НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

«На правах рукопису»
УДК 519.23-519.25

«До захисту допущено»
Завідувач кафедри
_____ О. Л. Тимощук
«16» травня 2018 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 124 Системний аналіз

на тему: «Скорингові моделі поведінки гравців для оцінки фінансових показників компанії»

Виконав:
студент ІІ курсу, групи КА-61м
Фомін Олександр Володимирович _____

Науковий керівник:
к.т.н., доцент кафедри ММСА
Кузнєцова Н.В. _____

Рецензент:
д.т.н., проф., завідуючий кафедрою АУТС
Теленик С.Ф. _____

Засвідчую, що в цій магістерській дисертації немає запозичень із праць інших авторів без відповідних посилань.
Студент _____

Київ — 2018

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти — другий (магістерський)
Спеціальність (спеціалізація) – 112 «Системний аналіз» («Системи і методи прийняття рішень»)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ О. Л. Тимощук
«16» березня 2018 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Фоміну Олександру Володимировичу

1. Тема дисертації: «Скорингові моделі поведінки гравців для оцінки фінансових показників компанії», науковий керівник дисертації Кузнєцова Наталія Володимирівна, к.т.н., доцент кафедри ММСА, затверджені наказом по університету від «27» березня 2018 р. № 1028-с.

2. Термін подання студентом дисертації: 16 травня 2018 р.

3. Об’єкт дослідження: клієнти компанії, що працює на ринку онлайн-ігор, та їх поведінка, для ефективного планування діяльності цієї компанії.

4. Предмет дослідження: скорингові моделі поведінки та методи інтелектуального аналізу даних для кількісного та якісного оцінювання клієнтів.

5. Перелік завдань, які потрібно розробити: обрати предметну область, здійснити аналіз існуючих даних, вибрати підходящий математичний апарат, здійснити постановку задачі дипломної роботи, проаналізувати існуючі методи класифікації, проаналізувати існуючі методи оцінювання ризиків та часу життя гравців, здійснити огляд особливостей ігрової індустрії, систематизувати методику оцінювання поведінки клієнтів, здійснити моделювання, провести порівняння результатів та напрацювати рекомендації.

6. Орієнтовний перелік ілюстративного матеріалу: структура статистичних даних, схеми ієрархічної кластеризації гравців, графіки показників якості моделей, процедура побудови системи прийняття рішень, концептуальна схема прототипу системи прийняття рішень.

7. Орієнтовний перелік публікацій: стаття «Прогнозування ризику втрати користувачів онлайн-платформи» - тези та виступ на конференції SAIT-2018.

8. Дата видачі завдання: 16 березня 2018 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Оформлення концептуального вступу	16.03.2018-25.03.2018	
2	Написання першого та другого розділів	25.03.2018-05.04.2018	
3	Робота над третім розділом	05.04.2018-20.04.2018	
4	Оформлення четвертого розділу	20.04.2018-30.04.2018	
5	Формулювання висновків і синтез результатів	30.04.2018-07.05.2018	

Студент _____ О.В. Фомін

Науковий керівник дисертації _____ Н.В. Кузнєцова

РЕФЕРАТ

Магістерська дисертація: 91 с., 29 рис., 24 табл., 2 додатки і 19 джерел.

Об’єкт дослідження – клієнти компанії, що працює на ринку онлайн-ігор, та їх поведінка.

Предмет дослідження – скорингові моделі поведінки та методи інтелектуального аналізу даних для кількісного та якісного оцінювання клієнтів.

Мета роботи – розробка методів моделювання поведінки клієнтів та їх порівняння із існуючими загальноприйнятими.

Методи дослідження – моделі бінарної класифікації, ансамблі дерев, моделі ієрархічної кластеризації, моделі виживання.

У цій роботі наведені результати побудови моделей скорингу поведінки використовуючи бустингові ансамблі дерев. Проведено порівняльний аналіз отриманих моделей за допомогою різних критеріїв, а також зроблено висновки щодо їхньої точності. Виявлено, що таким моделям вдається найкраще описувати нелінійні залежності в даних, і вони показують найкращі результати. Тому для подальших досліджень рекомендовано використовувати саме такі моделі.

За матеріалами магістерської дисертації були написані тези та наукова стаття. Тези опубліковані в збірці тез доповідей конференції CAIT-2018. А стаття буде опублікована в електронній збірці доповідей у видавництві CEUR.

Прогнозні припущення щодо подальшого розвитку об’єкта дослідження – вдосконалення існуючих моделей скорингу поведінки. А також покращення існуючої системи прийняття рішень на основі побудованих моделей.

СКОРИНГ, СТАТИСТИЧНИЙ АНАЛІЗ ДАНИХ, МОДЕЛІ ПОВЕДІНКИ, АНСАМБЛІ ДЕРЕВ, XGBOOST, ЛОГІСТИЧНА РЕГРЕСІЯ, АНАЛІЗ ВИЖИВАННЯ.

ABSTRACT

Scoring models of players behavior for financial performance of the company estimation.

Master's thesis: 91 p., 29 fig., 24 tab., 2 appendixes and 19 sources.

The object of study – clients of the online-games provider and their behavior.

Subject of research – scoring behavioral models and methods of intellectual data analysis for quantitative and qualitative evaluation of clients.

Purpose – constructing behavioral models of clients and comparison for existing ones.

The method of research – binary classification models, tree ensembles, hierarchical classification models, survival analysis models.

The paper represents results of estimation of behavioral models, based on boosted tree ensembles. The results of the comparative analysis of the obtained models are described with the help of information criteria, and in terms of their accuracy. It was found that boosted trees are best suited for describing non-linear dependencies in the data, and the tend to better performance. Therefore, the usage of such models is strongly recommended for further studies.

Theses and a scientific article were written based on master's dissertation. The theses have been published in the SAIT-2018 conference Book of Abstracts. The scientific article is going to be published in the electronic collection of reports at the CEUR publishing house.

Recommendations for further development consist of improvement of existing behavioral scoring models, and improvement of existing prototype of decision support system that is based on built models.

SCORING, STATISTICAL DATA ANALYSIS, BEHAVIORAL MODELS, TREE ENSEMBLES, XGBOOST, LOGISTIC REGRESSION, SURVIVAL ANALYSIS.

ЗМІСТ

УМОВНІ ПОЗНАЧЕННЯ ТА СКОРОЧЕННЯ 9
ВСТУП 10
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 12
1.1 Ринок онлайн ігор 12
1.1.1 Онлайн гемблінг 14
1.1.2 Форми онлайн гемблінгу 15
1.1.3 Онлайн покер 16
1.1.4 Інтернет-казино 19
1.1.5 Ставки на спорт 20
1.1.6 Онлайн бінго та лотереї 21
1.2 Функціонування компанії в умовах невизначеності 21
1.2.1 Навантаження на сервер 22
1.2.2 Фінансові показники 23
1.2.3 Відтік клієнтів 24
1.3 Опис даних 24
Висновки. Постановка задачі дипломного проекту 26
РОЗДІЛ 2 ОПИС МАТЕМАТИЧНОГО АПАРАТУ 27
2.1 Методи кластеризації та класифікації 27
2.1.1 Математичне формулювання завдання 28
2.1.2 Регресійні моделі 28
2.1.3 Класифікація без учителя 31
2.2 Методи оцінки ймовірності відпаду 33
2.2.1 Застосування дерев рішень 33
2.2.2 Ансамблі дерев рішень 37
2.2.3 Градієнтний бустинг 38
2.3 Методи оцінки часу відтоку 44
2.3.1 Аналіз виживання 44
Висновки 48
РОЗДІЛ 3 СКОРИНГОВІ МОДЕЛІ 49

3.1 Цензорування даних та генерація змінних 49
3.2 Оцінка ймовірності відтоку клієнта 50
3.2.1 Логістична регресія 51
Результат логістичної регресії (рисунок 3.1): 51
3.2.2 XGBoost 52
3.2.3 Аналіз показників та порівняння моделей 58
Висновки 59
РОЗДІЛ 4 АНАЛІЗ ВИЖИВАННЯ ТА КЛАСИФІКАЦІЯ 61
4.1 Постановка задачі 61
4.2 Ієрархічна кластеризація гравців гри «Покер» 63
4.3 Побудова моделей виживання 67
Висновки 70
РОЗДІЛ 5 ПРОТОТИП СППР ДЛЯ ОЦІНКИ МОЖЛИВИХ ВТРАТ ДОХОДІВ 72
5.1 Архітектура СППР 72
5.1.1 Модуль вітрин SQL 73
5.1.2 Модуль оцінки параметрів 75
5.1.3 Модуль візуалізації одержаних результатів 76
5.2 Деталі реалізації 76
Висновки 77
РОЗДІЛ 6 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ 79
1.1 Опис ідеї проекту 79
1.2 Технологічний аудит ідеї проекту 81
1.3 Аналіз ринкових можливостей запуску стартап-проекту 81
1.4 Розроблення ринкової стратегії проекту 85
1.5 Розроблення маркетингової програми стартап-проекту 86
Висновки 88
ВИСНОВКИ 90
ПЕРЕЛІК ПОСИЛАНЬ 92
ДОДАТОК А. ІЛЮСТРАТИВНІ МАТЕРІАЛИ ДОПОВІДІ 95
ДОДАТОК Б. ЛІСТИНГ ПРОГРАМИ 104

УМОВНІ ПОЗНАЧЕННЯ ТА СКОРОЧЕННЯ

XGBoost – Extreme gradient boosting
PD – Probability of default
EAD – Exposure at default
LGD – Loss given default
PH – Proportional hazards
KM – Kaplan-Meier estimator
ROC – Receiver-operator characteristic
AUC – Area under curve
FPR – False positive rate
TPR – True positive rate

ВСТУП

Швидка зміна технологій, використання мобільних телефонів, велика кількість соціальних мереж, розвиток доповненої та віртуальної реальностей суттєво впливають на життя людей і змінюють уподобання клієнтів. Якщо декілька років тому онлайн-ігри займали весь простір вільного часу користувачів-гравців, то тепер дедалі актуальнішим постає задача втримання існуючих клієнтів, оскільки залучення нових гравців до системи онлайн-ігор стає лише короткостроковою подією, - наслідком реклами або переадресації з іншої гри. У компанії, що здійснює підтримку користувачів онлайн-ігор, виникає необхідність оцінювання ризику втрати клієнта та прогнозування моменту, коли це може відбутися, та, за ідеального випадку, коли він може повернутися до гри.

Це обумовлює *актуальність* проблеми. Якщо ж подивитися на функціонування компанії, як на систему, то в одразу можна помітити цілий ряд невизначеностей, таких як: нестабільність навантаження на сервер, неможливість точно спрогнозувати фінансові показники та випадковість процесу відпаду гравців. Ця проблематика наштовхує на ідеї боротьби із ними. І якщо у випадку з серверами, нічого кращого, аніж придбання резервних придумати не можна, то інші невизначеності можна і треба мінімузовувати. Для цього і покликана дана робота.

Таким чином, *мета* дослідження – це розробка моделей та методів моделювання поведінки клієнтів та їх порівняння із існуючими загальноприйнятими. Для її досягнення були поставлені наступні *завдання*:

- розглянути існуючі методи побудови скоринг-моделей;
- виявити найбільш актуальні та перспективні підходи до розробки моделей;

- розробити власні моделі поведінки клієнтів-гравців онлайн-платформи;
- порівняти отримані результати та зробити висновки;
- розробити прототип системи прийняття рішення для розв’язку схожих завдань у подальшому.

Об’єктом цього дослідження є клієнти компанії, що працює на ринку онлайн-ігор, та їх поведінка, для ефективного планування діяльності цієї компанії, а *предметом* – скорингові моделі поведінки та методи інтелектуального аналізу даних для кількісного та якісного оцінювання клієнтів. При дослідженні серед інших застосовувалися наступні *методи:* аналіз, синтез, абстрагування, порівняння та узагальнення.

В роботі приділяється увага базовим ідеям для оцінки ймовірності відпаду та розробляється власний алгоритм на основі XGBoost для покращення цього оцінювання.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Ринок онлайн ігор

Станом на 2017 рік 2,2 млрд. гравців по всьому світу генерують 109 млрд. доларів США доходу [2]. Це показує приріст на 7,8 млрд., тобто 7,8%, порівняно з минулим роком. Дохід цифрових ігор покриває 87% світового ринку. Мобільні ігри являються найбільш прибутковим сегментом – швидкими темпами набирають популярності ігри на мобільних телефонах та планшетах – у середньому спостерігається приріст 19% кожного року [2]. Наразі сегмент мобільних ігор покриває 42% глобального ринку, а за прогнозами в 2020 мобільна ігрова індустрія буде представляти більше ніж половину усього ринку. Ігри на ПК та консолі згенерували 29,4 млрд. та 33,5 млрд. доходу в 2017.

При цьому 43% – 47,11 млрд. доларів США усього ринку становить індустрія онлайн гемблінгу, або азартних онлайн ігор. Онлайн гемблінг характеризується тим, що гравець ставить певну річ, що має цінність, зазвичай гроші, на результат певної події або гри, використовуючи Інтернет. Онлайн гемблінг включає такі види, як:

- покер;
- казино (де гравці грають в традиційні ігри казино: рулетка, блекджек, але онлайн) та слоти;
- ставки на спорт;
- бінго та лотереї.

За даними [3] ринок онлайн гемблінгу досягне об’єму в 51,96 млрд. доларів США, що майже вдвічі більше, порівняно з 2009 роком (див. рисунок 1.1).

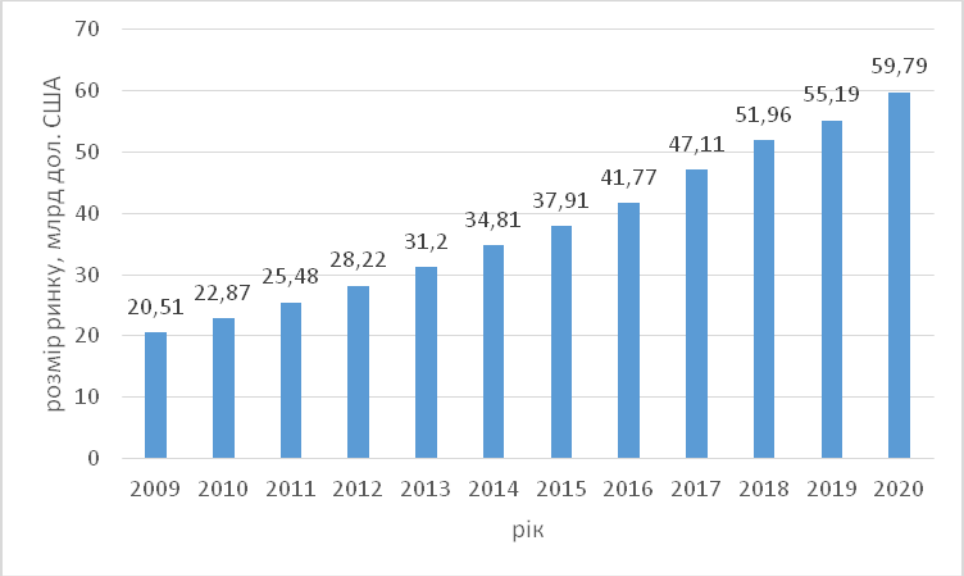


Рисунок 1.1 – Динаміка ринку онлайн гемблінгу

Ринок онлайнових азартних ігор складає значну частку загального ринку онлайн ігор (див. рисунок 1.2). Однак незважаючи на швидкий ріст складає лише незначну частку усієї гемблінгової індустрії. В опитуванні, проведеному в 2016-му Нільсеном Скарборо, майже 83 млн. американців призналися, що відвідували казино у минулому році [2]

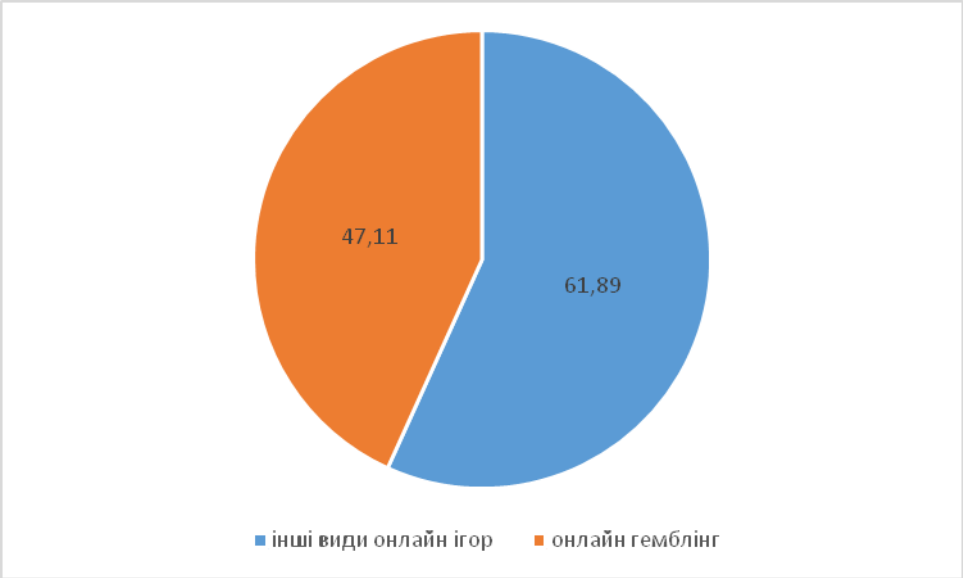


Рисунок 1.2 – Доля онлайн гемблінгу на ринку онлайн ігор, млрд. дол. США

1.1.1 Онлайн гемблінг

У 1994 році Антигуа і Барбуда прийняли Закон "Про вільну торгівлю та переробку", що дозволило отримати ліцензії організаціям, які подають заявку на відкриття онлайн казино. Перше повнофункціональне програмне забезпечення для грального бізнесу було розроблено компанією Microgaming, компанією з програмного забезпечення на острові Мен. Це було реалізовано на основі програмного забезпечення, розробленого компанією CryptoLogic, що займається онлайн-захистом програмного забезпечення. Безпечні операції стали життєздатними та призвели до першого інтернет-казино в 1994 році.

У 1996 році була створена комісія з ігрових технологій Kahnawake, яка регулювала онлайн-ігрову діяльність з території Mohawk Kahnawake і видає ігрові ліцензії для багатьох інтернет-казино та покер-румів у світі. Це спроба зберегти операції ліцензованих онлайн-азартних ігор ярмарок і прозорості.

Наприкінці 1990-х років інтернет-азартні ігри отримали популярність. Сайти, пов'язані з азартними іграми в Інтернеті, зросли з лише 15 веб-сайтів у 1996 році та до 200 веб-сайтів у 1997 році. У доповіді, опублікованому Frost & Sullivan, виявлено, що доходи від онлайн-азартних ігор перевищили 830 мільйонів доларів лише в 1998 році. У цьому ж році з'явилися перші онлайн-покер-руми. Незабаром після того, як у 1999 році було запроваджено Закон про заборону Інтернет-гемблінгу, це означає, що компанія не може запропонувати будь-якого азартного онлайн-продукту для будь-якого громадянина США. Це не пройшло. Мультіплеер онлайн-азартних ігор був також представлений в 1999 році. Це перший раз, коли люди могли грати в азартні ігри, спілкуватися та спілкуватися один з одним в інтерактивному онлайн-середовищі.

У 2000 році перший федеральний уряд Австралії прийняв Закон про інтерактивний акт про мораторій, що робить його незаконним для будь-якого

он-лайн-казино, який не був ліцензований та працював до травня 2000 року. Нове законодавство означало, що Інтернет-канал Lasseter's Online став єдиним інтернет-казино, який може юридично працювати в Австралії; однак, вони не можуть брати ставки від австралійських громадян.

До 2001 року очікувана кількість людей, які брали участь у онлайн-азартних іграх, зросла до 8 мільйонів, а ріст продовжувався б, незважаючи на законодавство та суперечки, які і надалі будуть надані онлайн-азартними іграми.

1.1.2 Форми онлайн гемблінгу

Інтернет дозволив формувати нові види азартних ігор в Інтернеті. Поліпшення в технології змінили ставки, подібно до того, як термінали для відео лотереї, кено та скретч-карт змінили азартні ігри на початку 20-го століття.

Азартні ігри стали одним з найпопулярніших та найприбутковіших сфер у мережі Інтернет. У 2007 році комісія з азартних ігор у Великобританії заявила, що азартна промисловість, згідно з Комісією з азартних ігор у Великій Британії, досяг свого обороту понад 84 мільярдів фунтів стерлінгів. Частково це пов'язано з широким вибором варіантів азартних ігор, доступних для різних людей. У статті, присвяченій Даррену Р. Крістенсену, Нікі А. Даулінг, Алун К. Джексон та Шейн А. Томас, опитування, зареєстроване в Австралії, показує, що найбільш поширеними формами азартних ігор є лотереї (46,5%), кено (24,3%), миттєві чекові квитки (24,3%) та електронні ігрові автомати (20,5%) (рисунок 1.3).

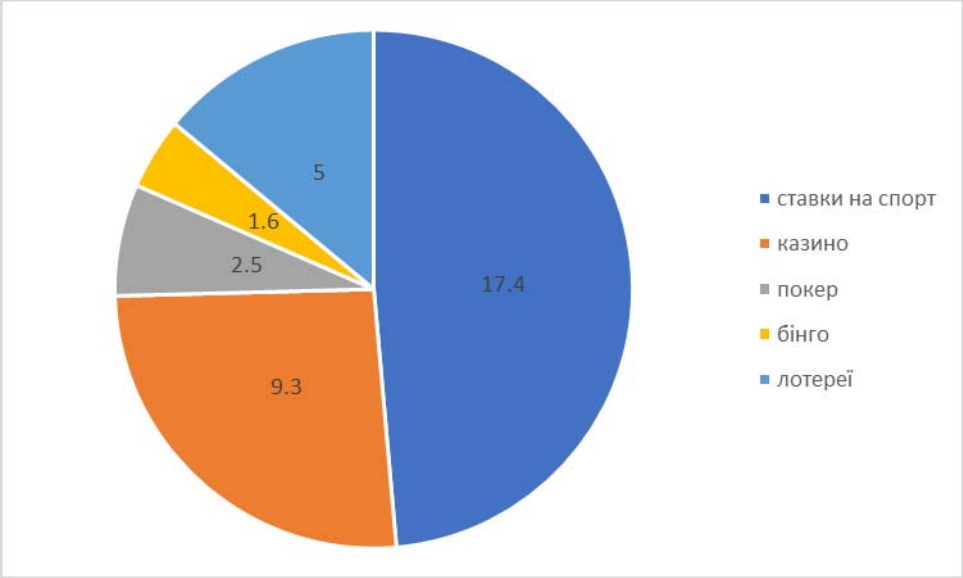


Рисунок 1.3 – Розподіл форм онлайн гемблінгу за розміром ринку

1.1.3 Онлайн покер

Традиційні майданчики для гри у покер, такі як казино та покерні зали, можуть відлякувати початківців і часто розташовуються в географічно невідповідних місцях. Також офлайн казино не активно рекламують покер, тому що їм важко отримувати прибуток з реклами. Незважаючи на те, що загальна суми рейку та погодинної тарифікації досить високі, альтернативні витрати на керування покер-румом ще вищі. Казино здатні генерувати більший дохід, позбуваючись покерних залів та додаючи ігрові автомати - наприклад, бухгалтерська фірма Джозефа Єви підрахувала, що покер складає лише 1% доходів офлайн казино.

Інтернет-сайти, навпаки, значно дешевші, оскільки мають набагато менші накладні витрати. Наприклад, ще один покерний стіл не займає цінного простору, як це було б у випадку звичайних казино. Онлайнові покер-руми також дозволяють гравцям грати на низьких ставках і часто пропонують

турніри з фріролів у покер (там, де немає вхідного внеску), залучаючи початківців та/або менш багатих клієнтів.

Інтернет-сайти можуть бути більш вразливими до певних видів шахрайства, особливо змов між гравцями. Проте у них є здібності розпізнавання таких змов та нечесної поведінки, які не існують в звичайних казино. Наприклад, представники служби безпеки онлайн-покеру можуть переглядати історію рук, які раніше грав будь-який гравець на сайті. Це значно полегшує виявлення моделей поведінки, порівняно з казино, де гравці можуть просто скласти руки, і тоді ніхто ніколи не дізнається про силу їхніх карт. Кімнати інтернет-покеру також перевіряють IP-адреси гравців, щоб запобігти грі в одній і тій самій мережі або з одного відкритого проксі-сервера. Відбитки цифрових пристроїв також дозволяють покерним сайтам розпізнавати та блокувати гравців, які створюють нові облікові записи, намагаючись обійти попередні заборони, обмеження чи закриття рахунків [15].

Безкоштовний онлайн покер був вперше імплементований ще наприкінці 1990-х років у вигляді покеру IRC. "Планета Покер" - перша покерна кімната, яка запропонувала ігрові гроші в 1998 році. Перша гра на реальні гроші була проведена 1 січня 1998 року.

Основні онлайн-покерні сайти пропонують різні функції, щоб заманити нових гравців. Однією з поширених функцій є надання турнірів, які називаються супутниками, за якими переможці отримують доступ до реальних покерних турнірів.

У жовтні 2004 року Sportingbet, на той час найбільша в світі онлайн-ігрова компанія, що продала публічну торгівлю (SBT.L), оголосила про придбання ParadisePoker.com, одного з перших і найбільших кардових центрів онлайн-покер-індустрії. Придбання у розмірі 340 мільйонів доларів США ознаменувалося вперше, коли в приміщенні онлайнова карткова кімната належала публічна компанія. З тих пір ще декілька материнських компаній, які займають картки, стали публічними [15].

У червні 2005 року PartyGaming, - материнська фірма одного з найбільших покерних румів, - PartyPoker, розмістила свої акції на Лондонській фондовій біржі, досягши первинного публічного розміщення, вартістю понад 8 млрд. дол. США. На момент проведення IPO покерні операції складали 92% доходів PartyGaming.

Станом на лютий 2010 року існує близько 545 онлайн-сайтів для покеру. В межах 545 активних сайтів приблизно два десятки є автономними сайтами (порівняно з 40 у березні 2008 року), а решту сайтів називають "скінами" і працюють у 21 різних спільних мережах, найбільша мережа - iPoker, що має десятки скінів що працюють у своїй мережі. З усіх онлайн-покер-румів PokerStars.com вважається найбільшим в світі покерним сайтом за кількістю гравців на сайті в будь-який час. До травня 2012 року PokerStars.com збільшив свою частку ринку до більш ніж 56%.

2011 рік відомий як ганебний рік Чорної п'ятниці, коли Департамент юстиції США заборонив доменні імена PokerStars, Full Tilt та Absolute Poker, які ефективно заморожували свої грошові кошти на своїй базі гравців. Full Tilt був звинувачений в DoJ, який виступав у ролі схеми Понзі і виграв з 300 мільйонів доларів. З іншого боку, PokerStars негайно заплатило штраф у розмірі 1 мільярда доларів.

Багато онлайн-покер сайтів стимулюють гравців, особливо нових вкладників, у вигляді бонусів. Зазвичай бонуси виплачуються поступово, залежно від того, як гравець заробляє певні суми. Наприклад, сайт може запропонувати гравцю, який розміщує 100 доларів США бонус у розмірі 50 доларів, який нараховує 5 доларів кожен раз, коли гравець заробить рейком 25 доларів. Щоб заробити всю бонусну суму в розмірі 50 доларів, гравець повинен виручити поркерному руму 250 доларів.

Крім того, у декількох онлайн-кабінетах розроблені VIP-програми для нагородження постійних гравців. Покерні кімнати часто пропонують додаткові бонуси для гравців, які бажають поповнювати свої рахунки. Вони відомі під

назвою «релоад бонусів». Також багато онлайнових номерів також пропонують рейкбек, а деякі пропонують покерну підтримку.

1.1.4 Інтернет-казино

Інтернет-казино в цілому пропонують проценти і процентні ставки окупності, які трохи вище, ніж наземні казино. Деякі публікують процентні аудити виплат на своїх веб-сайтах. Припускаючи, що онлайн-казино використовує правильно запрограмований генератор випадкових чисел, в настільних іграх, таких як блекджек, казино завжди має перевагу. Відсоток виплат для цих ігор встановлюється правилами гри.

Багато онлайн-казино орендують або купують своє програмне забезпечення від таких компаній, як CryptoLogic Inc, International Game Technology, Microgaming, Playtech та Realtime Gaming.

Існує декілька типів інтернет-казино:

- браузерні, тобто такі, що основані на web: Macromedia Flash, Shockwave, Java;
- настільні, тобто такі, для яких потрібно завантажувати та встановлювати клієнт;
- віртуальні (скриптові) ігри, результат яких визначений наперед генератором псевдовипадкових чисел;
- казино з живими круп’є, тобто такі, в яких справжні казино знімаються і транслюються гравцям.

Більшість казино заманює нових гравців бонусними програмами, що є формою маркетингу. До основних типів належать:

- бонус за перший депозит;
- бонус за реферального «друга»;

- кешбек;
- бездепозитні бонуси;
- нагородження балами.

1.1.5 Ставки на спорт

Спортивні ставки - це діяльність із прогнозування спортивних результатів та розміщення ставки на результат. Переважна більшість ставок поширюється на футбольні асоціації, американський футбол, баскетбол, бейсбол, хокей, велоспорт, автоперегони, змішане бойове мистецтво та бокс як на любительському, так і на професійному рівнях. Спортивні ставки також можуть поширюватися на спортивні заходи, такі як конкурси, реаліті-шоу та політичні вибори, а також не-людські змагання, такі як скачки, та незаконні собачі бої.

Гравці ставлять свої ставки як на законних підставах, через букмекерську/спортивну книгу, так і незаконно через приватні підприємства. Термін "книга" - це посилання на книги, які використовуються найманими брокерськими компаніями для відстеження ставок, виплат та боргів. Більшість легальних букмекерських контор функціонують онлайн. Вони беруть ставки "наперед", а це означає, що учасник повинен сплатити букмекеру перед розміщенням ставки. Незаконні букмекери, в залежності від характеру свого бізнесу, можуть працювати буквально в будь-якому місці, але вимагають лише грошей у випадку, якщо ставка не зіграла, що створює можливість боргу букмекеру. Це створює ряд інших кримінальних елементів, що сприяє підвищенню незаконності.

1.1.6 Онлайн бінго та лотереї

На відміну від кульок, які використовуються в звичайних залах бінго, сайти онлайн-бінго використовують генератор випадкових чисел. Більшість залів із бінго також пропонують посилання на онлайн-покер та казино, оскільки заядлі гравці в бінго часто являються їхньою цільовою аудиторією. Одна помітна особливість онлайн-бінго - це функціонал спілкування за допомогою чату. Бінго сайтів прагнуть сприяти почуттю спільноти та взаємодії між гравцями, оскільки це допомагає утримувати клієнтів.

1.2 Функціонування компанії в умовах невизначеності

Компанія, що надає послуги онайн-ігор постійно перебуває в умовах невизначеності. Функціонування будь-якої компанії передбачає існування найрізноманітніших видів невизначеності для всіх суб’єктів господарювання. Найпоширенішою є класифікація невизначеності за ступенем настання події. Ця класифікація дає можливість розрізнити повну та часткову невизначеність, повну визначеність. Часткова чи повна невизначеність пояснюється тим, що, по суті, економічні проблеми зводяться до задач вибору з деякої кількості альтернатив. При цьому економічні суб’єкти не мають повної інформації про стан систем для розробки оптимального рішення й достатніх можливостей для адекватного обліку всіх доступних даних. Невизначеність інформації можливо зняти, визначивши ймовірність, з якою можна очікувати цю інформацію. Залежно від засобів визначення ймовірності розрізняють два типи невизначеності – статистичну та нестатистичну. Якщо мається на увазі статистична невизначеність, то іноді кажуть, що рішення приймається в умовах

ризику, якщо нестатистична – то рішення приймається в умовах невизначеності. У чистому вигляді той чи інший вид імовірності трапляється рідко – найчастіше можна зустріти мішаний вид [16].

До основних причин, що породжують цю невизначеність належать:

- перепади навантаження на сервер;
- неможливість точно спрогнозувати фінансові показники;
- відтік клієнтів як випадковий процес.

1.2.1 Навантаження на сервер

Однією із причин невизначеності діяльності є хаотичний характер навантаженості сервера. По-перше, неможливо точно спрогнозувати кількість активних гравців та протяжність сесії кожного.

По-друге, можливі різноманітні атаки на сервер. Найпоширеніша з них – це атака на відмову в обслуговуванні та розподілена атака на відмову в обслуговуванні.

Одним із найпоширеніших методів нападу є насичення атакованого комп'ютера або мережевого устаткування великою кількістю зовнішніх запитів (часто безглуздих або неправильно сформульованих) таким чином атаковане устаткування не може відповісти користувачам, або відповідає настільки повільно, що стає фактично недоступним [4]. Взагалі відмова сервісу здійснюється:

- примусом атакованого устаткування до зупинки роботи програмного забезпечення/устаткування або до витрат наявних ресурсів, внаслідок чого устаткування не може продовжувати роботу;

- заняттям комунікаційних каналів між користувачами і атакованим устаткуванням, внаслідок чого якість сполучення перестає відповідати вимогам.

Якщо атака відбувається одночасно з великої кількості IP-адрес, то її називають розподіленою.

По-третє, можливі різного роду форс-мажорні ситуації, що полягають у відмові обладнання або втраті зв'язку між серверами, блокування серверу або сервісу.

Такого роду невизначеності не розглядаються у даній роботі, оскільки вони носять яскраво виражений не статистичний характер і погано підлягають процедурі прогнозування. Найкращий спосіб боротьби з ними – це метод запасання та резервування, коли є декілька резервних серверів, що здатні розподілити навантаження за умови якось лиха.

1.2.2 Фінансові показники

Ще однією невизначеністю є значення майбутніх фінансових показників компанії. Будь-яка організація повинна оцінювати свої фінансові показники, щоб планувати майбутні витрати та обирати стратегію подальшого розвитку. Якщо компанія взагалі не розуміє на який прибуток розраховувати, то вона не може обрати правильний розподіл витрат і наражає себе на небезпеку стагнації.

Отож очевидною є доцільність прогнозування фінансових показників компанії. Для цього є безліч інструментів. Починаючи з часових рядів і закінчуючи регресійними моделями та нейронними сітками. Однак найбільш точний прогноз можна отримати, моделюючи діяльність компанії на рівні індивідуальних її клієнтів. Особливо це стосується онлайнових платформ та провайдерів онлайн-ігор і гемблінгу.

У цій роботі показується статистичність характеру невизначеності, пов’язаної з фінансовими показниками, на прикладі прогнозованого доходу

компанії, шляхом зведення її до простої статистичної проблеми відтоку клієнтів.

1.2.3 Відтік клієнтів

Заманити нових клієнтів неважко. Набагато важче утримати їх і змусити користуватися продуктом тривалий час, змусити клієнтів повертатися до вас. Це так звана проблема ретеншену, що пов’язана із «відмиранням» когорт клієнтів.

Невизначеність ця пов’язана з тим, що досить важко сформувати портрет клієнта. Однак в даній роботі продемонстровано, як її можна позбутися звичайними статистичними інструментами.

1.3 Опис даних

Отож на вході маємо 150 тис. спостережень за діями гравців. При кожному спостереженні вимірюються наступні змінні:

- *user_id* – унікальний ідентифікатор гравця;
- *data* – день спостережння;
- *bets_cnt, wins_cnt* – кількість ставок та кількість виграшів відповідно;
- *bets_sum, wins_sum* – сума ставок та сума виграшів відповідно;
- *ng_sum* – величина програшу;

- *buyin_sum, rebuyin_sum, cashout_sum, payout_sum, pokerbet_sum, rake_sum* – суми внесків, повторних внесків, доходу, виграшів, ставок і рейку відповідно;
- *casino_hour_cnt, poker_hour_cnt* – кількість зіграних годин в казино і покер відповідно;
- *bonus_sum* – сума нарахованих бонусів;
- *dep_in_cnt, dep_out_cnt* – кількість депозитів та виводів;
- *dep_in_sum, dep_out_sum* – сума депозитів та виводів;
- *is_only_bonus* – ознака того, що в цей день клієнту лише нараховувалися бонуси, тобто з його боку не було ніякої активності;
- *avg_seats* – середнє арифметичне кількості місць за столами, якими він грав;
- *cum_rake* – кумулятивний рейк;
- *age* – вік (у днях);
- *cum_avg_rake, cum_bonus_sum, cum_avg_bonuses, cum_deps_in_cnt, cum_dep_in_sum, cum_dep_in_sum_over_cum_dep_in_cnt, cum_dep_out_cnt, cum_dep_out_sum, cum_dep_out_sum_over_cum_dep_out_cnt, cum_ng_sum* – кумулятивні значення середнього рейку, бонусів, виводів, депозитів тощо;
- *days_from_last_dep* – кількість днів після останнього депозиту.

Усього 42 змінні (див. рисунок 1.1)

```
$ user_id                          : Factor w/ 421 levels "53edff7b89051d144990579b",..: 1 1 1 1 1 1 1 1 1 1 ...
$ date                             : Date, format: "2014-08-15" "2014-08-16" "2014-08-17" "2014-08-19" ...
$ bets_sum                         : num  0 0 0 0 0 0 0 0 0 0 ...
$ wins_sum                         : num  0 0 0 0 0 0 0 0 0 0 ...
$ buyin_sum                        : num  58.2 0 0 0 0 ...
$ rebuyin_sum                      : num  0 0 0 0 0 0 0 0 0 0 ...
$ cashout_sum                      : num  54.0612 0 0 0.0183 0 ...
$ payout_sum                       : num  69.6453 0 0 0.0183 0 ...
$ pokerbet_sum                     : num  73.8 0 0 0 0 ...
$ rake_sum                         : num  3.59 0 0 0 0 ...
$ casino_hour_cnt                  : num  0 0 0 0 0 0 0 0 0 0 ...
$ poker_hour_cnt                   : num  7 3 5 4 3 5 6 5 4 1 ...
$ dep_in_cnt                       : num  1 0 0 0 0 0 0 0 0 0 ...
$ dep_in_sum                       : num  4.16 0 0 0 0 ...
$ dep_out_cnt                      : num  0 0 0 0 0 0 0 0 0 0 ...
$ dep_out_sum                      : num  0 0 0 0 0 0 0 0 0 0 ...
$ age                              : num  1 2 3 5 6 9 11 12 13 37 ...
$ cum_avg_rake                     : num  3.591 1.796 1.197 0.718 0.599 ...
$ quart_dep_out_sum                : num  1 4 4 4 4 4 4 4 4 4 ...
$ quart_dep_in_sum                 : num  1 4 4 4 4 4 4 3 4 4 ...
$ quart_wins_sum                   : num  2 3 2 4 2 2 4 4 4 4 ...
$ quart_bets_sum                   : num  2 3 2 3 2 2 3 3 3 3 ...
$ cum_avg_bonuses                  : num  0 0 0 0 0 0 0 0 0 0 ...
$ cum_dep_in_cnt                   : num  1 1 1 1 1 1 1 1 1 1 ...
$ cum_dep_in_sum_over_cum_dep_in_cnt  : num  4.16 4.16 4.16 4.16 4.16 ...
$ cum_dep_out_cnt                  : num  0 0 0 0 0 0 0 0 0 0 ...
$ cum_dep_out_sum_over_cum_dep_out_cnt: num  0 0 0 0 0 0 0 0 0 0 ...
$ cum_ng_sum                       : num  0 0 0 0 0 0 0 0 0 0 ...
$ is_poker                         : num  1 1 1 1 1 1 1 1 1 1 ...
$ is_casino                        : num  0 0 0 0 0 0 0 0 0 0 ...
$ last_poker                       : num  0 0 0 0 0 0 0 0 0 0 ...
$ last_casino                      : num  0 0 0 0 0 0 0 0 0 0 ...
```

Рисунок 1.4 – Фрагмент стандартного виводу даних середовища R

Висновки. Постановка задачі дипломного проекту

На основі вхідних даних, описаних в підрозділі 1.3, для прогнозування фінансових показників компанії, що надає послуги онлайн-гемблінгу:

- звести невизначеність фінансових показників до невизначеності, пов’язаної з відтоком клієнтів;
- проаналізувати характер відтоку клієнтів та декомпонувати на елементи, що можна розв’язати, використовуючи існуючий статистичний інструмент;
- визначити найкращі моделі прогнозування відтоку клієнтів;
- синтезувати результат і зробити висновки;
- спроектувати та побудувати прототип СППР для розв’язку таких задач у майбутньому.

РОЗДІЛ 2 ОПИС МАТЕМАТИЧНОГО АПАРАТУ

2.1 Методи кластеризації та класифікації

Задача класифікації — формалізована задача, яка містить множину об'єктів (ситуацій), поділених певним чином на класи. Задана скінченна множина об'єктів, для яких відомо, до яких класів вони відносяться. Ця множина називається вибіркою. До якого класу належать інші об'єкти невідомо. Необхідно побудувати такий алгоритм, який буде здатний класифікувати довільний об'єкт з вихідної множини.

Класифікувати об'єкт — означає, вказати номер (чи назву) класу, до якого відноситься даний об'єкт.

Класифікація об'єкта — номер або найменування класу, що видається алгоритмом класифікації в результаті його застосування до даного конкретного об'єкту.

В математичній статистиці задачі класифікації називаються також задачами дискретного аналізу. В машинному навчанні завдання класифікації вирішується, як правило, за допомогою методів штучної нейронної мережі при постановці експерименту у вигляді навчання з учителем.

Існують також інші способи постановки експерименту — навчання без вчителя, але вони використовуються для вирішення іншого завдання — кластеризації або таксономії. У цих завданнях поділ об'єктів навчальної вибірки на класи не задається, і потрібно класифікувати об'єкти тільки на основі їх подібності. У деяких прикладних областях, і навіть у самій математичній статистиці, через близькість завдань часто не відрізняють завдання кластеризації від завдання класифікації.

Деякі алгоритми для вирішення задач класифікації комбінують навчання з учителем і навчання без вчителя, наприклад, одна з версій нейронних мереж

Кохонена — Мережі векторного квантування, яких навчають способом навчання з учителем.

2.1.1 Математичне формулювання завдання

Нехай X — множина описів об'єктів, Y —множина номерів (чи назв) класів. Існує невідома цільова залежність – відображення $y^*: X \to Y$, значення якої відомі лише на елементах скінченної навчальної вибірки $X^m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$. Потрібно побудувати алгоритм $a: X \to Y$, здатний класифікувати довільний об'єкт $x \in X$.

Сформулювати задачу класифікації також можна в термінах імовірнісної міри. Припускається, що множина пар «об'єкт, клас» $X \times Y$ є ймовірнісним простором з невідомою ймовірнісною мірою P. Є скінченна навчальна вибірка спостережень $X^m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, згенерована згідно з ймовірнісною мірою P. Необхідно побудувати алгоритм $a: X \to Y$, здатний класифікувати довільний об'єкт $x \in X$.

2.1.2 Регресійні моделі

Найбільш розповсюдженим методом класифікації є логістична регресія. Найбільш розповсюджена логістична регресія – це бінарна (коли вихід може набувати лише двох значень), однак є більш загальні моделі, що розглядають ситуації полізначних змінних виходу.

Логістична регресія обходить проблему не гаусівського розподілу та не лінійності, використовуючи логіт-перетворення залежної змінної:

$$f(\mathbb{E}(y|x)) = f(p) = \ln\left(\frac{p}{1-p}\right) = \beta^T x \tag{2.1}$$

де $Y \in \{0,1\}$.

Функцію f у такому випадку називають логіт-перетворенням, а відношення $\frac{p}{1-p}$ – шансами (з англ. odds).

$$p = \mathbb{E}(y|x) = \frac{e^{\beta^T x}}{1 + e^{\beta^T x}} = \sigma\left(e^{\beta^T x}\right) \tag{2.2}$$

Функцію σ називають сигмоїдою, або логістичною функцією. Важливою особливістю цієї функції є її область значень: $E(\sigma) = [0; 1]$, - що як ніяк краще підходить для оцінювання імовірності.

Таким чином з припущення:

$$\mathbf{P}(y = 1|x) = \sigma(\beta^T x) \tag{2.3}$$

Що в іншій формі має вигляд: $\mathbf{P}(y = 0|x) = 1 - \sigma(\beta^T x)$, одержуємо:

$$\mathbf{P}(y|x) = \left(\sigma(\beta^T x)\right)^y \left(1 - \sigma(\beta^T x)\right)^{1-y} \tag{2.4}$$

Нехай маємо вектор спостережень виходу та матрицю значень незалежних змінних: $Y \in \mathbf{R}^n$, $X = (X_1 X_2 \dots X_n) \in Mat(m \times n)$, - відповідно. Тоді функція правдоподібності та її натуральний логарифм матимуть вигляд:

$$L(\beta) = \prod_{i=1}^{n} \mathbf{P}(y = Y_i|x = X_i), \tag{2.5}$$

$$l(\beta) = \ln L(\beta) = \sum_{i=1}^{n} (Y_i \ln(\sigma(\beta^T X_i)) + (1 - Y_i)\ln(1 - \sigma(\beta^T X_i))) \tag{2.6}$$

Природнім чином постає задача максимізації цих функцій.

$$\frac{\partial}{\partial \beta_j}\left(Y_i \ln\left(\sigma(\beta^T X_i)\right) + (1 - Y_i)\ln\left(1 - \sigma(\beta^T X_i)\right)\right) = X_{ij}\left(Y_i - \sigma(\beta^T X_i)\right) \quad (2.7)$$

Звідси градієнт логарифмічної функції правдоподібності:

$$l'(\beta) = \nabla l(\beta) = \sum_{i=1}^{n}\left(Y_i - \sigma(\beta^T X_i)\right)X_i \quad (2.8)$$

Знайдемо її матрицю Гессе:

$$\frac{\partial}{\partial \beta_k}\left(Y_i - \sigma(\beta^T X_i)\right)X_i = -X_{ik}\sigma'(\beta^T X_i)X_i = -X_{ik}\sigma(\beta^T X_i)\left(1 - \sigma(\beta^T X_i)\right)X_i, \quad (2.9)$$

$$\frac{\partial}{\partial \beta_k}\nabla l(\beta) = -\sum_{i=1}^{n} X_{ik}\sigma(\beta^T X_i)\left(1 - \sigma(\beta^T X_i)\right)X_i \quad (2.10)$$

Тоді матриця Гессе матиме вигляд:

$$l''(\beta) = H(\beta) = -\left(D(\beta)\right)^T X = -X^T D(\beta)X, \quad (2.11)$$

де

$$D = diag(\sigma(\beta^T X_1 1)(1 - \sigma(\beta^T X_1 1)), \quad \sigma(\beta^T X_1 2)(1 - \sigma(\beta^T X_1 2)), \ldots, \sigma(\beta^T X_1 n)(1 - \sigma(\beta^T X_1 n))$$

.

Тепер можна знайти оцінку вектору β , наприклад, методом Ньютона:

$$\beta^* = \arg\max_{\beta} L(\beta) = \arg\max_{\beta} l(\beta) \quad (2.12)$$

Основна задача логістичної регресії – це коректно спрогнозувати категорію, в яку потрапляє, змінна виходу. Для побудови такої моделі можна

застосовувати покрокову побудову, коли якість моделі оцінюється разом з додаванням або, навпаки, видаленням можливих змінних-претендентів. Результат такого процесу – набір регресорів, що мають оптимальні властивості, такі як зміщення та варіація.

2.1.3 Класифікація без учителя

Будь-який регресійний підхід потребує існування тренувальної вибірки, тобто історичних спостережень, що вже так чи інакше класифіковані. Однак дуже часто виникають випадки, коли дослідник апріорі не знає ні самі класи, в які попадають спостереження, а інколи й навіть кількість класів. У таких випадках застосовують методи кластеризації, або класифікації без учителя [17].

Кластерний аналіз – задача розбиття заданої вибірки об'єктів (ситуацій) на підмножини, що називаються кластерами, так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися. Завдання кластеризації відноситься до статистичної обробки, а також до широкого класу завдань навчання без вчителя.

Кластерний аналіз — це багатовимірна статистична процедура, яка виконує збір даних, що містять інформацію про вибірку об'єктів і потім упорядковує об'єкти в порівняно однорідні групи — кластери (Q-кластеризація, або Q-техніка, власне кластерний аналіз).

Основна мета кластерного аналізу — знаходження груп схожих об'єктів у вибірці. Спектр застосувань кластерного аналізу дуже широкий: його використовують в археології, антропології, медицині, психології, хімії, біології, державному управлінні, філології, маркетингу, соціології та інших дисциплінах. Однак універсальність застосування привела до появи великої кількості

несумісних термінів, методів і підходів, що ускладнюють однозначне використання і несуперечливу інтерпретацію кластерного аналізу [17].

Нехай X — множина об'єктів, Y — множина номерів (імен, міток) кластерів. Задано функцію відстані між об'єктами $\rho(x, x')$. Є кінцева вибірка об'єктів $X^m = \{x_1, ..., x_m\} \subset X$. Потрібно розбити вибірку на непересічні підмножини, що називаються кластерами, так, щоб кожен кластер складався з об'єктів, близьких по метриці ρ, а об'єкти різних кластерів істотно відрізнялися. При цьому кожному об'єкту $x_i \in X^m$ приписується номер кластеру y_i.

Алгоритм кластеризації — це функція $a: X \to Y$, яка будь-якому об'єкту $x \in X$ ставить у відповідність номер кластера $y \in Y$. Множина Y в деяких випадках відома заздалегідь, проте частіше ставиться завдання визначити оптимальне число кластерів, з погляду деякого критерію якості кластеризації.

Об'єднання схожих об'єктів у групи може бути здійснене різними способами. Саме для цього етапу існує цілий ряд методів:

- K-means;
- C-means;
- графові алгоритми кластеризації;
- статистичні алгоритми кластеризації;
- алгоритми сімейства FOREL;
- ієрархічна кластеризація;
- нейронна мережа Кохонена;
- ансамбль кластеризаторів;
- EM-алгоритм.

Розглянемо 2 найбільш популярних із них.

2.1.3.1 K-Means

Нехай задана вибірка спостережень $x = (x_1, x_2, ... x_n)$, де кожне спостереження $x_i \in R^d$, алгоритм k-середніх на те, щоб розбити n спостережень на k множин $S = \{S_1, S_2, ..., S_k\}$ таким чином, щоб мінімізувати варіацію всередині кожного із класів. Формально цільову функцію можна записати наступним чином:

(2.13)

де μ_i – математичне сподівання точок в S_i. Це еквівалентно мінімізації попарної варіації точок одного кластеру:

(2.14)

Еквівалентність можна отримати з рівності

$$\sum_{x \in S_i} \|x - \mu_i\|^2 = \sum_{x \neq y \in S_i} (x - \mu_i)(\mu_i - y) \quad (2.15)$$

А оскільки загальна варіації є константою, то ця умова є також еквівалентною максимізації варіації між кластерами.

2.2 Методи оцінки ймовірності відпаду

2.2.1 Застосування дерев рішень

Дерева рішень являються досить потужним інструментом для оцінки ймовірності відпаду. Основними поняттями дерев рішень є:

- вузли;
- гілки;
- листки.

В «листках» записані значення цільової функції. На «гілках» записані атрибути, від яких залежить цільова функція, а в інших вузлах – атрибути, за якими розрізняють випадки. Для того, щоб отримати значення цільової функції потрібно спуститися від кореню до листка, слідуючи шляху гілок відповідно до вхідних даних. Ціль полягає в тому, щоб створити модель, яка б давала значення на множині дійних чисел залежно від цілого вектору значень незалежних змінних. Приклад дерева рішень на рисунку 2.1.

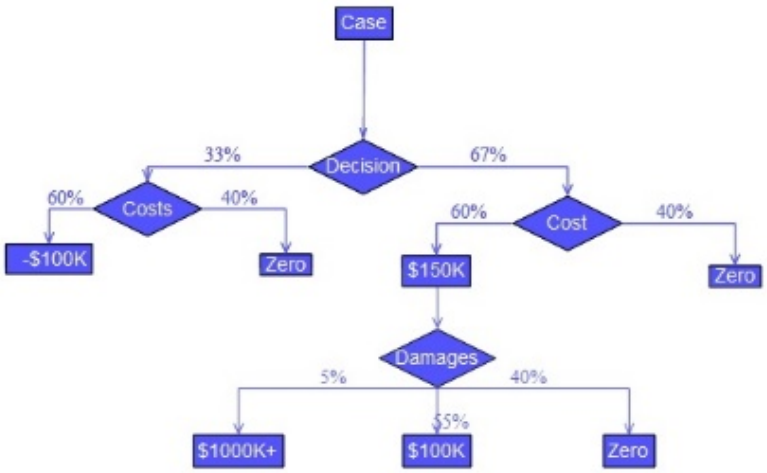


Рисунок 2.1 – Приклад простого дерева рішень

Дерева рішень, що використовуються в інтелектуальному аналізі даних бувають двох типів:

- дерева для класифікації, коли значення цільової функції – це клас, до якого належить спостереження;

- дерева для регресії, коли цільова функція може приймати значення на множині дійсних чисел, наприклад: ціна на житло, кількість років життя тощо.

Вище згадані терміни були вперше введені Брейманом та іншими [5].

2.2.1.1 Алгоритми побудови дерев

Загальна схема побудови дерева виглядає наступним чином:

1. Обираємо черговий атрибут Q та поміщаємо його в корінь.
2. Для всіх його значень і:
    a. Залишаємо з тестових прикладів лише ті, для яких значення атрибуту Q має значення і.
    b. Рекурсивно будуємо дерево в його потомку.

Основне питання – це яким чином обирати черговий атрибут.

Є різні способи обирання наступного атрибуту:

- алгоритм ID3, коли вибір атрибуту основується на максимізації приросту інформації (gain) або на основі критерію Джині.
- алгоритм C4.5, коли атрибут обирається на основі нормалізованого приросту інформації (gain ration).
- алгоритм CART та ін.

Кожен алгоритм використовує певну метрику однорідності цільової змінної в межах підмножин. Ці метрики застосовують для кожного кандидату-підмножини, а потім комбінуються (наприклад, усередненням) для визначення міри якості розділу. До основних метрик при виборі найкращого атрибут і його поділу належать:

- Gini impurity – є мірою того, як часто елемент, обраний випадковим чином з множини, буде невірно(!) класифікованим, якщо клас, до якого він віднесений, відповідає розподілу класів в підмножині. Для обрахунку цієї метрики для множини із J класів, в якій p_i – доля елементів, що належить до і-го класу застосовується наступна формула:

$$I_G(p) = \sum_{i=1}^{J} p_i \sum_{k \neq i} (1 - p_k) = \sum_{i=1}^{J} p_i (1 - p_i) = 1 - \sum_{i=1}^{J} p_i^2 \qquad (2.16)$$

- приріст інформації (information gain):

$$IG(T, a) = H(T) - H(T, a) =$$

$$= -\sum_{i=1}^{J} p_i \log p_i - \sum_{a} p(a) \sum_{i=1}^{J} p(i|a) \log p(i|a) \qquad (2.17)$$

- зменшення варіації (variance reduction):

$$I_V(N) = \frac{1}{|S|^2} Var(S) - \left(\frac{1}{|S_f|^2} Var(S_f) + \frac{1}{|S_t|^2} Var(S_t) \right) \qquad (2.18)$$

де $Var(S) = \sum_{i \in S} \sum_{j \in S} \frac{1}{2} (x_i - x_j)^2$

Незважаючи на свою універсальність дерева рішень мають ряд недоліків:

- зазвичай являються менш точними, порівняно з іншими методами [6];
- не являються робастними [6];
- задача побудови оптимального дерева є NP-повною;
- дерева можуть швидко стати занадто складними і погано узагальнюватися від тренувальної вибірки, тобто вони часто піддаються перенавчанню.

Для подолання вище зазначених проблем застосовуються ансамблі дерев.

2.2.2 Ансамблі дерев рішень

2.2.2.1 Bootstrap aggregating (bagging)

Нехай дана множина для побудови моделі D розміром n. Бутстреп агрегація генерує m нових множин D_i, кожна розміром n', відбіром вибірки з D рівномірно і з повторами. Усі m вибірок використовуються для побудови дерев, а потім об'єднуються усередненням (для регресій) або голосуванням (для класифікації).

Таким чином концептуально алгоритм bagging можна зобразити наступним чином.

1. m разів відбираємо вибірку (X_i, Y_i) з головної вибірки (X, Y), $i = 1, m$.
2. Будуємо m дерев f_i на сформованих вибірках (X_i, Y_i).
3. Після побудови прогнозне значення можна отримати, усереднюючи значення окремих дерев: $\widehat{f(x')} = \frac{1}{m}\sum_{i=1}^{m} f_i(x')$ або використовуючи правило більшості голосів у випадку класифікації.

Процедура бутстрепінгу веде до кращих результатів, тому що зменшує варіацію моделі без збільшення зміщення. Це означає, що в той час, як прогноз кожного окремого дерева є чутливим до шуму в тренувальній множині, середнє для сукупності багатьох дерев, - ні, якщо забезпечується відсутність кореляції між деревами. Відбір вибірки за допомогою бутстрепінгу – це лише один із методів позбування кореляції.

На додачу можна отримати оцінку невизначеності прогнозу як стандартне відхилення прогнозів усіх регресійних дерев:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{m} \left(f_i(x') - \hat{f}(x')\right)^2}{m-1}}, \quad (2.19)$$

де m – кількість дерев – вільний параметр.

Зазвичай використовують від декількох сотень до декількох тисяч дерев залежно від розміру вибірки. Оптимальну кількість таких дерев можна знайти використовуючи перехресну валідацію або розраховуючи out-of-bag error.

2.2.2.2 Random forest

Випадковий ліс відрізняється від звичайного алгоритму bagging лише однією особливістю: він використовує модифікований алгоритм навчання дерев. Модифікація полягає в тому, що на кожному етапі вибору атрибуту для розгалуження використовується випадкова підмножина атрибутів. Цей процес інколи називають bagging атрибутів. Основна ідея для такого алгоритму – це кореляція між деревами в звичайному bagging алгоритму. Якщо один або декілька атрибутів є сильними предикторами, то вони будуть входити до багатьох дерев, призводячи до того, що між ними з’явиться кореляція

2.2.3 Градієнтний бустинг

Як і будь-який інший метод бустингу, градієнтний бустинг поєднує слабкі моделі в одну сильну модель ітеративним чином. Розглянемо задачу

побудови регресії $y^* = F(x)$, що мінімізує середньо квадратичну похибку

.

На кожному кроці m, $1 \leq m \leq M$, градієнтного бустингу можна припустити, що існує певна неідеальна модель F_m (на початку використовується найслабша модель, значення якої – середнє арифметичне усіх спостережень). Алгоритм градієнтного бустингу покращує модель F_m наступним чином: будується ще одна модель h і додається до попередньої:

$$F_{m+1}(x) = F_m(x) + h(x) \tag{2.20}$$

Для знаходження h, метод градієнтного бустингу використовується той же критерій:

$$F_{m+1}(x) = F_m(x) + h(x) = y \tag{2.21}$$

Рівняння (2.32) є еквівалентним наступному:

$$h(x) = y - F_m(x) \tag{2.22}$$

Таким чином градієнтний бустинг буде навчати h на похибці попередньої моделі. Тобто кожна наступна модель намагається виправити свого нащадка.

Узагальнення цієї ідеї до довільної функції втрати (а не лише квадрату похибки) слідує із спостереження, що нев'язка $y - F(x)$ для певної моделі – це

від'ємний градієнт квадрату похибки . Отже, градієнтний

бустинг – це алгоритм градієнтного спуску, а його узагальнення дозволяє використовувати будь-яку функцію втрати.

2.2.3.1 AnyBoost

Нехай (x, y) – спостереження, елементи множини $X \times Y$, де X – простір вимірів, а Y - множина значень. Нехай H – певний простір функцій (слабких моделей) $X \to Y$ і $\langle , \rangle$ - скалярний добуток в цьому просторі, а також функціонал втрати на просторі H:

$$C : H \to R \tag{2.23}$$

Наша ціль – знайти функцію F, що мінімізує $C(F)$. Для цього застосуємо метод градієнтного спуску.

Припустимо, що є певна функція $F \in H$ і потрібно знайти нову $f \in H$, щоб при додаванні до F значення цільової функції $C(F + \epsilon f)$ зменшується при певному малому значенні ϵ. В термінах функціонального простору нам потрібно знайти напрямок f, такий, що $C(F + \epsilon f)$ найшвидше спадає. Потрібний напрямок – це просто від'ємний градієнт функції C в точці F : $-\nabla C(F)$.

Оскільки ми обмежені вибором нової функції f з простору H, в загальному неможливо обрати $f = -\nabla C(F)$, тому для знаходження f необхідно

$$\langle -\nabla C(F), f \rangle$$

максимізувати . Це можна обґрунтувати тим, що до першого порядку:

$$C(F + \epsilon f) = C(F) + \epsilon \langle \nabla C(F), f \rangle \tag{2.24}$$

Таким чином алгоритм, що дозволяє обрати f, намагаючись максимізувати $< -\nabla C(F), f >$ і породжує слабкі моделі. [7]

2.2.3.2 XGBoost

Наступні викладки описані в першопочатковій статті [8] і зводять задачу градієнтного спуску в функціональному просторі до більш елементарних кроків, що дозволяють абстрагуватися від цільових функцій і слабких моделей, а також максимально паралелізувати обчислення.

Для заданої вибірки із n спостережень та m змінних $D = \{(x_i, y_i)\} (|D| = n, x_i \in R^m, y_i \in R)$, модель ансамблів дерев застосовує K адитивних функцій для прогнозування виходу:

$$y_i^* = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{2.25}$$

де $F = \{f(x) = w_{q(x)}\} (q: R^m \to T, w \in R^T)$ – це простір регресійних дерев (CART);

q – визначає структуру кожного дерева як відображення із множини спостережень на множину листів;

T – кількість листів в дереві.

Кожна f_k відповідає незалежній структурі дерева q і вектору ваг листів w. На відміну від дерев рішень кожне регресійне дерево містить неперервний бал на кожному із листів (w_i – бал -го листа).

Для знаходження послідовності функцій, що використовуються в моделі, мінімізується наступна регуляризована цільова функція:

$$L(\phi) = \sum_i l(y_i^*, y_i) + \sum_k \Omega(f_k) \tag{2.26}$$

де $\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$;

l – довільна диференційовна випукла функція, що вимірює різницю між прогнозом y_i^* та ціллю y_i.

Модель ансамблю дерев містить функції як параметри тому не може бути оптимізована класичними методами оптимізації в Евклідових просторах. Для цього модель будується адитивно. Нехай $y_i^{*(t)}$ – прогнозне значення -го спостереження на t -му кроці. Потрібно додати f_t, щоб мінімізувати наступну цільову функцію:

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, y_i^{*(t-1)} + f_t(x_i)\right) + \Omega(f_t) \tag{2.27}$$

Для спрощення цього завдання використовується наближення 2-го порядку:

$$L^{(t)} \approx \sum_{i=1}^{n} \left[l\left(y_i, y_i^{*(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{2.28}$$

де $g_i = \partial_{y^{*(t-1)}} l\left(y_i, y^{*(t-1)}\right)$, $h_i = \partial^2_{y^{*(t-1)}} l\left(y_i, y^{*(t-1)}\right)$ – градієнтні статистики цільової функції відповідно 1-го і 2-го порядку.

Для спрощення можна позбутися константи і отримати наступну ціль на кожному етапі t :

$$L^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{2.29}$$

Нехай тепер $I_j = \{i | q(x_i) = j\}$ – множина спостережень листа j. Перепишемо рівняння (2.50) наступним чином:

$$L^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 =$$

$$= \sum_{j=1}^{T} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T \tag{2.51}$$

Для фіксованої структури $q(x)$ можна знайти оптимальну вагу листа w_j^*:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{2.51}$$

Тоді оптимальне значення цільової функції:

$$L^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{\left(\sum_{i \in I_j} g_i \right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T \tag{2.52}$$

Рівняння (2.52) можна використовувати як метрику якості структури q. Однак зазвичай неможливо перебрати всі можливі структури дерев q. Тому застосовують жадібний алгоритм, який на кожному кроці вибирає найкращий

поділ. Нехай I_L, I_R – це відповідно ліва і права підмножини після поділу. Якщо $I = I_L \cup I_R$, то зменшення значення цільової функції можна обчислити за допомогою:

$$L_{split} = \frac{1}{2}\left[\frac{\left(\sum_{t \in I_L} g_t\right)^2}{\sum_{t \in I_L} h_t + \lambda} + \frac{\left(\sum_{t \in I_R} g_t\right)^2}{\sum_{t \in I_R} h_t + \lambda} - \frac{\left(\sum_{t \in I} g_t\right)^2}{\sum_{t \in I} h_t + \lambda}\right] - \gamma = \quad (2.53)$$

2.3 Методи оцінки часу відтоку

2.3.1 Аналіз виживання

Аналіз виживання використовується для дослідження моменту загибелі деякої популяції. Час, що проходить до настання цього моменту, називається часом виживання.

Аналізу виживання передували таблиці смертності, що використовувалися в страхуванні життя та демографічних науках XVII ст. Це і призвело до вживання слова «виживання» у контексті рівня смертності. На початку свого існування метод таблиць смертності базувався на широких часових проміжках та великих об'ємах даних. Близько 1950-х Каплан та Мейер запропонували статистичну оцінку кривої виживання [12]. Вони розробили метод для коротких часових відрізків та менших вибірок, порівняно із тими, що використовувалася в демографічних дослідженнях. XX ст. характеризується подальшим розвитком методів обробки даних виживання.

Кокс в [12] запропонував метод, що дозволяв додавати коваріанти до аналізу подібних даних, що зараз відомий під назвою PH («Модель

пропорційних ризиків Кокса»). Така модель використовує регресори, що не залежать від часу або статичні змінні та припускає, відношення ризиків не змінюється із плином часу. Однак в реальних даних часто виникають саме характеристики, що змінюються із часом. Такі змінні порушують припущення про постійність відношення, тому модель Кокса була модифікована і доповнена. На сьогодні відомі її стратифікована та узагальнена модифікації.

2.3.1.1 Теоретичні засади аналізу виживання

Аналіз виживання дозволяє включати спостереження, що не дійшли до свого логічного завершення:

- вводиться поняття цензури даних.
- час спостереження закріплюється за такими випадками, що означає останній час, коли його спостерігали.

Загальний підхід до аналізу даних базується на використання функції ризику, значення якої відповідає ймовірності смерті в певний час:

$$h(t) = \lim_{\delta \to 0} \left(\frac{P(t \le T < t + \delta | T \ge t)}{\delta} \right) \quad (2.54)$$

де T – випадкова величина, що відповідає часу виживання.

Ймовірність виживання в певний час t може бути записаною в термінах функції ризику:

$$S(t) = \mathbb{P}(T \geq t) \tag{2.55}$$

Формулу можна трактувати, як ймовірність виживання в інтервалі часу з 0 до t . Звідси можна отримати ймовірність відвалу:

$$P_D(t) = 1 - S(t) \tag{2.56}$$

Що являється нічим іншим, як функцією розподілу $F = \mathbb{P}(T < t)$ випадкової величини T . Відповідно до формул:

$$S(t) = 1 - F(t), \tag{2.57}$$

$$f(t) = -\frac{dS(t)}{dt} \tag{2.58}$$

Використовуючи введену функцію ризику можна отримати ще один вираз для її обчислення:

$$h(t) = \frac{\lim_{\delta \to 0} \left(\frac{\mathbb{P}(t \leq T < t + \delta)}{\delta} \right)}{\mathbb{P}(T \geq t)} = \frac{f(t)}{S(t)} \tag{2.59}$$

де f – відома нам щільність випадкової величини T .

Звідси неважко отримати вирази для $h = h(S)$:

$$h(t) = -\frac{\frac{dS(t)}{dt}}{S(t)} \tag{2.60}$$

Розв’язуючи диференціальне рівняння отримуємо вираз для S , що залежить від h :

$$\int_0^t h(u)du = \int_0^t -\frac{S'(u)}{S(u)}du = \int_{S(0)}^{S(t)} -\frac{1}{s}ds = \log S(0) - \log S(t) =$$

$$= \log 1 - \log S(t) = -\log S(t),$$

(2.61)

$$S(t) = e^{-\int_0^t h(u)du} \tag{2.62}$$

Існує декілька підходів до оцінювання функції ризику. Найбільш розповсюджений – модель пропорціональних ризиків Кокса:

$$h(t, x(t), \beta) = h_0(t)e^{\beta^T x(t)} \tag{2.63}$$

Потрібно оцінити вектор коефіцієнтів β .

Якщо x не залежить від t , то:

$$\frac{h(t, x_1, \beta)}{h(t, x_2, \beta)} = e^{\beta(x_1 - x_2)} = const \tag{2.64}$$

Це пояснює слово «пропорціональні» в назві методу. Однак у випадку, коли коваріанти залежать від часу: $x = x(t)$, - це не так.

Виявляється, що для оцінки вектору параметрів β достатньо розглядати функцію часткової правдоподібності. Як імовірність того, що відбулася загибель певного конкретного індивіда i , за умови, що нам відомо про його загибель.

- t_i – час спостереження, тобто загибель або цензура;
- c_i – індикатор відвалу, $c_i = 1 \Leftrightarrow t_i$ – час загибелі.

Ймовірність того, що спостереження провалюється в певний час t серед інших спостережень обраховується наступним чином:

$$\frac{h(t, x_i(t), \beta)}{\sum_{j \in R(t_i)} h(t_i, x_j(t), \beta)} = \frac{e^{\beta^T x_i(t)}}{\sum_{j \in R(t)} e^{\beta^T x_j(t)}} \tag{2.65}$$

де $R(t) = \{j | t_j \geq t\}$.

Позначимо $\theta_j(t) = e^{\beta^T x_j(t)}$, $\theta_j^i = \theta_j(t_i)$. Тоді часткова функція правдоподібності та лог-перетворення від неї матимуть наступний вигляд:

$$L_p(\beta) = \prod_{i=1}^{n} \left(\frac{\theta_i^i}{\sum_{j \in R(t_i)} \theta_j^i} \right)^{c_i}, \tag{2.66}$$

$$l_p(\beta) = \log L_p(\beta) = \sum_{i:c_i=1} \left(\beta^T x_i(t_i) - \log \sum_{j \in R(t_i)} \theta_j^i \right) \tag{2.67}$$

Звідси можна отримати вирази для градієнта та матриці Гессе, що застосовуються при знаходженні оптимального значення. Маючи такі характеристики проводиться максимізація функції l_p для знаходження оцінки параметра β^* аналогічно.

Висновки

В другому розділі наведено основні теоретичні відомості щодо математичних методів та підходів до аналізу даних та моделювання поведінки клієнтів, що застосовуються при дослідженні ризиків відтоку клієнтів. Зокрема розглянуто такі сучасні моделі як ансамблі дерев, XGBoost. XGBoost є досить перспективним, однак малодослідженим об’єктом в управлінні відтоку клієнів.

У зв’язку з цим проведено побудова скоринг-моделей на основі пропорційних ризиків моделі XGBoost та проведено її тюнинг, а також порівняння з іншими більш простими моделями.

РОЗДІЛ 3 СКОРИНГОВІ МОДЕЛІ

В даному розділі проводить моделювання відпаду гравців, шляхом побудови бінарних класифікаторів спочатку на основі логістичної регресії, а потім на основі бустингового ансамблю дерев за допомогою алгоритму XGBoost. Завдяки використання ансамблю дерев вдалося значно покращити результат, що виражається в таких показниках, як AUC і точність класифікації – ROC-крива.

3.1 Цензорування даних та генерація змінних

Побудова моделі розпочиналася із цензорування та генерації незалежних змінних та цільового поля. Цензорування відбувалося формуванням цільового поля за ознакою того, чи повернеться гравець після певного фіксованого дня гри.

Далі відбувалася побудова цільового поля за наступною схемою:

- якщо гравець в день спостереження грав в покер, то це спостереження враховується у вибірці для покеру;
- якщо гравець в день спостереження грав в казино, то це спостереження враховується у вибірці для казино;

Таким чином одне спостереження може потрапити до обох вибірок.

Наступний логічний крок – це формування «міри відсутності» як перерва в днях між двома послідовними спостереженнями. При цьому пропуск вважається значимим, якщо його міра відсутності задовольняє 3 умовам:

- якщо клієнт був відсутнім більше 3 днів;

- якщо значення міри відсутності є екстремальним, а саме попадає в 0,975 квартиль розподілу усіх мір відсутності, що спостерігалися в певного гравця;
- максимальна перерва між іграми складає щонайменше 15 днів.

Для побудови моделей було згенеровано ряд лагових змінних, тобто значення спостережень із лагом в один день. Якщо гравець був відсутнім у цей день, то здійснювалася лінійна апроксимація між двома найближчими спостереженнями його гри. Наприклад, середній кумулятивний рейк рахувався наступним чином:

$$\frac{rake_{cavg_0} * age_0}{age_0 + absence} \tag{3.1}$$

3.2 Оцінка ймовірності відтоку клієнта

Далі відбувалася побудова моделей на основі цього цільового поля та вхідних даних за допомогою 2 типів моделей.

Для цього кожна вибірка (для покеру та для казино) поділилася на 2 частини: тренувальну та тестову. Поділ відбувався випадковим чином, але так, щоб тренувальна покривала 80% усіх даних. В якості метрики оцінювання використовувався AUC, а цільовою функцією виступала нев’язка логістичної регресії.

3.2.1 Логістична регресія

За базову модель для порівняння було взято звичайну логістичну регресію. Її побудова відбувалася із використанням формул (2.16)-(2.27), що імплементовані в стандартній бібліотеці мови програмування R.

Тест Стьюдента на значимість коефіцієнтів регресії показав, що більшість регресорів мало корелюють із цільовим полем. Більш детальний аналіз значимості регресорів наведено в п. 3.2.2.

Результат логістичної регресії (рисунок 3.1):

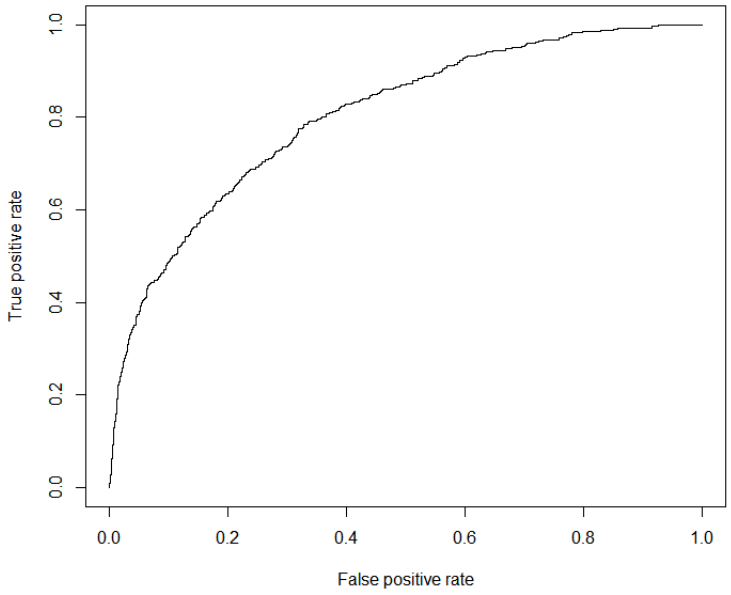


Рисунок 3.1 – ROC-крива логістичної регресії

Для покращення результатів було вирішено взяти ще більшу глибину лагу змінних, а саме: один тиждень. Однак це не дало очікуваних результатів. Більш детальне порівняння в п. 3.2.2.

3.2.2 XGBoost

Для побудови бустингового ансамблю дерев використовувався бібліотека xgboost з офіційного репозиторію.

3.2.2.1 Побудова моделей

Для підбору найоптимальніших значень параметрів моделі (тюнинг) здійснювався емпіричний пошук на сітці (max.depth; eta; nrounds) із кроками (1; 0,01; 1). Див рисунки 3.2-3.5.

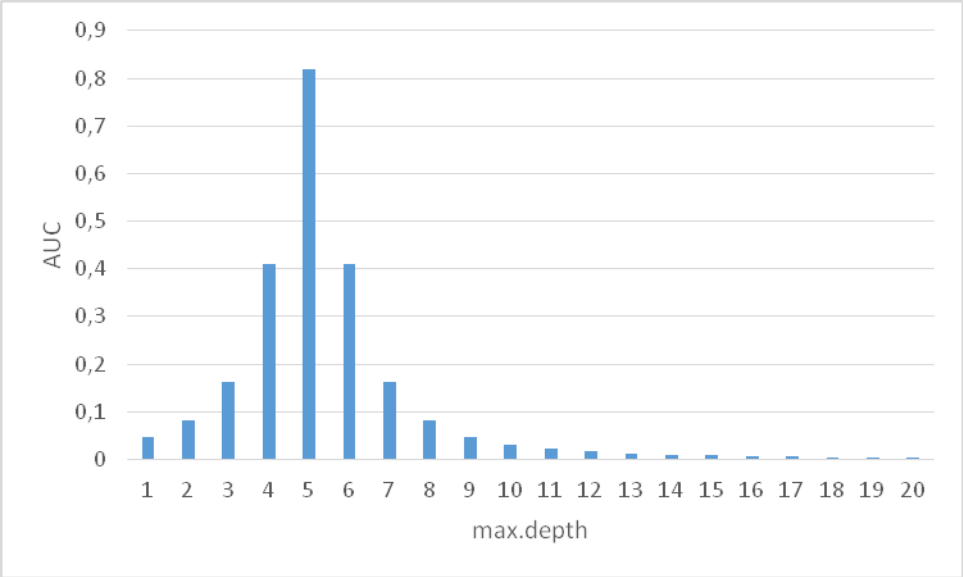


Рисунок 3.2 – Залежність AUC від параметру max.depth (покер)

Залежність між показником AUC і параметром max.depth має яскраво виражений екстремум в точці 5. Таке значення цього параметра і взято за остаточе, оскільки воно максимізує бажану точність.

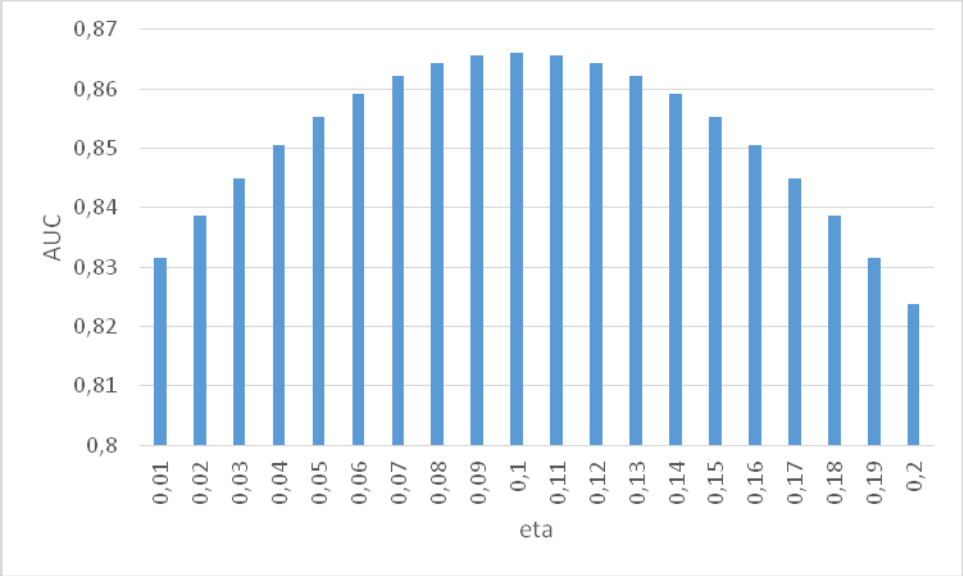


Рисунок 3.3 – Залежність AUC від параметру eta (покер)

Залежність між показником AUC і параметром eta має розподіл, схожий на нормальний. Максимум досягається в районі точки 0.1.

Побудова вібдувалася з використанням відповідних функцій бібліотеки, а значення параметрів підбиралися вручну.

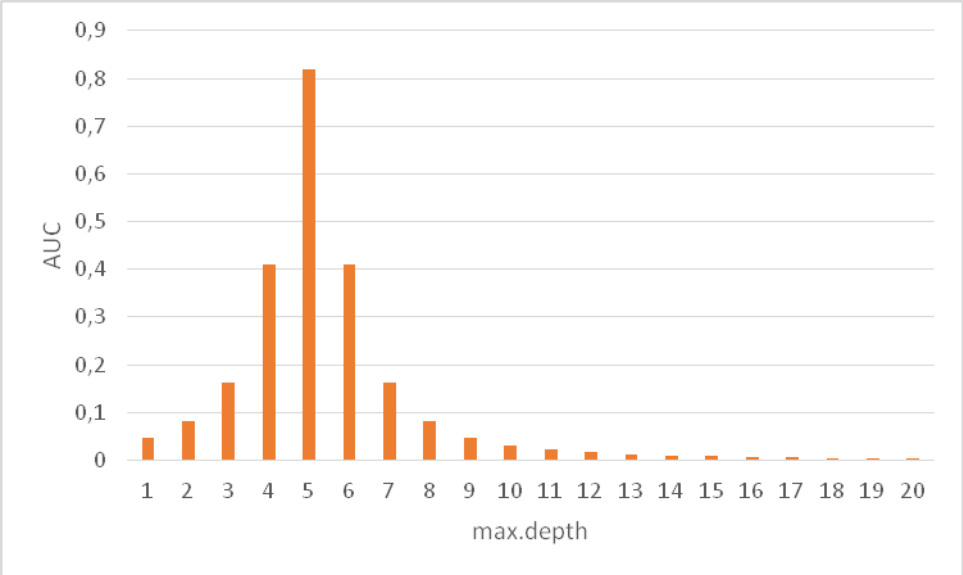


Рисунок 3.4 – Залежність AUC від параметру max.depth (казино)

Залежність для моделі гравців казино аналогічна до гравців покеру. Оптимальне значення параметру те, в якому досягається максимум.

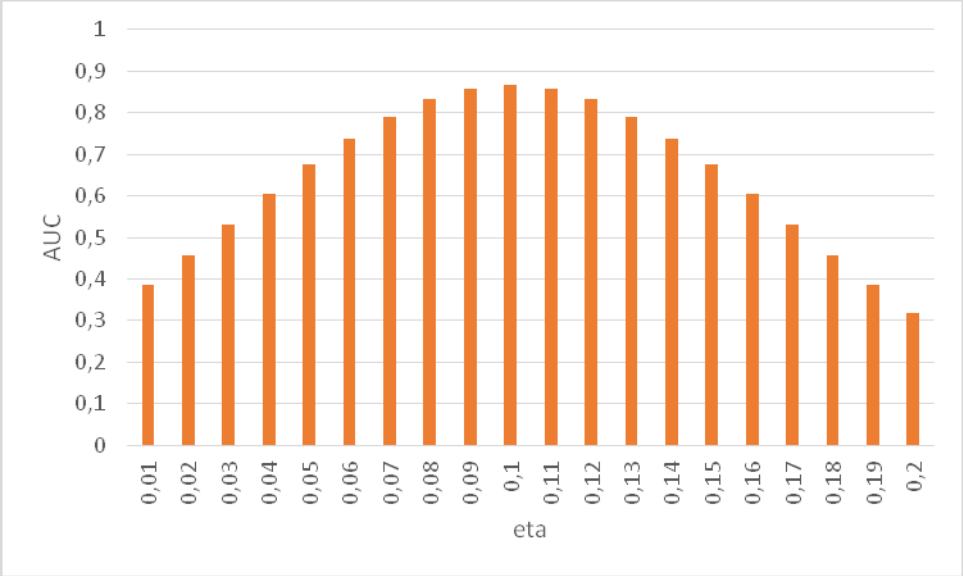


Рисунок 3.5 – Залежність AUC від параметру eta

3.2.2.2 Оцінка якості побудованих моделей

Порівняння регросорів відбувалося за середньозваженим приростом інформації (див. рисунок 3.6).

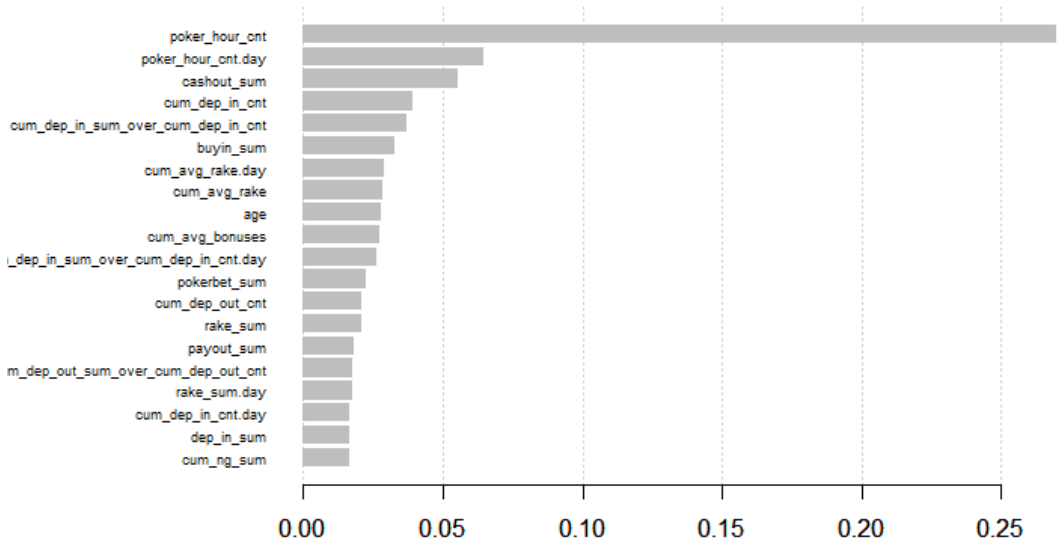


Рисунок 3.6 – Графік важливості регресорів (покер)

Найбільш важливим показником для моделі відтоку покерних гравців є кількість зіграних годин протягом дня. Також вижливими є наступні показники:

- кількість зіграних годин протягом попереднього дня;
- сума виводів;
- кумулятивна кількість депозитів;
- кумулятивне середнє значення величини депозиту;
- сума грошвих внесків;
- середнє значення кумулятивного рейку за попередній день та ін.

З огляду на список найбільш важливих параметрів можна сказати, що на ймовірність деофолту впливають не лише показники, що стосуються гри клієнта, але також і його фінансові показники.

Ітерації побудови моделі з оптимальними значеннями параметрів зображені на рисунку 3.7.

	iter	eval_auc	train_auc
1:	1	0.759258	0.755914
2:	2	0.763035	0.760968
3:	3	0.764583	0.763180
4:	4	0.766408	0.764485
5:	5	0.766565	0.765421

496:	496	0.865355	0.865666
497:	497	0.865526	0.865818
498:	498	0.865629	0.865890
499:	499	0.865690	0.865993
500:	500	0.865817	0.866121

Рисунок 3.7 - Перші та останні 5 кроків побудои моделі (покер)

Порівняння важливості регресорів моделі гравців в казино відбувалося по аналогії з гравцями в покер (див. рисунок 3.8).

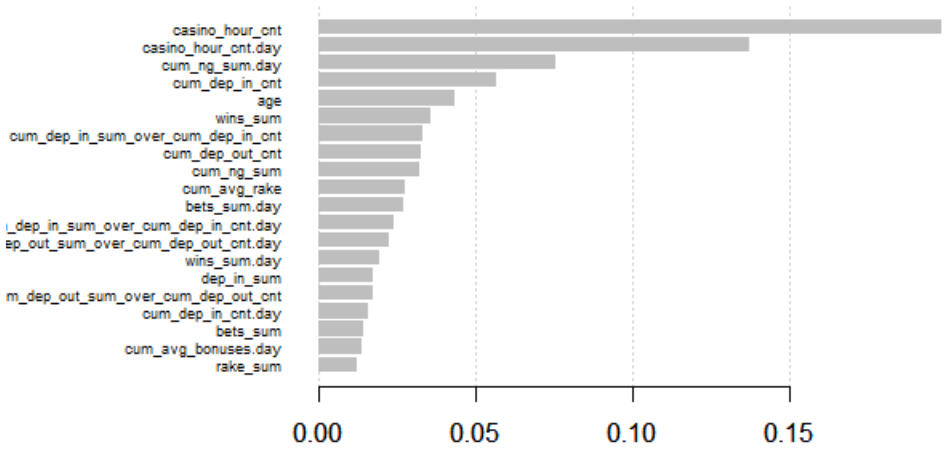


Рисунок 3.8 – Графік важливості регресорів (казино)

Найважливішим, як і у випадку моделі для гравців в покер, є кількість зіграних годин протягом дня. Однак на противагу покеру для гравців в казино також важливим є показник їхньої прибутковості, тобто скільки вони програли кумулятивно протягом всього свого життя, а також вік гравця. Варто також відзначити, що до одних із найбільш важливих показників потрапили і фінансові показники, такі як:

- сума виграшів;
- середня сума депозиту;
- кількість виводів;
- і що характерно середній кумулятивний рейк.

Ітерації побудови моделі, яких у випадку казино значно менше – 200, - з оптимальними значеннями параметрів зображені на рисунку 3.9.

	iter	eval_auc	train_auc
1:	1	0.703681	0.703509
2:	2	0.701701	0.705508
3:	3	0.709720	0.713998
4:	4	0.711215	0.714522
5:	5	0.710437	0.714352

196:	196	0.772752	0.817831
197:	197	0.772749	0.817970
198:	198	0.772964	0.818524
199:	199	0.772871	0.818680
200:	200	0.773151	0.819021

Рисунок 3.9 – Перші та останні 5 значень AUC для тренувальної та тестової вибірок

Загалом нам вдалося досягти досить високої точності оцінки ймовірносі відпаду за допомогою тюнигу бустингових ансамблів дерев. Чутливість цих моделей до такого налаштування, а також яскраво виражені оптимальні значення параметрів дозволяють досить гнучко будувати модель під будь-які потреби та дані.

3.2.2.3 Інтерпретація одержаних результатів

Одним із результатів цієї роботи є інтерпретація одержаних результатів моделі див. формулу (3.2). Оскільки в моделі оцінки ймовірності відтоку не враховується те, скільки днів гравець уже був відсутнім, через значну кореляцію із цільовим полем, то для формування остаточного балу використовується перетворення у вигляді функції, що приймає на вхід два параметри: ймовірність, одержану ансамблем дерев та фактичну кількість днів, що гравець був відсутнім:

$$s = \frac{1}{1 + e^{\frac{-(x-\mu)}{\sigma}}} \tag{3.2}$$

$$\mu = m \cdot \frac{\ln\frac{1-s_0}{s_0}}{\ln\frac{(1-s_0)s_1}{s_0(1-s_1)}}$$

де

$$\sigma = \frac{m}{\ln\frac{(1-s_0)s_1}{s_0(1-s_1)}};$$

n – кількість днів відсутності на момент розрахунку;

$$s_1(s_0) = \begin{cases} 0.95, & s_0 < 0.95 \\ s_0 + \frac{1-s_0}{2}, & s_0 \geq 0.95 \end{cases};$$

s_0 – початкова йомвіність, отрмана за допомогою моделі ансамблю дерев;

m – медіана значення s_0 для тестової вибірки.

3.2.3 Аналіз показників та порівняння моделей

Моделями xgboost вдалося досить добре спрогнозувати імовірність відпаду, про що свідчать значення AUC для моделей (див. таблиця 3.1).

Таблиця 3.1 – AUC на тренувальній та тестовій виборках

Тип гри	Тренувальна вибірка	Тестова вибірка
Покер	86,6%	86,5%
Казино	82%	77%

Порівняння з моделями логістичної регресії (М2 – один рівень лаговості, М3 – 2 рівня лаговості змінних). Як бачимо з таблиці 3.2 навіть збільшення лагів в регресорах логістичних моделей не приводить до значного покрещення їх точності. Моделі бустингових ансамблів дерев значно краще справляються із задачею.

Таблиця 3.2 – Порівння з моделями логістичної регресії

	M1	M2	M3
Покер	86%	71%	75%
Казино	82%	70%	72%

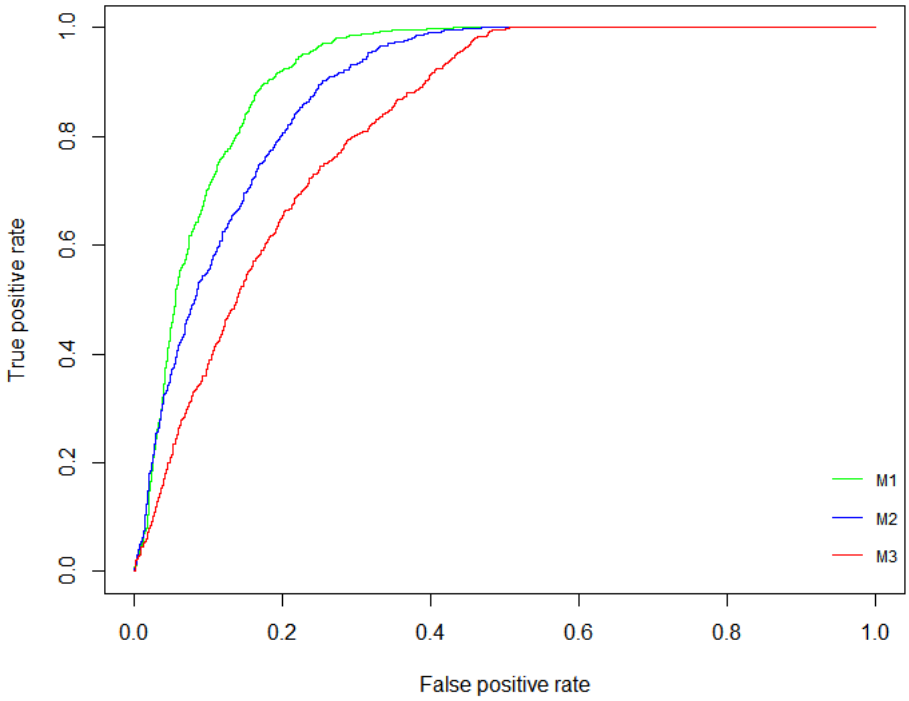


Рисунок 3.10 - Порівння з моделями логістичної регресії

Висновки

Отже, в цьому розділі було побудовано моделі ймовірності відпаду клієнтів за допомогою стандартних методів логістичної регресії та

спеціалізованих на основі градієнтного бустингу дерев. Для цього був розроблений власний алгоритм та підібрані оптимальні значення параметрів моделі. У результаті вдалося досягти дуже гарних результатів прогнозування імовірності відпаду клієнтів, про що свідчать показники AUC і ROC-криві.

РОЗДІЛ 4 АНАЛІЗ ВИЖИВАННЯ ТА КЛАСИФІКАЦІЯ

У даному розділі була здійснена спроба деталізувати портрет поведінки клієнта шляхом оцінювання не лише ймовірності відпаду, а й часу, коли цей відпад настане. Для цього було скомбіновано методи ієрархічної кластеризації та моделі аналізу виживання.

4.1 Постановка задачі

На вході маємо простір статистик 10 тис. гравців. Потрібно провести категоризацію клієнтів на основі кластеризації за допомогою ієрархічної кластеризації гравців гри «Покер» та побудувати модель виживання для кожного кластеру зважаючи на результати, отримані в [14].

Для аналізу використовується простір статистик гри в покер (див. рисунок 4.1, де зображено розподіл кожної із стаистик, нормованої в межах від 0 до 1), що складається зі 100 загально прийнятих статистик гри в покер. Для візуалізації також використовується зображення інтерквартильного розмаху (див. рисунок 4.2). Аналіз інтерквартильного розмаху може доповнювати класифікацію у випадку відшукання схожих поведінок.

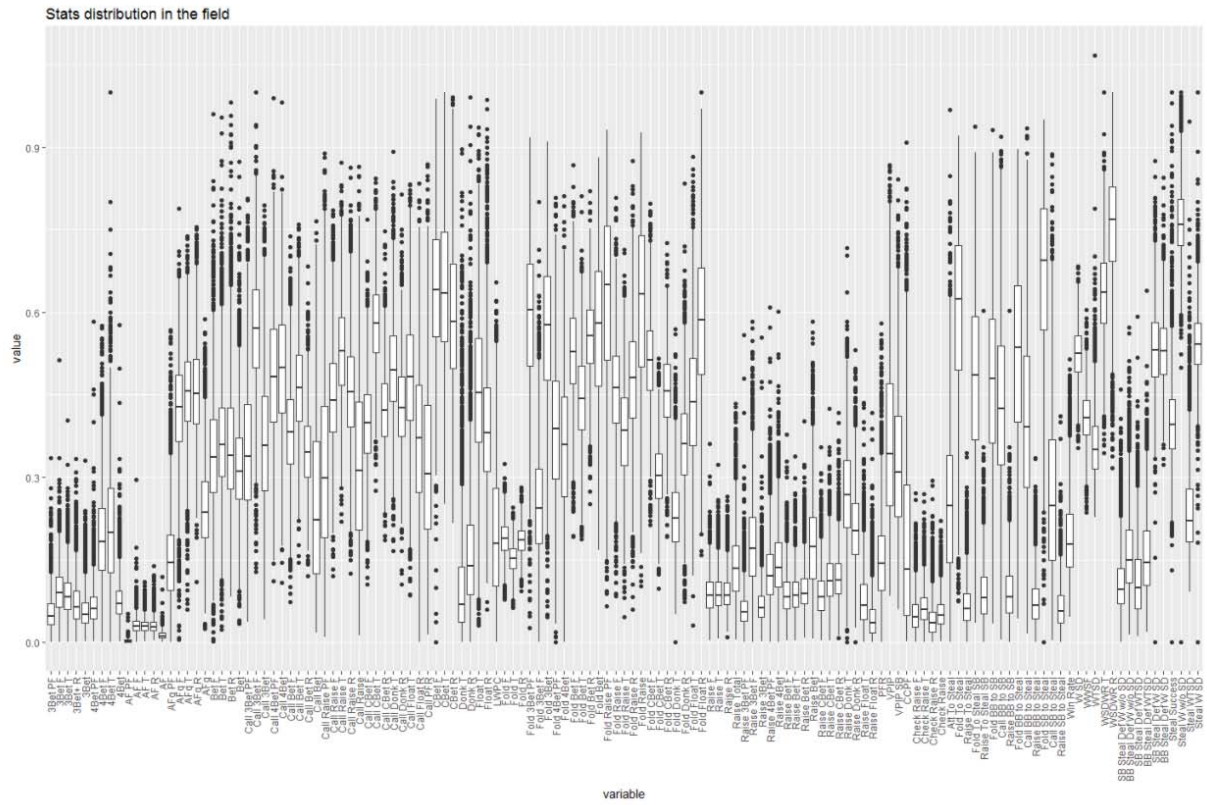


Рисунок 4.1 – Простір статистик гравців

Візуальний аналіз розподілу статистик говорить про їх різноманітність і досить значний розмах, що в свою чергу демонструє їх потенційну силу детермінації. Для більшої начності варто відсортувати статистики за зростанням медіани їхнього розподілу.

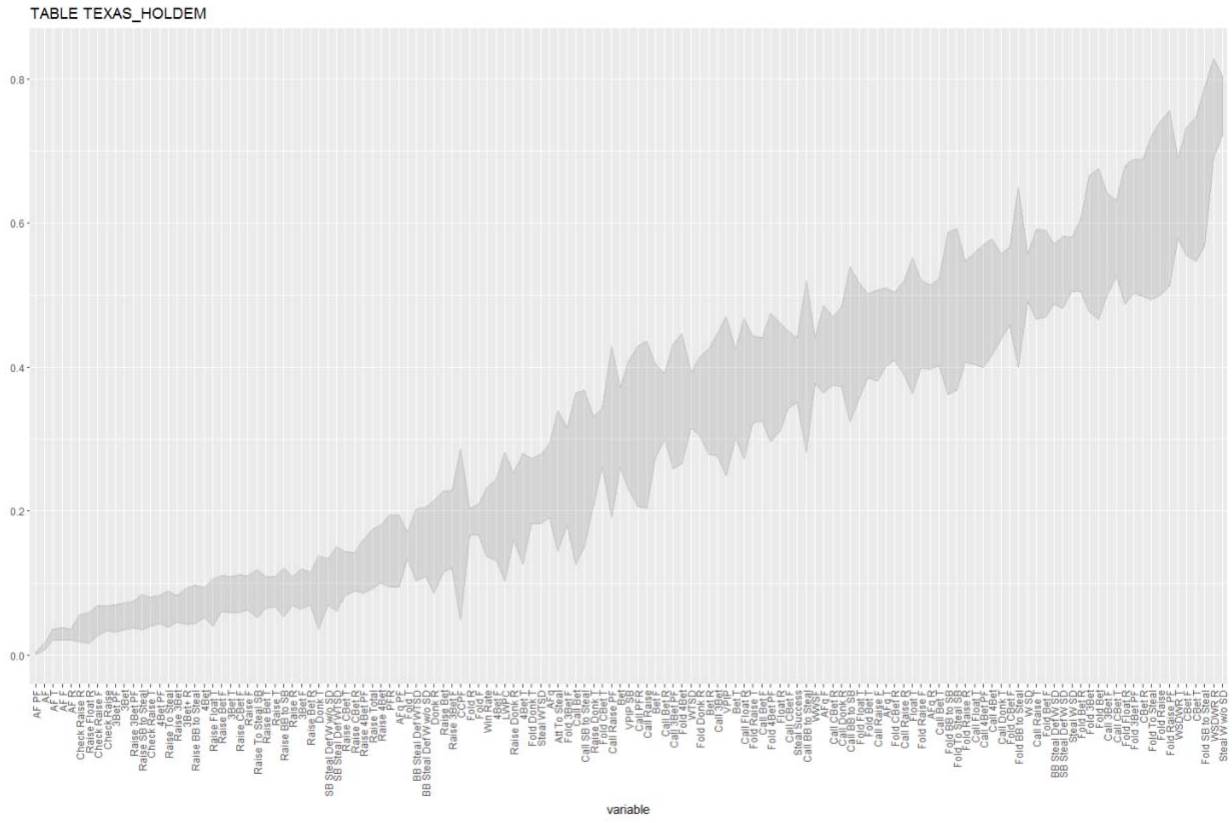


Рисунок 4.2 – Довірчі інтервали статистик у вигляді інтерквартилей

Таке зображення розподілу ще більш явно демонструє різноманітність як середніх значень, так і варіації статистик, що в свою чергу дозволяє сформувати досить детальний портрет кожного гравця.

4.2 Ієрархічна кластеризація гравців гри «Покер»

Спочатку побудуємо ієрархію на основі відстані між точками в просторі статистик. Агрегуючи його, додаванням точок в найближчі кластери і відрізаючи дендрограму на певній висоті можна отримати будь-яку кількість кластерів. Найоптимальнішим з точки зору зміщення і варіації, а також кількості кластерів, що нас цікавлять є відрізання на висоті 3. У такому випадку

отримуємо 3 групи гравців зі схожою поведінкою в межах кожного кластеру (рисунок 4.3).

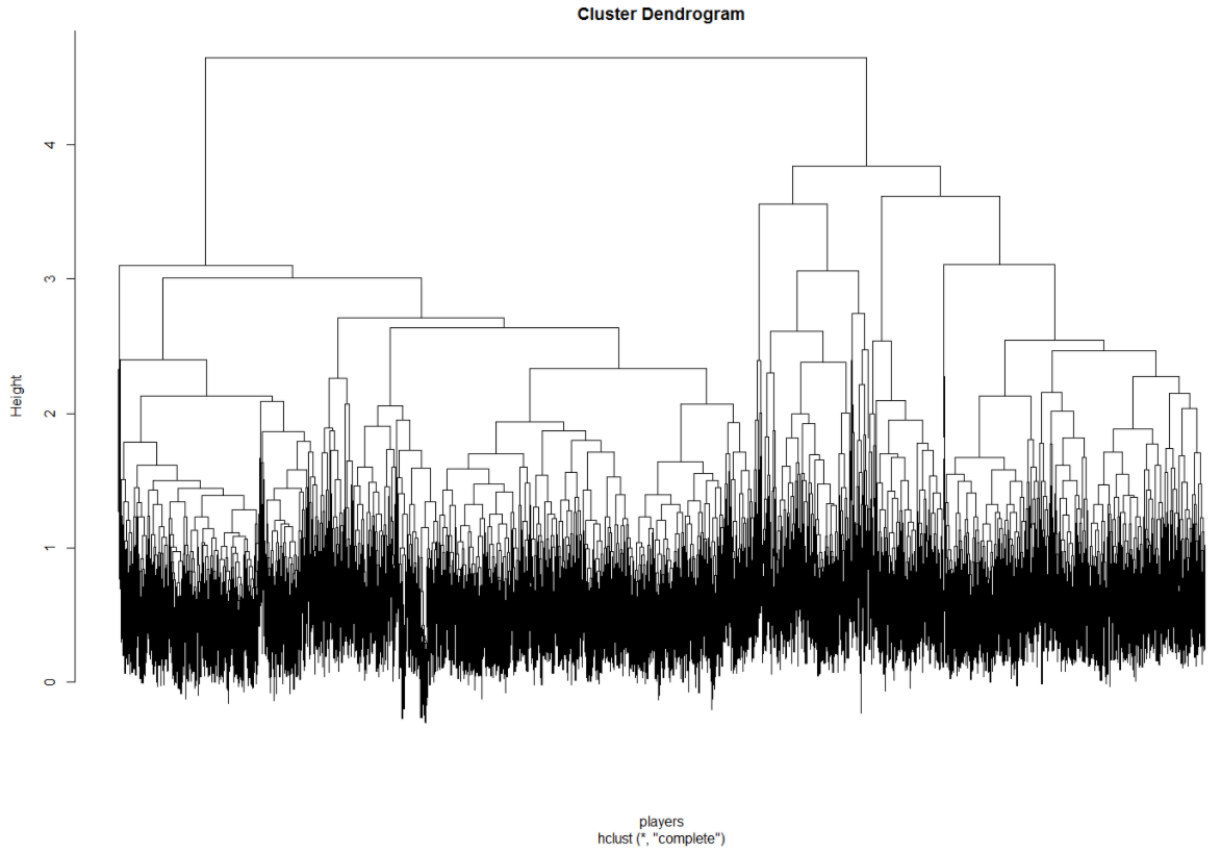


Рисунок 4.3 – Дендрограма гравців

Перша група гравців (рисунок 4.4) характеризується агресивною грою та необережністю. Це призводить до того, що гравці в такому кластері гравці приносять більший дохід, середній дохід в день з такого гравця становить 43,6 дол. США.

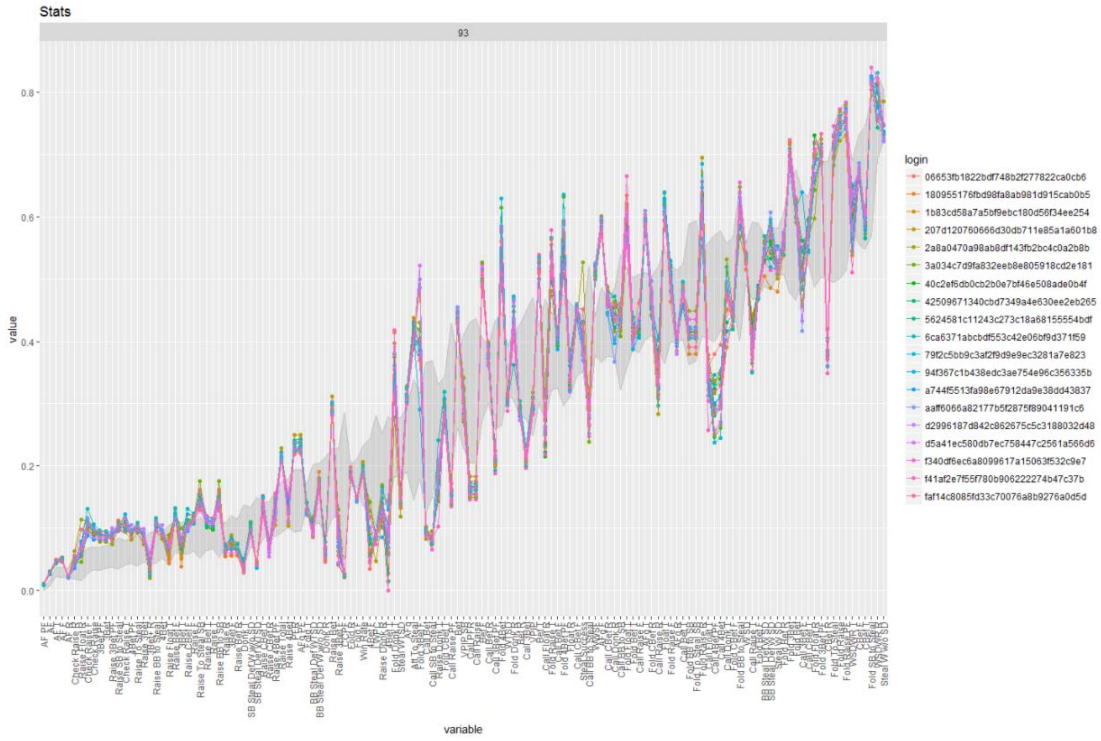


Рисунок 4.4 – Представники 1-ї групи

Друга група (рисунок 4.5) відзначається більш спокійним темпом гри. Представники цієї групи грають обережно і не ризикують. Проявляють витримку та не намагаються диктувати свої правила гри. Це також проектується на дохід, що ми отримуємо з них у вигляді рейку. В цьому випадку він є помірним і становить в середньому 3 дол. США в день.

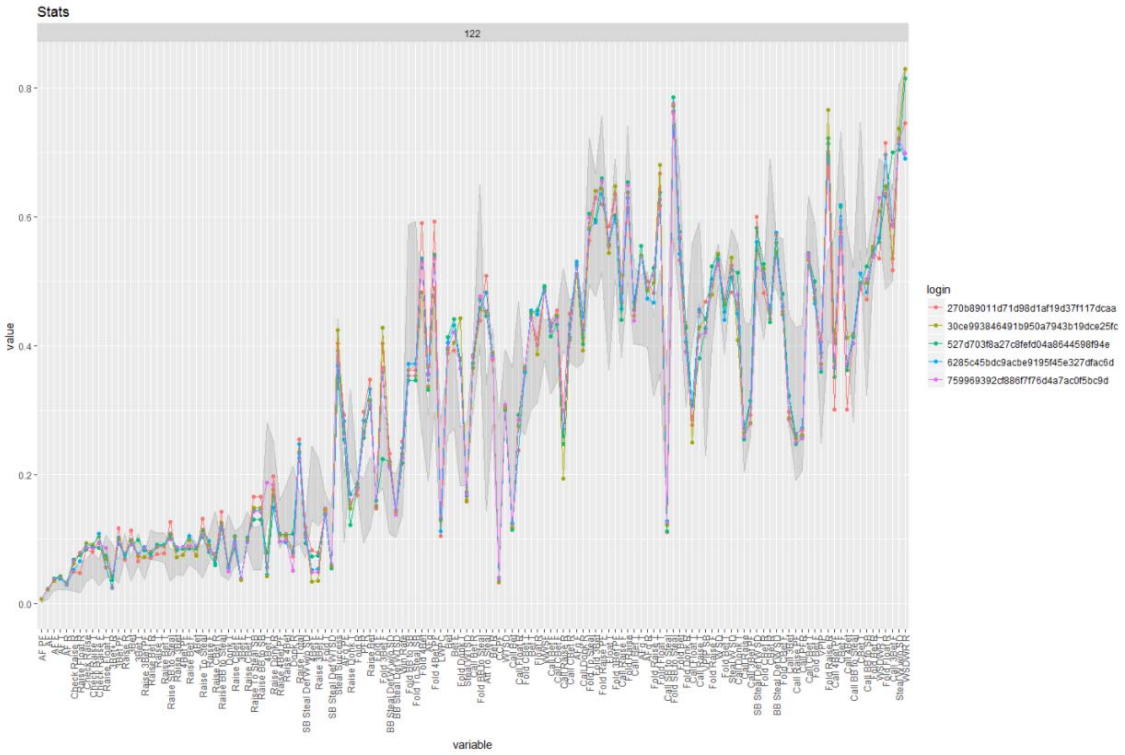


Рисунок 4.5 – Представники 2-ї групи

Третя група (рисунок 4.6) є найбільш обережною і через це найменш прибутковою. Середнє значення доходу, що становить лише 0,02 дол. США в день, говорить про те, що такі гравці грають в основному фріроли або інші безкоштовні покерні столи і турніри.

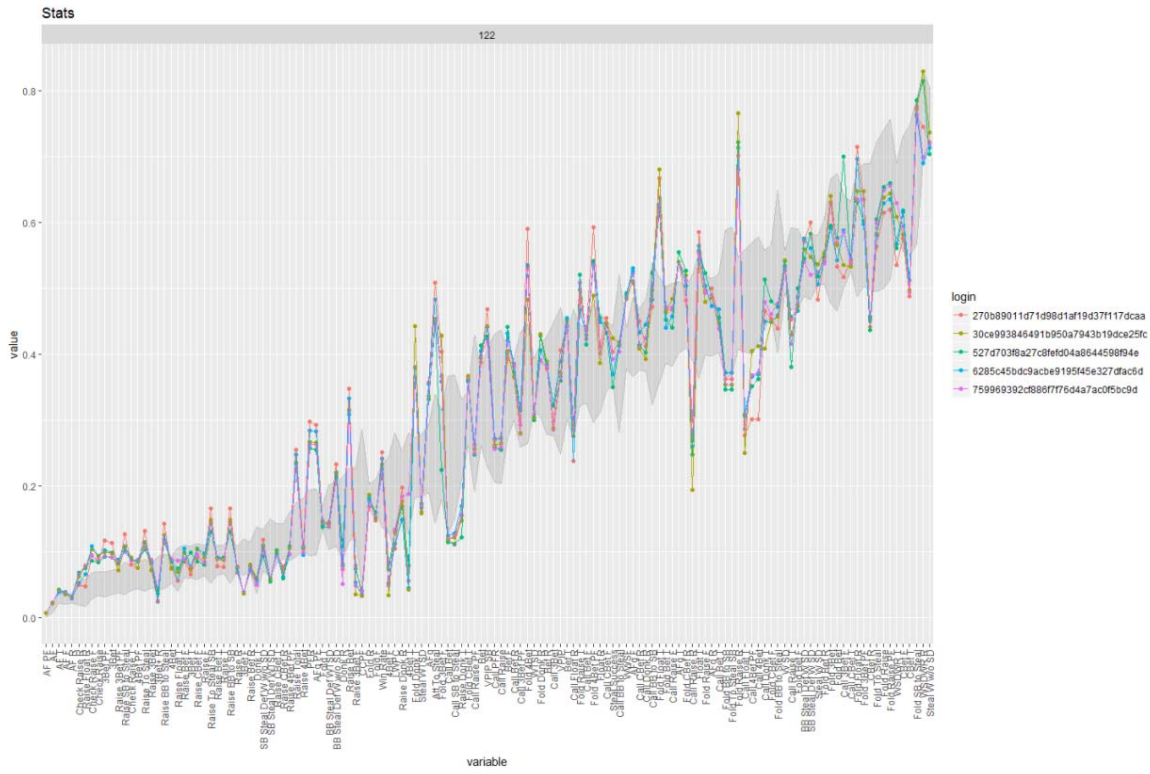


Рисунок 4.6 – Представники 3-ї групи

Отже, для сформованих груп характерні свої особливості гри. Причому ці особливості яскраво проявляються в значеннях доходу, який вони приносять компанії. Саме ця особливість і являється першочерговою причиною необхідності подібної кластеризації для уточнення портрету гравця і складання більш детальної моделі його поведінки.

4.3 Побудова моделей виживання

Наступним кроком являється побудова моделі виживання Каплан-Мейер, що описана в підрозділі 2.3. Для цього використовується інструментарій бібліотеки survival. Візуалізація одержаних результатів проводилася за допомогою пакету survminer та ggplot2.

Спочатку була побудована непараметрична модель для всієї вибірки (див. риунок 4.7)

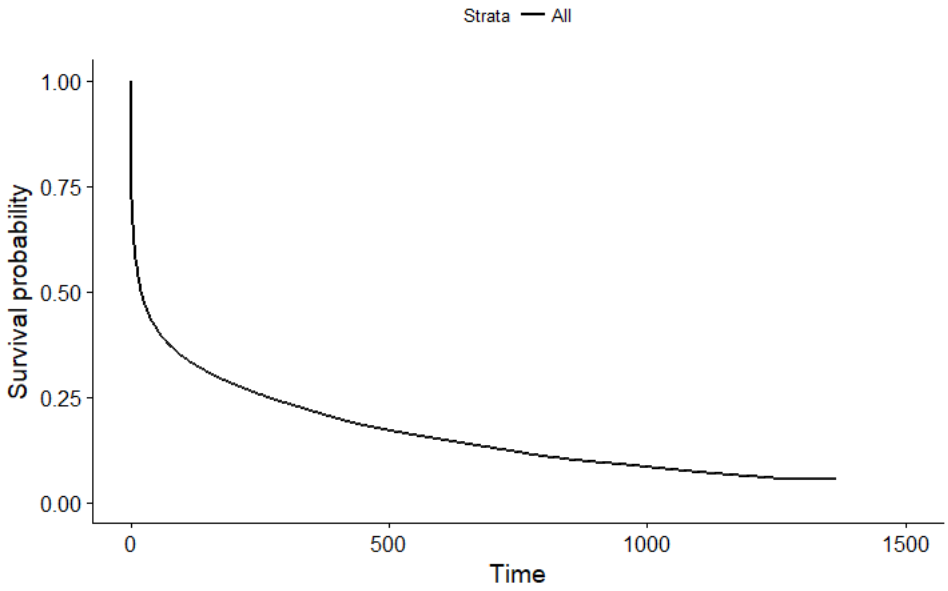


Рисунок 4.7 – Крива виживання для всієї популяції

Медіана життя для всієї вибірки складає 7 днів, що є досить низькою. Завдяки попередній кластеризації клієнтів вдалося виділити клас типічної поведінки, що відповідає більшості гравців в популяції; клас короткострокових гравців та довгострокових гравців (див. рисунок 4.8).

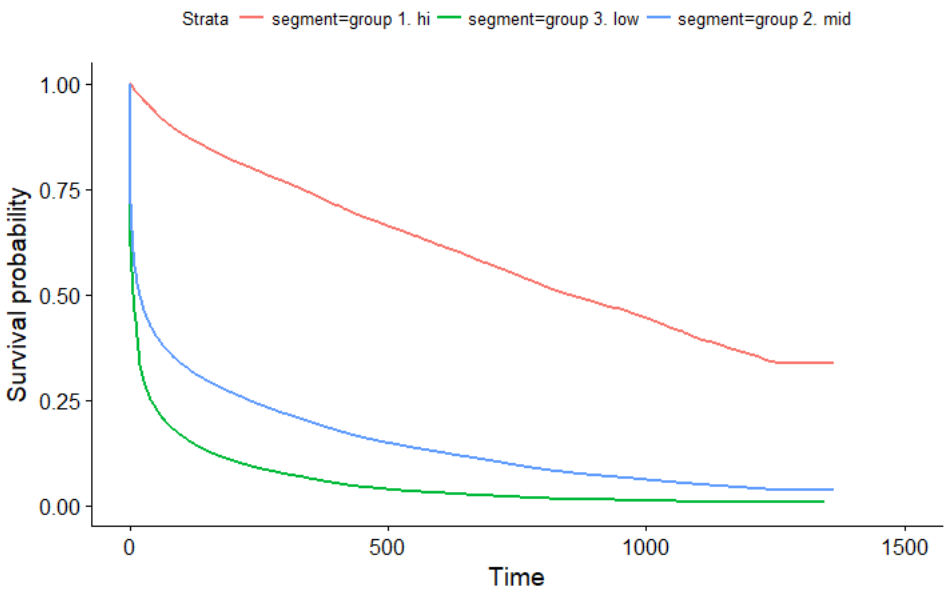


Рисунок 4.8 – Криві відмирання для отриманих кластерів

Також виявилося, що для отриманих кластерів характері різні типи характеру відмирання (рисунок 4.9). Чим більший дохід вони приносять, тим вони менш схильні до відмирання. Так, наприклад, половина популяції 3-ї групи відмирає на 3-й день свого життя. А у випадку 1-ї групи половина відмирає лише після 1000-го дня свого життя.

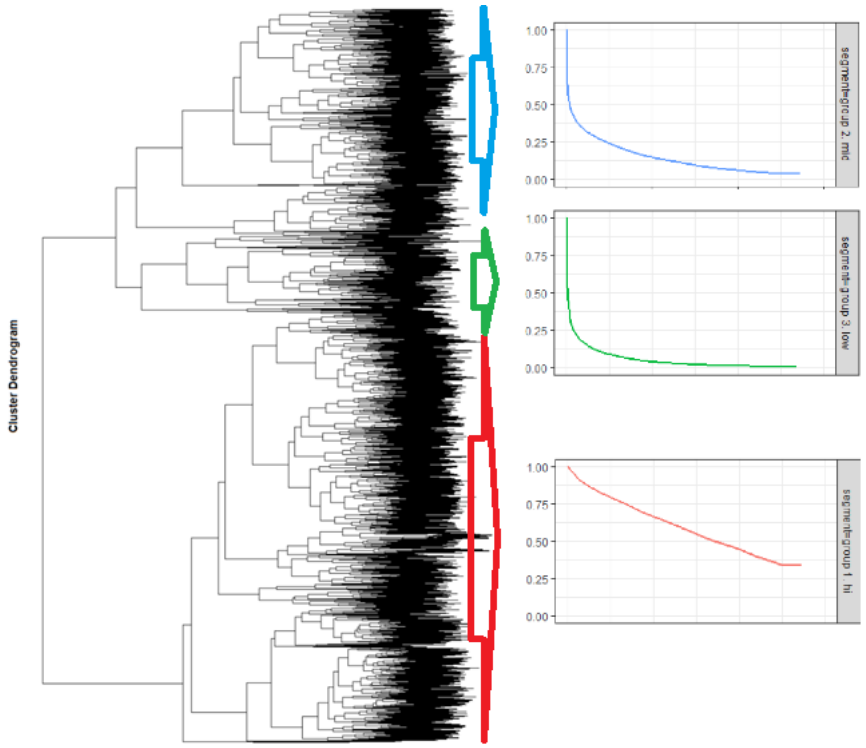


Рисунок 4.9 – Криві відмирання для отриманих кластерів у вигляді піддерев дендрограми

Застосування аналізу виживання дозволяє більш детально оцінювати поведінку кожного гравця. А це в свою чергу веде до того, що можна більш точно оцінювати фінансові показники компанії, такі як прогнозований дохід, про що більш детально буде розписано в наступному розділі.

Висновки

Отже, в цьому розділі була проведена кластеризації відрізанням дерева на висоті 3. У результаті було одержано 3 кластери клієнтів, що відрізняються певним типом поведінки. Тип поведінки відображається в середньому доході, який ми отримуємо із гравців, що дозволяє більш детально оцінювати цей фінансовий показник. Цей результат дозволяє більш детально описати гравців та виділити деталі, важливі для прогнозу, в їхній поведінці.

Усе це разом наштовхує на ідею систематизації процесу оцінки прогнозованої втрати доходу у вигляді певної системи підтримки прийняття рішень. Реалізація цієї ідеї на рівні певного прототипу системи описана в наступному розділі.

РОЗДІЛ 5 ПРОТОТИП СППР ДЛЯ ОЦІНКИ МОЖЛИВИХ ВТРАТ ДОХОДІВ

В цьому розділі проектується та реалізовується система підтримки прийняття рішень, що основується на величині прогнозних втрат доходу. Для цього використовується наступна модель оцінки прогнозованих втрат, що має вигляд:

$$E = \sum_{i=1}^{n} PD_i \cdot \left(T_i^{(m)} - T_i\right) \cdot R_{avg_i} \tag{5.1}$$

де PD_i – ймовірність відпаду, розрахована за моделлю із розділу 3;

$T_i^{(m)}$ – медіанний час життя клінта, розрахований за моделлю із розділу 4;

T_i – це час життя на момент розрахунку;

R_{avg} – середньоденний дохід від клієнта.

5.1 Архітектура СППР

В якості основного буферу між модулями аналізу і візуалізації результатів виступає база данних PostgreSQL. Для аналізу використовуються два підмодулі: модуль вітрин, написаний на SQL та модуль оцінки параметрів, що реалізований на мові програмування R. Обидва модулі зберігаються на сервері і разом формують бек-енд СППР (рисунок 5.1).

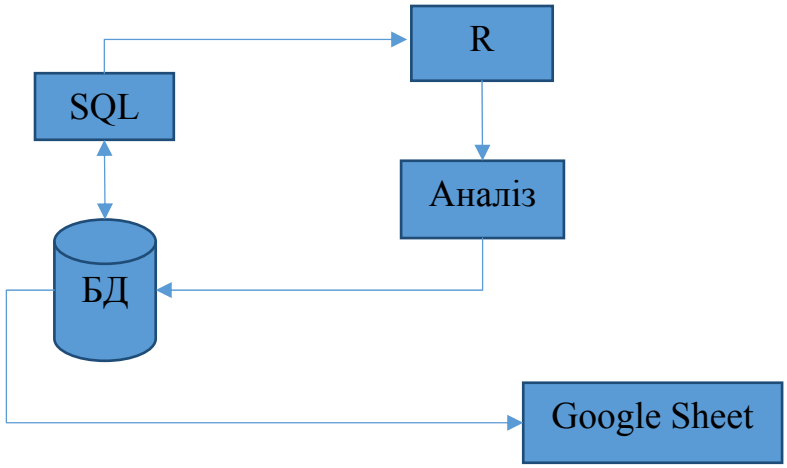


Рисунок 5.1 – Концептуальна схема архітектури СППР

5.1.1 Модуль вітрин SQL

Для формування зрізу активності клієнта за день та актуалізації цих даних із частотою 1 раз на добу використовується функціонал вітрин стандартного SQL (див. рисунок 5.2).

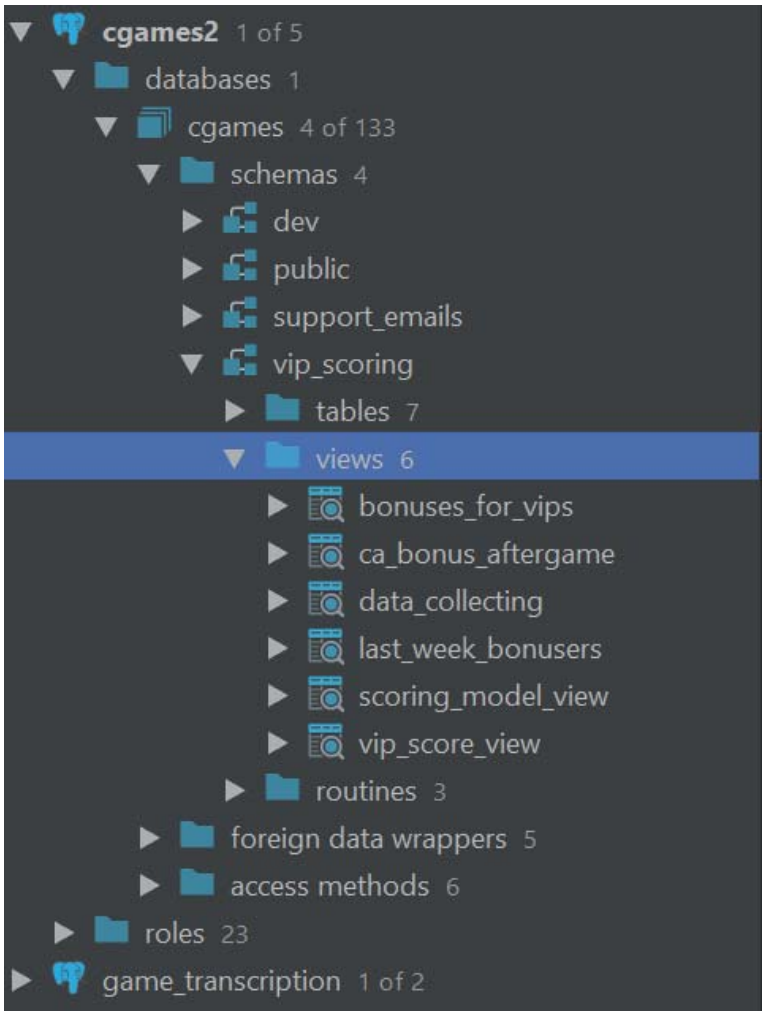


Рисунок 5.2 – Структура вітрин на сервері SQL

Основною вітриною для аналізу моделей являється scoring_model_view (див. додаток Б), що оновлює вхідні дані, необхідні для моделі, кожного дня.

Також для роботи системи необхідна вітрина vip_score_view (див. рисунок 5.3). Вона використовується для візуалізації оцінки ймовірності відтоку клієнта.

	user_id	type	last_date	score	is_bad	rake	ng	dep_media
1	4df7b89051d144990579b	(casino,poker)	(2018-03-22,2018-03-22)	(0.05138151301129300,0.81927405365424000)	(true,true)	75.54932443600992	611.9283223321622	21.25450139976
2	74b3489051d1449913aa7	(casino,poker)	(2017-03-13,2018-03-23)	(0.9999998351055600,0.11729414321023300)	(true,false)	1742.6350460125905	-0.7538907119558649	16.31409139494
3	87e0b89051d1449515ac6	(casino,poker)	(2018-03-15,2018-03-22)	(0.8210866531111800,0.7782837237511080)	(true,true)	3310.7453112442358	-95.50048127184523	76.2628756798
4	b2def89051d144991e28e	(poker)	(2018-01-28)	(1.0000000000000000)	(true)	61.78845567200418	<null>	21.7883810402
5	c7e5c89051d144992 4f3f	(casino,poker)	(2018-03-22,2018-03-22)	(0.13516164919569700,0.74275359462105400)	(false,true)	298.799256318495596	440.486359013910 67	17.19045998232
6	e1cef89051d14492b135	(casino,poker)	(2018-02-05,2018-03-22)	(0.9831416473518860 0,0.3570516591 9301900)	(true,false)	244.44652911497944	0.05472203585007 1106	43.3651344319
7	e44b189051d144992be4b	(casino,poker)	(2017-12-24,2018-03-17)	(0.9500000000000000,0.9989800892172930 0)	(true,true)	120.756112593675 27	4.44	
8	0777d89051d1449938e02	(casino,poker)	(2018-03-05,2018-03-22)	(0.9994075899873140 0,0.7907064143752370 0)	(true,true)	106.53782181965543	-61.34895998767093	15.14114576078
9	1e44189051d144993fc0f	(casino,poker)	(2018-03-21,2018-03-17)	(0.8560692522786450 0,0.9989123320584110 0)	(true,true)	1.6061365101542613	4082.1483197172806	21.19760406509
10	21ec089051d144994126b	(poker)	(2018-03-23)	(0.0498436146629134 0)	(false)	1937.9003916621816	<null>	16.94026683595
11	3564f89051d1449946e7f	(casino,poker)	(2018-03-09,2018-03-22)	(0.9999584277054140 0,0.7486520250331050 0)	(true,true)	67.19115669032887	-356.31811667027483	10.63214480130
12	4156989051d1449948e6c	(poker)	(2018-03-19)	(0.6113326850551950 0)	(true)	191.31820097096002	<null>	17.77894723741
13	4c14789051d144994cd29	(poker)	(2018-03-17)	(0.9988011330315050 0)	(true)	0	<null>	77.08043960516
14	a2543895051d1449965732	(casino,poker)	(2018-03-11,2018-03-20)	(0.9999996715027080 0,0.9500000000000000)	(true,true)	367.11000335390554	152.90800920546567	14.23102648394
15	c923989051d1449971385	(casino,poker)	(2017-11-10,2018-03-22)	(1.0000000000000000,0.9500000000000000)	(true,true)	271.3757076245187	-317.35522690137026	12.42614211778

Рисунок 5.3 – Перші 15 рядків уже сформованої вітрини vip_score_view

До основних полів цієї вітрини належать:

- *user_id* – ідентифікатор коритувача;
- *type* – вектор типу гравця: казино, покер або обидва;
- *score* – вектор розрахованої ймовірності відтоку;
- *is_bad* – ознака того, чи клієнт перестане грати в покер або казино.

5.1.2 Модуль оцінки параметрів

Цей модуль являє собою два скрипти fit_models.R та eval_score.R (див. додаток Б). В першому зібрано всі теоретичні засади та результати експериментів описаних в попередніх розділах у вигляді наступним, не враховуючи стандартних, бібліотек:

- *RODBC* – бібліотека для підключення до БД;
- *ggplot2* – основна бібліотека, що використовується для візуалізації;
- *survival, survminer* – бібліотеки для побудови моделей виживання та їх візуалізації;
- *dplyr, reshape2, zoo* – бібліотеки для попередньої обробки даних та маніпуляцій із ними;
- *xgboost, ROCR* – бібліотеки для побудови ансамблів дерев та оцінки їх якості;

Модуль забезпечує швидке перетворення даних в потрібний формат, оцінювання моделей та прогнозування необхідних параметрів для оцінки ризику втрати доходу на основі побудованих моделей.

5.1.3 Модуль візуалізації одержаних результатів

Фронт-ендом виступають таблиці Google Sheet, що реалізовують веб-інтерфейс для результатів аналізу. Вони дозволяють ефективно представляти результати у вигляді таблиць, обмежувати доступ певним користувачам та оновлювати дані в режими реального часу (див. рисунок 5.4).

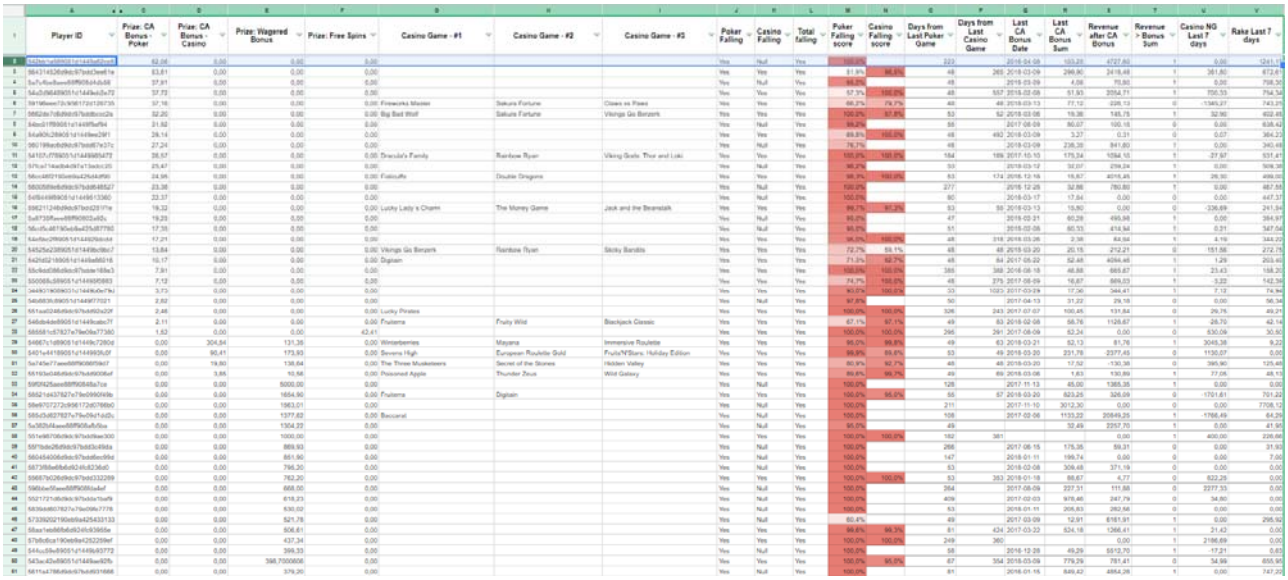


Рисунок 5.4 – Приклад візуалізації результатів за один робочий день

Вигрузка даних із бази в інтерфейс користувача відбувається за допомогою скриптів на мові програмування Python із такою ж періодичністю, як і оновлення даних за кожним із клієнтів – тобто 1 раз на добу.

5.2 Деталі реалізації

Для актуалізації даних по гравцям потрібно оновлювати дані кожного дня. Для цього використовується процес-демон на сервері cron, що кожного дня

запускає скриптів оцінки параметрів. А потім, після успішного їх виконання запускає ще один скрипт, що оновлює таблиці Google Sheet. Весь лісинг необхідних скриптів розміщено в Додатку А. Весь розпорядок процесу, що відповідає за періодичний запуск необхідхин модулів зображено на рисунку 5.5.

	jobid ▲ 1	schedule	command	nodename	nodeport	database	username
1	1	0 7 * * *	refresh materialized view public.nick_pin_view; commit;#select clans.dbl	localhost	5432	cgames	rdfraud
2	2	0 0 * * *	select public.upd_currency_rates(); commit;#select clans.dblink_exec('ho	localhost	5432	cgames	rdfraud
3	4	0 0 * * *	select public.upd_getsubidserver()	localhost	5432	cgames	rdfraud
4	5	0 0 * * *	select public.upd_aff_data_server_new()	localhost	5432	cgames	rdfraud
5	6	5 0 * * *	refresh materialized view public.pd_rolsegm_view; refresh materialized v.	localhost	5432	cgames	rdfraud
6	7	38 9 * * *	refresh materialized view public.aqua_curr_deps	localhost	5432	cgames	rdfraud
7	8	38 9 * * *	refresh materialized view public.pomodorro_spins	localhost	5432	cgames	rdfraud
8	9	38 9 * * *	refresh materialized view public.pomadorro_casino_games	localhost	5432	cgames	rdfraud
9	10	38 9 * * *	refresh materialized view public.poker_games	localhost	5432	cgames	rdfraud
10	11	38 9 * * *	refresh materialized view public.pomadorro_bonuses	localhost	5432	cgames	rdfraud
11	17	0 0 3 * *	set search_path = clans, antifraud; select antifraud.fom_parse_pokerhand:	localhost	5432	cgames	rdfraud
12	18	0 0 5 * *	set search_path = clans, antifraud; select antifraud.fom_gt_upd_blinds_a:	localhost	5432	cgames	rdfraud
13	19	0 0 5 * *	set search_path = clans, antifraud; select antifraud.fom_gt_upd_fin_resu.	localhost	5432	cgames	rdfraud
14	20	0 0 5 * *	set search_path = clans, antifraud; select antifraud.fom_gt_upd_game_rou:	localhost	5432	cgames	rdfraud
15	21	0 0 5 * *	set search_path = clans, antifraud; select antifraud.fom_gt_upd_game_sum	localhost	5432	cgames	rdfraud
16	22	0 0 5 * *	set search_path = clans, antifraud; select antifraud.fom_gt_upd_hand_sta	localhost	5432	cgames	rdfraud
17	23	0 0 5 * *	set search_path = clans, antifraud; select antifraud.fom_gt_upd_uncalled_	localhost	5432	cgames	rdfraud
18	24	0 19 2 2 Fri	set search_path = clans, antifraud; select antifraud.fom_get_players_net:	localhost	5432	cgames	rdfraud
19	25	* * * * *	select clans.clans_collecting_data_function()	localhost	5432	cgames	rdfraud
20	28	30 5 * * Mon	SELECT * from reporting.tableau_ph_report()	localhost	5432	cgames	rdfraud
21	29	0 5 * * *	REFRESH MATERIALIZED VIEW reporting.tableau_ph_report_funnel_view; REFRE:	localhost	5432	cgames	rdfraud
22	30	0 1-23/3 * * *	select public.upd_tbl_multi_paym_gac_cell()	localhost	5432	cgames	rdfraud
23	31	1 21 * * *	select fom.upd_rdfraud_currency()	localhost	5432	cgames	rdfraud
24	32	20 1-23/3 * * *	select fom.upd_rdfraud_multiacc()	localhost	5432	cgames	rdfraud

Рисунок 5.5 – Структура планувальника cron, необхідного для періодичного запуску модулів

Висновки

В цьому розділі описано концепуальну схему побудови прототипу СППР. Обрано найкращий варіант архітектури, якою виявилася централізвана модель «зірка» із буфером – базою даних на PostgreSQL.

Принциповою відміністю даної реалізації є те, що вона являється повністю безкоштовною завдяки застосування open source технологій, таких як: Ubuntu, PostgresSQL, R. Також характерною особливістю є простий інтерфейс, що основується на продукті від Google під назвою Google Sheet.

Однак за простотою цієї системи ховається гнучкість у її доповненні. Модульна архітектура цієї системи дозволяє поетапно покращувати її шляхом

допрацювання кожного модуля окремо. Можливість такого оновлення робить її досить цікавою для подальшого розвитку та розробки.

РОЗДІЛ 6 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

1.1 Опис ідеї проекту

У даному розділі наведемо таблиці 6.1-6.22, які описують розробку стартап-проекту.

Таблиця 6.1 – Опис ідеї стартап-проекту

Зміст ідеї	*Напрямки застосування*	*Вигоди для користувача*
Ідея стартапу – СППР оцінки можливої втрати доходу	1. Відносини з клієнтами	Суттєве покаращення відносин з клієнами
	2. Можливість агрегації показників	Зручне агрегування статистики для відображення у звітах та при прийнятті рішень
	3. Можливість швидкого реагування на зміну в поведінці гравців	Використання прогнозу доходу та заповнюваності для подальшого планування витрат та стратегічного планування подальшої роботи по певним напрямкам

У наступній таблиці виберемо бальну шкалу для оцінювання від 1 до 20.

Таблиця 6.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	*Техніко-економічні характеристики ідеї*	*(потенційні) товари/концепції конкурентів*			*W (слабка сторона)*	*N (нейтральна сторона)*	*S (сильна сторона)*
		Мій проект	*Конкурент1*	*Конкурент2*			
1.	Функціональність	15	10	8	–	–	Спрощення збору статистики
2.	Складність виконання проекту	15	16	13	Складність виконання наукоємних завдань	–	–
3.	Наявність фінансових ресурсів	9	12	5	Недостатні фінансові ресурси	–	–
4.	Ціна продукту	12	9	10	–	Прийнятна ціна продукту	–
5.	Якість продукту	13	10	8	–	–	Висока якість продукту
6.	Простота збору статистики	15	10	8	–	–	Вихід на нові ринки збуту

1.2 Технологічний аудит ідеї проекту

Таблиця 6.3 – Технологічна здійсненність ідеї проекту

№ п/п	*Ідея проекту*	*Технології її реалізації*	*Наявність технологій*	*Доступність технологій*
1	Ідея стартапу – СППР оцінки можливої втрати доходу	Розробка агрегатора статистики із використанням мови програмування R та бази даних MySQL	Дана технологія наявна на ринку	Дана технологія частково доступна автору проекту
2		Розробка агрегатора статистики із використанням мови програмування R та бази даних PostgreSQL	Дана технологія наявна на ринку	Дана технологія доступна автору проекту
3		Розробка агрегатора статистики із використанням мови програмування Python та бази даних PostgreSQL	Дана технологія наявна на ринку	Дана технологія частково доступна автору проекту
Обрана технологія реалізації ідеї проекту: Отже, із наведеної вище таблиці, можна зробити висновок, що потрібно використовувати для розробки продукту мову програмування R та базу даних PostgreSQL.				

1.3 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 6.4 – Попередня характеристика потенційного ринку стартап проекту

№	*Показники стану ринку (найменування)*	*Характеристика*
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн / ум. од	---
3	Динаміка ринку (якісна оцінка)	Зростає

4	Наявність обмежень для входу (вказати характер обмежень)	Обмежень немає
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	25%

Таблиця 6.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	*Потреба, що формує ринок*	*Цільова аудиторія (цільові сегменти ринку)*	*Відмінності у поведінці різних потенційних цільових груп клієнтів*	*Вимоги споживачів до товару*
1	Ринок потребує агрегатор статистики, який буду спрощувати її збір.	Компанії, що займаються цифровою рекламою	Кожна компанія має свій набір специфічних вимог, які наперед передбачити неможливо. Тому продукт має бути універсальним	Швидкодія роботи програми
2				Функціональність
3				Наявність великої кількості рекламодавців

Таблиця 6.6 – Фактори загроз

№ п/п	*Фактор*	*Зміст загрози*	*Можлива реакція компанії*
1	Нерентабельність	Затратність розробки	Зниження собівартості
2	Підвищення конкуренції	Наявність аналогів	Збільшення функціоналу

Таблиця 6.7 – Фактори можливостей

№ п/п	*Фактор*	*Зміст можливості*	*Можлива реакція компанії*

1	Новизна	Новий продукт на ринку	Постійний аналіз ринку
2	Прибутковість	Підвищення прибутковості	Подальший розвиток

Таблиця 6.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	*В чому проявляється дана характеристика*	*Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)*
1. Вказати тип конкуренції - монополія/олігополія/ монополістична/чиста	Олігополія	Покращення функціоналу
2. За рівнем конкурентної боротьби - локальний/національний/…	Локальний	Глобалізація
3. За галузевою ознакою - міжгалузева/ внутрішньогалузева	Внутрішньогалузева	Покращення позицій в галузі
4. Конкуренція за видами товарів: - товарно-родова - товарно-видова - між бажаннями	Між бажаннями	Генерація нових ідей
5. За характером конкурентних переваг - цінова / нецінова	Нецінова	Покращення якості
6. За інтенсивністю - марочна/не марочна	Не марочна	Підвищення відомості бренду

Таблиця 6.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	*Прямі конкуренти в галузі*	*Потенційні конкуренти*	*Постачальники*	*Клієнти*	*Товари-замінники*

	Admixer	*Початковий капітал*	*Відсутні*	*Внутрішній продукт*	*Агрегатори-аналоги*
Висновки:	Інтенсивність конкуренції невисока	Конкурентів небагато. Строки виходу на ринок – 12 місяців	---	Клієнти не диктують умови	Вихід на глобальний ринок потрібно досліджувати додатково

Таблиця 6.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	*Фактор конкурентоспроможності*	*Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)*
1	Функціональність	Достатній функціонал для збору статистики

Таблиця 6.11 – Порівняльний аналіз сильних та слабких сторін «агрегатора статистики»

№ п/п	*Фактор конкурентоспроможності*	*Бали 1-20*	*Рейтинг товарів-конкурентів у порівнянні з Mediawayss*						
			–3	*–2*	*–1*	*0*	*+1*	*+2*	*+3*
1	Функціональність	15			+				
2	Складність виконання проекту	15			+				
3	Наявність фінансових ресурсів	9					+		
4	Ціна продукту	12				+			
5	Якість продукту	13			+				
6	Простота збору статистики	15			+				

Таблиця 6.11 – SWOT-аналіз стартап-проекту

Сильні сторони: - Спрощення збору статистики - Прийнятна ціна продукту - Висока якісь продукту	Слабкі сторони: - Недостатні фінансові ресурси - Складність виконання наукоємних завдань
Можливості: - Підвищення прибутковості	Загрози: - Підвищення конкуренції

- Вихід на нові ринки збуту	- Зміна потреб клієнтів - Брак коштів у клієнтів

Таблиця 6.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	*Альтернатива (орієнтовний комплекс заходів) ринкової поведінки*	*Ймовірність отримання ресурсів*	*Строки реалізації*
1	Внутрішній продукт	100%	12 місяців

1.4 Розроблення ринкової стратегії проекту

Таблиця 6.124 – Вибір цільових груп потенційних споживачів

№ п/п	*Опис профілю цільової групи потенційних клієнтів*	*Готовність споживачів сприйняти продукт*	*Орієнтовний попит в межах цільової групи (сегменту)*	*Інтенсивність конкуренції в сегменті*	*Простота входу у сегмент*
1	Компанії, що займаються цифровою рекламою	Споживачі готові сприйняти продукт	Попит знаходиться на середньому рівні	Конкуренція незначна	Увійти в сегмент досить просто
Які цільові групи обрано: компанії, що займаються цифровою рекламою					

Таблиця 6.15 – Визначення базової стратегії розвитку

№ п/п	*Обрана альтернатива розвитку проекту*	*Стратегія охоплення ринку*	*Ключові конкурентоспроможні позиції відповідно до обраної альтернативи*	*Базова стратегія розвитку*
1	Розробка прототипу СППР як внутрішнього продукту з подальшою можливістю виводу на ринок	Пропонування продукту компаніям-партнерам	Невеликі початкові затрати Можливість реального тестування до виводу на ринок	Стратегія диференціації

Таблиця 6.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	*Чи є проект «першопрохідцем» на ринку?*	*Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?*	*Чи буде компанія копіювати основні характеристики товару конкурента, і які?*	*Стратегія конкурентної поведінки*
1	Ні, проект є вдосконаленням вже існуючих підходів до збору статистики	Компанія буде шукати нових споживачів	Компанія не буде копіювати основні характеристики товару конкурента	Стратегія виклику лідера

Таблиця 6.17 – Визначення стратегії позиціонування

№ п/п	*Вимоги до товару цільової аудиторії*	*Базова стратегія розвитку*	*Ключові конкурентоспроможні позиції власного стартап-проекту*	*Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)*
1	Швидкодія роботи програми	Наступальна стратегія	Висока швидкодія	Висока швидкість Відсутність зависань Наявність можливості повторного виконання запитів
2	Функціональність		Просунута функціональність	Надійність Багато функцій Якість
3	Наявність великої кількості рекламодавців		Значна кількість рекламодавців	Багато рекламодавців Гнучкість Масштабованість

1.5 Розроблення маркетингової програми стартап-проекту

Таблиця 6.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	*Потреба*	*Вигода, яку пропонує товар*	*Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)*
1	Швидкодія	Висока швидкодія	Збільшення швидкості збору статистики
2	Функціональність	Висока функціональність	Надання широкої функціональності
3	Велика кількість рекламодавців	Велика база рекламодавців	Збільшення кількості рекламодавців

Таблиця 6.19 – Опис трьох рівнів моделі товару

Рівні товару	*Сутність та складові*
I. Товар за задумом	- Швидкий збір необхідної статистики - Агрегування статистики для її подальшого аналізу
II. Товар у реальному виконанні	Властивості / характеристики
	1. Функціональність
	2. Швидкодія
	3. Велика база рекламодавців
	Назва організації-розробника: ОС ЕКСПЕРТ
III. Товар із підкріпленням	До продажу: різноманітні знижки на придбання продукту
	Після продажу: гарантія якості, подальша підтримка продукту
За рахунок чого потенційний товар буде захищено від копіювання: За рахунок прив’язки програми до особистого акаунту на сайті компанії-виробника.	

Таблиця 6.20 – Визначення меж встановлення ціни

№ п/п	*Рівень цін на товари-замінники*	*Рівень цін на товари-аналоги*	*Рівень доходів цільової групи споживачів*	*Верхня та нижня межі встановлення ціни на товар/послугу*
1	$ 2000	$ 10000	$ 200000	$ 8000 - $ 12000

Таблиця 6.21 – Формування системи збуту

№ п/п	*Специфіка закупівельної поведінки цільових клієнтів*	*Функції збуту, які має виконувати постачальник товару*	*Глибина каналу збуту*	*Оптимальна система збуту*
1	Одноразова купівля програмного продукту з його подальшою підтримкою	Встановлення контактів із споживачами і підтримання їх; Формування попиту і стимулювання збуту; Організація руху товару.	0 - 1	Безпосередньо від виробника до споживача

Таблиця 6.22 – Концепція маркетингових комунікацій

№ п/п	*Специфіка поведінки цільових клієнтів*	*Канали комунікацій, якими користуються цільові клієнти*	*Ключові позиції, обрані для позиціонування*	*Завдання рекламного повідомлення*	*Концепція рекламного звернення*
1	Найбільший рівень довіри проявляють до партнерів	Взаємодія з компаніями-партнерами	Функціонал Ціна Швидкодія	Розповісти споживачам про наявність даного продукту та його потенційну користь	Контент-маркетинг

Висновки

Отже, в результаті виконання даного розділу магістерської дисертації, можна зробити висновок, що існує реальна можливість ринкової комерціалізації проекту. При цьому, швидше за все буде попит, динаміка ринку є позитивною. Також варто зазначити, що такий вид діяльності на ринку буде рентабельним.

З огляду на потенційні групи клієнтів та бар’єри, які стоять на шляху, можна сказати, що у даного проекту є досить непогані перспективи впровадження. При цьому рівень конкуренції на даний момент є не дуже високим, а конкурентоспроможність проекту є достатньою.

Для ринкової реалізації проекту, на даний момент, краще обрати варіант розробки продукту, при якому використовується мова програмування R та база даних PostgreSQL.

ВИСНОВКИ

Аналіз поведінки гравців онлайн-ігор є важливим аспектом управління ризиками компанії. Крім того, що вчасне передбачення відтоку клієнта може зберегти прибуток та забезпечити сталий розвиток установи. Фактично будь-який адекватний підхід до формування взаємовідносин між компанією та клієнтами основується на оцінюванні ймовірності втрати такого клієнта. Це ще раз говорить про актуальність скоринг моделей, котрі дозволяють порівнювати клієнтів між собою та визначати вразливі місця.

Особливу увагу привертає такий вид онлайн-ігор, як онлайн-гемблінг. У зв’язку з його динамічним характером та значною капіталізацією на ринку, постає проблема вчасного реагування на зміни в поведінці гравця платформ, що надають послуги в сфері онлайн-гемблінгу. Найбільш підходящим інструментом для розв’язання цієї задачі є поведінковий скоринг, а найрозповсюдженішою моделлю такої скоринг карти – логістична регресія. Однак, як показує практика, цей підхід не дає бажаних результатів. По-перше, така модель є статичною, а по-друге, її важко застосовувати для прогнозування. Тому було запропоновано розглянути дещо альтернативну методологію, що основується на прийомах аналізу виживання та ансамблю бустингових дерев рішень.

У даній роботі було описано основні засади теорії аналізу виживання. Введено такі поняття, як бустинговий ансамбль XGBoost, модель ієрархічної кластеризації та KM. Тобто описано математичний апарат побудови моделі. Виявляється, що XGBoost дозволяє набагато краще моделювати нелінійні залежності між регресорами та цільовим полем, що призводить до кращої та більш точної класифікації і оцінки ймовірності відпаду.

Також продемонстровано можливість збільшити точність моделей виживання шляхом попередньої кластеризації клієнтів. Використання такого функціоналу дозволяє досить точно описати поведінку клієнтів, про що свідчать довірчі інтервали розподілу ймовірності відтоку статистичних моделей.

Також важливим аспектом роботи є розробка прототипу СППР, що дозволяє: по-перше, ефективно боротися з проблемою відтоку клієнтів, шляхом відловлювання гравців, що перебувають під ризиком, по-друге, система дозволяє ранжувати клієнтів за очікуваними втратами при його відтоці і таким чином направляти методи для запобігання відтокові в найбільш доцільні місця, і на кінець, оцінювати загальне фінансове положення компанії через оцінку загального об’єму можливих втрат. Це зайвий раз підкреслює потужність оцінювання фінансових показників шляхом моделювання поведінки кожного окремого клієнта.

При порівнянні отриманих результатів виявилось, що можливість класифікації логістичних моделей збільшується із збільшення лаговості коваріант, однак модель ансамблю дерев показує набагато кращі результати.

У даній роботі було проведено вичерпний аналіз бустингових ансамблів дерев, однак не було проведено аналізу інших видів ансамблів, таких як random forest, а також нейронних мереж. Це може стати темою для подальшого дослідження. Також перспективним є розвиток побудованого прототипу СППР, наприклад, розробка більш зручного інтерфейсу користувача, оскільки Google Sheet забезпечує лише базовий функціонал. При оцінці часу відпаду була проведена попередня кластеризація клієнтів-гравців в покер, а для оцінки гравців в казино було застосовану загальну модель. Її властивості можна значно покращити, якщо реалізувати подібну класифікацію. Дослідженням такої можливості та її потенційною реалізацією також можна доповнити дану роботу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Згуровський М. З. Основи системного аналізу / М. З. Згуровський, Н. Д. Панкратова. – Київ: Видавнича група BHV, 2007. – 544 с.

2. McDonald E. The Global Games Market Will Reach $108.9 Billion in 2017 With Mobile Taking 42% [Електронний ресурс] / McDonald E. – 2017. – Режим доступу до ресурсу: https://newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/.

3. Size of the online gambling market from 2009 to 2020 [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://www.statista.com/statistics/270728/market-volume-of-online-gaming-worldwide/.

4. Denial of service attack [Електронний ресурс]. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Denial-of-service_attack#Distributed_DoS.

5. Classification and regression trees / L.Breiman, J. Friedman, R. Olshen, C. Stone. – Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software, 1984. – 300 с..

6. Gareth J. An Introduction to Statistical Learning / J. Gareth, D. Witten, T. Hastie, R. Tibshirani // Springer. – 2015. – Vol. 52, No. 3. – P. 315.

7. Boosting Algorithms as Gradient Descent [Електронний ресурс] / L.Mason, J. Baxter, P. Bartlett, M. Frean. – 1999. – Режим доступу до ресурсу: https://papers.nips.cc/paper/1766-boosting-algorithms-as-gradient-descent.pdf.

8. Cheng T. XGBoost: A Scalable Tree Boosting System [Електронний ресурс] / T. Cheng, C. Carlos. – 2016. – Режим доступу до ресурсу: https://arxiv.org/pdf/1603.02754.pdf.

9. Stepanova M. Survival analysis methods for personal loan data / M. Stepanova, L. C. Thomas // Operations Research. — 2002. — Vol. 50, No. 2. — P. 277–289.

10. Щодо організації та функціонування систем ризик-менеджменту в банках україни: методичні рекомендації, схвалені постановою правління НБУ від 02 серпня 2004 № 361 // [Електронний ресурс]. — Режим доступу : http://zakon.nau.ua/doc/?uid=1045.5945.1&nobreak=1.

11. Cao R. Modelling consumer credit risk via survival analysis / R. Cao, J. M. Vilar, A. Devia // SORT Statistics and Operations Research Transactions. — 2009. — Vol. 33, No. June. — P. 3-30.

12. Cox D. R. Regression models and life-tables / D. R. Cox, S. Society, S. B. // Methodological. — 2007. — Vol. 34, No. 2. — P. 187–220.

13. Dabrowska D. Non-parametric regression with censored survival time data / D. Dabrowska // Scandinavian Journal of Statistics. — 1987. — Vol. 14, No. 3. — P. 181–197.

14. Perianez A. Churn Prediction in Mobile Social Games: Towards a Complete Assessment Using Survival Ensembles [Електронний ресурс] / A. Perianez, A. Saas, A. Guitart. – 2017. – Режим доступу до ресурсу: https://arxiv.org/pdf/1710.02264.pdf.

15. Online poker [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Online_poker.

16. Якось І. Ризик як міра невизначеності [Електронний ресурс] / І.С. Якось. – 2009. – Режим доступу до ресурсу: http://dspace.nbuv.gov.ua/bitstream/handle/123456789/22970/23-Yakos.pdf.

17. Cluster analysis [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Cluster_analysis.

18. Фомін О. В. Прогнозування ризику втрати користувачів онлайн-платформи / О. В. Фомін, Н. В. Кузнєцова. // IASA. – 2018. – №20. – С. 139–140.

19. Фомін О.В. Скорингові моделі поведінки клієнтів-власників кредитних карток для оцінки їх платоспроможності/ Фомін О.В., Кузнєцова Н.В. // Системні науки та кібернетика. — 2016. — №5. — с.56-67.

ДОДАТОК А. ІЛЮСТРАТИВНІ МАТЕРІАЛИ ДОПОВІДІ

Скорингові моделі поведінки гравців для оцінки фінансових показників компанії

Фомін О.В.
КА-61м

Актуальність теми

Класичний підхід скорингу поведінки

- **Логістична регресія** – статистичний регресійний метод, що використовується у випадку, коли пояснювана змінна може набувати тільки двох значень (чи, більш загально, скінченну множину значень).

$$p = \mathbb{E}(y|x) = \frac{e^{\beta^T x}}{1 + e^{\beta^T x}} = \sigma\left(e^{\beta^T x}\right)$$

Постановка завдання

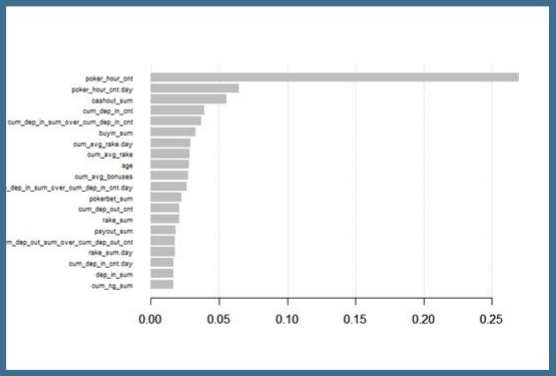
- **Мета роботи:** розробка моделей та методів моделювання поведінки клієнтів та їх порівняння із існуючими загальноприйнятими
- **Завдання:**
  - розглянути існуючі методи побудови скоринг-моделей;
  - виявити найбільш актуальні та перспективні підходи до розробки моделей;
  - розробити власні моделі поведінки клієнтів-гравців онлайн-платформи;
  - порівняти отримані результати та зробити висновки;
  - розробити прототип системи прийняття рішення для розв’язку схожих завдань у подальшому.

Модель оцінки ймовірності відтоку. Теорія

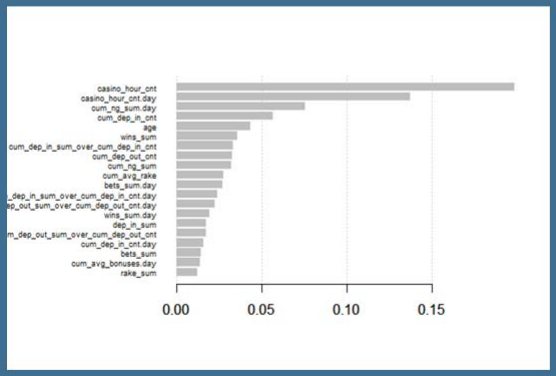
- $y_i^* = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in F$ – *простір дерев прийняття рішень;*
- $L(\phi) = \sum_i l(y_i^*, y_i) + \sum_k \Omega(f_k), \Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2;$
- $L^{(t)} \approx \sum_{i=1}^{n} \left[l\left(y_i, y_i^{*(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t);$
- $L_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i \in I_L} g_i\right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma.$

Модель оцінки ймовірності відтоку

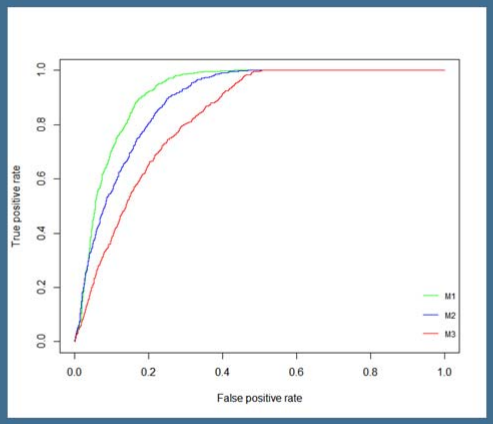
Покер



Казино



Порівняння із логістичною регресією

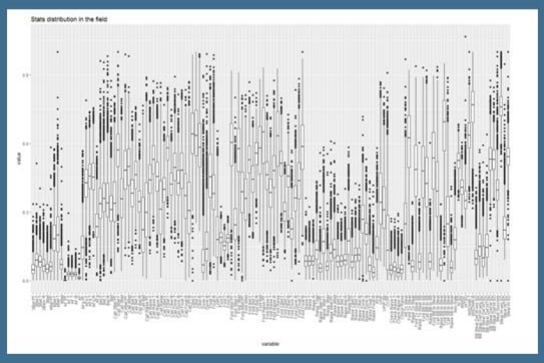


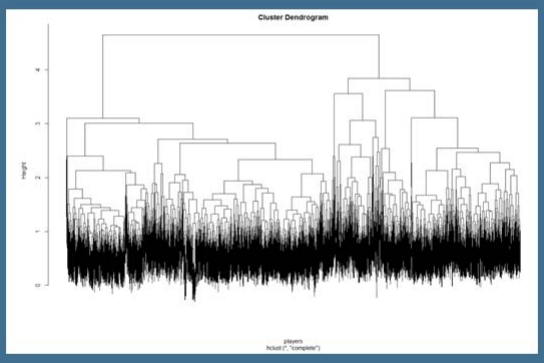
	M1	M2	M3
Покер	86%	71%	75%
Казино	82%	70%	72%

Модель виживання. Теорія

- $h(t, x(t), \beta) = h_0(t)e^{\beta^T x(t)}$;
- $\mathbb{P}(T \geq t) = \frac{h(t,x_i(t),\beta)}{\sum_{j \in R(t_i)} h(t_i,x_j(t),\beta)} = \frac{e^{\beta^T x_i(t)}}{\sum_{j \in R(t)} e^{\beta^T x_j(t)}}$, $R(t) = \{j | t_j \geq t\}$.

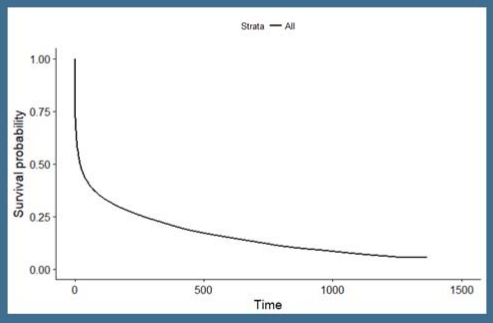
Класифікація покерних гравців



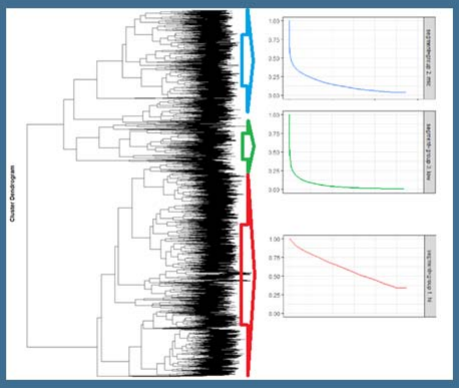


Модель виживання для всієї популяції

- Медіанний час життя – 7 днів:



Модель виживання



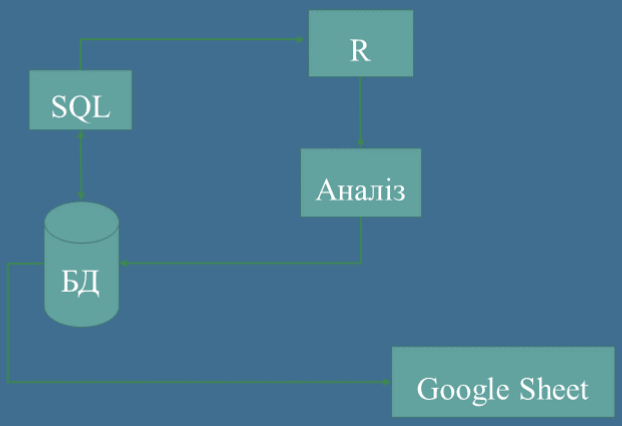
- Медіанний час життя – 3 дні;
- Медіанний час життя – 7 днів;
- Медіанний час життя – 100 днів;

Оцінка можливих втрат доходів

$$E = \sum_{i=1}^{n} PD_i \cdot \left(T_i^{(m)} - T_i\right) \cdot R_{avg_i}$$

- PD_i – ймовірність відпаду;
- $T_i^{(m)}$ – медіанний час життя клінта;
- T_i – це час життя на момент розрахунку;
- R_{avg} – середньоденний дохід від клієнта

Прототип СППР для автоматизації процесу

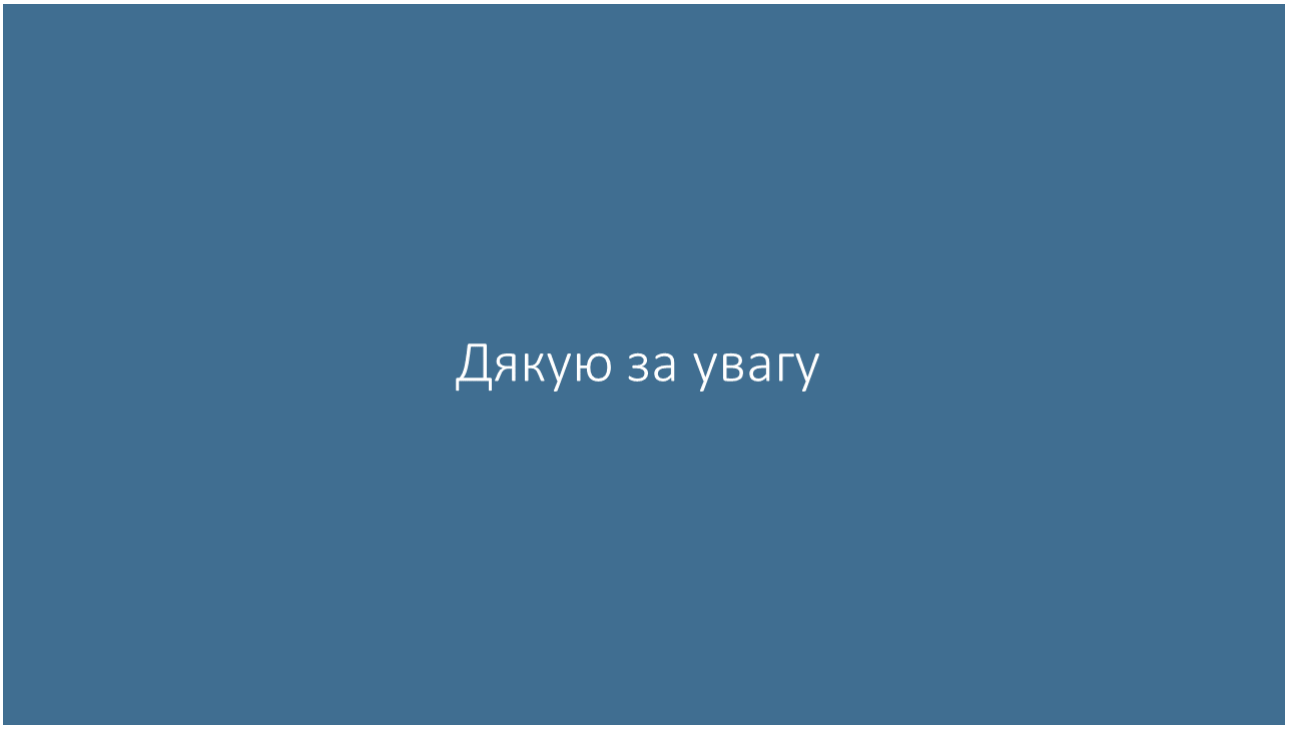


Висновки

- Побудовано модель на основі бустингового ансамблю дерев, що значно краще оцінює ймовірність відвалу порівняно із логістичною регресією;
- Здійснено класифікацію клієнтів;
- Проведно моделювання часу життя клієнта на основі проведеної класифікації, що значно точніше описує гравців;
- Результати проведеного аналізу синтезовано в показник оцінки можливих втрат доходу, як один із основних фінансових показників;
- Імплементовано прототип СППР на основі вище описаних експериментів.

Перспективи

- Проведення аналізу інших ансамблів дерев, таких як random forest та моделей на основі нейронних мереж;
- Дослідження можливості кластеризації гравців казино;
- Розвиток прототипу СППР.


Дякую за увагу

ДОДАТОК Б. ЛІСТИНГ ПРОГРАМИ

fitting_models.R:

```
library(RODBC)
library(dplyr)
library(reshape2)
library(ROCR)
library(xgboost)
library(zoo)

con = odbcConnect('PostgreSQL35W')

df = sqlQuery(con, 'select
                      t.*,
                      coalesce(t.poker_hour_cnt, 0) > 0 is_poker,
                      coalesce(t.casino_hour_cnt, 0) > 0 is_casino,
                      date = max(case when poker_hour_cnt > 0 then date end)
over (partition by user_id) last_poker,
                      date = max(case when casino_hour_cnt > 0 then date end)
over (partition by user_id) last_casino
                    from public.vip_scoring_view t
               ;')

#FUNCTIONS
get_essential = function(absence_measure, lower_bound, target.quantile) {
  #function for target field
  total_diff = sum(absence_measure)
  diff_sum = ifelse(total_diff == 0, 1, cumsum(absence_measure)/total_diff)
  return(ifelse(diff_sum <= target.quantile & absence_measure > lower_bound, 1,
0))
}

prepare_data = function(df0) {
  #function for preparing data

  #Preparing lagging 1-day [1-week, 1-month (30-days)]
  df3 = df0 %>%
    select(user_id, date) %>%
    mutate(date1 = date - 1) %>%
    melt(id.vars = c('user_id'), value.name = 'date') %>%
    select(user_id, date) %>%
    distinct() %>%
    left_join(df, by = c('user_id' = 'user_id', 'date' = 'date'))

  #LOCF-ing lagged 1-day [1-week, 1-month (30-days)]
  df4 = df3 %>%
    select(user_id, date,
           cum_avg_rake,
           cum_avg_bonuses,
           cum_dep_in_sum_over_cum_dep_in_cnt,
           cum_dep_out_sum_over_cum_dep_out_cnt,
           cum_ng_sum,
           date0,
           age) %>%
    arrange(user_id, date) %>%
```

```
    group_by(user_id) %>%
    mutate_all(funs(na.locf(., na.rm = F))) %>%
    mutate(absence_weight = as.numeric(date - date0),
           cum_avg_rake = cum_avg_rake*age/(age + absence_weight),
           cum_avg_bonuses = cum_avg_bonuses*age/(age + absence_weight)) %>%
    select(user_id, date,
           cum_avg_rake,
           cum_avg_bonuses,
           cum_dep_in_sum_over_cum_dep_in_cnt,
           cum_dep_out_sum_over_cum_dep_out_cnt,
           cum_ng_sum) %>%
    ungroup()

  df3 = df3 %>%
    select(-c(cum_avg_rake,
              cum_avg_bonuses,
              cum_dep_in_sum_over_cum_dep_in_cnt,
              cum_dep_out_sum_over_cum_dep_out_cnt,
              cum_ng_sum,
              age)) %>%
    filter(!is.na(date)) %>%
    mutate_all(funs(na.fill(., 0, na.rm = F)))

  df5 = df0 %>%
    mutate(date1 = date - 1) %>%
    inner_join(df3, by = c('user_id' = 'user_id', 'date1' = 'date'), suffix =
c('', '.day')) %>%
    inner_join(df4, by = c('user_id' = 'user_id', 'date1' = 'date'), suffix =
c('', '.day')) %>%
    filter(!is.na(cum_avg_rake.day))

  return(df5)
}

createWatchlist = function(df, labels, train_id) {
  #function for creating watchlist for xgboost
  dtrain = xgb.DMatrix(data.matrix(df[train_id, ]), label = labels[train_id])
  dtest = xgb.DMatrix(data.matrix(df[-train_id, ]), label = labels[-train_id])
  list(eval = dtest, train = dtrain)
}

get_cutoff = function(model, newdata, labels) {
  pred = prediction(predict(model, newdata = newdata), labels)
  perf = performance(pred, 'tpr', 'fpr')
  cutoffs = data.frame(cut = perf@alpha.values[[1]], fpr = perf@x.values[[1]],
tpr = perf@y.values[[1]])
  cutoffs = cutoffs %>%
    mutate(min_fpr = ceiling(fpr*100)/100) %>%
    group_by(min_fpr) %>%
    summarise(max_tpr = max(tpr),
              min_cut = min(cut)
    )
  return(list(cutoffs = cutoffs, perf = perf))
}

plot_feature_importance = function(xgb.model, df) {
  names <- dimnames(data.matrix(df))[[2]]
  importance_matrix <- xgb.importance(names, model = xgb.model)
  xgb.plot.importance(importance_matrix[1:20, ])
}

df = df %>% filter(cashout_sum >= 0) %>% select(-c(bets_cnt, wins_cnt, ng_sum,
                                                   bonus_sum, is_only_bonus,
```

```
                                                    cum_rake, cum_bonus_sum,
cum_dep_in_sum, cum_dep_out_sum,
                                                    avg_seats, avg_stakes,
rub_minutes_share, usd_minutes_share,
                                                    days_from_last_dep
                                                    ))

df$date0 = df$date

for(i in 3:(dim(df)[2] - 1)) {
  df[[i]][is.na(df[[i]])] = 0
}

now = Sys.Date()

#FOR POKER
poker.df = df %>%
  filter(is_poker == 1) %>%
  group_by(user_id) %>%
  arrange(date) %>%
  mutate(absence_measure = as.numeric(lead(date, default = as.Date(now)) - date)
- 1) %>%
  arrange(desc(absence_measure)) %>%
  mutate(is_bad = get_essential(absence_measure, 3, 0.975)) %>%

  mutate(max_absence = max(absence_measure, na.rm = T)) %>%
  filter(max_absence > 14) %>%
  select(-max_absence) %>%

  ungroup()

poker.df5 = prepare_data(poker.df)

# poker.df.eval = poker.df5 %>% filter(last_poker == 1)

poker.df.model = poker.df5 %>% filter(last_poker == 0)

poker.df.labels = poker.df.model$is_bad
poker.df.fin = poker.df.model %>%
  select(-c(user_id, date, is_poker, is_poker.day,
            is_casino, is_casino.day, date1,
            last_poker, last_poker.day,
            last_casino, last_casino.day,
            absence_measure, is_bad))

#FIT MODEL
set.seed(666)
poker.train_id = createDataPartition(poker.df.labels, p = 0.8, list = F)
poker.watchlist = createWatchlist(poker.df.fin, poker.df.labels, poker.train_id)
poker.param = list(max.depth = 5, eta = 0.05, objective = 'binary:logistic',
eval_metric = 'auc')

poker.xgb.fit = xgb.train(
  params = poker.param,
  data = poker.watchlist$train,
  nrounds = 600,
  watchlist = poker.watchlist,
  seed = 666
)

saveRDS(poker.xgb.fit, 'poker.xgb.fit v2')

#POKER CUTOFFS
```

```
poker.cutoffs = get_cutoff(poker.xgb.fit, data.matrix(poker.df.fin),
poker.df.labels)

sqlQuery(con, 'truncate forecast.cutoffs_poker_2')

sqlSave(con,
        poker.cutoffs$cutoffs,
        'forecast.cutoffs_poker_2',
        rownames = F,
        append = T)

#FOR CASINO
casino.df = df %>%
  filter(is_casino == 1) %>%
  group_by(user_id) %>%
  arrange(date) %>%
  mutate(absence_measure = as.numeric(lead(date, default = as.Date(now)) - date)
- 1) %>%
  arrange(desc(absence_measure)) %>%
  mutate(is_bad = get_essential(absence_measure, 3, 0.975),
         date0 = date) %>%

  mutate(max_absence = max(absence_measure, na.rm = T)) %>%
  filter(max_absence > 14) %>%
  select(-max_absence) %>%

  ungroup()

casino.df5 = prepare_data(casino.df)

# casino.df.eval = casino.df5 %>% filter(last_casino == 1)

casino.df.model = casino.df5 %>% filter(last_casino == 0)

casino.df.labels = casino.df.model$is_bad
casino.df.fin = casino.df.model %>%
  select(-c(user_id, date, is_poker, is_poker.day,
            is_casino, is_casino.day, date0, date1, date0.day,
            last_poker, last_poker.day,
            last_casino, last_casino.day,
            absence_measure, is_bad))

#FIT MODEL
set.seed(666)
casino.train_id = createDataPartition(casino.df.labels, p = 0.8, list = F)
casino.watchlist = createWatchlist(casino.df.fin, casino.df.labels,
casino.train_id)
casino.param = list(max.depth = 3, eta = 0.1, objective = 'binary:logistic',
eval_metric = 'auc')

casino.xgb.fit = xgb.train(
  params = casino.param,
  data = casino.watchlist$train,
  nrounds = 200,
  watchlist = casino.watchlist,
  seed = 666
)

saveRDS(casino.xgb.fit, 'casino.xgb.fit v2')

#CASINO CUTOFF
casino.cutoffs = get_cutoff(casino.xgb.fit, data.matrix(casino.df.fin),
casino.df.labels)
```

```
sqlQuery(con, 'truncate forecast.cutoffs_casino_2')

sqlSave(con,
        casino.cutoffs$cutoffs,
        'forecast.cutoffs_casino_2',
        rownames = F,
        append = T)

close(con)
```

eval_score.R:

```
# library(RODBC)
library(RPostgreSQL)
library(xgboost)
library(dplyr)
library(reshape2)
library(zoo)

#code for geting folder of the script
initial.options = commandArgs(trailingOnly = FALSE)
file.arg.name = "--file="
script.name = sub(file.arg.name, "", initial.options[grep(file.arg.name,
initial.options)])
script.basename = dirname(script.name)
setwd(script.basename)

# con = odbcConnect('rdfraud')
drv = dbDriver('PostgreSQL')
con = dbConnect(drv, dbname = "cgames",
                host = "localhost", port = 5432,
                user = "rdfraud", password = 'IOnjVGHzgvbsy12')

# df = sqlQuery(con, 'select
#                t.*,
#                coalesce(t.poker_hour_cnt, 0) > 0 is_poker,
#                coalesce(t.casino_hour_cnt, 0) > 0 is_casino,
#                date = max(case when poker_hour_cnt > 0 then date end) over
(partition by user_id) last_poker,
#                date = max(case when casino_hour_cnt > 0 then date end) over
(partition by user_id) last_casino
#                from vip_scoring.scoring_model_view t
#                ;')
#
# sqlQuery(con, 'truncate vip_scoring.vip_score')

df = dbGetQuery(con, 'select
                t.*,
                coalesce(t.poker_hour_cnt, 0) > 0 is_poker,
                coalesce(t.casino_hour_cnt, 0) > 0 is_casino,
                date = max(case when poker_hour_cnt > 0 then date end) over
(partition by user_id) last_poker,
                date = max(case when casino_hour_cnt > 0 then date end) over
(partition by user_id) last_casino
                from vip_scoring.scoring_model_view t
                ;')
dbSendQuery(con, 'truncate vip_scoring.vip_score')

prepare_data = function(df0) {
  #function for preparing data

  #Preparing lagging 1-day [1-week, 1-month (30-days)]
```

```
  df3 = df0 %>%
    select(user_id, date) %>%
    mutate(date1 = date - 1) %>%
    melt(id.vars = c('user_id'), value.name = 'date') %>%
    select(user_id, date) %>%
    distinct() %>%
    left_join(df, by = c('user_id' = 'user_id', 'date' = 'date'))

  #LOCF-ing lagged 1-day [1-week, 1-month (30-days)]
  df4 = df3 %>%
    select(user_id, date,
           cum_avg_rake,
           cum_avg_bonuses,
           cum_dep_in_sum_over_cum_dep_in_cnt,
           cum_dep_out_sum_over_cum_dep_out_cnt,
           cum_ng_sum,
           date0,
           age) %>%
    arrange(user_id, date) %>%
    group_by(user_id) %>%
    mutate_all(funs(na.locf(., na.rm = F))) %>%
    mutate(absence_weight = as.numeric(date - date0),
           cum_avg_rake = cum_avg_rake*age/(age + absence_weight),
           cum_avg_bonuses = cum_avg_bonuses*age/(age + absence_weight)) %>%
    select(user_id, date,
           cum_avg_rake,
           cum_avg_bonuses,
           cum_dep_in_sum_over_cum_dep_in_cnt,
           cum_dep_out_sum_over_cum_dep_out_cnt,
           cum_ng_sum) %>%
    ungroup()

  df3 = df3 %>%
    select(-c(cum_avg_rake,
              cum_avg_bonuses,
              cum_dep_in_sum_over_cum_dep_in_cnt,
              cum_dep_out_sum_over_cum_dep_out_cnt,
              cum_ng_sum,
              age)) %>%
    filter(!is.na(date)) %>%
    mutate_all(funs(na.fill(., 0, na.rm = F)))

  df5 = df0 %>%
    mutate(date1 = date - 1) %>%
    inner_join(df3, by = c('user_id' = 'user_id', 'date1' = 'date'), suffix =
c('', '.day')) %>%
    inner_join(df4, by = c('user_id' = 'user_id', 'date1' = 'date'), suffix =
c('', '.day')) %>%
    filter(!is.na(cum_avg_rake.day))

  return(df5)
}

get_essential = function(absence_measure, lower_bound, target.quantile) {
  #function for target field
  total_diff = sum(absence_measure)
  diff_sum = ifelse(total_diff == 0, 1, cumsum(absence_measure)/total_diff)
  max_absence = max(absence_measure)
  return(ifelse(diff_sum <= target.quantile & absence_measure >
ifelse(max_absence <= 3, 1, lower_bound), 1, 0))
}

df = df %>% filter(cashout_sum >= 0) %>% select(-c(bets_cnt, wins_cnt, ng_sum,
```

```
                                                    bonus_sum, is_only_bonus,
                                                    cum_rake, cum_bonus_sum,
cum_dep_in_sum, cum_dep_out_sum,
                                                    avg_seats, avg_stakes,
rub_minutes_share, usd_minutes_share,
                                                    days_from_last_dep
))

df$date0 = df$date

for(i in 3:(dim(df)[2] - 1)) {
  df[[i]][is.na(df[[i]])] = 0
}

now = Sys.Date()

#EVAL POKER
poker.df = df %>%
  filter(is_poker == 1) %>%
  mutate(date0 = date)

poker.df5 = prepare_data(poker.df)

poker.df.eval = poker.df5 %>% filter(last_poker == 1)

poker.fin = poker.df.eval %>%
  select(-c(user_id, date, is_poker, is_poker.day,
            is_casino, is_casino.day, date1,
            last_poker, last_poker.day,
            last_casino, last_casino.day))

poker.xgb.fit = readRDS('poker.xgb.fit v2')

poker.pred = predict(poker.xgb.fit, newdata = data.matrix(poker.fin))

poker.df.m = df %>%
  filter(is_poker == 1) %>%
  group_by(user_id) %>%
  arrange(date) %>%
  mutate(absence_measure = as.numeric(lead(date, default = as.Date(now)) - date)
- 1) %>%
  arrange(desc(absence_measure)) %>%
  mutate(is_bad = get_essential(absence_measure, 3, 0.975)) %>%
  filter(is_bad == 1) %>%
  summarise(m = median(absence_measure))

tmp.db = data.frame(
  model_date = as.Date(now),
  type = 'poker',
  user_id = poker.df.eval$user_id,
  last_date = as.Date(poker.df.eval$date),
  score0 = poker.pred
) %>%
  inner_join(poker.df.m, by = 'user_id') %>%
  mutate(score1 = ifelse(score0 < 0.95, 0.95, score0 + (1- score0)/2),
         mu = m*log((1-score0)/score0)/log(((1-score0)*score1)/(score0*(1-
score1))),
         s = m/log(((1-score0)*score1)/(score0*(1-score1))),
         n = as.numeric(now - last_date) - 1,
         score = 1/(1 + exp(-(n - mu)/s))
  )

# sqlSave(con,
```

```
#         tmp.db,
#         'vip_scoring.vip_score',
#         rownames = F,
#         append = T
# )
dbWriteTable(con,
             c('vip_scoring', 'vip_score'),
             tmp.db,
             row.names = F,
             append = T)

#EVAL CASINO
casino.df = df %>%
  filter(is_casino == 1) %>%
  mutate(date0 = date)

casino.df5 = prepare_data(casino.df)

casino.df.eval = casino.df5 %>% filter(last_casino == 1)

casino.fin = casino.df.eval %>%
  select(-c(user_id, date, is_poker, is_poker.day,
            is_casino, is_casino.day, date0, date1, date0.day,
            last_poker, last_poker.day,
            last_casino, last_casino.day))

casino.xgb.fit = readRDS('casino.xgb.fit v2')

casino.pred = predict(casino.xgb.fit, newdata = data.matrix(casino.fin))

casino.df.m = df %>%
  filter(is_casino == 1) %>%
  group_by(user_id) %>%
  arrange(date) %>%
  mutate(absence_measure = as.numeric(lead(date, default = as.Date(now)) - date)
- 1) %>%
  arrange(desc(absence_measure)) %>%
  mutate(is_bad = get_essential(absence_measure, 3, 0.975)) %>%
  filter(is_bad == 1) %>%
  summarise(m = median(absence_measure))

tmp.db = data.frame(
  model_date = now,
  type = 'casino',
  user_id = casino.df.eval$user_id,
  last_date = as.Date(casino.df.eval$date),
  score0 = casino.pred
) %>%
  inner_join(casino.df.m, by = 'user_id') %>%
  mutate(score1 = ifelse(score0 < 0.95, 0.95, score0 + (1- score0)/2),
         mu = m*log((1-score0)/score0)/log(((1-score0)*score1)/(score0*(1-
score1))),
         s = m/log(((1-score0)*score1)/(score0*(1-score1))),
         n = as.numeric(now - last_date) - 1,
         score = 1/(1 + exp(-(n - mu)/s))
  )

# sqlSave(con,
#         tmp.db,
#         'vip_scoring.vip_score',
#         rownames = F,
#         append = T
# )
```

```
dbWriteTable(con,
             c('vip_scoring', 'vip_score'),
             tmp.db,
             row.names = F,
             append = T)

# close(con)
dbDisconnect(con)
```

scoring_model_view.sql:

```
create view scoring_model_view as
  SELECT fs.user_id,
    fs.date,
    fs.bets_cnt,
    fs.wins_cnt,
    fs.bets_sum,
    fs.wins_sum,
    fs.ng_sum,
    fs.buyin_sum,
    fs.rebuyin_sum,
    fs.cashout_sum,
    fs.payout_sum,
    fs.pokerbet_sum,
    fs.rake_sum,
    fs.casino_hour_cnt,
    fs.poker_hour_cnt,
    fs.bonus_sum,
    fs.dep_in_cnt,
    fs.dep_in_sum,
    fs.dep_out_cnt,
    fs.dep_out_sum,
    fs.is_only_bonus,
    fs.avg_seats,
    fs.avg_stakes,
    fs.rub_minutes_share,
    fs.usd_minutes_share,
    COALESCE(sum(fs.rake_sum) OVER (PARTITION BY fs.user_id ORDER BY fs.user_id,
fs.date), (0)::double precision) AS cum_rake,
    ((fs.date - min(fs.date) OVER (PARTITION BY fs.user_id)) + 1) AS age,
    (COALESCE(sum(fs.rake_sum) OVER (PARTITION BY fs.user_id ORDER BY
fs.user_id, fs.date), (0)::double precision) / (((fs.date - min(fs.date) OVER
(PARTITION BY fs.user_id)) + 1))::double precision) AS cum_avg_rake,
        CASE
            WHEN ((fs.dep_out_sum IS NULL) AND (vp.id IS NULL)) THEN
NULL::integer
            ELSE ntile(4) OVER (PARTITION BY
            CASE
                WHEN ((fs.dep_out_sum IS NULL) AND (vp.id IS NULL)) THEN 1
                ELSE 2
            END ORDER BY fs.dep_out_sum)
        END AS quart_dep_out_sum,
        CASE
            WHEN ((fs.dep_in_sum IS NULL) AND (vp.id IS NULL)) THEN
NULL::integer
            ELSE ntile(4) OVER (PARTITION BY
            CASE
                WHEN ((fs.dep_in_sum IS NULL) AND (vp.id IS NULL)) THEN 1
                ELSE 2
            END ORDER BY fs.dep_in_sum)
        END AS quart_dep_in_sum,
```

```
        CASE
            WHEN ((((fs.wins_sum IS NULL) AND (fs.bets_sum IS NULL)) OR
((fs.wins_sum = (0)::double precision) AND (fs.bets_sum = (0)::double
precision))) AND (vp.id IS NULL)) THEN NULL::integer
            ELSE ntile(4) OVER (PARTITION BY
            CASE
                WHEN (((fs.wins_sum IS NULL) AND (fs.bets_sum IS NULL)) OR
((fs.wins_sum = (0)::double precision) AND (fs.bets_sum = (0)::double precision)
AND (vp.id IS NULL))) THEN 1
                ELSE 2
            END ORDER BY fs.wins_sum)
        END AS quart_wins_sum,
        CASE
            WHEN ((((fs.wins_sum IS NULL) AND (fs.bets_sum IS NULL)) OR
((fs.wins_sum = (0)::double precision) AND (fs.bets_sum = (0)::double
precision))) AND (vp.id IS NULL)) THEN NULL::integer
            ELSE ntile(4) OVER (PARTITION BY
            CASE
                WHEN (((fs.wins_sum IS NULL) AND (fs.bets_sum IS NULL)) OR
((fs.wins_sum = (0)::double precision) AND (fs.bets_sum = (0)::double precision)
AND (vp.id IS NULL))) THEN 1
                ELSE 2
            END ORDER BY fs.bets_sum)
        END AS quart_bets_sum,
    COALESCE(sum(fs.bonus_sum) OVER (PARTITION BY fs.user_id ORDER BY
fs.user_id, fs.date), (0)::double precision) AS cum_bonus_sum,
    (COALESCE(sum(fs.bonus_sum) OVER (PARTITION BY fs.user_id ORDER BY
fs.user_id, fs.date), (0)::double precision) / (((fs.date - min(fs.date) OVER
(PARTITION BY fs.user_id)) + 1))::double precision) AS cum_avg_bonuses,
    COALESCE(sum(fs.dep_in_cnt) OVER (PARTITION BY fs.user_id ORDER BY
fs.user_id, fs.date), (0)::numeric) AS cum_dep_in_cnt,
    COALESCE(sum(fs.dep_in_sum) OVER (PARTITION BY fs.user_id ORDER BY
fs.user_id, fs.date), (0)::double precision) AS cum_dep_in_sum,
    (COALESCE(sum(fs.dep_in_sum) OVER (PARTITION BY fs.user_id ORDER BY
fs.user_id, fs.date), (0)::double precision) /
(COALESCE(NULLIF(sum(fs.dep_in_cnt) OVER (PARTITION BY fs.user_id ORDER BY
fs.user_id, fs.date), (0)::numeric), (1)::numeric))::double precision) AS
cum_dep_in_sum_over_cum_dep_in_cnt,
    COALESCE(sum(fs.dep_out_cnt) OVER (PARTITION BY fs.user_id ORDER BY
fs.user_id, fs.date), (0)::numeric) AS cum_dep_out_cnt,
    COALESCE(sum(fs.dep_out_sum) OVER (PARTITION BY fs.user_id ORDER BY
fs.user_id, fs.date), (0)::double precision) AS cum_dep_out_sum,
    (COALESCE(sum(fs.dep_out_sum) OVER (PARTITION BY fs.user_id ORDER BY
fs.user_id, fs.date), (0)::double precision) /
(COALESCE(NULLIF(sum(fs.dep_out_cnt) OVER (PARTITION BY fs.user_id ORDER BY
fs.user_id, fs.date), (0)::numeric), (1)::numeric))::double precision) AS
cum_dep_out_sum_over_cum_dep_out_cnt,
    COALESCE(sum(fs.ng_sum) OVER (PARTITION BY fs.user_id ORDER BY fs.user_id,
fs.date), (0)::double precision) AS cum_ng_sum,
    (fs.date - max(
        CASE
            WHEN (fs.dep_in_cnt > 0) THEN fs.date
            ELSE NULL::date
        END) OVER (PARTITION BY fs.user_id ORDER BY fs.date)) AS
days_from_last_dep
   FROM (vip_scoring.scoring_model_sample fs
     JOIN vips vp ON ((vp.id = fs.user_id)));
```

vip_score_view.sql:

```
create view vip_score_view as
  WITH tmp_games AS (
```

```
         SELECT vs_1.user_id,
            (sum(sp.rakeusd) / (3)::double precision) AS rake,
            (sum(sp.ngusd) / (3)::double precision) AS ng
           FROM (vip_scoring.vip_score vs_1
             JOIN pomodorro_spins sp ON ((sp.user_id = vs_1.user_id)))
          WHERE (true AND (sp.date >= (vs_1.last_date - '21 days'::interval)))
          GROUP BY vs_1.user_id
        ), tmp_dep AS (
         SELECT vs_1.user_id,
            percentile_disc((0.5)::double precision) WITHIN GROUP (ORDER BY
dep.sumusd) AS dep_median
           FROM (vip_scoring.vip_score vs_1
             JOIN aqua_curr_deps dep ON (((vs_1.user_id = dep.id) AND (dep.isin
= 1) AND (dep.depdate <= vs_1.last_date))))
          GROUP BY vs_1.user_id
        ), tmp_vip AS (
         SELECT vs_1.user_id,
            array_agg(vs_1.type ORDER BY vs_1.type) AS type,
            array_agg(vs_1.last_date ORDER BY vs_1.type) AS last_date,
            array_agg(vs_1.score ORDER BY vs_1.type) AS score,
            array_agg(
                CASE
                    WHEN (vs_1.type = 'poker'::text) THEN
(vip_scoring.get_poker_cutoff() < vs_1.score)
                    WHEN (vs_1.type = 'casino'::text) THEN
(vip_scoring.get_casino_cutoff() < vs_1.score)
                    ELSE NULL::boolean
                END ORDER BY vs_1.type) AS is_bad
           FROM vip_scoring.vip_score vs_1
          GROUP BY vs_1.user_id
        )
 SELECT vs.user_id,
    vs.type,
    vs.last_date,
    vs.score,
    vs.is_bad,
    tg.rake,
    tg.ng,
    td.dep_median
   FROM ((tmp_vip vs
     LEFT JOIN tmp_games tg ON ((vs.user_id = tg.user_id)))
     LEFT JOIN tmp_dep td ON ((tg.user_id = td.user_id)));
```