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

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

«На правах рукопису»
УДК 004.9:004.021

До захисту допущено:

В. о. завідувача кафедри

_____ Олександр РОЛІК

«___»_____2021 р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою «Інформаційні управляючі системи та технології»
зі спеціальності 126 «Інформаційні системи та технології»
на тему: «Інформаційна система прогнозування періоду затримки тварини в притулку**»**

Виконав:
студент VI курсу, групи ІС-з01мп
Марченков Іван Денисович _____

Керівник:
доцент кафедри ІСТ
Жураковська Оксана Сергіївна _____

Рецензент:
доцент кафедри ІПІ
Лісовиченко Олег Іванович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2021 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ
В. о. завідувача кафедри
_____ Олександр РОЛІК
«___»_____2021 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Марченкова Івана Денисовича

1. Тема дисертації «Інформаційна система прогнозування періоду затримки тварини в притулку», науковий керівник дисертації Жураковська Оксана Сергіївна, доцент кафедри ІСТ, затверджені наказом по університету від «25» 10 2021 р. № 3575-с

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження *процес оцінювання інтервалу перебування тварин в притулку*

4. Вихідні дані *Прогноз періоду перебування тварини в притулку та рекомендації щодо покращення параметрів тварини для зменшення цього періоду*

5. Перелік завдань, які потрібно розробити *виконати порівняльний аналіз методів побудови класифікаторів періоду затримки; розробити алгоритми обробки текстових даних для використання текстових описів в навчальній виборці; на основі методів комп'ютерного зору розробити алгоритми для оцінювання ступеня привабливості улюбленців за зображеннями; Розробити алгоритм надання рекомендацій за результатами вирішення задачі класифікації.*

6. Орієнтовний перелік графічного (ілюстративного) матеріалу *діаграма прецедентів, діаграма вимог, IDEF0, IDEF0 – декомпозована, діграма послідовностей, архітектура Inception-v3, ER-діаграма, схема БД.*

7. Орієнтовний перелік публікацій *Марченков І. Д. Прогнозування періоду затримки тварини в притулку з використанням методів NLP / Іван Денисович Марченков. //*

Norwegian Journal of development of the International Science. – 2021. – №73. – С. 65–69.

9. Дата видачі завдання 1 вересня 2021 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз та формування вимог задачі	09.09.2021	
2	Порівняльний аналіз існуючих методів та алгоритмів	21.09.2021	
3	Опис предметної області та побудова математичної моделі задачі	28.09.2021.	
4	Побудова та оцінка ефективності моделей класифікації зображень та обробки тексту	09.10.2021	
5	Розробка структури БД	16.10.2021	
6	Реалізація програмного забезпечення	24.10.2021	
7	Побудова графічних матеріалів	12.11.2021	
8	Оформлення документації	19.11.2021	
9	Подання роботи на попередній захист	23.11.2021	
10	Подання роботи на основний захист		

Студент Іван МАРЧЕНКОВ

Науковий керівник Оксана ЖУРАКОВСЬКА

АНОТАЦІЯ

Магістерська дисертація: 166 с., 61 рис., 46 табл., 50 джерел, 2 додатки.

Актуальність теми. Щодня в усьому світі мільйони бездомних тварин страждають на вулицях або піддаються евтаназії в притулках для тварин. Кількість котів і собак без домівок з часом тільки збільшується. Тварина, опинившись на вулиці, відчуває величезні страждання через те, що вона не в змозі подбати про себе. Якщо тваринам вдасться знайти домівки, можна буде врятувати багато дорогоцінних життів і створити більше щасливих сімей.

Вирішення таких проблем, як утримання безхатніх тварин, їх годування, лікування є фінансово затратними. Зменшення фінансових витрат притулками є актуальною задачею.

Малайзійська компанія «PetFinder», яка є платформою для захисту тварин, що містить базу даних понад 150 тис. тварин. Дана інформація може виявитися корисною для визначення терміну затримки тварини у притулку.

Ефективне прогнозування періоду затримки надасть можливість притулкам оптимально розподіляти ресурси, щоб поліпшити загальну продуктивність усиновлення, та як наслідок, зменшити витрати на притулок та виховання.

Метою роботи є підвищення ефективності роботи притулку для тварин та розподілу ресурсів на утримання тварин в притулку за рахунок прогнозування періоду перебування тварин в притулку.

Для досягнення мети поставлено та вирішено такі задачі:

Виконати порівняльний аналіз методів побудови класифікаторів періоду затримки.

Розробити алгоритми обробки текстових даних для використання текстових описів в навчальній виборці.

На основі методів комп’ютерного зору розробити алгоритми для оцінювання ступеня привабливості улюбленців за зображеннями (фото).

Побудувати моделі для вирішення задач класифікації за зображенням та за текстовим описом, підібрати параметри побудованих моделей: кількість шарів

нейронної мережі, кількість епох навчання, розмірність зображень, яка подається на вхід тощо.

Розробити алгоритм надання рекомендацій за результатами вирішення задачі класифікації.

Об’єктом дослідження є процес оцінювання інтервалу перебування тварин в притулку.

Предметом дослідження є задача класифікації за зображенням та за текстовим описом.

Наукова новизна одержаних результатів полягає у тому, що для вирішення задачі класифікації застосовано комбінацію методів класифікації за зображенням та за текстовим описом.

Практичне значення отриманих результатів.

Ефективне прогнозування періоду перебування надасть можливість притулкам оптимально розподіляти ресурси, щоб поліпшити загальну продуктивність усиновлення, та як наслідок, зменшити витрати на притулок та виховання.

Публікації. Марченков І. Д. Прогнозування періоду затримки тварини в притулку з використанням методів NLP / Іван Денисович Марченков. // Norwegian Journal of development of the International Science. – 2021. – №73. – С. 65–69.

КЛЮЧОВІ СЛОВА: ПРИВЛАСНЕННЯ ДОМАШНІХ ТВАРИН, ОБРОБКА ПРИРОДНОЇ МОВИ, КОМП’ЮТЕРНИЙ ЗІР, ВИПАДКОВИЙ ЛІС

ABSTRACT

Master's dissertation consists 166 pages, 61 images, 46 tables, 50 referring sources, 2 appendices.

Topicality. Every day, millions of stray animals around the world suffer on the streets or are euthanized in animal shelters. The number of cats and dogs without homes is only increasing over time. Once on the street, the animal suffers greatly because it is unable to take care of itself. If animals can find homes, many precious lives can be saved and more happy families created.

Solving such problems as keeping stray animals, feeding them, and treating them is financially costly. Reducing the financial costs of shelters is an urgent task.

Malaysian company PetFinder, which is an animal protection platform with a database of more than 150,000 animals. This information may be useful in determining the period of detention of the animal in the shelter.

Effective forecasting of the delay period will allow shelters to allocate resources optimally to improve overall adoption productivity and, as a result, reduce shelter and upbringing costs.

The purpose of the dissertation research is to increase the efficiency of the animal shelter and the allocation of resources for keeping animals in the shelter by forecasting the period of stay of animals in the shelter.

To achieve this goal, the following tasks were set and solved:

Perform a comparative analysis of methods for constructing classifiers of the delay period.

Develop algorithms for processing text data for the use of text descriptions in the training sample.

Based on computer vision methods, develop algorithms to assess the degree of attractiveness of pets by images (photo).

Build models to solve the problems of classification by image and text description, choose the parameters of the built models: the number of layers of the neural network, the number of epochs of learning, the dimensionality of the images supplied to the input, and so on.

Develop an algorithm for providing recommendations based on the results of solving the classification problem.

The object of the study is the process of estimating the interval of stay of animals in the shelter.

The subject of the study is the problem of classification by image and textual description.

The scientific novelty of the obtained results is that a combination of classification methods by image and text description is used to solve the problem of classification.

The practical value of the obtained results.

Effective stay forecasting will allow shelters to allocate resources optimally to improve overall adoption productivity and, as a result, reduce shelter and upbringing costs.

Publications. Ivan Marchenkov. Animal delay period prediction in the shelter using NLP methods. Norwegian Journal of development of the International Science: 73, pages 65-69, 2021.

KEY WORDS: PET ADOPTION, NATURAL LANGUAGE PROCESSING, COMPUTER VISION, RANDOM FOREST

ЗМІСТ

ВСТУП

Актуальність теми. Щодня в усьому світі мільйони бездомних тварин страждають на вулицях або піддаються евтаназії в притулках для тварин [1]. Кількість котів і собак без домівок з часом тільки збільшується. Тварина, опинившись на вулиці, відчуває величезні страждання через те, що вона не в змозі подбати про себе. Якщо тваринам вдасться знайти домівки, можна буде врятувати багато дорогоцінних життів і створити більше щасливих сімей.

Вирішення таких проблем, як утримання безхатніх тварин, їх годування, лікування є фінансово затратними. Зменшення фінансових витрат притулками є актуальною задачею.

Малайзійська компанія «PetFinder», яка є платформою для захисту тварин, що містить базу даних понад 150 тис. тварин. Дана інформація може виявитися корисною для визначення терміну затримки тварини у притулку.

Ефективне прогнозування періоду затримки надасть можливість притулкам оптимально розподіляти ресурси, щоб поліпшити загальну продуктивність усиновлення, та як наслідок, зменшити витрати на притулок та виховання.

Метою роботи є підвищення ефективності роботи притулку для тварин та розподілу ресурсів на утримання тварин в притулку за рахунок прогнозування періоду перебування тварин в притулку.

Для досягнення мети поставлено та вирішено такі задачі:

Виконати порівняльний аналіз методів побудови класифікаторів періоду затримки.

Розробити алгоритми обробки текстових даних для використання текстових описів в навчальній виборці.

На основі методів комп’ютерного зору розробити алгоритми для оцінювання ступеня привабливості улюбленців за зображеннями (фото).

Побудувати моделі для вирішення задач класифікації за зображенням та за текстовим описом, підібрати параметри побудованих моделей: кількість шарів

нейронної мережі, кількість епох навчання, розмірність зображень, яка подається на вхід тощо.

Розробити алгоритм надання рекомендацій за результатами вирішення задачі класифікації.

Об’єктом дослідження є процес оцінювання інтервалу перебування тварин в притулку.

Предметом дослідження є задача класифікації за зображенням та за текстовим описом.

Наукова новизна одержаних результатів полягає у тому, що для вирішення задачі класифікації застосовано комбінацію методів класифікації за зображенням та за текстовим описом.

Практичне значення отриманих результатів.

Розроблене рішення за допомогою методів машинного навчання, обробки природньої мови, комп’ютерного зору, а також нейронних мереж будує модель, вилучаючи корисну інформацію з текстового опису тварини та її фотокарток.

Інформаційна система у вигляді боту має зручний інтерфейс для взаємодії користувача із нею.

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Word2Vec та GloVe

На відміну від завдання загальної класифікації, п’ять класів періоду затримки тварини в нашому наборі даних мають також порядкову властивість. Френк і Холл стверджують, що стандартний підхід може не врахувати властивість впорядкування інформації, тому пропонують використати дерева рішень [2]. Ми дослідимо методи дерева рішень, а також методи глибокого навчання. В оригінальній задачі Kaggle метрикою оцінки є квадратично зважена каппа, яка вимірює узгодження між двома рейтингами, і враховує вартість помилки, як у [3]. Беручи до уваги нелінійні часові інтервали різних класів, ми вибираємо точність класифікації як нашу оціночну метрику. Word2Vec [4] та GloVe [5] - це дві моделі вбудовування слів, які обчислюють векторні уявлення слів. Мета - вставити семантично подібні слова в подібний векторний простір. Word2Vec-це метод на основі прогнозування, навчається на великому наборі даних з досить низькими обчислювальними вимогами.

Він перевершує багато традиційних методів. GloVe, з іншого боку, є методом підрахунку. У роботі [5] Пеннінгтон та інші стверджують, що обидва методи поділяють фундаментальні ідеї та охоплюють подібну сементичну інформацію, але GloVe є більш ефективним у виконанні завдання. Таким чином, ми використовуємо GloVe як модель вкладення для отримання векторів вкладення. Нещодавня робота, проведена Мірсамаді та іншими [6] щодо розпізнавання мовленнєвих емоцій, свідчить про те, що використання простої схеми локальної уваги допомагає визначити емоційно помітні слова. На основі цієї схеми уваги вони пропонують новий метод, який дає змогу моделі надати більше ваги на вивчених словах, щоб досягти кращої точності класифікації.

1.2 Emotional intelligence

Підходи до емоційного інтелекту (англ. emotional intelligence) не мають за мету чітко змоделювати хід думок, навіть якщо підходи намагаються відтворити реакцію людини або тварини. Дослідники надали пріоритет, оцінювати емоції людини під час заняття справою, коли людина акцентує свою увагу на подразнику, та джерела впливу на емоційне реагування є очевидними. У публікації Натана Енсменгера [7] гру в шахи розглянуто, як експериментальну технологію, що може ефективно використовуватися для отримання достовірних знань про інші, більш складні системи. Зокрема, гра може використовуватись для визначення емоцій та реакції гравців, в залежності від їх поточного результату у партії.

1.3 Computer vision

Надалі визначимо виявлення емоційного стимулу, привабливості, як проблеми комп'ютерного зору (англ. computer vision). Люди проявляють позитивну емоційну реакцію, коли бачать привабливі речі, включаючи дітей, тварин та неживі предмети, що мають інфантильні риси [8]. Це реагування є досить специфічним для людського мозку та може бути виявлене за допомогою фМРТ (функціональної магнітно – резонансної томографії) [9]; та це реагування значно впливає на нашу поведінку, розпізнавання якої може бути корисно застосоване як для маркетингу продуктів, так і для збереження життя тварин [10]. Привабливість забезпечує цікавий випадок низькорівневої інстинктивної реакції на подразники, що може бути змодельована без побудови складної штучної сенсорно-моторної системи, але є досить зрозумілою, для того, щоб наші емоції були миттєво розпізнані.

Були проаналізовані дві публікації пов'язані з темою магістерської дисертації, зокрема щодо тематики виявлення привабливості. Ю. Бао та співавтори користуються методом опорних векторів (англ. Support vector machine, SVM), для визначення, чи є фотографії собак, котів чи кроликів привабливими [11]. Вони використовують складний процес вилучення ознак, що включає подання вектором Фішера характеристик SIFT, і розглядають виявлення привабливості як проблему

класифікації. SVM працює добре, класифікуючи зображення з точністю 84%. На жаль у роботі не вказано, яким саме чином було розмічено набір даних.

К. Ван та співавтори побудували набір даних людських обличь з позначеними оцінками привабливості, після чого застосували класичні функції комп'ютерного зору (фільтр Ґабора, локальні бінарні шаблони (англ. Local Binary Patterns, LBP) та гістограму напрямлених градієнтів (англ. histogram of oriented gradients, HOG)), а також використовували метод SVM для прогнозування показників привабливості як проблеми регресії, а не класифікації [12]. Вони визначили ground truth, попросивши сорок учасників ранжувати невеликі підмножини набору даних, та дати оцінку привабливості за шкалою 0-10. Схоже на те, що такий підхід розмітки набору даних працює добре, в результаті чого абсолютна похибка оцінки за десятибальною шкалою становить 1,27.

У спорідненій задачі, К. Ю та співавтори використовують згорткові нейронні мережі (англ. convolutional neural network, CNN) для класифікації зображень за емоціями [13]. Вони зібрали великий набір фотографій та заплатили працівникам Amazon Mechanical Turk (AMT), щоб вони розмітили зображення по восьми категоріям: радість, гнів, трепет, задоволення огида, хвилювання, страх і смуток. Задачу класифікації восьми класів було успішно вирішена, досягнувши точності 58%. Була використана CNN типу AlexNet.

Висновки до розділу

Word2Vec та GloVe є моделями вбудовування слів, які обчислюють векторні уявлення слів. Мета - вставити семантично подібні слова в подібний векторний простір. Проаналізовано дві публікації пов’язані з темою магістерської дисертації, зокрема щодо тематики виявлення привабливості. Ю. Бао та співавтори користуються методом опорних векторів, для визначення, чи є фотографії собак, котів чи кроликів привабливими.

2 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ

2.1 Варіанти використання

Актором інформаційної системи є користувач – працівник притулку. В таблиці 2.1 наведено варіанти використання.

Таблиця 2.1 – Типи залежностей між варіантами використання

Актор	Варіант використання	Опис дії варіанта використання
Користувач	Введення вхідних даних	Користувач вводить дані параметрів тварини до системи: текст та зображення
	Запит на введення інформації	Система послідовно надає запити користувачу щодо введення множини параметрів тварини
	Підтвердження введеної інформації	Після вводу усієї необхідної вхідної інформації система надає можливість перевірити введену інформацію та підтвердити її правильність
	Збереження вхідних даних до бази	Після вводу інформації користувачам система зберігає її до БД
	Отримання результату	Користувач отримує результат класифікації – період перебування тварини у притулку та рекомендації щодо покращення параметрів тварини для зменшення періоду перебування у притулку

Актор	Варіант використання	Опис дії варіанта використання
	Прогноз періоду перебування тварини у притулку	Користувач отримує прогноз щодо періоду перебування тварини у притулку на основі її параметрів
	Препроцесінг даних	Система підготовлює дані для прогнозу
	Препроцесінг текстової інформації	Система трансформує текстову інформацію методами NLP
	Препроцесінг зображень	Система трансформує зображення у масив чисел
	Класифікація зображень	Система класифікує зображення за рівнем привабливості
	Рекомендація щодо покращення параметрів	Користувач отримує рекомендацію щодо покращення параметрів тварини, що збільшить шанси привласнення із притулку

Відповідно визначених варіантів використання побудуємо загальну модель варіантів використання, яка наведена на графіку.

2.2 Функціональні вимоги

Відповідно визначених варіантів використання виявлено функціональні вимоги, результат для загальної частини наведено у таблиці 2.2.

Таблиця 2.2 – Функціональні вимоги

Актор	Варіант використання	Функціональна вимога
Користувач	Введення вхідних даних	1. Система надає можливість ввести дані – параметри тварини

Актор	Варіант використання	Функціональна вимога
	Запит на введення інформації	1.1. Система послідовно надсилає запити користувачу щодо введення множини параметрів тварини
	Підтвердження введеної інформації	1.2. Система надає можливість користувачу перевірити правильність введеної інформації та підтвердити у випадку достовірності
	Збереження вхідних даних до бази	1.2.1. Система повинна зберігати дані до БД для подальшого аналізу
	Отримання результату	2. Система повинна надавати результат класифікації та рекомендації щодо покращення параметрів тварини
	Прогноз періоду перебування тварини у притулку	2.1. Система повинна прогнозувати період перебування тварини на основі вхідних даних
	Обробка даних	2.1.1. Система повинна здійснити обробку даних у необхідний для класифікації за допомогою моделі формат
	Препроцесінг даних	2.1.1.1. Система повинна вилучити потрібні ознаки з даних різного формату (зображення, текст) для прогнозу
	Препроцесінг текстової інформації	2.1.1.1.1. Система повинна трансформувати текстову

Актор	Варіант використання	Функціональна вимога
		інформацію у числовий вигляд методами NLP
	Препроцесінг зображень	2.1.1.1.2. Система повинна трансформувати зображення у числовий вигляд
	Рекомендація щодо покращення параметрів	2.1.2. Система повинна надавати користувачу можливі рекомендації щодо покращення параметрів тварини
	Класифікація зображень	3. Система повинна класифікувати зображення за рівнем привабливості

Відповідно виявлених вимог побудуємо загальну модель вимог та представимо її на графіку.

Висновки до розділу

ІС має ряд варіантів використання: введення вхідних даних, запит на введення інформації, підтвердження введеної інформації, збереження вхідних даних до бази, прогноз періоду перебування тварини у притулку, препроцесінг даних, класифікація зображень та рекомендацію щодо покращення параметрів. Відповідно визначених варіантів використання виявлено функціональні вимоги.

3 МОДЕЛІ ТА МЕТОДИ ВИРІШЕННЯ ЗАДАЧІ

Для розуміння логіки побудови математичної моделі задачі, яку я розглядаю, спочатку проаналізую структуру набору даних, який використовуватиметься для дослідження проблеми прогнозування усиновлення домашніх тварин із притулку.

Набір даних розміщено на провідній платформі захисту тварин Малайзії PetFinder та у соціальній мережі спеціалістів по обробці даних та машинному навчанню (англ. Machine Learning) Kaggle. Набір складається з трьох частин. Перший – це файл CSV із детальною інформацією, що включає тип тварини (собака чи кіт), породу, стать, забарвлення, довжину хутра, стерилізацію, стан здоров’я та опис. Другий – це файли описів JSON з оцінкою привабливості, а третій – велика колекція файлів зображень. Загалом набір складається з 14993 записів домашніх тварин та 23 характеристик по кожній із них.

Для подальшої роботи із даними видалимо ті характеристики тварин, що вказані у файлах, які очевидно не впливають на цільову змінну, що потребує передбаченню: PetID – унікальний ідентифікатор домашньої тварини, RescuerID - унікальний ідентифікатор рятівника.

Файли зображень відповідають кожному запису тварини, але кожне зображення може містити більше одного вихованця. Це створює додаткову задачу ідентифікації конкретного домашнього улюбленця, що відповідає запису. Цільова змінна, коефіцієнт усиновлення, складаються з п’яти категорій:

0 – усиновлення в той же день;

1 – від 1 до 7 днів (1-й тиждень);

2 – від 8 до 30 днів;

3 – між 31 і 90 днями (2-й і 3-й місяці);

4 – Відсутність усиновлення через 100 днів після внесення до списку.

У наборі даних відсутні домашні улюбленці, які чекали усиновлення від 90 до 100 днів. Тривалість діапазону у кожному класі неоднакова. Наприклад, клас 0 включає 1-денний інтервал часу, тоді як клас 3 включає 60-денний інтервал часу.

Дана задача є нетривіально та потребує навчання на великому масиві інформації. Набір даних містить цільові змінні, тобто змінні які потрібно передбачити на основі побудованої моделі, у якій аргументами є незалежні змінні.

Задача належить до виду машинного навчання із вчителем (англ. supervised learning) та є задачею класифікації (classification problem). Після створення на вхід до моделі подаватимуться аргументи, для яких значення ground truth не визначено. У штучному інтелекті (англ. AI, Artificial Intelligence) і машинному навчанні завданням класифікації є поділ безлічі спостережень (об'єктів) на групи, звані класами, на основі аналізу їх формального опису. При класифікації кожна одиниця спостереження відноситься певної групи або номінальної категорії на основі деякої якісної властивості. Сформуємо математичну модель задачі класифікації.

Нехай X – множина характеристик об’єкту, Y – кінцева множина номерів класів. Існує невідома цільова залежність y^*: $X \to Y$, значення якої відомі на об’єктах навчальної вибірки $X^m = (x_1, y_1), \dots, (x_m, y_m)$. Треба побудувати алгоритм a: $X \to Y$, здатний класифікувати довільний об’єкт $x \in X$.

Розглянемо методи машинного навчання, які застосовуються для вирішення проблеми класифікації.

3.1 Логістична Регресія (Logistic Regression)

Логістична регресія – це статистичний метод, розроблений для вирішення проблем бінарної класифікації. Це досягається шляхом проходження вхідних даних через лінійну функцію, а потім перетворення вихідних даних у значення ймовірності за допомогою сигмоїдної функції.

Математичний вигляд логістичної регресії наведеено у формулах (3.1), (3.2).

$$h_\theta(x) = P(Y = 1|x; \theta), \tag{3.1}$$

$$Z = \theta^T x. \tag{3.2}$$

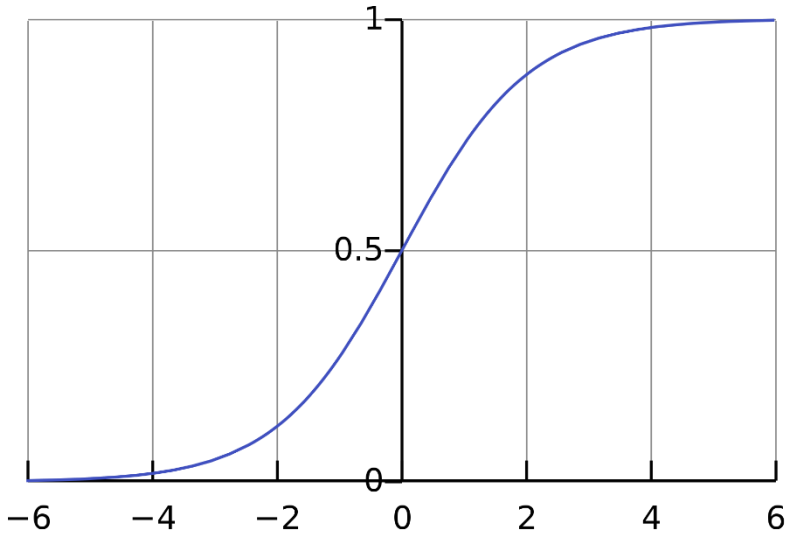

Рисунок 3.1 – Сигмоїдна функція

Без сигмоїдної функції логістична регресія була б просто лінійною регресією. Це означає, що вихід моделі може варіюватися від $-\infty$ до $+\infty$.

Для бінарної класифікації вихідним результатом має бути значення ймовірності. В такому випадку стає в нагоді сигмоїдна функція. Вона стискає вихідні дані лінійної функції x між 0 і 1. Усі вхідні значення, що перевищують 0, дають вихідне значення більше 0,5. Усі вхідні значення, менші за 0, дають вихідне значення менше ніж 0,5.

Математично вигляд сигмоїдної функції наведено у формулі (3.3).

$$f(x) = \frac{1}{1+e^{-x}}. \tag{3.3}$$

Межа рішення

Щоб отримати дискретне значення класу (0 або 1), необхідно визначити межу рішення. Межа рішення визначає, наскільки високою має бути ймовірність отримання вихідного значення - 1.

Як правило, межа рішення обирається 0,5, так що при вихідному значенні більшому або рівному 0,5 отримується клас 1, інакше клас 0.

Градієнтний спуск та функція витрат

Практично кожен алгоритм машинного навчання має в його основі алгоритм оптимізації. Градієнтний спуск – це алгоритм оптимізації, який використовується для пошуку значень параметрів (коефіцієнтів) функції (f), що мінімізує функцію витрат ($J(w)$) [14]. Найпоширенішою є функція витрат середньої квадратичної помилки (Mean-Squared Error cost function). Приклад функції витрат для лінійної функції на рисунку 3.1. Математичний вигляд функції витрат середньої квадратичної помилки наведено у формулі (3.4).

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta\left(x^{(i)}\right) - y^{(i)})^2, \tag{3.4}$$

де $h_\theta\left(x^{(i)}\right)$ – передбачуване значення і-го спостереження;

$y^{(i)}$ – істинне значення і-го спостереження.

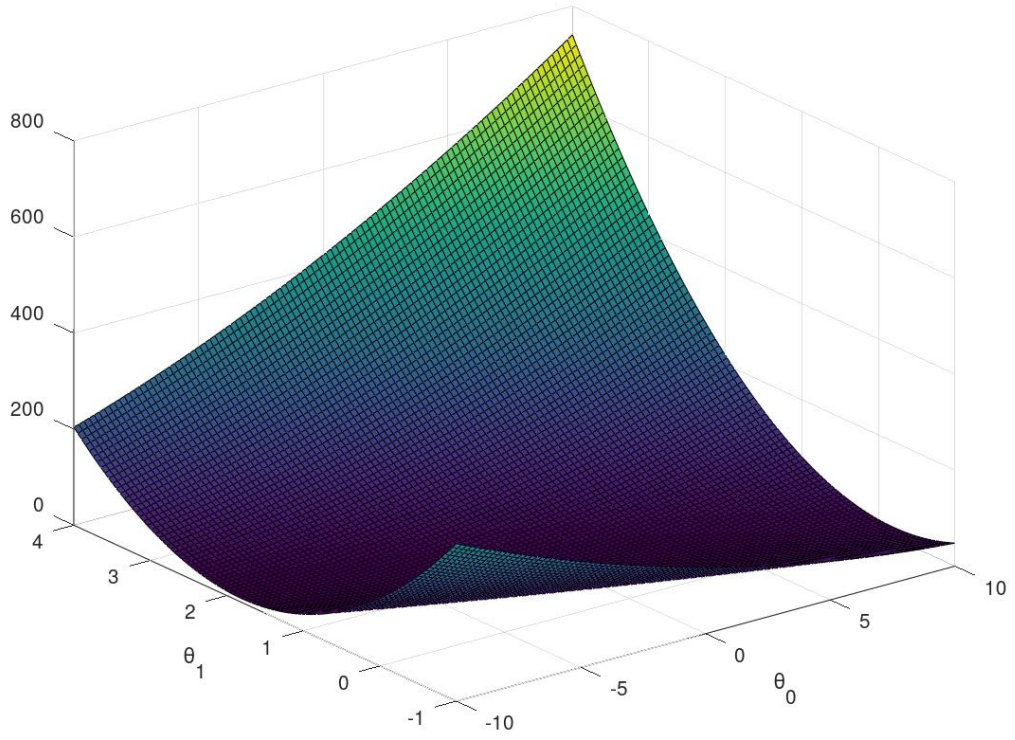

Рисунок 3.2 – Функція витрат $J(\theta_0, \theta_1)$ побудована на основі двох вагів.

Похідна функції витрат відносно будь-якої ваги. Формула (3.5) демонструє розрахунок градієнта для лінійної регресії [15]:

$$\frac{\partial}{\partial\theta_j}J(\theta) = \frac{1}{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}. \quad (3.5)$$

Похідна відображає нахил функції в заданій точці. Нам потрібно знати нахил, щоб дізнатися напрямок (знак) для переміщення значень вагів θ_j, щоб отримати нижчу вартість на наступну ітерацію.

Для пошуку оптимальних значень вагів необхідно повторити оновлення вагів до збіжності:

Повторювати до збіжності {

$$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta)$$

},

де α – коефіцієнт швидкості навчання.

Процес ітеративного обчислення значень вагів та функції вартості зображено на рисунку 1.2.

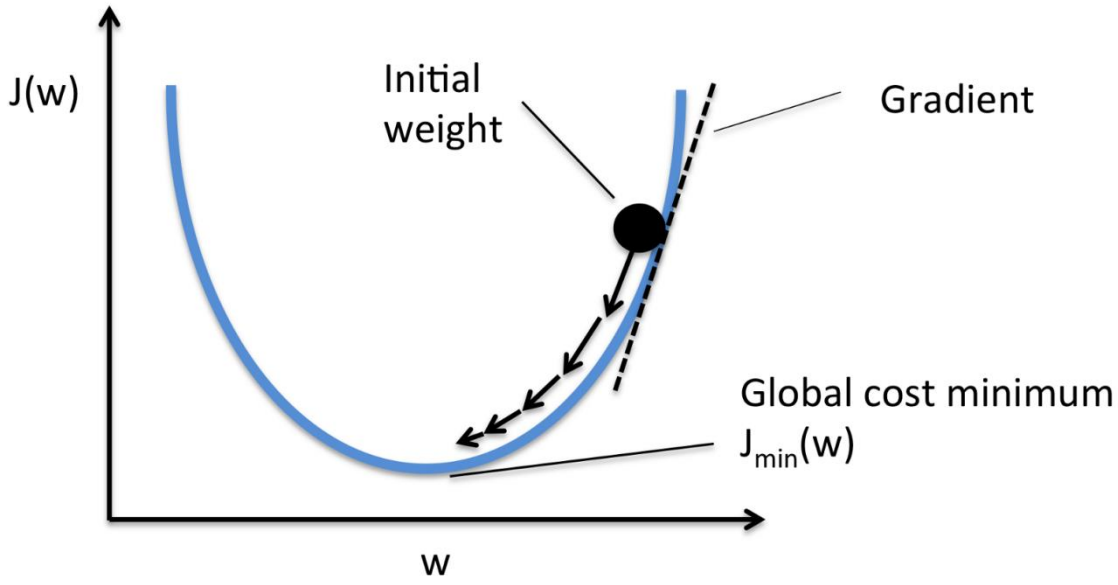

Рисунок 3.3 – Досягнення мінімуму функції витрат

Стохастичний градієнтний спуск

Стохастичний градієнтний спуск [16] (Stochastic Gradient Descent - SGD) - це варіація алгоритму градієнтного спуску, яка обчислює функцію витрат та оновлює модель для кожного прикладу в навчальному наборі даних.

Переваги:

- Часті оновлення негайно дають зрозуміти працездатність моделі та швидкість вдосконалення.
- Збільшення частоти оновлення моделі може призвести до більш швидкого вивчення деяких проблем.
- Шумний процес оновлення може дозволити моделі уникати локальних мінімумів.

Недоліки:

- Оновлення моделі настільки часто є обчислювально дорожчим, ніж інші конфігурації градієнтного спуску, що займає значно більше часу для навчання моделей на великих наборах даних.
- Часті оновлення можуть спричинити шумний градієнтний сигнал, через що функція витрат моделі стрибне (матиме більшу дисперсію в порівнянні з навчальними епохами).
- Шумний процес навчання вниз по градієнту помилок також може ускладнити алгоритм встановлення мінімуму помилок для моделі.

Алгоритм

1) Випадково упорядкувати навчальний набір даних
2) Повторювати {

for $i := \underline{1, m}$ {

$$\theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

(для кожного $j = \underline{1, n}$) }

}

Результати вагів кожного прикладу можна обчислюються в окремих потоках.

Міні-пакетний градієнтний спуск

Міні-пакетний градієнтний спуск - це варіація алгоритму градієнтного спуску, яка розбиває навчальний набір даних на невеликі партії, які використовуються для обчислення функції витрат моделі та оновлення вагів моделі.

Реалізації можуть вибрати підсумок градієнта по міні-партії, що зменшує дисперсію градієнта.

Міні-пакетний градієнтний спуск прагне знайти баланс між стійкістю стохастичного градієнтного спуску та ефективністю пакетного градієнтного спуску. Це найпоширеніша реалізація градієнтного спуску, що використовується в галузі глибокого навчання.

Переваги:

- Частота оновлення моделі вище, ніж пакетного градієнтного спуску, що дозволяє досягти більш міцної збіжності, уникаючи локальних мінімумів.
- Пакетні оновлення забезпечують обчислювально ефективніший процес, ніж стохастичний градієнтний спуск.

Недоліки:

- Вимагає конфігурації додаткового параметра «міні-партії» для алгоритму навчання.
- Інформація про функції витрат повинна накопичуватися в міні-партіях навчальних прикладів.

Алгоритм

1) Повторювати {

for $i := 1, b+1, b+2, \dots, m-b+1$ {

$$\theta_j := \theta_j - \alpha \frac{1}{b} \sum_{k=i}^{i+b-1} (h_\theta(x^{(k)}) - y^{(k)}) x_j^{(k)}$$

(для кожного $j = \underline{1, n}$) }

},

де b – розмір пакету.

Багатокласова логістична регресія може бути застосована для класифікації 5 змінних. Ймовірнісний розподіл відповіді по заданим вхідним параметрам наведено у формулах (3.6) – (3.8).

$$P(Y_i = c|X;\beta) = \frac{e^{\beta_c * x_i}}{\sum_{k=1}^{K} e^{\beta_k * x_i}}, \tag{3.6}$$

де Y_i – елемент із множини класів, X – матриця ознак, що описують спостереження, β – набір коефіцієнтів регресії, c – клас, ймовірність належності до якого треба визначити, $e^{\beta_c * x_i}$ – оцінка віднесення спостереження i до класу c, $\sum_{k=1}^{K} e^{\beta_k * x_i}$ – сума оцінок віднесення спостереження i до кожного класу k із множини K.

$$h_\theta(x) \geq 0.5 \rightarrow y = 1, \tag{3.7}$$

$$h_\theta(x) < 0.5 \rightarrow y = 0. \tag{3.8}$$

Багатокласова логістична регресія є розширенням бінарного випадку. Модель передбачає ймовірність належності об’єкту до кожного із класів.

Множина класів $y \in \{0, 1\}$ розширюється таким чином, що $y \in \{0, 1, .. n\}$. Базовий варіант багатокласової логістичної регресії – запуск бінарної класифікації кілька разів, по одному для кожного класу.

Алгоритм:

1. Розділити задачу на n задач бінарної класифікації.
2. Для кожного класу i:
   - спрогнозувати ймовірність по ознаках для цього окремого класу.
   - $y_{pred}^{(i)} = max(P(Y_i = c|X;\beta))$,

     де $y_{pred}^{(i)}$ – прогнозоване значення класу для поточної задачі

3. Для кожної підзадачі обирається один клас (ТАК), а всі інші об’єднуються в другий клас (НІ). Тоді береться клас із найвищим прогнозованим значенням.

Функція Softmax

Функція softmax (softargmax або нормалізована експоненціальна функція) — це функція, яка приймає на вхід вектор із K дійсних чисел і нормалізує його у розподіл ймовірностей, що складається з K ймовірностей, пропорційних експонентам вхідних чисел. Тобто перед застосуванням softmax деякі компоненти вектора можуть бути від’ємними або більшими за одиницю; і можуть не складати 1; але після застосування softmax кожен компонент буде в інтервалі [0 , 1], щоб їх можна було інтерпретувати як ймовірності. Стандартна (одинична) функція softmax визначається формулою (3.9).

$$\sigma(z_i) = \frac{e^{z_{\{i)}}}{\sum_{j=1}^{K} e^{z_{(j)}}}, \tag{3.9}$$

де $i = 1,.,.,.,K;\ z = z_1,.,.,.,z_K$.

Іншими словами: застосовується стандартна експоненційна функція до кожного елемента z_i вхідного вектора z і ці значення нормалізуються шляхом ділення на суму всіх цих показників; ця нормалізація гарантує, що сума компонентів вихідного вектора $\sigma(z)$ дорівнює 1.

3.2 Наївний Байєс (Naive Bayes)

Метод базується на теоремі Байєса із передбаченням класу на основі незалежних предикторів. Наївну Байєсову модель легко побудувати та вона особливо корисна для великих масивів даних. Відомо, що Наївний Байєсовий метод не тільки простий, але й у деяких випадках перевершує найскладніші методи класифікації [14]. Теорема Байєса представляє спосіб обчислення апостеріорної ймовірності $P(c|x)$ з $P(c)$, $P(x)$ та $P(x|c)$. Рівняння теореми Байєса:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}, \tag{3.10}$$

де $P(c|x)$ – апостеріорна ймовірність класу c для предиктора x, $P(c)$ – апріорна ймовірність класу c, $P(x|c)$ – умовна ймовірність предиктора x при класі c, $P(x)$ – апріорна ймовірність предиктора.

Теорема Баєса зазначає наступне співвідношення між змінною класів y та залежними ознаками $x_1, x_2 \ldots, x_n$:

$$P(y|x_1, \ldots, x_n) = \frac{P(y)P(x_1,\ldots,x_n|y)}{P(x_1,\ldots,x_n)}. \tag{3.11}$$

Використовуючи наївне припущення умовної незалежності, що

$$P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y), \tag{3.12}$$

для всіх i співвідношення спрощено до:

$$P(y|x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1,\ldots,x_n)}. \tag{3.13}$$

Оскільки $P(y|x_1, \ldots, x_n)$ є константою з урахуванням вхідних даних, ми можемо використовувати таке правило класифікації:

$$P(y|x_1, \ldots, x_n) = P(y)\prod_{i=1}^{n} P(x_i|y), \tag{3.14}$$

$$\hat{y} = arg\max_{y} P(y)\prod_{i=1}^{n} P(x_i|y). \tag{3.15}$$

і ми можемо використовувати максимальну апостеріорну оцінку (Maximum A Posteriori, MAP), щоб оцінити $P(y)$ та $P(x_i|y)$.

Різні наївні баєсові класифікатори відрізняються головним чином за припущенням, які вони роблять стосовно розподілу $P(x_i|y)$.

Незважаючи на свої, на перший погляд, надто спрощені припущення, наївні байєсівські класифікатори досить добре працюють в багатьох реальних ситуаціях, як відомо, класифікація документів і фільтрація спаму. Вони вимагають невеликої кількості навчальних даних для оцінки необхідних параметрів.

Наївні байєсівські класифікатори можуть бути надзвичайно швидкими в порівнянні з більш складними методами. Відокремлення розподілів умовних ознак класу означає, що кожен розподіл може бути незалежно оцінений як одновимірний розподіл. Це, у свою чергу, допомагає спростити проблеми, пов'язані з розмірністю.

Гаусівський наївний баєсів класифікатор

Співвідношення гаусівського наївного баєсовогу класифікатора:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}}. \quad (3.16)$$

3.3 Метод Опорних Векторів (SVM, Support Vector Machines)

SVM – це алгоритм керованого машинного навчання, який допомагає в проблемах класифікації або регресії. Він спрямований на пошук оптимальної межі між можливими виходами. Простіше кажучи, SVM виконує складні перетворення даних залежно від вибраної функції ядра і на основі цих перетворень намагається максимізувати межі поділу між точками даних залежно від міток або класів, які визначено.

Основна ідея методу – переведення вихідних векторів у простір більш високої розмірності і пошуку роздільної гіперплощини з максимальним зазором у цьому просторі (рис. 3.4). Дві паралельні гіперплощини будуються по обидва боки гіперплощини, що розділяє класи. Роздільною гіперплощиною буде гіперплощина, що максимізує відстань до двох паралельних гіперплощин. Алгоритм працює у

припущенні, що чим більша різниця або відстань між цими паралельними гіперплощинами, тим менше буде середня помилка класифікатора.

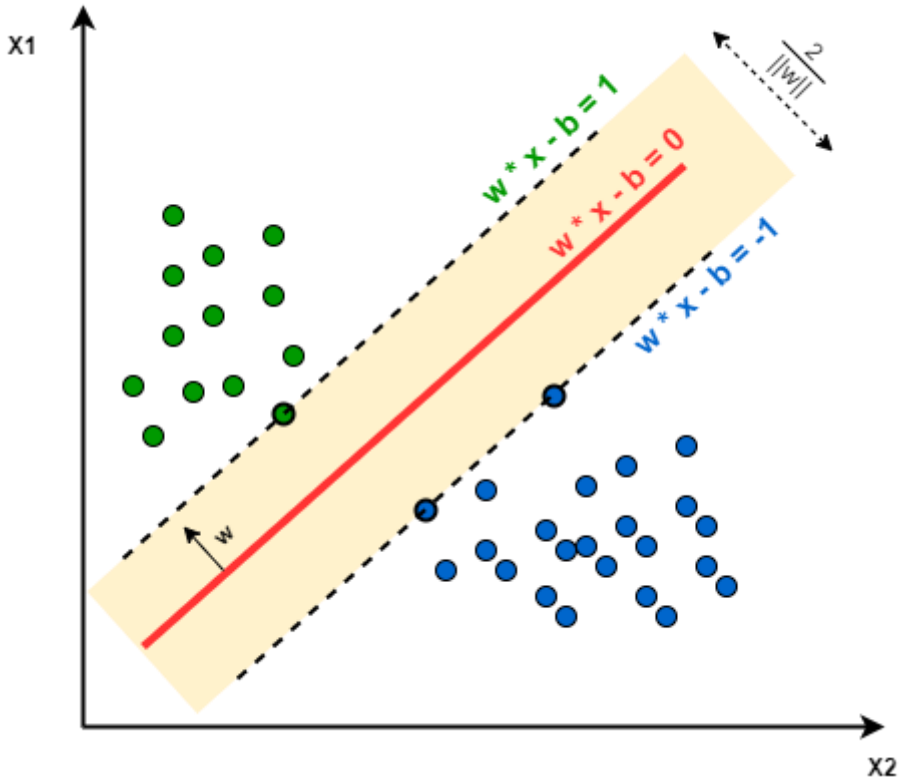

Рисунок 3.4 – Оптимальна роздільна лінія між відокремленими зразками даних

Багатокласова класифікація за допомогою SVM

У своєму найпростішому типі SVM не підтримує багатокласову класифікацію. Він підтримує бінарну класифікацію та поділ точок даних на два класи. Для багатокласової класифікації той самий принцип використовується після розбиття задачі мультикласифікації на кілька задач бінарної класифікації.

Ідея полягає в тому, щоб відобразити точки даних у високовимірний простір, щоб отримати взаємне лінійне розділення між кожними двома класами. Це називається підходом «один до одного» (One-to-One approach), який розбиває багатокласову проблему на кілька проблем бінарної класифікації. Бінарна класифікація для кожної пари класів.

Інший підхід, який можна використати, — це один до решти (One-to-Rest). У цьому підході розбивка встановлюється на бінарний класифікатор для кожного класу.

Одиничний SVM виконує бінарну класифікацію і здатний розрізняти два класи. Таким чином, згідно з двома підходами розбивки, класифікувати точки даних із m наборів даних класів:

– У підході One-to-Rest класифікатор може використовувати m SVM. Кожен SVM передбачить належність до одного з m класів.

– У підході One-to-One класифікатор може використовувати $\frac{(m-1)m}{2}$ SVM.

Розглянемо приклад задачі класифікації із 3 класів: зелений, червоний і синій (рис. 3.5):

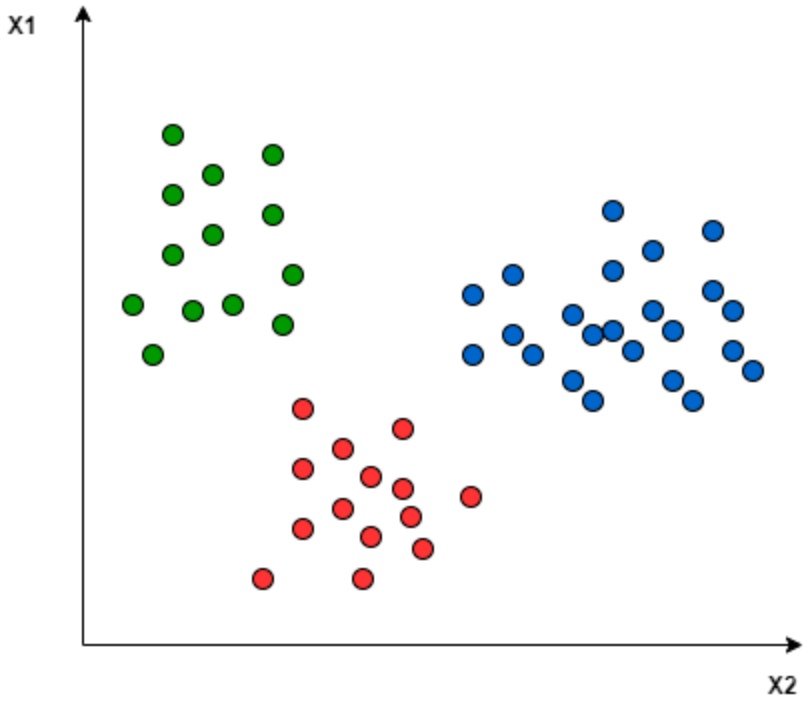

Рисунок 3.5 – Набір даних із 3 класів

Застосування двох підходів до цього набору даних призводить до наступного:

– У підході One-to-One потрібна гіперплощина для розділення між кожними двома класами, нехтуючи точками третього класу. Це означає, що поділ враховує лише точки двох класів у поточному розщепленні. Наприклад, червоно-синя лінія намагається максимізувати поділ лише

між синьою та червоною точками. Це не має відношення до зелених точок (рис. 3.6).

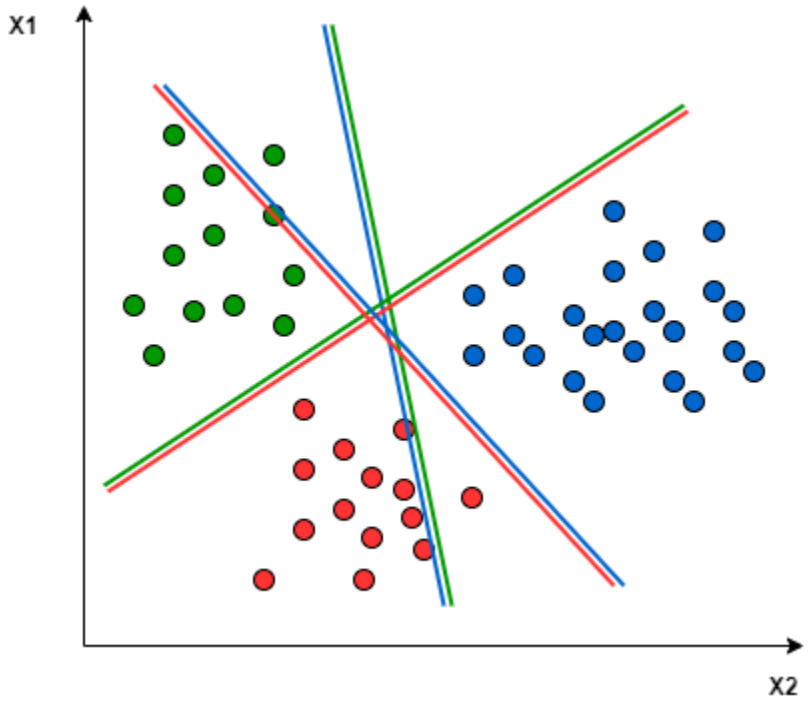

Рисунок 3.6 – Гіперплощини розділу класів у підході One-to-One

– У підході One-to-Rest потрібна гіперплощина, щоб відокремити клас від усіх інших одночасно. Це означає, що поділ враховує всі точки, розділяючи їх на дві групи; група точок одного класу і група для всіх інших точок. Наприклад, зелена лінія намагається максимізувати поділ між зеленими точками та всіма іншими точками одночасно (рис. 3.7).

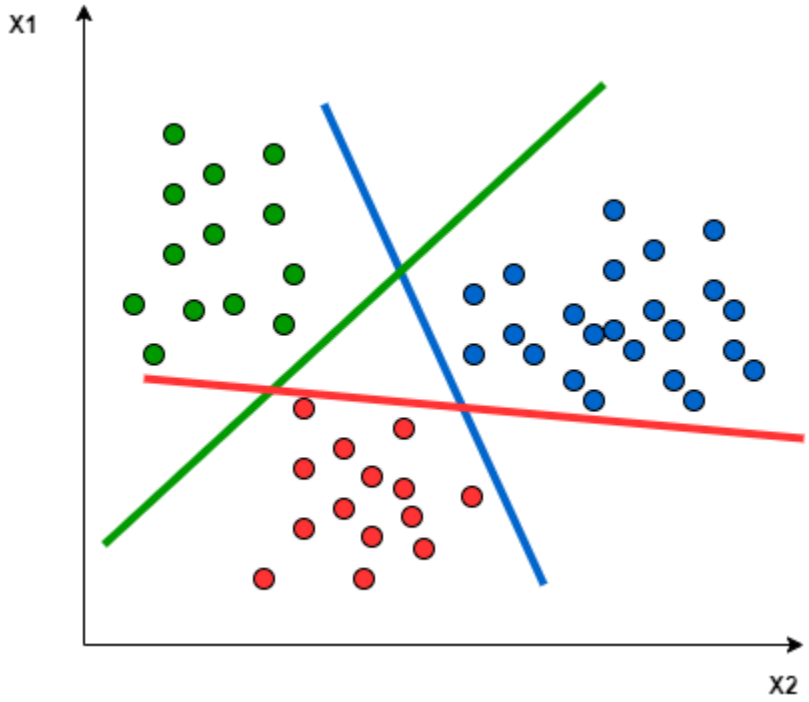

Рисунок 3.7 – Гіперплощини розділу класів у підході One-to-Rest

Однією з найпоширеніших проблем реальних для багатокласової класифікації за допомогою SVM є класифікація тексту. Наприклад, класифікація статей новин, твітів чи наукових статей.

3.4 Дерева Рішень (Decision Trees)

Дерево рішень, або метод дерева класифікації, на кожному кроці обирає змінну, яка є оптимальною для поділу, а потім знаходить точку поділу. З точки зору оптимального поділу, ми інтерпретуємо це як поділ, який найкраще зменшує похибку та покращує чистоту листкових вузлів [17].

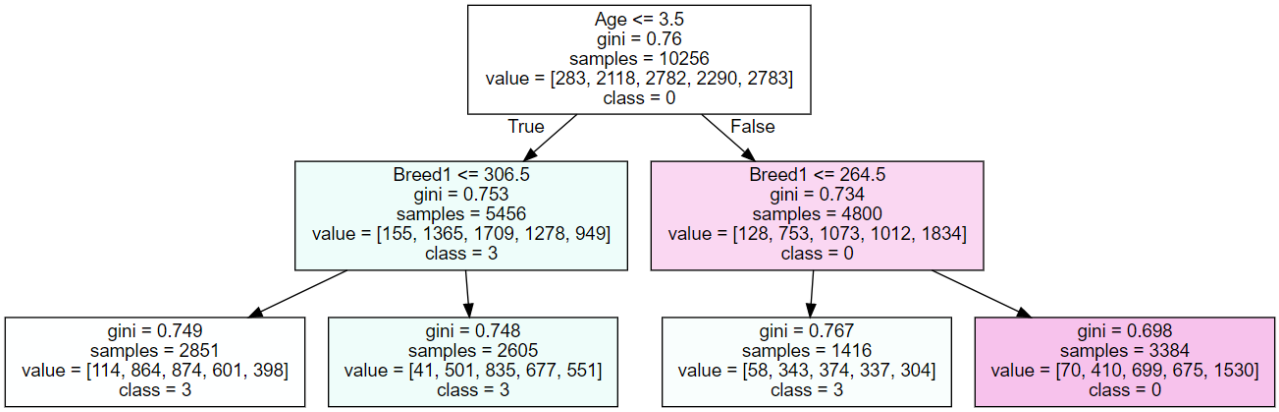

Рисунок 3.8 – Дерево рішень

Дерево рішень — це тип алгоритму ML, що складається з білого ящика. Він поділяє внутрішню логіку прийняття рішень, яка недоступна в алгоритмах типу чорного ящика, таких як нейронна мережа. Його час навчання швидше в порівнянні з алгоритмом нейронної мережі. Часова складність дерев рішень є функцією кількості записів і кількості атрибутів у наборі даних. Дерево рішень — це метод без розподілу або непараметричний, який не залежить від припущень щодо розподілу ймовірностей. Дерева рішень можуть обробляти дані великої розмірності з хорошою точністю.

Заходи вибору атрибутів

Міра вибору атрибутів — це евристика для вибору критерію поділу, який найкращим чином розподіляє дані. Він також відомий як правила розбиття, оскільки він допомагає нам визначити точки зупину для кортежів на даному вузлі. ASM надає ранг кожній функції (або атрибуту), пояснюючи даний набір даних. Атрибут найкращого результату буде вибрано як атрибут розділення (Джерело). У випадку атрибута з безперервним значенням, точки розділення для гілок також потрібно визначити. Найпопулярнішими показниками вибору є приріст інформації, коефіцієнт посилення та індекс Джині.

Приріст інформації

Шеннон винайшов концепцію ентропії, яка вимірює домішку вхідного набору. У фізиці та математиці ентропією називають випадковість або домішку в системі. У теорії інформації це відноситься до домішки в групі прикладів. Приріст інформації –

це зменшення ентропії. Приріст інформації обчислює різницю між ентропією до поділу та середньою ентропією після поділу набору даних на основі заданих значень атрибутів. Алгоритм дерева рішень ID3 (Ітеративний дихотомайзер) використовує приріст інформації (формула 3.17).

$$IG(T,a) = H(T) - H(T|a) = -\sum_{i=1}^{J} p_i \log_2 p_i - \sum_{i=1}^{J} -\Pr(i|a) \log_2 \Pr(i|a), \quad (3.17)$$

де $IG(T,a)$ – приріст інформації, $H(T)$ – ентропія, $H(T|a)$ – сума ентропій.

Коефіцієнт посилення

Прибуток інформації є упередженим для атрибута з багатьма результатами. Це означає, що він віддає перевагу атрибуту з великою кількістю різних значень.

C4.5, удосконалення ID3, використовує розширення для отримання інформації, відоме як коефіцієнт посилення. Коефіцієнт посилення вирішує проблему зміщення шляхом нормалізації підсилення інформації за допомогою Split Info (формула 3.18).

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2 \left(\frac{|D_j|}{|D|}\right), \quad (3.18)$$

де $\frac{|D_j|}{|D|}$ – вага j-го елемента, v-кількість дискретних значень в атрибуті A.

Коефіцієнт посилення можна визначити як:

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}. \quad (3.19)$$

Атрибут з найвищим коефіцієнтом посилення обирається як атрибут розщеплення.

Індекс Джині

Інший алгоритм дерева рішень CART (дерево класифікації та регресії) використовує метод Джині для створення точок розщеплення (формула 3.20).

$$Gini(D) = 1 - \sum_{i=1}^{m} P_i^2, \quad (3.20)$$

де p_i- ймовірність того, що кортеж у D належить до класу C_i.

Індекс Джині розглядає бінарне розщеплення для кожного атрибута. Якщо бінарне розщеплення атрибута A розбиває дані D на D_1 і D_2, індекс Джині для D:

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2). \quad (3.21)$$

У випадку атрибута з дискретним значенням, підмножина, яка дає мінімальний індекс Джині для цього вибраного, вибирається як атрибут розщеплення. У разі атрибутів з безперервними значеннями стратегія полягає у виборі кожної пари сусідніх значень як можливої точки розщеплення та точки з меншим індексом Джині, обраної як точка розщеплення (формула 3.22).

$$\Delta Gini(A) = Gini(D) - Gini_A(D). \quad (3.22)$$

Атрибут з мінімальним індексом Джині обирається як атрибут розщеплення.

3.5 Випадковий ліс (Random Forest)

Випадковий ліс – окремий випадок ансамблевого навчання, в якому індивідуальні моделі конструюються з використанням дерев рішень (рис. 3.9). Отриманий ансамбль використовується для прогнозування результату. При конструюванні окремих дерев використовують випадкові підмножини тренувальних даних. Це гарантує розкладання даних між різними деревами рішень. Як зазначалося вище, в ансамблевому навчанні дуже важливо забезпечити різнорідність ансамблю індивідуальних моделей.

Однією з найбільших переваг випадкових лісів є те, що вони не перенавчаються. У машинному навчанні ця проблема трапляється досить часто. Конструюючи

неоднорідну кількість дерев рішень за рахунок використання різних випадкових підмножин, ми гарантуємо відсутність перенавчання моделі на тренувальних даних. У процесі конструювання дерева рішень його вузли послідовно розщеплюються, і їм вибираються найкращі порогові значення, що знижують ентропію кожному рівні. У процесі розщеплення вузлів враховуються в повному обсязі ознаки, що характеризують дані вхідного набору. Натомість вибирається найкращий спосіб розщеплення вузлів, заснований на поточному випадковому піднаборі ознак, що розглядаються. Включення фактора випадковості збільшує зміщення випадкового лісу, проте дисперсія зменшується завдяки усередненню. Це зумовлює робастність результуючої моделі.

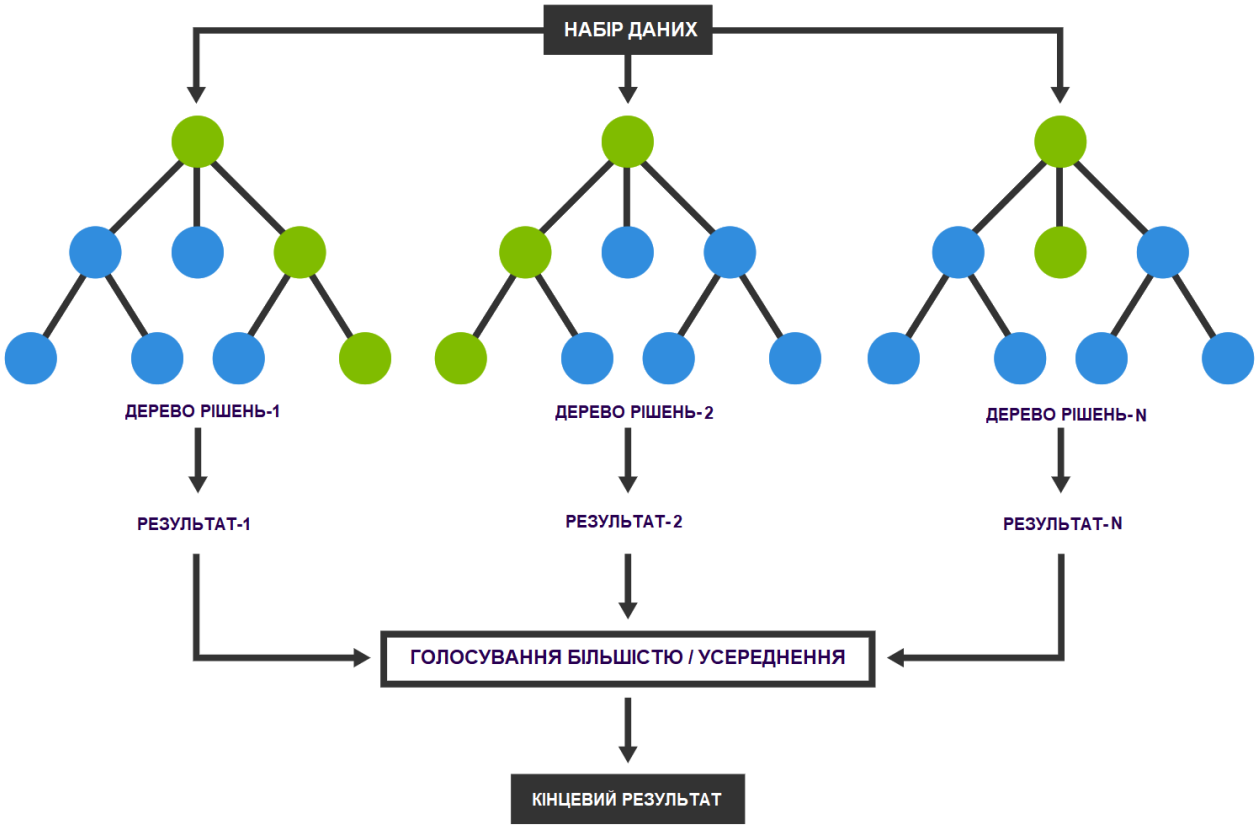

Рисунок 3.9 – Випадковий ліс

Випадковий ліс та градієнтний бустінг є ансамблевими методами підвищення продуктивності дерев рішень (шляхом досягнення кращої точності передбачення). Дерева мають велику дисперсію і дуже нестійкі. Випадкові ліси зменшують

дисперсію, виконують бутстреп-агрегування (беггінг), будують великий набір декорельованих дерев, а потім усереднюють їх [18]. Бустінг поєднує слабкі класифікатори (дерева) і створює більш потужну модель. Функціональність значно ширша, ніж у одного дерева, а функція бустінгу набагато згладженіша [19].

Гранично випадкові ліси (extremly random forests) ще більше посилюють роль випадковості. Поряд із випадковим вибором ознак випадково вибираються також порогові значення. Ері випадково генеровані значення стають правилами розбиття, що додатково зменшують варіативність моделі. Тому використання гранично випадкових лісів зазвичай призводить до більш гладких кордонів прийняття рішень, порівняно з тими, які вдається отримати за допомогою випадкових лісів.

3.6 Нейронні Мережі (Neural Networks)

Нейронні Мережі – група алгоритмів, що імітує роботу людського мозку, використовуючи штучні нейрони, як базові одиниці структури. Вони використовуються для вирішення різних задач, таких як кластеризація, класифікація та регресія. Для цілей цієї роботи використаємо два типи архітектур мереж: з повноз’єднаними шарами (англ. fully connected layers) та довгу короткострокову пам’ять (англ. Long short-term memory; LSTM). Мережа з повноз’єднаними шарами складається з декількох повністю з'єднаних шарів, тобто кожен нейрон поточного шару з'єднаний з кожним нейроном попереднього шару, і кожен зв'язок між двома нейронами має свою вагу. Через це мережа з повноз’єднаними шарами є ресурсозатратною, але підходить для загального користування. LSTM – тип рекурсивної нейронної мережі (англ. Recursive Neural Network, RNN), що складається із комірок та вихідних бар’єрів. Вихідні бар’єри Шлюзи використовуються для вирішення, яку і скільки інформації передати в наступну комірку. Комірка враховує як попередню, так і нову інформацію, для отримання остаточного результату. Ця структура LSTM дає можливість мережі добре обробляти послідовну сукупність даних.

Завданням роботи є передбачення швидкості усиновлення домашніх тварин за відповідною інформацією, такою як біологічні деталі, опис. Я використовуватиму традиційні методи машинного навчання, такі як логістична регресія, Naive Bayes, SVM, дерева рішень, випадковий ліс та градієнтний бустінг на категоріальних даних.

3.7 Класифікація фотокарток тварин за привабливістю

Для класифікації фотографій домашніх тварин за привабливістю можна скористуватися виявленням емоцій (emotion detection) [20]. Емоції людей та тварин корелюють між собою, оскільки мають спільні еволюційні характеристики. Вчені вже працювали із тваринами для виявлення їхніх емоцій. Розроблено багато різних технологій та методологій для ідентифікації та кількісної оцінки болю, що відчуває тварина [21].

Перебуваючи в притулку для тварин, коти та собаки відчувають широкий спектр емоцій, які глибоко впливають на їхнє емоційне здоров'я в короткостроковій перспективі, а також можуть мати довгострокові наслідки. Це особливо вірно, коли їх оточують люди, місця або речі, які постійно викликають страх або розчарування, або коли тривають стрес та пов'язані з ним негативні емоційні стани. Щоденний догляд за конем пов’язаний з ризиком отримання травм. У публікації повідомляється, що у моменти, коли кінь відчуває себе щасливим, він потребує більшої близькості, ніж у протилежній ситуації [22].

Зчитування і ідентифікація почуттів тварин, таких як примати, є більш простою задачею, оскільки їх жести та переживання мають подібність із людськими. Але розпізнавання емоційної складової риб, китів, пінгвінів, черепах та багатьох видів домашніх тварин. Розглянемо деякі індикатори. Коні із поганим настроєм закривають очі, при почутті страху корови рівно кладуть вуха рівно на голову, розлючені собаки мають жорстку позу тіла, тоді як щаслива собака має розслаблені очі та вуха. Існує чітка різниця між гавкотом розлюченої собаки, що захищає свій будинок, і щасливим гавканням тієї під час гри. Ми сприймаємо гавкіт собаки, як ознаку гніву, нявкання

котів, як ознаку голоду, але існує набагато більше факторів та ознак, від яких залежать їхні емоції. Примати, такі як шимпанзе, можуть видавати 32 звуки із різними тлумаченнями.

Це ознаки лише деяких тварин. Виявлення їхніх емоцій є важливою задачею обчислювальних методологій, в тому числі комп'ютерного зору. Маємо припущення, що емоційна складова, на фотографіях собак та котів, психологічно впливає на вибір людини, що забирає тварину з притулку, та є вагомим фактором визначення швидкості усиновлення із притулку. Тому має сенс кількісно визначити показник емоційності на фото домашньої тварини.

Чому визначення емоцій тварин є важливим?

Люди – істоти, які відчувають біль. Деякі тварини цієї властивості не мають, наприклад, міжнародна команда нейробіологів довела, що риби не відчувають болі. Оцінка рівня болю тварин є важливою для забезпечення ліками або належного лікування. Це є актуальним і для притулків тварин, в який тварини можуть потрапити у хворобливому стані, або ж набути захворювання безпосередньо там. Людина при проблемах зі здоров'ям звертається до лікаря, так само допомоги потребують і тварини. Виявлення рівню болю у вівці дозволяє передбачити такі захворювання, як копитна гниль та мастит. В більшості випадків доклінічних досліджень болю використовуються щури. Виявлення ступеню болі у щурів може допомогти у дослідженнях, а також позбавити їх від смерті чи проблем зі здоров'ям [23]. Своєчасно надана допомога може покращити привабливість тварини, що може збільшити швидкість усиновлення домашніх тварин із притулку.

При прийомі медикаментів тваринами, лікарі можуть краще підібрати препарат та дозування, використовуючи дані щодо емоційності тварин. Це дозволить запобігти прийому великих доз, що може ще більше нашкодити здоров'ю тварини. Медичні препарати тестуються на биках, щурах тощо, не знаючи інтенсивності емоцій та болю, що відчуває тварина. Рівень болю можна визначати за рухом обличчя, а також за звуком, що видає тварина. Зниження тиску та психічні захворювання можна визначити лише за емоціональністю. Не виявлення цих факторів може виявитися небезпечним як для господарів, так і для тварин, оскільки це може призвести до

смерті та ще гірших наслідків психічного здоров'я. Проблеми, які вирішує розпізнавання емоцій наведено на рис. 3.10.

Рисунок 3.10 – Застосування емоційного розпізнавання у лікуванні

Захист тварин

Однією з можливостей розпізнавання емоцій тварин, як диких, так і домашніх, може стати використання програм, встановлених на мобільний телефон, що допомагатиме визначати, коли тварина заспокоїлась. Розпізнавання емоцій тварин може допомогти виявляти командам порятунку тварин насильство та жорстоке поводження, що здійснюється у бік тварини. Здебільшого рятувальні бригади страждають, оскільки тварини вважають їх здобиччю. Такі злочини, як насильство та жорстоке поводження з тваринами, також можна ідентифікувати за емоціями. Команди захисту можуть безпечно доставити тварин, до яких застосовується насильство, до притулків.

Розпізнавання емоцій за фотокартками може допомогти визначити привабливість тварини, що знаходиться у притулку. Привабливість є критерієм, що впливає на вибір людини усиновлення тварини з притулку. У задачі визначення швидкості усиновлення оцінка привабливості, її вага, можуть сприяти покращенню

прогнозування, та використовувати інформацію не тільки щодо типу тварини, статі, породи тощо. Ефективне прогнозування швидкості усиновлення надасть можливість притулкам оптимально розподіляти ресурси, щоб поліпшити загальну продуктивність усиновлення, та як наслідок, зменшити витрати на притулок та виховання.

Зручна комунікація

Зручна комунікація – це умовний міст розуміння почуттів між людьми, тваринами або між людьми та тваринами. Люди можуть розпізнати тонкі відмінності між своїми емоціями та емоціями тварини. Любителі тварин здатні розуміти емоції тварин, тоді як для тих, хто любить тварин, або людей, які мають нейтральні емоції до тварин, їм стає важко зрозуміти свої почуття, і наслідки цих явищ спостерігаються у багатьох випадках щодня. Тварини виражають свої емоції здебільшого своїми діями, які можна диференціювати, коли всі ознаки добре відомі, наприклад - якщо собака чухається, що означає голод, або це свідчить про щось не так, якщо вона стрибає через вас, це не вказує на те, що воно намагається напасти на вас, бо вони також стрибають від відчуття щастя.

Безпека

Домашні тварини є не лише частиною родин, також вони захищають будинки від небезпеки, як в присутності господаря, так і без. Системи виявлення емоцій тварин можна встановити в охоронних камерах будинків і на робочих місцях, що по емоціям та дивній поведінці тварини зможуть сигналізувати про загрозу, що відбувається навколо них, або у середині будинку.

Розглянемо існуючі рішення, що стосуються класифікації зображень.

Згорткові мережі є основою більшості сучасних рішень комп’ютерного зору для вирішення широкого спектру завдань. З 2014 року глибокі згорткові мережі почали ставати мейнстрімом, що принесло значний виграш у різних контрольних показниках. Хоча збільшення розміру моделі та обчислювальної вартості, як правило, призводить до негайного підвищення якості для більшості завдань (за умови, що надається достатньо розмічених даних для навчання), обчислювальна ефективність і низька кількість параметрів все ще є факторами для різних випадків використання, таких як мобільний зір і сценаріїв великих даних. Досліджуються способи масштабування

мереж таким чином, щоб якомога ефективніше використовувати додаткові обчислення за допомогою відповідних факторних згорток і агресивної регуляризації. Порівняно з методами з набору перевірки класифікації ILSVRC 2012, демонструють значний виграш у порівнянні з сучасним рівнем техніки: 21,2% похибки топ-1 і 5,6% помилок топ-5 для оцінки одного фрейму з використанням мережі з обчислювальною вартістю 5 мільярдів параметрів на висновок і з використанням менше 25 мільйонів параметрів. Завдяки ансамблю з 4 моделей та оцінці кількох культур повідомляється про 3,5% помилок з топ-5 і 17,3% топ-1 помилок.

Розглянуто кілька принципів проєктування, заснованих на масштабних експериментах з різними архітектурними рішеннями зі згортковими мережами. На даному етапі корисність наведених нижче принципів є спекулятивною, і для оцінки їх точності та області достовірності знадобляться додаткові експериментальні докази в майбутньому. Проте серйозні відхилення від цих принципів, як правило, призводили до погіршення якості мереж, а виправлення ситуацій, коли ці відхилення були виявлені, призводило до покращення архітектури в цілому.

1. Уникнення вузьких місць, особливо на початку мережі. Мережі прямого зв'язку можуть бути представлені ациклічним графіком від вхідного рівня до класифікатора або регресора. Це визначає чіткий напрямок інформаційного потоку. Для будь-якого розрізу, що відокремлює входи від виходів, можна отримати доступ до кількості інформації, що проходить через розріз. Слід уникати вузьких місць з екстремальним стисненням. Загалом розмір представлення повинен плавно зменшуватися від входів до виходів, перш ніж досягти остаточного представлення, що використовується для відповідної задачі. Теоретично зміст інформації не можна оцінити лише за розмірністю представлення, оскільки воно відкидає такі важливі фактори, як кореляційна структура; розмірність дає лише приблизну оцінку змісту інформації.

2. Представлення більших розмірностей легше обробляти локально в мережі. Збільшення кількості активацій на комірку в згортковій мережі дозволяє отримати більше роз’єднаних функцій. Отримані мережі тренуватимуться швидше.

3. Просторова агрегація може бути виконана на більш низьких розмірах вбудовування без значної або будь-якої втрати репрезентативної сили. Наприклад, перед виконанням більш розгорнутої (наприклад, 3×3) згортки можна зменшити розмірність вхідного представлення перед просторовою агрегацією, не очікуючи серйозних негативних наслідків. Зроблено припущення, що причиною цього є сильна кореляція між сусідніми одиницями, що призводить до значно меншої втрати інформації під час зменшення розмірів, якщо вихідні дані використовуються в контексті просторової агрегації. Враховуючи, що ці сигнали повинні легко стискатися, зменшення розмірності навіть сприяє швидшому навчанню.

4. Збалансування ширини та глибину мережі. Оптимальна продуктивність мережі може бути досягнута шляхом збалансування кількості фільтрів на етапі та глибини мережі. Збільшення як ширини, так і глибини мережі може сприяти підвищенню якості мереж. Однак оптимальне покращення для постійної кількості обчислень можна досягти, якщо обидва збільшувати паралельно. Тому обчислювальний ресурс слід розподіляти збалансовано між глибиною та шириною мережі.

Хоча ці принципи можуть мати сенс, використовувати їх для покращення якості мереж непросто. Ідея полягає в тому, щоб використовувати їх розумно лише в неоднозначних ситуаціях.

Розкладання згорток на множники з великим розміром фільтра.

Значна частина оригінальних переваг мережі GoogLeNet [24] випливає з дуже щедрого використання зменшення розмірності. Це можна розглядати як окремий випадок факторізації згорток обчислювально ефективним способом. Розглянемо для прикладу випадок згорткового шару 1×1, потім згортковий шар 3×3. У зоровій мережі очікується, що результати активації поблизу мають високу кореляцію. Таким чином, ми можемо очікувати, що їх активація може бути зменшена перед агрегацією, і що це має призвести до подібних виразних локальних уявлень.

Досліджуються інші способи факторізації згорток у різних налаштуваннях, особливо з метою підвищення обчислювальної ефективності рішення. Оскільки початкові мережі є повністю згортковими, кожна вага відповідає одному множенню

за активацію. Тому будь-яке зниження витрат на обчислення призводить до зменшення кількості параметрів. Це означає, що при відповідній факторізації ми можемо отримати більше роз'єднаних параметрів і, отже, швидше навчання. Крім того, можливо використовувати економію обчислень і пам'яті, щоб збільшити розміри банку фільтрів нашої мережі, зберігаючи при цьому нашу здатність навчати кожну репліку моделі на одній обчислювальній машині.

3.7.1 Розкладання на дрібні згортки

Згортки з більшими просторовими фільтрами (наприклад, 5 × 5 або 7 × 7), як правило, є непропорційно дорогими з точки зору обчислень. Наприклад, згортка 5 × 5 з n фільтрами над сіткою з m фільтрами в 25/9 = 2,78 разів дорожча обчислень, ніж згортка 3 × 3 з такою ж кількістю фільтрів. Звичайно, фільтр 5×5 може фіксувати залежності між сигналами між активаціями одиниць, що знаходяться далі на попередніх шарах, тому зменшення геометричного розміру фільтрів коштує великої вартості виразності. Однак ми можемо запитати, чи можна замінити згортку 5 × 5 на багатошарову мережу з меншими параметрами з тим же розміром вхідного сигналу та вихідною глибиною. Якщо ми збільшимо графік обчислень згортки 5 × 5, ми побачимо, що кожен вихід виглядає як невелика повністю пов'язана мережа, що ковзає по плитках 5 × 5 над своїм вхідним сигналом (див. рисунок 3.13). Оскільки ми створюємо мережу бачення, здається природним знову використовувати інваріантність трансляції та замінити повністю зв'язаний компонент двошаровою згортковою архітектурою: перший шар — це згортка 3 × 3, другий — повністю зв'язаний шар поверх Вихідна сітка 3 × 3 першого шару (див. рис. 3.11). Переміщення цієї маленької мережі по сітці активації вхідних даних зводиться до заміни згортки 5 × 5 двома шарами згортки 3 × 3 (порівняння на рисунках 3.11 та 3.12).

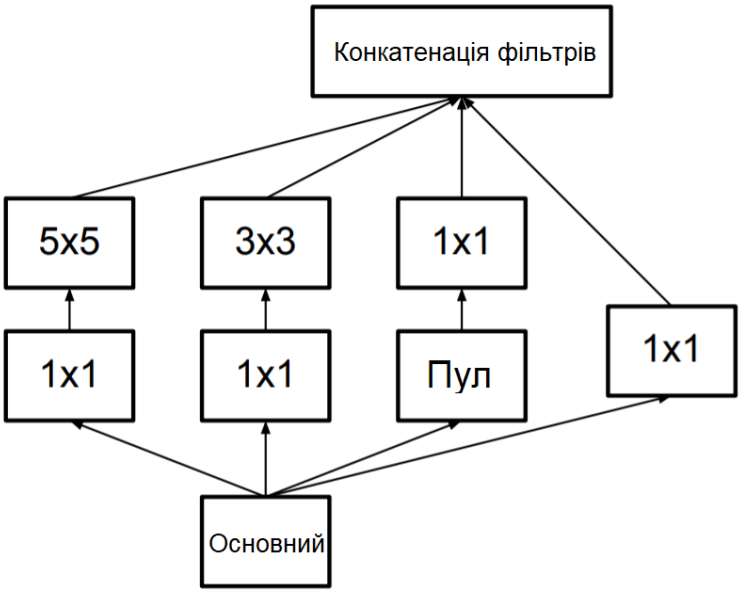

Рисунок 3.11 - Оригінальний модуль Inception

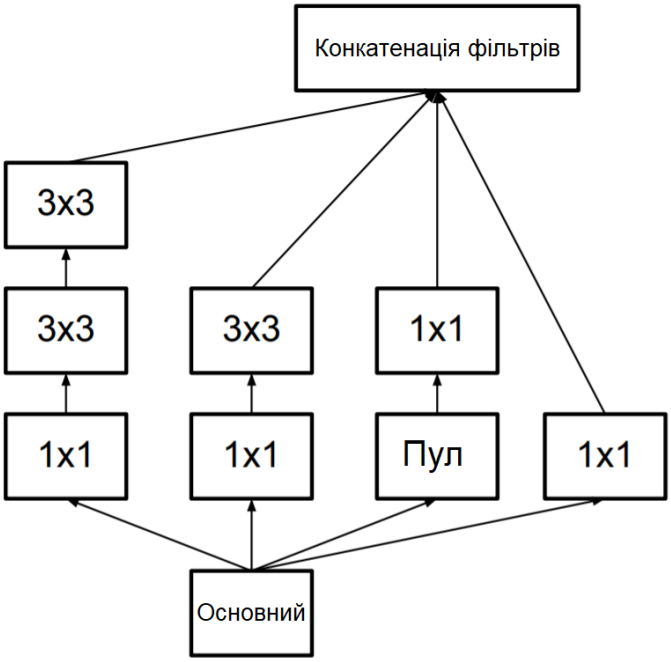

Рисунок 3.12 – Модулі Inception, де кожна згортка 5×5 замінена двома згортками 3×3

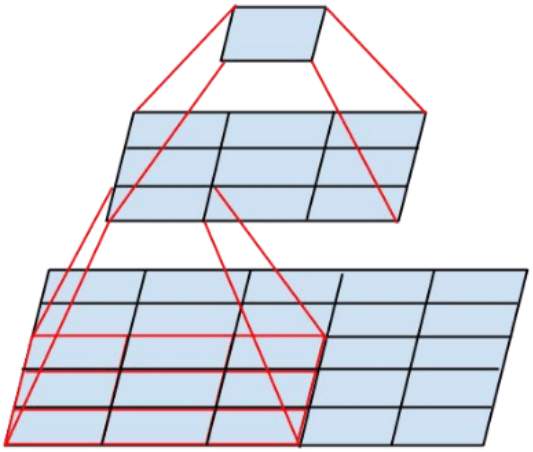

Рисунок 3.13 – Міні-мережа, яка трансформує згортку 5 × 5

Це налаштування чітко зменшує кількість параметрів, розподіляючи вагу між сусідніми плитками. Щоб проаналізувати очікувану економію обчислювальних витрат, ми зробимо кілька спрощуючих припущень, які застосовуються для типових ситуацій: Ми можемо припустити, що $n = \alpha m$, тобто ми хочемо змінити кількість активацій на одиницю на постійний альфа-фактор. . Оскільки згортка 5 × 5 агрегується, α зазвичай трохи більше одиниці (близько 1,5 у випадку GoogLeNet). Маючи двошарову заміну шару 5×5, здається розумним досягти цього розширення в два кроки: збільшення кількості фільтрів на $\sqrt{\alpha}$ на обох кроках. Щоб спростити нашу оцінку, вибравши $\alpha = 1$ (без розширення), якби ми наївно переміщали мережу без повторного використання обчислень між сусідніми плитками сітки, ми б збільшили обчислювальні витрати. ковзання цієї мережі може бути представлено двома згортковими шарами 3×3, які повторно використовують активації між сусідніми плитками. Таким чином, ми отримуємо чисте скорочення обчислень у $\frac{9+9}{25}$ ×, що призводить до відносного виграшу на 28% від цієї факторізації. Точно так само зберігається і для кількості параметрів, оскільки кожен параметр використовується рівно один раз під час обчислення активації кожного блоку. І все-таки ця установка викликає два загальні питання: чи призведе ця заміна до втрати виразності? Якщо головна мета — розкласти на множники лінійну частину обчислення, чи не

запропонує це зберегти лінійні активації на першому рівні? Використання лінійної активації завжди поступається використанню випрямлених лінійних одиниць на всіх етапах факторізації. Цей виграш пояснюється розширеним простором варіацій, які може засвоїти мережа, особливо якщо ми пакетно нормалізуємо [23] вихідні активації. Подібні ефекти можна побачити при використанні лінійних активацій для компонентів зменшення вимірів.

3.7.2 Просторова факторізація на асиметричні згортки

Наведені вище результати свідчать про те, що згортки з фільтрами більше 3×3 а можуть бути некорисними, оскільки їх завжди можна звести до послідовності згорткових шарів 3×3. І все-таки ми можемо поставити питання, чи слід розкладати їх на менші, наприклад 2×2 згортки. Однак виявляється, що можна зробити навіть краще, ніж 2×2, використовуючи асиметричні згортки, напр. $n \times 1$. Наприклад, використання згортки 3×1 з наступною згорткою 1×3 еквівалентно ковзанню двошарової мережі з тим же сприйнятливим полем, що й у згортки 3×3 (рис. 3.14). Все-таки двошарове рішення на 33% дешевше при однаковій кількості вихідних фільтрів, якщо кількість вхідних і вихідних фільтрів однакова. Для порівняння, розкладання згортки 3×3 на дві згортки 2×2 дає лише 11% економії обчислення.

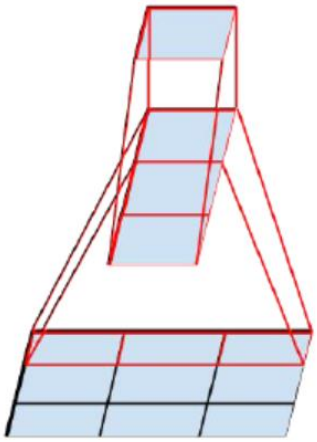

Рисунок 3.14 – Міні-мережа, яка трансформує згортку 3×3

3.7.3 Експериментальні результати та порівняння

У таблиці 3.1 показані експериментальні результати щодо ефективності розпізнавання архітектури Inceptionv2. Кожен рядок Inception-v2 показує результат сукупних змін, включаючи виділену нову модифікацію та всі попередні. Розкладене на множники 7 × 7 включає зміну, яка розкладає на множники перших 7 × 7 згорткових шарів у послідовність з 3 × 3 згорткових шарів. BN-допоміжний відноситься до версії, в якій повністю зв'язаний шар допоміжного класифікатора також пакетно-нормований, а не тільки згортки. Ми називаємо модель в останньому рядку таблиці 3.2 як Inception-v3 і оцінюється її продуктивність у налаштуваннях кількох кадрів та ансамблю.

Таблиця 3.1 – Експериментальні результати multi-crop, що порівнюють кумулятивний вплив на різні сприяючі фактори

Мережа	Топ-1	Топ-5	Вартість
GoogLeNet	29%	9.2%	1.5
BN-GoogLeNet	26.8%	-	**1.5**
BN-Inception	25.2%	7.8	2.0
Inception-v2	23.4%	-	3.8
Inception-v2 RMSProp	23.1%	6.3	3.8
Inception-v2 Label Smoothing	22.8%	6.1	3.8
Inception-v2 Factorized 7 × 7	21.6%	5.8	4.8
Inception-v2 BN-auxiliary	**21.2%**	**5.6%**	4.8

Таблиця 3.2 – Одномодельні експериментальні результати «multi-crop», що порівнюють кумулятивний вплив на різні сприяючі фактори

Мережа	Crop-оцінка	Топ-5	Топ-1
GoogLeNet	10	-	9.15%
BN-GoogLeNet	144	-	7.89%
VGG	-	24.4%	6.8%
BN-Inception	144	22%	5.82%
PReLU	10	24.27%	7.38%
PReLU	-	21.59%	5.71%
Inception-v3	12	19.47%	4.48%
Inception-v3	144	**18.77%**	**4.2%**

Найкращий показник точності у моделі Inception-v3, який досягає 21.2% при топ-1, та 5.6% при топ-5 помилках для «multi-crop» за класифікацією ILSVR 2012. Це досягається за допомогою відносно невеликого (2,5×) збільшення обчислювальних витрат порівняно з мережею, описаною в [25]. Ансамбль із чотирьох моделей Inception-v3 досягає 3,5% з оцінкою «multi-crop», досягає 3,5% помилки топ-5, що становить більш ніж 25% зменшення до найкращих опублікованих результатів і майже половину помилки ILSVRC 2014, що краще ансамблю GoogLeNet.

Для навчання моделі нейронної мережі буде використовуватися набір даних (фотокарток), розділений на 3 класи:

- привабливі
- нейтральні за привабливістю
- не привабливі

Набір даних для навчання складається з 4274 розмічених зображень: привабливих – 1387 штук, нейтральних – 1681, не привабливих – 1206. На рисунку 3.15 наведено фрагмент фотографій тварин, що будуть використовуватись для навчання.

За замовчуванням для навчання класифікатору було заплановано 30 епох. На 16-й епосі навчання зупинилось, оскільки в попередніх епохах 5 разів поспіль не відбулося покращення результату (рис. 3.16). На рисунку 3.17 наведено графік зміни параметрів точності та витрат по епохах навчання. Отримана точність класифікації – 0,7775.

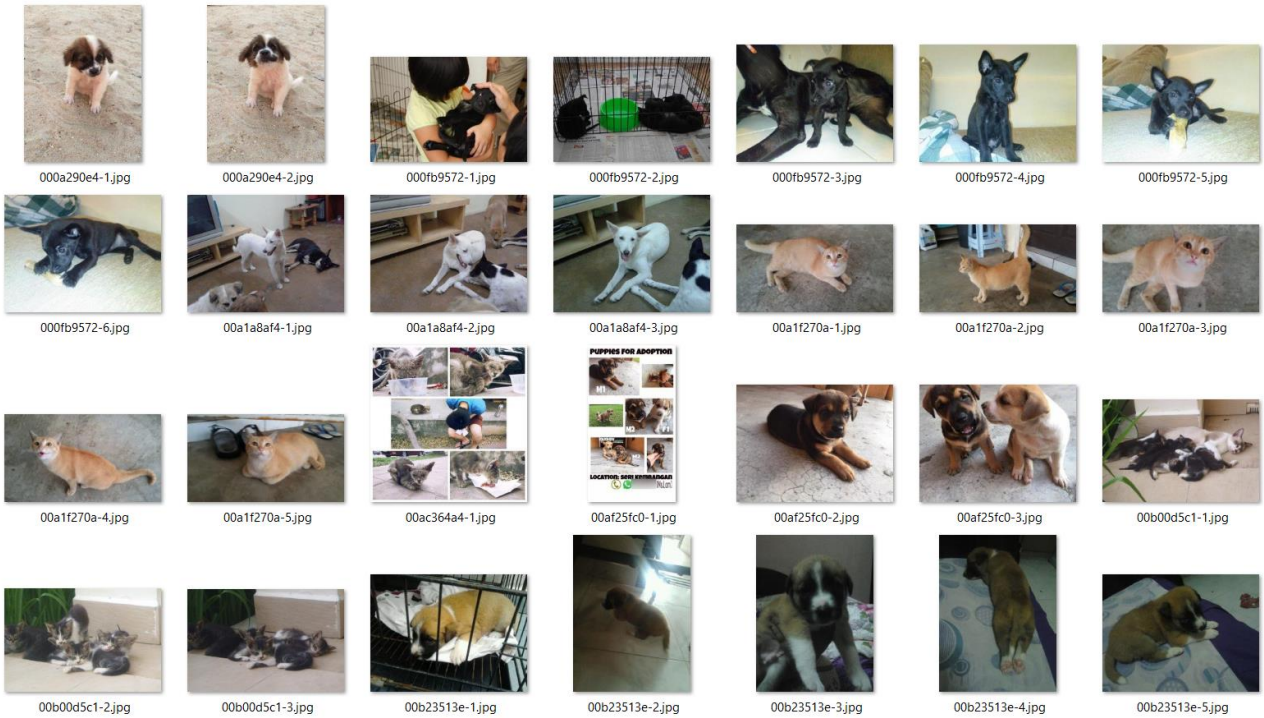

Рисунок 3.15 – Фрагмент фотокарток тварин, що будуть використовуватись для навчання класифікатору

```
history_model = model.fit_generator(image_data, steps_per_epoch = 10, epochs = 30, callbacks=cb)
```

```
Epoch 1/30
10/10 [==============================] - ETA: 0s - loss: 17.2764 - accuracy: 0.3662
Epoch 00001: accuracy improved from -inf to 0.36620, saving model to ./classification_model.h5
10/10 [==============================] - 43s 871ms/step - loss: 17.2764 - accuracy: 0.3662
Epoch 2/30
10/10 [==============================] - ETA: 0s - loss: 11.0653 - accuracy: 0.5183
Epoch 00002: accuracy improved from 0.36620 to 0.51831, saving model to ./classification_model.h5
10/10 [==============================] - 8s 774ms/step - loss: 11.0653 - accuracy: 0.5183
Epoch 3/30
10/10 [==============================] - ETA: 0s - loss: 8.7792 - accuracy: 0.5493
Epoch 00003: accuracy improved from 0.51831 to 0.54930, saving model to ./classification_model.h5
10/10 [==============================] - 8s 776ms/step - loss: 8.7792 - accuracy: 0.5493
Epoch 4/30
10/10 [==============================] - ETA: 0s - loss: 4.0835 - accuracy: 0.5493
Epoch 00004: accuracy did not improve from 0.54930
10/10 [==============================] - 7s 668ms/step - loss: 4.0835 - accuracy: 0.5493
Epoch 5/30
10/10 [==============================] - ETA: 0s - loss: 2.8053 - accuracy: 0.6225
Epoch 00005: accuracy improved from 0.54930 to 0.62254, saving model to ./classification_model.h5
10/10 [==============================] - 8s 773ms/step - loss: 2.8053 - accuracy: 0.6225
Epoch 6/30
10/10 [==============================] - ETA: 0s - loss: 2.6461 - accuracy: 0.6306
Epoch 00006: accuracy improved from 0.62254 to 0.63056, saving model to ./classification_model.h5
10/10 [==============================] - 8s 787ms/step - loss: 2.6461 - accuracy: 0.6306
Epoch 7/30
10/10 [==============================] - ETA: 0s - loss: 1.8518 - accuracy: 0.6873
Epoch 00007: accuracy improved from 0.63056 to 0.68732, saving model to ./classification_model.h5
10/10 [==============================] - 8s 785ms/step - loss: 1.8518 - accuracy: 0.6873
Epoch 8/30
10/10 [==============================] - ETA: 0s - loss: 1.4048 - accuracy: 0.7352
Epoch 00008: accuracy improved from 0.68732 to 0.73521, saving model to ./classification_model.h5
10/10 [==============================] - 8s 775ms/step - loss: 1.4048 - accuracy: 0.7352
Epoch 9/30
10/10 [==============================] - ETA: 0s - loss: 1.3831 - accuracy: 0.7211
Epoch 00009: accuracy did not improve from 0.73521
10/10 [==============================] - 7s 666ms/step - loss: 1.3831 - accuracy: 0.7211
Epoch 10/30
10/10 [==============================] - ETA: 0s - loss: 1.2286 - accuracy: 0.7521
Epoch 00010: accuracy improved from 0.73521 to 0.75211, saving model to ./classification_model.h5
10/10 [==============================] - 8s 772ms/step - loss: 1.2286 - accuracy: 0.7521
Epoch 11/30
10/10 [==============================] - ETA: 0s - loss: 1.0276 - accuracy: 0.7917
Epoch 00011: accuracy improved from 0.75211 to 0.79167, saving model to ./classification_model.h5
10/10 [==============================] - 8s 769ms/step - loss: 1.0276 - accuracy: 0.7917
Epoch 12/30
10/10 [==============================] - ETA: 0s - loss: 1.0371 - accuracy: 0.7803
Epoch 00012: accuracy did not improve from 0.79167
10/10 [==============================] - 7s 669ms/step - loss: 1.0371 - accuracy: 0.7803
Epoch 13/30
10/10 [==============================] - ETA: 0s - loss: 0.9618 - accuracy: 0.7803
Epoch 00013: accuracy did not improve from 0.79167
10/10 [==============================] - 7s 665ms/step - loss: 0.9618 - accuracy: 0.7803
Epoch 14/30
10/10 [==============================] - ETA: 0s - loss: 1.2689 - accuracy: 0.7887
Epoch 00014: accuracy did not improve from 0.79167
10/10 [==============================] - 7s 663ms/step - loss: 1.2689 - accuracy: 0.7887
Epoch 15/30
10/10 [==============================] - ETA: 0s - loss: 1.2160 - accuracy: 0.7521
Epoch 00015: accuracy did not improve from 0.79167
10/10 [==============================] - 7s 657ms/step - loss: 1.2160 - accuracy: 0.7521
Epoch 16/30
10/10 [==============================] - ETA: 0s - loss: 0.9478 - accuracy: 0.7775
Epoch 00016: accuracy did not improve from 0.79167
10/10 [==============================] - 7s 657ms/step - loss: 0.9478 - accuracy: 0.7775
Epoch 00016: early stopping
```

Рисунок 3.16 – Етапи навчання класифікатору

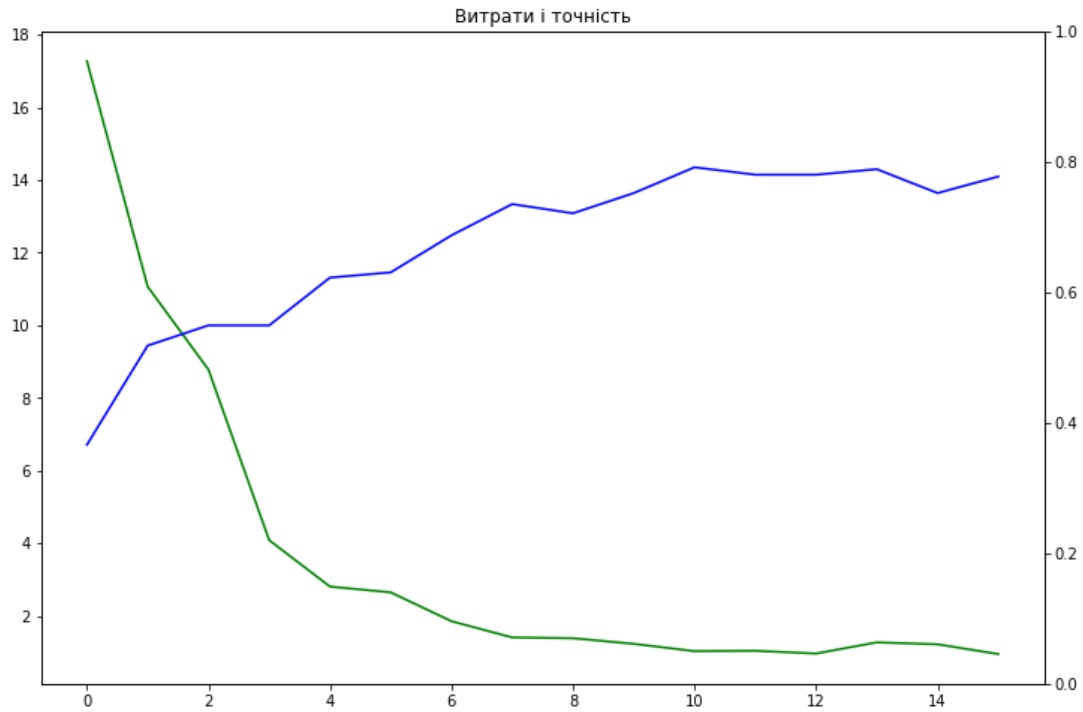

Рисунок 3.17 – Зміна параметрів у процесі навчання класифікатору

3.8 Обробка тексту методами NLP

У наборі даних присутнє поле опису тварин – колонка «Description». Для побудови моделі можемо використати методи NLP (Natural language processing; укр., Обробка природної мови). До обробленого тексту можна застосувати алгоритми машинного навчання та нейронних мереж.

Обробка природної мови (NLP) - це область штучного інтелекту, в якій комп'ютери обробляють, аналізують, розуміють та виводять значення людської мови. Використовуючи NLP, розробники можуть організувати та структурувати знання для виконання таких завдань, як автоматичне підведення підсумків (automatic summarization), переклад, розпізнавання іменованих сутностей (named entity recognition), вилучення зв'язків (relationship extraction), аналіз настроїв (sentiment analysis), розпізнавання мови (speech recognition) та сегментація тем.

У статті [26] експерт з NLP у Meltwater Group, Джон Релінг, «Окрім звичайних операцій обробки текстів, які трактують текст як просту послідовність символів, NLP

враховує ієрархічну структуру мови: кілька слів складають фразу, кілька фраз складають речення і, зрештою, речення передають сенс». Системи NLP вже довготривалий період виконують важливу роль в таких задачах визначення сенсу мови, як виправлення граматики, перетворення мови в текст і автоматичний переклад між мовами.

NLP використовується для аналізу тексту, дозволяючи машинам розуміти людське спілкування. Ця взаємодія між людиною та комп'ютером дає змогу реалізовувати такі застосунки, як автоматичне узагальнення тексту, аналіз настроїв, вилучення теми, розпізнавання іменних сутностей, визначення частин мови, вилучення зв'язків, стемінг тощо. NLP зазвичай використовується для аналізу тексту, машинного перекладу та автоматизованих відповідей на запитання.

NLP є складною проблемою інформатики. Розуміння людської мови означає розуміння не лише слів, а й понять, та їх зв'язків. Незважаючи на те, що мова є однією з найпростіших для вивчення людським розумом, неоднозначність мови робить обробку природної мови складною проблемою для розумінню комп'ютерами.

Перший етап обробки тексту

Токенізація (Tokenization)

Токенізація – процес розбиття фрагмента тексту на лексеми (токени). Лексемами можуть виступати слова, частини слів або розділові знаки. Це одне з найбільш фундаментальних і складних завдань NLP, оскільки кожна мова має свої граматичні конструкції, за якими важко визначити правила токенізації. Обробка природної мови йде паралельно із "офіційними мовами", галуззю між лінгвістикою та інформатикою, яка по суті вивчає мовні аспекти мов програмування. Як і в природній мові, офіційні мови мають різні рядки, що мають значення; у більшості випадків рядками виступають слова, але щоб уникнути плутанини, в офіційних мовах рядки називають лексемами.

Навіщо розбивати текст на лексеми (токени)?

Мови програмування працюють, розбиваючи необроблений код на токени, а потім комбінують їх за певною логікою (граматикою програми) в NLP. Але в процесі обробки природної мови протягом багатьох років розвиваються різні способи

поєднання лексем, а також безліч методів токенізації. Але сенс токенізації залишилається незмінним - представити комп'ютеру деякий кінцевий набір символів, які він може об'єднати для отримання бажаного результату.

Перетворення токенів у смислову конструкцію

Word2Vec [26] та GloVe [27] - це дві моделі побудови векторних подань слів. Ідея полягає в тому, щоб подати семантично подібні слова у подібному векторному просторі. Word2Vec - це метод, заснований на передобчисленні, який навчається на великому наборі даних із досить низькими обчислювальними вимогами. Це перевершує багато традиційних методів. GloVe є методом підрахунку. У роботі [27] Пеннінгтон та співавтори стверджують, що обидва методи поділяють основні ідеї та фіксують подібну семантичну інформацію, але GloVe є більш ефективним у виконанні завдань.

Робота 2017 року, виконана Мірсамаді та співавторами [28] щодо розпізнавання емоцій у мовленні, свідчить про те, що використання простої моделі локальної уваги дозволяє визначити емоційно значущі слова. На основі цієї моделі уваги вони пропонують новий метод, який дозволяє моделі надавати більшу вагу словам, що приймають участь при навчанні, для досягнення кращої точності класифікації.

Алгоритми Word2Vec або GloVe присвоюють токену вектор, який надає лексемі сенс для розуміння комп'ютером. Використовуючи дані моделі можна провести математичні операції додавання та віднімання типу: $w_{\text{король}} - w_{\text{чоловік}} + w_{\text{жінка}}$ (рис. 3.18).

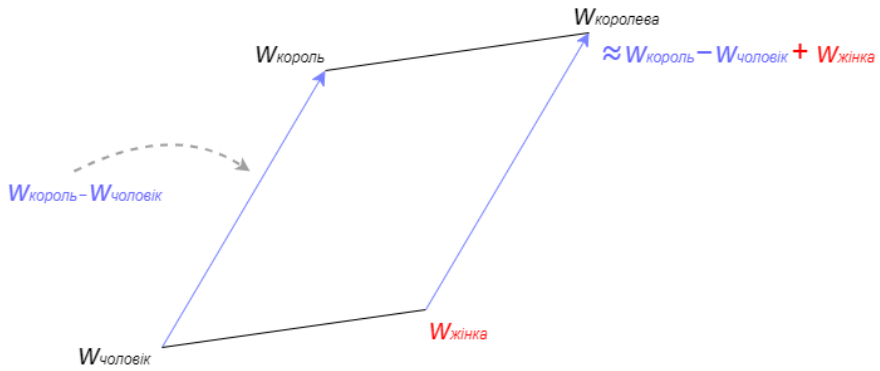

Рисунок 3.18 – Векторне подання лексем

На перший погляд, урахування при побудові моделі всіх токенів здається оптимальним рішенням. Однак такий вибір часто має собівартість. Наявність великої кількості дефіцитних лексем робить системи NLP схильними до перенавчання [29].

Сучасні методи глибокого навчання інтерпретують токени, в залежності їх контексту, із врахуванням появи нових [30]. Ця властивість дає більшу гнучкість при появі рідкісних лексем, оскільки система може розпізнати значення в залежності від контексту. Ці відносно нові можливості NLP змінили спосіб токенізації тексту. Системи глибокого навчання, як правило, характеризуються pipeline- підходом [31].

Більш сучасні схеми токенізації, які часто називають токенізаторами підслів, такі як фрагмент речення або алгоритми BPE, розділяють і об’єднують токени в більш складні форми [32].

Stemming and Lemmatization (стемінг та лематизація)

Першим етапом NLP є приведення текстових даних до придатної для розуміння обчислювальною технікою форми [33]. В обробці широко використовуються такі методи нормалізації слів, як стемінг та лематизація. Часто вони використовуються при впровадженні пошукових систем для коректної обробки варіантів того ж самого слова. Обидві техніки мають свої унікальні переваги й недоліки. Далі розберемо, коли краще використовувати ту або іншу.

Стемінг (Stemming) - це прямий прийом нормалізації, який найчастіше реалізується як сукупність правил, які послідовно застосовуються до слова для отримання нормалізованої форми. Ці правила різняться залежно від мови, і вони відображають морфологічну структуру мови, про яку йдеться. Наприклад, для англійської мови можливим правилом може бути видалення «s» у кінці слова, щоб перетворити його у форму однини.

Техніка стемінгу не вимагає, щоб нормалізоване слово існувало насправді. Стемінг знаходить у множині слів спільний корінь, цей корінь і є результатом. Розглянемо, як дана техніка може бути застосована для слів англійської мови. На рисунку 3.20 маємо множину слів: «change», «changing», «changes», «changed», «changer». Спільний корінь - «chang», що не є дійсним словом англійської мови.

Стемінг використовується для внутрішньої обробки в обчислювальній техніці, тобто звичайний користувач доступу до нормалізованих слів не отримує.

Лематизація (Lemmatization) є більш витонченою версією стемінгу. Ця техніка трансформує слово до базової форми, тобто нормалізоване слово існує насправді. Результат нормалізації залежить від наявної інформації про частину мови вхідного слова.

Для прикладу, розглянемо англійське слово «changing». В залежності від контексту воно може бути іменником («changing your report is essential»; укр., зміна звіту є дуже важливою), дієсловом («changed my plans»; укр., «змінив мої плани»), або прикметником («the changing weather»; укр., «мінлива погода»). Стемінг у цьому випадку просто видалив би суфікси «ing», «ed», не вдаючись у подробиці сенсу слова.

Перевагою стемінгу є його простота впровадження та швидкість у використанні. Але результат може містити неточності, що можуть бути нерелевантними для деяких задач. Лематизація дає кращі результати у задачах, в яких бажано враховувати частину мови слова, але техніка є складнішою у реалізації та менш продуктивною.

З точки зору якості, лематизація є кращим вибором. За сучасних обчислювальних ресурсів запуск алгоритмів лематизації не повинен значно впливати на загальну продуктивність. Однак, якщо в значній мірі необхідно оптимізувати швидкість, стемінг є задовільним варіантом.

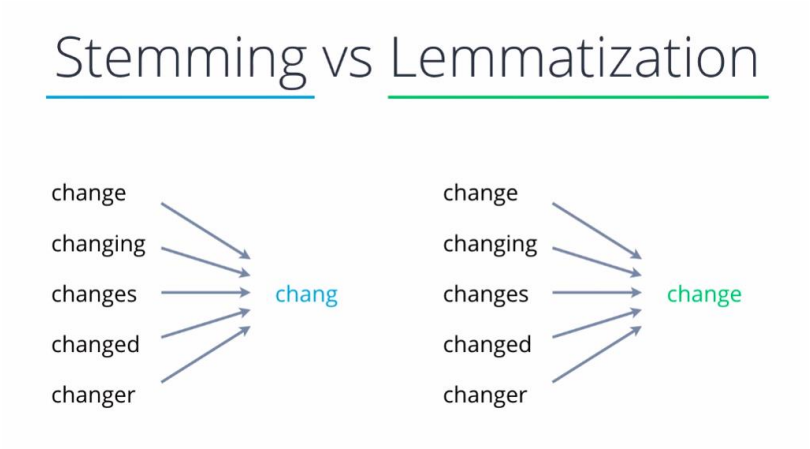

Рисунок 3.19 – Порівняння стемінгу та лематизації

Stop Words (стоп-слова)

Стоп-слова – це слова, які у будь-якій мові не надають особливого сенсу реченню. Вони можуть бути безпечно проігноровані без втрати сенсу речення. Одними з найпоширеніших коротких функціональних слів в англійській мові є «is», «at», «which» та «on».

В обчисленнях стоп-слова відфільтровуються до або після обробки тексту засобами NLP. [34] Хоча «стоп-слова» зазвичай стосуються найпоширеніших слів у мові, цілком засоби природної обробки мови не використовують єдиний універсальний список стоп-слів.

Коли стоп-слова можуть бути видалені

Якщо перед нами стоїть завдання класифікації тексту або семантичного аналізу, то нам слід видалити стоп-слова, оскільки вони не надають ніякої цінної інформації нашій моделі, але якщо ми маємо завдання перекладу мови, то збереження стоп-слів приносить користь, оскільки їх необхідно перекладати разом з іншими словами.

Не існує чіткого правила щодо того, коли треба позбавлятися від стоп-слів. Зазвичай стоп-слова видаляються при задачах Language Classification (мовна класифікація), Spam Filtering (фільтрація спаму), Caption Generation (генерація підписів), Auto-Tag Generation (генерація авто-тегів), Sentiment analysis (семантичний аналіз), або інших, що стосуються текстової класифікації.

З іншого боку, якщо нашим завданням є Machine Translation (машинний переклад), Question-Answering problems (проблема «Питання-Відповідь»), Text Summarization (узагальнення тексту), Language Modeling (моделювання мови).

Переваги і недоліки видалення стоп-слів

Переваги

– Стоп-слова часто видаляються з тексту перед тренуванням моделей глибокого та машинного навчання, оскільки вони не надають майже жодної унікальної інформації, яка може бути використана для класифікації або кластеризації.

– При видаленні стоп-слів розмір набору даних зменшується, відповідно час навчання моделі також зменшується, без значного впливу на точність моделі.

– Видалення стоп-слів може потенційно допомогти в покращенні продуктивності, оскільки залишаються лише значущі токени.

Недоліки

– Неправильний підбір та видалення стоп-слів може змінити значення тексту, що розглядається.

Text to Features (Перетворення тексту для побудови моделей)

Feature - це назва, яку дають вибраним або обробленим даним, які підготовлені для використання як вхідні дані в алгоритми (зазвичай це алгоритми машинного навчання). Feature можуть бути такі речі, як ціна будинку, значення RGB пікселя або, у нашому випадку, подання слова.

Обчислювальна техніка не здатна працювати з текстовими даними без попередньої обробки. Вона потребує приведення тексту до числового вигляду, який є «зрозумілим» для неї.

У цьому випадку стають у нагоді концепції «Bag-of-Words» (BoW; укр., «торба слів») та TF-IDF (від англ. TF — term frequency, IDF — inverse document frequency). BoW та TF-IDF здатні конвертувати текстові речення у числові вектори.

Bag of Words (BoW) Model (Модель «торба слів»)

Модель BoW - найпростіша форма подання тексту в числовому форматі. Як і сам термін, ми можемо представити речення як вектор «торби слів» (рядок чисел).

Недоліки використання моделі BoW

Якщо нові речення містять нові слова, тоді розмір нашого словника збільшується, а отже, збільшується і довжина векторів [36]. Крім того, вектори також міститимуть багато 0, таким чином, утворюється розріджена матриця. Модель не зберігає інформацію про граматику речень та впорядкування слів у тексті.

Term Frequency-Inverse Document Frequency (TF-IDF)

TF (term frequency — частота слова) — відношення числа входжень обраного слова t_i до загальної кількості слів документа (формула 3.23). Таким чином, оцінюється важливість слова в межах обраного документа.

$$TF = \frac{n_i}{\sum_k n_k}, \tag{3.23}$$

де n_i є числом входжень слова в документ, в знаменнику - загальна кількість слів в документі.

IDF (inverse document frequency - обернена частота документа) — інверсія частоти, з якою слово зустрічається в документах колекції (формула 3.24). Використання IDF зменшує вагу широковживаних слів.

$$IDF = log \frac{|D|}{|(d_i) \supset t_i|}, \tag{3.24}$$

де $|D|$ - кількість документів колекції; $|(d_i) \supset t_i|$ - кількість документів, в яких зустрічається слово t_i (коли $n_i \neq 0$).

Більшу вагу TF-IDF отримають слова з високою частотою появи в межах документа та низькою частотою вживання в інших документах колекції (формула 3.25).

$$TFIDF = TF \times IDF. \tag{3.25}$$

Підведемо підсумок: Модель «Bag of Words» лише створює набір векторів, що містять інформацію про кількість слів у документів, в той час, як TF-IDF враховує вагу слова у корпусі документів. BoW простіше реалізувати, однак TF-IDF зазвичай ефективніше працює в моделях машинного навчання.

Побудували частотний графік (рис. 3.21), що показує процентний розподіл тварин по типам, в залежності від діапазону кількості днів, через який тварина була усиновлена з притулку. На графіку бачимо, що процентний розподіл усиновлення

котів на ранніх етапах домінує над усиновленням собак. Тобто цей фактор є значимим для людей, які усиновляють тварин. Коти можуть здаватися людям більш привабливими.

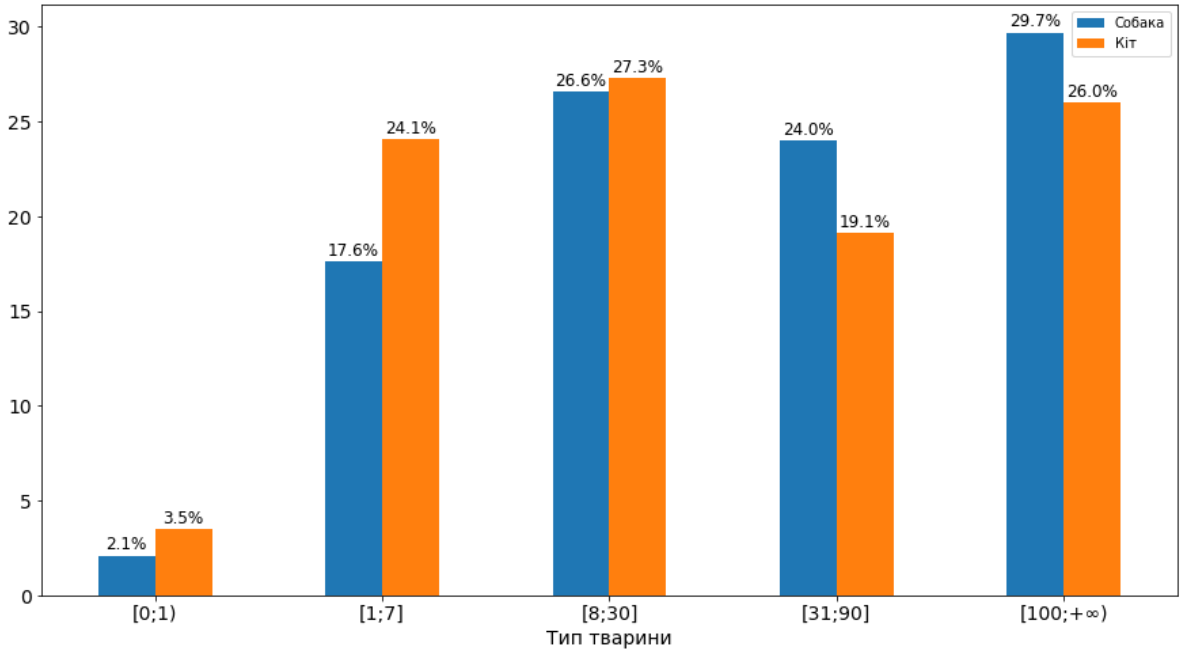

Рисунок 3.20 – Частотний розподіл по швидкості усиновлення тварин

У наборі даних представлено 175 унікальних порід тварин. 86 порід представлено менш, ніж чотирма представниками включно, які складають 1,14% набору даних. Для побудови моделей, які будуть включати породу тварини, від цих даних можна позбутися [37]. Оскільки породи з малою кількістю представників є не ключовими, вони можуть зіпсувати точність моделі. Також множину цих порід можна позначити, як «Інша». Далі визначимо наскільки назва породи корелює зі швидкістю усиновлення. Якщо кореляція слабка, то при побудові моделі враховувати цей фактор не потрібно, бо він не є значущим, а отже, інформацію про породи з невеликою кількістю представників можна залишити у наборі даних.

Побудуємо графік (рис. 3.21) з відносною частотою присутності найпопулярніших п’яти порід собак у наборі даних, що складають 81,2% записів у

множині записів собак. Найпопулярніші породи собак у наборі даних: змішана (72,9%), лабрадор-ретрівер (2,5%), ши-тцу (2,3%), пудель (2,1%), тер'єр (2%).

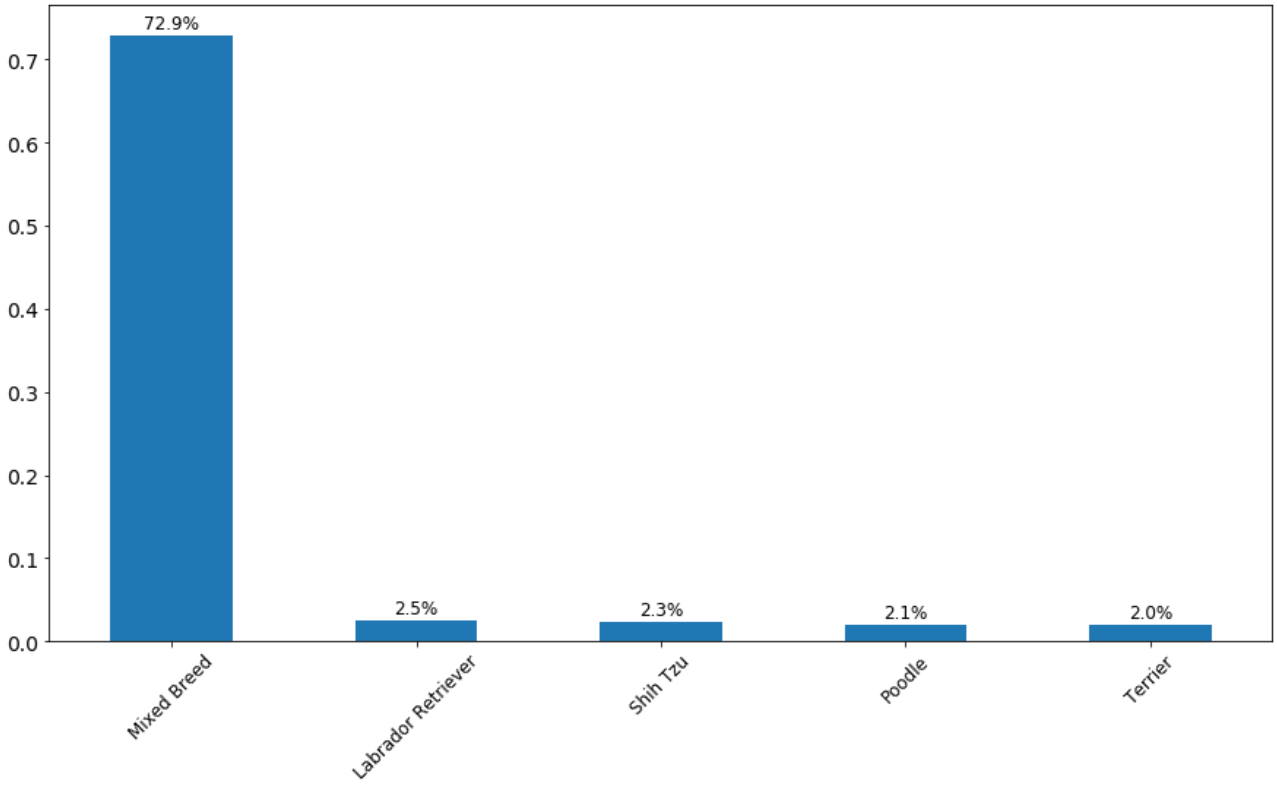

Рисунок 3.21 – Частотний розподіл порід собак у наборі даних

Побудуємо графік (рис. 3.22) з відносною частотою присутності найпопулярніших п’яти порід котів у наборі даних, що складають 84,4% записів у множині записів котів.

Найпопулярніші породи котів у наборі даних: домашня короткошерста (53,0%), домашня середньошерста (18,3%), таббі (5,0%), домашня довгошерста (4,3%), сіамська (3,8%).

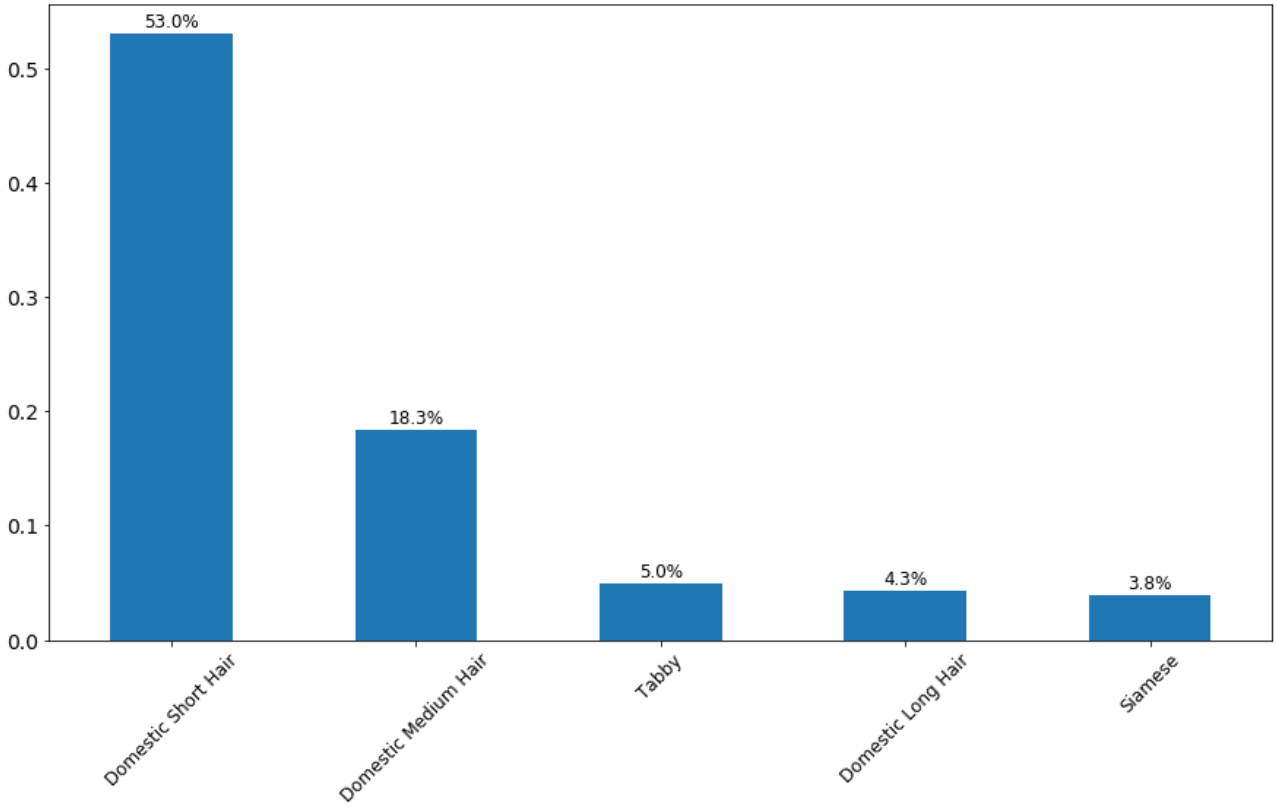

Рисунок 3.22 – Частотний розподіл порід котів у наборі даних

Використовуючи вищенаведену теоретичну інформацію про токенізацію, стемінг та лематизацію реалізуємо дані методи NLP для побудови моделі заснованій на опису тварин, що міститься у колонці «Description».

Протестуємо ефективність стемінгу та лематизації, при цьому видаливши усі стоп-слова та попередньо провівши токенізацію. Позбувшись стоп-слів побудуємо гістограму частот найчастіше використаних слів опису собак (графік 4). 15 слів складають близько 12% використаних слів для опису собак. Найчастіше використані слова: dog (собака), home (дім), puppy (цуценя), please (від слова pleased, задоволений), good (гарний). Перед побудовою графіку (рис. 3.23) слова було приведено до іменникової форми, за допомогою лематизації.

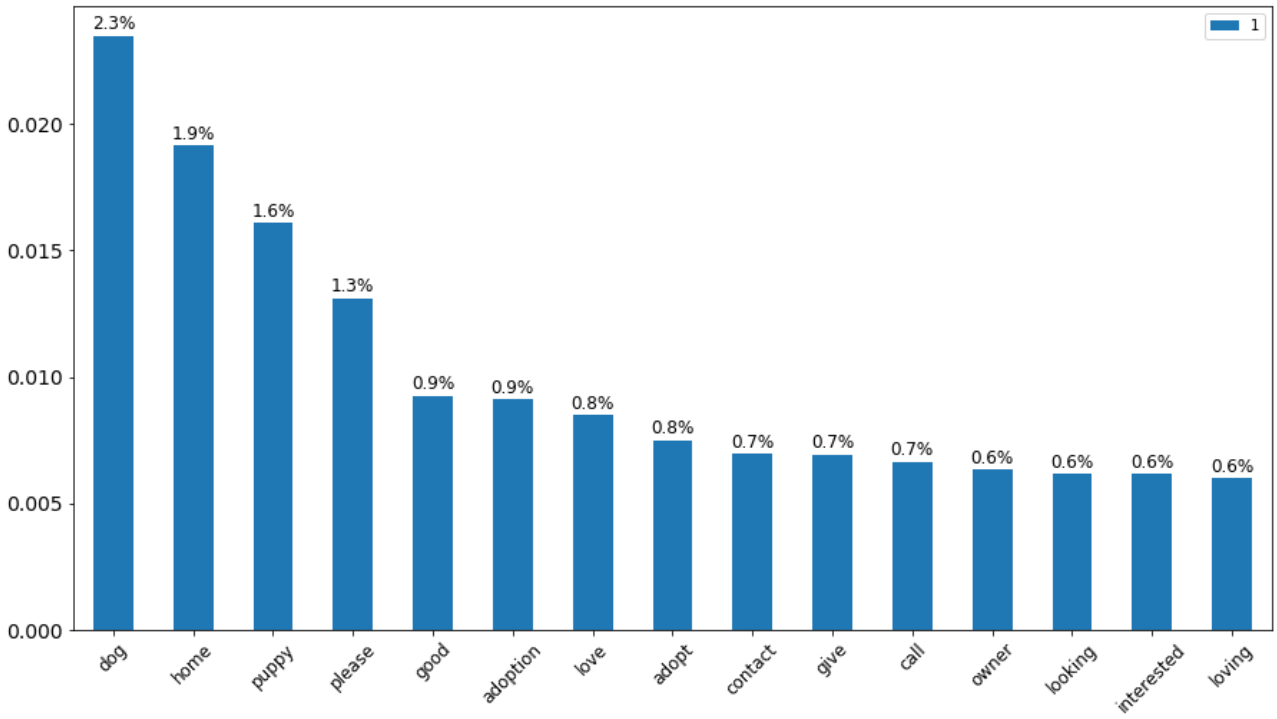

Рисунок 3.23 – Найчастіші за використанням слова опису собак

По тому ж самому принципу побудуємо гістограму частот найчастіше використаних слів опису котів (рис 3.24). 15 слів складають близько 12% використаних слів для опису котів. Найчастіше використані слова: cat (кіт), kitten (кошеня), home (цуценя), please (від слова pleased, задоволений), love (любити).

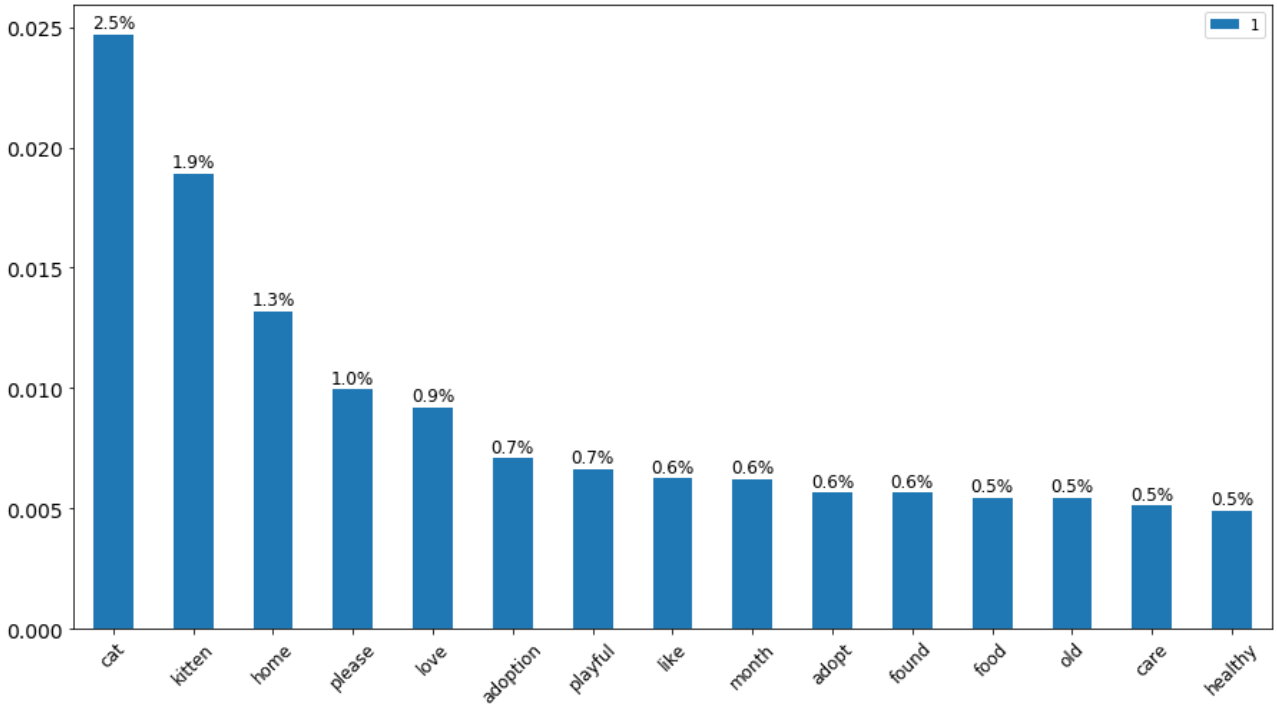

Рисунок 3.24 – Найчастіші за використанням слова опису котів

Матриця невідповідностей

Матриця невідповідностей (confusion matrix) – це таблиця, яка використовується для демонстрації ефективності класифікатора [38]. Зазвичай вона будується для тестового набору даних, котрій відомі базові справжні значення. Аналізуються результати віднесення кожного класу та визначення частки невірно визначених класів. У процесі конструювання таблиці маємо справу з кількома ключовими метриками, що грають важливу роль машинному навчанні. Розглянемо вигляд матриці на прикладі бінарного класифікатору, коли вихідна змінна має два можливі значення: 0 та 1.

Матриця невідповідностей складається з чотирьох основних характеристик (значень), які використовуються для визначення показників вимірювання класифікатора. Ці чотири значення (рис. 3.25):

– ІП (істинно позитивний): ІП представляє кількість випадків, які були належним чином класифіковані до позитивного класу.

– ІН (істинно негативний): ІН представляє кількість випадків, які були належним чином класифіковані до негативного класу.
– ХП (хибно позитивний): ХП представляє кількість неправильно класифікованих випадків до позитивного класу. ХП також відомий як помилка типу I.
– ХН (хибно негативний): ХН представляє кількість неправильно класифікованих випадків до негативного класу. ХН також відомий як помилка типу II.

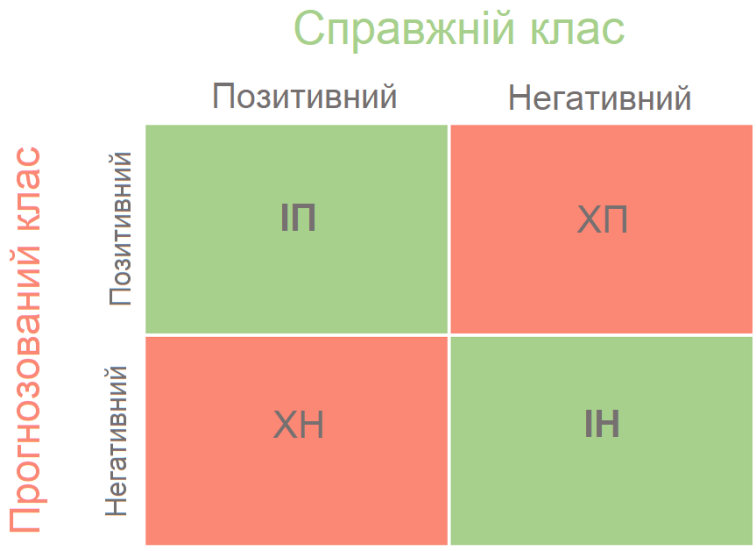

Рисунок 3.25 – Вигляд матриці невідповідностей

Показниками продуктивності алгоритму є accuracy, precision, recall та оцінка F_1, які розраховуються на основі вищезазначених ІП, ІН, ХП та ХН.

Точність (accuracy) алгоритму представляється як відношення правильно класифікованих випадків (ІП+ІН) до загальної кількості випадків (ІП+ІН+ХП+ХН) (формула 3.26).

$$Accuracy = \frac{(\text{ІП}+\text{ІН})}{(\text{ІП}+\text{ІН}+\text{ХП}+\text{ХН})}. \quad (3.26)$$

Точність (precision) алгоритму представлена як відношення правильно класифікованих випадків (ІП) до загальної кількості випадків, що були прогнозовані до позитивного класу (ІП+ХП) (формула 3.27).

$$Precision = \frac{\text{ІП}}{\text{ІП+ХП}}. \quad (3.27)$$

Показник recall визначається як відношення правильно класифікованих випадків (ІП), поділене на загальну кількість випадків, що належить позитивному класу (формула 3.28).

$$Recall = \frac{\text{ІП}}{\text{ІП+ХН}}. \quad (3.28)$$

Оцінка F_1 також відома як $F\ Measure$. Оцінка F_1 визначає рівновагу між $precision$ та $recall$ (формула 3.29).

$$F_1 = \frac{2precision*recall}{precision+recall}. \quad (3.29)$$

Побудуємо матрицю невідповідностей (confusion matrix) для побудованої моделі Bag of Words [39] для стемінгу. Дана матриця інформує про точність передбачення за класами, враховуючи не коректно передбачені класи.

Точність моделі - 0.345. Виходячи з візуалізації бачимо, що найкраще розпізнаються тварини з більшою кількістю днів проведених у притулку.

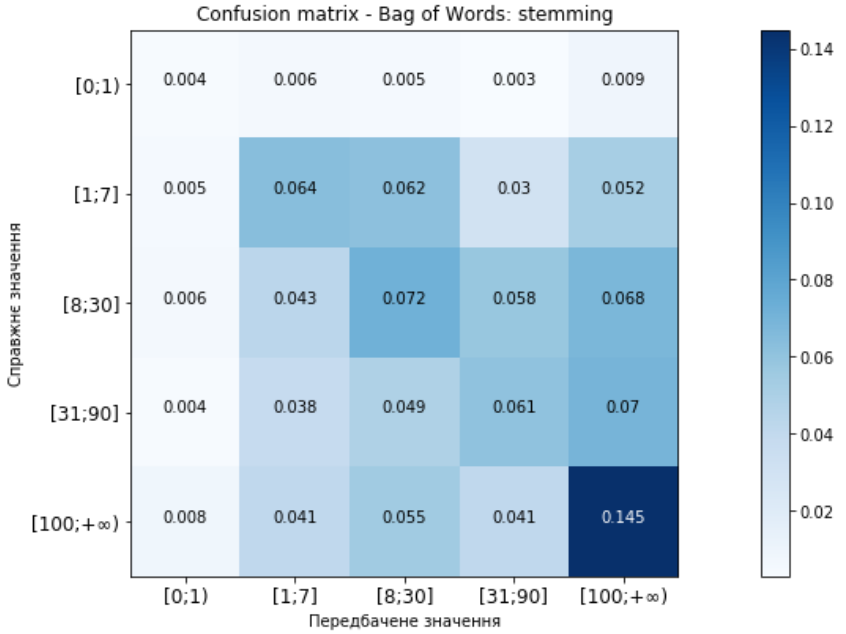

Рисунок 3.26 – Матриця невідповідностей методу Bag of Words для стемінгу

Побудуємо ту ж саму матрицю для побудованої моделі Bag of Words для лематизації (рис. 3.27). Точність моделі - 0.35, що є кращим ніж для тої ж моделі побудованій на лексемах, утворених за допомогою стемінгу.

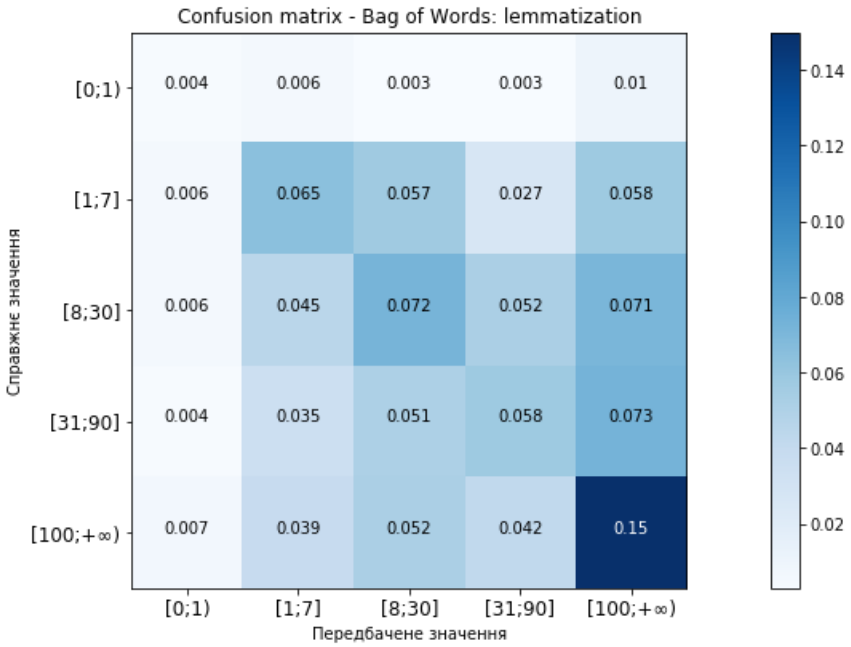

Рисунок 3.27 – Матриця невідповідностей методу Bag of Words для лематизації

Побудували матрицю невідповідностей для побудованої моделі TFidfVectorizer для стемінгу (рис. 3.28). Точність моделі - 0.355, що є кращим результатом, відносно обох моделей Bag of Words.

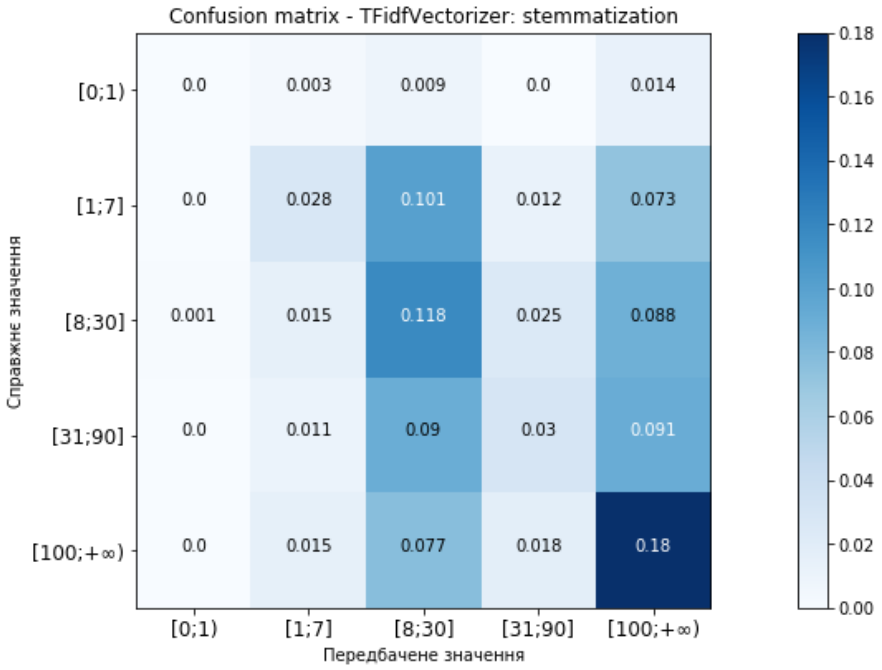

Рисунок 3.28 – Матриця невідповідностей методу TFidfVectorizer для стемінгу

Точність моделі TFidfVectorizer для лематизації - 0.356, що є кращим результатом, відносно обох моделей Bag of Words та моделі TFidfVectorizer для стемінгу (рис. 3.29).

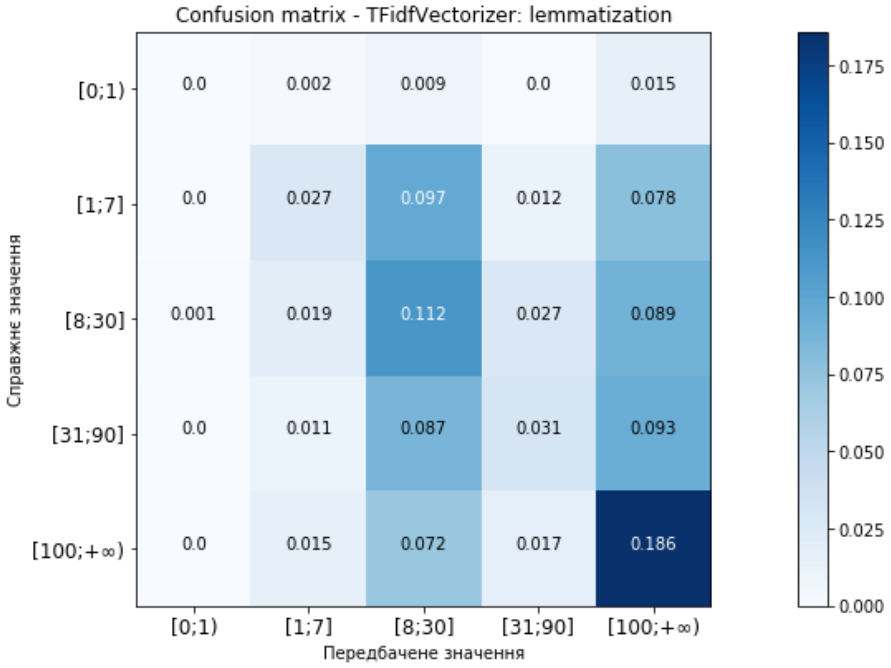

Рисунок 3.29 – Матриця невідповідностей методу TFidfVectorizer для лематизації

Окрім методів Bag of Words та TFidfVectorizer протестуємо LSTM рекурентну нейромережу (RNN). Для початку розберемо, що таке штучна нейронна мережа (ANN).

ANN — розподілена обчислювальна система [40], що має природну концепцію навчанню на попередній інформації, тобто набувати обчислювального досвіду. Вона може надати відповідні рішення для задач, які, як правило, характеризуються нелінійними зв'язками, достатньо високими розмірами, складними, неточними та недосконалими даними, схильних до помилок або відсутністю чітко заявленого математичного рішення чи алгоритму. Ключовою перевагою нейронних мереж є те, що модель системи може бути побудована з наявних даних.

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN) - мережі, що містять зворотні зв'язки та дозволяють зберігати інформацію [41]. На рисунку 3.30 фрагмент нейронної мережі A приймає вхідне значення x_t і повертає значення h_t. Наявність зворотного зв'язку дозволяє передавати інформацію від одного кроку мережі до іншого.

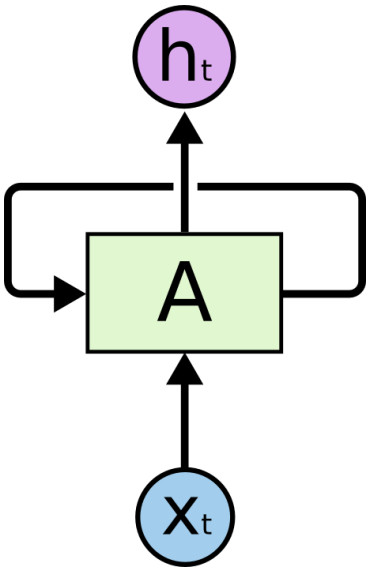

Рисунок 3.30 – Фрагмент рекурентної нейронної мережі

Зворотні зв'язку надають рекурентним нейронних мереж загадковість. Проте, вони не сильно відрізняються від звичайних нейронних мереж. Рекурентну мережу можна розглядати, як кілька копій однієї і тієї ж мережі, кожна з яких передає інформацію подальшої копії. Що станеться, якщо ми розгорнемо зворотний зв'язок, зображено на рисунку 8.

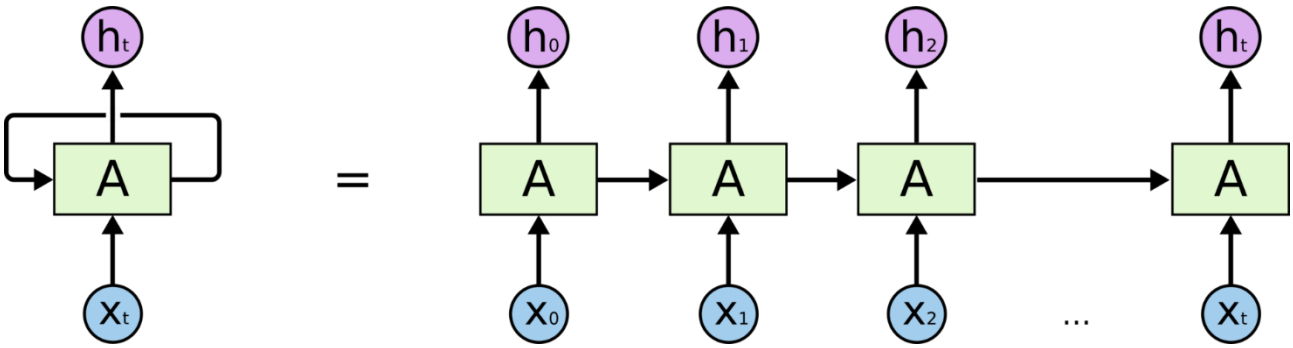

Рисунок 3.31 – Рекурентна нейронна мережа в розгортці

RNN нагадують ланцюжок, це говорить про те, що вони тісно пов'язані з послідовностями і списками. RNN – найбільш природна архітектура нейронних мереж для роботи з даними такого типу.

За останні кілька років RNN з неймовірним успіхом застосували до цілого ряду завдань: розпізнавання мови, мовне моделювання, переклад, розпізнавання зображень.

Чимала роль у цих успіхах належить LSTM - незвичайній модифікації рекурентної нейронної мережі, яка на багатьох задачах значно перевершує стандартну версію. Майже всі вражаючі результати RNN досягнуті саме за допомогою LSTM.

Проблема довгострокових залежностей

Одна з привабливих ідей RNN полягає в тому, що вони потенційно вміють пов'язувати попередню інформацію з поточною задачею.

Іноді для виконання поточного завдання нам необхідна тільки недавня інформація. Розглянемо, наприклад, мовну модель, яка намагається передбачити наступне слово на підставі попередніх. Якщо ми хочемо передбачити останнє слово в реченні "хмари пливуть по небу", нам не потрібен більш широкий контекст; в цьому випадку досить очевидно, що останнім словом буде "небо". У цьому випадку, коли дистанція між актуальною інформацією і місцем, де вона знадобилася, невелика, RNN можуть навчитися використанню інформації з минулого (рис. 3.32).

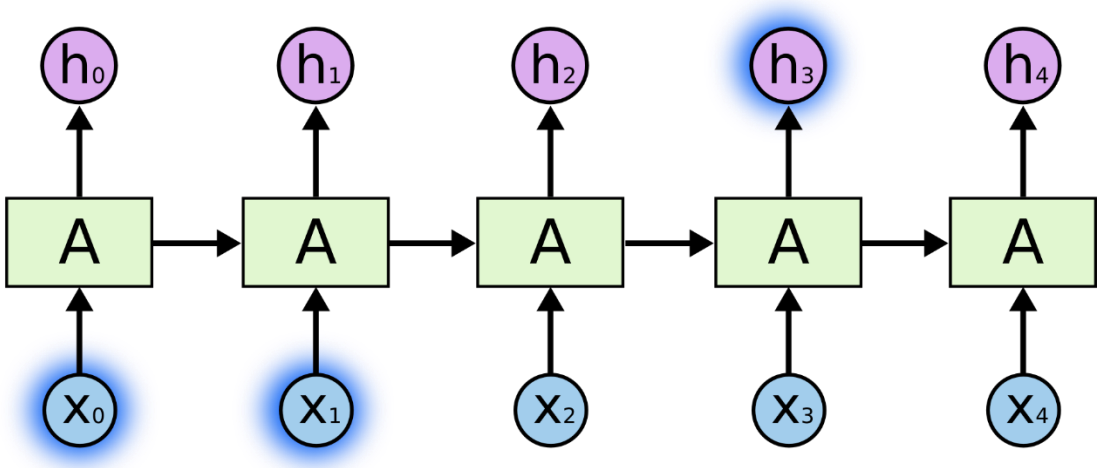

Рисунок 3.32 – Використання RNN попередньої інформації

Але бувають випадки, коли нам необхідно більше контексту. Припустимо, ми хочемо передбачити останнє слово в тексті "Я виріс у Франції... Я вільно розмовляю французькою". Найближчий контекст передбачає, що останнім словом буде назва мови, але щоб встановити, якої саме мови, нам потрібен контекст Франції з більш віддаленої дистанції. Таким чином, розрив між актуальною інформацією і точкою її застосування може стати дуже великим.

На жаль, у міру зростання цієї відстані, RNN втрачають здатність зв'язувати інформацію (рис. 3.33).

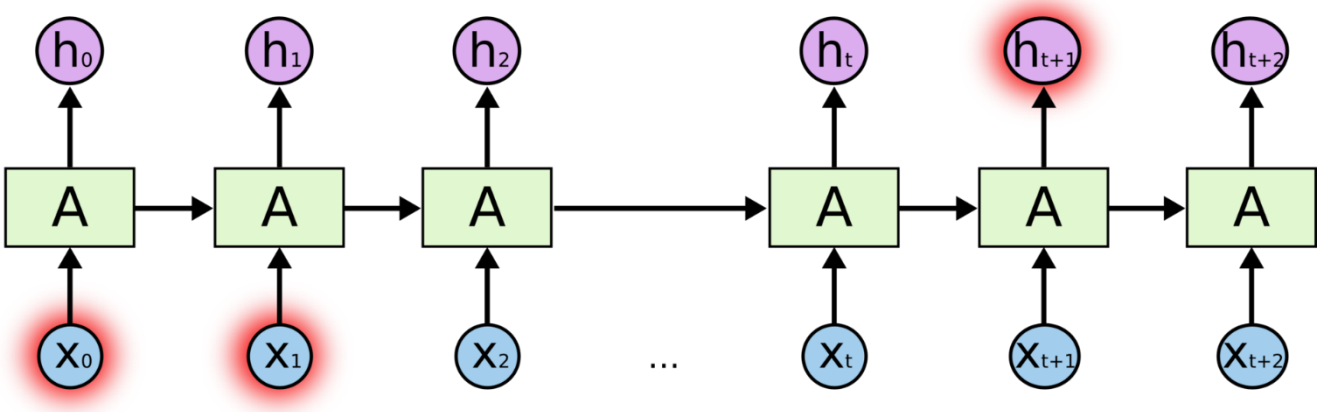

Рисунок 3.33 – Втрата контексту інформації RNN з минулого

В теорії проблеми з обробкою довготривалих залежностей у RNN бути не повинно. Людина може акуратно підбирати параметри мережі для вирішення завдання такого типу. На жаль, на практиці навчити RNN цим параметрам неможливо. Цю проблему детально дослідив Зепп Хохрайтер [42]. Він знайшов беззаперечні причини, за якими це може бути досить складно. LSTM таких проблем не мають.

Довга короткострокова пам'ять (Long short-term memory; LSTM) - особливий різновид архітектури рекурентних нейронних мереж, здатна до навчання довготривалим залежностям. Вони були представлені Зеппом Хохрайтер і Юргеном Шмідхубер (Jürgen Schmidhuber) в 1997 році [43], а потім вдосконалені і популярно

викладені в роботах багатьох інших дослідників. Вони прекрасно вирішують цілий ряд різноманітних завдань і в даний час широко використовуються.

LSTM розроблені спеціально, щоб уникнути проблеми довготривалої залежності. Запам'ятовування інформації на довгі періоди часу - це їх звичайна поведінка, а не щось, чого вони насилу намагаються навчитися.

Будь-яка рекурентна нейронна мережа має форму ланцюжка повторюваних модулів нейронної мережі. У звичайній RNN структура одного такого модуля дуже проста, наприклад, він може являти собою один шар з функцією активації tanh (гіперболічний тангенс) (рис. 3.34).

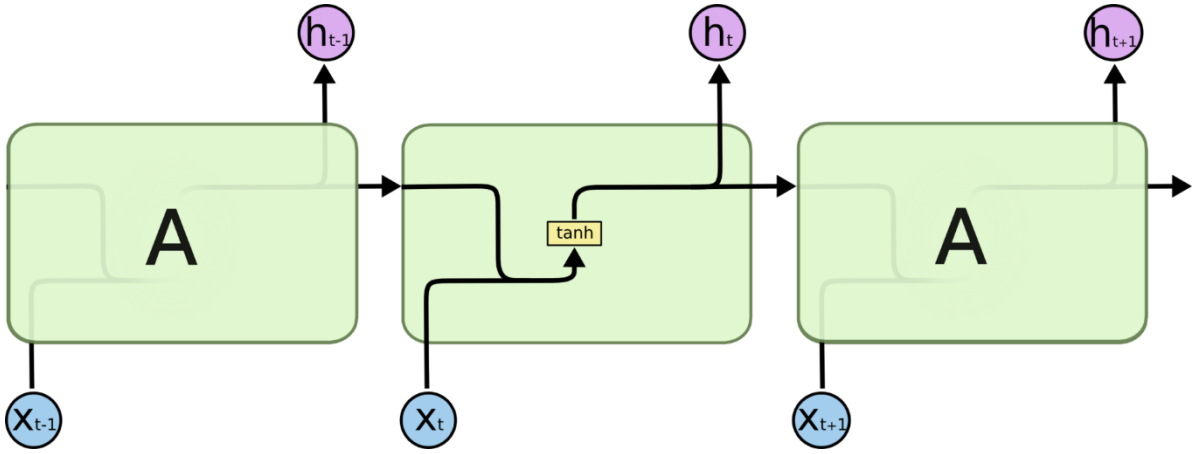

Рисунок 3.34 – Періодичний модуль в стандартній RNN

Структура LSTM також нагадує ланцюжок, але модулі виглядають інакше. Замість одного шару нейронної мережі вони містять цілих чотири (рис. 3.35), і ці шари взаємодіють особливим чином.

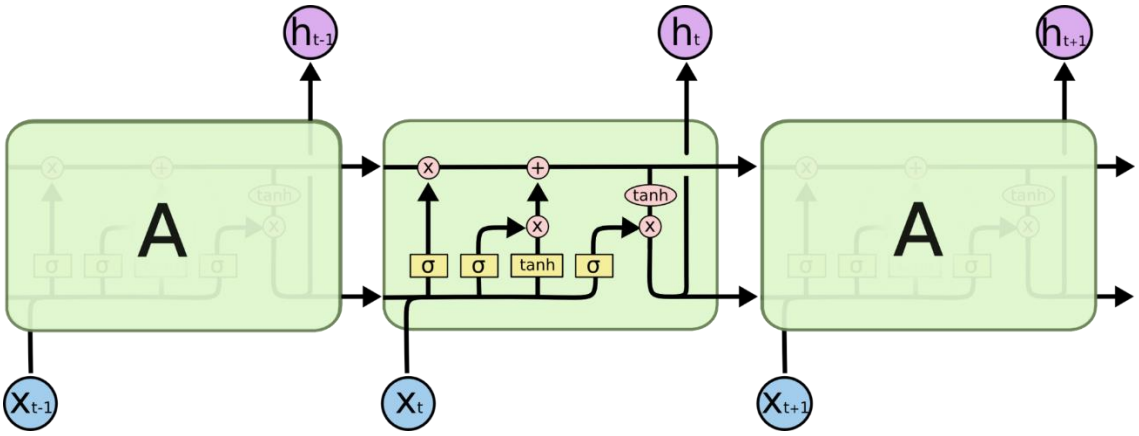

Рисунок 3.35 – Періодичний модуль в LSTM мережі

Побудували LSTM-модель для стемінгу та лематизації, з різною кількістю епох навчання (табл. 3.2).

Таблиця 3.2 – Ефективність LSTM-моделі для стемінгу та лематизації

Кількість епох	LSTM: стемінг	LSTM: лематизація
5	0.3307	0.3335
10	0.3388	0.3311
15	0.3279	0.3287
20	0.3390	0.3350

В обох випадках, з кількістю епох рівною 20, маємо незначну перевагу точності моделі.

Результати усіх застосованих методів текстової класифікації наведені в таблиці 3.3.

Таблиця 3.3 – Результати методів текстової класифікації

	Техніка перетворення слів	
Метод	Стемінг	Лематизація
bag-of-words	0.355	0.356
TFidfVectorizer	0.345	0.35
LSTM	0.3390	0.3350

Найкращу точність показав алгоритм bag-of-words. Для даного алгоритму методи обробки слів стемінгу та лематизації мають майже однаковий результат.

Висновки до розділу

Для знаходження оптимального результату були задіяні моделі логістичної регресії, наївного Баєсу, методу опорних векторів, дерев рішень, випадкового лісу та гранично випадкових лісів. Для класифікації фотографій домашніх тварин за привабливістю можна використано технологію emotion detection. Класифікацію зображень реалізовано за допомогою нейронної мережі архітектури Inception-v3. У наборі даних присутній текстовий опис тварин. Для трансформації тексту використані методи NLP.

4 ОЦІНКА ЕФЕКТИВНОСТІ МОДЕЛЕЙ

Для кінцевого набору даних побудовано 6 моделей: логістичну регресію, наївний баєсів класифікатор, метода опорних векторів, дерева рішень, випадковий ліс та гранично випадкові ліси.

Перед навчанням було зроблено препроцесінг даних, додано поле «LenDescr», що є значенням довжини текстового опису тварини. Поля «RescuerID», «PetID», «id», «Name», «pet_id» видалено з тренувального набору даних, оскільки вони не несуть цінної для навчання моделей інформації.

Для кожної моделі буде 2 варіанти реалізації: з використанням даних з текстового опису тварини і розпізнавання привабливості тварин з фотокарток та без цих двох додаткових кроків.

1.1 Логістична регресія

Матриця невідповідностей для варіанту моделі без використанням даних з текстового опису тварини і розпізнавання привабливості тварин з фотокарток (рис. 4.1). Матрицю побудовано на основі тренувальних даних.

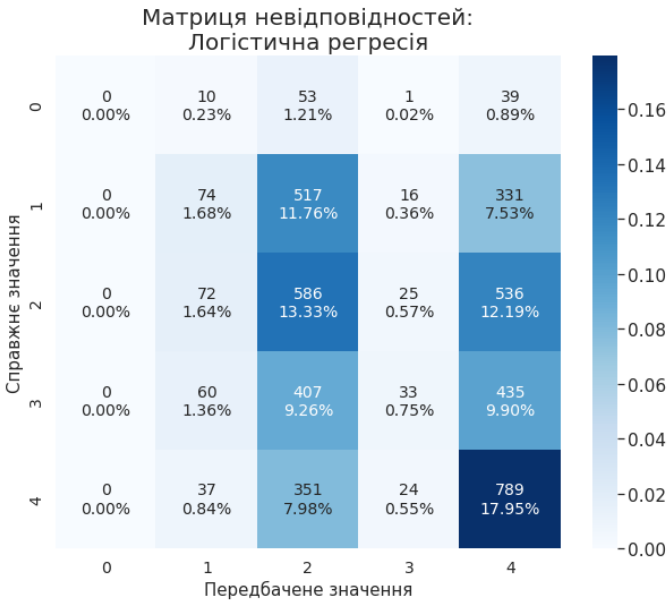

Рисунок 4.1 – Матриця невідповідностей для логістичної регресії

Таблиця 4.1 – Звіт точності класифікації логістичної регресії

class	precision	recall	f1-score	support
0	0	0	0	103
1	0.29	0.08	0.12	938
2	0.31	0.48	0.37	1219
3	0.33	0.04	0.06	935
4	0.37	0.66	0.47	1201
accuracy			0.34	4396
macro avg	0.26	0.25	0.21	4396
weighted avg	0.32	0.34	0.27	4396

Отримана точність моделі логістичної регресії на тестових даних - 0.3371.

Матриця невідповідностей для варіанту моделі з використанням даних з текстового опису тварин і розпізнавання привабливості тварин з фотокарток (рис. 4.2).

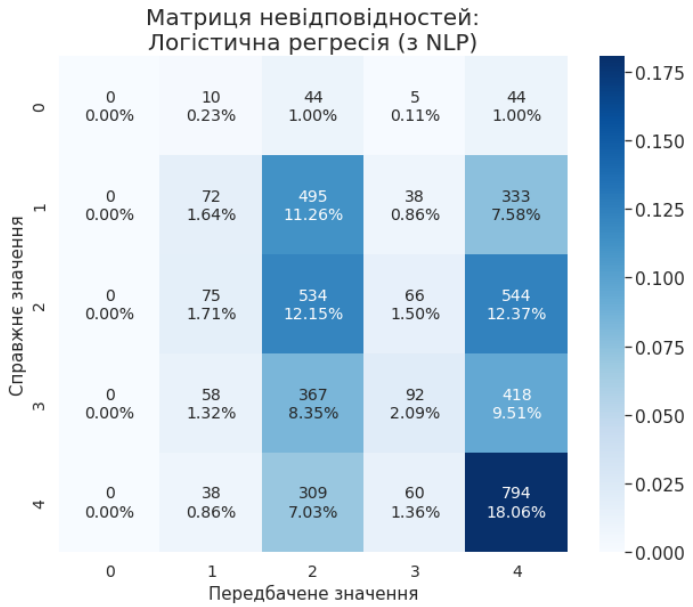

Рисунок 4.2 – Матриця невідповідностей для логістичної регресії

Таблиця 4.2 – Звіт точності класифікації логістичної регресії

class	precision	recall	f1-score	support
0	0	0	0	103
1	0.28	0.08	0.12	938
2	0.31	0.44	0.36	1219
3	0.35	0.1	0.15	935
4	0.37	0.66	0.48	1201
accuracy			0.34	4396
macro avg	0.26	0.25	0.22	4396
weighted avg	0.32	0.34	0.29	4396

Отримана точність логістичної регресії для варіанту моделі з використанням даних з текстового опису тварин і розпізнавання привабливості тварин з фотокарток на тестових даних - 0.3394.

Використання даних з текстового опису тварин і розпізнавання привабливості тварин з фотокарток у побудованій моделі покращило її точність на 0,68%.

1.2 Наївний Баєс

Матриця невідповідностей для варіанту моделі без використання даних з текстового опису тварини і розпізнавання привабливості тварин з фотокарток (рис. 4.3).

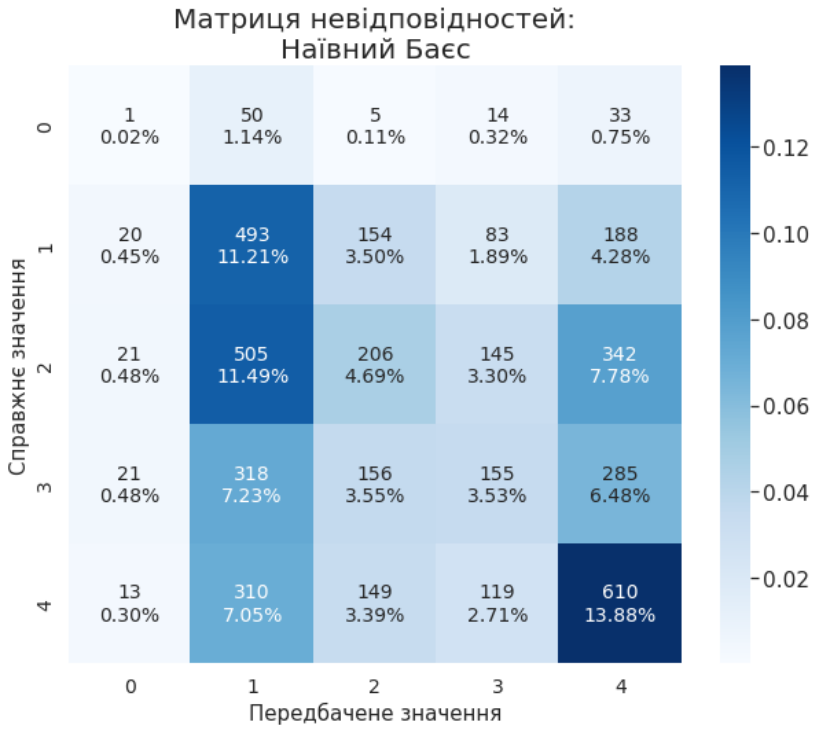

Рисунок 4.3 - Матриця невідповідностей для наївного баєсового класифікатора

Таблиця 4.3 – Звіт точності наївного баєсового класифікатора

class	precision	recall	f1-score	support
0	0.01	0.01	0.01	103
1	0.29	0.53	0.38	938
2	0.31	0.17	0.22	1219
3	0.3	0.17	0.21	935
4	0.42	0.51	0.46	1201
accuracy			0.33	4396
macro avg	0.27	0.28	0.26	4396
weighted avg	0.33	0.33	0.31	4396

Отримана точність наївного баєсового класифікатора для варіанту моделі без використання даних текстового опису тварин і розпізнавання привабливості тварин з фотокарток на тестових даних - 0.3332.

Матриця невідповідностей для варіанту моделі з використанням даних текстового опису тварин і розпізнавання привабливості тварин з фотокарток (рис. 4.4).

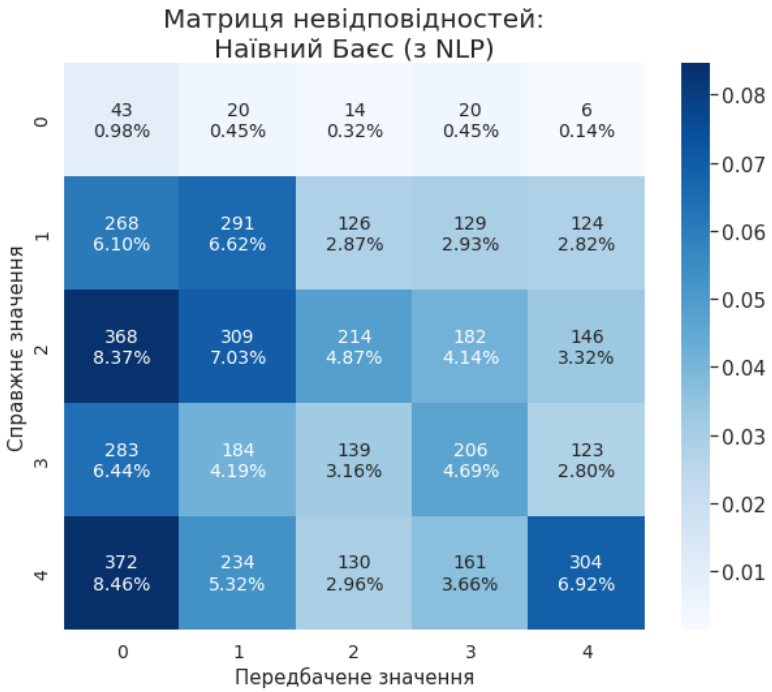

Рисунок 4.4 - Матриця невідповідностей для наївного баєсового класифікатора

Таблиця 4.4 – Звіт точності наївного баєсового класифікатора

class	precision	recall	f1-score	support
0	0.03	0.42	0.06	103
1	0.28	0.31	0.29	938
2	0.34	0.18	0.23	1219
3	0.3	0.22	0.25	935
4	0.43	0.25	0.32	1201
accuracy			0.24	4396
macro avg	0.28	0.28	0.23	4396
weighted avg	0.34	0.24	0.27	4396

Отримана точність наївного баєсового класифікатора для варіанту моделі з використанням даних з текстового опису тварин і розпізнавання привабливості тварин з фотокарток на тестових даних - 0.2407.

1.3 Метод опорних векторів

Матриця невідповідностей методу опорних векторів для варіанту моделі без використання даних з текстового опису тварини і розпізнавання привабливості тварин з фотокарток (рис. 4.5).

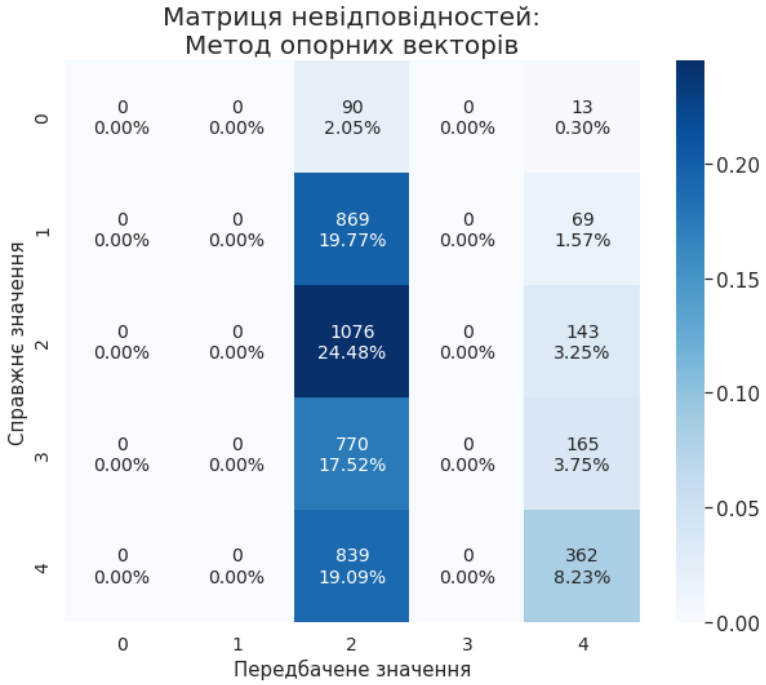

Рисунок 4.5 – Матриця невідповідностей для методу опорних векторів

Таблиця 4.5 – Звіт точності методу опорних векторів

class	precision	recall	f1-score	support
0	0.00	0.00	0.00	103
1	0.00	0.00	0.00	938
2	0.30	0.88	0.44	1219

class	precision	recall	f1-score	support
3	0.00	0.00	0.00	935
4	0.48	0.30	0.37	1201
accuracy			0.33	4396
macro avg	0.16	0.24	0.16	4396
weighted avg	0.21	0.33	0.22	4396

Отримана точність методу опорних векторів для варіанту моделі без використання даних текстового опису тварин і розпізнавання привабливості тварин з фотокарток на тестових даних - 0.3271.

Матриця невідповідностей методу опорних векторів для варіанту моделі з використанням даних текстового опису тварин і розпізнавання привабливості тварин з фотокарток (рис. 4.6).

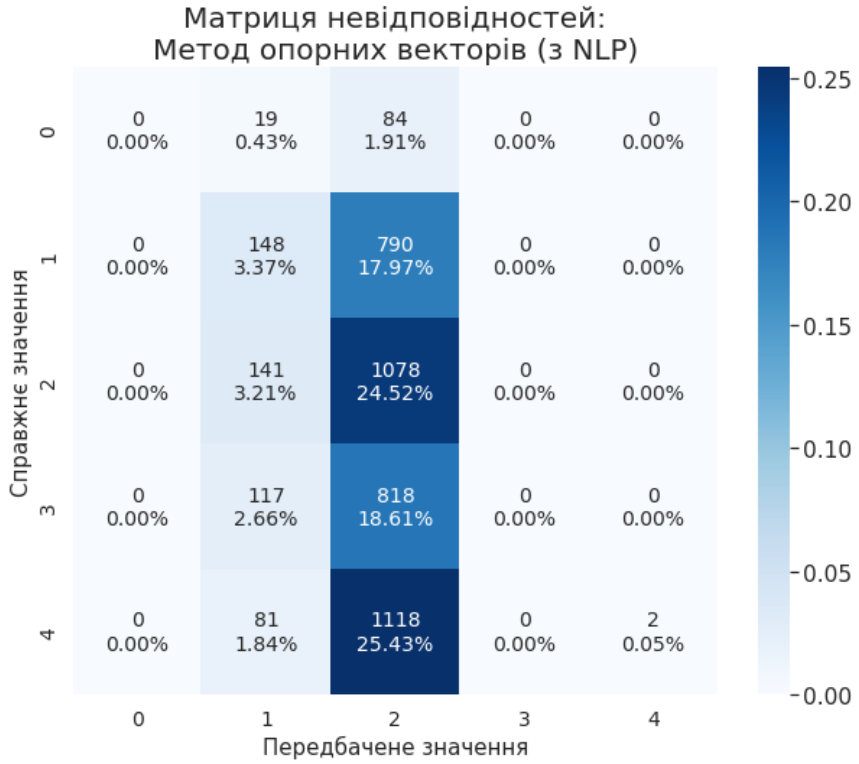

Рисунок 4.6 - Матриця невідповідностей для методу опорних векторів

Таблиця 4.6 – Звіт точності методу опорних векторів

class	precision	recall	f1-score	support
0	0.00	0.00	0.00	103
1	0.29	0.16	0.20	938
2	0.28	0.88	0.42	1219
3	0.00	0.00	0.00	935
4	1.00	0.00	0.00	1201
accuracy			0.28	4396
macro avg	0.31	0.21	0.13	4396
weighted avg	0.41	0.28	0.16	4396

Отримана точність методу опорних векторів для варіанту моделі з використанням даних з текстового опису тварин і розпізнавання привабливості тварин з фотокарток на тестових даних - 0.2793.

Для дерев рішень, випадкового лісу та гранично випадкових лісів було визначено оптимальні параметри моделей за допомогою пошуку по ґратці.

Пошук по ґратці (або варіація параметрів) - метод налаштування, який намагається обчислити оптимальні значення гіперпараметрів [44]. Це вичерпний пошук, який виконується за заданими значеннями параметрів моделі. Завдання пошуку по ґратці може заощадити час, зусилля та ресурси.

Пошук по ґратці використовує різну комбінацію всіх вказаних гіперпараметрів та їх значень, обчислює продуктивність для кожної комбінації та обирає найкраще значення для гіперпараметрів (рис. 4.7).

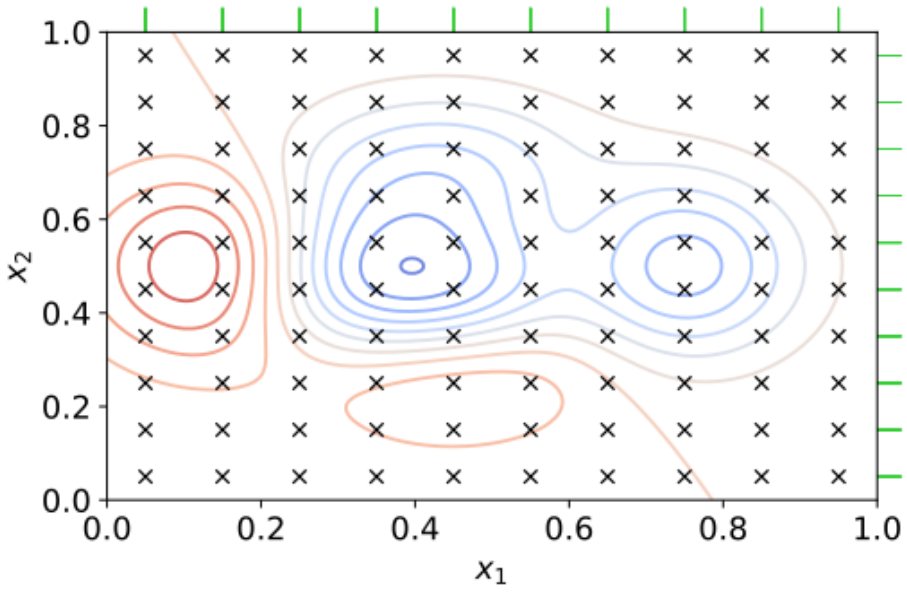

Рисунок 4.7 - Пошук по ґратці за двома параметрами

Цей метод є досить поширеним, тому бібліотека мови Python Scikit-learn має цю функціональність, вбудовану в GridSearchCV. CV (Cross-Validation) розшифровується як перехресна перевірка, яка є ще одним методом оцінки та покращення моделі машинного навчання.

1.4 Дерево рішень

Найкраща точність для дерев рішень без використання даних привабливості тварин отримана при максимальній глибині дерева – 5, random_state – будь-якому значенні (табл. 4.7). Точність – 37,53%.

Таблиця 4.7 - Пошук по ґратці дерева рішень

random_state / max_depth	0	1	2	3
3	35.4%	35.4%	35.4%	35.4%
5	37.5%	37.5%	37.5%	37.5%
8	37.1%	37.1%	37.1%	37.1%

random_state / max_depth	0	1	2	3
10	36.0%	36.0%	36.0%	36.0%
12	35.0%	34.7%	34.8%	35.0%

Отримана точність методу дерев рішень для варіанту моделі без використання даних текстового опису тварин і розпізнавання привабливості тварин з фотокарток на тестових даних - 0.3735.

Матриця невідповідностей методу дерева рішень для варіанту моделі без використання даних текстового опису тварин і розпізнавання привабливості тварин з фотокарток (рис. 4.8).

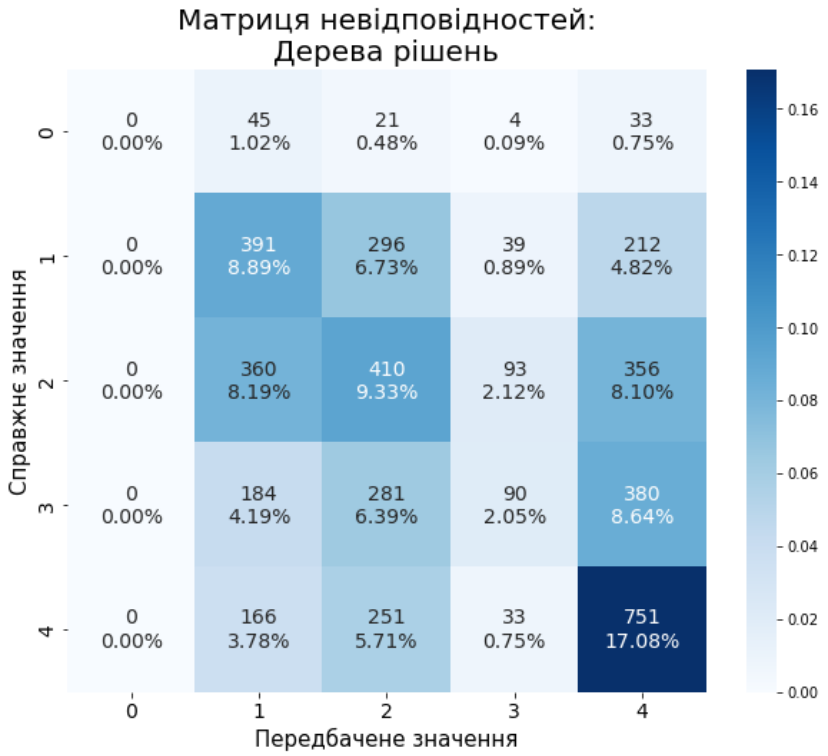

Рисунок 4.8 - Матриця невідповідностей для методу дерев рішень

Таблиця 4.8 – Звіт точності дерева рішень

class	precision	recall	f1-score	support
0	0.00	0.00	0.00	103
1	0.34	0.42	0.38	938
2	0.33	0.34	0.33	1219
3	0.35	0.10	0.15	935
4	0.43	0.63	0.51	1201
accuracy			0.37	4396
macro avg	0.29	0.21	0.27	4396
weighted avg	0.36	0.37	0.34	4396

Найкраща точність для дерев рішень з використанням даних привабливості тварин отримана при максимальній глибині дерева – 5, random_state – 1 (табл. 4.9). Точність – 38,8%.

Таблиця 4.9 - Пошук по ґратці дерева рішень

random_state / max_depth	0	1	2	3
3	36.97%	36.97%	36.97%	36.97%
5	38.78%	38.80%	38.78%	38.78%
8	37.97%	37.93%	37.95%	37.95%
10	37.14%	37.26%	37.16%	37.05%
12	36.31%	36.36%	36.38%	36.37%

Матриця невідповідностей дерева рішень для варіанту моделі з використанням даних текстового опису тварин і розпізнавання привабливості тварин з фотокарток (рис. 4.9).

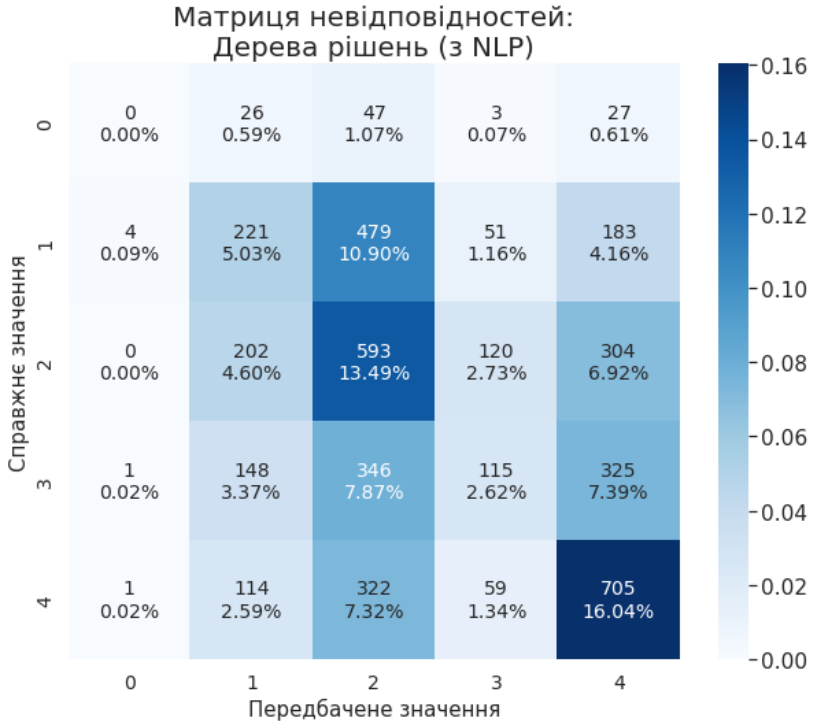

Рисунок 4.9 - Матриця невідповідностей для методу дерев рішень

Таблиця 4.10 – Звіт точності дерева рішень

class	precision	recall	f1-score	support
0	0.00	0.00	0.00	103
1	0.35	0.44	0.39	938
2	0.33	0.35	0.34	1219
3	0.35	0.10	0.15	935
4	0.43	0.63	0.51	1201
accuracy			0.37	4396
macro avg	0.29	0.21	0.27	4396
weighted avg	0.36	0.37	0.34	4396

1.5 Випадковий ліс

Найкраща точність для випадкового лісу без використання даних привабливості тварин отримана при максимальній глибині дерева – 12, кількості дерев, що використовуватимуться у випадковому лісі – 200 (табл. 4.11). Точність – 40,88%.

Таблиця 4.11 – Пошук по ґратці випадкового лісу

n_estimators / max_depth	100	150	200	250
3	36.06%	36.26%	36.30%	36.40%
5	38.92%	38.21%	38.40%	38.45%
8	39.78%	39.78%	40.07%	40.50%
10	39.73%	39.26%	39.50%	39.92%
12	40.35%	39.97%	40.88%	40.59%

Матриця невідповідностей методу випадкового лісу для варіанту моделі без використання даних текстового опису тварин і розпізнавання привабливості тварин з фотокарток (рис. 4.10).

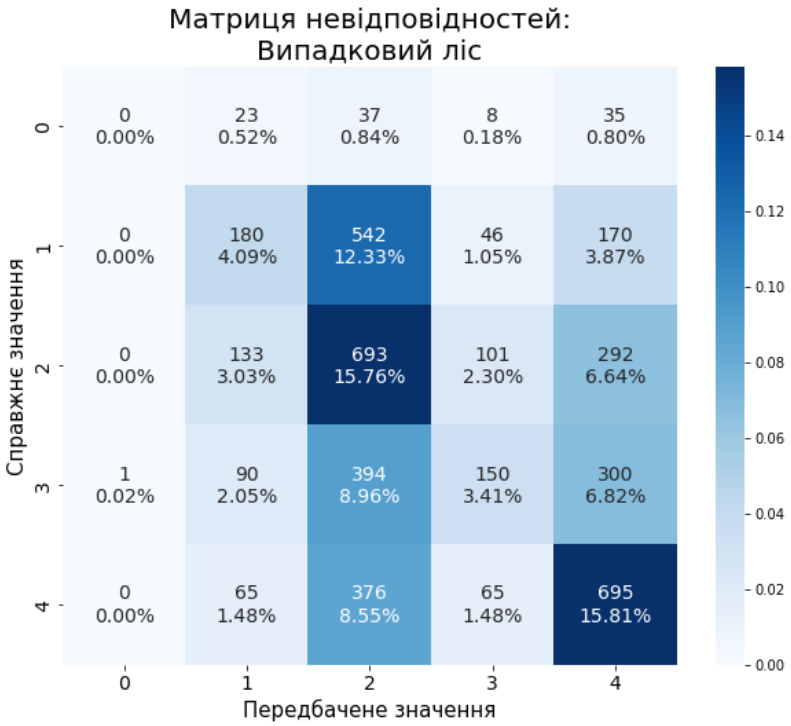

Рисунок 4.10 - Матриця невідповідностей для методу випадкового лісу

Таблиця 4.12 – Звіт точності випадкового лісу

class	precision	recall	f1-score	support
0	0.00	0.00	0.00	103
1	0.39	0.32	0.35	938
2	0.37	0.47	0.41	1219
3	0.43	0.19	0.26	935
4	0.48	0.66	0.56	1201
accuracy			0.39	4396
macro avg	0.33	0.33	0.32	4396
weighted avg	0.41	0.42	0.40	4396

Найкраща точність для випадкового лісу з використанням даних привабливості тварин отримана при максимальній глибині дерева – 12, кількості дерев, що використовуватимуться у випадковому лісі – 150 (табл. 4.13). Точність – 41,53%.

Таблиця 4.13 - Пошук по ґратці випадкового лісу

n_estimators / max_depth	100	150	200	250
3	36.39%	36.53%	36.38%	36.38%
5	38.85%	38.62%	38.87%	38.74%
8	40.64%	40.73%	40.64%	40.69%
10	40.85%	41.06%	41.07%	41.18%
12	41.28%	41.53%	41.42%	41.40%

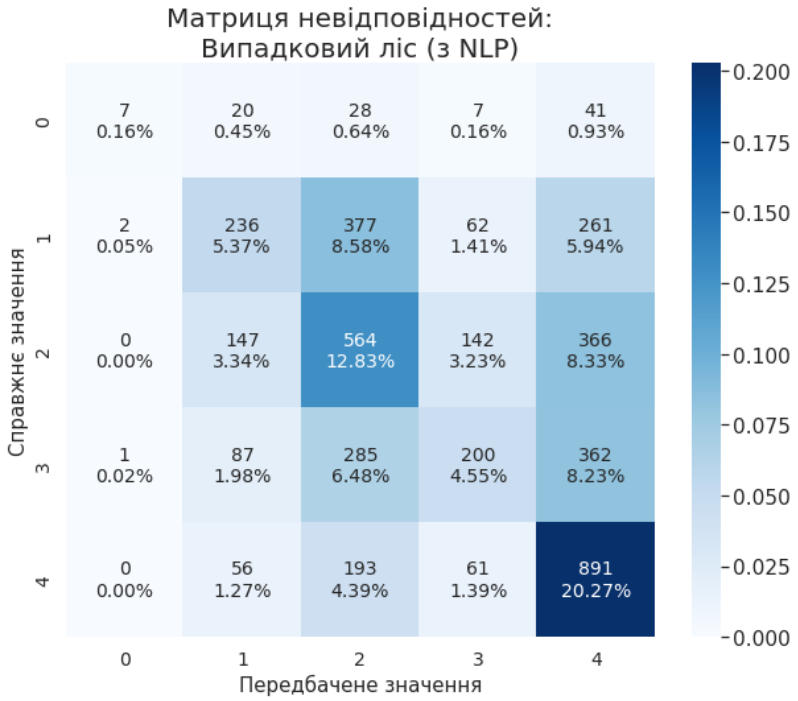

Рисунок 4.11 - Матриця невідповідностей для методу випадкового лісу

Використання даних з текстового опису тварин і розпізнавання привабливості тварин з фотокарток у побудованій моделі покращило її точність на 1,59%.

1.6 Гранично випадкові ліси

Найкраща точність для гранично випадкових лісів без використання даних привабливості тварин отримана при максимальній глибині дерев – 10, кількості дерев, що використовуватимуться у гранично випадкових лісах – 150 (табл. 4.14). Точність – 38,73%.

Таблиця 4.14 – Пошук по ґратці гранично випадкових лісів

n_estimators / max_depth	100	150	200	250
3	34.40%	34.06%	34.16%	34.16%
5	36.21%	36.06%	36.16%	36.30%
8	37.40%	37.73%	38.07%	37.73%
10	38.64%	38.73%	38.59%	38.49%

n_estimators / max_depth	100	150	200	250
12	38.30%	38.35%	37.97%	38.30%

Матриця невідповідностей методу гранично випадкових лісів для варіанту моделі без використання даних текстового опису тварин і розпізнавання привабливості тварин з фотокарток (рис. 4.12).

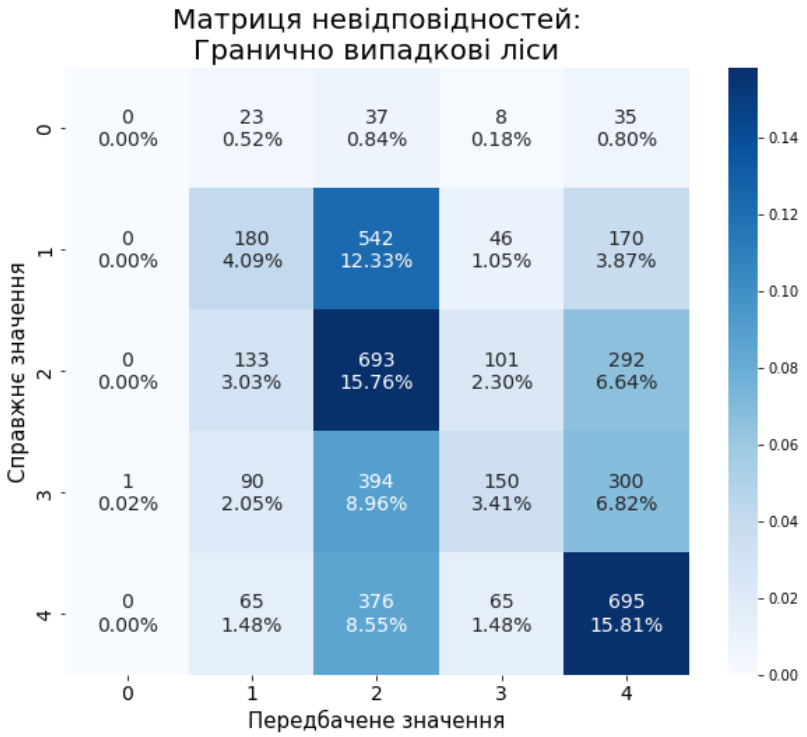

Рисунок 4.12 - Матриця невідповідностей для методу гранично випадкових лісів

Таблиця 4.15 – Звіт точності гранично випадкових лісів

class	precision	recall	f1-score	support
0	0.00	0.00	0.00	103
1	0.37	0.19	0.25	938
2	0.34	0.57	0.43	1219
3	0.41	0.16	0.23	935
4	0.47	0.58	0.52	1201

class	precision	recall	f1-score	support
accuracy			0.39	4396
macro avg	0.32	0.30	0.28	4396
weighted avg	0.39	0.39	0.36	4396

Найкраща точність для гранично випадкових лісів з використанням даних привабливості тварин отримана при максимальній глибині дерев – 12, кількості дерев, що використовуватимуться у гранично випадкових лісах – 200 (табл. 4.14). Точність – 39,05%.

Таблиця 4.16 – Пошук по ґратці гранично випадкових лісів

n_estimators / max_depth	100	150	200	250
3	35.40%	35.16%	34.99%	34.96%
5	36.18%	36.23%	36.33%	36.24%
8	37.20%	37.46%	37.60%	37.53%
10	38.47%	38.51%	38.62%	38.63%
12	39.02%	38.87%	39.05%	38.88%

Матриця невідповідностей методу гранично випадкових лісів для варіанту моделі з використанням даних текстового опису тварин і розпізнавання привабливості тварин з фотокарток (рис. 4.14).

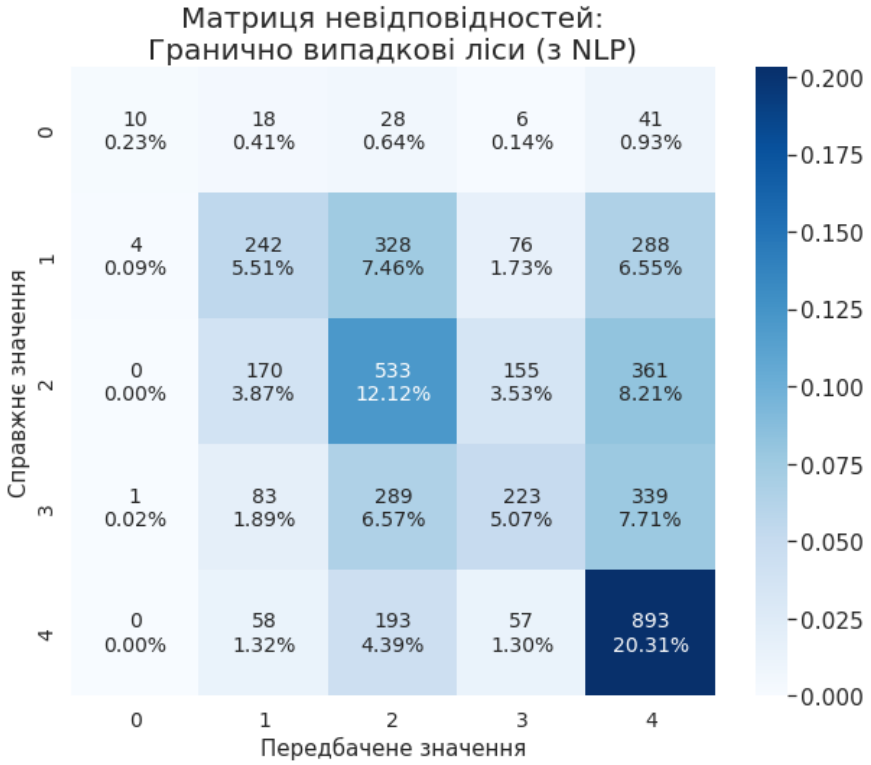

Рисунок 4.14 - Матриця невідповідностей для методу гранично випадкових лісів

Використання даних з текстового опису тварин і розпізнавання привабливості тварин з фотокарток у побудованій моделі покращило її точність на 0,83%.

Висновки до розділу

Ансамблеві методи машинного навчання показали кращі результати у вирішенні даної задачі відносно методів логістичної регресії, наївного баєсового класифікатору та методу опорних векторів. Для визначення найкращого методу серед ансамблевих був задіяний пошук по ґратці (варіація параметрів), що є традиційним методом оптимізації гіперпараметрів.

Серед ансамблевих методів найкращий показник точності на тестових даних показав метод випадкового лісу, що становив – 41,53%. При цьому максимальна глибина дерева – 12, кількість дерев у випадковому лісі - 150. Модель випадкового лісу буде задіяна в інформаційній системі.

5 ER-ДІАГРАМА

5.1 Опис схеми бази

У таблицях 5.1 – 5.3 наведено опис таблиць бази даних.

Таблиця 5.1 – Опис таблиці Users

Назва поля	Опис поля	Тип даних	Примітка
UserID	Ідентифікатор користувача ІС	int	Первинний ключ
TelegramUserID	Ідентифікатор користувача у Telegram	bigint	

Таблиця 5.2 – Опис таблиці Pets

Назва поля	Опис поля	Тип даних	Примітка
PetID	Ідентифікатор тварини	int	Первинний ключ
UserID	Ідентифікатор користувача ІС	int	Зовнішній ключ до таблиці User
CreatedOn	Дата та час створення запису тварини в системі	timestamp	
Type	Тип тварини	varchar	
Age	Вік тварини у місяцях	int	
Breed	Порода тварини	varchar	
Gender	Стать тварини	varchar	
Color	Колір шерсті	varchar	
MaturitySize	Розмір тварини у зрілому віці	varchar	
FurLength	Довжина хутра	varchar	
Vaccinated	Наявність вакцинації	varchar	
Sterilized	Наявність стерилізації	varchar	

Назва поля	Опис поля	Тип даних	Примітка
Health	Стан здоров’я	varchar	
Fee	Вартість привласнення тварини	int	
PhotoAmt	Кількість фотокарток	int	
Description	Текстовий опис тварини	text	
AttractivenessLevel	Підсумковий рівень привабливості тварини по фотографіям	int	

Таблиця 5.3 – Опис таблиці Images

Назва поля	Опис поля	Тип даних	Примітка
ImageID	Ідентифікатор фотографії	int	Первинний ключ
PetID	Ідентифікатор тварини	int	Зовнішній ключ до таблиці Pet

Висновок до розділу

База даних містить 3 таблиці: Users, Pets та Images. Таблиця Users містить інформацію про користувача системи. Таблиця Pets містить інформацію про тварин, що в подальшому оброблюється ІС для видачі результату. Таблиця Images містить інформацію про фотографії тварин.

6 ВИБІР ТА ОБҐРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ

Для реалізації програмного продукту було використано мову програмування Python та фреймворк TensorFlow. Були використані наступні бібліотеки numpy, pandas, matplotlib, scikit-learn, opencv, nltk, tensorflow, keras мови Python.

6.1 Python

Python (Пайтон) – високорівнева мова програмування загального призначення, спрямована на прискорення роботи розробника та має код, що легко читається. Синтаксис ядра мови мінімалістичний, в той же час стандартна бібліотека містить великий набір корисних функцій [44]. Python підтримує функціональне, структурне та об`єктно-орієнтоване програмування. Основними архітектурними особливостями є динамічна типізація, автоматичне управління пам`яттю, підтримка багатопоточних обчислень, механізм обробки виключень, високорівневі структури даних. Підтримується розподілення програми на модулі, які, в свою чергу, можуть інтегруватися в пакети.

Python на сьогодні є найпопулярнішою мовою програмування для досліджень та розробок у машинному навчанні. За даними Google Trends [45], інтерес до Python для машинного навчання займає передові позиції у порівнянні з іншими мовами ML (рис. 1), такими як R, Java, Scala, Julia тощо, які значно відстають.

Причини, через які Python найкраще підходить для машинного навчання: 1. Python простий у використанні. Простота використання Python є однією з головних причин, чому він настільки популярний для машинного навчання. Синтаксис легко читається. Розробники можуть зосередитись на вирішенні проблеми, а не на технічних нюансах мови. На додаток до цього, Python є надзвичайно ефективним. Це дозволяє розробникам використовувати меншу кількість рядків коду. Код Python простий для людського розуміння, що робить його ідеальним інструментом для реалізації моделей машинного навчання;

Python має велику кількість бібліотек та фреймворків Python має сотні різних бібліотек та фреймворків, якими можуть користуватися розробники. Ці бібліотеки та фреймворки економлять час, що, в свою чергу, робить Python ще більш популярним.

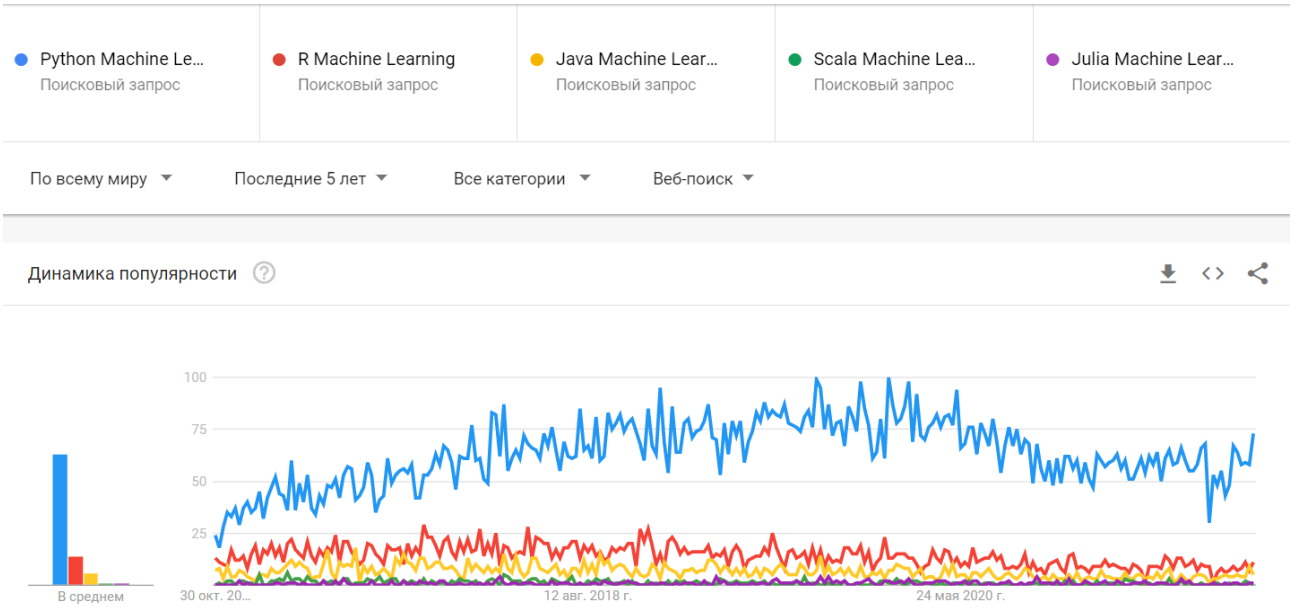

Рисунок 6.1 – Динаміка популярності використання мов програмування для машинного навчання

Розглянемо бібліотеки мови Python, які є найпопулярнішими для побудови моделей штучного інтелекту та машинного навчання [47] та відповідають задачам інформаційної системи:

Keras – бібліотека з відкритим кодом, яка особливо зосереджена на експериментах з глибокими нейронними мережами;

TensorFlow – безкоштовна бібліотека програмного забезпечення, яка використовується для багатьох програм машинного навчання, таких як нейронні мережі;

Scikit-learn – безкоштовна бібліотека програмного забезпечення для машинного навчання, що пов'язане з різними алгоритмами класифікації, регресії та кластеризації. Також Scikit-learn можна використовувати разом із NumPy та SciPy.

NLTK – є провідним інструментом для обробки людської мови. Він надає набір бібліотек обробки тексту для класифікації, токенізації, стемінгу, тегів, парсингу та семантичного аналізу, є обгорткою для бібліотек NLP. NLTK призначений для підтримки досліджень і викладання в обробці природньої мовлення або близько пов'язаних областях, включаючи емпіричну лінгвістику, когнітивну науку, штучний інтелект, пошук інформації та машинне навчання [50]. NLTK успішно використовується як навчальний інструмент, як індивідуальний навчальний інструмент, а також як платформа для створення прототипів і побудови дослідницьких систем.

6.2 Google Colab

Google Colab - це безкоштовне середовище, яке повністю працює у хмарі. Colab підтримує багато популярних і високорівневих бібліотек машинного навчання, які можна легко завантажити у редактор.

Особливості Google Colab:

- Написання коду на Python
- Імпорт зовнішніх наборів даних
- Інтеграція PyTorch, TensorFlow, Keras, OpenCV
- Безкоштовний хмарний сервіс із безкоштовними GPU та TPU

6.3 Telegram Bot API

Для взаємодії користувача з ІС реалізуємо її у вигляді Telegram Bot. Для реалізації боту скористаємося Telegram Bot API. Bot API — це інтерфейс на основі HTTP [45]. Бот містить функціонал, який потрібен для взаємодії з користувачем:

- Дозволяє взаємодіяти з ботом за допомогою команд.
- Надає можливість вводити на вхід текст, завантажувати зображення для подальшої ІС обробки.

– Містить клавіатуру для взаємодії з ботом.

6.4 PostgreSQL

PostgreSQL – потужна система об’єктно-реляційних баз даних з відкритим вихідним кодом, яка використовує та розширює мову SQL у поєднанні з багатьма функціями, які безпечно зберігають і масштабують найскладніші робочі навантаження даних [46].

PostgreSQL постачається з багатьма функціями, які допомагають розробникам створювати програми, адміністраторам захищати цілісність даних і створювати відмовостійкі середовища, а також допомагають керувати даними незалежно від того, наскільки великий чи малий набір даних.

Висновки до розділу

Для побудови та тестування ефективності моделей машинного навчання використано середовище Google Colab, що дозволяє використовувати хмарні обчислення. Інформаційна система та дослідження в Google Colab, реалізовані на мові Python. Бібліотеки numpy та pandas дозволяють швидко обробляти великі масиви даних.

Для обробки зображень використовується бібліотека OpenCV, для обробки тексту – nltk, моделі машинного навчання будуються за допомогою бібліотеки scikit-learn.

Інформаційна система реалізована у вигляді Telegram боту та використовує Telegram Bot API. Для мови Python функціонал даного API реалізований у бібліотеці telegram. Для збереження даних використано СКБД PostgreSQL.

7 РЕАЛІЗАЦІЯ БІЗНЕС-ЛОГІКИ СИСТЕМИ

7.1 Polling та long polling

Чат-боти повинні отримувати сповіщення від серверу моментально. Вони не можуть перевіряти оновлення кожну секунду, це не є ефективним. Більшість відповідей від сервера будуть неінформативними, на кшталт «Нові повідомлення відсутні».

Такий підхід, коли раз на n секунд опитується сторонній сервіс, називається polling (рис. 7.1).

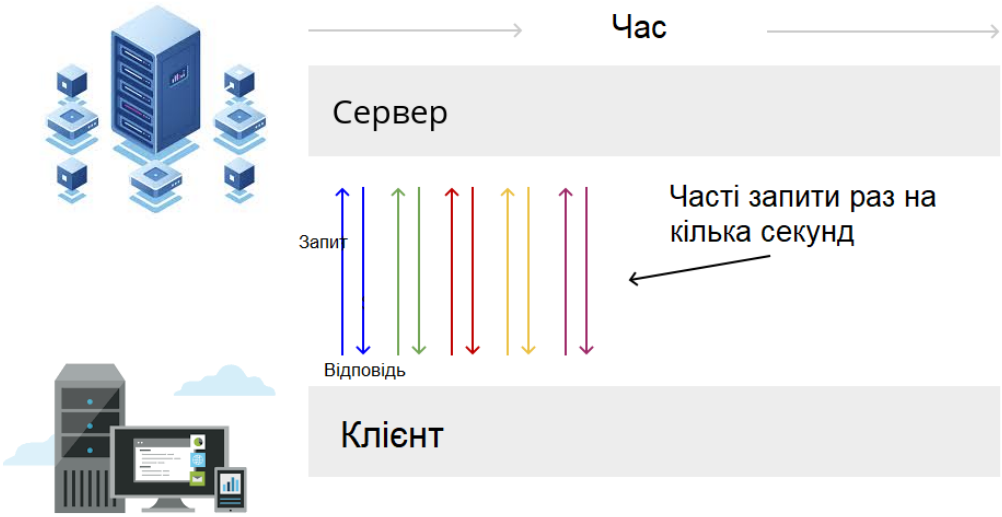

Рисунок 7.1 – Підхід обробки запитів типу polling

Щоб заощадити на ресурсах, використовуватимемо long polling. Влаштований він так само, як і polling з однією відмінністю: сервер відповідає довше. Взагалі, при long polling сервер відповідає у двох випадках: або тому, що надійшло нове повідомлення, або тому, що з'єднання пора розривати.

Кожен запит має timeout — час, протягом якого потрібно дати відповідь. Якщо за цей час на запит відповідь не отримана, вважається, що сервер не відповість взагалі. Тому сервер орієнтується на значення timeout і вирішує так:

1. Якщо за цей час у не з'явиться оновлення для клієнта, сервер відповість йому, що їх немає.
2. Якщо оновлення з'являться, сервер відправить оновлення відразу, не чекаючи timeout.

Щоб реалізувати long polling на стороні клієнта, потрібно виставити великий timeout: 30 чи 60 секунд.

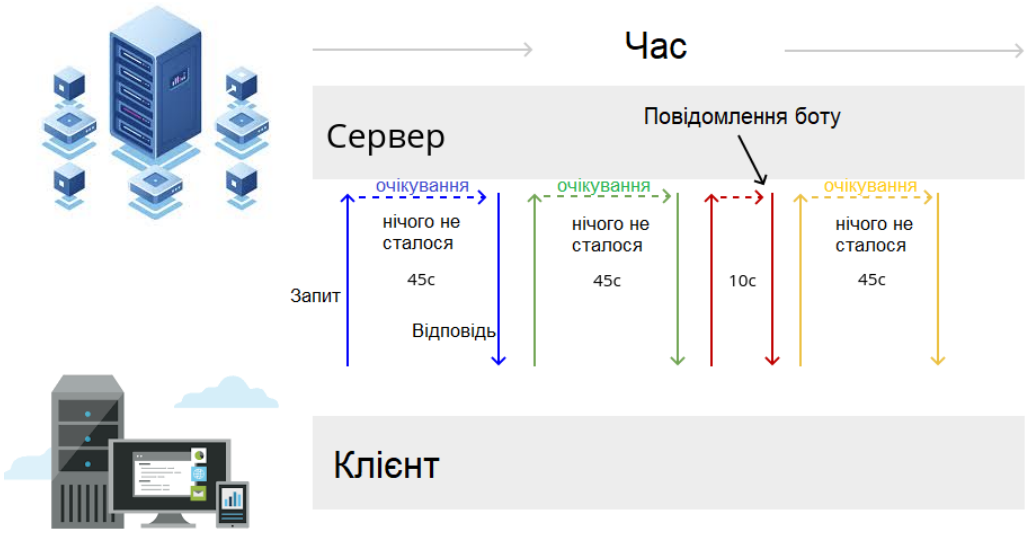

Рисунок 7.2 - Підхід обробки запитів типу long polling

На рисунку 7.3 зображено реалізацію polling з боку клієнта на Python.

```python
from time import sleep
import requests

while True:
    response = requests.get("http://someurl.com")
    for message in response:
        bot.answer(message)
        sleep(1)
```

Рисунок 7.3 – Реалізація polling на Python

На рисунку 7.4 зображено реалізацію long polling з боку клієнта на Python.

```python
import requests

while True:
    response = requests.get("http://someurl.com", timeout=60)
    for message in response:
        bot.answer(message)
```

Рисунок 7.4 – Реалізація long polling на Python

7.2 Функціонал інформаційної системи

Для прогнозування періоду перебування домашнього улюбленця було реалізовано Telegram бот. При першому використанні боту з'являється опис, що містить інформацію щодо функціоналу системи (рис. 7.5).

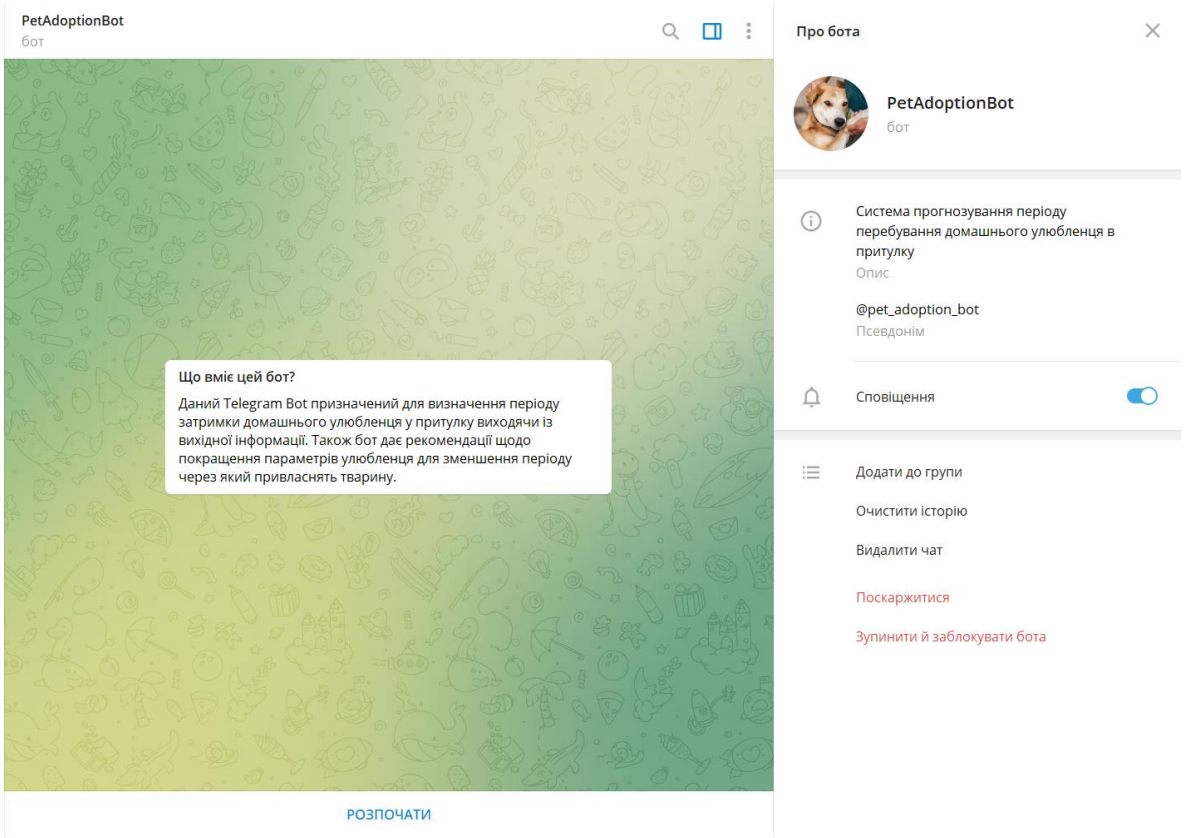

Рисунок 7.5 – Стартова сторінка Telegram боту

Після запуску боту він інформує щодо майбутніх дій користувача та надсилає запит щодо типу тварини (рис. 7.6). Пропонується 2 варіанти: собака або кіт. Комунікація з ботом у наступних запитах виглядає схожим чином: питання – варіанти відповідей. Бот задає користувачу 13 запитань щодо параметрів тварини, що стосуються типу тварини, віку, породи, статі, кольору хутра, розміру тварини у зрілому віці, розміру хутра, наявності вакцинації та стерилізації, стану здоров'я, вартості привласнення; бот просить завантажити фотографію тварини та ввести текстовий опис.

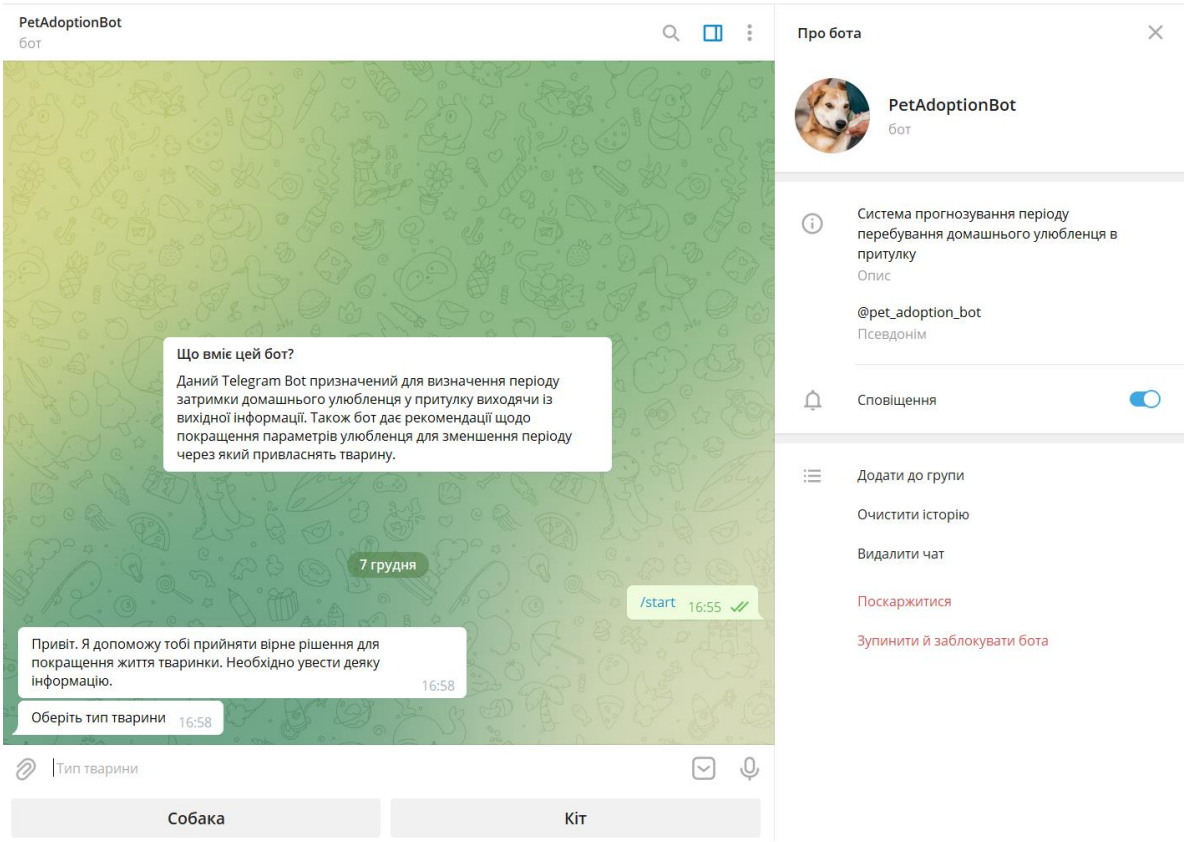

Рисунок 7.6 – Telegram бот після запуску

Послідовна комунікація з ботом із запланованим сценарієм можливих питань реалізована за допомогою обробника розмов ConversationHandler() бібліотеки python-telegram-bot.

Обробник розмов ConversationHandler() представляє собою список, що містить три стани:

– точку входу в розмову - список обробників повідомлень, які своєю функцією зворотного виклику запускають комунікацію.
– етапи розмови - словник, який як ключ містить етап розмови, а значення ключа представляє список обробників повідомлень цього етапу. Ключі етапів повинні повертати функції зворотного виклику повідомлень цих етапів. Перемикання між етапами відбувається залежно від того, який ключ повертає функція обробника конкретного повідомлення.
– точку виходу з розмови – список обробників повідомлень, функції зворотного виклику яких, для виходу з розмови, мають повертати return ConversationHandler.END.

В обробнику ConversationHandler() міститься логіка розмови (подібно до оператора вибору switch ... case), яка будується та управляється за допомогою обробників повідомлень і значень, що повертаються функціями зворотних викликів.

На рисунку 7.7 подано реалізацію розмови для інформаційної системи. Точкою входу у розмову є команда «/start». Етапи діалогу містяться у словнику «states». Для кожного параметра тварини створено власний етап.

```
conv_handler = ConversationHandler(
    entry_points=[
        CommandHandler('start', start_handler)
        ,MessageHandler(Filters.photo, photo_handler, pass_user_data=True)
    ],
    states = {
        TYPE: [MessageHandler(Filters.text, type_handler, pass_user_data=True)],
        AGE: [MessageHandler(Filters.text, age_handler, pass_user_data=True)],
        BREED: [MessageHandler(Filters.text, breed_handler, pass_user_data=True)],
        GENDER: [MessageHandler(Filters.text, gender_handler, pass_user_data=True)],
        COLOR: [MessageHandler(Filters.text, color_handler, pass_user_data=True)],
        MATURITYSIZE: [MessageHandler(Filters.text, maturity_size_handler, pass_user_data=True)],
        FURLENGTH: [MessageHandler(Filters.text, furlength_handler, pass_user_data=True)],
        VACCINATION: [MessageHandler(Filters.text, vaccination_handler, pass_user_data=True)],
        STERILIZATION: [MessageHandler(Filters.text, sterilization_handler, pass_user_data=True)],
        HEALTH: [MessageHandler(Filters.text, health_handler, pass_user_data=True)],
        FEE: [MessageHandler(Filters.text, fee_handler, pass_user_data=True)],
        PHOTO: [MessageHandler(Filters.photo, photo_handler, pass_user_data=True)],
        DESCRIPTION: [MessageHandler(Filters.text, description_handler, pass_user_data=True)],
    },
    fallbacks=[
        CommandHandler('cancel', cancel_handler),
    ]
)
```

Рисунок 7.7 – Реалізація обробника ConversationHandler

При запиті на уведення породи тварини бот надсилає файл, що містить список порід та їх номера (рисунок 7.8).

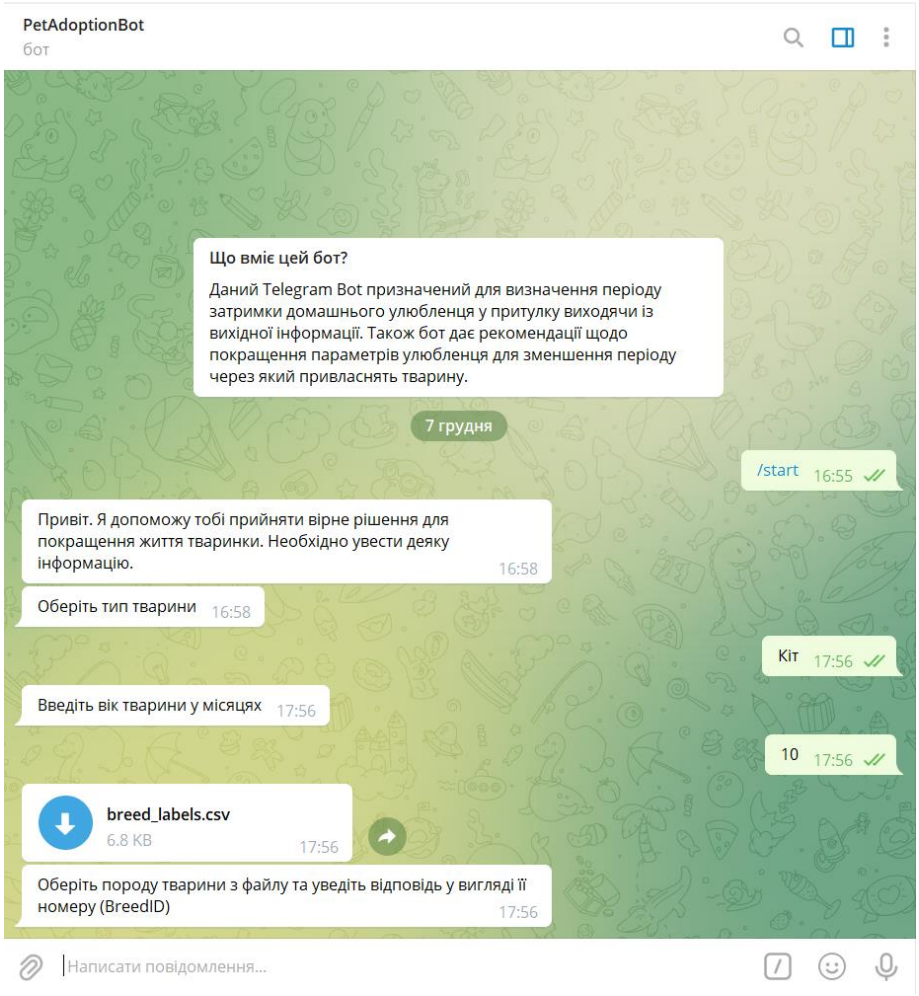

Рисунок 7.8 – Надсилання ботом файлу «breed_labels.csv» з найменуванням порід

Файл «breed_labels.csv» зі списком порід тварин виглядає таким чином (рис. 7.9). Колонка «BreedID» є номером, яким необхідно дати відповідь боту, «Type» - тип тварини (1 – собака, 2 – кіт), «BreedName» - найменування породи.

```
BreedID,Type,BreedName
1,1,"Affenpinscher"
2,1,"Afghan Hound"
3,1,"Airedale Terrier"
4,1,"Akbash"
5,1,"Akita"
6,1,"Alaskan Malamute"
7,1,"American Bulldog"
8,1,"American Eskimo Dog"
9,1,"American Hairless Terrier"
10,1,"American Staffordshire Terrier"
11,1,"American Water Spaniel"
12,1,"Anatolian Shepherd"
13,1,"Appenzell Mountain Dog"
14,1,"Australian Cattle Dog/Blue Heeler"
15,1,"Australian Kelpie"
16,1,"Australian Shepherd"
17,1,"Australian Terrier"
18,1,"Basenji"
19,1,"Basset Hound"
20,1,"Beagle"
21,1,"Bearded Collie"
22,1,"Beauceron"
23,1,"Bedlington Terrier"
24,1,"Belgian Shepherd Dog Sheepdog"
25,1,"Belgian Shepherd Laekenois"
26,1,"Belgian Shepherd Malinois"
27,1,"Belgian Shepherd Tervuren"
28,1,"Bernese Mountain Dog"
29,1,"Bichon Frise"
30,1,"Black and Tan Coonhound"
31,1,"Black Labrador Retriever"
32,1,"Black Mouth Cur"
33,1,"Black Russian Terrier"
34,1,"Bloodhound"
35,1,"Blue Lacy"
```

Рисунок 7.9 – Структура файлу «breed_labels.csv»

Після введення усіх даних бот запитує користувача про вірність введених даних (рис. 7.10). Після уведення текстового опису тварини спочатку текст перекладається на англійську мову, оскільки для навчання моделі використовувалась англомовна інформація. Після цього текст трансформується у модель мішку слів (bag-of-words).

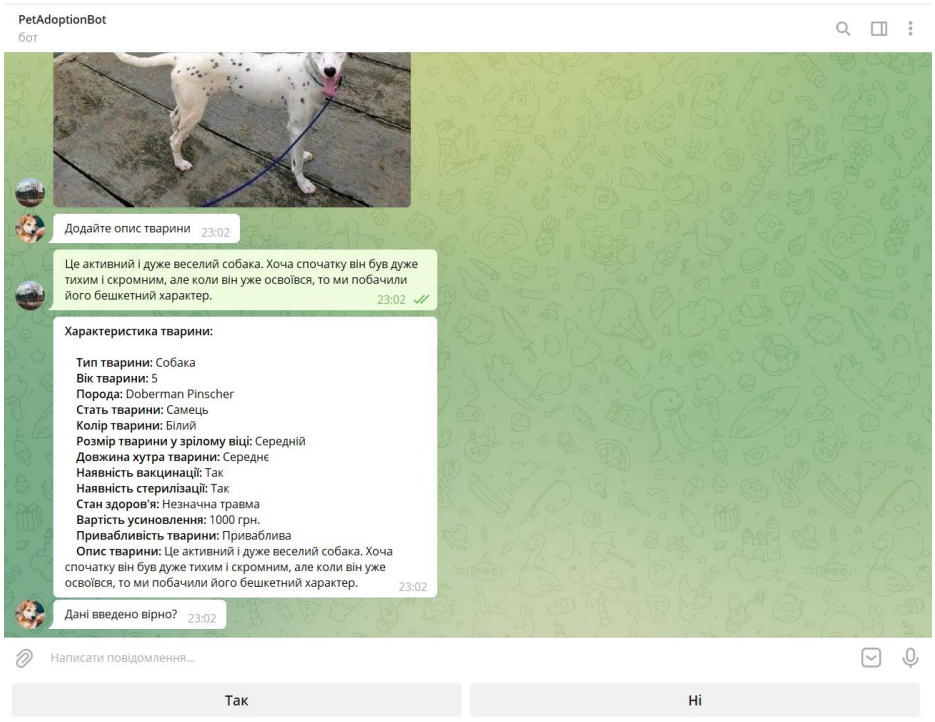

Рисунок 7.10 – Запит боту про вірність введення інформації

Після підтвердження вірності даних бот надає прогноз періоду перебування тварини у притулку та дає рекомендації щодо покращення параметрів тварини, які зможуть зменшити цей період (рис. 7.11).

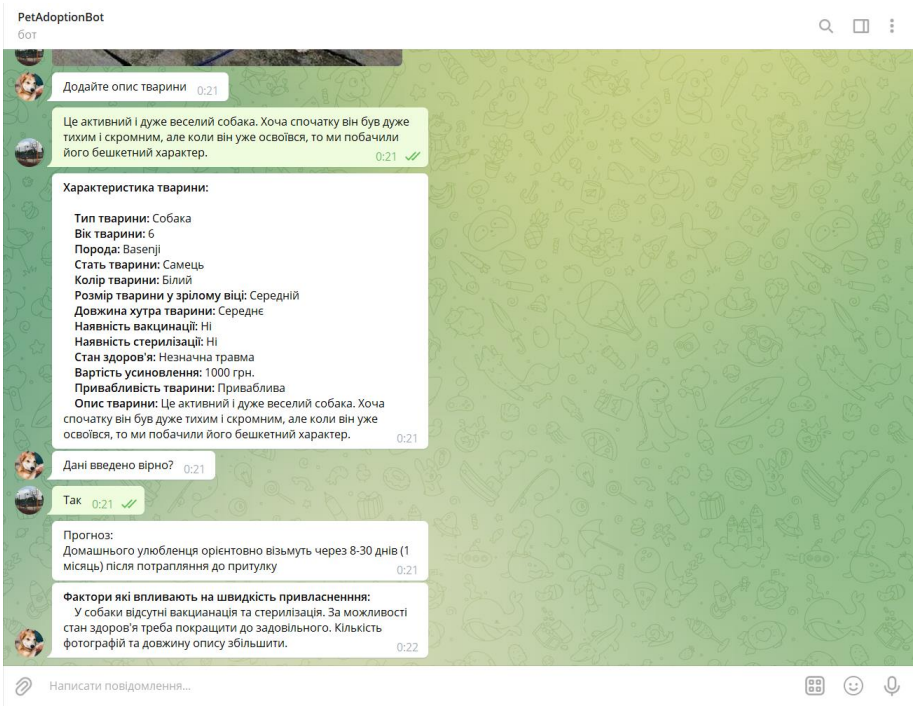

Рисунок 7.11 – Результат прогнозу періоду перебування у притулку

Висновки до розділу

Інформаційна система реалізована у вигляді Telegram боту. Бот дозволяє зручно комунікувати з користувачем, приймати на вхід дані у форматах тексту та зображень. Це є достатнім для того щоб отримати необхідну для прогнозування інформацію.

Бот використовує 3 заздалегідь натренованих моделі: модель визначення привабливості тварини по зображенню, модель трансформації тексту та модель прогнозування періоду перебування тварини у притулку.

Бот реалізовано за допомогою технології long polling, що має ряд переваг у порівнянні зі звичайним polling. Комунікацію з ботом реалізовано за допомогою методу ConversationHandler бібліотеки python-telegram-bot. Бот містить 13 етапів введення вхідної інформації: 11 з них мають варіанти відповідей із присутністю клавіатурою, 1 етап – додавання зображень та 1 етап – текстовий опис, що вводиться вручну. Після введення та підтвердження вірності даних бот надає прогноз періоду перебування тварини у притулку та дає рекомендації щодо покращення параметрів тварини, які зможуть зменшити цей період.

8 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

8.1 Опис ідеї проєкту

У таблиці 8.1 наведено опис ідеї стартап-проєкту.

Таблиця 8.1 – Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка та реалізація системи для прогнозування періоду перебування тварини в притулку	Для притулків та рятувальників	Покращення привабливості профілів домашніх тварин, зменшення страждань та евтаназії тварин. Ефективне прогнозування періоду перебування надасть можливість притулкам оптимально розподіляти ресурси, щоб поліпшити загальну продуктивність усиновлення, та як наслідок, зменшити витрати на притулок та виховання.

У таблиці 8.2 визначено сильні, слабкі та нейтральні характеристики ідеї проєкту.

Таблиця 8.2 – Визначення сильних, слабких та нейтральних характеристики ідеї проєкту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проєкт	Web pet adoption prediction			
1	Визначення привабливості за зображенням тварини	Наявний	Відсутній			S
2	Аналіз текстового опису тварини	Наявний	Відсутній			S
3	Гнучкість в розробці візуальних форм та дизайну системи	Відсутній	Наявний	W		
4	Зручність та швидкість реалізації поправок та нововведень до системи	Наявний	Відсутній			S
5	Прогнозування періоду перебування домашньої	Наявний	Наявний		N	

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проєкт	Web pet adoption prediction			
	тварини в притулку					
6	Підтримка української мови	Наявний	Відсутній			S

8.2 Технологічний аудит ідеї проєкту

Таблиця 8.3 – Технологічна здійсненність ідеї проєкту

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Визначення привабливості за зображенням тварини	Комп’ютерний зір	Наявна	Доступна
		Бібліотека OpenCV	Наявна	Доступна
2	Обробка тексту засобами NLP	Модель bag-of-words	Наявна	Доступна
		Бібліотека nltk	Наявна	Доступна
3	Прогнозування періоду перебування домашньої тварини в притулку	Модель випадкового лісу	Необхідно доробити	Доступна
		Бібліотека scikit-learn	Наявна	Доступна

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
Обрана технологія реалізації ідеї проєкту: модель машинного навчання випадковий ліс. Реалізована у бібліотеці мови Python - scikit-learn.				

8.3 Аналіз ринкових можливостей запуску стартап-проєкту

Таблиця 8.4 – Попередня характеристика потенційного ринку стартап-проєкту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	1 (Web pet adoption prediction)
2	Загальний обсяг продаж, грн/ум.од	27 200 грн.
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Зацікавленість обмеженої кількості регіонів
5	Специфічні вимоги до стандартизації та сертифікації	1. Вимоги до роботи інформаційної системи у притулках для тварин 2. Доступна для розуміння документація до ІС
6	Середня норма рентабельності в галузі (або по ринку), %	14,4%

Середня норма рентабельності по ринку – 12,4%, що є вищим із банківськими відсотками на вкладення (у середньому 10%), тому проєкт є привабливим для входження.

Визначені потенційні групи клієнтів, їх характеристики, та сформовано орієнтовний перелік вимог до товару для кожної групи (табл. 8.5).

Таблиця 8.5 – Характеристика потенційних клієнтів стартап-проєкту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	1. Роботизація, мінімізація людського ресурсу 2. Інтелектуальний аналіз 3. Зручна комунікація з користувачами 4. Оптимальне розподілення ресурсів притулків для тварин	1. Приватні фірми (притулки) 2. Приватні підприємці – власники притулків	– Необхідність оптимізації бізнесу – Орієнтація на якість продукту – Підтримка репутації	– Підвищення ефективності бізнесу – Мінімізація людського ресурсу – Залучення клієнтів, за допомогою сучасних технологій

Фактори, що перешкоджають ринковому впровадженню проєкту наведено в таблиці 8.6.

Таблиця 8.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Зависока ціна для цільової групи клієнтів	Покупець не вбачає належної цінності товару за запропоновану вартість	Випуск альтернативної моделі з менш ефективними методами. Нарощування продажів для зменшення ціни. Акційні пропозиції
2.	Недостатня рекламна компанія	Недостатнє розповсюдження реклами та недостатня зацікавленість цільової аудиторії	Агресивніша рекламна політика. Заміна SMM партнерів.
3.	Банкротство партнерів	Компанії-партнери з певних причин більше не приймають участь у бізнесі	Пошук нових партнерів

Фактори, що сприяють ринковому впровадженню проєкту наведено в таблиці 8.7.

Таблиця 8.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Поява нових технологій	Створення моделей з кращою точністю	Виробництво продукту з використанням новішої технології
2.	Відсутність конкурентів на вітчизняному ринку	Вільний ринок	Можливість швидкого розвитку

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
3.	Розширення множини типів тварин	Множина типів тварин, що враховуються у складанні прогнозів	Збільшення множини типів тварин, що враховуються у складанні прогнозів
4.	Програма лояльності	Реферальна програма – знижка обом учасникам	Рекламна компанія, націлена на розширення кількості користувачів

Проведемо аналіз пропозиції: визначимо загальні риси конкуренції на ринку (табл. 8.8).

Таблиця 8.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Конкуренція Чиста конкуренція	Компанія, користується досить високою популярністю, але існують і інші компанії, які створюють конкуренцію	Впровадження сучасних технологій, залучення капіталу у рекламні кампанії.
За рівнем конкурентної боротьби Міжнародний рівень	Охоплення розвинутих країн світу	Співпраця з міжнародними компаніями.
За галузевою ознакою внутрішньогалузева	Конкуренція між компаніями, які займаються прогнозуванням	Підстроювання під потреби клієнтів.

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
	швидкості привласнення тварин з притулків	
4. Конкуренція за видами товарів: - товарно-видова	Конкуренція між товарами одного виду	Оновлення продукту, виправлення багів, створення додаткового функціоналу.
5. За характером конкурентних переваг - нецінова	Конкуренція проводиться за рахунок вдосконалення якості продукту	Вдосконалення ефективності моделей
6. За інтенсивністю - не марочна	Роль торгової марки незначна	Заохочення клієнтів якістю товару, а не брендом.

Аналіз умов конкуренції в галузі (табл. 8.9).

Таблиця 8.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Приватні фірми	Чиста конкуренція	Розміри поставок	Зручний функціонал	Ціна
Висновки:	Конкурентна боротьба присутня, але кількість	Можливість виходу на ринок є легкою, оскільки	Зменшення ціни на обладнання, при	Потребують зручної взаємодії з ІС	Обмежень немає

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Приватні фірми	Чиста конкуренція	Розміри поставок	Зручний функціонал	Ціна
	конкурентів незначна	присутня невелика кількість аналогів, які мають функціонал не кращий, а то і гірший.	закупівлі оптом.		

Аналіз переліку факторів конкурентоспроможності в таблиці 8.10.

Таблиця 8.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
1	Сучасні технології ШІ	Реалізація сучасних алгоритмів та методів аналізу інформації
2	Доступна для розуміння документація	Клієнт потребує відсутності будь-яких складнощів, при використанні продукту
3	Технічне обслуговування	Наявність технічної бази з усіх точок
4.	Ціна та собівартість продукції	Конкурентоспроможна ціна

За визначеними факторами конкурентоспроможності (табл. 8.10) проведено аналіз сильних та слабких сторін стартап-проєкту (табл. 8.11).

Таблиця 8.11 – Порівняльний аналіз сильних та слабких сторін «PetAdoptionBot»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з PetAdoptionBot						
			–3	–2	–1	0	+1	+2	+3
1	Сучасні технології ШІ	16	+						
2	Доступна для розуміння документація	13			+				
3	Технічне обслуговування	12				+			
4	Ціна та собівартість продукції	13				+			

На основі виділених ринкових загроз та ринкових можливостей, та сильних і слабких сторін складено SWOT-аналіз (матриця аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 8.12).

Таблиця 8.12 – SWOT- аналіз стартап-проєкту

Сильні сторони: – Підвищення шансів знаходження нового господаря для тварини – Висока надійність – Простота використання – Зрозумілість необхідності товару.	Слабкі сторони: – Висока вартість продукту – Необхідність регулярного перенавчання застарілих моделей на основі нових даних
Можливості: – Поява нових технологій – Відсутність конкурентів на вітчизняному ринку – Розширення множини типів тварин – Програма лояльності.	Загрози: – Зависока ціна для цільової групи клієнтів – Недостатня рекламна компанія – Банкротство партнерів.

Визначені альтернативи проаналізовано з точки зору строків та ймовірності отримання ресурсів (табл. 8.13).

Таблиця 8.13 – Альтернативи ринкового впровадження стартап-проєкту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Вихід на ринки нових країн	Необхідні великі фінансові вкладення. Середня ймовірність	3 місяці
2.	Кооперація з відомими мережами притулків	Необхідні великі фінансові вкладення. Середня ймовірність	2 місяць

Варіант з кооперацією з відомими мережами притулків є швидшим по реалізації, тому при альтернативному ринковому впровадженні цей варіант є більш привабливим.

8.4 Розроблення ринкової стратегії проєкту

Обрано цільові групи потенційних споживачів (табл. 8.14).

Таблиця 8.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Притулки, що відносяться до державної власності	Низький, держава не готова до прозорих витрат на проєкт та чесних тендерів	5-10% - низький попит	Низька конкуренція, але державні установи не зацікавлені проєктами, з яких не має можливості «відмити» гроші.	Важко, присутня корупційна складова
2	Притулки, які належать приватним фірмам	Високий, оскільки приватні фірми зацікавлені залученням клієнтів, за допомогою нових технологій	40-50% - середній попит	Низька конкуренція	Досить легко

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
3	Притулки, що належать до приватної власності (підприємці)	Високий, підприємці зацікавлені у залученні автоматизації бізнесу	40-50% - середній попит	Низька конкуренція	Досить легко
Які цільові групи обрано: притулки приватної форми власності					

Сформовано базову стратегію розвитку (табл. 8.15).

Таблиця 8.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Орієнтування на розвитку ІТ, розробці ПЗ та алгоритмів комп`ютерного зору	Концентрований маркетинг	Швидкі терміни впровадження ІС; дорогі, але якісні технології; залучення професіоналів.	Концентрична форма розвитку стратегії «Диверсифікація»

Обрано стратегію конкурентної поведінки (табл. 8.16).

Таблиця 8.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Ні, проект «зайшов» на ринок паралельно з невеликою кількістю аналогічних.	Акцент буде зроблений на якість реалізації ПЗ, зручності у використанні. Компанія буде шукати нових споживачів, залучаючи СРА-мережі та переманювати споживачів, акцентуючи увагу на якості ПЗ та застосуванні передових алгоритмів ШІ.	Буде копіювати, але не в значній мірі. Пильно будуть відстежуватися успішні рішення конкурентів. Для даного аналізу будуть залучені спеціалісти.	Контрнаступ

На основі вимог споживачів з обраних сегментів до постачальника та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розроблено стратегію позиціонування (табл. 8.17), що

полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 8.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Простота і зручність у використанні	Диверсифікація	Логування помилок системи та відправлення її на сервер компанії, що відповідає за проект, для подальшої обробки. Створення зручного інтерфейсу ІС та адмін-панелі.	– Стеження за продуктом – Залучення професіоналів – Використання методів ШІ для розуміння потреб клієнта
2	Якість обладнання	Диверсифікація	Ретельне тестування обладнання перед встановленням. Акуратна доставка продукції встановлення. Орієнтування на якість, а не на ціну.	– Партнерство з ключовими виробниками обладнання – Тестування обладнання – Дорого, але якісно

8.5 Розроблення маркетингової програми стартап-проєкту

Підсумовано результати попереднього аналізу конкурентоспроможності товару (табл. 8.18).

Таблиця 8.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Визначення привабливості за зображенням тварини	Методи комп'ютерного зору для класифікації зображень за рівнем привабливості	Залучення передових алгоритмів
2	Прогнозування періоду перебування домашньої тварини в притулку	Зручна взаємодія із системою	Покращені алгоритми класифікації.
3	Обробка тексту засобами NLP	Підтримка декількох мов при використанні системи	Багатомовність

Розроблено трирівневу маркетингову модель товару: ідея продукту, його фізичні складові, особливості процесу його надання (табл. 8.19).

Таблиця 8.19 - Опис трьох рівнів моделі товару

<table>
<tr><td>Рівні товару</td><td colspan="3">Сутність та складові</td></tr>
<tr><td>I. Товар за задумом</td><td colspan="3">Інформаційна системи для прогнозування періоду перебування тварини у притулку, що реалізована у вигляді Telegram боту. Система приймає на вхід дані – параметри тварини, фото та текстовий опис. Після цього система визначає рівень привабливості тварини по фото та трансформує текст до подальшої обробки. Наостанок система робить прогноз та дає рекомендацію.</td></tr>
<tr><td rowspan="5">II. Товар у реальному виконанні</td><td>Властивості/характеристики</td><td>М/Нм</td><td>Вр/Тх /Тл/Е/Ор</td></tr>
<tr><td>1. Сховище даних
2. Зручний інтерфейс
3. Кросплатформенність</td><td>М
М
М</td><td>Тх/Вр
Тх/Ор
Тх/Ор</td></tr>
<tr><td colspan="3">Якість: Тестування моделей перед використанням.
стандарти, нормативи, параметри тестування тощо</td></tr>
<tr><td colspan="3">Пакування
Документація в електронному вигляді</td></tr>
<tr><td colspan="3">Марка: PetUp</td></tr>
<tr><td rowspan="2">III. Товар із підкріпленням</td><td colspan="3">До продажу
Інтеграція сервісу з чат-ботом (Telegram Bot).</td></tr>
<tr><td colspan="3">Після продажу
Безкоштовне обслуговування боту упродовж 1 року</td></tr>
<tr><td colspan="4">За рахунок чого потенційний товар буде захищено від копіювання:
патенту та авторського права</td></tr>
</table>

Визначено цінові межі, якими необхідно керуватись при встановленні ціни на потенційний товар (табл. 8.20).

Таблиця 8.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	23 000 грн.	36 000 грн.	36 000 грн.	24 000 – 28 000 грн.

Визначено оптимальну систему збуту, в межах якої приймається рішення (табл. 8.21).

Таблиця 8.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Клієнти можуть купувати продукт безпосередньо у компанії-розробника або ж залучаючи своїх постачальників	– Вчасна доставка – Довгострокова гарантія	0-глибина: прямі та непрямі канали збуту	Сайт постачальника

Розроблено концепцію маркетингових комунікацій (табл. 8.22).

Таблиця 8.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Клієнти дізнаються про товар із соціальних мереж (таргетована реклама)	Мережа	Сучасна автоматизована система	Переконати в необхідності використання	Змінимо життя відданих, вірних та незахищених представників вусатих-смугастих: «PetAdoptionBot – покращить життя домашніх тварин у притулку».

Висновки до розділу

Проєкт PetAdoptionBot має кращий показник рентабельності відносно відсотків на банківських вкладів, тому можлива його ринкова комерціалізація. Обраною технологією реалізації ідеї проєкту є модель машинного навчання випадковий ліс. З розвитком технологій можливе покращення продукту, подальша імплементація має перспективу.

ВИСНОВКИ

Щодня в усьому світі мільйони бездомних тварин страждають на вулицях або піддаються евтаназії в притулках для тварин. Вирішення таких проблем, як утримання безхатніх тварин, їх годування, лікування є фінансово затратними. Зменшення фінансових витрат притулками є актуальною задачею. Ефективне прогнозування періоду затримки надасть можливість притулкам оптимально розподіляти ресурси, щоб поліпшити загальну продуктивність усиновлення, та як наслідок, зменшити витрати на притулок та виховання.

Для знаходження найкращього результату були задіяні моделі логістичної регресії, наївного Баєсу, методу опорних векторів, дерев рішень, випадкового лісу та гранично випадкових лісів. Для класифікації фотографій домашніх тварин за привабливістю можна використано технологію emotion detection. Класифікацію зображень реалізовано за допомогою нейронної мережі архітектури Inception-v3. У наборі даних присутній текстовий опис тварин. Для трансформації тексту використані методи NLP.

Для вирішення задачі реалізована інформаційна система. ІС має ряд варіантів використання: введення вхідних даних, запит на введення інформації, підтвердження введеної інформації, збереження вхідних даних до бази, прогноз періоду перебування тварини у притулку, препроцесінг даних, класифікація зображень та рекомендацію щодо покращення параметрів.

Інформаційна система реалізована у вигляді Telegram боту та використовує Telegram Bot API. Для мови Python функціонал даного API реалізований у бібліотеці telegram. Для збереження даних використано СКБД PostgreSQL.

Бот використовує 3 заздалегідь натренованих моделі: модель визначення привабливості тварини по зображенню, модель трансформації тексту та модель прогнозування періоду перебування тварини у притулку.

В магістерській дисертації наведено вирішення усіх поставлених завдань.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Mel Hanson. Pet Adoption Statistics / Mel Hanson. – 2020. URL: https://spots.com/pet-adoption-statistics/ .

2. Arie Ben-David. Comparison of classification accuracy using cohen's weighted kappa. Expert Systems with Applications, 34(2):825–832, 2008.

3. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

4. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, 2014.

5. Seyedmahdad Mirsamadi, Emad Barsoum, and Cha Zhang. Automatic speech emotion recognition using recurrent neural networks with local attention. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2227–2231. IEEE, 2017.

6. Leo Breiman. Classification and regression trees. Routledge, 2017.

7. Nathan Ensmenger. Is chess the drosophila of artificial intelligence? A social history of an algorithm / Nathan Ensmenger. // Social Studies of Science. – 2011. – С. 5–30.

8. Baby Schema in Infant Faces Induces Cuteness Perception and Motivation for Caretaking in Adults / M. Glocker, D. Langleben, K. Ruparel та ін.]. // Ethology. – 2009. – С. 257–263.

9. Baby schema modulates the brain reward system in nulliparous women / M. Glocker, D. Langleben, K. Ruparel та ін.]. // PNAS. – 2009. – С. 9115–9119.

10. G. Genosko. Natures and Cultures of Cuteness. Invisible Culture / G. Genosko // An Electronic Journal for Visual Culture. – 2005. URL: https://www.rochester.edu/in_visible_culture/Issue_9/issue9_genosko.pdf .

11. Cuteness Recognition and Localization in the Photos of Animals / Yu Bao, Jing Yang, Liangliang Cao та ін.]. // Proceedings of the 22nd ACM international conference on Multimedia. – 2014. – С. 1193–1196.

12. Sense Beyond Expressions: Cuteness / Kang Wang, Tam V. Nguyen, Jiashi Feng, Jose Sepulveda. // Proceedings of the 23rd ACM international conference on Multimedia. – 2015. – С. 1067–1070.

13. Building a Large Scale Dataset for Image Emotion Recognition: The Fine Print and The Benchmark / Quanzeng You, Jiebo Luo, Hailin Jin, Jianchao Yang. // AAAI 2016. – 2016.

14. Machinelearningmastery , 2020. URL: https://machinelearningmastery.com/gradient-descent-for-machine-learning/.

15. S. Ray. 6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R / S. Ray // Analytics Vidhya. – 2017. URL: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/.

16. Hackernoon , 2020. – URL: https://hackernoon.com/gradient-descent-aynk-7cbe95a778da .

17. Classification And Regression Trees / L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone. – Boca Raton, Florida, United States: Routledge, 2017. – 368 с.

18. T. Hastie. The Elements of Statistical Learning / T. Hastie, R. Tibshirani, J. Friedman. – New York, NY: Springer, 2009. – 745 с. – (Second Edition).

19. J. Pennington. GloVe: Global Vectors for Word Representation / J. Pennington, R. Socher, C. Manning // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) / J. Pennington, R. Socher, C. Manning. – Doha, Qatar: Association for Computational Linguistics, 2014. – С. 1532–1543.

20. Animal Emotion Detection and Application / B. K.Singh, T. Dua, D. S. Prasad, A. Adane. – 2020.

21. Alan MacCormack, Chris F. Kemerer, Michael Cusumano, Bill Crandall, Trade-offs beetween Productivity and Quality in Selecting Software Development Practices, IEEE Com-puter Society, 2003

22. Schrimpf, A.; Single, M.-S.; Nawroth, C. Social Referencing in the Domestic Horse. Animals 2020, 10, 164.

23. Susana G Sotocinal, Robert E Sorge, Austin Zaloum, Alexander H Tuttle, Loren J Martin, Jeffrey S Wieskopf, Josiane CS Mapplebeck, Peng Wei, Shu Zhan,

Shuren Zhang, Jason J McDougall, Oliver D King, and Jeffrey S Mogil, The Rat Grimace Scale: A partially auto-mated method for quantifying pain in the laboratory rat via facial expressions, Sage Publi-cations, 2011 Jul 29. doi: [10.1186/1744-8069-7-55]

24. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.

25. S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of The 32nd International Conference on Machine Learning, pages 448–456, 2015.

26. Rehling J. How Natural Language Processing Helps Uncover Social Media Sentiment / John Rehling. – 2011. URL: https://mashable.com/archive/natural-language-processing-social-media#VbWC8PySNqqy.

27. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

28. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, 2014.

29. Seyedmahdad Mirsamadi, Emad Barsoum, and Cha Zhang. Automatic speech emotion recognition using recurrent neural networks with local attention. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2227–2231. IEEE, 2017.

30. Cai E. Machine Learning Lesson of the Day – Overfitting and Underfitting / Eric Cai. // The Chemical Statistician. – 2014.

31. Goyal P. Deep Learning for Natural Language Processing / P. Goyal, S. Pandey, K. Jain., 2018.

32. El-Amir H. Deep Learning Pipeline: Building a Deep Learning Model with TensorFlow / H. El-Amir, M. Hamdy., 2020.

33. Gallé M. Investigating the Effectiveness of BPE: The Power of Shorter Sequences / Matthias Gallé. // Proceedings of the 2019 Conference on Empirical Methods

in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). – 2019.

34. Teja S. Stop Words in NLP / Sai Teja. – 2020. URL: https://medium.com/@saitejaponugoti/stop-words-in-nlp-5b248dadad47 .

35. An Interpretation of Lemmatization and Stemming in Natural Language Processing Divya Khyani1 , Siddhartha B S2 , Niveditha N M3 ,Divya B M4 1 BGS Institute of Technology,2,3,4 Adichunchanagiri University-BGSIT. URL: https://www.baeldung.com/cs/stemming-vs-lemmatization .

36. Tharwat A. (August 2018). "Classification assessment methods". Applied Computing and Informatics.

37. Webster J. Tokenization as the initial phase in NLP / J. Webster, C. Kit. // Proceedings of the 14th conference on Computational linguistics. – 1992. – №4.

38. Feature Selection: A Data Perspective / [J. Li, K. Cheng, S. Wang та ін.]. // ACM Computing Surveys. – 2016. – №50.

39. Susmaga R. Confusion Matrix Visualization / Robert Susmaga. // Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'04 Conference held in Zakopane, Poland. – 2004.

40. Neural Network a Comprehensive Foundation; a Computational Approach to Learning and Machine Intelligence / S Haykin, - Macmillan, NY, 1994.

41. Yasin H. Prediction of Crude Oil Prices using Support Vector Regression (SVR) with grid search – cross validation algorithm / H. Yasin, R. C. Eko, A. Hoyyi. // Global Journal of Pure and Applied Mathematics. – 2016. – №12. – С. 3009–3020.

42. Du K. Recurrent Neural Networks / K. Du, M. Swamy // Neural Networks and Statistical Learning / K. Du, M. Swamy., 2014.

43. Hochreiter J. Diplomarbeit im fach informatik / Hochreiter Josef, 1991.

44. Hochreiter S. Long Short-term Memory / S. Hochreiter, J. Schmidhuber. // Neural Computation. – 1997. – №9.

45. Telegram Bot API . – 2021. URL: https://core.telegram.org/bots/api .

46. Documentation - PostgreSQL . – 2021. URL: https://www.postgresql.org/about/ .

47. Преимущества Python перед другими языками программирования. URL: https://www.digest.pro/news/preimushhestva-python-pered-drugimi-jazykamiprogrammirovanija/ , 2019.

48. Динамика популярности языков программирования для использования в машинном обучении . URL: https://trends.google.com/trends/explore?date=today%205-y&q=Python%20Machine%20Learning,R%20Machine%20Learning,Java%20Machine%20Learning,Scala%20Machine%20Learning,Julia%20Machine%20Learning .

49. Top 10 Python Libraries for Machine Learning . URL: https://www.zenesys.com/blog/top-10-python-libraries-for-machine-learning , 2021.

50. Multidisciplinary Instruction with the Natural Language Toolkit / S.Bird, E. Klein, E. Loper, J. Baldridge., 2008.