Машинаны Үйрету

Мағжан
ҚАЙРАНБАЙ

Машинаны үйрету

Кітаптың жазылуына әкем — жазушы Жұмабай Қайранбай Қожақынұлының ақыл-кеңестері себепші болды. Сонымен бірге жазылып біткен осы кітапты бастан-аяқ оқып, редакциялауға өз үлесін қосқан Арман Жармағамбетовке және Жасан Байпакбаевқа алғысым шексіз. Кітап 2021 жылдың Шілде айынан жазыла бастап, 1.5 жыл көлемінде толық аяқтадым.

Кітаптың жазылу барысында ғалым Эндрью Ыңның еңбектері басшылыққа алынды.

Қайранбай М.Ж.

Алғы сөз

Бұл оқу құралы машинаны үйрету саласына жаңадан аяқ басқан оқушы, студент немесе осы салада жұмыс істеймін деушілерге арналған. Ағылшын және орыс тілдерінде машинаны үйрету саласы бойынша ақпараттар өте көп болғанымен, қазақ тілінде біздің білуімізше ондай ақпараттар жоқ. Сол себепті бұл оқылықтың орнын толтыру үшін осы кітапты жазуды қолға алдық.

Бұл кітаптың басты мақсаты — машинаны үйрету саласын оқырмандарға таныстыра келіп нөлдік деңгейден халықаралық дәрежедегі Junior Machine Learning Engineer деңгейіне дейінгі білім беруді іске асыру. Өз тарапымыздан Junior Machine Learning Engineer деңгейін көздейтін тақырыптарды қарастырып, оларға толықтай түсінік беруге тырыстық. Әр тақырыптағы өз білімдеріңізді тексеру үшін тест сұрақтар мен жаттығулар беріп отырдық. Сол сұрақтардың кітап соңындағы жауабын қарап, өз білімдеріңізді тексере аласыздар. Сонымен бірге, әр тақырып үшін Python тілінде код жазылып, сол тақырыпты теориялық түрде түсініп қана қоймай, сол кодты бізбен бірге жазып, практикалық тұрғыдан білімдеріңізді арттыра алуға қолайлы жағдай жасадық. Кітаптағы әр бөлімді оқып болғаннан кейін https://itboo.kz/mlproblems.php сайтынан сол тақырыпқа байланысты есептер шығара аласыздар. Автоматтандырылған тексеру жүйесі сіздің кодыңызды тестілеу арқылы дұрыс не бұрыс екенін анықтап бере алады. Осылайша, өтілген әрбір тақырыпты толығырақ түсіну үшін осы есептерді шығара отырып, тек теориялық тұрғыдан ғана емес, сонымен бірге практикалық тұрғыдан да түсінік аласыздар. Машинаны үйрету саласында математика пәні көптеп қатысады. Сол себепті математикадан біліміңіз төмендеу болса, оқу құралын еркін түсініп оқу қиынға соғады. Сонымен қатар, кодтарды түсініп, бізбен бірге үлгеріп жазып отыру үшін бастапқы деңгейдегі бағдарламалау тілдерінің бірін жақсы білген қажет. Осы кітапта бағдарламау тілі ретінде Python тілі қолданылғандықтан, оны білсеңіздер тіпті жақсы болар еді. Нөлдік деңгейден Junior Machine Learning Engineer деңгейіне жету үшін кітапты асықпай, мұқият оқи отырып, түсінбеген сұрақтарға жауап іздеп және оларға толық жауап қайырып, берілген есептерді түгелдей шығару керек.

Егер әлемдік деңгейдегі Machine Learning Engineer болуға іштей дайын болсаңыздар, онда қолыңыздағы бұл оқу құралын оқып үйренуге кірісейік.

Кодтың нұсқаулықтары

Бұл кітаптың ішіндегі барлық кодтар Python тілінде жазылған. Python тілі машинаны үйрету саласындағы ең негізгі тіл болып табылады. Ол басқа тілдерге қарағанда үйрену мен жазуға өте жеңіл тіл болып есептеледі. Python тілін компьютерге орнату процесін өзіңізге қалдырып отырмыз.

Мазмұны

I

бөлім

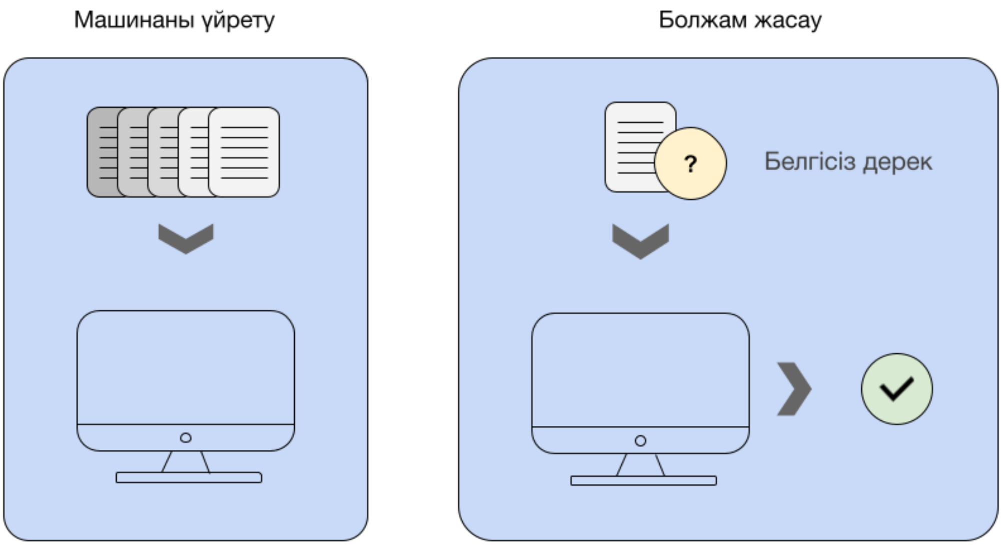
КІРІСПЕ

1. Кіріспе

Сұрақ - 5, есеп - 0

1.1. Машинаны үйрету дегеніміз не?

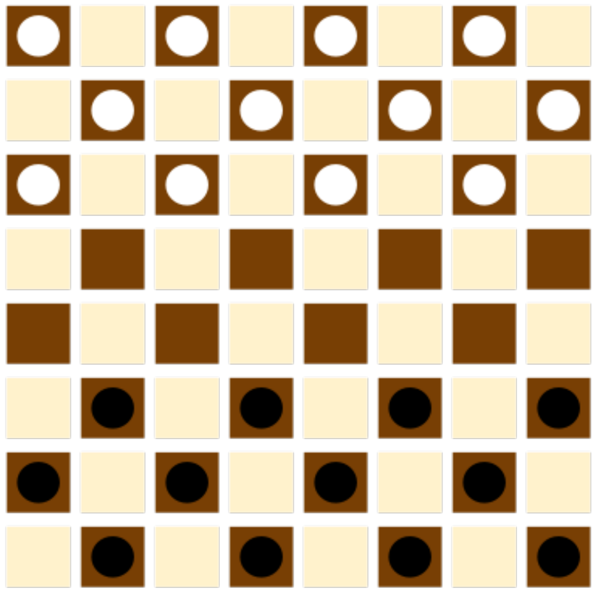
Машинаны үйрету — 20-ғасырда пайда болған ғылым саласы. Машинаны үйретудің ең басты мәні — деректерді қолдана отырып компьютерге білім беру. Ал ол білімнің басты мақсаты — болашақта дұрыс болжам жасауға қол жеткізу. Яғни қолда бар үлкен деректерді пайдаланып, болашақта дұрыс болжам жасау (1.1-сурет).



1.1-сурет. Машинаны үйрету және болжам жасау

"Машинаны үйрету" дегенге ғалымдар әрқандай түсінік берген. Артур Самуэльдің айтуы бойынша (1959 ж.) машинаны үйрету — компьютердің қандай да бір білімді үйреніп, түсіне алу қабілеті. 1959 жылы алғашқы машинаны үйрету бағдарламаларының бірі "дойбы" ойыны еді. Артур Самуэльдің құрған дойбы ойыны кәдімгі адам сияқты ойнау қабілетіне ие болған. Артур Самуэльдің өзі дойбы ойынын жақсы ойнамағанымен, компьютерді 10 000-дай дойбы ойынымен ойнатқызып, оған қай жүріс жеңіске, қай жүріс жеңіліске алып келетінін үйреткен. 10 000-дай ойыннан кейін компьютер дойбы ойынын Артур Самуэльдің өзінен жақсы ойнау қабілетіне ие болған.

Мұндағы бір ерекшелік — адам 10 000-дай ойын ойнауға көп уақыт жұмсайтындығы және ол үлкен шыдамдылық қажет етендігі. Ал компьютер аз ғана уақыт аралығында шыдамдылықты қажет етпестен тез үйрене алады. Бұл тәжірибенің соңында дойбы ойынын нашар ойнайтын Артур Самуэль өзінен әлде қайда жақсы ойнайтын бағдарлама жазып шығарды (1.2-сурет).



1.2-сурет. Дойбы ойнының тақтасы

Кейінірек, 1998 жылы Том Мигель машинаны үйретуге мынадай анықтама берді.

Машинаны үйрету:

Компьютерлік бағдарламаның ***E*** (experience) тәжірибесінен үйреніп, ***T*** (task) тапсырмасында ***P*** (performance) нәтижесін жақсартумен анықталады.

Артур Самуэльдің дойбы ойынына жасаған бағдарламасындағы ***E*** тәжірибе — 10 000-дай дойбы ойынын ойнауы, ***T*** тарсырма — дойбы ойынының өзі, ал ***P*** нәтиже — келесі ойында ұту ықтималдығы. Машинаны үйретуді жақсы түсіну үшін келесі сұрақтарға жауап беріңіздер.

1.1-сұрақ.

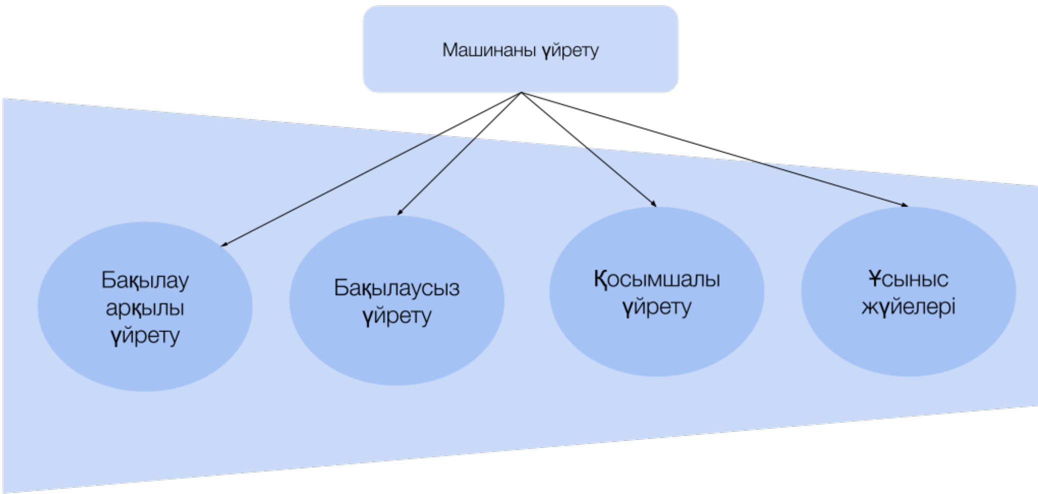
Суретте мысық немесе ит бейнеленгенін анықтайтын бағдарламада ***E***-тәжірибе, ***T***-тапсырма және ***P***-нәтиже нені білдіреді?

1. ***E*** - тәжірибе — суреттегі мысық пен иттің айырмашылығын түсініп, білу процессі.
2. ***T*** - тапсырма — суреттегі мысық пен иттің айырмашылығын түсініп, білу процессі.
3. ***P*** - нәтиже — суреттегі мысық пен иттің айырмашылығын түсініп, білу процессі.
4. ***E*** - тәжірибе — 100 суреттің 90 суретінде не мысық, не ит бейнеленгенін анықтау.
5. ***T*** - тапсырма — 100 суреттің 90 суретінде не мысық не, ит бейнеленгенін анықтау.
6. ***P*** - нәтиже — 100 суреттің 90 суретінде не мысық не, ит бейнеленгенін анықтау.
7. ***E*** - тәжірибе — суреттегі не мысық не, ит екенін анықтау.
8. ***T*** - тапсырма — суреттегі не мысық не, ит екенін анықтау.
9. ***P*** - нәтиже — суреттегі не мысық не ит, екенін анықтау.

Дұрыс жауабын кітап соңынан көріп анықтай аласыздар.

1.2. Машинаны үйретудің түрлері.

Машинаны үйретудің ең негізгі екі түрі бар, олар: бақылау арқылы үйрету (supervised learning) және бақылаусыз үйрету (unsupervised learning). Машинаны үйретудің бұлардан басқа қосымшалы үйрету (reinforcement learning) және ұсыныс жүйелері (recommended systems) дейтін түрлері де бар (1.3-сурет).



1.3-сурет. Машинаны үйретудің түрлері

Бақылаусыз үйретуге қарағанда, бақылау арқылы үйрету алгоритмдері әрі көп әрі кең тараған. Бақылау арқылы үйрету мен бақылаусыз үйретудің анықтамалары төменде қысқаша түрде берілді.

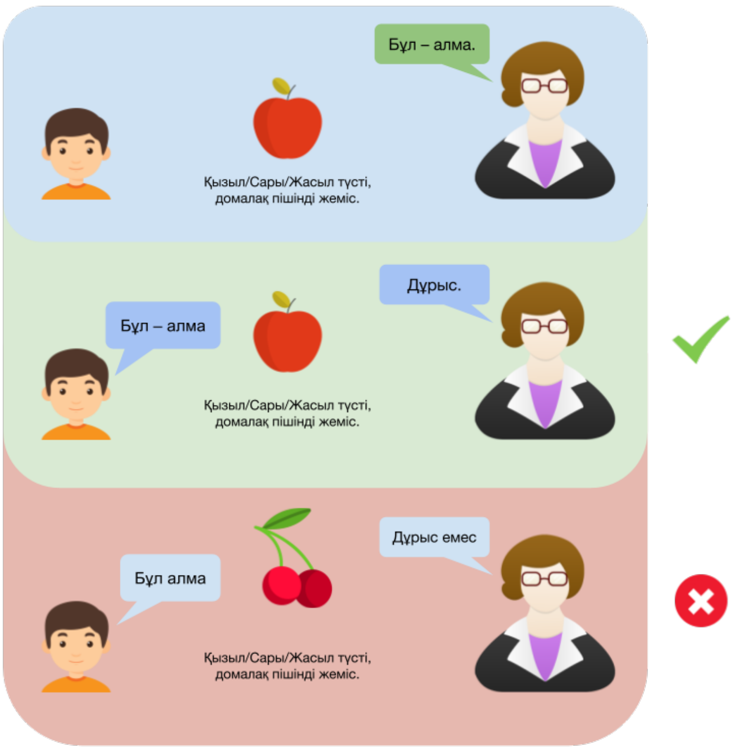
Бақылау арқылы және бақылаусыз үйрету:

Бақылау арқылы үйретуде біз компьютерді үйретеміз, ал бақылаусыз үйретуде компьютер біздің араласуымызсыз өздігінен үйренеді.

Бақылау арқылы үйрету мен бақылаусыз үйрету түрлеріне келесі тақырыпта толықтай түсінік беріледі.

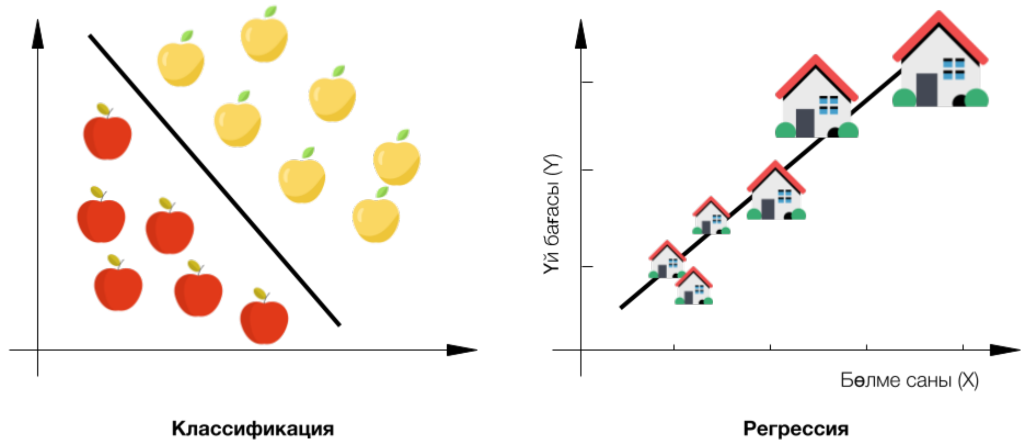
1.3. Бақылау арқылы үйрету

Бақылау арқылы үйрету дегеніміз — X айнымалысымен және Y нәтижесімен берілген f функциясын үйрену. Мұндағы $f(X) = y$. f функциясын анықтағаннан кейін берілген кез келген X айнымалысы үшін Y нәтижесін болжай аламыз. Бақылау арқылы үйретуде кез келген X айнымалысы үшін Y нәтижесі бар болғандықтан, қателікпен жасаған Y' болжамына ескерту жасай аламыз. Үйрету процессі осылайша жүзеге асады. Үйрету деңгейі жақсы нәтижеге жеткенде үйрету процессі тоқтатылады. Бақылау арқылы үйретуді жеңілдетілген тілмен түсіндіру үшін кішкентай бала алманы алғаш қалай танығанын мысалға келтірейік. Бала алманың не екенін білмей тұрғанда мұғалім немесе ата-ана алманы балаға көрсету арқылы таныстырады. Яғни бала алманың қызыл не сары немесе жасыл, домалақ пішінді жеміс екенін танып үйренеді. Келесі жолы бала алманы көргенде, оны түсіне, пішініне қарап алма екенін айтады. Бірақ бала жаңылысқан жағдайда, мұғалім немесе ата-ананың бақылауы арқылы баланың жіберген қателігі түзетіледі. Яғни бұл бақылау арқылы үйрету түрі болып табылады (1.4-сурет).



1.4-сурет. Бақылау арқылы үйрету

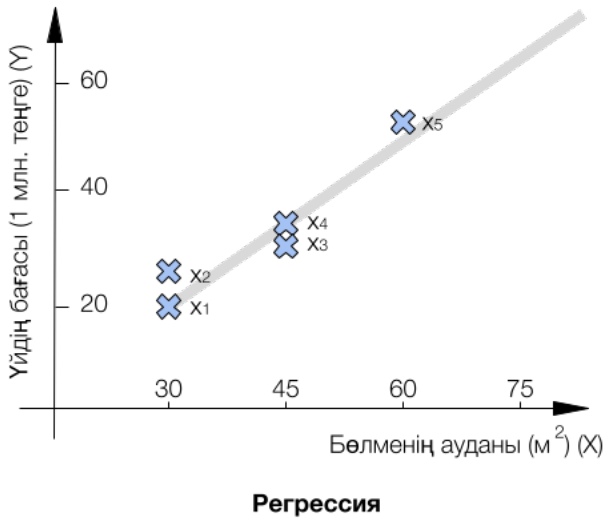
Бақылау арқылы үйретудің екі түрі бар, олар: Классификация және Регрессия. Классифицияда берілген объектінің қандай да бір класқа тиесілі екенін анықтаймыз. Ал регрессияда нәтиже — қандай да бір мәнді білдіретін сан. Мысалы, классификация есептері мынадай болуы мүмкін: алманың сары немесе қызыл екенін анықтау. Яғни бізде 2 класс бар, бірі — сары алма, екіншісі — қызыл алма. Ал регрессияның мысалы мынадай болуы мүмкін: үш бөлмелі пәтердің бағасын анықтау. Яғни мұнда нәтиже — класс емес, сан, яғни үйдің бағасын көрсететін сан (1.5-сурет).



1.5-сурет. Классификация және регрессия мысалдары

1.3.1. Бақылау арқылы үйретуде болжам жасау (Регрессиялық есеп)

Берілген деректерді қолдана отырып, бақылау арқылы үйретуде қалай болжам жасау керектігін анықтайық. Ол үшін төмендегі мысалды қарастырайық (1.6-сурет).



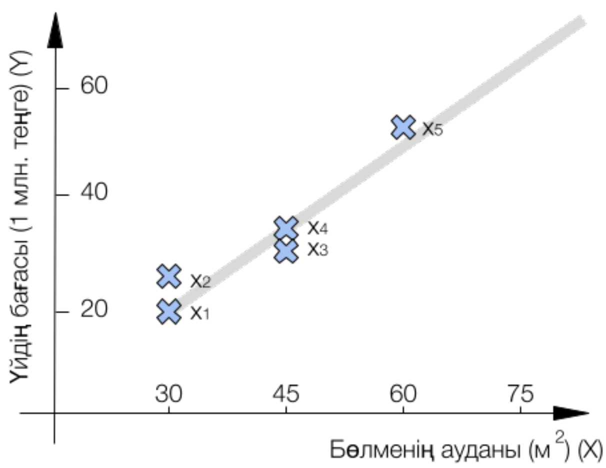
1.6-сурет. Регрессияның мысалы

Бізге 5 үй берілсін (x_1 , x_2 , x_3 , x_4 және x_5). Әр үй үшін оның ауданын (m^2)(X) әрі бағасын (Y) білеміз (1.1-кесте).

дерек	үйдің ауданы (m^2)	үйдің бағасы (млн. теңге)
x_1	15	20
x_2	15	25
x_3	45	30
x_4	45	35
x_5	60	50

1.1-кесте. Регрессиялық есепке берілген деректер

Осы деректерді қолдана отырып, ауданы 75 (m^2)-қа тең үйдің бағасы неше теңге болады деген сұрақ туындаса, онда графикке қарап 65 млн. теңге деп жауап бере аламыз. Себебі үйлердің бағасы виртуалды түзу бойымен өсіп бара жатқанын көреміз (1.7-сурет).



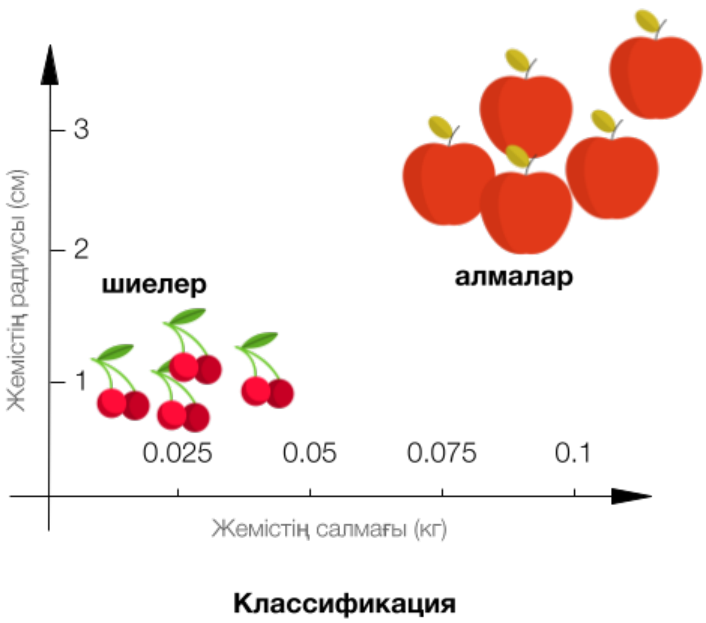
Регрессия

1.7-сурет:Берілген деректер бойынша үйдің бағасын болжау

Бұл есеп регрессиялық есеп болып табылады. Себебі, берілген үйдің ауданы (X) бойынша үйдің бағасын Y анықтап жатырмыз. Мұнда үйдің бағасы — сан, яғни класс/категория емес (классификациялық есептегідей).

1.3.2. Бақылау арқылы үйретуде болжам жасау (Классификациялық есеп)

Классификация есебінде болжам жасауды төмендегі мысалмен түсіндірейік (1.8-сурет).

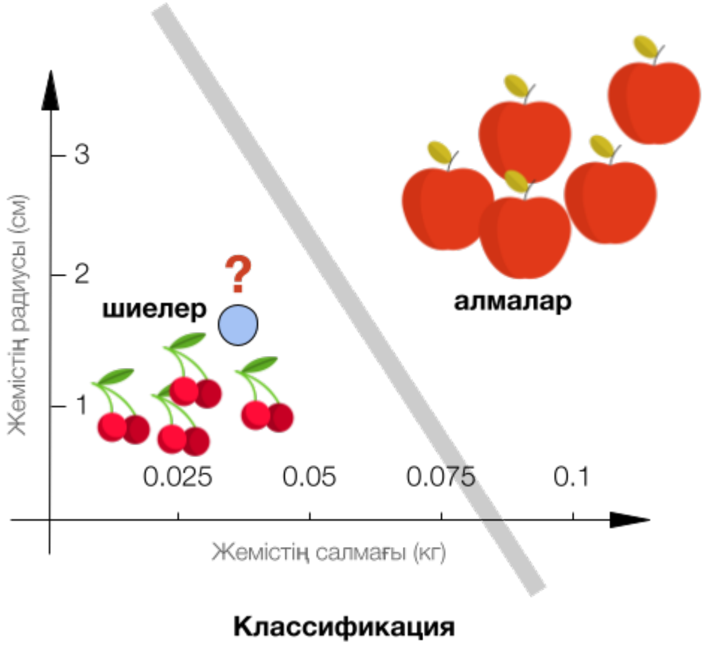


1.8-сурет. Берілген деректер бойынша жемістің түрін анықтау.

Бізде екі жеміс бар делік, оның бірі — алма, екіншісі — шие. Олар бір-бірінен көлемімен, салмағымен, түсімен және дәмімен ажыратылады. Бұл жолы 2 қасиетін таңдап алайық, олар жемістің салмағы және радиусы болсын (1.8-сурет). Осы екі қасиетін пайдалана отырып, берілген жеміс не алма, не шие екенін анықтаймыз. Егер бізге 0.0375 кг салмағымен және 1.5 см радиусымен өлшенетін жеміс берілсе, онда оның шие болу ықтималдығы жоғары (1.8-сурет).

Себебі, шие жемісі көбінше осындай қасиетке ие. Әрі графикте (1.9-сурет) көрсетілгендей, бұл екі жемісті түзу сызықпен бөле аламыз. Яғни бұл сызықтың жоғарғы жағындағы жемістер — алма, ал төменгі жағындағы жатқан жемістер — шие. Осылайша, бізге берілген деректерді қолдана отырып, берілген жемісті шие немесе алма классына жатқыза аламыз. Бұл — бақылаумен үйретудегі классификация есебінің мысалы.

Бақылау арқылы үйретуді тереңірек түсіну үшін мына сұрақтарға жауап беріңіздер.



1.9-сурет. Берілген деректер бойынша бұл жеміс алма немесе шие екенін болжау

1.2-сұрақ.

Берілген автокөліктің түсін анықтау (сұр, қара, ақ) қандай есепке жатады?

1. Классификациялық
2. Регрессиялық

Дұрыс жауабын кітап соңынан көре аласыздар

1.3-сұрақ.

Адамның түріне қарап жасын анықтау қандай есеп болып табылады?

1. Классификациялық
2. Регрессиялық

Дұрыс жауабын кітап соңынан көре аласыздар

1.4-сұрақ.

Адам қай мемлекеттің (Қазақстан, Малайзия т.с.с.) азаматы екенін анықтау қай есепке жатады?

1. Классификациялық
2. Регрессиялық

Дұрыс жауабын кітап соңынан көре аласыздар

1.5-сұрақ.

Ұялы телефонның неше түрлі қасиеттеріне қарай (брэнд, жад т.с.с.) бағасын анықтау қай есепке жатады?

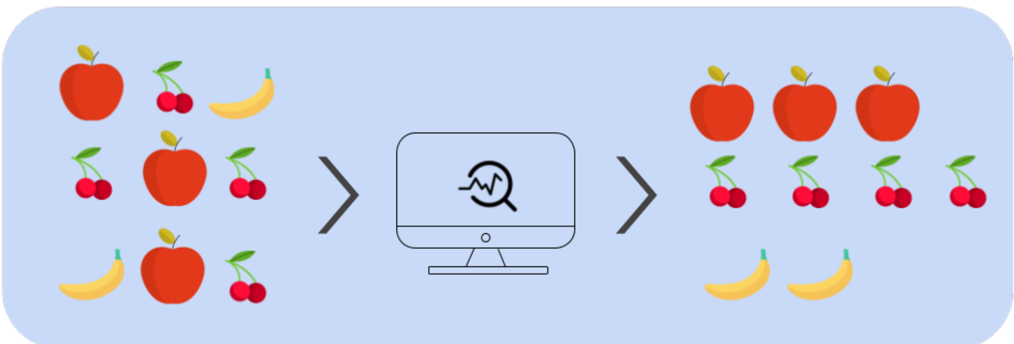
1. Классификациялық
2. Регрессиялық

Дұрыс жауабын кітап соңынан көре аласыздар

1.4. Бақылаусыз үйрету

Бақылаусыз үйретуде тек X дерегі ғана беріледі. Яғни бақылау арқылы үйретудегідей Y нәтижесі жоқ. Бақылаусыз үйретудің мақсаты — берілген X дерегі бойынша сол деректің ішіндегі қасиеттерді анықтау. Бақылаусыз үйрету деп аталу себебі — мұнда бақылау арқылы үйрету сияқты Y нәтижесінің жоқтығында. Яғни болжам жасау барысында қандай да бір қателік кетсе, онда біз ол қателікті түзете алмаймыз.

Бақылаусыз үйретуді түсіну үшін төмендегі мысалды қарастырайық. Сізге бір себет өзіңіз білмейтін жемістер берілсін дейік. Бұл жемістерді алғаш рет көріп тұрған сіз бір-бірінен ажыратып, оларды топтарға бөлуіңіз керек (1.10-сурет). Яғни мұнда сіз жемістің түріне, пішініне, түсіне және басқа қасиеттеріне қарап, оларды бір-бірінен ажырата аласыз. Бақылау арқылы үйретудің бақылаусыз үйретуден төменгідей айырмашылықтары бар (1.2-кесте).



1.10-сурет. Бақылаусыз үйретудің мысалы

Бақылау арқылы үйрету	Бақылаусыз үйрету
Әр X айнымалысы үшін Y нәтижесі бар.	Әр X айнымалысы үшін Y нәтижесі жоқ.
Болжам жасау барысында қателік кетсе, қателікті Y нәтижесі арқылы түзете аламыз.	Болжам жасау барысында қателік кетсе, оны түзете алмаймыз. Себебі X айнымалысы үшін Y нәтижесі жоқ.
Бақылау арқылы үйрету X айнымалысының қай класқа жататынын (классификация) немесе берілген X айнымалысының мәнін анықтайды (регрессия).	Бақылаусыз үйрету берілген X дерегін белгілі бір қасиеттеріне қарап топтастырады. Бұл кластеризациялық есеп деп аталады.

1.2-кесте: Бақылау арқылы және бақылаусыз үйретудің айырмашылықтары

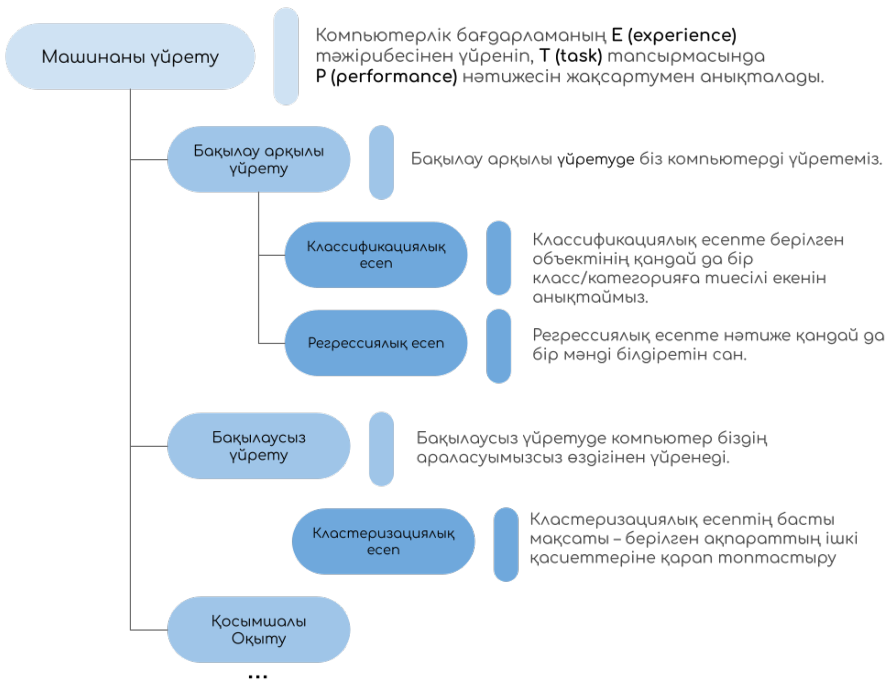
1.4.1. Кластеризация

Бақылаусыз үйретуде кластеризациялық есеп көптеп кездеседі. Кластеризациялық есептің басты мақсаты — берілген деректің ішкі қасиеттеріне қарап топтастыру. Яғни бақылау арқылы үйретуде берілген объектінің атын білсек (мысалы, домалақ, қызыл/сары/жасыл түсті жеміс — объект ***X***, ал аты — алма ***Y***) және сол аты бойынша оларды топтастыру арқылы болжам жасайтын болсақ, бақылаусыз үйретуде ол объектінің аты бізде жоқ. Яғни топтастыруды ішкі қасиеттеріне қарап жасаймыз. Жемістерді топтастыру мысалында пішініне, түсіне қарап олар бір-бірінен ажыратылады.

Қорытынды

Төмендегі суретте (1.11-сурет) осы бөлімнің ең маңызды бөліктері көрсетілген. Ол бөліктер мына сұрақтарға жауап береді:

- Машинаны үйрету дегеніміз не?
- Машинаны үйретудің қандай түрлері бар?
- Бақылау арқылы үйрету дегеніміз не?
- Бақылаусыз үйрету дегеніміз не?
- Бақылау арқылы үйретудің қандай түрлері болады?
- Бақылаусыз үйретудің қандай түрлері болады?
- Классификациялық есеп дегеніміз не?
- Регрессиялық есеп дегеніміз не?
- Кластеризациялық есеп дегеніміз не?

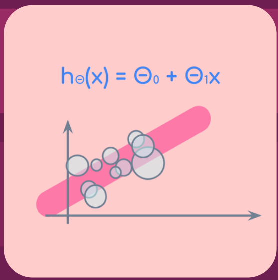


1.11-сурет. Машинаны үйрету және оның түрлері

II

бөлім

СЫЗЫҚТЫҚ РЕГРЕССИЯ

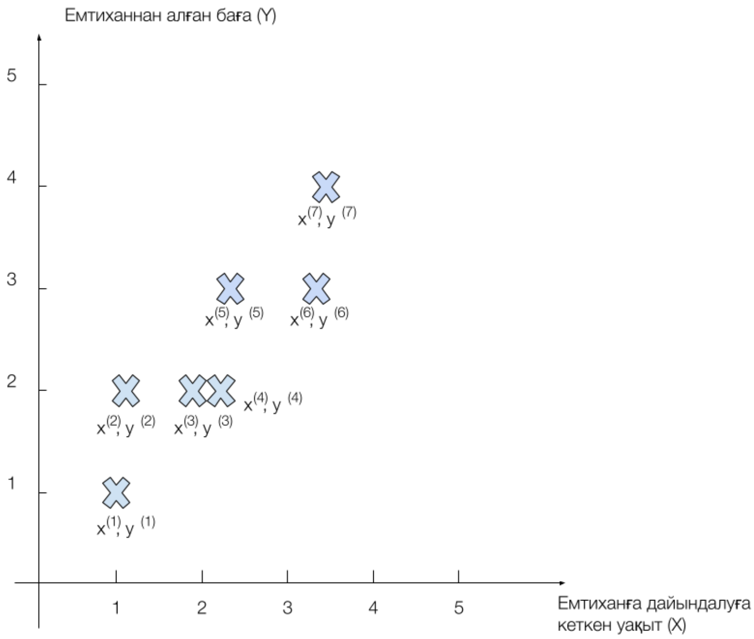


2. Сызықтық регрессия

Сұрақ — 6, Есеп — 3

2.1. Сызықтық регрессия

Алдыңғы бөлімде регрессиялық есеп бақылау арқылы үйретудің бір түрі екенін айтқан едік. Ал регрессиялық есептің кең тараған шешімдерінің бірі — сызықтық регрессия. Сызықтық регрессияны түсіндіру үшін төмендегі есепті қарастырайық (2.1-сурет).



2.1-сурет. Машинаны үйрету және болжам жасау

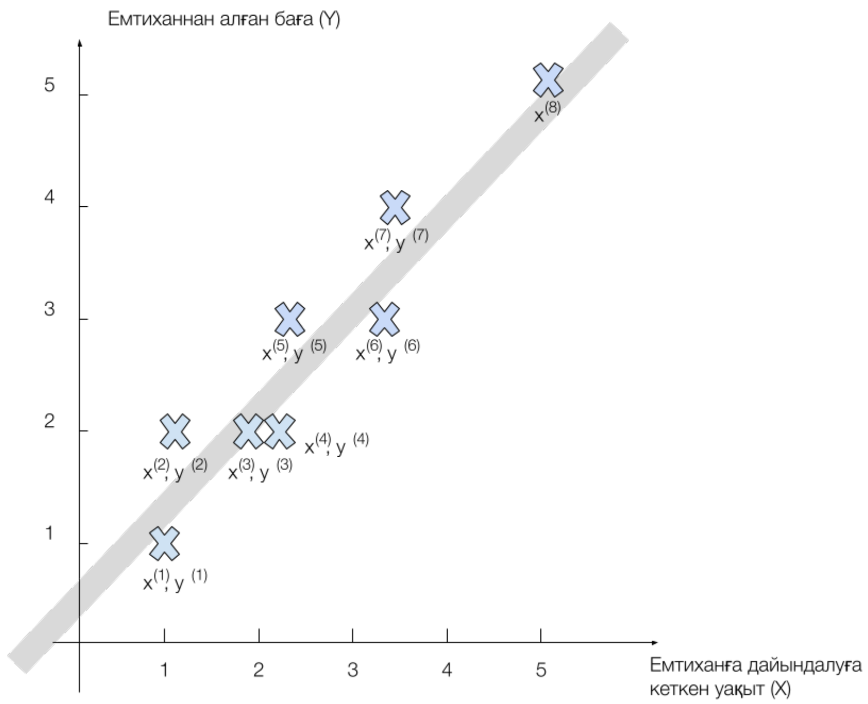
Бізде 7 оқушы бар дейік. Әр оқушының емтиханнан алған бағасы және сол емтиханға дайындалуға кеткен уақыт бар екені белгілі. X осінде емтиханға дайындалуға кеткен уақыт, ал Y осінде соған сәйкес емтиханнан алған бағасы болсын. Әр оқушы $(x^{(i)}, y^{(i)})$ сандарымен анықталады. Яғни $(x^{(i)}, y^{(i)})$ — біздегі i-інші оқушы, мұндағы $x^{(i)}$ — i-інші оқушының емтиханға дайындалуға кеткен уақыты, ал $y^{(i)}$ — i-інші оқушының емтиханнан алған бағасы.

Бізде барлығы m оқушы болсын делік. Біздің жағдайда $m = 7$. Яғни i мәні 1-ден 7 (m)-ге дейін анықталады. $x^{(i)}$, $y^{(i)}$ мәндерін толығырақ түсіну үшін төмендегі кестеге көз жібертейік (2.1-кесте).

i	x	y
$i = 1$	$x^{(1)} = 1$	$y^{(1)} = 1$
$i = 2$	$x^{(2)} = 1.1$	$y^{(2)} = 2$
$i = 3$	$x^{(3)} = 1.9$	$y^{(3)} = 2$
$i = 4$	$x^{(4)} = 2.2$	$y^{(4)} = 2$
$i = 5$	$x^{(5)} = 2.3$	$y^{(5)} = 3$
$i = 6$	$x^{(6)} = 3.3$	$y^{(6)} = 3$
$i = 7$	$x^{(7)} = 3.4$	$y^{(7)} = 4$

2.1-кесте. Сызықтық регрессияға берілген деректер жиынтығы

Енді бізге жаңа $x^{(8)}$ мәні берілсін (2.2-сурет). $x^{(8)}$ мәні 5-ке тең болсын. Біздегі қолда бар деректерді пайдалана отырып, $x^{(8)}$ мәні үшін $y^{(8)}$ мәні 5-ке тең болуы мүмкін деген болжам жасай аламыз. Себебі $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$ мәндері виртуалды сызық бойымен орналасқан сияқты болады (2.2-сурет). Біздегі дайындалуға кеткен уақыт пен баға бір-біріне пропорционал. Яғни арасындағы қатынас түзу сызықпен анықталады. Сол себепті бұл мысал сызықтық регрессияға жатады. Жалпылама айтқанда:



2.2-сурет: Дайындалуға кеткен уақыт пен бағаның арасындағы сызықтық қатынас

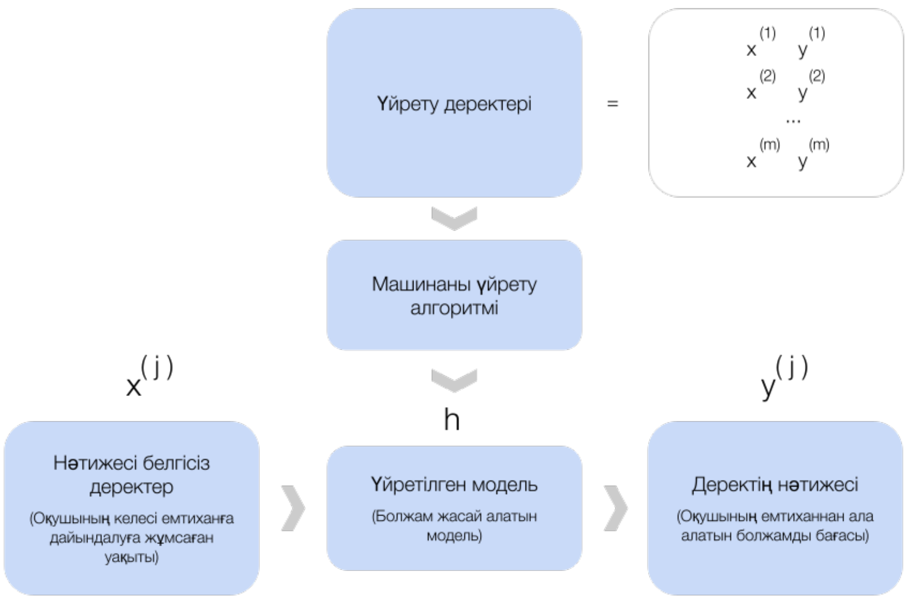
Сызықтық регрессия:

X айнымалы мен Y нәтижесі арасындағы сызықтық қатынас.

2.1.1. Машинаны үйрету моделінің жалпы жағдайы

Жалпы жағдайда бізге берілген $x^{(i)}$ — i-інші деректің қасиеті (мысалы, оқушыны сипаттайтын қасиет — сол емтиханға дайындалуға кеткен уақыт), ал $y^{(i)}$ — ол i-інші деректің нәтижесі (мысалы, оқушының емтиханнан алған бағасы). m — үйретуге қажетті деректердің саны. Осы мәліметтерді пайдалана отырып, машинаны үйрету модельдерінің үйрету сызбасын қарастырайық (2.3-сурет). Үйретуге берілген $(x^{(i)}, y^{(i)})|i = 1...m$ дерегін пайдалана отырып, біз h (hypothesis — гипотеза) машинаны үйрету моделін құрамыз. Бұл модельдің басты мақсаты - болашақ $x^{(j)}$ дерегі үшін соған сәйкес $y^{(j)}$ нәтижесін табу (болжау). Математика тілімен айтқанда:

h — X айнымалысын Y нәтижесіне айналдыратын функция: $y = h(x)$



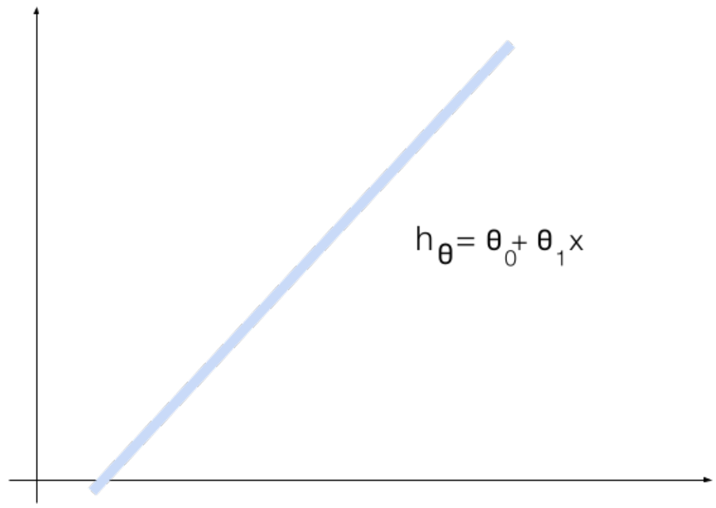
2.3-сурет. Машинаны үйрету модельдерінің жалпы сызбасы

Сызықтық регрессияда h функциясы былайша анықталады:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Мұндағы $\theta = (\theta_0, \theta_1)$ яғни θ — θ_0 және θ_1 параметрлерімен анықталады.

$h_\theta(x) = \theta_0 + \theta_1 x$ теңдеуі — сызық теңдеуі (2.4-сурет) және де болашақ $x^{(j)}$ мәні беріліп, соған сәйкес $y^{(j)}$ мәнін табу үшін, бізге θ_0 және θ_1 мәндерін табу жеткілікті болады. Себебі, $y^{(j)} = h_\theta(x^{(j)}) = \theta_0 + \theta_1 x^{(j)}$.



2.4-сурет. Сызықтық регрессияның теңдеуі

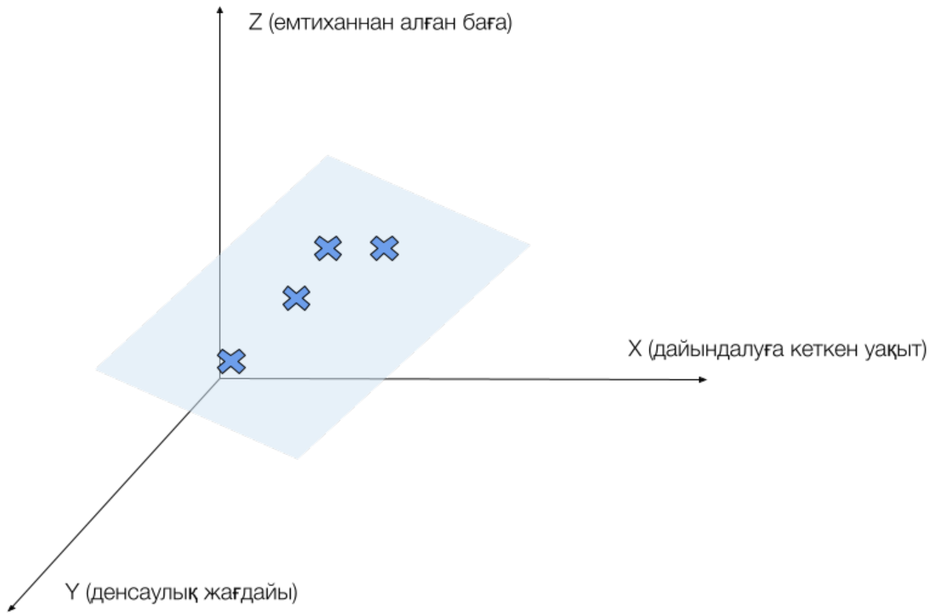
Берілген деректің бір ғана қасиеті бар болғандықтан (оқушының көрсеткен нәтижесі тек дайындалу уақытысымен анықталғандықтан), бұл есеп "бір ғана қасиетпен анықталған сызықтық регрессия" деп аталады. Ал басқа есептерде берілген дерек бірнеше қасиеттермен анықталуы мүмкін. Мысалы: оқушының емтиханнан алған бағасы 1) дайындалуға кеткен уақыт, 2) денсаулығының жағдайы т.с.с. параметрлерге байланысты болуы мүмкін. "Бір айнымалымен анықталатын сызықтық регрессия" екі өлшемді кеңістікте бейнеленетін болса (2.4-сурет), онда k қасиеті бар сызықтық регрессия $k+1$ өлшемді кеңістікте анықталады. Мысалы, $k = 2$ болса: 1) дайындалуға кеткен уақыт, 2) денсаулық жағдайы, онда нәтиже (емтихан бағасы) 3-тік кеңістікте анықталады (2.5-сурет).

Осы және алдыңғы тақырыптарды жете түсіну үшін, төмендегі сұрақтарға жауап беріңіздер:

2.1-сұрақ.

Сызықтық регрессия X және Y айнымалыларының арасындағы қандай қатынасты көрсетеді?

1. квадраттық
2. кубтық
3. сызықтық



2.5-сурет:Сызықтық регрессия үштік кеңістікте

2.2-сұрақ.

Сызықтық регрессиядағы $h_\theta(x)$ функциясы қалай анықталады?

1. $h_\theta(x) = \theta_2 x^2 + \theta_1 x + \theta_0$
2. $h_\theta(x) = \theta_1 x + \theta_0$
3. $h_\theta(x) = \theta_3 x^3 + \theta_2 x^2 + \theta_1 x + \theta_0$

2.3-сұрақ.

Бір өлшемді түзулік регрессия қайсысы?

1. X айнымалысы адамның салмағы және бойымен анықталады
2. X айнымалысы адамның салмағы, бойы және жасымен анықталады
3. X айнымалысы адамның салмағымен анықталады

2.1-есеп.

Бір айнымалы сызықтық регрессия үшін болжам функциясын анықтайтын есепті шығарыңыз.

Бір айнымалы сызықтық регрессия үшін болжам функциясы

Сізге x, θ_0, θ_1 айнымалылары берілген. Осы үш айнымалы бойынша $\theta_1 x + \theta_0$ мәнін есептеңіз.

Берілген деректер

Сізге input_1.txt файлының әр жолында x, θ_0, θ_1 айнымалылары берілген. Мысалы:

input_1.txt

```
1,2,3
2,3,4
4,5,6
...
```

Сыртқа шығатын деректер

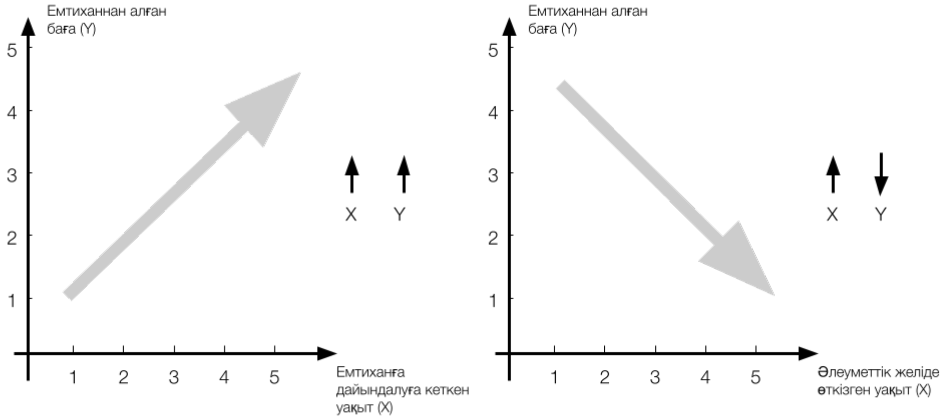
input_1.txt файлының әр жолы үшін $\theta_1 x + \theta_0$ мәнін есептеп, output_1.txt файлының әр жолына сол мәнді жазып шығыңыз. Мысалы:

output_1.txt

```
5
10
29
...
```

2.1.2. Сызықтық регрессияның түрлері

Сызықтық регрессияның мынадай екі түрі болады: позитивті қатынастағы сызықтық регрессия және негативті қатынастағы сызықтық регрессия. Алдынғы қарастырған "емтиханға дайындалуға кеткен уақыт" мысалы позитивті қатынастағы сызықтық регрессия болып табылады. Себебі, тәуелсіз айнымалы X (дайындалуға кеткен уақыт) артқан сайын тәуелді айнымалы Y да (емтиханнан алған баға) артады. Ал тәуелсіз айнымалы X өскен сайын тәуелді айнымалы Y кемісе, онда мұндай қатынасты негативті қатынас деп атаймыз. Мысалы, әлеуметтік желіде көп отырған сайын емтиханнан алу бағамыз кеми беруі мүмкін (2.6-сурет). Біз көрсеткен позитивті және негативті қатынастағы сызықтық регрессия мысалдарының графигі төмендегі суретте (2.6-сурет) көрсетілген. Келесі тарауда сызықтық регрессияға қолданылатын "баға функциясымен" танысатын боламыз.



2.6-сурет. Позитивті және негативті сызықтық регрессия

Позитивті және негативті қатынасты сызықтық регрессияны дұрыс түсіну үшін, келесі сұрақтарға жауап беріп көріңіздер.

2.4-сұрақ.

Тамақ көп жеген сайын салмақ артады. Бұл тұжырым қай қатынастағы сызықтық регрессияға жатады?

1. позитивті қатынас
2. негативті қатынас

2.5-сұрақ.

Көп жүрген сайын адамның энергиясы азаяды. Бұл тұжырым қай қатынасқа жатады?

1. позитивті қатынас
2. негативті қатынас

2.1.3. Баға функциясы

"Емтиханға дайындалуға кеткен уақыт пен емтиханнан алған баға" мысалын қайта қарастырайық. Біздегі бар деректер төмендегі кестеде берілген (2.2-кесте).

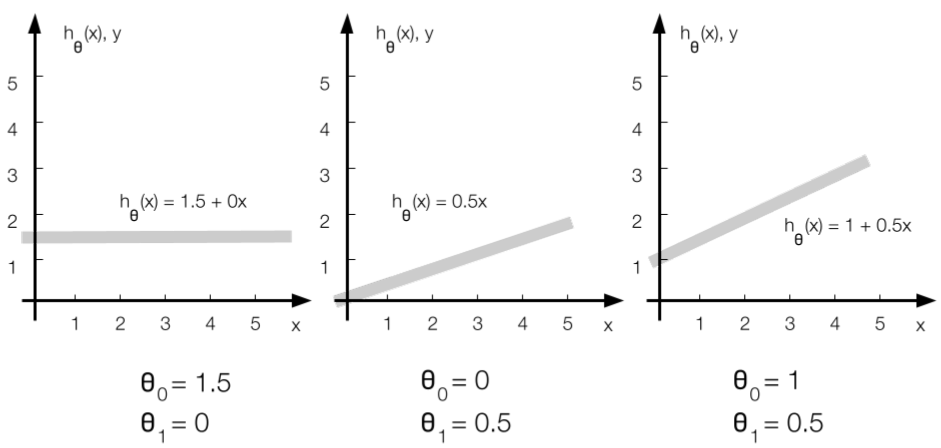
Бұл деректерді қолдана отырып, сызықтық регрессия арқылы болжам жасалатындықтан, біздің болжам жасайтын функциямыз мынадай болады:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Мұндағы θ_0 және θ_1 функциясының параметрлері және олар кез келген нақты санға тең болуы мүмкін ($\theta_0, \theta_1 \in R$). Енді θ_0 және θ_1 мәндерін қалай таба аламыз деген сұрақ туындайды. Ол мәндерді табу үшін мынадай жағдайларды қарастырайық (2.7-сурет). θ_0 және θ_1 әр түрлі мәнге ие болғанда, біз әр түрлі сызық графигін шығарып аламыз. Ол графикті шығарып алу үшін $h_\theta(x) = \theta_0 + \theta_1 x$ теңдеуіне θ_0 және θ_1 мәндерін қойып және неше түрлі X мәндері үшін ($X \in (-\infty, +\infty)$), $h_\theta(x)$ мәнін тауып, $(x, h_\theta(x))$ екі өлшемді график салу жеткілікті болады.

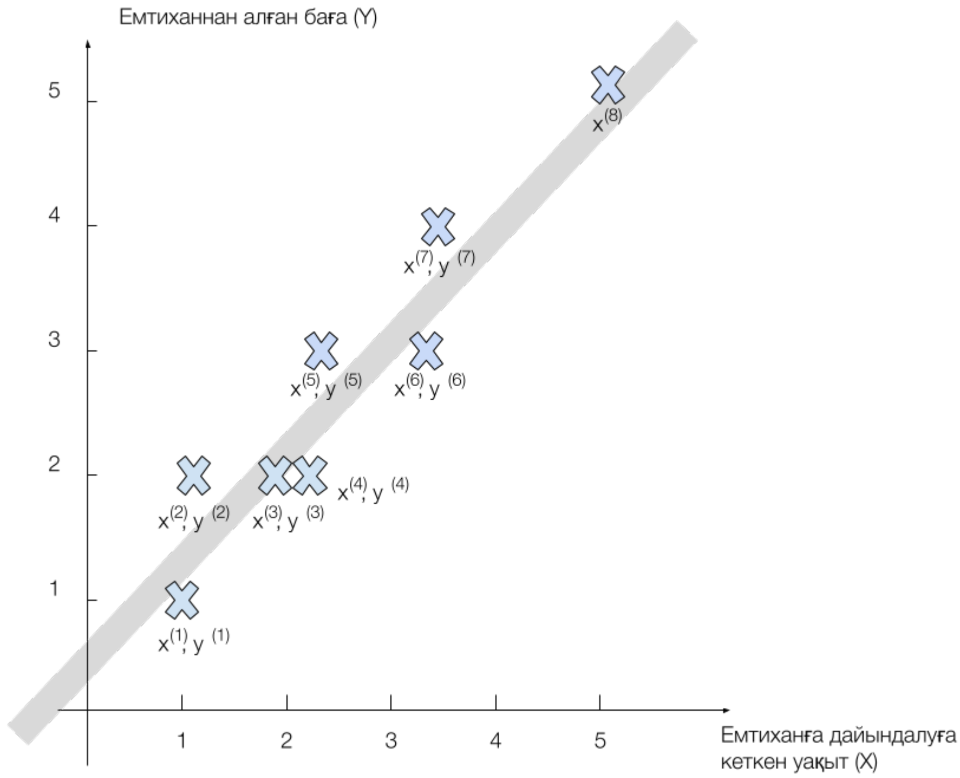
i	x	y
$i = 1$	$x^{(1)} = 1$	$y^{(1)} = 1$
$i = 2$	$x^{(2)} = 1.1$	$y^{(2)} = 2$
$i = 3$	$x^{(3)} = 1.9$	$y^{(3)} = 2$
$i = 4$	$x^{(4)} = 2.2$	$y^{(4)} = 2$
$i = 5$	$x^{(5)} = 2.3$	$y^{(5)} = 3$
$i = 6$	$x^{(6)} = 3.3$	$y^{(6)} = 3$
$i = 7$	$x^{(7)} = 3.4$	$y^{(7)} = 4$

2.2-кесте. Сызықтық регрессияға берілген деректер жиынтығы



2.7-сурет. Сызықтық регрессия параметрлері

Ал енді алдыңғы мысалымызда (2.2-кесте) θ_0 және θ_1 мәндерін табу керек делік. Яғни осы деректердің бойымен жүретін сызықтың θ_0 және θ_1 параметрлерін табу қажет (2.8-сурет).



2.8-сурет. Сызықтық регрессия параметрлері

Ол үшін бізге берілген $(x^{(i)}, y^{(i)})$ дерегіндегі $y^{(i)}$ мен $h_\theta(x^{(i)})$ мәндері бір-біріне өте жақын болатын θ_0 және θ_1 мәндерін таңдау қажет. Немесе кез келген i үшін $h_\theta(x^{(i)}) - y^{(i)}$ мәні өте кішкентай болу қажет (0-ге жақын болуы тиіс). Жалпы жағдайда θ_0 және θ_1-ді табу үшін мынадай тұжырым жасай аламыз:

$$minimize_{\theta_0,\theta_1} = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Мұндағы m-барлық деректер саны, $h_\theta(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$, i-інші — деректегі біздің болжам, ал $y^{(i)}$ i-інші — деректің нәтижесі. Яғни, барлық i 1-ден m-ге дейінгі $h_\theta(x^{(i)}) - y^{(i)}$ мәндерінің қосындысы өте кішкентай мән (0-ге жақын мән) болатындай θ_0 және θ_1 мәндерін таңдауымыз қажет. Мұндағы $\frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 = J(\theta_0, \theta_1)$ "баға функциясы" деп аталады. Яғни θ_0 және θ_1 мәндерін таңдау үшін баға функциясын ықшамдауымыз (minimize) қажет. Баға функциясының басқа да түрлері бар. Бірақ сызықтық регрессия үшін біз оның мынадай түрін қолданатын боламыз.

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Баға функциясын дұрыс түсіну үшін келесі сұраққа жауап беріп көріңіз.

2.6-сұрақ.

2.2-кестедегі деректер үшін баға функциясын есептеңіз. Мұндағы $\theta_0 = 1$ және $\theta_1 = 2$.

2.2-есеп.

Сызықтық регрессия үшін баға функциясын есептейтін есепті шығарыңыз.

Сызықтық регрессия үшін баға функциясы

Сізге x, y,m, θ_0, θ_1 айнымалылары берілген. Осы айнымалылар бойынша $\frac{1}{2m}\sum_{i=1}^{m}(\theta_1 x + \theta_0 - y)^2$ мәнін есептеңіз.

Берілген деректер

Сізге input_2.txt файлының бірінші жолында m, θ_0, θ_1 айнымалылары берілген. Келесі m жолда x, y айнымалылары берілген. Мысалы:

input_2.txt

```
3,1,2
1,2,2,3,4,5
```

Сыртқа шығатын деректер

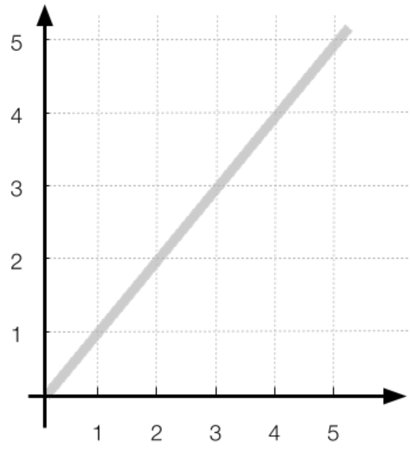
output_2.txt файлына $\frac{1}{2m}\sum_{i=1}^{m}(\theta_1 x + \theta_0 - y)^2$ мәнін есептеп жазыңыз. Жоғардағы мысал үшін:

output_2.txt

```
3.5
```

2.1.4. Баға функциясының экстрим нүктесі (Жеке жағдай)

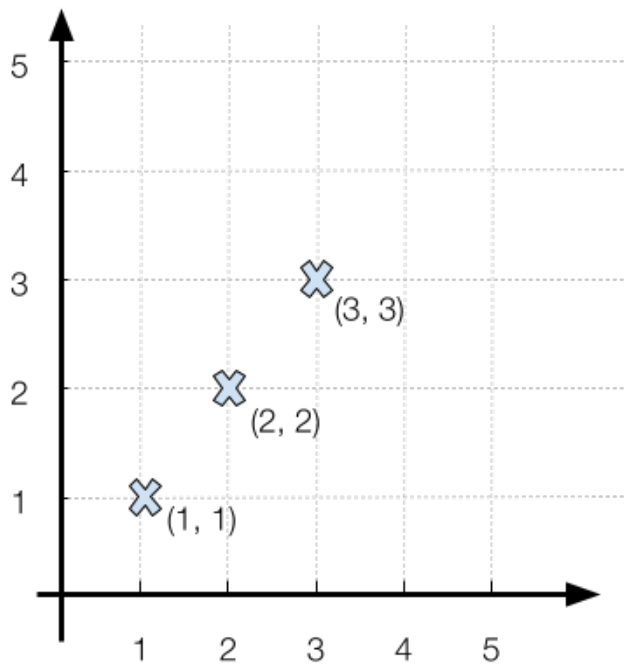
$J(\theta_0, \theta_1)$ баға функциясы екі параметрмен анықталады. Олар: θ_0 және θ_1. Бірақ, егер біз $\theta_0 = 0$ дейтін болсақ, онда $J(\theta_0, \theta_1) = \theta_1 x$ мәніне ие болып, тек бір айнымалымен, яғни θ_1 ғана анықталады ($J(\theta_0, \theta_1) = \theta_1 x$). $J(\theta_0, \theta_1) = \theta_1 x$ функциясының графигі төмендегідей болуы мүмкін (2.9-сурет).



2.9-сурет. Координата осьтерінің басы арқылы өтетін түзу.

Бұл суретте $\theta_1 = 1$ жағдайы кескінделген. Бірақ, жалпы жағдайда $h_\theta(x)$ сызығы координата осьтерінің басы арқылы өтеді.

Енді келесі суретте (2.10-сурет) бейнеленген деректерді қарастырайық. Осы деректерді қанағаттандыратын бірнеше θ_1 мәнін тексеріп көрейік:



2.10-сурет. Үйретуге берілген деректер

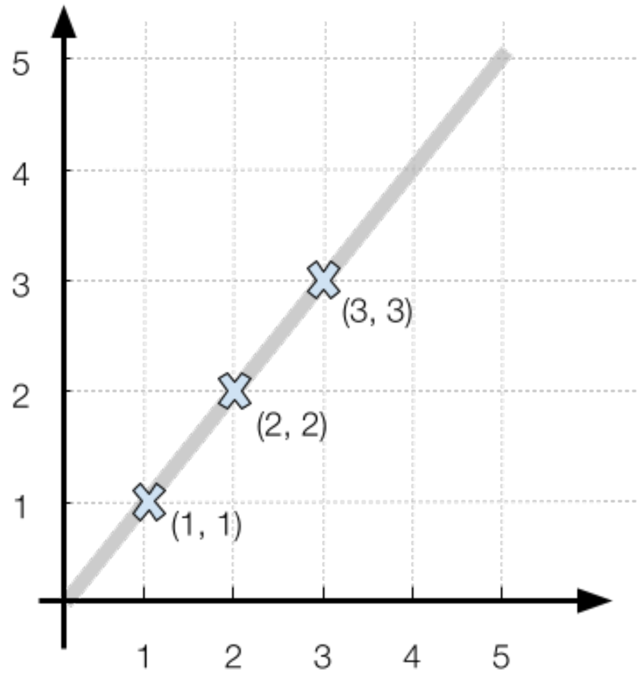
- Егер $\theta_1 = 1$ болса, онда біз мынадай график шығарып аламыз: $h_\theta(x) = x$. Ал $J(\theta_1)$ мәнін есептейтін болсақ, онда $J(1) = \frac{1}{6}((1-1)^2 + (2-2)^2 + (3-3)^2) = 0$ болады (2.11-сурет).

- Егер $\theta_1 = 0.5$ болса, онда біз мынадай график шығарып аламыз: $h_\theta(x) = 0.5x$. Ал $J(\theta_1)$ мәнін есептейтін болсақ, онда $J(0.5) = \frac{1}{6}((1-0,5)^2 + (2-0.5)^2 + (3-0.5)^2) = 0.58$ болады (2.12-сурет).

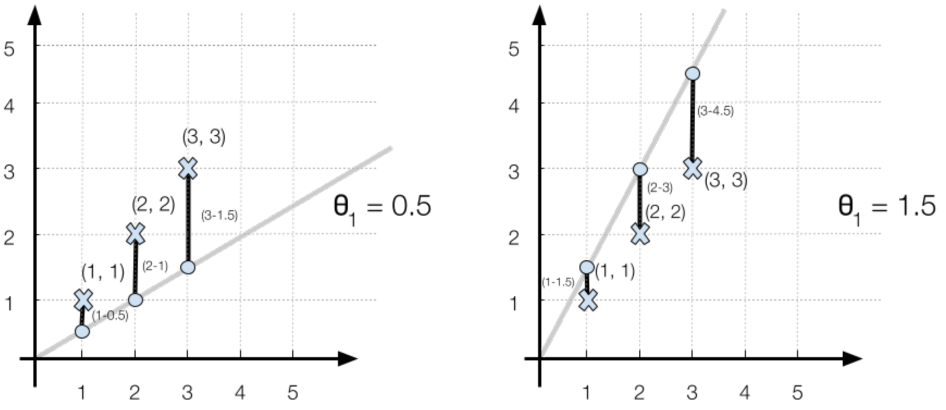
- Егер $\theta_1 = -0.5$ болса, онда біз мынадай график шығарып аламыз: $h_\theta(x) = -0.5x$. Ал $J(\theta_1)$ мәнін есептейтін болсақ, онда $J(-0.5) = \frac{1}{6}((1+0.5)^2 + (2+0.5)^2 + (3+0.5)^2) = 5.25$ болады (2.13-сурет).

- Егер $\theta_1 = 1.5$ болса, онда біз мынадай график шығарып аламыз ($h_\theta(x) = 1.5x$). Ал $J(\theta_1)$ мәнін есептейтін болсақ, онда $J(\theta_1)$ мәнін есептейтін болсақ, онда $J(1.5) = \frac{1}{6}((1-1.5)^2 + (2-1.5)^2 + (3-1.5)^2) = 0.58$ (2.12-сурет) болады.

- Егер $\theta_1 = 2.5$ болса, онда біз мынадай график шығарып аламыз ($h_\theta(x) = 2.5x$). Ал $J(\theta_1)$ мәнін есептейтін болсақ, онда $J(2.5) = \frac{1}{6}((1-2.5)^2 + (2-2.5)^2 + (3-2.5)^2) = 5.25$ (2.13-сурет) болады.



2.11-сурет. $h_\theta(x) = x$ функциясының графигі

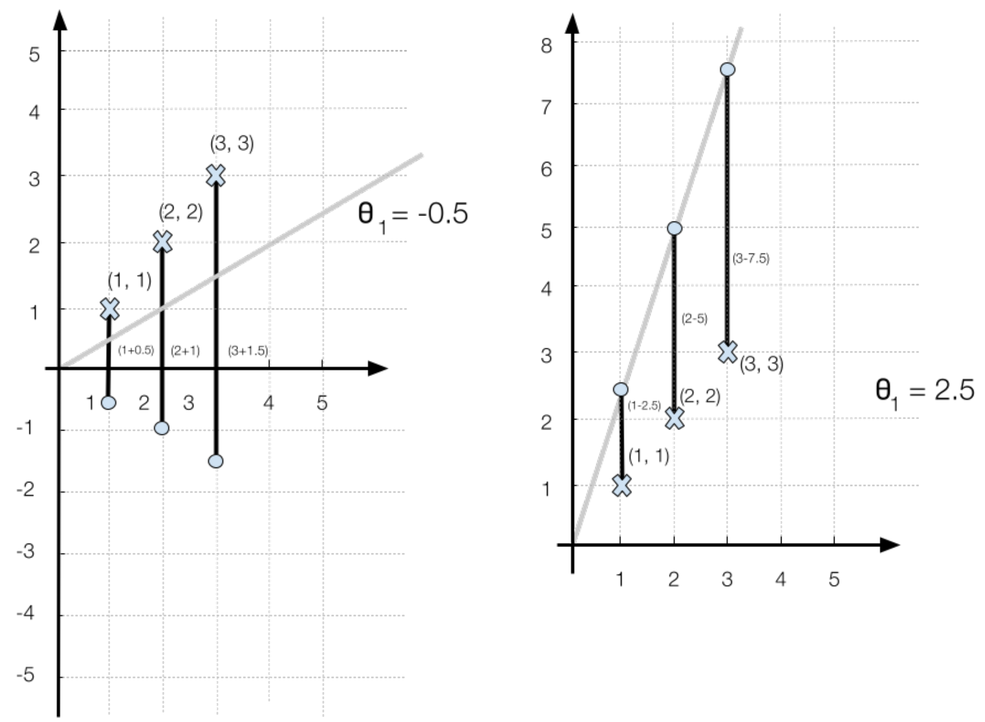


2.12-сурет. $h_\theta(x) = 0.5x$ және $h_\theta(x) = 1.5x$ функциясының графиктері

Ал енді осы 5 жағдайды ($\theta_1 = 1$, $\theta_1 = 0.5$, $\theta_1 = 1.5$, $\theta_1 = -0.5$, $\theta_1 = 2.5$) топтастырайық (2.3-кесте).

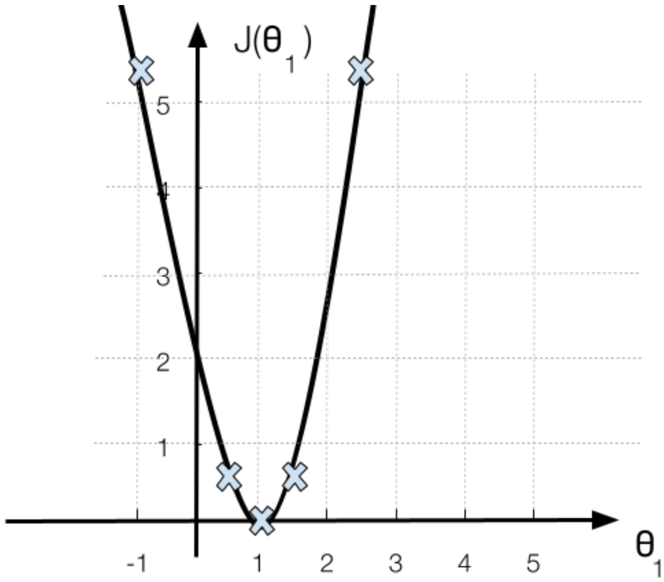
θ_1	-0.5	0.5	1	1.5	2.5
$J(\theta_1)$	5.25	0.58	0	0.58	5.25

2.3-кесте. Түрлі θ_1 мәндері үшін $J(\theta_1)$ мәндерінің анықталуы



2.13-сурет. $h_\theta(x) = -0.5x$ және $h_\theta(x) = 2.5x$ функциясының графиктері

Әр θ_1 мәні үшін $J(\theta_1)$ мәндері есептеу арқылы табылады. Енді осы жағдайларды графикке енгізсек, төмендегідей парабола шығады (2.14-сурет).



2.14-сурет. $J(\theta_1)$ функциясының графигі

Ал осы $J(\theta_1)$ функциясының экстрим нүктесі $\theta_1 = 1$ болғанда ғана орындалады. Яғни $\theta_0 = 0$ болған жағдайда $h_\theta(x) = \theta_1 x$ функциясы X мәндерімен анықталса, $J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$ баға функциясы θ_1 мәнімен анықталады және $J(\theta_1)$ функциясының мәні парабола болады. $minimize J(\theta_1)$ мәнін есептеу үшін осы параболаның төбесін табу жеткілікті.

2.1.5. Баға функциясының экстрим нүктесі (Жалпы жағдай)

Енді баға функциясы $\theta_0 \neq 0$ және $\theta_1 \neq 0$ болғандағы жағдайды қарастырайық. Бұл жағдайда $h_\theta(x) = \theta_1 x + \theta_0$ сызығы координата осьтерінің басы арқылы өтпейді (2.15-сурет). Яғни $h_\theta(x)$ функциясы X-қа тәуелді болса, онда $J(\theta_0, \theta_1)$ функциясы θ_0 мен θ_0-ге байланысты болады. Мысалы, өзіміздің емтиханға дайындалу және емтиханнан баға алу жағдайын қарастырайық. Алдыңғы мысалдағы жағдайдан бөлек жаңа деректер берілсін делік (2.16-сурет).

Біздің мақсатымыз — осы деректерді қанағаттандыратын сызықты табу ($h_\theta(x) = \theta_1 x + \theta_0$). Ол үшін алдыңғы тақырыпта көрсетілгендей θ_0 және θ_1 мәндерін қарастырып көрейік:

- **1-жағдай** $(\theta_0 = 3, \theta_1 = 0.25)$. Егер $x = 0$ болса, онда $h_\theta(0) = 0.25 \times 0 + 3 = 3$ болады. Егер $x = 4$ болса, онда $h_\theta(4) = 0.25 \times 4 + 3 = 4$ болады. Яғни төмендегідей график шығарып аламыз (2.17-Сурет). Ал енді осы сызық үшін және осы деректер үшін $J(\theta_0, \theta_1)$-дің мәнін есептейік, яғни

$$J(3, 0.25) = \frac{1}{2 \times 4}((0.25 \times 0 + 3 - 2)^2 +$$
$$(0.25 \times 1 + 3 - 2)^2 + (0.25 \times 1 + 3 - 3)^2 +$$
$$(0.25 \times 3 + 3 - 4)^2) = 0.33$$

болады.

- **2-жағдай** ($\theta_0 = 3, \theta_1 = -0.25$). Егер $x = 0$ болса, онда $h_{\theta_0}(0) = -0.25 \times 0 + 3$ болады. Егер $x = 4$ болса, онда $h_\theta(4) = -0.25 \times 4 + 3 = 2$ болады. Яғни төмендегідей график шығарып аламыз (2.18-Сурет). Ал енді осы сызық үшін және осы деректер үшін $J(\theta_0, \theta_1)$-дің мәнін есептейік, яғни

$$J(3, -0.25) = \frac{1}{2 \times 4}((-0.25 \times 0 + 3 - 2)^2 +$$
$$(-0.25 \times 1 + 3 - 2)^2 + (-0.25 \times 1 + 3 - 3)^2 +$$
$$(-0.25 \times 3 + 3 - 4)^2) = 0.58$$

болады.

- **3-жағдай** ($\theta_0 = 3, \theta_1 = -3$). Егер $x = 0$ болса, онда $h_\theta(0) = -3 \times 0 + 3 = 3$ болады. Егер $x = 1$ болса, онда $h_\theta(1) = -3 \times 1 + 3 = 0$ болады. Яғни төмендегідей график шығарып аламыз (2.19-сурет). Ал енді осы сызық үшін және осы деректер үшін $J(\theta_0, \theta_1)$-дің мәнін есептейік, яғни

$$J(3, -3) = \frac{1}{2 \times 4}((-3 \times 0 + 3 - 2)^2 +$$
$$(-3 \times 1 + 3 - 2)^2 + (-3 \times 1 + 3 - 3)^2 +$$
$$(-3 \times 3 + 3 - 4)^2) = 14.25$$

болады.

- **4-жағдай** ($\theta_0 = 3, \theta_1 = 3$). Егер $x = 0$ болса, онда $h_\theta(0) = 3 \times 0 + 3 = 3$ болады. Егер $x = -1$ болса, онда $h_\theta(-1) = 3 \times (-1) + 3 = 0$ болады. Яғни төмендегідей график шығарып аламыз (2.20-сурет). Ал енді осы сызық үшін және осы деректер үшін $J(\theta_0, \theta_1)$-дің мәнін есептейік, яғни

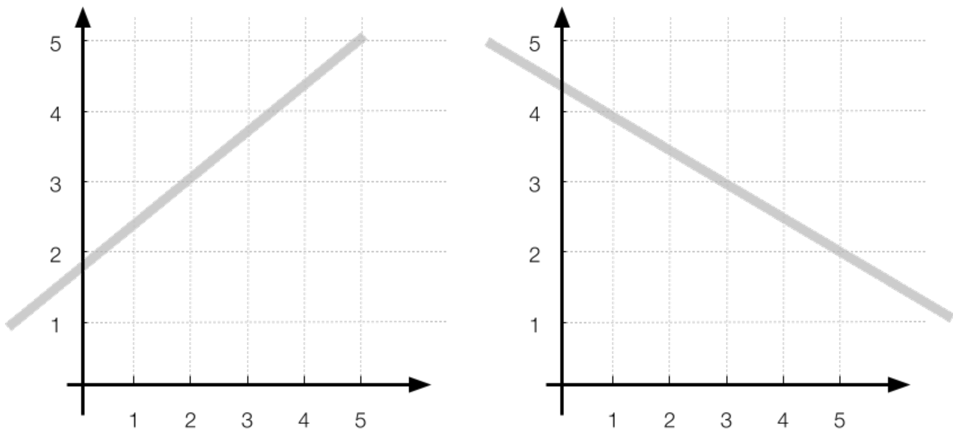
$$J(3, 3) = \frac{1}{2 \times 4}((3 \times 0 + 3 - 2)^2 +$$
$$(3 \times 1 + 3 - 2)^2 + (3 \times 1 + 3 - 3)^2 +$$
$$(3 \times 3 - 4)^2) = 11.25$$

болады.

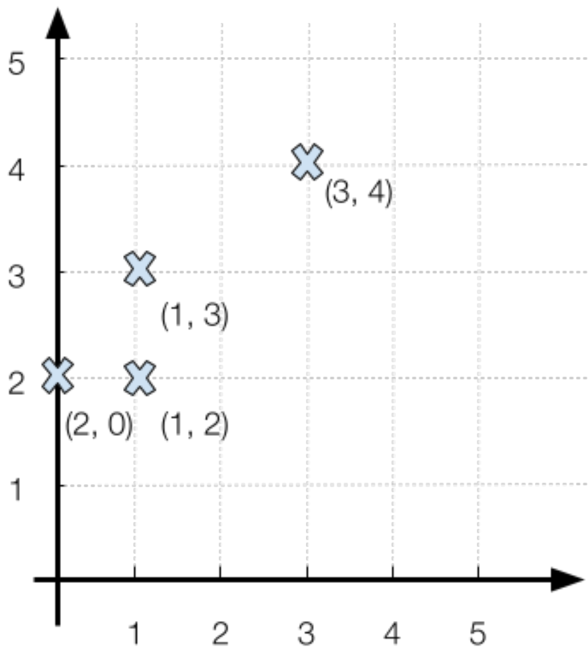
- **5-жағдай** ($\theta_0 = 2, \theta_1 = \frac{2}{3}$). Егер $x = 0$ болса, онда $h_\theta(0) = \frac{2}{3} \times 0 + 2 = 2$ болады. Егер $x = 3$ болса, онда $h_\theta(3) = \frac{2}{3} \times 3 + 2 = 4$ болады. Яғни төмендегідей график шығарып аламыз (2.21-сурет). Ал енді осы сызық үшін және осы деректер үшін $J(\theta_0, \theta_1)$-дің мәнін есептейік, яғни

$$J(2, \frac{2}{3}) = \frac{1}{2 \times 4}((\frac{2}{3} \times 0 + 2 - 2)^2 +$$
$$(\frac{2}{3} \times 1 + 2 - 2)^2 + (\frac{2}{3} \times 1 + 2 - 3)^2 +$$
$$(\frac{2}{3} \times 3 + 2 - 4)^2) = 0.0689$$

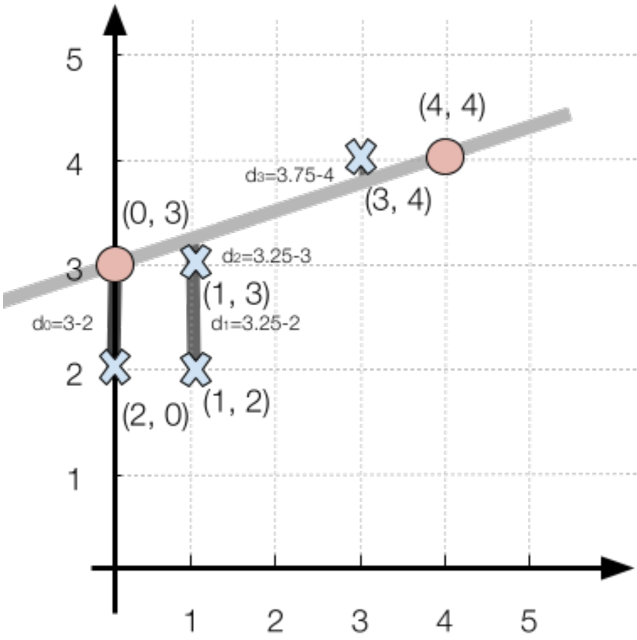
болады.



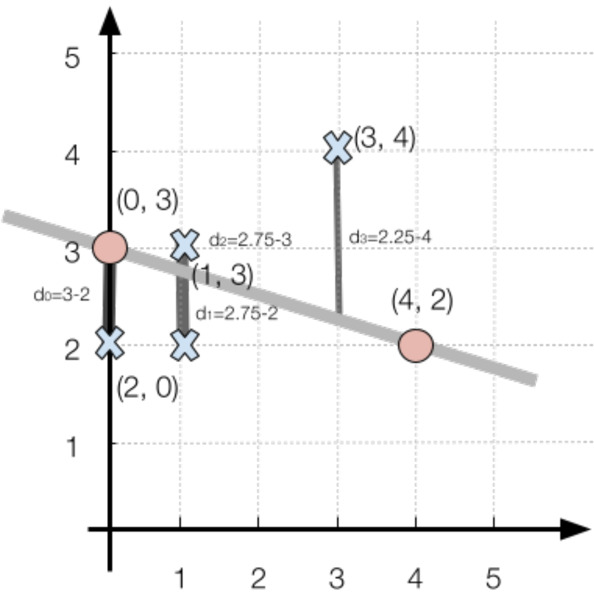
2.15-сурет. Координата осьтерінің басы арқылы өтпейтін сызықтар



2.16-сурет. Үйретуге арналған жаңа дерек



2.17-сурет. $h_\theta(x) = \theta_1 x + \theta_0$ сызығының теңдеуі, мұндағы $\theta_0 = 3, \theta_1 = 0.25$

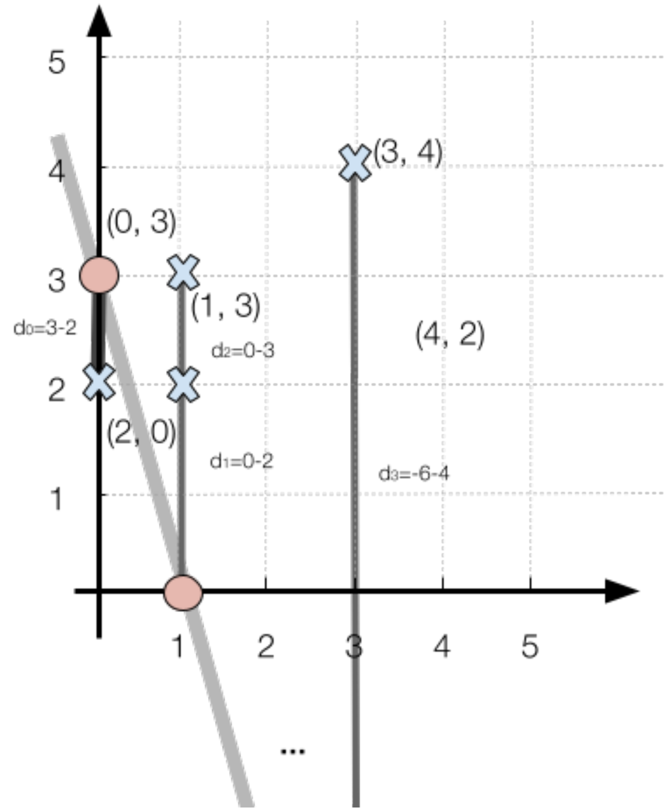


2.18-сурет. $h_\theta(x) = \theta_1 x + \theta_0$ сызығының теңдеуі, мұндағы $\theta_0 = 3, \theta_1 = -0.25$

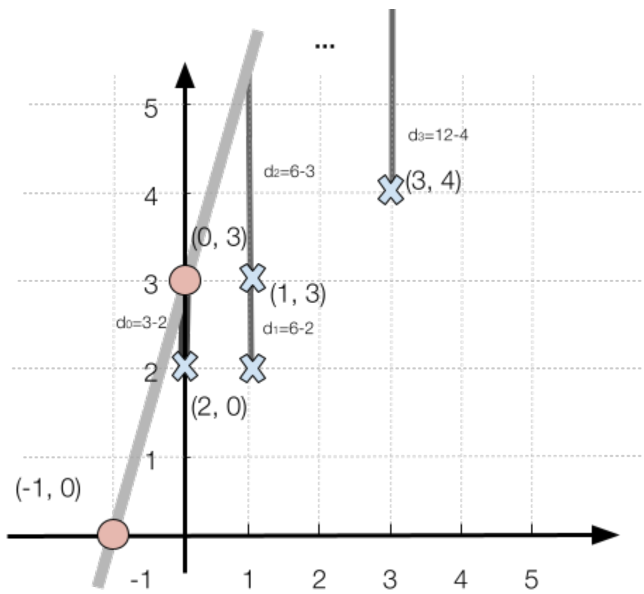
Осы 5 жағдайды біріктірсек, мынадай қортынды шығады (2.4-кесте).

θ_0	2	3	3	3	3
θ_1	$\frac{2}{3}$	3	-3	0.25	-0.25
$h_\theta(x)$	0.0689	11.25	14.25	0.5859	0.3359

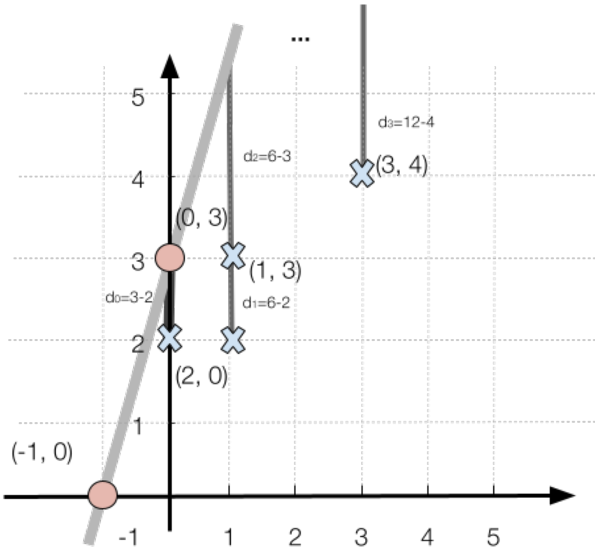
2.4-кесте. Сызықтық регрессияның түрлі θ_0 және θ_1 мәндері



2.19-сурет. $h_\theta(x) = \theta_1 x + \theta_0$ сызығының теңдеуі, мұндағы $\theta_0 = 3, \theta_1 = -3$

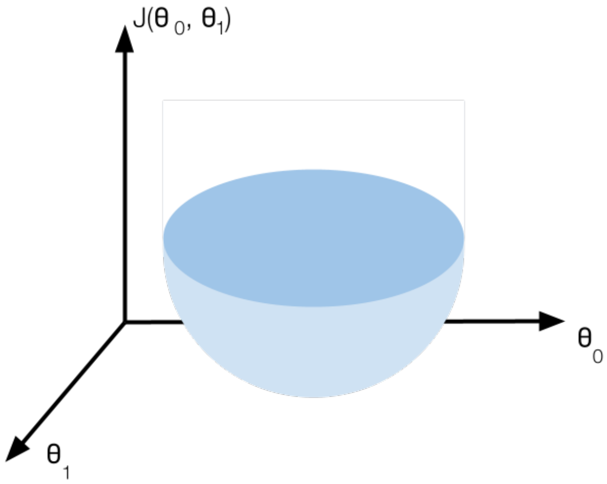


2.20-сурет. $h_\theta(x) = \theta_1 x + \theta_0$ сызығының теңдеуі, мұндағы $\theta_0 = 3, \theta_1 = 3$



2.21-сурет. $h_\theta(x) = \theta_1 x + \theta_0$ түзуінің теңдеуі, мұндағы $\theta_0 = 2, \theta_1 = \frac{2}{3}$

Яғни $\theta_0 = 2$ және $\theta_1 = \frac{2}{3}$ болғанда, ең кіші мәнді ($h_\theta(x) = 0.0689$) шығарып аламыз. Енді осы мәліметтерді үштік кеңістіктегі координата жүйелеріне енгізейік. X абциссасы θ_0 мәнін қабылдасын, Y ординатасы θ_1 мәнін қабылдасын, ал Z $h_\theta(x)$ мәнін қабылдасын. Бұл мәліметтерді үштік кеңістіктегі координата жүйелеріне енгізсек, төмендегідей график шығарып аламыз (2.22-Сурет).



2.22-сурет. Үштік кеңістіктегі $J(\theta_0, \theta_1)$ теңдеуінің графигі

Яғни график үштік кеңістіктегі парабола пішіндес болады. $h_\theta(x)$ функциясы біздің деректерімізді қанағаттандыратындай θ_0, θ_1 мәндерін табу үшін, $J(\theta_0, \theta_1)$ функциясының экстрим нүктесін табу қажет. $J(\theta_0, \theta_1)$ функциясының экстрим нүктесін табу үшін, $J(\theta_0, \theta_1)$ функциясының төбесін табу керек. Үштік кеңістіктегі параболаның төбесі біздің жағдайымызда $\theta_0 = 2$ және $\theta_1 = \frac{2}{3}$ болғанда ғана орындалады.

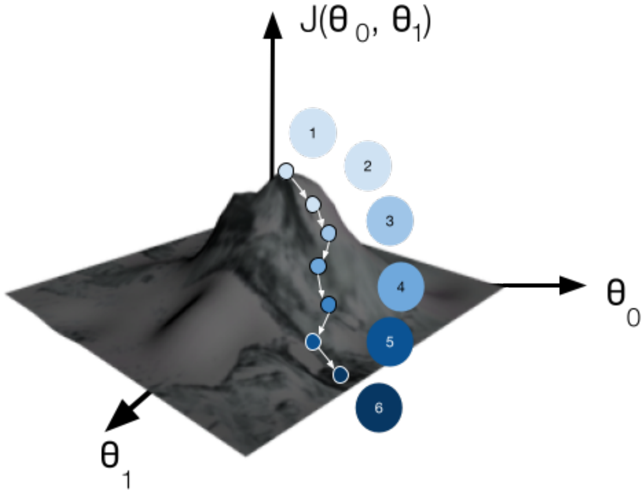
2.1.6. Градиенттік түсу

Жалпы жағдайда θ_0 мен θ_1 таңдап ала беру оптималды шешім емес. Яғни біз барлық θ_0 мен θ_1 қамтып, барлық θ_0 мен θ_1 үшін $J(\theta_0, \theta_1)$ мәнін есептеп табу еш мүмкін емес. Сол себепті "градиенттік түсу" сияқты алгоритмдер қолданылады. Градиенттік түсу тек сызықтық регрессияға ғана емес, сонымен бірге басқа да машинаны үйрету модельдерінде қолданылады. Градиенттік түсудің басты мақсаты — машинаны үйрету модельдерінің оптималды шешімін табу. Біздің жағдайда $J(\theta_0, \theta_1)$ мәндерінің ең кіші мәні үшін θ_0 мен θ_1 мәнін есептеп табу. Сызықтық регрессияда градиенттік түсу алгоритмінің қалай жұмыс істейтінін қарастырайық:

Алгоритм (Градиенттік түсу)

- θ_0 мен θ_1 мәндерін кездейсоқ санға теңеу.
- $J(\theta_0, \theta_1)$ минимум нүктесіне дейін жеткенше θ_0, θ_1 мәндерін өзгертіп отыру.

Графикалық тұрғыда алгоритм былайша түсіндіріледі (2.23-сурет).



2.23-сурет. Үштік кеңістіктегі $J(\theta_0, \theta_1)$ функциясының графигі

- Бастапқыда біздің θ_0, θ_0 мәндері 1-ші нүктеде орналассын (2.23-сурет). Біздің мақсатымыз 6-шы нүктедегі $J(\theta_0, \theta_1)$ минимумды қабылдайтын нүктеге жету. Әрбір қадамда біз 360° болатын айналамызға қарап шығып, ең төмен бағытты көрсететін аймақты таңдаймыз.
- 2-ші нүктеде тұрып, қайтадан 360° болатын айналамызға қарап шығып, ең кіші бағытты таңдап, сол бағытта кішкентай ғана қадам жасаймыз. Осылайша 3-ші нүктеге өтеміз. Соған сәйкес θ_0 мен θ_1 мәндері де өзгереді.
- 3-ші нүктеде жоғарыда аталған қадамдарды қайта орындап барып, 4-ші нүктеге өтеміз.

- 4-ші нүктеден 5-ші нүктеге.
- 5-ші нүктеден 6-шы нүктеге.
- 6-шы нүктеде $J(\theta_0, \theta_1)$ мәні минимумды қабылдап, әрі қарай (θ_0, θ_1) нүктелерінің жылжуы тоқталады. Осылайша градиенттік түсу арқылы $J(\theta_0, \theta_1)$ мәні минимумды қабылдайтын θ_0 мен θ_1 мәндерін табамыз. Әрбір қадамда біз айналамызға қарап, төменге түсу бағытын іздедік. Бұл операция функцияның туындысын табу арқылы жүзеге асады. Математика және физика курстарынан білетіндеріңіздей, туынды қандай да бір функцияның өзгеру шамасын анықтайды. Біздің жағдайда әр қадамда төменге түсу бағыты аса қызықты болғандықтан, $J(\theta_0, \theta_1)$ функциясының θ_0 мен θ_1 бағытындағы туынды да қызықты. Яғни градиенттік түсу алгоритмі былайша жұмыс істейді.

$J(\theta_0, \theta_1)$ функциясы минимум нүктесіне дейін жеткенше {

$$tempVal_0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$tempVal_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 = tempVal_0$$
$$\theta_1 = tempVal_1$$

}

Енді осы алгоритмдегі әрбір қадамды жеке қарастырайық.

- Алдымен $J(\theta_0, \theta_1)$ функциясында θ_0 және θ_1 бағытындағы туындыны есептеп алуымыз қажет $\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$, $\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$.
- Кейін осы шамаларды α-ға (үйрету жылдамдығын анықтайтын шамаға) көбейтуіміз қажет $\alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$, $\alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$.
- θ_0 мен θ_1-ден 2-пунктте есептелген мәндерді алып тастау қажет: $tempVal_0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$, $tempVal_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$.
- 3-пункттегі мәндерді θ_0 мен θ_1-ге меншіктеу қажет: $\theta_0 = tempVal_0$, $\theta_1 = tempVal_1$
- Жаңа θ_0 мен θ_1 мәндерін қолдана отырып, $J(\theta_0, \theta_1)$ мәнін есептеп, ол мән алдыңғы мәннен аз болса, онда қайтадан 1-пунктті орындау керек. Әйтпесе алгоритмді тоқтату қажет.
- Мұндағы ескеретін бірінші жайт: 4-пунктті орындамас бұрын 3-пунктты орындау қажет. Яғни $\theta_0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$ деп есептеп алып, жаңа θ_0 мәнімен $\theta_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$ мәнін есептемейміз, керісінше, 3-пунктті толығымен орындап, яғни $\theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$, $\theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$ мәндерін есептеп және оларды $tempVal_0$, $tempVal_1$ мәндеріне теңеп, кейін $tempVal_0$ мен $tempVal_1$-ді θ_0 мен θ_1-ге меншіктейміз.

- Екінші бір айта кететін жайт — α үйрету шамасы жайлы болмақ, α — үйрету жылдамдығын анықтайтын шама. Яғни бұл — әрбір қадамда туындыны табу барысында төменге түсетін бағытта қаншалықты үлкен/кіші қадам жасауды анықтайтын шама. Егер α мәні үлкен болса, онда сәйкесінше сол бағытта үлкен қадам жасаймыз. Керісінше, α мәні кіші мәнге ие болса, онда сол бағытта кіші қадам жасаймыз. Көбіне α кіші мәнге ие болып, кіші қадам жасағаны дұрыс. Оның себебін келесі параграфта түсіндіретін боламыз.

2.1.6.1. Градиенттік түсу параметрлері

Алдыңғы параграфта айтылғандай, θ_0 мен θ_1 мәні градиенттік түсуде былайша өзгереді:

$J(\theta_0, \theta_1)$ мәні минимумды қабылдамағанша {

$$tempVal_0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

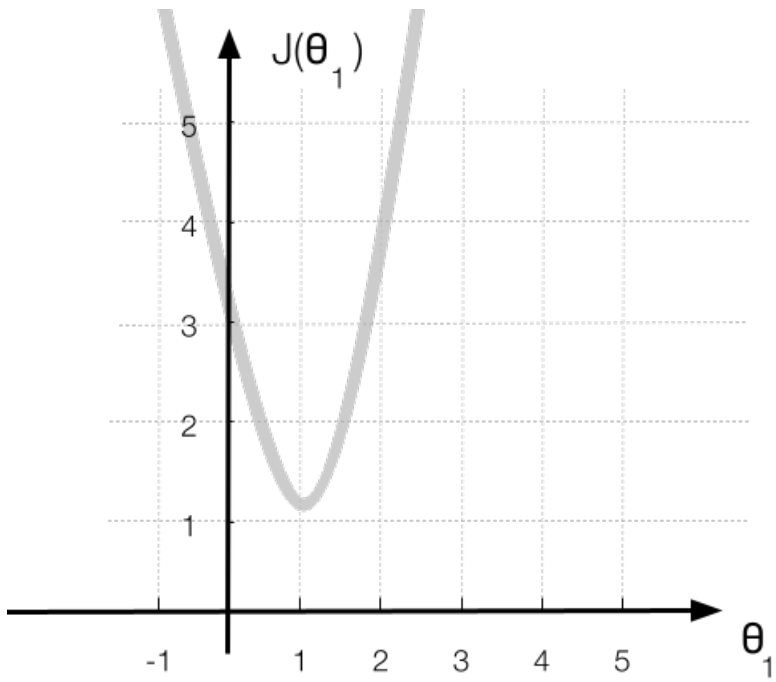
$$tempVal_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 = tempVal_0$$

$$\theta_1 = tempVal_1$$

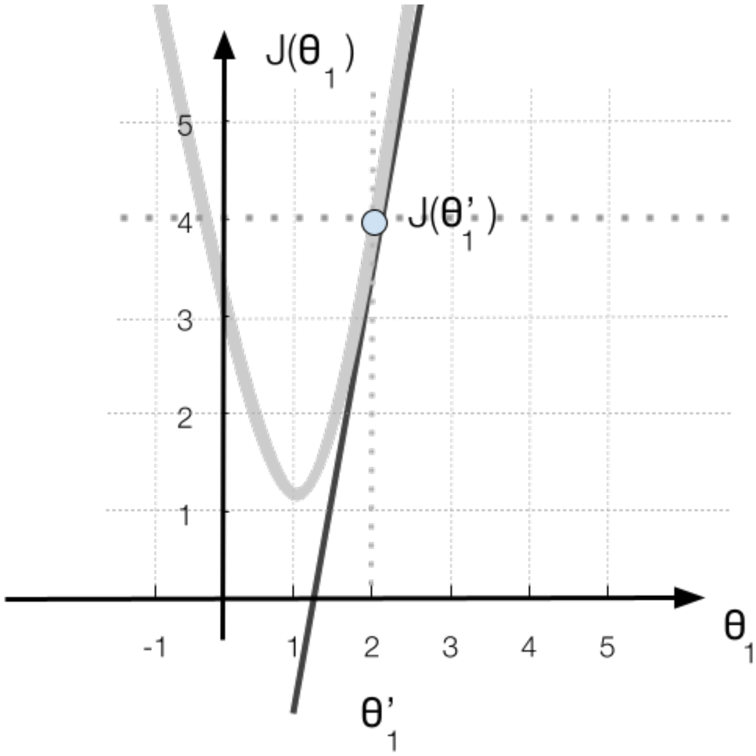
}

Алдымен мұндағы туынды бөлігін толығымен түсіндіріп шығайық. Ол үшін $J(\theta_0, \theta_1)$ функциясының ең қарапайым түрі $J(\theta_1)$ функциясын қарастырайық ($\theta_0 = 0$). $\theta_0 = 0$ болғандағы $J(\theta_1)$ функциясы төмендегідей параболамен кескінделеді (2.24-сурет).

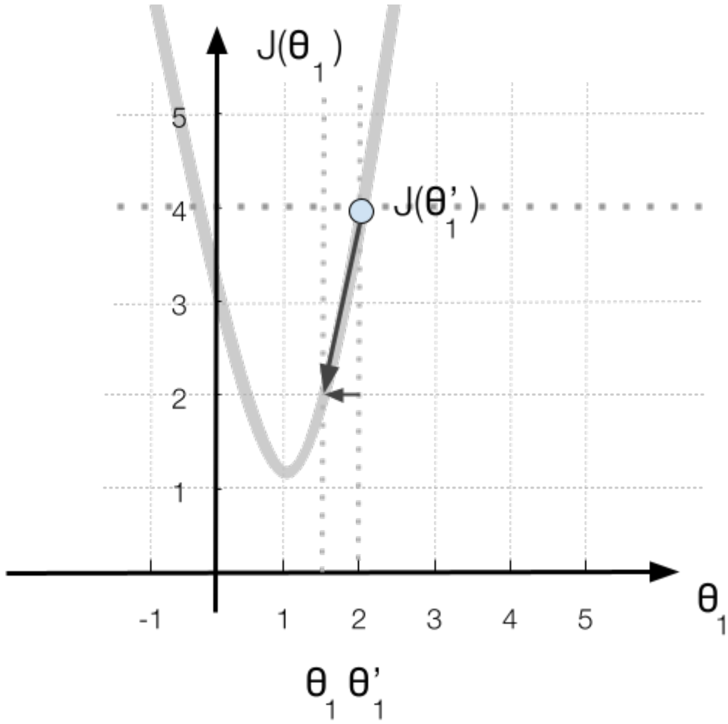


2.24-сурет. $J(\theta_1)$ функциясының парабола графигі

θ_1' нүктесі үшін $\frac{\partial}{\partial\theta_1}J(\theta_1)$ туындысы сол нүктедегі жанамамен анықталады (2.25-сурет). Ал жанаманың бағыты біздің жағдайда оң болады (2.26-сурет). Сол себепті $\theta_1 = \theta_1' - \alpha\frac{\partial}{\partial\theta_1}J(\theta_1)$ жанамасын былайша жазсақ болады: $\theta_1 = \theta_1' - \alpha \times positiveNumber$ немесе θ_1-дің мәні бұрынғы θ_1'-тің мәнінен кемиді (θ_1'-тан оң санды азайтып, оны θ_1-ге меншіктейміз).

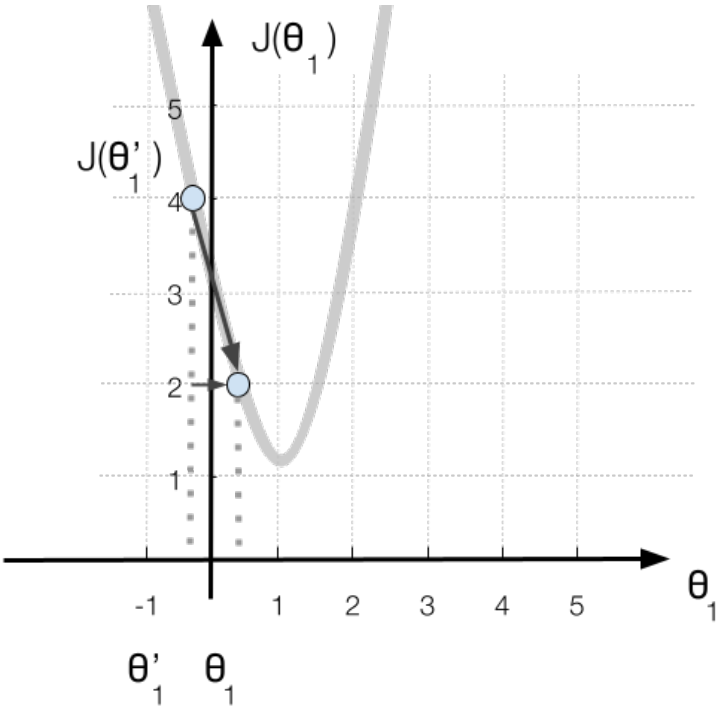


2.25-сурет. $J(\theta_1)$ функциясына $\theta_1 = \theta_1'$ нүктесіндегі жанама



2.26-сурет. $J(\theta_1)$ функциясында θ_1'-ден θ_1 нүктесіне көшу

Ал θ_1' саны мынадай мәнге ие болса (2.27-сурет), онда θ_1 мәні, керісінше, артады. Себебі, θ_1' нүктесінде туынды (немесе жанама) теріс болғандықтан, $\frac{\partial}{\partial\theta_1}J(\theta_1)$-дің мәні теріс сан, ал $\theta_1' - \alpha\frac{\partial}{\partial\theta_1}J(\theta_1)$-дің мәні бұрынғы θ_1-дің мәнінен артады (2.26-сурет).



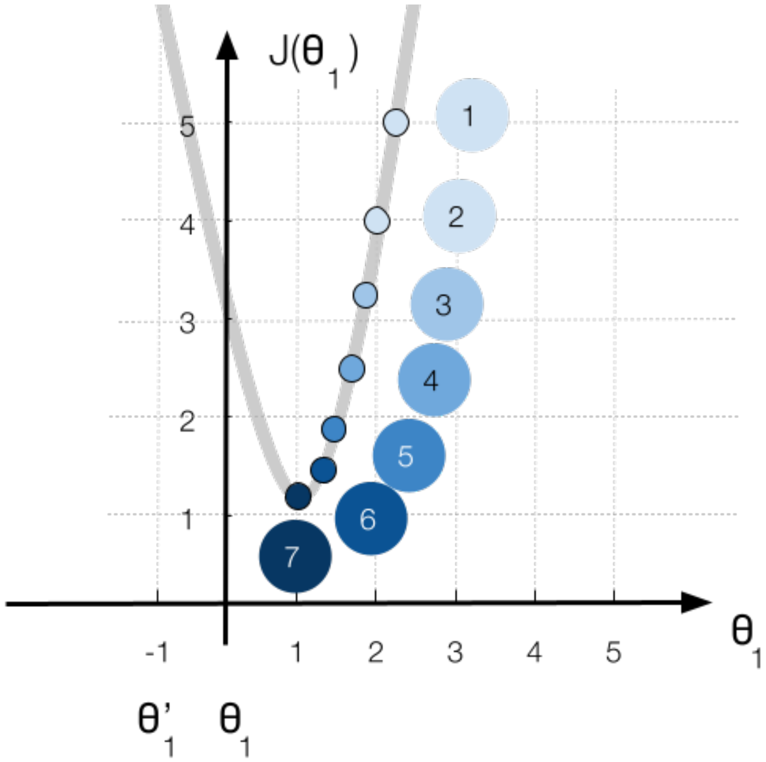
2.27-сурет. $J(\theta_1)$ функциясында θ_1'-ден θ_1 нүктесіне көшу

Ал θ_1' — парабола төбесінде болса, онда жанама бағыты 0-ге тең болады. Соған сәйкес $\frac{\partial}{\partial\theta_1}J(\theta_1)$-дің мәні 0-ге тең және $\theta_1' - \alpha\frac{\partial}{\partial\theta_1}J(\theta_1) = \theta_1'$ болады. Яғни ешқандай өзгеріс болмай, θ_1-дің мәні бұрынғы мәнінде қалады. Осылайша градиенттік түсу арқылы $J(\theta_1)$-дің мәні минимумды қабылдайтындай θ_1-дің мәнін таба аламыз.

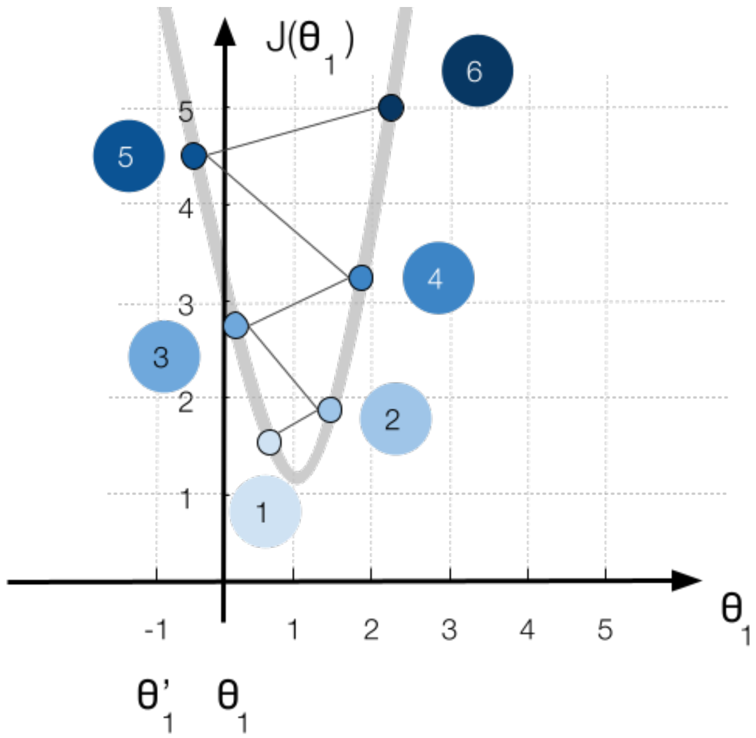
Екінші қарастыратынымыз — α үйрету шамасы. Алдыңғы бөлімде айтқанымыздай, α кіші және үлкен мән қабылдауы мүмкін. Егер α өте кіші мән қабылдаса, $J(\theta_1)$ үшін θ_1 мәні төмендегіше өзгереді (2.28-сурет).

Яғни алғашында θ_1 мәні 1-ші нүктеде орналассын. Әрбір $\theta_1 = \theta_1 - \alpha\frac{\partial}{\partial\theta_1}J(\theta_1)$-ді меншіктеуі арқылы θ_1 мәні 1-ден 2-ге, 2-ден 3-ке, 3-тен 4-ке, тағы сол сияқты 6-дан 7-ге ауысады. Яғни әрбір қадамымыз кіші мәнге өзгеріп отырады. Ал егер α үлкен мәнге ие болса, онда θ_1 және $J(\theta_1)$ мәндері төмендегіше өзгереді (2.29-сурет).

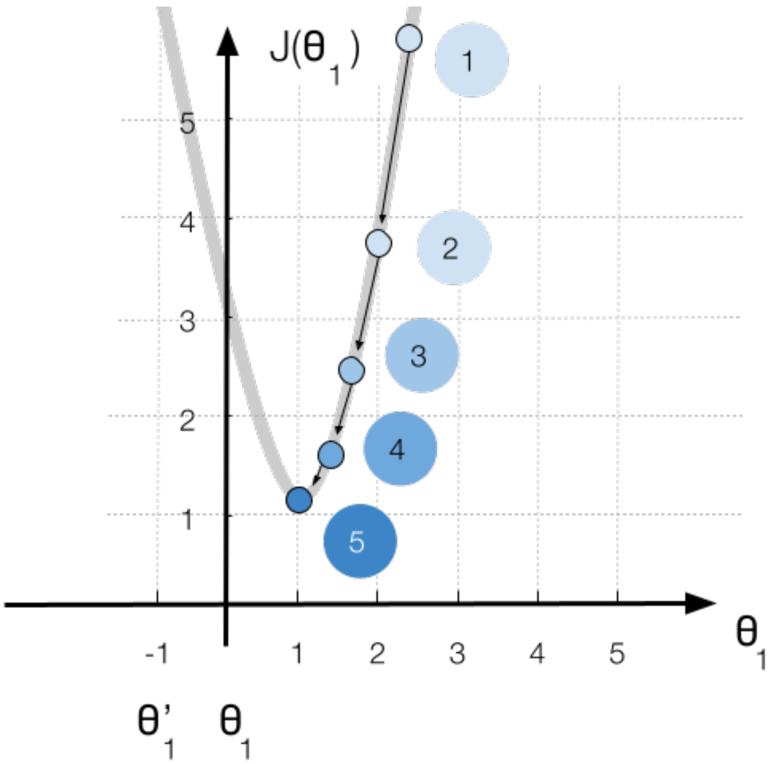
Алғашында θ_1 мәні 1-ші нүктеде болса, онда $\theta_1 = \theta_1 - \alpha\frac{\partial}{\partial\theta_1}J(\theta_1)$ өзгеру арқылы 2-ге үлкен мәнді қадаммен көшеді. 2-ші нүктеден 3-ші нүктеге, 3-тен 4-ке тағы сол сияқты көшеді. Яғни әрбір қадамда θ_1 мәні үлкен өзгеріске ұшырағандықтан, парабола өз төбесінен алшақтай береді.



2.28-сурет. Градиенттік түсудегі α мәнінің кіші болғандағы жағдайы



2.29-сурет. Градиенттік түсудегі α мәнінің үлкен болғандағы жағдайы



2.30-сурет. Градиенттік түсудегі α мәнінің үйретуге тигізетін әсері

Сонымен қатар тұрақты α кіші мәні үшін $\theta_1 - \alpha\frac{\partial}{\partial\theta_1}J(\theta_1)$ шамасы әрбір қадам сайын кішірейе береді (2.30-сурет). Мысалы, бастапқыда θ_1 мәні 1-ші нүктеде болса, онда 2-ші нүктеге көшкенде d_1 шамасына өзгерсін. Кейін 2-ші нүктеден 3-ші нүктеге d_2 шамасына көшсін. Ал 3-ші нүктеден 4-шіге d_3 және 4-шіден 5-ші нүктеге көшкенде d_4 шамасына өзгерсін. Сонда әрбір қадамда $(d_1 > d_2 > d_3 > d_4 > \ldots)$ көшу шамасы өзгере береді. Яғни алғашында $\theta_1 - \alpha\frac{\partial}{\partial\theta_1}J(\theta_1)$ шамасы үлкен шамаға өзгерсе, кейін $J(\theta_1)$ мәні минимумға жақындаған сайын $\theta_1 - \alpha\frac{\partial}{\partial\theta_1}J(\theta_1)$ мәні 0-ге жақындай береді. 0-ге жақындағанда θ_1-дің мәні өзгермей, үйрету тоқтатылады.

2.1.6.2. Сызықтық регрессияға арналған градиенттік түсу

$J(\theta_0, \theta_1)$ шамасы θ_0 және θ_0 мәндерімен анықталғандықтан, сызықтық регрессияда градиенттік түсудің толықтай қолданылуын қарастырайық. Ол үшін бізге қалғаны тек $\frac{\partial}{\partial\theta_0}J(\theta_0, \theta_1)$ және $\frac{\partial}{\partial\theta_1}J(\theta_0, \theta_1)$ шамаларын есептеп шығару. $\frac{\partial}{\partial\theta_0}J(\theta_0, \theta_1)$ және $\frac{\partial}{\partial\theta_1}J(\theta_0, \theta_1)$ туындыларын есептесек, онда олар мынадай мәндерге ие болады:

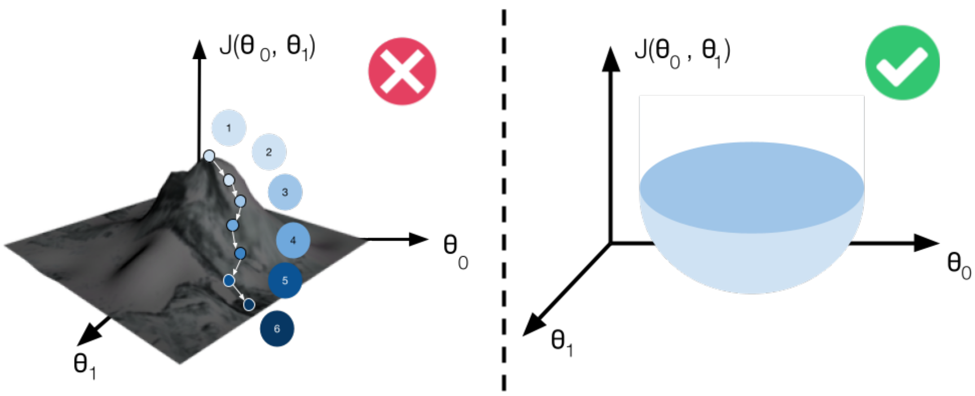
- $\frac{\partial}{\partial\theta_0}J(\theta_0, \theta_1) =$
  $\frac{\partial}{\partial\theta_0}\left(\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2\right) =$
  $\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})$

- $\frac{\partial}{\partial\theta_1}J(\theta_0,\theta_1) =$
  $\frac{\partial}{\partial\theta_1}(\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2) =$
  $\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x^{(i)}$

Яғни барлық алгоритм толығымен былайша жазылады:

$J(\theta_0,\theta_1)$ мәні минимум қабылдамағанша {
$$\theta_0 = \theta_0 - \alpha(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)}))$$
$$\theta_1 = \theta_1 - \alpha(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)}))x^{(i)}$$
}

$J(\theta_0,\theta_1)$ функциясының графигі 2.31-суреттің сол жағындағы көрсетілген формада емес, 2.31-суреттің оң жағындағы көрсетілген дөңес формада болады. Яғни дөңес формадағы функцияның тек бір ғана минимум нүктесі болады әрі ол нүкте сол дөңес функцияның төбесінде болады.



2.31-сурет. Сызықтық регрессиядағы градиенттік түсу графигі

Кейде "градиенттік түсуді" "топтық градиенттік түсу" деп те атайды. Себебі, $\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2$ шамасында i мәні 1-ден m-ге дейін жүгіріп, топтық $(h_\theta(x^{(i)})-y^{(i)})^2$ мәні есептелгендіктен, градиенттік түсу "топтық градиенттік түсу" деп аталады.

Градиенттік түсу тек сызықтық регрессияға ғана емес, сонымен бірге басқа машинаны үйрету модельдеріне жиі қолданылады. Градиенттік түсудің басқа да машинаны үйрету модельдеріне қолданылуы кейінгі параграфтарда түсіндіріледі.

2.1.6.3. Сызықтық алгебра

Алдыңғы параграфтарда бір (x_1) және екі (x_1, x_2) қасиеттермен анықталатын сызықтық регрессияны қарастырдық. Бірақ, көп жағдайда машинаны үйретудің есептерін шығару барысында деректер бір немесе екі қасиетпен емес, сонымен бірге бірнеше қасиетпен анықталуы мүмкін. Мысалы, емтиханнан жақсы баға (y) алу үшін оқушы:

- дайындалуға көп уақыт жұмсау керек (x_1);
- әлеуметтік желіде аз отыру керек (x_2);
- мүмкіндігінше үй тапсырмаларын орындап тұру қажет (x_3);
- сабақта қол көтеріп, жиі тақтаға шығып, бес алып тұру қажет (x_4);
- тағы сол сияқты.

Осындай жағдайларда сызықтық алгебраны қолдану есеп шығаруды жеңілдете түседі. Сызықтық алгебрада векторлар мен матрицаларға қолданылатын амалдар бізді қатты қызықтырады. Сол себепті "бірнеше айнымалыға тәуелді сызықтық регрессияны" түсіндірмей тұрып, сызықтық алгебраға көңіл аударатын боламыз.

2.2. Код

Код жазу барысында алдымен өзімізге керекті кітапханаларды бағдарламаға қосып алуымыз қажет. Біздің жағдайымызда *numpy*, *random*, *matplotlib* кітапханаларын бағдарламаға қосуымыз керек.

numpy — операцияларды сандармен, сандар тізбегімен тез әрі жеңіл түрде орындауға арналған кітапхана.

random — кездейсоқ сандарды генерация жасауға қажетті кітапхана.

matplotlib - график құруға арналған кітапхана.

```
import numpy as np
import random
import matplotlib.pyplot as plt
```

Керекті кітапханаларды кодқа қосқаннан кейін, сызықтық регрессияға керекті X және Y деректерін генерация жасауға кірісе берсек болады.

2.2.1. Деректер генерациясы

Деректер генерациясын бастамас бұрын өзімізге белгілі сызықтық болжам функциясын жазайық.

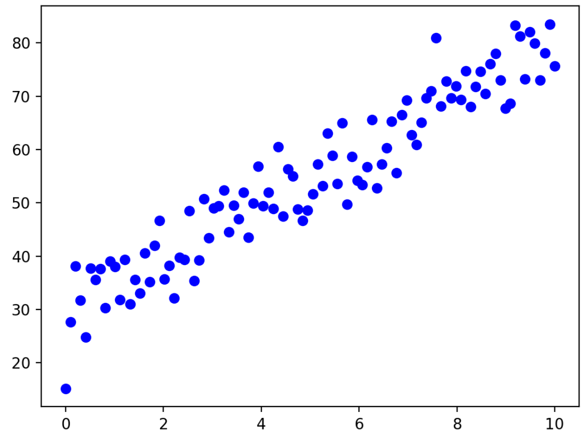
```
def h(x, thetaZero, thetaOne):
    return x * thetaOne + thetaZero
```

Енді $\theta_0 = 10$ және $\theta_1 = 5$ болған жағдайда X және Y деректерін генерация жасайық. X дерегі 0 мен 10 арасында орналасқан. Ал Y дерегін генерация жасау үшін сызықтық болжам функциясы және Гаусс шуы қолданылады. Гаусс функциясын кейінгі параграфтарда толығымен қарастыратын боламыз. Қазірше, бізде (X, Y) деректері ғана бар деп ойлаңыз.

```
trueThetaZero = 10
trueThetaOne = 5
X = np.linspace(0, 10, 100)
Y = []
for i in range(len(X)):
    Y.append(h(X[i], trueThetaZero, trueThetaOne) + abs(random.gauss(20,5)))
```

Егер бұл арқпараттардың графигін сызсақ, ол мынадай кескінде болады (2.32-сурет).

```
plt.scatter(X, Y, c='b')
plt.show()
```



2.32-сурет. Сызықтық регрессия үшін генерацияланған дерек

Деректерді генерациялауға арналған толық код мынадай:

```
import numpy as np
import random
import matplotlib.pyplot as plt

def h(x, thetaZero, thetaOne):
    return x * thetaOne + thetaZero

trueThetaZero = 10
trueThetaOne = 5
X = np.linspace(0, 10, 100)
Y = []
for i in range(len(X)):
    Y.append(h(X[i], trueThetaZero, trueThetaOne)
                + abs(random.gauss(20,5)))

plt.scatter(X, Y, c='b')
plt.show()
```

2.2.2. Сызықтық регрессия

Сызықтық регрессия кодын толығымен жазуды баға функциясынан бастаймыз. Баға функциясы былайша анықталады: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$. Ол математикалық өрнек кодта былайша жазылады:

```
def costFunction(yTrue, yPred):

    totalSum = 0
    m = len(yTrue)

    for i in range(m):
        totalSum += (yPred[i] - yTrue[i])
                        * (yPred[i] - yTrue[i])

    return totalSum / (2 * m)
```

Ал енді сызықтық регрессияға арналған градиенттік түсу функциясын жазайық.

```
def gradientDescent(y, x, alpha, errorRate):

    m = len(y)

    thetaZero = random.randint(0, 1e3)
    thetaOne = random.randint(0, 1e3)

    yPred = []
    for i in range(m):
        yPred.append(h(x[i], thetaZero, thetaOne))

    cost = costFunction(y, yPred)

    iteration = 0
    while cost > errorRate:
        print(iteration, cost, thetaZero, thetaOne)

        sumZero = 0
        sumOne = 0
        for i in range(m):
            sumZero += (yPred[i] - y[i])
            sumOne += (yPred[i] - y[i]) * x[i]
        sumZero =  sumZero / m
        sumOne =  sumOne / m

        tempZero = thetaZero - alpha * sumZero
        tempOne = thetaOne - alpha * sumOne

        thetaZero = tempZero
        thetaOne = tempOne

        for i in range(m):
            yPred[i] = h(x[i], thetaZero, thetaOne)

        cost = costFunction(y, yPred)
        iteration += 1

    return thetaZero, thetaOne
```

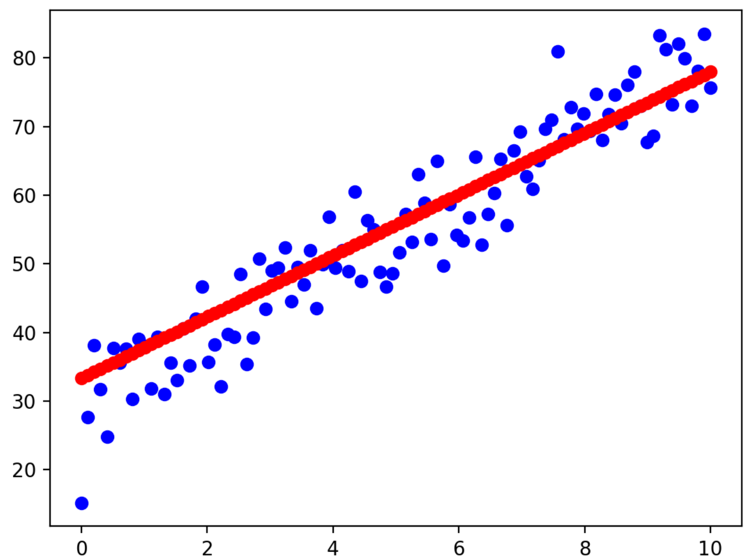
Функция (X, Y) дерегін, үйрету шамасын және баға функциясының қателік шегін көрсететін параметрлер қабылдасын. Алдымен θ_0, θ_1 (thetaZero, thetaOne) мәндерін кездейсоқ сандарға теңейміз. Кейін сол (θ_0, θ_1) параметрлер арқылы болжамды yPred немесе $h_\theta(x)$ мәндерін есептейміз. $h_\theta(x)$ (немесе yPred) мәндері арқылы баға функциясын есептеуімізге болады. Енді градиенттік түсудің басты бөлігіне көшейік. Градиенттік түсу бойынша:

$J(\theta_0, \theta_1)$ мәні минимумды қабылдамағанша {
$$tempVal_0 = \theta_0 - \alpha(\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}))$$
$$tempVal_1 = \theta_1 - \alpha(\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})x^{(i)})$$
$$\theta_0 = tempVal_0$$
$$\theta_1 = tempVal_1$$
}

Яғни, алдымен θ_0, θ_1 мәндерін есептеп, кейін θ_0, θ_1 мәндерін өзгертуімізге болады. (θ_0, θ_1)-дің жаңа мәндері арқылы $h_\theta(x)$-дің жаңа мәнін есептеп, қайтадан жаңа баға функциясын есептеуімізге болады. Баға функциясы қандайда бір қателік шамасынан кіші болмағанша осы операциялар тізімін жалғастырып орындай береміз. Бұл функцияны шақыру үшін келесі кодты жазу керек.

```
errorRate = 14
learningRate = 0.05
thetaZero, thetaOne =
        gradientDescent(Y, X, learningRate, errorRate)
```

Мұндағы үйрету шамасы 0.05-ке ($learningRate = 0.05$), ал қателік шамасы 14-ке тең ($errorRate = 14$). Енді (θ_0, θ_1)-дің табылған мәндері арқылы әр X дерегі үшін $h_\theta(x)$-тің мәнін есептесек, мынадай түзу сызықты болжам шығады.



2.33-сурет. Сызықтық регрессияда генерацияланған деректерге табылған түзу сызықты болжам функциясы

Толық код мынадай:

```
import numpy as np
import random
import matplotlib.pyplot as plt

def h(x, thetaZero, thetaOne):
    return x * thetaOne + thetaZero

def costFunction(yTrue, yPred):

    totalSum = 0
    m = len(yTrue)

    for i in range(m):
        totalSum += (yPred[i] - yTrue[i]) *
                                (yPred[i] - yTrue[i])

    return totalSum / (2 * m)

def gradientDescent(y, x, alpha, errorRate):

    m = len(y)

    thetaZero = random.randint(0, 1e3)
    thetaOne = random.randint(0, 1e3)

    yPred = []
    for i in range(m):
        yPred.append(h(x[i], thetaZero, thetaOne))

    cost = costFunction(y, yPred)

    iteration = 0
    while cost > errorRate:
        print(iteration, cost, thetaZero, thetaOne)

        sumZero = 0
        sumOne = 0
        for i in range(m):
            sumZero += (yPred[i] - y[i])
            sumOne += (yPred[i] - y[i]) * x[i]
        sumZero =  sumZero / m
        sumOne =  sumOne / m

        tempZero = thetaZero - alpha * sumZero
        tempOne = thetaOne - alpha * sumOne

        thetaZero = tempZero
        thetaOne = tempOne

        for i in range(m):
            yPred[i] = h(x[i], thetaZero, thetaOne)

        cost = costFunction(y, yPred)
        iteration += 1

    return thetaZero, thetaOne

trueThetaZero = 10
trueThetaOne = 5
X = np.linspace(0, 10, 100)
Y = []
for i in range(len(X)):
    Y.append(h(X[i], trueThetaZero, trueThetaOne) +
                abs(random.gauss(20,5)))

errorRate = 14
learningRate = 0.05
thetaZero, thetaOne = gradientDescent(Y, X,
                learningRate, errorRate)

Y_pred = []
for i in range(len(X)):
    Y_pred.append(h(X[i], thetaZero, thetaOne))

plt.scatter(X, Y, c='b')
plt.scatter(X, Y_pred, c='r')
plt.show()
```

Осы жоғардағы кодты өздеріңіз жазып шыққандарыңыз дұрыс болады.

2.2. Қорытынды

Бұл бөлімдегі ең басты мәліметтер осы бөлікте топтастырылған.

- Сызықтық регрессия X айнымалысы мен Y нәтижесі арасындағы сызықтық қатынасты атайды.
- Сызықтық регрессияда болжам функциясы мына жолмен анықталады: $h_\theta(x) = \theta_0 + \theta_1 x$.
- Сызықтық регрессияда баға функциясы былайша анықталады: $J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$.
- Сызықтық регрессияда дұрыс θ_0 және θ_0 мәндерін табу үшін баға функциясын барынша 0-ге жақындату керек ($minimize_{\theta_0,\theta_1} J(\theta_0, \theta_1)$).
- $J(\theta_0, \theta_1)$ функциясын барынша кішірейту үшін $J(\theta_0, \theta_1)$ функциясының экстрим нүктесін табу керек.
- Сызықтық регрессияда градиенттік түсу арқылы дұрыс θ_0 және θ_1 мәндерін мына жолмен табамыз:

$J(\theta_0, \theta_1)$ мәні минимум қабылдамағанша {

$$tempVal_0 = \theta_0 - \alpha(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)}))$$
$$tempVal_1 = \theta_1 - \alpha(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)}))x^{(i)}$$
$$\theta_0 = tempVal_0$$
$$\theta_1 = tempVal_1$$

}

Сұрақтар жауабы

Сұрақ	Дұрыс жауап
1	1, 6, 8
2	1
3	2
4	1
5	2
6	3
7	2
8	3
9	1
10	2
11	4.8
12	3
13	3
14	1.3
15	$\begin{pmatrix} -15 & 11 \\ 18 & 26 \end{pmatrix}$
16	$\begin{pmatrix} 88 & 27 \\ 22 & 55 \end{pmatrix}$
17	$\begin{pmatrix} 6 \\ 16 \\ 21 \end{pmatrix}$
18	$\begin{pmatrix} 20 \\ 26 \\ 48 \end{pmatrix}$
19	$\begin{pmatrix} 44 & 44 & 21 \\ 39 & 54 & 21 \\ 109 & 21 & 52 \end{pmatrix}$
20	$\begin{pmatrix} 4 & 8 \\ 7 & 15 \end{pmatrix}$
21	3x2
22	2x2
23	$\begin{pmatrix} 1 & 3 \\ 4 & 7 \\ 7 & 0 \end{pmatrix}$
24	$\begin{pmatrix} 3 & 9 \\ 5 & 5 \end{pmatrix}$
25	28
26	57
27	—
28	1
29	1
30	0.99
31	1
32	0.000024095
33	15.87
34	2

Сұрақ	Дұрыс жауап	Сұрақ	Дұрыс жауап	Сұрақ	Дұрыс жауап
35	1	51	1	67	1
36	0.99	52	4	68	3
37	4	53	2	69	1
38	4	54	—	70	1,2
39	50.4085	55	2	71	4
40	—	56	2,4	72	4
41	4 модель, 91%	57	2,3	73	3
42	1	58	1	74	1
43	2	59	1	75	1
44	2	60	1	76	1
45	2	61	3	77	2
46	2	62	1	78	3
47	1	63	1	79	1,2
48	3	64	1	80	1
49	3	65	2	81	2,3
50	3	66	2	82	3

Сұрақ	Дұрыс жауап
83	2,4
84	2

Айнымалылар және олардың анықтамасы

- **Кіріспе**
  - $E(Experience)$ — Машинаны үйрету моделінің үйрену тәжірибесі
  - $P(Performance)$ — Машинаны үйрету моделінің нәтижесі
  - $T(Task)$ — Машинаны үйрету моделы шешетін тапсырма
  - X, x — Машинаны үйрету моделыне берілген дерек
  - Y, y — деректің шындық мәні
  - Y', y' — берілген X, x дерегіне машинаны үйрету моделы жасаған болжам
- **Сызықтық регрессия**
  - $x^{(i)}$ — i-інші дерек
  - $y^{(i)}$ — i-інші дерегіне арналған шындық мәні
  - m — Үйренуге қажетті дерек саны
  - h — Болжам функциясы
  - n — Қасиеттер саны
  - θ — Үйрену параметрлері
  - θ_i — Үйрену параметрінің i-інші мүшесі
  - $h_\theta(x)$ — θ үйрену параметрімен анықталатын h болжам функциясының x дерегіндегі болжам мәні
  - $\sum_{i=1}^{m}$ — i мәні 1-ден m-ге дейін жүгіргендегі қандай да бір амалдың қосындысы. Мысалы: $\sum_{i=1}^{m} i^2$ - 1-ден m-ге дейінгі барлық сандардың квадраттарының қосындысы
  - $J(\theta)$ — баға функциясы
  - $\frac{\partial}{\partial \theta_i} J(\theta)$ — баға функциясының θ_i параметрі бойынша туындысы
  - α — үйрену жылдамдығын анықтайтын шама

- **Сызықтық алгебра**
  - n — Матрицадағы қатар саны
  - m — Матрицадағы баған саны
  - $A^{n \times m}$ — n қатардан және m бағаннан тұратын матрицасы
  - $A_{i,j}$ — A матрицасының i-ші жолы мен j-ші бағанында тұрған элемент
  - i — бірлік матрицасы. Бастапқы диагональдағы элементтердің бәрі 1-ге тең, ал қалған элементтер 0-ге тең A^{-1} - A матрицасына кері матрица
  - A^T — A матрицасының транспозициясы
- **Бірнеше айнымалы сызықтық регрессия**
  - μ — x_i мәндерінің арифметикалық ортасы ($\mu = \frac{\sum_{i=1}^{n} x_i}{n}$)
  - s — x_i мәндерінің аралық ұзындығы. s мәнін табу үшін мына формула қолданылады: $s = max_{i=1}^{n} x_i - min_{i=1}^{n} x_i + 1$, яғни x_i мәндерінің үлкен және кіші шамаларының айырмашылығы
- **Логистикалық регрессия**
  - $g(\theta^T x)$ — g функциясы сигмоид немесе логистикалық функция деп аталады және былайша анықталады: $g(z) = \frac{1}{1+e^{-z}}$.
  - $p(y = 1|x; \theta)$ функциясы берілген x ақпараты мен θ параметрі үшін y болу ықтималдылығын көрсетеді.
  - $cost$ — Баға функциясын анықтайтын шама
  - log — Екілік логарифм
  - $h_\theta^{(i)}(x)$ — i-інші класс үшін i-інші классқа тиесілі екенін анықтайтын ықтималдық
- **Регуляризация**
  - λ — регуляризация параметрі
- **Нейрондық жүйелер**
  - g — функциясын активация функциясы немесе сызықтық емес функция
  - $a_i^{(j)}$ — j-ші қабаттағы i-ші нейронның активациялық функциядан шыққан мәні
  - $\Theta^{(j)}$ — j-ші қабат пен $(j+1)$-ші қабатты қосатын параметрлер матрицасы
  - $h_\theta(x)$ — нейрондық жүйедегі болжам

- NOT — логикалық терістеу амалы
- OR — логикалық немесе амалы
- AND — логикалық және амалы

- **Нейрондық жүйедегі оқыту**
  - L — нейрондық жүйедегі барлық қабат саны
  - s_l — L-ші қабаттағы барлық нейрон саны
  - K — сыртқы қабаттағы нейрон саны немесе класс саны
  - $\Theta_{ij}^{(l)}$ — нейрондық жүйенің L-ші қабатындағы i-ші нейрон мен $(l+1)$-ші қабатындағы j-ші нейронды қосатын оқыту параметрі
  - $\delta_j^{(l)}$ — L-ші қабаттағы j-ші нейрон үшін қателік мәні. Ол мән $a_j^{(l)}$ мәнінің шындық мәнінен қаншалықты алыс/жақын екенін көрсететін қателік мәні
  - $\odot$ — амалы екі вектордың соған сәйкес элементтерін көбейту арқылы құрылған вектор
  - ϵ — өте кіші шама ($\epsilon \approx 10^{-4}$).

- **Машинаны үйретудегі кеңестер**
  - $(x_{test}^{(i)}, y_{test}^{(i)})$ — i-ші тест дерегі мен сол деректің шындық мәні
  - $(x_{train}^{(i)}, y_{train}^{(i)})$ — i-ші оқыту дерегі мен сол деректің шындық мәні
  - $(x_{val}^{(i)}, y_{val}^{(i)})$ — i-ші валидация дерегі мен сол деректің шындық мәні
  - m_{train} — үйретуге керекті барлық дерек саны
  - m_{test} — тестілеуге керекті барлық дерек саны
  - m_{val} — валидацияға керекті барлық дерек саны
  - $J_{test}(\theta)$ — тестілеу дерегіне арналған баға функциясы
  - $J_{train}(\theta)$ — үйрету дерегіне арналған баға функциясы
  - $J_{val}(\theta)$ — валидация дерегіне арналған баға функциясы
  - d — полиномдық функцияның жоғары дәрежесі

- **Машинаны үйрету жүйесін құру**
  - F_1 метрикасы — дәлдік пен шолу арасындағы гармоникалық ортасы

- **Тіреу векторының машиналары**
  - C — λ параметрі сияқты A мен B-ның арасындағы ымыралы шешім табуға арналған параметр
  - $p^{(i)}$ — $x^{(i)}$ векторынан θ векторына түскен проекцияның ұзындығы
  - f_i — x_i қасиетін алмастыратын жаңа қасиет
  - l_i — f_i қасиетін табуға арналған бағдар нүктесі
  - σ^2 орташа квадраттық ауытқу
- **k Жақын Көршілер**
  - $dis^{(i)}$ — k жақын көршілер алгоритміндегі ара қашықтық
- **Шешім ағашы**
  - i — қоспа деңгейі
  - j — барлық класс саны
  - p_i — i классына тиесілі деректер санының барлық деректер санына қатынасы
- **Қарапайым Байес**
  - $P(A|B)$ — шартты түрдегі ықтималдық, яғни егер B жағдайы орындалса, онда A жағдайының орындалу ықтималдығы
  - $P(B|A)$ — шамасы, егер A жағдайы орындалса, онда B жағдайының орындалу ықтималдығы
  - $P(A)$ және $P(B)$ — A және B жағдайларының бір-бірінен тәуелсіз орындалу ықтималдықтары
- **Кластеризация есебі**
  - K — кластер саны
  - $c^{(i)}$ — $x^{(i)}$ дерегі үшін ең жақын кластер индексі (1 мен K аралығы)
  - μ_k — K-ші кластерге тиесілі деректердің арифметикалық ортасы
- **Өлшемдікті қысқарту**
  - $u^{(1)}, u^{(2)}, \ldots, u^{(k)}$ — K векторлы кеңістік
  - Σ — коварианттық матрица
  - z — x өлшемдігі қысқарған жаңа координаталар
  - λ — меншік мәні

- x — меншік векторы
- A — матрица
- det — матрица детерминанты
- i — бірлік матрицасы
- x_{approx} — өлшемдігі қысқартылған дерек

- **Аномалияны анықтау**
  - $p(x)$ — x дерегі аномалиялық құбылыс екенін анықтайтын ықтималдық
  - ϵ — өте кіші шама
  - $N(\mu, \sigma^2)$ — Гаусстық таралу, мұндағы μ — орта мәні, ал Σ — стандартты ауытқу мәні
- **Ұсыныс жүйелері**
  - $r(i,j)$ — i-ші қолданушы j-ші тауарға баға беруін анықтайтын шама
  - $y(i,j)$ — i-ші қолданушы j-ші тауарға берген бағасы
  - n_u — қолданушылар саны
  - n_m — тауар саны
  - $\Theta^{(j)}$ — j-ші қолданушы үшін параметр векторы
  - $x^{(i)}$ — i-ші қолданушы үшін қасиеттер векторы
  - $m^{(j)}$ — j-ші қолданушы үшін баға берген тауар саны
- **Үлкен деректермен жұмыс**
  - $const_1, const_2$ — тұрақты сандар
  - $iteration$ — итерация саны
  - B — кіші топтағы дерек саны
- **Оптикалық Символды Анықтау мысалы**

Кітаптың толық нұсқасы

Кітаптың толық нұсқасын www.itboo.kz сайтынан сатып алуға болады. Кітап бағасы 10 000 теңге. Сатып алар алдында academica промокодын енгізсеңіз кітапқа 10% жеңілдік аласыздар. Төлем ақы жасағаннан кейін сіздің электронды поштаңызға кітап жіберіледі.

Автор жайлы

Мағжан Қайранбай

Машинаны үйрету саласының ғылым докторы (Малайзияда ғылыми жұмысын қорғаған). 10 жылдан астам "Машинаны үйрету" саласында жұмыс істеп келеді. Мағжан жасаған машинаны үйрету жобалары Оңтүстік Шығыс Азия мемлекеттерінде кеңінен қолданылады.

Кітап дизайнері Мағжан Қайранбай

Куала Лумпур, 2022