МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки
Факультет Комп'ютерних наук
Кафедра Програмної Інженерії

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Дослідження методів прогнозування прибутковості фільмів та серіалі

Виконав:
студент 2 курсу, групи ІПЗм-21-2
Волоховський В. Є.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного Забезпечення
Тип програми освітньо-наукова
Керівник доц. Назаров О.С.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ Дудар З.В.
(підпис) (прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної Інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Волоховському Віталію Євгеновичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження методів прогнозування прибутковості фільмів та серіалів

затверджена наказом університету від «29» березня 2023 р. № 302 Ст

2. Термін подання студентом до екзаменаційної комісії «15» травня 2023 р.

3. Вихідні дані до роботи календарний план роботи, методичні вказівки до оформлення пояснювальної записки, методи прогнозування з використанням нейронних мереж.

4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної галузі, виявлення існуючих систем, аналіз існуючих досліджень, огляд та визначення математичних моделей, створення плану проведення експерименту, реалізація алгоритмів, проведення експерименту, аналіз отриманих результатів.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів	Примітка
1	Аналіз предметної галузі	28.01.2023	виконано
2	Аналіз існуючих досліджень	30.01.2023	виконано
3	Здійснення огляду математичних моделей	10.02.2023	виконано
4	Теоретична валідація обраних моделей	10.02.2023	виконано
5	Розробка плану наукового дослідження	01.03.2023	виконано
6	Формування набору даних для аналізу	15.03.2023	виконано
7	Аналіз методів попередньої обробки даних	31.03.2023	виконано
8	Аналіз програмних засобів реалізації моделей	31.03.2023	виконано
9	Побудова плану експерименту	31.03.2023	виконано
10	Імплементація обраних моделей	05.04.2023	виконано
11	Проведення експерименту	30.04.2023	виконано
12	Аналіз результатів експерименту	01.05.2023	виконано
13	Написання пояснювальної записки	05.05.2023	виконано
14	Підготовка презентації	12.05.2023	виконано
15	Попередній захист роботи	12.05.2023	виконано
16	Захист кваліфікаційної роботи	15.05.2023	виконано

Дата видачі завдання 29 березня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Назаров О.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до кваліфікаційної роботи, 68 сторінок, 18 рисунків, 4 таблиці, 30 джерел, 4 додатки.

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, КІНОІНДУСТРІЯ, НЕЙРОННА МЕРЕЖА, ПРИБУТОК, ПРОГНОЗУВАННЯ, РЕКУРЕНТНА НЕЙРОННА МЕРЕЖА, ФІЛЬМ.

Об’єктом дослідження є методи прогнозування прибутковості фільмів у сучасних ринкових умовах.

Метою кваліфікаційної роботи є визначення ефективності використання нейронних мереж, поліноміальної та сегментованої регресійних моделей для прогнозування прибутковості фільмів.

У результаті кваліфікаційної роботи було проведено аналіз предметної галузі, визначено моделі для подальшого аналізу, виконано математичний опис вказаних моделей, виконано програмну реалізацію обраних моделей та алгоритмів попередньої обробки даних, проведено експеримент для дослідження ефективності прогнозування прибутку фільмів обраними моделями.

INTELLIGENT DATA ANALYSIS, FILM INDUSTRY, NEURAL NETWORK, REVENUE, FORECASTING, RECURRENT NEURAL NETWORK, MOVIE.

The object of the study is the methods of forecasting the profitability of films and in modern movie market conditions.

The purpose of the qualification work is to determine the effectiveness of using neural networks, polynomial and segmented regression models for forecasting the profitability of films.

As a result of the qualification work, an analysis of the subject area was performed, models for further analysis were determined, a mathematical representation of the specified models was provided, a software implementation of the selected models and data pre-processing algorithms was performed, an experiment was conducted to define the effectiveness of predicting the profit of films by the selected models.

Умови публікації пояснювальної записки

Я, Волоховський Віталій Євгенович
(прізвище, ім’я, по батькові)
студент групи ІПЗм-21-2 здобувач вищої освіти на другому (магістерському) рівні
кафедра програмної інженерії,
(повна назва кафедри)
заявляю: моя кваліфікаційна робота на тему
Дослідження методів прогнозування прибутковості фільмів та серіалів
(назва роботи)
що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений(а) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

Історія кінематографа налічує понад 100 років, а фільми відіграють важливу роль у житті мільйонів людей у всьому світі [1]. Під час пандемії темпи росту кіноіндустрії значно знизилися, що вплинуло на доходи кінокомпаній, спонукаючи їх шукати нові підходи для залучення глядачів [2].

Після закінчення пандемії галузь почала стрімко відновлюватися. У 2022 році кількість проданих квитків у кіно зросла більш ніж на 60 відсотків порівняно з попереднім роком, але показники все ще на 40 відсотків нижчі, ніж у 2019 році [3, 4].

Змінилося і співвідношення вікових груп, які відвідують кінотеатри. Кількість глядачів віком від 12 до 17 років порівняно з 2020 роком зросла на 127%, а на вікову групу 18-34 припадає половина відвідувань кінотеатру [5]. Серед основних факторів походу до кінотеатру є якість зображення у фільмі, вартість квитків, а також доступність фільму на стрімінгових платформах. Втім 55% дорослого населення у США все ж таки віддають перевагу перегляду фільмів удома, аргументуючи це тим, що вони не зацікавлені у фільмах, які показують у кінотеатрах [6]. Найбільш популярними стрімінговими сервісами є Netflix, Amazon Prime Video, HBO Max, Apple TV+.

Також відбулись зміни у жанрі фільмів, якому віддають перевагу глядачі. У 2021 році найбільшу частку займали фільми у жанрі бойовик, змістивши фільми у жанрі пригоди на друге місце. Також фільми у жанрі жахів посіли третє місце за прибутковістю [7].

Зміна трендів та підходів до ведення бізнесу є значним викликом навіть для великих кінокомпаній, оскільки їм необхідно швидко підлаштуватися до нових умов ринку. Деякі компанії вже використовують сторонні системи для прогнозування успіху фільмів, наприклад, Warner Bros. використовує платформу Cinelytic, 20th Century Fox інтегрована з системою Merlin, а Netflix використовує власну систему [8-10].

Щоб бути конкурентоспроможним та мати високий рівень прибутку, кінокомпаніям необхідно розуміти своїх глядачів, аналізувати їхні вподобання та інтереси. Базуючись на цих даних, компанії можуть розробити стратегії розвитку та визначити, які саме фільми треба створювати, щоб утримувати існуючих клієнтів та залучати нову аудиторію.

Для вирішення цієї проблеми треба розуміння того, які проекти мали успіх, які сподобались глядачам більше, а які не мали зовсім успіху, та що на це вплинуло. Оскільки кількість знятих фільмів велика, а на успіх може впливати багато різних факторів, проведення аналізу є складною задачею. Тому для вирішення цієї проблеми можна використовувати сучасні методи машинного навчання.

Для проведення дослідження було обрано поліноміальну регресію, сегментована регресію та багатошарову нейронну мережу прямого зв’язку. Поліноміальну регресію та сегментовану регресію було обрано, тому ще ці моделі працюють швидко та дозволяють моделювати нелінійні зв’язки у даних. З іншого боку, нейрона мережа прямого зв’язку є більш складною системою, проте дозволяє працювати з більшими наборами даних та значною кількістю вхідних характеристик.

Метою роботи є визначення методів прогнозування прибутковості фільмів, порівняння їхньої ефективності та розробка програмної системи, яка дозволить прогнозувати прибутковість фільмів за заданим набором характеристик.

Об’єктом дослідження є розмір прибутку фільму, який кінокомпанія отримує в результаті продажів квитків, авторських прав.

Предметом дослідження є методи прогнозування прибутку фільмів у сучасних умовах ринку кіноіндустрії.

1 ОПИС ПРОБЛЕМНОЇ ГАЛУЗІ

1.1 Аналіз предметної області

Згідно з даними сайту The Numbers загальні збори кіноіндустрії у США та Канаді у 2022 році склали більше, ніж 7,5 млрд. дол. Це майже на 3 млрд. дол. більше ніж у попередньому році (див. рис. 1.1) [11].

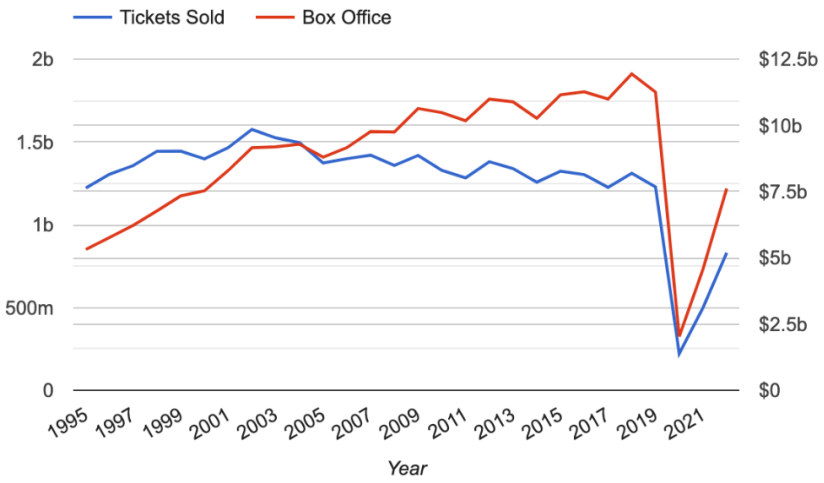

Рисунок 1.1 – Загальний розмір зборів кіноіндустрії

Також ми можемо побачити, скільки заробила кіноіндустрія у різних країнах світу у 2021 році, дані наведені у млрд. дол. США (див. рис. 2) [12].

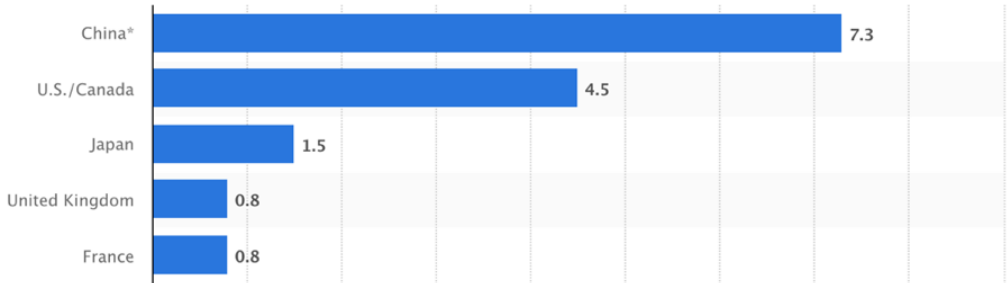

Рисунок 1.2 – Прибуток кіноіндустрії по країнах у 2021 році

У 2020 році кіноіндустрія Китаю, США та Японії отримала 21 млрд, 7 млрд та 3,4 млрд доларів відповідно, що робить цю сферу досить цікавою для інвесторів.

Великобританія та Франція мають нижчий загальний прибуток у цій сфері на рівні 0,8 млрд. доларів.

Розглянемо, яким чином ринок кіно розподілений між компаніями та як він змінювався з часом (див. рис. 1.3) [13].

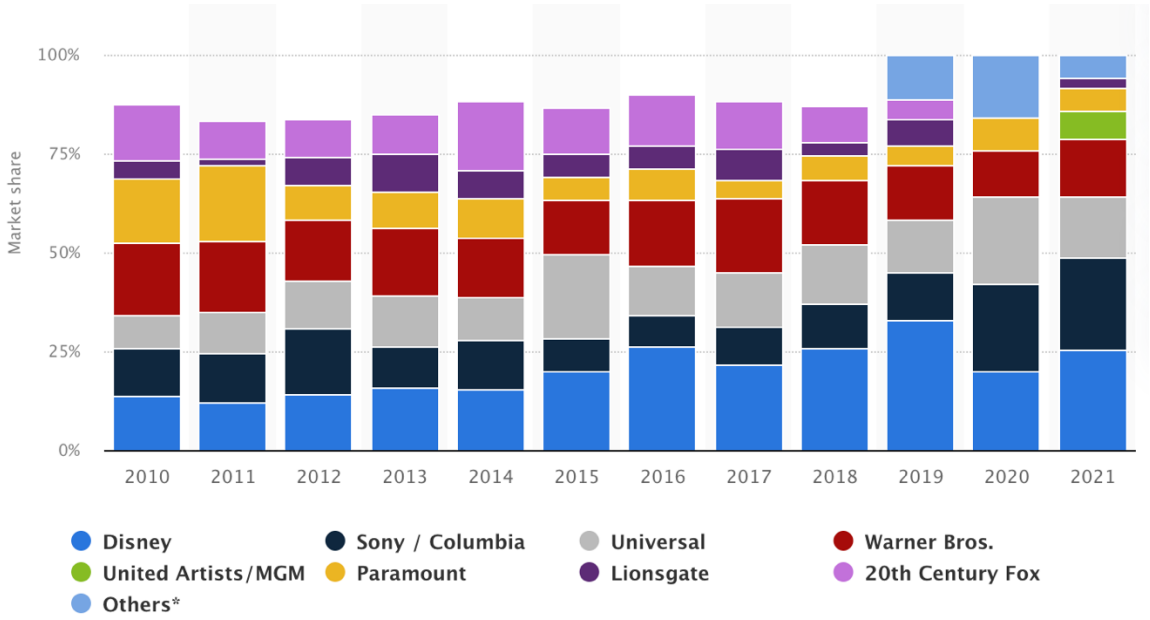

Рисунок 1.3 – Розподіл ринку між компаніями

Серед найбільших компаній, представленими на ринку США, можна виділити наступні:

– Disney;

– Sony Pictures;

– Universal Pictures;

– Warner Bros.

При цьому на графіку зовсім не представлені молоді кінокомпанії, які також є стрімінговими сервісами, такі як Netflix, Amazon Prime Video, HBO Max, Apple TV+. Треба зауважити, що Disney має власний стрімінговий сервіс Disney+, на якому відбуваються прем'єри фільмів та серіалів. Але в більшості випадків кінокомпанії орієнтуються на покази у кінотеатрах, або ж використовують стрімінгові сервіси інших компаній для залучення аудиторії.

За рахунок того, що стрімінгові сервіси співпрацюють з багатьма кінокомпаніями та мають доступ до значно більшої аудиторії, ніж будь-яка окрема кінокомпанія, вони можуть краще адаптуватися до змін трендів кіноіндустрії, мають більше даних щодо вподобань користувачів та менш схильні до ризиків за рахунок великої різноманітності картин, які вони пропонують користувачам. Саме це обумовило стрімкий зріст компанії Netflix під час пандемії з середини 2019 до початку 2022 років (див. рис. 1.4) [14].

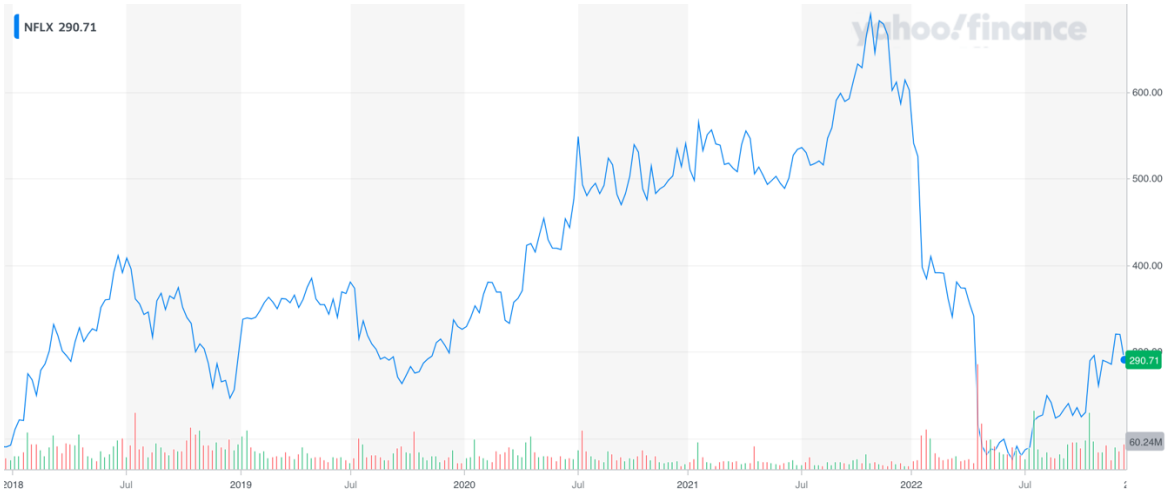

Рисунок 1.4 – Вартість акцій компанії Netflix

Але навіть для таких компаній, як Netflix, закінчення пандемії спричинило стрімкий спад в прибутках. Люди все більше проводять часу поза межами дому, віддаючи перевагу живому спілкуванню та активному відпочинку. Тому залучити глядачів до перегляду фільмів та серіалів стає більш складним завданням, а конкуренція між кінокомпаніями за час і увагу глядачів тільки збільшується.

В таких умовах виникає проблема того, як швидко і точно визначити нові тренди, нові вподобання та запити глядачів і спрогнозувати потенційні прибутки від нових фільмів, щоб не витрачати ресурси на зйомки фільму, який не буде мати успіху.

Розглянемо, які інструментами та методи існують на ринку та чи користуються компанії з кіноіндустрії ними для прийняття таких рішень.

З відкритих джерел було отримано інформацію про подібні системи тільки для кількох компаній.

Netflix, як сучасна компанія орієнтована на онлайн-покази, має велику кількість даних як про фільми та серіали, так і про користувачів, що дозволяє їм аналізувати тренди та відповідати новим умовам. Згідно з блогом розробників Netflix, компанія розробила власну систему, яка використовує методи машинного навчання для аналізу історичних даних та виконання прогнозів щодо нових фільмів.

Для прогнозування розміру аудиторії компанія використовує нейрону мережу, яка на основі опису нового проекту та історичних даних робить припущення щодо того, яка кількість глядачів у різних країнах буде зацікавлена у цьому фільмі [10]. Для визначення того, які існуючі фільми схожі на новий проект, та відношення його до певної категорії використовується softmax функція.

Компанія 20th Century Fox використовує більш складний підхід для прогнозування розміру аудиторії. Вони розробили власний інструмент комп'ютерного зору Merlin Video, який вивчає візуальне представлення трейлерів до фільмів, визначаючи різні характеристики відео, такі як кольори, освітлення, типи обличчя, різні об'єкти та ландшафти [9]. А потім, використовуючи історичні дані та визначені характеристики, виконує передбачення майбутньої аудиторію певного трейлера.

З іншої сторони, Warner Bros. та Sony Pictures використовують сторонню систему аналізу Cinelytic – продукт, який дозволяє проаналізувати фільм, зробити прогнози щодо різних метрик, включаючи доходи, витрати на зйомки та розмір аудиторії, і запропонувати покращення, такі як набір акторів, час зйомок і прем'єри (див. рис. 1.5) [8].

Як і Netflix, Cinelytic використовує історичні дані для прогнозування майбутніх метрик. Вони взаємодіють з іншими компаніями, які мають доступ до великих об'ємів даних, що дозволяє їм використовувати методи машинного навчання для аналізу. Саме через доступ до великих об'ємів даних такі великі

компанії як Warner Bros. та Sony Pictures зацікавлені в Cinelytic, оскільки самі вони не мають можливості виконувати такий точний аналіз.

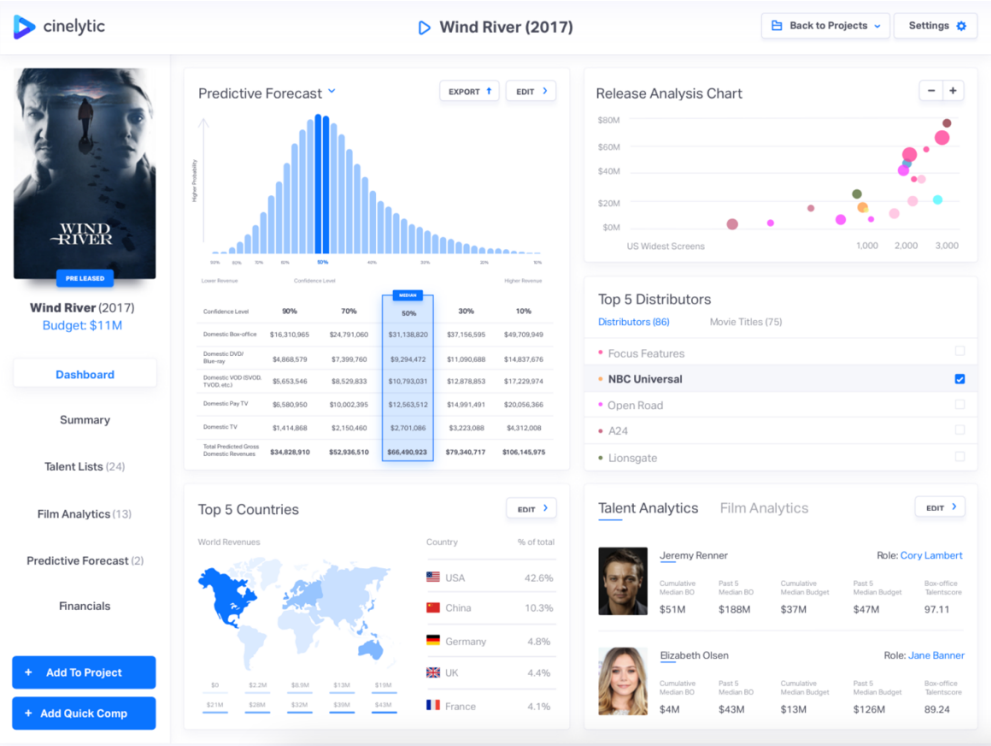

Рисунок 1.5 – Продукт Cinelytic

Нажаль, Cinelytic не має у відкритому доступі інформації про те, які саме методи та підходи вони використовують для створення прогнозів, але можна припустити, що вони використовують власні нейроні мережі, які адаптовані саме під ті дані та задачі, з якими компанія працює найбільше.

1.2 Аналіз попередніх досліджень

Вирішенням проблеми прогнозування прибутку фільмів займались і інші науковці. Розглянемо декілька з попередніх досліджень.

У 2018 було опубліковано статтю «Forecasting Movie Box Office Profitability», яка розглядає методи прогнозування прибутковості фільмів [15].

У цьому досліджені було розглянуто три методи прогнозування: багатошарові нейронні мережі, дерева рішень та множинна лінійна регресія. Дані для навчання було взято з сервісу IMDB, а набір даних складався з 295 фільмів.

Нейронна мережа показала найкращі результати передбачення порівняно з іншими методами. Також було порівняно методи прогнозування цільового значення: відношення цільового значення до одного з багатьох визначений класів (multi-class prediction); відношення цільового значення до одного з двох класів: прибуток дорівнює або більше бюджету та прибуток менше бюджету; передбачення дискретного цільового значення.

За результатами дослідження було визначено, що передбачення дискретної цільової змінної може бути застосовано на практиці та краще відображає прогнозоване значення. З іншої сторони відношення результату до визначених категорій показало високий рівень помилки, що робить цей підхід непрактичним у застосуванні для вирішення проблеми. Також було визначено, що характеристики бюджету, режисеру та наявність продовження найбільше впливають на прибутковість фільму.

Інше дослідження з назвою «Comparing performance of ensemble methods in predicting movie box office revenue» було проведено у 2020 [16]. У ньому автори сфокусувалися на досліджені ефективності використання моделей дерев рішень, лінійної регресії та К-найближчих сусідів з використанням ансамблевих методів. Для дослідження було використано вибірку, яка налічувала 1439 фільмів.

В результаті дослідження було визначено, що дерева рішень з використанням ансамблевих методів показали кращий результат для передбачень прибутку фільмів, порівняно з іншими моделями та з класичною моделлю дерева рішень. Варто зазначити, що для дослідження використовувалась вибірка виключно корейських фільмів. Ці дані є дуже специфічними, та не дозволяють перенести ці результати на більш широкий спектр фільмів з інших країн світу, тому автори, як зазначено у досліджені, планують провести подальші дослідження для перевірки своїх результатів на більш різноманітній вибірці.

1.3 Аналіз методів прогнозування

Загалом, для проблем передбачення дискретних числових значень використовують регресійні моделі та нейроні мережі.

Серед регресійних моделей розглянемо ті, які можуть моделювати нелінійні зв’язки для даних, що мають декілька вхідних характеристик, оскільки характер залежності прибутку фільму від інших характеристик не є лінійним. Проаналізувавши підходи до вирішення схожих проблеми, було виділено наступні моделі:

– поліноміальна регресія (polynomial regression);

– сегментована регресія (segmented regression);

– гребнева регресія (ridge regression);

– регресійне дерево рішень (decision tree regression);

– опорно-векторна регресія (support vector regression).

Оскільки у цьому дослідженні є необхідність працювати зі змішаними типами даних (числові, текстові, категоріальні), а точності прогнозування є важливішою за час навчання моделі, було вирішено розглянути методи поліноміальної регресії та сегментованої регресії. Поліноміальна регресія є найпростішою моделлю, яка дозволяє виявляти нелінійні залежності в даних, а також показує гарну продуктивність у навчанні за рахунок використання поліномів вищих порядків [17].

Вибір сегментованої регресії обумовлено тим, що фільми можна розділити на декілька категорій, де дані у середині групи будуть мати більш схожі характеристики між собою, ніж з елементами з інших груп. Тому використання сегментованої функції може дати більш точний прогноз, оскільки різні її частини будуть краще відповідати певній групі даних, ніж одна неперервна функція.

Серед нейронних мереж можна навести наступні:

– одношарова нейронна мережа (one-layer neural network);

– багатошарова нейронна мережа прямого зв’язку (multi-layer feed-forward neural network);

– рекурентна нейронна мережі (recurrent neural network);

– LSTM-мережа.

Оскільки більшість характеристик представлені числовими або категоріальними типами даних, а окремі приклади не залежать один від одного, було обрано багатошарову нейронну мережу прямого зв’язку, оскільки вони працюють досить швидко та підтримують обробку даних з великою кількістю параметрів.

1.4 Аналіз джерел даних

Було виконано аналіз наступних відкритих баз даних та інших доступних ресурсів, де зберігаються дані про фільми та серіали:

– IMDb datasets;

– Movielens datasets;

– TMDb API;

– OMDb API.

Було вирішено обрати сервіс TMDb API. Цей сервіс надає зручний та гнучкий API, який дозволяє отримати детальну інформацію про фільм або серіал, включаючи наступні характеристики: назва, бюджет, прибуток, список жанрів, опис, кількість голосів, середню оцінку, рейтинг, мова оригіналу та інші [18]. За допомогою API було отримано інформацію про найбільш популярні та прибуткові фільми. та для проведення дослідження було сформовано набір з 5000 фільмів. Сервіс надає дані у форматі JSON, які попередньо необхідно конвертувати у формат CSV.

1.5 Постановка задачі

Сформуємо задачу даної наукової роботи: провести дослідження методів прогнозування прибутковості фільмів та серіалів, порівняти їхню ефективність за

допомогою обраних критеріїв порівняння та обрати ефективніший метод для вирішення поставленої задачі.

Для порівняння ефективності підходів будемо використовувати наступні критерії:

– точність прогнозування;

– час прогнозування;

– можливість узагальнення моделі у разі недостатньої кількості даних.

Вирішення поставленої задачі розіб’ємо на декілька менших кроків:

– визначення та ознайомлення з математичними методами та моделями для прогнозування;

– визначення методів обробки даних;

– реалізація алгоритму прогнозування прибутковості фільмів;

– розробка плану проведення дослідження;

– проведення дослідження;

– аналіз отриманих результатів.

2 МАТЕМАТИЧНЕ ПРЕДСТАВЛЕННЯ

2.1 Багатошарова нейронна мережа прямого зв’язку

В загальному вигляді багатошарову мережу можна представити наступним чином (див. рис. 2.1) [19].

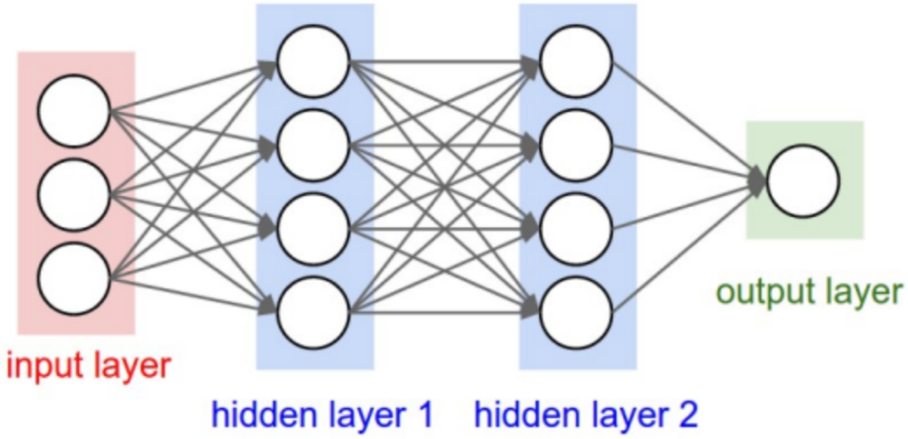

Рисунок 2.1 – Графічне представлення багатошарової мережі

На рисунку 2.1 можна побачити, що червоним кольором виділено вхідний шар (input layer), який утворений набором характеристик об’єкту, і який зазвичай не рахується як шар нейронної мережі. Синім кольором позначені приховані (hidden) шари моделі, кожен з яких виконує певні перетворення даних, використовуючи функцію активації, та передає результат наступному шару. Зеленим кольором позначено вихідний (output) шар, який формує кінцевий результат роботи моделі.

Ця мережа є прикладом простої нейронної мережі прямого зв’язку. Така архітектура передбачає, що інформація передається лише вперед у мережі від вхідних вузлів, через приховані вузли і до вихідних вузлів, не утворюючи циклів.

Розглянемо математичне представлення цих моделей [20].

Нехай задано набір вхідних даних у вигляді пар вхідного вектору-харатеристик $\overrightarrow{x_i}$ та вихідного значення y_i:

$$X = \{(\overrightarrow{x_1}, y_1), \dots, (\overrightarrow{x_i}, y_i)\}, i = \overline{1, m}, \quad (1)$$

де $\vec{x} = (x_1, \dots, x_j), j = \overline{1, n}$ – вектор характеристик об'єкту дослідження,

n – кількість характеристик,

m – кількість елементів тренувальної вибірки.

Задачею навчання є визначення такої функції h, щоб $h(x)$ давала значення максимально наближене до вихідного значення y.

Функція $h(x)$ задається наступним чином:

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x, \quad (2)$$

де x_i – вектор характеристик i-го елементу тренувального набору,

θ_i – вектор параметрів або вагів (weights), які налаштовуються в процесі навчання алгоритму.

Для визначення наскільки значення функції $h(x)$ відрізняється від очікуваного значення y, використовують функцію втрат [21]:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h(x_i) - y_i)^2. \quad (3)$$

Для мінімізації функції втрат використовується метод градієнтного спуску, який можна записати наступним чином:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} J(f(x_i, \theta), y_i), \quad (4)$$

де $J(f(x_i, \theta), y_i)$ – функції втрат представлена для наочності як функція, яка залежить від вектора характеристик, вектору параметрів та вихідного значення.

Метод градієнтного спуску знаходить мінімум функції, з'ясовуючи, у якому напрямку у просторі параметрів θ, нахил функції зростає найкрутіше та рухається у протилежному напрямку.

Навчання моделі, тобто адаптація та зміна параметрів θ, відбувається послідовно для всіх елементів тренувального набору наступним чином:

$$\theta_{j+1} = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta), \tag{5}$$

де α – параметр швидкості навчання (learning rate), який впливає на швидкість зміни параметрів θ.

Кожен нейрон у прихованому шарі перетворює значення з попереднього шару за допомогою зваженого лінійного підсумовування, за яким слідує нелінійна функція активації. Функція активації необхідна для введення нелінійної залежності у моделі та використовується на виході прихованих шарів. Позначається функція активації наступним чином $g(z)$.

Розглянемо функції активації.

Найпоширенішою функцією активації є функція ReLU (rectified linear unit) – випрямлена лінійна одинична функція (див. рис. 2.2).

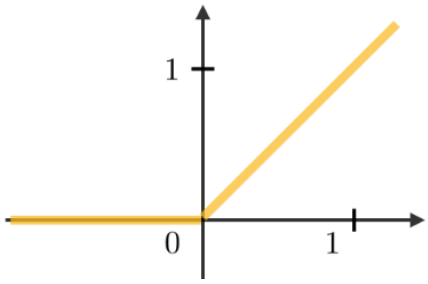

Рисунок 2.2 – Графік ReLU функції

Ця функція дозволяє зберігати багато властивостей даних, які дозволяють легко оптимізувати лінійні моделі за допомогою методів на основі градієнта

Вона задається наступним чином:

$$g(z) = \begin{cases} x, & \text{якщо } x \geq 0 \\ 0, & \text{якщо } x < 0 \end{cases}. \quad (6)$$

Іншою функцією активації є логістична функція, яка задається наступним чином:

$$g(z) = \frac{1}{1+e^{-z}}. \quad (7)$$

Функція логістичної регресії повертає значення у проміжку (0, 1) та широко застосовується для класифікації. Вона дозволяє зробити висновок щодо належності об'єкта до певного класу. Якщо значення функції $g(z) \geq 0.5$, то можна вважати, що об'єкт належить до цього класу, інакше – ні (див. рис. 2.3).

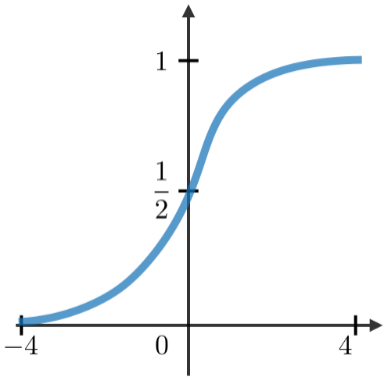

Рисунок 2.3 – Графік логістичної функції

У нейронній мережі функції активації можуть комбінуватися, тобто шари можуть мати різні функції активації, для отримання різних вихідних значення, наприклад приховані шари можуть мати функцію ReLU, а вихідний шар використовуватиме логістичну функцію для відношення результату прогнозування до одного з визначений класів.

Для вирішення поставленої задачі буде використано ReLU функція як для прихованих шарів, так і для вихідного шару, оскільки необхідно прогнозувати дискретне значення прибутку.

2.2 Поліноміальна регресія

Розглянемо математичне представлення поліноміальної регресійної моделі. Використаємо початкові умови з виразу (1).

Тоді для кожної пари вектору характеристик $\overrightarrow{x_i}$ та вихідного значення y_i [22]:

$$\begin{aligned} h(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_1^p + \\ + \beta_{p+1} x_2 + \cdots + \beta_{2p} x_2^p + \\ \cdots \\ + \beta_{p(n-1)+1} x_n + \cdots + \beta_{pn} x_n^p, \end{aligned} \quad (7)$$

де p – порядок поліному;

β – вектор коефіцієнтів поліному.

Для спрощення запису формул представмо значення незалежних змінних для вибірки розміром m за допомогою символу X – матриця розміром $m \times pn$, а вектор розрахованих значень $h(x)$ за допомогою символу Y – вектор довжиною m. Для визначення різниці між розрахованим значенням функції та очікуваним, використовуються функції помилок. У даному випадку наведено функцію MSE – середня квадратична помилка:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \operatorname{MSE}(\beta) = (X^T X)^{-1} X^T Y. \quad (8)$$

Задачею начання моделі є мінімізація значення помилки за рахунок зміни вектору параметрів β. Аналогічним чином, за допомогою методу градієнтного спуску, відбувається налаштування коефіцієнтів моделі.

2.3 Сегментована регресія

Розглянемо математичне представлення регресійної моделі, яка використовує B-spline функцію для опису залежності між змінними. Spline-функція порядку q – це сегментованою поліноміальною функцією ступеня $q-1$ для вектора незалежних змінних x. B-spline порядку q є базисною функцією для spline-функції того ж порядку, визначених для тих самих вузлів. Spline-функція порядку q для визначеного набору вузлів може бути представлена як лінійна комбінація декількох B-spline [23]:

$$h(x) = \sum_{i=1}^{t} \alpha_i B_{i,q}(x), \tag{9}$$

де q – порядок функції;

t – кількість вузлів;

α_i – spline-коефіцієнт;

$B_{i,q}(x)$ – значення i-го B-spline порядку q в точці x.

Сегменти поліноміальної функції можуть бути представлені за допомогою наступних рекурсивних виразів:

$$B_{i,0}(x) = \begin{cases} 1 & \text{if } t_i \le x < t_{i+1} \\ 0 & \text{otherwise} \end{cases},$$

$$B_{i,k}(x) = \frac{x - t_i}{t_{i+k} - t_i} B_{i,k-1}(x) + \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x). \tag{10}$$

Для визначення різниці між розрахованим значенням функції та очікуваним, використовується метод найменших квадратів (8).

Порівняно з поліноміальною регресією, B-spline функції мають декілька переваг:

– вони більш гнучкі та дозволяють налаштовувати не тільки ступень полінома, але й кількість вузлів;

– вони краще оброблюють значення, які лежать за межами відомих значень моделі.

Розглянемо графік, на якому представлені графіки поліномів наступних порядків (3, 4, 5) та функція B-spline (див. рис. 2.4) [24]. На рисунку можна побачити, що вихідні дані (позначені чорним кольором) мають складний характер залежності та не можуть бути описані лінійною функцією.

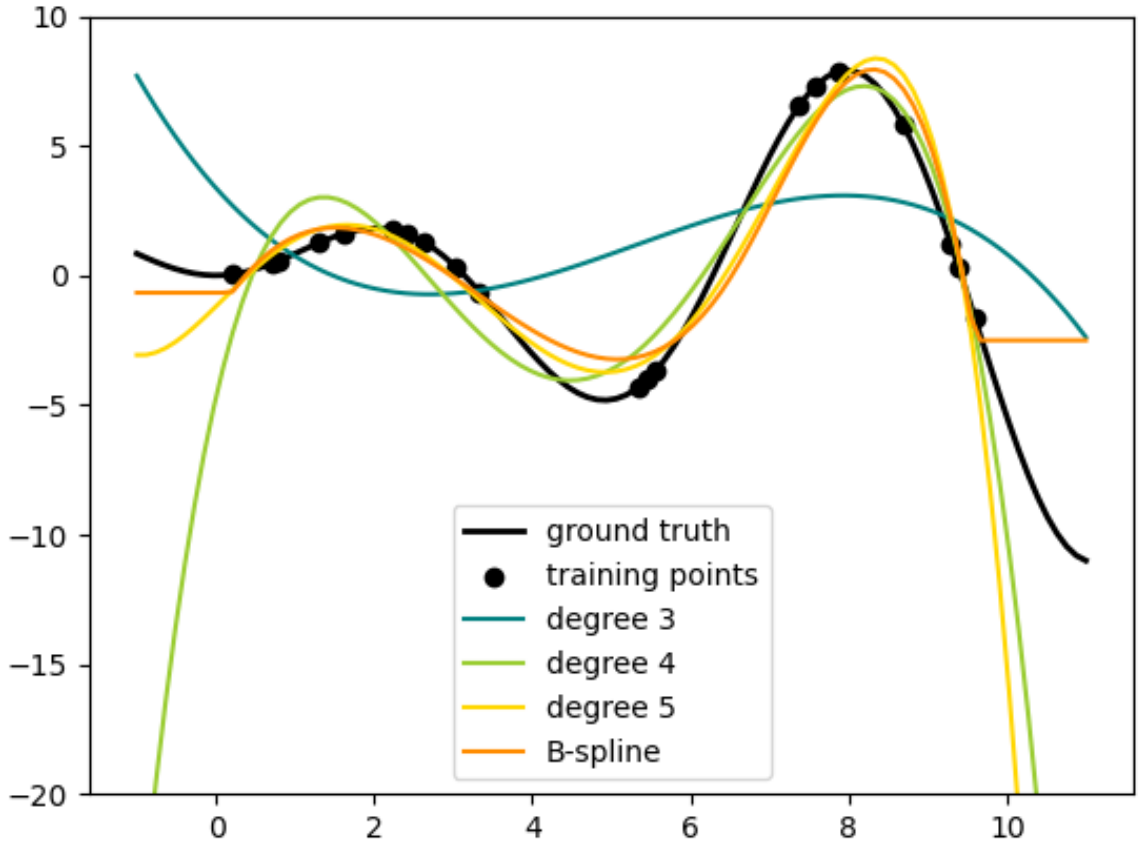

Рисунок 2.4 – Порівняння поліноміальної та B-spline функцій

Далі до графіку послідовно було додано графіки поліноміальних функцій різних порядків. Поліном третього порядку (позначено синім кольором) має велику помилку у передбачені вихідних значень та некоректно описує залежність у них. Поліном четвертого порядку (позначено зеленим кольором) краще описує дані зі значно меншою помилкою. Проте для значень, які виходять за межі навчальних

даних, поліном показує погані здібності до екстраполяції, тому ми бачимо, що функція у лівій частині графіку швидко зменшується до від’ємних значень, хоча вихідні дані мають значення більше нуля. Поліном п’ятого порядку ще точніше описує залежність у даних, а також краще екстраполює дані у лівій частині графіку. Функція B-spline має ступінь 3, при цьому описує залежність так само точно, як і поліноміальна функція 5-го ступеня. Це досягається за рахунок адаптації кількості вузлів функції. Також, вона краще екстарполює дані за межами навчального набору, використовуючи, наприклад, константне значення або лінійну функцію.

3 ІМПЛЕМЕНТАЦІЯ МОДЕЛЕЙ

3.1 Огляд даних

Розробка програми для вирішення поставленої задачі починається з попередньої обробки даних та представлення їх у формі, зрозумілій методам машинного навчання.

Наведемо графік розподілення прибутку у наборі даних (див. рис. 4.1).

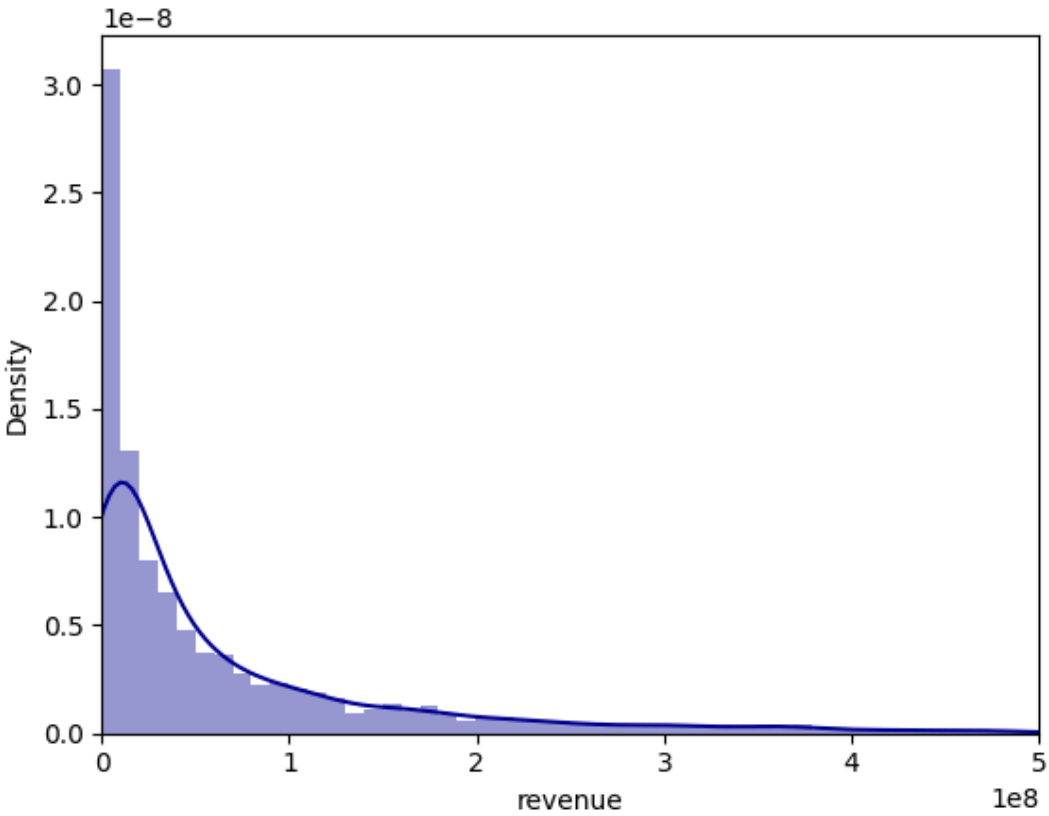

Рисунок 4.1 – Розподіл прибутку

Можна побачити, що більше ніж 75% фільмів у наборі даних має прибуток менше 100 мільйонів доларів, в той час як інші 25% розподілені у проміжку від 100 мільйонів до 2,7 мільярдів доларів. Такий розподіл формує довгий «хвіст» даних, який може вплинути на результати прогнозування моделі, оскільки їй буде складно виявити залежності у невеликій кількості даних та узагальнити їх.

Розглянемо розподіл характеристики бюджету (див. рис. 4.2).

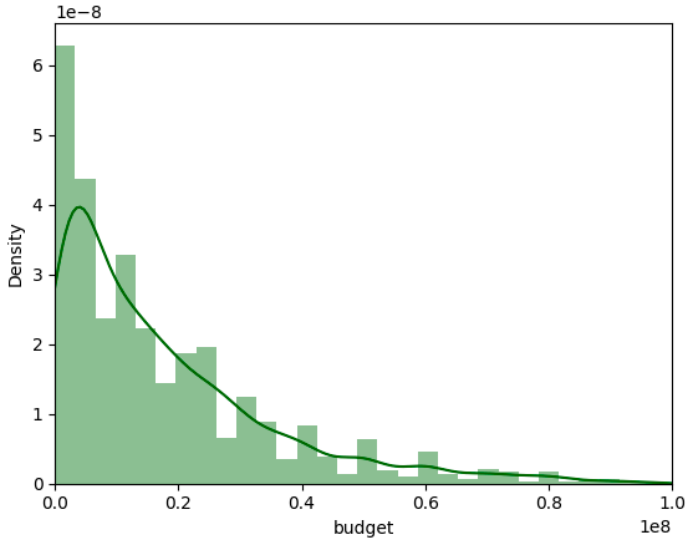

Рисунок 4.2 – Розподіл бюджету

Аналогічно розподілу прибутку, бюджет 75% фільмів зосереджено у проміжку до 25 мільйонів доларів, а інші 25% розподілені на проміжку від 25 до 255 мільйонів. Однак щільність розподілу бюджету в декілька разів вища.

Розглянемо розподіл фільмів за жанрами (див. рис. 4.3).

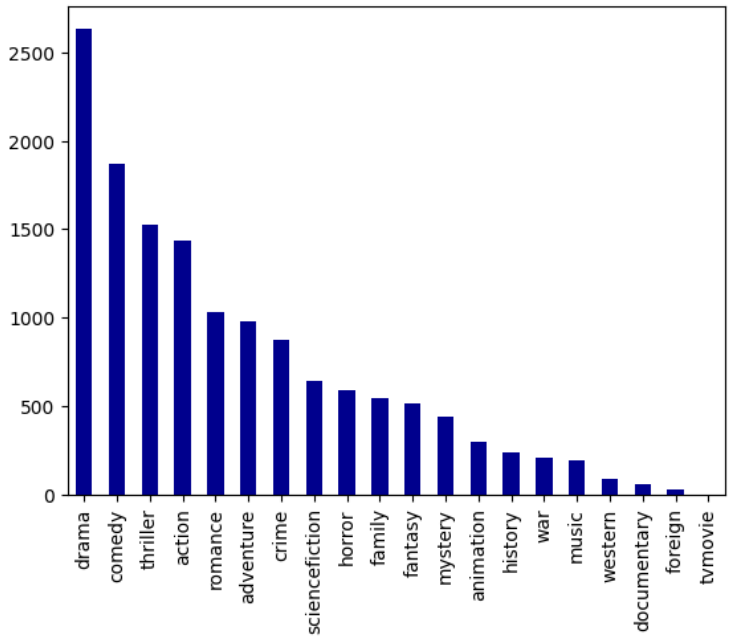

Рисунок 4.3 – Розподіл жанрів

На графіку 4.3 видно, що у вибірці наявно найбільше фільмів у жанрі драми, також значну частину складають комедії, трилери та бойовики. Треба зазначити,

що фільми у вибірці можуть відноситися до декількох жанрів одночасно. Втім, документальні фільми та вестерни займають незначну частину.

Розглянемо розподіл кількості знятих фільмів за режисерами, які працювали над цими фільмами (див. рис. 4.4).

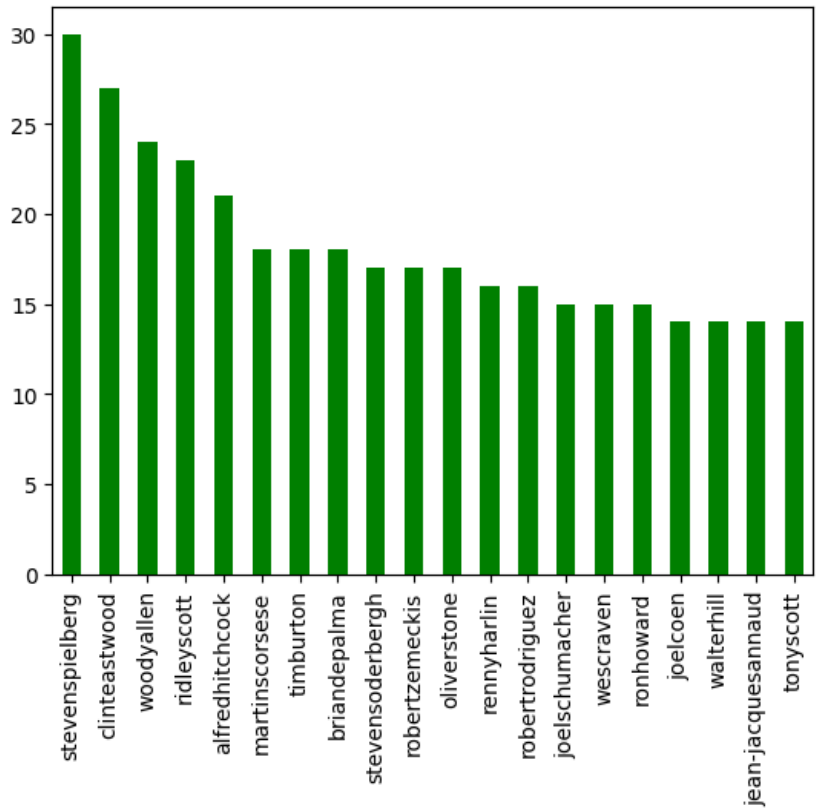

Рисунок 4.4 – Розподіл фільмів за режисерами

На графіку 4.4 можна побачити вибірку з 20 режисерів, які мають найбільше знятих фільмів, представлених у вибірці. Steven Spielberg зняв найбільшу кількість фільмів – 30, а Clint Eastwood та Woody Allen зняли 27 та 24 фільми відповідно. Також ми бачимо, що більшість режисерів зняли від 15 до 20 фільмів.

Наведемо фрагмент набору даних, який буде використаний для подальшого аналізу (див. рис. 4.5). На рисунку можна побачити, що дані у кожній з колонок представлені у різному вигляді та мають різні властивості.

genres	original_title	overview	popularity	productio...	runtime	vote_aver...
[{'id': 16, 'name': 'Animation'}, {'id': 35, 'name': 'Comedy'}, {'id': 10751, 'name': 'Family'}]	Toy Story	Led by Woody, Andy's toys live happily in his room until Andy's birthday brings Buzz Lightyear onto ...	21.946943	[{'name': 'Pixar Animation Studios', 'id': 3}]	81.0	7.7
[{'id': 12, 'name': 'Adventure'}, {'id': 14, 'name': 'Fantasy'}, {'id': 10751, 'name': 'Family'}]	Jumanji	When siblings Judy and Peter discover an enchanted board game that opens the door to a magical world...	17.015539	[{'name': 'TriStar Pictures', 'id': 559}, {'name': 'Teitler Film', 'id': 2550}, {'name': 'Interscope...	104.0	6.9
[{'id': 10749, 'name': 'Romance'}, {'id': 35, 'name': 'Comedy'}]	Grumpier Old Men	A family wedding reignites the ancient feud between next-door neighbors and fishing buddies John and...	11.7129	[{'name': 'Warner Bros.', 'id': 6194}, {'name': 'Lancaster Gate', 'id': 19464}]	101.0	6.5

Рисунок 4.5 – Опис фільмів у наборі даних

Виділимо типи даних, які наявні у вибірці:

– числові дані;

– категоріальні дані з одним або з багатьма значеннями;

– текстові дані.

Визначимо, до яких категорії відносяться наведені у наборі даних характеристики.

До числових даних можна віднести наступні характеристики:

– бюджет (budget);

– популярність (popularity);

– прибуток (revenue);

– тривалість фільму (runtime);

– середня оцінка (vote average);

– кількість голосів (vote count).

До категоріальних даних з одним або багатьма значеннями можна віднести наступні характеристики:

– жанри (genres);

– мови оригіналу (original language);

– компанії, які зняли фільм (production companies);

– країни, де знімався фільм (production countries);

– мови, на яких розмовляють у фільмі (spoken languages);

– ключові слова (keywords);

– актори (cast);

– режисер (director);

– статус фільму (status).

До текстових даних можна віднести огляд фільму (overview).

Кожна з категорій потребує різних методів попередньої обробки даних. Розглянемо їх детальніше.

3.2 Обробка числових даних

Числові дані потребують попередньої нормалізації. Нормалізацію числових даних виконаємо за допомогою шару попередньої обробки Normalization бібліотеки Tensorflow. Шар має два параметри: середнє значення (mean) та дисперсія (variance). За допомогою цих параметрів, шар зміщує та масштабує вхідні дані до розподілу з центром навколо значення параметру mean та з дисперсією, значення якої дорівнює параметру variance.

Наведемо програмну реалізацію нормалізації числових даних:

```
def preprocess_numeric_feature(dataset, feature_name):
    feature = dataset[feature_name]
    normalization_layer = layers.Normalization()
    normalization_layer.adapt(feature)
    normalized_feature = normalization_layer(feature)
    feature_n = normalized_feature.numpy()
    result_data = pd.DataFrame({feature_name: feature_n[0]})
    return result_data
```

Наведений програмний код виконує нормалізацію для кожної з числових характеристик окремо та додає нормовані значення до результуючого набору даних.

3.3 Обробка категоріальних даних

Категоріальні дані з одним або багатьма значеннями потребують складнішої попередньої обробки. Характеристики з цієї категорії у наборі даних представлені у вигляді рядка у форматі JSON, який зберігає масив об’єктів, тому виконаємо спочатку перетворення цього рядка до об’єкту у мові python за допомогою бібліотеки ast. Наступним кроком у кожного об’єкта у масиві візьмемо поле name, видалимо з нього усі пробіли та переведемо отриманий рядок до нижнього регістру. З отриманих рядків сформуємо новий масив для кожного рядка даних.

Наведемо програмну реалізацію перетворення категоріальних даних до масиву рядків:

```
def process_category_feature_to_array(data):
    item = ast.literal_eval(data)
    names = [i['name'].replace(" ", "").lower() for i in item]
    return np.asarray(names, dtype=np.str_)
```

Далі виконаємо перетворення отриманих рядків до числових значень. Використаємо шар попередньої обробки StringLookup. Цей шар для кожного унікального рядка даних ставить у відповідність числове значення, в результаті формуючи словник відповідності.

Використовуючи отриманий словник виконаємо кодування масивів рядків до категорій за допомогою шару попередньої обробки CategoryEncoding. Цей шар кодує кожен зразок у вхідних даних в масив розміру, який відповідає розміру сформованого словника на попередньому етапі, що містить одиницю для кожного значення, що наявне у цьому зразку, та нуль для значень, що не містяться у цьому зразку.

Розглянемо приклад перетворення даних.

Маємо два елементи у вибірці, кожен з яких представлений набором жанрів:

[["animation", "comedy", "family"], ["comedy", "family", "drama"]].

Виконаємо перетворення рядків даних до числових значень за допомогою шару обробки StringLookup:

[[4, 2, 1], [2, 1, 3]].

Таким чином, можна побачити, що у вибірці є 4 унікальних елементи, кожному з яких у відповідність поставлено числове значення.

Виконаємо формування категорій даних, використовуючи попереднє перетворення, за допомогою шару CategoryEncoding:

[[0., 1., 1., 0., 1.], [0., 1., 1., 1., 0.]].

Для кожного зразку даних було створено масив, у якому наявні усі елементи зі словника, та проставлено значення 1 тільки для тих елементів, які наявні у цьому зразку. Перша позиція у масиві зарезервована для значень, які не було знайдено у словнику. Таким чином, було виконано формування категорій для кожного рядку даних, які можуть розглядатися як окремі характеристики даних.

Наведемо програмну реалізацію обробки категоріальних даних з багатьма значеннями:

```
    def preprocess_category_feature(dataset, feature_name):
    feature = tf.ragged.constant(dataset[feature_name])
    lookup_layer = layers.StringLookup()
    lookup_layer.adapt(feature)
    processed_feature = lookup_layer(feature)
    category_encoder_layer = layers.CategoryEncoding(
num_tokens=lookup_layer.vocabulary_size(), output_mode="multi_hot")
    encoded_feature = category_encoder_layer(processed_feature)
    encoded_features_numpy = encoded_feature.numpy()
    data_frame = pd.DataFrame(encoded_features_numpy,
columns=lookup_layer.get_vocabulary(), dtype=float)
    frequent_features = get_features_sorted_by_frequency(data_frame)
    result_data = data_frame[frequent_features]
    return result_data
```

Оскільки кількість отриманих категорій може бути великою, і частина з них буде зустрічатися всього у декількох зразках даних, додатково виконаємо сортування отриманих категорій за частотою їхнього використання у зразках даних та оберемо для подальшого використання тільки найбільш вживані. Це дозволить зменшити кількість окремих характеристик, які не притаманні більшості записів, що у свою чергу прискорить час навчання моделі.

3.4 Обробка текстових даних

Для даних, які представлені текстовим описом, використаємо tf-idf характеристику, щоб визначити ступінь важливості окремих слів у тексті для опису певного фільму. tf-idf характеристику можна представити у наступному вигляді:

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D), \tag{13}$$

де t – слово;

d – документ;

D – колекція документів (корпус);

$tf(t, d)$ – частота слова;

$idf(t, D)$ – обернена частота документу.

Частота слова визначається як відношення кількості входжень певного слова у документі до загальної кількості слів у цьому документі:

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}, \tag{14}$$

де $f_{t,d}$ – кількість входжень слова t в документі d.

Обернена частота документу визначається як відношення кількості документів у корпусі до кількості документів, де зустрічається певне слово:

$$idf(t, D) = \log \frac{N}{n_t} \tag{15}$$

де n_t – кількість документів, де слово t зустрічається;

N – кількість документів у колекції D.

Перед використанням tf-idf характеристики виконаємо обробку текстових даних за допомогою наступних операцій [25, 26]:

– очистка тексту від нетекстових літералів, таких як розділові знаки;

– виконання токенізації тексту, виділивши окремі слова;

– приведення слова до їхньої основи (stemming);

– визначення леми слів (lemmatization);

– видалення слів, які не мають значного лінгвістичного забарвлення та часто повторюються;

Наведемо реалізацію кроків очистки тексту:

```
def process_text_row(sample, stop_words):
    stemmer = PorterStemmer()
    lemmatizer = WordNetLemmatizer()
    processed_words = []
    words = word_tokenize(sample)
    words = [w for w in words if (not w.lower() in stop_words) &
                                                    (w.isalpha())]
    for word in words:
        word = stemmer.stem(word)
        word = lemmatizer.lemmatize(word)
        processed_words.append(word)
    words = [w for w in processed_words if (not w.lower() in
                                    stop_words) & (w.isalpha())]
    result = " ".join(words)
    return result
```

Наступним кроком, за допомогою шару TextVectorization виконаємо розрахунок tf-idf характеристики для обробленого тексту.

3.5 Реалізація моделей

Для реалізації моделі багатошарової нейронної мережі використаємо клас MLPRegressor бібліотеки sklearn. Налаштуємо наступні параметри моделі: функція активації – ReLU; метод оптимізації вагів – stochastic gradient descent. Параметри кількості ітерацій навчання, рівень навчання, кількість та розмір шарів моделі визначимо експериментальним шляхом [27].

Для реалізації поліноміальної регресійної моделі використаємо клас PolynomialFeatures, який генерує матрицю поліноміальних комбінацій вхідних характеристик зі степенем, який дорівнює або менше заданого [28]:

```
transformer = PolynomialFeatures(degree=degree)
features = transformer.fit_transform(x_train)
polynomial_model = LinearRegression()
polynomial_model.fit(features, y_train)
score = polynomial_model.score(features, y_train)
```

Для реалізації B-spline регресії використаємо клас SplineTransformer, який генерує одновимірні B-spline функції для вхідних характеристик з заданими значеннями ступеня полінома та кількістю вузлів [29]:

```
tʳansformer = SplineTransformer(degree=degree, n_knots=knots,
knots='quantile')
features = transformer.fit_transform(x_train)
spline_model = LinearRegression()
spline_model.fit(features, y_train)
score = spline_model.score(features, y_train)
```

Цей клас надає можливість задавати ступінь функції, а також кількість вузлів функції B-spline.

4 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

4.1 Умови експерименту

Для виконання експериментів було використано комп’ютерне обладнання з наступними характеристиками:

– CPU: Apple M1;

– RAM: 8 GB;

– OS: MacOS 12.5.

У якості основного програмного середовища буде використано мову програмування python 3.7. Для виконання навчання та тестування моделі будемо використовувати фреймворк Tensorflow, який надає широкі можливості для обробки даних, розробки моделей та систем машинного навчання, їхнього тестування та валідації. Для роботи з даними застосуємо бібліотеку pandas та numpy. У якості середовища розробки (IDE) скористуємося Jupyter Notebook.

4.2 Критерії порівняння моделей

Щоб порівняти алгоритми та мати змогу обрати оптимальний для вирішення задачі, визначимо критерії оцінки алгоритмів:

– точність прогнозування – наскільки точно алгоритм може передбачати вихідне значення;

– час роботи алгоритму – час необхідний для навчання алгоритму;

– час підготовки даних – час необхідний для попередньої обробки даних;

Для кожного з критеріїв визначимо відповідну шкалу. Точність прогнозування вимірюється у відсотках і може приймати значення від 0 до 100. Час роботи алгоритму та час підготовки даних вимірюється у секундах. Для визначення найбільш ефективної моделі будемо використовувати лінійну адитивну згортку з нормуючими множниками.

4.3 Етапи експерименту

Визначимо етапи проведення експерименту:

– виконання попередньої обробки даних для різних моделей та порівняння часу обробки;
– визначення оптимального набору характеристик для навчання моделі (feature selection)
– виконання навчання моделей;
– розрахунок метрик-точності моделей та порівняння результатів
– порівняння отриманих результатів з результатами інших досліджень.

4.4 Виконання експерименту

Виконаємо виміри визначених критеріїв порівняння моделей.

Розпочнемо експеримент з вимірювання часу підготовки даних перед тим, як модель зможе з ними працювати (див. табл. 4.1).

Таблиця 4.1. Час підготовки даних (у мілісекундах)

Спроба	Нейрона мережа	Поліноміальна регресія	Spline регресія
1	5	57	42
2	12	37	57
3	7	37	66
4	9	42	63
5	15	34	42
Середнє	10,75	37,5	57

З отриманих результатів можна побачити, що додатковий час на підготовку даних для нейронної мережі найменший, в той час як spline регресія потребує

значно більше часу. Більший час підготовки для spline регресії обумовлено тим, що вхідні характеристики необхідно перетворити на матрицю, яка зберігає B-spline функції для кожної з цих характеристик, а для поліноміальна регресія вхідні характеристики необхідно перетворити матрицю поліноміальних комбінацій цих характеристик.

Перейдемо до відбору характеристик, які найкраще описують дані. Спочатку побудуємо матрицю кореляції між характеристиками даних, використовуючи коефіцієнт кореляції Пірсона. Це дозволить виявити статистичні залежності в даних та обрати початковий набір характеристик для аналізу (див. рис. 4.1).

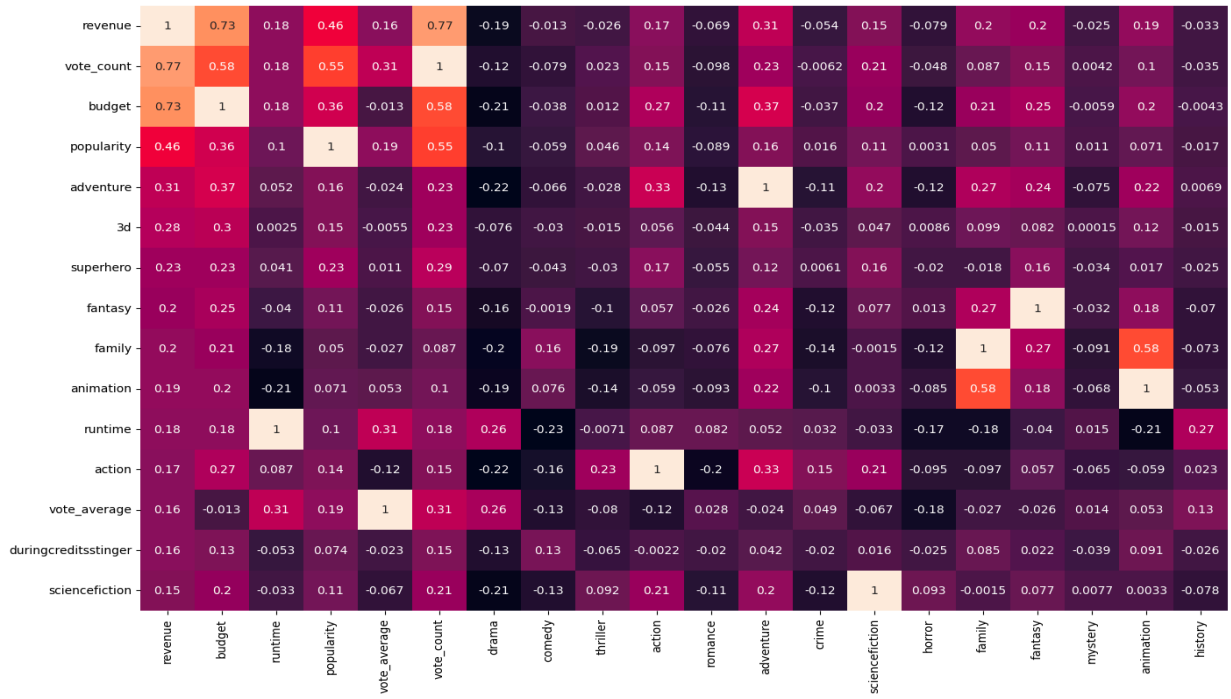

Рисунок 4.1 – Матриця кореляції характеристик

На рисунку 4.1 можна побачити, що найбільшу кореляцію з прибутком фільму мають наступні характеристики: бюджет, кількість голосів, рівень популярності, належність фільму до пригодницького, сімейного та фентезійного жанрів.

Далі виконаємо аналіз важливості цих характеристик для навчання моделі, використовуючи SelectFromModel клас. Він дозволяє виявити характеристики даних, які найбільше впливають на результат передбачення моделі, порівнюючи

розраховані ваги моделі для кожної з характеристик з середнім значенням усіх вагів (див. рис. 4.2). Можна побачити, що рівень важливості характеристик, отриманий в результаті навчання моделі значно відрізняється від результатів, отриманих в статистичного аналізу даних за допомогою коефіцієнта кореляції Пірсона.

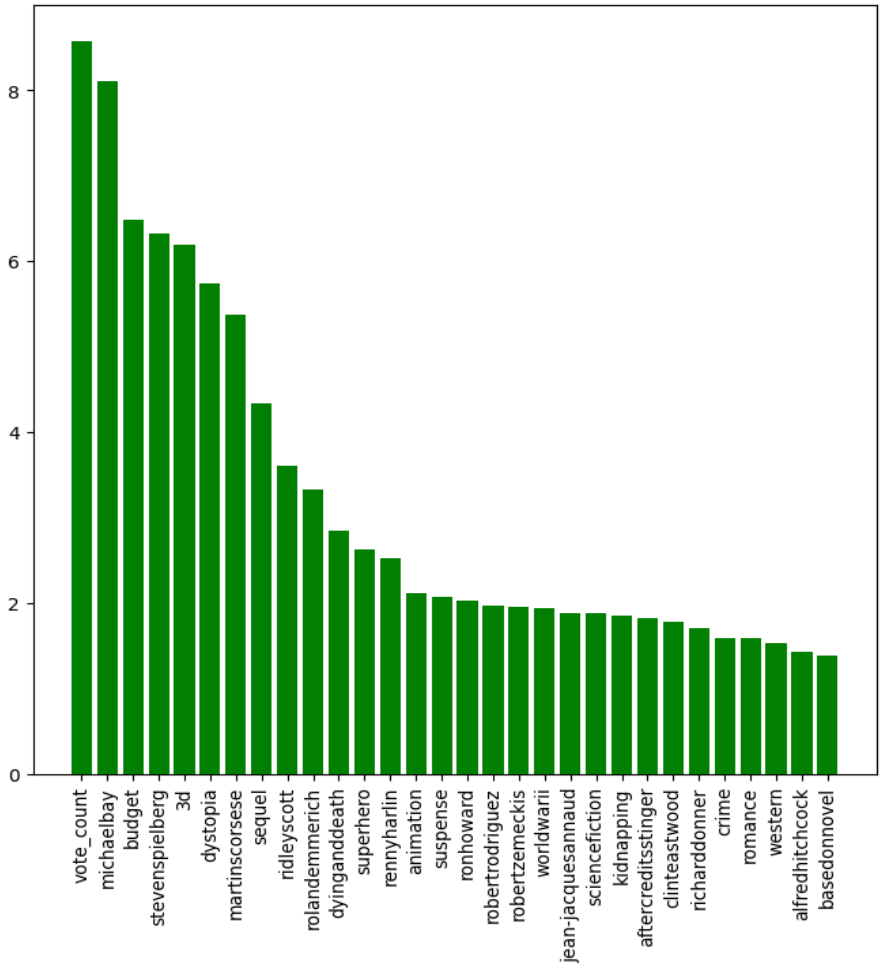

Рисунок 4.2 – Рівень важливості характеристик

Найбільше значення мають характеристики кількості голосів, бюджет, фільми, де режисерами були Michael Bay та Steven Spielberg. На діаграмі можна побачити, що інші відомі режисери теж мають значний вплив на прибуток фільму. В той час як приналежність фільму до того чи іншого жанру має менше значення для передбачення прибутку фільму. Якщо фільм є продовженням або наступною частиною серії фільмів, це теж впливає значно впливає на рівень його прибутку. Це можна пояснити тим, що кінокомпанії, які випускають продовження фільму, мають

значно більше даних про успіх попереднього фільму та що на це вплинуло, ніж можна отримати з відкритих джерел, тому рішення про зйомки продовження є обґрунтованим.

Визначивши важливість характеристик у навчанні моделі, перейдемо до визначення кількості характеристик, необхідних для ефективного навчання моделі та отримання найкращих результатів прогнозування.

Виконаємо серію експериментів, зменшуючи кількість характеристик, які використовує модель для навчання. Для порівняння ефективності роботи моделі, будемо використовувати метрики R^2 – коефіцієнт детермінації та середню абсолютну помилку (mean absolute error – MAE) (див. табл. 4.2) [30].

Таблиця 4.2. Залежність метрик від кількості характеристик моделі

Кількість	Нейрона мережа		Поліноміальна регресія		Spline регресія	
	R^2	*MAE*	R^2	*MAE*	R^2	*MAE*
95	0.79	44,017,316	0.76	48,036,825	0.5	87,808,004
68	0.75	48,174,270	0.70	49,862,239	0.63	63,091,778
50	0.76	42,519,380	0.76	42,198,162	0.70	47,169,635
40	0.76	44,800,872	0.76	44,805,337	0.72	49,965,401
30	0.75	47,054,065	0.76	46,390,542	0.73	47,948,089
20	0.76	45,940,372	0.75	47,128,688	0.71	48,548,421

Побудуємо графік залежності середньої абсолютної помилки від кількості характеристик (див. рис. 4.3).

На графіку можна побачити, що найменше значення помилки для усіх моделей було отримано, коли кількість вхідних характеристик дорівнювала 50. При зменшені кількості характеристик середня абсолютна помилка збільшується повільно. При цьому коефіцієнт детермінації майже не змінюється. Це можна пояснити тим, що характеристики, які залишились, краще описують залежність у даних та є найбільш важливими для прогнозування. Окремо варто зазначити, що

для spline регресії найкраще значення R^2 було отримане, коли кількість характеристик дорівнювала 30, при цьому значення помилки було таким, як і для 50 характеристик.

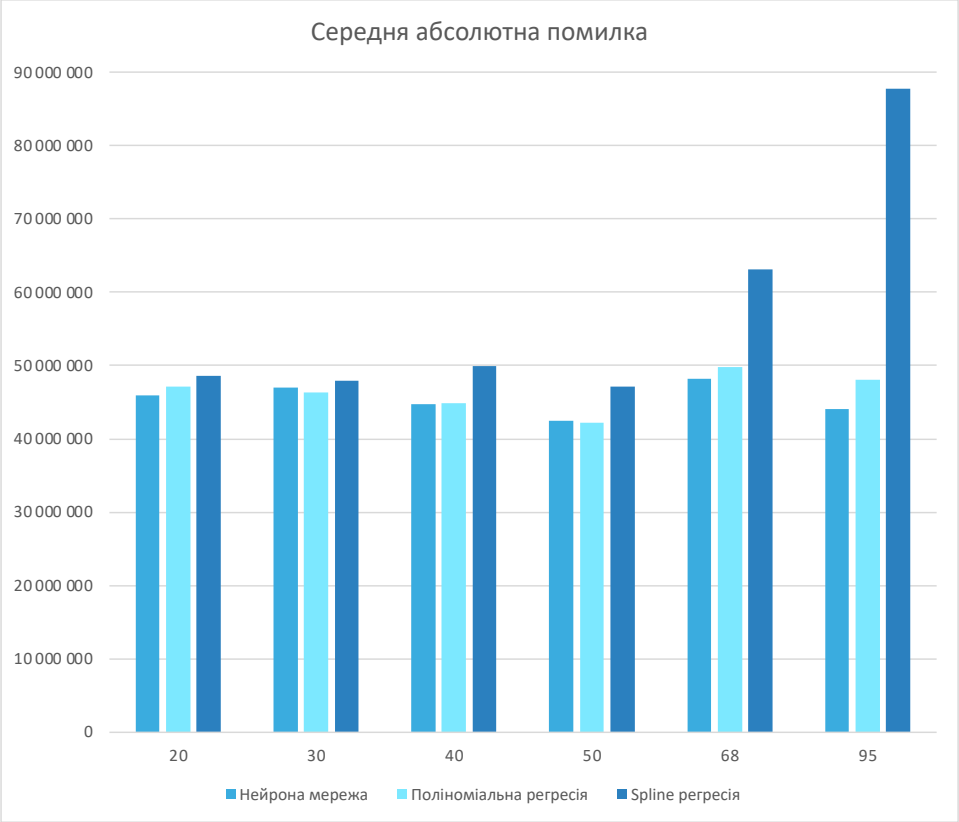

Рисунок 4.3 – Залежність помилки від кількості характеристик

При збільшені кількості характеристик для spline регресії значення R^2 значно зменшується, а значення помилки стрімко зростає. Це обумовлено тим, що модель не може точно налаштувати параметри для великої кількості вхідних характеристик та виділити закономірності. Для нейронної мережі значення помилки теж зростає, але повільніше. При цьому для експерименту з максимальної кількістю характеристик значення R^2 збільшилось. Така поведінка обумовлена тим, що за рахунок наявності декількох прихований шарів, модель краще підлаштовує параметри під час навчання, виявляючи складні залежності в даних. Аналогічну поведінку можна побачити і для поліноміальної регресії. Оскільки кількість параметрів моделі збільшується поліноміально відносно кількості вхідних

характеристик, модель матиме багато окремих параметрів, що дозволяють краще описувати залежності в даних.

Однак зі значень метрики абсолютної помилки можна побачити, що це не завжди є оптимальним рішенням, оскільки більша кількість параметрів призводить до перенавчання моделі на тренувальній вибірці, погіршуючи можливості моделі до узагальнення, тому на тестовій вибірці така модель показує менш точне значення.

Виходячи з отриманих результатів, можна зробити висновок, що використання 50 найбільш значимих параметрів є оптимальним рішенням для усіх розглянутих моделей. Для подальших експериментів буде використано саме цю кількість вхідних характеристик даних.

Наступним кроком виконаємо вимірювання часу навчання моделей (див. табл. 4.3).

Таблиця 4.3. Час навчання моделей (у мілісекундах)

Спроба	Нейрона мережа	Поліноміальна регресія	Spline регресія
1	3987	2511	2313
2	3497	2501	2264
3	4120	2536	2180
4	3201	2528	2278
5	4588	2566	2196
Середнє	3878,6	2528,4	2246,2

З результатів експерименту видно, що нейронна мережа потребує більше часу на навчання, ніж інші підходи. Це обумовлено кількістю прихованих шарів моделі, розміром кожного з них та кількістю епох навчання. Окрім цього, ми бачимо, що різниця у часі навчання поліноміальна регресія та spline регресія є значною.

Якщо кількість вхідних характеристик перевищує 100, час навчання поліноміальної регресії значно збільшується, оскільки кількість вхідних параметрів

моделі збільшується поліноміально відносно кількості вхідних характеристик та експоненціально відносно обраного ступеня поліному. Для Spline регресії зі збільшенням кількості вхідних характеристик збільшується тільки кількість вузлів, а значення ступеня не змінюється, тому час навчання змінюється повільніше.

Далі виміряємо точність прогнозування моделей на тренувальній та тестовій вибірках, використавши R^2 – коефіцієнт детермінації та середню абсолютну помилку (див. табл. 4.4).

Таблиця 4.4. Метрики моделей

Модель	R^2		MAE	
	Тренування	Тест	Тренування	Тест
Нейрона мережа	0,79	0,76	40 986 071	42 519 380
Поліноміальна регресія	0,8	0,76	40 278 759	42 198 162
Spline регресія	0,74	0,73	44 690 691	47 948 089

З отриманих результатів видно, що на тестовій вибірці найкращий результат був отриманий за допомогою алгоритму поліноміальної регресії, хоча інші моделі показали дуже схожі результати. На тестовому наборі даних ми бачимо, що нейрона мережа та поліноміальна регресія мають однакові значення метрики R^2. Однак spline регресія має значно нижчий показник. Це можна пояснити тим, що налаштування параметрів моделі для всього набору даних виявилось ефективніше, ніж налаштування параметрів для його окремих частин за рахунок використання B-spline функцій.

Виходячи з отриманих результатів, можна зробити висновок, що багатошарові нейронні мережі прямого зв’язку краще підходять для вирішення задачі прогнозування прибутку фільмів, оскільки вони краще масштабуються, дозволяючи обробляти велику кількість вхідних характеристик, дають найбільш точний результат як під час навчання, так і під час валідації, однак мають більший час навчання, який не є вирішальним у розглянутій предметній галузі.

Порівняємо отримані результати з висновками статті «Forecasting Movie Box Office Profitability». В результаті експериментів, як і у вказаній статті, було визначено, що наступні характеристики мають значний вплив на прибуток фільмів: бюджет, режисер та чи є фільм продовженням (sequel). Також було виявлено, що кількість голосів має великий вплив на прогнозування, як і певні характеристики сюжету фільму, таких як антиутопія, супер герой, війна. Аналогічно було визначено, що нейроні мережі краще підходять для вирішення задачі прогнозування прибутковості фільмів.

За допомогою побудованої моделі поліноміальної регресії виконаємо передбачення прибутку фільмів та порівняємо результати з реальними значеннями (див. рис. 4.4).

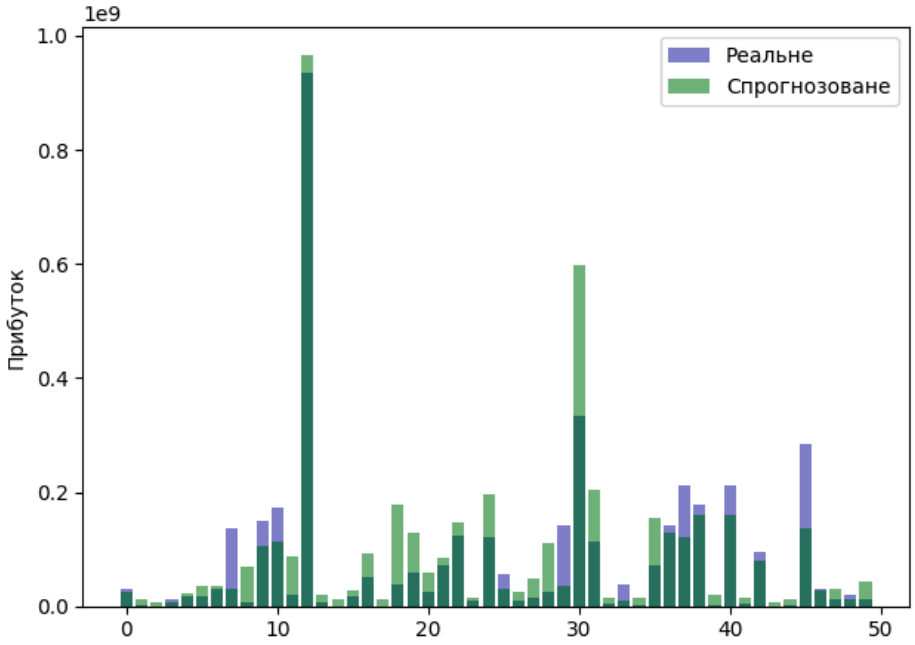

Рисунок 4.4 – Порівняння прогнозованого та реального значень прибутку

На рисунку 4.4 можна побачити, що у більшості випадків модель дає завищену оцінку прибутку фільмів, якщо він менше, ніж 100 мільйонів. При цьому для фільмів, прибуток яких більший, модель дає менший результат, ніж реальний. Як видно з таблиці 4.4 середня абсолютна помилка моделі складає 40 мільйонів, що є вагомим значенням для передбачення прибутку малобюджетних фільмів, в той час як для фільмів зі значним прибутком, ця помилка не є значною.

4.5 Подальше дослідження

Для подальшого дослідження та поліпшення точності прогнозування моделей планується:

– формування більшого набору вхідних даних таким чином, щоб розподіл значень прибутку відповідав нормальному розподілу, що повинно покращити точність передбачення моделей;
– збільшення кількості метрик оцінки точності прогнозування моделей для глибшого аналізу роботи моделей;
– додавання нових характеристик фільмів;
– дослідження методів виявлення схожих фільмів та прогнозування прибутку на основі додаткових даних у певному кластері фільмів;
– використання підходу трансферного навчання (transfer learning) для точнішого налаштування моделі для окремих груп фільмів;
– розробка програмної системи, яка, на основі вхідних даних користувача, робить комплексний аналіз фільму та прогнозування прибутку, надаючи користувачу інформацію про схожі фільми, вплив певних характеристик на прибуток та рекомендації, які можуть допомогти збільшити потенційний прибуток.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було проаналізовано ринок кіноіндустрії у сучасних умовах, визначено його розмір та найбільші компанії індустрії. Було проаналізовано, яким чином відбувається процес прийняття рішень щодо зйомок нових фільмів, розглянуто можливості компаній до швидкої адаптації до умов ринку, який постійно змінюється. Визначено та розглянуто, які системи та підходи допомагають приймати рішення та оцінювати потенціал нового продукту. Виявлено проблему прогнозування прибутковості фільмів та серіалів у сучасних умовах. Також було проаналізовано існуючі дослідження за цією темою.

Було сформовано завдання до кваліфікаційної роботи, визначено етапи виконання наукового дослідження, проаналізовано існуючі методи прогнозування та обрано наступні підходи:

– багатошарова нейрона мережа прямого зв’язку;

– поліноміальна регресія;

– сегментована регресія;

Було розглянуто математичне представлення обраних підходів, виконано програмну реалізацію моделей за допомогою мови python та додаткових бібліотек машинного навчання, проведено серію експериментів та порівняно характеристики різних моделей.

В результаті експериментів, було визначено, що:

– багатошарова нейрона мережа прямого зв’язку має високий рівень точності та краще масштабуються;

– поліноміальна регресія також дозволяє отримати високій рівень точності прогнозування, втім значна кількість параметрів значно збільшує час навчання моделі та може привести до перенавчання за рахунок великої кількості параметрів самої моделі;

– сегментована регресія має нижчий результат точності прогнозування на заданому наборі даних, втім масштабується краще за поліноміальну регресія при збільшені кількості вхідних параметрів.

– розподіл даних значно впливає на точність прогнозування моделі.

Ми порівняли отримані результати з результатами існуючих досліджень, та підтвердили їхні результати, зокрема доцільність використання нейронних мереж, а також важливість характеристик бюджету, режисеру та продовження на прибуток фільму.

Було визначено задачі для подальшого дослідження, які допоможуть покращити точність моделі та дозволять використовувати її у реальних умовах, а саме: кращий підбір даних для навчання, збільшення розміру вибірки, додавання нових характеристик фільмів та використання додаткових методів, таких як трансферне навчання, для кращої адаптації моделі до певних груп фільмів.

Таким чином, за результатами дослідження можна зробити висновок, що багатошарова нейрона мережа прямого зв’язку найкраще підходять для вирішення задачі прогнозування прибутковості фільмів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Video-on-demand users 2017-2026 by segment. Statista. URL: https://www.statista.com/forecasts/456771/video-on-demand-users-in-the-world-forecast (дата звернення: 21.04.2023).

2. Box office revenue in the U.S. and Canada 2022. Statista. URL: https://www.statista.com/statistics/187069/north-american-box-office-gross-revenue-since-1980 (дата звернення: 21.04.2023).

3. North American box office: Tickets sold 2022. Statista. URL: https://www.statista.com/statistics/187076/tickets-sold-at-the-north-american-box-office-since-2001 (дата звернення: 21.04.2023).

4. U.S. & Canada: Number of movie tickets sold 2022. Statista. URL: https://www.statista.com/statistics/187073/tickets-sold-at-the-north-american-box-office-since-1980 (дата звернення: 21.04.2023).

5. U.S. per capita movie attendance by age 2021. Statista. URL: https://www.statista.com/statistics/380129/number-of-movies-seen-in-the-theater-usa-ba-ag (дата звернення: 21.04.2023).

6. Why some people do not go to the movies 2022. Statista. URL: https://www.statista.com/statistics/1315975/reasons-infrequent-moviegoers-do-not-go-movie-theaters-united-states (дата звернення: 21.04.2023).

7. Movie genre distribution in the United States and Canada from 2010 to 2021. Statista. URL: https://www.statista.com/statistics/668712/movie-genres-in-north-america-by-average-box-office-revenue (дата звернення: 21.04.2023).

8. Cinelytic Home Page. Cinelytic. URL: https://www.cinelytic.com (дата звернення: 21.04.2023).

9. How 20th Century Fox uses ML to predict a movie audience. Google Cloud Blog. URL: https://cloud.google.com/blog/products/ai-machine-learning/how-20th-century-fox-uses-ml-to-predict-a-movie-audience (дата звернення: 21.04.2023).

10. Dye M., Ekanadham C., Saluja A. Supporting content decision makers with machine learning. Netflix TechBlog. URL: https://netflixtechblog.com/supporting-

content-decision-makers-with-machine-learning-995b7b76006f (дата звернення: 21.04.2023).

11. Movie market summary 1995 to 2023. The Numbers. URL: https://www.the-numbers.com/market (дата звернення: 21.04.2023).

12. Top 20 countries by filmed entertainment revenue. Statista. URL: https://www.statista.com/statistics/296431/filmed-entertainment-revenue-worldwide-by-country (дата звернення: 21.04.2023).

13. U.S. & Canada: Market share of film studios 2021. Statista. URL: https://www.statista.com/statistics/187171/market-share-of-film-studios-in-north-america-2010 (дата звернення: 21.04.2023).

14. Netflix, Inc. (NFLX) Stock Price. Yahoo Finance. URL: https://finance.yahoo.com/quote/NFLX?p=NFLX (дата звернення: 21.04.2023).

15. Galvão M., Henriques R. Forecasting Movie Box Office Profitability. Journal of Information Systems Engineering & Management. 2018. T. 3, № 3. URL: https://doi.org/10.20897/jisem/2658 (дата звернення: 01.05.2023).

16. Lee S., KC B., Choeh J. Y. Comparing performance of ensemble methods in predicting movie box office revenue. Heliyon. 2020. T. 6, № 6. URL: https://doi.org/10.1016/j.heliyon.2020.e04260 (дата звернення: 01.05.2023).

17. Fabozzi F. J., Focardi S. M., Rachev, S. T. Appendix E: Model Selection Criterion: AIC and BIC. The Basics of Financial Econometrics. 2014. С. 399–403. URL: https://doi.org/10.1002/9781118856406.app5 (дата звернення: 01.05.2023).

18. API Overview. The Movie Database. URL: https://www.themoviedb.org/documentation/api (дата звернення: 21.04.2023).

19. Feed-Forward Neural Networks. Mukul Rathi. URL: https://mukulrathi.com/demystifying-deep-learning/feed-forward-neural-network (дата звернення: 21.04.2023).

20. Jurafsky D., Martin J. H. Logistic Regression. Stanford University. URL: https://web.stanford.edu/~jurafsky/slp3/5.pdf (дата звернення: 01.05.2023).

21. Ng A. Supervised learning. Stanford Engineering Everywhere. URL: https://see.stanford.edu/materials/aimlcs229/cs229-notes1.pdf (дата звернення: 01.05.2023).

22. Higher-order Multivariable Polynomial Regression to Estimate Human Affective States / J. Wei та ін. Scientific Reports. 2016. Т. 6, № 1. URL: https://doi.org/10.1038/srep23384 (дата звернення: 01.05.2023).

23. Eilers P. H. C., Marx B. D. Flexible smoothing with B-splines and penalties. Statistical Science. 1996. Т. 11, № 2. С. 89–121. URL: https://doi.org/10.1214/ss/1038425655 (дата звернення: 01.05.2023).

24. Polynomial Interpolation. Scikit Learn. URL: https://scikit-learn.org/stable/auto_examples/linear_model/plot_polynomial_interpolation.html (дата звернення: 01.05.2023).

25. Khovrat A., Kobziev V., Nazarov A. та ін. Parallelization of the VAR Algorithm Family to Increase the Efficiency of Forecasting Market Indicators During Social Disaster. Information Technology and Implementation, Kyiv, Ukraine. 2022. С. 222–233. URL: https://ceur-ws.org/Vol-3347/Paper_19.pdf (дата звернення: 01.05.2023).

26. Effectiveness of Preprocessing Algorithms for Natural Language Processing Applications / K. Smelyakov та ін. 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine. 2020. URL: https://doi.org/10.1109/picst51311.2020.9467919 (дата звернення: 01.05.2023).

27. MLP Regressor. Scikit Learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html (дата звернення: 01.05.2023).

28. Polynomial Features. Scikit Learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html (дата звернення: 01.05.2023).

29. Spline Transformer. Scikit Learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.SplineTransformer.html (дата звернення: 01.05.2023).

30. Smelyakov K., Bizkrovnyi O., Sharonova N. та ін. Building of Regression Models for Cryptocurrency Price Prediction. COLINS 2022, Gliwice, Poland. 2022. Т. 1, № 6. С. 1216–1232.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

24. Khovrat A., Kobziev V., Nazarov A. та ін. Parallelization of the VAR Algorithm Family to Increase the Efficiency of Forecasting Market Indicators During Social Disaster. Information Technology and Implementation, Kyiv, Ukraine. 2022. С. 222–233. URL: https://ceur-ws.org/Vol-3347/Paper_19.pdf (дата звернення: 01.05.2023).

25. Effectiveness of Preprocessing Algorithms for Natural Language Processing Applications / K. Smelyakov та ін. 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine. 2020. URL: https://doi.org/10.1109/picst51311.2020.9467919 (дата звернення: 01.05.2023).

29. Smelyakov K., Bizkrovnyi O., Sharonova N. та ін. Building of Regression Models for Cryptocurrency Price Prediction. COLINS 2022, Gliwice, Poland. 2022. Т. 1, № 6. С. 1216–1232. URL: https://ceur-ws.org/Vol-3171/paper90.pdf (дата звернення: 01.05.2023).