

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-
МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики
Бакалаврська програма

**Використання машинного навчання для створення мобільної
рекомендаційної системи для підбору вбрання**

**Текстова частина до курсової роботи
за спеціальністю «Інженерія програмного забезпечення» - 121**

Керівник курсової роботи
Франків О.О.

(Підпис)

“ ___ ” _____ 2023 року

Виконала студентка БП ІПЗ-3

Авраменко К.В.

“ ___ ” _____ 2023 року

Київ 2023

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики технологій факультету інформатики

ЗАТВЕРДЖУЮ

Викладач кафедри інформатики ,
канд. фіз-мат. наук, доц. _____ Гороховський С.С.
(підпис)
„_____” _____ 2023р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу
студентці Авраменко Катерині Вадимівні
факультету інформатики 3 курсу бакалаврської програми
ТЕМА: Використання машинного навчання для створення мобільної
рекомендаційної системи для підбору вбрання

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Вступ

Теоретичні основи

Використаний підхід до рішення проблеми

Імплементация обраних технологій та створення застосунку

Висновки

Список покликань

Дата видачі „_____” _____ 2022 р.

Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Тема: Використання машинного навчання для створення мобільної рекомендаційної системи для підбору вбрання

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	23.10.2022	
2.	Огляд літератури за темою роботи.	23.11.2022	
3.	Проведення досліджень	24.11.2022- 24.01.2023	
3.	Опис результатів дослідження	28.02.2023	
4.	Аналіз отриманих результатів з керівником,	10.03.2023	
6.	Корегування роботи	10.04.2023	
7.	Створення презентації та написання доповіді.	10.05.2023	
8.	Остаточне оформлення пояснювальної роботи та слайдів.	10.05.2023	
9.	Захист курсової роботи	24.05.2023	

Студентка Авраменко К.В.

Керівник _____

“ _____ ” _____

Зміст

Анотація	5
Вступ	6
Розділ 1: Теоретичні основи	8
1.1 Індустрія вбрання та технології	8
1.2 Комп'ютерний зір, як спосіб розпізнавання типу та кольору вбрання.....	9
1.3 Рекомендаційні системи: Основи та технології.....	12
1.4 Алгоритм, використаний для навчання табличного класифікатора	16
1.5 Розробка та інтеграція додатків для iOS.....	17
1.6 Висновки до розділу 1	19
Розділ 2. Використаний підхід до рішення проблеми	20
2.1 Core Data: фреймворк, для оперування базою даних вбрання	20
2.2 SwiftUI: фреймворк для написання користувацького інтерфейсу	21
2.3 CoreML як спосіб інтеграції машинного навчання в iOS застосунки.....	22
2.4 Обґрунтування вибору CreateML для тренування моделей.....	25
2.5 Висновки до розділу 2	27
Розділ 3. Імплементация обраних технологій та створення застосунку	28
3.1 Навчання та тестування моделей класифікації типу та кольору вбрання	28
3.2 Навчання та тестування табулярного класифікатора для рекомендації вбрання в контексті погоди	30
3.3 Інтеграція моделей в iOS застосунк.....	32
3.4 Застосунок iDresser як результат дослідження	33
3.5 Висновки до розділу 3	35
Висновки	36
Список покликань	38

Анотація

У цій роботі досліджується застосування методів машинного навчання для розробки додатку для iOS, призначеного для управління предметами одягу та надання рекомендацій щодо вбрання. Три моделі машинного навчання були створені з використанням фреймворку CoreML від Apple: модель класифікатора зображень для категоризації типу та кольору одягу та модель табличного класифікатора для генерування рекомендацій щодо одягу на основі даних про одяг та поточні погодні умови.

У роботі детально розглянуто процес створення, навчання та інтеграції моделей в iOS-додаток. Вона висвітлює проблеми, що виникають під час інтеграції моделі, та пропонує практичні рішення. Створений додаток з двома основними екранами - "Гардероб" та "Історія вбрання" - дозволяє користувачам ефективно керувати своїм гардеробом та отримувати пропозиції щодо вбрання, яке підходить для поточних погодних умов.

Вступ

В останні роки машинне навчання стало трансформаційною технологією з потенціалом змінити різні аспекти нашого повсякденного життя. За своєю суттю машинне навчання передбачає розробку алгоритмів, які можуть вивчати шаблони з даних, дозволяючи комп'ютерам приймати керовані даними рішення та прогнозувати без явного програмування. Використовуючи потужність машинного навчання, будь-яка програма може аналізувати величезні масиви даних, визначати складні зв'язки та адаптувати свої пропозиції на основі вподобань користувачів і контекстних факторів.

У сучасному світі люди постійно шукають способи заощадити час і зробити свою повсякденну роботу ефективнішою. Однією з таких сфер є вибір правильного вбрання, що може бути трудомістким і складним завданням, особливо якщо враховувати такі фактори, як особистий стиль, тип одягу та погодні умови. У даній роботі представлено програму для рекомендацій одягу, розроблену для вирішення цієї проблеми шляхом використання машинного навчання, бази даних вбрання користувачів та інформації про погоду в реальному часі, щоб надавати користувачам персоналізовані пропозиції одягу з урахуванням місцезнаходження користувача.

Неможливо переоцінити важливість машинного навчання в розробці програми для рекомендацій одягу. Це дає змогу додатку зрозуміти гардероб користувача, дізнатися його вподобання та надати рекомендації, адаптовані до погодних умов. Основною метою цього дослідження є розробка програми, яка може точно класифікувати предмети одягу за кольором і типом, зберігати їх у віртуальному гардеробі та генерувати рекомендації щодо одягу відповідно до місця розташування користувача та переважаючих погодних умов. Для досягнення цієї мети програма

використовує модель Tabular Classifier (Core ML) для класифікації одягу та рекомендацій, а також Core Data для ефективного зберігання та пошуку даних. Крім того, програма дозволяє користувачам вручну додавати вбрання та зберігати їх у програмі, пропонуючи повний та персоналізований досвід.

Ця робота поділена на три основні розділи, кожен з яких містить детальне розуміння різних аспектів процесу розробки програми. У першому розділі розглядаються теоретичні основи дослідження, обговорюється Core ML, його особливості, переваги та обмеження, а також порівнюється з альтернативними інструментами машинного навчання. У другому розділі обґрунтовується обраний підхід і розглядаються SwiftUI, модель Object Classifier та алгоритми, що використовуються для класифікації одягу та розпізнавання кольорів. В останньому розділі представлено докладний опис впровадження програми, включаючи навчання моделі, перевірку та тестування, а також інтеграцію класифікації одягу, рекомендацій, даних про погоду та функцій збереження даних.

Дана робота присвячена використанню методів машинного навчання (ML) для розробки системи рекомендацій щодо одягу в додатках для iOS. Для виконання цього завдання використовується три різні моделі ML: одну для класифікації предметів одягу, іншу для класифікації кольорів вбрання і третю для рекомендації комбінацій одягу. Ці моделі працюють у тандемі, щоб забезпечити безперебійний та ефективний користувацький досвід, зрештою надаючи користувачам відповідні пропозиції щодо вбрання, які відповідають погоді, в залежності від місцезнаходження користувача. Запропонований додаток для рекомендацій одягу прагне покращити щоденний процес прийняття рішень користувачами, допомогти їм заощадити час і покращити їхній загальний досвід у виборі ідеального вбрання для будь-якої погоди.

Розділ 1: Теоретичні основи

1.1 Індустрія вбрання та технології

Протягом останніх років індустрія одягу зазнала значних трансформацій, пов'язаних зі стрімким розвитком технологій та зростаючим впливом соціальних медіа-платформ. Як наслідок, споживачі постійно шукають персоналізовані та ефективні рішення, щоб не відставати від постійно мінливих трендів. Системи рекомендацій щодо вбрання стали перспективним інструментом, який допомагає користувачам приймати обґрунтовані рішення щодо свого гардеробу, пропонуючи персоналізовані чи контекстно-орієнтовані пропозиції, пристосовані до їхніх уподобань, потреб чи погодніх умов.

Окрім того темп життя людей з кожним роком зростає, тому люди шукають нові та ефективніші рішення для повсякденних проблем, аби витратити менше часу на рутинні задачі. Поява вже таких звичних нам додатків, як Google Maps, чи веб-застосунки, де можна дізнатись прогноз погоди, значно спростили повсякденне життя людини. З метою економії часу, люди все більше використовують рекомендаційні системи товарів під час онлайн покупок чи слів, під час набирання тексту. Рекомендація повсякденного вбрання є чи не найбільш вдалим прикладом, як технології та зручний інтерфейс можуть допомогти людині зробити найбільш правильний вибір.

Телефон людина вже давно використовує як асистент по життю, зберігаючи там робочі документи чи персональні фотографії. Із цього випливає, що маючи в телефоні віртуальну шафу із власним одягом, людина змогла би зручно оперувати своїм гардеробом та одразу бачити все вбрання, що вона має у будь-який час, не залежно від того – людина знаходиться поруч зі своєю шафою чи ні.

Алгоритми комп'ютерного зору для розпізнавання одягу мають широкий спектр потенційних результатів. Одні лише можливості електронної комерції величезні. Оскільки сотні мільярдів доларів витрачаються на купівлю одягу щороку, ефективний додаток для автоматичної ідентифікації та пошуку одягу за візуальною схожістю мав би виняткову цінність. Крім того, існує сильний контекстуальний зв'язок між предметами одягу та частинами тіла - наприклад, ми носимо капелюхи на голові, а не на ногах. Для задач візуального розпізнавання, таких як виявлення людини або ідентифікація пози, знання того, які предмети одягу носить людина, і локалізація цих предметів може призвести до вдосконалення алгоритмів оцінки конфігурації тіла.

Незважаючи на потенційні дослідницькі та комерційні вигоди від оцінки одягу, відносно небагато дослідників вивчали проблему розпізнавання одягу, в основному зосереджуючись на дослідженні проблеми в обмежених областях або розпізнаючи дуже невелику кількість типів одягу. Підхід, обраний для даного дослідження вирішує проблему оцінювання одягу на доволі широкому масштабі для реальних зображень. Розглядається велика кількість різних типів одягу, яка згодом класифікується на такі як футболки, верхнє зимове чи демісезонне вбрання, взуття тощо і досліджуються методи для точного аналізу зображень людей в одязі, аби класифікувати його тип та колір.

1.2 Комп'ютерний зір, як спосіб розпізнавання типу та кольору вбрання

Комп'ютерний зір - це галузь, яка включає в себе методи отримання, обробки та аналізу зображень. Комп'ютерний зір описується як підприємство автоматизації та інтеграції широкого спектру процесів і уявлень для зорового сприйняття, а також це штучні системи, які витягують інформацію із зображень. Методами комп'ютерного зору є взаємодія

комп'ютера з людиною, виявлення подій, ідентифікація та розпізнавання об'єктів. Виявлення об'єктів, класифікація об'єктів і відстеження об'єктів використовуються для вирішення завдань спостереження. Розпізнавання об'єктів - це загальний термін для опису сукупності пов'язаних завдань для комп'ютерного зору, які передбачають ідентифікацію об'єктів на цифрових фотографіях. Класифікація зображень передбачає передбачення класу одного об'єкта на зображенні. Локалізація об'єктів - це визначення місцезнаходження одного або декількох об'єктів на зображенні та нанесення рамки навколо них. Виявлення об'єктів поєднує ці два завдання і локалізує та класифікує один або декілька об'єктів на зображенні. У цій роботі використовуватиметься термін класифікації об'єктів в широкому сенсі, щоб охопити як класифікацію зображень (завдання, що вимагає алгоритму для визначення класів об'єктів, присутніх на зображенні), так і виявлення об'єктів (завдання, що вимагає алгоритму для локалізації всіх об'єктів, присутніх на зображенні). Таким чином, можна розрізнити такі завдання комп'ютерного зору:

А. Виявлення об'єктів

Виявлення об'єктів - це технологія, яка пов'язана з комп'ютерним зором і обробкою зображень. Виявлення об'єктів полягає у виявленні екземплярів семантичних об'єктів у класах, таких як транспортні засоби, будівлі та люди. Основною галуззю досліджень в області виявлення об'єктів є виявлення облич та пішоходів чи виявлення облич та транспортних засобів. В системі виявлення об'єктів є два основних етапи:

- а) Виділення ряду ознак
- б) Навчання класифікатора.

В. Класифікація об'єктів

Класифікатор - це алгоритм, який бере набір ознак, що характеризують об'єкти, і використовує їх для визначення класу кожного об'єкта. Класичним прикладом в астрономії є розрізнення зірок і галактик. Для кожного об'єкта

вимірюється низка властивостей (швидкість, розмір, компактність, граничне поле тощо); класифікатор потім використовує ці властивості, щоб визначити, чи є кожен об'єкт зіркою або галактикою. Існує два типи класифікації: контрольована і неконтрольована. У контрольованій класифікації, тобто коли людина-експерт не тільки визначила властивості об'єкта, але й на які класи можна віднести об'єкт, а також надала набір зразків об'єктів з відомими класами. Цей набір відомих об'єктів називається навчальною вибіркою, оскільки він використовується програмами класифікації для навчання класифікації об'єктів. У методах неконтрольованої класифікації індуктивний двигун працює безпосередньо з даними, і немає ні навчальних наборів, ні заздалегідь визначених класів.

Для написання додатку із рекомендацією одягу важливо імплементувати зручну можливість додавання одягу, оскільки користувач може доволі швидко втратити інтерес. Не існує конкретного часу, який би стосувався кожного додатка, оскільки інтерес користувачів може змінюватися залежно від призначення, дизайну, цільової аудиторії та інших факторів. Однак загально визнано, що є дуже короткий проміжок часу, щоб справити сильне перше враження та залучити користувачів. Цей проміжок часу часто називають "правилом трьох секунд", тобто є близько трьох секунд, щоб привернути увагу користувача, коли він вперше відкриває додаток. Важливо, щоб користувацький інтерфейс додатку був візуально привабливим, зручним для навігації, а процес використання максимально автоматизованим. Базуючись на цьому, при розробці додатку для зберігання вбрання та для зручного відображення доданого вбрання важливо класифікувати одяг за ознаками типу та кольору. Для того, аби користувач не виконував класифікацію вручну і його користувацький досвід був швидшим та більш автоматизованим, в додатку використовується класифікація об'єктів за допомогою комп'ютерного зору. У даній роботі використовувалась контрольована класифікація, оскільки

був наданий навчальний набір із заздалегідь визначеними класами для навчання класифікації об'єктів.

1.3 Рекомендаційні системи: Основи та технології

Рекомендаційні системи відіграють вирішальну роль у допомозі користувачам орієнтуватися у широкому спектрі доступних їм варіантів вибору. Машинне навчання має потенціал для революційних змін у сфері рекомендаційних систем. Розробка ефективної системи рекомендацій щодо вибору вбрання може залежати від багатьох факторів. У даній роботі врахований такий фактор, як погодні умови за місцезнаходженням користувача.

Система рекомендацій - це штучний інтелект або алгоритм штучного інтелекту, зазвичай пов'язаний з машинним навчанням, який використовує великі дані, щоб пропонувати або рекомендувати споживачам додаткові продукти. Вони можуть ґрунтуватися на різних критеріях, включаючи минулі покупки, історію пошуку, демографічну інформацію та інші фактори. Рекомендаційні системи є дуже корисними, оскільки вони допомагають користувачам знаходити продукти та послуги, які вони могли б не знайти самостійно.

Рекомендаційні системи навчені розуміти вподобання, попередні рішення та характеристики людей і продуктів, використовуючи дані, зібрані про їхню взаємодію. Сюди входять покази, кліки, вподобання та покупки. Завдяки своїй здатності передбачати інтереси та бажання споживачів на високо персоналізованому рівні, рекомендаційні системи є фаворитами серед постачальників контенту та продуктів. Вони можуть підштовхнути споживача до будь-якого продукту чи послуги, що його цікавить, від книг до відео, від занять зі здоров'я до одягу.

Крім того системи рекомендацій допомагають користувачам орієнтуватися у великих колекціях товарів і знаходити товари, що відповідають їхнім інтересам, використовуючи великі обсяги інформації про товари, такі як переглянуті або проігноровані товари, покупки або відвідування веб-сторінок, щоб визначити, як, коли і що рекомендувати своїм клієнтам. Системи рекомендацій стали невід'ємною частиною всіх великих інтернет-магазинів, на них припадає до 35% продажів Amazon або понад 80% контенту, що переглядається на Netflix.

Хоча існує величезна кількість алгоритмів і методів рекомендацій, більшість з них поділяються на такі широкі категорії: спільна фільтрація, фільтрація контенту і контекстна фільтрація.

Алгоритми спільної фільтрації рекомендують елементи на основі інформації про вподобання багатьох користувачів. Цей підхід використовує схожість поведінки користувачів, враховуючи попередні взаємодії між користувачами та об'єктами, алгоритми рекомендацій вчать передбачати майбутню взаємодію. Ці рекомендаційні системи будують модель на основі минулої поведінки користувача, наприклад, товарів, придбаних раніше, або оцінок, наданих цим товарам, а також подібних рішень інших користувачів. Ідея полягає в тому, що якщо деякі люди приймали подібні рішення і робили покупки в минулому, наприклад, вибирали фільми, то існує висока ймовірність того, що вони погодяться з додатковим вибором у майбутньому. Наприклад, якщо рекоменатор спільної фільтрації знає, що у вас з іншим користувачем схожі смаки в кіно, він може порекомендувати вам фільм, який, як він знає, вже сподобався іншому користувачеві.

На противагу цьому, контент-фільтрація використовує атрибути або особливості елемента (це контентна частина), щоб рекомендувати інші елементи, схожі на вподобання користувача. Цей підхід ґрунтується на схожості характеристик елемента та користувача, враховуючи інформацію про користувача та елементи, з якими він взаємодіяв (наприклад, вік

користувача, категорію кухні ресторану, середній відгук про фільм), моделює ймовірність нової взаємодії.

Гібридні системи рекомендацій поєднують переваги вищезгаданих типів для створення більш комплексної системи рекомендацій.

Контекстна фільтрація включає контекстну інформацію користувачів у процес рекомендацій. Представники Netflix розповіли про створення кращих рекомендацій шляхом формування рекомендацій у вигляді передбачення контекстної послідовності. Цей підхід використовує послідовність контекстних дій користувача, а також поточний контекст, щоб передбачити ймовірність наступної дії. У прикладі Netflix, маючи одну послідовність для кожного користувача - країну, пристрій, дату та час перегляду фільму - вони навчили модель передбачати, що дивитися далі.

На даний момент існує багато систем рекомендацій на основі моди, таких як генерація вбрання відповідно до кольору тіла користувача та рекомендація вбрання користувачеві відповідно до нагоди, наприклад, весілля, побачення тощо, обидві системи використовують ручні анотовані дані. На відміну від кількості додатків для рекомендації одягу на основі кольору, існує мало досліджень, які б аналізували одяг, що людина може вдягнути в різну погоду, а потім рекомендувати його.

Для даного завдання був вибраний підхід не класичної рекомендаційної системи. Натомість використовувався табличний класифікатор, який чудово допоміг дати користувачам вдалу початкову рекомендацію одягу, базуючись на тому, яке вбрання було попередньо додане у шафу. У цій роботі, я припускаю, що використання табличного класифікатора є найефективнішим способом для початкової рекомендації одягу користувачам. Традиційні системи рекомендацій, які часто базуються на спільній фільтрації або методах на основі контенту, не враховують складну взаємодію із фактором погоди. Табличний класифікатор пропонує потужну альтернативу, фіксуючи ці взаємозв'язки у структурованому та

зрозумілому вигляді. Табличні дані складаються з рядків і стовпців, де кожен рядок представляє екземпляр, а кожен стовпець - ознаку. У контексті рекомендацій щодо одягу ознаки можуть включати погодні умови (температура, вологість, опади), атрибути одягу (тип, колір, матеріал). Таке структуроване представлення дає змогу всебічно зрозуміти взаємозв'язки між цими ознаками.

Табличні класифікатори мають кілька переваг для рекомендацій щодо одягу:

а. Зрозумілість: Табличні класифікатори, такі як дерева рішень і моделі на основі правил, містять зрозумілі для людини правила, що полегшує розуміння і пояснення обґрунтування рекомендацій щодо вбрання.

б. Гнучкість: Табличні класифікатори можуть враховувати різні типи даних, включаючи числові (наприклад, температуру), категоріальні (наприклад, тип одягу) та порядкові (наприклад, рейтинг переваг). Така гнучкість дозволяє більш точно відображати вподобання користувачів та погодні умови.

с. Масштабованість: Табличні класифікатори можуть обробляти великі набори даних, що робить їх придатними для великих колекцій предметів одягу та вподобань користувачів.

До табличних даних можна застосовувати різні алгоритми машинного навчання, зокрема Decision Tree, Random Forest, Boosted Tree чи Support Vector Machine, і підходи глибокого навчання, такі як вбудовування шарів у нейронні мережі. Ці методи дозволяють класифікатору вивчати складні, нелінійні взаємозв'язки між ознаками, що в кінцевому підсумку призводить до більш точних і релевантних рекомендацій щодо одягу.

1.4 Алгоритм, використаний для навчання табличного класифікатора

Для навчання моделі рекомендацій щодо одягу на основі типу одягу та погодних умов було використано алгоритм Random Forest, оскільки він має низку бажаних властивостей та переваг. Ось основні причини вибору Random Forest для цієї задачі:

1. Надійність: Random Forest - це ансамблевий метод, який будує кілька дерев рішень і об'єднує їхні прогнози. Такий підхід робить модель більш надійною і здатною краще узагальнювати невидимі дані.

2. Робота зі змішаними типами даних: Random Forest може ефективно обробляти змішані типи даних, такі як категоріальні (тип одягу, колір) та числові (температура, вологість) характеристики. Це має вирішальне значення для рекомендацій щодо одягу, оскільки дозволяє моделі вловлювати взаємозв'язки між різними типами даних.

3. Важливість ознак: Random Forest може автоматично оцінювати важливість ознак, надаючи уявлення про найбільш релевантні фактори, що впливають на рекомендації щодо одягу. Це може допомогти визначити ключові атрибути одягу та погодні умови, які суттєво впливають на вибір одягу.

4. Нелінійність і взаємодія: Random Forest може моделювати складні нелінійні взаємозв'язки та взаємодії між елементами. Ця можливість є важливою для рекомендацій щодо одягу, оскільки вона дозволяє моделі врахувати складні взаємозв'язки між типами одягу, вподобаннями користувача та погодними умовами. Це також корисно при роботі з великими наборами даних або при необхідності швидкого навчання моделі

1.5 Розробка та інтеграція додатків для iOS

Зростаюче поширення смартфонів, зокрема пристроїв на базі iOS, призвело до збільшення попиту на мобільні додатки, які задовольняють різноманітні потреби користувачів. Однією з таких потреб є можливість зберігати дані, наприклад одяг, чи рекомендації щодо одягу на основі таких факторів, як погодні умови та вподобання в одязі. Щоб задовольнити цей попит, розробники все частіше використовують методи машинного навчання, які можуть надавати користувачам точні та релевантні пропозиції.

Розробка додатків для iOS стала життєво важливим аспектом індустрії програмного забезпечення, оскільки пристрої Apple продовжують набирати популярність у всьому світі. Мільйони користувачів покладаються на iPhone та iPad для виконання повсякденних завдань, тому платформа iOS пропонує розробникам широкі можливості для створення інноваційних та цікавих додатків. Крім того, суворі стандарти контролю якості та кураторське середовище App Store заохочують розробників надавати пріоритет користувацькому досвіду та продуктивності додатків, що сприяє загальному успіху платформи.

При розробці системи рекомендацій одягу на основі машинного навчання для пристроїв iOS слід враховувати кілька факторів:

а. Збір та попередня обробка даних: повинен бути різноманітний набір даних, що містить релевантну інформацію про предмети одягу, погодні умови. Потім ці дані повинні бути попередньо оброблені, очищені та перетворені у відповідний формат для навчання та тестування моделі машинного навчання.

б. Вибір моделі та навчання: Вибір моделі машинного навчання відіграє вирішальну роль у роботі системи.

с. Оптимізація та оцінка моделі: точно налаштованими мають бути параметри моделі, аби оцінити її продуктивність, використовуючи відповідні метрики, такі як точність, достовірність, пригадування та оцінка F1(Оцінка F1 - це метрика, яка використовується для оцінки ефективності моделей класифікації, особливо в ситуаціях, коли існує дисбаланс у розподілі міток класів. Це середнє гармонійне значення точності та запам'ятовування, що враховує як хибно-позитивні, так і хибно-негативні результати. Оцінка F1 коливається від 0 до 1, де 1 означає ідеальну класифікацію, а 0 - найгіршу з можливих. Показник F1 розраховується за наступною формулою:
$$\text{Оцінка F1} = 2 * (\text{точність} * \text{відгук}) / (\text{точність} + \text{відгук}).$$
 Високий показник F1 вказує на те, що модель має хороший баланс між точністю і пригадуванням, ефективно ідентифікуючи релевантні екземпляри, мінімізуючи помилкові спрацьовування і помилкові несприятливі результати.). Цей процес допомагає гарантувати, що модель надає точні та релевантні рекомендації щодо екіпірування.

d. Користувацький інтерфейс та досвід: Користувацький інтерфейс додатку має бути візуально привабливим, простим у навігації та ефективно демонструвати доданий одяг.

Фреймворк Core ML від Apple спрощує інтеграцію моделей машинного навчання в додатки для iOS. Core ML підтримує різні типи моделей, і забезпечує безперешкодний спосіб перетворення та оптимізації моделей для розгортання на пристроях. Фреймворк дозволяє розробникам використовувати обчислювальну потужність пристроїв iOS і виконувати прогнозування в реальному часі на пристрої, пропонуючи користувачам швидкий і гнучкий досвід.

Інтеграція моделей машинного навчання в додатки для iOS є важливим компонентом даного дослідження. У цій роботі надано огляд розробки додатків для iOS та підтверджується її актуальність на сучасному ринку.

1.6 Висновки до розділу 1

В даному розділі було розглянуто мету та головну мотивацію для дослідження та створення додатку для мобільних пристроїв на базі iOS для оперування та рекомендації вбрання. Класифікатор об'єктів був обраним для класифікації типу одягу та його кольору. Табличні класифікатори є оптимальним рішенням для рекомендацій вбрання на основі погодних умов і переваг в одязі. Використовуючи переваги табличного представлення даних і можливості машинного навчання, ці класифікатори можуть надавати персоналізовані, контекстно-залежні рекомендації щодо вбрання, які відповідають унікальним смакам і вимогам користувачів. Random Forest було обрано для навчання моделі рекомендацій одягу на основі типу одягу та погодних умов, оскільки це надійний, універсальний та ефективний метод, який може врахувати складні взаємозв'язки в даних. Його здатність обробляти змішані типи даних, моделювати нелінійні взаємодії та надавати уявлення про важливість ознак робить його добре придатним для цього завдання.

Розділ 2. Використаний підхід до рішення проблеми

2.1 Core Data: фреймворк, для оперування базою даних вбрання

Core Data - це потужний фреймворк графів та постійності даних, який надається компанією Apple для додатків iOS та macOS. Він дозволяє розробникам керувати даними та забезпечувати ефективний доступ до них та їх зберігання. Для даного дослідження було вибрано Core Data як основний фреймворк для управління даними у проекті через його переваги над іншими альтернативами.

Переваги Core Data:

- Надає високорівневий рівень абстракції над SQLite, що дозволяє ефективно та оптимізовано здійснювати доступ до даних та їх зберігання.
- Core Data надає багатий набір функцій, включаючи об'єктне моделювання, керування версіями та міграцію схем, які полегшують роботу з даними.
- Він пропонує вбудовану підтримку збоїв та попередньої вибірки, що зменшує використання пам'яті та підвищує продуктивність.
- Core Data підтримує зв'язки між об'єктами, що дозволяє моделювати та підтримувати складні структури даних.
- Core Data забезпечує підтримку паралелізму, що уможливорює багато потоковий доступ до даних, забезпечуючи при цьому їхню цілісність.

Інші альтернативи для управління даними в iOS включають SQLite, Realm та Firebase. Хоча ці альтернативи мають свої переваги, Core Data часто надають перевагу завдяки інтеграції з екосистемою Apple та багатому набору функцій.

SQLite є популярним вибором для управління даними, але він вимагає більше ручного кодування для налаштування та підтримки, що може зайняти багато часу. Realm - новіша альтернатива, відома своєю продуктивністю та простотою використання, але вона має обмежену підтримку зв'язків між об'єктами. Firebase - це хмарна альтернатива, яка пропонує синхронізацію в режимі реального часу, але вона вимагає підключення до Інтернету і може бути дорогою для великомасштабних додатків.

Загалом, Core Data пропонує надійну та надійну основу для управління даними в додатках iOS, а її функції роблять її чудовим вибором для розробників, які хочуть ефективно та результативно керувати даними.

2.2 SwiftUI: фреймворк для написання користувацького інтерфейсу

Swift UI - це сучасний фреймворк від Apple для створення користувацьких інтерфейсів у додатках для iOS, macOS, watchOS і tvOS. Він використовує декларативний синтаксис для опису користувацького інтерфейсу, що полегшує читання та написання коду. Для даної роботи було вибрано SwiftUI як основний фреймворк інтерфейсу для iOS-дodatku для гардеробу з наступних причин:

Переваги SwiftUI:

- Його синтаксис простий і лаконічний, що полегшує читання і написання коду.
- Він надає декларативний синтаксис для побудови користувацьких інтерфейсів, що дозволяє легко міркувати про макет і поведінку інтерфейсу користувача програми.
- SwiftUI призначений для безперебійної роботи з іншими фреймворками Apple, такими як Core Data чи Combine. Ця

інтеграція полегшує створення надійних і масштабованих додатків.

- Він надає функцію попереднього перегляду, яка дозволяла бачити інтерфейс в реальному часі під час редагування коду, що прискорює процес розробки та зменшує ймовірність помилок.
- Swift UI пропонує автоматичне оновлення інтерфейсу, що дозволяє легко керувати станом програми та надавати користувачам зворотній зв'язок у режимі реального часу.

Інші альтернативи для створення інтерфейсу в iOS включають UIKit та Storyboards. Хоча ці альтернативи мають свої переваги, Swift UI пропонує значні переваги над цими варіантами.

UIKit - це зрілий і широко використовуваний фреймворк для створення користувацьких інтерфейсів в iOS додатках. Однак він вимагає більше коду для реалізації та управління інтерфейсом порівняно зі Swift UI, що може зайняти багато часу та призвести до помилок.

Storyboards - це візуальний редактор для створення користувацького інтерфейсу iOS-додатків. Однак вони можуть стати складними і важкими в управлінні тоді як додаток зростає в розмірі і складності. Декларативний синтаксис Swift UI та функція попереднього перегляду в реальному часі пропонують більш простий підхід до створення та управління інтерфейсом в додатках для iOS.

2.3 CoreML як спосіб інтеграції машинного навчання в iOS застосунки

У контексті додатку iDresser, що став результатом цього дослідження, методи машинного навчання були використані для класифікації типу та кольору одягу, а також для надання порад щодо вбрання у певний день. Серед переваг машинного навчання:

- Машинне навчання дозволяє автоматично вивчати закономірності та взаємозв'язки в даних, що може призвести до більш точних і надійних прогнозів.

- Воно може обробляти великі та складні набори даних, які людині було б важко обробити вручну.

У роботі було використано дві моделі класифікатора зображень для класифікації типу одягу та кольору. Ці моделі були навчені на основі набору зображень, що були попередньо поділені на класи, які були позначені правильним типом одягу кольором. Вибір відповідного фреймворку машинного навчання для додатку має вирішальне значення, оскільки він впливає на процес розробки, ефективність виконання та загальну продуктивність системи. Для розробки даного додатку для iOS, Core ML від Apple було використано як основний фреймворк. Це рішення зумовлене кількома причинами, починаючи від безперешкодної інтеграції Core ML з пристроями iOS і закінчуючи підтримкою різних типів моделей, що полегшує наші завдання класифікації. Дані причини є підтвердженням вибору Core ML як ідеального фреймворку для цього проекту:

Бездоганна інтеграція з екосистемою Apple: Оскільки Core ML є продуктом Apple, його природно розроблено для безперешкодної інтеграції з усіма пристроями в екосистемі Apple. Це робить його природним вибором для додатків, орієнтованих виключно на пристрої iOS. Він пропонує пряму підтримку різних бібліотек і фреймворків Apple, таких як Vision для аналізу зображень, Natural Language для обробки тексту і SoundAnalysis для аудіоданих. Наш додаток, який використовує класифікацію зображень за типом і кольором одягу, значно виграє від цієї інтеграції, спрощуючи процес розробки та покращуючи загальну продуктивність.

Машинне навчання на пристрої: Ще однією важливою перевагою Core ML є його здатність виконувати обчислення на пристрої. Це має два ключові наслідки: покращена продуктивність і підвищена

конфіденційність. Обробка даних безпосередньо на пристрої значно зменшує затримку, оскільки немає необхідності надсилати дані на сервер. Це гарантує, що додаток надає рекомендації та класифікує об'єкти майже в режимі реального часу, покращуючи користувацький досвід. З точки зору конфіденційності, обробка на пристрої означає, що дані користувача не потрібно передавати на сервер, тим самим знижуючи ризик витоку даних. Це особливо важливо у випадку даної роботи, оскільки користувачі можуть мати занепокоєння щодо конфіденційності, коли діляться вмістом свого гардеробу.

Окрім того, Core ML підтримує широкий спектр типів моделей машинного навчання, включаючи глибоке навчання, машини опорних векторів та узагальнені лінійні моделі. Це особливо корисно для даного додатку, який використовує як класифікатори зображень для класифікації одягу, так і табличний класифікатор для рекомендацій. Гнучкість Core ML дозволяє використовувати найбільш підходящий тип моделі для кожного завдання, тим самим оптимізуючи продуктивність системи.

Також є важливим те, що Core ML оптимізовано для роботи на пристроях Apple. Він може використовувати CPU, GPU та Neural Engine на чіпах Apple, розподіляючи завдання у найбільш ефективний спосіб, щоб максимізувати обчислювальну потужність та мінімізувати використання батареї. Це гарантує, що додаток працюватиме безперебійно та ефективно, забезпечуючи чудовий користувацький досвід.

Саме тому вибір Core ML як основного фреймворку машинного навчання для даної системи рекомендацій одягу на основі погоди виправданий його безшовною інтеграцією з екосистемою iOS, можливостями обчислень на пристрої, підтримкою декількох типів моделей, ефективними інструментами перетворення моделей та оптимізацією для пристроїв Apple. Всі ці функції в сукупності сприяють

створенню ефективного, зручного та безпечного для користувача додатку, який надає точні та своєчасні рекомендації.

2.4 Обґрунтування вибору CreateML для тренування моделей

Вибір відповідного інструменту для тренування моделей машинного навчання є важливим, оскільки він впливає на процес розробки, ефективність виконання та загальну продуктивність системи. Для розробки додатку для класифікації та рекомендації одягу на основі погодніх умов було обрано Create ML від Apple. Цей вибір ґрунтується на кількох причинах, в основному на ефективності для тренування класифікаторів зображень і табличних класифікаторів.

Одним з ключових переваг Create ML є його простота використання. Він надає зручний графічний інтерфейс, що дозволяє навчати моделі без необхідності писати велику кількість коду. Це є надважливо для даного проекту, оскільки це дозволяє швидко розробляти, тестувати і вдосконалювати моделі, що використовуються для класифікації одягу та кольору.

Create ML є частиною екосистеми Apple, і він бездоганно інтегрується з Core ML. Тренування моделей за допомогою Create ML гарантує, що вони будуть сумісні з Core ML, що спрощує процес розгортання. Крім того, моделі, навчені за допомогою Create ML, автоматично оптимізуються для найкращого виконання на пристроях Apple.

Create ML підтримує широкий спектр типів даних, включаючи зображення, текст, табличні дані, тощо. В даному випадку, це особливо корисно, оскільки використовуються зображення для класифікації типу та кольору одягу, а також табличні дані для формування рекомендацій. Create ML надає вбудовані інструменти, які спеціально розроблені для обробки цих типів даних, значно полегшуючи процес тренування.

Create ML також використовує потужні алгоритми машинного навчання, які автоматично адаптуються до даних. Він використовує передові методи, такі як трансферне навчання для зображень і тексту, що дозволяє нам використовувати передтреновані моделі як вихідний пункт, зменшуючи час і обчислювальні ресурси, необхідні для тренування моделей.



Рисунок 1. Результат тренування моделі класифікації типу вбрання в CreateML

Він автоматично налаштовує параметри моделі на основі даних, що було внесено, що звільняє розробників від необхідності ручного налаштування. Це не тільки пришвидшує процес тренування, але і збільшує шанси на досягнення високої точності моделі. Окрім цього, даний ресурс включає інструменти для валідації та оцінки моделі прямо під час процесу тренування. Якщо розглядати цю можливість на прикладі класифікації вбрання – розробнику дається можливість надати картинку для тестування моделі відразу в ресурсі, аби перевірити, чи задовольняє для роботи результат. Це дозволяє безпосередньо відстежувати якість моделей і робити необхідні корективи в процесі тренування.

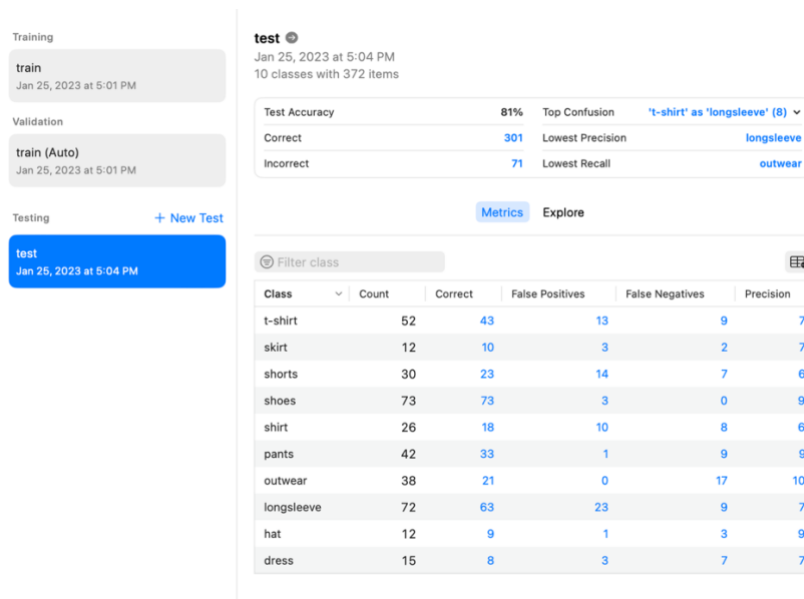


Рисунок 2. Результати тестування роботи моделі класифікації типу вбрання в CreateML

2.5 Висновки до розділу 2

Для збереження даних про вибір одягу та рекомендацій було використано фреймворк Core Data. Це дозволило зручно зберігати дані та маніпулювати ними в додатку.

Для розробки користувацького інтерфейсу було використано фреймворк SwiftUI. Він дозволив швидко та легко створити ефективний інтерфейс користувача, що дозволяє взаємодіяти з додатком та отримувати рекомендації з використанням машинного навчання.

Вибір Create ML як інструменту для тренування моделей машинного навчання є обґрунтованим, враховуючи його простоту використання, інтеграцію з Core ML, підтримку різних типів даних, потужні алгоритми машинного навчання, автоматичне налаштування параметрів та вбудовані інструменти валідації та оцінки моделі.

Розділ 3. Імплементация обраних технологій та створення застосунку

3.1 Навчання та тестування моделей класифікації типу та кольору вбрання

Навчальна модель - це набір даних(частіше використовується термін датасет), який використовується для навчання алгоритму ML. Вона складається з вихідних даних вибірки та відповідних наборів вхідних даних, які впливають на вихідні дані. Навчальна модель використовується для того, щоб пропустити вхідні дані через алгоритм і зіставити оброблений результат з вихідними даними вибірки. Результат цієї кореляції використовується для модифікації моделі.

Навчання моделі на машинній мові - це процес надання алгоритму ML даних, які допомагають ідентифікувати і вивчити хороші значення для всіх задіяних атрибутів.

Навчання під контролем можливе, коли навчальні дані містять як вхідні, так і вихідні значення. Кожен набір даних, що містить вхідні дані та очікуваний вихід, називається навчальним сигналом. Навчання відбувається на основі відхилення обробленого результату від задокументованого результату, коли вхідні дані подаються на модель.

Для розробки застосунку iDresser, що є практичною демонстрацією даної роботи, використовуються дві моделі машинного навчання для класифікації типу та кольору вбрання, що були навчені контрольованим типом навчання. Перш ніж почати тренування моделі, було зібрано набір даних для класифікації типу та кольору вбрання. Для створення моделі для класифікації типу одягу було використано датасет, із більш ніж п'ятьма тисячами фотографій різного типу одягу. Датасет був поділений на такі класи як “dress”, “hat”, “longsleeve”, “winter outdoor”, “pants”, “shoes”,

“shorts”, “skirt”, “t-shirt”, “demi-season outerwear”. Для створення моделі для класифікації кольору було використано датасет із більш ніж шести тисячами фотографій вбрання різного одягу. Датасет було поділено на дванадцять різних кольорів для якнайбільш точної класифікації кольору вбрання.

Після того, як було зібрано набір даних, було розпочато підготовку даних для тренування моделей. Спочатку було розділено набір даних на тренувальний та тестовий набори. Використовуючи Create ML для тренування моделей машинного навчання, для класифікації як типу вбрання, так і кольору вбрання було використано модель Image Classifier. Провівши навчання моделі на тренувальному наборі даних, можна було оцінити її точність на тестовому наборі даних. За допомогою Create ML була можливість налаштувати параметри моделі та провести валідацію для покращення її точності.

Після тренування моделей, було оцінено їх точність на тестовому наборі даних. Для класифікації типу вбрання модель Image Classifier показала точність на рівні 83%, що є доволі високим показником. Для класифікації кольору вбрання наша модель Tabular Classifier показала точність на рівні 63,2%, що також є дуже хорошим результатом.

Важливим аспектом розробки моделі є її перевірка та оцінка в реальному середовищі. Важливо провести ряд тестів, щоб переконатися, що модель працює ефективно та точно в контексті відповідного додатку. Це включає в себе перевірку точності рекомендацій, швидкості відгуку, а також перевірку того, як модель справляється з новими даними. Для цього можна було використано можливість Preview в програмі, завантаживши реальну власну фотографію. Таким чином можна перевірити, чи задовольняє відповідь моделі.

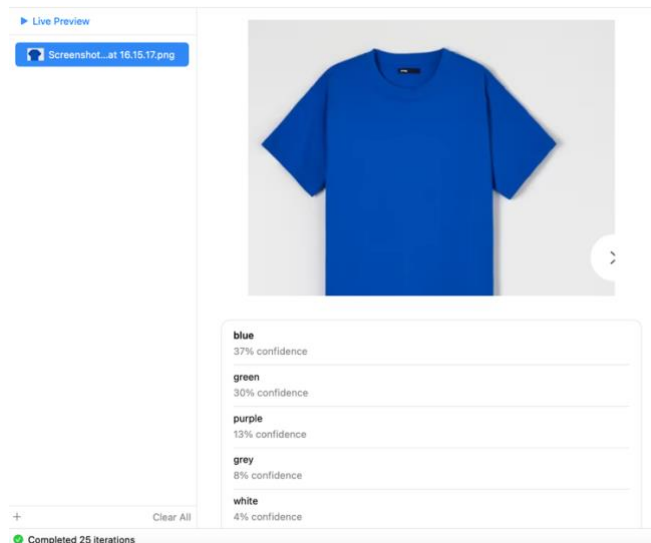


Рисунок 3. Тестування моделі класифікації вбрання використовуючи Preview

3.2 Навчання та тестування табулярного класифікатора для рекомендації вбрання в контексті погоди

Табулярні дані - це структуровані дані, що представлені в рядках і стовпцях, аналогічно тому, як дані відображаються в табличному форматі, як-то електронні таблиці. Кожен рядок представляє окремий випадок або спостереження, а стовпці представляють характеристики цих спостережень. В даному випадку, будуть використовуватись табулярні дані для представлення різноманітних атрибутів одягу, таких як тип, а також погодні умови, такі як температура.

Apple CreateML - це потужний інструмент для машинного навчання, що дозволяє розробникам легко тренувати моделі безпосередньо на Mac, використовуючи Swift або її графічний інтерфейс. CoreML, з іншого боку, - це фреймворк, що дозволяє використовувати ці моделі в iOS, macOS, watchOS та tvOS додатках. Використовуючи обидва інструменти, можна

створювати і використовувати моделі, оптимізовані для швидкості та ефективності на пристроях Apple.

Для тренування табулярного класифікатора вбрання було використано CreateML. Дані для тренування складаються з атрибутів одягу, зазначених вище, а також даних про погоду.

Процес тренування включає в себе кілька кроків:

Підготовка даних: Дані перетворюються в формат, придатний для CreateML. Для створення моделі Tabular Classifier, дані повинні бути подані у, наприклад, .csv чи .json файлі, де сепаратором буде кома. Для цього можна також використати будь-який додаток на зразок Microsoft Excel, звідки потім буде експортований файл у потрібному форматі. В даному випадку було використано ресурс від Apple – Numbers. Набором даних для реалізації додатку стала таблиця, яка містила стовпець із вбранням та температурою, коли це вбрання носилось. Стовпець із вбранням містить у собі одразу два типи одягу, який згодом ділиться на верхній атрибут одягу та нижній.

Очищення та представлення даних: На цьому етапі очищуються усі дані від шуму та виключаємо неповні або невідповідні записи. Також нормалізуються дані, якщо це необхідно, та перетворюються у формат, який придатний для моделі.

Розділення даних: Дані діляться на тренувальний та тестовий набори. Тренувальний набір використовується для тренування моделі, а тестовий набір використовується для перевірки її точності.

Тренування моделі: Використовуємо CreateML для тренування нашої моделі на основі тренувального набору даних. Це включає в себе визначення структури моделі, вибір відповідного алгоритму та оптимізацію параметрів.

Тестування та оцінювання моделі: Після тренування моделі іде перевірка на її точність, використовуючи тестовий набір даних. Також оцінюється її продуктивність та робляться необхідні корективи.

Експорт та інтеграція моделі: Після того, як модель була натренована та перевірена, її можна експортувати в формат `.mlmodel` для подальшого використання в CoreML.

Під час тренування моделі Tabular Classifier потрібно також вказати атрибути, які модель повинна аналізувати, а також цільовий атрибут (`target`), який буде слугувати вихідним результатом роботи моделі. У цьому випадку модель аналізує температуру, а вихідним параметром є саме вбрання.

3.3 Інтеграція моделей в iOS застосунок

Після тренування моделі, вона екпортується в формат `.mlmodel` для інтеграції в iOS додаток. За допомогою CoreML, ми можемо використовувати цю модель для аналізу зображень вбрання, що завантажуються користувачами, і визначення їх типу та кольору.

Однак, інтеграція Image Classifier може мати деякі труднощі:

- **Обробка зображень:** Зображення, які завантажуються користувачами, можуть мати різний розмір, орієнтацію та якість. Це може вимагати додаткової обробки, щоб перетворити ці зображення в формат, який придатний для моделі.

- **Точність моделі:** Точність моделі може варіюватися в залежності від якості зображення та різноманітності вбрання. Наприклад, модель може мати складнощі при визначенні кольору вбрання при поганому освітленні або при визначенні типу вбрання, якщо воно частково приховане.

Для зручності інтеграції моделей класифікації типу та кольору вбрання було створено клас `VissionClassifier`. В даному класі відбувається

ініціалізація моделі, а також знаходиться головний метод `classify`. Даний клас є універсальним, тому в реалізації додатку його можна використовувати для будь-якої моделі типу `Image Classifier`. Оскільки і класифікація типу, і кольору відбувається моделями цього типу, клас `VisionClassifier` використовується для них обох.

`Tabular Classifier`, модель, яка використовує табулярні дані для видачі рекомендацій вбрання в додатку також інтегрується її в iOS застосунок за допомогою `CoreML`.

Така інтеграція також може мати деякі труднощі:

- Точність моделі: Як і з `Image Classifier`, точність `Tabular Classifier` також може варіюватися. Це може бути зумовлено різноманітністю даних, неоднорідністю вхідних даних або недостатнім обсягом даних для тренування.

Для інтеграції моделі в додаток було написано клас `ClothingRecommender`, де здійснюється ініціалізація моделі, а результат поділяється на верхню частину вбрання та нижню. Поділ результату здійснюється для того, аби зручніше шукати відповідний одяг у базі даних вбрання, яке користувач попередньо зберіг у додаток.

3.4 Застосунок `iDresser` як результат дослідження

Кульмінацією цього дослідження стала розробка застосунку для iOS під назвою `iDresser`, який використовує моделі машинного навчання для класифікації одягу та надання рекомендацій щодо вбрання. Додаток складається з двох основних екранів: "Гардероб" та "Історія вбрання", і використовує `Core Data` від Apple для локального збереження даних.

Екран "Гардероб" слугує основним інтерфейсом для керування елементами одягу. Тут користувачі можуть додавати, редагувати, видаляти

та зберігати свій одяг. Для додавання одягу користувач має можливість використовувати фотоплівку, якщо до неї було надано дозвіл, чи камеру, аби фотографувати річ в реальному часі. Коли додається новий предмет одягу, зображення автоматично проходить через інтегровані дві моделі класифікатора зображень. Моделі класифікують річ за типом (наприклад, футболка, штани, сукня) і кольором, а потім призначає її до відповідної категорії або "полиці" в цифровому гардеробі.

Однією з ключових особливостей екрану "Гардероб" є кнопка "Рекомендація". Якщо попередньо було надано дозвіл до локації користувача, під час роботи додатку, він використовує цю інформацію, щоб отримати поточні дані про температуру з WeatherAPI. Ці дані, разом з категоризованим одягом з гардеробу користувача, проходять через інтегровану модель табличного класифікатора, яка генерує рекомендацію щодо вбрання, що відповідає поточним погодним умовам. Якщо потрібного одягу не було додано до бази даних попередньо – користувачу буде надана можливість додати потрібний одяг, у вигляді кнопки.

Екран "Історія вбрання" містить записи про вбрання, які користувач носив відповідно до рекомендацій програми. Кожне вбрання має позначку дати і може бути переглянуте в деталях, що може слугувати джерелом натхнення для майбутнього вибору, а також вбрання записується у .json файл із майбутньою перспективою додати можливість персоніфікованих рекомендацій для користувача. Якщо користувач забув додати вбрання, що носив сьогодні – додаток нагадуватиме це зробити, використовуючи локальні пуш-нотифікації.

Всі елементи одягу та історія вбрання зберігаються локально на пристрої за допомогою Core Data, платформи Apple для управління та збереження даних. Це забезпечує надійний та ефективний засіб управління даними користувача, а також гарантує, що додаток залишається функціональним навіть в автономному режимі. Зміни, внесені на екранах

"Гардероб" або "Історія вбрання", негайно відображаються в сховищі Core Data, гарантуючи, що дані користувача залишаються актуальними та узгодженими в усіх частинах програми.

Цей додаток для iOS є практичним застосуванням проведеного дослідження щодо використання моделей машинного навчання для категоризації одягу та рекомендацій щодо вбрання. Він надає користувачам інтерактивну, зручну платформу для управління своїм гардеробом і отримання рекомендацій щодо вбрання, а також демонструє потенціал інтеграції моделей машинного навчання в мобільні додатки. Подальша робота може бути спрямована на подальше вдосконалення алгоритму рекомендацій, збільшення набору даних для більш точної класифікації типу та кольору вбрання, покращення користувацького інтерфейсу та створення моделі, що буде навчатись на даних із історії вбрання користувача для надання ще більш персоналізованих і точних рекомендацій.

3.5 Висновки до розділу 3

У цьому розділі було описано тренування та тестування моделей класифікації типу та кольору вбрання за допомогою Create ML. Збір та підготовка даних, налаштування параметрів моделей та проведення валідації допомогли досягти доволі високої точності моделей.

Моделі були використані в додатку, щоб забезпечити точну класифікацію вбрання та релевантну рекомендацію для користувачів. Фреймворк Core ML дозволив ефективно розробити та впровадити наші моделі в додаток, що забезпечує високу продуктивність та точність.

Практичним застосуванням дослідження став додаток iDresser, що надає користувачам інтерактивну, зручну платформу для управління своїм гардеробом і отримання рекомендацій щодо вбрання.

Висновки

У цій роботі досліджено використання машинного навчання (ML) для створення додатку для iOS, призначеного для зберігання одягу та рекомендацій щодо вбрання. Дослідження почалося з детального вивчення існуючої літератури про застосування машинного навчання у сфері моди, категоризації одягу та рекомендацій щодо вбрання. Це дослідження дозволило отримати чітке уявлення про сучасний стан справ у цих галузях, а також виявити прогалини та можливості для власного застосування.

Це дозволило перейти далі до практичного аспекту роботи, розробивши дві ML-моделі з використанням фреймворку Core ML від Apple: модель класифікатора зображень для категоризації типів одягу та кольорів, а також модель табличного класифікатора для рекомендацій одягу. Ці моделі було навчено на відповідних даних, включаючи зображення предметів одягу та табличні дані, що представляють різні фактори, такі як тип одягу, колір та погодні умови.

Згодом ці моделі були інтегровані в додаток для iOS. Додаток, що складається з двох основних екранів - "Гардероб" та "Історія вбрання", надає користувачам можливість керувати своїм гардеробом та отримувати рекомендації щодо вбрання на основі їхніх предметів одягу та поточних погодних умов.

Ця курсова робота робить внесок в існуючу сукупність знань декількома способами. Вона надає практичний приклад того, як машинне навчання може бути застосоване в контексті мобільного додатку для надання корисного сервісу - в даному випадку, управління гардеробом і надання рекомендацій щодо вбрання. Воно також сприяє розумінню викликів і рішень, пов'язаних з розробкою та інтеграцією моделей машинного навчання в мобільний додаток.

Крім того, це дослідження має практичне значення як для користувачів, так і для розробників. Для користувачів розроблений додаток надає інструмент для управління своїми предметами одягу та отримання персоналізованих рекомендацій щодо вбрання. Для розробників ця курсова робота надає детальний опис процесу розробки та інтеграції моделей машинного навчання в iOS-додаток, який може слугувати керівництвом для подібних проектів.

Незважаючи на успішну розробку та інтеграцію моделей машинного навчання в iOS-додаток, це дослідження не позбавлене недоліків. Наприклад, точність моделей залежить від якості та різноманітності навчальних даних.

Подальша робота може бути зосереджена на вдосконаленні моделей шляхом розширення навчальних даних або дослідження більш складних алгоритмів. Також можна вдосконалити користувацький інтерфейс додатку, наприклад, додати більше інтерактивних елементів або функцій персоналізації. Нарешті, інтеграція інших джерел даних або сервісів, таких як тренди в соціальних мережах, може забезпечити більш контекстно-орієнтовані та персоналізовані рекомендації щодо вбрання.

Отже, ця курсова робота продемонструвала потенціал машинного навчання у розробці практичного і зручного додатку для iOS для управління гардеробом і рекомендацій щодо вбрання. Успіх цього проекту ілюструє, як можна використовувати машинне навчання для покращення функціональності та зручності мобільних додатків, прокладаючи шлях для майбутніх досліджень і розробок у цій галузі.

Список покликань

1. Apple Inc. Core ML Documentation [Електронний ресурс] / Apple Inc. – Режим доступу до ресурсу:
<https://developer.apple.com/documentation/coreml>.
2. Hollemans M. Core ML Survival Guide / Matthijs Hollemans., 2018. – 363 с.
3. Yamaguchi K, Kiapour MH, Ortiz LE, Berg TL (2012) Parsing clothing in fashion photographs. In: 2012 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 3570–3577
4. Chaudhari, K., Thakkar, A. A Comprehensive Survey on Travel Recommender Systems. Arch Computat Methods Eng 27, 1545–1571 (2020). <https://doi.org/10.1007/s11831-019-09363-7>
5. Ramesh, Nikita, "Outfit Recommender System" (2018). Master's Projects. 611. DOI: <https://doi.org/10.31979/etd.8c8x-txe7>
https://scholarworks.sjsu.edu/etd_projects/611
6. Kaur, Er & Kaur, Yadwinder. (2014). Object classification Techniques using Machine Learning Model. International Journal of Computer Trends and Technology. 18. 170-174. 10.14445/22312803/IJCTT-V18P140.
7. Oden T. What Is Model Training? [Електронний ресурс] / Technologies Oden – Режим доступу до ресурсу: <https://oden.io/glossary/model-training/>.
8. Huang Y. Outfit Recommendation System Based on Deep Learning / Huang Ying, 2016. – 164 с.