НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

До захисту допущено:
Завідувач кафедри

_____ Оксана ТИМОЩУК
«___»_____20__ р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою «Системи і методи штучного інтелекту»
спеціальності 122 «Комп'ютерні науки»

на тему: «Система прогнозування результатів виконання проектів на платформі Kickstarter»

Виконав (-ла):
студент (-ка) IV курсу, групи КА-76
Хомич Олександр Ростиславович _____

Керівник:
професор, д.т.н., Зайченко Олена Юріївна _____

Консультант з економічного розділу:
доцент, к.е.н., Рощина Надія Василівна _____

Консультант з нормоконтролю:
доцент, к.т.н. Коваленко Анатолій Єпіфанович _____

Рецензент:
професор, д.т.н., АПЕПС ТЕФ Аушева Наталія Миколаївна _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2021 року

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

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітня програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ Оксана ТИМОЩУК
«26» травня 2021 р.

ЗАВДАННЯ

на дипломну роботу студенту

Хомичу Олександру Ростиславовичу

1. Тема роботи «Система прогнозування результатів виконання проектів на платформі Kickstarter», керівник роботи Зайченко Олена Юріївна, професор, доктор технічних наук, затверджені наказом по університету від «26» травня 2021 р. № 1344-с.

2. Термін подання студентом роботи: 08.06.2021.

3. Вихідні дані до роботи: інформація про кампанії на сайті Kickstarter за 2009-2021 роки, отримані за допомогою скрапінгу.

4. Зміст роботи: основні поняття, завдання і напрямки краудфандингу, дослідження предметної області, теоретичні основи методів дослідження, огляд даних та їх попередня обробка, опис програмного продукту, фінансово-економічний аналіз.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): Актуальність роботи, Постановка задачі дослідження, Критерії оцінювання якості моделі, Програмний продукт, Приклад виконання програми, Практична значущість результатів роботи.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Рощина Н. В., доцент		

7. Дата видачі завдання: 21 лютого 2021 року.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Вивчення літератури за темою роботи	13.03.2021	
2.	Підготовка першого розділу	21.04.2021	
3.	Підготовка другого розділу	01.05.2021	
4.	Розробка програмного продукту	16.05.2021	
5.	Підготовка третього розділу	23.05.2021	
6.	Підготовка економічної частини	31.05.2021	
7.	Оформлення розділів відповідно до нормоконтролю	01.06.2021	
8.	Підготовка презентації доповіді	29.05.2021	
9.	Оформлення дипломної роботи	02.06.2021	

Студент Олександр ХОМИЧ

Керівник Олена ЗАЙЧЕНКО

РЕФЕРАТ

Дипломна робота: 138 с., 10 табл., 31 рис., 3 додатки, 23 джерела.

ПРОГНОЗУВАННЯ. КРАУДФАНДИНГ. СТАРТАП. KICKSTATER. ГРАДІЄНТНИЙ БУСТИНГ.

Тема: система прогнозування результатів виконання проектів на платформі Kickstarter.

У роботі розглянуто різні моделі класифікації та регресії для прогнозування результатів виконання проектів на краудфандинговій платформі Kickstarter.

Об’єкт дослідження: застосування математичних методів прогнозування в фінансово-інвестиційній сфері.

Предмет дослідження: методи машинного навчання та програмні засоби їх реалізації.

Мета роботи: розробка програмного забезпечення для прогнозу результатів виконання проектів на платформі Kickstarter.

Створено програмний продукт для прогнозування результатів виконання проектів на платформі Kickstarter. Для проведення аналізу було використано реальні дані з сайту Kickstarter, отримані за допомогою скрапінгу за період з 2009 по 2021 роки.

ABSTRACT

Thesis: 138 p., 10 tabl., 31 fig., 3 appendices, 23 sources.

FORECASTING. CROWDFUNDING. STARTUP. KICKSTATER. GRADIENT BOOSTING.

Topic: system for forecasting the projects results on the Kickstarter platform.

The paper considers different classification and regression models for forecasting projects results on the crowdfunding platform Kickstarter.

Object of research: application of mathematical forecasting methods in the financial and investment sphere.

Subject of research: methods of machine learning and software for their implementation.

Purpose: development of software for forecasting the results of projects on the Kickstarter platform.

A software product has been created for forecasting the results of project implementation on the Kickstarter platform. The analysis used real data from the Kickstarter site, obtained by scraping for the period from 2009 to 2021.

ЗМІСТ

ВСТУП

Часто люди, які створюють ідеї чи стартапи, які дійсно потрібні суспільству та певній галузі, не можуть втілити свій проект у життя лише через брак інформаційних навичок. Тому багато стартапів задаються питанням: як представити свій інноваційний та перспективний (і навіть дуже крутий) проект, щоб викликати інтерес у потенційного інвестора?

За допомогою краудфандингу фінансується широкий спектр комерційних підприємницьких підприємств, таких як художні та творчі проекти, медичні витрати, подорожі та проекти соціального підприємництва, орієнтовані на громаду. Незважаючи на те, що краудфандинг, як вважають, тісно пов'язаний із стійкістю, емпіричне підтвердження показало, що стійкість відіграє лише незначну роль у краудфандингу. Його використання також критикується за фінансування шарлатанства, особливо дорогих та шахрайських методів лікування раку.

Зокрема, в межах даної роботи вивчаються кілька питань, важливих для розуміння швидкого зростання краудфандингу, та представлений попередній аналіз деяких основних динамічних явищ. Спочатку буде зроблений огляд краудфандингу, включаючи робоче визначення та пояснення того, як працює краудфандинг. Далі буде описано природу даних краудфандингу, які використовуються для дослідницьких аналізів, і наведені основні описові результати щодо краудфандингу. Після цього буде проведено кілька поглиблених аналізів того, коли краудфандинг призводить до успішної розробки продукту; змінні, пов’язані з успіхом у краудфандингових підприємствах. На додаток до опису основної динаміки краудфандингу, аналіз цієї нової обстановки також дає загальне уявлення про те, як характеристики засновників та спосіб їх презентації можуть впливати на результати фінансування підприємництва.

Окрім теоретичної частини роботи, також буде описаний створений програмний продукт, заснований на використанні моделей машинного навчання, будуть описані алгоритми та процедури, що були використані для побудови та інтерпретації результатів.

РОЗДІЛ 1 ОСНОВНІ ПОНЯТТЯ, ЗАВДАННЯ І НАПРЯМКИ КРАУДФАНДИНГУ. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Краудфандинг як явище. Основні поняття

Останнім часом Інтернет послужив не лише фактором, що сприяє явищу, коли люди з усього світу збираються разом, щоб внести фінансові ресурси для фінансування справ, ідей, проектів, продуктів або бізнесу; але це також стимулювало його дифузію та зростання, спонукаючи її до опори для фінансування. Незважаючи на те, що це не зовсім нове явище, в сенсі того, що окремі люди збираються разом, щоб внести або об'єднати фінансові ресурси, Інтернет дозволив процвітати явищу, яке називають «краудфандингом». В статті журналу Fortune краудфандинг називається демократизацією збору коштів [1].

В умовах стійкого зростання та диверсифікації краудфандингу його соціально-економічний потенціал стає все більш очевидним. Від допомоги людям у збиранні грошей на покриття медичних витрат та витрат на поховання до надання капіталу на ранніх стадіях новаторам, які допомагають реалізувати творчі ідеї, краудфандинг продовжує доводити свою цінність. Найвидатнішим прикладом краудфандингу є Oculus Rift – компанія, яка спеціалізується на випуску ігрових апаратних засобів для віртуальної реальності, що фінансується за допомогою механізму краудфандингу, згодом придбала Facebook приблизно на 2 мільярди доларів [2]. Промислові звіти про краудфандинг показують, що в 2019 році ринок краудфандингу зріс на 167% і, як очікується, не лише досягне 90-96 млрд доларів до 2025 року [1], але також повинен перерости світовий ринок венчурного капіталу в 1,8 рази того ж часу.

1.1.1 Визначення краудфандингу

Краудфандинг – це практика фінансування проекту або підприємства шляхом залучення невеликої суми грошей від великої кількості людей, як правило, через

Інтернет. Краудфандинг – це форма краудсорсингу та альтернативного фінансування. Це дуже ефективний спосіб отримати трохи грошей, і, наприклад, у 2015 році за допомогою краудфандингу було залучено понад 34 мільярди доларів у всьому світі [1]. Краудфандинг використовується різними людьми та групами, такими як підприємці, благодійні організації, компанії та приватні особи. Тому різні цілі краудфандингу можуть бути використані: від збору грошей на фінансування нового автомобіля або відпустки, зйомки фільму, до створення компанії чи розробки інноваційної ідеї. Метою може бути все, тільки поки люди готові інвестувати в ідею.

Незважаючи на те, що подібні концепції також можуть бути реалізовані за допомогою підписки на замовлення поштою, подій переваг та інших методів, термін краудфандинг відноситься до реєстрів, що опосередковуються Інтернетом. Ця сучасна модель краудфандингу, як правило, базується на трьох типах суб'єктів: ініціатор проекту, який пропонує фінансувати ідею чи проект, окремі особи або групи, які підтримують ідею, та організація модерації ("платформа"), яка об'єднує сторони запустити ідею. За допомогою краудфандингу фінансується широкий спектр комерційних, підприємницьких підприємств, таких як художні та творчі проекти, медичні витрати, подорожі та проекти соціального підприємництва, орієнтовані на громаду. Його використання також критикували за фінансування шарлатанства, особливо дорогих та шахрайських методів лікування раку.

Тобто суть краудфандингу полягає в тому, що у вас є гарна ідея, але для її реалізації потрібні гроші, а є звичайні люди (спонсори), які готові профінансувати ваш проект за певну винагороду (копії товару, знижки на послуги/продукцію). Краудфандинг використовується у всьому світі, але є найбільш регульованим у США.

Краудфандинг черпає натхнення у таких концепціях, як мікрофінансування та краудсорсинг, але представляє власну унікальну категорію збору коштів, сприяючи зростаючій кількості веб-сайтів, присвячених цій темі. Як і в будь-якій новій галузі, популярні та академічні концепції краудфандингу перебувають у стані еволюційного руху, що робить повні визначення довільно обмежувальними. В

одному з небагатьох опублікованих оглядів теми визначають краудфандинг як «відкритий заклик, по суті через Інтернет, щодо надання фінансових ресурсів або у формі пожертв, або в обмін на якусь винагороду та/або права голосу на підтримку ініціатив для конкретних цілей» [3].

Однак навіть це експансивне визначення потенційно не включає приклади, які вчені в різних галузях називали "краудфандингом", включаючи однорангове кредитування в Інтернеті та акції збору коштів, ініційовані шанувальниками музичної групи, серед багатьох інших випадків. Отже, широке визначення краудфандингу є невловимим, особливо оскільки краудфандинг охоплює так багато поточних (і, можливо, майбутніх) застосувань у багатьох дисциплінах. Натомість існує твердження, що для науковців, які вивчають нові підприємства та підприємницьке фінансування, де краудфандинг є особливо помітним, кращим є більш вузьке визначення терміну [4]. У підприємницькому контексті наступне визначення надає конкретність, забезпечуючи при цьому простір для подальшої еволюції концепції: краудфандинг відноситься до зусиль підприємницьких осіб та груп – культурних, соціальних та комерційних – для фінансування своїх підприємств, спираючись на відносно невеликі внески від відносно великої кількості осіб, що користуються Інтернетом, без стандартних фінансових коштів посередників.

Два аспекти визначення не розглядаються у цьому формулюванні: мета зусиль краудфандингу та мета інвесторів. Обидва типи цілей, очевидно, мають велике значення, але вони також є аспектами краудфандингу, що зазнають найбільших змін.

1.1.2 Основні поняття та ключові слова

Засновник – це особа, команда або організація, яка ініціює кампанію збору коштів на платформі краудфандингу з певною метою. Засновника також називають

різними іншими іменами, наприклад, ініціатором, одержувачем, командою/групою проектів тощо.

Фінансист – ця особа зобов'язується фінансово підтримати ініціативу, яку фінансують краудфандинги. Фінансисти можуть розраховувати не отримати відчутної віддачі, продукту, віддачі від своїх інвестицій з відсотками або власності на акції. Залежно від зобов’язань окремого проекту, донатора також можна назвати донором, спонсором, позикодавцем, інвестором, замовником перед покупкою тощо.

Платформа для краудфандингу – це Інтернет-платформа, яка використовується як портал для демонстрації ідеї кампанії та відповідну інформацію, таку як детальний опис проекту, зображення, аудіо та відео, варіанти застави, тривалість проекту, деталі засновника тощо.

Кампанія – це відкритий заклик до фінансової підтримки проекту чи ідеї протягом певного періоду. Важливими елементами кампанії є опис проекту та ефективна взаємодія, такі як відповідь на запити та занепокоєння, надання оновлень та інтеграція відгуків. Зазвичай кампанія триває 30–45 днів.

Інвестиційний поріг визначає успіх або невдачу краудфандингової кампанії. Найпопулярнішим варіантом є модель All-or-Nothing (AON – все або нічого), коли всі кошти повертаються спонсорам, якщо вказана ціль кампанії не досягнута. Тим часом модель Keep-it-All (KIA – «залишай все собі») надає засновнику доступ до заставлених коштів, навіть коли цільова сума не досягнута.

Краудфандинг – метод залучення капіталу колективними зусиллями друзів, сім'ї, клієнтів та окремих інвесторів.

Акредитований інвестор – фізична особа, чистий капітал якої перевищує 1 млн. дол. США або дохід якої перевищує 200 тис. дол. США за останні 2 роки. В даний час Комісія з цінних паперів та бірж США (SEC) санкціонує, що лише акредитовані інвестори мають законну можливість інвестувати в приватні компанії.

Краудфандинг на основі пожертв – будь-яка краудфандингова кампанія, при якій інвесторам чи вкладникам не приносять фінансової віддачі.

Краудфандинг, що базується на винагородах – будь-яка краудфандингова кампанія, в якій беруть участь люди, які вносять вклад у ваш бізнес в обмін на «винагороду», як правило, форму товару чи послуги, яку пропонує ваша компанія.

Краудфандинг на основі акціонерного капіталу – будь-яка краудфандингова кампанія, яка дозволяє вкладникам стати частковими власниками вашої компанії, обмінюючи капітал на акції.

1.1.3 Різниця між краудфандингом та традиційним збором коштів

Традиційний збір коштів для бізнес-проекту або підприємства включає в себе розміщення кількох інвесторів, банків або венчурних капіталістів на значну суму. Але в рамках краудфандингу "натовп" фінансує ідею чи проект через Інтернет-платформу, тобто Інтернет використовується для спілкування з людьми, які можуть внести відносно невеликі суми у бізнес-ідею, проект чи підприємство, щоб допомогти їй почати роботу.

Різниця між краудфандингом та традиційним збором коштів полягає в тому, що краудфандинг повинен залучити велику групу людей, які всі пожертвують або вкладають (невелику) суму грошей, щоб врешті досягти загальної необхідної суми. Традиційні фінанси здебільшого здійснюються залученням однієї або двох організацій або підприємств, які інвестують загальну суму необхідних грошей, це також може бути позика в банку чи іншій фінансовій установі.

1.1.4 Характеристики краудфандингу

На ранніх стадіях краудфандингу він переважно використовувався для фінансування мистецьких та інших творчих починань. Пізніше він виник як засіб залучення капіталу для підтримки інновацій, підприємницьких ідей та підприємств [3]. Деякі основні елементи відрізняють краудфандинг від традиційних інвесторів, таких як банки та венчурні капіталісти, що робить його новим джерелом

фінансування. По-перше, фінансування стартапів дуже обмежене, тому краудфандинг допомагає заповнити прогалину на ранній стадії фінансування. По-друге, краудфандинг використовує впливову силу Інтернету, особливо через соціальні медіа. По-третє, це новий маркетинговий канал. По-четверте, він заохочує унікальні ідеї перевірятись громадськістю та в кінцевому підсумку реалізовуватись. Нарешті, це полегшує пряму взаємодію зі споживачами.

Основні характеристики краудфандингу, виділені вище, є одними з ключових особливостей, які відрізняють його від традиційних методів фінансування. Ми тепер детальніше розглянемо ці відмінності та повернемось до більш детального обговорення цих характеристик Це має надати більш глибокий огляд краудфандингу як інтригуючого альтернативного підходу до підприємницького фінансування порівняно з традиційними джерелами фінансування поза мережею.

Загальновідомо, що багато нових підприємців стикаються з труднощами у залученні коштів, необхідних для реалізації своїх ідей за допомогою традиційних постачальників коштів, таких як банки та венчурні капіталісти. Отже, підхід до краудфандингу представляє інший та привабливий варіант для цих підприємців. Краудфандинг також допомагає підприємцям отримати широкий вплив, що врешті допоможе їм залучити подальше фінансування. Те, що пропонує краудфандинг і чого не вистачає традиційному фінансуванню, полягає в тому, що воно створює багаторівневу віддачу, тим самим створюючи спільноту. Іншими словами, краудфандингові інвестори отримують не лише фінансову віддачу від своїх інвестицій.

Відповідно до звіту про краудфандингову галузь за 2015 рік, очікувалося, що ця галузь залучить 34,4 мільярда доларів США протягом 2015 року [6], вдвічі більше, ніж у 2014 році, і в п’ять разів збільшилася в 2013 році. значення. Цікаво, що в 2014 р. Ріст краудфандингу зріс на 320 відсотків до 3,4 млрд. Доларів в Азії, перевищивши показник для Європи (3,26 млрд. Доларів). Інвестиції окремих людей у проект краудфандингу, як правило, коливаються від 1 до 100 доларів, але спостерігаються також величезні інвестиції від одного донора. Наприклад, найбільші індивідуальні інвестиції у розмірі 100 000 доларів спостерігались на

платформі Crowdcube, що базується на акціонерному капіталі у Великобританії [6]. Найпопулярніші краудфандингові ініціативи потрапляють до категорії бізнесу та підприємництва, за ними йдуть соціальні справи, розваги та виконавське мистецтво та нерухомість [2]. Платформи для краудфандингу, такі як Kickstarter, IndieGoGo, RocketHub та GoFundMe, сприяють зростанню кількості транзакцій. Крім того, краудфандингові платформи, такі як Kiva.org та Prosper.com, сприяють новим способам мікрофінансування в області однорангового кредитування.

Краудфандинг відрізняється від інших концепцій, що базуються на натовпі, таких як краудсорсинг, розробка програмного забезпечення з відкритим кодом, рух за відкриті інновації. Всі ці концепції, засновані на громаді, побудовані на уявленні про використання сили/мудрості натовпу. На відміну від цього, краудфандинг зазвичай передбачає грошові внески громади, що підтримує фінансово. Основна мотивація засновників до участі у краудфандингу – це залучення коштів, оскільки це забезпечує зручну та організовану платформу, де розподілені фінансові внески можна вимагати та збирати від багатьох людей, отже, дозволяючи підприємцям, що зароджуються, обійти сторону традиційних фінансистів бізнесу [4]. Інші концепції, що базуються на натовпі, зазвичай включають участь та інтелектуальні вкладення, крім грошового.

1.1.5 Типологія краудфандингу

Краудфандинг можна широко класифікувати на два типи – комерційний та благодійний, хоча проект іноді підпадає під обидві групи. У комерційних ініціативах спонсори очікують певної віддачі від своїх інвестицій, тоді як спонсори не очікують жодної особистої віддачі від благодійних проектів, оскільки вони люблять бути частиною більшої справи. Визначається ще один тип (який називається проміжним), який характеризується ознаками, притаманними як комерційним, так і благодійним типами. Майбутній комерційний успіх деяких

проектів незрозумілий. Наприклад, такі проекти, як Skype, Facebook та YouTube, потрапляли б у цей тип на дуже ранніх стадіях [4].

Краудфандинг класифікується також на три групи залежно від характеру винагороди: пожертви, активні інвестиції та пасивні інвестиції [3]. На противагу цьому наявні п’ять різних категорій краудфандингу на основі прибутковості, яку спонсори очікують отримати від своїх інвестицій:

- модель пожертви;
- модель винагороди;
- модель купівлі;
- модель позики;
- модель власного капіталу.

Громадським фінансуванням для журналістики в основному нехтує краудфандингова література. Коли ми розглядаємо зобов’язання між засновником та спонсорами, краудфандинг також може бути широко розподілений на чотири типи:

- моделі на основі пожертв;
- винагороди;
- кредитування;
- акціонерного капіталу.

У додатку А представлено запропоновану [5] типологію для виявлення подібності та відмінності між різними типами краудфандингу у вигляді таблиці.

1.1.5.1 Краудфандинг на основі пожертв

Це найпростіший і найпопулярніший вид краудфандингу. За цією моделлю донори роблять пожертви на благодійні цілі. Ці пожертви зазвичай робляться на соціальні та благодійні ініціативи, при цьому спонсори не очікують віддачі від своїх інвестицій [4]. Пожертви також можна робити підприємствам, орієнтованим на прибуток, але платформи чистого пожертвування є рідкістю і, як правило, фокусуються на запитах благодійних та некомерційних організацій. Фундатори, як

правило, пожертвують на цілі, у які вони вірять, наприклад, на збір грошей, щоб дати можливість музичному гурту поїхати на гастролі. Ці донори можуть отримати якусь символічну віддачу, наприклад, подяку від засновників, але матеріальної винагороди немає. Однак модель, яка базується на пожертвах, не обмежується некомерційними організаціями, оскільки деякі люди можуть робити пожертви на ініціативу, якщо це дозволить їм згодом придбати бажаний продукт на відкритому ринку [3]. Природно, що ризик, пов’язаний з краудфандингом на основі пожертв, є дуже низьким, оскільки засновники не зобов’язані надавати прибутковість, і спонсори не очікують такої.

1.1.5.2 Краудфандинг на основі винагороди

Краудфандинг, що базується на винагороді, забезпечує спонсорам немонетарну віддачу, таку як одна з перших виготовлених продуктів. За цією моделлю підприємці запрошують потенційних клієнтів попередньо замовляти свої товари, іноді за ціною нижчою за звичайну. Засновники можуть також пропонувати подарунки та інші негрошові винагороди своїм спонсорам, але вони ніколи не платять відсотків або частки свого прибутку від бізнесу. Існує середній рівень ризику як для засновника, так і для спонсорів. У найгіршому сценарії засновник може бути не в змозі виготовити запропонований продукт з якихось причин, тому спонсори не отримають очікуваної винагороди. Взагалі, люди з особливостями раннього впровадження, як правило, стають спонсорами цього типу краудфандингу.

1.1.5.3 Краудфандинг на основі акціонерного капіталу

Краудфандинг на основі акціонерного капіталу є моделлю, за допомогою якої спонсори очікують фінансової віддачі від своїх інвестицій. Його також називають моделлю розподілу прибутку. У цій моделі підприємці закликають людей вкладати

гроші, щоб отримати частку майбутніх доходів підприємства [3]. Акціонерний краудфандинг вимагає великої кількості ділових та юридичних розглядів як з боку посередницьких краудфандингових організацій, так і з боку інвесторів, оскільки це явно потрапляє у сферу продажу цінних паперів. Crowdcube, Seedrs та IPOVillage – деякі популярні платформи для краудфандингу на основі акціонерного капіталу. Наприклад, у США розробляються закони щодо краудфандингу на основі акціонерного капіталу. Модель на основі власного капіталу, схоже, залучає більші суми капіталу, ніж модель, що базується на винагороді, незважаючи на те, що більшість платформ значною мірою займаються моделлю, яка базується на винагороді [3].

1.1.5.4 Краудфандинг на основі кредитування

Модель, що базується на кредитуванні, передбачає однорангове кредитування. Kiva та Prosper – два яскраві приклади платформ, що використовують цю модель. За цією моделлю спонсори фінансують кошти на узгоджений період з розрахунком на повернення своїх коштів, можливо, з відсотками. Деякі платформи краудфандингу, що базуються на кредитуванні, мають виключно процентний характер. Модель, що базується на кредитуванні, відрізняється від інших моделей краудфандингу тим, що обмінюються лише гроші. Процес для цієї моделі відносно простий, але спонсори ризикують втратити свої основні інвестиції у випадках, коли позичальники не можуть їх повернути. Нещодавнє дослідження показало, що краудфандинг на основі кредитування становить близько двох третин (69%) загального збору коштів [6].

1.1.6 Переваги краудфандингу

Переваги краудфандингу виходять за рамки залучення грошей.

Отримати доступ до капіталу важко для більшості нових підприємств. Багато компаній на початковій стадії передаються венчурними компаніями з різних причин, і отримання грошей у банках чи багатих членах родини навряд чи є здоровою стратегією. Краудфандинг вирівнює умови, зменшуючи опору на традиційні, а іноді і ексклюзивні методи збору коштів. Кампанії з краудфандингу також унікальні своєю здатністю залучати інтерес до нових користувачів та стимулювати зацікавленість. Оскільки для успіху потрібно залучати натовп, кампанії створюють неймовірну платформу для підвищення обізнаності про компанію, бренд, товар чи послугу. Ціль та розвиток кампанії створюють відчуття актуальності, яка мотивує інвесторів. Таким чином, краудфандинг надає стартапам можливість нарощувати аудиторію. Стартапи можуть зв’язуватися з потенційними клієнтами, які також можуть виконувати функції інвесторів та послів брендів. Успішна кампанія доводить, що існує інтерес до продукту, одночасно забезпечуючи платформу, необхідну для підтримки нових проектних ініціатив.

З точки зору інвестора, краудфандинг забезпечує простий спосіб фінансування проектів та людей, у яких ви щиро вірите та про яких турбуєтесь. Крім того, краудфандинг дозволяє інвесторам вкладати невеликі суми в кілька підприємств, тим самим диверсифікуючи свої портфелі та максимізуючи шанси на велику виплату [7]. Хоча краудфандинг – це інвестиція з високим ризиком, де інвестори повинні інвестувати лише той капітал, який їм зручно втрачати, гіпотетично, все, що інвестору насправді потрібно, – це одна інвестиція, щоб компенсувати інші втрати та отримати велику віддачу.

1.1.7 Ризики та недоліки краудфандингу

З точки зору компанії, краудфандинг може стати чудовим способом швидкого залучення коштів. Хоча для успішної краудфандингової кампанії потрібно багато роботи, просування та уваги, понад 90% стартапів у Республіці

успішно зібрали. Хоча успіх ніколи не гарантований, такі платформи, як Republic, використовують усі доступні ресурси, щоб привернути увагу до кожної кампанії.

З точки зору інвестора, як і інші форми інвестування, краудфандинг має свої ризики. З одного боку, інвестори можуть втратити всі свої інвестиції. Те, що компанія виконує цілі кампанії краудфандингу, ще не означає, що компанія обов’язково досягне успіху. Насправді більшість стартапів провалюються, і якщо цей бізнес не вдасться, інвестор, швидше за все, втратить усі вкладені гроші. Навіть якщо компанія все-таки досягне успіху, може знадобитися багато років, поки повернеться якась віддача [7].

Завжди існують ризики, пов’язані з краудфандинговими кампаніями. Інвестори повинні обов'язково перевірити будь-який проект фінансування, щоб переконатися, що їх кошти будуть використані належним чином та спрямовані на надійну компанію чи справу.

Варто зазначити, що в міру розширення ринку приватних інвестицій на платформах для стартових інвестицій є внутрішні групи, що займаються зменшенням ризиків, пов'язаних з приватним ринком. Незважаючи на те, що ці компанії не є настільки жорстко регламентованими та ретельно вивчаються, як публічні компанії, запобігання шахрайству та інші запобіжні заходи застосовуються для забезпечення доступності точної інформації. Таким чином, інвестори можуть приймати обґрунтовані рішення.

1.1.8 Краудфандинг в Україні

До створення вітчизняних платформ українці використовували міжнародні платформи для краудфандингу, такі як Kickstarter, Indiegogo та GoFundMe, і продовжують використовувати їх зараз. Kickstarter та Indiegogo - це платформи, засновані на винагородах, які в основному використовуються у стартапах, хоча вони також підтримують благодійні цілі. Безкоштовна платформа для

краудфандингу GoFundMe набула популярності завдяки збору коштів під час Євромайдану та кризи на Сході України [12].

За останні роки з'явилося кілька українських платформ для краудфандингу для вирішення різноманітних питань, серед яких гуманітарна допомога жертвам Євромайдану та конфлікту на сході України, старту бізнесу та інноваційні проекти для розвитку громадянського суспільства.

Одним з найуспішніших сайтів краудфандингу в Україні є biggggidea.com. Він був створений у 2009 році для обміну ідеями з соціально активними людьми, і тому його називають Великою ідеєю. У 2012 році компанія Big Idea запустила платформу краудфандингу Громадського фонду, яка стала першою платформою колективного фінансування в Україні. На думку засновників платформи, вона спрямована на реалізацію проектів, здатних забезпечити систематичні зміни в суспільстві та сприяти економічному зростанню національної економіки. Таким чином, за допомогою таких платформ люди можуть фінансувати соціальні зміни. На платформі представлені проекти у галузі охорони здоров'я, освіти, літератури, спорту, музики, науки, професійних подорожей та журналістики.

Ще одна популярна платформа в Україні – na-starte.com. Це платформа для фінансування стартапів, комерційних та соціальних проектів, бізнес-ідей. На цій платформі розпочато багато різних проектів, а найбільшим серед них є фінансування фільму Георгія Делієва «Одеський підкидьок» – на його реалізацію було залучено 3,7 млн грн. За весь період роботи платформа залучила 12,2 млн. грн [12].

Український краудфандинг має такі характеристики:

- переважно фінансуються проекти у галузі охорони здоров'я, культури, освіти тощо;
- відсутність ефективного державного регулювання;
- діяльність платформ для краудфандингу, розробників проектів та приватних інвесторів залишається не повністю легалізованою, оскільки немає конкретного закону про краудфандинг;
- нерозвиненість інфраструктури краудфандингу;

- існує ризик шахрайства з боку нечесних авторів проекту, які можуть «ховатися» за краудфандинговою платформою для збору коштів на фальшивий проект.

Підводячи підсумок, слід зазначити, що ринок краудфандингу в Україні є дуже перспективним. Але для подальшого активного розвитку життєво важливе законодавче поле (оскільки в Україні немає незалежного регулювання краудфандингу).

1.2 Актуальність предметної області та досліджень в цьому напрямку

Краудфандинг – явище, що розвивається, і на даний момент характеризується високою динамікою. Хоча вона сягає своїм корінням у часи, коли Бетховен або Моцарт фінансували свої концерти за державною підпискою, глобальний процес оцифрування дозволяє цій екосистемі інновацій та фінансів процвітати. Все більша кількість проектів різної природи намагається заробити гроші у громадськості в Інтернеті за допомогою краудфандингу. У той же час кількість платформ для краудфандингу масово зростає і вдвічі збільшилась між 2008-2012 роками [1, 3], забезпечуючи різні способи збору коштів для просування своїх ідей. Відповідно, нещодавно краудфандинг набув великого розголосу в новинах завдяки кільком вражаючим кампаніям, які залучили кілька мільйонів доларів капіталу від натовпу. Pebble Smartwatch – основний приклад надзвичайно успішної кампанії, яка сьогодні користується статусом найвищої фінансування Kickstarter, але проект зазнав труднощів у отриманні достатньої кількості коштів перед початком кампанії. Однак дострокове фінансування не лише змінює динаміку циклу розробки нового продукту, але й швидкість, з якою кампанії можуть залучити ранніх клієнтів до виходу на ринок. Наприклад, Ouya зібрав понад мільйон доларів з 9000 прихильників лише за вісім годин і отримав понад 2,5 мільйони доларів від понад 20 000 спонсорів протягом 24 годин, що зробило його найуспішнішим у Kickstarter

[2]. Ці історії успіху демонструють, що краудфандинг дедалі більше визнається підприємцями як здійсненний засіб виходу на ринок.

Аналіз також виявляє, що краудфандинг змінює динаміку інноваційного процесу за рахунок того, що життєвий цикл розробки нового продукту може бути скорочений, що дозволить раніше запускати продукт. Запуск кампанії з краудфандингу не лише залучає гроші на дослідження та розробку, а також служить недорогим та швидким інструментом дослідження, який надає проекту цінну інформацію про спонсорів та відгуки аудиторії [8]. Як вже згадувалося, це дозволяє швидше запустити ринковий продукт, використовуючи ресурс Натовпу. Згідно з отриманими висновками, надзвичайно важливо активно розвивати спільноту, яка підтримує цю ідею, дуже рано. Натовп є ключовим фактором для розмноження інформації, щоб швидко генерувати критичну обізнаність про проект, залучаючи раніше недоступні мережі та медіа-канали. Нарешті, сформована кількість спонсорів підтверджує ідею та ринковий попит, надаючи проекту певну довіру до взаємодії з такими партнерами, як виробники або дистриб’ютори. Визначені проблеми, такі як технологічність, проблеми з якістю та великі зусилля з комунікацією з підтримками, можуть потенційно уповільнити реалізацію ідеї.

1.3 Стартап як явище, пов’язане з краудфандингом. Особливості предметної області

Стартапи – це молоді компанії, засновані для розробки унікального продукту чи послуги, виведення їх на ринок та зробити їх непереборними та незамінними для клієнтів.

Стартапи засновані на інноваціях, усуваючи недоліки існуючих продуктів або створюючи абсолютно нові категорії товарів і послуг, тим самим порушуючи усталені способи мислення та ведення бізнесу для цілих галузей. Ось чому багато стартапів відомі у відповідних галузях як «руйнівники».

На високому рівні стартап працює як будь-яка інша компанія. Група працівників спільно працює над створенням продукту, який купуватимуть клієнти. Однак, що відрізняє стартап від інших підприємств, це те, як стартап займається цим.

Звичайні компанії дублюють те, що було зроблено раніше. Потенційний власник ресторану може відкрити франшизу існуючого ресторану. Тобто вони працюють за існуючим шаблоном того, як повинен працювати бізнес. З іншого боку, стартап має на меті створити абсолютно новий шаблон. У харчовій промисловості це може означати пропонування наборів їжі, щоб забезпечити те саме, що і ресторани – страву, яку готує шеф-кухар, але з зручністю та вибором місця для сидіння не можуть зрівнятися. У свою чергу, це забезпечує масштаб, якого окремі ресторани не можуть торкнутися: десятки мільйонів потенційних клієнтів замість тисяч.

Це також вказує на ще один ключовий фактор, який відрізняє стартапи від інших компаній: швидкість та зростання. Стартапи прагнуть дуже швидко спиратися на ідеї. Вони часто роблять це за допомогою процесу, який називається ітерацією, під час якого вони постійно вдосконалюють продукти за допомогою зворотного зв'язку та даних про використання. Часто стартап починається з основного скелета продукту, який називається мінімально життєздатним продуктом (MVP), який він буде тестувати та переглядати, поки не буде готовий вийти на ринок.

Хоча вони вдосконалюють свою продукцію, стартапи також, як правило, прагнуть швидко розширити свої клієнтські бази. Це допомагає їм встановлювати дедалі більші частки ринку, що, у свою чергу, дозволяє їм залучати більше грошей, а потім ще більше збільшувати свою продукцію та аудиторію.

Все це швидке зростання та інновації, як правило, неявно чи явно, слугують кінцевій цілі: виходу на біржу. Коли компанія відкривається для державних інвестицій, це створює можливість для ранніх інвесторів отримати гроші та отримати свої плоди – концепція, що називається стартап, яка називається «виходом на ринок».

1.3.1 Ознаки стартапів

Виокремлюють наступні ознаки стартапів, які притаманні більшості з них:

1) Стартапи зосереджені на зростанні.

В одному вирішальному аспекті стартапи не схожі на будь-які інші невеликі компанії: для стартапу недостатньо залишатися в стагнації та отримувати стабільний дохід. Натомість засновники та команди стартапів прагнуть до одного, і лише до одного: зростання.

Стартап – це за визначенням компанія, яка призначена для зростання. Однак те, що таке зростання і як воно виглядає, залежить від фази запуску в циклі фінансування стартапів. На перших етапах стартапи часто втрачають гроші, збільшуючи свою клієнтську базу, команду та/або оцінку, тоді як на пізніх стадіях після беззбиткового зростання ріст доходу є ключовим.

Типовий шлях зростання для стартапу включає 4 основні етапи:

- на початку стартапи покладаються на початковий капітал, придбаний, наприклад, за допомогою акселераторів, краудфандингу, ангельських інвесторів або навіть друзів та сім'ї. На цьому початковому етапі стартап використовує капітал для створення достатнього продукту та найму команди для отримання доходу.
- після інвестицій більшість стартапів потрапляють у так звану «Долину смерті». Вони не заробляють достатньо грошей, щоб повернути початкові капіталовкладення. Наприклад, Uber все ще перебуває на цій стадії; вона все ще зростає, але витрати компанії набагато більші за поточні доходи. Тому прогнозований рівень беззбитковості Uber має настати у 2021 році [7].
- в ідеалі, в якийсь момент стартап приніс достатньо доходу, щоб повернути початкові інвестиції, і може переключити свої погляди з генерування «достатньої кількості грошей, щоб окупити», на «більше грошей, про які

ми могли б колись подумати». На цьому етапі більшість стартапів сприяють зростанню темпів росту за допомогою стратегічних альянсів, придбань або додаткового фінансування через венчурний капітал або краудсорсинг.

- після того, як стартап пройде тверду траєкторію зростання, він розгляне можливість відкриття компанії для громадськості за допомогою пропозицій до IPO (Initial Public Offering – перша публічна пропозиція), IPO та подальших, вторинних пропозицій.

2) Стартап складається з більш ніж 1 особи, але менше, ніж з 500 осіб

Почнемо з очевидного: потрібна команда, яка називатиме бізнес стартапом. Як дуже влучно висловився Startup Commons, «Підприємець – це індивідуальність, стартап – це підприємницька команда» [9]. Більше того, однією з причин, чому стартапи зазнають невдачі, є те, що вони не прийняли на роботу належним чином. Можете розпочати бізнес самостійно, але для його розвитку знадобляться потрібні люди на борту – співробітники, фрілансери чи співзасновники.

Справжній розмір команди стартапів трохи більш неоднозначний: він перестає бути стартапом, як тільки кількість працівників перевищує 30, або 50, або 100? Аргументація обмеження в 500 осіб заснована на публікації Crunchbase за 2016 рік, згідно з якою в більшості єдинорогів з оцінками від 1 до 2,5 мільярдів доларів працює приблизно 600 людей [10].

3) Стартапи працюють з технологіями

Сам термін стартап був введений в середині кінця століття технологічної революції. Сьогодні стартапи, як і раніше, працюють із програмним забезпеченням. Але все більше і більше стартапи також зосереджуються на рішеннях, які, здавалося б, нічого або зовсім мало пов’язані з технологіями.

Ці відносно «низькотехнологічні» стартапи часто є наступними стартапами наступного покоління: наприклад, Planet Nusa, яка перетворює рибні сітки на прекрасний спортивний інвентар, або Entis, який впровадив цвіркунів на скандинавську кухню. Жоден з цих стартапів не працює безпосередньо у створенні технологічних рішень, але, безумовно, залежить від нових технологій, щоб

реалізувати свої концепції. Тому можна сказати, що стартапи працюють із технологіями, навіть якщо продукт сам по собі не є технічним.

4) Стартапи є інноваційними

Основне емпіричне правило: стартапи знаходять нові, масштабовані рішення відомих проблем. Будь то абсолютно новий продукт, послуга в новому місці, монетизація в новій формі або доставка по-новому, інновації в одній з її форм є запорукою успіху запуску. Відомі стартапи, які створюють пластикову упаковку з дерева, додають датчики до обладнання тренажерного залу, щоб відстежувати ваш прогрес, або навіть забезпечують доставку першого покоління безпілотників до окремих країн світу.

1.4 Огляд платформи для краудфандингу на прикладі Kickstarter

Kickstarter – американська корпорація з суспільною вигодою, яка підтримує глобальну платформу для краудфандингу, орієнтовану на творчість. За даними компанії, станом на липень 2020 року понад 17 мільйонів людей внесли понад 4 мільярди доларів на реалізацію проектів з 2009 року. Щоб залучити кошти, «творець» (автор) описує проект, встановлює мету для збору та звертається до «спонсорів» з проханням інвестувати в кампанію. Якщо мета досягнута, то творець отримує всі закладені кошти. В іншому випадку кампанія не буде фінансуватися, і всі гроші повертаються спонсорам. Аналіз історичних даних Kickstarter дасть важливу інформацію як для авторів, так і для авторів [11].

Kickstarter пишається тим, що допомагає «реалізувати творчі проекти». Платформа успішно забезпечила фінансування проектів, які зрештою призвели до процвітання компаній, включаючи технології Pebble та Ouya. Станом на лютий 2021 року Kickstarter залучив понад 5,6 млрд доларів на 196942 проектів [2]. На рисунку 1.1 представлена типова домашня сторінка кампанії для краудфандингових кампаній Kickstarter.

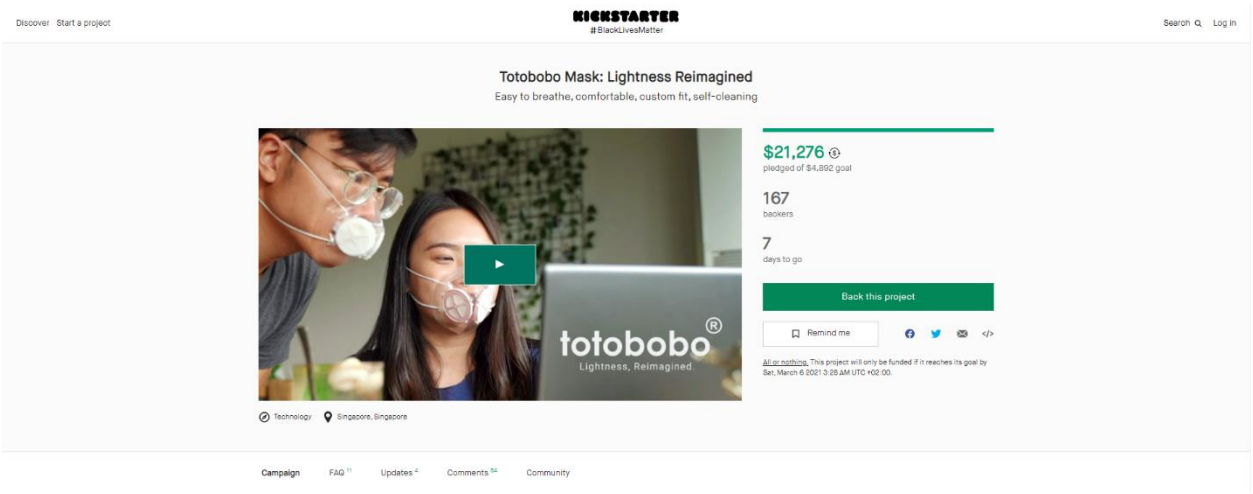


Рисунок 1.1 – Домашня сторінка кампанії на платформі Kickstarter [2]

Участь у Kickstarter вимагає, щоб приватні особи приєднувались до спільноти через безкоштовну реєстрацію. Учасники можуть створювати проекти для фінансування, фінансово сприяти проектам та коментувати проекти. Платформа відкрита для спонсорів з будь-якої точки світу та для творців з багатьох країн, включаючи США, Великобританію, Канаду, Австралію, Нову Зеландію, Нідерланди, Данію, Ірландію, Норвегію, Швецію, Іспанію, Францію, Німеччину, Австрію, Італію, Бельгію, Люксембург, Швейцарію та Мексику.

Kickstarter застосовує комісію в розмірі 5% від загальної суми залучених коштів. Їхній платіжний процесор забирає додаткову плату в розмірі 3–5%. На відміну від багатьох форумів для збору коштів або інвестицій, Kickstarter не претендує на право власності на проекти та роботу, яку вони виробляють. Веб-сторінки проектів, що запускаються на сайті, постійно архівуються та є доступними для громадськості. Після завершення фінансування проекти та завантажені носії інформації не можна редагувати чи видаляти з сайту. Власники проектів можуть взяти в заставу максимум 10000 доларів США або їх еквівалент. Проекти на Kickstarter зазвичай тривають від 1 до 60 днів, а 30 днів - рекомендований часовий проміжок [2].

У Kickstarter є кілька факторів, які відрізняють його від інших платформ для краудфандингу. По-перше, він працює за моделлю збору коштів «все або нічого». Це означає, що проект повинен повністю відповідати своїй цілі фінансування

протягом періоду збору коштів, інакше кошти повертаються спонсорам. По-друге, вкладники на Kickstarter не отримують власного капіталу в проектах, які вони фінансують, але можуть отримувати скромні винагороди (наприклад, демоверсії продуктів тощо) [4].

Сам Kickstarter пропонує керівництво для засновників проектів щодо створення та реалізації кампанії. «Підручник для творців» Kickstarter та форум пропонують надійну основу для запуску та запуску проекту.

Немає гарантії, що люди, які розміщують проекти на Kickstarter, виконуватимуть їх, використовуватимуть гроші на реалізацію своїх проектів або що виконані проекти відповідають очікуванням спонсорів. Kickstarter радить спонсорам використовувати своє судження щодо підтримки проекту. Вони також попереджають керівників проекту, що вони можуть нести відповідальність за збитки, які надали спонсори за невиконання обіцянок. Проекти можуть також провалитися навіть після успішної кампанії збору коштів, коли творці занижують загальні витрати або технічні труднощі, які потрібно подолати.

1.5 Постановка задачі дослідження

Для створення успішної кампанії зі збору фінансів на краудфандинговій платформі (зокрема Kickstarter) необхідно враховувати наступні моменти, які ставлять багато запитань:

- Історії успіху (або тематичні дослідження) окремих проектів мають слабку аргументаційну репутацію. Поради успішних засновників проектів на Kickstarter можуть бути правильними та чесними. Однак узагальнення цього (або декількох) успішних проектів неможливе, оскільки обставини відрізняються. Деякі поради також видаються суперечливими.
- Відсутні деякі важливі поради. Наприклад, які підкатегорії мають найвищий рівень успіху чи невдачі? Який день тижня збільшує

ймовірність запуску успішного проекту? У яких категоріях спонсори особливо активні?

Незважаючи на стрімке зростання кількості користувачів та проектів, рівень успіху проекту в цілому зменшується через запуск проектів без достатньої підготовки та досвіду. Для вирішення цієї проблеми в даній роботі необхідно буде:

- зібрати найбільші набори даних від Kickstarter, що складаються з усіх профілів проектів, тимчасових даних проектів та інформації користувачів у соціальних мережах;
- аналізувати характеристики успішних проектів, поведінку користувачів та розуміти динаміку платформи краудфандингу;
- запропонувати нові статистичні підходи для прогнозування того, чи буде проект успішним, та передбачування об’єму вкладених в проект грошей;
- розробити прогнозні моделі та оцінювати їх ефективність.

1.6 Аналіз існуючих робіт з поставленої задачі

Оскільки краудфандинг стає все більш популярним альтернативним фінансуванням, багато дослідників досліджували різні методи, щоб зрозуміти динаміку його розвитку. Етан Моллік дотримувався цілісної точки зору і припустив, що особисті мережі та основна якість проектів, а також географія є найважливішими факторами, що визначають успіх краудфандингу [4]. Дослідники з Масачусетського університету застосували інший підхід і зосередились на аналізі мови, яка використовується в краудфандингу [1]. Вивчаючи величезний корпус текстів, представлених у 45 000 проектів, вони виявили, що фрази, що відповідають певним принципам, таким як взаємність, дефіцит та соціальна ідентичність, збільшують шанс на успіх. Інший дослідник Вуонг обрав групу факторів, які вважаються критично важливими для розуміння підприємницьких зусиль на основі існуючої літератури про підприємництво [13]. Було знайдено докази, що підтверджують взаємозв'язок між соціокультурними рисами та результатами

діяльності, пов’язаними з підприємництвом, або рисами, що пристосовуються до географічного розташування.

Визначення причин усіх змін і тенденцій, які, як видається, трапляються випадковим чином у будь-який момент часу чи місцезнаходження у постійно мінливих умовах – це те, що відвертає потенційних творців проектів від підприємницької діяльності. Якби можна було зробити висновок про причини успішної краудфандингової кампанії в будь-якому конкретному місці та за часом, це могло б покращити розуміння таких видів підприємств та допомогти майбутнім підприємцям скласти правильні плани під час початку кампанії. Пошук причин може також допомогти нам передбачити майбутній результат будь-якої кампанії або навіть передбачити результати багатьох кампаній, таким чином, можливо, знайти тенденцію на її ранніх стадіях ще до її початку. Просторово-часова мінливість є ключем до розуміння всього цього.

Складність основної структури залежності просторово-часового компонента, схоже, помітно зростає. Можна спостерігати зміну тенденцій у різні часи та місця. Багато локацій зазнали змін у типах ідей, які з часом є найбільш успішними, а інші, здається, вперше виявляють краудфандинг в Інтернеті. Розуміння динаміки краудфандингу у різні часи та місця корисне для допомоги підприємцям або малим підприємствам у залученні основного капіталу з аудиторії для підтримки своїх проектів чи бізнесу. Інша стаття вперше проводить дослідження, вивчаючи просторово-часовий зразок успіху кампаній у краудфандингу з використанням просторово-часового статистичного підходу для додаткового розуміння динаміки краудфандингу [14].

1.7 Висновки до розділу 1

Краудфандинг – це новий метод та потенційно руйнівна інновація для фінансування різноманітних нових підприємницьких підприємств, що дозволяє окремим засновникам комерційних, культурних чи соціальних проектів вимагати

фінансування у багатьох людей в Інтернет-спільнотах, часто в обмін на майбутні товари чи капітал. Сьогодні краудфандинг стає основним способом досягнення підприємцями своїх мрій. Краудфандинг переважно розглядається з підприємницької точки зору як фінансування, включаючи стартовий капітал, один з найбільш критичних щодо ресурсів, необхідних для успіху нових підприємств. Вченим та людям, які користуються послугами краудфандингу, досі невідомо, що сприяє справді успішному прагненню отримати фінансування, і чи підсилюють чи намагання краудфандингу існуючі теорії щодо динаміки успішного підприємницького фінансування та загального розподілу та використання механізмів краудфандингу.

З огляду на стрімке зростання, динаміка краудфандингу в основному не вивчалася. Проекти, як правило, досягають успіху з невеликими відстанями або проваляться з великими. Соціальний капітал та готовність пов’язані із збільшенням шансів на успіх проекту, що свідчить про те, що сигнали якості відіграють певну роль у результатах проекту. Географія також пов'язана з характером та рівнем успіху проектів. Нарешті, переважна більшість засновників намагається доставити продукти, обіцяні спонсорам, але порівняно невелика кількість це робить своєчасно, проблема посилюється у великих або надмірно фінансованих проектах.

Для посередників із краудфандингового фінансування та політиків також є чіткі наслідки. Хоча в даний час рівень шахрайства в рамках краудфандингу є дуже низьким, незважаючи на відсутність значної сторонніх перевірок проектів, це може не мати місце в усіх формах краудфандингу. Взаємодія між низкою особливостей Kickstarter – включаючи порогове фінансування, активну участь великих спільнот, часту взаємодію між засновниками та потенційними спонсорами та здатність засновників передавати якісні сигнали через багаті описи та біографічну інформацію – ймовірно, відіграє ключову роль зменшення шахрайства. Видалення деяких з цих елементів може зменшити здатність громад визначати якісні проекти та збільшити шанс шахрайства. Крім того, посередники та політики повинні розглянути способи, як допомогти засновникам створити реалістичні плани та цілі,

щоб гарантувати, що краудфандинг підтримує низький рівень шахрайства та високий темп зростання.

РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ МЕТОДІВ ДОСЛІДЖЕННЯ

2.1 Дослідження існуючих методів вирішення задачі

Бурхлива тенденція поширення то розвитку краудфандингу привернула велику увагу з боку наукових кіл. Традиційний підхід полягав у створенні класифікатора машинного навчання, такого як SVM, на основі мета-функцій із профілю кампанії. Грінберг та ін. [15] показав покращення, використовуючи різні алгоритми дерева рішень та SVM, навчений таким функціям, як наявність відео, кількість пропозицій у профілі, цілі проекту, тривалість проекту та інші можливі додаткові фактори, такі як демографічні атрибути творців. Деякі вдосконалені підходи використовують текстовий опис, тоді як певні моделі додатково використовують динамічну інформацію шляхом моніторингу соціальних мереж або краудфандингової кампанії. Мітра та Гілберт [16] проаналізували лінгвістичні особливості разом із 59 іншими загальними рисами для прогнозування успіху проекту та запропонував аналітичну тематичну основу для прогнозування успіху збору коштів шляхом вилучення прихованої семантики з текстового опису із поєднанням загальних числових ознак. Пізніші дослідження вивчали фактори динамічного часового ряду, відстежуючи соціальні медіа та відстежуючи динамічні особливості, такі як статус застави та грошей під час кампанії та виявили, що ступінь активності соціальної мережі творця позитивно пов’язана з успіхом проекту, оскільки творці можуть транслювати свої краудфандингові проекти для більшої аудиторії. Проблему прогнозування успіху також формулювали з точки зору цензурованої регресії та досягли кращих показників, використовуючи тимчасові особливості від застави та динаміки соціальних мереж [13].

З наведених вище описів ми можемо помітити, що більшість попередніх робіт були зосереджені на текстовому профілі та інформації про запуск, що робить як творців проектів, так і платформи не в змозі передбачити результат вчасно. Тому, щоб зробити можливим прогнозування перед публікацією, підхід даного дослідження фокусується на аналізі текстової та візуальної інформації, зібраної на

етапі перед запуском, який ще не був повністю вивчений у попередніх дослідженнях (або не є широко відомим).

Так як невідомо, які з методів виявляться найкращими для прогнозування успіху проекту, тому коротко розглянемо усі, які будемо використовувати й порівнювати. Окремо необхідно зауважити, що робота не буде стосуватися лише класифікації, так як в рамках поставленого завдання також є вимога до прогнозу суми, яку вдасться зібрати в рамках краудфандингу, тому не слід забувати й про регресійні методи.

2.2 Опис методів вирішення задачі

Перш ніж описувати методи, пов’язані з підходом, використаним у роботі, спочатку необхідно виділити недоліки стандартної класифікації та регресійних моделей для вирішення проблеми прогнозування успіху, згаданої раніше.

Моделювання даних краудфандингу створює нову проблему з точки зору включення проектів, де ми знаємо дату успіху, та проектів, де ми маємо лише часткову інформацію про те, що вони не досягли успіху до певної дати цілі проекту. Такі проекти називаються цензурними. У традиційному режимі регресії/класифікації ці проекти просто трактуються як відсутні або пропущені дані, і вони не вносять жодної інформації, якщо тільки не зроблено досить жорстких припущень, за якими слідують важкі обчислення, наприклад, багаторазове обчислення. Однак, використовуючи цензуровані моделі регресії, функція вірогідності будується з використанням часткової інформації.

У проблемах регресії змінна результату є безперервною і є будь-яким дійсним числом, тоді як час за цією природою буде строго невід’ємним. Стандартні методи машинного навчання, такі як лінійна та логістична регресія, не можуть бути використані для прогнозування часу. Це зумовлено тим, що не можна застосовувати лінійні та логістичні алгоритми регресії для прогнозування негативних результатів. Цензуровані моделі регресії за своєю суттю можуть

обробляти це негативне обмеження і будувати моделі, що передбачають лише невід'ємні змінні результату.

З наведеного вище ясно, що цензуровані моделі регресії мають деякі критичні переваги порівняно зі стандартною регресією/класифікацією. Хоча це не слід розглядати як конкурента стандартному регресійному аналізу, скоріше, такі цензуровані моделі застосовні до більш спеціалізованих та складних сценаріїв моделювання, а саме до моделювання даних часу до події. У цьому дослідженні ми вважаємо за подію, що представляє інтерес, успіх проекту, а на меті передбачити, коли проект може потенційно стати успішним порівняно з іншими наявними. Отже, у таких проблемах можна буде отримати повну інформацію про події лише для успішних проектів. Невдалі проекти не матимуть події і спостерігатимуться лише до дати цілі проекту. Критичною відмінністю між цим формулюванням та стандартними підходами до регресії є той факт, що ця робота включає як успішні, так і невдалі проекти одночасно, на відміну від використання лише тих успішних проектів, як це зроблено у формулюваннях на основі регресії.

2.2.1 Модель Тобіта (цензурована регресія)

Ідея Джеймса Тобіна полягала в тому, щоб змінити функцію правдоподібності таким чином, щоб вона відображала неоднакову ймовірність вибірки для кожного спостереження залежно від того, чи досягла прихована змінна значення вище або нижче визначеного порогу. Для вибірки, яка, як і в початковому випадку Тобіна, була цензурована знизу при нулі, ймовірність вибірки для кожного необмеженого спостереження – це просто значення відповідної функції щільності. Для будь-якого граничного спостереження це кумулятивний розподіл, тобто інтеграл нижче нуля відповідної функції щільності. Таким чином, функція правдоподібності до тобіту являє собою суміш щільностей та кумулятивних функцій розподілу.

Нижче наведені функції правдоподібності та логарифм правдоподібності для тобіту першого типу (в літературі [17] описуються 5 типів тобітів, але оберемо найпростіший). Це тобіт, який цензурується знизу на y_L, коли прихована змінна $y_j^* \leq y_L$. Виписуючи функцію правдоподібності, спочатку необхідно визначити функцію індикатора I:

$$I(y) = \begin{cases} 0 \text{ якщо } y \leq y_L, \\ 1 \text{ якщо } y > y_L. \end{cases} \tag{2.1}$$

Далі нехай Φ – стандартна нормальна кумулятивна функція розподілу, а φ – стандартна нормальна функція щільності ймовірності. Використовуючи формулу 2.1 для набору даних з N спостереженнями функція правдоподібності для описаного вище тобіту дорівнює

$$\mathcal{L}(\beta, \sigma) = \prod_{j=1}^{N} \left(\frac{1}{\sigma} \varphi \left(\frac{y_j - X_j \beta}{\sigma} \right) \right)^{I(y_j)} \left(1 - \Phi \left(\frac{X_j \beta - y_L}{\sigma} \right) \right)^{1 - I(y_j)} \tag{2.2}$$

Тоді логарифм правдоподібності з урахуванням виразу 2.2 буде мати наступний вигляд:

$$\begin{aligned} \log \mathcal{L}(\beta, \sigma) &= \sum_{j=1}^{N} I(y_j) \log \left(\frac{1}{\sigma} \varphi \left(\frac{y_j - X_j \beta}{\sigma} \right) \right) + \\ &+ \left(1 - I(y_j) \right) \log \left(1 - \Phi \left(\frac{X_j \beta - y_L}{\sigma} \right) \right) = \\ &= \sum_{y_j > y_L} \log \left(\frac{1}{\sigma} \varphi \left(\frac{y_j - X_j \beta}{\sigma} \right) \right) + \sum_{y_j = y_L} \log \left(\Phi \left(\frac{y_L - X_j \beta}{\sigma} \right) \right) \end{aligned} \tag{2.3}$$

Коефіцієнт β у формулі 2.3 не слід інтерпретувати як вплив x_i на y_i, що справедливо для лінійної моделі регресії. Натомість це слід інтерпретувати як

комбінацію зміни y_i тих, що перевищують межу, зважену ймовірністю перевищення межі; та зміни ймовірності перевищення межі, зваженої на очікуване значення y_i, якщо вище.

Наприклад, дана модель застосовувались для оцінки факторів, що впливають на отримання грантів, включаючи фінансові трансферти, що розподіляються урядам субнаціональних органів, які можуть подати заявку на отримання цих грантів. У цих випадках одержувачі грантів не можуть отримувати негативні суми, і, таким чином, дані піддаються цензурі вліво. Однак дані можуть бути цензуровані вліво в точці, що перевищує нуль, з ризиком неправильної специфікації. Моделі тобіту також застосовувались при аналізі попиту для забезпечення спостережень з нульовими витратами на деякі товари. У відповідному застосуванні моделей система нелінійних моделей регресій тобіту була використана для спільної оцінки системи попиту на бренд з гомосцедастичними, гетероскедастичними та узагальненими гетероскедастичними варіантами [17].

Що стосується дослідження з краудфандингу, моделі цензурованої регресії містять два компоненти:

- час до події, тобто час, необхідний для того, щоб відбулася конкретна подія, що представляє інтерес (успіх проекту);
- цензура, тобто часткова інформація про проекти, де успіху не відбулося.

Форма цензури у даній задачі – це правильна цензура, коли, як відомо, час існування кампанії довший за певне значення, але його точне значення невідоме. Крім того, є також особливості, які потрібно пов’язати з часом, щоб пояснити явище часової події (наприклад, час успіху проекту). Такі моделі перевіряють різницю в часі успіху для двох або більше проектів, що представляють інтерес, одночасно дозволяючи скоригувати особливості проекту. Зовсім недавно лише деякі проблеми в обчислювальній рекламі були ефективно вирішені за допомогою таких моделей [15]. Для моделювання цензурованих даних деякі з цих підходів використовують апроксимацію функції ймовірності, яка називається частковою логарифмічною ймовірністю.

2.2.2 Метод опорних векторів (SVM – Support Vector Machine)

SVM будує модель навчання, яка призначає нові приклади тій чи іншій групі. За допомогою цих функцій SVM називають неімовірнісним, двійковим лінійним класифікатором. У налаштуваннях ймовірнісної класифікації SVM можуть використовувати такі методи, як масштабування Платта.

Як і інші навчальні машини з учителем, SVM вимагає навчання маркованих даних. Групи матеріалів позначені класифікацією. Навчальні матеріали для SVM класифікуються окремо в різних точках простору та організовуються у чітко відокремлені групи. Після обробки численних прикладів навчання SVM можуть виконувати навчання без учителя. Алгоритми намагатимуться досягти найкращого поділу даних з максимальною межами навколо гіперплощини і навіть між обома сторонами.

Системи використовуються в класифікації тексту, гіпертексту та зображень. SVM можуть працювати з рукописними символами, і алгоритми використовуються в лабораторіях біології для виконання таких завдань, як сортування білків. Системи навчання, що контролюються та не контролюються, серед іншого використовуються в чат-ботах, автономних автомобілях, програмах розпізнавання обличчя, експертних системах та роботах.

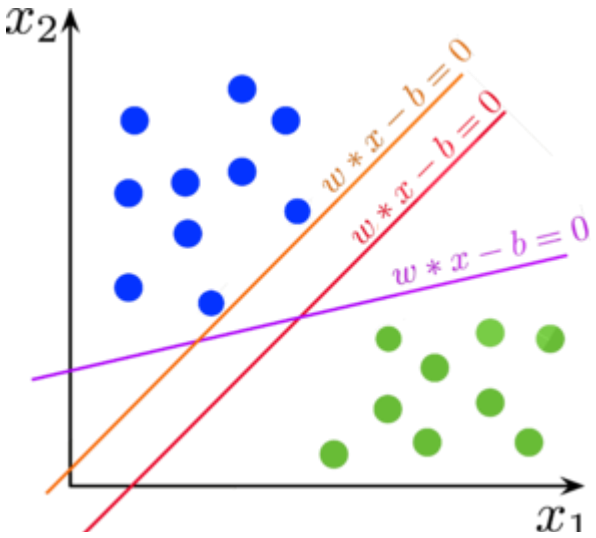


Рисунок 2.1 – Серед великої кількості розділяючих прямих SVM знаходить рішення з найбільшим відступом [8]

Відповідно до алгоритму SVM, ми знаходимо точки, найближчі до лінії з обох класів. Ці точки називаються опорними векторами. Тепер ми обчислюємо відстань між лінією та опорними векторами. Ця відстань називається відступом. Наша мета – максимізувати її. Гіперплощина, для якої відступ є максимальним, є оптимальною.

Переваги SVM перед методом стохастичного градієнта і нейронними мережами:

- завдання опуклого квадратичного програмування добре вивчена і має єдине рішення;
- метод опорних векторів еквівалентний двошарової нейронної мережі, де число нейронів на прихованому шарі визначається автоматично як число опорних векторів;
- принцип оптимальної розділяючої гіперплощини призводить до максимізації ширини розділяючої смуги, а отже, до більш впевненої класифікації.

Недоліки класичного SVM:

- нестійкість до шуму: викиди в вихідних даних стають опорними об'єктами-порушниками і безпосередньо впливають на побудову розділяючої гіперплощини;
- не описані загальні методи побудови ядер і спрямляючих просторів, найбільш придатних для конкретного завдання.
- немає відбору ознак.
- необхідно підбирати константу C за допомогою крос-валідації.

2.2.3 Дерева рішень та випадковий ліс

Дерево рішень – це карта можливих результатів низки пов'язаних між собою варіантів вибору. Це дозволяє окремій особі чи організації зважувати можливі дії

між собою на основі їхніх витрат, ймовірностей та вигод. Їх можна використовувати або для стимулювання неформальних дискусій, або для складання алгоритму, який передбачає найкращий вибір математично.

Дерево рішень зазвичай починається з одного вузла, який розгалужується на можливі результати. Кожен із цих результатів веде до додаткових вузлів, які розгалужуються на інші можливості. Це надає йому деревоподібну форму.

Існує три різні типи вузлів: випадкові вузли, вузли прийняття рішень та кінцеві вузли. Вузол шансів, представлений колом, показує ймовірності певних результатів. Вузол рішення, представлений квадратом, показує рішення, яке потрібно прийняти, а кінцевий вузол – кінцевий результат шляху прийняття рішення.

Переваги:

- простий для розуміння, інтерпретації, візуалізації;
- дерева рішень неявно виконують скринінг змінних або вибір функцій;
- може обробляти як числові, так і категоріальні дані. також може обробляти проблеми з кількома виходами;
- дерева рішень вимагають від користувачів відносно невеликих зусиль для підготовки даних;
- нелінійні взаємозв'язки між параметрами не впливають на продуктивність дерева.

Недоліки:

- листові вузли дерева прийняття рішень можуть створювати надскладні дерева, які погано узагальнюють дані. Це називається перенавчанням;
- дерева рішень можуть бути нестабільними, оскільки невеликі варіації даних можуть призвести до генерування зовсім іншого дерева. це називається дисперсією;
- жадібні алгоритми не можуть гарантувати повернення глобально оптимального дерева рішень. це можна пом'якшити шляхом навчання кількох дерев, де ознаки та вибірки випадково відбираються із заміною.

- листові вузли дерева прийняття рішень створюють упереджені дерева, якщо деякі класи домінують. тому рекомендується збалансувати набір даних перед використанням дерева рішень.

Випадковий ліс, як випливає з назви, складається з великої кількості окремих дерев рішень, які діють як ансамбль. Кожне окреме дерево в випадковому лісі випливає передбачення класу, і клас з найбільшою кількістю голосів стає прогнозом нашої моделі (у найпростішому випадку).

Фундаментальним поняттям, що стоїть за випадковим лісом, є проста, але потужна концепція – мудрість натовпу. Велика кількість відносно некорельованих моделей (дерев), що діють як комітет, перевершить будь-яку з окремих складових моделей.

Низька кореляція між моделями є ключовим фактором. Подібно до того, як інвестиції з низькою кореляцією (наприклад, акції та облігації) об'єднуються, щоб сформувати портфель, який перевищує суму його частин, некорельовані моделі можуть створювати ансамблеві прогнози, які є більш точними, ніж будь-які окремі прогнози. Причиною цього чудового ефекту є те, що дерева захищають одне одного від своїх індивідуальних помилок (якщо вони постійно не помиляються в одному напрямку). Хоча деякі дерева можуть помилятися, багато інших дерева матимуть рацію, тому, будучи групою, дерева можуть рухатись у правильному напрямку.

2.2.4 Наївний Байєс Бернуллі та Гаусса

Наївні класифікатори Байєса базуються на теоремі Байєса. Одним з припущень є тверді припущення про незалежність між ознаками. Ці класифікатори припускають, що значення певної ознаки не залежить від будь-якої іншої ознаки. У навчальній ситуації під час навчання з учителем класифікатори наївного Байєса дуже ефективно навчаються. Наївні класифікатори потребують невеликих навчальних даних, щоб оцінити параметри, необхідні для класифікації. Наївні

класифікатори Байєса мають просту розробку та реалізацію, і вони можуть застосовуватися до багатьох реальних життєвих ситуацій.

Наївні класифікатори Байєса є дуже масштабованими, вимагаючи ряду параметрів, лінійних за кількістю змінних (ознак/предикторів) у навчальній задачі. Навчання з максимальною правдоподібністю можна здійснити, оцінивши вираз із закритою формою, який займає лінійний час, а не шляхом дорогого ітераційного наближення, як це використовується для багатьох інших типів класифікаторів.

Це проста техніка побудови класифікаторів: моделі, які присвоюють мітки класів примірникам проблем, представленим у вигляді векторів значень ознак, де мітки класів витягуються з деякого кінцевого набору. Існує не єдиний алгоритм навчання таких класифікаторів, а сімейство алгоритмів, заснованих на загальному принципі: усі наївні класифікатори Байєса припускають, що значення певної ознаки не залежить від значення будь-якої іншої ознаки, враховуючи змінну класу. Наприклад, фруктом можна вважати яблуко, якщо воно червоне, кругле і має діаметр близько 10 см. Наївний класифікатор Байєса вважає, що кожна з цих ознак вносить незалежний внесок у ймовірність того, що цей фрукт є яблуком, незалежно від будь-яких можливих співвідношень між кольором, округлістю та діаметром.

Для деяких типів ймовірнісних моделей наївні класифікатори Байєса можуть бути дуже ефективно навчені в контрольованому навчальному середовищі. У багатьох практичних додатках для оцінки параметрів для наївних моделей Байєса використовується метод максимальної ймовірності; іншими словами, можна працювати з наївною моделлю Байєса, не приймаючи байєсівської ймовірності або використовуючи будь-які байєсівські методи.

Незважаючи на наївний дизайн та, мабуть, надто спрощені припущення, наївні класифікатори Байєса працювали досить добре у багатьох складних реальних ситуаціях. У 2004 р. Аналіз проблеми класифікації Байєса показав, що існують вагомі теоретичні причини очевидно неправдоподібної ефективності наївних класифікаторів Байєса [18]. Проте всебічне порівняння з іншими алгоритмами класифікації в 2006 році показало, що класифікація Байєса перевершує інші підходи, такі як підняті дерева або випадкові ліси [18].

Перевагою наївного Байєса є те, що для оцінки параметрів, необхідних для класифікації, потрібна лише невелика кількість навчальних даних.

2.2.4.1 Гауссівський наївний Байєс

При роботі з неперервними даними типовим припущенням є те, що неперервні значення, пов'язані з кожним класом, розподіляються відповідно до нормального розподілу. Наприклад, припустимо, дані навчання містять неперервний атрибут, x. Дані спочатку сегментуються за класом, а потім обчислюється середнє значення та дисперсія x у кожному класі. Нехай є середнім значенням у x, пов'язаному з класом C_k, і нехай σ_k^2 – виправлена дисперсія значень x, пов’язана з класом C_k. Припустимо, хтось зібрав якесь значення спостереження v. Потім щільність ймовірності v для класу $C_k, p(x = v|C_k)$, можна обчислити, підключивши v до рівняння 2.4 для нормального розподілу, параметризованого σ_k^2.

$$p(x = v|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}}\, e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}} \tag{2.4}$$

Інший поширений прийом для обробки безперервних значень – використання дискретизації значень ознак, для отримання нового набору розподілених функцій Бернуллі. Це необхідно для застосування наївних Байєсів. Іноді розподіл умовно-класових граничних щільностей далеко не нормальний. У цих випадках оцінку щільності ядер можна використовувати для більш реалістичної оцінки граничних щільностей кожного класу.

2.2.4.2 Наївний Байєс Бернуллі

У багатовимірній моделі подій Бернуллі ознаками є незалежні булеві значення (двійкові змінні), що описують вхідні дані. Як і багаточленна модель, ця

модель популярна для завдань класифікації документів, де використовуються бінарні ознаки входження термінів, а не частотні терміни. Якщо x_i є логічним значенням, що виражає появу або відсутність і-го терміна зі словникового запасу, то ймовірність отримання документом класу C_k задано за допомогою формули 2.5:

$$p(x|C_k) = \prod_{i=1}^{n} p_{k_i}^{x_i}(1 - p_{k_i})^{1-x_i} \quad (2.5)$$

де p_{k_i} – ймовірність класу C_k, що породжує термін x_i.

Ця модель події особливо популярна для класифікації коротких текстів. Це має перевагу явного моделювання відсутності термінів. Необхідно також звернути увагу, що наївний класифікатор Байєса з моделлю події Бернуллі не є таким самим, як багаточленний класифікатор із підрахунком частоти, зрізаним до одиниці.

2.2.5 Метод *k* найближчих сусідів (k-NN, KNN – k nearest neighbors)

k-NN – це тип класифікації, де функція апроксимується лише локально, а всі обчислення відкладаються до оцінки функції. Оскільки цей алгоритм покладається на відстань для класифікації, якщо ознаки представляють різні фізичні одиниці або мають дуже різні масштаби, нормалізація навчальних даних може значно покращити його точність.

Як для класифікації, так і для регресії корисним прийомом може бути присвоєння ваги внеску сусідів, так що ближчі сусіди вносять більше в середнє, ніж більш віддалені. Наприклад, загальна схема зважування полягає у наданні кожному сусідові ваги $\frac{1}{d}$, де d – відстань до сусіда.

Сусіди беруться з набору об'єктів, для яких відомий клас (для класифікації k-NN) або значення властивості об'єкта (для регресії k-NN). Це можна сприймати як навчальний набір для алгоритму, хоча явного кроку навчання не потрібно. Особливістю алгоритму k-NN є те, що він чутливий до локальної структури даних.

Навчальні приклади – це вектори в багатовимірному просторі ознак, кожен з яких має позначку класу. Навчальний етап алгоритму складається лише із збереження векторів ознак та міток класів навчальних зразків. На етапі класифікації k – це визначена користувачем константа, а немічений вектор (запит чи точка тесту) класифікується шляхом присвоєння мітки, яка найчастіше зустрічається серед k-зразків навчання, найближчих до цієї точки запиту.

Часто використовуваною метрикою відстані для неперервних змінних є евклідова відстань. Для дискретних змінних, таких як класифікація тексту, може бути використана інша метрика, така як метрика перекриття (або відстань Хеммінга). Часто точність класифікації k-NN можна значно покращити, якщо метрику відстані вивчити за допомогою спеціалізованих алгоритмів, таких як аналіз компонентів з найближчим сусідством із великим полем чи сусідством.

Недолік базової класифікації "більшості голосів" виникає, коли розподіл класів є нерівним. Тобто, приклади більш частого класу, як правило, домінують у передбаченні нового прикладу, оскільки вони, як правило, поширені серед k найближчих сусідів через їх велику кількість. Одним із способів подолання цієї проблеми є зважування класифікації, враховуючи відстань від точки тесту до кожного з найближчих сусідів. Клас (або значення в задачах регресії) кожної з k найближчих точок множиться на вагу, пропорційну зворотній відстані від цієї точки до контрольної точки. Іншим способом подолання перекосу є абстракція у поданні даних.

2.2.6 Екстремальний градієнтний бустинг (XGBoost – eXtreme Gradient Boosting)

XGBoost – це алгоритм машинного навчання на основі дерева рішень, який використовує фреймворк градієнтного бустингу. При проблемах прогнозування, пов’язаних з неструктурованими даними (зображеннями, текстом тощо), штучні нейронні мережі, як правило, перевершують всі інші алгоритми або структури.

Однак, коли мова йде про структуровані/табличні дані від малого до середнього, алгоритми, засновані на дереві рішень, зараз вважаються найкращими у своєму класі.

Реалізація алгоритму була розроблена для ефективності обчислення часу та ресурсів пам'яті. Метою дизайну було якнайкраще використовувати наявні ресурси для навчання моделі. Деякі ключові функції реалізації алгоритму включають:

- реалізація з розрідженою інформацією з автоматичною обробкою відсутніх значень даних;
- структура блоку для підтримки розпаралелювання побудови дерева;
- продовження навчання, щоб ви могли додатково вдосконалити вже встановлену модель на нових даних.

Бустинг – це ансамблева техніка, де додаються нові моделі, щоб виправити помилки, допущені існуючими моделями. Моделі додаються послідовно, доки подальших удосконалень зробити не вдасться. Популярним прикладом є алгоритм AdaBoost, який зважує точки даних, які важко передбачити.

Градієнтний бустинг – це підхід, коли створюються нові моделі, які передбачають залишки або помилки попередніх моделей, а потім додають разом, щоб зробити остаточне прогнозування. Це називається посиленням градієнта, оскільки воно використовує алгоритм градієнтного спуску, щоб мінімізувати втрати при додаванні нових моделей.

Цей підхід підтримує як проблеми регресії, так і класифікації прогнозного моделювання.

XGBoost – це більш швидкий алгоритм порівняно з іншими алгоритмами завдяки його паралельним та розподіленим обчисленням. XGBoost розроблений як з глибокими міркуваннями з точки зору оптимізації систем, так і принципів машинного навчання. Метою цієї бібліотеки є розширення граничних обчислювальних значень машин, щоб забезпечити масштабовану, портативну та точну бібліотеку.

2.3 Критерії оцінки якості рішення

Ідея побудови моделей машинного навчання працює на принципі конструктивного зворотного зв’язку. Створюється модель, надходить зворотний зв'язок з метриками, вноситься вдосконалення і продовжується, поки не буде досягнута бажана точність. Показники оцінки пояснюють ефективність моделі. Важливим аспектом метрик оцінювання є їх здатність розрізняти результати моделі.

Просто побудова прогнозної моделі це не професіонально. Йдеться про створення та вибір моделі, яка забезпечує високу точність на основі даних вибірки. Отже, дуже важливо перевірити точність моделі перед обчисленням прогнозованих значень.

У якості основного показника якості моделі було обрано AUC (площа під кривою) ROC, звертаючи увагу на оцінку точності для класу успішного проекту у своєму аналізі, щоб переконатися, що ми будемо не передбачати занадто багато успіхів, які могли б виявитись невдачами (тим самим мінімізуючи помилковий позитивний показник).

Крива ROC – це графік між чутливістю та специфічністю). Специфічність також відомий як хибнопозитивний показник (false positive rate), а чутливість – як істинний позитивний показник (true positive rate).

AUC ROC враховує прогнозовані ймовірності для визначення ефективності нашої моделі. Однак є проблема з AUC ROC, вона враховує лише порядок ймовірностей, а отже, не враховує можливості моделі передбачити вищу ймовірність для зразків, які, швидше за все, будуть позитивними.

Також необхідно звертати увагу на більш прості метрики якості, такі як точність, повнота, показник F1 та інші, так як вони можуть швидко вказати на недоліки моделі. AUC будемо використовувати лише для кінцевої валідації.

2.4 Висновки до розділу 2

Для вирішення зростаючого різноманіття та складності проблем управлінського прогнозування протягом останніх років було розроблено багато методів прогнозування. Кожен має своє особливе використання, і слід подбати про те, щоб вибрати правильний для конкретного застосування.

Вибір методу залежить від багатьох факторів – контексту прогнозу, релевантності та доступності історичних даних, ступеня бажаної точності, періоду часу, який слід прогнозувати, вартості/вигоди (або вартості) прогнозу.

В даному розділі було описано методи, які лежать в основі роботи, а також виявлено їх сильні та слабкі сторони. Їх порівняння буде наведено в наступних розділах, в чому стануть в нагоді обрані для цієї мети метрики оцінки якості моделей.

РОЗДІЛ 3 ОГЛЯД ДАНИХ ТА ЇХ ПОПЕРЕДНЯ ОБРОБКА

3.1 Вихідні дані дослідження

Дані в оригінальній формі, складаються з 104 наборів даних у форматі csv, що охоплюють час з 2009 по 2017 рік. При об'єднанні набори даних містять 36 мільйонів спостережень, багато з яких є дублікатами через скрапінг, що відбувається щомісяця, та проекти, що йдуть від початку до закінчення терміну протягом місяців. Ідентифікатори кампанії спостережень використовувались для видалення дублікатів. Багато спостережень також бракувало. Тому змінні із занадто неповними спостереженнями були вилучені із загального набору даних для цього дослідження. В результаті було створено набір даних із 99 036 спостереженнями на загальну суму 1064 392 179 доларів США під заставу. Дані містять спостереження з усього світу, більшість із них походять із США та Північної Америки. Цей набір даних повинен добре відображати кампанії Kickstarter і, можливо, будь-яку платформу для краудфандингу, яка може бути використана в Інтернеті, щоб допомогти підприємцям отримати стартовий капітал.

Ці дані були отримані за допомогою скрапінгу сайту Kickstarter. Веб-скрапінг, збір веб-сайтів або витяг веб-даних – це метод отримання даних, що використовується для вилучення даних з веб-сайтів. Програмне забезпечення для вискоблювання веб-сторінок може безпосередньо отримувати доступ до Всесвітньої павутини за допомогою протоколу передачі гіпертексту або веб-браузера. Незважаючи на те, що скрапінг може виконуватися користувачем програмного забезпечення вручну, термін, як правило, відноситься до автоматизованих процесів, реалізованих за допомогою бота або веб-сканера. Це форма копіювання, при якій конкретні дані збираються та копіюються з Інтернету, як правило, в центральну локальну базу даних або електронну таблицю для подальшого пошуку або аналізу.

3.1.1 Короткий опис головних колонок даних

- Year: рік запуску;
- Goal_amount_USD: сума цілі, встановлена автором (сума грошей, яку автор хотів би отримати);
- Duration: час від дати початку до дати закінчення кампанії;
- ContentImageCount: кількість зображень, використаних у вмісті проекту;
- ContentVideoCount: кількість відеозаписів, використаних у змісті проекту;
- PackageCount: кількість пропонованих пакетів у проекті;
- DescriptionWordCount: кількість слів в описі проекту;
- ContentWordCount: кількість слів у змісті проекту;
- RiskWordCount: кількість слів у частині ризику проекту;
- MinPackageAmount: мінімальна сума пропонованих пакетів у доларах США;
- MaxPackageAmount: максимальна сума пропонованих пакетів у доларах США;
- MeanPackageAmount: середня сума пропонованих пакетів у доларах США;
- Pledged_amount_USD: заставлена сума, яку отримав проект після завершення кампанії;
- Category: категорія проекту. наприклад мистецтво, їжа, технології;
- ChildCategory: наприклад, проект, що належить до категорії Мистецтво, може належати до однієї з наступних категорій: Ілюстрація, Публічне мистецтво, Живопис, тощо;
- Goal_currency: валюта встановлення цілі.

3.1.2 Встановлення важливостей ознак

У багатьох випадках не менш важливо мати не тільки точну, але й інтерпретовану модель. Часто, окрім того, що ми хочемо знати, яким є

прогнозування ціни на житло для нашої моделі, ми також дивуємось, чому саме цей високий/низький рівень і які особливості є найбільш важливими для визначення прогнозу. Іншим прикладом може бути прогнозування відтоку клієнтів – дуже приємно мати модель, яка успішно передбачає, які клієнти схильні до відтоку, але виявлення важливих змінних може допомогти нам у ранньому виявленні та, можливо, навіть вдосконаленні продукту/послуги. Знання важливості функцій, зазначеної моделями машинного навчання, може принести вам багато користі.

Отримавши краще розуміння логіки моделі, можна не тільки перевірити її правильність, але й попрацювати над вдосконаленням моделі, зосередившись лише на важливих змінних, які вище можна використовувати для вибору змінних. Можна видалити деяку кількість змінних, які не є настільки важливими і, маючи подібні чи кращі показники за значно коротший час навчання, у деяких бізнес-випадках має сенс пожертвувати деякою точністю заради зрозумілості. Наприклад, коли банк відхиляє заявку на позику, він також повинен мати обґрунтування рішення, яке також може бути представлене клієнту.

Необхідно зазначити одне: чим точнішою є модель, тим більше можна довіряти заходам важливості ознак та іншим тлумаченням. Відповідні дані можна побачити на рисунку 3.1. Як можна побачити, найважливішою ознакою є рік, коли відбувалася кампанія. Це достатньо логічно, адже тренди змінюються не те що за рік, а за лічені місяці. Далі за спаданням важливості можна побачити обсяги грошей, які вдалося зібрати та які вимагалося. Ці ознаки також є вкрай важливими для аналізу успішності проекту.

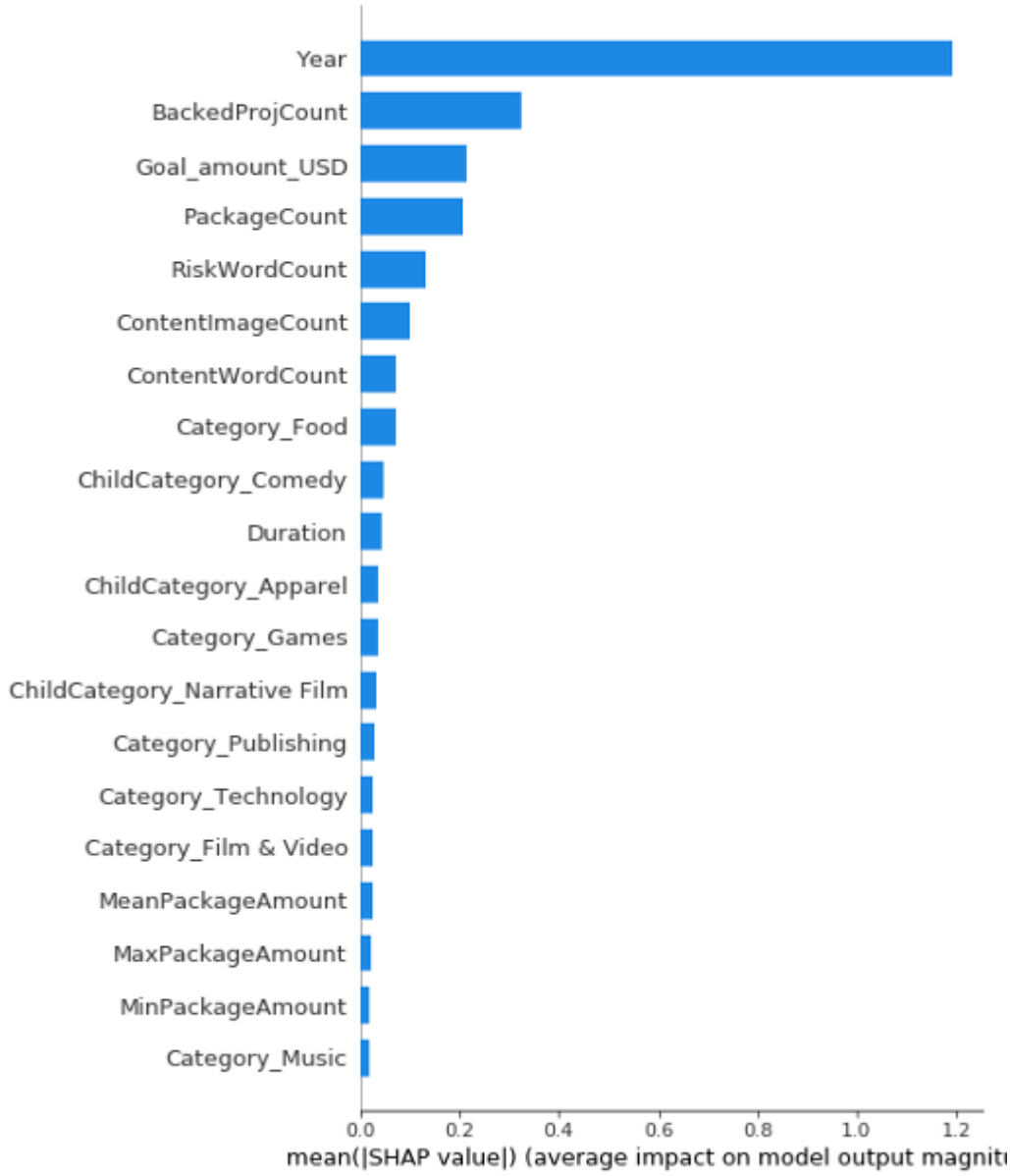


Рисунок 3.1 – Важливість ознак з набору даних

3.2 Методи попередньої обробки даних

Попередньо обробляючи дані, можна досягти наступних цілей:

- база даних стане точнішою. Внаслідок усунення неправильних або відсутніх значень, які присутні внаслідок людського фактора або помилок;
- якщо у даних або дублікатах є невідповідності, це впливає на точність результатів, тому їх видалення стане хорошим кроком;

- базу даних стане повнішою. За потреби можна заповнити відсутні атрибути;
- дані будуть більш «гладкими» внаслідок нормалізації. Таким чином можна полегшити використання та інтерпретацію.

3.2.1 Оцінка якості даних

Перш за все, потрібно добре поглянути на свою базу даних та провести оцінку якості даних. Випадковий збір даних (такий як скрапінг) часто має недостатню повноту та коректність.

Невідповідність у типах даних. Досить часто можна поєднувати набори даних, які використовують різні формати даних. Отже, невідповідності цілі та числа з плаваючою точкою чи UTF8 та ASCII.

Різні розміри масивів даних. Коли відбувається агрегація даних з різних наборів даних, наприклад, із п’яти різних масивів даних для розпізнавання голосу, три поля, які є в одному з них, можуть бути відсутніми в чотирьох інших масивах.

Щоб очистити цей набір даних, необхідно переконатися, що використовується та сама назва, що і дескриптор у наборі даних.

Викиди в наборі даних. Викиди дуже небезпечні. Вони можуть сильно вплинути на результати моделі машинного навчання. Зазвичай дослідники оцінюють викиди, щоб визначити, чи є кожен конкретний запис результатом помилки у зборі даних або унікальним явищем, яке слід враховувати при обробці даних.

Відсутні дані. Можна помітити, що відсутні деякі важливі значення. Ці проблеми виникають через людський фактор, помилки програми чи інші причини. Вони вплинуть на точність прогнозів, тому перед тим, як продовжувати роботу з базою даних, потрібно виконати очищення даних.

3.2.2 Очищення даних

Мета очищення даних – надати прості, повні та зрозумілі набори прикладів для машинного навчання.

Відсутні дані. Ситуація, коли у наборі даних відсутні дані, є досить поширеною. У цьому випадку необхідно відшукати додаткові набори даних або більше спостережень.

При об’єднанні двох або більше наборів даних в одну базу даних, щоб отримати більший навчальний набір, деякі невідповідності полів даних є досить поширеними. Якщо в об’єднаних масивах представлені не всі поля, краще їх заздалегідь видалити перед об’єднанням.

Якщо для будь-якого з рядків або стовпців бази даних відсутнє більше 50% значень, доведеться видалити цілий рядок/стовпець, якщо неможливо заповнити відсутні значення.

Зашумлені дані. Велика кількість додаткових безглуздих даних називається шумом. Це можуть бути дублікати або напівдублікати записів даних, сегменти даних, які не мають значення для конкретного дослідження, непотрібні інформаційні поля для кожної зі змінних.

Можна застосувати один із таких методів для вирішення цієї проблеми:

- Бінінг. Якщо є пул відсортованих даних. Необхідно поділити усі дані на менші сегменти однакового розміру та застосувати методи підготовки набору даних окремо на кожному сегменті. Наприклад, можна розділити значення віку на такі категорії, як 21-35, 36-59 та 60-79.
- Регресія. Регресійний аналіз допомагає визначити, які змінні справді впливають на результат прогнозу великих обсягів даних. Це дозволить працювати лише з ключовими ознаками, а не намагатися проаналізувати переважну кількість змінних.
- Кластеризація. Нарешті, можна застосувати алгоритми кластеризації для групування даних. Тут потрібно бути обережним з викидами.

Викиди. Важливо не замінювати викиди, сприймаючи їх як шум. Наприклад, при побудові алгоритму, який сортує різні сорти яблук. У наборі даних ми можемо зустріти два типи відхилень: зображення містять екзотичні фрукти, такі як ананаси та ківі. Вони можуть бути знайдені у даних через помилку вибірки та представляти шум у наборі даних.

Також можуть бути фотографії деяких «дивних яблук», наприклад, які мають дивну форму. Коли мета – навчити машину розпізнавати сорти яблук, важливим є відхилення від груп. Такі відхилення допоможуть навчити модель розпізнавати спеціальні символи та підвищити точність прогнозу.

3.2.3 Перетворення даних

Насправді, очистивши та згладивши дані, вже здійснено модифікацію даних.

Агрегація. У разі агрегування даних дані об’єднуються разом і подаються в єдиному форматі для аналізу даних. Робота з великою кількістю високоякісних даних дозволяє отримати більш надійні результати від моделі. Якщо ми хочемо побудувати алгоритм нейронної мережі, що імітує стиль Вінсента Ван Гога, нам потрібно надати якомога більше картин цього відомого художника, щоб забезпечити достатньо матеріалів для навчання. Зображення повинні мати однаковий цифровий формат, і для цього ми використаємо методи перетворення даних.

Нормалізація допомагає масштабувати дані в межах діапазону, щоб уникнути побудови неправильних моделей під час навчання та виконання аналізу даних. Якщо діапазон даних дуже широкий, порівняти цифри буде важко. За допомогою різних методів нормалізації можна трансформувати вихідні дані лінійно, виконувати десяткове масштабування або нормалізацію Z-оцінки.

Вибір ознак – це вибір змінних у даних, які є найкращими предикторами для змінної, яку ми хочемо передбачити. Якщо функцій багато, тоді час роботи класифікатора збільшується. Крім того, точність прогнозування часто знижується.

Особливо, якщо в даних є багато функцій, які не співвідносяться з цільовою змінною.

Дискретизація. Під час дискретизації програміст перетворює дані на набори невеликих інтервалів. Наприклад, віднести людей до категорій «молодий», «середній вік», «старший», а не працювати з постійними віковими значеннями. Дискретизація допомагає підвищити ефективність.

Генерація концепції ієрархії. Якщо ви використовуєте метод генерації ієрархії концепції, ви можете створити ієрархію між атрибутами там, де вона не була вказана. Наприклад, якщо у вас є інформація про місцезнаходження, яка включає вулицю, місто, провінцію та країну, але вони не мають ієрархічного порядку, цей метод може допомогти вам перетворити дані.

Узагальнення. За допомогою узагальнення можна перетворити функції даних низького рівня в характеристики даних високого рівня. Наприклад, адреси будинків можна узагальнити до визначень вищого рівня, таких як місто чи країна.

3.2.4 Зменшення даних

Коли необхідно працювати з великими обсягами даних, важче знайти надійні рішення. Скорочення даних можна використовувати для зменшення обсягу даних та зменшення витрат на аналіз.

Дослідникам дуже потрібне зменшення даних при роботі з наборами словесних мовних даних. Масивні датасети містять індивідуальні особливості динаміків, наприклад, вставки та наповнення слів. У цьому випадку величезні бази даних можна зменшити до репрезентативної вибірки для аналізу.

Вибір функції атрибута. Методи перетворення даних також можуть бути використані для зменшення даних. Якщо ви створюєте нову функцію, що поєднує дані функції, щоб зробити процес видобутку даних більш ефективним, це називається вибором атрибутів. Наприклад, ознаки чоловіка/жінки та студента можна перетворити на студента/студентку. Це може бути корисно, якщо ми

проводимо дослідження щодо того, скільки чоловіків або жінок є студентами, але їхня сфера навчання нас не цікавить.

Зменшення розмірності. Набори даних, які використовуються для вирішення реальних завдань, мають величезну кількість функцій. Комп'ютерний зір, генерація мови, переклад та багато інших завдань не можуть погіршити швидкість роботи заради якості. Можна зменшити розмірність, щоб зменшити кількість використовуваних функцій. Зменшення чисельності – це метод зменшення даних, який замінює вихідні дані меншою формою подання даних. Існує два типи методів зменшення чисельності - параметричний та непараметричний.

3.3 Виконання попередньої обробки даних

Застосовуючи усі описані вище методи та процеси, виконаємо первинну обробку даних, щоб навчання моделі та її подальша результативність були якнайкращими. Ці процедури будуть підкріплюватися графіками та відповідними перетвореннями даних.

Отже, на рисунку 3.2 наведено частину датасету до будь-якої роботи з ним (наведено лише частину колонок). Загальна кількість записів сягає більше 183 тисяч. Так як для більшості моделей машинного навчання така кількість є навіть надлишковою, то маємо деякі свободи у видаленні неякісних даних без втрати загальності.

	backers_count	blurb	category	converted_pledged_amount	country	country_displayable_name	created_at
0	56	The Backyard will be a community garden in Lak...	{"id":305,"name":"Community Gardens","slug":"f...	5507	US	the United States	1487809696
1	1	Modern London is home to a poverty driven ambi...	{"id":293,"name":"Drama","slug":"film & video/...	1	GB	the United Kingdom	1469267948
2	35	Announcing a twist of our original party game ...	{"id":273,"name":"Playing Cards","slug":"games...	1119	US	the United States	1519236804
3	37	Overstock card renovation plan	{"id":273,"name":"Playing Cards","slug":"games...	1236	HK	Hong Kong	1566103145
4	0	Straight Up Photography is a project to bring ...	{"id":277,"name":"Nature","slug":"photography/...	0	US	the United States	1422146426

Рисунок 3.2 – Частина датасету до його первинної обробки

3.3.1 Очищення від зайвих в контексті дослідження колонок та створення нових

Слід також зауважити, що всього в датасеті наявні 38 колонок, більшість з яких не несе важливої інформації для аналізу та передбачення успіху кампанії. Отже, після очищення колонок було прийняти рішення залишити лише 11:

- кількість тих, хто підтримав фінансово кампанію;
- категорія;
- країна;
- мета (в грошовому еквіваленті);
- час запуску кампанії (у вигляді Unix Timestamp);
- назва кампанії;
- об’єм зібраних коштів;
- чи пропонується даний проект серед рекомендацій сайту;
- чи заручилася кампанія підтримкою сайту;
- стан (успішний чи ні);
- час зміни стану (на успішний чи провальний);
- батьківська категорія (якщо наявна).

Також окремо було виділено ще деякі колонки:

- рік запуску кампанії;
- час «життя» кампанії, тобто протягом якого часу відбувався збір коштів.

Внаслідок проведення цих перетворень були видалені дві колонки, які відповідали часу запуску та часу зміни стану.

Відповідні перетворення відбуваються паралельно з тестовим набором даних (це новіші дані за другу половину квітня – початок травня). Після проведення цих первинних перетворень у ньому було 7347 рядків. Його вигляд наведено на рисунку 3.3.

	name	country	year	parent_category	category	spotlight	staff_pick
51	Suburban Legend - EP	the United States	2021	Music	Pop	True	True
96	Craig's Brother: Full-Length Album Phase 2 and...	the United States	2021	Music	Punk	True	False
125	Ödesboxen med Charta 77	Sweden	2020	Music	Punk	True	False
312	California Classical	the United States	2021	Music	Classical Music	True	False
322	Startide Vinyl Pressing	the United States	2021	Music	Electronic Music	True	False
...	...	...	...	...	...	...	...
182911	Emily Scott Robinson's Sophomore Album	the United States	2021	Music	Country & Folk	True	False
182915	Mathæus Bech – debut album	Denmark	2020	None	Music	True	False
182943	"The Wilderness and the Wasteland Shall Be Glad"	the United States	2021	None	Music	True	False

Рисунок 3.3 – Тестовий набір даних після первинної обробки даних

3.3.2 Кореляційний аналіз даних перед очисткою даних

Кореляційний аналіз дозволяє виявити на ранніх етапах деякі залежності між даними та на цій підставі залишити чи викинути їх з розгляду. Кореляційна матриця на даному етапі представлена на рисунку 3.4.

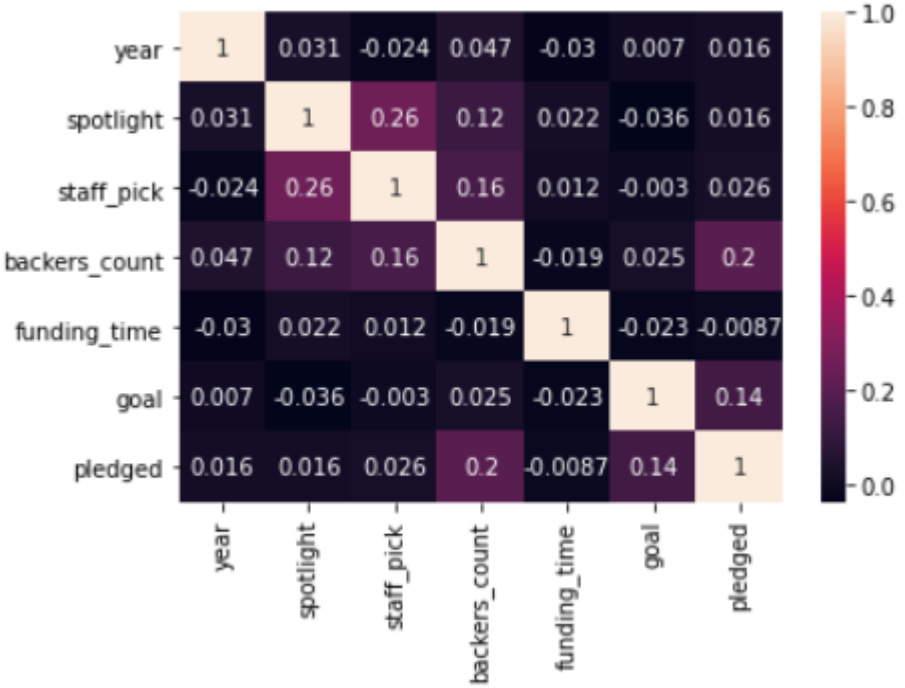


Рисунок 3.4 – Кореляційна матриця перед очисткою даних

Як ми бачимо, сильних залежностей не простежується, а такі змінні як рік та час збору коштів взагалі мають незначний вплив на результат та інші змінні. Проте поки що робити висновки зарано, адже наявні лише колонки з неперервними даними, а категоріальні все ще не були включені до аналізу. Також будемо намагатися посилити вплив змінних одна на одну за допомогою створення додаткових штучних змінних.

3.3.3 Очистка даних від викидів

Продемонструємо приклад очистки даних від викидів на прикладі стовпчика з грошовою метою кампанії. Для цього будемо використовувати стандартний підхід, заснований на квантилях. Відповідні графіки перетворення зображені на рисунку 3.5. Зазначимо також, що максимальне значення цієї колонки зменшилося від 100000000 до 45237.

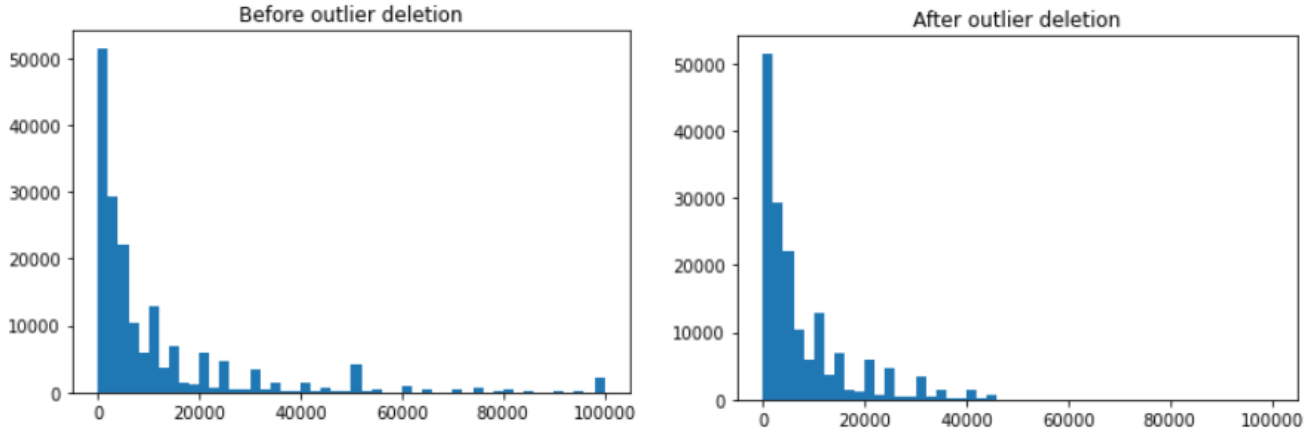


Рисунок 3.5 – Значення змінних до та після видалення викидів

3.3.4 Нормалізація даних

Використаємо декілька підходів до нормалізації даних та порівняємо їх. Спочатку застосуємо логарифмічне перетворення. Результат зображено на рисунку 3.6. Максимальне значення після такого перетворення становить 15, а мінімальне – -6.

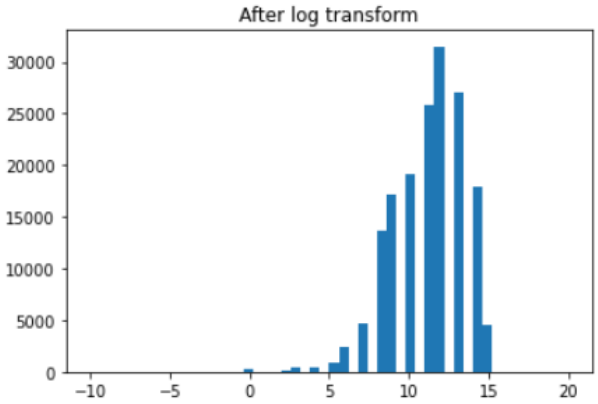


Рисунок 3.6 – Після логарифмічного перетворення

Тепер використаємо мінімаксне перетворення. Результат наведено на рисунку 3.7. Максимальне значення за такого перетворення становить 100, а мінімальне – 1. Це відбулося внаслідок того, що такі границі були задані перед перетворенням.

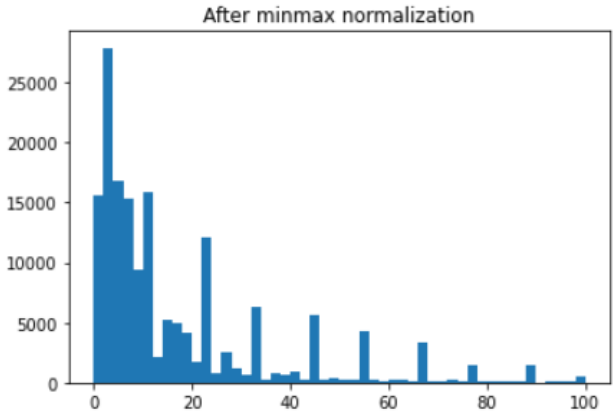


Рисунок 3.7 – Після мінімаксного перетворення

Тепер настала черга нормалізації за допомогою взяття кореня. Результат наведено на рисунку 3.8. Максимальне значення за такого перетворення становить 213, а мінімальне – 1.

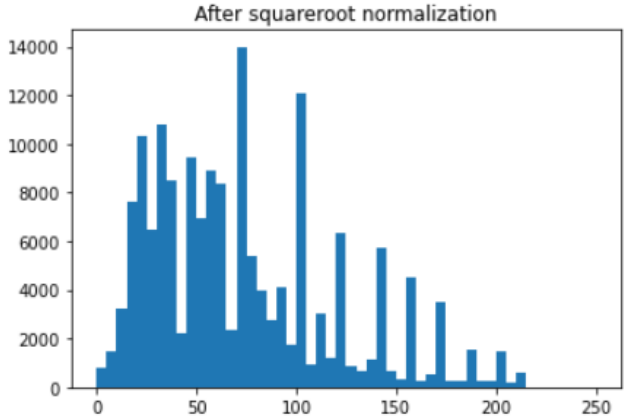


Рисунок 3.8 – Після перетворення за допомогою взяття кореня

Після проведення цих перетворень визначимо найкраще з них, яке будемо застосовувати надалі. Будемо використовувати два критерії: близькість до нормального розподілу та «розділеність» даних за значеннями. Логарифмічне перетворення дуже нагадує нормальний розподіл, але усі значення розміщені на відрізку довжиною 21, що не дуже добре з боку опису, так як значення 12 мають близько 30 тисяч прикладів з набору даних. Мінімаксне перетворення не змогло значно змінити розподіл, він так і залишився схожим на експоненційний, хоча й можна задати будь-які границі для розміщення даних. Тому в цьому плані найкращою є нормалізація з взяттям кореня. Вона дає прийнятний розподіл і має широкий спектр значень.

3.3.5 Інші перетворення даних та створення нових

Щоб нормалізувати рік запуску кампанії, було вирішено відняти від актуального рік заснування Kickstarter [2] 2008, щоб отримати менші значення. Вони наведені на рисунку 3.9.

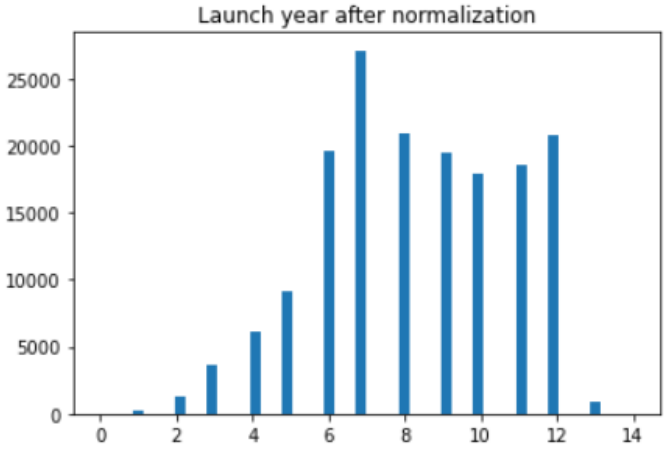


Рисунок 3.9 – Рік запуску кампанії після перетворення

З отриманого графіку можна зробити декілька висновків за темою дослідження. По-перше, можна помітити, що перші роки роботи платформа зростала не дуже активно, проте зі збільшенням популярності краудфандингових платформ у 2014-2015 спостерігаємо значне зростання кількості проектів. По-друге, в останні роки активність залишається приблизно на такому ж високому рівні, а так як даних за 2021 рік не дуже багато, то це не вважаємо за неточність, а лише як нестачу даних (частина яких потрапила до тестової вибірки).

Дані, які описують час збору коштів, не були нормалізовані, оскільки більша частина проектів завершується через 30-31 днів (це стандартний період для проектів на платформі Kickstarter [2]). Підтвердження цього факту можна знайти на рисунку 3.10.

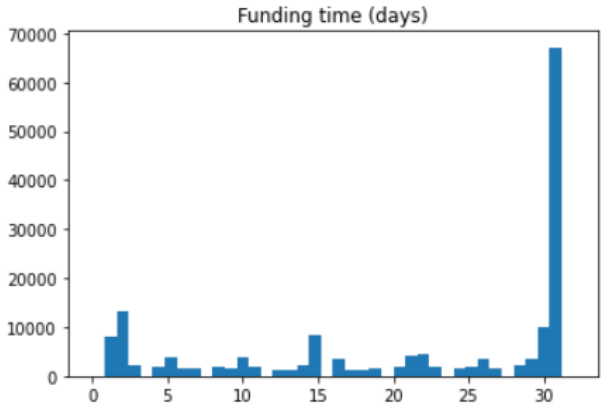


Рисунок 3.10 – Графік розподілу часу збору коштів

Далі створимо нову змінну – очікуваний середній приток коштів за добу. Для цього достатньо розділити очікувану суму на час збору коштів. Отриману метрику додамо до набору даних та проведемо його нормалізацію за допомогою взяття кореня, щ обуло показано вище. Це перетворення показано на рисунку 3.11.

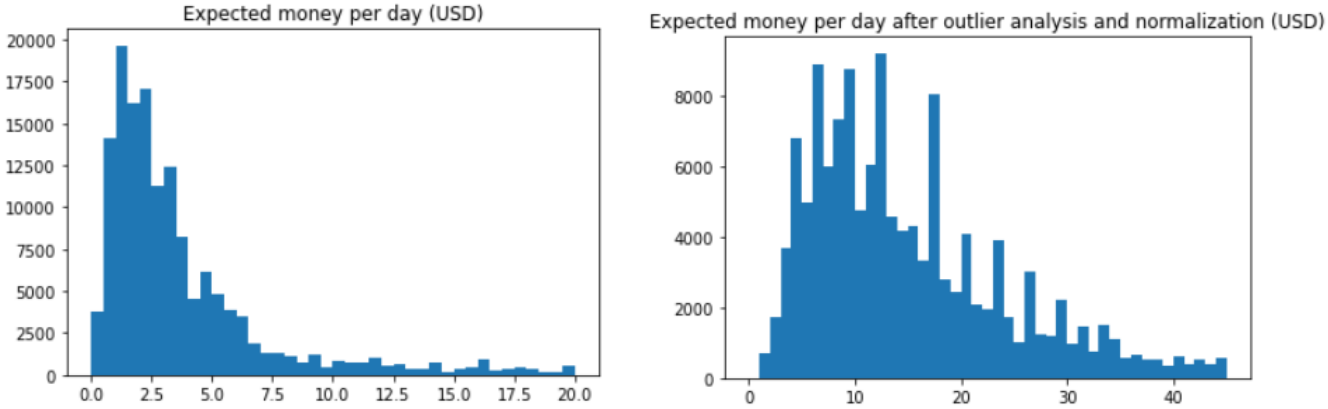


Рисунок 3.11 – Очікуваний приток коштів за добу до та після нормалізації

3.3.6 Перетворення категоріальних даних на неперервні

Для більшості алгоритмів як класифікації так і регресії категоріальні дані не представляють особливої користі. Зазвичай вони викидаються з аналізу. Тому щоб не допустити цього у нашому випадку ми виконаємо перетворення за допомогою LabelEncoder з модуля sklearn Python. Остаточний вигляд набору даних наведено на рисунку 3.12. Зауважимо, що кількість прикладів у тренувальному наборі даних зменшилася до 144 тисяч.

	country	year	category	staff_pick	funding_time	goal	pledged	state	expectation_per_day
0	24	9	27	0	30	73	5507.77	1	13
1	23	8	40	0	31	188	1.00	0	31
2	24	10	106	0	31	32	1119.00	1	6
3	8	11	106	0	31	90	9699.00	1	15
4	24	10	141	0	31	24	766.00	1	4
...	...	...	...	...	...	...	...	...	...
183265	24	13	137	1	22	174	196586.00	1	40
183266	24	13	137	0	11	11	907.50	1	6
183267	3	13	37	0	8	23	3582.29	1	15
183268	13	13	142	0	30	185	34000.80	1	31
183269	24	13	75	0	31	62	2527.00	0	11

Рисунок 3.12 – Тренувальний набір даних після перетворень

3.3.7 Кореляційна матриця після перетворень даних

Тепер вже виконаємо кореляційний аналіз на основі повного набору змінних. І порівняємо з тим, щ обуло до обробки даних на рисунку 6. Нова кореляційна матриця зображена на рисунку 3.13. В цілому залежності дуже сильно змінилися, адже було виконано декілька нормалізацій та очищення. Якщо звернути увагу на цільову змінну класифікації, то помітимо, що найбільший вплив на неї мають вибір працівників платформи, очікувана сума збору та зібрана сума (але цю змінну доведеться видалити з розгляду, адже для проектів, які ще не завершилися і які є об'єктом дослідження, невідоме це значення).

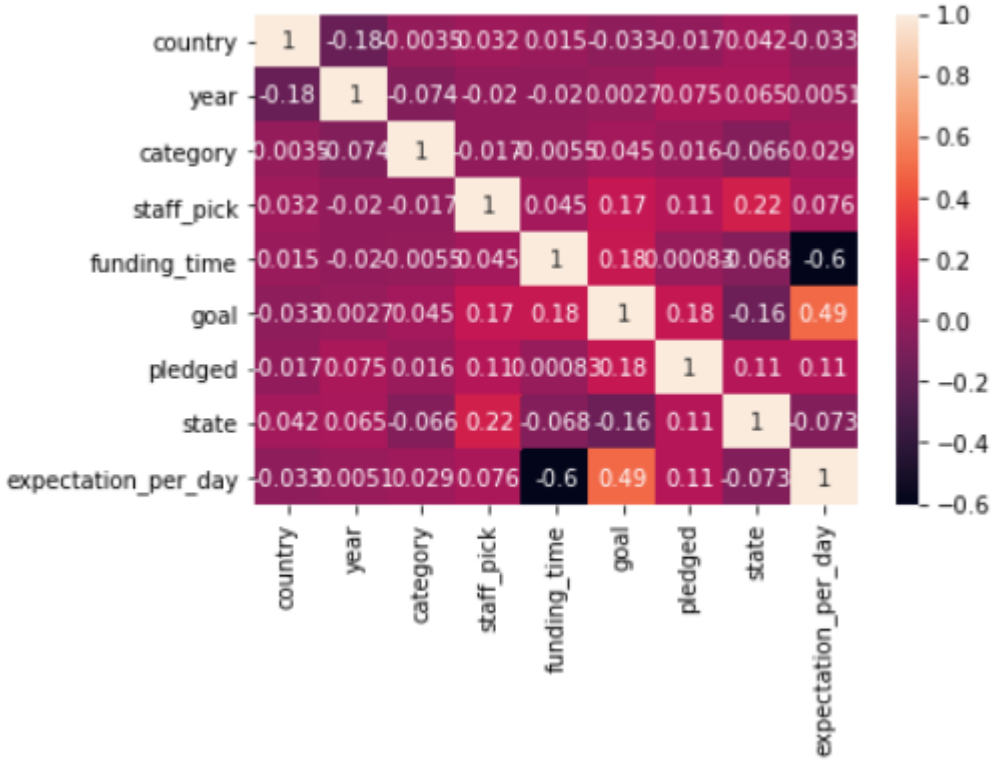


Рисунок 3.13 – Кореляційна матриця після перетворень даних

Також можна помітити незначний вплив таких змінних як рік, категорія та час збору коштів на цільову змінну класифікації. Це можна пояснити тим, що для такого загального набору даних, в якому присутні більше 140 тисяч прикладів, такі вищезазначені категорії не відіграють великої ролі. Проте кореляційна матриця вказує лише на лінійні залежності (що яскраво демонструють ознаки, які ми самостійно додали до датасету на основі вже існуючих за допомогою простої операції ділення), тому не можна бути впевненими, що інших видів залежностей між цими змінними не існує.

3.3.8 Пост-аналіз отриманих після обробки даних

Тепер, отримавши дані у тому вигляді, у якому вони будуть застосовуватися для навчання моделі, можемо проаналізувати та знайти деякі залежності, які будуть корисними в рамках дослідження впливу на успіх або невдачу при створенні кампанії на платформі Kickstarter.

Наприклад, проаналізуємо, в яких категоріях найімовірніший успіх проекту. Відповідні дані наведені на рисунку 3.14.

category	state
Chiptune	0.769231
Latin	0.757009
Fashion	0.736842
Social Practice	0.725000
Taxidermy	0.714286
...	...
Movie Theaters	0.584746
Toys	0.575342
Publishing	0.571429
Crochet	0.570423
Film & Video	0.570312

Рисунок 3.14 – Ймовірності успіху проекту у різних категоріях

Всього в наявності 161 категорія і треба сказати, що для такої великої кількості різниця у 20% між крайніми категоріями не є дуже суттєвою. Такі показники пов’язані з тим, що взагалі середня ймовірність успіху проекту рівна 65-67%, тобто набір даних буде трохи «скошеним» в бік успіху. Але незважаючи на це, найуспішнішими категоріями є чиптюн (різновид музики), латинська (музика), мода, соціальна практика та таксидермія. А найменш успішними є кінотеатри, іграшки, публікації, в’язання гачком, фільми та відео.

Тепер спробуємо проаналізувати, як впливає країна походження проекту на його успіх чи невдачу. Відповідні дані наведені на рисунку 3.15.

country	state
Slovenia	1.000000
Poland	0.714286
Austria	0.707317
Greece	0.692308
Switzerland	0.664723
Belgium	0.662757
Germany	0.661214
Singapore	0.660714
Mexico	0.659280
Ireland	0.655172
Australia	0.654503

Рисунок 3.15 – Залежність країни і успіху проекту

В даному випадку ми віднайшли ще одні дані, які претендують на звання викидів, тому що в Словенії було започатковано всього 4 стартапи і всі вони виявилися успішними. Тому це ніщо інше, як аномальні значення. Далі серед «найуспішніших» країн для започаткування кампанії є Польща, Австрія, Греція та Швейцарія. Найгірші показники показують азіатські країни. Але це пов’язано з тим, що для них існують інші платформи для краудфандингу, спрямовані на їх регіон. Отже, як ми впевнилися, країна також не дуже сильно впливає на успішність проекту, адже різниця успіху складає приблизно 10-14% (для Японії цей показник становить 0,576923).

Якщо вдатися до аналізу залежності успіху від очікуваної суми інвестицій, то він виявиться більш вдалим за ті, що були розглянуті до цього. Відповідний графік зображено на рисунку 3.16. Тут яскраво простежується лінійна залежність: чим більше запит ініціатора проекту тим менше ймовірність успіху. Проте на графіку видно, що стаються час від часу випадки, коли це правило не працює, особливо для більших грошових сум. Проект може стати як дуже успішним внаслідок залучення

великого об’єму інвестицій, так і стати провальним через перевищення очікуваної суми бажання користувачів інвестувати в даний проект.

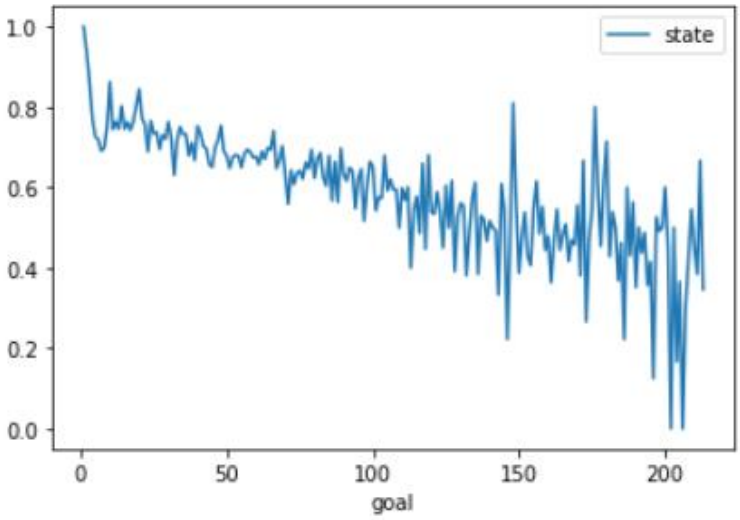


Рисунок 3.16 – Графік залежності успіху проекту від очікуваної суми інвестицій

Тепер дослідимо залежність успіху від року запуску проекту. Якщо тримати в голові графік з рисунку 3.9, то стає очевидною причина провалу проектів у 2014-2015. Цей період є тим часом, коли краудфандинг став дуже популярним і кожен, хто хотів, створював свої проекти. І часто такі пробні проекти завершувалися невдачею. Відповідний графік можна побачити на рисунку 3.17.

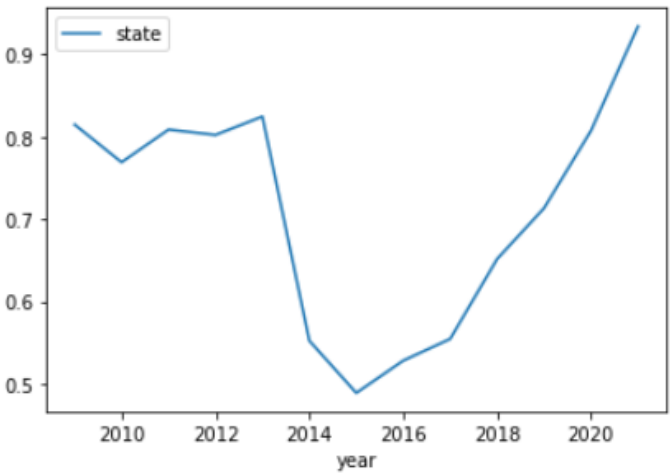


Рисунок 3.17 – Графік залежності успіху проекту від року запуску

Саме завдяки цьому графіку можна зробити висновок, що більшість проектів з тестової вибірки будуть мати успішний стан, так як в останні роки спостерігається зростання якості проектів і сягає 80-90%.

Інших залежностей успіху від, наприклад, часу збору коштів або очікуваного притоку коштів за добу, виявлено не було. Це доводять графіки, зображені на рисунку 3.18.

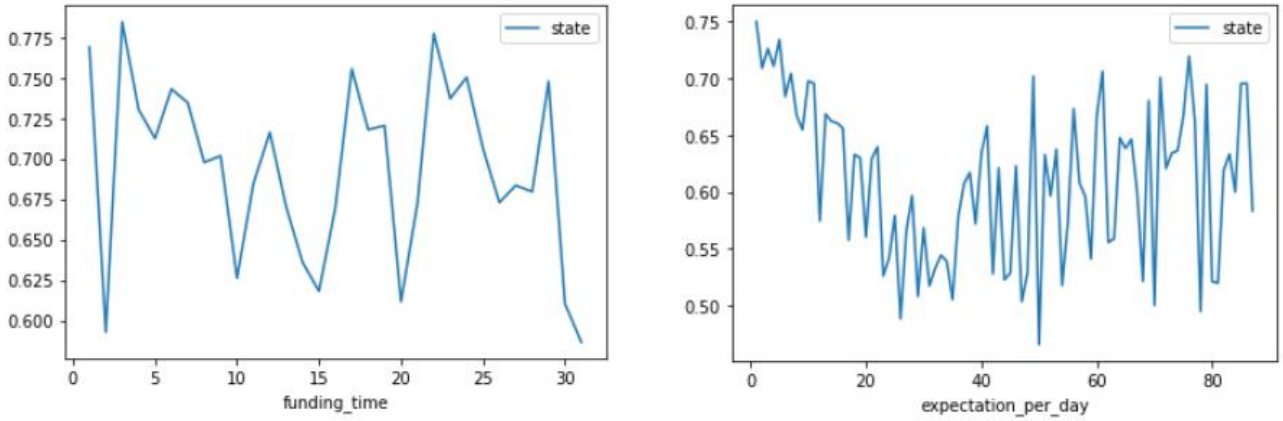


Рисунок 3.18 – Графіки залежності успіху від інших змінних

3.4 Висновки до розділу 3

Попередня обробка даних дозволяє видалити небажані дані за допомогою очищення даних, це дозволяє користувачеві мати набір даних, який міститиме більш цінну інформацію після етапу попередньої обробки для обробки даних пізніше в процесі видобутку даних. Редагування такого набору даних для виправлення пошкодження даних або людської помилки є важливим кроком для отримання точних кванторів, таких як справжні позитиви, справжні негативи, помилкові позитиви та помилкові негативи, знайдені в матриці неточностей, які зазвичай використовуються для медичної діагностики. Користувачі можуть об'єднувати файли даних разом і використовувати попередню обробку для фільтрації будь-якого непотрібного шуму від даних, що може забезпечити більш високу точність. Користувачі використовують сценарії програмування Python, що супроводжуються бібліотекою pandas, що дає їм можливість імпортувати дані із

значень, розділених комами, як фрейм даних. Потім фрейм даних використовується для обробки даних, які можуть бути складними завданнями в іншому випадку в Excel. pandas, який є потужним інструментом, що дозволяє аналізувати дані та маніпулювати ними; що полегшує візуалізацію даних, статистичні операції та багато іншого.

Причина, по якій користувач перетворює наявні файли на нові, полягає в багатьох причинах. Попередня обробка даних має на меті додавати відсутні значення, сукупну інформацію, мітити дані за категоріями (збігання даних) та згладжувати траєкторію. Більш досконалі методи, такі як аналіз основних компонентів та вибір функцій, працюють із статистичними формулами і застосовуються до складних набори даних, які реєструються GPS-трекерами та пристроями зйомки руху.

Отже, в рамках даного розділу було досліджено різні методи попередньої обробки даних, які були застосовані до вихідного датасету.

РОЗДІЛ 4 ОПИС ПРОГРАМНОГО ПРОДУКТУ

4.1 Опис засобів та методів програмування

Для побудови моделей та їх дослідження було використано засоби мови Python та модулів NumPy, pandas, sklearn. А для розробки самого продукту, з яким буде працювати користувач, написана за допомогою фреймворку flask.

4.1.1 Python

Python – інтерпретована, об’єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Вбудовані структури даних високого рівня в поєднанні з динамічним набором тексту та динамічним прив'язуванням роблять його дуже привабливим для швидкої розробки додатків, а також для використання в якості мови сценаріїв або склеювання для з'єднання існуючих компонентів разом. Простий, легкий у вивченні синтаксис Python підкреслює читабельність і, отже, зменшує витрати на обслуговування програми. Python підтримує модулі та пакети, що заохочує модульність програми та повторне використання коду. Інтерпретатор Python та велика стандартна бібліотека доступні у вихідній або двійковій формі безкоштовно для всіх основних платформ і можуть вільно розповсюджуватися [19].

Часто програмісти закохуються в Python через підвищену продуктивність, яку він забезпечує. Оскільки не існує кроку компіляції, цикл редагування-тестування-налагодження неймовірно швидкий. Налагодження програм Python дуже просто: помилка або неправильне введення ніколи не спричинить помилку сегментації. Натомість, коли інтерпретатор виявляє помилку, виникає виняток. Коли програма не вловлює виняток, інтерпретатор друкує трасування стека. Налагоджувач рівня джерела дозволяє перевіряти локальні та глобальні змінні, оцінювати довільні вирази, встановлювати точки зупинку, переходити через код за рядком за один раз тощо. Налагоджувач написаний на самому Python, що свідчить про інтроспективну силу Python. З іншого боку, часто найшвидшим способом

налагодження програми є додавання кількох операторів друку до джерела: швидкий цикл редагування-тестування-налагодження робить цей простий підхід дуже ефективним [19].

4.1.2 NumPy

NumPy – це основний пакет наукових обчислень на Python. Це бібліотека Python, яка забезпечує використання багатовимірного об'єкту масиву, різні похідні об'єкти (наприклад, масковані масиви та матриці) та асортимент підпрограм для швидких операцій над масивами, включаючи математичні, логічні, маніпуляції з фігурами, сортування, вибір, введення/виведення, дискретне перетворення Фур'є основної лінійної алгебри, основні статистичні операції, випадкове моделювання та багато іншого.

В основі пакету NumPy лежить об'єкт ndarray. Це інкапсулює n-вимірні масиви однорідних типів даних, причому багато операцій виконуються в скомпільованому коді для продуктивності. Існує кілька важливих відмінностей між масивами NumPy та стандартними послідовностями Python:

Масиви NumPy мають фіксований розмір при створенні, на відміну від списків Python (які можуть динамічно зростати [19]). Зміна розміру ndarray створить новий масив і видалить оригінал [20].

Всі елементи масиву NumPy повинні мати однаковий тип даних, і, отже, вони матимуть однаковий розмір у пам'яті. Виняток: можна мати масиви об’єктів (Python, включаючи NumPy), дозволяючи таким чином масиви різних за розміром елементів.

Масиви NumPy сприяють вдосконаленим математичним та іншим типам операцій над великою кількістю даних. Як правило, такі операції виконуються ефективніше та з меншим кодом, ніж це можливо за допомогою вбудованих послідовностей Python [20].

Зростаюча кількість науково-математичних пакетів на основі Python використовують масиви NumPy; хоча вони, як правило, підтримують введення послідовності Python, вони перетворюють такий вхід у масиви NumPy перед обробкою, і вони часто виводять масиви NumPy. Іншими словами, для ефективного використання більшої частини сучасного науково-математичного програмного забезпечення на базі Python, просто знання того, як використовувати вбудовані типи послідовностей Python, недостатньо - потрібно також знати, як використовувати масиви NumPy [20].

4.1.3 Pandas

Pandas – це пакет Python, що забезпечує швидкі, гнучкі та виразні структури даних, покликані зробити роботу з «реляційними» або «позначеними» даними одночасно легкою та інтуїтивно зрозумілою. Він має на меті стати фундаментальним елементом високого рівня для практичного аналізу даних у реальному світі на Python. Окрім того, вона має більш широку мету – стати найпотужнішим та найбільш гнучким інструментом аналізу та маніпуляції з відкритим кодом, доступним будь-якою мовою [21].

Pandas добре підходить для багатьох різних типів даних:

- табличні дані з неоднорідно набраними стовпцями, як у таблиці SQL або таблиці Excel;
- впорядковані та невпорядковані (не обов’язково фіксовані частоти) дані часових рядів;
- довільні дані матриці (однорідно введені або неоднорідні) з мітками рядків і стовпців;
- будь-яка інша форма спостережних або статистичних наборів даних. Дані зовсім не потрібно мітити, щоб розміщувати їх у структурі даних pandas.

Дві основні структури даних pandas, Series (1-мірна) та DataFrame (2-мірна), обробляють переважну більшість типових випадків використання у фінансах,

статистиці, соціальних науках та багатьох галузях техніки. Для користувачів R DataFrame надає все, що забезпечує R. DataFrame, та багато іншого. pandas побудований на базі NumPy і призначений для доброї інтеграції в науково-обчислювальне середовище з багатьма іншими сторонніми бібліотеками [21].

4.1.4 Scikit-learn (sklearn)

Проект scikit-learn розпочався як scikits.learn, проект Google Summer of Code від Девіда Курно. Його назва походить від уявлення про те, що це "SciKit" (SciPy Toolkit), окремо розроблене та розповсюджене стороннім розширенням SciPy [19]. Пізніше оригінальну базу коду переписали інші розробники. У 2010 році Фабіан Педрегоза, Гаель Вароку, Александр Грамфорт і Вінсент Мішель, усі з Французького інституту досліджень в галузі комп'ютерних наук та автоматики в Роккенкурі, Франція, взяли на себе керівництво проектом і зробили перший публічний реліз 1 лютого 2010 року. [22] У листопаді 2012 року серед різноманітних науково-дослідних робіт науково-дослідний процес, а також науково-дослідний імідж були названі «добре підтримуваними та популярними» [22]. Scikit-learn - одна з найпопулярніших бібліотек машинного навчання на GitHub.

Scikit-learn в основному написаний на Python і широко використовує NumPy для високопродуктивних лінійних алгебр та операцій з масивами. Крім того, деякі основні алгоритми написані на Cython для підвищення продуктивності. Машини векторної підтримки реалізовані обгорткою Cython навколо LIBSVM; логістична регресія та лінійна підтримка векторних машин подібною обгорткою навколо LIBLINEAR. У таких випадках розширення цих методів за допомогою Python може бути неможливим.

Scikit-learn добре інтегрується з багатьма іншими бібліотеками Python, такими як Matplotlib та інтуїтивно для побудови графіків, NumPy для векторизації масивів, фреймів даних Pandas, SciPy та багатьох інших [22].

4.1.5 Flask

Flask – це веб-фреймворк. Це означає, що він надає вам інструменти, бібліотеки та технології, що дозволяють створювати веб-додаток. Ця веб-програма може бути декількома веб-сторінками, щоденником, вікі-програмою чи мати такий великий розмір, як веб-програма календаря чи комерційний веб-сайт [23].

Flask є частиною категорії мікрофреймворку. Мікрофреймворк – це зазвичай фреймворк, який майже не залежить від зовнішніх бібліотек. У цьому є плюси і мінуси. Плюси полягають у тому, що фреймворк є легким, тут мало залежностей для оновлення та стеження за помилками безпеки, мінуси в тому, що іноді вам доведеться зробити більше роботи самостійно або збільшити собі список залежностей, додавши плагіни. У випадку з Flask його залежності:

- Werkzeug – утиліта WSGI, яка дозволяє встановлювати з’єднання з зовнішнім сервером;
- jinja2 – механізм шаблонів.

4.2 Порівняння методів класифікації

В рамках виконання поставленої задачі передбачення результату виконання проекту на краудфандинговій платформі Kickstarter було досліджено та порівняно наступні моделі машинного навчання для класифікації:

- k найближчих сусідів;
- метод опорних векторів;
- дерево рішень;
- випадковий ліс;
- нейронна мережа;
- наївний Байєс;
- градієнтний бустинг дерева рішень;
- AdaBoost дерева рішень.

Зазначимо, що усі параметри для моделей були підібрані за допомогою решітчастого пошуку, тому вони є найоптимальнішими серед можливих. Хоча для деяких алгоритмів не всі параметри включалися до пошуку, але можна з великою ймовірністю стверджувати, що їх результативність все одно є кращою за моделі за замовчуванням. Результати порівняння за допомогою метрик якості наведені у таблиці 4.1. Для класифікації такими метриками є точність, повнота, метрика F1 та ROC-AUC, значення яких ми і будемо аналізувати. А на рисунках 4.1 та 4.2 зображено відповідні матриці неточностей, на основі яких можна обчислити такі показники, як помилки першого на другого роду, специфічність.

Таблиця 4.1 – Порівняння методів класифікації

Метод	Клас	Точність	Повнота	F1	ROC-AUC
k найближчих сусідів	успіх	0.84	0,85	0,85	0.6447333269421276
	невдача	0.46	0,43	0,45	
метод опорних векторів	успіх	0,85	0,90	0,87	0,6674793084715431
	невдача	0,55	0,44	0,49	
дерево рішень	успіх	0,81	0,98	0,89	0,579577815230243
	невдача	0,76	0,18	0,28	
випадковий ліс	успіх	0,79	1,00	0,88	0,5483298181701979
	невдача	0,86	0,10	0,18	
нейронна мережа	успіх	0,80	0,98	0,88	0,5571656600517687
	невдача	0,64	0,14	0,88	
наївний Байєс	успіх	0,88	0,82	0,84	0,7074427188189053
	невдача	0,48	0,60	0,54	
градієнтний бустинг дерева рішень	успіх	0,87	0,97	0,92	0,7352532515259005
	невдача	0,84	0,50	0,63	
AdaBoost дерева рішень	успіх	0,78	1,00	0,88	0,5100586393123062
	невдача	0,92	0,02	0,04	

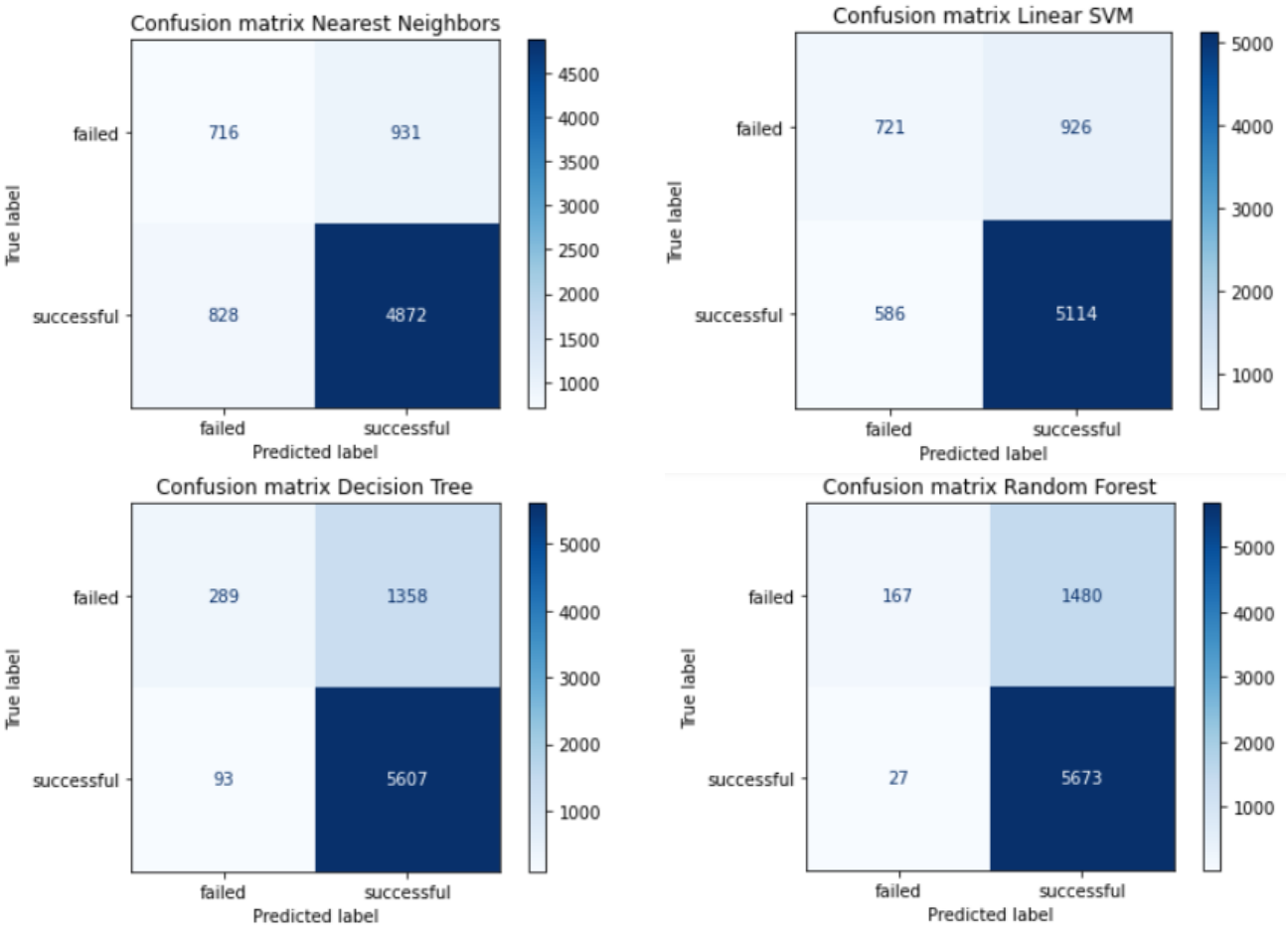


Рисунок 4.1 – Перша частина матриць неточностей для розглянутих алгоритмів

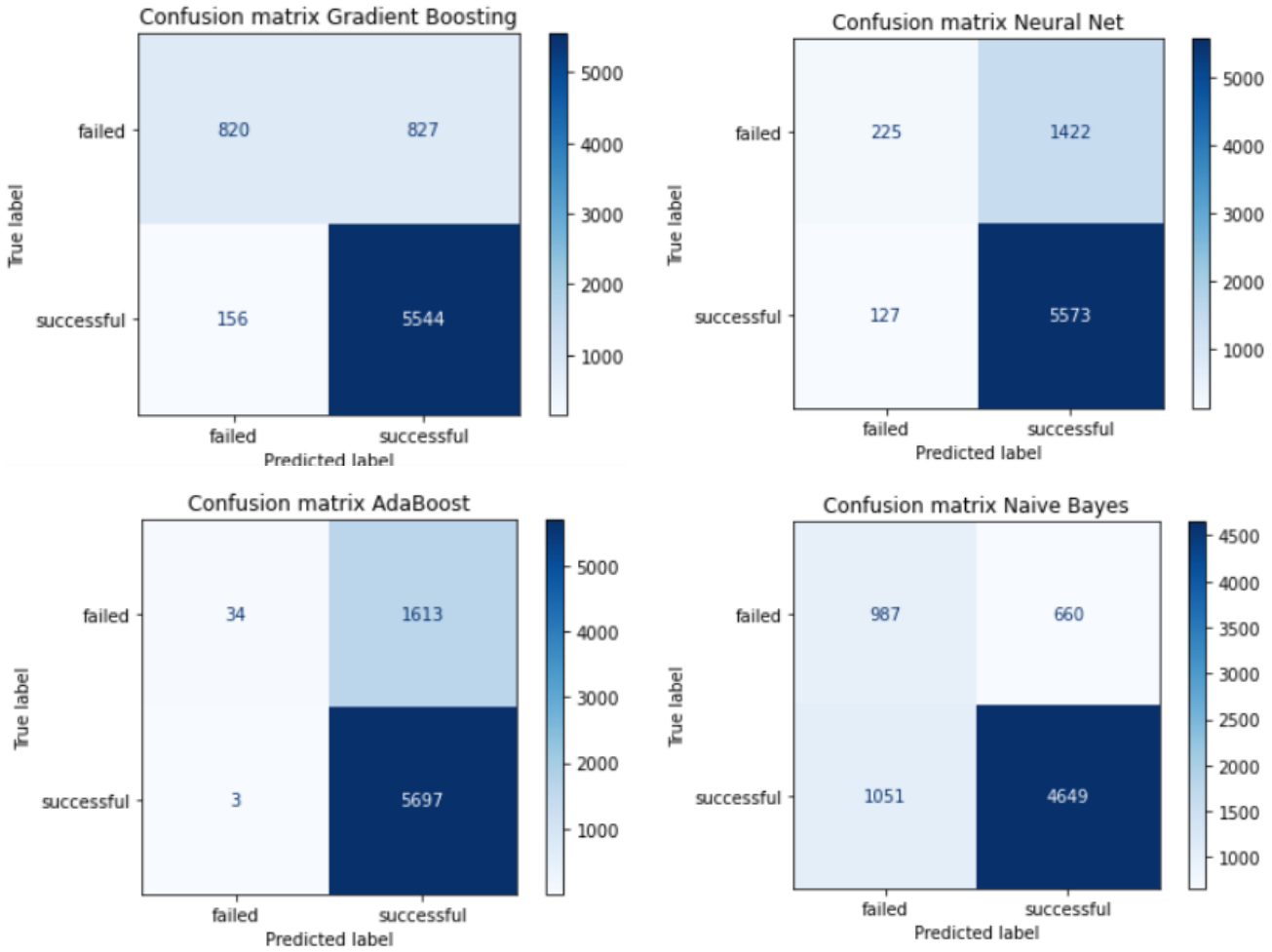


Рисунок 4.2 – Друга частина матриць неточностей для розглянутих алгоритмів

Отже, з отриманих даних метрик можемо зробити висновок, що найкращим алгоритмом для даного датасету є градієнтний бустинг дерева рішень, що є не одинокою моделлю, а ансамблем. Для цього методу виявився найвищим показник ROC-AUC, а так як це ансамбль, то його якість можна значно покращити за допомогою варіювання параметрів. також непогану якість моделі показав алгоритм наївного байєса, але якість його передбачення класу проектів з невдачею дуже низька, що видно з дуже низькох повноти для цього класу.

4.3 Вибір найкращої моделі класифікації

Виходячи з отриманих результатів для різних моделей класифікації, спробуємо змінити параметри моделі градієнтного бустингу, щоб ще покращити її.

Спробуємо варіювати кількість дерев (n_estimators – N_est) для уточнення результатів та параметр швидкості навчання (learning rate – LR). Окрім наведених вище метрик якості будемо також для порівняння використовувати значення точності для навчальної та тестової вибірки. Результати наведені у таблиці 4.2.

Таблиця 4.2 – Порівняння моделей градієнтного бустингу дерев з різними параметрами

LR	N_est	Клас	Точність	Повнота	F1	Навчальний набір	Тестовий набір
0,1	200	успіх	0,87	0,97	0,92	0,7833338028	0,86620389274
		невдача	0,84	0,50	0,63		
	400	успіх	0,89	0,96	0,92	0,8021311660	0,87858990063
		невдача	0,81	0,60	0,69		
	700	успіх	0,91	0,95	0,93	0,8129208102	0,88389818973
		невдача	0,78	0,66	0,72		
0,7	200	успіх	0,89	0,94	0,91	0,8182452038	0,86157615353
		невдача	0,74	0,58	0,65		
	400	успіх	0,88	0,95	0,91	0,8253584809	0,85871784401
		невдача	0,76	0,54	0,63		
	700	успіх	0,87	0,94	0,90	0,8315209736	0,84490936354
		невдача	0,72	0,51	0,60		
1,2	200	успіх	0,89	0,95	0,92	0,8150125362	0,86838165237
		невдача	0,77	0,59	0,67		
	400	успіх	0,88	0,93	0,31	0,8189311603	0,85164012522
		невдача	0,71	0,57	0,63		
	700	успіх	0,88	0,95	0,91	0,8203580584	0,85950672383
		невдача	0,75	0,56	0,64		

Виходячи з отриманих результатів, можна зробити висновок, що найкращою моделлю класифікації для даного набору даних буде модель градієнтного бустингу з швидкістю навчання 0,1 та кількістю дерев, що апроксимують рішення, рівною 700. Отже, саме цю модель будемо використовувати для передбачення в нашому веб-додатку.

4.4 Порівняння методів регресії

В рамках виконання поставленої задачі передбачення результату виконання проекту на краудфандинговій платформі Kickstarter було досліджено та порівняно наступні моделі машинного навчання для регресії:

- k найближчих сусідів;
- метод опорних векторів;
- дерево рішень;
- випадковий ліс;
- нейронна мережа;
- гребенева регресія;
- ласо регресія;
- регресія ElasticNet;
- регресія автоматичного визначення мети;
- байєсівська гребенева регресія;
- стохастичний градієнтний спуск;
- ансамбль Bagging для дерева рішень;
- градієнтний бустинг дерева рішень;
- AdaBoost дерева рішень.

Зазначимо, що усі параметри для моделей були підібрані за допомогою решітчастого пошуку, тому вони є найоптимальнішими серед можливих. Хоча для деяких алгоритмів не всі параметри включалися до пошуку, але можна з великою ймовірністю стверджувати, що їх результативність все одно є кращою за моделі за замовчуванням. Для регресії такими метриками є метрика R2, середньоквадратична похибка, а також середня абсолютна помилка. Відповідні дані наведені у таблиці 4.3.

Таблиця 4.3 – Порівняння досліджуваних видів регресій

Метод	R2	MSE	MAE
гребенева регресія	0,0143183619	608021775151	74442
ласо регресія	0,0143183385	608021789631	74104
регресія ElsticNet	0,0136731343	608419786481	74104
регресія автоматичного визначення мети	0,0141610535	608118811496	74054
байєсівська гребенева регресія	0,0143142114	608024335404	74441
метод опорних векторів	-0,007090367	621227837895	68159
стохастичний градієнтний спуск	-73831727851	4.554340519224001e+23	364020895744
k найближчих сусідів	-0,0012594386	617361005578	72277
дерево рішень	-0,0516325717	648703880051	96200
AdaBoost	-0,0002347797	616998940540	128421
Bagging	0,02357102275	602314233269	85203
градієнтний бустинг	0,03803829523	593390036697	73343
випадковий ліс	0,03647377647	594355116532	83683
нейронна мережа	-0,0907665037	672843806714	87250

Для регресій також найкращим методом виявився градієнтний бустинг. Проте якість усіх моделей не є дуже високою. Помилки дуже великі і сягають дуже великих значень. Це можна було б побороти за допомогою регуляризації або нормалізації самої колонки, але так як вона невідома до проведення аналізу, то ці способи не представляється можливим зробити. Метод опорних векторів має нижче значення R2, але помилку нижчу. Можна було б обрати за основний

алгоритм саме його, але час навчання моделі сягає більше години для такого великого набору даних. В той час як градієнтний бустинг витрачає всього декілька хвилин на тренування. Отже, робимо вибір на користь градієнтного бустингу для дерева рішень.

4.5 Вибір методу регресії

Всі варіювання параметрів відбуваються аналогічно до тих, що були проведені для алгоритму класифікації. Відповідні результати наведені у таблиці 4.4.

Таблиця 4.4 – Порівняння моделей градієнтного бустингу для регресії

Швидкість навчання	Кількість дерев	Тренувальний R2	Тестовий R2
0,1	200	0,199656153	0,0331572780
	400	0,238378263	0,0281068264
	700	0,268401614	0,0352253281
0,7	200	0,3002253116	0,0246549367
	400	0,3485503389	0,0246243327
	700	0,3908531509	0,0110292154
1,2	200	0,3052967827	-0,147952706
	400	0,3665281402	-0,028947139
	700	0,4114378660	-0,082041746

Проаналізувавши отримані результати, можемо зробити висновок про перенавченість моделі, адже результативність на тестовій вибірці на порядок менша за ту, що показує модель не тренувальній. Причому чим більше зростає

метрика якості (R2) на тренувальній вибірці, тим більше зменшується він на тестовій вибірці. Отже, в якості опорної моделі оберемо ту, для якої показник якості на тестовій вибірці був найвищим. Ця модель зі швидкістю навчання 0,1, а кількістю дерев рівною 700. Також сам цю модель будемо використовувати для розробки веб-додатку.

4.6 Програмний продукт

Як зазначалося вище, програмний продукт написаний за допомогою мікрофреймворку flask, який є доволі легковісним та буде ідеальним рішенням для нашого веб-додатку. Робота складається з двох основних частин:

- передбачення успіху проекту за допомогою скрапінгу сайту Kickstarter за посиланням, яке буде надано користувачем. Для цього необхідно з самого сайту завантажити та обробити відповідні об’єкти: рік, категорія, мета краудфандингу, розташування, вибір редакції сайту, час збору коштів та інші;
- передбачення успіху проекту за допомогою даних, які були перелічені у попередньому пункті і які будуть надані користувачем через інтерактивну форму на веб-сторінці.

Повний процес роботи програми описано за допомогою блок-схеми, зображеної на рисунку 4.3.

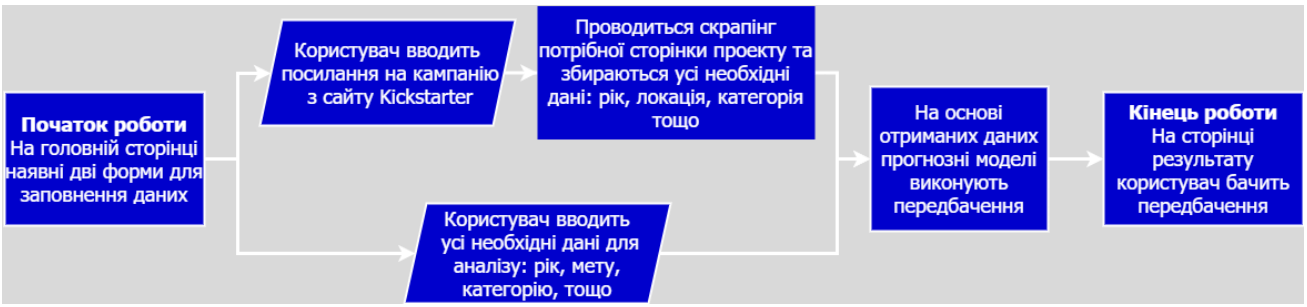


Рисунок 4.3 – Блок-схема роботи програми

Покажемо повний процес роботи програми з боку користувача. По-перше, користувач має обрати проект на сайті Kickstarter, який його цікавить. Це може бути навіть його власний проект. Необхідно лише мати посилання для продовження роботи. Нехай користувач обере проект, головна сторінка якого зображена на рисунку 4.4. Посилання на нього: https://www.kickstarter.com/projects/insurgence-coffee/a-yawn-is-but-a-silent-scream-for-coffee?ref=discovery_newest. Ця кампанія спрямована на відкриття кавового ресторану з оригінальними блендами та кавовими міксами з усіх куточків світу, як стверджує власник.

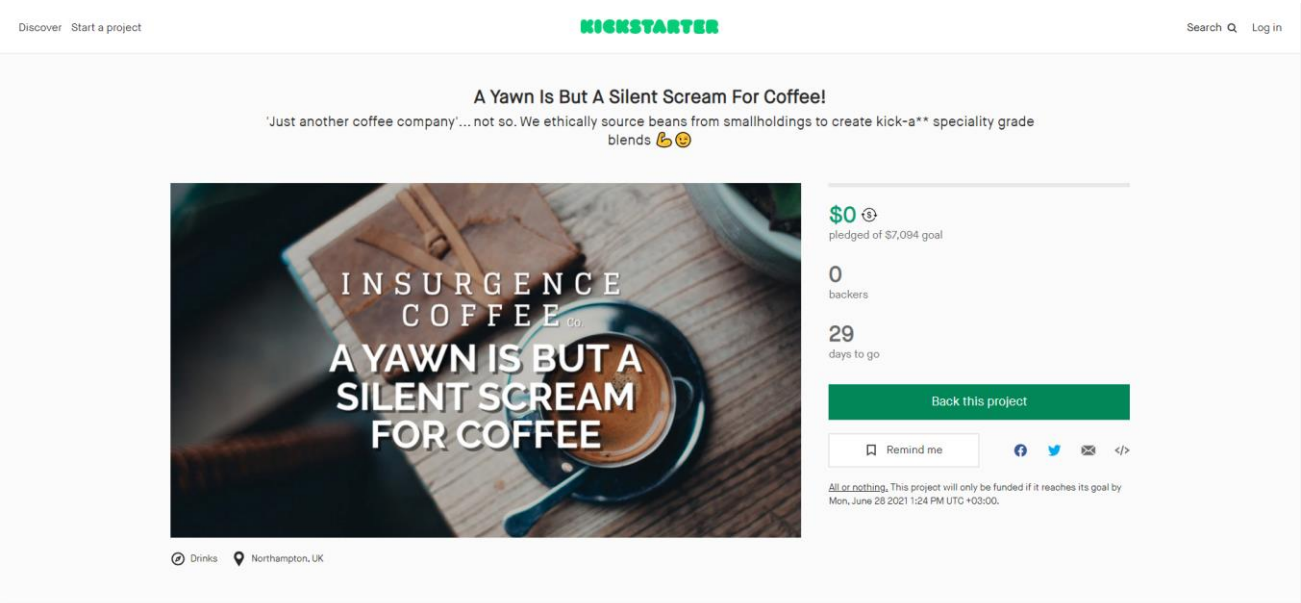


Рисунок 4.4 – Головна сторінка кампанії, яка цікавить користувача

Після вибору проекту та отримання посилання на нього, користувач переходить до веб-додатку та вводить його до відповідного поля. Цей процес зображено на рисунку 4.5. Оформлення сайту дуже просте, проте воно може змінюватися, проте головні свої функції воно виконує справно.

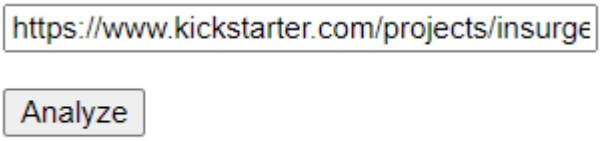


Рисунок 4.5 – Форма для надання посилання на цікавлячий проект

Після натискання на кнопку Analyze, користувач матиме зачекати 3-5 секунд для того, аби веб-драйвер під керуванням Selenium зміг отримати цікавлячі дані про проект. Далі ці дані будуть передані до моделей, які були визначені на попередніх етапах роботи. В результаті чого користувач отримає відповідний прогноз щодо успіху або невдачі проекту, а також прогнозованого об’єму грошей інвестицій. Ці дані, відповідно до прикладу, наведені на рисунку 4.6.

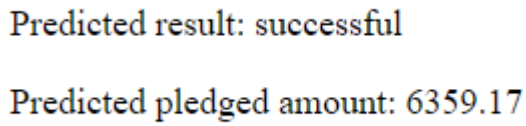


Рисунок 4.6 – Результати виконання програми

Отже, як можна побачити, модель дала відповідь, що цей проект після свого завершення буде вдалим та матиме 6359,17 доларів інвестицій. На жаль, такий прогноз не має особливого сенсу, адже мета збору проекту становить близько 7000 доларів. Але такий прогноз також може мати право на життя. В будь-якому разі, слід віддавати перевагу прогнозу, який дає модель класифікації, так як вона показала якість ROC-AUC близько 89%. А модель регресії є не дуже результативною, хоча прогноз є реальним, якщо взяти до уваги саме мету збору коштів.

4.7 Висновки до розділу 4

В рамках цього розділу було описано розроблений програмний продукт, засоби та моделі, які були використані для цього. Особливу увагу приділено моделям класифікації та регресії, які були використані для порівняння. Найкращі моделі виявилися градієнтним бустингом дерев рішень, як для класифікації, так і для регресії. Після вибору та решітчастого пошуку за параметрами були отримані

бажані моделі, які потім застосовувалися для використання в рамках веб-додатку, який був розроблений за допомогою мікрофреймворку flask для мови програмування Python.

Для моделі класифікації була отримана непогана якість, яка доводиться показником ROC-AUC рівним майже 89% на тестовій вибірці. Для регресії таких показників, на жаль, досягти не вдалося, але достатній адекватний рівень передбачається.

РОЗДІЛ 5 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

У даному розділі проводиться оцінка основних характеристик програмного продукту, розробленого для вирішення задач прогнозування результатів виконання проектів на платформі Kickstarter за допомогою моделей машинного навчання.

Нижче наведено аналіз різних варіантів реалізації модулю з метою вибору оптимальної, з огляду при цьому як на економічні фактори, так і на характеристики продукту, що впливають на продуктивність роботи і на його сумісність з апаратним забезпеченням. Для цього було використано апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) – це технологія, яка дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. ФВА проводиться з метою виявлення резервів зниження витрат за рахунок ефективніших варіантів виробництва, кращого співвідношення між споживчою вартістю виробу та витратами на його виготовлення. Для проведення аналізу використовується економічна, технічна та конструкторська інформація.

Алгоритм функціонально-вартісного аналізу включає в себе визначення послідовності етапів розробки продукту, визначення повних витрат (річних) та кількості робочих часів, визначення джерел витрат та кінцевий розрахунок вартості програмного продукту.

5.1 Постановка задачі проектування

У роботі застосовується метод ФВА для проведення техніко-економічного аналізу розробки системи прогнозування успіху проекту на платформі Kickstarter. Оскільки рішення стосовно проектування та реалізації компонентів, що розробляється, впливають на всю систему, кожна окрема підсистема має її задовольняти. Тому фактичний аналіз представляє собою аналіз функцій

програмного продукту, призначеного для збору, обробки та проведення аналізу даних по коронавірусу.

Технічні вимоги до програмного продукту є наступні:

- функціонування на персональних комп'ютерах із стандартним набором компонентів;
- зручність та зрозумілість для користувача;
- швидкість обробки даних та доступ до інформації в реальному часі;
- можливість зручного масштабування та обслуговування;
- мінімальні витрати на впровадження програмного продукту.

5.2 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка програмного продукту, який вирішує задачу прогнозування успішності кампанії та будує його модель. Беручи за основу цю функцію, можна виділити наступні:

- F_1 – вибір мови програмування;
- F_2 – вибір фреймворку машинного навчання;
- F_3 – вибір середовища розробки.

Кожна з цих функцій має декілька варіантів реалізації:

Функція F_1:

а) Python

б) C++

Функція F_2:

а) Tensorflow;

б) Sklearn

Функція F_3:

а) Jupyter Notebook;

б) PyCharm Professional Edition.

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рисунок 5.1).

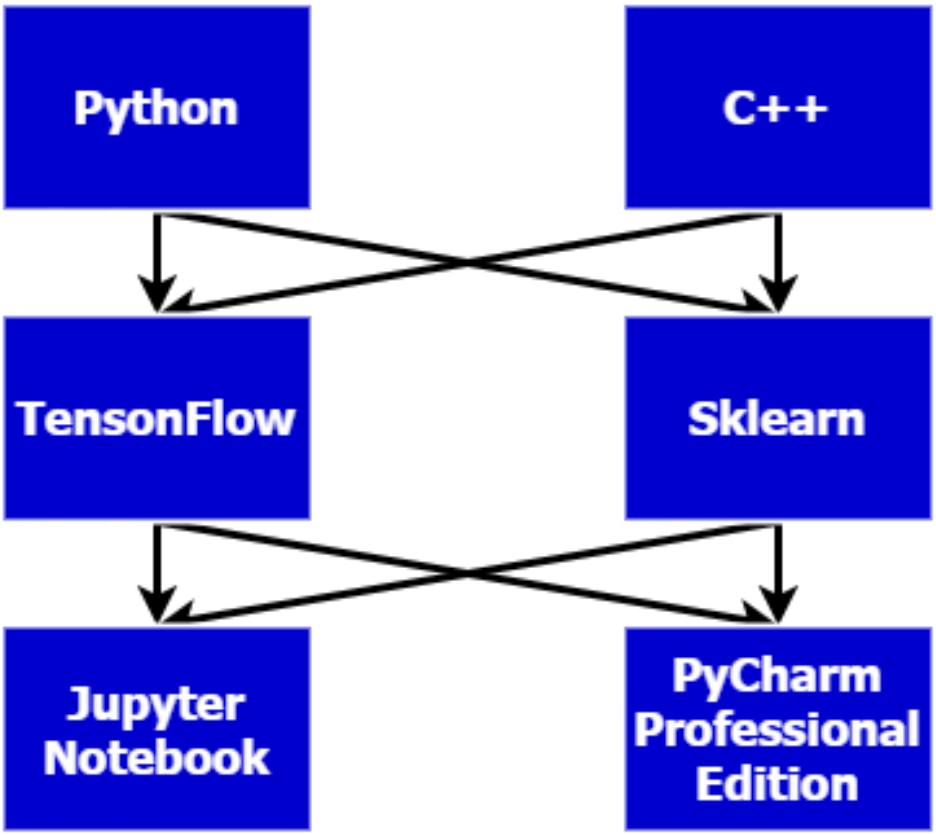


Рисунок 5.1 – Морфологічна карта

Морфологічна карта відображає множину всіх можливих варіанти основних функцій.

На основі цієї карти будуємо позитивно-негативну матрицю варіантів основних функцій (таблиця 5.1). Робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Таблиця 5.1 – Позитивно-негативна матриця

Функції	Варіанти реалізації	Переваги	Недоліки
F_1	*А*	Швидка розробка програми, доступність бібліотек, кросплатформеність	Низька швидкість роботи, особливо, якщо потрібно обробляти велику кількість даних
	Б	Код швидко виконується	Іде багато часу на розробку програми
F_2	*А*	Надійно працює зі складними проектами	Додатковий час на інсталяцію та вивчення
	Б	Надійність	Не підтримується багатьма мовами
F_3	*А*	Підтримується багатьма мовами програмування, легко запускається на будь-якому сервері	Відсутня можливість роботи без інтернету
	Б	Багато інструментів, безпечна	Підтримує одночасно лише одну мову програмування

Функція F_1: перевагу віддаємо швидкості вивчення, простоті використання та наявності стандартних бібліотек для обчислення. Для спрощення роботи по написанню коду варіант Б має бути відкинутий.

Функція F_2: обидва варіанти можна використовувати в розробці.

Функція F_3: віддаємо перевагу варіанту А в разі вибору мови програмування Python.

Таким чином, будемо розглядати такі варіанти реалізації ПП:

- F_1а – F_2а – F_3а
- F_1а – F_2б – F_3а

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

5.3 Обґрунтування системи параметрів ПП

На основі даних, розглянутих вище, визначаються основні параметри вибору, які будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- *X1* – швидкодія мови програмування;
- *X2* – об’єм пам’яті для обчислень та збереження даних;
- *X3* – час навчання даних;
- *X4* – потенційний об’єм програмного коду.

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію ПП як показано у таблиці 5.2.

Таблиця 5.2 – Основні параметри ПП

Назва параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкодія мови програмування	X1	оп/мс	8000	15000	20000
Об’єм пам’яті	X2	Мб	1024	512	256
Час попередньої обробки даних	X3	мс	50	20	10
Потенційний об’єм програмного коду	X4	кількість рядків коду	1500	1000	750

За даними таблиці 5.3 будуються графічні характеристики параметрів – рисунки 5.2 – 5.5.

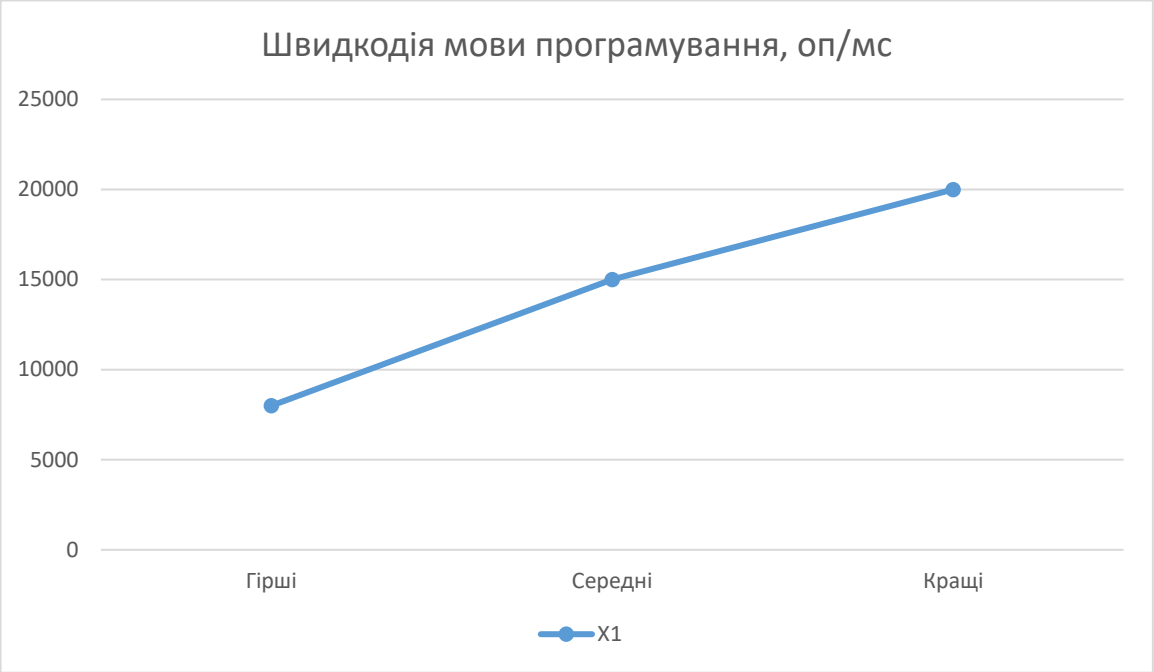


Рисунок 5.2 – X1, швидкодія мови програмування

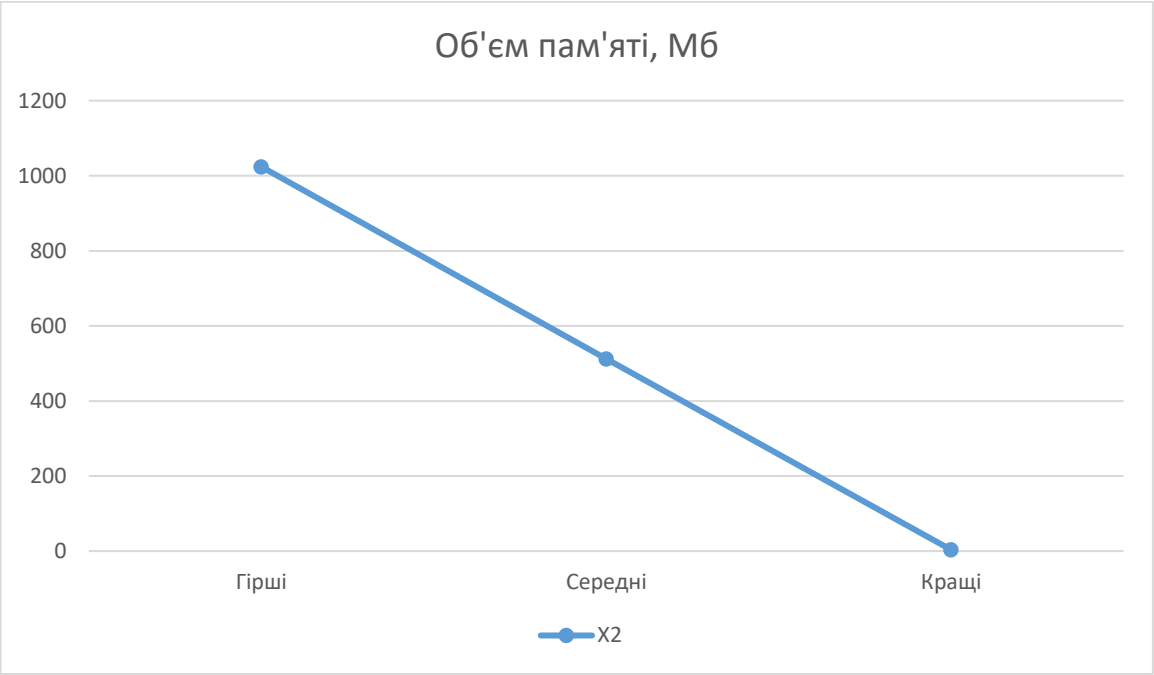


Рисунок 5.3 – X2, об’єм пам’яті

Рисунок 5.4 – X3, час попередньої обробки даних

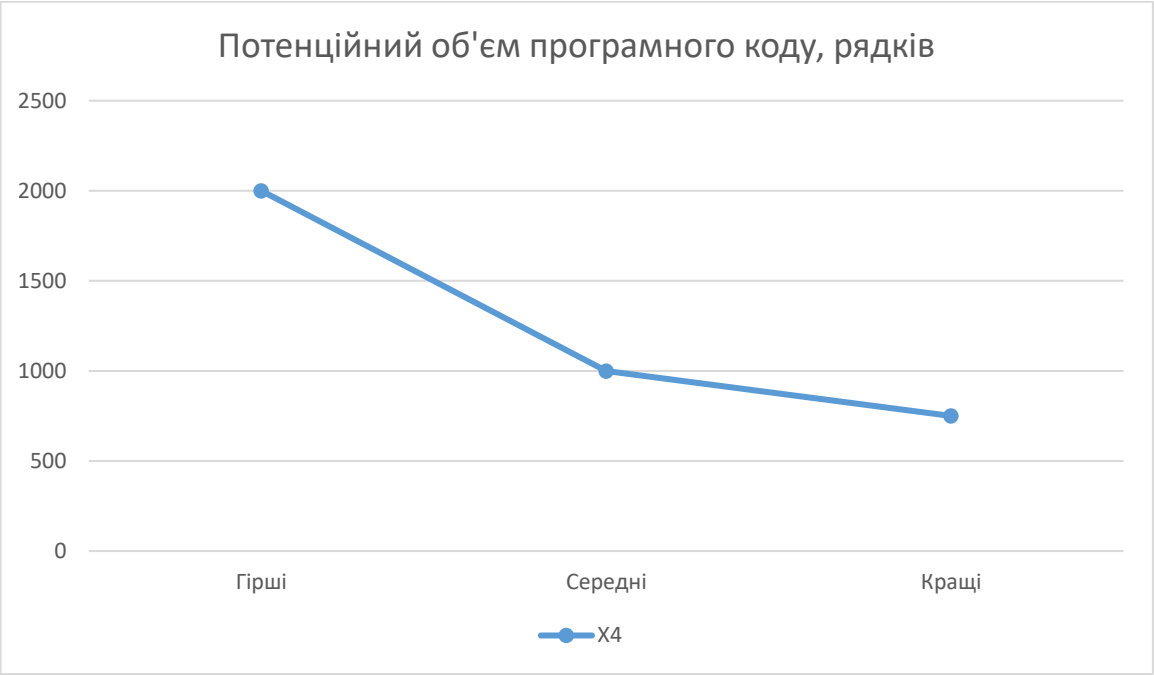


Рисунок 5.5 – X4, потенційний об’єм програмного коду

5.4 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні результати при знаходженні параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 5 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 5.3.

Таблиця 5.3 – Результати ранжування параметрів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта					Сума рангів R_i	Відхи-лення Δ_i	Δ_i^2
			1	2	3	4	5			
X1	Швидкодія мови програмування	Оп/мс	4	1	1	2	1	9	-3,5	12,25
X2	Об’єм пам’яті	Мб	2	5	2	3	2	14	1,5	2,25
X3	Час попередньої обробки даних	мс	1	3	5	2	2	13	0,5	0,25
X4	Об’єм програмного коду	Кількість рядків коду	3	1	2	3	5	14	1,5	2,25
Разом			10	10	10	10	10	50	0	17

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

- сума рангів кожного з параметрів і загальна сума рангів наведені у формулі 5.1:

$$R_i = \sum_{j=1}^{N} r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 50, \quad (5.1)$$

де N – число експертів;

n – кількість параметрів.

- середня сума рангів (формула 5.2):

$$T = \frac{1}{n} R_{ij} = 12{,}5 \quad (5.2)$$

- відхилення суми рангів кожного параметра від середньої суми рангів визначено формулою 5.3:

$$\Delta_i = R_i - T. \quad (5.3)$$

Сума відхилень по всім параметрам повинна дорівнювати 0.

- загальна сума квадратів відхилення (формула 5.4):

$$S = \sum_{i=1}^{N} \Delta_i^2 = 75. \quad (5.4)$$

Обчислимо коефіцієнт узгодженості, який визначається згідно формули 5.5:

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 75}{5^2(4^3 - 4)} = 0{,}69 > W_k = 0{,}67. \quad (5.5)$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 5.4.

Таблиця 5.4 – Попарне порівняння параметрів.

Параметри	Експерти					Кінцева оцінка	Числове значення
	1	2	3	4	5		
X1 і X2	>	>	<	<	>	>	1,5
X1 і X3	=	=	<	=	<	=	1,0
X1 і X4	>	>	<	=	>	>	1,5
X2 і X3	<	<	<	<	<	<	0,5
X2 і X4	<	<	<	<	<	<	0,5
X3 і X4	>	<	>	>	<	>	1,5

Числове значення, що визначає ступінь переваги *i*–го параметра над *j*–тим, a_{ij} визначається за формулою 5.6:

$$a_{ij} = \begin{cases} 1.5 \text{ при } X_i > X_j \\ 1.0 \text{ при } X_i = X_j \\ 0.5 \text{ при } X_i < X_j \end{cases}. \quad (5.6)$$

З отриманих числових оцінок переваги складемо матрицю $A = \| a_{ij} \|$.

Для кожного параметра зробимо розрахунок вагомості $K_{ві}$ за наступними формулами 5.7 та 5.8:

$$K_{\text{ві}} = \frac{b_i}{\sum_{i=1}^{n} b_i} \quad (5.7)$$

$$b_i = \sum_{i=1}^{N} a_{ij} \tag{5.8}$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами 5.9 та 5.10:

$$K_{\text{ві}} = \frac{b_i'}{\sum_{i=1}^{n} b_i'}, \tag{5.9}$$

$$b_i' = \sum_{i=1}^{N} a_{ij} b_j \tag{5.10}$$

Як видно з таблиці 5.5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 5.5 – Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.	
	X1	X2	X3	X4	b_i	$K_{\text{ві}}$	b_i^1	$K_{\text{ві}}^1$
X1	1,0	1,5	1,0	1,5	5	0,294	20,5	0,293
X2	0,5	1,0	1,5	0,5	3,5	0,206	15,25	0,218
X3	1,0	1,5	1,0	1,5	5	0,294	20,5	0,293
X4	0,5	1,5	0,5	1,0	3,5	0,206	13,75	0,196
Разом					17	1	70	1

5.5 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів відповідають технічним вимогам умов функціонування даного ПП.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 5.6) згідно формули 5.11:

$$K_K(j) = \sum_{i=1}^{n} K_{\text{в}i,j} B_{i,j}\,, \tag{5.11}$$

де n – кількість параметрів;

$K_{вi}$– коефіцієнт вагомості i–го параметра;

B_i – оцінка i–го параметра в балах.

Таблиця 5.6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Параметри	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	А	X1	15000	3	0,293	0,879
F2	А	X2	512	3	0,218	0,654
	Б	X2	256	5	0,218	1,09
F3	А	X3	20	3	0,293	0,879

Визначаємо рівень якості кожного з варіантів за даними з таблиці 5.6 згідно формули 5.12:

$$K_K = K_{\text{ТУ}}[F_{1k}] + K_{\text{ТУ}}[F_{2k}] + \ldots + K_{\text{ТУ}}[F_{zk}], \tag{5.12}$$

$$\text{К}_{\text{К1}} = 0{,}879 + 0{,}654 + 0{,}879 = 2{,}412,$$
$$\text{К}_{\text{К2}} = 0{,}879 + 1{,}09 + 0{,}879 = 2{,}848.$$

Як видно з розрахунків, кращим є другий варіант, для якого коефіцієнт технічного рівня має найбільше значення.

5.6 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе два окремих завдання:

- Розробка проекту програмного продукту;
- Розробка програмної оболонки;

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 – до групи Б. За складністю алгоритми, які використовуються в завданні 1 належать до групи 1; а в завданні 2 – до групи 3.

Для реалізації завдання 1 використовується довідкова інформація, а завдання 2 використовує інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Загальна трудомісткість обчислюється за формулою 5.13:

$$T_O = T_P \cdot K_П \cdot K_{СК} \cdot K_М \cdot K_{СТ} \cdot K_{СТ.М}, \quad (5.13)$$

де T_P – трудомісткість розробки ПП;

$K_П$ – поправочний коефіцієнт;

$K_{СК}$ – коефіцієнт на складність вхідної інформації;

$K_М$ – коефіцієнт рівня мови програмування;

$K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм;

$K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення.

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру степеню новизни А та групи складності алгоритму 1, трудомісткість

дорівнює: T_P = 90 людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання: $К_П$ = 1,7. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх семи завдань рівний 1: $К_{СК}$ = 1. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $К_{СТ}$ = 0,8. Тоді загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 90 \cdot 1{,}7 \cdot 0{,}8 = 122{,}4 \text{ людино} - \text{днів}.$$

Проведемо аналогічні розрахунки для подальших завдань.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б), тобто T_P = 27 людино-днів, $К_П$ = 0,9, $К_{СК}$ = 1,$К_{СТ}$ = 0,8:

$$T_2 = 27 \cdot 0{,}9 \cdot 0{,}8 = 19{,}44 \text{ людино} - \text{днів}.$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = (122.4 + 19.44 + 4.8 + 19.44) \cdot 8 = 1328{,}64 \text{ людино} - \text{годин}.$$
$$T_{II} = (122.4 + 19.44 + 6.91 + 19.44) \cdot 8 = 1345.52 \text{ людино} - \text{годин}.$$

Найбільш високу трудомісткість має варіант II.

В розробці беруть участь два програмісти з окладом 18000 грн., один аналітик в області даних з окладом 15500 грн. Визначимо середню зарплату за годину за формулою 5.14:

$$С_Ч = \frac{М}{T_m \cdot t}, \quad (5.14)$$

де М – місячний оклад працівників;

T_m – кількість робочих днів на тиждень;

t – кількість робочих годин в день.

$$С_ч = \frac{18000 + 18000 + 15500}{3 \cdot 21 \cdot 8} = 102{,}18 \text{ грн}$$

Тоді, розрахуємо заробітну плату за формулою:

$$С_{ЗП} = С_ч \cdot Т_i \cdot К_Д, \quad (5.15)$$

де $С_ч$– величина погодинної оплати праці програміста;

$Т_i$ – трудомісткість відповідного завдання;

$К_Д$ – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

$$С_{ЗП1} = 102{,}18 \cdot 1328{,}64 \cdot 1{,}2 = 162912{,}52 \text{ грн}$$
$$С_{ЗП2} = 102{,}18 \cdot 1345.52 \cdot 1.2 = 164982{,}28 \text{ грн}$$

Відрахування на єдиний соціальний внесок становить 22%:

$$С_{ВІД} = С_{ЗП} \cdot 0{,}22 = 162912{,}52 \cdot 0{,}22 = 35840{,}75 \text{ грн}$$
$$С_{ВІД} = С_{ЗП} \cdot 0{,}22 = 164982{,}28 \cdot 0{,}22 = 36296{,}10 \text{ грн}$$

Тепер визначимо витрати на оплату однієї машино-години ($С_М$).

Так як одна ЕОМ обслуговує одного програміста з окладом 18000 грн, з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$С_Г = 12 \cdot M \cdot K_З = 12 \cdot 18000 \cdot 0{,}2 = 43200 \text{ грн}$$

З урахуванням додаткової заробітної плати:

$$С_{ЗП} = С_Г \cdot (1 + К_З) = 43200 \cdot (1 + 0{,}2) = 51840 \text{ грн}$$

Відрахування на соціальний внесок:

$$С_{ВІД} = С_{ЗП} \cdot 0{,}22 = 51840 \cdot 0{,}22 = 11404{,}80 \text{ грн}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 20000 грн.

$$С_А = К_{ТМ} \cdot К_А \cdot Ц_{ПР} = 1{,}15 \cdot 0{,}25 \cdot 20000 = 5750 \text{ грн},$$

де $К_{ТМ}$ – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

$К_А$ – річна норма амортизації;

$Ц_{ПР}$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$С_Р = К_{ТМ} \cdot Ц_{ПР} \cdot К_Р = 1{,}15 \cdot 20000 \cdot 0{,}05 = 1150 \text{ грн},$$

де $К_Р$ – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$Т_{ЕФ} = (Д_К - Д_В - Д_С - Д_Р) \cdot t \cdot К_В = (365 - 104 - 12 - 16) \cdot 8 \cdot 0{,}9 =$$
$$= 16776 \text{ годин},$$

де $Д_К$ – календарна кількість днів у році;

$Д_В$, $Д_С$ – відповідно кількість вихідних та святкових днів;

$Д_Р$ – кількість днів планових ремонтів устаткування;

t – кількість робочих годин в день;

$К_В$ – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$С_{ЕЛ} = Т_{ЕФ} \cdot N_С \cdot K_З \cdot Ц_{ЕН} = 1677,6 \cdot 0,6 \cdot 0,6 \cdot 3,52 = 2125,85 \text{ грн},$$

де $N_С$ – середньо-споживча потужність приладу;

$К_З$ – коефіцієнтом зайнятості приладу;

$Ц_{ЕН}$ – тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$С_Н = Ц_{ПР} \cdot 0,67 = 20000 \cdot 0,67 = 13400 \text{ грн}$$

Тоді, річні експлуатаційні витрати будуть визначатися за формулою 5.16:

$$С_{ЕКС} = С_{ЗП} + С_{ВІД} + С_А + С_Р + С_{ЕЛ} + С_Н, \quad (5.16)$$

$$С_{ЕКС} = 51840 + 11404,80 + 5750 + 1150 + 2125,85 + 13400 = \\ = 85670,65 \text{ грн}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$С_{М-Г} = С_{ЕКС} / Т_{ЕФ} = 85670,65 / 1677,6 = 51,07 \text{ грн/год}$$

Оскільки в даному випадку всі роботи, які пов‘язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складають згідно формули 5.17:

$$С_М = С_{М-Г} \cdot Т, \quad (5.17)$$

$$С_{М1} = 51,07 \cdot 1328,64 = 67853,64 \text{ грн}$$
$$С_{М2} = 51,07 \cdot 1345,52 = 68715,70 \text{ грн}$$

Накладні витрати складають 67% від заробітної плати визначені формулою 5.18:

$$С_Н = С_{ЗП} \cdot 0,67 \quad (5.18)$$

$$С_Н = 162912,52 \cdot 0,67 = 109151,04 \text{ грн}$$
$$С_Н = 164982,28 \cdot 0,67 = 110538,13 \text{ грн}$$

Отже, вартість розробки ПП за варіантами становить (формула 5.19):

$$С_{ПП} = С_{ЗП} + С_{ВІД} + С_М + С_Н, \quad (5.19)$$

$$С_{ПП} = 162912,52 + 35840,75 + 67853,64 + 109151,04 = 375757,95 \text{ грн}$$
$$С_{ПП} = 164982,28 + 36296,10 + 68715,70 + 110538,13 = 380532,21 \text{ грн}$$

5.7 Вибір кращого варіанту ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою 5.20:

$$К_{ТЕР j} = К_{Кj} / С_{Фj}, \quad (5.20)$$

$$К_{ТЕР1} = 2,412 / 375757,95 = 6,419 \cdot 10^{-6}$$
$$К_{ТЕР2} = 2,848 / 380532,21 = 7,484 \cdot 10^{-6}$$

Як бачимо, найбільш ефективним є другий варіант реалізації програми з коефіцієнтом техніко-економічного рівня $К_{ТЕР2}= 7,484 \cdot 10^{-6}$.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишились

після першого відбору двох варіантів виконання програмного комплексу оптимальним є перший варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості $К_{ТЕР} = 7,484 \cdot 10^{-6}$.

Цей варіант реалізації програмного продукту має такі параметри:

мова програмування – Python;

використання модулю sklearn;

використання Jupyter Notebook в якості середи розробки.

Даний варіант виконання програмного комплексу дає користувачу зручний інтерфейс, непоганий функціонал і швидкодію.

5.8 Висновки до розділу 5

Проведено повний функціонально-вартісний аналіз програмного продукту. Визначено та проведено оцінку основних функцій програмного продукту. Визначено параметри, які характеризують програмний продукт. Проведено експертне оцінювання параметрів та аналіз якості варіантів реалізації функцій.

Проведено економічний аналіз варіантів розробки – трудомісткість, витрати на заробітну плату та інші витрати.

На основі аналізу вибрано варіант реалізації програмного продукту.

ВИСНОВКИ

Краудфандинг стабільно скорочує цикл розробки нових продуктів, тим самим забезпечуючи більш ранній вихід на ринок. Він служить багатогранним інструментом підтримки на ранніх стадіях для впровадження інновацій. Це не тільки забезпечує приток коштів за розробку та виробництво продукції, що більш важливо, це дозволяє продемонструвати тягу через перевірку ринкового попиту, що базується на функції натовпу як мультиплікатора інформації, що генерує загальний огляд та зворотній зв'язок.

У рамках провадження даного дослідження було виконано усі поставлені задачі:

- зібрані найбільші набори даних з сайту Kickstarter;
- були проаналізовані характеристики успішних проектів та досліджено основні залежності між ознаками;
- досліджені існуючі методи розв’язання поставленої задачі та запропоновано альтернативні за допомогою аналізу даних;
- виконано попередню обробку, очистку та кореляційний аналіз даних;
- розроблено прогнозні моделі класифікації та регресії та оцінено їх якість;
- розроблено програмний продукт з використанням отриманих моделей.

Актуальність дослідження важко переоцінити, адже, згідно зі звітом про краудфандинг, він перейшов пік популярності та перебуває на плато використання, тобто активно застосовується в наш час [6]. А розроблений програмний продукт допоможе усім зацікавленим сторонам інвестування проаналізувати предметну область та оцінити кампанії.

РЕКОМЕНДАЦІЇ ЩОДО ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Отримані результати, беззаперечно, можна покращити і на їх основі зробити більш точний аналіз предметної області й тематики дослідження в цілому.

По-перше, можна було б обрати конкретизовані дані, наприклад, за однією категорією. Так як до різних категорій привертається різна ступінь уваги, то існує сенс спробувати розглянути їх окремо одна від одної, а згодом об’єднати їх у групи за подібністю моделей. Окрім цього, можна розділяти дані за іншими ознаками, як-от рік чи сума зібраних коштів.

З описаної вище мети також випливає можливість категоризації деяких змінних. У пункті 3.3.6 було показано зворотній процес, але в даному випадку наголос саме на перетворенні неперервних даних до категоріальних.

По-друге, можна використати інші методи машинного навчання. Наприклад, так як модуль sklearn представляє не такі широкі можливості для створення нейронних мереж, як, скажімо, TensorFlow, то і якість цих моделей (внаслідок недостатньо гнучких конфігурацій) може бути значно нижчою за потенційно можливі.

І по-третє, окрім покращення результативності моделі, значно можна змінити графічну оболонку програмного продукту. Так як основна мета була саме в створенні та дослідженні різних моделей машинного навчання, то цьому аспекту було приділено значно менше часу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1) Noyes K. Why Investors Are Pouring Millions into Crowdfunding. *Fortune Magazine*. 2014. URL: http://fortune.com/2014/04/17/why-investors-arepouring-millions-into-crowdfunding/ (дата звернення: 26.04.2021).
2) Kickstarter Stats. 2020. URL: https://www.kickstarter.com/help/stats?ref=press (дата звернення: 26.04.2021).
3) Schwienbacher, Armin, Larralde, Benjamin. Crowdfunding of Small Entrepreneurial Ventures. *The Oxford Handbook of Entrepreneurial Finance*. 2010. URL: https://www.researchgate.net/publication/228252861_Crowdfunding_of_Small_Entrepreneurial_Ventures (дата звернення: 26.04.2021).
4) Mollick E. The dynamics of crowdfunding: An exploratory study. January 2014. Vol. 29, No. 1, P. 1-16. URL: https://www.sciencedirect.com/science/article/pii/S088390261300058X (дата звернення: 26.04.2021).
5) Hossain M., Onyema G. Oparaocha Crowdfunding: Motives, Definitions, Typology and Ethical Challenges. *De gruyter*: entrepreneurship research journal. 2017. URL: https://www.degruyter.com/document/doi/10.1515/erj-2015-0045/html (дата звернення: 28.04.2021).
6) The Crowdfunding Industry Report. *Massolution*. 2015. URL: https://www.smv.gob.pe/Biblioteca/temp/catalogacion/C8789.pdf (дата звернення: 28.04.2021).
7) Kurani S. What is crowdfunding and how does it work? 2021. URL: https://republic.co/blog/investor-education/what-is-crowdfunding-and-how-does-it-work (дата звернення: 29.04.2021)
8) Nadine S. The Relevance of Crowdfunding: The Impact on the Innovation Process of Small Entrepreneurial Firms. 2015. URL: https://link.springer.com/chapter/10.1007/978-3-658-09837-7_6. (дата звернення: 30.04.2021)

9) About Us. *Startup Commons* URL: https://www.startupcommons.org/about-us.html. (дата звернення: 30.04.2021)

10) Global Innovation Investment Report. *Crunchbase*. 2016 . URL: https://static.crunchbase.com/reports/annual_2016_yf42a/crunchbase_annual_2016.pdf. (дата звернення: 04.05.2021)

11) Васильчук І. П. Краудфандинг як феномен постіндустріальної економіки. *Ефективна економіка*: електронний журнал. Вип. 11, 2013. URL: http://www.economy.nayka.com.ua/?op=1&z=2500 (дата звернення: 04.05.2021).

12) Machuskyy V. Crowdfunding: the nature and features of implementation in Ukraine. 2020. URL: https://www.businesslaw.org.ua/crowdfunding-the-nature-and-features-of-implementation-in-ukraine/ (дата звернення: 05.05.2021).

13) Vuong Q. Impacts of geographical locations and sociocultural traits on the Vietnamese entrepreneurship. *SpringerPlus*, No. 5(1), Art. 1189. 2016. URL: https://doi.org/10.1186/s40064-016-2850-9 (дата звернення: 05.05.2021).

14) Woods C., Huang, H. Predicting the success of entrepreneurial campaigns in crowdfunding: a spatio-temporal approach. *Innovation Entrepation* Vol. 9, No. 13. 2020 URL: https://doi.org/10.1186/s13731-020-00122-8 (дата звернення: 05.05.2021).

15) Greenberg M., Pardo B., Hariharan K., Gerber E. Crowdfunding support tools: predicting success & failure. *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. 2013. P. 1815–1820.

16) Mitra T., Gilbert E. The language that gets people to give: Phrases that predict success on kickstarter. Proceedings of the 17th ACM conference on *Computer supported cooperative work & social computing*. 2014. P. 49–61.

17) Takeshi A. Tobit Models. Advanced econometrics. *Cambridge Mass*: Harvard University Press. 1985. P. 384.

18) Zhang H. The Optimality of Naive Bayes. *Flairs2004* conference. 2004. URL: https://www.researchgate.net/publication/221439320_The_Optimality_of_Naive_Bayes (дата звернення: 08.05.2021).

19) What is Python? Executive Summary. URL: https://cutt.ly/knpz1tS (дата звернення: 28.05.2021).

20) What is NumPy? URL: https://numpy.org/doc/stable/user/whatisnumpy.html (дата звернення: 28.05.2021).

21) Package overview – pandas documentation. URL: https://pandas.pydata.org/pandas-docs/stable/getting_started/overview.html (дата звернення: 28.05.2021).

22) Scikit-learn – Вікіпедія. URL: https://en.wikipedia.org/wiki/Scikit-learn (дата звернення: 28.05.2021).

23) Introduction to Flask. URL: https://pymbook.readthedocs.io/en/latest/flask.html#:~:text=Flask%20is%20a%20web%20framework,application%20or%20a%20commercial%20website (дата звернення: 28.05.2021)

ДОДАТОК А ЗАПРОПОНОВАНА [5] ТИПОЛОГІЯ КРАУДФАНДИНГУ

Тип	Краудфандинг, що базується на пожертвах	Краудфандинг на основі винагороди	Краудфандинг на основі акціонерного капіталу	Краудфандинг на основі кредитування
Мотивація фінансування	внутрішня та соціальна	внутрішня, соціальна та зовнішня	фінансова вигода	соціальна та / або фінансова
Тип внеску	пожертва	попереднє замовлення	інвестиція	кредит
Очікувана віддача фінансування	нематеріальні вигоди	матеріальні та нематеріальні вигоди	рентабельність інвестицій	рентабельність інвестицій
Основний фокус	журналістика/ гідна справа/ філантропія	товари для тих, хто приймає дітей/ подарунки	стартапи	короткострокові позичальники
Складність процесу	дуже низька	низька	висока	середня
Приклади основних бенефіціарів	власник(и) проекту, музиканти, некомерційні організації	стартапи, фінансуючі	стартапи	фізичні особи, суб’єкти господарювання
Тип контракту	контракт без екзистенціальної винагороди	договір купівлі-продажу	акціонерний контракт	договір позики

ДОДАТОК Б ЛІСТИНГ ПРОГРАМИ

Первинна обробка даних

```
import os
import json
import datetime
import pandas as pd
import numpy as np

train_df = pd.DataFrame()

for dir in [d.path[2:] + os.sep for d in os.scandir() if 'train' in
d.path and not 'csv' in d.path]:
    for file in os.listdir(dir):
        temp = pd.read_csv(dir + file)
        train_df = pd.concat([train_df, temp])
train_df = train_df.drop_duplicates(subset='name')
train_df.head()

train_df.columns

train_df = train_df.drop(['blurb', 'country', 'currency',
'currency_symbol', 'currency_trailing_code', 'current_currency',
                          'disable_communication', 'friends',
'fx_rate', 'id', 'is_backing', 'is_starrable', 'is_starred',
                          'permissions', 'photo', 'slug',
'source_url', 'static_usd_rate', 'urls', 'usd_pledged',
                          'usd_type', 'converted_pledged_amount',
'location', 'creator', 'profile', 'created_at', 'deadline'],
                         axis=1)
train_df = train_df[(train_df.state == 'successful') |
(train_df.state == 'failed')]
train_df = train_df.reset_index(drop=True)

parent_cat, cat = [], []
for i in train_df.category:
    item_json = json.loads(i)
    try:
        parent_cat.append(item_json['parent_name'])
    except KeyError:
        parent_cat.append(None)
    cat.append(item_json['name'])

train_df.drop(columns=['category'])
train_df['category'] = cat
train_df['parent_category'] = parent_cat
train_df.head()

# train_df.created_at = train_df.created_at.apply(lambda x:
datetime.datetime.utcfromtimestamp(x))
# train_df.deadline = train_df.deadline.apply(lambda x:
datetime.datetime.utcfromtimestamp(x))
```

```
train_df.launched_at = train_df.launched_at.apply(lambda x:
datetime.datetime.utcfromtimestamp(x))
train_df.state_changed_at = train_df.state_changed_at.apply(lambda
x: datetime.datetime.utcfromtimestamp(x))
train_df.head(10)

# train_df['predicted_funding_time'] = train_df.deadline -
train_df.launched_at
train_df['funding_time'] = train_df.state_changed_at -
train_df.launched_at
train_df['year'] = train_df.launched_at.apply(lambda x: x.year)

train_df = train_df.drop(columns=['launched_at',
'state_changed_at'])
train_df

train_df.funding_time = train_df.funding_time.apply(
    lambda x:
datetime.datetime.utcfromtimestamp(x.total_seconds()).day)
train_df.rename(columns={'country_displayable_name': 'country'},
inplace=True)

train_df

train_df = train_df[['name', 'country', 'year', 'parent_category',
'category', 'spotlight', 'staff_pick',
                     'backers_count', 'funding_time', 'goal',
'pledged', 'state']]
train_df.to_csv('train_data.csv')

test_df = pd.DataFrame()
for dir in [d.path[2:] + os.sep for d in os.scandir() if 'test' in
d.path and not 'csv' in d.path]:
    for file in os.listdir(dir):
        temp = pd.read_csv(dir + file)
        test_df = pd.concat([test_df, temp])
test_df = test_df.drop_duplicates(subset='name')
test_df.head()

test_df = test_df.drop(['blurb', 'country', 'currency',
'currency_symbol', 'currency_trailing_code', 'current_currency',
                        'disable_communication', 'friends',
'fx_rate', 'id', 'is_backing', 'is_starrable', 'is_starred',
                        'permissions', 'photo', 'slug',
'source_url', 'static_usd_rate', 'urls', 'usd_pledged',
                        'usd_type', 'converted_pledged_amount',
'location', 'creator', 'profile', 'created_at', 'deadline'],
                       axis=1)
test_df = test_df[(test_df.state == 'successful') | (test_df.state
== 'failed')]
test_df = test_df.reset_index(drop=True)

parent_cat, cat = [], []
for i in test_df.category:
    item_json = json.loads(i)
    try:
```

```
            parent_cat.append(item_json['parent_name'])
        except KeyError:
            parent_cat.append(None)
        cat.append(item_json['name'])

test_df.drop(columns=['category'])
test_df['category'] = cat
test_df['parent_category'] = parent_cat
test_df.head()

test_df.launched_at = test_df.launched_at.apply(lambda x:
datetime.datetime.utcfromtimestamp(x))
test_df.state_changed_at = test_df.state_changed_at.apply(lambda x:
datetime.datetime.utcfromtimestamp(x))
test_df.head(10)

test_df['funding_time'] = test_df.state_changed_at -
test_df.launched_at
test_df['year'] = test_df.launched_at.apply(lambda x: x.year)

test_df = test_df.drop(columns=['launched_at', 'state_changed_at'])
test_df

test_df.funding_time = test_df.funding_time.apply(
    lambda x:
datetime.datetime.utcfromtimestamp(x.total_seconds()).day)
test_df.rename(columns={'country_displayable_name': 'country'},
inplace=True)

test_df

test_df = test_df[['name', 'country', 'year', 'parent_category',
'category', 'spotlight', 'staff_pick',
                   'backers_count', 'funding_time', 'goal',
'pledged', 'state']]

test_df = test_df[~test_df.name.isin(train_df.name)]
test_df

test_df = test_df.reset_index(drop=True)
test_df.to_csv('test_data.csv')
```

Очищення та перетворення даних

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sn

train_df = pd.read_csv('train_data.csv')
train_df = train_df.iloc[:, 1:]
train_df.head()
test_df = pd.read_csv('test_data.csv')
```

```
test_df = test_df.iloc[:, 1:]
test_df.head()

corrMatrix = train_df.corr()
print(corrMatrix)
sn.heatmap(corrMatrix, annot = True)
plt.show()

# y_train_state = train_df.state
# y_train_pledged = train_df.pledged
X_train = train_df.drop(columns=['name', 'parent_category',
'backers_count', 'spotlight'])

print(f'Maximum value in goal: {max(X_train.goal)}')
print(f'Minimum value in goal: {min(X_train.goal)}')

plt.hist(X_train.goal, range=(0, 100000), bins=50)
plt.title('Before outlier deletion')
plt.show()

X_train = X_train[X_train.goal > X_train.goal.quantile(0.5) -
        3*(X_train.goal.quantile(0.75) -
X_train.goal.quantile(0.25))]
X_train = X_train[X_train.goal < X_train.goal.quantile(0.5) +
        3*(X_train.goal.quantile(0.75) -
X_train.goal.quantile(0.25))]

print(f'Maximum value in goal after outlier deletion:
{max(X_train.goal)}')
print(f'Minimum value in goal after outlier deletion:
{min(X_train.goal)}')
plt.hist(X_train.goal, range=(0, 100000), bins=50)
plt.title('After outlier deletion')
plt.show()

X_train['goal_log'] = np.log2(X_train.goal).astype(int)
print(f'Maximum value in goal after logarithmic normalization:
{max(X_train.goal_log)}')
print(f'Minimum value in goal after logarithmic normalization:
{min(X_train.goal_log)}')
plt.hist(X_train.goal_log, range=(-10, 20), bins=50)
plt.title('After log transform')
plt.show()
a = 1
b = 100
x, y = min(X_train.goal), max(X_train.goal)
X_train['goal_minmax'] = (X_train.goal - x) / (y - x) * (b - a) + a
print(f'Maximum value in goal after minmax normalization:
{max(X_train.goal_minmax)}')
print(f'Minimum value in goal after minmax normalization:
{min(X_train.goal_minmax)}')
plt.hist(X_train.goal_minmax, range=(0, 100), bins=50)
plt.title('After minmax normalization')
plt.show()

X_train.goal = (np.sqrt(X_train.goal)+1).astype(int)
```

```
print(f'Maximum value in goal after squareroot normalization:
{max(X_train.goal)}')
print(f'Minimum value in goal after squareroot normalization:
{min(X_train.goal)}')
plt.hist(X_train.goal, range=(0, 250), bins=50)
plt.title('After squareroot normalization')
plt.show()

X_train.drop(columns=['goal_log', 'goal_minmax'], inplace=True)

print(f'Maximum value in pledged: {max(X_train.pledged)}')
print(f'Minimum value in pledged: {min(X_train.pledged)}')

plt.hist(X_train.pledged, range=(0, 10000), bins=50)
plt.title('Pledged before squareroot normalization')
plt.show()

pledged = (np.sqrt(X_train.pledged) + 1).astype(int)
print(f'Maximum value in pledged after squareroot normalization:
{max(pledged)}')
print(f'Minimum value in pledged after squareroot normalization:
{min(pledged)}')
plt.hist(pledged, range=(0, 250), bins=50)
plt.title('Pledged after squareroot normalization')
plt.show()

X_train.year = X_train.year - 2008
plt.hist(X_train.year, range=(0, 14), bins=50)
plt.title('Launch year after normalization')
plt.show()

print(f'Maximum value in funding time:
{max(X_train.funding_time)}')
print(f'Minimum value in funding time:
{min(X_train.funding_time)}')

plt.hist(X_train.funding_time, range=(0, 32), bins=40)
plt.title('Funding time (days)')
plt.show()

X_train['expectation_per_day'] = X_train.goal /
X_train.funding_time
plt.hist(X_train.expectation_per_day, range=(0, 20), bins=40)
plt.title('Expected money per day (USD)')
plt.show()

print(f'Minimum value in daily expectations:
{min(X_train.expectation_per_day)}')
print(f'Maximum value in daily expectations:
{max(X_train.expectation_per_day)}')

print(f'Number of data in daily expectations before outlier
elemination: {len(X_train.expectation_per_day)}')

X_train = X_train[X_train.expectation_per_day <
X_train.expectation_per_day.quantile(0.5) +
```

```
                    3*(X_train.expectation_per_day.quantile(0.75) -
X_train.expectation_per_day.quantile(0.25))]

print(f'Number of data in daily expectations after outlier
elemination: {len(X_train.expectation_per_day)}')

X_train.expectation_per_day = (X_train.expectation_per_day * 5 +
1).astype(int)

plt.hist(X_train.expectation_per_day, range=(0,45), bins=45)
plt.title('Expected money per day after outlier analysis and
normalization (USD)')
plt.show()

X_train.staff_pick.replace([False, True], [0, 1], inplace=True)
X_train.state.replace(['failed', 'successful'], [0, 1],
inplace=True)

X_train

X_train.category.fillna('None', inplace=True)

from sklearn.preprocessing import LabelEncoder

enc_country = LabelEncoder()
enc_category = LabelEncoder()
X_train.country = enc_country.fit_transform(X_train.country)
X_train.category = enc_category.fit_transform(X_train.category)

X_train

# corrMatrix = pd.concat([X_train, y_train_pledged, y_train_state],
axis=1).corr()
corrMatrix = X_train.corr()
print(corrMatrix)
sn.heatmap(corrMatrix, annot = True)
plt.show()

categorySuccess = X_train[['category', 'state']]
categorySuccess.category =
pd.Series(enc_category.inverse_transform(X_train.category))
categorySuccess = categorySuccess.groupby(['category']).mean()
categorySuccess.sort_values('state', ascending=False)

countrySuccess = X_train[['country', 'state']]
countrySuccess.country =
pd.Series(enc_country.inverse_transform(X_train.country))
countrySuccess = countrySuccess.groupby(['country']).mean()
countrySuccess.sort_values('state', ascending=False)

goalSuccess = X_train[['goal', 'state']]
goalSuccess = goalSuccess.groupby(['goal']).mean()
goalSuccess.plot()
plt.show()

launchYearSuccess = X_train[['year', 'state']]
```

```
launchYearSuccess.year += 2008
launchYearSuccess = launchYearSuccess.groupby(['year']).mean()
launchYearSuccess.plot()
plt.show()

activetimeSuccess = X_train[['funding_time', 'state']]
activetimeSuccess =
activetimeSuccess.groupby(['funding_time']).mean()
activetimeSuccess.plot()
plt.show()

expectationSuccess = X_train[['expectation_per_day', 'state']]
expectationSuccess =
expectationSuccess.groupby(['expectation_per_day']).mean()
expectationSuccess.plot()
plt.show()

y_train_pledged = X_train.pledged
y_train_state = X_train.state
X_train.drop(columns=['pledged', 'state'], inplace=True)

X_train.to_csv('X_train.csv')
y_train_state.to_csv('y_train_state.csv')
y_train_pledged.to_csv('y_train_pledged.csv')

X_test = test_df.drop(columns=['name', 'parent_category',
'backers_count', 'spotlight'])

X_test.goal = (np.sqrt(X_test.goal)+1).astype(int)
print(f'Maximum value in goal after squareroot normalization:
{max(X_test.goal)}')
print(f'Minimum value in goal after squareroot normalization:
{min(X_test.goal)}')
plt.hist(X_test.goal, range=(0, 250), bins=50)
plt.title('After squareroot normalization')
plt.show()

X_test.year = X_test.year - 2008
plt.hist(X_test.year, range=(0, 14), bins=50)
plt.title('Launch year after normalization')
plt.show()

X_test['expectation_per_day'] = X_test.goal / X_test.funding_time
plt.hist(X_test.expectation_per_day, range=(0, 20), bins=40)
plt.title('Expected money per day (USD)')
plt.show()

X_test.expectation_per_day = (X_test.expectation_per_day * 5 +
1).astype(int)

plt.hist(X_test.expectation_per_day, range=(0,45), bins=45)
plt.title('Expected money per day after normalization (USD)')
plt.show()
X_test.staff_pick.replace([False, True], [0, 1], inplace=True)
X_test.state.replace(['failed', 'successful'], [0, 1],
inplace=True)
```

```
X_test

X_test.country = enc_country.transform(X_test.country)
X_test.category = enc_category.transform(X_test.category)

X_test

y_test_pledged = X_test.pledged
y_test_state = X_test.state
X_test.drop(columns=['pledged', 'state'], inplace=True)

X_test.to_csv('X_test.csv')
y_test_state.to_csv('y_test_state.csv')
y_test_pledged.to_csv('y_test_pledged.csv')

pickle.dump(enc_country, open('country_enc.sav', 'wb'))
pickle.dump(enc_category, open('category_enc.sav', 'wb'))
```

Навчання та порівняння моделей

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neural_network import MLPClassifier
from sklearn.neighbors import KNeighborsClassifier
# from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier,
AdaBoostClassifier, GradientBoostingClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn import metrics

import warnings
warnings.filterwarnings('ignore')

names = ["Nearest Neighbors",
#          "Linear SVM",
         "Decision Tree", "Random Forest",
         'Gradient Boosting', "Neural Net", "AdaBoost", "Naive
Bayes"]
class_names = ['failed', 'successful']

X_train = pd.read_csv('X_train.csv')
y_train_state = pd.read_csv('y_train_state.csv')
y_train_pledged = pd.read_csv('y_train_pledged.csv')

X_train = X_train.iloc[:, 1:]
y_train_state = y_train_state.iloc[:, 1:]
y_train_pledged = y_train_pledged.iloc[:, 1:]

X_test = pd.read_csv('X_test.csv')
y_test_state = pd.read_csv('y_test_state.csv')
```

```
y_test_pledged = pd.read_csv('y_test_pledged.csv')

X_test = X_test.iloc[:, 1:]
y_test_state = y_test_state.iloc[:, 1:]
y_test_pledged = y_test_pledged.iloc[:, 1:]

classifiers = [
    KNeighborsClassifier(3),
#     SVC(kernel="linear", C=0.025),
    DecisionTreeClassifier(max_depth=5),
    RandomForestClassifier(max_depth=5, n_estimators=100),
    GradientBoostingClassifier(n_estimators=200),
    MLPClassifier(alpha=1, max_iter=1000),
    AdaBoostClassifier(),
    GaussianNB()]

for name, clf in zip(names, classifiers):
    clf.fit(X_train, y_train_state)
    y_pred = clf.predict(X_test)
    print(name)
    print(f'ROC-AUC score: {metrics.roc_auc_score(y_test_state,
y_pred)}')
    disp = metrics.plot_confusion_matrix(clf, X_test, y_test_state,
                                 display_labels=class_names,
                                 cmap=plt.cm.Blues)
    disp.ax_.set_title(f'Confusion matrix {name}')
    plt.show()
    print(disp.confusion_matrix)
    print(metrics.classification_report(y_test_state, y_pred,
target_names=['failed', 'successful']))

learning_rate = [0.1, 0.7, 1.2]
n_estimators = [200, 400, 700]

examples = [(i, j) for i in learning_rate for j in n_estimators]

def variate_params(examples, X_train, y_train, X_test, y_test):
    for lr, n_est in examples:
        clf = GradientBoostingClassifier(n_estimators=n_est,
learning_rate=lr)
        clf.fit(X_train, y_train)
        y_pred = clf.predict(X_test)
        print(f'Model with {n_est} n_estimators and {lr} learning
rate')
        print(f'Train dataset  score: {clf.score(X_train,
y_train)}')
        print(f'Test dataset  score: {clf.score(X_test, y_test)}')
        print(f'{metrics.classification_report(y_test,
y_pred)}\n\n')

variate_params(examples, X_train, y_train_state, X_test,
y_test_state)

import joblib
```

```
best_clas = GradientBoostingClassifier(n_estimators=700,
learning_rate=0.1)
best_clas.fit(X_train, y_train_state)
joblib.dump(best_clas, 'clas_model.das')

from sklearn import linear_model
# from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import AdaBoostRegressor, BaggingRegressor,
GradientBoostingRegressor, RandomForestRegressor
from sklearn.neural_network import MLPRegressor

names_reg = ['Ridge Regression', 'Lasso Regression', 'ElasticNet
Regression',
          'Automatic Relevance Determination Regression', 'Bayesian
Ridge Regression',
#           'Support Vector Regression',
          'Stochastic Gradient Descent Regression', 'k Neighbors
Regression', 'Decision Tree Regression',
          'AdaBoost Regression', 'Bagging Regression', 'Gradient
Boosting Regression', 'Random Forest Regression',
          'MLP Regression']

regressors = [
    linear_model.Ridge(alpha=.5),
    linear_model.Lasso(alpha=0.1),
    linear_model.ElasticNet(),
    linear_model.ARDRegression(),
    linear_model.BayesianRidge(),
#     SVR(),
    linear_model.SGDRegressor(),
    KNeighborsRegressor(),
    DecisionTreeRegressor(),
    AdaBoostRegressor(),
    BaggingRegressor(),
    GradientBoostingRegressor(),
    RandomForestRegressor(),
    MLPRegressor()
    ]

for name, reg in zip(names_reg, regressors):
    reg.fit(X_train, y_train_pledged)
    y_pred = reg.predict(X_test)
    print(name)
    print(f'R2 score: {metrics.r2_score(y_test_pledged, y_pred)}')
    print(f'MSE: {metrics.mean_squared_error(y_test_pledged,
y_pred)}')
    print(f'MAE: {metrics.mean_absolute_error(y_test_pledged,
y_pred)}\n')

def variate_params(examples, X_train, y_train, X_test, y_test):
    for lr, n_est in examples:
        reg = GradientBoostingRegressor(n_estimators=n_est,
learning_rate=lr)
        reg.fit(X_train, y_train)
```

```
        y_pred = reg.predict(X_test)
        print(f'Model with {n_est} n_estimators and {lr} learning
rate')
        print(f'Train dataset  score: {reg.score(X_train,
y_train)}')
        print(f'Test dataset  score: {reg.score(X_test, y_test)}')
        print(f'{metrics.r2_score(y_test, y_pred)}\n')

variate_params(examples, X_train, y_train_pledged, X_test,
y_test_pledged)

best_regr = GradientBoostingRegressor(n_estimators=700,
learning_rate=0.1)
best_regr.fit(X_train, y_train_pledged)
joblib.dump(best_regr, 'regr_model.das')
```

Основний веб-застосунок

app.py

```
from flask import Flask, render_template, url_for, redirect,
request
from static.predictor import predict_with_url
from static.forms import UrlForm
from config import Config


app = Flask(__name__)
app.config.from_object(Config)


@app.route('/', methods=['GET', 'POST'])
def main():
    form = UrlForm()
    if form.validate_on_submit():
        return redirect(url_for('analysis', url=form.url.data))
    return render_template('main.html', title='Main', form=form)


@app.route('/result', methods=['GET'])
def analysis():
    url = request.args.get('url', type=str)
    state, pledged = predict_with_url(url)
    return render_template('analysis.html', title='Analysis',
state=state, pledged=pledged)


if __name__ == '__main__':
    app.run(debug=True)
```

config.py

```
import os


class Config(object):
    SECRET_KEY = os.environ.get('SECRET_KEY') or 'you-will-never-
guess'
```

forms.py

```
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField
from wtforms.validators import ValidationError, DataRequired


class UrlForm(FlaskForm):
    url = StringField('Url', validators=[DataRequired()],
render_kw={"placeholder": "   Project Url"})
    submit = SubmitField('Analyze')
```

predictor.py

```
from selenium import webdriver
import joblib
import json
from datetime import datetime
import numpy as np
import pandas as pd


def predict_with_url(url: str):
    clf_model = joblib.load('model/clas_model.das')
    reg_model = joblib.load('model/regr_model.das')

    driver = webdriver.Chrome()
    driver.get(url)
    project = driver.find_element_by_css_selector('#react-project-
header')
    project = json.loads(project.get_attribute('data-initial'))
    driver.quit()

    category = project['project']['category']['name']
    staff_pick = project['project']['isProjectWeLove']
    location = project['project']['location']['displayableName']
    country = location.replace(' ', '').split(',')[1]
    if country == 'UK':
```

```
        country = 'the United Kingdom'
    elif country.upper() == country:
        country = 'the United States'

    goal = int(float(project['project']['goal']['amount']))
    year =
datetime.fromtimestamp(project['project']['timeline']['edges'][0]['
node']['timestamp']).year
    funding_time = project['project']['duration']

    test_df = pd.DataFrame({'country': country, 'year': year,
'category': category, 'staff_pick': staff_pick,
                            'funding_time': funding_time, 'goal':
goal}, index=[0])

    enc_country = joblib.load('model/country_enc.sav')
    enc_category = joblib.load('model/category_enc.sav')

    test_df.goal = (np.sqrt(test_df.goal) + 1).astype(int)
    test_df.year = test_df.year - 2008
    test_df['expectation_per_day'] = test_df.goal /
test_df.funding_time
    test_df.expectation_per_day = (test_df.expectation_per_day * 5
+ 1).astype(int)
    test_df.staff_pick.replace([False, True], [0, 1], inplace=True)
    test_df.country = enc_country.transform(test_df.country)
    test_df.category = enc_category.transform(test_df.category)

    state = 'successful' if clf_model.predict(test_df)[0] == 1 else
'failed'
    pledged = np.round(reg_model.predict(test_df)[0], 2)
    return state, pledged
```

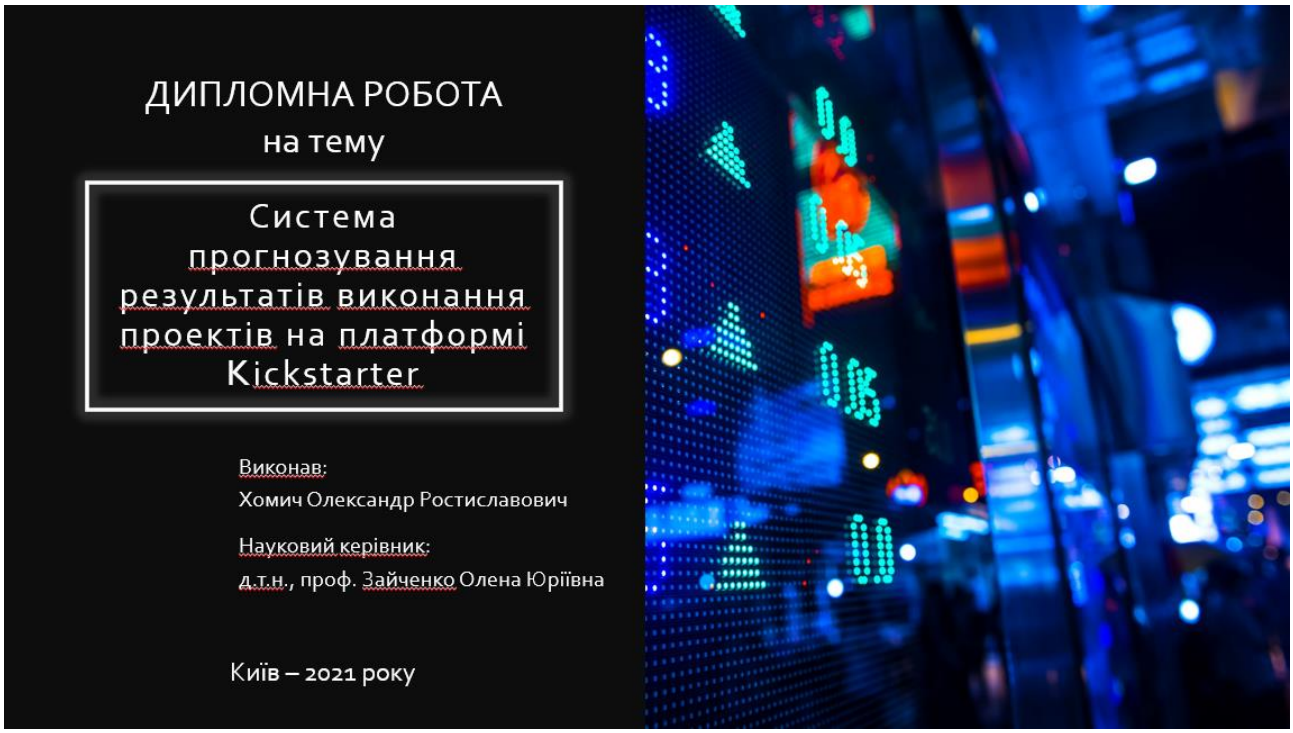
analysis.html

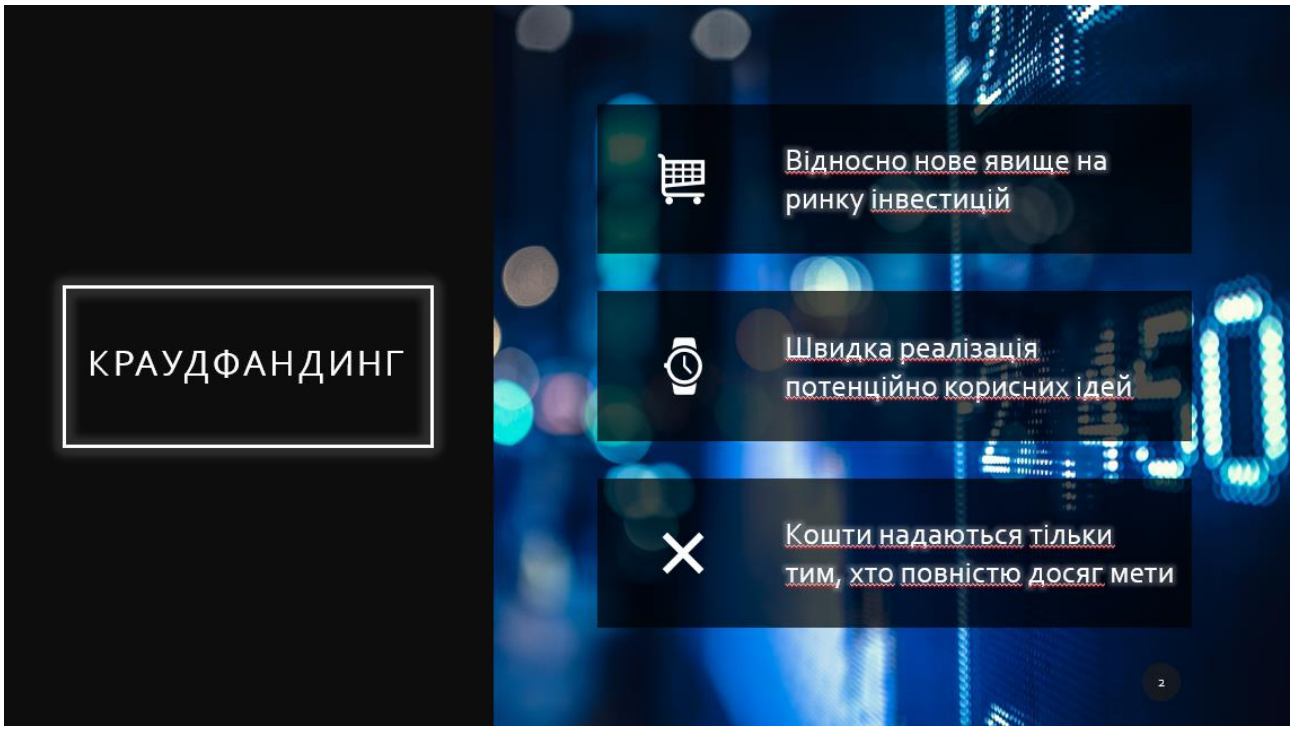
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{{ title }}</title>
</head>
<body>
<p>Predicted result: {{ state }}</p>
<p>Predicted pledged amount: {{ pledged }}</p>
</body>
</html>
```

main.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{{ title }}</title>
</head>
<body>
<form action="" method="post" novalidate>
    {{ form.hidden_tag() }}
    <div class="url">
        {{ form.url(size=32) }}
    </div>
    <p>{{ form.submit() }}</p>
</form>
</body>
</html>
```

ДОДАТОК В ІЛЮСТРАТИВНИЙ МАТЕРІАЛ





АКТУАЛЬНІСТЬ РОБОТИ

В умовах стійкого зростання та диверсифікації краудфандингу його соціально-економічний потенціал стає все більш очевидним.
Від допомоги людям у збиранні грошей на покриття медичних витрат та витрат на поховання до надання капіталу на ранніх стадіях новаторам, які допомагають реалізувати творчі ідеї, краудфандинг продовжує доводити свою цінність.

3

Мета роботи

- Розробка програмного забезпечення для прогнозу результатів виконання проекту на платформі Kickstarter

Об’єкт дослідження

- Застосування математичних методів прогнозування в фінансово-інвестиційній сфері

Предмет дослідження

- Методи машинного навчання та програмні засоби їх реалізації

4

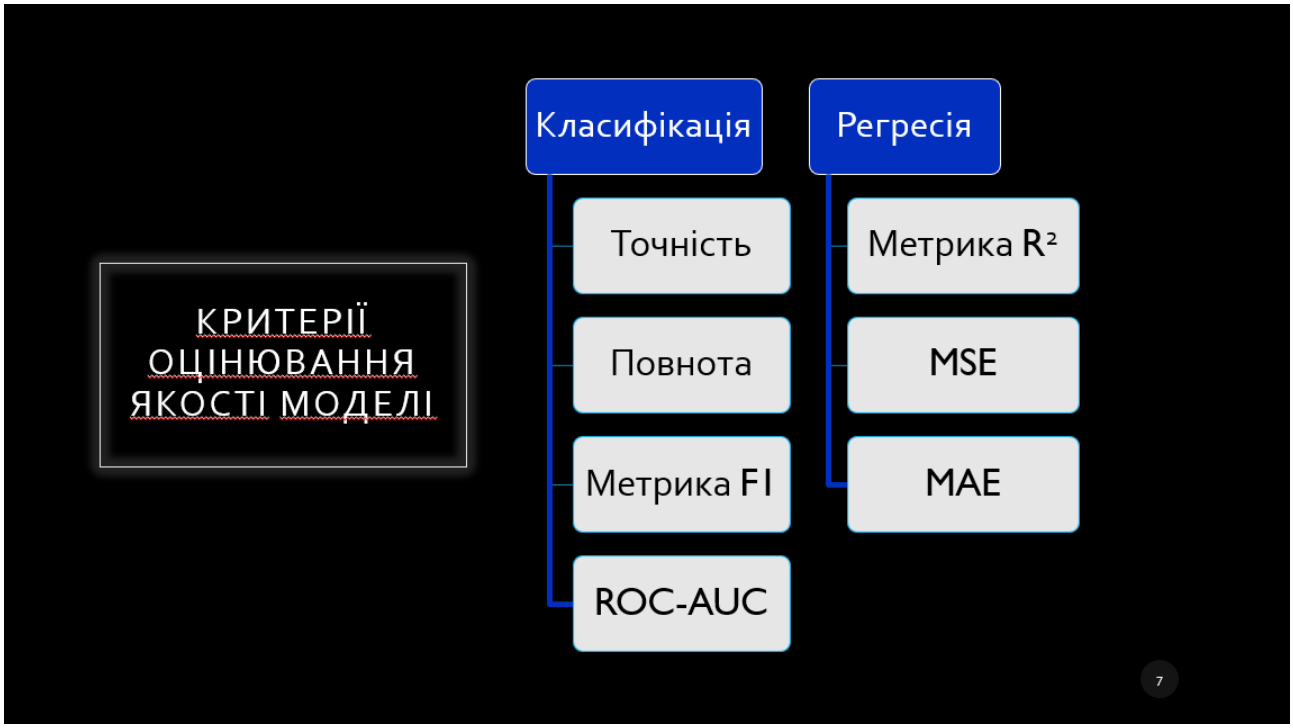
ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

5

ОСОБЛИВІСТЬ МОДЕЛЮВАННЯ ДАНИХ КРАУДФАНДИНГУ

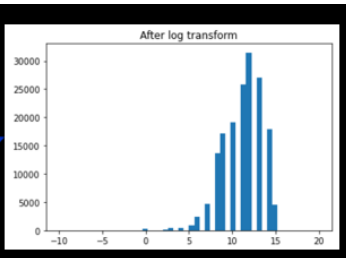
Моделювання даних краудфандингу створює нову проблему з точки зору включення проектів, де ми знаємо дату успіху, та проектів, де ми маємо лише часткову інформацію про те, що вони не досягли успіху до певної дати цілі проекту. Такі проекти називаються цензурованими. Використовуючи особливі моделі регресії, функція вірогідності будується з використанням часткової інформації.

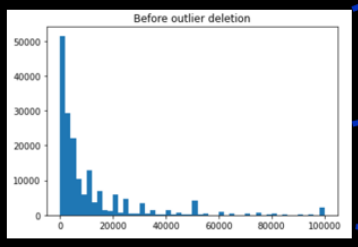
6

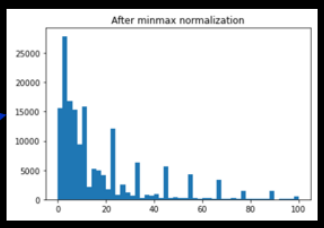

КРИТЕРІЇ ОЦІНЮВАННЯ ЯКОСТІ МОДЕЛІ
Класифікація
Точність
Повнота
Метрика F1
ROC-AUC
Регресія
Метрика R^2
MSE
MAE
7

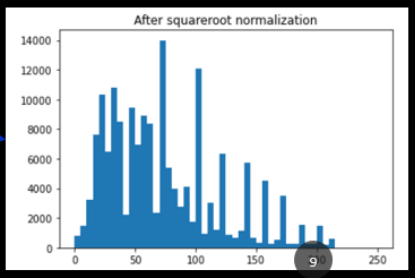

ОПИС ОСНОВНИХ ОЗНАК ДАНИХ
кількість інвесторів
категорія
країна
мета (в грошовому еквіваленті)
час запуску кампанії
об'єм зібраних коштів
стан (успішний чи ні)
8

НОРМАЛІЗАЦІЯ ДАНИХ

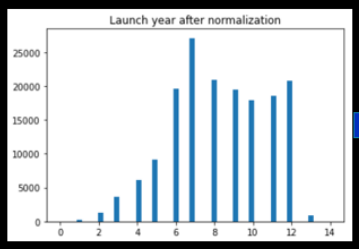


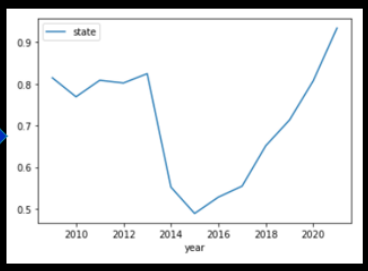


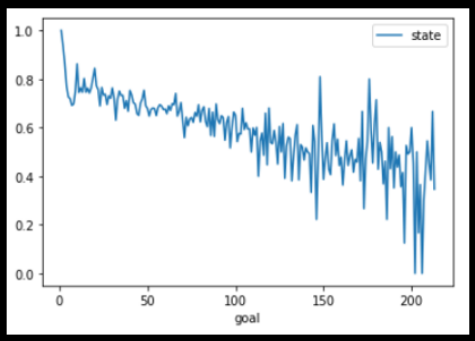


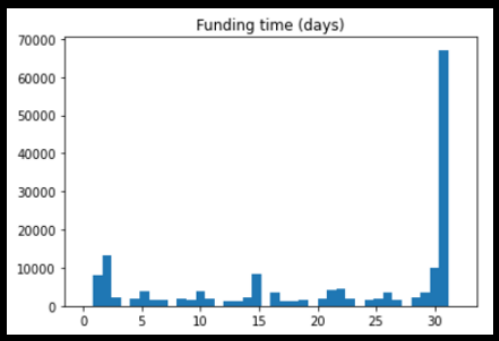


АНАЛІЗ ОТРИМАНИХ ДАНИХ

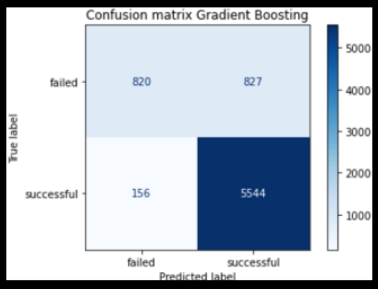








МОДЕЛЬ КЛАСИФІКАЦІЇ



Модель GradientBoostingClassifier

ROC-AUC на тестовій вибірці 89%

F1 score для класу:

- **Успіху 93%**
- **Невдачі 72%**

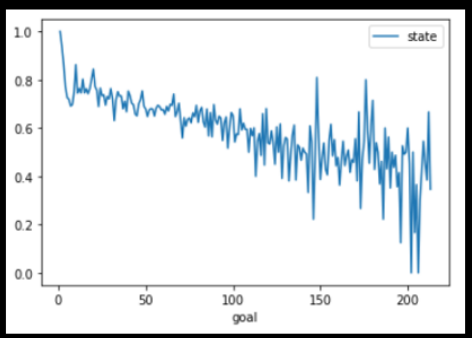
11

МОДЕЛЬ РЕГРЕСІЇ

Модель GradientBoostingRegressor

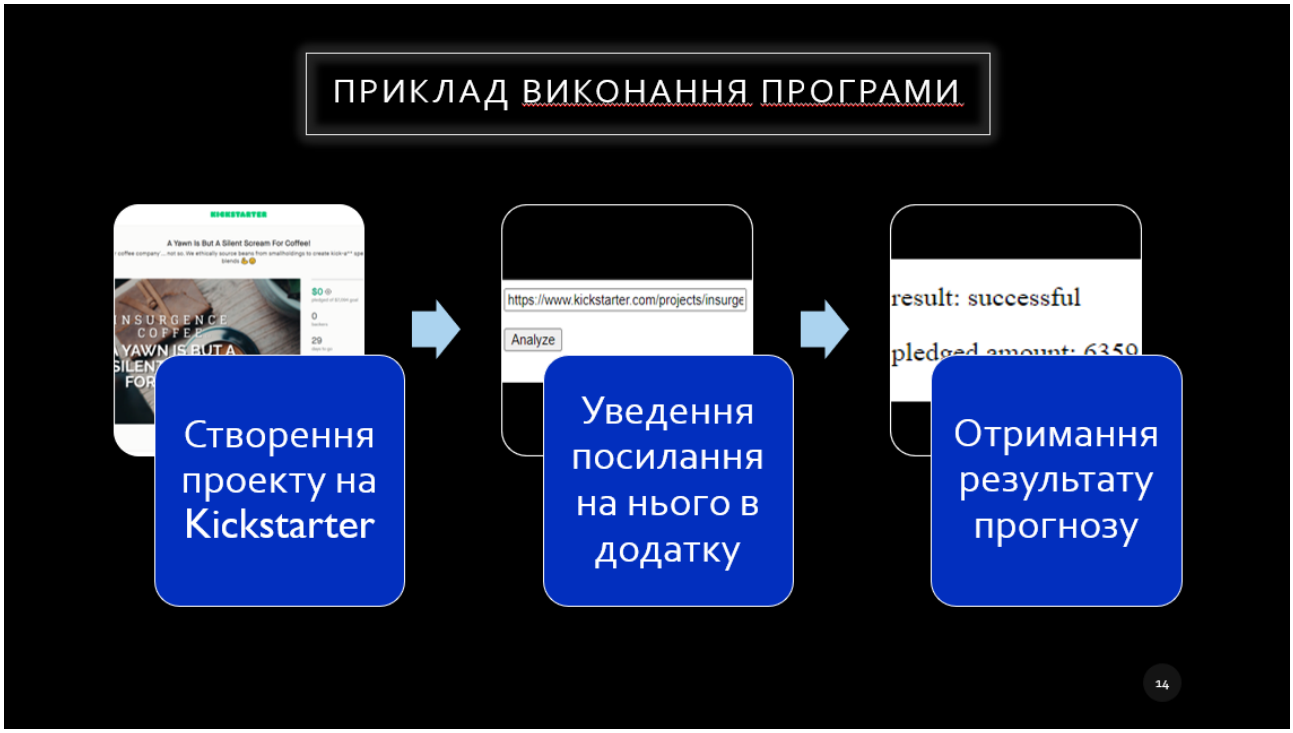
R^2 на тестовій вибірці 0,035

R^2 на тренувальній вибірці 0,27



12


ПРОГРАМНИЙ ПРОДУКТ
Початок роботи
На головній сторінці наявні дві форми для заповнення даних
Користувач вводить посилання на кампанію з сайту Kickstarter
Користувач вводить усі необхідні дані для аналізу: рік, мету, категорію, тощо
Проводиться скрапінг потрібної сторінки проекту та збираються усі необхідні дані: рік, локація, категорія тощо
На основі отриманих даних прогнозні моделі виконують передбачення
Кінець роботи
На сторінці результату користувач бачить передбачення
13


ПРИКЛАД ВИКОНАННЯ ПРОГРАМИ
Створення проекту на Kickstarter
https://www.kickstarter.com/projects/insurge
Analyze
Уведення посилання на нього в додатку
result: successful
Отримання результату прогнозу
14

ПРАКТИЧНА ЗНАЧУЩІСТЬ РЕЗУЛЬТАТІВ РОБОТИ

Розроблений веб-застосунок дозволить усім зрозуміти доцільність запровадження кампанії по залученню інвестицій на краудфандинговому сайті Kickstarter, а аналіз та дослідження предметної області дозволять зацікавленим сторонам прогнозувати поведінку ринку та підвищувати інвестиційну привабливість власних проектів.

15

ВИСНОВКИ

- Розроблено модель прогнозування успіху проекту з точністю 89%
- Досліджено основні залежності між ознаками проектів
- Розроблено веб-застосунок для зручності прогнозування

16

РЕКОМЕНДАЦІЇ ЩОДО ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

- Виокремити більш специфічні дані, наприклад, для конкретної категорії
- Категоризація змінних для використання у навчанні моделей
- Використання інших або модифікованих методів машинного навчання
- Розробка кращого графічного інтерфейсу користувача

17

ДЯКУЮ ЗА УВАГУ!