**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**Факультет інформаційних технологій**

Кафедра технологій управління

Освітня програма «Інформаційна аналітика та впливи»

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

на тему:

«Вдосконалення методів прогнозування для покращення кредитної системи в Україні»

**Студентки 6-го курсу групи ІАВ-21 Науковий керівник:**

Сокол Олена Олександрівни д.т.н., доцент, професор кафедри

*(прізвище, ім’я, по батькові) (науковий ступінь, вчене звання)*

Колеснікова Катерина Вікторівна

*(прізвище, ім’я, по батькові)*

*(підпис студента) (дата) (підпис)*

**Попередній захист:**

*(Висновок: “До захисту в Державній екзаменаційній комісії”)*

|  |  |  |
| --- | --- | --- |
| Завідувач кафедри |  | |
| технологій управління |  |  |
| *(підпис)* | *(прізвище, ініціали)* | *(дата)* |

**Київ 2019**

ЗАТВЕРДЖУЮ

Завідувач кафедри

Морозов В.В.

*(підпис)*

«\_\_» \_\_\_\_\_\_\_\_ 20\_\_р.

**ЗАВДАННЯ**

**НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студентка: Сокол Олена Олександрівна

Група: ІАВ-21

1. **Тема дипломної роботи**: «Вдосконалення методів прогнозування для покращення кредитної системи в Україні».

2. **Строк подання студентом готової роботи** – «\_\_» \_\_\_\_\_\_\_\_ 20\_\_р.

3. **Цільова установка та вихідні дані до роботи**: дослідити методи оцінювання кредитоспроможності клієнтів банку. Зібрати репрезентативну вибірку клієнтів банку і вибрати математичні методи та алгоритми моделювання, які покращуватимуть точність прогнозу.

4. **Зміст роботи**:

1. Дослідити проблему підвищення точності скорингового аналізу даних у банках України.

2. Розробити методи та алгоритми для покращення прогнозування.

3. Обрати програмні засоби.

4. Виконати програмну реалізацію.

5. Провести порівняльний аналіз результатів отриманого дослідження.

6. Проаналізувати практичну цінність роботи.

5. **Перелік графічного матеріалу**:

* структурні елементи процесу скорингу у вигляді таблиць та схем;
* графічні приклади роботи алгоритмів;
* візуалізація результатів запропонованих методів та алгоритмів, порівняння моделей;
* представлення бізнес-моделі стартапу у вигляді таблиць.

6. **Календарний план виконання роботи:**

|  |  |
| --- | --- |
| **Етапи виконання дипломних робіт** | **Термін виконання** |
| 1. Постановка завдання.   Збір статистичної інформації | 01.12.2018 |
| 2. Дослідження існуючих рішень і підходів | 28.01.2019 |
| 3. Дослідження алгоритмів прогнозування | 10.03.2019 |
| 4. Розробка програмної моделі | 15.03.2019 |
| 5. Налаштування гіперпараметрів | 25.03.2019 |
| 6. Тестування алгоритмів | 25.04.2019 |
| 7. Перевірка точності обраних методів з відібраними гіперпараметрами | 30.04.2019 |
| 8. Оформлення дипломної роботи | 11.05.2019 |
| 9. Попередній захист дипломної роботи | 17.05.2019 |
| 10. Захист роботи | 31.05.2019 |

Дата видачі завдання «\_\_» \_\_\_\_\_\_\_\_ 20\_\_р.

Керівник роботи: д.т.н., доцент, проф. кафедри Колеснікова К.В.\_\_\_\_\_\_\_\_\_\_

(підпис)

Завдання прийняла до виконання:

студентка групи ІАВ-21 Сокол О.О. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(підпис)

ЗМІСТ

[**ВСТУП**](#_1v9tyqnrnvev) **10**

[**РОЗДІЛ 1 ОЦІНКА ПЛАТОСПРОМОЖНОСТІ КЛІЄНТА**](#_ldz9x9s4c5m2) **11**

[1.1 Сутність кредитного ризику і методи боротьби з ним](#_tz5og59i3q64) 11

[1.2 Нові джерела даних](#_x88qenhqsoj) 13

[1.2.1 Причини залучення нових даних](#_tig4r39jpy08) 13

[1.2.2 Дані соціальних мереж](#_aggaign5g9o) 14

[1.2.3 Дані “Яндекс Таксі” або Uber](#_uwkyqggmdc8a) 16

[1.2.4 Дані про поведінку користувача на сайті банку](#_ivut2lfif4b6) 17

[1.2.5 Дані телеком-оператора](#_fni0btbvwhbd) 18

[Висновки до розділу 1](#_3pgcp62djbr) 20

[Постановка задачі дисертаційного дослідження](#_u9i741d15f3i) 20

[**РОЗДІЛ 2 ПЕРВИННА ОБРОБКА ДАНИХ ТА ПОБУДОВА МОДЕЛІ**](#_dmt7ao9tbz6x) **22**

[2.1 Препроцессінг даних](#_dcf41795lfpq) 22

[2.2.1 Обробка категоріальних даних](#_kz3p4cq4lwvl) 22

[2.2.2 Обробка неперервних даних](#_y0eu6gqtlzd9) 23

[2.2.3 Weight-decay](#_mfx89kewzky1) 25

[2.2.3.1 L2 penalty](#_yoadng9ylz8e) 25

[2.2.3.2 L1 penalty](#_fjkuffz280k4) 27

[2.2.4 Відсів викидів](#_r8dgzmek2pr4) 28

[2.2.4.1 Відсів спостережень](#_hb0lfp6g7xoq) 28

[2.2.4.2. Відсів характеристик](#_mqobfnnm2uyd) 29

[2.2 Побудова моделей](#_n9pfvbun44bo) 29

[2.2.1 Логістична регресія](#_xdde4nw1to05) 29

[2.2.2 Дерева рішень](#_irv2qf7zitei) 30

[2.3 Ансамблі алгоритмів](#_utwbjzeedpui) 31

[2.3.1 Загальна ідея](#_tyn864hrav3) 31

[2.3.2 Bagging](#_37jn4kusey71) 33

[2.3.3 Boosting](#_dco17b4osaha) 35

[2.4 Критерії якості класифікаційних моделей](#_31ds4p832au) 36

[2.4.1 Загальна точність моделі](#_ro1628e8yxq2) 36

[2.4.2 Помилки І-го та ІІ-го роду;](#_lc1ju6qiqj2l) 37

[2.4.3 ROC -крива та індекси AUC та GINI.](#_32528hxxj4qg) 37

[2.4.4 Розбиття датасету](#_dhez8chaahzo) 40

[Висновки за розділом 2](#_w0ii2r1k37e4) 43

[**РОЗДІЛ 3 ПОБУДОВА МОДЕЛЕЙ ЗА ФАКТИЧНИМИ ДАНИМИ**](#_vl7w98iexnh8) **44**

[3.1 Збір даних](#_q2wofaafl0kc) 44

[3. 2 Побудова моделі](#_mgplh5xleei) 47

[3.2.1 Оцінка платоспроможності абонентів “Київстар”](#_8lydxdlijvam) 47

[3.2.2 Оцінка платоспроможності абонентів інших операторів за даними компанії “Київстар”](#_pcrlfzbvici8) 50

[3.3 Аналіз роботи моделей](#_gvx2whsphid6) 51

[3.3.1 Аналіз роботи моделі для прогнозування абонентів “Київстар”](#_mv0y4zm8sx7s) 51

[3.3.2 Аналіз роботи моделі для прогнозування платоспроможності абонентів інших операторів за даними компанії “Київстар”](#_r1bgjx9ibawv) 55

[Висновки за розділом 3](#_o698nl633v2u) 58

[**РОЗДІЛ 4 РОЗРОБКА СТАРТАПУ**](#_1bk9n3qi5gbw) **59**

[4.1 Опис проекту](#_5v5vwbras7g5) 59

[4.2 Команда проекту](#_sh8ie2rgs5k9) 59

[4.3 Бізнес модель проекту](#_kfdkeav9h3iv) 61

[4.4 Аналіз ринкових можливостей стартап проекту](#_4hxatonw4zy6) 63

[4.5 Розроблення ринкової стратегії проекту](#_cu3be5qw26pw) 72

[4.6 Розроблення маркетингової програми стартап-проекту](#_5poi6riqwbg3) 75

[Висновки за розділом 4](#_91vh437nak80) 80

[**ВИСНОВКИ ЗА РОБОТОЮ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ**](#_soeyepmkwj01) **81**

[**ПЕРЕЛІК ПОСИЛАНЬ**](#_svn4mrnahy34) **83**

[**Додаток А Код програмного продукту**](#_ftrk0qoip1ts) **86**

[**Додаток Б Акт впровадження**](#_5d3ns4stghk) **92**

[**Додаток В Ілюстративні матеріали доповіді**](#_fqjz8u2ycte3) **93**

# 

# РОЗДІЛ 1. CКОРИНГОВІ СИСТЕМИ ЯК МЕТОД УПРАВЛІННЯ КРЕДИТНИМИ РИЗИКАМИ

## Кредитний ризик та методи боротьби з ним. Постановка задачі магістерської роботи.

## Основні поняття регулюванням кредитного ризику.

Основним завданням цього підрозділу - є пояснення причини виникнення та впровадження кредитного скорингу, щоб, в подальшому, провести оцінку автоматизованої системи кредитного скорингу.

Кредитний ризик — ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *credit risk*)ймовірність відхилення від запланованих (очікуваних) показників через невиконання позичальником зобов'язань перед банком. Кредитний ризик доцільно розділяти на індивідуальний (конкретний контрагент банку) та портфельний (сукупна заборгованість) ризики[1].

Щоб правильно оцінити платоспроможність потенційного клієнта, використовують скоринг.

Скоринг - це система оцінки позичальників, яка заснована на математико-статистичних методах. Мета скорингу – це оцінити рівень платоспроможності клієнта по деяким факторам і відібрати потенційних позичальників. Кредитні менеджер, під час співбесіди з позичальником, збирає максимум інформації про нього, після чого скорингова система обробляє інформацію, нараховуючи бали за кожен сприятливий фактор [2].

Найперша скорингова модель з'явилася в банках США, коли, під час Другої світової війни, практично всі кредитні інспектори були покликані захищати батьківщину. За проханням банків та інших кредитних організацій, перед відходом, кредитні аналітики розробили модель для прийняття рішень про видачу кредиту. Цією моделлю могли користатися навіть неспеціалісти.

Наступний крок у розвитку скорингу - це консалтингова компанія Fair Issac Corporation (FIC), яка займалася розвитком скорингових моделей. Сьогодні FIC продовжує розробляти скорингові системи для банків. А використовувана шкала FICO (Таблиця 1), яку створила компанія FIC, застосовується в більшості банків США. До уваги приймаються наступні складові:

* якість кредитної історії,
* наявність і розмір поточних боргів,
* тривалість відносин з кредиторами,
* співвідношення кількості поданих заявок і виданих кредитів,
* типи виданих кредитів [3].

|  |  |
| --- | --- |
| ***Бали*** | ***Опис*** |
| ***750+*** | Відмінний кредитний бал. Позичальнику надається кредит на самих вигідних умовах |
| ***700–750*** | Хороший кредитний бал. Немає проблем з видачею кредиту. Позичальник отримує кредит під хороший  відсоток. |
| ***640–700*** | Середній кредитний бал. Банк може видати позичальникові кредит, але на не дуже вигідних умовах.  Також можливий відмова банку. |
| ***580–640*** | Слабкий кредитний бал. Довгий процес прийняття рішення про видачу кредиту позичальникові і високі  процентні ставки |
| ***>580*** | Поганий кредитний бал. відмова у видачі кредиту позичальникові. |

Таблиця 1.1. Шкала FICO - показник якості позичальника [4]

На розвиток скорингу вплинуло ще два фактора. У США був прийнятий закон, про рівні отримання кредиту, який зобов'язав банки розглядати всі заявки (боротьба з расовою дискримінацією). Наступний фактор – це розвиток інформаційних технологій, які могли обробляти велику кількість надходження кредитних заявок.

Скорингові системи діляться на кілька видів, самі популярні з них:

• скоринг на підставі кредитної історії, який враховує кредитну історію потенційного позичальника (позики різних банків, прострочення (якщо вони є), спроби взяти позику у банку, наявність кредитних карт).

Головний недолік такого підходу очевидний. Вибірка класифікується тільки по тим клієнтам, яким вже давали кредит. Залишалося невідомим, як повели б себе ті клієнти, яким було відмовлено в кредиті або які за ним навіть не зверталися[3].

• Соціально-демографічний скоринг – це оцінка позичальника на підставі таких показників як вік або гендерні показники, сімейне положення, стаж роботи, професія.

Також скоринг враховує заробітню плату, як правило, за останні 6 місяців [3].

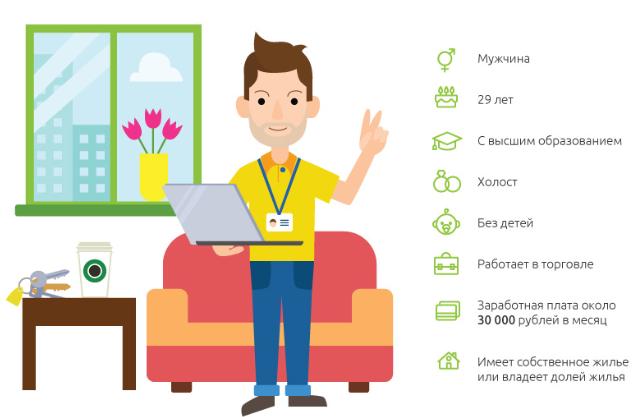


Рис. 1.1.1. Соціально-демографічні дані про клієнта

Кредитний менеджер проводить співбесіду і анкетування з потенційним позичальником, після чого вносить дані в програму. На підставі цих даних скорингова система присвоює бали за кожен фактор, а в кінці процедури відносить позичальника до певної «групи ризику» і віддає висновок про можливість надання кредиту.

Незважаючи на скоринговую систему, в кожному банку існує власна система

оцінки потенційних позичальників. Наприклад, є певні пороги за рівнем заробітної плати, якщо значення нижче поставленого порога, клієнт отримує відмову банку ще на рівні прескорінга.

В умовах кризи, для мінімізації фінансових ризиків деякі банки ввели мораторій на видачу позички непрацюючим пенсіонерам, чий дохід потенційно не зможе забезпечити повного погашення кредиту. Крім цього, співробітники можуть візуально оцінювати потенційного контрагента, чия неадекватна поведінка або неприйнятний зовнішній вигляд може стати перешкодою до отримання кредиту.

Після всіх, вищевикладених методів оцінки, заявка повинна пройти етап андеррайтингу і отримати схвалення у департаменту аналізу ризиків та служби

безпеки. У разі, якщо всі етапи пройдено, це ще не гарантія отримання кредиту. В умовах нестабільної економіки в регіонах, внутрішній контроль має право відмовити клієнту, проте оформлення страхового договору, дає високі

шанси на успіх позичальника [5].

Управління кредитними ризиками займає окреме місце в ефективному менеджменті будь-якого банку. Під кредитним ризиком мається на увазі невиконання контрагентом своїх кредитних зобов'язань. Найбільш поширений кредитний ризик - дефолт позичальника, коли контрагент не виконує зобов'язання перед банком щодо повернення грошових коштів згідно з умовами кредитного угоди в силу економічної неспроможності або небажання.

Фактори кредитного ризику носять як зовнішній характер, так і внутрішній. Зовнішні фактори пов'язані з можливістю реалізації кредитного ризику через нездатність позичальника погасити заборгованість перед банком. В той час

як, внутрішні фактори пов'язані з помилками кредитних менеджерів, департаменту аналізу ризиків або інших співробітників, які були допущені в ході оформлення заявки або оцінки позичальника [6].

|  |  |
| --- | --- |
| **Внутрішні фактори** | **Характеристика факторів кредитного ризику** |
| *Фактори,* | * Зміст і умови комерційної діяльності позичальника |
| *пов'язані з* | * Кредитоспроможність позичальника |
| *діяльністю* | * Рівень менеджменту позичальника |
| *позичальника* | * Банкрутство позичальника |
|  | * Шахрайство з боку позичальника |
|  | * Репутація позичальника |
| *Фактори,* | * Адекватність вибору кредитної політики |
| *пов'язані з* | * Помилкові дії кредитних працівників |
| *діяльністю*  *банка- кредитора* | * Якість технологій * Кваліфікація персоналу * Тип ринкової стратегії |
|  | * Здатність розробляти і просувати |
|  |  |

Таблиця 1.2.Внутрішні чинники кредитного ризику

1.1.2 Проблема кредитного скорингу в Україні

Проблема кредитного скорингу в Україні досить актуальна на сьогодні. У кожного банку своя кредитна політика, але спільне завдання - мінімізувати ризики при видачі кредитів.

Зараз банки, в основному, вибирають консервативну політику, що, на наш погляд, є правильним рішенням, особливо після кризи 2008 року, коли в США іпотечні менеджери видавали позику практично всім підряд. В українській практиці існує два основних підходи оцінки ризику кредитування, які, зазвичай, застосовуються в поєднанні один з одним [7]:

• Суб'єктивне укладення кредитних менеджерів

• Автоматизовані системи скорингу.

Основним завданням цих двох підходів є з'ясування ймовірності повернення кредиту контрагентом. Скорингові системи в Україні більш зосереджені на використанні оцінки ризиків при кредитуванні фізичних осіб, ніж юридичних. Автор зазначає, що це пов'язано, перш за все, з труднощами оцінки фінансового стану підприємств, так як всі компанії різні за своєю діяльністю, працюють в різних секторах, і у кожної компанія різні масштаби. Для визначення ризиків кредитування юридичних осіб, крім скорингових систем, використовується моніторинг фінансового стану компанії шляхом оцінки вартості бізнесу та його активів [7]. Для роздрібного кредитування все набагато простіше, так як потенційні позичальники сегментируются на «хороших» і «поганих». В Україні банки стали використовувати скорингові моделі на початку нового століття. З високою конкуренцією і активним розвитком споживчого кредитування, без скорингових систем, банкам не вдасться конкурувати на ринку [8]. Кредитний скоринг розділяється на кілька типів. Далі ми розглянемо, які типи актуальні для українських банків:

• Application-скоринг. Application-скоринг - оцінка кредитоспроможності позичальників для отримання банківської позики по даних із заяви на кредит. Частка прострочення досягає 4,3% і продовжує зростати. Таким чином, можна сказати, Application-скоринг найбільш актуальний тип скорингу для України.

• Collection-скоринг. Останнім часом українські банки наголошують на необхідності збільшити використання Collection-скорингу. Цей тип скорингу допомагає проводити планомірну роботу з простроченою заборгованістю до моменту її передачі в колекторський відділ. Досвід показує, що значну частину заборгованості в ході цієї роботи вдається ліквідувати [9].

• Behavioral-скоринг. Behavioral-скоринг - це оцінка динаміки стану кредитного рахунку позичальника. Така модель може спрогнозувати зміни платоспроможності контрагента на підставі різних факторів. Поведінковий скоринг дозволяє визначити оптимальні для позичальника строки погашення, суми та ліміти.

• Fraud-скоринг. Спільно з вищеописаними типами скорингу і службою безпеки, банки використовують Fraud-скоринг, який дозволяє визначити ймовірність шахрайських дій з боку клієнта [9].

Хочеться відзначити, що з розвитком інформаційних технологій і появою автоматизованих скорингових систем, процес прийняття рішень про видачу кредиту став набагато зручніше, безпечніше і швидше. В умовах конкурентної боротьби на ринку кредитування неможливо уявити банк, який не використав би скорингові системи і інші методи управління кредитними ризиками. В даний час, підвищений попит позичальників на кредити зобов'язує банки використання політики управління кредитними ризиками .

Всякий ризик - це ймовірність, як сприятливого, так і негативного результату. Щоб не зазнати збитків банкам необхідно мінімізувати ризики, а скорингові системи дозволяють зробити це. Тому завдання магістерської роботи – освоїти методи побудови скорингу та покращити точність скорингової моделі.

* 1. Аналіз існуючих у світі способів її вирішення.

Скорингова модель - це методика оцінки кредитного ризику, яка дозволяє, оцінивши набір ознак, що характеризують позичальника, визначити, чи варто надавати йому кредит.

В основі скорингової системи лежить припущення, що люди зі схожими соціальними показниками поводяться однаково. Суть скорингу полягає в тому, що кожному параметру, що характеризує позичальника, надається реальна оцінка в балах.

## Побудова скорингової моделі проходить такі етапи:

## **C:\Users\User\Desktop\практика\Схема.PNG**

Рис.1.2.1. Етапи побудови скорингової моделі

Щоб побудувати скорингову модель, нам потрібно класифікувати клєнтів.

Тому розлянемо основні методи класифікації.

* + 1. Методи класифікації

Класифікація - це віднесення об'єктів до певного класу по набору ознак. Наприклад, розпізнавання номерів машин, або в медицині діагностика захворювань, або кредитний скоринг в банківській сфері. Цеодин з розділів машинного навчання, що присвячений вирішенням наступного завдання. Нехай ми маємо безліч об'єктів (ситуацій), розділених деяким чином на класи. Дано скінченну кількість об'єктів, для яких відомо, до яких класів вони належать. Ці об’єкти називається навчальною вибіркою. Класова приналежність інших об'єктів невідома. Потрібно побудувати алгоритм, здатний класифікувати довільний об'єкт.

Класифікувати об'єкт - значить, вказати номер (або найменування класу), до якого належить даний об'єкт.

На сьогодні існує 4 типи задач[10]:

* Навчання з учителем;
* Навчання без вчителя;
* Навчання з частковим залученням вчителя;
* Навчання з підкріпленням.

**Навчання з вчителем**

Навчання з учителем - це навчання на тренувальному наборі даних, шляхом підгонки результатів навчання на тренувальний набір даних.

Основне завдання - знайти найбільш оптимальні параметри моделі для прогнозування. Якщо результат прогнозу є дійсним числом, то це завдання регресії. Якщо має обмежену кількість значень, де ці значення є неврегульованими, то це завдання класифікації.

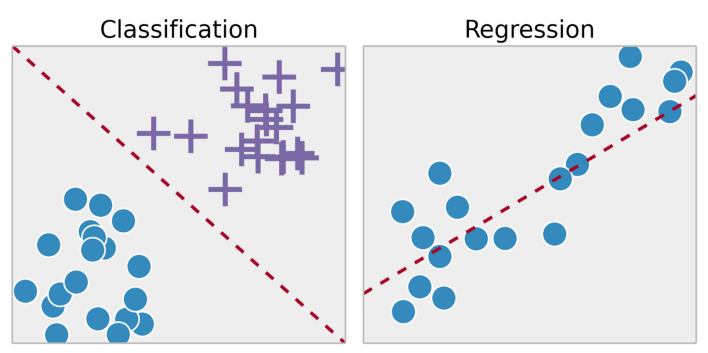


Рис1.2.2. Навчання із вчителем

**Навчання без вчителя**

У навчанні без контролю у нас менше інформації про об'єкти. Зокрема, тренувальний набір не має маркованих даних, що відносяться до певного класу заздалегідь зумовлених даних. Яка наша мета зараз? Спостерігати деяку схожість між групами об'єктів і включати їх до відповідних кластерів.

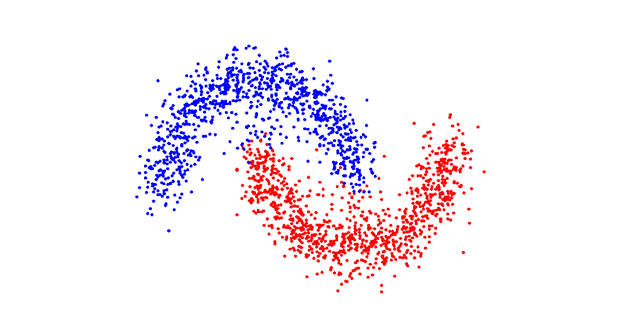


Рис1.2.3. Навчання без вчителя

**Навчання з частковим залученням вчителя**

Навчання з частковим залученням вчителя включає обидві проблеми, які ми описали раніше: вони використовують марковані і немарковані дані. Це відмінна можливість для тих, хто не може розподілити маркувати свої дані. Цей метод дозволяє значно підвищити точність, оскільки ми можемо використовувати немарковані дані в тренувальному наборі даних з невеликою кількістю маркованих даних.

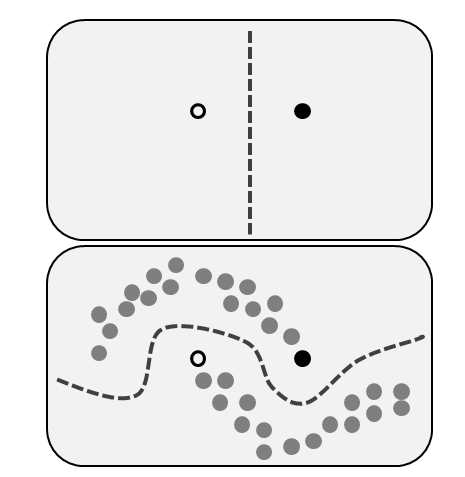


Рис1.2.4. Навчання з частковим залученням вчителя

**Навчання з підкріпленням**

Навчання з підкріпленням не схоже на будь-яку з наших попередніх завдань, тому що тут ми не маємо в своєму розпорядженні ні зумовленими маркованими даними, ні немаркованими наборами даних. Навчання з підкріпленням - область машинного навчання, пов'язана з тим, як агенти програмного забезпечення повинні вживати дії в деякому середовищі, щоб максимізувати деяке поняття кумулятивної нагороди.

Ідея навчання з підкріпленням полягає в тому, що система буде вчитися в середовищі, взаємодіяти з нею і отримувати винагороду за виконання дій.

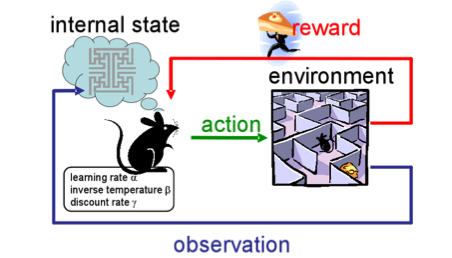


Рис.1.2.5. Навчання з підкріпленням

Уявіть, що ви робот в якомусь дивному місці. Ви можете виконувати дії і отримувати нагороди від навколишнього середовища за них. Після кожної дії поведінка навколишнього середовища стає більш складнішою тому ви тренуєтеся, щоб вести себе найбільш ефективним способом на кожному кроці. У біології це називається адаптацією до природного середовища.

Завдання класифікації відноситься до розділу навчання з учителем.

Основними методами класифікації для побудови скорингової моделі серед банків України:

- Дерева прийняття рішень.

- Логістична регресія;

1.2.3. Дерева прийняття рішень (С5.0, CART)

Алгоритм С5.0

Алгоритм C5.0 будує класифікатор в формі дерева рішень.

Припустимо, що у нас є набір даних - це дані про групу потенційних клієнтів банку. Ми знаємо різні параметри кожного пацієнта: вік, освіта, наявність страхування наявність нерухомості, історію сім'ї і так далі (це параметри). На підставі цих параметрів ми хочемо передбачити, чи може пацієнт отримати кредит. Пацієнт може потрапити в один з 2 класів: отримає та не отримає кредит. Алгоритму C5.0 повідомляє клас кожного клієнта використовуючи набір параметрів пацієнта і відповідний клас, C5.0 будує дерево рішень, здатне передбачити клас для нових пацієнтів на підставі їх параметрів.

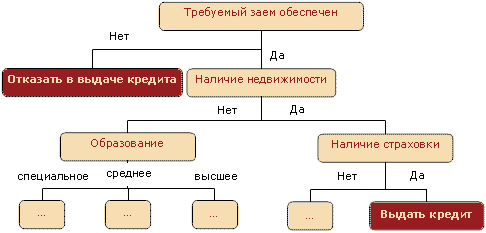


Рис.1.2.6. Дерево рішень

Класифікація методом дерева рішень створює своєрідні блок-схеми для розподілу нових даних. У кожній точці блок-схеми задається питання про значимість того чи іншого параметра, і в залежності від цих параметрів він або вона [клієнти] потрапляють в певний клас.

Задля того, щоб визначити найбільш інформативну змінну(змінна, що буде стояти на початковому етапі відбору) використовують принцип припливу інформації(Entropy and Information Gain).

Ентропія (Entropy) – це те, як багато інформації вам не відомо про стан об'єкта (або джерела інформації)[10].

Використовується, щоб визначити скільки інформації наявно у певному джерелі інформації.

У загальному випадку, якщо опиратися на теорію ймовірності, джерело інформації – це є ансамблі таких станів K = {k1, k2, ..., kN} , тобто ймовірність цих станів - {Р(k1), Р(k2), ..., Р(kN)} при цьому сума ймовірностей станів рівна 1.

Кількість інформації, яку ми можемо отримати з певного джерела інформації розраховується за формулою, яку запропонував К. Шеннон в 1946 році:

d

B(k) = -Σ P(k\_d) \* log2(P(k\_d)).

1

d - к-сть змінних,

P(k\_d) – ймовірність класу k змінної d.

Розрахуємо ентропію та знайдемо найбільш інформативні змінні.

Нехай клієнти банку розподілені таким чином:

Таблиця 1.2.1 Розподіл клієнтів банку

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Клієнт | Наявність  забезпечення | Наявність нерухомості | Наявність страхування |
| 1 | Good | yes | no | yes |
| 2 | Bad | no | yes | yes |
| 3 | Good | yes | yes | no |
| 4 | Bad | no | no | no |
| 5 | Good | no | no | no |
| 6 | Good | yes | no | yes |
| 7 | Bad | no | yes | no |

Обрахуємо ентропію в загальному випадку, тільки на наданих клієнта – «Good»/«Bad»:

B(Клієнт) = -p('yes')log2(p('yes')) - p('no')log2(p('no'))

B(Клієнт) = -(4/7)log2(4/7) - p(3/7)log2(3/7)

B(Клієнт) = 0.985228136034

Тож значення загальної ентропії - 0.985.

Щоб дізнатися порядок змінних у дереві застосовують критерій - Information Gain.

Gain(variable) = total\_entropy - remainder(variable)

branch\_d

remainder(variable) = Σ P(variable\_branch\_d)\*B(branch)

branch

total\_entropy – загальна ентроія;

B(branch) – ентропія змінної;

P(variable\_branch\_n) – ймовірність кожного класу змінної.

Обрахуємо Information Gain для змінної «Наявність забезпечення».

remainder(Наявність забезпечення) = P('yes, Good')\*B('yes') + P(‘yes, Bad')\*B('yes') + P('no, Good')\*B('no') + P('no, Bad')\*B('no').

remainder(Наявність забезпечення) = (2/7)\*B('yes') + (1/7)\*B('yes') + (2/7)\*B('no') + (2/7)\*B('no').

remainder(Наявність забезпечення) = 0.463587.

Gain(Наявність забезпечення) = 0.985228 - 0.463587 = 0.521641.

Для порівняння обраховуємо Information Gain для інших змінних:

Gain('Наявність страхування') = 0.020244;

Gain('Наявність нерухомості') = 0.128085.

Отже, змінна «Наявність забезпечення» буде розташована на «вершині» дерева.

Відмінності C5.0 від інших систем, що використовують дерева рішень:

o По-перше, C5.0 використовує приплив інформації, при створенні дерева рішень.

o По-друге, щоб уникнути перенавчання(overfiting) вибірки використовують метод 'pruning' - відсікання гілок, які збільшують похибку моделі.

o По-третє, C5.0 може працювати з дискретними і неперервними значеннями. Обмежуючи діапазони і встановлюючи пороги даних, трансформуючи неперервні дані в дискретні.

Ймовірно, найбільшим гідністю дерев рішень є їх проста інтерпретація, також вони мають досить високу швидкість роботи, а отримані дані легкі для розуміння.

Алгоритм CART

CART (classification and regression trees) - це абревіатура, що позначає методи класифікації і регресії з використанням дерева рішень. Це методика навчання, заснована на деревах рішень, яка повертає класифікаційні або регресивні дерева.

Наприклад, знову візьмемо набір даних про клієнта. Ви можете спробувати передбачити, чи отримає клієнт кредит. Тут можливе використання двох класів: «Good» - отримає, «Bad» - не отримає.

Задля того, щоб визначити найбільш інформативну змінну(змінна, що буде стояти на початковому етапі відбору) у методі CARD використовують Gini Impurity .

d

G(k) = Σ P(k\_d) \* (1 - P(k\_d))

i=1

d - к-сть змінних,

P(k\_d) – ймовірність класу k змінної d.

Застосуємо Gini Impurity на нашому прикладі:

G(Клієнт) = P('yes') \* (1 - P('yes')) + P('no') \* (1 - P('no'));

G(Клієнт) = 4 / 7 \* (1 - 4/7) + 3 / 7 \*( 1 - 3/7);

G(Клієнт) = 0.489796.

Gini Impurity - це показник того, як часто рандомним чином вибраний елемент з набору буде неправильно позначений, тобто у нашому випадку неправильно класифікованих об`єктів буде ~ 49%.

Визначення найбільш інформативної змінної робиться аналогічним чином, як обчислення інформації для ентропії, за винятко, що замість того, щоб взяти зважену суму ентропій кожної змінної, ми беремо зважену суму Gini Impurity.

Gini\_Gain(variable) = total\_impurity - impurity\_remainder(variable)

branch\_d

remainder(variable) = Σ P(attribute\_branch\_d)\*G(branch)

branch

total\_impurity – загальний показник Gini Impurity;

G(branch) – Gini Impurity змінної;

P(variable\_branch\_d) – ймовірність кожного класу змінної.

Таблиця 1.2.2. Співвідношення CART і C.4.5

|  |  |
| --- | --- |
| **C5.0** | **Cart** |
| Для оцінки точності моделі в процесі класифікації використовується приплив інформації(Entropy and Information Gain) до сегменту даних. | Для оцінки точності моделі в процесі класифікації використовується критерій Джині(Gini Impurity) |
| Використовує однопрохідний метод проріджування, щоб зменшити перенавчання. | Починаючи знизу дерева, CART оцінює помилку класифікації в вузлі і поза вузла. Якщо похибка перевищує граничну, то гілка відкидається |
| Вузли дерева рішень можуть мати дві або більше гілок | Вузли рішення мають дві гілки |

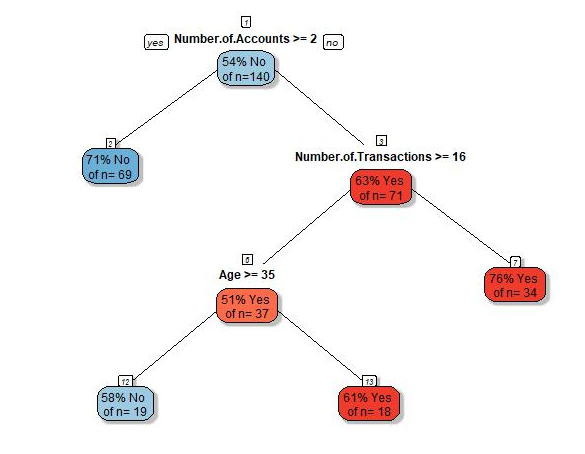


Рис.1.2.7. приклад побудови дерева рішень на основі алгоритму CART

Як і C5.0, CART досить швидкий та отримані дані легкі для розуміння.

Логарифмічна регресія

Лінійна регресійна модель не завжди здатна якісно передбачати значення залежної змінної. Вибираючи для побудови моделі лінійне рівняння, ми природним ніяк не накладаємо ніяких обмежень на значення залежної змінної. А такі обмеження можуть бути істотними. Наприклад, при проектуванні оптимальної довжини шахти ліфта в новій будівлі необхідно врахувати, що ця довжина не може перевищувати висоту будівлі взагалі.

Лінійна регресійна модель може дати результати, несумісні з реальністю. З метою вирішення даних проблем корисно змінити вид рівняння регресії і підстроїти його для вирішення конкретного завдання.

Логарифмічна регресія - це різновид множинної регресії, призначення якої полягає в аналізі зв'язку між декількома незалежними змінними (предикторами) і залежною змінною. За допомогою бінарної логістичної регресії можна оцінювати вірогідність того, що подія настане для конкретного випробуваного (повернення кредиту / дефолт).

Математична основа логістичної регресії

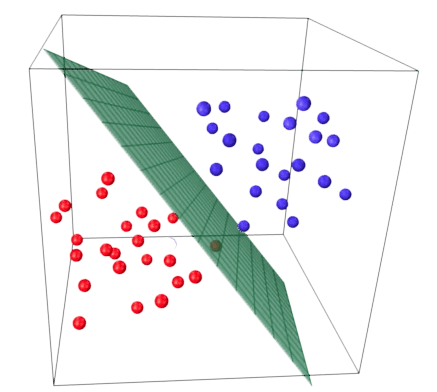
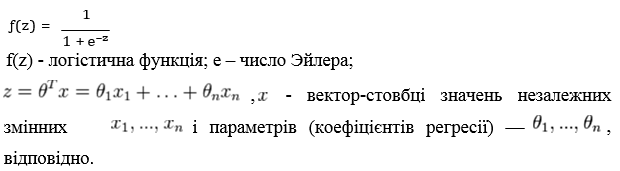


Рис.1.2.8. Класифікація за допомогою логарифмічної регресії

Завдання регресії може бути сформульована інакше: замість передбачення бінарної змінної ми передбачаємо неперервну змінну зі значеннями на відрізку [0,1] при будь-яких значеннях незалежних змінних. Це досягається

застосуванням наступного регресійного рівняння (логіт-еквіваленті):



Залежність, що зв'язує ймовірність події і величину y, показана в наступній діаграмі :

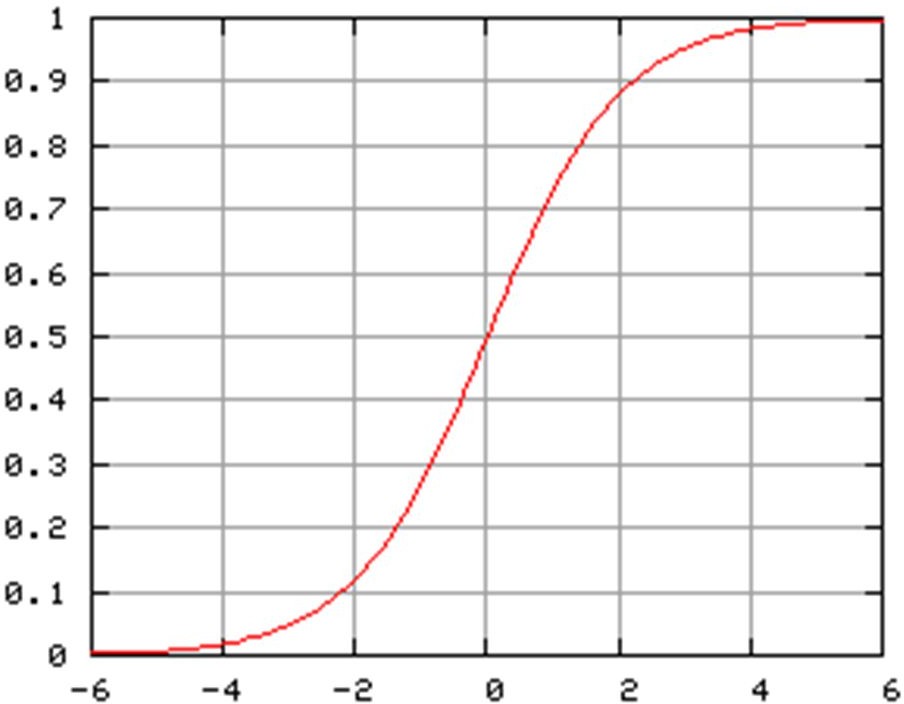


Рис1.2.9. Логарифмічна функція

Легко побачити, що незалежно від регресійних коефіцієнтів чи величин х, передбачені значення (у) в цій моделі завжди будуть лежати в діапазоні від 0 до 1.

Логістична регресія – один з найпопулярніших методів класифікації, адже цей метод досить точно і швидко працює з великим об’ємом вибірки. Також при вирішенні задач класифікації об'єкти можна розділяти на кілька груп. Наприклад, не тільки - (0 - поганий, 1 - хороший), а й кілька груп (1, 2, 3, 4 групи ризику).

Проте є недоліки при роботі із малою кількістю спостережень у вибірці. Коли розмірніcть вибірки(кількість спостережень) < 500, то можливе завищення оцінки коефіцієнтів регресії.

Присутні обов’язкові умови до вибірки при побудові моделі. Потрібно мінімально 10 спостережень на кожну незалежну змінну (рекомендоване значення 30-50):

Наприклад, смерть пацієнта. Якщо 50 пацієнтів з 100 вмирають, то максимальне число незалежних змінних в моделі = 50/10 = 5.

* 1. Визначення вхідних даних для вирішення поставленої задачі.

Вибір і побудова моделі, впровадження кредитного скорингу та його застосування, безумовно, складна і трудомістка задача. В умовах існування кредитних бюро завдання дещо спрощується щодо позичальників, які ще не

були клієнтами конкретного банку, однак існують заявники, які не зверталися до послуг кредиторів раніше. Тому цілком природно існує практика окремо оцінювати заявників, що мають і не мають кредитну історію.

Особливу увагу варто звернути на дані, за якими навчається і оцінюється модель. З одного боку, вибірка повинна бути репрезентативними, тому в них відображені як хороші, так і погані кредитні ризики.

Для прикладу була використана тестова вибірка даних. За початкові дані були взяті репрезентативна вибірка банку АТ «Альфа-Банк».

Вибірка складається із 20 змінних:

1. Стан поточного рахунку
2. Тривалість кредиту
3. Кредитна історія
4. Мета кредиту
5. Сума кредиту
6. Ощадний рахунок / облігації
7. Стаж поточної роботи
8. Ставка внеску (у відсотках від наявного доходу)
9. Сімейний статус і стать
10. Відповідальна особа
11. Нинішня резиденція
12. Тип власності
13. Вік
14. Заявник має інший кредит на розстрочку
15. Тип власності на дім
16. Кількість кредитів у цьому банку
17. Тип зайнятості
18. Кількість людей, які несуть відповідальність за кредит
19. Номер телефону(Є/ Немає)
20. Іноземний працівник (Так / Ні)

1.4. Висновки

На даний момент ринок інформаційних рішень скоринга ще не досить розвинутий в Україні.

Логістична регресія - достатньо поширений метод в кредитному скорингу.

Вона менш чутлива до розміру вибірки у порівнянні з багатьма іншими методами, що застосовуються для класифікації кредитного скоринга. Наразі існує кілька методів її вдосконалення – лассо регресія, яка «регулює» змінні, які входять у модель.

Також для класифікації поганих і хороших кредитів використовуються дерева рішень. В залежності від алгоритму породження дерева (C5.0, CART) метод має деякі переваги, серед яких простота в інтерпретації і розумінні, менша необхідність в передній обробці даних (може працювати одночасно зі змінними, які оцінені за різними шкалами, не вимагає нормування або заповнення пропусків в даних). Проте зараз існують нові методи, основані на ідеї дерева рішень, які показують кращу точність – метод випадкоаих лісів та градієнтний бустинг.

# РОЗДІЛ 2 МЕТОДИ ТА ЗАСОБИ ОБРОБКИ ІНФОРМАЦІЇ

2.1 Первинна обробка даних для побудови моделі. Методи обробки та аналізу даних

Дано репрезентативну вибірку клієнтів банку(1000 клієнтів, 20 змінних).

\* процес «Data Preparation» реалізований на мові R, скрипт знаходиться у додатку до магістерської роботи.

Підготовка даних - це не лише перший крок, а й багато повторних кроків(алгоритмів) протягом аналізу, коли з'являються нові проблеми або з'являються нові дані. Тому не дивно, що є поширена думка, що до 80% процесу аналізу даних - це час, витрачений на їх підготовку[12].

Дані для вибірки можуть мати неточності, «викривлені» значення та ін. Щоб зробити модель більш точною наведемо кілька прикладів вдосконалення перинної структури даних.

2.1.1. Заміна пропущених значеннь

Часто в даних, з якими необхідно працювати, присутні пропуски, в результаті чого аналітик виявляється перед вибором: відкинути або ж заповнити пропущені значення. Взагалі існує кілька варіантів боротьби із пропущеними значеннями:

• якщо змінна має занадто багато відсутніх значень, то видаляємо її;

• Якщо у змінної дуже мало пропущених значень, то краще видалити клієнтів, у яких є пропущені значення;

• Якщо видалення клієнтів з відсутніми значеннями зменшить суттєво розмір вибірки, то заміняємо прощущені значення.

Заповнення пропусків часто, і цілком обґрунтовано, здається більш привабливим рішенням. Однак це не завжди так. Невдалий вибір методу заповнення пропусків може не тільки не поліпшити, а й сильно погіршити результати моделі.

Виняток ігнорування рядків з пропущеними значеннями стало рішенням за замовчуванням в деяких популярних прикладних пакетах, в результаті може виникнути уявлення, що дане рішення - правильне. Крім того, існують досить прості в реалізації і використанні методи обробки пропусків, що отримали назву «ad-hoc методи»: заповнення пропусків нулями, медіаной, середнім арифметичним значенням, введення індикаторних змінних тощо, простота яких може послужити причиною для вибору саме цих методів .

Ймовірно, саме через свою простоту ad-hoc методи широко використовувалися на зорі розвитку сучасної теорії обробки пропусків. І хоча за станом на сьогоднішній день відомо, що застосування цих методів може призводити до спотворення статистичних властивостей вибірки і, як наслідок, до погіршення результатів, одержуваних після такої обробки пропусків [13], їх як і раніше часто використовують. Так, відомі статті, присвячені збору та оцінки статистики використання методів заповнення пропусків в наукових роботах медичної тематики [14], з результатів яких можна зробити висновок, що навіть вчені часто віддають перевагу інтуїтивно-зрозумілим ad -hoc методам і ігнорування / видалення рядків, незважаючи на те, що застосування цих методів в контексті розв'язуваної задачі часом недоречно.

Механізми формування пропусків

Для того щоб зрозуміти, як правильно обробити пропуски, необхідно визначити механізми їх формування.

Розрізняють такі 3 механізму формування пропусків: MCAR, MAR, MNAR.

MCAR (Missing Completely At Random) - механізм формування пропусків, при якому ймовірність пропуску для кожного запису набору однакова. Наприклад, якщо проводилося соціологічне опитування, в якому кожному десятому респонденту одне рандомно вибране питання не ставилося, причому на всі інші поставлені запитання респонденти відповідали, то тут має місце механізм MCAR. В такому випадку ігнорування / виключення записів з даних не призведе до спотворення результатів.

MAR (Missing At Random) - на практиці дані зазвичай пропущені не випадково, а з огляду на деяких закономірностей. Пропуски відносять до MAR, якщо ймовірність пропуску може бути визначена на основі іншої наявної в наборі даних інформації (стать, вік, займана посада, освіта ...), яка не містить пропуски. В такому випадку видалення або заміна пропусків на значення «Пропуск», як і в разі MCAR, не призведе до суттєвого спотворення результатів.

MNAR (Missing Not At Random) - механізм формування пропусків, при якому дані відсутні в залежності від невідомих чинників. MNAR передбачає, що ймовірність пропуску могла б бути описана на основі інших атрибутів, але інформація по цим атрибутам в наборі даних відсутня. Як наслідок, ймовірність пропуску неможливо виразити на основі інформації, що міститься в наборі даних.

Розглянемо відмінності між механізмами MAR і MNAR на прикладі.

Люди, що займають керівні посади і / або які отримали освіту в престижному вузі частіше, ніж інші респонденти, які не відповідають на питання про свої доходи. Оскільки посада і освіту сильно корелюють з доходами, то в такому випадку пропуски в поле доходи вже не можна вважати абсолютно випадковими, тобто говорити про випадок MCAR не представляється можливим.

Якщо в наборі даних є інформація про освіту і посади респондентів, то залежність між підвищеною ймовірністю пропуску в графі доходів і цією інформацією може бути виражена математично, отже, виконується гіпотеза MAR. У разі MAR виключення пропусків цілком прийнятно.

Однак якщо інформація про займану посаду та освіті у нас відсутній, то тоді має місце випадок MNAR. При MNAR просто ігнорувати або виключити пропуски вже не можна, так як це призведе до значного спотворення розподілу статистичних властивостей вибірки.

Розглянемо прості методи обробки пропусків і пов'язані з ними проблеми.

Видалення / ігнорування пропусків

Complete-case Analysis (Listwise Deletion Method) - метод обробки пропусків, який застосовується в побудові моделі як метод за замовчуванням. Полягає у виключенні з набору даних записів / рядків або атрибутів / колонок, що містять пропуски.

У разі першого механізму пропусків (MCAR) застосування даного методу не призведе до суттєвого спотворення параметрів моделі. Проте видалення рядків призводить до того, що при подальших обчисленнях використовується не вся доступна інформація, стандартні відхилення зростають, отримані результати стають менш репрезентативними. У випадках коли пропусків в даних багато, це стає відчутною проблемою.

Крім того, в разі другого (MAR) і, особливо, третього механізму пропусків (MNAR) зміщення статистичних властивостей вибірки, значень параметрів побудованих моделей і збільшення стандартних відхилень стають ще сильніше.

Таким чином, незважаючи на широке поширення, застосування даного методу для вирішення практичних завдань обмежена.

Available-case analysis (він же Pairwise Deletion) - методи обробки, заснований на ігноруванні пропусків в розрахунках. Ці методи, як і Complete-case Analysis, теж часто застосовуються за замовчуванням.

Статистичні характеристики, такі як середні значення, стандартні відхилення, можна розрахувати, використовуючи всі непропущені значення для кожного з атрибутів / стовпців. Як і в разі Complete-case Analysis, за умови виконання гіпотези MCAR, застосування даного методу не призведе до суттєвого спотворення параметрів моделі.

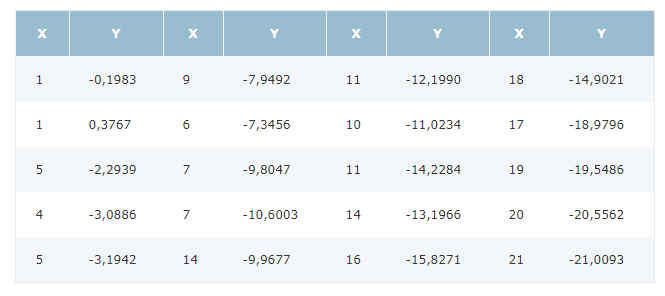
Перевага даного підходу в тому, що при побудові моделі використовується вся доступна інформація.

Головний же недолік даних методів полягає в тому, що вони можуть бути застосовані для розрахунку далеко не всіх показників і, як правило, пов'язані з алгоритмічними і обчислювальними труднощами, що призводять до некоректних результатів.

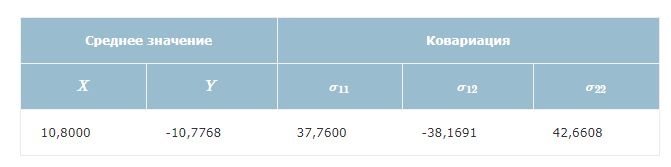
Наприклад, розраховані значення коефіцієнтів кореляції можуть опинитися поза діапазону [-1; 1]. Крім того, не завжди вдається однозначно відповісти на питання про оптимальний вибір числа змінних у моделі, якщо ми хочемо обрати оптимальне число змінних за допомогою методів, які засновані на розрахунку стандартних відхилень.

Наведемо приклад, який демонструє проблеми методів Available-case analysis.

Розглянемо наступну задачу: необхідно розрахувати лінійний коефіцієнт кореляції (коефіцієнт кореляції Пірсона) між двома факторами / змінними X і Y.



Таблиця 2.1.1. Значення для аналізу кореляції Пірсона

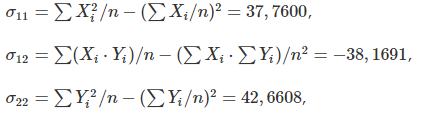


Таблиця 2.1.2. Значення кореляції Пірсона

Середнє значення X: 10,8000.

Середнє значення Y: -10,7768.

Оцінка (ко) варіації:



де n - число спостережень (n = 20).

Значення коефіцієнта кореляції:

F:\Дипломна робота_2\картинки\4_4.PNG

Розглянемо результати аналогічних розрахунків при наявності пропусків в даних (дані представлені в таблиці 2).

Таблиця 2.1.3. Дані із пропущеними значеннями

Тобто працюємо з тим же набором даних (що і в таблиці 2.1.1), з тією лише відмінністю, що в даному випадку нам невідомі два перших значення змінної X.

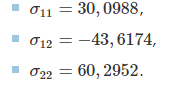
В рамках Available-case analysis підходу ми вважаємо середнє значення, використовуючи всю доступну інформацію, тобто для змінної X на основі 18 відомих значень, а для змінної Y на основі всіх 20 значень.

Таким чином, на основі таблиці 2 отримаємо наступні результати:

cреднее значення X: 11,8889,

cреднее значення Y: -10,7768,

оцінка (ко) варіації:



Значення коефіцієнта кореляції:

r = -1,0239.

Таким чином, розрахунок середнього значення на основі підходу Available-case Analysis призвела до зміщення даного значення, що в свою чергу, проявилося в розрахованому значенні коефіцієнта кореляції меншим -1. Таким чином, розраховане значення вийшло за межі теоретично можливого діапазону [-1; 1]), що суперечить фізичному змісту.

Якщо ж розрахувати значення коефіцієнта кореляції в рамках підходу Complete-case Analysis, то отримаємо значення коефіцієнта кореляції: -0,9311.

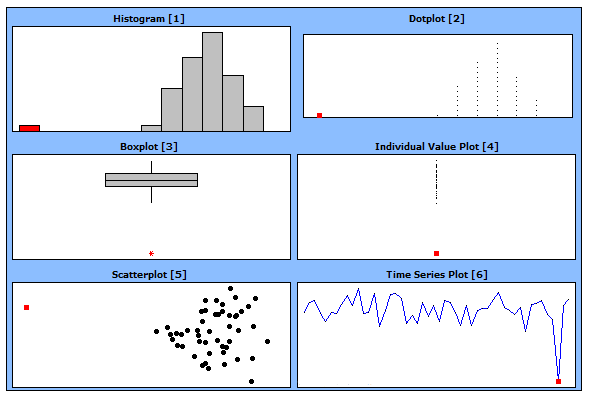
Коли гіпотеза MCAR не виконується, методи Available-case analysis так само, як і методи Complete-case Analysis призводять до суттєвих перекручень статистичних властивостей вибірки (середнього значення, медіани, варіації, кореляції).

До недоліків перших двох методів обробки пропусків (Complete-case Analysis і Available-case analysis) відноситься і те, що, далеко не завжди виключення рядків прийнятно. Нерідко процедури подальшої обробки даних припускають, що всі рядки і колонки беруть участь в розрахунках (наприклад, коли пропусків в кожному стовпчику і в рядках небагато).

Тобто, це було очевидно, але ми переконалися, що вставляти пропущені значення потрібно, проте потрібно вставляти їх покладаючись а певні залежності в даних.

2.1.2. Викиди (надто малі або надто великі значення)

Викид - це результат вимірювання, що виділяється із загальної вибірки. Іншими словами, викиди - це незвично низькі або високі значення спостережуваної величини, причому настільки, що це помітно неозброєним оком: в ході графічного аналізу спостережень ви можете помітити значення, яке не належить популяції спостережень. Визначити викиди можна за допомогою: гістограм [1], точкових [2] і ящикових [3] діаграм, діаграм індивідуальних значень [4], розсіювання [5] і навіть діаграм часових рядів [6]:



Таблиця 2.1.4. Різне представлення викидів

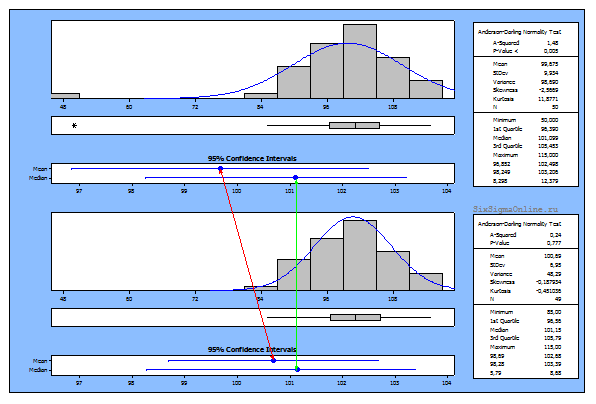
Червоні точки, зірочки і стовпці на діаграмах відповідають викидам.

В теорії статистичного аналізу немає однозначного критерію ідентифікації викидів, і це - перша причина, по якій викиди становлять небезпеку.

З визначення випливає, що всі незвично низькі або високі значення спостережуваної величини можуть бути викидами. Як же визначити, яке значення змінної є надзвичайно високим або низьким. Один з найпростіших способів: використовувати діапазон трьох стандартних відхилень навколо середнього значення. Ймовірність виходу величини за межі ± 3σ становить 0,0027, а значить, з великою часткою ймовірності, значення, яке виходить за межі ± 3σ не належить до популяції.

З іншого боку, можна привести ряд доводів проти цього твердження. Наприклад, воно втрачає сенс, якщо функція розподілу відрізняється від нормальної або розмір вибірки занадто малий, щоб представити генеральну сукупність значень. Крім того, з імовірністю 0,0027 спостереження все ж може вийти за межі діапазону трьох стандартних відхилень.

Друга небезпека, яку представляють викиди - спотворення статистик або результатів статистичних розрахунків. Такі показники як середнє арифметичне (Mean), стандартне відхилення (StDev), асиметрія (Skewness), ексцес (Kurtosis), а також критерій згоди з нормальним законом вельми схильні до впливу викидів. На відміну від середнього арифметичного, медіана менш схильна до впливу викидів. На наступному малюнку медіана і середнє арифметичне до і після виключення викиду позначені зеленою і червоною стрілками відповідно:



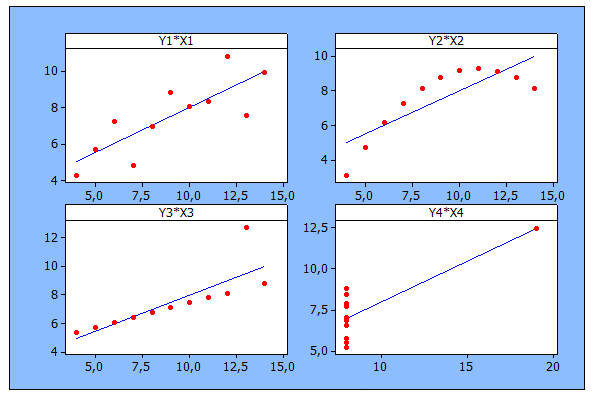
Таблиця 2.1.5 Вплив викидів на статистичну оцінку даних

Ще один класичний приклад - квартет Енскомба (Anscomb):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **X1** | **Y1** | **X2** | **Y2** | **X3** | **Y3** | **X4** | **Y4** |
| 10 | 8,04 | 10 | 9,14 | 10 | 7,46 | 8 | 6,58 |
| 8 | 6,95 | 8 | 8,14 | 8 | 6,77 | 8 | 5,76 |
| 13 | 7,58 | 13 | 8,74 | 13 | 12,74 | 8 | 7,71 |
| 9 | 8,81 | 9 | 8,77 | 9 | 7,11 | 8 | 8,84 |
| 11 | 8,33 | 11 | 9,26 | 11 | 7,81 | 8 | 8,47 |
| 14 | 9,96 | 14 | 8,1 | 14 | 8,84 | 8 | 7,04 |
| 6 | 7,24 | 6 | 6,13 | 6 | 6,08 | 8 | 5,25 |
| 4 | 4,26 | 4 | 3,1 | 4 | 5,39 | 19 | 12,5 |
| 12 | 10,84 | 12 | 9,13 | 12 | 8,15 | 8 | 5,56 |
| 7 | 4,82 | 7 | 7,26 | 7 | 6,42 | 8 | 7,91 |
| 5 | 5,68 | 5 | 4,74 | 5 | 5,73 | 8 | 6,89 |

Таблиця 2.1.6. Дані квартету Енскомба

Квартет Енскомба - це чотири набори числових даних, які використовують як свідчення важливості візуальної оцінки спостережень в кореляційному і регресійному аналізі:



Таблиця 2.1.7. Приклади кореляції

Не дивлячись на відмінності взаємозв'язку змінних X і Y, у всіх чотирьох випадках статистичні показники, як і рівняння лінійної регресії, однакові:

**Характеристика Значення**

Середнє значення змінної X 9

Дисперсія змінної X 10

Середнє значення змінної Y 7,5

Дисперсія змінної Y 3,75

Коефіцієнт кореляції Пірсона 0,816

Рівняння лінійної регресії Y = 3 + 0.5X

Викид в третьому прикладі спотворює рівняння залежності, а в четвертому - змушує прийняти рішення про наявність кореляції, в той час як її насправді немає.

І, нарешті, третя небезпека, яку приховують викиди - це легкість їх невірного тлумачення, що, в свою чергу, призведе до невірного напрямку подальшого аналізу. Наявність викидів може означати помилку введення даних, недостатню величину вибірки або присутність спеціальної причини відхилення - дія конкретного фактора або причини. Діагностуючи викиди, легко допустити помилку, виключивши потрібні для аналізу дані або навпаки - розрахувавши показники процесу, використовуючи неправильні результати спостережень

Неуважне ставлення до викидів спостережень ставить під загрозу висновки про спостереження процесу і ставить під загрозу результати подальшого аналізу. Отже, виявивши незвично низькі або високі значення спостережуваної величини, дослідник повинен визначити причину їх появи, перш ніж робити висновки про спостерігається змінної або приступати до подальшого аналізу даних.

Методи боротьби із викидами:

* якщо виключити викиди із змінної, то це призведе до змін середнього значення та дисперсії. Враховуючи, що викиди також вважаються значеннями, виключення їх з аналізу робить цей підхід неадекватним для «регулювання» змінної;

## встановити всі відхилення до заданого процентилу даних; наприклад, 98% - це всі дані без викидів. Остальні: 1% - дані з надто малими значеннями, інший 1% - з надто великими значеннями. Ці дані з надто малими та великими значеннями замінюємо на найменші та ,відповідно, найбільші значення 98% даних без викидів.

Метод бінінгу використовують, щоб не змінювати цілісність даних при цьому позбавитися проблеми із викидами.

2.1.3. Бінінг

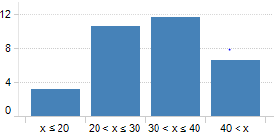
В скоринговії моделі можуть використовуватися в якості незалежних змінних категоріальні і кількісні предиктори. Багато розробників скорингових систем використовують завжди метод категоризації кількісних змінних(binning).

Рис. 2.1.2. Змінна «Вік» із застосуванням методу бінінгу

Категоризація кількісних змінних дозволяє досягти таких основних переваг при побудові скорингової карти: полегшити обробку викидів та екстремальних значень кількісних змінних; спростити інтерпретацію скорингової карти; відобразити складні нелінійні зв'язки. Розглянемо основні методи бінінгу, які застосовуються у скорингу на основі змінної «Вік» (age\_in\_yrs – вік, 1 о.в. – 1 рік).Розглянемо емпіричний розподіл змінної(без бінінгу):

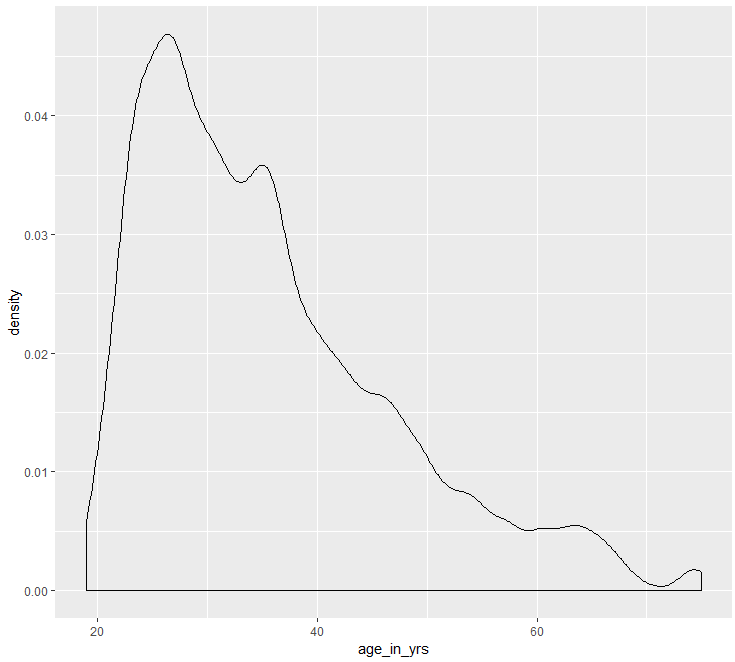
****

Рис.2.1.3. Емпіричний розподіл змінної «age\_in\_yrs»

## Методи бінінгу:

Типові методи, які широко застосовуються у побудовах моделі;

*Equal-width (distance):*

## ділить діапазон на N інтервалів однакового розміру;

## якщо A і B є найнижчими і найвищими значеннями змінної, то ширина інтервалів буде: W = (B-A) / N;

## чутливий до відхилень, нерівні дані не обробляються добре.

## **C:\Users\User\Desktop\практика\eq_width.PNG**

Рис.2.1.4. Застосування «Equal-width (distance)» на змінній «age\_in\_yrs»

## *Equal-depth (frequency):*

## Він ділить діапазон на N інтервалів, кожен з яких містить приблизно однакову кількість спостережень;

## Гарне масштабування даних.

## **C:\Users\User\Desktop\практика\eq_depth.PNG**

Рис.2.1.4. Застосування «Equal-width (frequency)» на змінній «age\_in\_yrs»

## Проте можна застосувати нові методи бінінгу:

## *“Jenks Natural Breaks” method:*

## метод почергово перебирає всі можливі варіанти класів, рахуючи при цьому міжкласову та міжгрупову дисперсію, отже, вимагає затратів по часу;

## ідея методу полягає у мінімізації дисперсії в класах і максималізації міжкласової дисперсії.

## **C:\Users\User\Desktop\практика\jenks.PNG**

Рис.2.1.5. Застосування «Jenks Natural Breaks» на змінній «age\_in\_yrs»

## Ще одним методом бінінгу, метод кластерного аналізу, - *K-means.*

## Jenks та K-means відрізняються тим, як вони мінімізуються в межах відстаней групи.

## Jenks використовує той факт, що одновимірні дані сортуються, що робить його більш швидким алгоритмом для одновимірних даних.

## K-means є більш загальними алгоритмом, він ефективний із даними великої розмірності.

## Надалі розглянемо методи обробки інформації за допомогою методу *K-means* та застосуємо метод на наших даних.

## Мета методу — розділити n спостережень на k кластерів, так щоб кожне спостереження належало до кластера з найближчим до нього [середнім значенням](https://uk.wikipedia.org/wiki/Середнє_арифметичне). Метод базується на мінімізації суми квадратів відстаней між кожним спостереженням та центром його кластера, тобто функції

,

де d — метрика, xi{\displaystyle x\_{i}} — і-ий об'єкт даних, а {\displaystyle m\_{j}(x\_{i})}mj(xi) — центр кластера, якому на j-ій ітерації приписаний елемент {\displaystyle x\_{i}} xi{\displaystyle x\_{i}} .

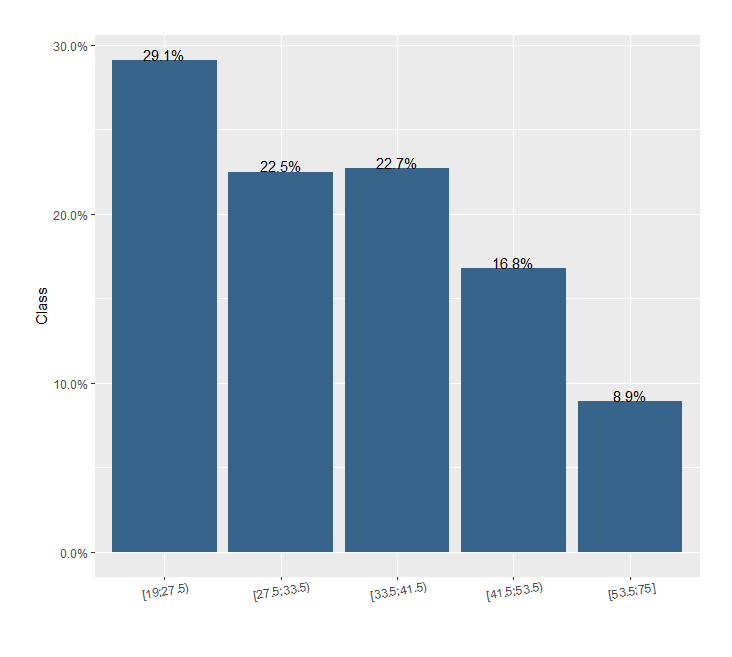


Рис.2.1.5. Застосування методу «K-means» на змінній «age\_in\_yrs»

Ми провели ознайомлення із методами обробки інформації. Надалі проведемо аналітичну діяльність над магістерським завданням та протестуємо запропоновані методи аналізу інформації.

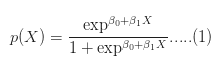
2.2. Розробка та аналіз методів, які можуть покращити прогностичну властивість скорингової моделі

2.2.1. LASSO регресія

LASSO (least absolute shrinkage and selection operator - потужний метод оцінки коефіцієнтів регресії. Лассо є по суті методом регуляризації. Даний метод зменшує перенавчання, використовуючи менш складні функції . Один із способів зробити це - викинути менш важливі змінні після перевірки того, що вони не є важливими. Це можна зробити, вивчивши p-значення коефіцієнтів і відкинувши ті змінні, коефіцієнти яких не є значущими. Проте, це займає багато часу для проблем класифікації з багатьма незалежними змінними. У таких ситуаціях лассо пропонує спосіб моделювання залежної змінної при автоматичному виборі значущих змінних шляхом зменшення коефіцієнтів неважливих змінних до нуля.

Як діє логістична регресія із використанням Лассо-регулізації?

Припустимо, що результат (передбачена змінна) і предиктори позначаються відповідно Y і X, і два класи, що представляють інтерес, позначаються + і - відповідно. Ми хочемо моделювати умовну ймовірність того, що результат Y дорівнює +, враховуючи, що вхідними змінними (предикторами) є X. Умовна ймовірність позначається p (Y = + | X), яку ми будемо скорочувати як p(X):



Можна переконатися, що p(X) дійсно лежить між 0 і 1, оскільки X змінюється від - ∞ до + ∞. На малюнку також показано типову S-подібну криву, характерну для логістичної регресії.

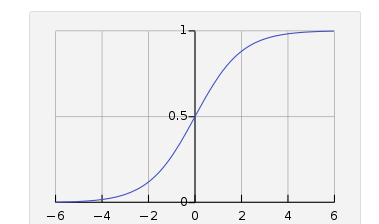


Рис.2.2.1. Приклад логістичної регресії

З’ясуємо звідки походить назва «логістична». Еквівалентним способом вираження наведеного рівняння є:

F:\Дипломна робота_2\картинки\модель\3.PNG

Значення ліворуч - логарифм шансів. Отже, модель являє собою лінійну регресію логарифмічних коефіцієнтів, іноді званий logit, і звідси назва логістична.

Проблема полягає в тому, щоб знайти значення beta\_0 і beta\_1 і т.д., які призводять до p(X), що найбільш точно класифікує всі спостережувані точки даних - тобто ті, які належать до позитивного класу, мають ймовірність максимально наближеною до 1 і ті, що належать до негативного класу, мають ймовірність максимально близьку до 0. Один із способів сформулювати цю проблему:

F:\Дипломна робота_2\картинки\модель\4.PNG

Де представлені продукти над i та j, які проходять над + і - класифікованими точками відповідно. Такий підхід, який називається оцінкою максимальної правдоподібності, досить часто зустрічається у багатьох налаштуваннях машинного навчання, особливо тих, що стосуються ймовірностей.

Більше того, мінімізується ймовірність негативного логарифму правдоподібності, що, звичайно, є таким же, як максимізація ймовірності правдоподібності. Мінімізується таким чином:

F:\Дипломна робота_2\картинки\модель\5.PNG

Проте, це теоретичні деталі, які я згадую тільки для повноти. Як ви побачите далі, вони мало впливають на практичне використання логістичної регресії із лассо регулізацією.

Lasso регресія

Отже, значення коефіцієнтів логістичної регресії beta\_0 і beta\_1 і т.д. виявляються шляхом мінімізації ф-ції правдоподібності ,описаного в рівнянні (3). Ще одним поширеним методом регуляризаціє є рідж регуляризацяя. Рідж і лассо регуляризації працюють шляхом додавання штрафу на функцію правдоподібності. У разі рідж регресії термін штрафу – це сума «beta» коефіцієнтів у квадраті, а у випадку ласо - це сума «beta» коефіцієнтів взятих по модулю .Тобто для рідж резуляризації:

F:\Дипломна робота_2\картинки\модель\6.PNG

Для лассо:

F:\Дипломна робота_2\картинки\модель\7.PNG

Де лямбда - це вільний параметр, який зазвичай вибирається таким чином, щоб отримана модель мінімізувала помилку вибірки.

У разі рідж регресії ефект регулізації полягає в зменшенні коефіцієнтів, які найбільше впливають на помилку моделі. Іншими словами, це зменшує величину коефіцієнтів, що сприяють збільшенню L. На відміну від цього, у випадку регресії лассо, ефект штрафу проявляється у зменшенні цих коефіцієнтів практично до нуля! Це добре, тому що це означає, що регресія лассо працює як селектор змінних, тобто вибирає найважливіші змінні, ті, які мають найбільшу прогностичну силу(мають найменші значення p).

Ми розглянули оптимізацію логістичної регресії, одну з найпростіших (і найстаріших) методів класифікації в арсеналі практикуючих методів. Незважаючи на простоту (або, треба сказати, через це), логістична регресія добре працює для багатьох бізнес-проблем, які часто мають прості рішення. Більш того, через її простоту вона менш схильна до перенавчання, ніж методи, побудовані на деревах рішень. Але даний метод можна оптимізувати усуненням змінних, які сприяють перенавчанню, змінні будуть усунені за допомогою лассо регуляризації, без погіршення точності. Враховуючи ці переваги та властиву їй простоту, не дивно, що логістична регресія залишається одним із найпопулярніших методів для скоронгових моделей.

2.2.2. Випадковий ліс

Bagging - використовує паралельне навчання базових класифікаторів. В ході беггінга відбувається наступне:

З безлічі вихідних даних випадковим чином відбирається кілька підмножин, що містять кількість прикладів, що відповідає кількості прикладів вихідного.

Оскільки відбір здійснюється випадковим чином, то набір прикладів завжди буде різним:

* деякі приклади потраплять в кілька підмножин, а деякі не потраплять ні в одне.
* На основі кожної вибірки будується класифікатор.
* Висновки класифікаторів агрегируются (шляхом голосування або усереднення).

Оскільки беггінг – це ансамбль дерев рішень, тобто модель, яка побудована на багатьох деревах рішень. Згодом переконаємося, що результат точності прогнозу буде кращим, ніж одинична модель на тому ж наборі даних.

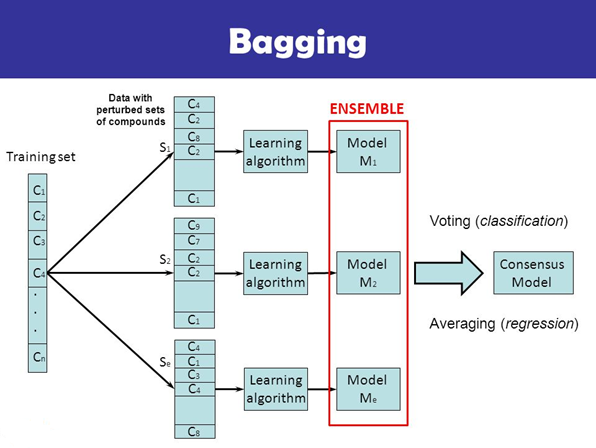


Рис.2.2.2. Bagging

Random forest (англ. Випадковий ліс) - алгоритм машинного навчання, що полягає у використанні комітету (ансамблю) дерев. Алгоритм поєднує в собі ідею методу беггінга. Алгоритм застосовується для задач класифікації, регресії і кластеризації.

Для задання алгоритму спочатку треба задати початкові параметри. Нехай навчальна вибірка складається з N прикладів, загальна кількість змінних у вибірці - M, кількість параметрів у кожному дереві – m.

Всі дерева будуються незалежно один від одного на різних частинах(«ділянках») вибірки:

Згенеруємо випадкову підвибірку з повторенням розмірності N з навчальної вибірки. (Таким чином, деякі приклади потраплять в неї кілька разів, а приблизно N / 3 прикладів не ввійдуть в неї взагалі).

Побудуємо дерево, що класифікує приклади даної підвибірки, причому в ході створення чергового вузла дерева будемо вибирати ознаки, на основі яких проводиться розбиття, не з усіх M ознак, а лише з m випадково обраних. Вибір найкращого з цих m ознак можна здійснюватися різними способами. Використовується невизначеність Джині(Gini Impurity), що застосовується також в алгоритмі побудови вирішальних дерев CART або критерій приросту інформації(Entropy and Information Gain).

Дерево будується до повного вичерпання підвибірки і не піддається процедурі прунінга (на відміну від вирішальних дерев, побудованих за таким алгоритмам, як CART або C5.0).

Класифікація об'єктів проводиться шляхом голосування: кожне дерево комітету відносить об'єкт до одного з класів, і перемагає той клас, за який проголосувала найбільша кількість дерев.

Оптимальне число дерев підбирається таким чином, щоб мінімізувати помилку класифікатора на тестовій вибірці.

Оскільки приблизно N / 3 прикладів не ввійдуть в тренувальну вибірку, тому на цих даних можна обрахувати точність моделі. Вибірку, що не ввійшла у навчальну називають – out-of-bag.

Як і в CARD та C5.0, Random Forest також має метод обрахунку найбільш інформативної змінної.

Після побудови моделі на навчальній вибірці кожну змінну тестують на даних out-of-bag та обраховують похибку для кожної змінної. Оскільки змінна задіяна у декількох деревах навчальної вибірки, тому і похибок буде декілька. Щоб отримати одне значення ці похибки усереднюють. Потім значення змінної перемішують на навчальній вибірці і знову рахують похибку на даних out-of-bag. Чим більша різниця похибки до і після перемішання значень змінної, тим більша інформативність змінної. Даний підхід має недолік – завищує значущість категоріальних змінних, якщо великий об’єм вибірки.

Основною перевагою даного методу є точність. Якість краща, ніж у моделі класифікації нейронних мереж та методу опорних векторів.Також здатна ефективно обробляти дані з великим числом ознак і класів. Однаково добре обробляються як неперервні, так і дискретні ознаки. Метод не чутливий до «викидів» та легко справляється із пропущеними значеннями. Існують методи оцінювання значущості окремих ознак в моделі.

Ці всі переваги роблять прогностичну властивість скорингової моделі, побудованої на методі ВЛ, набагато кращою, ніж на CART або C5.0.

Але також треба врахувати, що результати погано інтерпретується, на відміну від інших алгоритмів дерев рішень.

2.2.3

Градієнтний бустинг

Boosting - метод побудови ансамблю моделей, при якому створюється послідовність композиції алгоритмів, де кожна модель повинена компенсувати помилки, допущені композицією попередніх моделей.

F:\Дипломна робота_2\картинки\модель\11.PNGВідобразимо ідею бегінгу в алгоритмі градієнтного бустингу. Нам потрібно знайти модель F(x). Ми оцінюємо ефективність моделі, вибираючи функцію втрат L(y, F (x)). У нашому випадку функція втрат - це функція правдоподібності:

(Детальний огляд функції у розділі 2.2.2.(LASSO регресія)

F (x) = f (x) + λh(x) + λh'(x) + ...

f(x) -> модель над фактичним y.

h(x) -> модель, яка побудована на основі залишкової помилки (y-y'), де y' - прогноз f(x).

h''(x) побудована над залишком комбінованої моделі f (x) + h'(x). І так далі, до того моменту, коли модель досягне найкращої прогностичної здібності.

Де λ – коефіцієнт, який наз.learning rate(від англ. - швидкість навчання) – кількість інформації, яку ми будемо брати із нової моделі h.

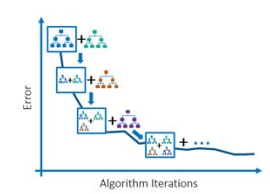


Рис.2.2.3. Алгоритм градієнтного бустингу

Отже, ми розглянули методи моделювання, за яким скорингова модель показує кращу точність. Ми отримаємо кращу прогностичну модель, якщо використаємо логістичну регресію із методом регуляризації – LASSO регресія. А також, якщо використати методи ансамблів, які основані та методах дерев рішень, то ми отримаємо набагато кращі результати, ніж на одиничній моделі дерев рішень.

2.3.1. Засоби обробки інформації

Статистичний аналіз є невід'ємною частиною наукового дослідження. Якісна обробка даних підвищує шанси опублікувати статтю в солідному журналі, і вивести дослідження на міжнародний рівень. Існує багато програм, здатних забезпечити якісний аналіз, однак більшість з них платні, і часто ліцензія коштує від кількох сотень доларів і вище. Як засіб обробки інформації я використала мову R, за яку не треба платити, а її надійність і популярність конкурують з кращими комерційними стат. пакетами: статистична мова програмування - R.

Що таке R?

Перш ніж дати чітке визначення, слід зазначити, що R - це щось більше, ніж просто програма: це і середовище, і мова.

R - це середовище обчислень, розроблене вченими для обробки даних, математичного моделювання та роботи з графікою. R можна використовувати як простий калькулятор, можна редагувати в ньому таблиці з даними, можна проводити прості статистичні аналізи (наприклад, t-тест, ANOVA або регресійний аналіз) і більш складні тривалі обчислення, перевіряти гіпотези, будувати векторні графіки і карти. Це далеко не повний перелік того, що можна робити в цьому середовищі. Варто відзначити, що вона поширюється безкоштовно і може бути встановлена ​​як на Windows, так і на операційні системи класу UNIX (Linux і MacOS X). Іншими словами, R - це вільний і багатоплатформовий продукт.

R - це мова програмування, завдяки чому можна писати власні програми (скрипти) за допомогою керуючих конструкцій, а також використовувати і створювати спеціалізовані розширення (пакети). Пакет - це набір R функцій, файлів з довідковою інформацією і прикладами, зібраних разом в одному архіві. R пакети грають важливу роль, так як вони використовуються як додаткові розширення на базі R. Кожен пакет, як правило, присвячений конкретній темі, наприклад: пакет 'ggplot2' використовується для побудови векторних графіків певного дизайну, а пакет 'qtl' ​​ідеально підходить для генетичного картування. Таких пакетів в бібліотеці R налічується на даний момент більше 7000! Всі вони перевірені на предмет помилок і знаходяться у відкритому доступі.

Як виглядає середу R?

Існує багато "оболонок" для R, зовнішній вигляд і функціональність яких можуть сильно відрізнятися. Але ми коротко розглянемо лише три найбільш популярних варіанти: Rgui, Rstudio і R, запущений в терміналі Linux / UNIX у вигляді командного рядка. Rgui - це стандартний графічний інтерфейс (https://cran.r-project.org/), вбудований в R за замовчуванням. Ця оболонка має вигляд командного рядка у вікні, званим консоллю. Командний рядок працює за принципом "питання-відповідь".

наприклад:

> 2 + 2 \* 2 # наше запитання / запит

[1] 6 # відповідь комп'ютера

Однак, для запису складного алгоритму команд в Rgui існує додаткове вікно, де пишеться програма (скрипт). Третім елементом даної оболонки є графічний модуль, який з'являється при необхідності відображення графіків.

На наведеному нижче малюнку, показана повна версія Rgui: консоль (зліва), скриптовими вікно і графічний модуль (праворуч).

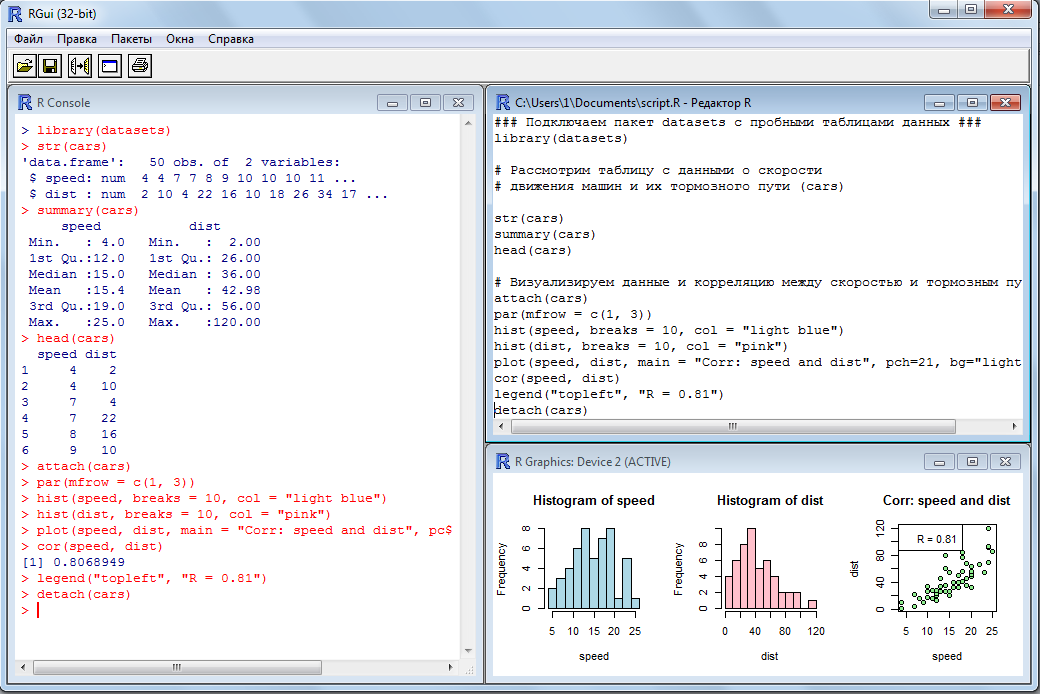


Рис2.3.1. Rgui - зовнішній вигляд

Rstudio - інтегроване середовище розробки (IDE) (https://www.rstudio.com/). На відміну від Rgui, у даній оболонки є заздалегідь розділені області та додаткові модулі (наприклад, історія команд, робоча область). На думку деяких користувачів, Rstudio має більш зручний інтерфейс, що спрощує роботу з R. Ряд особливостей, таких як підсвічування різними кольорами і автоматичне завершення коду, зручна навігація по скрипту робить Rstudio привабливою не тільки для новачків, але і для досвідчених програмістів.

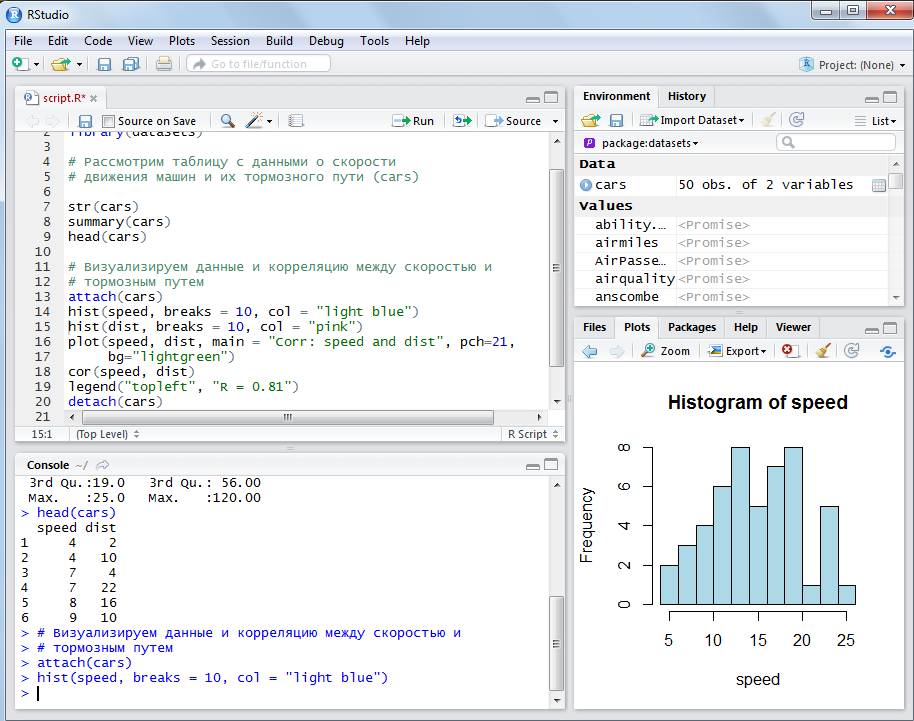


Рис.2.3.2.Rstudio зовнішній вигляд

R в терміналі Linux / UNIX.

Даний варіант кращий для аналізу великого обсягу даних через сервер, суперкластер або суперкомп'ютер. Більшість з них працюють на операційних системах класу Linux / UNIX, доступ до яких здійснюється через термінал команд (наприклад, bash). R в терміналі є додаток, запущене у вигляді командного рядка.

Порівняння R з мовами програмування MatLab, Python і Julia

Серед мов програмування, використовуваних в статистичних розрахунках, лідируючі позиції займають R і Matlab. Вони схожі між собою, як за зовнішнім виглядом, так і по функціональності. Історично MatLab був орієнтований на прикладні науки інженерних спеціальностей, тому його сильними сторонами є мат. моделювання та розрахунки, до того ж він набагато швидше R. Але так як R розроблявся як вузькопрофільна мова для статистичної обробки даних, то багато експериментальних стат. Методів з'являлися і закріплювалися саме в ньому. Цей факт і нульова вартість зробили R ідеальним майданчиком для розробки та використання нових пакетів, що застосовуються в фундаментальних науках.

Іншими "конкуруючими" мовами є Python і Julia. На мою думку, Python, будучи універсальний мовою програмування, більше підходить для машинного навчання, ніж для статистичного аналізу та візуалізації. А ось статистична мова Julia - досить молодий і претензійний проект. Основною особливістю цієї мови є швидкість обчислень, в деяких тестах перевищує R в 100 разів! Поки Julia знаходиться на ранній стадії розвитку і має мало додаткових пакетів і послідовників, але в віддалений перспективі Julia - це, мабуть, єдиний потенційний конкурент R.

Таким чином, в даний час мова R є одним з провідних статистичних інструментів в світі. Він активно застосовується в генетиці, молекулярній біології, науках про навколишнє середовище (екологія, метеорологія) та сільськогосподарських дисциплінах. Також R все більше використовується в обробці медичних даних, витісняючи з ринку такі комерційні пакети, як SAS і SPSS.

Переваги середовища R:

безкоштовна;

багатий арсенал стат. методів;

якісна векторна графіка;

більше 7000 перевірених пакетів;

гнучка у використанні:

- дозволяє створювати / редагувати скрипти і пакети,

- взаємодіє з іншими мовами, такими: C, Java і Python,

- може працювати з форматами даних для SAS, SPSS та STATA;

активна спільнота користувачів та розробників;

регулярні оновлення, хороша документація і тех. підтримка.

Недоліки:

невеликий обсяг інформації російською/українською мовою (хоча за останні п'ять років з'явилося кілька навчальних курсів і цікавих книг).

Невисока швидкість обрахунків.

2.4. Висновки

У даному розділі містяться методи обробки даних, за допомогою яких якість скорингової моделі стає кращою. Зокрема, ідеї обробки пропущених значень та відсіву викидів, а це потрібно для того щоб побудувати адекватну скорингову модель. Описаний метод біннінгу – розбиття неперервних данх на категорії(для обробки даних при побудові моделей). Також розглянуті сучасні алгоритми для оцінки кредитоспроможності клієнтів банку. Це моделі інтелектуального аналізу даних - логістична регресія з використанням LASSO регуляризації, дерева рішень, Random forest, градієнтний бустинг. Описані основні ідеї бегінгу та бустінгу.

Тобто, ми описали основні засоби та етапи побудови скорингової моделі за допомогою яких можна оптимізувати її точність.

# РОЗДІЛ 3 ПОБУДОВА СКОРИНГОВОЇ МОДЕЛІ

3.1 Візуалізація та детальна обробка вхідних даних

3.1.1. Аналіз вхідних даних та заміна пропущених значень

Щоб побудувати скорингову модель були відібрані реальні дані з бази даних «Альфа-Банку», який за розмірами активів та часткою на роздрібному ринку входить в десятку найбільших банків України.

Загальні активи Альфа-Банку, на кінець 2015 року, становили понад 42 мільярди гривень. Роздрібна мережа обслуговування включає в себе близько 100 відділень та 200 банкоматів. За початкові дані були взяті репрезентативна вибірка, яка складається із 20 змінних:

1. Стан поточного рахунку
2. Тривалість кредиту
3. Кредитна історія
4. Мета кредиту
5. Сума кредиту
6. Ощадний рахунок / облігації
7. Стаж поточної роботи
8. Ставка внеску (у відсотках від наявного доходу)
9. Сімейний статус і стать
10. Відповідальна особа
11. Нинішня резиденція
12. Тип власності
13. Вік
14. Заявник має інший кредит на розстрочку
15. Тип власності на дім
16. Кількість кредитів у цьому банку
17. Тип зайнятості
18. Кількість людей, які несуть відповідальність за кредит
19. Номер телефону(Є/ Немає)
20. Іноземний працівник (Так / Ні)

Охарактеризуємо детально змінні. У вибірці представлено такі типи даних:

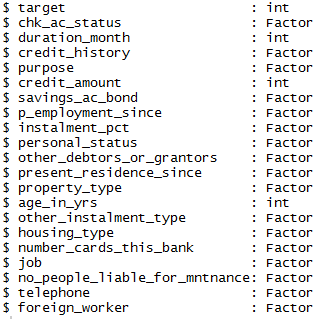


Рис.3.1.1. Типи даних у вибірці

Змінна «target» відповідає за ціль моделі, значення target=1 відповідає платоспроможному клієнту, відповідно 0 – неплатоспроможному. «Target», «duration\_month», «credit\_amount», «age\_in\_yrs» – мають тип даних «int»(скорочення від integer), тобто - цілі числа. Інші змінні мають тип даних «factor» - категоріальний тип даних.

Перевіримо наявність пропусків серед змінних:

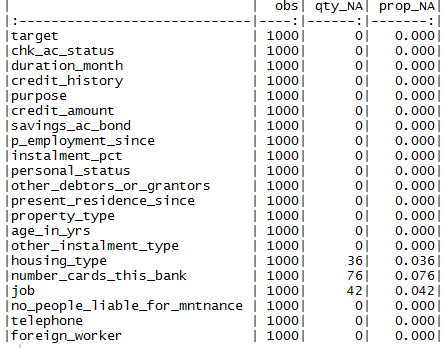


Рис.3.2.1. Розподіл пропущених значень у вибірці

Колонка «obs» - всого кількість спостережень у змінній;

«qty\_NA» - к-сть пропущених значень;

«prop\_NA» - відсоткове відношення кількості пропущених значень до загальної кількості спостережень у вибірці.

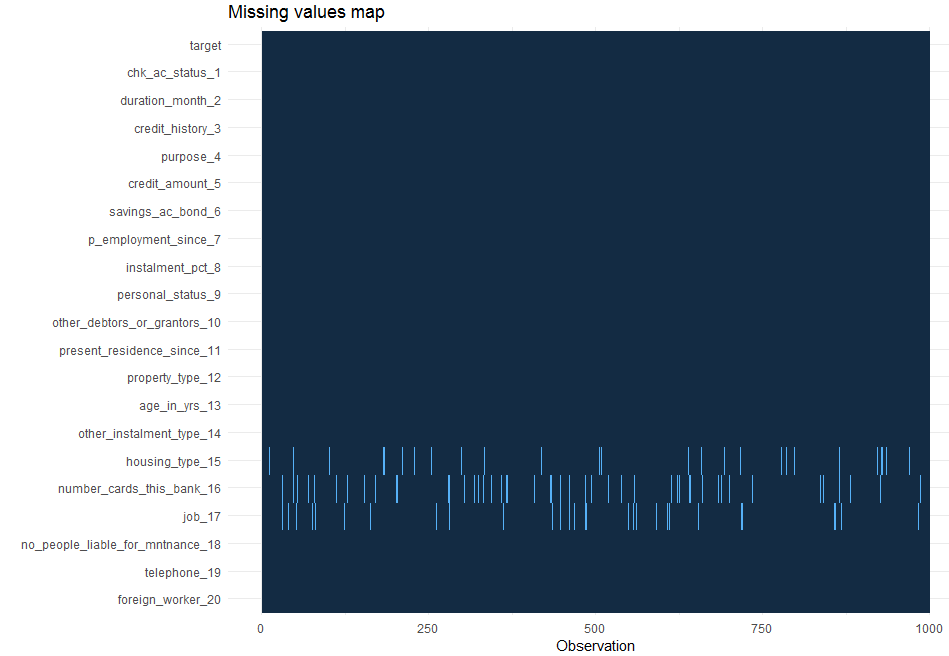
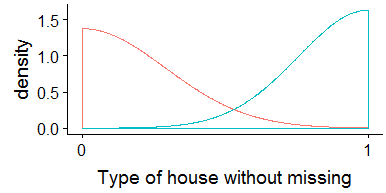
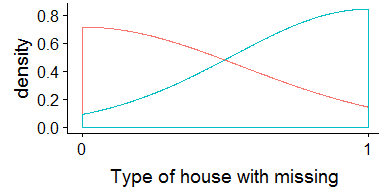
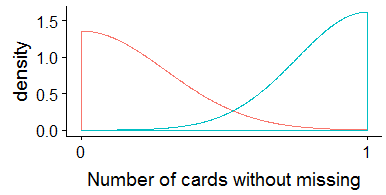
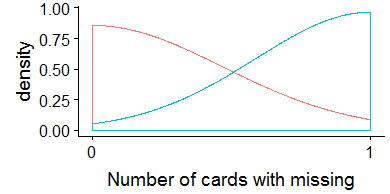
Охарактеризуємо розподіл пропущених значень:

Рис.3.1.2. Мапа пропущених значень

Змінні – «housing\_type»(тип власності на дім), «number\_cards\_this\_bank»(кількість кредитів у цьому банку), «job»(тип зайнятості) – мають пропущені значення. Пропуски у даних розподілені рандомно, тобто тип пропущених значень – MCAR (Missing Completely At Random). Це механізм формування пропусків, при якому ймовірність пропуску для кожного запису набору однакова.

Переконаємося у правильності припущень. Оскільки головною задачею моделі є класифікація клієнтів на платоспроможних і неплатоспроможних, тому нам потрібно знати, чи розподіл платоспроможних клієнтів у всій вибірці загалом і у пропущених значеннях відрізняється.





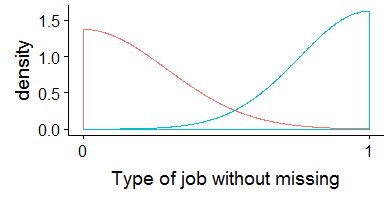
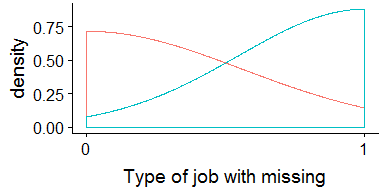


Рис. 3.1.3. Аналіз пропущених значень

Із візуалізації можна зробити висновок, що розподіли суттєво не відрізняються. Щоб прийняти нульову гіпотезу(підтвердити, що відмінності у розподілах немає) обрахуємо хи-квадрат Пирсона[11111].

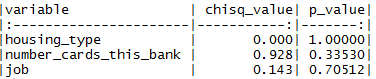


Табл.3.1.1. Значення хі-квадрат Пірсона

Колонка «variable» відповідає за назву змінної;

«chisq\_value» - за значення хі-квадрат Пірсона;

«p\_value» - за «p\_value» хі-квадрат Пірсона.

Якщо «p\_value» > 0.05, тоді приймають нульову гіпотезу. Отже, ми можемо зробити висновок, що суттєвої відмінності у розподілах немає.

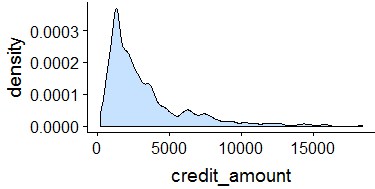
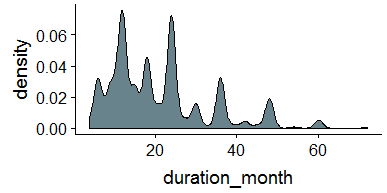
У мові програмування R наявна бібліотека «MICE»( Multivariate Imputation by Chained Equations) – багатоваріаційне заповнення значень за допомогою ланцюгових рівнянь Маркова[11111]. Скористаємося цієї бібліотекою і заповнимо пропущені значення.

3.1.2. Біннінг

Багато розробників скорингових систем використовують завжди метод категоризації кількісних змінних(binning). Категоризація кількісних змінних дозволяє досягти таких основних переваг при побудові скорингової карти: полегшити обробку викидів та екстремальних значень кількісних змінних; спростити інтерпретацію скорингової карти; відобразити складні нелінійні зв'язки.

Наявні 3 числові змінні - «duration\_month», «credit\_amount», «age\_in\_yrs».

Проаналізуємо емпіричний(без біннінгу) розподіл змінних.



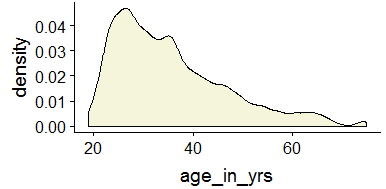


Рис.3.1.3. Емпіричний розподіл змінних

Проведемо біннінг скориставшись методами, які описані у Розділі2.

Для оцінки ефективності змінної використовують критерій – Information Value(IV)[11111].

IV - вважається найпоширенішою мірою визначення значущості змінних і вимірювання різниці в розподілі «поганих» і «хороших» клієнтів. Інформаційне Значення визначається за формулою:

IV = Σ (Gi - Bi)\*ℓn(Gi / Bi),

де Gi - відсоток всіх «хороших» випадків, Bi - відсоток всіх «поганих» випадків.

Значення даного коефіцієнта трактуються наступним чином:

* менше 0,02 - статистично незначна змінна;
* 0,02 - 0,1 - статистично мало значуща змінна;
* 0,1 - 0,3 - статистично значуща змінна;
* 0,3 і більше - статистично сильна змінна.

Угруповання всередині показників дозволяє легше зрозуміти існуючі залежності в моделі, скорингові бали стають більш транспарентні, стійкі до невеликих змін в клієнтській базі, вирішується проблема екстремальних величин і рідкісних значень і надає більше статистичної значущості скоринговим змінним.

Duration\_month:

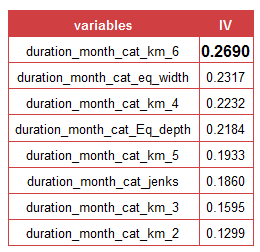
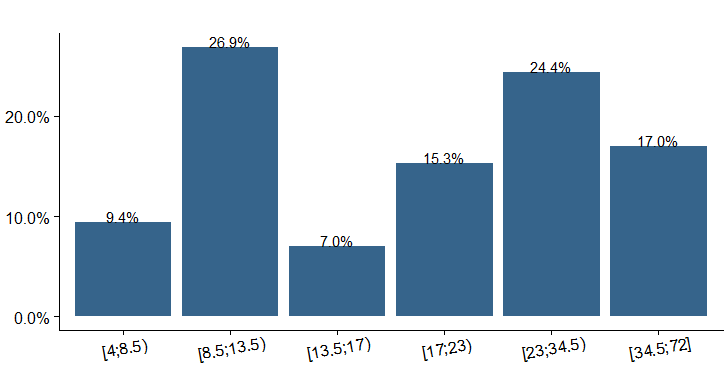


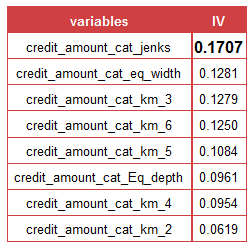
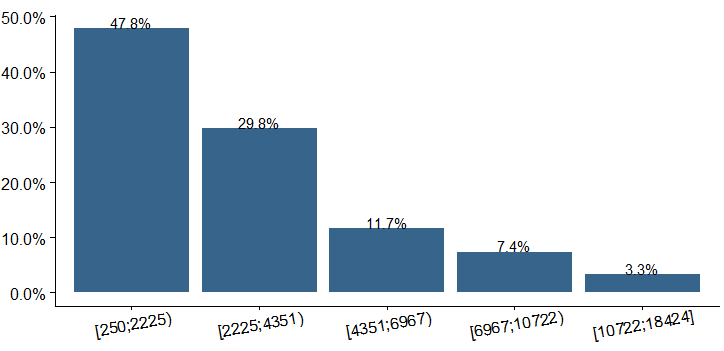
Рис.3.1.4. Біннінг змінної «Duration\_month»

Бінінг змінної «Duration\_month» виконали за допомогою методу кластеризації – k-means(«duration\_month\_cat\_km\_6»). Зауважимо, що запропонований

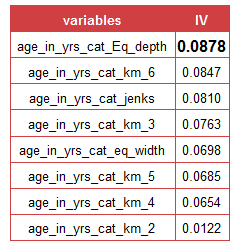
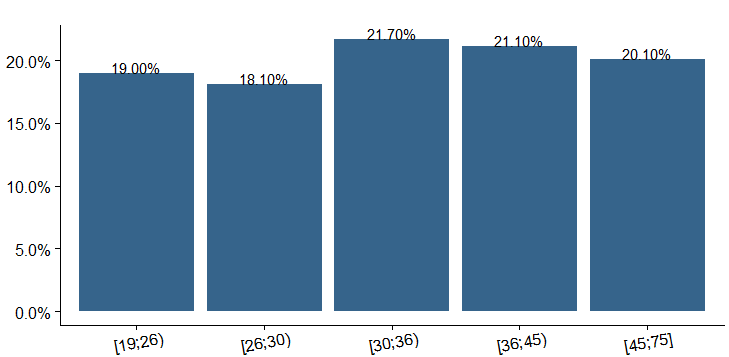
меод біннінгу, k-means, показав кращі результати, ніж універсальні методи у

скорингу – equal\_width, equal\_depth.

Сredit\_amount:



Age\_in\_yrs:



3.1.3. Корреляція між змінними

Якщо скорингова модель містить змінні, які сильно коорелюють між собою (значення коефіцієнта корреляції більше за 0.7), то усуваємо одну з цих змінних, тому що це може призвести до перенавчання алгоритму.

3.2. Побудова моделі

3.2.1. Розбиття вибірки

Для перевірки моделі потрібно зрозуміти, яким буде якість прогнозування. Для цього необхідно використовувати одну частину нашого навчального набору (зазвичай 70–80 відсотків) для навчання моделі, а іншу частину навчального набору (30–20 відсотків) використовуватиметься як тестовий набір, на якому необхідно прогнозувати цільові значення змінних і потім порівнювати прогноз з фактичними відомими значеннями. Різниця в прогнозованих значеннях і фактичних значеннях буде основою для розрахунку метрики, яка розповість нам, як добре виконується ваш алгоритм.

Що таке стратифікація?

Наш набір даних складається із 1000 рядків, де цільова змінна – платоспроможність клієнта.

Якщо ми просто випадково розділимо цей набір на дві частини, то одна частина може включати в себе в основному платоспроможних, а інша частина може включати більш неплатоспроможних клієнтів, тоді розподіл цільової змінної в обох наборах буде значно відрізнятися.

Це може призвести до того, що модель буде бачити лише платоспроможних клієнтів як приклад, який, у свою чергу, може погано навчити нові дані.

Для того, щоб зберегти розподіл цільових змінних у наборах навчальної та тестової вибірки, необхідно поділити дані за допомогою методу стратифікації[11111].

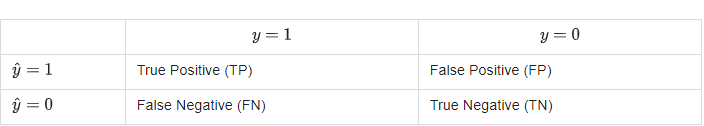
3.2.2. Метрики оцінювання моделі

Площа під ROC-кривой - один з найпопулярніших функціоналів якості в задачах бінарної класифікації. У нас вирішується завдання класифікації з двома класами {0, 1}. Алгоритм видає деяку оцінку приналежності об'єкта до класу 1. Можна вважати, що оцінка належить відрізку [0, 1].

Результат роботи алгоритму на фіксованій тестової вибірці візуалізують за допомогою ROC-кривої (ROC = receiver operating characteristic, іноді говорять «крива помилок»), а якість оцінюють як площа під цією кривої - AUC (AUC = area under the curve).

Перед переходом до самих метрик необхідно ввести важливу концепцію для опису цих метрик в термінах помилок класифікації - confusion matrix (матриця помилок).

Ми створили алгоритм,який прогнозує приналежність кожного об'єкта одному з класів, тоді матриця помилок класифікації буде виглядати наступним чином:



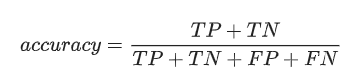
Таблиця3.2.1. Матриця помилок

Тут стовпці  - це прогноз, а рядки - справжня мітка класу.

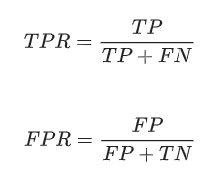
Таким чином, помилки класифікації бувають двох видів: False Negative (FN) і False Positive (FP).

Accuracy(точність)

Інтуїтивно зрозумілою метрикою є accuracy - частка правильних відповідей алгоритму:



Одним із способів оцінити модель в цілому, не прив'язуючись до конкретного порогу, є AUC-ROC (або ROC AUC). Дана крива представляє із себе лінію від (0,0) до (1,1) в координатах True Positive Rate (TPR) і False Positive Rate (FPR):



Кожна точка на графіку відповідає вибору деякого порога. Площа під кривою в даному випадку показує якість алгоритму (більше - краще), крім цього, важливою є крутизна самої кривої - ми хочемо максимізувати TPR, мінімізуючи FPR, а значить, наша крива в ідеалі повинна прагнути до точки (0,1).

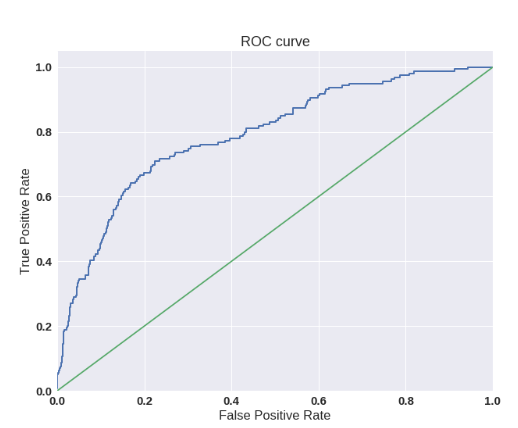


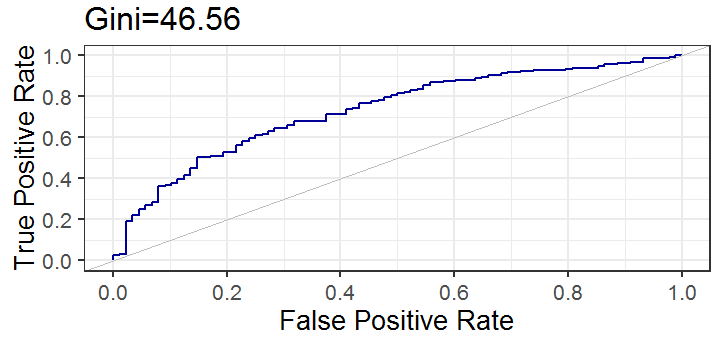
Рис.3.2.1. ROC-крива

3.2. Побудова моделі методам CART, C4.5, логістичною регресією

Логістична регресія

У цьому прикладі ми будемо використовувати параметр логістичної регресії, реалізований у функції glm, яка постачається з базовою інсталяцією в R. Ця функція відповідає класу моделей, загальновідомих як узагальнені лінійні моделі. Код досить простий, достатньо вказати тільки один параметр family = "binomial" , який вказує, що ми маємо справу із бінарною класифікацією.

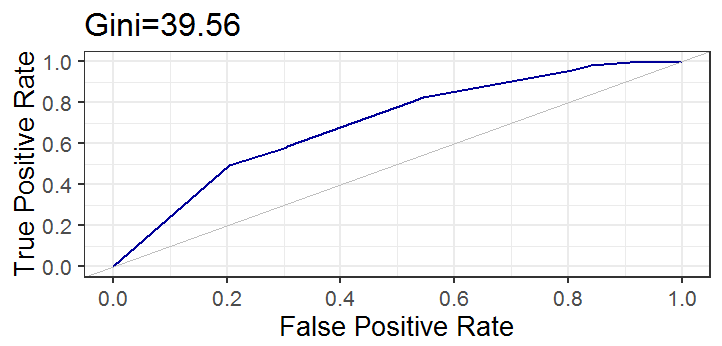
Відберемо змінні, які мають найбільше інформаційне значення(IV), адже велика кількість змінних може перенавчити модель.



F:\Дипломна робота_2\картинки\models_eval\1loggini.PNG

Рис.3.2.2. Оцінка моделі (Логістична регресія)

CART

Метод дерева рішень - це потужна і популярна методика інтелектуального машинного навчання, яка використовується як для класифікації, так і для регресії. Таким чином, він також відомий як Дерева класифікації та регресії (CART).

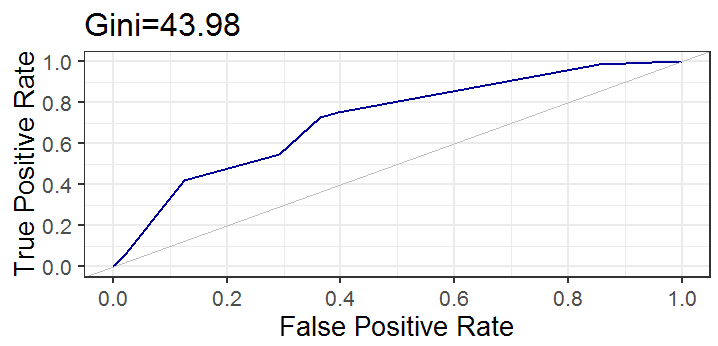
F:\Дипломна робота_2\картинки\models_eval\3cardgini.PNG

Рис.3.2.3. Оцінка моделі(CART)

Зауважимо, що R-реалізація алгоритму CART називається RPART (рекурсивне розбиття і регресія дерев).

C5.0

Алгоритм C5.0 реалізує дерева рішень. Алгоритм починається з усіх спостережень в одній групі, потім повторно розбиває дані на основі змінних, поки кожене спостереження не буде класифіковане. Щоб уникнути перенавчання, іноді дерево обрізається. Мова R реалізує C5.0 автоматично.



F:\Дипломна робота_2\картинки\models_eval\4c45gini.PNG

Рис.3.2.4. Оцінка моделі(CART)

3.3. Вдосконалення методів прогнозування та представлення результатів.

3.3.1. Розбиття вибірки

Зрозуміло, що вдосконалені методи мають більш складну, тому більш схильні до перенавчання. Для більш точного прогнозу розбивають вибірку за допомогою методу кросс-валідації.

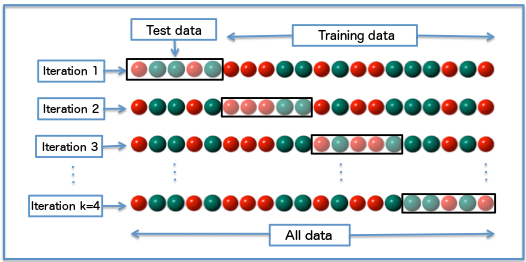


Рис 3.3.1. – Cross-validation

Крос-валідація - це метод, при якому даний набір даних розбивається на K множин, де кожна з множин використовується в якості тестового набору в якийсь момент. Візьмемо 4-кратну крос-валідацію(K = 4). У першій ітерації перша множина вибірки використовується для тестування моделі, а решта використовується для навчання моделі. У другій ітерації в якості тестового набору використовують 2-у множину, а решту – для тренувального набору. Цей процес повторюється до тих пір, поки не буде використано кожну множин як тест-набір.

3.3.2. Побудова моделей

LASSO регресія

Для скорингу LASSO регресія – це вдосконалена логістична регресія із параметром регулювання. Регуляризація карає коефіцієнти за те, що вони вносять занадто велику «вагу» змінним. Як правило, штраф зростає в геометричній прогресії, тому чим більший коефіцієнт набуває, тим сильніший штраф.

Ми будемо використовувати функцію cv.glmnet, яка влаштована у бібліотеку glmnet у мові R. Функція автоматично виконує крос-валідацію і знаходить оптимальне значення 𝝺 (коефіцієнт регуляризації).

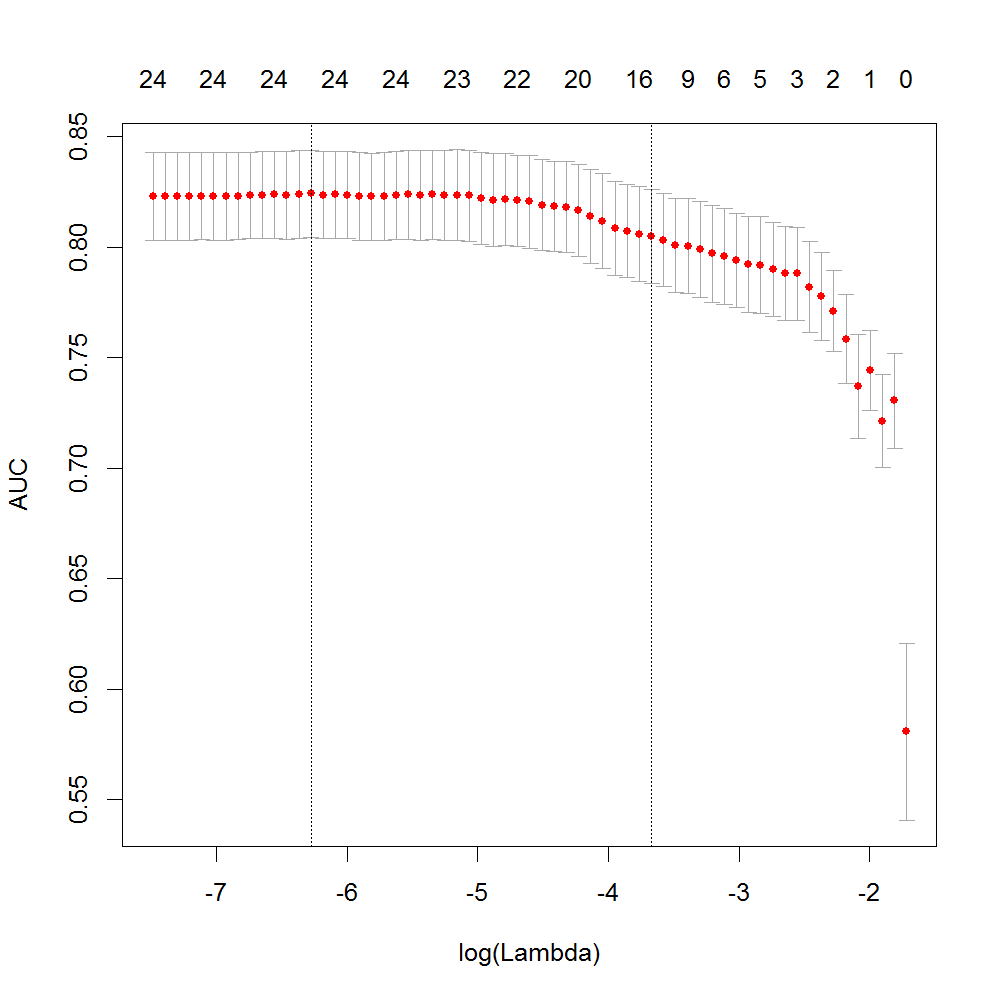
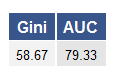


Рис3.3.2.Залежність точності моделі від значення 𝝺

Найкращу точність модель набуває при log(𝝺) = -6.27,

Обрахуємо точність моделі.

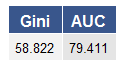


Таблиця.3.3.1. Оцінка моделі(LASSO регресія)

Випадкові ліси

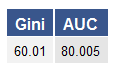
Щоб вдало побудувати модель потрібно підібрати гіперпараметри.

Спочатку побудуємо простий алгоритм, який складається із 200 дерев із кількістю змінних для кожного дерева - 3.



Таблиця.3.3.2. Оцінка моделі(Випадкові ліси)

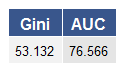
Підібравши параметри модель стала показувати кращі показники.



Таблиця.3.3.3. Оцінка моделі після підбору гіперпараметрів(Випадкові ліси)

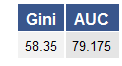
Градієнтний бустинг

Спочатку побудуємо алгоритм, який складається зі 200 дерев із глибигою для кожного дерева - 6 та параметрами регуляризації 𝝺 = 0.1 та 𝞪 = 1.



Таблиця.3.3.4. Оцінка моделі(Градієнтний бустинг)

Після перебору гіперпараметрів точність стала значно кращою:



Таблиця.3.3.5. Оцінка моделі після підбору гіперпараметрів(Випадкові ліси)

3.4. Практичне застомування. Розробка стартапу

3.4.1 Опис стартапу

Оскільки моделювання в Україні перебуває у стадії розвитку, а кредитна система потребує змін, тому найкращим початком вдосконалення системи є розробка стартап-проекту.

Основна мета стартапу є оцінка платоспроможності клієнтів банку за допомогою оптимізованих методів.

Стартап використовує комбінацію методів машинного навчання, щоб допомогти кредиторам оцінити кредитний ризик позичальника. Компанія дивиться на певні параметри в кредитному звіті, а також аналізує використання телефону і соціальних мереж.

Назва стартапу: «GiniMachine»

F:\Дипломна робота_2\картинки\модель\15.PNG

Логотип проекту:

Реалізація стартапу: 10 - 18 місяців

Команда «GiniMaсhine»

Склад команди:

● Біл – DB administrator

● Алан – Data scientist

● Стив – Django developer

● Ілон – маркетолог

● Світлана – менеджер з продажів

Завдання проекту по ролям :

1. Завдання DB administrator:

• збір вибірки для побудови моделей;

• написання на мові sql процедур, які підтримують базу даних для повного функціонування моделей.

2. Завдання Data scientist:

• побудова скорингових моделей;

• проведення регулярної аналітики компанії.

3. Завдання Django developer:

• загортання результатів моделей в API для зовнішнього користувача .

4 Завдання маркетолога:

• аналізувати ринок;

• збір інформації про продажі, на основі яких аналізується попит, запускати рекламу для компанії;

• складання звітів і презентацій-рекомендацій по просуванню.

5. Завдання менеджера з продаж:

• знаходження партнерів(клієнтів) і налагодження ділових стосунків з ними;

• підготовка комерційних пропозицій і презентацій.

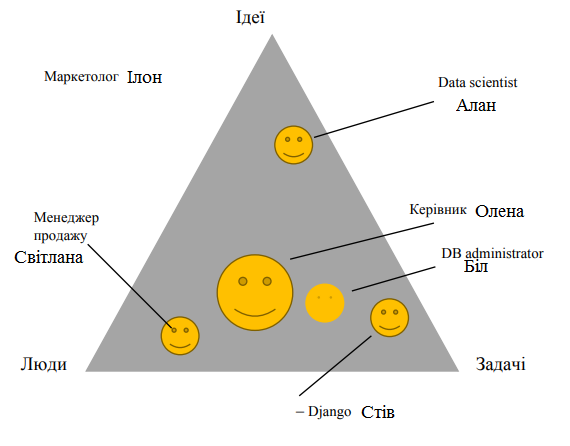


Рис.3.4.1. Розподіл ролей в компанії

3.4.2. Бізнес модель проекту

Потенційні партнери компанії:

- фінансові установи;

- банки;

- компанії, які основані на великих даних.

- банки.

Потенційні затрати:

- заробітна плата співробітників;

- робота та обслуговування серверів ;

- оренда офісу ;

- канцелярія та техніка для працівників.

Потенційні доходи:

- оплата запитів на надання клієнтського скорингу.

Потенційний розвиток продукту можливий за таких умов:

По-перше, це поганий розвиток Data Science ринку в Україні. Багато компаній, які мають великі об`єми даних не знають як вони могли б їх використовувати, а потім на них заробляти.

Деякі банки частину задач віддають на аутсорс, у зв`язку з недостатнім розвитком ІТ працівників в банківський сфері.

Пропозиції для інвесторів:

- 50% від основного прибутку;

- Реалізація - від 10 до 18 місяців.

Було проведено зрівняльний аналіз недоліків та сильних сторін стартапу

Розглянемо сильні та слабкі сторони проекту

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № п/п | Конкурентоспроможність/  Характеристики | Бали 1-20 | Порівняння конкурентів із GiniMachine | | | | | | |
| -3 | -2 | -1 | 0 | 1 | 2 | 3 |
| 1 | Нові технології | 9 |  |  | + |  |  |  |  |
| 2 | Обсяг інформації | 11 |  |  |  |  | + |  |  |
| 3 | Автоматизація обрахунків | 11 |  | + |  |  |  |  |  |
| 4 | Ціна | 17 |  |  |  | + |  |  |  |

Таблиця 3.4.1 – Конкурентоспроможність «GiniMachine»

Оцінимо слабкі та сильні сторони компанії за допомогою SWOT-аналізу.

|  |  |
| --- | --- |
| Strength:   * ціна, * використання автоматизація обрахунків * нові технології | Weakness:   * багато компаній-монополій |
| Opportunity:   * знаходження нових рішень, алгоритмів * збільшення капіталу * накопичення більшої кількості даних | Treats:   * недостатній об`єм даних * нестача досвіду |

Таблиця 3.4.2 – SWOT- аналіз компанії

3.5. Висновок

У даному розділі було описано загальний процес вдосконалення скорингової маделі та кредитної системи України загалом.

Предсталена поетапна первинна обробка даних. Показано, як можна усунути недоліки у даних(пропущені значення, викиди) та покращити струкктуру(біннінг). Було доведено, що застосування нових методів біннінгу(k-means, jenks) роблять змінні більш інформаційно значущими у порівнянні із універсальними методами.

При побудові моделі універсальними методами скорингу(логарифмічна регресія, CARD, C5.0) було виявлено, що логарифмічна регресія показує кращі результати, ніж методи дерев рішень. Цікавим є факт, що нові,оптимізовані, методи скорингу(Випадкові ліси), які засновані на деревах рішень показали кращі результати, ніж оптимізована логарифмічна регресія(Лассо регресія).

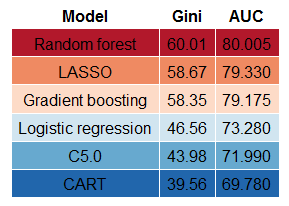


Рис.3.5.1. Результат моделей

Щоб правильно оцінити наші вдосколення, представимо шкалу оцінки моделі:



Таблиця.3.5.1. Шкала точності моделі

Тобто, ми вдосконалили модель із середньої точності до хорошої точності.

По результатам роботи була описана ідея стартапу, ціль якої покращити кредитну систему України за рахунок продажу скорингових моделей банкам.

ВИСНОВК.

ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

В роботі розглянуто математичні методи за допомогою яких можна вдосконалити кредитну систему України. Кредитний ризик уособлює непостійність та невизначеність у найбільш поширеній для банків діяльності – кредитуванні. Діяльність українських банків по кредитуванню фізичних осіб знаходиться в періоді вибухового розвитку. Одним з найбільш ефективних методів оцінки клієнтів банку для видачі їм кредитів – кредитний скоринг. Ось чому я вирішила дослідити тему вдосконалення методів моделювання у кредитній системі нашої країни. Дані для аналізу - це репрезентативна вибірка українського банку (АТ «Альфа-Банк»).

На першому етапі був проведений аналіз даних. Було показано, як можна усунути недоліки у даних(пропущені значення, викиди) та покращити структуру даних(біннінг).Ми переконалися, що застосування нових методів біннінгу(k-means, jenks) роблять змінні більш інформаційно значущими у порівнянні із універсальними методами.

На наступному етапі була побудова моделей. Спочатку типові методи для скорингу, а потім їх оптимізація. Найкращим виявився метод випадкових лісів, який оснований на деревах рішень.

Отже, точність моделі покращилася із 69-73% до 79-80%.

Надалі цікавою темою для дослідження - це залучення нової інформації. Це можуть бути дані із соціальних мереж, мобільних операторів, транзакцій по картці і т.д.

Список

- хи-квадрат Пирсона

- ланцюги маркова(<https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D0%BD%D1%86%D1%8E%D0%B3%D0%B8_%D0%9C%D0%B0%D1%80%D0%BA%D0%BE%D0%B2%D0%B0>)

--IV

-- стратифікація