Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра прикладної математики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розв'язання задачі прогнозування
відтоку гравців в ігровій індустрії
(тема)

Виконав:
студент _2_ курсу, групи _САУм-22-1
Ломія С.Г.
(прізвище, ініціали)
Спеціальність
124 Системний аналіз
(код і повна назва спеціальності)
Тип програми _освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма
Системний аналіз і управління
(повна назва освітньої програми)
Керівник доц. Гибкіна Н.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ _____ Сидоров М.В.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _інформаційно-аналітичних технологій та менеджменту
Кафедра _прикладної математики
Рівень вищої освіти _другий (магістерський)
Спеціальність _124 Системний аналіз
(код і повна назва)
Тип програми _освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма _Системний аналіз і управління
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ _____
(підпис)
“_06_” _листопада_ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _Ломії Сергію Гівійовичу
(прізвище, ім’я, по батькові)

1. Тема роботи _Розв’язання задачі прогнозування відтоку гравців в ігровій індустрії
затверджена наказом по університету від 2 листопада 2023 р. № 1277 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10 січня 2024 р.

3. Вихідні дані до роботи _набір даних гравців мобільної гри в жанрі Fantasy Collection RPG

4. Перелік питань, що потрібно опрацювати в роботі
1. Системний аналіз предметної області
2. Вибір і обґрунтування методу розв’язання
3. Програмна реалізація
4. Результати обчислювального експерименту
5. Аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

1. Актуальність теми роботи
2. Постановка задачі
3. Системний аналіз предметної області
4. Метод чисельного аналізу
5. Результати обчислювального експерименту

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	6 – 12 листопада 2023 р.	виконано
2	Вибір та обґрунтування методу	13 – 26 листопада 2023 р.	виконано
3	Розробка алгоритму і програми	27 листопада – 10 грудня 2023 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	11 грудня – 24 грудня 2023 р.	виконано
5	Робота над текстом пояснювальної записки	25 грудня 2023 р. – 9 січня 2024 р.	виконано
6	Представлення роботи на рецензію в ЕК	10 січня 2024 р.	виконано

Дата видачі завдання _6_ листопада 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Гибкіна Н.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 67 с., 7 табл., 17 рис., 1 дод., 11 джерела.

ІГРОВА ІНДУСТРІЯ, ЙМОВІРНІСТЬ ВІДТОКУ ГРАВЦІВ, ВИХІД З ГРИ, ОЗНАКА, ГРАДІЄНТНИЙ БУСТІНГ, ЗАДАЧА КЛАСИФІКАЦІЇ

Об’єкт дослідження – процес відтоку гравців мобільної гри.

Мета роботи – дослідження можливостей прогнозування відтоку гравців у ігровій індустрії.

Методи дослідження – метод градієнтного бустінгу.

У кваліфікаційній роботі розглянуто проблему прогнозування відтоку гравців у ігровій індустрії та проаналізовано існуючі підходи до розв’язання цієї задачі як задачі класифікації. Виконано системний аналіз проблеми із врахуванням особливостей набору даних обраної мобільної гри та обрано найбільш підходящий метод машинного навчання для прогнозування відтоку гравців. Розроблено програмний продукт, який дозволив розв’язати поставлену задачу, обрано найкращі для заданого датасету значення гіперпараметрів моделі та ті ознаки, які є найвпливовішими для ймовірності відтоку.

ABSTRACT

Introductory note: 67 pages, 7 tables, 17 figures, 1 appendix, 11 sources.

GAMING INDUSTRY, PROBABILITY OF PLAYER CHURN, GAME EXIT, FEATURE, GRADIENT BOOSTING, CLASSIFICATION.

Object of research – the process of outflow of mobile game players.

Purpose of work – studying the possibilities of predicting the outflow of players in the gaming industry.

Methods of research – Gradient boosting method.

The qualification work considers the problem of predicting player churn in the gaming industry and analyzes existing approaches to solving this problem as a classification problem. A systematic analysis of the problem is performed, taking into account the features of the data set of the selected mobile game, and the most suitable machine learning method for predicting player churn is selected. A software product was developed that allowed solving the problem, selecting the best values of the model hyperparameters for a given dataset and the features that are most influential for the probability of churn.

ЗМІСТ

ВСТУП

Актуальність теми. Ігрова індустрія сьогодні є однією з найшвидше зростаючих індустрій у світі розваг та розвитку технологій. Однак, незважаючи на її розвиток та популярність, виникає важлива проблема – відтік гравців, оскільки здатність зберегти і зацікавити користувачів у довгостроковій перспективі є одним з ключових аспектів успіху.

Припинення активності користувачів та їх відхід від гри стає серйозним викликом для розробників та видавців ігор. Ця проблема потребує ретельного аналізу та розробки стратегій, спрямованих на її вирішення. Вивчення та розгляд можливостей прогнозування відтоку гравців включає в себе аналіз методів збору та обробки даних, використання різноманітних моделей машинного навчання для прогнозування відтоку, а також розробку стратегій утримання гравців та попередження їх відтоку.

Застосування методів аналізу даних та технологій штучного інтелекту дозволяє розробникам і видавцям ігор покращувати стратегії утримання користувачів та забезпечувати стабільність та розвиток своїх продуктів. Вирішення проблеми відтоку є стратегічним кроком для забезпечення стабільності гри та доходів.

Мета і завдання кваліфікаційної роботи. Метою кваліфікаційної роботи є дослідження можливостей прогнозування відтоку гравців у ігровій індустрії. Для досягнення поставленої мети необхідно виконати наступні завдання:

– провести огляд і аналіз сучасного стану задачі «прогнозування відтоку гравців у ігровій індустрії»;

– розв’язати задачу прогнозування відтоку гравців у ігровій індустрії обраним методом машинного навчання;

– розробити програмний продукт, який дозволить виконувати прогноз відтоку гравців;

– провести обчислювальні експерименти та виконати аналіз отриманих результатів, зокрема визначити ефективність використання обраного алгоритму при прогнозуванні.

Об'єктом дослідження є процес відтоку гравців в ігровій індустрії.

Предметом дослідження є прогнозування відтоку гравців в ігровій індустрії.

Методи дослідження. У кваліфікаційній роботі використовуються методи машинного навчання для прогнозування відтоку гравців.

Публікації. Результати, отримані у кваліфікаційній роботі, було представлено на 27-му Міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті» (м. Харків, 10-12 травня 2023 р.) [1].

1 СИСТЕМНИЙ АНАЛІЗ ПРОБЛЕМИ ПРОГНОЗУВАННЯ ВІДТОКУ ГРАВЦІВ В ІГРОВІЙ ІНДУСТРІЇ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Системний аналіз проблеми прогнозування відтоку гравців в ігровій індустрії

1.1.1 Вербальна модель системи

Розглянемо вербальну модель системи «База даних характеристик гравців у комп’ютерній грі».

Об’єкт аналізу – «Поведінка гравців щодо їх можливого остаточного виходу з гри».

Предмет аналізу – «Проблема прогнозування відтоку гравців в ігровій індустрії».

Точка зору: дослідник.

Призначення системи: розв’язання проблеми прогнозування відтоку гравців в ігровій індустрії.

Мета системи – отримання прогнозу щодо конкретного гравця щодо того, чи буде він продовжувати грати чи схильний до припинення в найближчому майбутньому активності у грі.

Виконаємо класифікацію системи.

За функціоналом систему можна розглядати як виробничу, призначену для зберігання, обробки та надання даних про характеристики гравців у реальному часі для оперативної підтримки бізнес-процесів, таких як утримання клієнтів та аналіз їх поведінки, та як аналітичну, де основний акцент робиться на обробку даних для аналізу трендів, виявлення патернів і створення звітів для підтримки прийняття стратегічних рішень щодо подальшої поведінки гравців.

За масштабом система є локальною, тому що використовується для підтримки гравців в рамках конкретної гри або платформи.

За типом даних система є структурованою, оскільки більшість даних у ній представлено у форматі таблиць та структурованих записів, зокрема, даних про користувачів, транзакції та досягнення. Частина даних, наприклад, відгуки гравців та інші форми зворотного зв’язку, можуть містити текстову інформацію, що ускладнює їх аналіз.

За ступенем деталізації даних система є деталізованою, бо включає безліч характеристик і параметрів для більш точного аналізу та управління відтоком гравців. Ці дані можуть бути використані для широкого аналізу та статистики без детального аналізу.

За рівнем конфіденційності система є системою з обмеженим відповідно до політик безпеки та конфіденційності доступом.

Класифікація системи допомагає зрозуміти її основні характеристики, що важливо для правильного настроювання, використання та супроводу системи бази даних.

Ідентифікованість системи полягає у виборі методу розв’язання задачі прогнозування відтоку гравців. Саме це відрізнятиме розглядувану систему від інших схожих систем. Система забезпечуватиме високу продуктивність, оскільки всі її процеси спрямовані на оптимізацію роботи та ефективне використання ресурсів, таких як час, витрачений на розробку алгоритму та виконання програми з метою досягнення остаточного результату у розв’язанні задачі прогнозування відтоку гравців.

1.1.2 Морфологічний опис системи

Виконаємо морфологічний опис системи «База даних характеристик гравців у комп’ютерній грі». Морфологічна модель включає опис структури, складу, меж, зовнішнього середовища, у тому числі «чорну скриньку» [2].

Почнемо розгляд з моделі типу «чорна скринька» для системи «База даних характеристик гравців у комп’ютерній грі» (рисунок 1.1). Входом до моделі «чорна скринька» є різні характеристики гравців, такі як активність, час, проведений у грі,

виконані транзакції, рівень досягнень тощо, які можуть бути важливими при ухваленні рішення про остаточний вихід з гри. Виходом моделі «чорна скринька» буде прогноз того, чи буде конкретний гравець продовжувати грати або, навпаки, він схильний до припинення в найближчому майбутньому активності у цій грі. Склад «чорної скриньки» користувачем не досліджується, увага приділяється тільки межам системи, оскільки вони підкреслюють її цілісність, відокремленість від зовнішнього середовища та взаємодію системи і середовища.

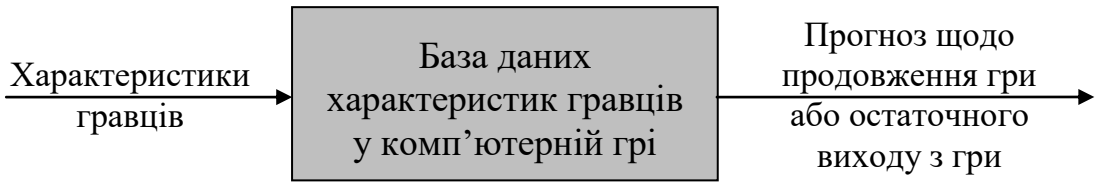

Рисунок 1.1 – Модель типу «чорна скринька»

Перейдемо до опису структури та складу системи. Розглядувана система є структурованим сховищем даних, метою якого є зберігання та управління інформацією про характеристики гравців комп'ютерної гри. База даних використовується для аналітики та прийняття управлінських рішень щодо утримання гравців та запобігання їх відтоку.

Ключовими компонентами системи є:

– таблиці даних;

– схема бази даних;

– метадані та документація;

– системи безпеки;

– модуль аналізу даних;

– API.

Розглянемо детально ці компоненти.

Таблиці даних містять інформацію про гравців, транзакції та активність у грі, зокрема:

– основні дані кожного гравця, включаючи унікальні ідентифікатори, демографічну інформацію та інформацію про час, проведений у грі;

– дані про фінансові транзакції гравців, такі як покупки всередині гри;

– дані про активність гравця, такі як частота входу, тривалість ігрових сесій та досягнення.

Схема бази даних включає інформацію про зв’язки між таблицями даних і компоненти прискорення доступу до даних та оптимізації продуктивності.

Метадані та документація містять інформацію про:

– опис полів таблиць даних з метою полегшення розуміння та використання даних;

– опис структури бази даних, типів даних та логіки зберігання інформації.

Системи безпеки реалізують організацію ролей та прав доступу з метою забезпечення доступу до даних відповідно до рівня привілеїв, а також механізми резервного копіювання та відновлення для запобігання втраті даних та забезпечення можливості їх відновлення у разі збоїв.

Модуль аналізу даних призначений для отримання відповідей на запити щодо для даних гравців з метою виявлення патернів їх поведінки та створення аналітичних звітів щодо відтоку гравців.

Інтерфейс інтеграції з ігровою платформою (API) забезпечує автоматичне оновлення даних.

Проаналізуємо межі системи. Межею системи «База даних характеристик гравців» є простір, в якому дослідник може досліджувати задачу прогнозування відтоку гравців та розробляти алгоритм для її розв’язання.

Зовнішнє середовище системи «База даних характеристик гравців у комп’ютерній грі» включає різні елементи, які взаємодіють з системою або знаходяться в її оточенні. До основних елементів зовнішнього середовища відносяться:

а) ігрова платформа – система взаємодіє з ігровою платформою через API для отримання актуальних даних про користувачів, транзакції та інші важливі події у грі;

б) інструменти аналізу даних – зовнішні інструменти, які можуть використовуватися для більш глибокого аналізу даних з бази даних, зокрема, модулі, які використовують дані з бази даних для прийняття рішень та розробки стратегій щодо утримання гравців;

в) персонал (адміністратори та оператори) – працівники, відповідальні за обслуговування та підтримку роботи бази даних, включаючи адміністрування, моніторинг та вирішення проблем;

г) законодавчі нормативи – закони та правила, що регулюють збір, зберігання та обробку даних, можуть впливати на політики та процеси всередині системи бази даних;

д) постачальники технологій та обладнання – компанії, що надають послуги хостингу та обладнання, на якому працює база даних;

е) користувачі та гравці – користувачі системи, включаючи аналітиків, розробників та інші зацікавлені сторони, які можуть взаємодіяти із даними через інтерфейс системи.

Обмін інформацією між системою «База даних характеристик гравців» та елементами зовнішнього середовища відіграє ключову роль у забезпеченні ефективного управління та аналізу даних з метою покращення характеристик гравців та запобігання їх відтоку.

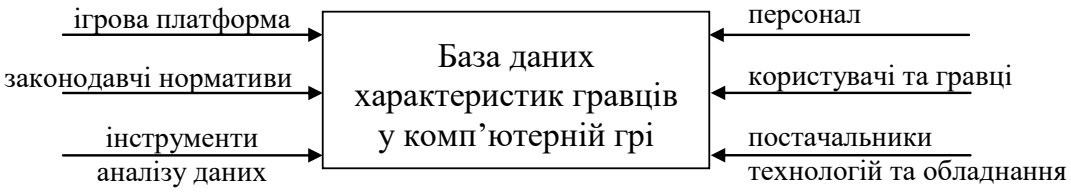

Рисунок 1.2 – Модель зовнішнього середовища системи

1.1.3 Функціональна модель системи

Метою системного аналізу є дослідження проблеми класифікації гравців з точки зору їх остаточного виходу з гри у системі «База даних характеристик гравців у комп'ютерній грі», тому розглянемо систему «Розв'язання задачі прогнозування відтоку гравців в ігровій індустрії».

Функціональна модель системи «Розв’язання задачі прогнозування відтоку гравців в ігровій індустрії» включає процеси, спрямовані на виявлення гравців, схильних до виходу з гри. До цих процесів належать:

– збір даних про ігрову активність (частота входу в гру, тривалість ігрових сесій, досягнення тощо), транзакції (покупки у грі тощо), особисті дані гравців (вік, стать, місцезнаходження тощо);

– підготовка даних, зокрема, їх очищення та перетворення з метою усунення помилок, заповнення пропущених значень та приведення до стандартних форматів, а також інтеграція даних, тобто об’єднання даних із різних джерел для створення загального набору характеристик;

– вибір ознак та інженерія ознак, зокрема, визначення найбільш значущих ознак, які можуть впливати на відтік, створення нових ознак з множини наявних, які можуть поліпшити модель;

– вибір моделі машинного навчання, що передбачає вибір відповідних алгоритмів класифікації, таких як логістична регресія, дерева рішень, нейронні мережі тощо, та налаштування гіперпараметрів моделі для досягнення кращої продуктивності;

– навчання моделі на тренувальних даних, що містять інформацію про гравців із відомим статусом відтоку;

– валідація та оцінка моделі, зокрема, використання метрик для оцінки ефективності моделі;

– прогнозування відтоку, тобто використання навченої моделі для прогнозування ймовірності відтоку нових гравців;

– інтеграція із системою управління відтоком для подальшого використання результатів прогнозування у стратегіях утримання гравців;

– моніторинг ефективності та оновлення, тобто регулярне відстеження роботи моделі та її оновлення за потреби за допомогою нових даних для підтримки актуальності прогнозів.

Графічне подання функціонального опису системи «Розв’язання задачі прогнозування відтоку гравців в ігровій індустрії» можна здійснити за

допомогою контекстної діаграми IDEF0 (рис. 1.3). Входами в систему є характеристики гравців, що зберігаються у табличному вигляді, зручному для подальшої обробки та аналізу. До механізмів системи відносяться обчислювальна техніка, за допомогою якої проводяться дослідження, та дослідник, який приймає рішення в процесі дослідження. До управління системою «Розв'язання задачі прогнозування відтоку гравців в ігровій індустрії» відносяться математичні методи та інформаційні технології, які є теоретичним підґрунтям та інструментами для розв'язання поставленої задачі, а саме: методи оптимізації (зокрема, градієнтні методи оптимізації), методи прийняття рішень (зокрема, дерева рішень), спеціалізовані засоби програмування (зокрема, бібліотеки Pandas та Sklearn, що використовуються для розширення можливостей мови програмування Python у аналізі даних). Виходом з системи є розв'язок задачі, тобто прогноз остаточного виходу з гри для окремих клієнтів.

Декомпозиція контекстної діаграми (рисунок 1.4) містить основні етапи, необхідні для розв'язання поставленої задачі прогнозування відтоку гравців у ігровій індустрії. Наступним етапом здійснюється декомпозиція кожного етапу системи до досягнення заданої глибини деталізації опису. Як приклад, наведемо декомпозицію етапу «Виконати попередню обробку даних» за допомогою побудови наступної діаграми в ієрархії із зазначенням функцій, які реалізуються на цьому етапі, та механізмів їх реалізації (рисунок 1.5).

На рисунку 1.6 зображена IDEF3-діаграма, яка дозволяє моделювати взаємодію функціональних блоків у системі, а також аналізувати потоки даних та управління між ними [3]. На IDEF3-діаграмі розглядуваного у роботі процесу прогнозування відтоку гравців вказано, що необхідними для вирішення поставленої задачі є дані, що містять інформацію про характеристики гравців. На наступному етапі ці дані підлягають попередній обробці, а також обирається відповідна модель машинного навчання, яка дозволить розв'язати поставлену задачу. Ця інформація є вхідною до наступного етапу – навчання моделі. Наступним кроком навчена модель подається на вхід завершального етапу, на якому виконується прогноз ймовірності відтоку окремих клієнтів.

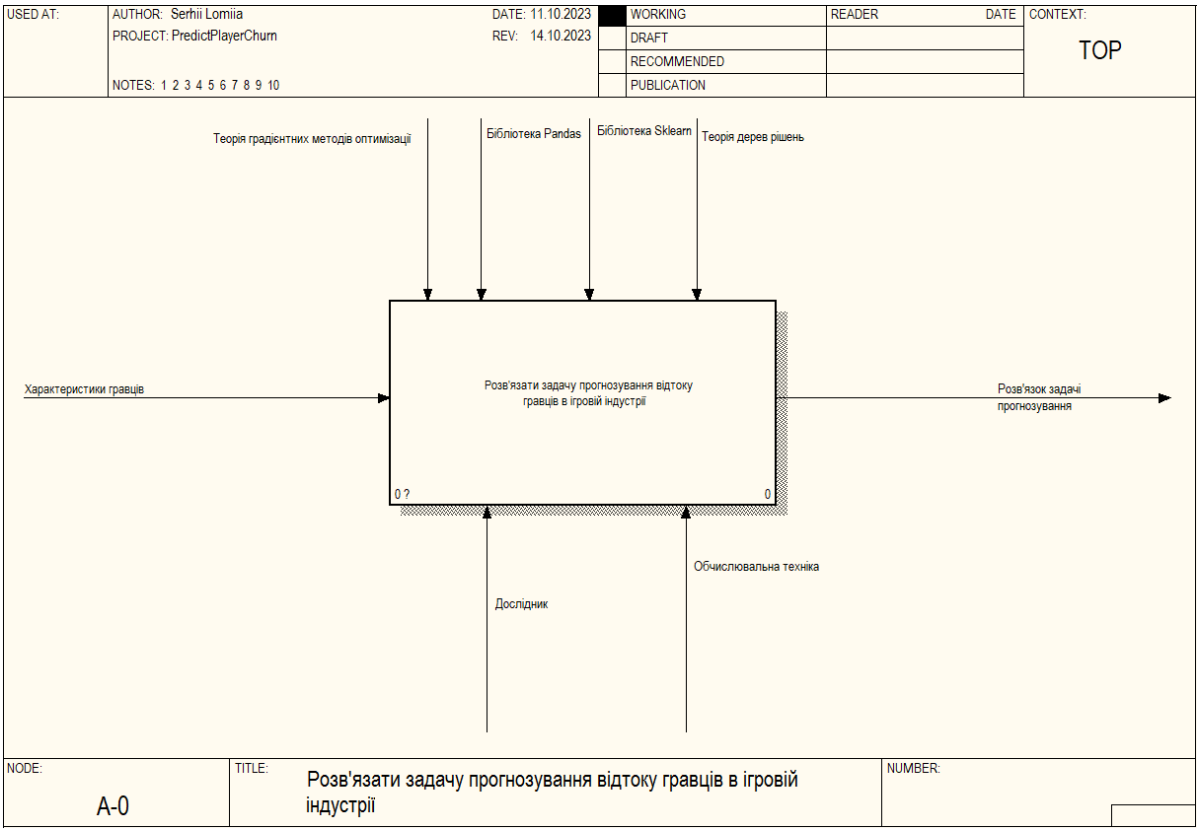

Рисунок 1.3 – Контекстна діаграма (рівень A-0)

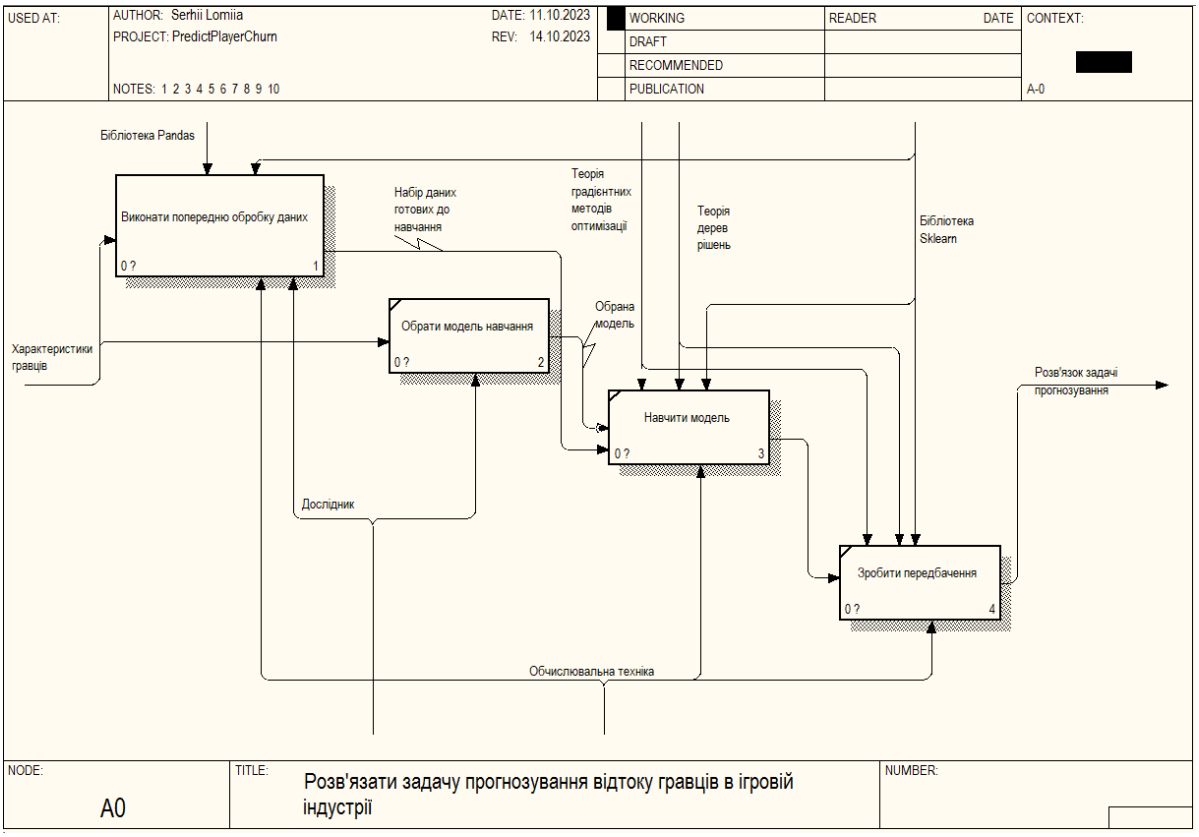

Рисунок 1.4 – Декомпозиція роботи «Розв’язати задачу прогнозування відтоку гравців в ігровій індустрії»: рівень A0

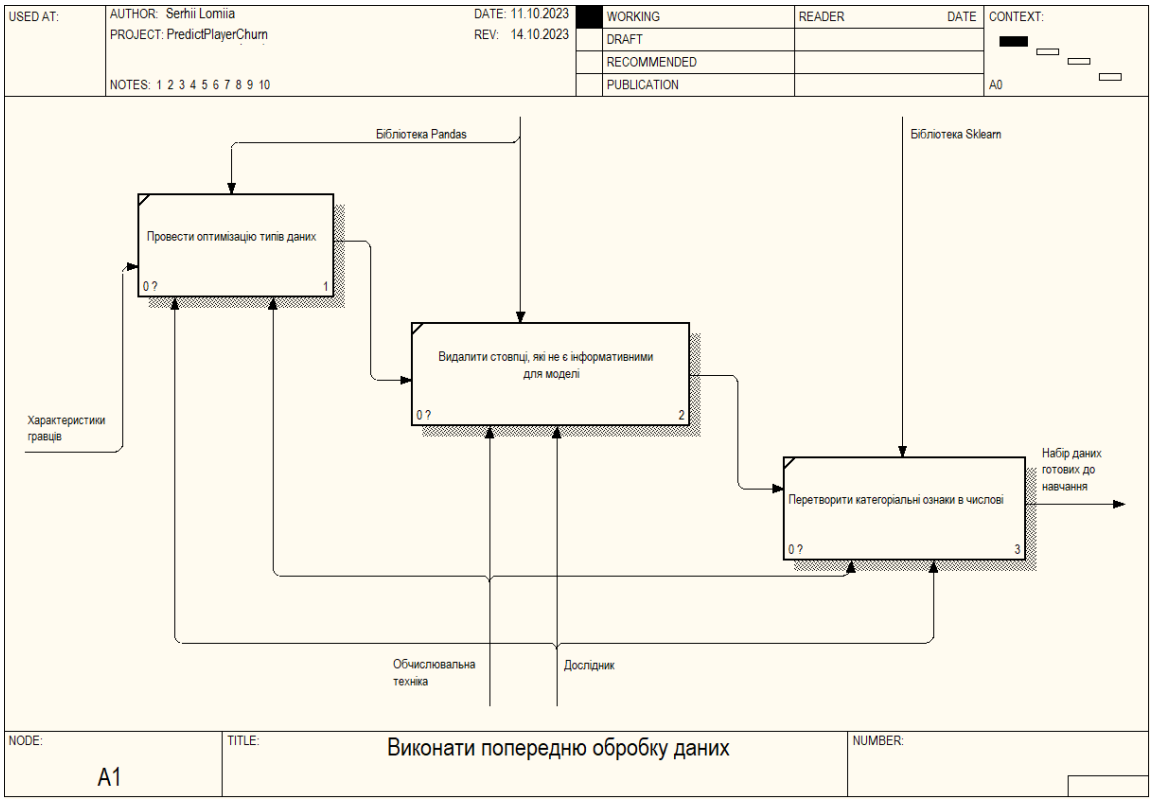

Рисунок 1.5 – Декомпозиція роботи «Виконати попередню обробку даних»: рівень A1

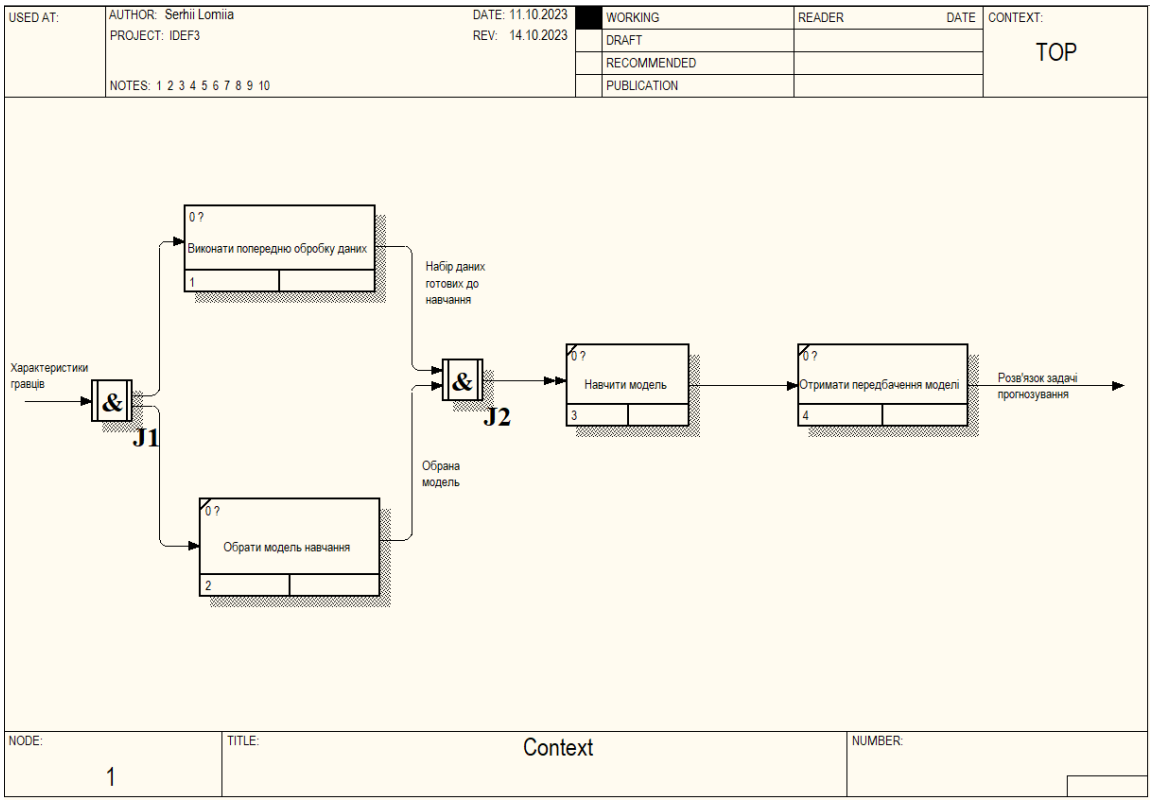

Рисунок 1.6 – Опис роботи «Розв’язати задачу прогнозування відтоку гравців в ігровій індустрії»: рівень A0 (в нотації IDEF3)

На рисунку 1.7 наведений опис роботи «Виконати попередню обробку даних»в нотації IDEF3, як одного з етапів розв’язання задачі прогнозування відтоку гравців.

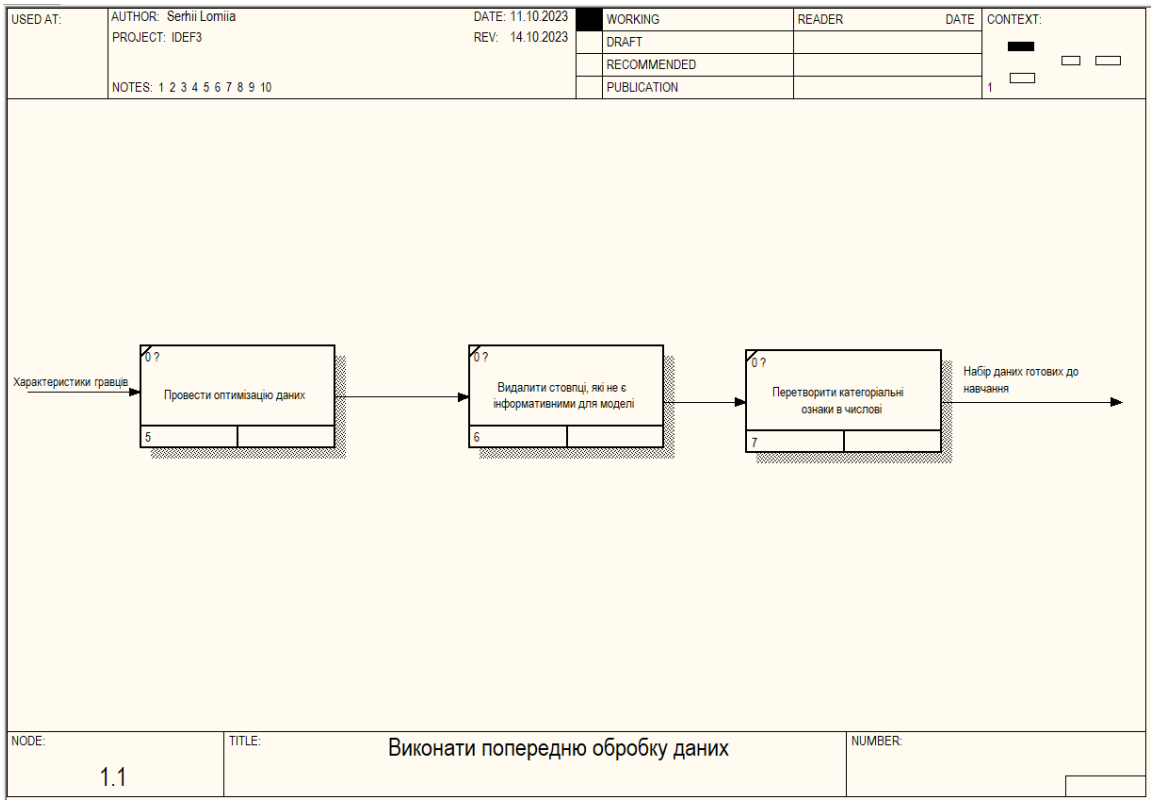

Рисунок 1.7 – Опис роботи «Виконати попередню обробку даних»: рівень A1 (в нотації IDEF3)

1.1.4 Інформаційна модель

Інформаційна модель системи «Розв’язання задачі прогнозування відтоку гравців у комп’ютерній грі» дозволяє описати та візуалізувати структуру даних, їх взаємозв’язки та основні елементи, необхідні для ефективного вирішення завдання прогнозування відтоку. Основними компонентами інформаційної моделі є наступні:

– ігрова активність, зокрема інформація про частоту входу в гру, тривалість ігрових сесій гравців у грі, записи про досягнення та успішно виконані місії;

– транзакції, тобто записи про покупку ігрових предметів, валюти та інших товарів;

– особиста інформація про гравців, зокрема, дані про вік, стать, місцезнаходження гравців;

– статус відтоку, тобто індикатор того, чи покинув гравець гру чи ні;

– модель машинного навчання, що використовується для прогнозування відтоку, її параметри та структура;

– дані для навчання, зокрема, історичні дані про попередні випадки відтоку та невідтоку, які використовуються для навчання моделі;

– результати прогнозування, тобто передбачене значення ймовірності відтоку для кожного гравця та визначення статусу його відтоку з урахуванням встановленого порога ймовірності;

– інтерфейс користувача, тобто елементи, що забезпечують взаємодію з моделлю та надають звіти про результати прогнозування;

– механізми передачі результатів прогнозування відтоку гравців до системи керування відтоком для вжиття відповідних заходів щодо утримання гравців.

Графічне подання потоків даних інформаційної моделі розглядуваної системи можна виконати за допомогою DFD-діаграми (Data Flow Diagram) [4]. Вона дозволяє наочно промоделювати, як дані переміщуються всередині системи, як вони обробляються, а також як взаємодіють різні компоненти системи.

DFD-діаграма інформаційної моделі системи «Розв’язання задачі прогнозування відтоку гравців у комп’ютерній грі» подана на рисунку 1.8. Початковою ланкою потоку даних є дослідник, який розв’язує задачу прогнозування відтоку гравців. Остаточним етапом є збір отриманих прогнозів у базу даних, яка містить звіти про ймовірності виходу з гри окремих гравців та статус цих гравців з точки зору можливості їх виходу. Перший рівень декомпозиції поданий на DFD-діаграмі на рисунку 1.9. Відповідно до неї для розв’язання задачі прогнозування відтоку гравців дослідник повинен обробити отримані дані, створити модель класифікації засобами бібліотеки Sklearn на натренувати її, виконати класифікацію гравців щодо можливості остаточного

виходу ними з гри та отримати відповідний список гравців, які потенційно можуть покинути гру.

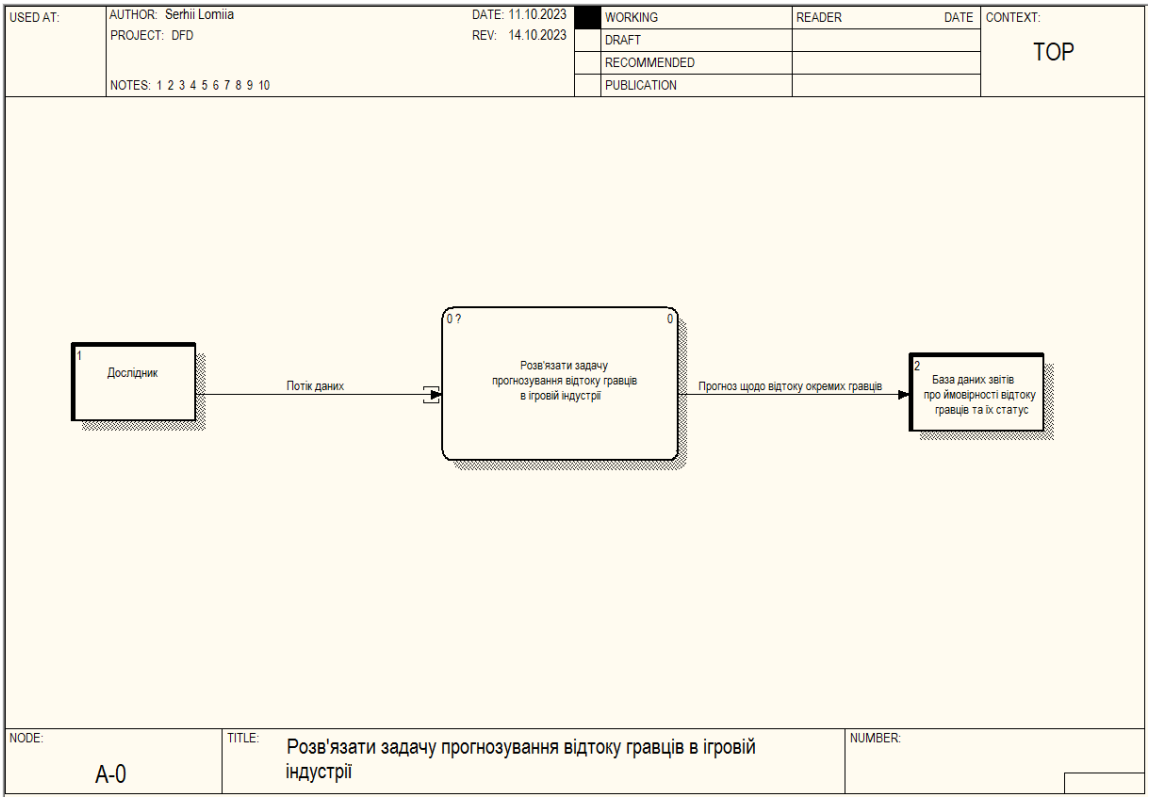

Рисунок 1.8 – DFD-діаграма

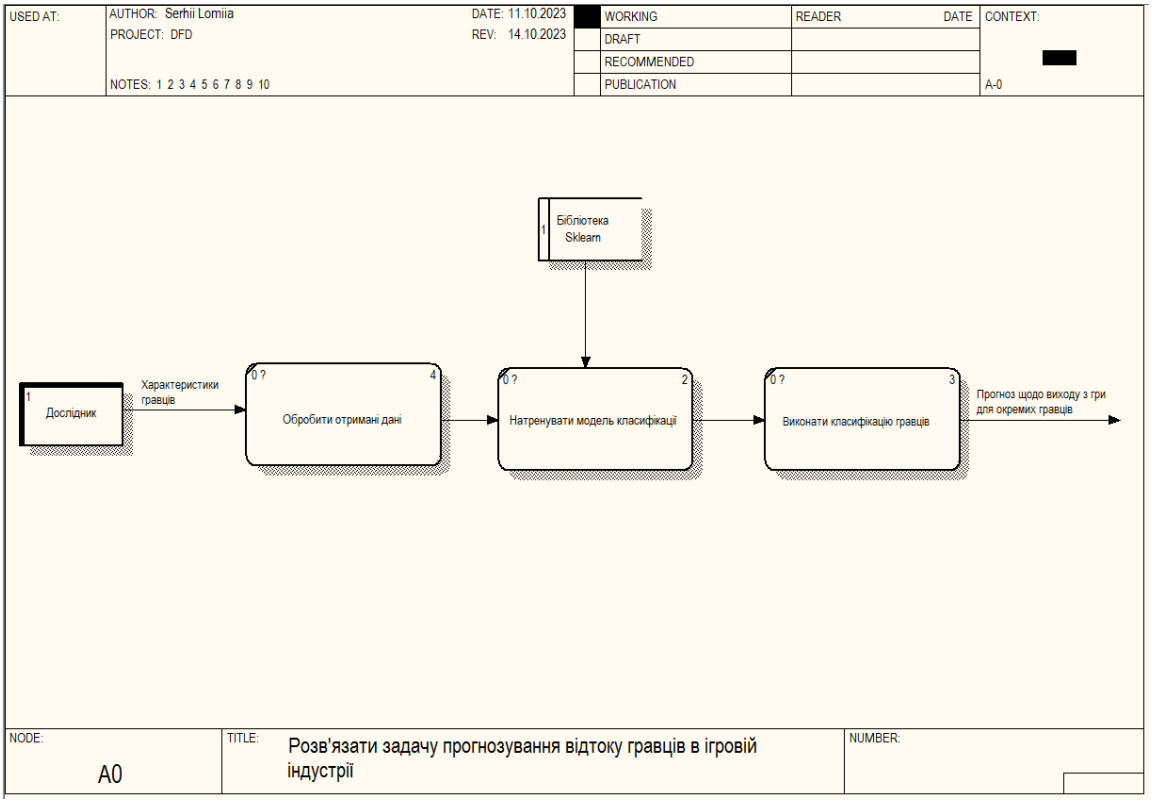

Рисунок 1.9 – DFD-діаграма 1-го рівня декомпозиції

1.2 Аналіз сценаріїв вирішення проблеми прогнозування відтоку гравців в ігровій індустрії

1.2.1 Модель аналізу проблеми

Наступним етапом системного аналізу проблеми прогнозування відтоку гравців в ігровій індустрії є вибір методу, застосування якого є найдоцільнішим для розв’язання поставленої задачі. Для того, щоб ухвалити рішення про вибір методу, визначимось з переліком методів, які принципово можуть бути використані для розв’язання задачі, та порівняємо їх за такими критеріями:

– критерій 1 (К1): простота алгоритму;

– критерій 2 (К2): чутливість алгоритму до пропусків у даних;

– критерій 3 (К3): чутливість алгоритму до мультиколінеарності у даних;

– критерій 4 (К4): здатність алгоритму працювати з розрідженими даними;

– критерій 5 (К5): час роботи програми.

Метод для розв’язання задачі будемо обирати з наступних альтернатив:

– альтернатива 1 (А1): логістична регресія;

– альтернатива 2 (А2): дерева рішень;

– альтернатива 3 (А3): градієнтний бустінг.

Ієрархічна модель вибору методу для вирішення проблеми прогнозування відтоку гравців в ігровій індустрії подана на рисунку 1.10. Фокусом ієрархії є проблема розв’язання поставленої задачі, першим рівнем ієрархії – критерії, за якими відбувається порівняння, другим рівнем – альтернативи, з яких обираємо метод розв’язання задачі.

1.2.2 Оцінювання вектора пріоритетів незадоволеностей методом аналізу ієрархій

Для аналізу ієрархії побудуємо матриці парних порівнянь моделі, а також критеріїв системи.

Матриця парних порівнянь критеріїв записана у таблиці 1.1. Останній стовпчик цієї таблиці містить результати розрахунків для вектора пріоритетів критеріїв.

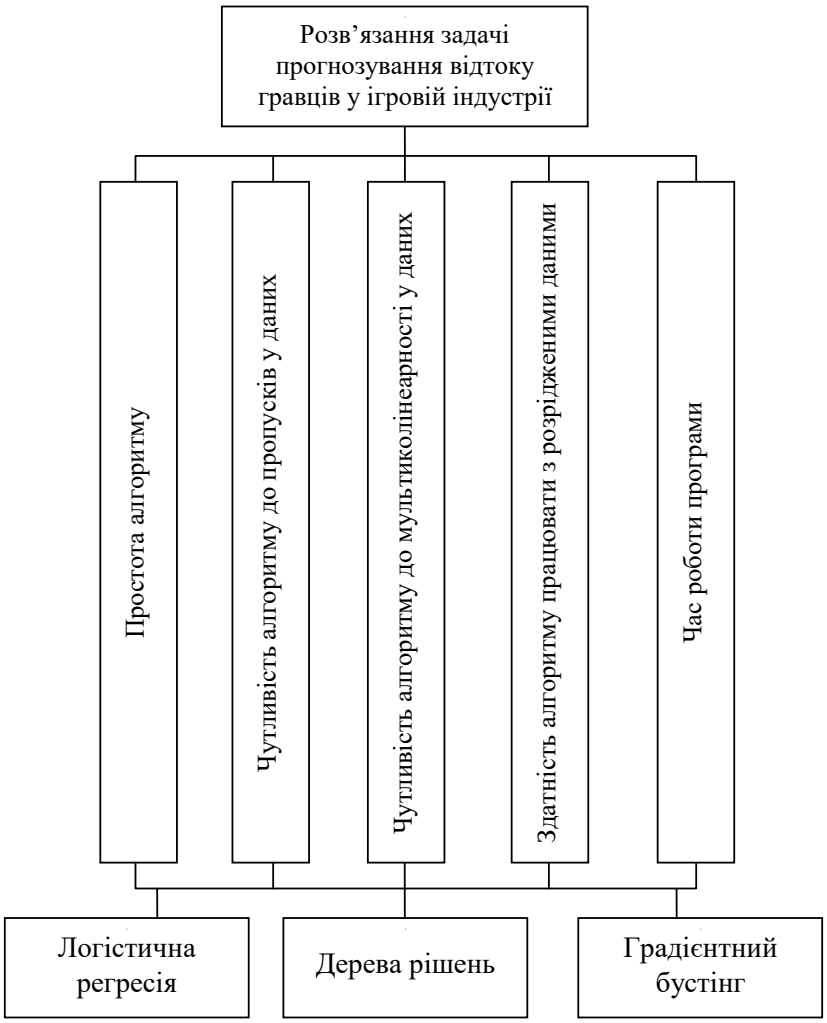

Рисунок 1.10 – Ієрархічна модель вибору методу вирішення проблеми прогнозування відтоку гравців в ігровій індустрії

Випадкова узгодженість для матриці п'ятого порядку дорівнює 1,12.

За даними таблиці 1.1:

– індекс узгодженості $ІУ = \frac{5,56 - 5}{5 - 1} \approx 0,14$;

– відносна узгодженість $ВУ = \frac{0,14}{1,12} \approx 0,12 = 12,5\%$.

Таблиця 1.1 – Матриця парних порівнянь критеріїв

Критерії оцінювання	К1	К2	К3	К4	К5	Оцінки компонентів	Вектор пріоритетів
К1	1	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{3}$	0,31	0,05
К2	5	1	2	$\frac{1}{3}$	4	1,68	0,26
К3	4	$\frac{1}{2}$	1	$\frac{1}{4}$	4	1,15	0,18
К4	6	3	4	1	2	2,70	0,42
К5	3	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	1	0,62	0,09
Усього						6,46	

Оскільки відносна узгодженість близька до 0,1, то робимо висновок, що матриця парних порівнянь критеріїв побудована правильно.

Вектор локальних пріоритетів критеріїв відносно проблеми вибору дорівнює $\vec{p}^{K} = (0,05; 0,26; 0,18; 0,42; 0,09)$.

Сформуємо матриці попарних порівнянь альтернатив за кожним критерієм (таблиці 1.2 – 1.6) та виконаємо розрахунки за ними.

За даними таблиці 1.2:

– індекс узгодженості $\text{ІУ} = \frac{3,02-3}{3-1} \approx 0,01$;

– відносна узгодженість $\text{ВУ} = \frac{0,01}{0,58} \approx 0,02 = 2\%$.

За даними таблиці 1.3:

– індекс узгодженості $\text{ІУ} = \frac{3,14-3}{3-1} \approx 0,07$;

– відносна узгодженість $\text{ВУ} = \frac{0,07}{0,58} \approx 0,12 = 12\%$.

Таблиця 1.2 – Матриця попарних порівнянь за першим критерієм

К1	А1	А2	А3	Оцінки компонентів	Вектор пріоритетів
А1	1	$\frac{1}{3}$	$\frac{1}{6}$	0,38	0,10
А2	3	1	$\frac{1}{3}$	1,00	0,25
А3	6	3	1	2,62	0,65
Усього				4,00	

Таблиця 1.3 – Матриця попарних порівнянь за другим критерієм

К2	А1	А2	А3	Оцінки компонентів	Вектор пріоритетів
А1	1	$\frac{1}{7}$	$\frac{1}{7}$	0,27	0,06
А2	7	1	$\frac{1}{3}$	1,33	0,30
А3	7	3	1	2,76	0,64
Усього				4,36	

Таблиця 1.4 – Матриця попарних порівнянь за третім критерієм

К3	А1	А2	А3	Оцінки компонентів	Вектор пріоритетів
А1	1	$\frac{1}{4}$	$\frac{1}{7}$	0,33	0,07
А2	4	1	$\frac{1}{5}$	0,93	0,20
А3	7	5	1	3,27	0,73
Усього				4,53	

Таблиця 1.5 – Порівняння за четвертим критерієм

К4	А1	А2	А3	Оцінки компонентів	Вектор пріоритетів
А1	1	$\frac{1}{4}$	$\frac{1}{5}$	0,37	0,09
А2	4	1	$\frac{1}{4}$	1,00	0,24
А3	5	4	1	2,71	0,66
Усього				4,08	

Таблиця 1.6 – Порівняння за п’ятим критерієм

К5	А1	А2	А3	Оцінки компонентів	Вектор пріоритетів
А1	1	$\frac{1}{2}$	$\frac{1}{3}$	0,55	0,15
А2	2	1	$\frac{1}{4}$	0,79	0,22
А3	3	4	1	2,29	0,63
Усього				3,63	

За даними таблиці 1.4:

– індекс узгодженості $\text{ІУ}=\frac{3,12-3}{3-1}\approx 0,06$;

– відносна узгодженість $\text{ВУ}=\frac{0,06}{0,58}\approx 0,10=10\%$.

За даними таблиці 1.5:

– індекс узгодженості $\text{ІУ}=\frac{3,15-3}{3-1}\approx 0,08$;

– відносна узгодженість $\text{ВУ}=\frac{0,08}{0,58}\approx 0,14=14\%$.

За даними таблиці 1.6:

– індекс узгодженості $\text{ІУ}=\frac{3,11-3}{3-1}\approx 0,06$;

– відносна узгодженість $\text{ВУ}=\frac{0,06}{0,58}\approx 0,10=10\%$.

1.2.3 Модель вирішення проблеми

На основі результатів п. 1.2.2 виконаємо остаточні розрахунки та зробимо за ними висновок про вибір методу вирішення проблеми прогнозування відтоку гравців в ігровій індустрії.

Результати розрахунків на цьому етапі наведені у таблиці 1.7. За даними цієї таблиці можна зробити висновок, що кращою для вирішення поставленої проблеми є третя альтернатива – градієнтний бустінг.

Таблиця 1.7 – Остаточні розрахунки

Критерій Альтернатива	К1	К2	К3	К4	К5	Узагальнені пріоритети
А1	0,10	0,06	0,07	0,09	0,15	0,09
А2	0,25	0,30	0,20	0,24	0,22	0,25
А3	0,65	0,64	0,73	0,66	0,63	0,66

1.3 Змістовна та формальна постановка задачі

1.3.1 Змістовна постановка задачі

Відтік гравців являє собою явище, коли гравці припиняють активну участь у грі або, в ширшому сенсі, перестають взаємодіяти з продуктом або послугою. Це явище є важливою проблемою для ігрової індустрії та інших

сфер, де залучення й утримання клієнтів відіграє ключову роль в успіху бізнесу. Виходячи з цього необхідно оптимізувати утримання гравців і підвищити їхню участь у грі, запобігаючи відтоку.

Задача зводиться до прогнозування ймовірності того, що конкретний гравець покине гру в найближчому майбутньому.

Використання методів машинного навчання для побудови моделей, здатних прогнозувати ймовірність відтоку для кожного гравця, може допомогти уважно ставитися до ризику відтоку і вживати заходів для утримання гравців.

1.3.2 Формальна постановка задачі

Задані X – простір об’єктів (гравці) та Y – множина відгуків, $Y = \{0,1\}$.

Кожен об’єкт $\vec{x} \in X$ описується вектором ознак: $\vec{x} = (x_1, x_2, ..., x_m)$.

Значення залежної змінної $y \in Y$: $y = 1$ – позитивний клас (гравець, що покидає гру), $y = 0$ – негативний клас (гравець, що лишається у грі).

Значення цільової змінної y відомі на об’єктах тренувальної вибірки: $X^n = (\vec{x}^{(i)}, y^{(i)})_{i=1}^n$, тобто розглядається задача навчання з вчителем.

Метою навчання є побудова алгоритму $a: X \to Y$, що прогнозує значення цільової змінної y, $y \in \{0,1\}$, для будь-якого $\vec{x} \in X$ та ймовірність p віднесення об’єкта до позитивного класу.

1.4 Постановка задач дослідження

Метою кваліфікаційної роботи є дослідження можливостей прогнозування відтоку гравців у ігровій індустрії. Виходячи з цього наведемо перелік задач, які необхідно виконати під час дослідження:

– провести огляд і аналіз сучасного стану задачі «прогнозування відтоку гравців у ігровій індустрії»;

– розв’язати задачу прогнозування відтоку гравців у ігровій індустрії обраним методом машинного навчання;

– розробити програмний продукт, який дозволить виконувати прогноз відтоку гравців;

– провести обчислювальні експерименти та виконати аналіз отриманих результатів, зокрема визначити ефективність використання обраного алгоритму при прогнозуванні.

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

2.1 Поняття відтоку гравців та фактори впливу

В ігровій індустрії відтік гравців – це втрата активних користувачів, які перестають грати в гру. Показник, що відображає відсоток втрати гравців за певний період часу, називається коефіцієнтом відтоку.

Базова формула розрахунку коефіцієнту відтоку досить проста:

$$CR = \frac{CL}{CS} \cdot 100\%,$$

де CR – коефіцієнт відтоку;

CL – кількість втрачених гравців за інтервал часу;

CS – кількість гравців на початку періоду.

Наприклад, якщо місячний коефіцієнт відтоку дорівнює 5%, то за місяць 5% гравців припиняють грати. Чим вище коефіцієнт відтоку, тим більше гравців перестають грати у гру. Тому завдання розробників полягає у тому, щоб зробити цей показник якомога нижчим.

Фактори та причини відтоку гравців можуть бути різними, але треба виділити головні з них:

– якість ігрового процесу, при якому низька якість геймплею може призвести до відтоку, оскільки гравці втрачають інтерес;

– технічні проблеми, наприклад, зависання гри чи проблеми з серверами можуть розчарувати гравців і викликати відтік;

– монотонність або відсутність контенту – коли гра не пропонує достатньо нового контенту, гравці можуть почати нудьгувати та врешті решт піти;

– соціальні особливості, при яких відсутність можливостей для спілкування та взаємодії між гравцями може бути ключовою причиною відтоку для багатьох з них;

– маркетинг та комунікація – коли неправильна маркетингова стратегія, недостатність комунікації з гравцями чи погана підтримка можуть призвести до втрати користувачів.

Треба зазначити, що не всі користувачі грають в ігри кожен день, а в багатьох жанрах можна побачити, що найбільше гравці заходять в гру тільки декілька днів на тиждень, здебільшого на вихідних. Тому важливо, що б в аналізі та прогнозуванні відтоку враховувалося, що якщо гравець не увійшов до гри протягом певного часу, наприклад 14 днів, то його можна вважати таким, що покинув гру.

На практиці краще орієнтуватися на дані своєї гри, а не брати стандартні періоди для жанру гри або конкурентів. Треба проаналізувати, після скількох днів відсутності відсоток гравців, котрі повертаються в гру, починає стрімко знижуватися, та обрати це значення для розрахунків відтоку гравців.

2.2 Підходи до прогнозування відтоку гравців

Прогнозування можливості відтоку гравців на основі різних предикторів допомагає утримувати гравців у грі і тим самим суттєво підвищує їх цінність. Результатом такого прогнозування є групи гравців, що потенційно можуть здійснити незворотний вихід з гри. Однією з цілей прогнозування є визначення особливостей поведінки, які призводять до виходу з гри, та стратегія запобігання виникнення подібних ситуацій в майбутньому.

Предиктори відтоку, також відомі як фактори або змінні, є характеристиками, параметрами або ознаками, які використовуються для прогнозування відтоку або виходу гравців з мобільної або комп’ютерної гри. Ці предиктори є ключовими компонентами у будь-якій моделі прогнозування відтоку і можуть включати різні типи даних, такі як:

– демографічні дані: вік, стать, країна, освіта, сімейний стан;

– поведінкові дані: частота використання гри, час, проведений у грі, рівень досягнень та розвитку у грі, взаємодія з іншими гравцями тощо;

– технічні дані: тип пристрою, версія операційної системи, час звернення до певних функцій гри, частота оновлень тощо;

– економічні дані: витрати грошей в грі, підписки та інші платіжні дії.

Ці предиктори аналізуються та використовуються в моделях прогнозування відтоку для ідентифікації патернів та кореляцій між ними та відтоком користувачів. Наприклад, модель може виявити, що гравці певної вікової групи з певним рівнем доходу та низьким рівнем витрат у грі мають більшу ймовірність відтоку. Такі спостереження дозволяють розробникам ігор вживати ці дані для вдосконалення стратегій утримання користувачів та запобігання відтоку.

2.3 Прогнозування відтоку гравців методами машинного навчання

2.3.1 Використання логістичної регресії для прогнозування відтоку гравців у мобільній грі

Розглянемо найбільш популярні методи прогнозування відтоку гравців в іграх. Логістична регресія є одним із найвідоміших алгоритмів машинного навчання, який відноситься до техніки навчання з вчителем.

Логістична регресія – це статистична модель, що використовується для прогнозування ймовірності належності об’єкта одному з декількох класів (для бінарної класифікації – одному з двох класів) за значеннями ознак цього об’єкта. Отже, логістична регресійна модель є лінійною моделлю бінарної класифікації, і може бути розширена на багатокласову класифікацію [5, 7].

Логістична регресія працює на даних, що є лінійно роздільними, тобто на таких даних, які можна розділити лінійною межею – гіперплощиною. Ця межа будується за наявними вихідними даними виходячи з алгоритму навчання.

Розглянемо постановку задачі логістичної регресії. Задані простір об’єктів X та множина відповідей $Y = \{0,1\}$. Кожен об’єкт описується вектором ознак $\vec{x} = (x_1, x_2, ..., x_m)$. Значеннями залежної змінної (мітки) є $y = 1$ та $y = 0$. У задачі

прогнозування відтоку гравців результат 0 або 1 є підставою для класифікації гравців на тих, котрі перестали грати, та тих, котрі ще грають. Значення цільової змінної задані на об'єктах тренувального вибірки $X^n = (\vec{x}^{(i)}, y^{(i)})_{i=1}^n$.

У ході навчання необхідно побудувати алгоритм $a: X \to Y$, який прогнозуватиме значення цільової змінної y для будь-якого об'єкта $\vec{x} \in X$.

Позначимо через $p_+ = \mathrm{P}(y=1 \mid \vec{x}) \in [0;1]$ – ймовірність того, що об'єкт $\vec{x}$ належить до класу $y=1$. Відношенням шансів називається відношення ймовірності настання цієї події до ймовірності її ненастання:

$$OR = \frac{p_+}{1-p_+}.$$

Логарифм цієї величини дорівнює:

$$\ln OR = \ln \frac{p_+}{1-p_+}.$$

Остання функція перетворює значення ймовірності p_+ у значення $\ln OR \in \mathbb{R}$, тому цей факт можна використати для встановлення лінійного зв'язку між ознаками та логарифмом відношення шансів:

$$\ln\left(\frac{p_+}{1-p_+}\right) = \vec{\omega}^T \vec{x},$$

де $\vec{\omega} = (\omega_0, \omega_1, ..., \omega_m)$ – вектор параметрів моделі;

$\vec{x} = (1, x_1, ..., x_m)$ – розширений вектор ознак, $x_0 = 1$.

Послідовність дій, які дозволяють спрогнозувати мітку класу об'єкта під час розв'язання задачі класифікації за допомогою логістичної регресії, полягає у наступному.

Крок 1. Обчислити $\omega_0 + \omega_1 x_1 + \omega_2 x_2 + ... + \omega_m x_m = \vec{\omega}^T \vec{x} \in \mathbb{R}$.

Крок 2. Обчислити відношення шансів $OR_+ = e^{\vec{\omega}^T \vec{x}}$ належності до класу 1.

Крок 3. Обчислити ймовірність належності до класу 1:

$$\hat{p}_+ = \frac{OR_+}{1+OR_+} = \frac{e^{\vec{\omega}^T \vec{x}}}{1+e^{\vec{\omega}^T \vec{x}}} = \frac{1}{1+e^{-\vec{\omega}^T \vec{x}}}.$$

Права частина останньої рівності задається функцією $\sigma(t) = \frac{1}{1+e^{-t}}$, яка називається сигмоїдою.

Крок 4. Класифікація об'єкта. Після обчислення ймовірності віднесення зразка до позитивного класу, необхідно виконати його класифікацію, тобто визначити мітку класу цього об'єкта:

$$\hat{y} = \begin{cases} 1, & \text{якщо } \sigma(t) \geq 0,5; \\ 0, & \text{якщо } \sigma(t) < 0,5. \end{cases}$$

Значення параметрів моделі логістичної регресії $\omega_0, \omega_1, ..., \omega_m$ визначаються у ході її навчання. Метою навчання є визначення таких значень вектора параметрів $\vec{\omega}$, за яких модель прогнозуватиме високі ймовірності належності до класу 1 для зразків, що дійсно до нього належать та низькі ймовірності належності до класу 1 для зразків, які дійсно належать до іншого класу.

Для навчання використовують метод максимальної правдоподібності. Функція правдоподібності має вигляд:

$$L(\vec{\omega}) = \mathrm{P}(y \mid \vec{x}, \vec{\omega}) = \prod_{i=1}^{n} \mathrm{P}(y^{(i)} \mid \vec{x}^{(i)}, \vec{\omega}) = \prod_{i=1}^{n} \left(\sigma(\vec{\omega}^T \vec{x}^{(i)})\right)^{y^{(i)}} \left(1 - \sigma(\vec{\omega}^T \vec{x}^{(i)})\right)^{1-y^{(i)}},$$

але на практиці зручніше використовувати не її, а логарифмічну функцію правдоподібності:

$$l(\vec{\omega}) = \ln L(\vec{\omega}) = \sum_{i=1}^{n} \Big[y^{(i)} \ln\left(\sigma(\vec{\omega}^T \vec{x}^{(i)})\right) + (1 - y^{(i)}) \ln\left(1 - \sigma(\vec{\omega}^T \vec{x}^{(i)})\right) \Big].$$

Задача оптимізації логарифмічної функції правдоподібності має вигляд:

$$l(\vec{\omega}) = \sum_{i=1}^{n} \Big[y^{(i)} \ln\left(\sigma(\vec{\omega}^T \vec{x}^{(i)})\right) + (1 - y^{(i)}) \ln\left(1 - \sigma(\vec{\omega}^T \vec{x}^{(i)})\right) \Big] \to \max_{\vec{\omega}}.$$

Її розв'язок можна знайти методом градієнтного спуску. Для реалізації методу градієнтного спуску обчислимо частинні похідні:

$$\frac{\partial l(\vec{\omega})}{\partial \omega_j} = \sum_{i=1}^{n} \left(\frac{y^{(i)}}{\sigma(\vec{\omega}^T \vec{x}^{(i)})} - \frac{1 - y^{(i)}}{1 - \sigma(\vec{\omega}^T \vec{x}^{(i)})} \right) \cdot \sigma(\vec{\omega}^T \vec{x}^{(i)}) \cdot (1 - \sigma(\vec{\omega}^T \vec{x}^{(i)})) \cdot \left. \frac{\partial (\vec{\omega}^T \vec{x})}{\partial \omega_j} \right|_{(x^{(i)}, y^{(i)})} =$$

$$= \sum_{i=1}^{n} \left(\frac{y^{(i)} - \sigma(\vec{\omega}^T \vec{x}^{(i)})}{\sigma(\vec{\omega}^T \vec{x}^{(i)})(1 - \sigma(\vec{\omega}^T \vec{x}^{(i)}))} \right) \cdot \sigma(\vec{\omega}^T \vec{x}^{(i)}) \cdot (1 - \sigma(\vec{\omega}^T \vec{x}^{(i)})) \cdot \left. \frac{\partial (\vec{\omega}^T \vec{x})}{\partial \omega_j} \right|_{(x^{(i)}, y^{(i)})} =$$

$$= \left| \omega_0 + \sum_{j=1}^{m} \omega_j x_j \right| = \sum_{i=1}^{n} \left(y^{(i)} - \sigma(\vec{\omega}^T \vec{x}^{(i)}) \right) \cdot x_j^{(i)}, \quad j = \overline{0, m}.$$

Відповідно до алгоритму методу градієнтного спуску оновлення ваг здійснюється за формулою:

$$\omega_j = \omega_j + \eta \frac{\partial l(\vec{\omega})}{\partial \omega_j} = \omega_j + \eta \sum_{i=1}^{n} (y^{(i)} - \sigma(\vec{\omega}^T \vec{x}^{(i)})) x_j^{(i)}, \quad j = \overline{0, m}.$$

Оскільки всі ваги оновлюються одночасно, правило оновлення можна записати у вигляді:

$$\vec{\omega} = \vec{\omega} + \Delta\vec{\omega},$$

де $\Delta\vec{\omega} = \eta \cdot \nabla l(\vec{\omega})$,

η – коефіцієнт навчання.

Функція вартості або крос-ентропійна функція логістичної регресії може бути побудована за функцією логарифмічної правдоподібності і дорівнює:

$$J(\vec{\omega}) = -l(\vec{\omega}) = -\sum_{i=1}^{n}\left[y^{(i)} \ln\left(\sigma(\vec{\omega}^T \vec{x}^{(i)})\right) + (1 - y^{(i)}) \ln\left(1 - \sigma(\vec{\omega}^T \vec{x}^{(i)})\right)\right].$$

Очевидно, що вищезнайдене правило для оновлення ваг $\vec{\omega}$ може бути також отримане шляхом розв’язання задачі мінімізації функції витрат:

$$J(\vec{\omega}) = -\sum_{i=1}^{n}\left[y^{(i)} \ln\left(\sigma(\vec{\omega}^T \vec{x}^{(i)})\right) + (1 - y^{(i)}) \ln\left(1 - \sigma(\vec{\omega}^T \vec{x}^{(i)})\right)\right] \to \min_{\vec{\omega}}.$$

Перевагами логістичної регресії є простота інтерпретації результатів, швидкість тренування та використання моделі. головним недоліком моделі є її неефективність при моделюванні складних нелінійних залежностей.

2.3.2 Використання дерева рішень для прогнозування відтоку гравців у мобільній грі

Далі розглянемо інший метод класифікації – дерева прийняття рішень. Дерева прийняття рішень є універсальним алгоритмом розв’язання задач класифікації (у тому числі багатокласової) і регресії, здатним добре працювати на складних наборах даних. Дерево рішень (decision tree) – схема класифікації об’єкта з допомогою ієрархічної послідовності правил виду «Якщо …, то…»,

під час відповіді на які дані розбиваються на підмножини, які потім знову поділяються на ще менші підмножини і так далі, доки алгоритм не визначить, що дані в підмножинах досить однорідні, або виконана інша умова зупинки побудови дерева рішень [5, 6].

Правила генеруються автоматично в процесі навчання за рахунок узагальнення навчальних прикладів. Для зразків навчальної вибірки має бути задано цільове значення, тому дерева рішень є задачею навчання з учителем. Аналізуючи ознаки об'єктів у навчальному наборі модель навчається робити висновки щодо класу цих об'єктів.

Дерева прийняття рішень є складовими випадкових лісів – одного з найпотужніших алгоритмів машинного навчання, доступних на сьогоднішній день.

Дерево рішень подає вирішальні правила у певній ієрархії, і включає елементи двох типів – вузли і листи. Кожен внутрішній вузол (не лист) відповідає одній з вхідних змінних. Початковий вузол дерева рішень називається кореневим вузлом. У вузлах здійснюється перевірка відповідності об'єктів встановленим у цих вузлах правилам. Кожне ребро, що виходить із внутрішнього вузла, представляє одну з можливих відповідей на питання, асоційоване з цим вузлом. Таким чином, множина зразків, що потрапили у вузол, розбивається на підмножини залежно від відповіді на відповідне цьому вузлу запитання. Далі до кожної підмножини знову застосовується правило і процедура повторюється, доки не буде виконана умова зупинки алгоритму. В останньому вузлі гілки запитання не ставляться, такий вузол називається листом. Він містить підмножину об'єктів, що задовольняють всім правилам цієї гілки. Тобто лист є прогнозним значенням цільової змінної. До кожного листа є лише один шлях по дереву.

Найпростішим випадком дерев рішень є бінарні дерева, у вузлах яких розгалуження може вестись лише у двох напрямах, тобто є лише дві відповіді на поставлене запитання – «так» чи «ні» (рис. 2.1). У загальному випадку гілок, що виходять із внутрішнього вузла дерева, може бути більше двох.

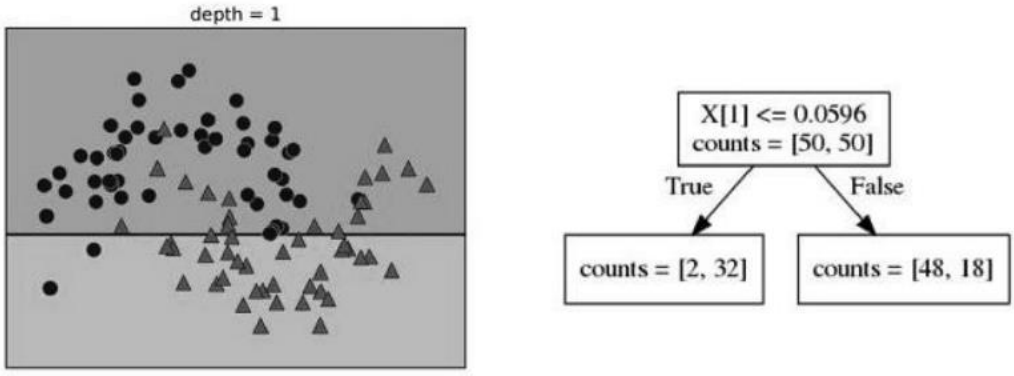

Рисунок 2.1 – Границя прийняття рішення, отримана за допомогою дерева глибиною 1 (ліворуч) і відповідне дерево рішень (праворуч)

Основним завданням при побудові дерева рішень є послідовне розбиття навчальної множини на підмножини доти, поки всі вузли в кінці гілок не стануть листами. Це може відбутись у одному з двох випадків: або коли вузол містить єдиний об’єкт чи об’єкти тільки одного класу або коли досягнута задана умови зупинки алгоритму [5].

Алгоритми побудови дерева рішень для організації чергового вузла обирають змінну, яка розбиває множину спостережень у вузлі так, щоб результуючі підмножини містили об’єкти лише з одного класу або кількість об’єктів з інших класів в кожній з цих множин була якомога меншою. Найбільш відомими критеріями, які дозволяють обрати черговий атрибут для розбиття, є міра ентропії та індекс Gini.

Міра інформаційного виграшу або міра ентропії для системи з n можливими класами визначається за формулою:

$$S(t) = -\sum_{i=1}^{n} p(i \mid t) \log_2 p(i \mid t),$$

де $p(i \mid t)$ – частка зразків, що належать до класу i для окремого вузла t (для непорожніх класів, $p(i \mid t) \neq 0$), $\sum p(i \mid t) = 1$.

Ентропія дозволяє оцінити ступінь неоднорідності підмножини за тими класами, з яких вона складається: чим вище ентропія, тим менш упорядкована

система. Коли класи у підмножині подані у рівних частках, ентропія максимальна. Якщо всі приклади у вузлі відносяться до одного класу, ентропія дорівнює нулю. за таких умов найкращим атрибутом розбиття є той, який забезпечить максимальне зниження ентропії результуючої підмножини щодо породної множини.

Невизначеність Джині обчислюється за формулою:

$$G(t) = 1 - \sum_{i=1}^{n} p(i \mid t)^2 ,$$

що дозволяє оцінити ймовірність помилкової класифікації. Міра невизначеності Джині максимальна, якщо класи у вузлі повністю перемішані. Мінімальне значення 0 має місце, якщо всі об’єкти у підмножині відносяться до одного класу. Найкращим буде таке розбиття, для якого значення індексу Джині мінімальне.

На практиці невизначеність Джині і приріст інформації призводять до схожих дерев, однак міру Джині обчислювати швидше.

Зменшення ентропії називають прирощенням інформації. Формально прирощення інформації при розбитті вибірки за ознакою A визначається як

$$IG(A) = S_0 - \sum_{i=1}^{l} \frac{N_i}{N} S_i ,$$

де l – кількість груп після розбиття;

N_i – кількість елементів вибірки, для яких ознака A набуває i-е значення;

S_i – ентропія у i-му вузлі.

Аналогічно обчислюється прирощення інформації на основі невизначеності Джині.

Теоретично алгоритм навчання дерева рішень працює до тих пір, доки не будуть отримані підмножини, у кожній з яких міститимуться об’єкти лише одного

класу. При цьому побудоване дерево може у кожному листі містити єдиний об’єкт, тобто буде перенавченим. Тому дерево зазвичай підрізується шляхом встановлення максимальної глибини. Дерево має враховувати інформацію з досліджуваного набору даних, але одночасно воно має бути досить простим.

Для побудови дерева оптимального розміру використовуються дві стратегії: нарощування дерева до певного розміру відповідно до заданих параметрів та скорочення дерева шляхом відсікання гілок.

Використовуються такі правила зупинки під час нарощування дерева:

– рання зупинка – алгоритм зупиняється після досягнення заданого значення частки правильно розпізнаних об’єктів;

– обмеження глибини дерева – завдання максимальної кількості розбиттів у гілках, після досягнення якої навчання зупиняється;

– завдання мінімально допустимої кількості об’єктів у вузлі – встановлюється обмеження на створення вузлів із кількістю об’єктів менше заданої, що дозволить уникнути створення малозначущих правил.

Альтернативним підходом є метод відсікання гілок, який реалізується у такій послідовності:

а) побудувати повне дерево, в якому листи містять об’єкти одного класу;

б) визначити відносну точність моделі – відношення кількості правильно розпізнаних об’єктів до загальної кількості навчальних об’єктів, та абсолютну помилку – кількість неправильно класифікованих об’єктів;

в) видалити з дерева листи і вузли, відсікання яких не призведе до значного зменшення точності моделі або до збільшення помилки.

Перевагою відсікання гілок у порівнянні з ранньою зупинкою є можливість пошуку оптимального співвідношення між точністю та зрозумілістю дерева. Недоліком є більший час навчання через необхідність спочатку збудувати повне дерево.

Отже, робота дерева прийняття рішень базується на наступних кроках:

– введення даних: передобробка та представлення даних в виді ознак і цільової змінної (відтік чи не відтік);

– розбиття даних: дерево починається з кореневого вузла в якому здійснюється розбиття даних на основі певної ознаки та її значення;

– побудова дерева: рекурсивно продовжується розбиття даних в кожному вузлі дерева за різними ознаками поки не буде досягнуто критерій зупинки;

– прийняття рішень: коли дерево збудовано, гравці можуть бути класифіковані шляхом проходження через дерево від кореневого вузла до листового вузла, де приймається кінцеве рішення щодо відтоку.

До недоліків використання даного методу можна віднести недостатню узагальнююча здатність, коли дерево стає складним.

2.3.3 Використання градієнтного бустінгу для прогнозування відтоку гравців у мобільній грі

Градієнтний бустінг – це ансамблевий алгоритм. Ансамблем називається набір класифікаторів, окремі відповіді яких узагальнюються для отримання остаточного прогнозу. Причина використання ансамблів полягає в тому, що кілька класифікаторів, які намагаються оцінити одну й ту ж змінну, за виконання певних умов разом можуть дати кращий результат, ніж кожен окремий з них. Існує декілька підходів до організації ансамблів, зокрема це бегінг (bagging) та бустінг (boosting) [8].

Розглянемо детальніше ансамблеві методи для більш детального розуміння принципу градієнтного бустінгу.

Ансамблеві методи (ensemble learning) використовують декілька навчальних алгоритмів з метою підвищення точності прогнозування, порівняно з кожним навчальним алгоритмом окремо [5, 9].

Для створення ансамблю класифікаторів використовуються такі підходи:

– мажоритарне голосування (majority voting) у бінарній класифікації, коли результатом класифікації є мітка того класу, який був визначений більшістю класифікаторів;

– плюралістичне голосування (plurality voting) в багатокласовій класифікації, коли результатом класифікації є мітка того класу, що отримав максимум голосів окремих класифікаторів.

Розглянемо, як відбувається навчання ансамблевих методів. За допомогою тренувального набору даних натренуємо m різних класифікаторів $C_1, ..., C_m$. При використанні мажоритарного або плюралістичного підходу міткою $\hat{y}$, яку прогнозуватиме ансамбль методів, буде мітка того класу, яка отримала максимум голосів зазначених класифікаторів $C_1, ..., C_m$:

$$\hat{y} = \text{moda}\{C_1(x), C_2(x), ..., C_m(x)\}.$$

Такий мажоритарний класифікатор називається класифікатором із жорстким голосуванням (hard voting classifier).

Зокрема, у задачі двокласової класифікації прогноз на основі мажоритарного голосування має вигляд:

$$\hat{y} = C(x) = \text{sign}\left[\sum_{j=1}^{m} C_j(x)\right] = \begin{cases} 1, & \sum_i C_i(x) \geq 0, \\ -1, & \sum_i C_i(x) < 0. \end{cases}$$

Навіть коли кожен класифікатор є слабким учнем (weak learner), тобто його точність ненабагато краще за випадкове вгадування, то ансамбль все одно може бути сильним учнем (strong learner) і забезпечуватиме високу точність за умови, що є достатня кількість слабких учнів і вони досить різноманітні.

Ансамблеві методи працюють краще, коли окремі учні є якомога більш незалежними один від одного. Один із способів отримати неоднорідні класифікатори полягає в тому, щоб навчати їх із застосуванням різних алгоритмів. Це підвищує ймовірність того, що помилки, які вони будуть робити, будуть різних типів, і, у свою чергу, сприятиме підвищенню якості ансамблю.

Якщо всі класифікатори можуть оцінювати ймовірність класів (тобто мають метод predict_proba), то остаточний прогноз класу можна зробити за найвищою серед усереднених ймовірностей класів за всіма індивідуальними класифікаторами. Це називається м'яким голосуванням (soft voting). Такий підхід часто є більш ефективним, ніж жорстке голосування, тому що призначає більшу вагу голосам з високою довірою.

Розглянемо далі один із способів організації ансамблів – бустінг. Бустінг (boosting, улучшение) – процедура послідовної побудови ансамблю алгоритмів машинного навчання, у якій кожен наступний алгоритм намагається компенсувати недоліки композиції всіх попередніх алгоритмів.

У бустінгу ансамбль складається з дуже простих базових класифікаторів (слабких учнів), які прогнозують ненабагато краще випадкового вгадування. Прикладом слабкого учня є однорівневе дерево рішень, тобто дерево рішень з єдиним внутрішнім вузлом – коренем, який безпосередньо пов'язаний з кінцевими вузлами (його листами).

Бустінг дозволяє зосередити увагу на тренувальних зразках, які важко класифікувати, тобто з метою покращення якості ансамблю слабким учням дається можливість навчатися на помилково класифікованих зразках.

Алгоритм бустінгу у своїй початковій постановці використовує випадкові підмножини тренувальних зразків, вилучені з набору тренувальних даних без повернення.

Процедура бустінгу складається з таких кроків [5, 9]:

а) вилучити випадкову підмножину зразків d_1 без повернення з тренувального набору для тренування слабкого учня C_1;

б) вилучити другу випадкову тренувальну підмножину d_2 без повернення з тренувального набору і додати 50% раніше помилково класифікованих зразків для тренування слабкого учня C_2;

в) знайти в тренувальному наборі тренувальні зразки d_3, за якими C_1 і C_2 не співпадають, для тренування третього слабкого учня C_3;

г) об’єднати слабких учнів C_1, C_2 та C_3 за допомогою мажоритарного голосування.

Сьогодні доступно багато популярних методів бустінгу: AdaBoost (Adaptive Boosting – адаптивний бустінг), Gradient Boosting (градієнтний бустінг), CatBoost, Light GBM і XGBoost.

Алгоритм градієнтного бустінгу найпростіше пояснити, спочатку представивши алгоритм адаптивного бустінгу (AdaBoost). Даний алгоритм для тренування слабких учнів використовує повний тренувальний набір, де зразки зважуються повторно на кожній ітерації з метою побудови сильного класифікатора, який навчається на помилках попередніх слабких учнів в ансамблі.

Отже, бустінг полягає у багаторазовому використанні слабких алгоритмів навчання на зважених по-різному версіях навчальних даних. Ваги на кожному раунді алгоритму залежать від точності попередніх класифікаторів, що дозволяє алгоритму зосередити свою увагу тих зразках, які досі неправильно класифіковані.

Розглянемо приклад ансамблю AdaBoost з трьох слабких учнів для двокласової класифікації. Перший раунд представляє тренувальний набір, де всім тренувальним зразкам призначені рівні ваги. На основі цього набору тренуємо пеньок рішення, який намагається класифікувати зразки. У наступному раунді призначаємо більшу вагу раніше помилково класифікованим зразкам і знижуємо вагу правильно класифікованих зразків. Наступний пеньок рішення буде більш сфокусований на тренувальних зразках з більшими вагами, тобто на тренувальних зразках, які ймовірно важко класифікувати. Якщо цей слабкий учень помилково класифікує деякі зразки, то у наступному раунді їм призначаються більші ваги. Оскільки ансамбль AdaBoost складається з трьох раундів бустінгу, об'єднуємо трьох слабких учнів, натренованих на різнозважених тренувальних підмножинах, шляхом зваженого мажоритарного голосування.

Таким чином, на кожній ітерації нова модель фокусується на тих прикладах, які були важкими для класифікації попередніми моделями.

Градієнтний бустінг (Gradient Boosting) – це потужний метод машинного навчання, що передбачає комбінування слабких учнів (зазвичай дерев рішень, але можуть використовуватись і інші методи) для отримання більш точної і сильної моделі. Основна відмінність градієнтного бустінгу полягає у способі коригування помилок. Навчання починається з простої моделі, а потім для кожної нової моделі обчислюється градієнт функції втрат щодо прогнозів поточного ансамблю. Новий слабкий учень навчається так, щоб мінімізувати функцію втрат. Нова модель додається до ансамблю, зменшуючи залишкові помилки.

Отже, градієнтний бустінг також працює за рахунок послідовної побудови моделей, але використовує градієнтні методи для ефективного коригування помилок попередніх моделей.

До переваг методу градієнтного бустінгу відносять високу точність прогнозування за рахунок послідовного уточнення моделей.

Висновки за розділом 2

У даному розділі розглянуто поняття відтоку гравців у ігровій індустрії та основні фактори впливу на цей процес. Формально встановлення факту, чи піде гравець із гри чи залишиться у ній, є задачею бінарної класифікації.

Основної уваги приділено розгляду найпопулярніших методів машинного навчання, які використовуються для розв’язання задачі класифікації, зокрема, логістичну регресію, дерева рішень та ансамблеві методи, такі як бустінг. Для кожного методу наведено короткі теоретичні відомості, алгоритм використання, переваги та недоліки.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Python як інструмент прогнозування в машинному навчанні

Python, як потужний та універсальний мова програмування, стає невід'ємною частиною аналітичних та прогностичних завдань у багатьох галузях. У сфері геймінгу та розваг важливою є задача прогнозування відтоку гравців, щоб ефективно управляти ресурсами та покращувати користувацький досвід [10, 11].

Python відомий своєю простотою та зручністю в використанні. Це високорівнева мова програмування, що дозволяє лаконічно виражати ідеї та швидко реалізовувати програми. Це особливо важливо в області аналізу даних, де розробники часто мають справу з великими обсягами інформації.

Python має обширну систему бібліотек для наукових обчислень та машинного навчання. Бібліотеки такі як NumPy, Pandas, Matplotlib та Scikit-learn дозволяють легко виконувати завдання з обробки та аналізу даних, що є важливим етапом у вирішенні задачі прогнозування відтоку гравців.

Для розв'язання поставленої задачі також використовується XGBoost – бібліотека машинного навчання, яка широко застосовується у завданнях класифікації та регресії. Її ефективність полягає у використанні градієнтного бустінгу та оптимізованому виборі параметрів для покращення точності прогнозувань.

Python та XGBoost в поєднанні створюють потужний інструмент для прогнозування відтоку гравців, що дозволяє гейм-розробникам ефективно вдосконалювати свої продукти та забезпечувати задоволення користувачів.

3.2 Алгоритм розв'язання задачі

Алгоритм розв'язання задачі прогнозування відтоку гравців методами машинного навчання передбачає виконання наступних дій.

Етап 1. Збір та підготовка даних:

– збір та попередній аналіз даних;

– розподіл даних на навчальну, валідаційну та тестову вибірки;

– передобробка даних (обробка пропущених значень та викидів, кодування категоріальних ознак, масштабування числових ознак, аналіз кореляції між ознаками).

Етап 2. Вибір моделі:

– вибір алгоритму машинного навчання та визначення гіперпараметрів моделі;

– розробка базової моделі для подальшого вдосконалення.

Етап 3. Навчання та оцінювання якості моделі:

– навчання моделі на навчальному наборі даних;

– оцінка якості моделі на валідаційній вибірці;

– налаштування гіперпараметрів для оптимізації моделі;

– оцінка якості навченої моделі на тестовій вибірці;

– аналіз метрик.

Етап 4. Впровадження моделі:

– отримання прогнозів щодо відтоку гравців на нових даних.

3.3 Опис програми

У ході виконання кваліфікаційної роботи було засобами Python та додаткових бібліотек було розроблено програмний продукт, за допомогою якого можна визначати ймовірність відтоку гравців і мобільній грі.

Першочергово було обрано ряд даних по користувачам мобільної гри в жанрі Fantasy Collection RPG та проведений попередній аналіз, що допомогло зрозуміти, які дані потрібні для реалізації прогнозування відтоку гравців.

Наступним кроком було обрання цільової групи гравців, в даному випадку обрано гравців які досягли як мінімум level 30 у грі.

Реалізація алгоритму прогнозування відтоку гравців відбулася за допомогою використання бібліотек Python.

На першому етапі після преданалізу та обрання необхідної вибірки для прогнозування ми розбиваємо наші дані на тренувальну, тестову та валідаційну вибірки.

Далі використовуємо Optuna – бібліотеку для оптимізації гіперпараметрів в машинному навчанні. Вона надає зручний інтерфейс для автоматизації та оптимізації вибору параметрів моделей, що дозволяє знайти найкращі конфігурації гіперпараметрів для покращення ефективності моделі.

Наступним етапом є перевірка обчислення ваг класів у балансованому наборі даних, що буде корисним для моделей класифікації, особливо у випадку незбалансованих класів як в нас, адже клас гравців, котрі не відвалюються (клас 0), домінує над класом гравців котрі перестають грати (клас 1). Для цього ми використовуємо бібліотеку Sklearn модуль class_weight.

Після цих етапів отримаємо ваги класів та набір гіперпараметрів для оптимальної конфігурації моделі, а саме кількість дерев у градієнтному бустінговому алгоритмі, а також параметр, що контролює обрізання дерев.

Далі використовуючи бібліотеку xGBoost разом з отриманими параметрами на минулих етапах проводимо моделювання та оцінюємо результати на тестовій і валідаційній множинах даних. Більшої уваги ми приділяємо оцінюванню саме AUC-PR ніж AUC, оскільки в нашому випадку важлива точність перед незбалансованістю. Також аналізуємо матрицю невідповідності (confusion matrix) та вагу кожної з ознак (використовуємо Shap values) для розуміння найбільш значущих ознак моделі.

Програма зручна у використанні завдяки своїй чіткій структурі та наявності засобів візуалізації результатів для полегшення їх сприйняття та аналізу. Результати роботи програми можуть бути використані для прогнозування відтоку гравців в іграх, що допоможе зберегти гравців якнайдовше в грі та підвищити дохід самої гри.

Висновки за розділом 3

В розділі 3 розглянуто особливості мови програмування Python, яка була використана для розробки програмного продукту, що дозволяє прогнозувати ймовірність відтоку гравців комп’ютерної гри. Вибір Python як мови реалізації завдання пояснюється його потужними можливостями у розробці проєктів машинного навчання, наявністю різноманітних бібліотек, які дозволяють більш якісно реалізовувати поставлені завдання та зосередитись на аналіз отриманих результатів замість самостійного програмування типових процедур та методів.

Окремої уваги було приділено опису основних етапів програмної реалізації поставленої задачі та розгляду етапів роботи готового програмного продукту. Програма є зручною у використанні та може знайти практичне застосування.

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ ТА ЇХ АНАЛІЗ

Маємо набір даних по гравцям за один з кварталів 2023 року по деяким країнам. Набір даних складається з 19882 рядків (тобто маємо стільки ж гравців, як і кількість рядків) та 165 стовпців (які є показниками гри кожного з гравця за останні 7, 14, 30 та 60 днів). Багато даних має пропуски, що означає, що даний гравець не має дій в тій чи іншій активності. Кожний стовпчик є певною ознакою (feature) гравця. Фрагмент набору даних наведений на рисунку 4.1.

	DayInARow	DaysInGame	EndLevel	RevenueD60	RevenueD30	RevenueD14	RevenueD7	DepositsD60	DepositsD30	DepositsD14	...	Battle_Hydra_BrutalD14	Battle_Hydra_HardD14	Battle_Hyd
0	84.0	1752.0	100.0	1056.755005	465.25	98.00	28.00	35.0	18.0	2.0	...	1.0	0.0	
1	3.0	1731.0	88.0	671.846008	231.00	0.00	0.00	22.0	6.0	0.0	...	0.0	1.0	
2	1711.0	1710.0	100.0	411.383789	90.50	68.25	68.25	11.0	6.0	3.0	...	8.0	2.0	
3	293.0	1710.0	100.0	2207.775146	1432.00	661.00	418.50	36.0	21.0	11.0	...	11.0	14.0	
4	52.0	1706.0	100.0	1127.777222	915.00	735.00	572.00	27.0	16.0	13.0	...	2.0	2.0	

Рисунок 4.1 – Фрагмент набору даних

Попередній аналіз даних свідчить про те що набір даних є незбалансованим, адже лише 11,2% гравців мають статус відвалу (рисунок 4.2).

```
AboutToChurn
False    17658
True      2224
Name: userid, dtype: int64
AboutToChurn
False    0.88814
True     0.11186
Name: userid, dtype: float64
```

Рисунок 4.2 – Співвідношення класів відвалу гравців

Розподіл даних на тестувальну, валідаційну та тренувальну вибірки відбувається випадковим чином, різні вибірки будуть відрізнятися розміром. Надалі будемо використовувати на різних етапах спочатку тренувальний набір, потім валідаційний та тестовий, щоб переконатися в коректності та стійкій точності моделі прогнозування.

```
-- TRAIN --
(9769, 164)
(9769,)
-- VALID --
(4188, 164)
(4188,)
-- TEST --
(3490, 164)
(3490,)
```

Рисунок 4.3 – Розміри вибірок після розподу

На наступному етапі налаштовуємо гіперпараметри, знаходимо ваги класів для моделі засобами бібліотек Optuna та Sklearn. Результати процесу налаштування гіперпараметрів наведені на рисунку 4.4.

```
Number of finished trials: 50
Best trial:
  Value: 0.7193670356600261
  Params:
    n_estimators: 410
    eta: 0.03570281219889413
    gamma: 0.17
    reg_lambda: 1.67
    early_stopping_rounds: 18
{'n_estimators': 410, 'eta': 0.03570281219889413, 'gamma': 0.17, 'reg_lambda': 1.67, 'early_stopping_rounds': 18}
```

Рисунок 4.4 – Результати процесу гіперпараметризації,
де було проведено 50 експериментів

Пояснимо значення параметрів, виведені на рисунку 4.4:

– n_estimators: кількість дерев у градієнтному алгоритмі;

– eta: швидкість навчання (learning rate);

– gamma: параметр, що контролює обрізання дерев;

– reg_lambda: коефіцієнт регуляризації для обмеження ваг дерев;

– early_stopping_rounds: кількість раундів без покращень, після яких навчання буде припинене.

Отже, отримано найкращий набір гіперпараметрів, який слід використовувати для побудови моделі з найвищою очікуваною ефективністю на основі визначеної цільової метрики.

Далі оцінимо отримані налаштування. Побудуємо графік порівняння AUC-PR між тренувальною вибіркою та валідаційною (рисунок 4.5).

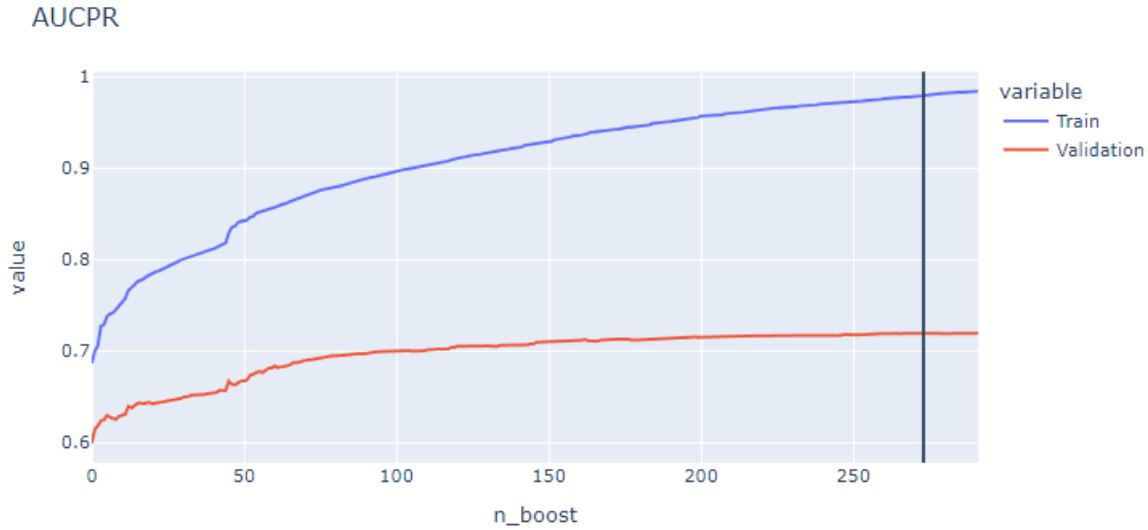

Рисунок 4.5 – Порівняння метрики AUC-PR тренувального та валідаційного набору даних

Наше головне завдання мінімізувати False Positive але при цьому мати не високе значення False Negative, тобто знайти найкращий баланс між цими показниками. AUC-PR вимірює площу під кривою Precision-Recall, яка відображає відношення точності (Precision) до чутливості (Recall) для різних порогів класифікації. AUC-PR може надати більш чутливу оцінку ефективності моделі, оскільки вона фокусується на класах меншості.

Далі проаналізуємо ROC-криву (рисунок 4.5), а також побудуємо матрицю невідповідностей (рисунок 4.6).

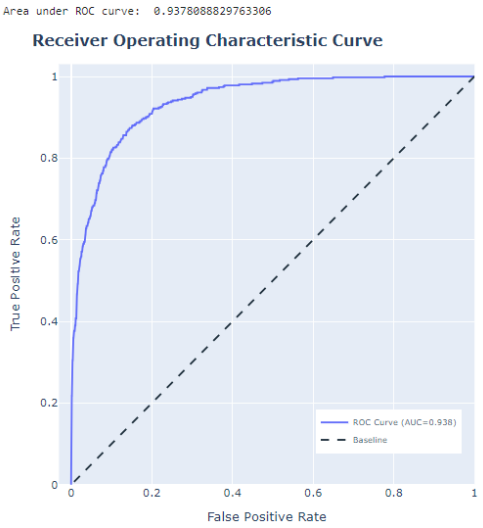

Рисунок 4.5 – ROC-крива для валідаційної вибірки даних

За значеннями матриці невідповідності розрахуємо оптимальне значення $F1score = 0,66$ для оптимального співвідношення показників Precision та Recall.

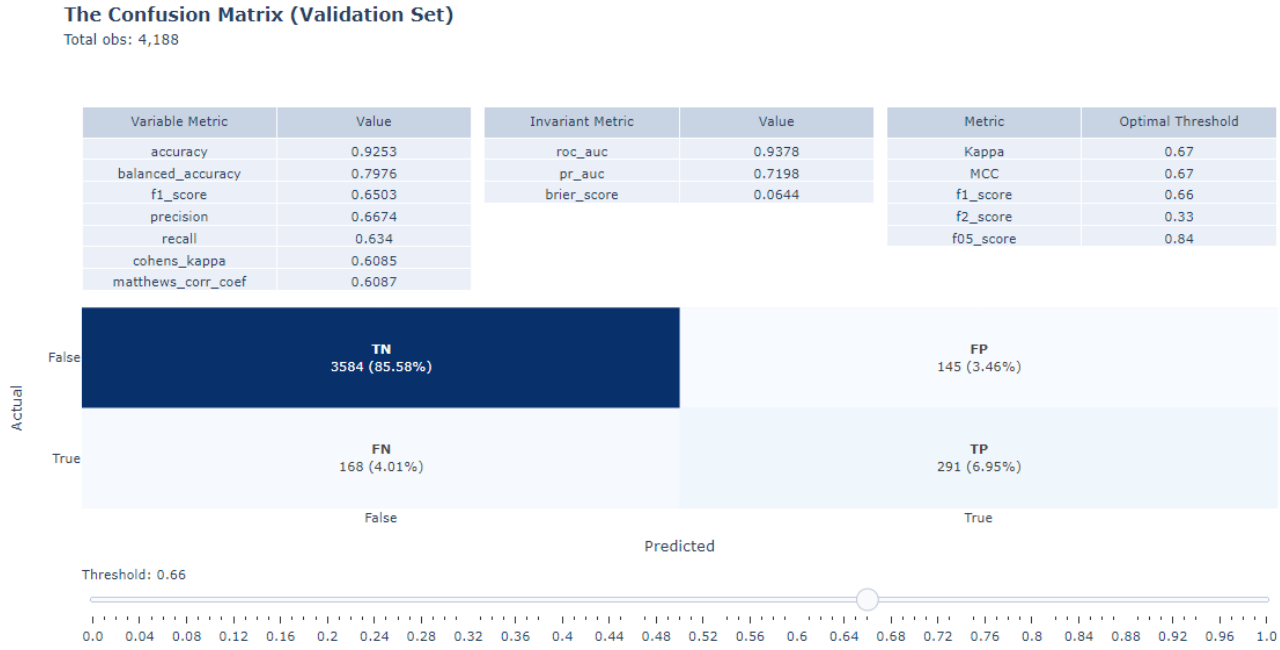

Рисунок 4.6 – Матриця невідповідності для валідаційної вибірки

З отриманих результатів бачимо, що з 4188 гравців у валідаційній вибірці наша модель може розмітити правильно 291 гравця як таких, що відвалюються (churn-гравець), але ще 168 гравців, які також є churn-гравцями, модель не розмітить. Тому ми бачимо Recall показник на рівні 0,634. В цілому це середня якість моделі, що в майбутньому може бути допрацьована.

Далі подивимося на вагу впливу окремих ознак (feature). Оцінку впливу будемо дивитися по показнику Shap value. Shap (SHapley Additive exPlanations) – це метод для інтерпретації прогнозів моделей машинного навчання. Shap використовує концепції з теорії ігор, зокрема, числа Шеплі (Shapley values), для призначення важливості кожної ознаки у моделі прогнозу. Сума SHAP values для всіх ознак для конкретного прогнозу дорівнює різниці між прогнозом моделі та середнім прогнозом моделі на тренувальному наборі даних. Це дозволяє розкрити, які ознаки призводять до збільшення чи зменшення

прогнозу. Значення показника Shap value для окремих ознак набору даних наведено на рисунку 4.7 у порядку спадання.

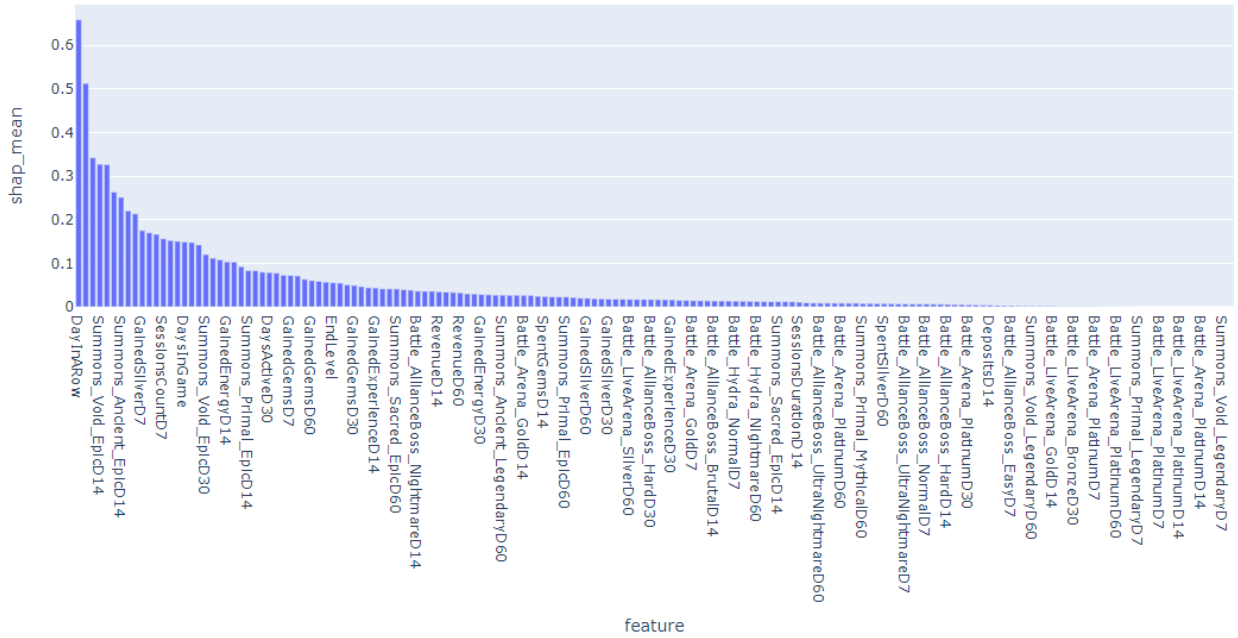

Рисунок 4.7 – Графік Shap value для кожної ознаки у моделі

Можна побачити, що є суттєво впливові ознаки та практично не впливові. Якщо залиши тільки топ ознак за показником Shap value, наприклад ті, котрі мають більше 0,1 ваги то отримаємо список, що наведений на рисунку 4.8.

```
['DayInARow',
 'Summons_Sacred_LegendaryD7',
 'Battle_DoomTowerD7',
 'Summons_Void_EpicD14',
 'GainedEnergyD7',
 'Summons_Ancient_EpicD30',
 'Summons_Ancient_EpicD14',
 'Summons_Primal_EpicD7',
 'Battle_DoomTowerD30',
 'GainedSilverD7',
 'Summons_Void_EpicD60',
 'Summons_Ancient_LegendaryD30',
 'SessionsCountD7',
 'DaysActiveD60',
 'GainedGemsD14',
 'DaysInGame',
 'DaysActiveD7',
 'GainedExperienceD60',
 'Summons_Void_EpicD30',
 'Summons_Sacred_EpicD30',
 'Battle_DoomTowerD14',
 'GainedEnergyD14',
 'DepositsD7']
```

Рисунок 4.8 – Список ознак з Shap value більшим за 0,1

Як можна побачити з рисунку 4.8, до тих що домінують, належать ознаки з наступних груп:

– ознаки з Summon групи, яка показує, як багато гравець відкриває нових героїв зі спеціального айтему у грі;

– ознаки, які показують активність за деякий період, зокрема DaysInAROW показує яку кількість дні підряд грає гравець;

– ознаки з групи Battle_DoomTower, яка показує кількість битв у спеціальному розділі на мапі, доволі складному етап для гравців.

Проаналізуємо, яким чином та чи інша ознака впливає на показник прогнозування моделі. Побудуємо графіки залежності для ознак, та подивимося на деякі з них (рисунок 4.9). Слід зазначити, що було проведено нормалізацію даних ознак, тобто перетворення даних до безрозмірних одиниць діапазону від 0 до 1. Тому по осі oX вказані не фактичні дані ознак, а їх нормалізовані значення.

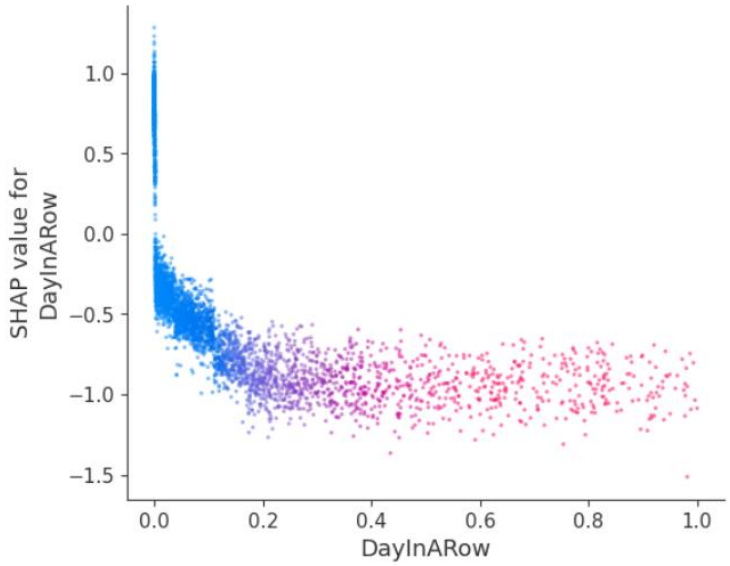

Рисунок 4.9 – Графік залежності для ознаки DaysInARow

На рисунку 4.9 можна побачити чітку залежність, яка свідчить про те, що при високому значенні цієї ознаки ймовірність відтоку (churn) гравця знижується.

Далі проаналізуємо аналогічний графік для ознаки Summons_Sacred_LegendaryD7. Ця ознака дає розуміння того, як багато

отримав гравець найбільш коштовних героїв з найбільш дорогого предмета у грі, тобто чи висока доля найкращих героїв з усіх отриманих саме з цього предмета. Бачимо, що при низькому значені цієї ознаки ймовірність відтоку підвищується, а як тільки показник підвищується, тобто гравці отримують найкращих героїв, то показник Shapvalue стає від'ємним, що свідчить про зниження ймовірності відтоку (churn).

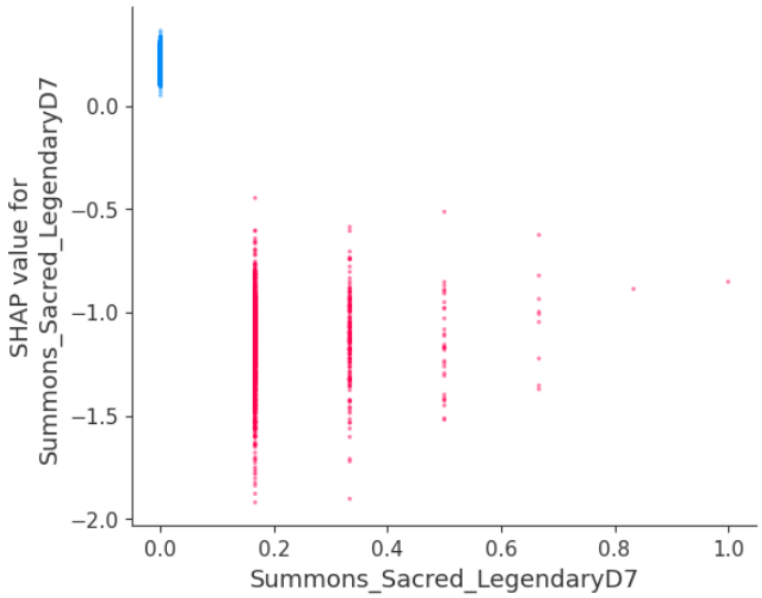

Рисунок 4.10 – Графік залежності для ознаки LegendaryD7

Так само виконується аналіз кожної ознаки для розуміння її впливу та виявлення нечітких ознак, котрі не дають чіткого розуміння впливу на ймовірність відтоку гравця.

Наступним етапом є прогон моделі на тестовій вибірці даних з урахуваням всіх вищенаведених параметрів та тільки на тих ознаках, для яких показник Shapvalue є більшим за 0,1. Результати наведено на рисунку 4.11.

З рисунку 4.11 можна побачити, ми отримали трохи кращі показники, ніж на валідаційній вибірці, але вони все ще не сутево кращі.

	Accuracy	Recall	Precision	F1
train	0.981984	0.992851	0.868647	0.926606
validation	0.925263	0.633987	0.667431	0.650279
test	0.936676	0.651351	0.723724	0.685633

Рисунок 4.11 – Графік показників моделі в порівнянні на різних даних

Побудуємо матрицю невідповідності (confusionmatrix) для тестового набору даних та подивимося на результати (рисунок 4.12).

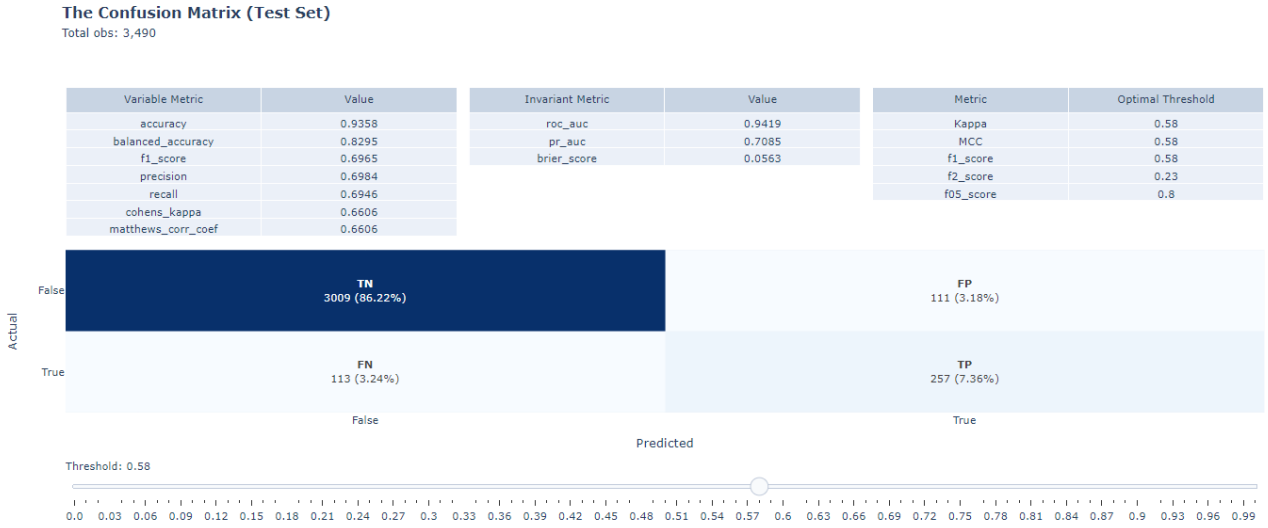

Рисунок 4.12 – Матриця невідповідності для тестової вибірки
(з урахування оптимального f1 показника)

Врахувавши усі проміжні результати навчання та аналізу ми отримали модель, яка прогнозує ймовірність відтоку гравців, спираючись на незбалансовані вибірки, та з використанням таких ознак як, наприклад, активність гравців у батлах на всіх існуючих локаціях у грі, у призові героїв за останні 7,14,30 та 60 днів.

Бачимо що вибір ознак дає ефект в поліпшенні якості моделі, але треба більше детально розбиратися в яких ще напрямках можна зробити покращення.

Висновки за розділом 4

У даному розділі наведено результати обчислювального експерименту щодо навчання та оцінювання якості програми, яка дозволяє прогнозувати ймовірність відтоку гравців мобільної гри в жанрі Fantasy Collection RPG.

Проаналізовано вхідний набір даних з точки зору пропущених значень, збалансованості класів. Обрано найкращий набір гіперпараметрів, який слід використовувати для побудови моделі з найвищою очікуваною ефективністю на основі визначеної цільової метрики.

Проаналізовано вплив окремих ознак та обрано ті з них, які є найвпливовішими на ймовірність відтоку гравців.

З врахуванням аналізу проміжних результатів побудовано остаточну модель, якість якої покращилась порівняно з початковою.

ВИСНОВКИ

У кваліфікаційній роботі було розглянуто проблему прогнозування відтоку гравців у ігровій індустрії та проаналізовано існуючі підходи до розв’язання цієї задачі. Найбільш перспективною групою методів є методи машинного навчання, які дозволяють аналізувати великі обсяги даних різної природи. Прогнозування відтоку гравців з використанням методів машинного навчання є задачею класифікації, де метою є визначення того факту, чи піде гравець з гри (клас «відтік») або залишиться (клас «не відтік»).

У ході системного аналізу проблеми із врахуванням особливостей набору даних обраної мобільної гри, було обрано найбільш підходящий метод прогнозування відтоку гравців, а саме метод градієнтного бустінгу.

У ході виконання роботи було розроблено програмний продукт, який дозволив розв’язати поставлену задачу. Шляхом обчислювальних експериментів на основі робочого датасету було обрано найкращі значення гіперпараметрів моделі та ті ознаки, які є найвпливовішими для ймовірності відтоку. З використанням отриманих значень було побудовано ефективно працюючу модель.

Результати, отримані у кваліфікаційній роботі, можуть бути корисними в прогнозуванні відтоку гравців в іграх, що допоможе зберегти гравців якнайдовше в грі та підвищити дохід самої гри.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ломія С. Г. Системний аналіз задачі сегментації гравців комп'ютерних ігор // 27-й Міжнародний молодіжний форум «Радіоелектроніка і молодь у XXI столітті». Зб. матеріалів форуму. Т. 7. Харків : ХНУРЕ. 2023. С.157-158.

2. Сорока К. О. Основи теорії систем і системного аналізу. Харків : "ХНАМГ", 2004. 115 с.

3. Катренко А. В., Пасічник В. В., Пасько В. П. Теорія прийняття рішень. Київ : Видавнича група BHV, 2009. 448 с.

4. Катренко А. В. Системний аналіз. Львів : "Новий світ – 2000", 2011. 396 с.

5. Raschka S., Mirjalili V. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow. Packt Publishing, 2017. 622 p.

6. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc., 2019. 850 p.

7. Flach P . Machine Learning. Cambridge University Press, 2012. 410 p.

8. Басюк Т. М., Литвин В. В., Захарія Л. М., Кунанець Н. Е. Машинне навчання. Львів : Видавництво «Новий Світ - 2000», 2021. 315 с.

9. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Системи штучного інтелекту. Львів : Магнолія-2006, 2013. 279 с.

10. Müller A, Guido S. Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media, 2016. 398 p.

11. Chollet F. Deep Learning with Python. Manning, 2021. 504 p.