Supervised Learning: Ангилал ба Регрессийн Мод /APM/

James J. Cochran
Associate Dean for Research
Rogers-Spivey Research Fellow
The University of Alabama
jcochran@cba.ua.edu
Ulaanbaatar, Mongolia
Thursday, June 28, 2018

Мод ба Дүрэм

Салангид үр дүнг таамаглагч багц хувьсагчдын утгад үндэслэн ангилах, урьдчилан таамаглах зорилгоор модны бүтцийг аргыг ашигладаг.

Гарах үр дүн нь багц дүрэм юм. Жишээ нь:

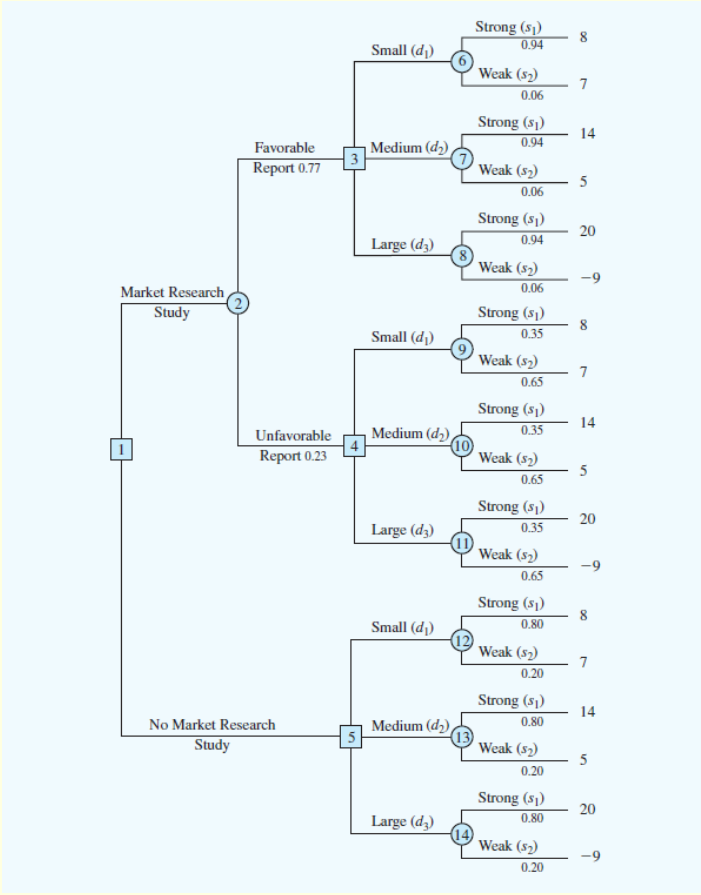
Зорилго:"Кредит карт авна" эсвэл “Авахгүй" гэдгийг тэмдэглэнэ.

Ийм дүрэм байж болно “IF (Income > 92.5) AND (Education < 1.5) AND (Family <= 2.5) THEN Class = 0 (nonacceptor).

Мод ба Дүрэм

Рекурсив хуваалт, АРМ, эсвэл Шийдвэрийн мод (энэ нь маш буруу)

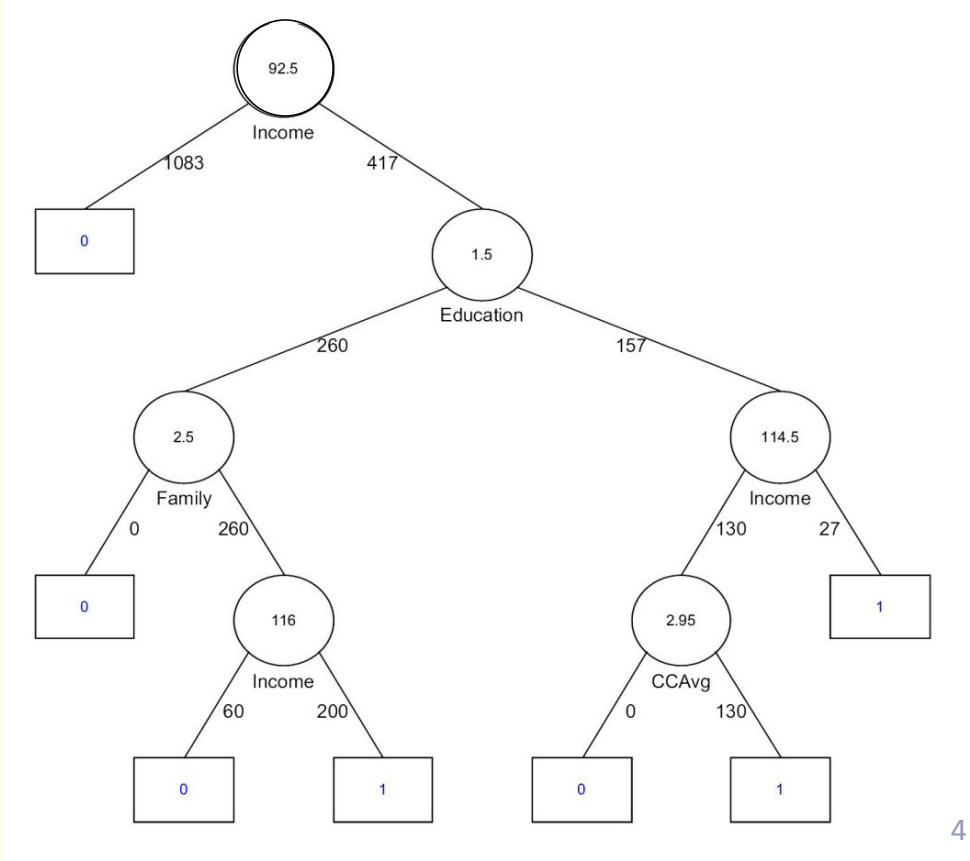
Энэ бол *жинхэнэ* шийдвэрийн мод юм:



Мод ба Дүрэм

АРМ шинжилгээний дүрмийг модны диаграмаар дүрсэлдэг.

АРМ–ийн чухал онцлог нь үр дүнг *маш хялбар* гаргадаг.



Мод ба Дүрэм- Гол үзэл баримтлал

Рекурсив хуваалт: Том бүлгийг(эх цэг) харилцан хамааралгүй хоёр бүлэг болон хамрагдсан бүх бүлэг (хүүхэд цэг)-т дахин дахин хуваадаг бөгөөд ингэснээр шинэ бүлэг доторх хамгийн төстэй байх болон хамгийн олон хувилбарыг бий болгодог.

Ангиллын мод: Зорилтот чанарын (нэрлэсэн болон дэс түвшин) хувьсагчийн рекурсив хуваагдал.

Регрессийн мод: Зорилтот тоон (интервал эсвэл харьцааны түвшин) хувьсагчийн рекурсив хуваагдал

Мод ба Дүрэм- Гол үзэл баримтлал

Эх цэг: бүх түүврийг агуулсан цэг

Хуваах: Урьдчилан таамаглах хувьсагчийн утгуудад үндэслэн хоёр болон түүнээс дээш дэд бүлгүүдэд өгөгдлийг хуваадаг.

Салбар: Хуваагдлын нэг чиглэл

Дэд мод: Хуваагдал ба дараагийн бүх мөчир, цэгний нэг чиглэл.

Эх цэг: Хуваагдалд орох сургалтын тоон мэдээлэл

Мод ба Дүрэм- Гол үзэл баримтлал

Хүүхэд цэг: Эх цэгний хуваагдлаас үүсэх өгөгдөл буюу сургалтын дэд бүлэг.

Эцсийн цэг (Навч): Хуваагдахгүй цэг.

Тайрах: Салбар эсвэл дэд модноос салгасан хэсэг.

Өргөн: Эх цэгний боловсруулсан шууд үр удам (хүүхэд цэг).

Гүн: Эх цэгний үйлдвэрлэх дараагийн үеийн тоо.

Рекурсив хуваах алхам

- Таамаглах хувьсагч x_i–ыг сонгоно.
- Xi-ийн утга болох сургалтын өгөгдлийг хоёр (заавал тэнцүү байх албагүй) бүлэгт хуваагдах s_i–ыг сонгох.
- Үр дүнд хүргэх бүлгийн хэр цэвэр байдал эсвэл ижилхэн байдлыг (нэг ангиллын хувь) хэмжих.
- Эхний хуваагдалд цэвэр байдлыг нэмэгдүүлэхийн тулд xi ба si-ийн янз бүрийн хослолуудыг туршиж үзэх.
- Нэмэлт хуваагдалд зориулж үйлдлийг давтах

Таамаглах тоон хувьсагчийг хэрхэн хуваах вэ?

- Нэг таамаглагч хувьсагч сонгох
- Сонгогдсон таамаглагч хувьсагчийн дагуу бүртгэх
- Дараалсан утгуудын эхний хосын дундах цэгүүдийг олох
- Бүртгэсэн цэгүүдийг дундын цэгээс их, дундын цэгүүдээс бага гэж ангилах
- Хуваагдлыг үнэлсний дараа дараалсан утгуудын хоорондох дундын цэгийг үнэлэх

Категорчилсон таамаглагч хувьсагчийг хэрхэн хуваах вэ?

- Категориудыг хуваах боломжит бүх аргуудыг судлах, ж.нь., A, B, C гэсэн хувьсагчдыг 3 аргаар хувааж болно::
  - {A} ба {B, C}.
  - {B} ба{A, C}.
  - {C} ба {A, B}.
- Категорийн тоо өсөхийн хэрээр хуваагдлын тоо эрс нэмэгддэг.

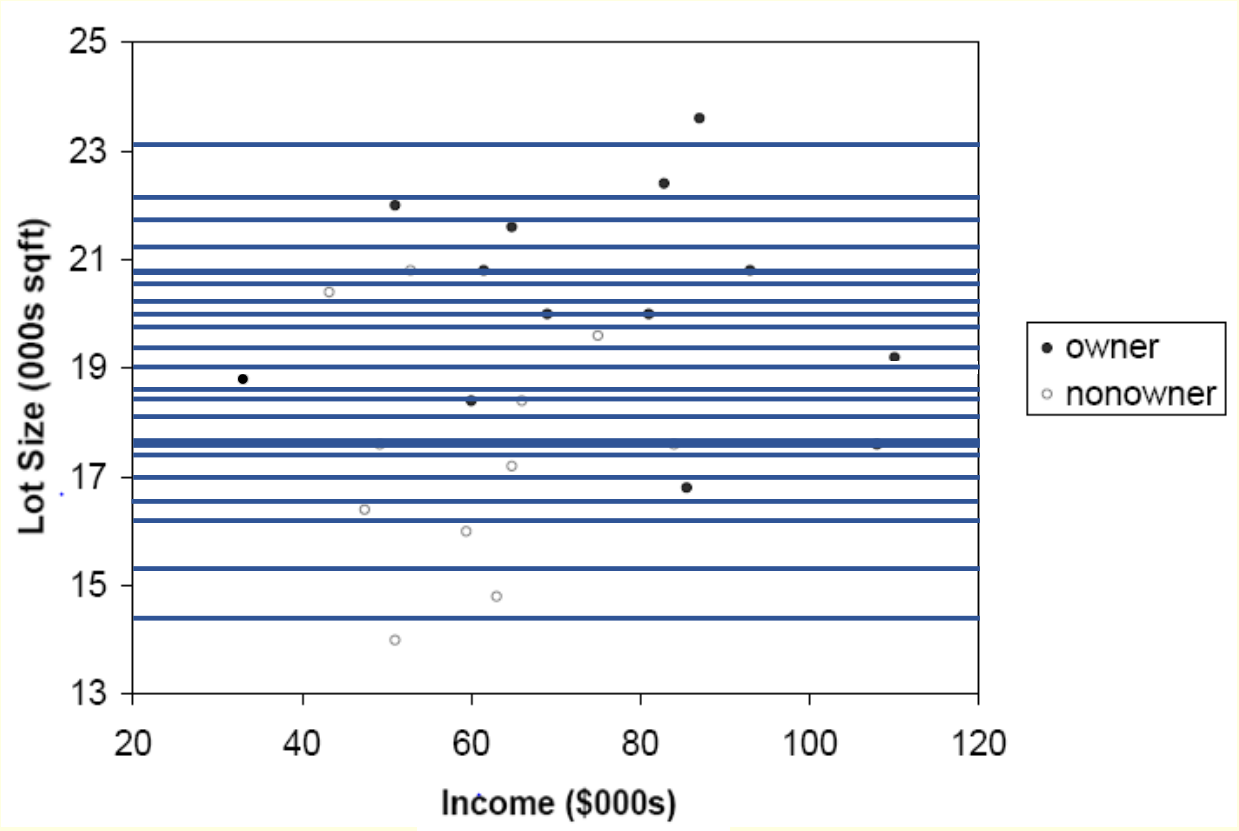
Жишээ нь: Өвс хаддаг машин унах

- Зорилго: Хадуур машин эзэмшигч эсэхээр нь 24 өрхийг ангилах (эзэмшигч эсвэл эзэмшигч биш)
- Таамаглагч нь Орлого, газрын хэмжээ

Орлого	Газрын хэмжээ	Эзэмшлийн хэлбэр
60.0	18.4	эзэмшигч
85.5	16.8	эзэмшигч
64.8	21.6	эзэмшигч
61.5	20.8	эзэмшигч
87.0	23.6	эзэмшигч
110.1	19.2	эзэмшигч
108.0	17.6	эзэмшигч
82.8	22.4	эзэмшигч
69.0	20.0	эзэмшигч
93.0	20.8	эзэмшигч
51.0	22.0	эзэмшигч
81.0	20.0	эзэмшигч
33.0	18.8	эзэмшигч
75.0	19.6	эзэмшигч биш
52.8	20.8	эзэмшигч биш
64.8	17.2	эзэмшигч биш
43.2	20.4	эзэмшигч биш
84.0	17.6	эзэмшигч биш
49.2	17.6	эзэмшигч биш
59.4	16.0	эзэмшигч биш
66.0	18.4	эзэмшигч биш
47.4	16.4	эзэмшигч биш
51.0	14.0	эзэмшигч биш
63.0	14.8	эзэмшигч биш

Жишээ нь: Өвс хаддаг машин унах

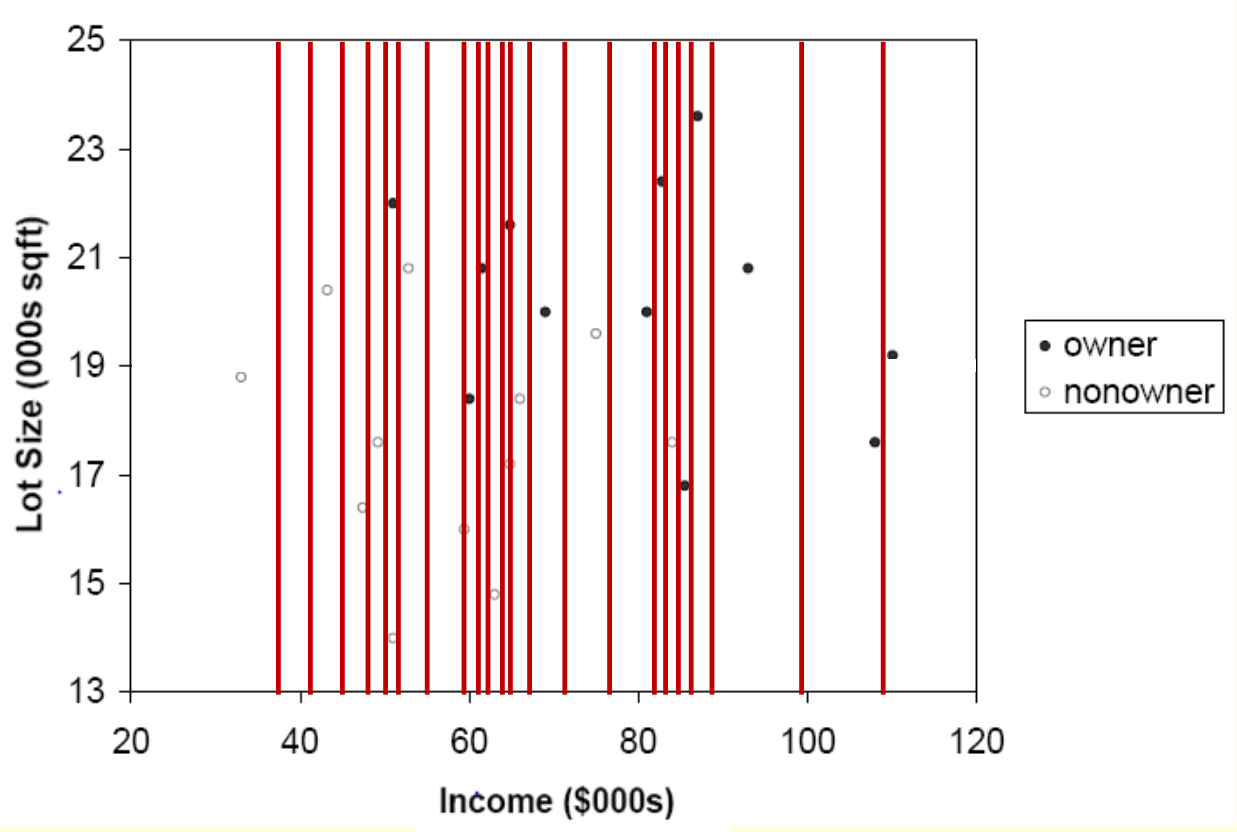
Боломжит хэдэн хуваалтыг авч үзэх ёстой вэ?



Жишээ нь: Өвс хаддаг машин унах

Боломжит хэдэн хуваалтыг авч үзэх ёстой вэ?

Эхний хуваагдлын хамгийн их утга нь m (n - 1)
Дараагийн хуваагдлын хамгийн их утга нь m (n - 1) - 1

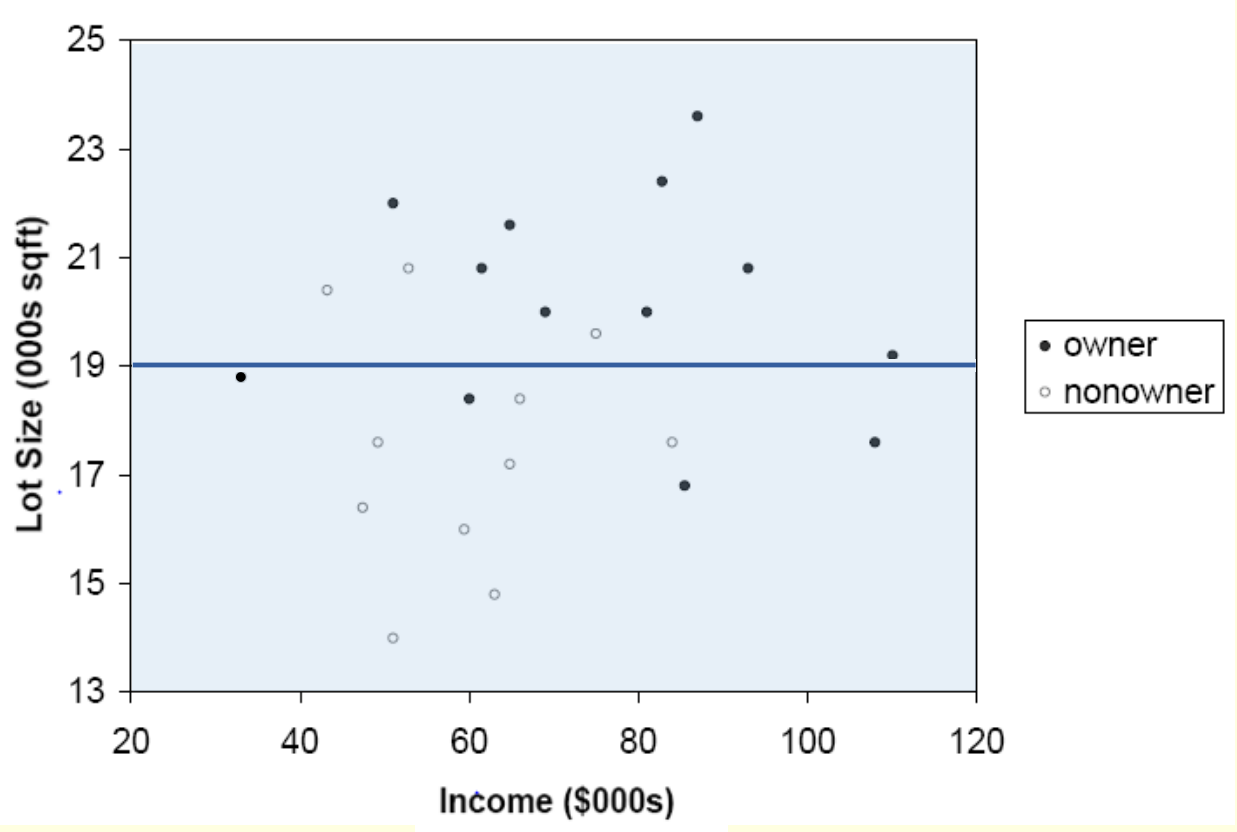


Жишээ нь: Өвс хаддаг машин унах

Эхний хуваагдал:

Газрын хэмжээ

Size = 19,000



Жишээ нь: Өвс хаддаг машин унах

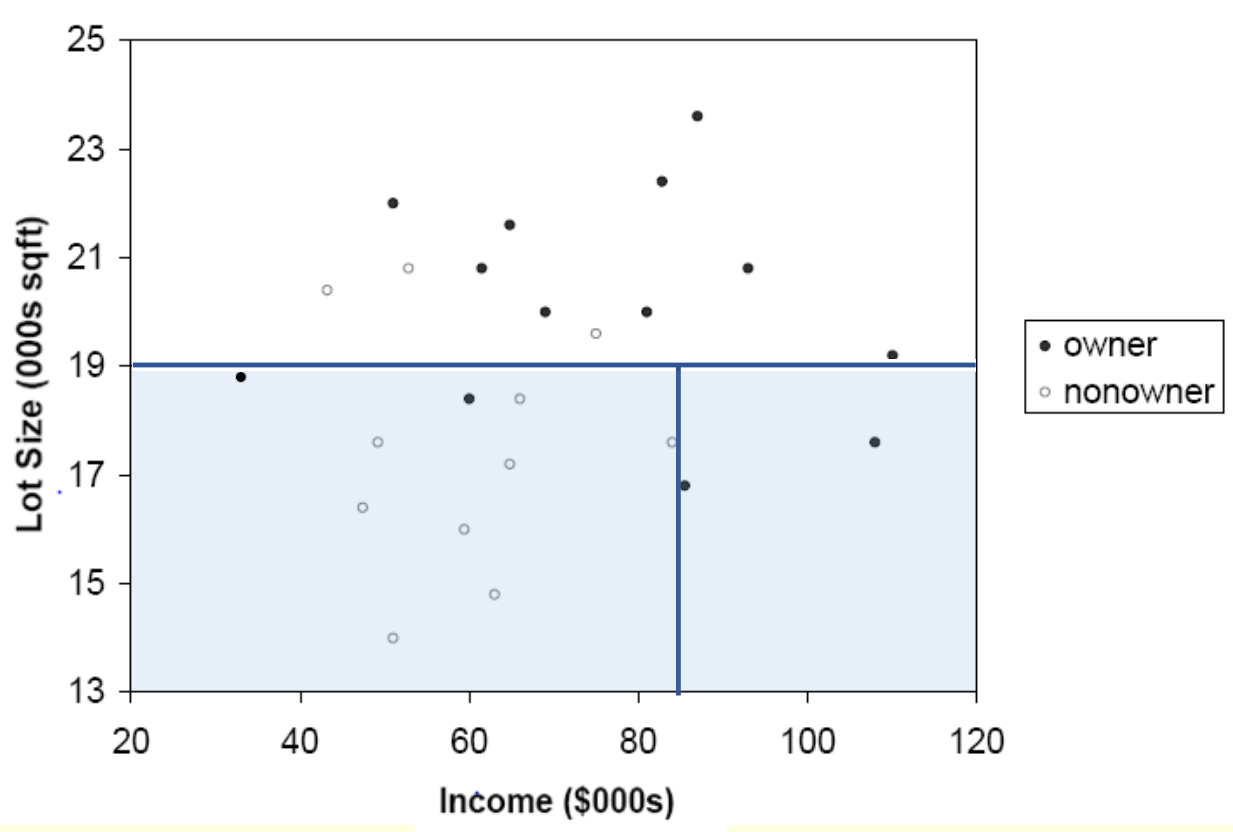
Жишээ нь: Өвс хаддаг машин унах

2 дах хуваагдал :

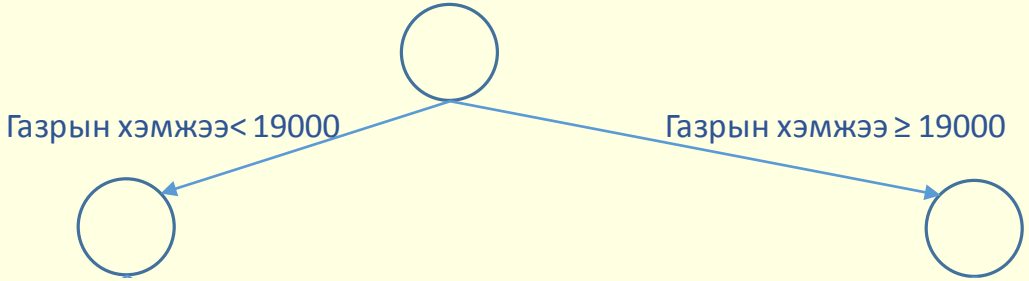
Орлого= $84,000

Газрын хэмжээ

< 19,000



Жишээ нь: Өвс хаддаг машин унах



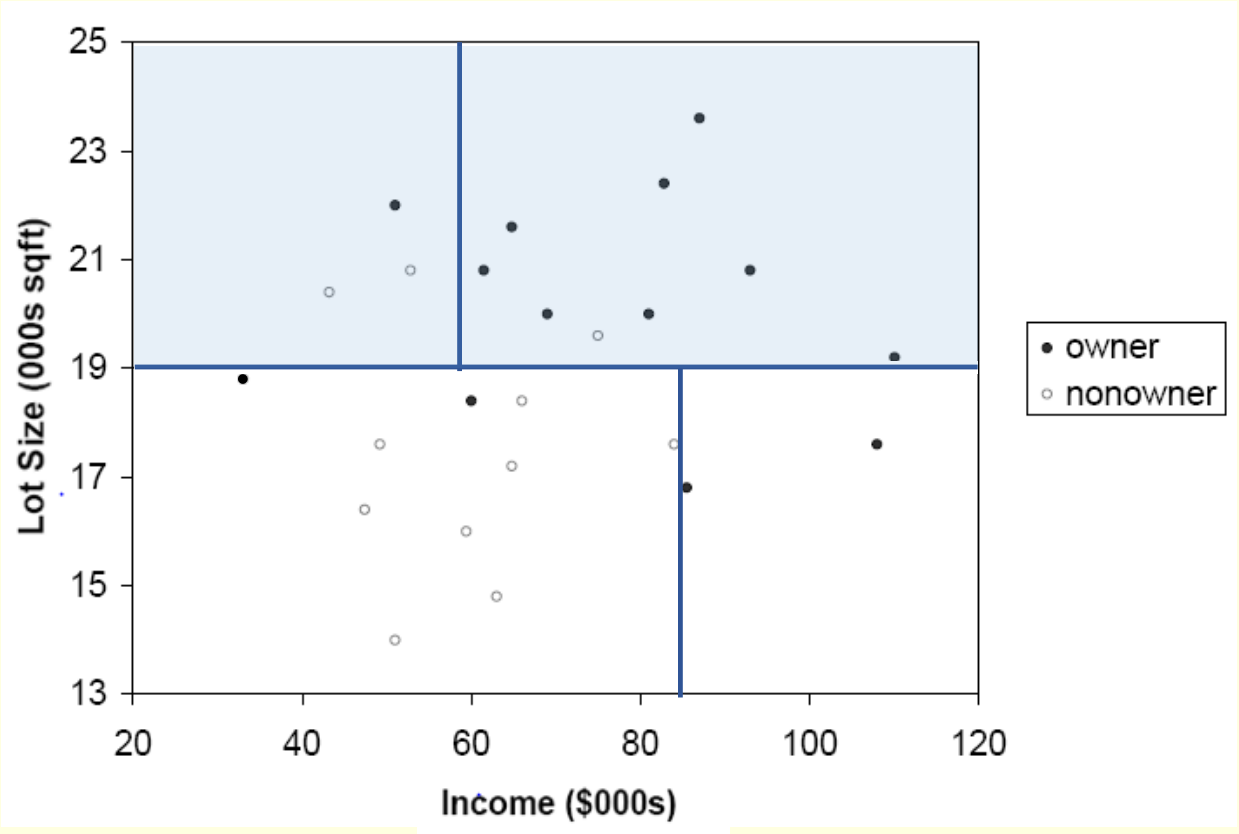
Жишээ нь: Өвс хаддаг машин унах

3 дах хуваагдал:

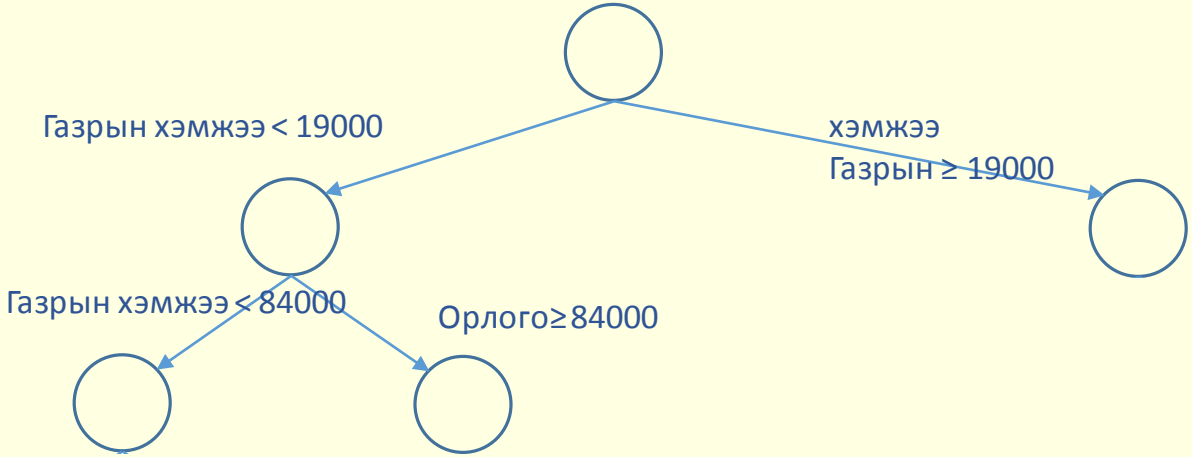
Орлого= $58,000

Газрын хэмжээ

≥ 19,000



Жишээ нь: Өвс хаддаг машин унах



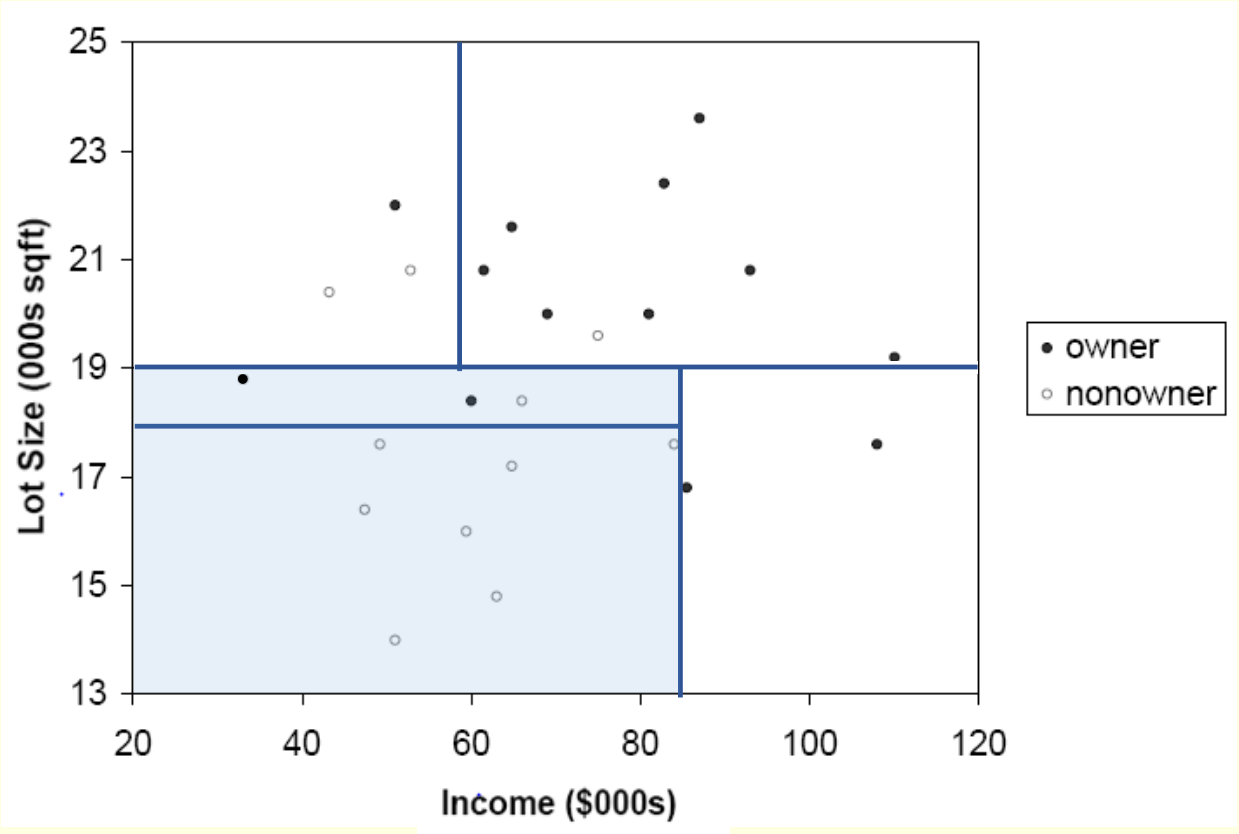
Жишээ нь: Өвс хаддаг машин унах

4 дэх хуваагдал :

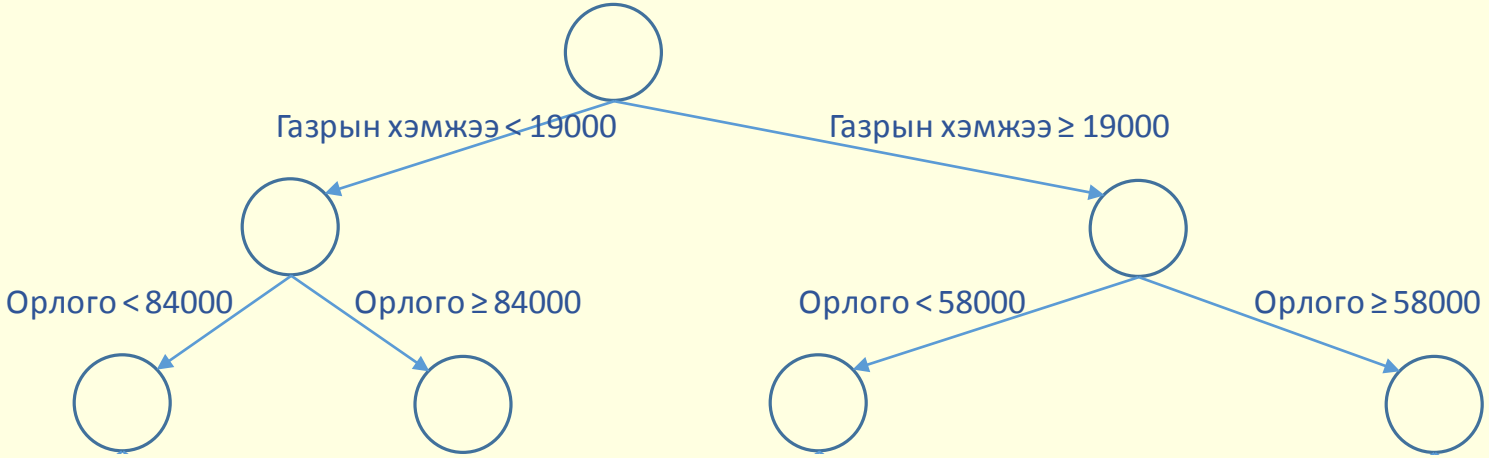
Газрын хэмжээ= 18,000

Газрын хэмжээ < 19,000

Орлого< $84,000



Жишээ нь: Өвс хаддаг машин унах



Жишээ нь: Өвс хаддаг машин унах

5 дах хуваагдал :

Орлого = $62,000

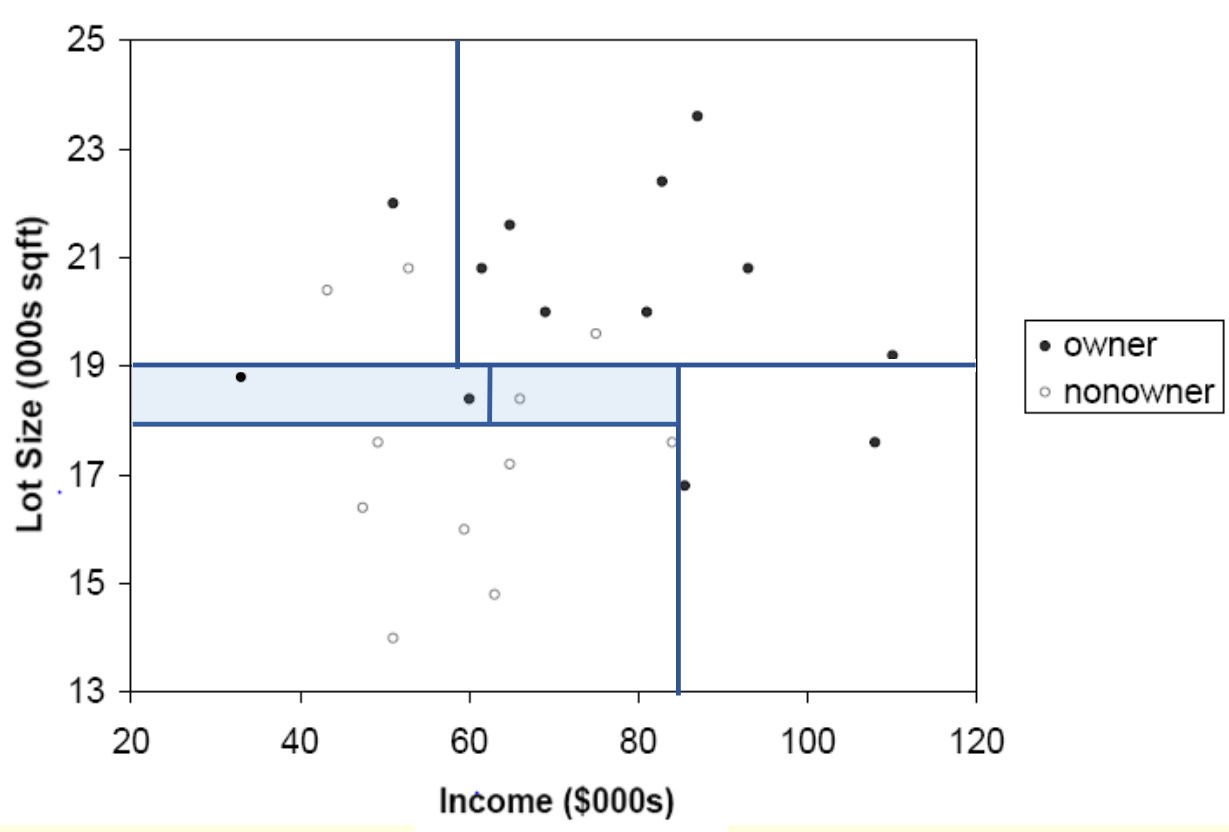
Газрын хэмжээ

< 19,000

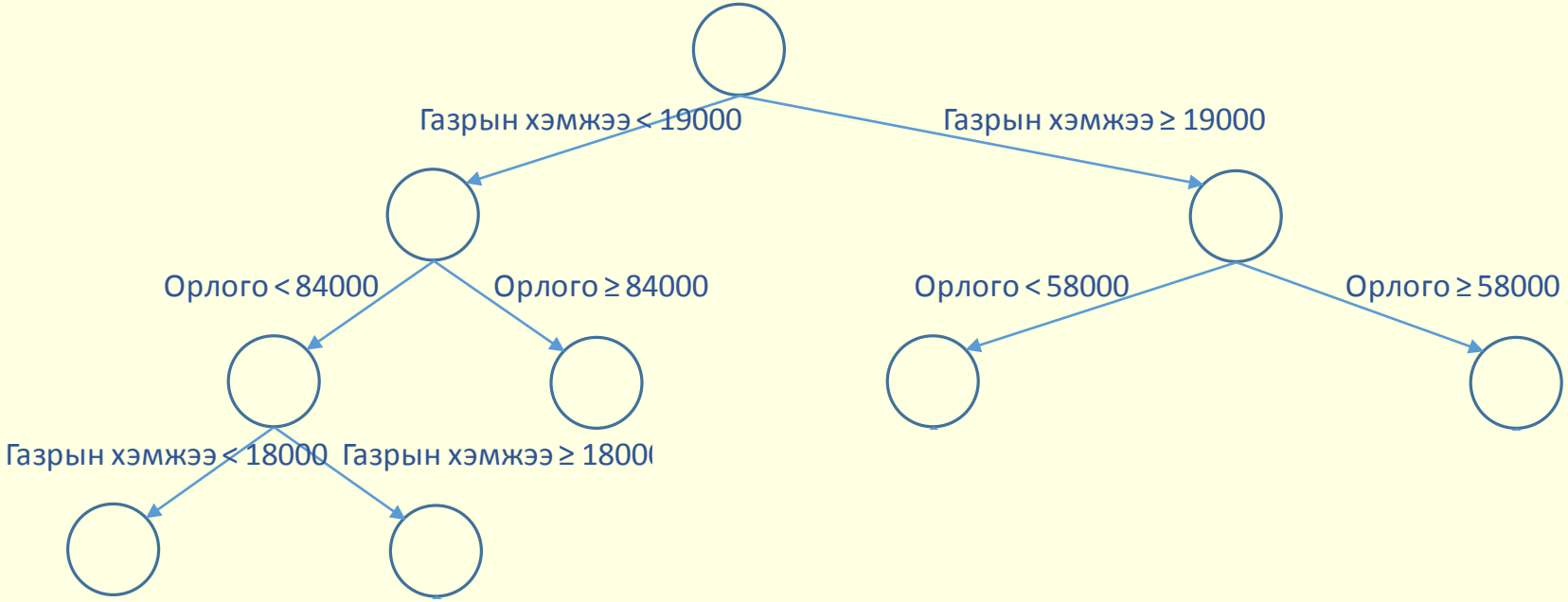
Орлого < $84,000

Газрын хэмжээ

≥ 18,000



Жишээ нь: Өвс хаддаг машин унах



Жишээ нь: Өвс хаддаг машин унах

6 дах хуваагдал :

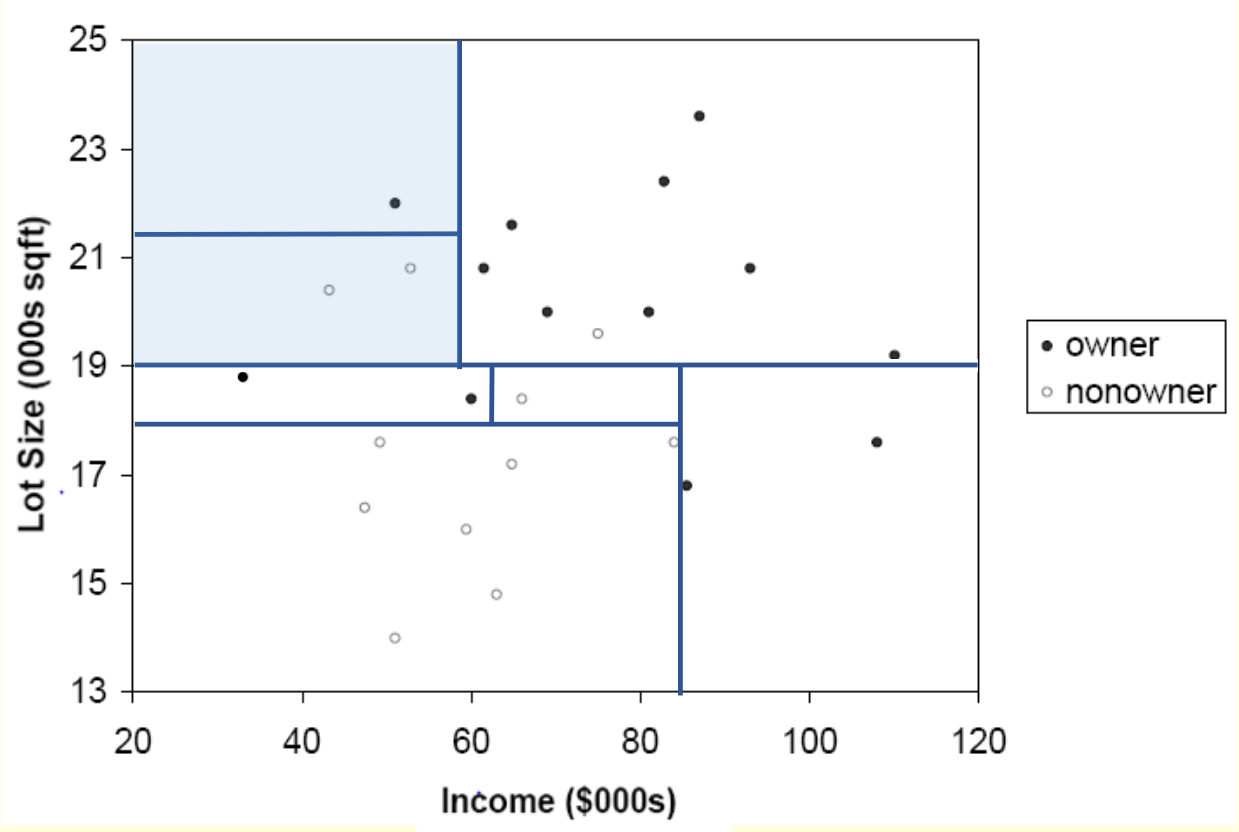
Газрын хэмжээ

= 21,250

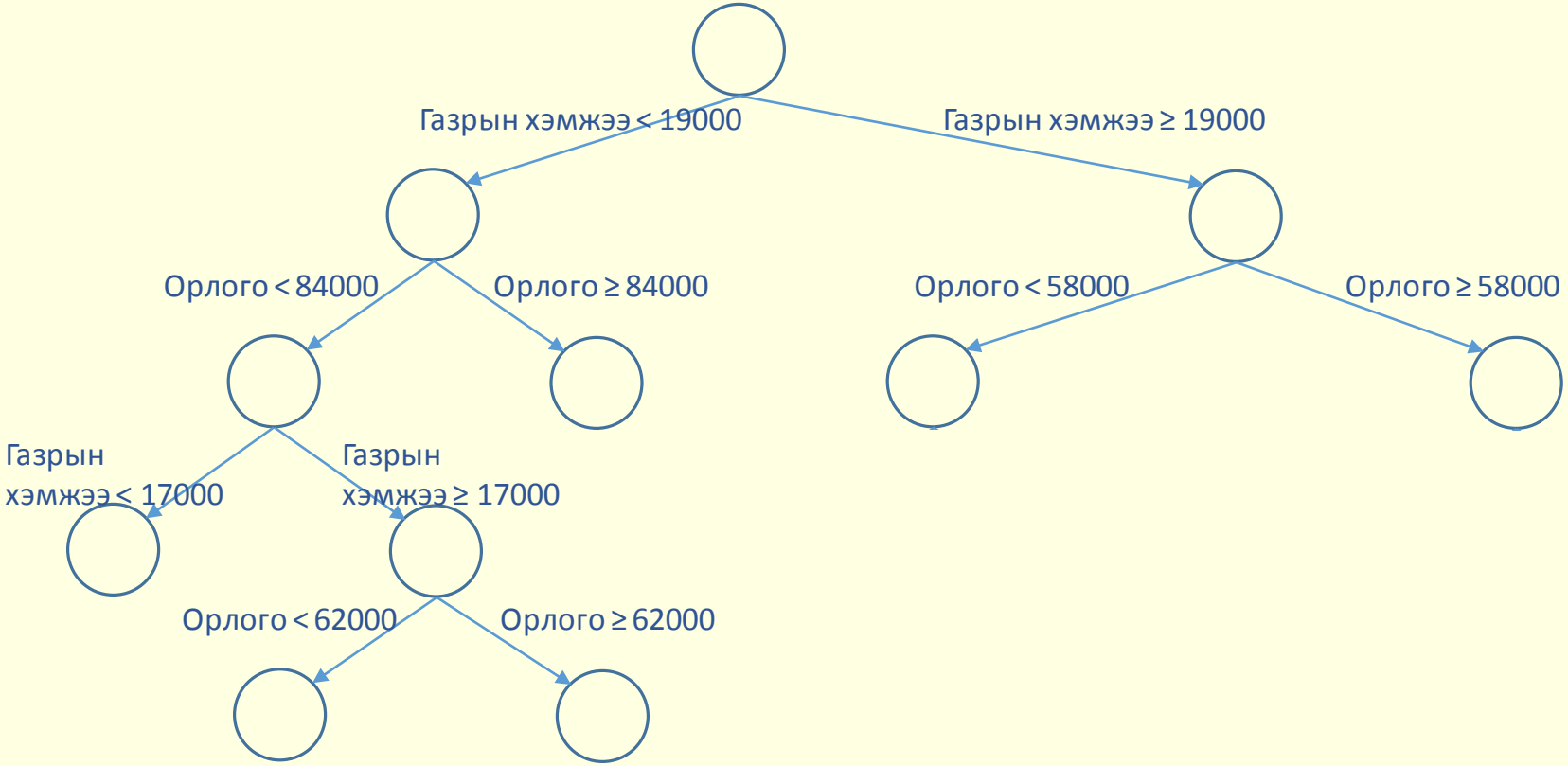
Газрын хэмжээ

≥ 19,000

Орлого < $58,000



Жишээ нь: Өвс хаддаг машин унах



Жишээ нь: Өвс хаддаг машин унах

7 дах хуваагдал :

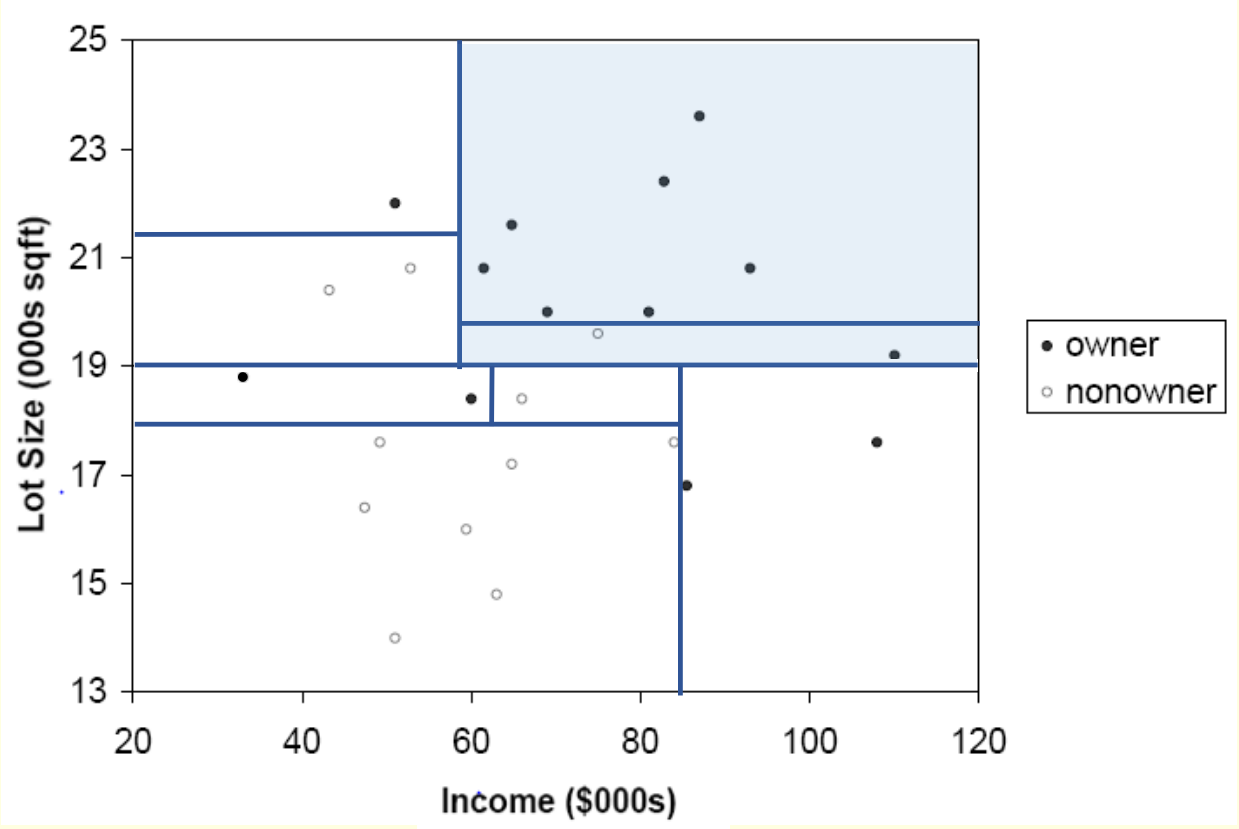
Газрын хэмжээ

= 19,500

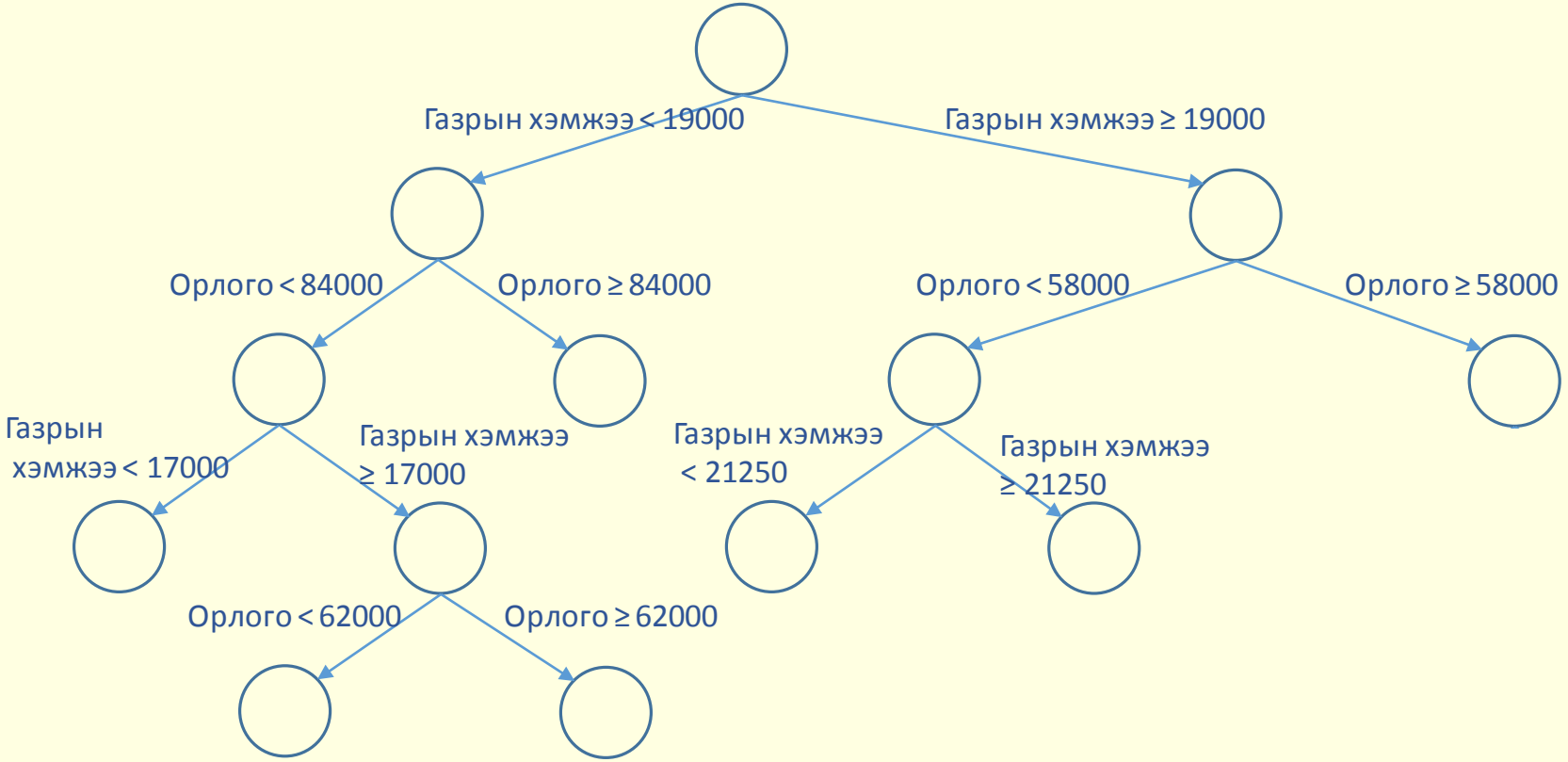
Газрын хэмжээ

≥ 19,000 ба

Орлого≥ $58,000



Жишээ нь: Өвс хаддаг машин унах



Жишээ нь: Өвс хаддаг машин унах

8 дах хуваагадл :

Орлого= 85,000

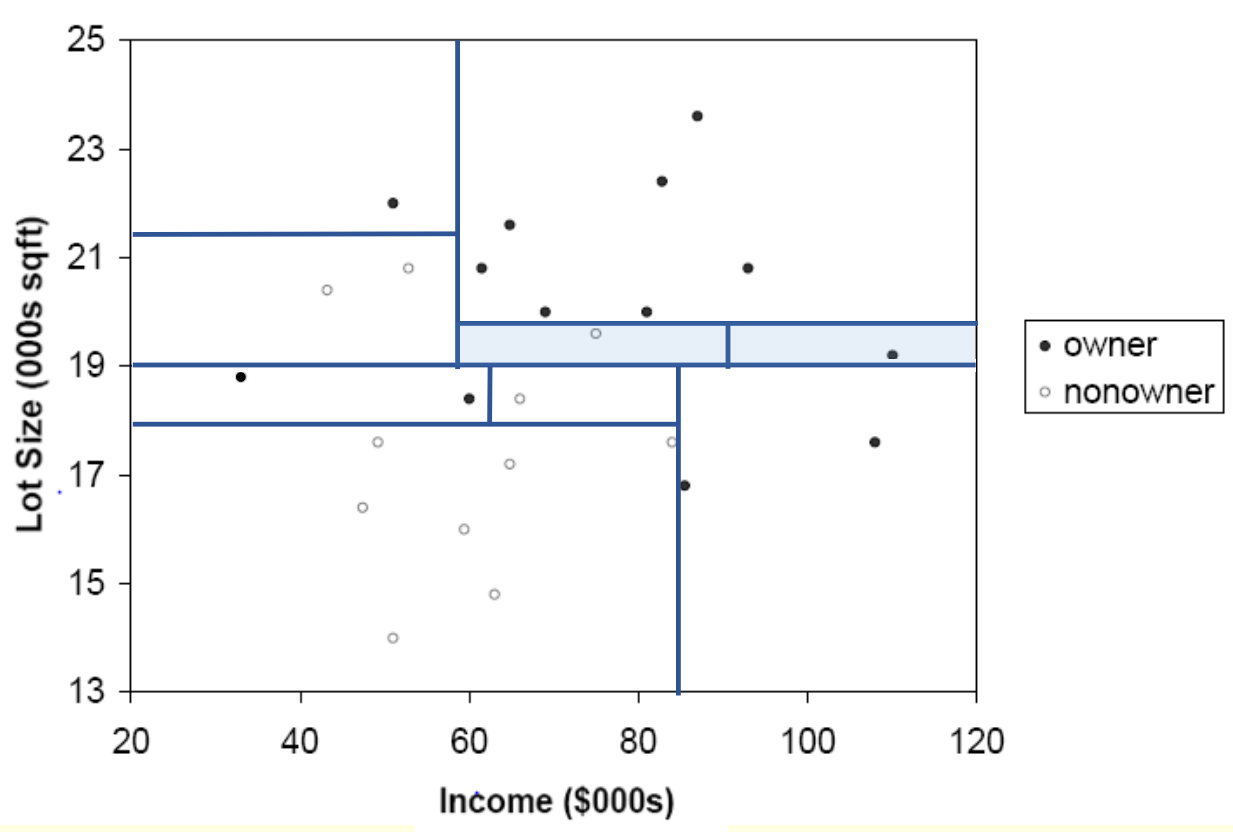
Газрын хэмжээ

≥ 19,000 ба

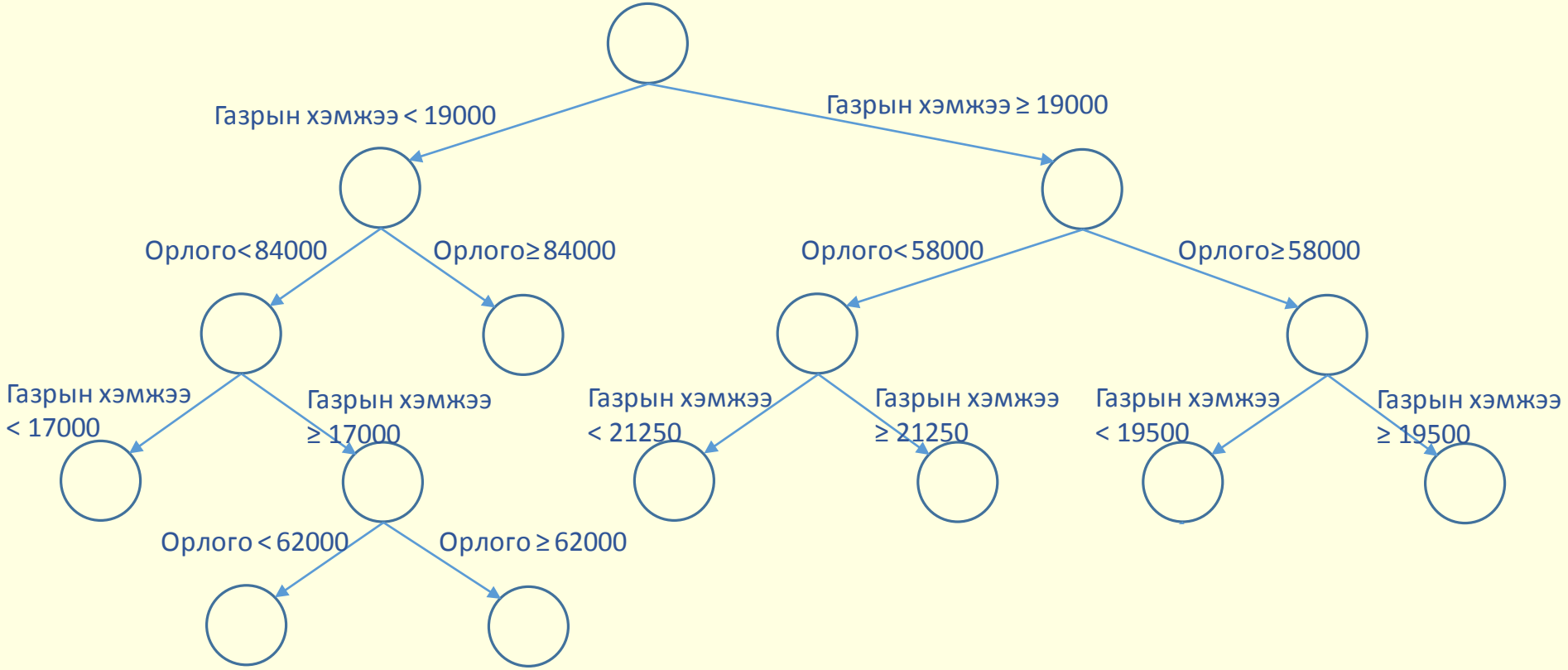
Орлого ≥ $58,000 ба

Газрын хэмжээ

< 19,500



Жишээ нь: Өвс хаддаг машин унах



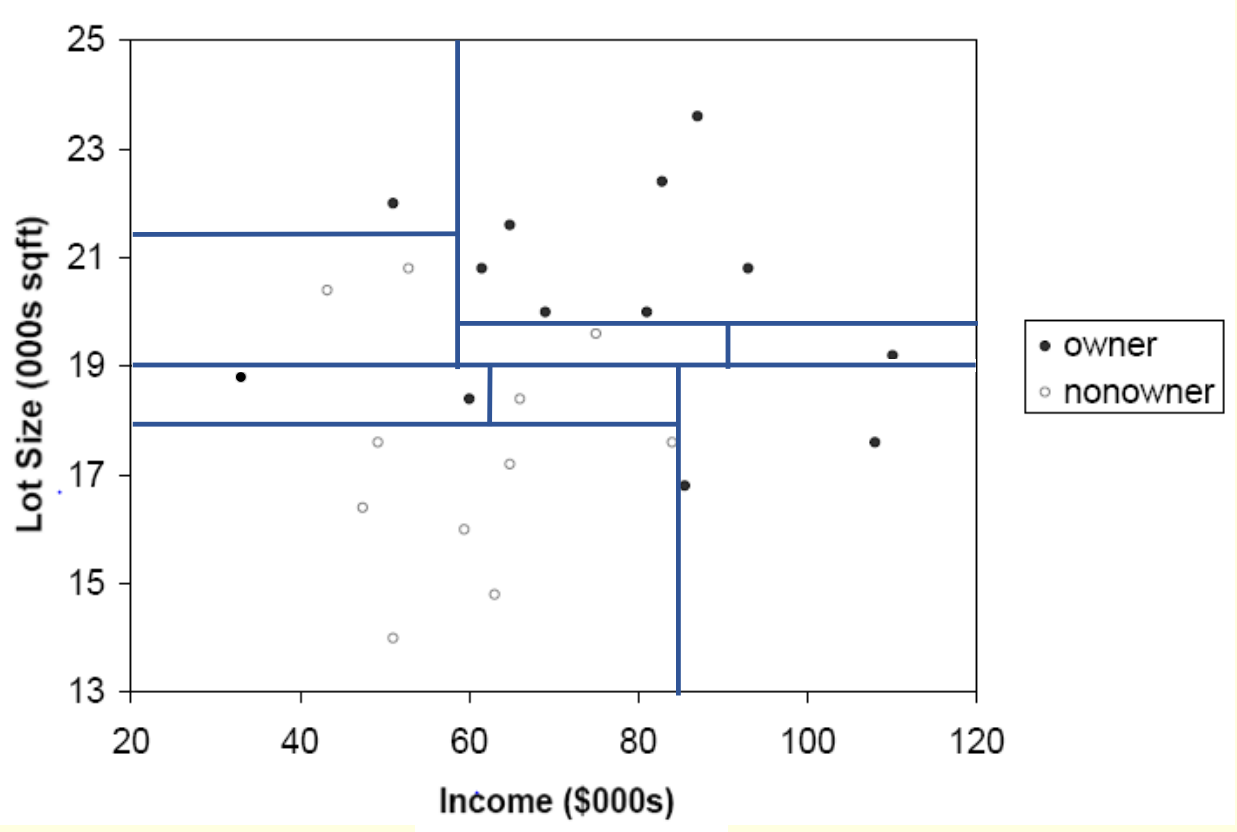
Цааш нь хувааж болох уу? Яагаад Тийм эсвэл Үгүй гэж ?

Жишээ нь: Өвс хаддаг машин унах

Бид яаж хийсэн бэ?

Ажиглалт бүрийг эзэмшигч ба эзэмшигч биш гэж хуваах

Тэгэхээр бид яаж хийсэн бэ?



Бохир байдал– Хуваагдлыг сонгох шалгуур

Хүүхэд цэгний боломжит цэвэр байдал буюу өөрөөр хэлбэл шинэ хүүхэд цэгнууд хоорондын төстэй чанарыг хэрхэн хуваахад хэрэглэдэг.

- Жини индекс (Gini Impurity)
- Энтропи
- Вариацын бууралт

Бохир байдал– Хуваагдлыг сонгох шалгуур

Жини индекс (Gini Impurity) – *m бүртгэлийг агуулж буй A* тэгш өнцөгтийг (цэг) *дараах байдлаар тооцно.*

$$I(A) = 1 - \sum_{k=1}^{m} p_k^2$$

p_k = k ангид хамаарах A тэгш өнцөгтийн (цэг) тохиох хувь

Бохир байдал– Хуваагдлыг сонгох шалгуур

Жини индекс (Gini Impurity)

- Хэрэв дэд хэсгийн хаягийг хуваарийн дагуу санамсаргүйгээр хаягласан бол багцаас хэр олон удаа санамсаргүй байдлаар элементийг сонгохыг хэмжинэ.
- APM– ийн алгоритмийг ашиглав.
- Хамгийн бага утга нь 0 нь төгс цэвэр байдлыг илэрхийлнэ (зөвхөн нэг ангиас ажигласан тохиолдолд)
- Хамгийн их утга нь тухайн асуудлаас хамаарах бөгөөд p_k нь цэгний бүх k-ийн хувьд тэнцүү байхад үүсдэг.

Бохир байдал– Хуваагдлыг сонгох шалгуур

Энтропи- *m бүртгэлийг ангуулж буй A* тэгш өнцөгтийг (цэг) *дараах байдлаар тооцно*

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

where p_k = *k* ангид хамаарах *A* тэгш өнцөгт (цэг) тохиох хувь

Бохир байдал– Хуваагдлыг сонгох шалгуур

Энтропи

- мэдээллийн томьёоноос энтропийн үзэл баримтлалд тулгуурлах
- CHAID (Chi-Square Automated Interaction Detection) алгоритм ашиглах
- Хамгийн бага утга 0 нь төгс цэвэр байдалд тохионо (цэг нь зөвхөн нэг ангиас авсан ажиглалтыг агуулдаг).
- p_k нь цэгэнд байгаа бүх k-тэй тэнцүү үед Log2 (m) -ийн хамгийн их утга агуулна.

Бохир байдал– Хуваагдлыг сонгох шалгуур

Энтропи

- Gain мэдэлээл

$$GAIN(A) = Entropy(A) - \sum_{k=1}^{m} Entropy(A_k) \frac{n_k}{n} log_2 \left(\frac{n_k}{n} \right)$$

A = эх цэг

A_k = k дах хүүхэд цэг

Бохир байдал– Хуваагдлыг сонгох шалгуур

Энтропи

- Gain харьцаа

$$GAINRatio(A) = \frac{GAIN(A)}{\sum_{k=1}^{m} \frac{n_k}{n} log_2\left(\frac{n_k}{n}\right)}$$

n = эх цэгний тохиолдлын тоо

n_k = хүүхэд цэг k – тохиолдлын тоо $k = 1,\ldots,m$

Бохир байдал– Хуваагдлыг сонгох шалгуур

χ^2 (Chi-Square) - *K* ангийг агуулах *A* тэгш өнцөгтийн хувьд

$$\chi^2 = \sum_{k=1}^{K} \frac{\left(e_k - f_k\right)^2}{e_k}$$

e_k = эх цэгэнд үндэслэсэн k ангийн давтамж

f_k = k ангийн бодит давтамж

Бохир байдал– Хуваагдлыг сонгох шалгуур

χ^2 (Chi-Square)

- χ^2 тархалтын үзэл баримтлалд суурилсан
- CHAID (Chi-Square Automated Interaction Detection) алгоритм ашиглана
- Эхийн цэгний к ангийн давтамж нь хүүхэд цэгэнд байгаа бүх k-ийн хувьд бодит давтамжтай тэнцэх үед хамгийн бага утга буюу 0 утга авна.
- Хамгийн их утга нь тухайн асуудлаас хамаардаг боловч төгс цэвэр байдалд тохиолддог (эхийн цэгн дах хүүхдийн цэг нь зөвхөн нэг ангиас ажиглагдана)

Бохир байдал– Хуваагдлыг сонгох шалгуур

Вариацын бууралт - тэгш өнцөгт А (цэг) нь зорилтот тоон хувьсагч Y

$$VR(A) = \sum_{i \in S} \left(y_i - \frac{\sum_{i \in S} y_i}{|S| - 1} \right)^2 - \sum_{k=1}^{K} \left[\frac{|S|}{|S_k|} \sum_{i \in S_k} \left(y_i - \frac{\sum_{i \in S_k} y_i}{|S_k| - 1} \right)^2 \right]$$

S нь эх цэгний ажиглалтын багц индекс юм , S_k нь хүүхэд цэг ны ажиглалтын багц индекс юм.

Бохир байдал– Хуваагдлыг сонгох шалгуур

Вариацын бууралт

- Зорилтот тоон хувьсагч у-ийн хамт АРМ алгоритмд ашиглагдана.
- Зорилтот тоон хувьсагчийн утга у-ын ялгаврын квадрат нь хүүхэд цэгн дах хамгийн их нь хамгийн их байвал хамгийн бага утга авна.
- Төгс цэвэр байдалд хамгийн их утга авна (хүүхэд цэгний бүх ажиглалтаар зорилтот тоон хувьсагчийн тогтмол утга).

Бохир байдал– Хуваагдлыг сонгох шалгуур

F-тест - *A* тэгш өнцөгт (цэг) ба зорилтот тоон утга *y*

$$F(A) = \frac{(|S|-1)\sum_{i\in S}\left(y_i - \frac{\sum_{i\in S} y_i}{|S|}\right)^2}{\sum_{k=1}^{K}(|S_k|-1)\sum_{i\in S_k}\left(y_i - \frac{\sum_{i\in S_k} y_i}{|S_k|}\right)^2}$$

S нь эх цэгний ажиглалтын багц индекс юм ,

S_k нь хүүхэд цэг ны ажиглалтын багц индекс юм.

Бохир байдал– Хуваагдлыг сонгох шалгуур

F-тест

- Зорилтот тоон хувьсагч y-ийн хамт хэд хэдэн алгоритмд ашиглагдана.
- Зорилтот тоон хувьсагчийн ялгаврын квадрат нь хүүхэд цэг бүрт хамгийн их байхад хамгийн бага утга авна.
- Төгс цэвэр байдалд хамгийн их утга авна (хүүхэд цэгний бүх ажиглалтаар зорилтот тоон хувьсах утгын тогтмол утга).

Бохир байдал– Хуваагдлыг сонгох шалгуур

Бохирдлын хэмжүүр сонгогдсон тохиолдолд

Бүх бохирдлын хэмжүүрийг авах (тэгш өнцөгт бүрийн жигнэсэн дундаж).

- Дараалсан үе шат бүрт бүх хувьсагчийн хувьд боломжит бүх хуваагдалтай энэ хэмжүүрийг харьцуулна.
- Бохир байдлыг хамгийн бага байлгах хуваагдлыг сонгох
- Сонгосон хуваагдлын цэгүүд нь модны дараагийн цэг болно.

Модны бүтцийг тайлбарлах

Модыг бий болгосны дараа

- Хуваагдлын цэгүүд нь модны цэг болдог(төвд байрлах хуваагдлын утгатай тойргууд).
- Тэгш өнцөгт нь навчыг төлөөлдөг (сүүлийн цэгүүд, цаашид хуваагдахгүй, ангиллын утга тэмдэглэсэн).
- цэг хоорондох шугамын тоо нь тохиолдлын тоог илэрхийлдэг.
- Дүрмийг ойлгохын тулд модыг доош нь уншина. Жишээ нь: газрын хэмжээ <19 ба орлого> 84.75 бол анги= эзэмшигч. Дүрмүүд нь нэг нэгэндээ багтсан байна (жишээ нь: харилцан хамаарал).

Модны бүтцийг тайлбарлах

Навч цэгний хаягыг дараах байдлаар тодорхойлно.

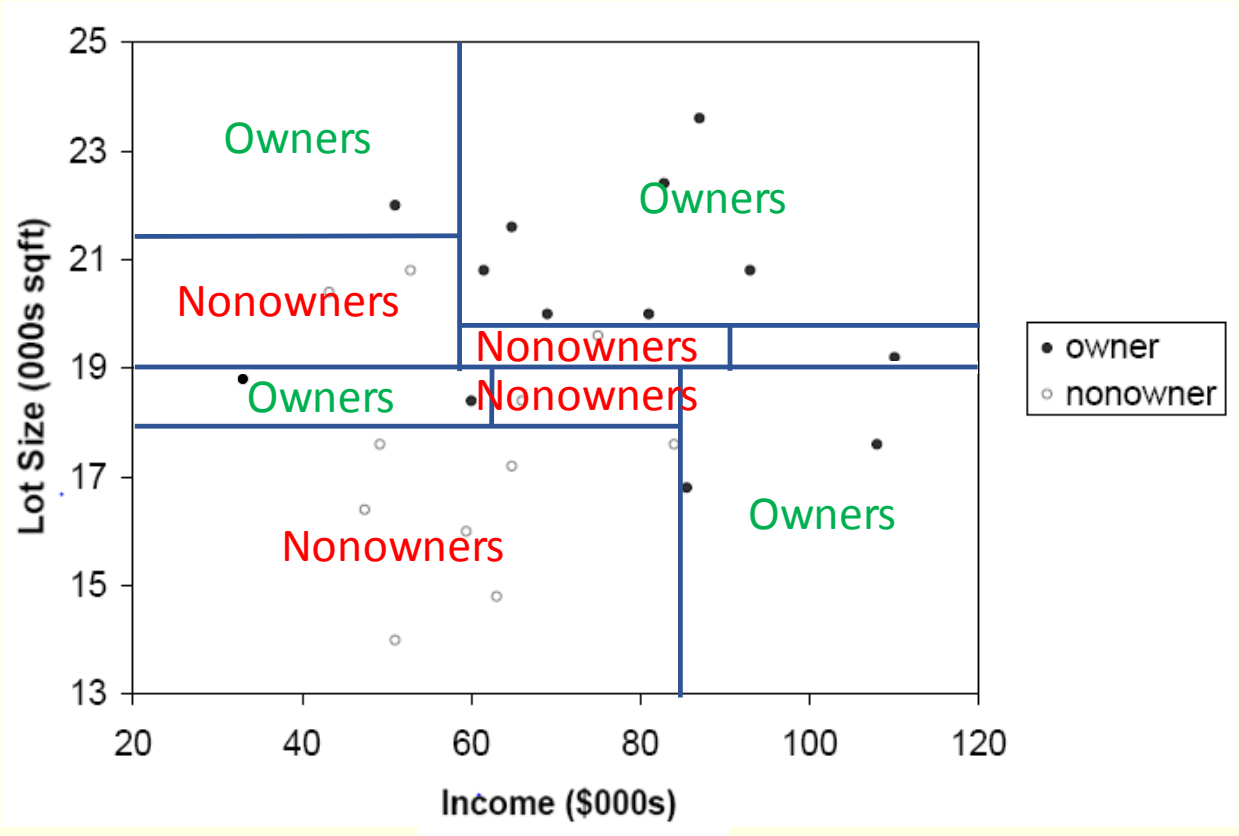
Навч цэгний хаяг бүр бүртгэлийн "санал" болон тасрах утга хоёроор тодорхойлно.

- Сургалтын өгөгдлөөс навч цэгний бүртгэл нь бүрдэнэ.
- Тогтсон хуваах утга = 0.5 бол навчны цэгний хаяг нь ангийн ихэнх байна гэсэн үг
- Хуваах утга = 0.75 байвал 75%-ийн ихэнхийг буюу "1“ гэж бүртгэлтэй навч "1“-р цэг гэж хаяг өгнө.

Жишээ нь: Өвс хаддаг машин унах

Дахин хэлэхэд бид үүнийг яаж хийсэн бэ?

Ажиглалт бүр нь эзэмшигч, үл эзэмшигч гэж цэвэр хуваагдсан байна.



Модыг дуусгах

Хэт тохирох

- Загвар нь эх олонлогийн харилцаанаас илүүтэйгээр санамсаргүй алдаа буюу дуу чимээ гаргавал хэт тохирох тохиолдоно. Жишээ нь: сургалтын өгөгдөл зан үйлийн онцлогоос хамаарна)
  - Эх олонлогийн шинж чанараас илүү түүврийн мэдээлэлд онцлогоо тусгах.
  - Түүврийн бус алдаанаас болж төөрөгдүүлэх
  - Загвар нь түүвэрт таарах боловч шинэ өгөгдөлд таарахгүй байх).

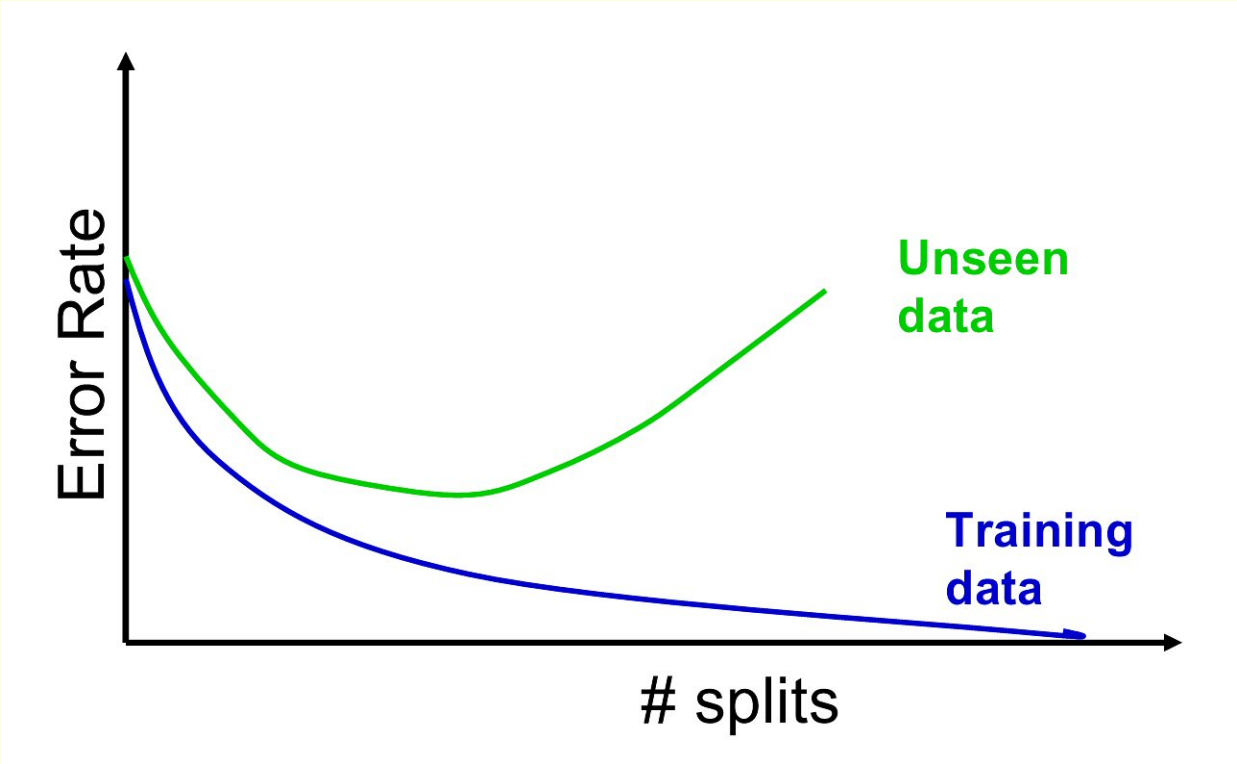
Модыг дуусгах

Хэт тохирох

- Үйл явцын байгалийн төгсгөл нь навч бүрт 100% цэвэр байдалтай байна.
- Энэ нь өгөгдөлд хэт тохирох бөгөөд сургалтын өгөгдөлд нийцэж буй онцлог шинж чанарыг харуулсан дохио өгдөг.
- Баталгаажуулалтын өгөгдлийн алдааны түвшин нь модны тодорхой хуваагдлыг өнгөрсөний дараагаар нэмэгдэж эхэлдэг.
- Хэт тохирох нь түүврийн өгөгдлөөс таамаглалын нарийвчлалыг муутгадаг.

Модыг дуусгах

Бүтэн модны нийтлэг алдааны түвшин:



Модыг дуусгах

Эрсдлийг дараах байдлаар бууруулдаг.

- Эх цэг хамгийн өргөн нь байх.
- Үндсэн цэг нь хамгийн гүн байх.
- Эх цэгг хамгийн бага хэмжээгээр хуваах
- Хуваагдлаас үүсэх хүүхэд цэг нь хамгийн жижиг байх.
- Сүүлийн цэг нь хамгийн бага хэмжээтэй байх.
- Сүүлийн цэг нь хамгийн их хэмжээтэй байх.
- Мод тайрах, Жишээ нь: захын мөчрүүдийг тайрч модыг хялбаршуулах.

Модыг дуусгах

Тайрах

- Модыг буцаж тайрах.
- Модны загвараас гүйцэтгэл доройтсон хуваагдлыг тайрах.
- Навчуудыг тайрч дараалсан жижиг мод үүсгэх.
- Тайралт бүрт аль болох олон мод үүсгэх
- үе шат бүрт хамгийн сайн модыг сонгохын тулд зардлын нарийн төвөгтэй байдал эсвэл баталгаажуулалтын алдаа зэрэг шалгуурыг ашиглах

Модыг дуусгах

Тайрахдаа зардлын төвөгтэй байдлын аргыг ашиглах

$$CCT = Err\left(T\right) + \alpha L\left(T\right)$$

CCT = модны зардлын төвөгтэй байдал
Err(*T*) = буруу ангилсан бүртгэлийн харьцаа
α= модны хэмжээнд хавсаргасан торгуулийн хүчин зүйл (хэрэглэгчээр тохируулсан)
$\alpha L(T)$ = навч цэгний тоо

Модыг дуусгах

Тайралтад Зардлын төвөгтэй байдлын /ЗТБ/ аргыг ашиглах

- Өгөгдсөн модноос ЗТБ хамгийн багыг нь сонгох
- Хэмжээ өөр мод бүрт үүнийг хийх.

Рекурсив хуваалтын тайлбар

Тайрахдаа Баталгаажуулалтын алдааг ашиглах

- Хамгийн бага алдааны мод - баталгаажуулалтын өгөгдөлд хамгийн бага алдааны түвшинтэй байна
- Хамгийн сайн тайрсан мод - Зарим стандартад нийцсэн хамгийн жижиг мод (ихэвчлэн зарим зайнд - магадгүй нэг стандарт алдаа - Хамгийн бага алдаа)

Рекурсив хуваалтын тайлбар

Давуу талууд

- Тодорхойлох, тайлбарлах, урьдчилан таамаглахад ашиглаж болно.
- Хэрэглэх, ойлгоход хялбар.
- Тайлбарлах, хэрэгжүүлэхэд хялбар дүрмийг гаргадаг.
- Хувьсагчийн сонголт ба бууралт автоматаар хийгддэг.
- Статистикийн загварын таамаглал шаарддаггүй (өөрөөр хэлбэл, энэ нь параметрийн бус).
- Байхгүй мэдээллийг хялбар зохицуулах боломжтой.

Рекурсив хуваалтын тайлбар

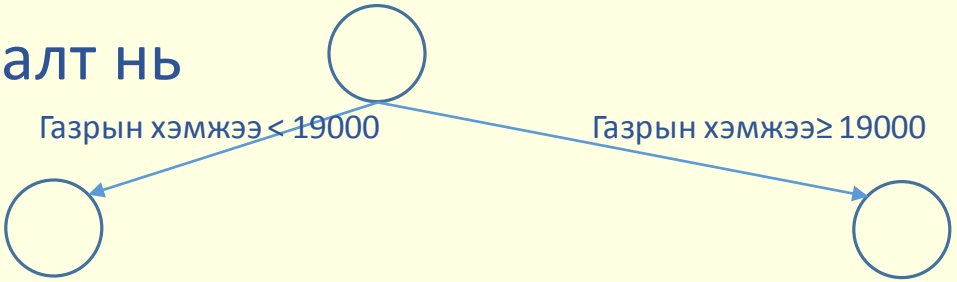
Сул талууд

- Өгөгдлийн бүтцийг хэвтээ эсвэл босоо тэнхлэгт хуваахад сайн ажилладаггүй (тэнхлэгтэй зэрэгцээ).
- Өгөгдлийн бүтцийг шугаман хуваахад сайн таардаггүй.
- Процесс нэг удаадаа зөвхөн нэг хувьсагчинд тохирох тул хуваагдал хоорондын хувьсагчийг танихгүй
- Статистикийн дүгнэлт хийх арга биш.
- Түүврийн их хэмжээг шаарддаг (жижиг цөөнхийн ангийг анхаарах).
- Хамгийн оновчтой үр дүн гардаггүй (*хамгийн сайн* мод биш).

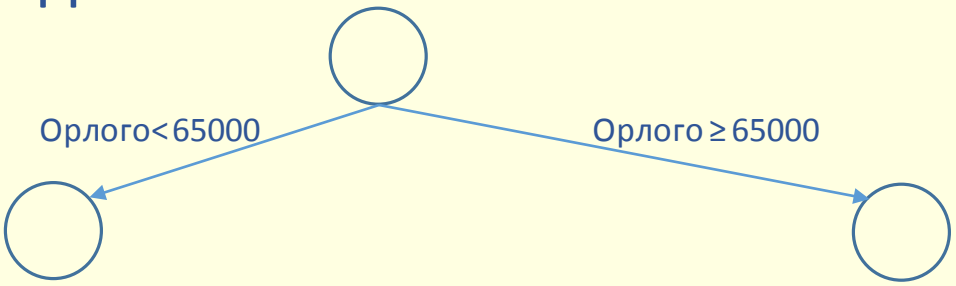
Example: Riding Mowers

Яагаад хамгийн оновчтой үр дүнг гардаггүй вэ?

Учир нь энэ хуваалт нь

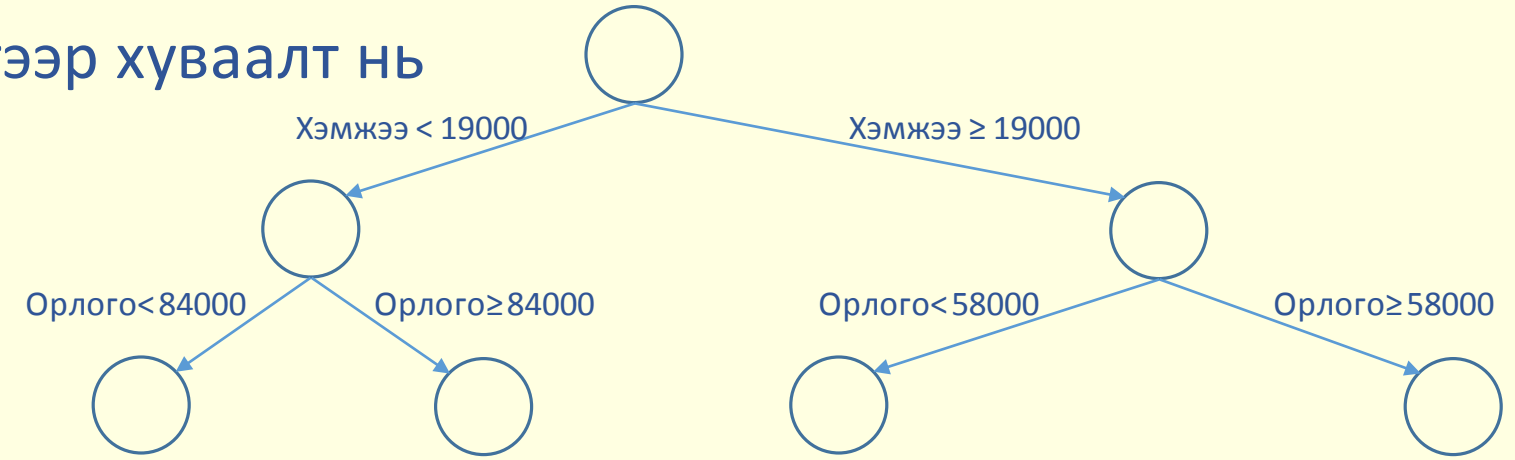


энэ хуваалтаас илүү байж болно

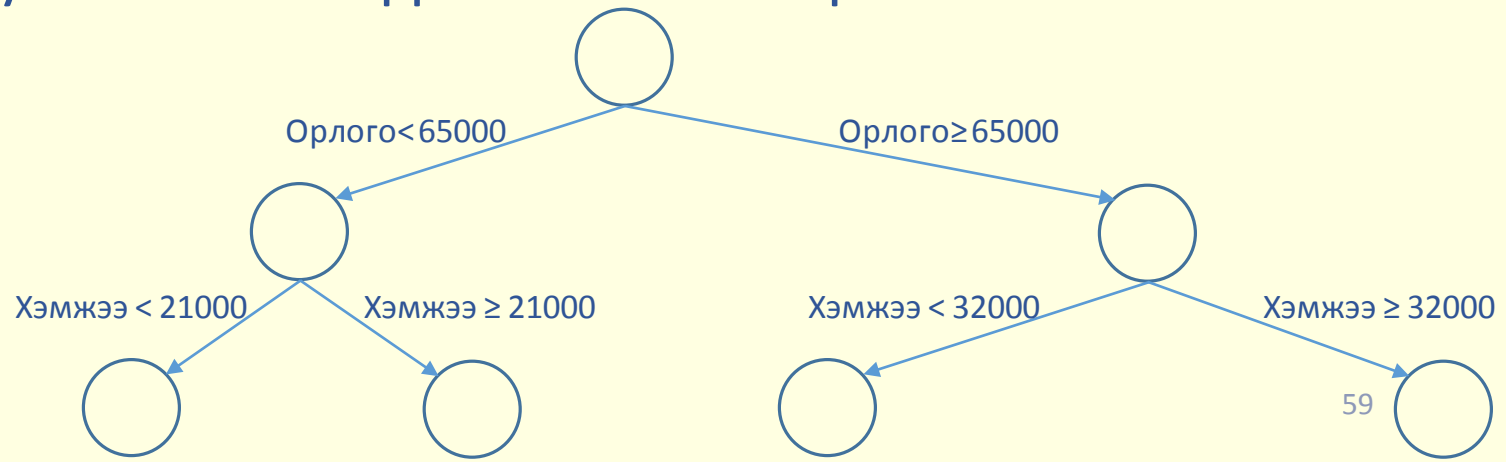


Жишээ нь: Өвс хаддаг машин унах

Гэвч эдгээр хуваалт нь



Эдгээр хуваалтаас илүү байж болохгүй



Олон төрлийн алгоритмууд

Алгоритмууд нь голчлон дараах зүйлсээс хамаарч өөр өөр байдаг. Үүнд:

- Салбарлах/хуваах шалгуур
- Ангилах мод, регрессийн мод, эсвэл хоёуланг ашиглах боломжтой эсэх
- Эх цэгний үйлдвэрлэх боломжтой хүүхэд цэгний тоо
- Overfitting-г хэрхэн шийдвэрлэдэг
- Дутуу өгөгдөлтэй хэрхэн шийдэх вэ

Олон төрлийн алгоритмууд

Хамгийн түгээмэл алгоритмууд

- АХИ(автомат харилцаа илрүүлэгч) – Морган, Сонкүист нарын боловсруулсан алгоритм нь зорилтот тоон хувьсагчтай хоёртын модыг нэмэхэд зориулсан алгоритм юм (Morgan, J.N. and Sonquist, J.A. (1963). Судалгааны өгөгдөлд дүн шинжилгээ хийхэд тулгарах асуудлууд
- *Journal of the American Statistical Association*, 58, 415–434).

Олон төрлийн алгоритмууд

Хамгийн түгээмэл алгоритмууд

- THAID (Тета автомат харилцаа илрүүлэгч) - Morgan, Messenger нарын боловсруулсан алгоритм нь theta шалгуур ашиглан зорилтот ангилсан хувьсагчдад АХИ-аргыг дэлгэрүүлсэн. (Morgan, J.N. and R.C. Messenger. 1973. *THAID: нэрлэсэн хэмжээнд хамааралтай хувьсагчдын дүн шинжилгээнд дараалсан дүн шинжилгээ хийх хөтөлбөр.* Survey Research Center, Institute for Social Research, University of Michigan, Ann Arbor).

Олон төрлийн алгоритмууд

Хамгийн түгээмэл алгоритмууд

- OHAXИ(олон хувьсагчтай автомат харилцаа илрүүлэгч) - Гилло, Шелли нарын боловсруулсан алгоритм нь олон хувьсагчийн зорилтот тоон хувьсагчдад зориулсан АХИ-ийн аргыг дэлгэрүүлсэн. (Gillo, M.W. 1972. OHAXИ, Автоматжуулсан судалгаанд дүн шинжилгээ хийх Honeywell 600 програм. *Behaviorial Science*, 17(2), 251–252; Gillo, M.W. and Shelly, M.W. 1974. Олон хувьсагчийн болон олон хувьсагчтай өгөгдлийн таамагласан загварчлал. *Journal of the American Statistical Association*, 69(347), 646–653.)

Олон төрлийн алгоритмууд

Хамгийн түгээмэл алгоритмууд

- CHAID (chi-square автомат харилцаа илрүүлэгч) - 1980 онд Gordon Kass-ийн диссертацид хэвлэгдсэн C2 шалгуурыг ашигладаг АХИ-ийн дэлгэргүүлсэн алгоритм.

Олон төрлийн алгоритмууд

Хамгийн түгээмэл алгоритмууд

- CLS-1 to CLS-9 (Концепт сургалтын систем 1-9) - Hunt болон хамтрагчдынх нь боловсруулсан шийдвэр гаргах модулийн алгоритмууд нь хиймэл оюун ухааны талаас нь зорилтот ангилсан хувьсагч дээр мод тарих; CLS-1 CLS-8 нь хоёртын модыг бүтээх ба CLS-9 нь олон салбар болгодог (Hunt, E.B., Martin, J., and Stone, P.J. 1966. *Experiments in induction*. New York and London: Academic Press.)

Олон төрлийн алгоритмууд

Хамгийн түгээмэл алгоритмууд

- ТЧСЗХХИ (Туршилтын чуулганы сегментчилэлээр холбоос ба харилцаах илрүүлэх) - Cellard, Labbé, Savitsky нарын боловсруулсан алгоритм бөгөөд зорилтот ангилсан хувьсагчаар хоёртын модыг нэмэх нь (Cellard, J.C., Labbé, B., et Savitsky, G. 1967. Le Programme ELISEE, Présentation et application. *Metra 3* (6), 511–519.)

Олон төрлийн алгоритмууд

Хамгийн түгээмэл алгоритмууд

- ИӨИДШХ (Интерактив өгөгдөлийг илрүүлэх, дүн шинжилгээ хийх) - Press, Рожерс болон Shure боловсруулсан алгоритм нь олон салбарлах боломжтой модыг интерактив аргаар тарих (Press, L.I., Rogers, M.S., and Shure, G.H. 1969. Олон хувьсагчтай өгөгдөлд дүн шинжилгээ хийх интерактив арга. *Behavioral Science*, 14(5), 364–370.)

Олон төрлийн алгоритмууд

Хамгийн түгээмэл алгоритмууд

- ID3 (Давтах Dichotomizer 3) – эхний 3 алгоритмыг Росс Куинлан боловсруулсан (Quinlan, J.R. 1986) Шийдвэрийн модны танилцуулга, 1 (1), 81-106).
- зөвхөн салангид функц-д болох;
- бүрэн бус бүртгэлийг зохицуулдаггүй
- Overfitting-г шийдэх системчилсэн арга байхгүй

Олон төрлийн алгоритмууд

Хамгийн түгээмэл алгоритмууд

- C4.5 – Quinlan-ы 2 дахь алгоритм (Quinlan, J.R. C4.5: Машины сургалтын програмууд, Morgan Kaufmann Publishers, 1993).
- салангид болон үргэлжилсэн функцэд ажилдаг
- Бүрэн бус бүртгэлийг зохицуулдаг
- Хуваах замаар хэт тохирох
- сургалтын өгөгдлүүдийн онцлог шинжүүдэд өөр өөр жинг хэрэглэж болдог

Олон төрлийн алгоритмууд

Хамгийн түгээмэл алгоритмууд

- C5.0 – Quinlan-ы хамгийн сүүлийн алгоритм. C4.5-ээс хурдан гэж тооцож байгаа.

Олон төрлийн алгоритмууд

Хамгийн түгээмэл алгоритмууд

- АРМ (Ангилал ба регрессийн мод) – өгөгдлийг тоон хуваагдлын шалгууураар модыг бүтээх алгоритмыг Станфордын судлаачдын бүлэг боловсруулсан (Breiman, L., Friedman, J., Stone, C.J., and Olshen, R.A. 1984, *Ангилал ба регрессийн мод*, Wadsworth Statistics/Probability)