Баггинг және кездейсоқ орман

1907 жылы статист сэр Фрэнсис Гальтон (Francis Galton) Англиядағы округтік жәрмеңкеге қатысты, онда көрмеде көрсетілген өгіздің сою салмағын болжау байқауы өтті. 800 болжам жасалды және жеке болжамдар айтарлықтай өзгергенімен, орташа және медиана өгіздің шынайы массасының 1% - га жетті. Джеймс Суровецки (Джеймс Суровиекки) бұл құбылысты өзінің "көпшіліктің даналығы" (the wisdom of Crowds) кітабында зерттеді. — Doubleday, 2004). Бұл қағида болжамды модельдерге де қолданылады: көптеген модельдерді — модельдерден тұратын ансамбльді орташалау (немесе көпшілік дауысты алу) тек бір модельді таңдаудан гөрі дәлірек болады.

Негізгі терминдер

Ансамбль (ансамбль) модельдер жиынтығының көмегімен болжамды қалыптастыру. Синоним: модельдердің орташалануы.

Баггинг *(bagging) деректерді жүктеу арқылы модельдер жиынтығын құрудың жалпы әдістемесі. Синонимдер: Bootstrap үлгілерін біріктіру, Bootstrap біріктіру.*

Кездейсоқ орман (random forest) шешім ағаштарының үлгілеріне негізделген жүктеме-жиынтық бағалау түрі. Синоним: Bootstrap-жиынтық шешім ағаштары.

Айнымалының маңыздылығы (variable importance) модельдің нәтижесі үшін болжамды айнымалының маңыздылығының метрикалық көрсеткіші.

Ансамбльдік тәсіл модельдеудің көптеген және әртүрлі әдістерінде қолданылады, бұл Netflix Contest байқауында айқын көрінді, бір уақытта Netflix компаниясы рейтингті болжауды 10% жақсартатын модельді ойлап тапқан кез - келген қатысушыға 1 миллион долларлық сыйлық ұсынды, оны Netflix клиенті марапаттайды.кинофильм. Ансамбльдердің қарапайым нұсқасында келесі көрініс бар:

1. Болжалды модель жасаңыз және нақты мәліметтер жиынтығы үшін болжамдарды жазыңыз.

2. Сол деректердегі көптеген модельдер үшін қайталаңыз.

3. Әрбір болжамды жазба үшін болжамдардың орташа мәнін алыңыз немесе көпшілік дауыс беру арқылы болжамды таңдаңыз. Ансамбльдік әдістер шешім ағаштарына жүйелі және тиімді қолданылады. Ансамбльдік ағаш үлгілері соншалықты күшті, олар салыстырмалы түрде аз күш жұмсай отырып, жақсы болжамды үлгілерді жасау жолын қамтамасыз етеді. Қарапайым ансамбльдік алгоритмнен тыс ансамбльдік модельдердің екі негізгі нұсқасы бар: бэггинг және бустинг. Егер біз ансамбльдік ағашты есте ұстасақ-көрнекті модельдер, содан кейін олар кездейсоқ орман модельдері және жүктелген ағаш модельдері деп аталады. Бұл бөлім бэггингке арналған; осы тараудың келесі аттас бөлімінде қарауды күшейту.

Бэггинг

Бэггинг (орыс тілінде —Bootstrap үлгілерін біріктіру) 1994 жылы Лео Брейман енгізген 4 болжамды айнымалылардың Y және P жауаптары бар делік x =x1, x2,.. xp. n- жазбаларымен.

Бэггинг ансамбльдер үшін негізгі алгоритмге ұқсас, тек бір ерекшелік — әр түрлі модельдерді бірдей деректерге сәйкестендірудің орнына, әрбір жаңа модель қайта таңдалған жүктеу үлгісіне сәйкес келеді. Төменде бұл алгоритм формальды түрде ұсынылған:

1. Инициализациялау M, сәйкестендіру үшін модельдер саны және n, таңдамалы жазбалар саны(n<N). Итерацияны m =1 етіп орнатыңыз .

2. Ym және Xm (пакет) қосалқы үлгісін қалыптастыру үшін жаттығу деректерінің n жазбаларынан қайта жүктеу үлгісін (яғни қайтарумен) алыңыз.

3. ˆf m (X) шешім ережелерінің жиынтығын жасау үшін Ym және Xm көмегімен модельді жаттықтырыңыз .

4. m=m+1 модель есептегішін көбейтіңіз. Егер m ≤ M болса, 1-қадамға өтіңіз.

Fm, Y=1 ықтималдығын болжайтын жағдайда, bagging негізіндегі бағалау (bagged estimate — пакеттік бағалау) келесі формуламен беріледі:

$$\hat{f} = \frac{1}{M}\left(\hat{f}_1(\mathbf{X}) + \hat{f}_2(\mathbf{X}) + \ldots + \hat{f}_M(\mathbf{X})\right).$$

Кездейсоқ орман

Кездейсоқ Орман бір маңызды кеңейтімі бар шешім ағаштарына багингті қолдануға негізделген: жазбаларды таңдаудан басқа, алгоритм айнымалыларды да таңдай. Дәстүрлі шешім ағаштарында a сегментінің ішкі

сегментін қалай құру керектігін анықтау үшін алгоритм критерийді, атап айтқанда Джи - ни гетерогенділік коэффициентін азайту арқылы айнымалы және бөлу нүктесін таңдайды (бөлімді қараңыз. Осы тараудың басында" біртектілікті немесе гетерогенділікті өлшеу"). Алгоритмнің әр кезеңінде кездейсоқ ормандарды қолданған кезде айнымалыны таңдау айнымалылардың кездейсоқ жиынтығымен шектеледі. Негізгі ағаш алгоритмімен салыстырғанда, кездейсоқ орман алгоритмі тағы екі қадамды қосады: бұрын талқыланған баггинг және әр бөлу нүктесінде айнымалыларды жүктеу:

1. Жазбалардан Bootstrap қосалқы үлгісін алыңыз (қайтарумен).

2. Бірінші бөлу үшін p < P айнымалыларын қайтарусыз кездейсоқ ретпен таңдаңыз.

3. Таңдалған айнымалылардың әрқайсысы үшін x j(1) , x j(2),... x j(p) x алго бөлу ритағын қолданыңыз: X-тен S j (k) әрбір мәні үшін:

- а сегментіндегі жазбаларды Xj(k) < Sj(k) - мен бір сегментке және қалған жазбаларға бөліңіз, мұндағы Xj(k) ≥ Sj(k), басқа сегментке;

- Әрбір а кіші сегментіндегі сыныптардың біртектілігін өлшеңіз сыныптың максималды сегментішілік біртектілігін тудыратын Sj(k) мәнін таңдаңыз.

4. Xj(k) айнымалысын және Sj(k) бөлу нүктесіндегі мәнді таңдаңыз , ол шегі-сыныптың максималды сегментішілік біртектілігін береді.

5. Келесі бөлімге өтіп, 2-қадамнан бастап алдыңғы қадамдарды қайталаңыз.

6. Ағаш өскенше сол процедураны орындай отырып, қосымша бөлуді жалғастырыңыз.

7. 1-қадамға оралыңыз, тағы бір жүктеу үлгісін алыңыз және процесті қайтадан бастаңыз. Әр қадамда қанша айнымалы таңдалады?

Ереже $\sqrt{P}$ таңдау туралы айтады, мұндағы P-болжамды айнымалылар саны. Randomforest бағдарламалық пакеті R-де кездейсоқ орманды жүзеге асырады.келесі үзінді бұл пакетті несие деректеріне қолданады (бөлімді

қараңыз. "K жақын көршілер" осы тараудың басында деректерді сипаттауға қатысты).

```
> library(randomForest)
> rf <- randomForest(outcome ~ borrower_score + payment_inc_ratio,
                     data=loan3000)
Call:

randomForest(formula = outcome ~ borrower_score + payment_inc_ratio,
             data = loan3000)
              Type of random forest: classification
                    Number of trees: 500
No. of variables tried at each split: 1

        OOB estimate of error rate: 38.53%
Confusion matrix:
         paid off default class.error
```

paid off 1089 425 0.2807133

default 731 755 0.4919246

Әдепкі бойынша 500 ағаш оқытылады. Болжаушылар жиынтығында тек екі айнымалы болғандықтан, алгоритм кез - келген тәртіпте айнымалыны таңдайды, оған сәйкес әр қадамда бөлуді орындауға болады (яғни 1 өлшемді жүктеу үлгісі).

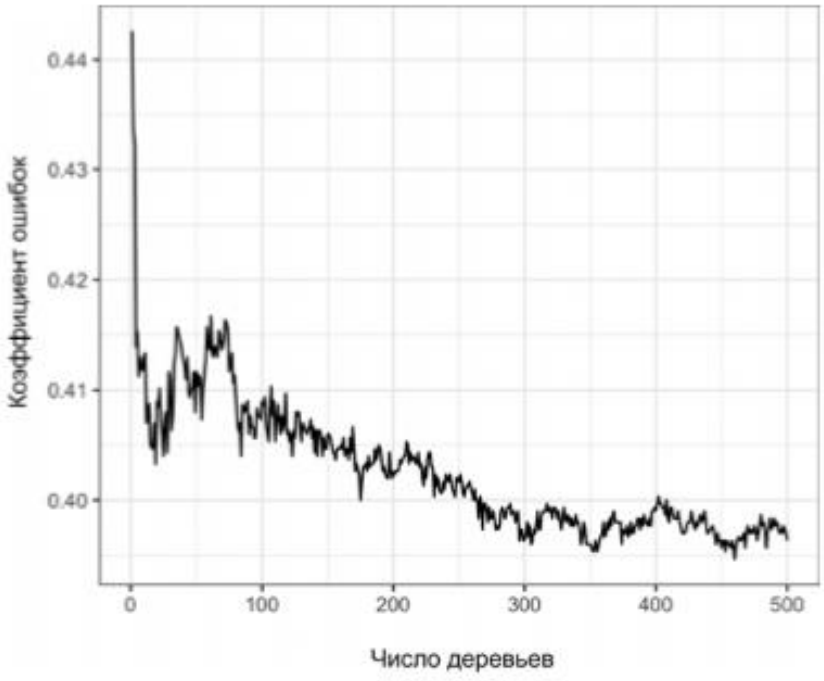
Пакеттен тыс бағалау OOB (out-of-bag — пакетке кірмеген) қателер - бұл осы ағашқа арналған жаттығу жинағынан алыс орналасқан деректерге қолданылатын оқытылған модельдер үшін қателік коэффициенті. Үлгінің шығысын пайдаланып, OOB қатесін кездейсоқ ормандағы ағаштар санына қарсы графикте көрсетуге болады:

```
error_df = data.frame(error_rate = rf$err.rate[,'OOB'],
num_trees = 1:rf$ntree)
ggplot(error_df, aes(x=num_trees, y=error_rate)) +
 geom_line()
```

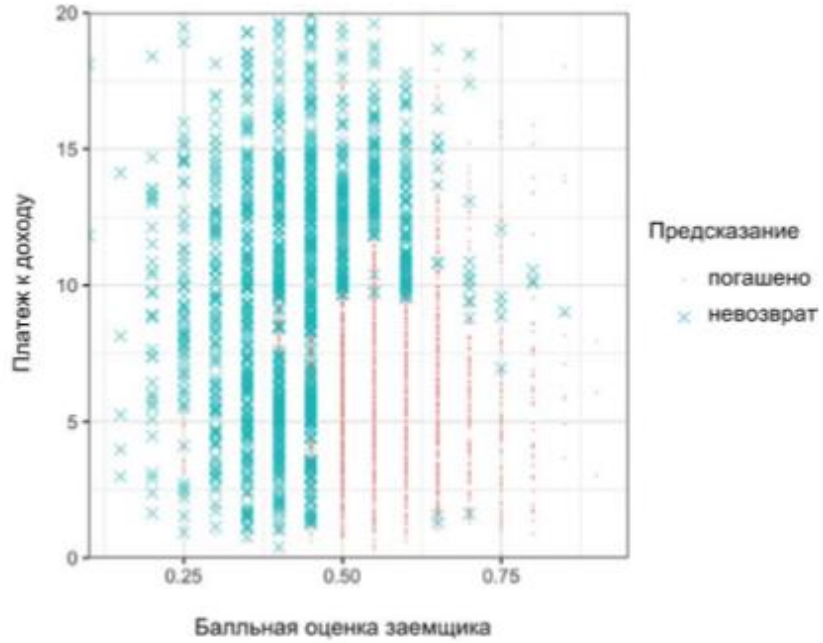
6.6.-суретте нәтиже көрсетілген. Қателік коэффициенті 0,44-тен 0,385 деңгейінде тұрақтанғанға дейін тез төмендейді. Болжалды мәндер predict функциясынан алынуы мүмкін және графикте келесідей көрсетілуі мүмкін

```
pred <- predict(loan_lda)
rf_df <- cbind(loan3000, pred_default=pred[,'default']>.5)
```

ggplot(data=rf_df, aes(x=borrower_score, y=payment_inc_ratio, color=pred_default, shape=pred_default)) +
geom_point(alpha=.6, size=2) +
scale_shape_manual(values=c(46, 4))



6.6. -Сурет. Көбірек ағаштарды қосу арқылы кездейсоқ орманның дәлдігін жақсарту



6.7-Сурет. Қайтарылмайтын несие деректеріне қатысты кездейсоқ орманнан болжамды нәтижелер

6.7- Суретте көрсетілген график, кездейсоқ орманның табиғатына қатысты өте айқын. Кездейсоқ орман әдісі - "қара жәшік" әдісі. Ол қарапайым ағашқа қарағанда дәлірек болжамдар жасайды, бірақ қарапайым ағашты шешудің интуитивті ережелері жоғалады. Болжамдар да біршама шулы: жоғары несиелік қабілеттілік туралы айтатын өте жоғары балл жинаған кейбір қарыз алушылар әлі де несиені қайтармау туралы болжамды алатынын ескеріңіз.
Бұл деректердегі бірнеше ерекше жазбалардың нәтижесі және кездейсоқ орман тудырған қайта құру қаупін көрсетеді

Айнымалылардың маңызы

Кездейсоқ орман алгоритмінің күші көптеген белгілері мен жазбалары бар мәліметтер үшін болжамды модельдер жасауда көрінеді. Алгоритм қандай болжаушылардың маңызды екенін автоматты түрде анықтай алады және өзара әрекеттесу мүшелеріне сәйкес келетін болжаушылар арасындағы күрделі байланыстарды анықтай алады.Мысалы, бағандарды қосу арқылы модельді қайтарылмайтын несие деректеріне сәйкестендірейік:

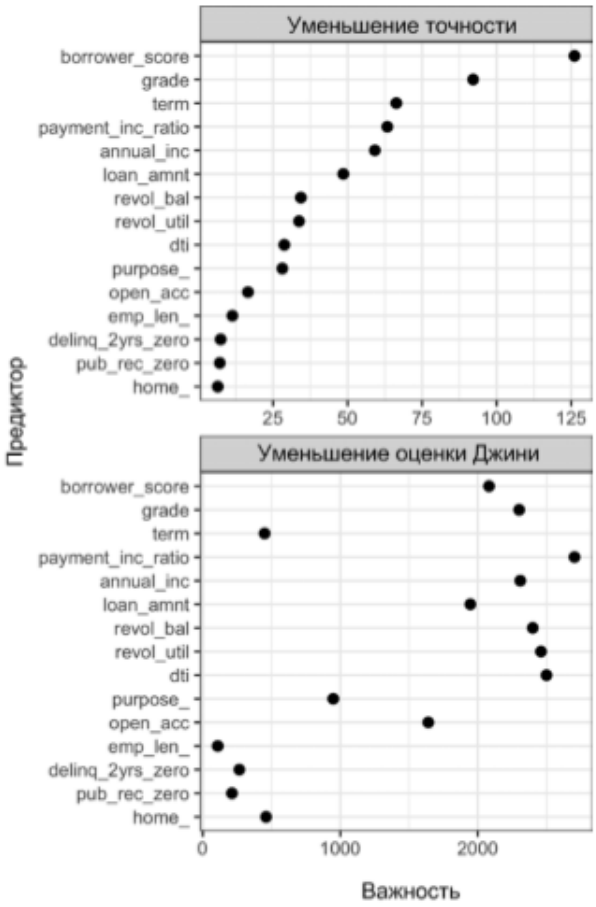
```
> rf_all <- randomForest(outcome ~ ., data=loan_data, importance=TRUE)
> rf_all
Call:
randomForest(formula = outcome ~ ., data = loan_data, importance = TRUE
Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 3
      OOB estimate of error rate: 34.38%
Confusion matrix:
paid off default class.error paid off    15078    8058   0.3482884
default      7849   15287   0.3392548
```

Importance=TRUE аргументі random Forest-тен әртүрлі айнымалылардың маңыздылығы туралы қосымша ақпаратты сақтауды сұрайды. VarImpPlot функциясы айнымалылардың салыстырмалы өнімділігін сызады:

```
varImpPlot(rf_all, type=1)

 varImpPlot(rf_all, type=2)
```

Нәтиже 6.8.-суретте көрсетілген.



6.8. -Сурет. Модельді несие деректеріне толық сәйкестендіру үшін айнымалылардың маңыздылығы

Айнымалылардың маңыздылығын бағалаудың екі әдісі бар.

- Егер айнымалы мәндер кездейсоқ ретпен қайта реттелсе, модельдің дәлдігін төмендету арқылы (type=1). Мәндерді ерікті түрде ауыстыру осы айнымалы үшін барлық болжамды күштерді жоюға әсер етеді. Дәлдік пакеттен тыс мәліметтерден есептеледі (сондықтан бұл шара іс жүзінде кросс - тексеру бағасы болып табылады).

- Джинидің гетерогенділігін бағалаудағы орташа төмендеу арқылы (бөлімді қараңыз. Осы тараудың басында" біртектілік немесе гетерогенділік өлшемі") айнымалы бойынша бөлінген барлық түйіндер үшін (type=2). Бұл айнымалының түйіндердің тазалығын жақсартуға қандай үлес қосатынын көрсетеді. Өлшем жаттығу жиынтығына негізделген, сондықтан пакеттен тыс деректерде есептелген өлшемге қарағанда сенімділігі төмен.

Жоғарғы және төменгі бөліктері 6.8 -сурет. көрсетілген тәртіпте Джинидің дәлдігі мен гетерогенділігінің төмендеуіне сәйкес айнымалылардың маңыздылығын көрсетеді. Екі бөліктегі айнымалылар дәлдіктің төмендеуіне байланысты. Осы екі шара арқылы жасалған айнымалылардың маңыздылығын бағалау өте әртүрлі. Дәлдіктің төмендеуі неғұрлым сенімді метрикалық көрсеткіш болғандықтан, неге Джинидің гетерогенділігін азайту шарасын қолдану керек? Әдепкі бойынша, randomForest тек Джинидің бағасын есептейді: Джинидің гетерогенділік өлшемі алгоритмнің қосымша өнімі болып табылады, ал айнымалыға байланысты модельдің дәлдігі қосымша есептеулерді қажет етеді (деректерді кездейсоқ ауыстыру және осы деректерді болжау). Есептеу күрделілігі маңызды болған жағдайда, мысалы, мыңдаған модельдер Орнатылатын жұмыс ортасында, ол (күрделілік) қосымша есептеу күшіне тұрарлық болмайды. Сонымен қатар, Джини шарасының азаюы кездейсоқ ормандар өздерінің бөлу ережелерін жасау үшін қолданатын өзгергіштерге жарық түсіреді (қарапайым ағашта оңай көрінетін бұл ақпарат кездейсоқ орманда жоғалып кететінін есте сақтаңыз). Джинидің гетерогенділігін азайту мен модельдің дәлдігі арқылы айнымалылардың маңыздылығы арасындағы айырмашылықты зерттеу модельді жақсарту жолдарын ұсына алады.

Гиперпараметрлер айнымалылардың маңыздылығын бағалаудың екі әдісі бар.

Егер айнымалы мәндер кездейсоқ ретпен қайта реттелсе, модельдің дәлдігін төмендету арқылы (type=1). Мәндерді ерікті түрде ауыстыру осы айнымалы үшін барлық болжамды күштерді жоюға әсер етеді. Дәлдік пакеттен тыс мәліметтерден есептеледі (сондықтан бұл шара іс жүзінде кросс - тексеру бағасы болып табылады).

Джинидің гетерогенділігін бағалаудағы орташа төмендеу арқылы (бөлімді қараңыз. Осы тараудың басында" біртектілік немесе гетерогенділік өлшемі") айнымалы бойынша бөлінген барлық түйіндер үшін (type=2). Бұл айнымалының түйіндердің тазалығын жақсартуға қандай үлес қосатынын көрсетеді. Өлшем жаттығу жиынтығына негізделген, сондықтан пакеттен тыс деректерде есептелген өлшемге қарағанда сенімділігі төмен. Жоғарғы және төменгі бөліктері сурет. 6.8 көрсетілген тәртіпте Джинидің дәлдігі мен гетерогенділігінің төмендеуіне сәйкес айнымалылардың маңыздылығын көрсетеді. Екі бөліктегі айнымалылар дәлдіктің төмендеуіне байланысты. Осы екі шара арқылы жасалған айнымалылардың маңыздылығын бағалау өте әртүрлі. Дәлдіктің төмендеуі неғұрлым сенімді метрикалық көрсеткіш

болғандықтан, неге Джинидің гетерогенділігін азайту шарасын қолдану керек? Әдепкі бойынша, randomForest тек Джинидің бағасын есептейді: Джинидің гетерогенділік өлшемі алгоритмнің қосымша өнімі болып табылады, ал айнымалыға байланысты модельдің дәлдігі қосымша есептеулерді қажет етеді (деректерді кездейсоқ ауыстыру және осы деректерді болжау). Есептеу күрделілігі маңызды болған жағдайда, мысалы, мыңдаған модельдер Орнатылатын жұмыс ортасында, ол (күрделілік) қосымша есептеу күшіне тұрарлық болмайды. Сонымен қатар, Джини шарасының азаюы кездейсоқ ормандар өздерінің бөлу ережелерін жасау үшін қолданатын өзгергіштерге жарық түсіреді (қарапайым ағашта оңай көрінетін бұл ақпарат кездейсоқ орманда жоғалып кететінін есте сақтаңыз). Джинидің гетерогенділігін азайту мен модельдің дәлдігі арқылы айнымалылардың маңыздылығы арасындағы айырмашылықты зерттеу модельді жақсарту жолдарын ұсына алады.

Гиперпараметрлер

Кездейсоқ орман, көптеген статистикалық Машиналық оқыту алгоритмдері сияқты, қораптың жұмыс сипатын құруға арналған түймелері бар "қара жәшік" алгоритмі ретінде қарастырылуы мүмкін. Бұл түймелер гиперпараметрлер деп аталады, яғни модельді орнатуды бастамас бұрын конфигурациялануы керек параметрлер; олар жаттығу процесінің құрамдас бөлігі ретінде оңтайландырылмайды. Дәстүрлі статистикалық модельдер таңдауды қажет етсе де (мысалы, регрессия моделінде қолдану үшін болжаушыларды таңдау), шай - шай орманының гиперпараметрлері, әсіресе қайта орнатудың алдын алу үшін үлкен маңызға ие. Атап айтқанда, кездейсоқ орманның ең маңызды екі гиперпараметрі келесідей:

- nodesize-терминал түйіндерінің ең аз мөлшері (ағаштағы жапырақтар), әдепкі мәні жіктеу үшін 1 және регрессия үшін 5;
- maxnodes-әр шешім ағашындағы түйіндердің максималды саны. Әдепкі бойынша шек жоқ және ең үлкен ағаш nodesize - де орнатылған шектеулерге сәйкес келеді.

Бұл опцияларды елемеу және жай ғана әдепкі мәндерден бастау қызықты болуы мүмкін. Дегенмен, әдепкі мәнді пайдалану шулы деректерге кездейсоқ орманды қолданған кезде қайта орнатуға әкелуі мүмкін. Nodesize-ді үлкейту немесе maxnodes - ті орнату кезінде алгоритм кішірек ағаштарды сәйкестендіреді және болжамды болжау ережелерін жасау ықтималдығы аз болады. Әртүрлі білімдердің гиперпараметрлерімен қабылдау әсерін тексеру үшін кросс-тексеру қолданылуы мүмкін

Бэггинг пен кездейсоқ орманға арналған негізгі идеялар

- *Ансамбльдік модельдер көптеген модельдердің нәтижелерін біріктіру арқылы модельдің дәлдігін жақсартады.*
- *Bagging-бұл көптеген модельдерді деректерден жүктеу үлгілеріне және модельдердің орташалануына сәйкес келетін ансамбльдік модельдің ерекше түрі.*
- *Кездейсоқ орман-бұл шешім ағаштарына қолданылатын баггингтің ерекше түрі. Деректерді қайта таңдаудан басқа, кездейсоқ орман алгоритмі ағаштарды бөлу кезінде болжамды айнымалыларды таңдайды.*
- *Кездейсоқ орманнан шығудың пайдалы деректері болжаушыларды модельдің дәлдігіне қосқан үлесі тұрғысынан бағалайтын айнымалылардың маңыздылығының өлшемі болып табылады.*
- *Кездейсоқ орманда бірнеше гиперпараметрлер бар, оларды қайта орнатуды болдырмау үшін кросс-тексеру арқылы реттеу керек.*

Бустинг

Ансамбльдік модельдер болжамды модельдеудің стандартты құралына айналды. Бустинг-бұл модельдер ансамблін құрудың жалпы әдісі. Ол бэггингпен бір уақытта жасалған. Bagging сияқты, boosting шешім ағаштарымен өте кең қолданылады. Олардың ортақ қасиеттеріне қарамастан, бустингте көптеген артық аксессуарлармен бірге жүретін мүлдем басқа тәсіл қабылданды. Нәтижесінде, бүктеуді салыстырмалы түрде аз түзетумен қолдануға болады, ал күшейту оны қолдануда әлдеқайда мұқият болуды талап етеді. Егер бұл екі әдіс автомобильдер болса, онда bagging accord моделінің Honda көлігі ретінде қарастырылуы мүмкін (сенімді және тұрақты), ал boosting Porsche болар еді (қуатты, бірақ көп күтімді қажет етеді).

Сызықтық регрессиялық модельдерде қалдықтардың сәйкестікті жақсартуға болатындығын тексеру үшін жиі тексеріледі (бөлімді қараңыз. "4-тараудың" жеке қалдық графиктері және сызықтық емес"). Бустинг бұл тұжырымдаманы әлдеқайда алға тартты және алдыңғы модельдердің қателіктерін азайту мақсатында әрбір келесі модель сәйкес келетін модельдер сериясын орындайды. Бұл алгоритмнің бірнеше нұсқалары кеңінен қолданылады: Adaboost, градиентті күшейту және стохастикалық градиентті күшейту. Жоғарыда айтылғандардың соңғысы, стохастикалық градиентті күшейту, ең жалпы болып табылады. Шын мәнінде, параметрлерді дұрыс таңдау арқылы бұл алгоритм кездейсоқ орманды еліктей алады.

Негізгі терминдер

Ансамбль (ансамбль) модельдер жиынтығын қолдану арқылы болжамды қалыптастыру.

Синонимдер: модельдердің орташалануы, модельдердің құрамы.

Boosting (boosting) әрбір келесі цикл үшін үлкен қалдықтары бар жазбаларға көбірек салмақ беру арқылы модельдер тізбегін орнатудың жалпы әдістемесі.

Adaboost деректердің қалдыққа негізделген артық салмағына негізделген күшейткіштің ерте нұсқасы.

Градиентті күшейту (градиентті күшейту) - бұл шығындар функциясын азайту тұрғысынан жасалған күшейтудің жалпы түрі.

Стохастикалық градиентті күшейту (stochastic gradient boosting) әр циклде жазбалар мен бағандарды қайта таңдауды қамтамасыз ететін ең жалпы күшейту алгоритмі.

Регуляризация (regularization) модельдегі бірқатар параметрлердегі құн функциясына айыппұл мүшесін қосу арқылы қайта құрудың алдын алу әдісі.

Гиперпараметрлер (hyperparameters) алгоритмді орнатпас бұрын Орнатылатын Параметрлер.

Күшейту алгоритмі

Әр түрлі күшейту алгоритмдерінің негізінде жатқан негізгі идея новости-Жаңалықтар бірдей. Түсіну оңай-adaboost алгоритмі, ол келесідей жұмыс істейді:

1. Инициализациялау M-сәйкес келетін модельдердің максималды саны және Итерация есептегішін m = 1 етіп орнатыңыз. үшін w i= 1/ N бақылау салмағын инициализациялау,i =1,2,…n. Ансамбль моделін инициализациялау F0 =0 .

2. Бақылау салмағын қолдана отырып, ˆM f көмегімен модельді жаттықтырыңыз

w 1, w2, ... wn Дұрыс жіктелмеген бақылаулардың салмағын қосу арқылы анықталған өлшенген M е қатесін азайтатын

3. Ансамбльге модель қосу:

: $\hat{F}_m = \hat{F}_{m-1} + \alpha_m \hat{f}_m$, где $\alpha_m = \frac{\log 1 - e_m}{e_m}$.

4. Салмақтарды жаңарту w1, w2, ... wn, дұрыс емес жіктелген бақылаулардың салмағын арттыратындай етіп. Үлкейту мөлшері M α-ға ілінеді, ал үлкен м α мәндері үлкен салмаққа әкеледі.

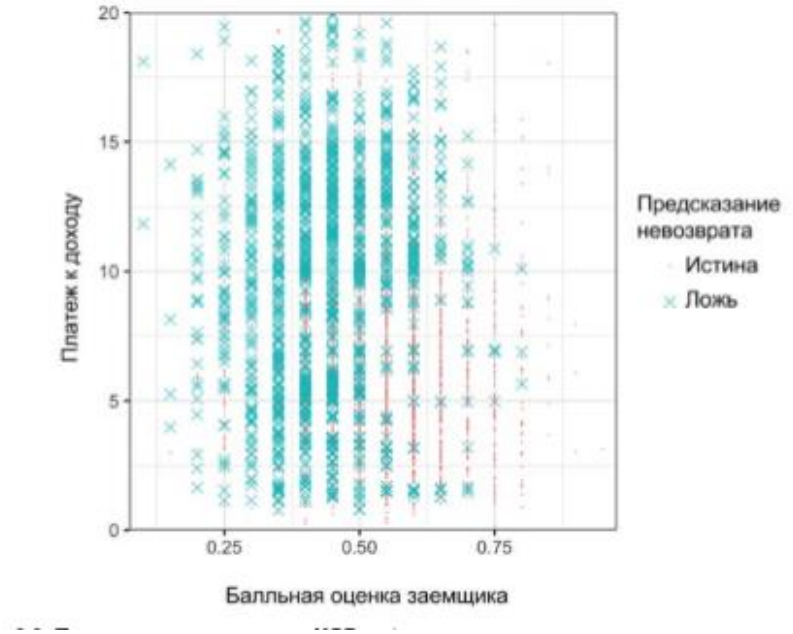
5. m =m+1 модель есептегішін жасаңыз. Егер m ≤m болса, онда 1-қадамға өтіңіз. Күшейтілген бағалау келесі формуламен беріледі:

$$\hat{F} = \alpha_1 \hat{f}_1 + \alpha_2 \hat{f}_2 + ... + \alpha_M \hat{f}_M.$$

Қате жіктелген бақылаулардың салмағын арттыру арқылы алгоритм модельдерді нашар өнімділік көрсеткен мәліметтерге белсенді түрде жаттығуға шақырады. M α факторы қателігі төмен модельдердің салмағы көп болуын қамтамасыз етеді. Градиентті күшейту Adaboost-қа ұқсас, бірақ тапсырманы шығындар функциясын оңтайландыру ретінде ұсынады. Салмақты түзетудің орнына, градиентті күшейту модельдерді жалған қалдыққа сәйкестендіреді, бұл үлкен қалдықтарға белсенді жаттығуларға әсер етеді. Кездейсоқ орман рухында стохастикалық градиентті күшейту алгоритмге әр кезеңде бақылаулар мен болжамды айнымалыларды таңдау арқылы еріктілік қосады.

XGBoost

XGBoost - бұл Вашингтон университетінде тянси Чен (Tianqi Chen) және Карлос Гестрин (Carlos Guestrin) шығарған стохастикалық градиентті күшейтуді жүзеге асыратын ең көп қолданылатын ақысыз бағдарламалар пакеті. Оның көптеген нұсқалары бар есептеуіш тиімді іске асырылуы деректер ғылымында қолданылатын негізгі бағдарламалау тілдерінің көпшілігі үшін бағдарламалық кітапхана ретінде қол жетімді. The R XGBoost xgboost бағдарламалық пакеті ретінде қол жетімді. Xgboost функциясында түзетуге болатын және қажет болатын көптеген параметрлер бар (бөлімді қараңыз. "Гиперпараметрлер және Кросс-тексеру" осы тарауда). Екі өте маңызды параметр - бұл әр Итерация кезінде таңдалуы керек тағамдардың үлесін басқаратын subsample және күшейту алгоритмінде αm — ге қолданылатын eta қысу факторы (бөлімді қараңыз. Осы тараудың басында" күшейту алгоритмі"). Subsample пайдалану күшейткішті кездейсоқ орман ретінде әрекет етуге мәжбүр етеді, тек іріктеу қайтарусыз жүзеге асырылады.



6.9. Сурет. Қайтарылмайтын несие деректеріне қатысты XGBoost болжамды нәтижелері

6.9.-суретте көрсетілген. Сапалы түрде бұл кездейсоқ орманның 6.7-сурет болжамдарына ұқсас. Болжамдар біршама шулы, өйткені қарыз алушының өте жоғары баллдық ұпайы бар кейбір қарыз алушылар әлі де несиені қайтармау туралы болжамды алады.

Негізгі күшейту идеялары

Бустинг - бұл кейінгі циклдарда үлкен қателіктері бар жазбаларға көбірек салмақ берілетін модельдердің сәйкестігіне негізделген ансамбльдік модельдер класы. •

Стохастикалық градиентті күшейту-ең жақсы өнімділікті қамтамасыз ететін күшейткіштің ең жалпы түрі. Стохастикалық градиентті күшейтудің жалпы қабылданған түрі ағаш үлгілерін пайдаланады. •

XGBoost-бұл стохастикалық градиентті күшейтуге арналған танымал және есептеуіш тиімді бағдарламалар жиынтығы; ол деректер ғылымында қолданылатын барлық қабылданған бағдарламалау тілдерінде қол жетімді. •

Бустинг деректердің қайта реттелуіне ұшырайды және оның алдын алу үшін гиперпараметрлерді конфигурациялау қажет. •

Регуляризация-бұл теңдеудің айыппұл мүшесін модельдегі параметрлер жиынтығына (мысалы, ағаш өлшемі) қолдану арқылы қайта құрудың алдын алудың бір әдісі. •

Кросс-тексеру, әсіресе, гиперпараметрлердің көптігіне байланысты күшейту үшін өте маңызды.