

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра комп'ютерних наук

Пояснювальна записка

до магістерська кваліфікаційної роботи.
на тему:

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ІНТЕРНЕТ-МАРКЕТИНГУ

Виконав: студент 2 курсу групи 1КН-19м
напряму підготовки 122
«Комп'ютерні науки»

Солоний М. А.
(прізвище та ініціали)

Керівник к. т. н., доц. Арсенюк І. Р.
(прізвище та ініціали)

Рецензент к. т. н., доц. Коваленко О. О.
(прізвище та ініціали)

Вінниця – 2020 року

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
д. т. н., проф. Яровий А. А.

_____ (підпис)
“ _____ ” _____ 2020 року

ЗАВДАННЯ

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра зі спеціальності: 122 – «Комп'ютерні науки»

08-22.МКР.011.19.000.ПЗ

Магістранта групи 1КН-19м Солоного Максима Андрійовича

Тема магістерської кваліфікаційної роботи: «Інформаційна технологія інтернет-маркетингу»

Вхідні дані: кількість товарів – не менше 300, кількість аналітичних даних для кожного товару – не менше 10, кількість товарів для обробки нейромережею (які були обрані користувачем за 1 сесію) – не менше 7, кількість категорій товарів не менше 20, мова програмування об'єктно-орієнтована.

Короткий зміст частин магістерської кваліфікаційної роботи:

1. Графічна: алгоритм функціонування інформаційної технології інтернет-маркетингу, загальна структурна схема інформаційної технології інтернет-маркетингу, загальна схема роботи алгоритму програмного засобу інформаційної технології інтернет-маркетингу, UML-діаграма класів програмного засобу інформаційної технології інтернет-маркетингу, вікно авторизації програмного засобу інформаційної технології інтернет-маркетингу, вікно реєстрації програмного засобу інформаційної технології інтернет-маркетингу, головний екран програмного засобу інформаційної технології інтернет-маркетингу.

2. Текстова (пояснювальна записка): вступ, аналіз предметної області інформаційної технології інтернет маркетингу, структурна організація та проектування інформаційної технології інтернет-маркетингу, програмна реалізація інформаційної технології інтернет-маркетингу, економічна частина, висновки, перелік використаних джерел, додатки.

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз сучасного рівня розвитку інформаційних технологій бізнес-прогнозування. Постановка задач дослідження			Аналітичний огляд літературних джерел, задачі досліджень, розділ 1 ПЗ
2	Розробка математичної моделі та інформаційної технології прогнозування купівлі товару			Математична модель, інформаційна технологія, розділ 2 ПЗ
3	Підготовка економічної частини			Розділ 4 ПЗ
4	Програмна реалізація розробленої інформаційної технології, тестування та оцінка якості			Програмне забезпечення розділ 3 ПЗ
5	Апробація та/або впровадження результатів дослідження			Тези доповідей/акт впровадження
6	Оформлення пояснювальної записки, графічного матеріалу та презентації			Пояснювальна записка, графічний матеріал, презентація

Консультанти з окремих розділів магістерської кваліфікаційної роботи

1. Науковий керівник _____ канд. техн. наук, доц., доц. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)
“ _____ ” _____ 20__ р. І. Р. Арсенюк
ініціали та прізвище

2. Економічна частина _____ канд. екон. наук, доц., доц. кафедри ЕПВМ
(підпис) наук. ступінь, вчене звання (посада)
М. В. Бальзан
ініціали та прізвище

“ _____ ” _____ 20__ р.

Дата попереднього захисту роботи “ _____ ” _____ 20__ р.

Рецензент _____ канд. техн. наук, доц., доц. кафедри ПЗ
(підпис) наук. ступінь, вчене звання (посада)
О. О. Коваленко
ініціали та прізвище

Завдання видав
науковий керівник _____ канд. техн. наук, доц., доц. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)
І. Р. Арсенюк
ініціали та прізвище

“ _____ ” _____ 20__ р.

Завдання отримав магістрант _____ М. А. Солоний
(підпис) ініціали та прізвище

“ _____ ” _____ 20__ р.

РЕФЕРАТ

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології інтернет-маркетингу. Розроблена технологія забезпечує прогнозування набору товарів для покупок, та на відміну від існуючих, з метою підвищення точності прогнозування, передбачає оброблення ряду додаткових аналітичних даних, основними з яких є час переглядання товарів (з урахуванням захисту від неактивності користувача), кількість разів переглядання конкретного товару, а також фільтри сортування товарів.

Розглянуто та проаналізовано існуючі методи створення мобільних додатків. Проаналізовано існуючі нейронні мережі для розв'язання задачі прогнозування, обґрунтовано вибір нейронного перцептрона з поєднанням вихідних нейронів. Розроблено загальний алгоритм функціонування додатку, що передбачає використання нейронної мережі для прогнозування. Розроблено та протестовано програмний засіб на мові програмування JavaScript для прогнозування набору товарів для покупок. Даний додаток має на 1.5 – 3% вищу точність прогнозування.

ABSTRACT

This master's thesis is devoted to the development of information technology for Internet marketing. This technology provides forecasting of a set of goods for purchases, and unlike existing technologies includes a number of additional analytical data such as: time of viewing goods (including protection against user inactivity), number of views of a particular product, sorting filters.

The existing methods of creating mobile applications are considered and analyzed. The existing neural networks are analyzed to solve the problem of prediction, the choice of a neural perceptron with a combination of source neurons is substantiated. The general algorithm of online store and neural network of forecasting is developed. Developed a software tool in the JavaScript programming language for predicting a set of goods for purchases.

ЗМІСТ

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ІНТЕРНЕТ-МАРКЕТИНГУ.....	12
1.1 Аналіз задачі прогнозування набору товарів для покупок.....	12
1.2 Аналітичний огляд аналогів.....	13
1.3 Постановка задачі розробки інформаційної технології.....	17
1.4 Висновки.....	19
2 СТРУКТУРНА ОРГАНІЗАЦІЯ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ІНТЕРНЕТ-МАРКЕТИНГУ.....	20
2.1 Аналіз існуючих нейронних мереж для розв'язання задачі прогнозування.....	20
2.2 Обґрунтування вибору типу нейронної мережі для роботи з бізнес-прогнозуванням.....	24
2.2.1 Обґрунтування вибору нейронного перцептрона з поєднанням вихідних нейронів.....	24
2.2.2 Аналіз та обґрунтування вибору методу для вирішення проблеми перенавчання.....	26
2.2.3 Обґрунтування вибору реверсивного Dropout методу нормалізації...31	31
2.2.4 Розробка архітектури мобільного додатку для інтернет-маркетингу.....	33
2.3 Розробка структури інформаційної технології інтернет-маркетингу.....	37
2.4 Висновки.....	39
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ІНТЕРНЕТ-МАРКЕТИНГУ.....	40
3.1 Обґрунтування вибору мови програмування.....	40
3.2 Обґрунтування вибору бібліотеки для створення нейронної мережі.....	42

3.3 Обґрунтування вибору середовища розробки.....	44
3.4 Побудова UML-діаграми класів розробленого програмного засобу.....	51
3.5 Тестування розробленого програмного засобу інтернет-магазину з використанням нейронної мережі бізнес-прогнозування та аналіз результатів його роботи.....	53
3.7 Обґрунтування вибору архітектурної організації обчислень на GPU.....	57
3.8 Висновок.....	60
4 ЕКОНОМІЧНА ЧАСТИНА.....	61
4.1 Оцінювання комерційного потенціалу розробки.....	61
4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи.....	63
4.3 Прогнозування комерційних ефектів від реалізації результатів розробки.....	66
4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності.....	68
ВИСНОВКИ.....	72
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	74
ДОДАТКИ.....	75
Додаток А Інструкція користувача.....	80
Додаток Б Лістинг програми.....	84
Додаток В Графічна частина.....	95

ВСТУП

Актуальність теми дослідження. На сьогоднішній день інтернет став невід'ємною частиною нашого життя, кожного дня ми переглядаємо сотні гігабайт інформації використовуючи різні сервіси. В наші дні найпопулярнішим девайсом для роботи в мережі інтернет – смартфон. Кожного дня користувач смартфона оновлює, скачує, інсталує мобільні додатки.

Ринку сучасних мобільних додатків уже понад п'ятнадцяти років, однак бурхливість його розвитку в останній час набирає все більших обертів. Попит на створення мобільних додатків з боку компаній постійно зростає та все ще помітно перевищує пропозицію, що призводить до постійного подорожчання розробок [3].

Кожного дня створюються сотні додатків різного призначення. Одними з найпопулярніших є додатки інтернет-магазинів. Дані застосунки допомагають власнику збільшити його прибуток в декілька разів, завдяки онлайн торгівлі, яка в нас час розвивається з неймовірною швидкістю.

Даний тип додатків, приносить свої переваги користувачу також, оскільки він завжди має зручний спосіб: придбати необхідні речі, переглянути асортимент товару, бути проінформованим про акції та знижки.

На сьогоднішній день більша частина додатків використовує back-end, він забезпечує: зв'язок додатка з базою даних, відправку даних на клієнтську сторону, тощо. Але в наш час набирають все більшого розвитку хмарні сервіси, а саме, serverless архітектура, або безсерверна архітектура. Один з найбільших та найпродуманіших сервісів для створення безсерверних додатків – Amazon Web Services [4].

Serverless технологія дозволяє значною мірою збільшити продуктивність додатку, та швидкість його збирання завдяки технологіям мікросервіса.

У більшій частині великих інтернет-магазинах (Amazon, Aliexpress, Rozetka), тощо, використовується технологія прогнозування, на основі якої будуються різні типи маркетингових трюків, такі як акції розпродажі, коливання цін, тощо. Один з таких трюків – це створення каталогу товару, який може зацікавити користувача на основі його попередніх переглядів товару. Також дана технологія корисна для створення аналітики, яка також дуже актуальна в сфері інтернет-маркетингу тим, що такі нейронні мережі мають змогу прогнозувати необхідну кількість закупівель певного товару, основну таргет групу, тощо.

Саме тому покращення точності прогнозування за допомогою нових модифікованих алгоритмів на основі штучних нейронних мереж є важливим та актуальним питанням.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою дослідження магістерської кваліфікаційної роботи є підвищення точності прогнозування ймовірної набору товарів для покупок.

Для досягнення поставленої мети слід розв'язати такі завдання:

- розглянути та проаналізувати існуючі програмні реалізації розв'язання задачі прогнозування набору товарів для покупок;
- запропонувати математичну модель прогнозування очікуваних покупок для інформаційних технологій інтернет-маркетингу;
- обґрунтувати стадії розробки інформаційної технології та на їх основі розробити структуру та алгоритм роботи програмного засобу;

- виконати програмну реалізацію запропонованої інформаційної технології інтернет-маркетингу;
- провести тестування додатку та виконати аналіз отриманих результатів.

Об’єкт дослідження – процес прогнозування набору товарів для покупок.

Предмет дослідження – інформаційна технологія інтернет-маркетингу.

Метою дослідження магістерської кваліфікаційної роботи є підвищення точності прогнозування набору товарів для покупок користувача.

Область застосування – інтернет-маркетинг, аналітичні додатки для бізнес прогнозування.

Методи дослідження. У роботі використані такі методи наукових досліджень: методи оброблення цифрової інформації, методи математичної статистики для обрахунків результатів отриманих за допомогою програмного засобу, програмування на мовах високого рівня.

Наукова новизна одержаних результатів полягає в наступному:

Модернізована математична модель прогнозування набору товарів для покупок шляхом врахування додаткових аналітичних даних, які не використовуються в програмних додатках аналогів.

Практичне значення одержаних результатів:

1. Розроблено алгоритм роботи інтернет-магазину з нейромережевим модулем прогнозування набору товарів для покупок.
2. Розроблено програмний продукт інтернет-магазин з використанням нейронної мережі прогнозування.
3. Розроблено структуру технології інтернет-маркетинг, що дозволяє більш точно прогнозувати набір товарів для покупок.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими та

збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок здобувача. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно.

Апробація результатів роботи. Результати досліджень апробовані на XLIX науково-технічній конференції підрозділів ВНТУ.

Публікації. За результатами досліджень подано свідоцтво про реєстрацію авторського права на твір – комп'ютерна програма «Нейромережевий модуль інтернет-маркетингу» [1], опубліковано тези доповіді міжнародних науково-практичних конференцій [2].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ІНТЕРНЕТ-МАРКЕТИНГУ

1.1 Аналіз задачі прогнозування набору товарів для покупок

У даній кваліфікаційній роботі вирішується задача прогнозування вірогідної купівлі для мобільних додатків інтернет-магазинів. Актуальність завдання обумовлена тим, що майже кожна людина хоч раз у своєму житті використовувала додаток інтернет-магазину для купівлі того чи іншого товару, та для того щоб ще більше зацікавити нас як користувачів данного типу додатку, великі компанії почали використовувати нейронні мережі прогнозування вірогідної купівлі товару. Додавання прогнозування у додатки інтернет-магазинів збільшує вірогідність того, що користувач продивляючись каталог з рекомендованих товарів виконає процедуру придбання.

Раніше, для створення таргет груп, статистики продажу на кожний тип продукції, прогнозування кількості продажу в подальшому, виконували великі відділи маркетингу, на сьогоднішній день, дану роботу автоматизували за допомогою різних видів аналітики, які в свою чергу використовують нейронні мережі. Друга причина дослідження – підвищення точності прогнозування. Нажаль точність прогнозів на даний час все ще недостатньо висока.

Процес складання прогнозу можна умовно поділити на кілька етапів [3]:

1. Ідентифікування користувача, та створення для нього запису в базі даних;
2. Визначення категорій товарів які користувач найчастіше переглядає, відстежуючи кожний перехід по посиланням сайту, або додатку;
3. Створення окремого каталогу товарів на основі попередніх даних.

Проаналізувавши схожі роботи було зроблено висновок, що основною проблемою, яка впливає на точність прогнозу, є мала кількість аналітичних

даних яку використовують для прогнозування. Для вирішення цієї проблеми доцільно використати повний аналіз товару включаючи його ціну, виробника, час переглядання товарів (з врахуванням захисту від неактивності користувача), кількість разів переглядання конкретного товару, фільтри сортування товарів.

1.2 Аналітичний огляд аналогів

Прогнозування товару який може зацікавити клієнта є досить нетривіальним завданням. Найчастіше прогнозування товару відбувається на основі категорії або груп категорій які переглядає користувач без додавання усіх аналітичних даних таких як (ціновий діапазон, дата перегляду, фірма виробник, колір, тощо). На сьогоднішній день є багато систем які використовують бізнес прогнозування. Кожні з них мають свої переваги та недоліки.

На мій погляд, можна виділити такі основні додатки де використовується бізнес-прогнозування: Aliexpress, Rozetka, Joom, UA-ТАО.

Проаналізуємо детально проаналізуємо переваги та недоліки цих сервісів.

Aliexpress – торгова площадка роздрібною торгівлі малого бізнесу Китаю, що пропонує товари міжнародним покупцям в Інтернеті. Належить групі компаній Alibaba. Сайт допомагає малим підприємствам продавати свою продукцію клієнтам по всьому світові і, так само як й на сайті Amazon або eBay, на AliExpress можна знайти практично все що завгодно для продажу [3].

Дана торгова площадка аналізує переглянуті товари по категоріям, які не завжди правильно виставлені, та на основі них підбирає цікавий для користувача товар, але не включає ціновий діапазон, сортування по категоріям, дату перегляду, тому часто вам можуть запропонувати товар, який ви навіть не планували переглядати.

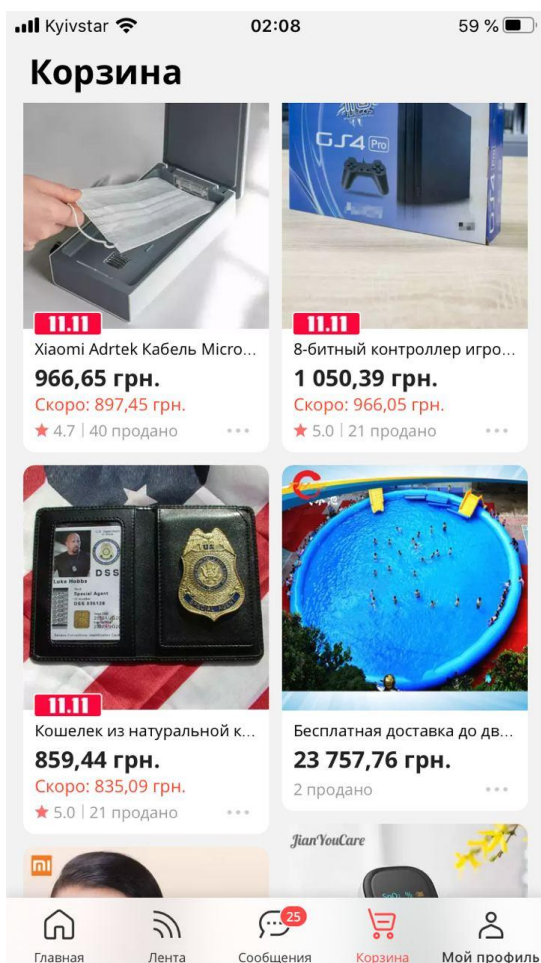


Рисунок 1.1 – Зпрогнозований каталог товарів від Aliexpress

Rozetka – український інтернет-магазин та маркетплейс, що з'явився 2005 року. Має відділення в Києві, Броварах та Одесі. Станом на серпень 2020 року сайт посідає 7 місце серед найвідвідуваніших в Україні [4].

Точність прогнозу в даному сервісі значно краща ніж в Aliexpress, так як визначені більш чіткі категорії товарів та аналізується цінова політика переглянутих товарів. Але й є свої недоліки, співвідношення останніх переглянутих товарів та рекомендуємих значно відрізняється.

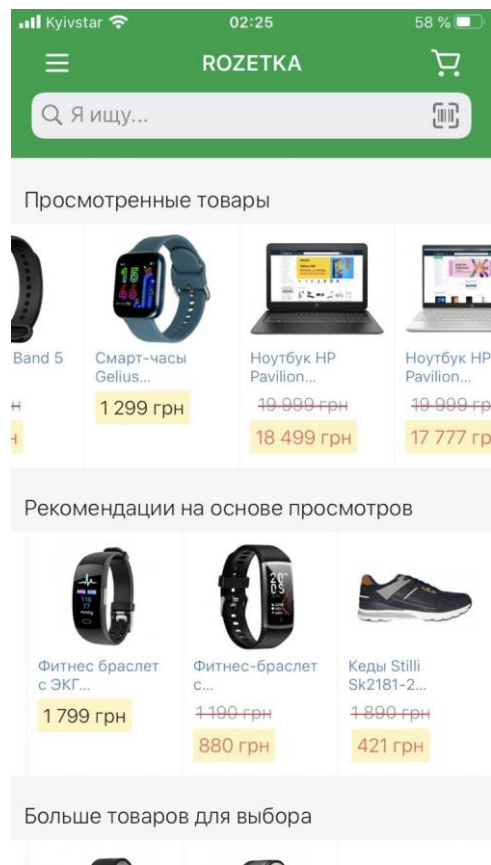


Рисунок 1.2 – Зпрогнозований каталог товарів від Rozetka

Ще одним яскравим прикладом є торгова площадка Joom – російська електронна торговельна площадка та однойменна програма для покупки товарів з Китаю та їхнього доставляння в Росію та інші країни. Штаб-квартира власника розташована в Москві. Компанія посідає друге (після AliExpress) місце серед російських інтернет-компаній, які доставляють товари з Китаю [5].

Дана площадка чітко розбиває каталог прогнозу на категорії які переглядав користувач, але не включає цінову політику.

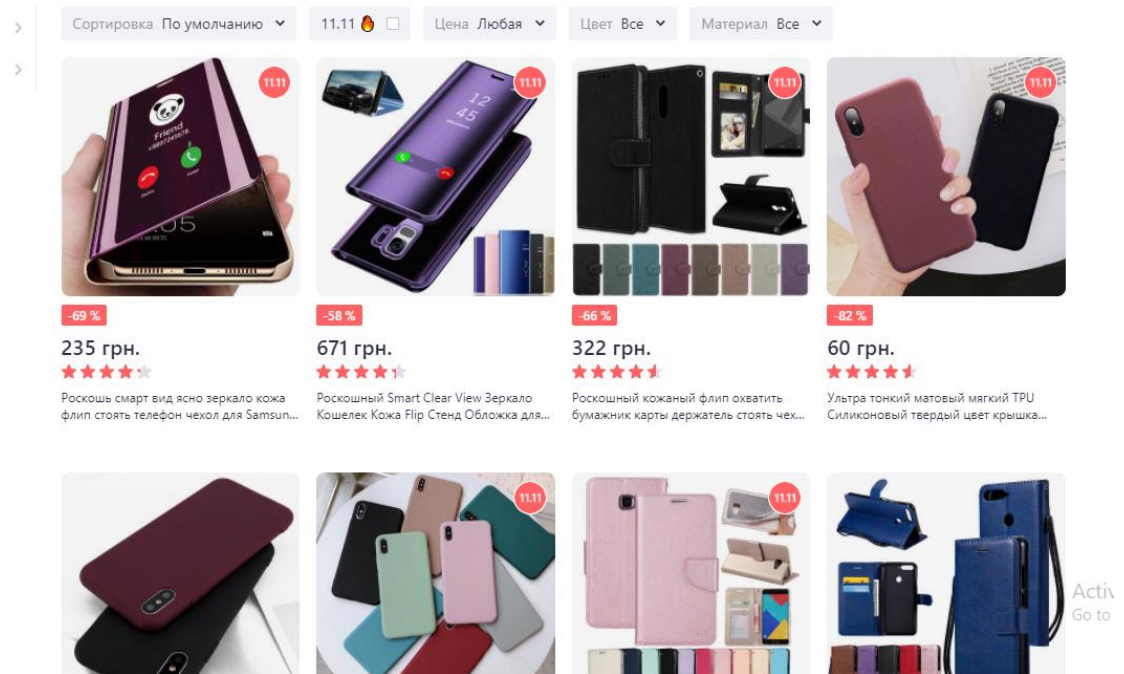


Рисунок 1.3 – Зпрогнозований каталог товарів від Joom

Також розглянемо UA-ТАО – онлайн Інтернет-магазин, орієнтований для споживачів у Китаї. Сайт працює за системою С2С, «Споживач для Споживача» – форма електронної торгівлі, яка полягає в продажі товарів та послуг між споживачами. У мережева крамниця Таобао виступає в ролі посередника між покупцем та продавцем [6].

В данному інтерне магазині, які на Aliexpress, дуже сильно страждає точність та впорядкування товару що пропонується, усі категорії перемішуються, цінова політика не відповідає очікуваному бюджету, актуальність деяких рекомендуємих товарів відсутня.

На мій погляд, серед даних сервісів, найбільш точно виконує прогнозування інтернет-магазин Joom, але й він має проблеми з точністю в наслідок недостатньої кількості аналітичних даних.

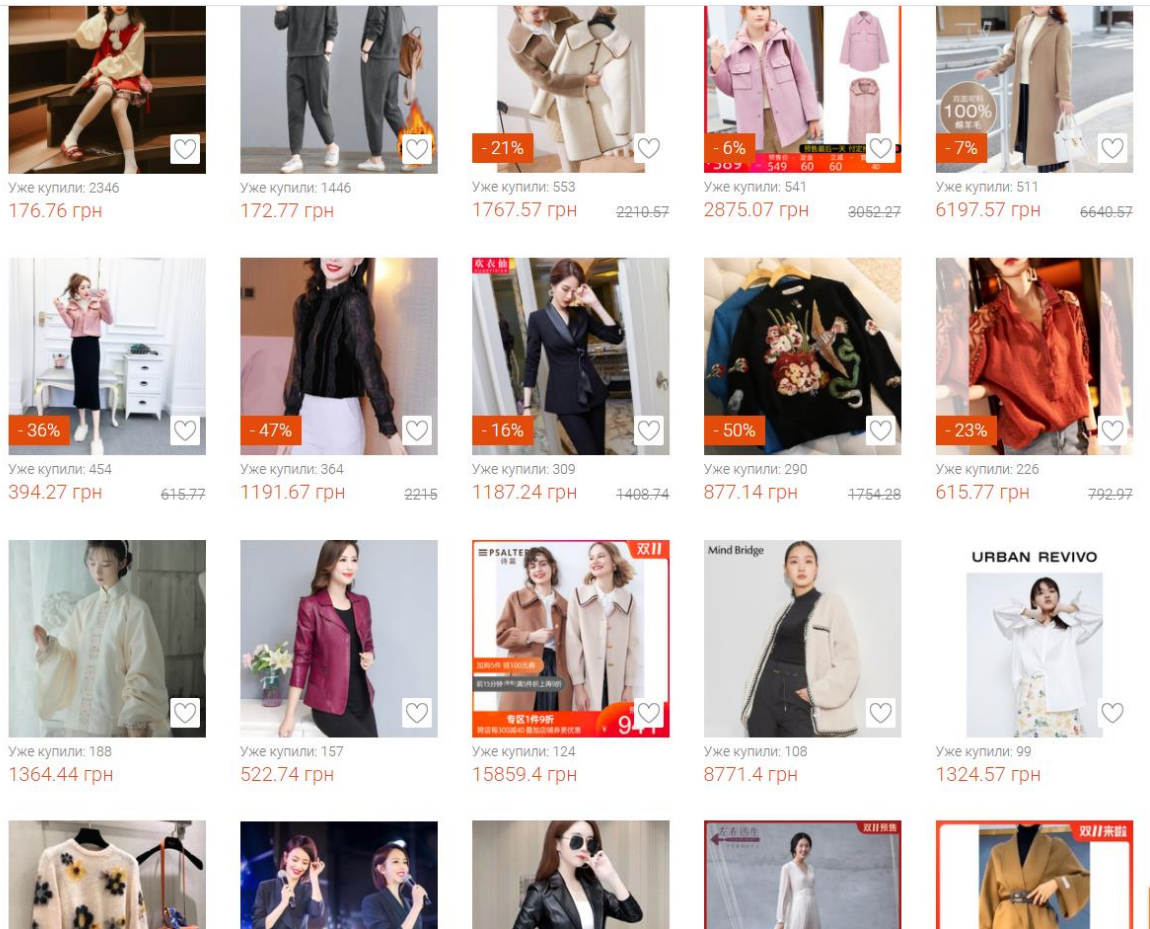


Рисунок 1.4 – Зпрогнозований каталог товарів від UA-TAO

1.3 Постановка задачі розробки інформаційної технології

У ході виконання дипломного проектування необхідно дослідити предметну область процесу прогнозування. Після цього необхідно вивчити аналоги та зробити висновок про необхідність розробки нового продукту, розробити технічне завдання для проектуємого додатка, необхідно визначити засоби розробки: операційну систему, мову й середовище програмування.

Отже, потрібно розробити інформаційну технологію інтернет-маркетингу. Розроблений програмний додаток має забезпечити користувачеві максимально точний підбір товару який може його зацікавити, на основі переглядаємих раніше товарів у цьому додатку, також додаток має забезпечити зручну аналітику для

власника цього додатку. Кожему новому користувачу буде запропоновано створити свій обліковий запис, дані кожного користувача буде збережено в хмарну базу даних. Дані про кожен переглянутий товар будуть оброблятися на стороні клієнта, та відправлятися в базу даних з повними характеристиками. За відсутності інтернет з'єднання дані будуть оброблятися та зберігатися у кеш пам'яті, а при повторному з'єднанні відправлені у базу даних.

Додаток має відповідати таким основним вимогам:

- 1) підтримуватись на платформах IOS, Android;
- 2) мати зв'язок з хмарною базою даних;
- 3) відстежувати кожен перегляд товару, та відправляти дані в базу, включаючи дані з фільтрів які використав користувач;
- 4) на основі вище зазначених даних оновлювати список товарів які можуть зацікавити користувача;

Основна складність полягає у тому щоб даний додаток не створював велике навантаження на апаратну частину девайса так як даний додаток розробляється для мобільних девайсів.

1.4 Висновки

У даному розділі проаналізовано предметну область, зазначено актуальність дослідження, визначено основну проблему під час прогнозування, що полягає в недостатній точності прогнозування набору товарів для покупок, у наслідок нехтування рядом аналітичних даних.

Виконано аналітичний огляд аналогів для вирішення технічної проблеми прогнозування набору товарів для покупок. Аналіз показав, що більшість реалізацій не враховують, зокрема, такі аналітичні дані для прогнозування, як: час переглядання товарів (з врахуванням захисту від неактивності користувача),

кількість разів переглядання конкретного товару, фільтри сортування товарів тощо. Одним із найкращих в цьому плані можна вважати програмне забезпечення Amazon, яке розбиває дані на максимально дрібні підкатегорії. Якість прогнозування в данному додатку вважається найкраща, але іноді запропоновані товари не відповідають очікуваній ціновій політиці на яку зорієнтований клієнт, особливо часто це відбувається коли користувач довгий час не користувався додатком.

Враховуючи недоліки існуючих програмних засобів прогнозування та аналітики, здійснено постановку задачі, спрямованої на розробку технології інтернет-маркетингу, що дозволяє підвищити точність прогнозування набору товарів для покупок.

2. СТРУКТУРНА ОРГАНІЗАЦІЯ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ІНТЕРНЕТ-МАРКЕТИНГУ

2.1 Аналіз існуючих нейронних мереж для розв'язання задачі прогнозування.

Під час розробки нейронної мережі прогнозування використовують часто такі архітектурні підходи як:

- навчання з вчителем (перцептрон);
- змішане навчання: (мережа радіально-баисних функцій).

Перцептрон – математична або комп'ютерна модель сприйняття інформації мозком. Перцептрон складається з трьох типів елементів, а саме: сигнали, що надходять від давачів, передаються до асоціативних елементів, а відтак до реагуючих. Таким чином, перцептрони дозволяють створити набір «асоціацій» між вхідними стимулами та необхідною реакцією на виході. У біологічному плані це відповідає перетворенню, наприклад, зорової інформації у фізіологічну відповідь рухових нейронів. Класичний метод навчання перцептрону – це метод корекції помилки. Він являє собою такий вид навчання з учителем, при якому вага зв'язку не змінюється доти, поки поточна реакція перцептрона залишається правильною. При появі неправильної реакції вага змінюється на одиницю, а знак (+/-) визначається протилежним від знаку помилки [6].

Також перцептрони мають ряд недоліків: так, їм притаманні обмеження у задачах, пов'язаних з інваріантним представленням образів, тобто незалежним від їхнього положення на сенсорному полі та положення щодо інших фігур [7].

Важливою властивістю перцептроїв є їхня здатність до навчання, причому за рахунок досить простого й ефективного алгоритму. Останнім часом

дослідники починають звертати увагу саме на оригінальну версію перцептрона, оскільки навчання багат шарового перцептрона за допомогою методу зворотного поширення помилки виявило істотні обмеження на швидкість навчання [8, 9]. Спроби навчати багат шаровий перцептрон методом зворотного поширення помилок приводять до експоненційного зростання обчислювальних витрат. Якщо ж користуватися методом прямого поширення, то обчислювальна складність алгоритму навчання стає лінійною.

Мережа радіально-базисних функцій це штучна нейронна мережа, яка використовує радіально-базисні функції у якості функції активації. Виходом мережі є лінійна комбінація радіально-базисних функцій входу та параметрів нейрона. Мережі радіально-базисних функцій (RBF) зазвичай мають три шари: вхідний шар, прихований шар з нелінійною RBF функцією активації та лінійний вихідний рівень. Мережі RBF, як правило, тренуються з пар вхідних й цільових значень, за двоетапним алгоритмом [9].

На першому етапі обирається центр вектору RBF функції у прихованому шарі. Цей етап виконується кількома способами; центри можуть бути випадково відібрані з деякого набору прикладів, або їх можна визначити за допомогою кластеризації методом k-середніх [10].

Радіально-базисні мережі характеризуються трьома особливостями:

- єдиний прихований шар;
- тільки нейрони прихованого шару мають нелінійну активаційну функцію;
- синаптичні ваги зв'язків вхідного та прихованого шарів дорівнюють одиниці. Дані мережі мають ряд переваг:

– вони моделюють довільну нелінійну функцію за допомогою всього одного проміжного шару, тим самим позбавляючи розробника від необхідності вирішувати питання про кількість шарів;

– параметри лінійної комбінації в вихідному шарі можна повністю оптимізувати за допомогою добре відомих методів лінійної оптимізації, які працюють швидко й не відчують труднощів з локальними мінімумами, так заважають при навчанні з використанням алгоритму зворотного поширення помилки [11].

Як й будь-які нейроні мережі RBF мережі мають свої недоліки. Зокрема, вони володіють поганими екстраполюється властивостями й виходять вельми громіздкими при великій розмірності вектора входів [12].

У ході аналізу даних методів, опираючись на недоліки кожного з них, було прийнято рішення взяти за основу модель перцептрона. Так як однією з основних параметрів є швидкодія додатка.

Під час вирішення задачі прогнозування за допомогою перцептрона нейромережева система будується таким чином: вхідний шар містить декілька нейронів на які подаються значення досліджуваного часовго ряду, а останній шар складається з єдиного нейрона на виході якого отримується прогноз. Але недоліком даного алгоритму є достатньо швидке накопичення помилки [13].

Поліпшення точності прогнозу відбувається за шляхом поєднання вихідних нейронів між собою так, щоб отримане значення на першому вихідному нейроні подавалося на вхід другого вихідного нейрона, а отримане на першому й другому – враховувалося на третьому тощо [14].

Іншими словами, на кожен наступний нейрон вихідного шару, крім сигналів з нейронів передостаннього шару, повинні подаватися сигнали, вже отримані на попередніх виходах мережі.

Приклад нейронного перцептрона без поєднання вихідних нейронів між собою наведено на рисунку 2.1 [15].

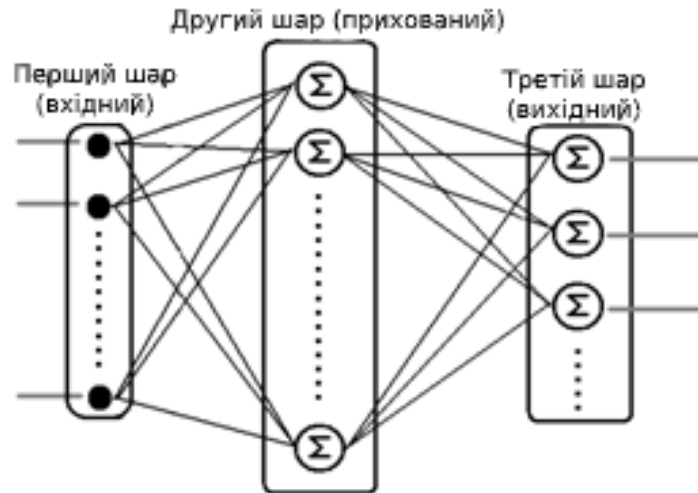


Рисунок 2.1 – Будова нейронного перцептрона без поєднання вихідних нейронів між собою

Приклад нейронного перцептрона з поєднанням вихідних нейронів між собою наведено на рисунку 2.2 [16].

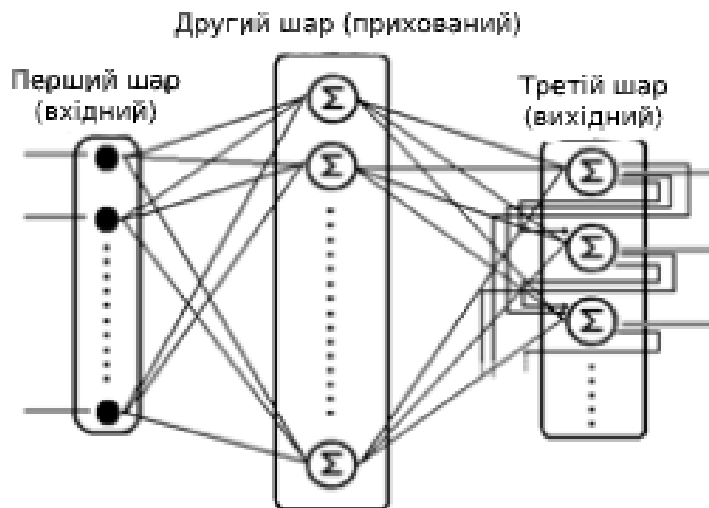


Рисунок 2.2 – Будова нейронного перцептрона з поєднанням вихідних нейронів між собою

2.2 Обґрунтування вибору типу нейронної мережі для роботи з бізнес-прогнозуванням

2.2.1 Обґрунтування вибору нейронного перцептрона з поєднанням вихідних нейронів

Як було зазначено вище, за допомогою поєднання зв'язків нейронів між собою так, щоб на кожен наступний нейрон вихідного шару, крім сигналів з нейронів передостаннього шару, повинні подаватися сигнали, вже отримані на попередніх виходах мережі.

Процес навчання данної мережі виконується ітераційно, у відповідності з алгоритмом зворотного поширення помилки.

На першому етапі кожної ітерації дані чергового навчального прикладу надходять на нейрони вхідного шару й поширюються від першого шару до останнього, при цьому вихідне значення кожного нейрона обчислюється за формулою (2.1) [17]

$$OUT_q = f_a(\sum_{p=1}^{N+\sigma} OUT_p w_{pq}), \quad (2.1)$$

де OUT_q OUT_p – вихідні значення нейронів q та p відповідно; f_a – функція активації; σ – множина додаткових аналітичних даних; w_{pq} – ваговий коефіцієнт зв'язку між нейронами p та q .

На другому етапі ітерації навчання відбувається перерахунок вагових коефіцієнтів нейронних зв'язків за формулою (2.2) [18]:

$$w_{pq}(i + 1) = w_{pq}(i) + n\delta_q OUT_p, \quad (2.2)$$

де $w_{pq}(i + 1)$ – нове значення вагового коефіцієнта зв'язку між нейронами p й q ;

w_{pq} – старе значення вагового коефіцієнта p q – зв'язку; n – швидкість навчання;
 σ – множина додаткових аналітичних даних; δ_q – дельта-коефіцієнт нейрона q ;
 OUT_p – вихідне значення нейрона p .

Перерахунок здійснюється, починаючи з останнього шару й закінчуючи першим [19].

Дельта-коефіцієнт, який бере участь у розрахунку вагових значень, для вихідного шару обчислюється за формулою (2.3), а для прихованих шарів за формулою (2.4)

$$\delta_q = OUT_q(1 - OUT_q)(\mu_q - OUT_q); \quad (2.3)$$

$$\delta_q = OUT_q(1 - OUT_q) \sum_{p=1}^{N+\sigma} OUT_p w_{pq}, \quad (2.4)$$

де OUT_q, OUT_p – вихідні значення нейронів q й p відповідно; σ – множина додаткових аналітичних даних; μ_q – істинне значення; w_{pq} – ваговий коефіцієнт зв'язку між нейронами p й q [20].

Як наведено на рисунку 2.3 кількість помилок у перцептрона з внутрішніми зв'язками між нейронами вхідного шару менша ніж у звичайного перцептрона [21].

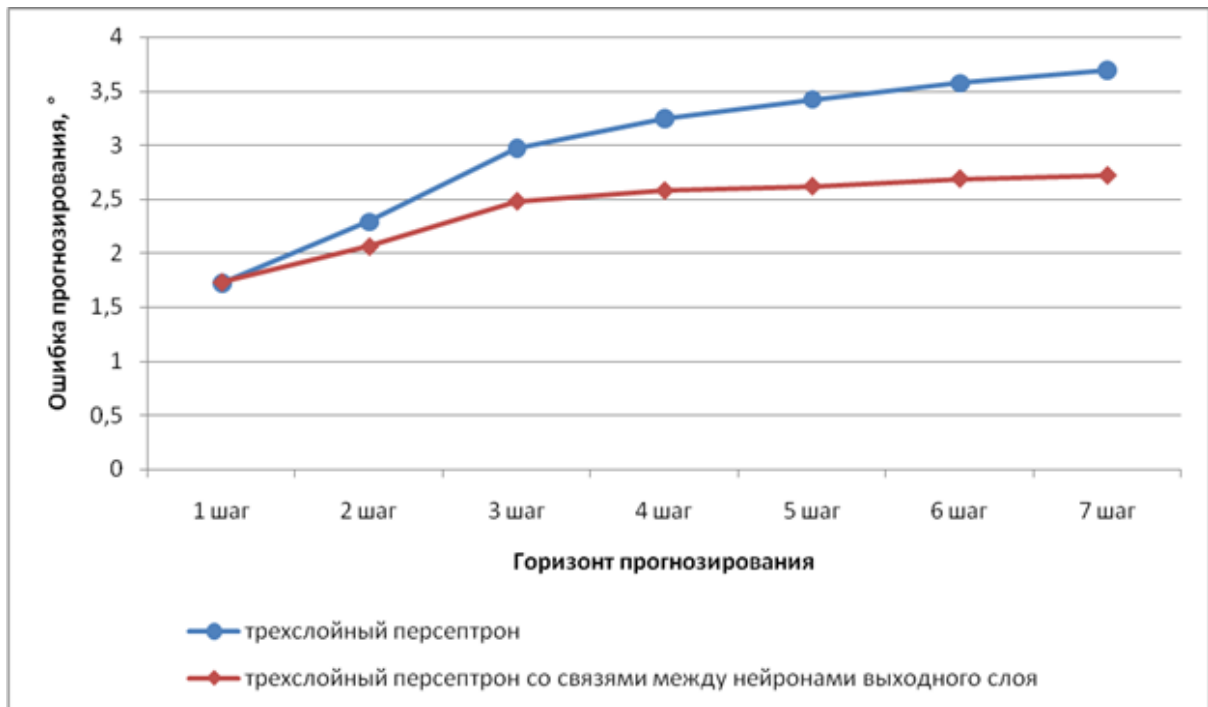


Рисунок 2.3 – Графік залежності помилки прогнозування від кроку прогнозування

Дана модель перцептрона найкраще підійде для навчання з даними дуже великого обсягу даними гарантуючи при цьому максимальну точність, та мінімальну просадку в продуктивності.

2.2.2 Аналіз та обґрунтування вибору методу для вирішення проблеми перенавчання

Одна з проблем глибоких нейронних мереж проявляється в наступному: модель добре пояснює тільки приклади з навчальної вибірки, адаптуючись до навчальних прикладів, замість того щоб вчитися класифікувати приклади, а не брати участі в навчанні.

На рисунку 2.4 наведено приклад перенавченої мережі де зелена лінія являє собою перенавчену модель, а чорна – регуляризовану модель. Хоча зелена лінія найточніше слідує за навчальними даними, вона надто залежить від цих даних і, ймовірно, матиме вищий рівень помилок щодо нових невидимих даних, порівняно з чорною лінією [22].

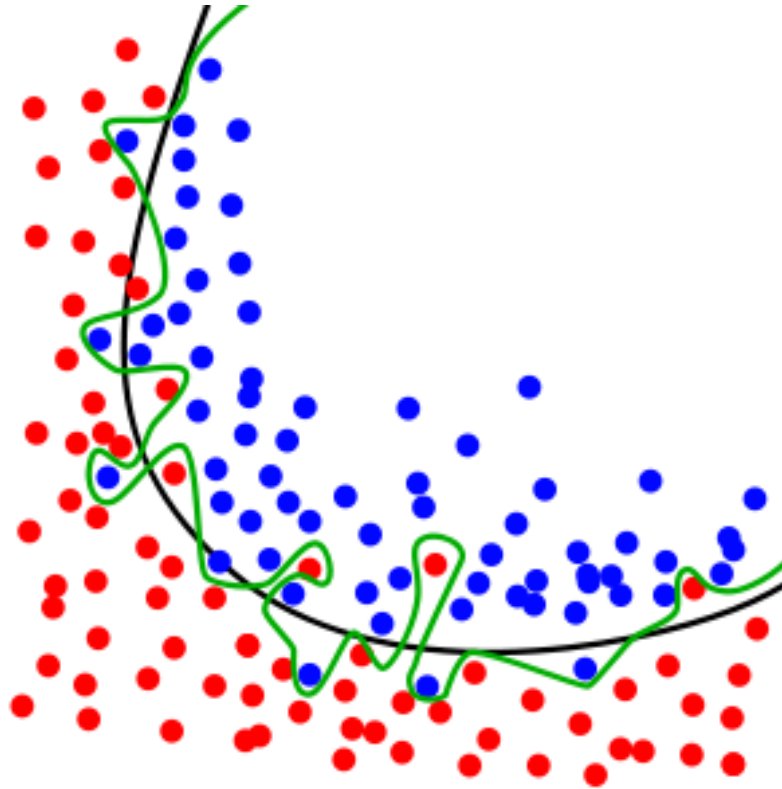


Рисунок 2.4 – Приклад методу виключення

Зазвичай алгоритм навчання навчається з використанням певного набору навчальних даних: зразкових ситуацій, для яких відомий бажаний результат. Мета полягає в тому, що алгоритм також буде добре працювати при прогнозуванні результату роботи при подачі перевірочних даних, які не зустрічалися під час навчання [23].

Перенавчання – це використання моделей або процедур, які порушують лезо Оккама, наприклад, включаючи більше регульованих параметрів, ніж є в кінцевому рахунку оптимальними, або використання більш складного підходу,

ніж є остаточно оптимальним. Для прикладу, коли забагато регульованих параметрів, розглянемо набір даних, де дані навчання для у можна адекватно передбачити за допомогою лінійної функції двох незалежних змінних. Для такої функції потрібні лише три параметри (перехоплення та два нахили). Заміна цієї простої функції новою, більш складною квадратичною функцією або новою, більш складною лінійною функцією для більш ніж двох незалежних змінних, несе ризик: лезо Оккама передбачає, що будь-яка дана складна функція апіорі менш вірогідна, ніж будь-яка проста функція [24].

Якщо замість простої функції обрано нову, більш складну функцію, та якщо не було достатньо великого виграшу в навчальних даних, придатних для компенсації збільшення складності, то нова складна функція "переставляє" дані, а комплекс переоснащує функція, швидше за все, буде працювати гірше, ніж простіша функція для даних перевірки поза навчальним набором даних, навіть якщо складна функція, що виконується також, або, можливо, навіть краще, на навчальному наборі даних [25].

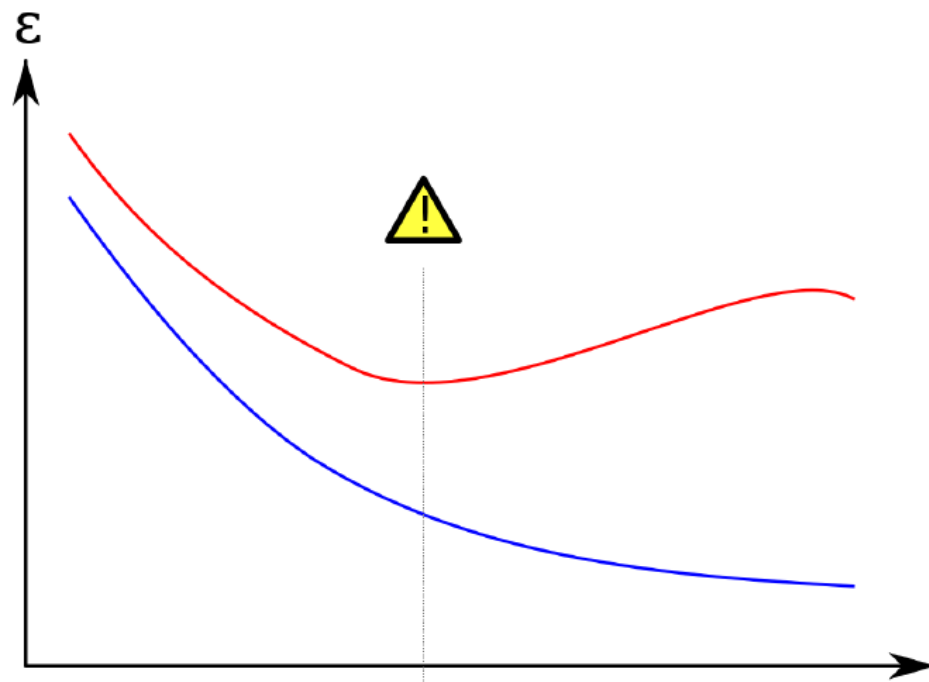


Рисунок 2.5 – Приклад перенавчання

На сьогодні існує досить велика кількість рішень проблеми перенавчання.

Для аналізу доцільно розглянути такі рішення:

- батч – нормалізація;
- рання зупинка;
- dropout.

Проаналізуємо переваги та недоліки кожного з цих рішень.

Батч-нормалізація пропонує просте рішення даної проблеми: нормалізувати вхідні дані так, щоб отримати нульове математичне очікування та одиничну дисперсію. Перед входом в кожен шар виконується нормалізація. Це означає, що під час навчання нормалізується розмір пакетів прикладів, а під час тестування виконується нормалізація статистики, отримана на основі всьої навчальної множини, так як заздалегідь побачити тестові дані немає можливості. А саме, обчислюється математичне сподівання й дисперсія для певного Батча (пакета) у такий спосіб [25]:

$$\mu\beta = \frac{1}{m} \sum_{i=1}^m x_i; \quad (2.5)$$

$$\sigma \frac{2}{\beta} = \frac{1}{m} \sum_{i=1}^m (x_i - \mu\beta)^2. \quad (2.6)$$

де $\mu\beta$ – математичне очікування пакета; $\sigma \frac{2}{\beta}$ – дисперсія пакета; x – значення з пакету $B = \{x_1, \dots, x_m\}$ [25].

За допомогою цих статистичних даних ми перетворимо функцію активації таким чином, щоб вона мала нульове маточікування та одиничну дисперсію на всьому Батчі

$$x_i = \frac{x_i - \mu\beta}{\sqrt{\sigma \frac{2}{\beta} + \epsilon}}, \quad (2.7)$$

де $\varepsilon > 0$ – параметр, що захищає нас від ділення на 0 (у разі, якщо середньоквадратичне відхилення Батча дуже мале або навіть дорівнює нулю) [25].

Для того щоб отримати функцію активації, нам необхідно переконатися, що ми не втратили здатності до узагальнення під час нормалізації, і так як до вихідних даних ми застосували операції масштабування та зсуву, ми маємо змогу дозволити довільне масштабування та зрушення нормалізованих значень, отримавши остаточну функцію активації:

$$y_i = \gamma x_i + \beta, \quad (2.8)$$

де β та γ – параметри Батч-нормалізації, яким системи можна навчити (їх можна оптимізувати методом градієнтного спуску на навчальних даних) [28].

Dropout – головна ідея даного методу полягає у тому, щоб замість навчання однієї DNN навчити ансамбль декількох DNN, а потім усереднити отримані результати.

Мережі для навчання виходять за допомогою виключення з мережі (dropping out) нейронів з ймовірністю p , таким чином, вірогідність того, що нейрон залишиться в мережі, становить $q = 1 - p$ [25]. "Виключення" нейрона означає, що при будь-яких вхідних даних або параметрах він повертає 0.

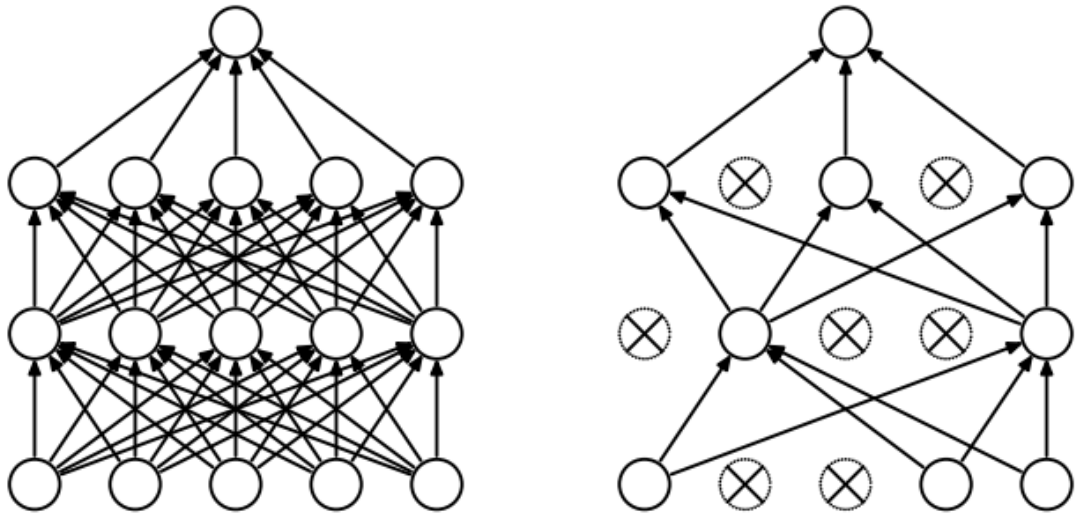


Рисунок 2.4 – Приклад методу виключення

2.2.3 Обґрунтування вибору реверсивного Dropout методу нормалізації

Як зазначалося раніше головна ідея Dropout – замість навчання однієї DNN навчити ансамбль декількох DNN, а потім усереднити отримані результати.

Мережі для навчання створюються за допомогою виключення з мережі (dropping out) нейронів з ймовірністю, таким чином вірогідність того, що нейрон залишиться в мережі становить $q = 1 - p$ [25, 26]. "Виключення" нейрона означає, що при будь-яких вхідних даних або параметрах він повертає 0.

Виключені нейрони не вносять свій вклад в процес навчання ні на одному з етапів алгоритму зворотного поширення помилки (back propagation), тому виключення хоча б одного з нейронів рівносильно навчання нової нейронної мережі.

Dropout виключає нейрони з вірогідністю p і, як наслідок, залишає їх включеними з ймовірністю

$$q = 1 - p \quad (2.9)$$

Ймовірність виключення кожного нейрона однакова [30]. Це означає наступне.

За умови умові, що

- $h(x)=xW+b$ — лінійна проекція вхідного d_i -мірного вектора x на d_h -мірний простір вихідних значень;
- $a(h)$ – функція активації;

Застосування Dropout до даної проекції на етапі навчання можна уявити як змінену функцію активації [26, 27]

$$f(h) = D \odot a(h), \quad (2.10)$$

де $D = (X_1, \dots, X_{d_h})$ – d_h мірний вектор випадкових величин X , розподілених за законом Бернуллі; \odot - матриця Адамара

X – має такий розподіл ймовірностей:

$$f(k; p) = \begin{cases} p, & \text{if } k = 1 \\ 1 - p, & \text{if } k = 0 \end{cases}, \quad (2.11)$$

де k – усі ймовірні вихідні значення

Очевидно, що ця випадкова величина ідеально відповідає Dropout, застосованого до одного нейрона [26, 27]. Дійсно, нейрон вимикають з $p = P(k = 1)$ ймовірністю в іншому випадку – залишають включеним.

Подивимося на застосування Dropout до i -му нейрона:

$$O_i = X_i a \left(\sum_{k=1}^{d_i} w_k k_x + b \right) = \begin{cases} a \left(\sum_{k=1}^{d_i} w_k k_x + b \right), & \text{if } X_i = 1 \\ 0, & \text{if } X_i = 0 \end{cases}, \quad (2.12)$$

Оскільки на етапі навчання нейрон залишається в мережі (не береться виключення) з ймовірністю q , на етапі тестування нам необхідно емулювати

поведінка ансамблю нейронних мереж, використаного при навчанні. Для цього автори пропонують на етапі тестування помножити функцію активації на коефіцієнт q [26, 27]. Таким чином

$$\text{На етапі навчання } O_i = X_i a \left(\sum_{k=1}^{d_i} w_k k_x + b \right) \quad (2.12)$$

$$\text{На етапі тестування } O_i = qa \left(\sum_{k=1}^{d_i} w_k k_x + b \right) \quad (2.13)$$

Можливо використовувати трохи інший підхід – зворотний Dropout. В даному випадку ми множимо функцію активації на коефіцієнт не під час тестового етапу, а під час навчання [32].

$$\text{На етапі навчання } O_i = \frac{1}{q} X_i a \left(\sum_{k=1}^{d_i} w_k k_x + b \right) \quad (2.14)$$

$$\text{На етапі тестування } O_i = a \left(\sum_{k=1}^{d_i} w_k k_x + b \right) \quad (2.15)$$

2.2.4 Розробка архітектури мобільного додатку для інтернет-маркетингу

Даний додаток можна розділити на 5 модулів. Структурна схема функціонування мобільного додатку інтернет-магазину з використанням бізнес прогнозування зображена на рисунку 2.2.

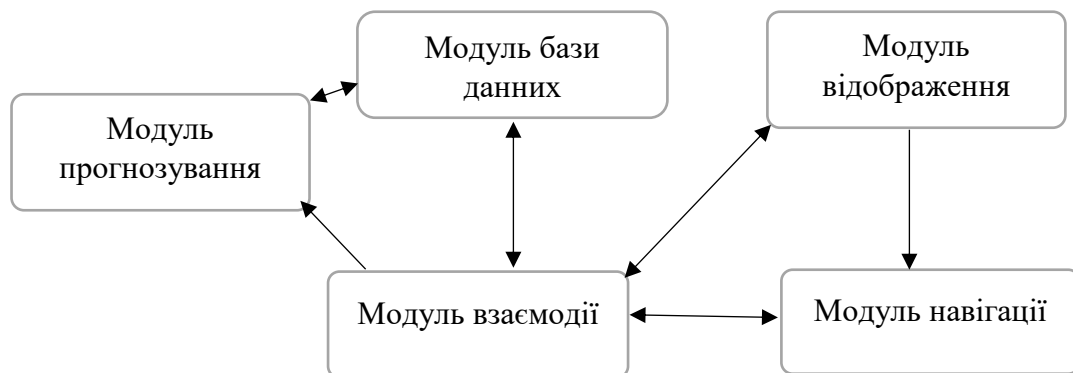


Рисунок 2.5 – Загальна схема роботи алгоритму програмного засобу інформаційної технології інтернет-маркетингу

Модуль бази даних виконує обов'язки структури додатку (збереження інформації, процес авторизації). Даний модуль відповідає за видачу, запис, збереження даних. База даних реалізується завдяки «Firebase». Firebase – набір інструментів серед яких є база даних NoSQL, різні технології авторизації, сервіс для зберігання медіа-файлів та ін [28]. Даний модуль взаємодіє з модулем взаємодії додатку.

Модуль відображення відображає користувацький інтерфейс та результати взаємодії з іншими модулями. Даний модуль працює та взаємодіє з модулем взаємодії та модулем навігації.

Модуль навігації відповідає за навігацію додатку. Завдяки ньому користувач має змогу переключатися між екранами додатка. Він реалізований завдяки бібліотеці React Native Navigation [29].

Модуль взаємодії відповідає за взаємодію усіх модулів. Даний модуль реалізується завдяки технології Redux, на рисунку 2.3 представлена схема роботи технології.

Модуль прогнозування відповідає за обробку даних отриманих з бази даних , та видачу результатів прогнозу.

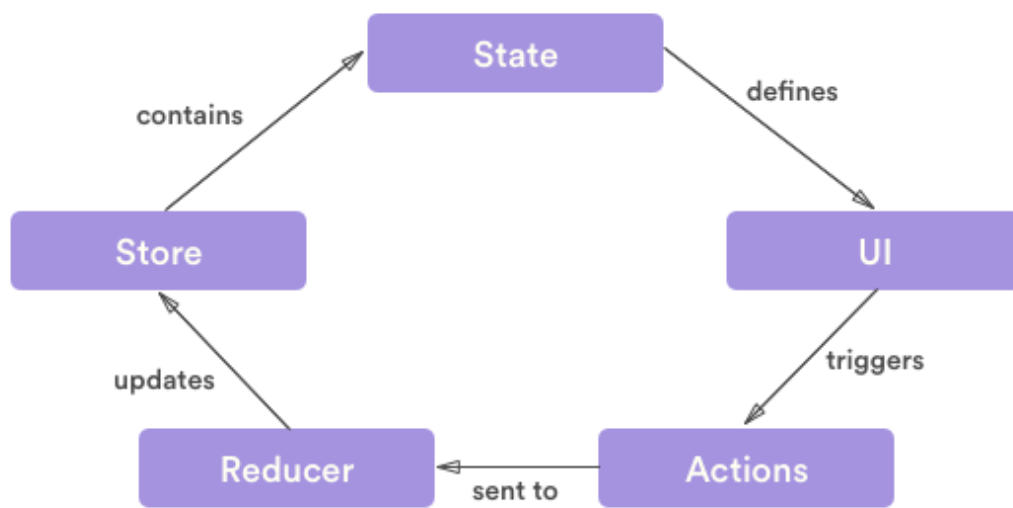


Рисунок 2.6 – Схема роботи технології Redux

Під час взаємодії з додатком, користувач викликає деякий action, який відсилається в reducer.

Action – функція, яка повертає об'єкт, який містить назву та деякі дані, які необхідно внести в store.

Reducer – функція, яка в свою чергу, по назві вибирає необхідний action, та оновлює store, додаючи дані з action.

Store – загальний стан додатку, він повністю перезаписується кожен раз, коли оновлюється.

Після оновлення store, оновлюються state UI компонентів, які «підписані» на зміну store.

Даний модуль взаємодіє з усіма модулями додатку.

На рисунку 2.7 зображено діаграму діяльності мобільного додатку інтернет-магазину з використанням модуля бізнес-прогнозування.

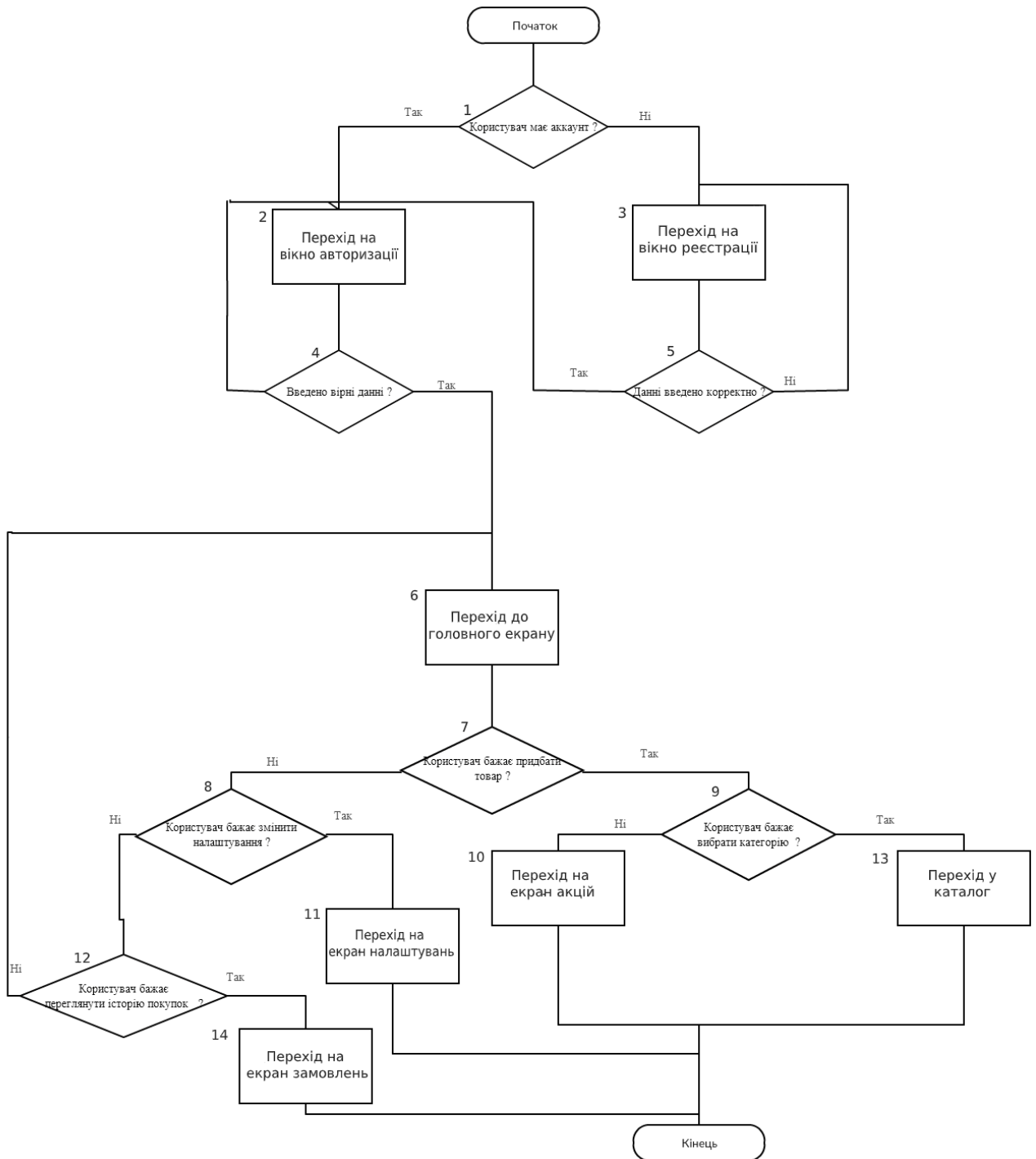


Рисунок 2.7 – Загальний алгоритм роботи мобільного додатку інтернет-магазину з використанням модуля бізнес-прогнозування

Дана діаграма відображає основні дії, які користувач може виконати користуючись додатком.

На початку потрібно розробити процес авторизації та реєстрації. Якщо користувач має запис в базі даних, то він вводить свої дані (логін та пароль) та відправляє запит на авторизацію, якщо дані введено вірно, він попаде на головний екран програми. Якщо користувач не має акаунта, тоді він переходить у вкладку реєстрації, вводить дані та відправляє запит на створення запису. Після цього він потрапляє на вікно авторизації.

Після того як користувач потрапляє на головний екран, він може придбати товар, змінити свої дані, подивитися історію операцій або вилогінитися.

Якщо користувач захоче придбати товар, у нього є вибір де він може це зробити. Якщо користувач захоче вибрати якусь певну категорію товарів, йому необхідно перейти у вікно каталогу. Також користувач має змогу придбати товар прямо з головного екрану, де відображаються нові товари.

Якщо користувач захоче переглянути список своїх придбань, він може перейти у вікно історії покупок.

Якщо користувач захоче змінити свої дані або вилогінитися, він може перейти у вікно налаштувань та виконати усі необхідні операції.

2.3 Розробка структури інформаційної технології інтернет-маркетингу.

На першому етапі користувач повинен пройти процедуру авторизації, в додаток, під час якої отримуємо дані про вже зпрогнозовані товари. Якщо користувач не зареєстрований, то він зобов'язаний пройти процедуру реєстрації під час якої створиться обліковий запис в базі даних.

Після цього починається процес відслідковування дій користувача, для збору аналітичних даних, запам'ятовуються усі перегляди товарів, включаючи фільтри які були в цей час застосовані.

Наступним кроком аналітичні дані подаються на нейронну мережу для отримання зпрогнозованих даних.

Після отримання зпрогнозованих даних, вони відправляються у базу даних для й оновлюють список рекомендацій для даного користувача.

Останнім кроком йде процес оновлення списку рекомендуємих товарів на девайсі користувача.

Структура інформаційної технології інтернет-маркетингу зображено на рисунку 2.8.

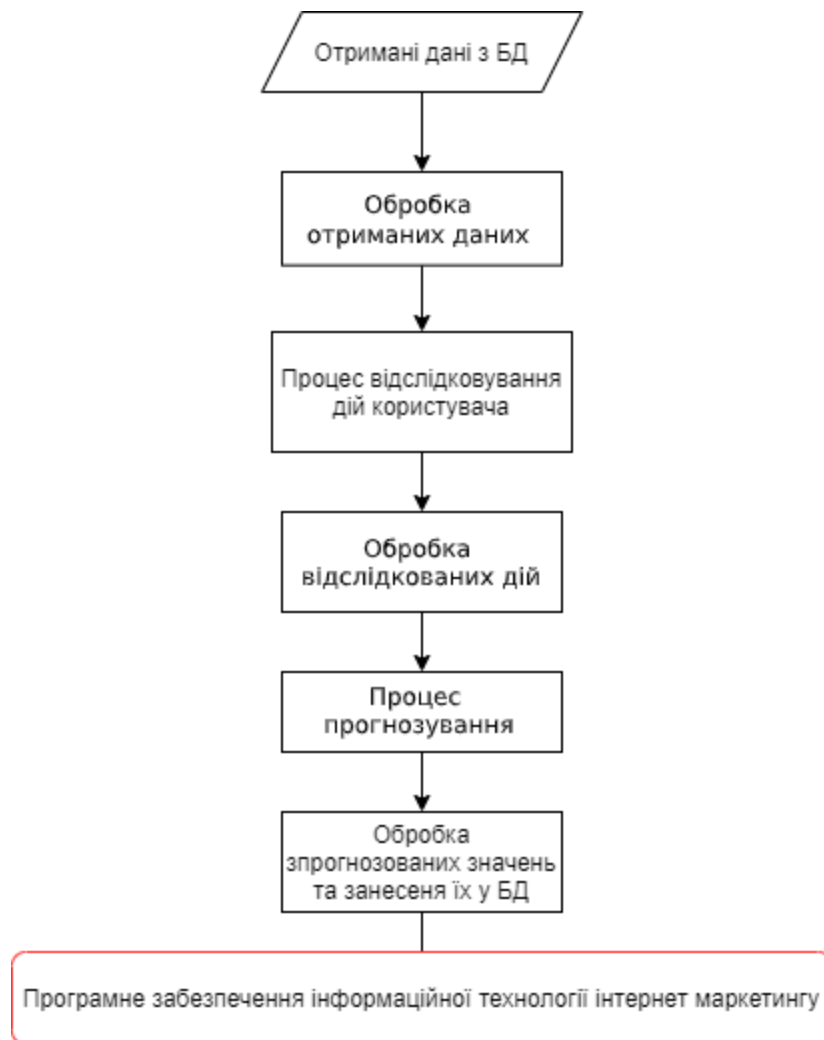


Рисунок 2.8 – Структура інформаційної технології інтернет-маркетингу

2.4 Висновки

У другому розділі було обґрунтовано доцільність застосування нейронних мереж для розв'язання задачі прогнозування. Розглянуто та проаналізовано існуючі нейронні мережі для розв'язання задачі прогнозування. Найбільш використовувані з яких – перцептрон та мережа радіально-баисних функцій. На основі виконаного аналізу було обрано модель перцептрона із поєднанням вихідних нейронів, оскільки ця модель володіє кращими властивостями екстраполювання та виходить не такою громіздкою при великій розмірності вектора входів.

Запропоновано математичну модель прогнозування очікуваних покупок для інформаційної технології інтернет-маркетингу, яка відрізняється врахуванням додаткових аналітичних даних, що характеризують як товари так і дії користувачів під час використання додатка. Це дозволить збільшити точність прогнозування.

Проаналізовано існуючі методи вирішення проблеми перенавчання (батч-нормалізація, метод ранньої зупинки, dropout). На основі виконаного аналізу обґрунтовано вибір методу реверсивного Dropout, оскільки він показав найкращі результати.

Розроблено алгоритм функціонування інформаційної технології інтернет-маркетингу, що передбачає використання перцептрона із поєднанням вихідних нейронів.

Розроблено загальну структурну схему інформаційної технології інтернет-маркетингу, що містить 7 етапів та визначає послідовність процесів, що відбуваються під час функціонування додатку.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ІНТЕРНЕТ-МАРКЕТИНГУ

3.1 Обґрунтування вибору мови програмування

Під час вибору засобів для розробки програмного продукту необхідно врахувати велику кількість аспектів, найбільш важливим з них є мова програмування.

Так як даний додаток розробляється для мобільних девайсів, тому слід розглядати мови програмування які створені спеціально для даних типів додатків.

На сьогоднішній день , для розробки мобільних додатків використовують такі основні мови програмування: Java, Swift/Objective-C, JavaScript.

Java – об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році як основний компонент платформи Java. Зараз мовою займається компанія Oracle, яка придбала Sun Microsystems у 2009 році. Синтаксис мови багато в чому походить від C та C++. У офіційній реалізації, Java програми компілюються у байткод, який при виконанні інтерпретується віртуальною машиною для конкретної платформи. Oracle надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License [30]. Дана мова програмування використовується для створення додатків під операційну систему Android.

Android – операційна система та платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux. Підтримується альянсом Open Handset Alliance (ОНА) [31].

Swift – багатопарадигмова компільована мова програмування, розроблена компанією Apple для того, щоб співіснувати з Objective C та бути стійкішою до

помилкового коду. Swift була представлена на конференції розробників WWDC 2014. Мова побудована з LLVM компілятором, включеного у Xcode 6 beta [32].

Objective-C — рефлексивна, високорівнева об'єктно-орієнтована мова програмування загального призначення, розроблена у вигляді набору розширень стандартної C [33]

Мови програмування Objective-C та Swift використовується для створення додатків під операційну систему iOS.

iOS – це власницька мобільна операційна система від Apple. Розроблена спочатку для iPhone, згодом також вдосконалена для використання на iPad (до літа 2019, коли на конференції Apple WWDC було представлено нову OS для iPad — iPadOS), iPod Touch та Apple TV (до 9 вересня 2015, коли на спеціальному заході Apple було представлено tvOS). Apple не дозволяє роботу ОС на мобільних телефонах інших фірм. [34]

JavaScript – динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв вебсторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки [35].

Основне призначення JavaScript це робота з веб-ресурсами, а точніше створення клієнтської частини додатка, але у наш час дана мова використовується в усіх сферах: Front-end, Back-end, кросплатформні розробки, створення нейронних мереж, тощо. За допомогою JS створена навіть операційна система JsOS.

Одним із основних критеріїв даного додатка є підтримка на всіх типах мобільних девайсів, тому слід звзунити критерій вибору з мов які підтримують кросплатформну розробку.

Як було зазначено раніше, одна з мов програмування яка підтримує кросплатформну розробку це JavaScript. Крім того для створення кросплатформних додатків також використовують Xamarin та Flutter.

Flutter це програмний каркас із відкритим кодом, для створення додатків для платформ Android та iOS, а також на веб, розроблений компанією Google. Він є основним способом створення додатків для Google Fuchsia. Весь графічний інтерфейс Google Fuchsia створено за допомогою Flutter [36].

Xamarin – американська компанія в галузі розробки ПЗ. Займається розробкою й підтримкою Mono та інструментів для розробки застосунків мовою C# для iOS, Android, Windows, Mac [37].

Для розробки даного додатка, також необхідно орієнтуватися на мову яка підтримує бібліотеки для розробки нейронних мереж. Найбільш якісні бібліотеки для розробки нейронних мереж наявні для мови програмування JavaScript.

На основі проаналізованих варіантів мов програмування було обрано мову JavaScript, так як вона підтримує кросплатформну розробку та велику кількість якісних бібліотек для створення нейронних мереж.

3.2 Обґрунтування вибору бібліотеки для створення нейронної мережі

Javascript бібліотеки використовують для визначення, навчання та запуску моделей глибокого навчання, візуалізації даних повністю в браузері. Вони значно полегшують життя розробнику.

Для вибору необхідної бібліотеки буде проаналізовано такі бібліотеки: Brain.js, Synaptic, Neataptic, ConventJs, Tensorflow, Neuro.js.

Brain.js – Javascript бібліотека для штучних нейронних мереж, а також заміна «мозкову» бібліотеку, яка пропонує різні типи мереж залежно від конкретних завдань. Використовується з Node.js або в браузері.

Synaptic – JavaScript бібліотека для нейронних мереж для node.js та браузера, яка дозволяє навчати архітектури нейронних мереж першого та другого порядків. Проект містить декілька вбудованих архітектур – багатошаровий перцептрон, багатошарова мережа довгої короткостроковій пам'яті, LSM (liquid state machine) й тренер (trainer), здатний навчати мережі.[38]

Neataptic – ця бібліотека надає можливість швидко здійснювати нейроеволюція та зворотне поширення для браузера та Node.js. Бібліотека містить кілька вбудованих мереж – перцептрон, LSTM, GRU, Nark та інші [38].

ConventJs – Ця популярна бібліотека, розроблена PhD студентом зі Стенфорда Андрієм Карпат, який зараз працює в Tesla. Хоча вона не підтримується останні 4 роки, Conventjs залишається одним з найцікавіших проектів у цьому списку. Conventjs вдає із себе написану на Javascript реалізацію нейронних мереж, що підтримує поширені модулі – класифікацію, регресію, експериментальний модуль навчання з підкріпленням. За допомогою цієї бібліотеки можна навіть навчати сверточноє нейросеть для обробки зображень [38].

Tensorflow – Бібліотека від Google (наступниця популярної deeplearnjs) дає можливість навчати нейронні мережі в браузері або запускати предобученние моделі в режимі виведення. Творці бібліотеки стверджують, що вона може бути використана як NumPy для вебу. Tensorflow.js з простим в роботі API може бути використана в різноманітних корисних додатках. Бібліотека також активно підтримується [39].

NeuroJs – Цей проект являє собою Javascript бібліотеку глибокого навчання та навчання з підкріпленням в браузері. Через реалізації повнофункціонального фреймворка машинного навчання на основі нейронних мереж з підтримкою навчання з підкріпленням, Neuro.js вважається наступником Conventjs [38].

На основі проаналізованих даних було прийнято рішення про використання бібліотеки Tensorflow, так як вона має асиміляцію з продуктами Google , такими як Firebase , так ідеально підходить для вирішення задач бізнес-прогнозування.

3.3 Обґрунтування вибору середовища розробки

Середовище розробки також грає важливу роль в процесі розробки програмного продукту, адже кожна IDE , або навіть текстовий редактор , мають певний встроений набір функціоналу та можливість його розширення.

Для розробки мовою JavaScript або TypeScript – типізований JavaScript, найчастіше використовують середовище для веб-розробки такі як: WebStrom, Visual Studio Code, IntelliJ IDE, якщо наявний MacBook то найчастіше використовують Atom.

Так як даний додаток є кросплатформним, то необхідно вибрати саме таке середовище , яке максимально обладнано для даної задачі.

Необхідне середовище повинно містити в собі:

- 1) Інструменти для роботи з Git, або можливість їх встановити. Git – це система контролю версій, вона допомагає оптимізувати , та структурувати розробку на логічні етапи, також дана система дозволяє чітко відслідкувати усі кроки розробки та при потребі вімінити деякі.

- 2) Інструменти для роботи з білдерами. Білдери – це проміжне програмне забезпечення яке дозволяє автоматизувати процес збору та встановлення додатка на девайс.
- 3) Підтримка автозаповнення та перевірка синтаксису кода. Це одна з найнеобхідніших речей, так як це значно пришвидшує написання коду, та зменшує кількість потенційних помилок.
- 4) Зручний дебагінг коду, так як під час розробки програмного продукту , виникає значна кількість логічних та синтаксичних помилок. Зручний дебагінг коду допомагає в найкоротші строки ідентифікувати та виправити недоліки.

Основні середовища розробки які підлягають під **дані** критерії: IntelliJ IDE, VS Code, WebStorm, Atom.

IntelliJ IDEA — комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP та ін.) від компанії JetBrains. Система поставляється у вигляді урізаної по функціональності безкоштовної версії «Community Edition» та повнофункціональної комерційної версії «Ultimate Edition», для якої активні розробники відкритих проектів мають можливість отримати безкоштовну ліцензію. Сирцеві тексти Community-версії поширюються рамках ліцензії Apache 2.0. Бінарні збірки підготовлені для Linux, Mac OS X й Windows.[40]

Visual Studio Code — засіб для створення, редагування та зневадження сучасних веб-застосунків та програм для хмарних систем. Visual Studio Code розповсюджується безкоштовно та доступний у версіях для платформ Windows, Linux та OS X.[41]

JetBrains WebStorm — інтегроване середовище розробки для JavaScript, HTML та CSS від компанії JetBrains, розроблена

на основі платформи IntelliJ IDEA. WebStorm є спеціалізованою версією PhpStorm, пропонуючи підмножину з його можливостей. WebStorm постачається з перед-установленим плагінами JavaScript (такими як для Node.js), котрі доступні для PhpStorm безкоштовно.

WebStorm підтримує мови JavaScript, CoffeeScript, TypeScript та Dart.

WebStorm забезпечує автодоповнення, аналіз коду на льоту, навігацію по коду, рефакторинг, зневадження та інтеграцію з системами управління версіями. Важливою перевагою інтегрованого середовища розробки WebStorm є робота з проектами (у тому числі, рефакторинг коду JavaScript, що міститься в різних файлах та бібліотеках проекту, а також вкладеного в HTML). Підтримується множинна вкладеність (коли в документ на HTML вкладений скрипт на Javascript, в який вкладено інший код HTML, всередині якого вкладений Javascript) — в таких конструкціях підтримується коректний рефакторинг. [41]

Atom — розроблений компанією «GitHub» вільний текстовий редактор й редактор коду, який може використовуватися як самодостатнє рішення, так й у ролі технологічного стека для побудови різних спеціалізованих рішень. Зокрема, на платформі Atom побудовані середовище розробки «Visual Studio Code» від компанії «Microsoft» та «Nuclide» від «Facebook». [41]

Проаналізувавши усі переваги та недоліки праналізованих середовищ розробки, було обрано середовище розробки IntelliJ IDE, так як данне середовище ідеально розширяєме під усіпотреби розробника, має встроєний функціона для роботи з Git, велику кількість вбудованих сніпетів для аналізу коду.

3.4 Процес розробки інформаційної технології інтернет-маркетингу за допомогою React Native

Для створення додатку, було обрано React Native, оскільки даний фреймворк, на даний момент є багатообіцяючим за рахунок того, що бере найкраще з веб-розробки та React.js.

Завдяки JS-рушію забезпечується висока продуктивність, порівнянна з нативною. Додаток будується з компонент платформи – це нативні модулі, загорнуті в React-компоненти.

```
return (  
  <Footer>  
    <div className="timeline-header">  
      <div className="timeline-header-left">  
        <div className="item">  
          <div className="name">  
            <Translate value="audio-waveform" />  
          </div>  
          <div className="value">  
            <input  
              defaultChecked={true}  
              type="checkbox"  
              onChange={event => {  
                if (!wf) return;  
                wf.setOptions({  
                  wave: event.target.checked,  
                });  
              }}  
            />  
          </div>  
        </div>  
      </div>  
    </div>  
  </div>  
)
```

Рисунок 3.1 – Приклад React-компонента

Так, програмісти пишуть код на JavaScript, який працює з нативними компонентами платформи, тим самим передаючи переваги й зручність використання React.js з браузера в мобільні додатки.

Уся навігація в додатку реалізується через Navigation, але через те, що NavigationBar, повністю відв'язаний від компонента [42].

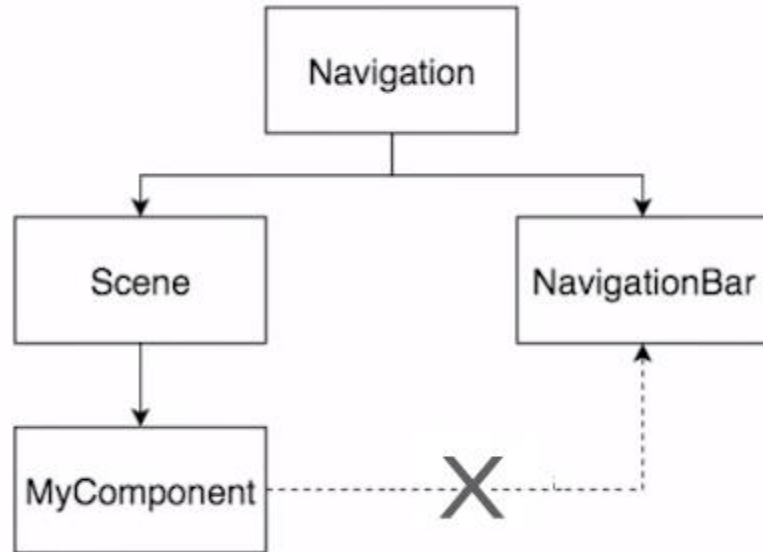


Рисунок 3.2 Схема роботи Navigation

Тому для вирішення цієї проблеми можна використовувати `redux`, тоді схема роботи навігації буде виглядати так:

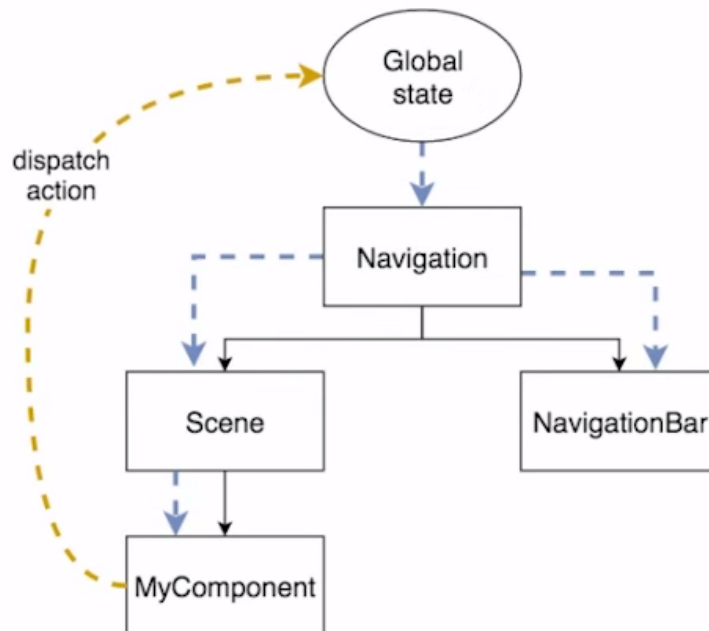


Рисунок 3.3 Схема роботи Navigation з Redux

Від програмістів робота з фреймворком вимагає знання JavaScript, Objective-C API та Android Java API.

Програма складається із компонентів, навігатора, бази даних та Redux. База даних відіграє роль моделі – структури додатка яка змінюються за допомогою контролера, Redux – контролер, модуль який змінює структуру в залежності від дій користувача, компоненти відповідають за відображення, що залежить від даних які поступають з структури. Також з компонентів відправляються сповіщення контролеру, коли користувач виконує певні дії.

Під час розробки додатка, було створено наступні компоненти.

CatalogueScreen – даний компонент відповідає за відображення екрану каталогу, з даного компонента відсилаються команди для завантаження вибраної категорії товарів. Відображається таблиця товарів яка складається з комірок товарів які містять: код товару, вартість, назву, зображення та зображення кошика, для того щоб купити даний товар.

HomeScreen – компонент, що відповідає за відображення домашнього екрану, даний екран відображає найновіші товари та являється стартовим під час авторизації у додаток. Даний компонент містить слайдер зображень в якому містяться зображення акцій та пропозицій, таблицю з новими товарами, та кнопку переходу до каталога.

Слайдер реалізований за допомогою бібліотеки «react-native-image-slider».

OrderScreen – компонент який відповідає за відображення придбаних товарів, сюди потрапляють товари після того як користувач натисну на кнопку придбати, він відображає таблицю товарів у якій міститься: назва товару, зображення, вартість, код товару та дата купівлі.

PromotionsScreen – компонент, що відповідає за відображення екрану з акціями, під час переходу в дане вікно, відсилається про початок завантаження даних про акційні товари.

`SettingsScreen` – компонент, що відповідає за вікно з налаштуваннями. Тут користувач матиме змогу скорегувати свої дані, такі як: прізвище, ім'я, пошту, стать, номер телефону. Усі дані користувача зберігаються в його записі в базі даних.

`LogInScreen` – компонент, що відповідає за авторизацію, з нього відсилаються нотифікації про авторизацію, якщо авторизація завершується успішно, «навігатор» направляє користувача на головний екран. Авторизація створена за допомогою `Firebase Auth API`.

`SignUpScreen` – компонент, що відповідає за відображення вікна реєстрації з нього відсилаються нотифікації про створення нового користувача. Під час реєстрації створюється запис в `Firebase Users`, кожен користувач має свій хеш. На основі хеша створюється запис в базі з стартовими даними.

`SideDrawerNavigator` – компонент, що відповідає за відображення виїжджаючого бокового меню, та навігацію між екранами. Навігація створена за допомогою бібліотеки «`react-navigation`». У виїжджаючому меню знаходяться зображення та вкладки на кожен з екранів додатка.

`Header` – компонент який, що відповідає за відображення «голови» додатка. В ньому міститься стрічка пошуку, назва сторінки, гамбургер-меню яке запускає відображення вікна з навігацією, та іконку кошика, що перенаправляє користувача на каталог-вікно.

`PredictionList` – компонент , що відповідає за створення списку прог

Компоненти `Header`, `sideDrawerNavigator`, слайдер – `dumb,stateless` або презентаційні компоненти. Вони не містять стану (`state`) або посилання на `DOM` елемент (`ref`). В них передаються параметри (`props`) та контекст. Вони можуть визначити `defaultProps`, `propTypes` й `contextTypes`. Основна їх задача – відображення контенту на основі переданих параметрів.

Усі інші – компоненти контейнери. Компоненти контейнери – компоненти, що слугують основними «будівельними» блоками додатку, вони

представляються як класи які наслідуються від React.Component, дані компоненти мають стан, в них входять презентаційні компоненти в які передаються параметри.

3.5 Побудова UML-діаграми класів розробленого програмного засобу

Під час проектування крос-платформного мобільного додатку було розроблено діаграму класів (рисунк. 3.1).

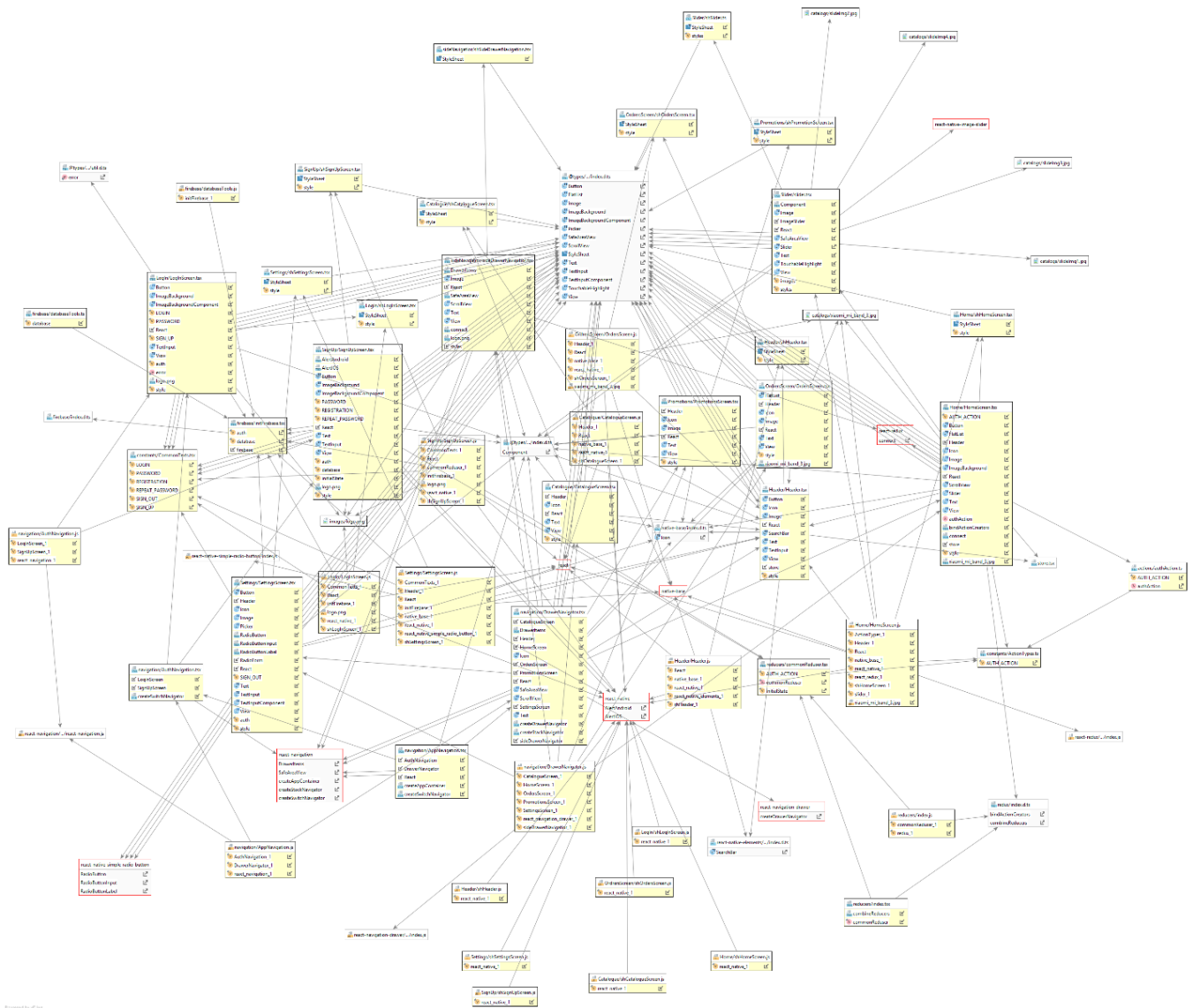


Рисунок 3.1 – UML-діаграма класів додатку інтернет-маркетингу

З діаграми класів видно, що весь додаток має модульну структуру, кожен елемент включає в себе презентаційний компонент `Header` в який передаються певні параметри.

Модульна структура передбачає в собі те, що кожен модуль програми є ізольованим від іншого, це допомагає покращити захищеність та якість коду.

Компонент-контейнер `HomeScreen` містить у собі презентаційний компонент `Slider`, який реалізує слайдер зображень, за допомогою пакета “`react-native-image-slider`”.

Компонент-контейнер `PredictionList` містить у собі компонент який містить реалізацію нейронної мережі, яка була реалізована за допомогою `TensorFlow`.

Кожен компонент-контейнер наслідується від базового `React.Component`, та з'єднується зі `store` за допомогою методу `connect`, який імпортовано з бібліотеки “`react-redux`”.

Компоненти через які здійснюється навігація по додатку, мають в собі імпорт `react-navigator`, який включає в себе набір інструментів для роботи з навігацією.

Усі компоненти мають зв'язок із `@types/react`, цей пакет необхідний для того, щоб `TypeScript` компілятор міг справно компілювати `.tsx` файли у `.js` файли.

Деякі компоненти мають векторні зображення, які можна використовувати імпортувавши “`Icon`” з бібліотеки `native-base`, яка розширює стандартний функціонал `react-native`.

В кожен компонент імпортується файл з префіксом “`sh`” в якому знаходиться об'єкт зі стилями, також кожен компонент містить у собі імпорт “`react`”, для того щоб мати доступ до всіх інструментів даного фреймворка.

3.6 Тестування розробленого програмного засобу інтернет-магазину з використанням нейронної мережі бізнес-прогнозування та аналіз результатів його роботи

Розроблена система інтернет-маркетингу була протестована, що підтвердило коректність її роботи.

Було проведено близько 1000 запусків додатку, протестовано можливості його роботи, що дало можливість адекватно оцінити його роботу.

Після запуску програми відкривається вікно авторизації, в якому відбувається процес створення нового користувача, або авторизація вже існуючого.

Коли ми створюємо новий акаунт, створюється відповідний запис у базі даних (рисунок 3.2)

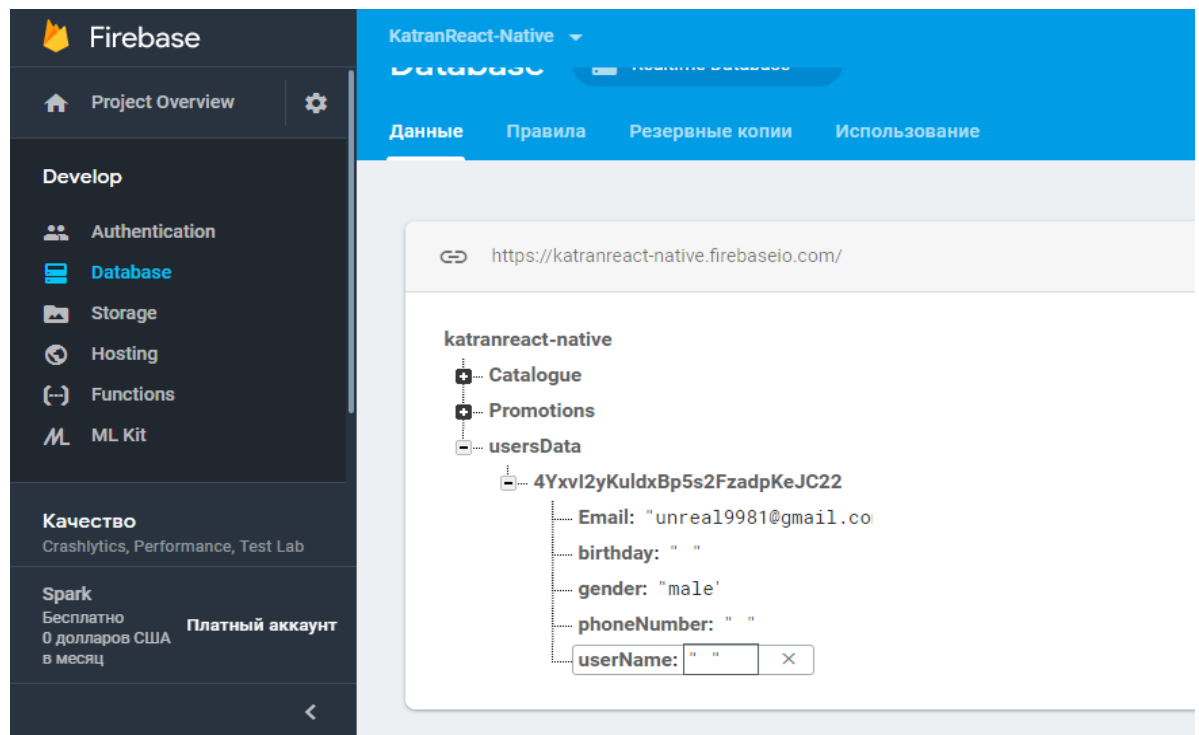


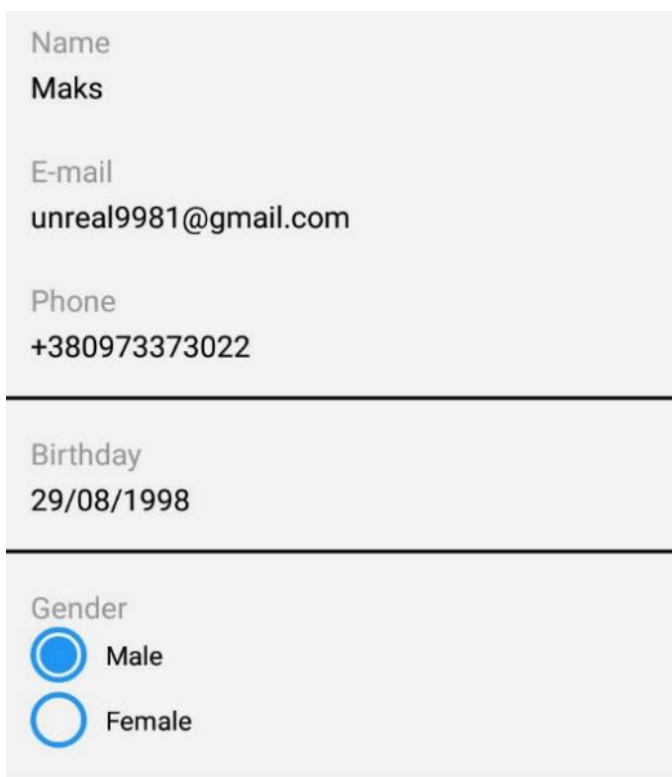
Рисунок 3.2 – Запис нового користувач в базі даних

Якщо ми вже маємо існуючий аккаунт, ми маємо нагоду авторизації.

Для перевірки авторизації, необхідно ввести e-mail та пароль які співпадають з даними зареєстрованого користувача.

Якщо дані співпали з існуючими, додаток одразу покаже головний екран, та в меню висвітиться ім'я користувача, а панель з налаштуваннями відобразить основну інформацію про користувача.

На рисунку 3.3 наведена інформація про раніше створеного користувача.



The image shows a user profile card with the following information:

- Name:** Maks
- E-mail:** unreal9981@gmail.com
- Phone:** +380973373022
- Birthday:** 29/08/1998
- Gender:** Male (selected), Female

Рисунок 3.3 – Інформація про існуючого користувача

Головне вікно розробленої системи інтернет-маркетингу містить такі елементи взаємодії з користувачем:

- Кнопка-меню– для відкриття глобальної навігації по додатку;
- Кнопка «Кошик» де ви можете переглянути відібрані товари;
- Кнопка «Каталог» – для переходу в Каталог додатку;

- Панель «Запропоновані товари» – товари які були запропоновані нейронною мережею на основі переглянутих раніше товарів користувачем;
- Панель-слайдер «Акції» – на ній відображаються діючі акції даного додатка;
- Кнопка «Налаштування» – для переходу в меню налаштувань. Де користувач має змогу скорегувати свої персональні дані;
- Кнопка «Переглянуті товари» містить редагуємий список переглянутих товарів;
- Кнопка «Вийти» – для закриття програми.

На рисунку 3.4 наведено порівняння результатів роботи розробленої програми з аналогом.

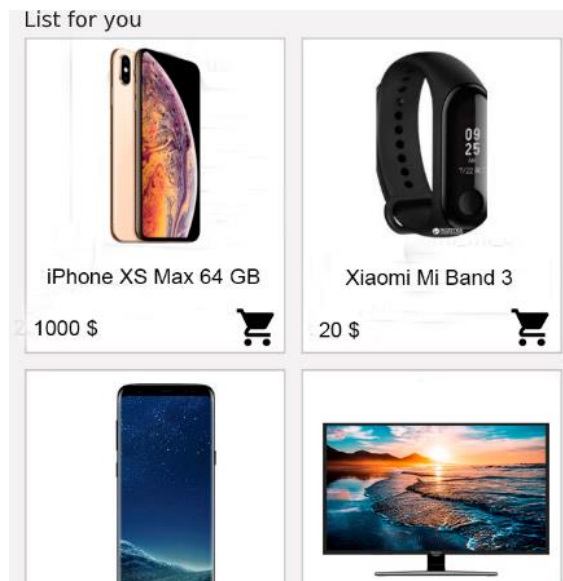


Рисунок 3.4 – Порівняння результатів роботи розробленої програми з аналогом.

Отже, після тестових запусків програми можна зробити такі висновки: програма створює достатньо точний список з прогнозованих товарів. Результати,

отримані під час тестування програми, відповідають очікуваним результатам. Розроблений програмний додаток працює відповідно до завдань проектування. Розроблений додаток має кращу точність прогнозування порівняно з аналоговими програмами. Точність зросла майже на 2,1% порівняно з аналогами, що є дуже хорошим результатом.

Оцінювання точності відбувалось в такий спосіб, ми взяли одну категорію товарів на всіх аналогах, та переглянули 30 товарів з неї. Потім перейшли в кожен із додатків. І порахували середню абсолютну помилку в процентах MAPE, яка застосовується для вимірювання точності короткочасного прогнозу [42].

$$MAPE = \frac{1}{N} \sum_{t=1}^N \frac{|Z(t) - \hat{Z}(t)|}{Z(t)} * 100\%, \quad (3.1)$$

де $Z(t)$ – фактичне значення тимчасового прогнозу, а $\hat{Z}(t)$ – прогнозоване.

Ми порахували цю оцінку для таких пар зображень: 100 переглянутих товарів категорії смартфони в додатку аналогу – Rozetka, 100 переглянутих товарів категорії смартфони в розробленому додатку, 100 переглянутих товарів категорії смартфони в додатку аналогу – UA-TAO, 100 переглянутих товарів категорії смартфони в додатку аналогу – Joom .

Таблиця 3.2 – Порівняльний аналіз достовірності роботи різних алгоритмів вирішення задачі

Програма	MAPE
Rozetka	80%
Розроблений додаток	82%
Joom	75%
UA-TAO	78%

3.6 Обґрунтування вибору архітектурної організації обчислень на GPU

У наш час швидкість навчання нейронної мережі можна збільшити в декілька разів при використанні відео процесора (GPU). В сучасних GPU міститься сотні (а в деяких моделях тисячі) обчислювальних ядер, тому розрахунки на них виконуються набагато швидше, ніж за допомогою CPU [42].

GPU (графічний процесор) – це процесор відеокарти, який виконує обробку 2D або 3D графіки. Наявність якого звільняє CPU від зайвої навантаженості завдяки чому комп'ютерний процесор може виконувати всі інші важливі завдання швидше.

У 2003 році GPU вперше отримала підтримку 32-бітних точних обчисленнях. Direct3D виділився основним інтерфейсом програмування, першим, хто надав підтримку шейдерів. З'явилися перші програми, які використовують GPU для високоефективних обчислень, напрямок GPGPU почав розвиватись. GPGPU (General-Purpose computing on Graphic Processing Units) – використання графічних процесорів для вирішення довільних обчислювальних задач. Для програмування GPU пропонується потоковий підхід до програмування. Цей підхід передбачає розбиття програми на відносно невеликі етапи (ядра), які обробляють елементи потоків даних. Ядра відображаються як шейдери, а потоки даних – як текстури в GPU [42].

Порівняння швидкості навчання нейронної мережі на CPU та GPU наведено у таблиці 3.3.

Таблиця 3.3 – Порівняння швидкості навчання нейронної мережі на CPU та GPU

	GPU (NVIDIA GeForce 920M)	CPU (Intel Core i3-5005u)
Швидкість навчання однієї епохи, ~хв.	8	31
Швидкість навчання п'яти епох з використанням fine tuning, ~хв.	34	169

Багато сучасних засобів навчання нейронних мереж підтримують GPU. Серед таких бібліотеки Brain.js, Tensorflow, deeplearn.js, propel.

Бібліотека – збірка підпрограм або об'єктів, що використовуються для розробки програмного забезпечення. У деяких мовах програмування (наприклад, в JavaScript) те ж, що модуль, в деяких – кілька модулів [42].

У наш час розробити великий проект без використання бібліотек майже неможливо. Бібліотеки дозволяють програмісту відразу зосередитися на поставлених завданнях, а не починати будувати структуру низького рівня для вирішення базових задач. Такий підхід значно скорочує час на розробку та тестування програмного продукту, спрощує його підтримку, а отже, зменшує його вартість.

Tensorflow використовує технологію CUDA, яка працює тільки на GPU виробництва компанії NVIDIA.

Compute Unified Device Architecture (CUDA) – це програмна модель, яка описує обчислювальну паралельність та структуру ієрархічної пам'яті безпосередньо для мови програмування. З точки зору програмного забезпечення, реалізація CUDA – це кросплатформна система для збирання та виконання програм, частини яких працюють на процесорі та графічному процесорі. CUDA

використовується для розробки GPGPU-додатків без прив'язки до графічних API та підтримується всіма GPU NVIDIA, починаючи з серії GeForce 8 [42].

Ця паралельна архітектура обчислень може значно підвищити продуктивність обчислень за рахунок використання графічних процесорів, які можуть дуже ефективно керувати великими блоками даних. Платформа паралельних обчислень CUDA використовується на сьогоднішній день в тисячах GPU-прискорених додатків і тисячах опублікованих наукових статтях [42].

CUDA використовується в багатьох галузях для дослідницьких та наукових цілей а саме, фінансове моделювання, медична візуалізація, енергетичні дослідження, тощо. Це також допомагає розробити нове покоління програм у багатьох сферах які пов'язані з роботою з графікою, наприклад: перетворення мобільного відео та поліпшення якості відео на комп'ютерах.

CUDA має ряд переваг перед традиційним обчисленням загального призначення на GPU (GPGPU), використовуючи графічні API:

- Розсіяне читання – код можна читати з довільних адрес у пам'яті;
- Єдина віртуальна пам'ять (CUDA 4.0 та вище);
- Єдина пам'ять (CUDA 6.0 і вище);
- Спільна пам'ять – CUDA відкриває зону швидкої спільної пам'яті, яку можна розділити між потоками. Це призводить до більш високої пропускної здатності, ніж це можливо за допомогою текстурних пошуків.
- Швидше завантаження та відтворення на GPU
- Повна підтримка цілих і побітових операцій, включаючи цілі текстурні переходи.

3.8 Висновок

У даному розділі здійснено аналіз існуючих мов програмування для створення мобільних додатків (Java, JavaScript, Swift/Objective-C). Обґрунтовано

доцільність використання мови програмування JavaScript, оскільки дана мова має багату кількість фреймворків та бібліотек для створення кросплатформних додатків, а також нейронних мереж прогнозування.

Розглянуто та проаналізовано існуючі бібліотеки для створення нейронних мереж на мові програмування JavaScript. Обрано бібліотеку Tensorflow, оскільки вона є найпотужнішим інструментом для роботи з нейронними мережами у даній мові програмування.

Розглянуто найсучасніші IDE та текстові редактори для мови програмування JavaScript, такі як Visual Studio Code, WebStorm, IntelliJ IDE. Було обрано IntelliJ IDE, оскільки воно безкоштовне та найзручніше середовище розробки для даної мови програмування.

Наведено процес розробки додатку за допомогою бібліотеки React Native в якому продемонстровано вирішення проблеми навігації за допомогою Redux.

Розроблено UML-діаграму класів додатку інтернет-маркетингу, яка відображає усі зв'язки між функціональними компонентами програмного забезпечення.

Розглянуто особливість архітектурної організації обчислень на GPU та обґрунтовано доцільність її використання для прискорення процесу прогнозування набору товарів.

Виконано тестування розробленого програмного додатку, яке підтвердило коректність його роботи. Також було здійснено порівняльний аналіз результатів прогнозування розробленого застосунку за критерієм точності. У підсумку, порівняно з аналогом точність прогнозування зросла приблизно на 1.5% – 3%.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено двох незалежних експертів. Такими експертами будуть Арсенюк І.Р. та Сілагін О.В.

Оцінювання комерційного потенціалу розробки буде здійснено за 12-ма критеріями за 5-ти бальною шкалою. Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.1.

Таблиця 4.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	Арсенюк І.Р.	Сілагін О.В.
	Бали, виставлені експертами:	
1	4	4
2	3	3
3	3	4
4	4	3
5	3	4
6	4	4
7	3	3
8	4	4
9	3	3
10	4	3
11	3	4
12	4	4
Сума балів	СБ ₁ = 43	СБ ₂ = 43
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{2} = 43$	

Отже, з отриманих даних таблиці 4.1 видно, що нова розробка має високий рівень комерційного потенціалу.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи.

Для розробки нового програмного продукту необхідні такі витрати.

Основна заробітна плата для розробників визначається за формулою (4.1):

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника;

T_p – кількість робочих днів у місяці, $T_p = 22$ дні;

t – число днів роботи розробника, $t = 50$ днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 4.2.

Таблиця 4.2 – Розрахунки основної заробітної плати

Працівник	Оклад M , грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	10000	454,55	5	2272,75
Інженер-програміст	8000	380,95	50	19047,61
Всього:				21320,36

Розрахуємо додаткову заробітну плату:

$$Z_{\text{дод}} = 0,1 \cdot 21320,36 = 2132,03 \text{ (грн.)}$$

Нарахування на заробітну плату операторів НЗП розраховується як 37,5...40% від суми їхньої основної та додаткової заробітної плати:

$$H_{зп} = (З_о + З_р) \cdot \frac{\beta}{100}, \quad (4.2)$$

$$H_{зп} = (21320,36 + 2132,03) \cdot \frac{36,3}{100} = 8513,21 \text{ (грн.)}$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12}, \quad (4.3)$$

де Ц – балансова вартість обладнання, грн;

H_a – річна норма амортизаційних відрахувань % (для програмного забезпечення 22%);

T – Термін використання (T=3 міс.).

Таблиця 4.3 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
Персональний комп'ютер	10000	22	3	550
Всього:				550

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_1^n H_i \cdot Ц_i \cdot K_i, \quad (4.4)$$

де n – кількість комплектуючих;

H_i – кількість комплектуючих i -го виду;

$Ц_i$ – покупна ціна комплектуючих i -го виду, грн;

K_i – коефіцієнт транспортних витрат (прийmemo $K_i = 1,1$).

Таблиця 4.4 – Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
1. Флешка	шт.	180	1	180
2. Пачка паперу	уп.	130	1	130
3. Ручка	шт.	10	1	10
Всього з урахуванням транспортних витрат				352

Витрати на силову електроенергію розраховуються за формулою:

$$B_e = B \cdot П \cdot \Phi \cdot K_n, \quad (4.5)$$

де B – вартість 1кВт-години електроенергії ($B=2$ грн/кВт);

$П$ – установлена потужність комп'ютера ($П=0,6$ кВт);

Φ – фактична кількість годин роботи комп'ютера ($\Phi=200$ год.);

K_n – коефіцієнт використання потужності ($K_n < 1$, $K_n = 0,7$).

$$B_e = 2 \cdot 0,6 \cdot 200 \cdot 0,7 = 168 \text{ (грн.)}$$

Розрахуємо інші витрати $V_{ін}$.

Інші витрати I_B можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$V_{ін} = (1..3) \cdot (Z_o + Z_p). \quad (4.6)$$

Отже, розрахуємо інші витрати:

$$V_{ін} = 1 * (21320,36 + 2132,03) = 23452,39 \text{ (грн.)}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$V = Z_o + Z_d + H_{зп} + A + K + V_e + I_B$$

$$V = 21320,36 + 2132,03 + 8513,21 + 625 + 352 + 168 + 23452,39 = 56563 \text{ (грн.)}$$

Розрахуємо загальну вартість наукової роботи $V_{заг}$ за формулою:

$$V_{заг} = \frac{V_{ін}}{\alpha}, \quad (4.7)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$V_{заг} = \frac{56563}{1} = 56563.$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{В_{заг}}{\beta}, \quad (4.8)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{56563}{0,9} = 62847,77 \text{ (грн.)}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо отримання прибутку від реалізації результатів нашої розробки. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Це забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності .

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{я} \cdot N + \Pi_{я}\Delta N)_i \quad (4.9)$$

де $\Delta\Pi_{я}$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки витрати на виготовлення інформаційної технології зменшаться на 40 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 40 грн), а кількість користувачів, які будуть користуватись збільшиться: протягом першого року – на 200 користувачів, протягом другого року – на 130 користувачів, протягом третього року – 90 користувачів. Реалізація інформаційної технології до впровадження результатів наукової розробки складала 500 користувачів, а прибуток, що отримував розробник до впровадження результатів наукової розробки – 400 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту $\Delta\Pi_1$ протягом першого року складатиме:

$$\Delta\Pi_1 = 40 \cdot 500 + (400 + 40) \cdot 200 = 90000 \text{ грн.}$$

Протягом другого року:

$$\Delta\Pi_2 = 40 \cdot 500 + (400 + 40) \cdot (200 + 130) = 147200 \text{ грн.}$$

Протягом третього року:

$$\Delta\Pi_3 = 40 \cdot 500 + (400 + 40) \cdot (200 + 130 + 90) = 186800 \text{ грн.}$$

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $E_{\text{абс}}$ вкладених інвестицій розраховується за формулою:

$$E_{\text{абс}} = (\text{ПП} - PV), \quad (4.10)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки 2, 3, 4.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 4.1.

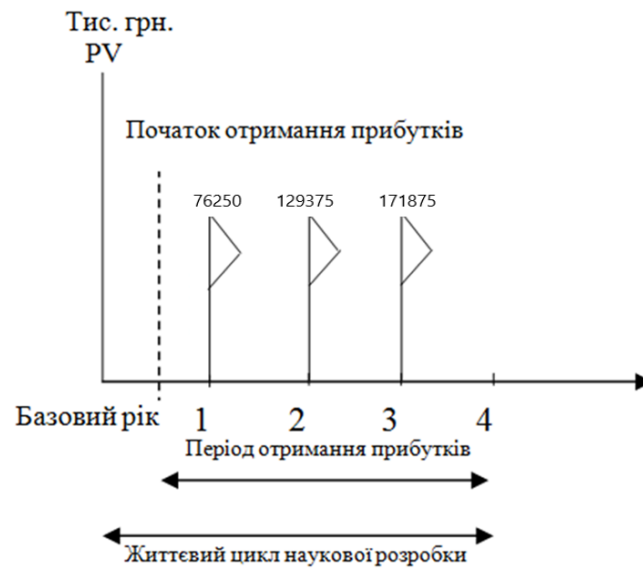


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо вартість чистих прибутків за формулою:

$$\text{ПП} = \sum_1^m \frac{\Delta\Pi_i}{(1+\tau)^t} \quad (4.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

τ – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки.

Отже, розрахуємо вартість чистого прибутку:

$$\text{ПП} = \frac{62847,77}{(1+0,1)^0} + \frac{90000}{(1+0,1)^2} + \frac{147200}{(1+0,1)^3} + \frac{186800}{(1+0,1)^4} = 375408.38 \text{ (грн.)}$$

Тоді розрахуємо E_{abc} :

$$E_{abc} = 375408,38 - 62847,77 = 312560,61 \text{ грн.}$$

Оскільки $E_{abc} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_B за формулою:

$$E_B = \sqrt[T]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.12)$$

де E_{abc} – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій $PV = 3B$, грн;

$T_{ж}$ – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_B = \sqrt[3]{1 + \frac{312560,6}{62847,77}} - 1 = 0,81 \text{ або } 81\%.$$

Далі, розраховану величина E_B порівнюємо з мінімальною (бар'єрною) ставкою дисконтування τ_{\min} , яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування τ_{\min} визначається за формулою:

$$\tau = d + f,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках;
в 2020 році в Україні $d = 0,2$;

f – показник, що характеризує ризикованість вкладень, величина $f = 0,1$.

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки $E_B = 81\% > \tau_{\min} = 0,3 = 30\%$, то у інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_B},$$

$$T_{ок} = \frac{1}{0,81} = 1,2 \text{ рік.}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи розроблено інформаційну технологію інтернет-маркетингу. Під час аналізу предметної області було відзначено, що нейромережеве прогнозування в сфері інтернет-маркетингу може знайти своє застосування в багатьох процесах розробки додатків та значно полегшує роботу людям. Проте сьогодні не існує досконалого засобу, який є зручним в використанні, та дозволяє з достатньою точністю прогнозувати набір товарів для покупок. Визначено що однією з проблем точності прогнозування є нехтування рядом аналітичних даних таких як: час переглядання товарів (з врахуванням захисту від неактивності користувача), кількість разів переглядання конкретного товару, фільтри сортування товарів.

Аналіз предметної області інформаційної технології інтернет-маркетингу показав, що виконання цього завдання з використанням штучних нейронних мереж та машинного навчання є найбільш якісним методом.

Розглянуто та проаналізовано існуючі нейронні мережі для розв'язання задачі прогнозування, такі як перцептрон та мережа радіально-баисних функцій. Було обрано модель перцептрона із поєднанням вихідних нейронів, оскільки дана модель володіє кращими властивостями екстраполявання та виходять не такими громіздкими при великій розмірності вектора входів.

У ході проектування було запропоновано математичну модель прогнозування очікуваних покупок для інформаційних технологій інтернет-маркетингу, що відрізняється врахуванням додаткових аналітичних даних, що характеризують як товари так і дії користувачів під час використання додатка.

Розроблено алгоритм функціонування інформаційної технології інтернет-маркетингу, що передбачає використання перцептрона із поєднанням вихідних нейронів. Також було розроблено загальну структурну схему інформаційної

технології інтернет-маркетингу, що містить 7 етапів та визначає послідовність процесів, що відбувається під час функціонування додатку.

Після проектування було здійснено аналіз існуючих мов програмування для створення мобільних додатків (Java, JavaScript, Swift/Objective-C). В даній роботі використано мову програмування JavaScript так як дана мова має багату кількість фреймворків та бібліотек для створення кросплатформних додатків, та нейронних мереж прогнозування.

Розглянуто та проаналізовано існуючі бібліотеки для створення нейронних мереж на мові програмування JavaScript. Після детального аналізу було обрано бібліотеку Tensorflow, так як вона є найпотужнішим інструментом для роботи з нейронними мережами у даній мові програмування.

Наведено процес розробки додатка за допомогою бібліотеки React Native в яких продемонстровано вирішення проблеми навігації за допомогою Redux. На основі створеного додатку була розроблена розроблена UML-діаграма класів. Яка відображає усі зв'язки між компонентами.

Здійснено тестування розробленого програмного додатку, яке підтвердило коректність його роботи. Також було здійснено порівняльний аналіз результатів прогнозування розробленого застосунку за критерієм точності. У підсумку, порівняно з аналогом точність прогнозування зросла приблизно на 1.5% – 3%.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Солоний М. А., Арсенюк І. Р. Порівняльний аналіз підходів розробки модулю прогнозування для мобільного додатку інтернет-магазину. // Тези доповідей XLIX науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії. – Вінниця: ВНТУ, 2020. Електронний ресурс (режим доступу <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2020/paper/view/8989/7752>)
2. Свідectво про реєстрацію авторського права на твір «Інформаційна технологія інтерне-маркетингу» / Арсенюк І. Р., Солоний М. А.
3. Нативный vs. Кроссплатформенный: различия подходов в разработке мобильных приложений: [Електронний ресурс]. – Режим доступу: <http://www.ironwaterstudio.com/ru/blog/nativnyj-vs-krossplatformennyj-razlichiya-podhodov-v-razrabotke-mobil-nyh-prilozhenij> (дата звертання 05.05.2020).
4. Основы Serverless приложений в среде Amazon Web Services: [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/309370/> (дата звертання 05.05.2020)
5. Rozetka.ua: [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Rozetka.ua> (дата звертання 05.05.2020)
6. Taobao: [Електронний ресурс]. – Режим доступу: <https://world.taobao.com/> (дата звертання 05.05. 2020).
7. Amazon.com: [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Amazon.com> (дата звертання 05.05. 2020).
8. Progressive Web Apps: [Електронний ресурс]. – Режим доступу: <https://developers.google.com/web/progressive-web-apps/> (дата звертання 07.05.2020).

9. Apache Cordova: [Электронный ресурс]. – Режим доступа: <https://cordova.apache.org/docs/en/latest/guide/overview/index.html> (дата звертання 08.05. 2020).
10. React Native 0.58: [Электронный ресурс]. – Режим доступа: <https://facebook.github.io/react-native/> (дата звертання 9.05. 2020).
11. Visual Studio Tools for Xamarin: [Электронный ресурс]. – Режим доступа: <https://visualstudio.microsoft.com/ru/xamarin/?rr=https%3A%2F%2Fru.wikipedia.org%2F> (дата звертання 9.05. 2020).
12. Flutter: [Электронный ресурс]. – Режим доступа: <https://flutter.dev/> (дата звертання 10.05. 2020).
13. Сравнение решений для кроссплатформенной разработки [Электронный ресурс]: [Веб-сайт]. – Режим доступа: <https://smartum.pro/ru/blog-ru/sravneniye-resheniy-krossplatformennoy-razrabotki-phonegap-xamarin-flutter-react-native/> (дата звернення 10.05. 2020).
14. React Native – одного JS мало [Электронный ресурс] : [Веб-сайт]. – Режим доступа: <https://habr.com/ru/post/323214/> (дата звернення 11.05.2020).
15. Powering UberEATS with React Native and Uber Engineering [Электронный ресурс]: [Веб-сайт]. – Режим доступа: <https://eng.uber.com/ubereats-react-native/> (дата звернення 13.05.2020).
16. React Native at Instagram [Электронный ресурс]: [Веб-сайт]. – Режим доступа: <https://instagram-engineering.com/react-native-at-instagram-dd828a9a90c7> (дата звернення 14.05.2020).
17. Visual Studio Code [Электронный ресурс]: [Веб-сайт]. – режим доступа : https://ru.wikipedia.org/wiki/Visual_Studio_Code. (дата звернення 17.05.2020) – Назва з екрана.
18. Dart vs JavaScript [Электронный ресурс]: [Веб-сайт]. – режим доступа: <https://blog.codemagic.io/dart-vs-javascript/>. (дата звертання 18.05.2020) – Назва з екрана.

19. Середовище розробки «IntelliJ IDEA» [Електронний ресурс]: [Веб-сайт]. – режим доступу : <https://www.jetbrains.com/idea/>. (дата звернення 17.05.2020) – Назва з екрана.
20. TypeScript – Wikipedia [Електронний ресурс]: [Веб-сайт]. – Режим доступу: <https://uk.wikipedia.org/wiki/TypeScript> (дата звернення 16.05.2020) – Назва з екрана.
21. TypeScript [Електронний ресурс]: [Веб-сайт]. – Режим доступу: <https://www.typescriptlang.org/> (дата звернення 17.05.2020) – Назва з екрана
22. Dumb Components and Smart Components [Електронний ресурс]: [Веб-сайт]. – режим доступу : <https://medium.com/@thejasonfile/dumb-components-and-smart-components-e7b33a698d43>. (дата звернення 18.05. 2020) – Назва з екрана.
23. Swift (язык программирования). Електронний ресурс (режим доступу <https://ru.wikipedia.org/wiki/Swift> (дата звертання 05.10.2020).
24. Dropout — метод решения проблемы переобучения в нейронных сетях: [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/company/wunderfund/blog/330814/> (дата звертання 05.05.2020).
25. 11 JavaScript-библиотек для машинного обучения: [Електронний ресурс]. – Режим доступу: <https://neurohive.io/ru/osnovy-data-science/js-libraries-machine-learning/> (дата звертання 16.09.2020)
26. Machine Learning in JavaScript with TensorFlow.js: [Електронний ресурс]. – Режим доступу: <https://www.udemy.com/course/machine-learning-in-javascript-with-tensorflow-js> (дата звертання 16.11.2020)
27. Ошибка прогнозирования: как рассчитать и применять: [Електронний ресурс]. – Режим доступу: <http://logist.ru/articles/oshibka-prognozirovaniya-kak-rasschitat-i-primenyat> (дата звертання 26.11. 2020).
28. What is internet marketing: [Електронний ресурс]. – Режим доступу: <https://glem.com.ua/> (дата звертання 12.09. 2020).

29. Dropout: A Simple Way to Prevent Neural Networks from Overfitting: [Электронный ресурс]. – Режим доступа: <https://jmlr.org/papers/v15/srivastava14a.html> / (дата звертання 09.09.2020).
30. Overfitting: [Электронный ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/Overfitting> (дата звертання 31.09. 2020).
31. Java: [Электронный ресурс]. – Режим доступа: [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)) (дата звертання 09.07. 2020).
32. Objective C: [Электронный ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/Objective-C> (дата звертання 23.08. 2020).
33. Переобучение (Overtraining):[Электронный ресурс]. – Режим доступа: <https://wiki.loginom.ru/articles/overtraining.html> (дата звертання 10.05. 2020).
34. Из чего состоит интернет-маркетинг [Электронный ресурс]:[Веб-сайт]. – Режим доступа: <https://tilda.education/courses/marketing/internet-marketing-beginning/>(дата звернення 10.05. 2020).
35. Переобучение [Электронный ресурс] : [Веб-сайт]. – Режим доступа: <https://www.datanomics.ru/news/pereobuchenie/> (дата звернення 30.11.2020).
36. Интернет-маркетинг: основы [Электронный ресурс]: [Веб-сайт]. – Режим доступа: <https://elama.ru/blog/internet-marketing-osnovy/> (дата звернення 13.05.2020).
37. Brain.js: GPU accelerated Neural networks in JavaScript for Browsers and Node [Электронный ресурс]: [Веб-сайт]. – Режим доступа: <https://brain.js.org/#/> (дата звернення 14.10.2020).
38. brain.js [Электронный ресурс]: [Веб-сайт]. – режим доступа : <https://github.com/BrainJS>. (дата звернення 17.09.2020) – Назва з екрана.

39. TensorFlow [Электронный ресурс]: [Веб-сайт]. – режим доступа: <https://en.wikipedia.org/wiki/TensorFlow/>. (дата звертання 18.10.2020) – Назва з екрана.
40. Tensor Flow [Электронный ресурс]: [Веб-сайт]. – режим доступа : <https://www.tensorflow.org/>. (дата звернення 17. 10.2020) – Назва з екрана.
41. TensorFlow is an end-to-end open source platform for machine learning [Электронный ресурс]: [Веб-сайт]. – Режим доступа: <https://github.com/tensorflow/tensorflow> (дата звернення 16.11.2020) – Назва з екрана.
42. Hello, TensorFlow. Библиотека машинного обучения от Google [Электронный ресурс]: [Веб-сайт]. – Режим доступа: <https://habr.com/ru/post/305578/> (дата звернення 17.10.2020) – Назва з екрана