Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра математики

# Кваліфікаційна робота

освітній ступінь – бакалавр

на тему: «**ПОШУК ОПТИМАЛЬНИХ ПАРАМЕТРІВ В АУКЦІОНІ ЗА ДОПОМОГОЮ НАВЧАННЯ З ПІДКРІПЛЕННЯМ**»

Виконала: студентка 4-го року навчання,

Освітньої програми «Прикладна математика», 113

Сивохіп Анна Вадимівна

Керівник: Крюкова Галина Віталіївна

кандидат фіз.-мат. наук

Рецензент _____
(прізвище та ініціали)

Кваліфікаційна робота захищена
з оцінкою _____

Секретар ЕК _____
«____» _____ 20____ р.

Київ – 2023

| № з/п | **ПЕРЕЛІК РОБІТ** | Термін виконання | Примітка |
|---|---|---|---|
| 1. | Вибір теми, затвердження її на засіданні кафедри та закріплення наукового керівника<br>Ознайомлення студента з критеріями оцінювання кваліфікаційної роботи (п. 8.5). | травень 2022 | |
| 2. | Вивчення джерел літератури, матеріалів архівів, періодичних видань, збір та узагальнення фактів, даних | лютий – березень | |
| 3. | Складання плану кваліф. роботи та узгодження з науковим керівником | лютий – березень | |
| 4. | Написання розділів роботи | лютий – березень | |
| 5. | Проміжний контроль виконання роботи | березень | |
| 6. | Написання кваліфікаційної роботи в цілому, ознайомлення з її першим варіантом наукового керівника | квітень – травень | |
| | **Формулювання задачі**<br>(постановка проблеми, теоретичні основи, огляд літературних джерел) | | |
| | **Ознайомлення з навчанням з підкріпленням**<br>(аналітично-дослідницька частина) | | |
| | **Вибір методу вирішення проблеми**<br>(проектно-рекомендаційна частина) | | |
| 7. | Повне завершення написання кваліфікаційної роботи, оформлення її згідно з вимогами й подання на відгук науковому керівнику | кінець травня | |
| 8. | Подання кваліфікаційної роботи для перевірки письмових робіт студентів НаУКМА на відповідність вимогам академічної доброчесності, | кінець травня | |
| 9. | Подання на зовнішню рецензію | кінець травня | |
| 10. | Підготовка до захисту кваліфікаційної роботи на засіданні кафедри: написання доповіді та виготовлення ілюстративного матеріалу | **до 28 травня** | |
| 11. | Попередній захист кваліфікаційної роботи на засіданні кафедри | **до 22 травня** | |
| 12. | Подання кваліфікаційної роботи на кафедру з усіма супроводжувальними документами | **до 28 травня** | |
| 13. | Публічний захист кваліфікаційної роботи перед екзаменаційною комісією | 5-6 червня | |

# ЗМІСТ

# ПЕРЕЛІК УМОВНИХ ОЗНАЧЕНЬ

| Скорочення, абревіатура | Значення |
|---|---|
| ADX | Ad Exchange Platform<br>Платформа з продажу реклами |
| DSP | Demand Side Platform<br>Платформа попиту, рекламодавець |
| MDP | Markov Decision Process<br>Марковський процес прийняття рішень |
| oRTB, openRTB | Open Real Time Bidding<br>Відкритий рекламний аукціон у реальному часі |
| RL | Reinforcement learning<br>Навчання з підкріпленням |
| SSP | Supply Side Platform<br>Платформа пропозиції, паблішер |
| Паблішер | Від англ. Publisher<br>Власник ресурсу, на якому можна розміщати рекламу: веб-сайт, мобільний додаток, сервіси розумного телебачення, тощо |

# ВСТУП

Рекламні біржі та платформи з продажу реклами стають все більш популярними і важливими у сучасному цифровому рекламному середовищі. Використання аукціонів рекламних оголошень дозволяє ефективно зв'язувати рекламодавців з власниками веб-сайтів, забезпечуючи оптимальний показ рекламних матеріалів для цільової аудиторії.

Коли рекламодавці мають намір розмістити свою рекламу на веб-сайті, вони використовують OpenRTB формат для створення запиту на показ реклами. Запит містить інформацію про цільову аудиторію, яку рекламодавці бажають привернути, таку як вікові групи, місце проживання, інтереси та інші параметри, які дозволені з точки зору законодавства.

Отримавши запит, рекламна біржа (Ad Exchange platform) виступає посередником між рекламодавцями і паблішерами. Паблішер є власником веб-сайту, на якому буде розміщуватись реклама. Запит від паблішера надсилається на рекламну біржу, яка у свою чергу пропонує його різним рекламодавцям, які мають бажання показати свою рекламу. Рекламодавці вказують свою ціну, виходячи з параметрів запиту. За допомогою OpenRTB визначається рекламодавець з найвищою пропозицією, який вважається переможцем.

Після визначення переможця, реклама надсилається на веб-сайт для показу користувачам. Усі ці процеси відбуваються автоматично, залучаючи незначний час, в середньому близько 200 мілісекунд. Люди не беруть безпосередньої участі в цих операціях, що дозволяє ефективно керувати рекламними кампаніями.

У даній роботі пропонується підхід до вирішення проблеми фіксованої націнки в контексті рекламних бірж. Часто фіксований відсоток націнки виявляється неприйнятно високим для рекламодавців або занизьким для досягнення максимального прибутку. Основною метою цієї роботи є створення автоматизованої опції налаштування націнки, яка ґрунтується на параметрах, наданих паблішером, а також на доступних відомостях про користувача.

Розробка моделі, яка може прогнозувати оптимальну націнку для кожного запиту на підставі доступної інформації, має великий потенціал для максимізації прибутку платформи з продажу реклами. У даній роботі ми використовуємо підхід Q-навчання для створення алгоритму, який допомагає вирішити цю проблему. Застосовуючи методи навчання з підкріпленням, наш алгоритм навчається на вхідних даних та здатний приймати рішення про оптимальну націнку в реальному часі. Врахування різних параметрів запиту, таких як ліміт рекламодавця, країна, операційна система, формат реклами та інші, дозволяє побудувати ефективну модель, що враховує особливості кожного запиту та забезпечує оптимальне управління рекламними кампаніями. Наш алгоритм має потенціал допомагати платформі з продажу реклами забезпечити максимальний прибуток і покращити ефективність рекламних кампаній рекламодавців.

# ОЗНАЙОМЛЕННЯ З АУКЦІОНОМ РЕКЛАМНИХ ОГОЛОШЕНЬ oRTB

OpenRTB — це протокол, який полегшує купівлю та продаж цифрових рекламних ресурсів у режимі реального часу через програмні аукціони. Він слугує стандартизованою платформою зв’язку між рекламодавцями, паблішерами та ADX (Ad Exchange Platform), забезпечуючи автоматичну торгівлю показами реклами (англ. *ad impression*).

Процес OpenRTB починається, коли користувач заходить на ресурс паблішера. Платформа попиту (SSP, Supply Side Platform) паблішера ініціює запит ставки у форматі OpenRTB, який містить важливу інформацію про конкретне розміщення реклами, розмір вікна, характеристики користувача та характеристики пристрою, з якого користувач заходить на ресурс. Цей запит зазвичай включає такі дані, як демографічні дані, географічне розташування, контекстна інформація та будь-які додаткові параметри таргетингу, визначені паблішером та дозволені для передачі законом.

Потім запит передається на різні ADX, де він стає доступним для потенційних рекламодавців, які беруть участь в аукціоні. Рекламодавці, представлені своїми відповідними платформами, відомими як Demand-Side Platforms (DSP), аналізують запит та оцінюють можливість показу своїх оголошень на основі різних факторів, зокрема критеріїв таргетингу, бюджетних обмежень і цілей кампанії.

Отримавши запит, кожен рекламодавець обробляє доступну інформацію та визначає ціну ставки, яка представляє максимальну суму, яку він готовий заплатити за цей показ.

Згодом рекламодавець надсилає відповідь назад до ADX. Відповідь містить ціну ставки та додаткові дані, що стосуються ставки, як-от творчі ресурси, URL-адреси рекламованих сторінок і будь-які конкретні інструкції щодо цілей рекламної кампанії. Ця відповідь на ставку порівнюється з іншими ставками, отриманими за той самий показ протягом періоду аукціону, що зазвичай відбувається протягом декількох сотень мілісекунд.

ADX передає відповідь на ставку паблішеру. Паблішер збирає та аналізує ставки, застосовує механізми аукціону та визначає виграшну ставку на основі попередньо визначених правил аукціону. Рекламодавець з найвищою ставкою виграє аукціон, і його оголошення вибирається для показу на ресурсі паблішера.

Упродовж цього процесу різні сторони, зокрема рекламодавці, DSP, SSP та ADX, обмінюються інформацією та беруть участь у ряді автоматизованих транзакцій, забезпечуючи ефективну та цілеспрямовану доставку реклами.

Протокол OpenRTB стандартизує ці взаємодії, сприяючи прозорості та масштабованості в екосистемі цифрової реклами.

Загалом OpenRTB революціонізує цифровий рекламний простір, автоматизуючи купівлю та продаж рекламних ресурсів, дозволяючи приймати рішення в реальному часі та максимізуючи ефективність рекламних кампаній.

# ПОСТАНОВКА ЗАДАЧІ

ADX є посередником між паблішером та рекламодавцем у аукціоні oRTB. Технологія працює даним чином:

1. Користувач заходить на сайт паблішера
2. Паблішер відправляє запити на показ реклами з вказуванням мінімальної ставки $m$ на різні платформи
3. Платформа додає свою націнку $a \in (0,100)$ і передає запит рекламодавцю з новою мінімальною ставкою $r/(1-\frac{a}{100})$
4. Рекламодавець приймає рішення, чи буде він відповідати на запит. Якщо мінімальна ставка є зависокою для рекламодавця, він ігнорує запит. У зворотньому випадку, він дає відповідь зі своєю ставкою $b$
5. Платформа отримує відповідь зі ставкою рекламодавця, віднімає свою націнку за формулою

$$final\ bid\ =\ b * \left(1 - \frac{a}{100}\right)$$

   та відправляє відповідь на запит паблішера
6. Паблішер на своєму боці проводить аукціон. Найвища ставка виграє та отримує можливість показати свою рекламу
7. Якщо виграє наш рекламодавець – платформа отримує прибуток, який визначається за формулою

$$profit\ =\ b - final\ bid\ =\ b * \frac{a}{100}$$

Задача полягає в тому, щоб розробити модель, яка буде прогнозувати оптимальну націнку для кожного запиту на основі наявної інформації, такої як ліміт рекламодавця, країна, ОС, формат реклами тощо, з метою максимізації заробітку платформи. Ця модель повинна бути навчена на історичних даних та здатна працювати в режимі реального часу, швидко приймаючи рішення про націнку для кожного нового запиту.

# ПОБУДОВА MDP ТА ФУНКЦІЇ ОНОВЛЕННЯ Q-ЗНАЧЕНЬ

Ми обрали навчання з підкріпленням для вирішення цієї проблеми через його здатність навчатися оптимальних стратегій шляхом проб та помилок в динамічних середовищах. Навчання з підкріпленням дозволяє моделі навчатися на основі історичних даних та взаємодіяти з середовищем, адаптуючи свою стратегію на основі отриманого винагородження. За допомогою формулювання проблеми як процесу прийняття рішень Маркова (MDP), ми можемо моделювати послідовний процес прийняття рішень та оптимізувати стратегію націнки з плином часу. Модель може досліджувати різні дії, оцінювати їх вплив на доходи платформи та оновлювати свою стратегію відповідно.

Щоб сформулювати проблему як MDP, нам потрібно визначити множини станів середовища S, множини дій A, функцію винагороди та функцію ймовірності переходу. Щоб застосувати метод Q-навчання до задачі визначення оптимальної націнки для кожного запиту в режимі реального часу, потрібно визначити компоненти алгоритму Q-навчання: стани, дії, винагороди та процес оновлення Q-значень.

В алгоритмі Q-навчання метою є ітераційне вивчення оптимальної функції Q-значення за допомогою рівняння оптимальності Белмана:

$$V(s) = \max_a(R(s, a + \gamma \sum_{s'} P(a, s, s')V(s'))$$

Для цього ми зберігаємо всі Q-значення в таблиці, яку будемо оновлювати на кожному кроці за допомогою ітерації Q-навчання.

1. **Множина станів**

Стан $S$ на момент часу $t$ включає інформацію, таку як ліміт рекламодавця, країна, операційна система (ОС), формат реклами та будь-які інші важливі характеристики, що впливають на процес прийняття рішень.

Визначимо стан $s$ як сукупність $(m, c, os, f)$, де:

- $m$ — мінімальна ставка
- $c$ — країна користувача,
- $os$ — операційна система користувача,
- $f$ — формат оголошення.

2. **Множина дій**

Дія $a$ в момент часу $t$ — це відсоток націнки, який платформа додає до мінімальної ставки у відповідь на запит. Націнка має бути від 0% до 100% (не включно). У цій задачі дії $A$ відповідатимуть набору всіх можливих націнок, які платформа може застосувати до ціни пропозиції, отриманої від рекламодавця.

Ми можемо представити цей набір дій як:

$$A = \{a_1, a_2, \ldots, a_n\}$$

де $a_i$ представляє $i$-й можливий відсоток націнки, який платформа може застосувати до мінімальної ставки. Платформа може вибрати будь-яку з цих розміток для певного запиту. Набір дійсних дій $A$ є скінченним, але він може мати багато елементів залежно від деталізації відсотків розмітки, які ми хочемо розглянути.

3. **Функція винагороди**

Винагорода $R(s, a)$ у момент часу $t$ — це прибуток, отриманий платформою за запит на аукціон, враховуючи відсоток націнки $a$.

Платформа отримує прибуток, коли рекламодавець виграє аукціон. Прибуток, отриманий платформою, визначається за формулою:

$$profit = b * \frac{a}{100}$$

Якщо рекламодавець виграє аукціон, платформа отримує прибуток. Якщо рекламодавець не виграє аукціон або не дає відповіді, платформа не отримує прибутку (тобто винагорода дорівнює нулю).

Тому функцію винагороди $R(s, a)$ можна визначити як:

$$R(s,a) = \begin{cases} b * \frac{a}{100}, \text{якщо } b \text{ є виграшною ставкою} \\ 0, \quad \text{якщо рекламодавець не відповів або } b \text{ не є виграшною ставкою} \end{cases}$$

**4. Оновлення Q-значень**

Q-значення представляє очікувану накопичувальну винагороду для певної пари стан-дія. Оновлення Q-значення виконується за таким рівнянням:

$$Q(s,a) \leftarrow Q(s,a) + \alpha * [r + \gamma * max_{a'} Q(s',a') - Q(s,a)] \quad (1)$$

Де:

- $Q(s, a)$ - Q-значення пари стан-дія $(s, a)$,
- $\alpha$ - швидкість навчання,
- $r$ - миттєва винагорода, отримана при виборі дії a у стані s,
- $\gamma$ - фактор знецінювання, який враховує миттєві та майбутні винагороди,
- $max_{a'} Q(s', a')$ - максимальне Q-значення серед всіх можливих дій $a'$ у наступному стані $s'$.

**5. Дисконтний коефіцієнт**

Фактор знецінювання або дисконтний коефіцієнт $\gamma$ – це параметр, який визначає відносну важливість майбутніх винагород порівняно з негайними винагородами. У цьому випадку високий коефіцієнт дисконтування віддасть пріоритет отриманню прибутку в довгостроковій перспективі, тоді як низький коефіцієнт дисконтування надасть пріоритет короткостроковому прибутку.

Дисконтний коефіцієнт $\gamma$ визначений у цій задачі як $0$. Вищі його значення можна включити в задачу оптимізації, налаштувавши функцію винагороди для відображення вартості грошей у часі.

# ПОБУДОВА АЛГОРИТМУ Q-НАВЧАННЯ

У цьому описі алгоритму Q-навчання будуть представлені основні кроки і концепції, що лежать в основі цього методу. Також будуть наведені деталі використання Q-навчання для прогнозування оптимальної націнки в аукціоні oRTB.

1. **Ініціалізація**: Довільно ініціалізувати Q-значення для всіх пар стан-дія в області задачі.

2. **Дослідження і використання (Exploration or Exploitation):** Вибрати дію, використовуючи стратегію exploration-exploitation. Спочатку алгоритм може досліджувати, випадково вибираючи дії або за допомогою стратегії дослідження. З плином часу він поступово перейде до використання вивчених Q-значень, вибираючи дію з найвищим Q-значенням у заданому стані.

3. **Вибір дії та перехід до нового стану:** Отримати запит, подивитись на стан і вибрати дію (націнку) на підставі обраної стратегії дослідження і використання.

4. **Обробка запиту:** Надіслати запит рекламодавцеві, отримати ставку $b$ та обчислити кінцеву ставку

$$final\ bid\ =\ b * \left(1 - \frac{m}{100}\right)$$

5. **Розрахунок винагороди та оновлення Q-значень:** Розрахувати миттєву винагороду на основі фінальної ставки, оновити Q-значення, використовуючи рівняння оновлення Q-значень (1).

6. Повторювати кроки 3-5 для кожного нового запиту, оновлюючи Q-значення на основі спостережених винагород.

З плином часу Q-значення збігатимуться до оптимальних значень, які відображають найкращі вибори націнки в кожному стані. Швидкість навчання $\alpha$ впливає на баланс між врахуванням нової інформації та урахуванням майбутніх винагород у оновленнях Q-значень.

# ПРИКЛАД ЗАСТОСУВАННЯ

Приклад застосування мови програмування Python для вирішення задачі. Ми використовуємо вигаданий модуль RTB для інтеграції з системою проведення аукціону, імплементація якого є поза межами даної роботи.

Спочатку ми імпортуємо необхідні бібліотеки.

```
import math
import numpy as np
import itertools as it
import math as m
import random
import rtb  # imaginary module to work with
```

**Визначення дискретних значень дій та кроку.** Задаються можливі значення націнок (дій) за допомогою numpy.arange().

```
action_step = 5
actions = np.arange(0, 100, action_step)[1:]
number_of_actions = len(actions)
```

Визначення дискретних значень мінімальної ставки, країн, типів реклами та операційних систем.

```
bid_floor_step = 0.5
bid_floors = np.arange(0, 50, bid_floor_step)
countries = np.array(["Ukraine", "USA"])
type_of_ads = np.array(["banner", "video"])
platforms = np.array(["IOS", "Android", "Windows"])
```

**Визначення станів.** Використовуються всі можливі комбінації бідфлору, країни, типу реклами та платформи для створення унікальних станів моделі.

```
states = [
    (b, c, t, p)
    for b in bid_floors
    for c in countries
    for t in type_of_ads
    for p in platforms
]
number_of_states = len(states)
states_map = {k: v for (v, k) in enumerate(states)}
```

**Створення Q-таблиці.** Створюється початкова Q-таблиця, яка містить нульові значення для всіх можливих пар стан-дія. Розмір таблиці Q визначається кількістю станів та дій.

```
Q = np.zeros((number_of_states, number_of_actions))
```

**Визначення параметрів.** Визначаються значення параметрів eps (Exploration or exploitation) і lr (швидкість навчання), які використовуються під час навчання моделі.

```
eps = 0.1
lr = 0.5
```

**Початок навчання.** Запускається головний цикл, який працює безкінечно. На кожній ітерації модель отримує новий запит від системи rtb.

```
while True:
    req = (
        rtb.get_next_request()
    )    # get next request with all info needed for algorithm to work
```

**Вибір дії.** Використовуючи епсилон-жадібний підхід, модель вибирає дію на основі поточного стану. За значення eps з ймовірністю 0.1 вибирається випадкова дія, в іншому випадку вибирається дія з найбільшим значенням Q для даного стану.

```
    state_index = states_map[
        req.state(bid_floors)
    ]  # state is a convinient method to get all needed states in tuple, including finding of corresponding bit floor bucket


    if random.uniform(0, 1) < eps:
        action_index = random.randint(0, number_of_actions)
    else:
        action_index = np.argmax(Q[state_index, :])
```

**Обробка дії та отримання винагороди.** Обробляється вибрана дія для запиту, і отримується результат. Винагорода обчислюється, використовуючи отриманий прибуток та мінімальну ставку.

```
    res = req.proceed(
        markup=actions[action_index]
    )    # method on request to proceed with selected markup and get results
    reward = res.income / req.bid_floor    # calculated normalized income
```

**Оновлення Q-значень.** За допомогою формули для оновлення Q-значень оновлюється Q-значення з використанням винагороди і швидкості навчання.

```
    Q[state_index, action_index] = Q[state_index, action_index] + lr * (
        reward - Q[state_index, action_index]
    )
```

Цей процес повторюється безкінечно, навчаючи модель приймати оптимальні рішення щодо націнки в реальному часі на основі навчальних даних, які надаються системою rtb.

# ВИСНОВКИ

Код представлений у даній роботі є прикладом імплементації опції оптимальної націнки. Загалом, використання методу Q-навчання є перспективним інструментом для оптимізації націнки в аукціоні oRTB, що може призвести до підвищення прибутковості платформи та поліпшення її здатності у прийнятті рішень.

У подальших дослідженнях можна розглянути розширення моделі на основі Q-навчання, включаючи додаткові фактори, які можуть бути передані у запиті, такі як тип трафіку, розмір рекламного вікна, інформація про паблішера та безпосередньо назва ресурсу.

Крім того, можна розглянути комбінацію Q-навчання з іншими методами машинного навчання для отримання більш точних та ефективних рішень. Ефективними також можуть бути алгоритми SARSA та Policy gradient methods.

Загалом, використання методу Q-навчання є перспективним інструментом для оптимізації націнки в аукціоні oRTB, що може призвести до підвищення прибутковості платформи та поліпшення її здатності у прийнятті рішень.

# ЛІТЕРАТУРА

[1] Richard S. Sutton and Andrew G. Barto, Reinforcement learning: An introduction, MIT Press, 1998.

[2] Benoit Liquet, Sarat Moka, and Yoni Nazarathy, Deep Learning Math, 2023 https://deeplearningmath.org/deep-reinforcement-learning.html

[3] Титаренко, Д., Оцінка ефективності стратегій навчання з підкріпленням на прикладі гри в 21, Бакалаврська робота, НТУУ "КПІ 2021.

[4] Han, X., Reinforcement Learning in Finance, Written and Oral Presentation, New York University, 2018.

[5] Eyal Even-Dar and Yishay Mansour, Learning Rates for Q-learning, 2023 https://www.jmlr.org/papers/volume5/evendar03a/evendar03a.pdf