНИЗМІ
↕
ФАКТІЛЕР / ЖҰМЫС ЖАДЫСЫ
↓
ҚОРЫТЫНДЫ / ҰСЫНЫС
↓
ТҮСІНДІРУ МОДУЛІ
↓
ПАЙДАЛАНУШЫ
Осы схеманы түсінсек, сараптамалық жүйенің негізгі логикасын түсіндік деп айтуға болады.

32. Қорытынды
Сонымен, бүгін біз **сараптамалық жүйелерді** қарастырдық.
Ең алдымен мына негізгі ойды есте сақтайық:
Сараптамалық жүйе — жай ғана деректер сақтайтын программа емес.
Ол белгілі бір пәндік саладағы білімді сақтайды және сол білімді нақты мәселені шешу үшін қолданады.
Сараптамалық жүйенің негізгі компоненттері:
- білім базасы;
- фактілер немесе жұмыс жадысы;
- логикалық шығару механизмі;
- пайдаланушы интерфейсі;

- түсіндіру модулі;
- білімді алу және жаңарту құралдары.

Білім базасында сарапшының тәжірибесі:

- ережелер;
- фактілер;
- объектілер;
- байланыстар

түрінде берілуі мүмкін.
Ал инференция механизмі осы білімді нақты жағдайға қолданады.
Біз екі негізгі бағытты қайталадық:
Forward Chaining:
фактілер → қорытынды
және
Backward Chaining:
мақсат → қажетті фактілер.
Сонымен қатар жақсы сараптамалық жүйе тек қорытынды беріп қоймай, мүмкіндігінше:
«Неге осындай шешім қабылданды?»
деген сұраққа да жауап беруі керек екенін көрдік.
Сараптамалық жүйені құрудағы ең күрделі мәселелердің бірі — **білімді алу**.
Өйткені тәжірибелі маманның көп жылдық тәжірибесін нақты және бір-біріне қайшы келмейтін компьютерлік ережелерге айналдыру оңай емес.
Сондықтан **сарапшы мен білім инженері** бірге жұмыс істейді.
Біз сондай-ақ сараптамалық жүйелердің машиналық оқытудан айырмашылығын қарастырдық.
Классикалық сараптамалық жүйеде:
білімді көбіне адам береді,
ал машиналық оқытуда:
заңдылықтар деректерден үйреніледі.
Қазіргі интеллектуалды жүйелерде бұл тәсілдер бір-бірін толықтыра алады.
Сондықтан бүгінгі дәрістің негізгі тұжырымын бір сөйлеммен айтсам:
Сараптамалық жүйе — сарапшы білімін формализациялап, оны нақты жағдайға қолдану арқылы негізделген қорытынды немесе ұсыныс беретін интеллектуалды жүйе.

33. Бақылау сұрақтары

1. Сараптамалық жүйе дегеніміз не?
2. Сараптамалық жүйенің қарапайым ақпараттық жүйеден айырмашылығы қандай?
3. Сараптамалық жүйенің негізгі компоненттерін атаңыз.
4. Білім базасының қызметі қандай?
5. Жұмыс жадысы қандай ақпаратты сақтайды?
6. Инференция механизмі дегеніміз не?
7. Білім базасы мен деректер базасының айырмашылығы қандай?
8. Forward Chaining қалай жұмыс істейді?
9. Backward Chaining қалай жұмыс істейді?
10. Forward және Backward Chaining арасындағы негізгі айырмашылық қандай?
11. Түсіндіру модулі не үшін қажет?
12. Білімді алу дегеніміз не?
13. Сарапшы дегеніміз кім?
14. Білім инженері қандай қызмет атқарады?
15. Сарапшы білімін алудың қандай әдістері бар?
16. Білімді формализациялау не үшін қажет?
17. Белгісіздік сараптамалық жүйеде қандай мәселе туғызады?

18. Сараптамалық жүйені әзірлеудің негізгі кезеңдері қандай?
19. Верификация мен валидацияның айырмашылығы неде?
20. Сараптамалық жүйелердің негізгі артықшылықтары қандай?
21. Сараптамалық жүйелердің қандай шектеулері бар?
22. Сараптамалық жүйе мен машиналық оқытудың айырмашылығы қандай?
23. Сараптамалық жүйелер қандай салаларда қолданылады?
24. Неліктен білім базасын үнемі жаңартып отыру қажет?
25. Өнімдік модель сараптамалық жүйемен қалай байланысады?

34. Сабақ соңындағы шағын тапсырма

Енді шағын жағдайды өздеріңіз талдап көріңіздер.

Университетте студентке **жеке оқу қолдауы қажет пе** екенін анықтайтын сараптамалық жүйе жасалып жатыр.

Студент **Данияр** туралы:

Көрсеткіш	**Мән**
GPA	2.42
Сабаққа қатысу	67%
Академиялық қарыз	1
Соңғы семестрдегі GPA өзгерісі	−0.55
Міндетті пәннен қайта тапсыру	бар
Бұрынғы академиялық ескерту	жоқ

Тапсырма

1. Осы жағдайды бағалау үшін қандай қосымша ақпарат қажет болуы мүмкін екенін анықтаңыз.

2. Кемінде **3 өнімдік ереже** құрыңыз.

Бір ғана көрсеткішке сүйенетін өте қарапайым ережелермен шектелмеңіз.

3. Жүйенің жұмыс жадысындағы бастапқы фактілерді көрсетіңіз.

4. Қай ережелер іске қосылатынын анықтаңыз.

5. Жүйенің соңғы ұсынысын жазыңыз.

6. Жүйе пайдаланушыға:

«Неліктен осындай ұсыныс берілді?» деген сұраққа қалай жауап беруі керектігін көрсетіңіз.

Бұл тапсырмада бір ғана дайын дұрыс шешім жоқ. Ең маңыздысы — құрған ережелеріңіздің логикалық негізді болуы және жүйенің қорытындысын фактілермен түсіндіре алуыңыз.