Жамбыл облысы әкімдігі білім басқармасы

«Білім» кәсіби гуманитарлық-техникалық колледжі

(білім беру ұйымының атауы)

**Оқу сабағының жоспары**

(теориялық немесе өндірістік оқыту)

**Бустинг циклдарын зерттеу.**

(сабақ тақырыбы)

**Модуль/Пән атауы** Бағдарламалық кодты қайта өңдеу

**Дайындаған педагог** Нургисаева У.М

**20\_**25**\_ жылғы** «\_\_\_\_» \_\_\_наурыз\_\_\_

**1. Жалпы мәліметтер**

Курс, оқу жылы, топ 3 курс, 3БҚ-1-22, 3БҚ-2-22

Сабақ түрі: Аралас

**2. Мақсаты, міндеттері:**

**Оқу:** Білім алушыларға python (пайтон) программалау тілінде Бустинг циклдарын зерттеу, екі өлшемді массивтерге есеп шығаруды жүзеге асыру, қолдану жолдарын ұйымдастыру, программаны өңдеу жұмыснегіздерін үйрету

#### Дамыту: Логикалық ойлау қабілеттерін дамыту, есептердің әртүрлі шешімдерін табуға дағдыландыру, программалық кодтың құрылымын дұрыс құру қабілеттерін қалыптастыру.

#### Тәрбиелік: Жауапкершілік пен ұқыптылыққа тәрбиелеу, алгоритмдік ойлауды дамыту, шығармашылық қабілеттерін жетілдіру.

**3. Оқу-жаттығу процесінде білім алушылар меңгеретін күтілетін нәтижелер және кәсіби дағдылар тізбесі:** Python программалау тілінде Бустинг циклдарын зерттеу, екі өлшемді массивтермен жұмыс, есеп шығаруды жүзеге асыру, қолдану жолдарын ұйымдастыру, программаны өңдеу және пайдалану дағдыларын меңгеру.

**4. Қажетті ресурстар:** ДК немесе ноутбуктер

Ю.Аляев, О.Козлов. Алгоритмизация и языки программирования  Python, C++, Visual Basic: Учебно-справочное пособие. – М.: Финансы и статистика, 2004

<https://ust.kz/powerpoint/eki_olsemdi_massivter-386835.html>

***5. Сабақтың барысы: (90 минут)***

**5.1. Ұйымдастыру кезеңі:** *( 3 мин )*

**5.2. Үй жұмысын жан-жақты тексеру:**

***“Алгоритм” әдісі*** *(15 минут)*

**Бустинг циклдарын зерттеу.**

Бустинг (Boosting) – бұл әлсіз классификаторлардың жиынтығын күшейтіп, күшті модель құруға мүмкіндік беретін ансамбльдік оқыту әдісі. Ол машиналық оқытудың маңызды техникаларының бірі болып табылады және әсіресе үлкен деректер жиынтықтарында жоғары өнімділік көрсетеді.

Бустингтің негізгі идеясы – бірнеше әлсіз модельдерді тізбекті түрде үйрету, әр қадамда алдыңғы модель қателіктерін түзету арқылы жалпы болжамның дәлдігін арттыру.

**2. Бустингтің негізгі принциптері**

Бустинг алгоритмдері келесі негізгі принциптерге сүйенеді:

* Бірнеше әлсіз модельдер қолданылады (мысалы, шешім ағаштары).
* Модельдер тізбекті түрде үйренеді, ал әр жаңа модель алдыңғы модель жіберген қателерді түзетуге тырысады.
* Соңында барлық модельдердің болжамдары біріктіріліп, күшейтілген болжам жасалады.

**3. Бустинг циклдарының құрылымы**

Бустинг циклдарын зерттегенде оның әрбір кезеңін талдау маңызды. Бір цикл келесі қадамдардан тұрады:

1. **Бастапқы салмақтарды белгілеу** – Барлық деректер нүктелеріне бірдей салмақ беріледі.
2. **Әлсіз модель құру** – Шешім ағашы немесе басқа жеңіл модельдер бастапқы деректер бойынша оқытылады.
3. **Қателіктерді бағалау** – Жаңа модельдің болжамдары мен нақты нәтижелер арасындағы айырмашылық есептеледі.
4. **Салмақтарды жаңарту** – Қате жіберілген үлгілердің салмағы артады, осылайша келесі модель сол қателерді түзетуге тырысады.
5. **Келесі әлсіз модельді құру** – Алдыңғы модельдің кемшіліктерін түзету үшін жаңа модель үйретіледі.
6. **Қорытынды модель құру** – Барлық әлсіз модельдердің нәтижелері біріктіріліп, соңғы күшейтілген модель жасалады.

**4. Негізгі бустинг алгоритмдері**

Бустингтің ең танымал алгоритмдеріне келесілер жатады:

**4.1 AdaBoost (Adaptive Boosting)**

* Бірінші бустинг алгоритмі.
* Қате жіберген үлгілердің салмағын арттыру арқылы әлсіз модельдерді жақсартады.
* Әдетте қысқа шешім ағаштарын қолданады.

**4.2 Gradient Boosting Machines (GBM)**

* Градиентті еңкіштікті төмендету әдісін пайдаланады.
* Алдыңғы модель жіберген қателерге негізделген жаңартулар енгізіледі.
* Қуатты, бірақ есептеу тұрғысынан ауыр алгоритм.

**4.3 XGBoost (Extreme Gradient Boosting)**

* GBM-нің жетілдірілген нұсқасы.
* Орындалу жылдамдығы жоғары және жадыны тиімді пайдаланады.
* Үлкен деректер жиынтықтарында кеңінен қолданылады.

**4.4 LightGBM (Light Gradient Boosting Machine)**

* XGBoost-қа қарағанда жылдам әрі жеңіл.
* Үлкен көлемді деректермен жақсы жұмыс істейді.
* Терең шешім ағаштарын қолдануға мүмкіндік береді.

**4.5 CatBoost (Categorical Boosting)**

* Категориялық деректермен тиімді жұмыс істейді.
* Деректерді алдын ала түрлендіруді қажет етпейді.
* Есептеу жылдамдығы жоғары.

**5. Бустинг циклдарын зерттеу әдістері**

Бустинг циклдарын терең зерттеу үшін келесі әдістер қолданылады:

1. **Гиперпараметрлерді баптау** – Оптималды параметрлерді таңдау арқылы модельдің тиімділігін арттыру.
2. **Қателіктерді талдау** – Әрбір қадамдағы қателіктерді бағалау арқылы модельдің жұмысын жақсарту.
3. **Айқаспалы тексеру (Cross-validation)** – Жалпыға жақсы жалпылау қабілетін қамтамасыз ету.
4. **Ерте тоқтату (Early stopping)** – Артық үйретуді (overfitting) болдырмау үшін оқытуды уақытында тоқтату.
5. **Функция маңыздылығын талдау** – Қандай белгілер модельге ең үлкен әсер ететінін анықтау.

**6. Қолдану салалары**

Бустинг алгоритмдері келесі салаларда кеңінен қолданылады:

* **Қаржы саласы** – Несие беру тәуекелін бағалау.
* **Медицина** – Ауруларды диагностикалау.
* **Маркетинг** – Тұтынушылардың мінез-құлқын болжау.
* **Киберқауіпсіздік** – Алаяқтық әрекеттерді анықтау.
* **Табиғи тілдерді өңдеу (NLP)** – Сөйлемдерді жіктеу, мәтіндік талдау.

**Шешім ағашы – Дерево решений**

**Бустингті түсіну үшін алдымен шешім ағашын түсіну керек. Бұл өте қарапайым ағаш:**



Дәл қазір бұл шешім ағашы, бірақ болжау ағашы болуы мүмкін. Ағаштың атауы «Пайдаланушы аты серуендеуге шығады ма?» деп жазылғанын елестетіп көріңіз. — және сіз дұрыс сұрақтар қойған жағдайда, Пайдаланушы аты мен ауа-райы туралы деректер негізінде Пайдаланушы аты туралы дәл болжам жасайтын болжау машинасын аласыз.

## **Болжау машинасы**

Енді мысал күрделірек. YouTube-те миллиондаған музыкалық бейнелер туралы деректер бар делік. Әрқайсысы үшін 100 критерий бар, мысалы:

* Клип үш минуттан ұзақ па?
* Онда түзу бөшке бар ма?
* Бұл трек хип-хоп па, жоқ па?
* Бейне танымал лейбл арқылы шығарылды ма?
* Клип аулада ұялы телефонға түсірілді ме?
* …

Сондай-ақ бізде клиптің миллионнан астам қаралғаны туралы деректер бар. Біз бұл критерийді болжауды үйренгіміз келеді - оны танымалдылық деп атаймыз. Яғни, сіз иә/жоқ пішімінде 100 клип критерийін беретін қандай да бір алгоритм түрін алғымыз келеді және нәтижеде ол сізге: «Бұл клип танымал болу керек» деп айтады.

Миллион клиптер, жүз критерий - алгоритм үшін жаттығу үлгісі

## **Бірінші мәселе**

Егер бұл тапсырма бойынша болжау ағашын құрастырғымыз келсе, алдымызда мәселе туындайды: біз қандай критерийлерді басына және қайсысын аяғына қою керектігін, қайсысын қай бұтақтарға қою керектігін білмейміз, ал қайсысы мүлдем қажет емес, сондықтан ешқандай әсер етпейді.

[](https://thecode.media/wp-content/uploads/2020/08/3-1.jpg)

## **Шешімнің бірінші қадамы**

Мұндай мәселеге ағаш құру үшін алдымен әрбір критерийдің қажетті нәтижеге қаншалықты жақын екенін есептеу керек:

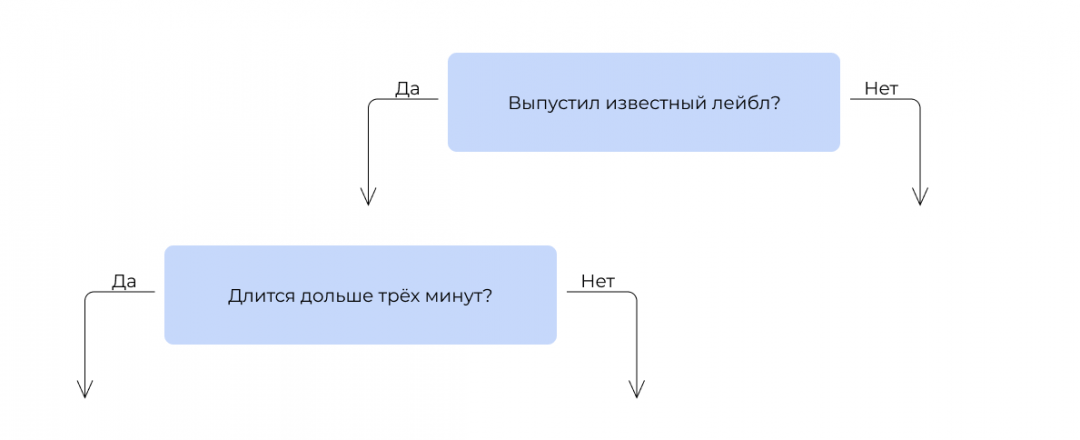
Нәтижесінде соңғы танымалдылықпен ең күшті байланыс «Бейне танымал белгімен шығарылды ма» критерийі үшін екенін көреміз. Бейне жапсырма арқылы шығарылса, бұл барабаннан немесе автотюден гөрі сәттілікке көбірек әсер етеді екен. Бұл критерий ағаштың басына көтеріледі.

[](https://thecode.media/wp-content/uploads/2020/08/5.jpg)

Содан кейін біз қандай критерий қою керектігін қарастырамыз. Біз белгілер шығарған 300 000 бейнені алып, қалған критерийлер арқылы іске қосамыз. Біз соңғы болжамның ең жоғары дәлдігін беретінін іздейміз.

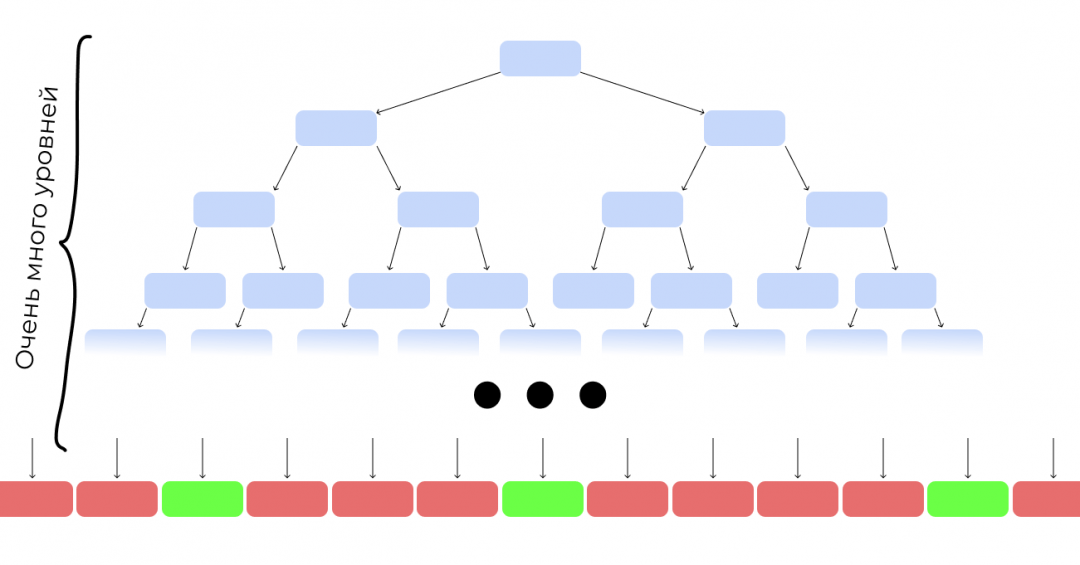
Жапсырмалар шығарған 300 000 бейненің 250 000-ы 3 минуттан ұзақ және олардың 90 000-ы миллионнан астам қаралған. Бұл критерий белгі бейнелерінің танымалдығының ең жақсы болжамы болып табылады.

Біз оны екінші орынға қойдық.

[](https://thecode.media/wp-content/uploads/2020/08/7.png)

Басқа филиал үшін де солай істейміз. Осылайша біз қалған критерийлер бойынша тізбекті құрастырамыз. Іс жүзінде бұл қолмен жасалмайды: мұны автоматты түрде жасайтын арнайы алгоритмдер бар.

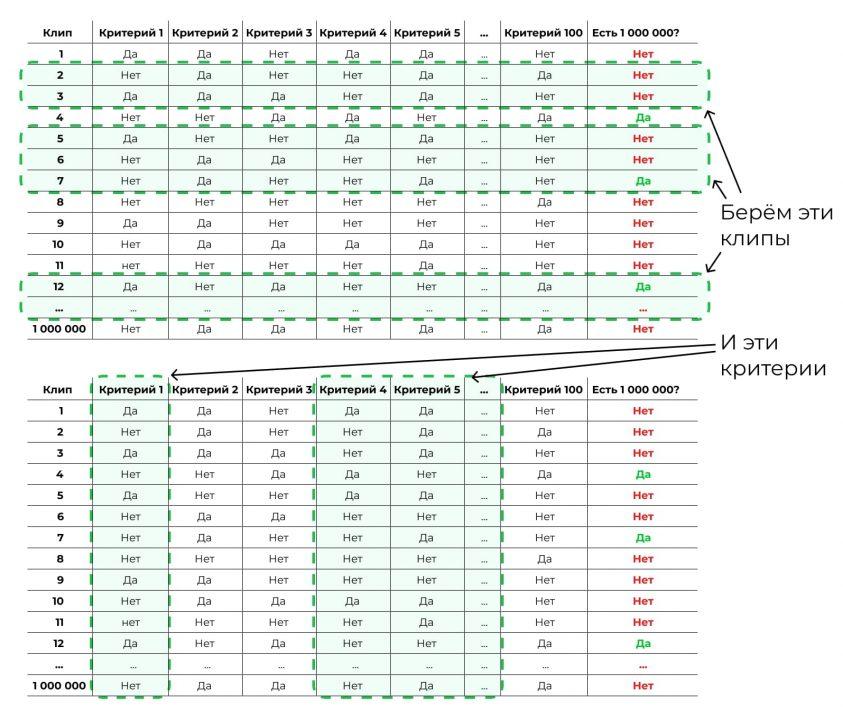
Ой, бізде шешім ағашы бар:

Бейнелердің көпшілігі шықпайды, бірақ кейбіреулері танымал болады.

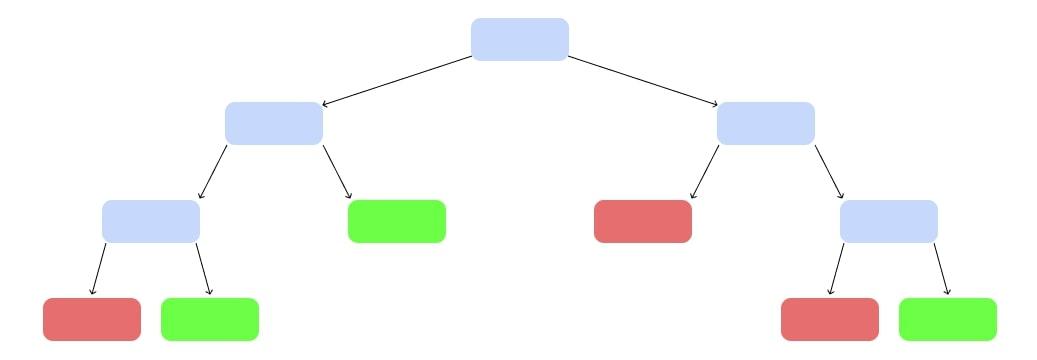
## **Кездейсоқ орман**

Мәселе бар: осылай салынған ағаш өте күрделі және өте дәл емес. Бір үлкен ағаш емес, бірнеше кішкентай ағаштар жасауға тырысайық.

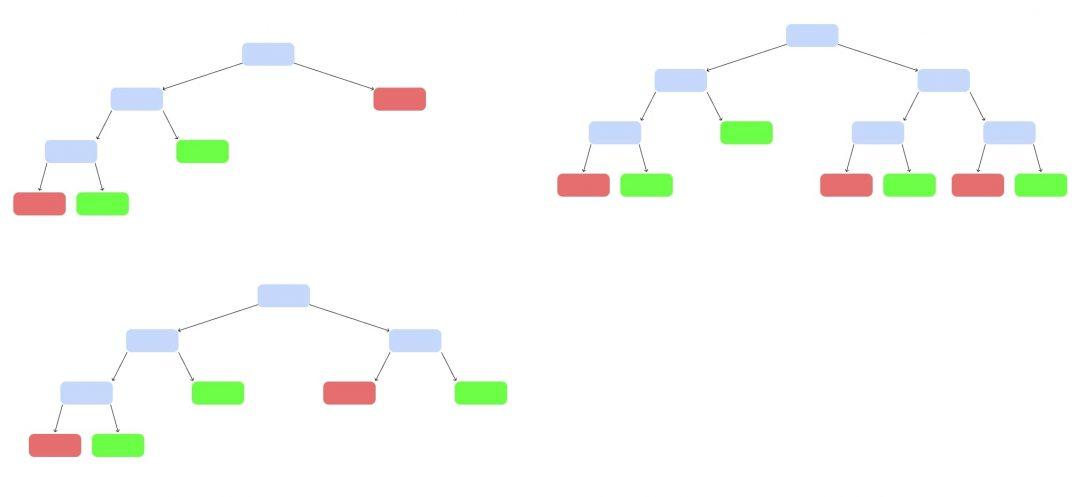
Бастапқы деректерден кездейсоқ үлгіні алайық. Миллион емес, 10 000 клип. Оларға - критерийлердің кездейсоқ жиынтығы, барлығы 100 емес, 5:

[](https://thecode.media/wp-content/uploads/2020/08/9.jpg)

Ал қарапайым ағашты құрастырайық:

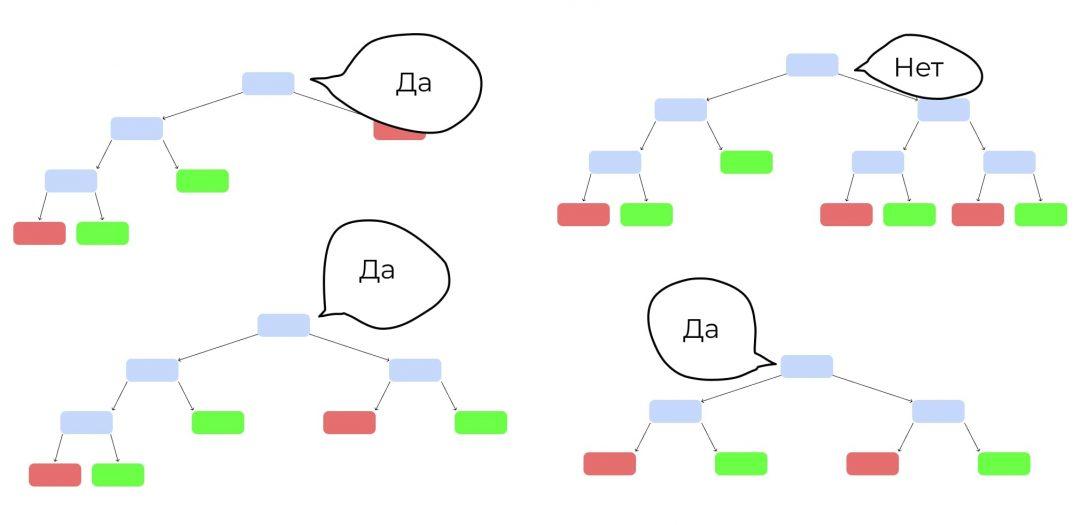
Деңгейлері көп болды, қазір азайып кетті

Ендеше, әрқайсысы өз деректері мен өз критерийлері бойынша тағы бірнеше ағаш құрастырайық:

Алгоритмдер орманы

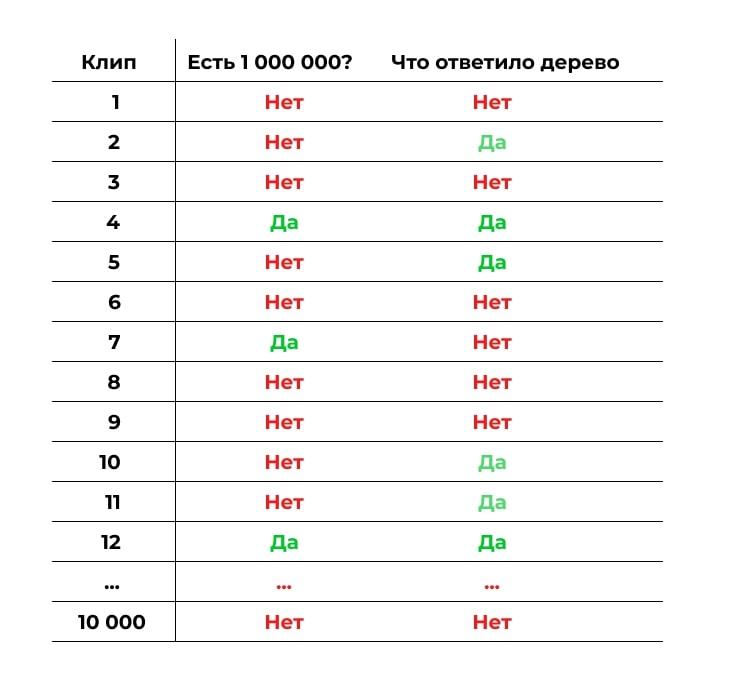
Қазір бізде кездейсоқ орман бар. Кездейсоқ – өйткені біз әр жолы кездейсоқ деректер мен критерийлер жинағын алатынбыз. Орман – өйткені ағаштар көп.

Енді жаттығу үлгісінде болмаған клипті орындайық. Әр ағаш танымал бола ма, жоқ па, соған байланысты өз үкімін береді. Сайлаудағы дауыс беру сияқты. Біз ең көп дауыс жинаған опцияны таңдаймыз.

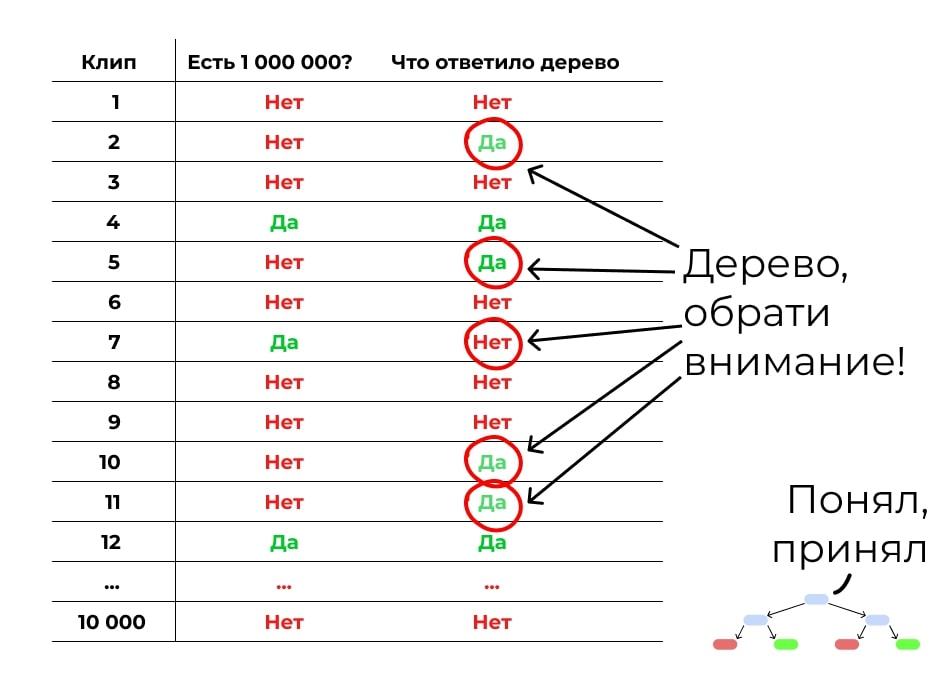
Үш жақта, бір қарсы - бейне сәтті болады. Мүмкін

## **Кездейсоқ емес орман - күшейту**

Енді ұқсас орманды құрастырайық, бірақ деректер жиынтығы кездейсоқ емес болады. Кездейсоқ деректер мен кездейсоқ критерийлерді пайдалана отырып, біз бірінші ағашты бұрынғыдай жасаймыз. Содан кейін біз осы ағаш арқылы бақылау үлгісін іске қосамыз: бізде барлық деректер бар, бірақ жаттығуға қатыспаған басқа клиптер. Ағаш қай жерде қателескенін көрейік:

Бірінші ағаш көптеген қателер бере алады

Енді келесі ағашты жасаймыз. Бірінші ағаш қателескен жерлерге назар аударайық. Оқыту үшін деректер мен критерийлерді таңдау кезінде осы қателерге көбірек мән берейік. Тапсырма – алдыңғысының қатесін түзететін ағаш жасау.

Ағашты алдыңғысының қателерін түзетуге үйрету

Бірақ екінші ағаш өз қателерін жасайды. Үшіншісін жасаймыз, ол оларды түзетеді. Содан кейін төртінші. Содан кейін бесінші. Сіз идеяны түсінесіз.

Біз мұндай ағаштарды қажетті дәлдікке жеткенше немесе шамадан тыс орнату салдарынан дәлдік төмендей бастағанша жасаймыз. Бізде ағаштар өте көп, олардың әрқайсысы онша мықты емес екен. Бірақ олар бірге жақсы дәлдік беретін орманды құрайды. Көтеруде!



***Тапсырмалар***

**1-Есеп: Спираль бойынша матрицаны толтыру**  
Берілген **n × n** өлшемді матрицаны **1**-ден **n²**-ге дейінгі сандармен **сағат тілімен** спираль бойымен толтырыңыз.

**2-Есеп: Матрицаны сағат тілімен 90° бұру**  
Берілген **n × n** матрицаны **90 градусқа** сағат тілімен бұрыңыз.

**3-Есеп: Екі матрицаны көбейту**  
Берілген **A(m × p)** және **B(p × n)** матрицаларын көбейтіп, **C(m × n)** нәтижелік матрицасын табыңыз.

**4-Есеп: Матрицаның барлық жолдары бойынша максималды мәндерді табу**  
Берілген **n × m** матрицаның әрбір жолындағы ең үлкен элементтерді табыңыз.

**5-Есеп: Екіөлшемді массивте ең ұзын өспелі жолды табу**  
Матрицадағы **көршілес сандар 1-ге артып отыратын** ең ұзын жолды табыңыз.

|  |  |  |
| --- | --- | --- |
| **Білдім** | **Білемін** | **Білгім келеді** |
|  |  |  |

**Рефлексия**

**Үй тапсырмасы –** Зертханалық жұмыс №8 орындау және қорғау