Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

Бакалаврська дипломна робота на тему:

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ РІВНІВ ЗАДОВОЛЕНОСТІ ПАСАЖИРІВ АВІАКОМПАНІЯМИ»

Виконала: студентка 4 курсу, групи 2ІСТ-20б спеціальності 126 – «Інформаційні системи та технології»
_____ Анна СУДЕЦЬ
Керівник: к.т.н., доц. каф. САІТ
_____Олексій КОЗАЧКО
«31» _____ 2024 р.
Рецензент: к.т.н., доц. каф. КН
_____Володимир ОЗЕРАНСЬКИЙ
«19» _____ 2024 р.

Допущено до захисту
Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
«03» 06 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність 126 – «Інформаційні системи та технології»
Освітньо-професійна програма – Прикладні інформаційні технології

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
д.т.н., проф. Віталій МОКІН
"05" 03 2024 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТЦІ
Судець Анні Олександрівні

1. Тема роботи: «Інформаційна технологія передбачення рівнів задоволеності пасажирів авіакомпаніями»
керівник роботи: Козачко О. М., к.т.н., доц. каф. САІТ
затверджені наказом ВНТУ від «11» 03 2024 року №80
2. Термін подання студентом роботи 31.05
3. Вихідні дані до роботи:
Дані про рівні задоволеності пасажирів авіакомпаніями.
4. Зміст текстової частини:
1) Характеристика об'єкту досліджень;
2) Аналіз методів машинного навчання;
3) Підготовка даних та розвідувальний аналіз;
4) Розроблення інформаційної технології передбачення рівнів задоволеності пасажирів авіакомпаніями;
5. Перелік ілюстративного матеріалу:
1) Кореляційна матриця показників;
2) Блок-схема алгоритму роботи інформаційної технології;
3) Матриці плутанини моделі оптимальних моделей;
4) Класифікаційний звіт оптимальних моделей.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 11.03

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Характеристика об'єкту досліджень	11.03	31.03	Вик
2	Аналіз методів машинного навчання	01.04	15.04	Вик
3	Проведення розвідувального аналізу	16.04	30.04	Вик
4	Розроблення інформаційної технології передбачення рівнів задоволеності пасажирів авіакомпаніями	01.05	15.05	Вик
5	Оформлення матеріалів до захисту БДР	16.05	31.05	Вик

Студентка _____ Анна СУДЕЦЬ

Керівник роботи _____ Олексій КОЗАЧКО

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 74 сторінок формату А4, на яких є 44 рисунків, список використаних джерел містить 20 найменувань.

Метою дослідження є підвищення точності передбачення рівнів задоволеності пасажирів авіакомпаніями, шляхом створення інформаційної технології та використання методів машинного навчання.

Під час розробки було здійснено огляд існуючих систем, підготовлено дані для подальшого аналізу, проведено розвідувальний аналіз даних, розроблено інформаційну технологію, яка використовує моделі машинного навчання та здійснено передбачення. Також було оцінено результати роботи моделей.

Ключові слова: інформаційна технологія, розвідувальний аналіз, передбачення, машинне навчання.

ABSTRACT

The bachelor thesis consists of 74 pages of A4 format, on which there are 44 figures, the list of used sources contains 20 links.

The purpose of the study is to increase the accuracy of predicting the levels of passenger satisfaction with airlines by creating information technology and using machine learning methods.

During development, an overview of existing systems was performed, data was prepared for further analysis, data intelligence analysis was performed, information technology was developed that uses machine learning models, and predictions were made. The results of the models were also evaluated.

Keywords: information technology, intelligence analysis, prediction, machine learning.

ЗМІСТ

ВСТУП

Актуальність теми. В сучасному світі авіаційні перевезення стають все більш популярними, а конкуренція між авіакомпаніями зростає. Для утримання конкурентної переваги та забезпечення високого рівня обслуговування, авіакомпанії повинні розуміти та передбачати потреби і очікування своїх пасажирів. Задоволеність пасажирів є критичним рівнем, що впливає на їх лояльність та подальше використання послуг авіакомпанії.

Традиційні методи аналізу задоволеності пасажирів часто не можуть впоратися з великими обсягами даних та складністю взаємозв'язків між різними рівнями, що впливають на задоволеність. Тому виникає потреба у розробці ефективних інформаційних технологій, які дозволяють здійснювати глибокий аналіз та точне передбачення рівнів задоволеності пасажирів.

Використання сучасних методів аналізу даних та машинного навчання може покращити точність передбачень, що в свою чергу дозволить авіакомпаніям швидко реагувати на зміни в потребах пасажирів, удосконалювати якість послуг та підвищувати рівень задоволеності клієнтів. Це не тільки допоможе збільшити прибутковість авіакомпаній, але й сприятиме підвищенню якості обслуговування в галузі авіаперевезень в цілому.

Таким чином, розробка інформаційної технології для передбачення рівнів задоволеності пасажирів авіакомпаніями є актуальною та своєчасною задачею, яка має практичний і науковий інтерес.

Мета і задачі дослідження. Метою дослідження є підвищення точності передбачення рівнів задоволеності пасажирів авіакомпаніями, шляхом створення інформаційної технології та використання методів машинного навчання.

Для досягнення поставленої мети необхідно розв’язати такі задачі:

- здійснити огляд існуючих систем;
- підготувати дані для подальшого аналізу;
- провести розвідувальний аналіз даних;

– розробити інформаційну технологію, яка використовує моделі машинного навчання та здійснити передбачення;

– оцінити результати роботи моделей.

Об’єктом дослідження бакалаврської дипломної роботи є процес розроблення інформаційної технології передбачення рівнів задоволеності пасажирів авіакомпаніями.

Предметом дослідження бакалаврської дипломної роботи є інформаційна технологія передбачення рівнів задоволеності пасажирів авіакомпаніями.

Публікації. За результатами роботи було опубліковано тези на тему «Інформаційна технологія передбачення рівнів задоволеності пасажирів авіакомпаніями» на міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (м. Вінниця, 2023-2024 рр.) [1].

1 ХАРАКТЕРИСТИКА ОБ'КТУ ДОСЛІДЖЕНЬ

1.1 Аналіз предметної області

Однією з найважливіших частин світової економіки є сфера послуг. Наприклад, у Сполучених Штатах протягом останньої чверті 20 століття сфера послуг створювала близько 70% нових робочих місць і близько 60% річного ВВП. Як підрозділ сфери послуг, обслуговування авіакомпаній також швидко розширилося в останні роки. Згідно зі звітом Міжнародної асоціації повітряного транспорту (IATA), очікується, що глобальна кількість пасажирів сягне 7,2 мільярда в 2035 році, що приблизно втричі збільшить попит порівняно з 3,8 мільярда пасажирів, які літали в 2023 році, при середньорічному зростанні близько 3,7%.

Швидкий розвиток систем управління та постійне вдосконалення технологій сприяли швидкому зростанню та поширенню стандартизації сфери послуг. Як наслідок, зараз зменшується розбіжність між підприємствами щодо фізичного обладнання та якості послуг, а стандартизація продуктів стає все більш очевидною. Таким чином, клієнти можуть вибирати з кількох варіантів подорожі, посилюючи конкуренцію в авіаційній галузі. Посилення конкуренції в галузі спонукає авіакомпанії докладати великих зусиль для підвищення якості послуг, щоб покращити задоволеність пасажирів і залучити клієнтів від інших постачальників.

Відомо, що якість обслуговування є ключовим чинником задоволеності в авіакомпанії, і було доведено, що задоволення має вирішальне значення для збереження поточних клієнтів і залучення нових. Таким чином, вивчення того, як якість послуг авіакомпанії впливає на задоволеність мандрівників, є однією з найпопулярніших тем в авіаційній галузі [2].

Розуміння впливу якості обслуговування в галузі авіаперевезень має вирішальне значення не лише для підвищення доходів бізнесу, але й завдяки унікальній природі задоволеності клієнтів. Наприклад, у 2021 було досліджено вплив якості обслуговування на задоволеність пасажирів авіакомпанією та

виявили, що коли стандарти обслуговування спочатку низькі, додаткові послуги можуть призвести до помітного підвищення рівня задоволеності клієнтів. Однак із підвищенням рівня обслуговування додатковий вплив на задоволеність зменшується.

Також було виявлено, що позитивне ставлення працівників та оперативні дії щодо відновлення обслуговування викликають більше позитивних емоцій у пасажирів низькобюджетних авіакомпаній та пасажирів економ-класу. З іншого боку, для мандрівників, які користуються послугами авіакомпаній з повним набором послуг, труднощі в обслуговуванні та своєчасні, надійні й задовільні заходи щодо відновлення мають більший вплив. Як наслідок, існує значна різниця в сприйнятті якості послуг різними групами споживачів.

Таким чином, розвиток інформаційних технологій для передбачення рівнів задоволеності пасажирів авіакомпаніями є важливою складовою для підвищення конкурентоспроможності та забезпечення високої якості обслуговування в авіаційній галузі [2].

1.2 Аналіз актуальності

Актуальність теми дипломної роботи щодо прогнозування рівнів задоволеності пасажирів авіакомпаній в Україні під час війни може викликати певні сумніви. На перший погляд, війна змінює пріоритети як держави, так і громадян, і здається, що дослідження задоволеності пасажирів не є першочерговим завданням. Однак, більш детальний аналіз дозволяє виявити декілька важливих аспектів, що обґрунтовують актуальність цієї теми навіть у такі складні часи.

Ситуація в авіаційній галузі під час війни:

– Зміни у транспортних маршрутах: Війна спричинила значні зміни в транспортній інфраструктурі країни, зокрема в авіаційній галузі. Частина повітряного простору може бути закрита, що впливає на маршрути польотів і логістику. Аналіз задоволеності пасажирів дозволяє адаптуватися до цих змін,

забезпечуючи кращу організацію авіаперевезень;

– Безпека і надійність: У воєнний час питання безпеки стають надзвичайно актуальними. Важливо зрозуміти, як фактори безпеки впливають на задоволеність пасажирів і які заходи можна впровадити для підвищення довіри до авіакомпаній;

– Евакуація і гуманітарна допомога: Авіакомпанії відіграють важливу роль у процесі евакуації населення та доставки гуманітарної допомоги. Дослідження задоволеності пасажирів допомагає виявити критичні фактори, що впливають на ефективність цих процесів.

Соціально-економічні аспекти:

– Психологічний комфорт: Війна впливає на психологічний стан людей. Дослідження задоволеності пасажирів допомагає зрозуміти, які фактори можуть покращити їхній психологічний комфорт під час польотів;

– Підтримка авіаційної галузі: Підтримка національної авіаційної галузі в складний період є важливою для економіки. Аналіз задоволеності пасажирів сприяє поліпшенню послуг, що, в свою чергу, підтримує стабільність і розвиток авіакомпаній.

Технологічний аспект:

– Інновації та адаптація: Використання сучасних інформаційних технологій для прогнозування задоволеності пасажирів дозволяє авіакомпаніям швидко адаптуватися до нових умов і викликів, забезпечуючи високу якість обслуговування;

– Оптимізація ресурсів: В умовах війни важливо оптимально використовувати ресурси. Інформаційні технології можуть допомогти визначити найважливіші рівні задоволеності пасажирів і спрямувати ресурси на їх покращення [3].

Незважаючи на складну ситуацію в Україні через війну, тема дипломної роботи «Інформаційна технологія передбачення рівнів задоволеності пасажирів авіакомпаніями» залишається актуальною. Вона сприяє покращенню якості обслуговування пасажирів, підвищенню безпеки та надійності авіаперевезень,

підтримці національної авіаційної галузі та забезпеченню психологічного комфорту громадян. Таким чином, дослідження в цій сфері має значний потенціал для позитивного впливу на соціально-економічну ситуацію в країні.

Станом на 2024 рік, українські авіакомпанії, які продовжують свою діяльність на території інших країн під час війни, включають:

– SkyUp Airlines: Ця авіакомпанія, одна з найбільших в Україні, продовжує виконувати рейси за кордоном, зокрема в Європу, Азію та інші регіони. Вони також займаються чартерними перевезеннями. За контрактами ACMI та за чартерною програмою для туроператорів у SkyUp Airlines протягом 2023 року виконали 10 528 рейсів, перевезли понад 1,5 млн пасажирів та перерахували державі майже 200 млн грн податків. За показником виручки SkyUp стала провідною авіакомпанію України;

– Ukraine International Airlines (UIA): Незважаючи на складні умови, UIA продовжує здійснювати міжнародні рейси. Вони також займаються репатріаційними рейсами, допомагаючи українцям повернутися додому з-за кордону;

– Windrose Airlines: Ця авіакомпанія також здійснює міжнародні рейси, включаючи чартерні перевезення до різних країн світу;

– Azur Air Ukraine: Спеціалізується на чартерних рейсах, переважно в туристичні напрямки. Вони також виконують рейси за кордоном, підтримуючи українських туристів та громадян.

Ці авіакомпанії докладають зусиль для підтримки своєї діяльності в умовах війни, забезпечуючи пасажирів необхідними транспортними послугами на міжнародних маршрутах [3].

Діяльність українських авіакомпаній за кордоном під час війни має кілька важливих переваг для України:

– Економічна підтримка: Продовження діяльності українських авіакомпаній за кордоном забезпечує надходження валютних коштів в економіку країни, що є особливо важливим у часи війни. Це допомагає підтримувати стабільність національної валюти та економіки загалом;

– Робочі місця: Збереження роботи для працівників авіаційної галузі, що є критично важливим під час війни. Навіть якщо операції здійснюються за кордоном, це все одно підтримує працевлаштування українців;

– Репутація та партнерства: Продовження діяльності авіакомпаній допомагає зберегти та покращити репутацію української авіації на міжнародному рівні. Це сприяє розвитку партнерських відносин з іншими країнами та авіакомпаніями, що може бути корисним для України в довгостроковій перспективі;

– Гуманітарна допомога: Виконання репатріаційних рейсів та гуманітарних місій дозволяє авіакомпаніям допомагати українцям за кордоном, а також транспортувати необхідні товари та допомогу в Україну;

– Підтримка туризму: Хоча війна значно знижує рівень туризму, підтримка авіаційних зв'язків з іншими країнами допомагає зберегти туристичний потік в перспективі, коли ситуація стабілізується. Це також важливо для українських туристів, які подорожують за кордон;

– Розвиток нових ринків: Вимушене переміщення діяльності за кордон може відкрити нові ринки та можливості для українських авіакомпаній, що в перспективі може збільшити їх конкурентоспроможність і прибутковість [3].

Таким чином, діяльність українських авіакомпаній за кордоном під час війни є стратегічно важливою для підтримки економічної стабільності, соціальної підтримки населення та зміцнення міжнародних зв'язків.

1.3 Інформаційні технології рівнів задоволеності

Сучасні інформаційні технології відіграють вирішальну роль у зборі, аналізі та передбаченні рівнів задоволеності клієнтів. Завдяки використанню великих даних, машинного навчання та аналітики, авіакомпанії можуть глибше зрозуміти потреби і очікування пасажирів, швидко реагувати на зміни та покращувати якість своїх послуг.

У цьому підрозділі ми розглянемо основні додатки та платформи, які використовуються для оцінки і моніторингу рівнів задоволеності пасажирів. Ці

інструменти дозволяють авіакомпаніям отримувати зворотний зв'язок, аналізувати відгуки і визначати ключові області для покращення обслуговування, що, у свою чергу, сприяє підвищенню лояльності клієнтів і зміцненню їхньої конкурентоспроможності на ринку.

Skytrax – це один з найвідоміших рейтингів авіакомпаній, що використовує відгуки пасажирів для оцінки якості обслуговування. Їхні рейтинги включають різні аспекти задоволеності пасажирів, такі як комфорт сидінь, якість харчування, обслуговування на борту і багато іншого [4].

На рисунку 1.1 зображено головну сторінку Skytrax.

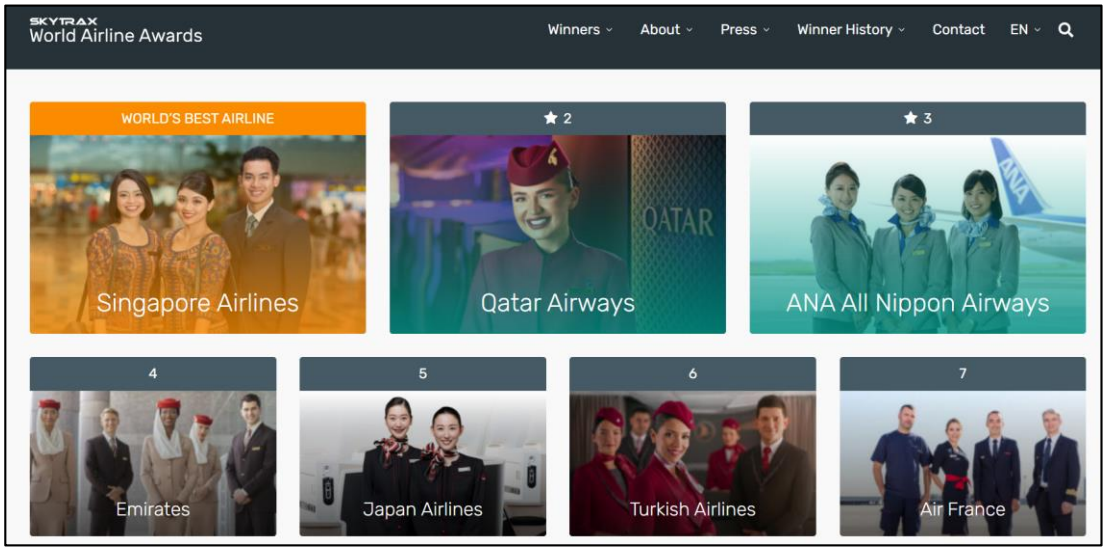

Рисунок 1.1 – Головна сторінка Skytrax

Переваги:

– Широке охоплення: Skytrax має глобальну базу даних з відгуками пасажирів про авіакомпанії з усього світу;

– Деталізовані рейтинги: Рейтинги враховують різні аспекти обслуговування, такі як комфорт сидінь, якість харчування, чистота, обслуговування на борту тощо.

Недоліки:

– Платне включення: Деякі користувачі скаржаться, що авіакомпанії можуть платити за включення в рейтинг, що може вплинути на неупередженість оцінок;

– Обмежена інформація: Хоча Skytrax надає детальні оцінки, інколи може бракувати інформації про конкретні аспекти, які цікавлять окремих пасажирів.

AirlineRatings – це онлайн-платформа, яка надає комплексні оцінки авіакомпаній з усього світу. Вона включає в себе різноманітні рейтинги, засновані на різних аспектах безпеки та обслуговування. Також оцінює авіакомпанії за кількома критеріями, включаючи безпеку, комфорт, обслуговування на борту та співвідношення ціни і якості. Вони також враховують відгуки пасажирів [5].

На рисунку 1.2 зображено вигляд головної сторінки.

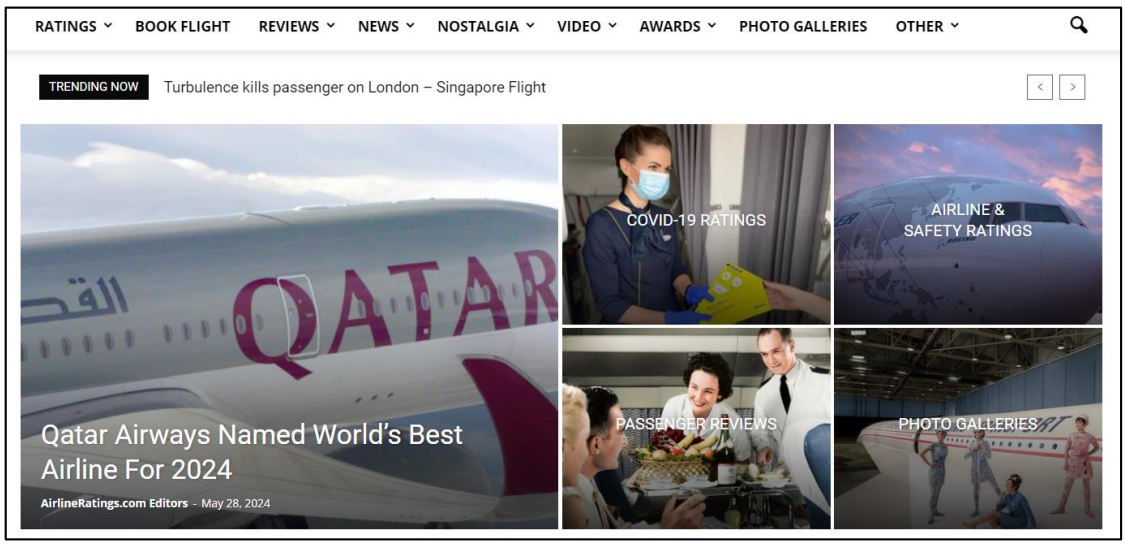

Рисунок 1.2 – Вигляд головної сторінки

Переваги:

– Оцінка безпеки: AirlineRatings надає детальні рейтинги безпеки, враховуючи історію аварій, дотримання міжнародних стандартів та аудити безпеки;

– Актуальність інформації: Рейтинги та відгуки регулярно оновлюються, забезпечуючи актуальність даних для користувачів.

Недоліки:

– Можлива упередженість: Існують зауваження, що деякі авіакомпанії можуть впливати на свої рейтинги через спонсорство або інші комерційні угоди;

– Обмежена глибина аналізу: Хоча платформа надає загальні оцінки, деталі можуть бути недостатніми для глибокого аналізу конкретних аспектів обслуговування.

Хоча кожна з цих платформ має свої переваги, такі як широке охоплення та актуальність інформації, вони також мають певні недоліки, включаючи можливу суб'єктивність та обмежену глибину аналізу. Використання цих інструментів сприяє підвищенню лояльності клієнтів та зміцненню конкурентоспроможності авіакомпаній на ринку.

1.4 Огляд готових рішень для передбачення

Оскільки програмне забезпечення буде створене на основі наявного датасету, важливо розглянути існуючі рішення, розроблені з його використанням. Після дослідження сторінки датасету на платформі Kaggle, було знайдено кілька готових проектів. Розглянемо одне з таких рішень під назвою «Airline Passenger Satisfaction» (рис. 1.3) [6].

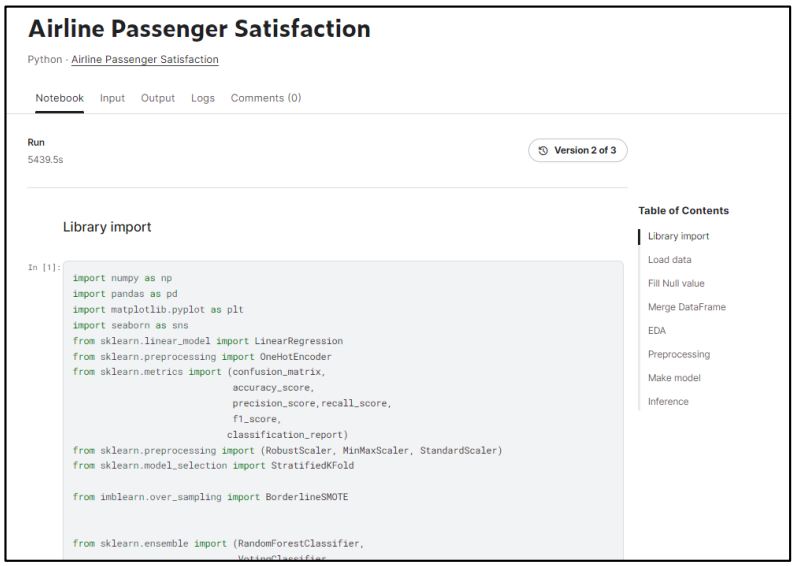

Рисунок 1.3 – Сторінка головного рішення «Airline Passenger Satisfaction»

Для впровадження інформаційної технології було ефективно застосовано кілька моделей машинного навчання, які показали високі результати на кожному етапі перевірки.

На рисунку 1.4 зображено фрагмент коду та результат побудови моделей машинного навчання [6].

```
print('Random Forest')
print('Accuracy:',accuracy_score(y_test, rf_y_pred))
print('Precision:', precision_score(y_test, rf_y_pred))
print('Recall:', recall_score(y_test, rf_y_pred))
print('F1 Score:', f1_score(y_test,rf_y_pred))
print('Macro F1 Score:', f1_score(y_test, rf_y_pred, average='macro'))

print('\nAdaBoost')
print('Accuracy:',accuracy_score(y_test, ab_y_pred))
print('Precision:', precision_score(y_test, ab_y_pred))
print('Recall:', recall_score(y_test, ab_y_pred))
print('F1 Score:', f1_score(y_test,ab_y_pred))
print('Macro F1 Score:', f1_score(y_test, ab_y_pred, average='macro'))

print('\nDecision Tree')
print('Accuracy:',accuracy_score(y_test, dt_y_pred))
print('Precision:', precision_score(y_test, dt_y_pred))
print('Recall:', recall_score(y_test, dt_y_pred))
print('F1 Score:', f1_score(y_test, dt_y_pred))
print('Macro F1 Score:', f1_score(y_test, dt_y_pred, average='macro'))
```

```
0-fold
Random Forest
Accuracy: 0.9605601270391223
Precision: 0.971258808898715
Recall: 0.9367004264392325
F1 Score: 0.9536666440539991
Macro F1 Score: 0.95966734233463

AdaBoost
Accuracy: 0.9283095134979067
Precision: 0.9232276812867473
Recall: 0.9102478678038379
F1 Score: 0.9166918302298273
Macro F1 Score: 0.9268876609656674
```

Рисунок 1.4 – Код та результат побудови моделей машинного навчання

Цей підхід має такі переваги: використання крос-валідації для стабільної оцінки моделей, порівняння кількох моделей (Random Forest, AdaBoost, Decision Tree), та детальний аналіз метрик (Accuracy, Precision, Recall, F1 Score, Macro F1 Score), що дозволяє обрати найефективнішу модель.

Недоліки: обмеження кількістю моделей, відсутність налаштування гіперпараметрів, можливе перенавчання, відсутність обробки незбалансованих даних, що може вплинути на точність моделей для рідкісних класів.

Однією з вже успішно впроваджених та ефективних інформаційних технологій є метод дослідження, відомий як «Will the Passenger be Satisfied?| ACC 94%» (рис. 1.5) [7]. Цей проект був визнаний через позитивні результати.

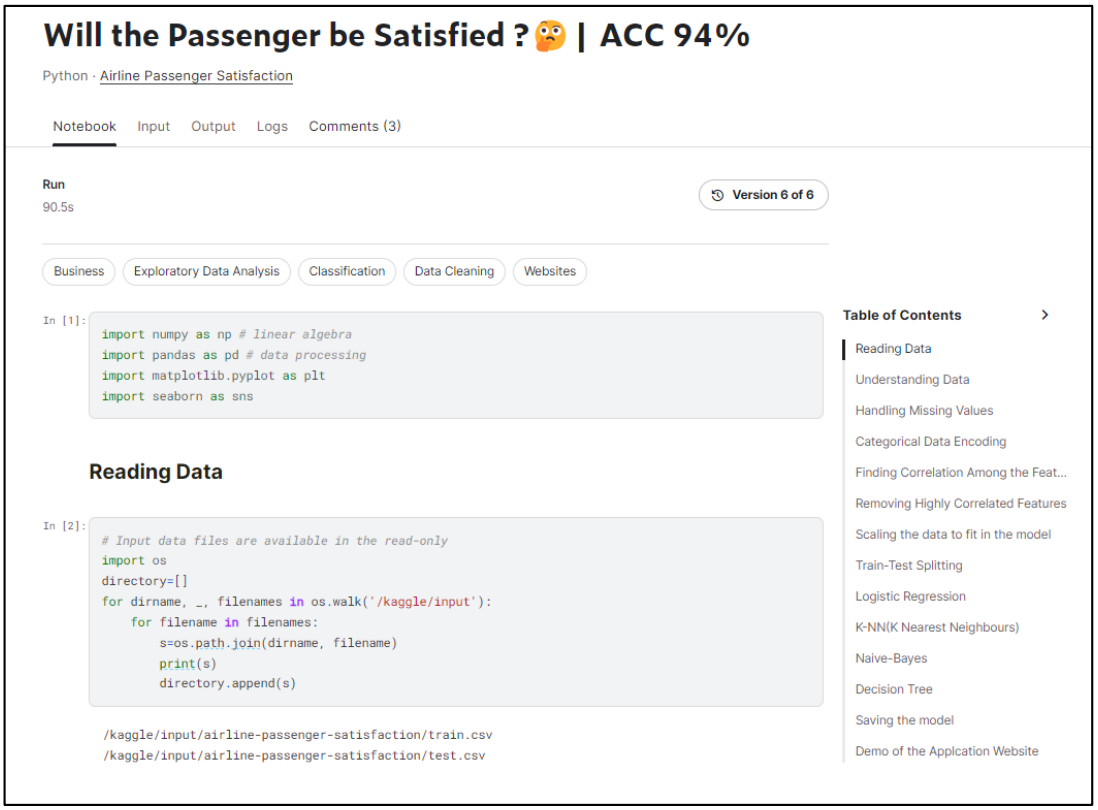

Рисунок 1.5 – Головна сторінка «Will the Passenger be Satisfied?| ACC 94%»

Для навчання та оцінки кількох моделей машинного навчання було використано Logistic Regression, KNN, Naive Bayes та Decision Tree. Результати передбачень оцінюються за допомогою метрик точності (accuracy) та classification report, який включає precision, recall, та f1-score для кожного класу.

Перевагами розробленої інформаційної технології можна вважати можливість порівняти декілька моделей машинного навчання для визначення найкращої та оцінку моделей за різними метриками дозволяє отримати повнішу картину ефективності. Але код не враховує оптимізацію гіперпараметрів для кращої ефективності моделей та не враховується можливість перенавчання моделей при використанні одного та того ж тестового набору даних для оцінки.

На рисунку 1.6 зображено фрагмент коду та результат побудови моделей.

```
y_pred_lr=model_lr.predict(X_test)
accuracy=accuracy_score(y_pred_lr,y_test)
print(f"Accuracy of Logistic Regression is : {round(accuracy*100,2)} % .")

report=classification_report(y_test,y_pred_lr)
print("Classification Report is : \n",report)
```

```
Accuracy of Logistic Regression is : 87.42 % .
Classification Report is :
              precision    recall  f1-score   support

           0       0.88      0.90      0.89     14622
           1       0.87      0.84      0.85     11354

    accuracy                           0.87     25976
   macro avg       0.87      0.87      0.87     25976
weighted avg       0.87      0.87      0.87     25976
```

Рисунок 1.6 – Фрагмент коду та результат побудови моделей

Розглянуті дві інформаційні технології виявили гарну точність у передбаченнях, що свідчить про їх потенціал та ефективність у вирішенні конкретних завдань. Проте, при їх застосуванні виникають певні недоліки, які можуть призвести до неточності або упередженості результатів. Давайте розглянемо ці недоліки детальніше [7].

Перш за все, однією з головних проблем є недостатня репрезентативність даних. Якщо дані, на яких навчається модель, не є достатньо різноманітними або не охоплюють всі можливі сценарії, це може призвести до того, що модель буде не в змозі коректно працювати з новими, раніше невідомими даними. Наприклад, якщо в навчальній вибірці відсутні певні групи населення або ж представлені в недостатній кількості, модель може демонструвати упередженість щодо цих груп, що в кінцевому результаті знижує точність передбачень.

Друге питання стосується методів попередньої обробки даних. Неправильне або недостатнє очищення даних може призвести до того, що модель навчиться на "шумі", тобто на випадкових або нерелевантних особливостях даних. Це може спотворити результати та знизити їх надійність. Важливо забезпечити правильне масштабування, нормалізацію, а також обробку

пропущених значень та аномалій, щоб мінімізувати ризики пов’язані з неякісною підготовкою даних.

Третій аспект стосується вибору гіперпараметрів моделей. Неправильний вибір гіперпараметрів може значно вплинути на продуктивність та точність моделі. Наприклад, занадто високий коефіцієнт регуляризації може призвести до недонавчання моделі, в той час як занадто низький може викликати перенавчання. Важливо проводити ретельний процес налаштування гіперпараметрів, використовуючи такі методи як крос-валідація, щоб знайти оптимальні значення, які забезпечать найкращу продуктивність моделі.

Крім того, слід зазначити, що всі ці фактори часто взаємопов’язані. Наприклад, недоліки в методах попередньої обробки даних можуть ускладнити процес налаштування гіперпараметрів, оскільки модель може сприймати нерелевантні особливості як значущі. Тому важливо підходити до процесу створення моделі комплексно, забезпечуючи високу якість даних на всіх етапах - від збору до налаштування гіперпараметрів.

Таким чином, хоча розглянуті інформаційні технології демонструють високу точність у передбаченнях, існує ряд факторів, які можуть негативно вплинути на їх результати. Усунення цих недоліків потребує уважного підходу до кожного етапу розробки та впровадження моделей, забезпечуючи тим самим більш надійні та точні передбачення.

1.5 Висновки

Перший розділ присвячений аналізу предметної області авіаційних послуг, зокрема, важливості якості обслуговування авіакомпаній для задоволеності пасажирів. У розділі підкреслюється зростання сфери послуг в економіці, роль стандартизації, вплив якості обслуговування на конкурентоспроможність та різницю у сприйнятті якості послуг різними групами пасажирів. Також розглядаються способи використання інформаційних технологій для покращення задоволеності пасажирів.

Після проведених досліджень аналогів було визначено, що більшість з них приділяють мало уваги попередній обробці даних, використанню кількох моделей передбачення, а також ігнорують налаштування гіперпараметрів, що буде враховано при розробленні інформаційної технології.

2 АНАЛІЗ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Вибір оптимального середовища розробки

Для вирішення поставленого завдання було вибрано платформу Kaggle, оцінюючи її ефективність у сфері аналізу даних та машинного навчання. Kaggle відома своїм великим співтовариством фахівців, доступом до потужних обчислювальних ресурсів та інтегрованими інструментами для обробки даних.

Також, на платформі проводяться змагання, що сприяє обміну ідеями та розвитку навичок у галузі аналізу даних. Впевненість у стабільності та надійності Kaggle робить її відмінним вибором для цього проекту. Kaggle - це онлайн-платформа для проведення змагань з машинного навчання та аналізу даних, яка збирає дані, завдання та спільноту фахівців з усього світу. Учасники можуть брати участь у різних випробуваннях, вирішувати завдання та долучатися до спільноти для обміну досвідом. Kaggle також надає інструменти для аналізу даних, візуалізації та роботи з моделями машинного навчання [8].

На рисунку 2.1 зображено головну сторінку Kaggle.

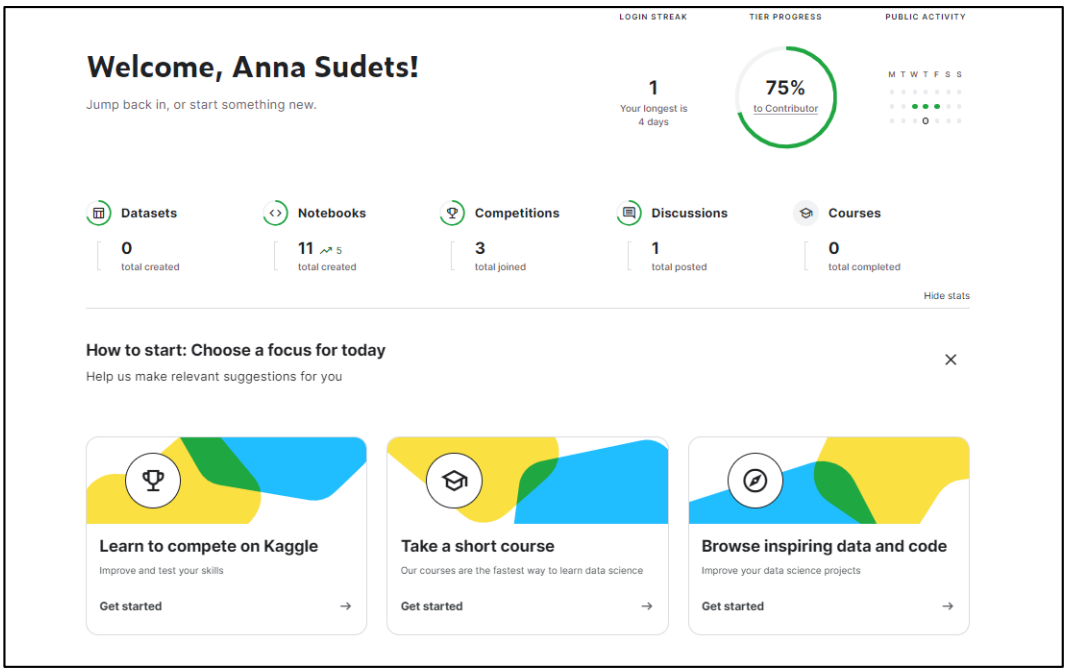

Рисунок 2.1 – Головна сторінка Kaggle

На Kaggle користувачі можуть завантажувати власні дані або використовувати наявні набори даних для вирішення конкретних завдань. Платформа підтримує різні мови програмування, що дає можливість учасникам використовувати їх улюблені інструменти та бібліотеки для створення моделей.

Змагання на платформі Kaggle є однією з головних характеристик, яка привертає учасників з усього світу. Ці конкурси часто спонсоруються компаніями або організаціями, які надають учасникам конкретні завдання у сфері машинного навчання, аналізу даних та інших суміжних галузях (рис. 2.2).

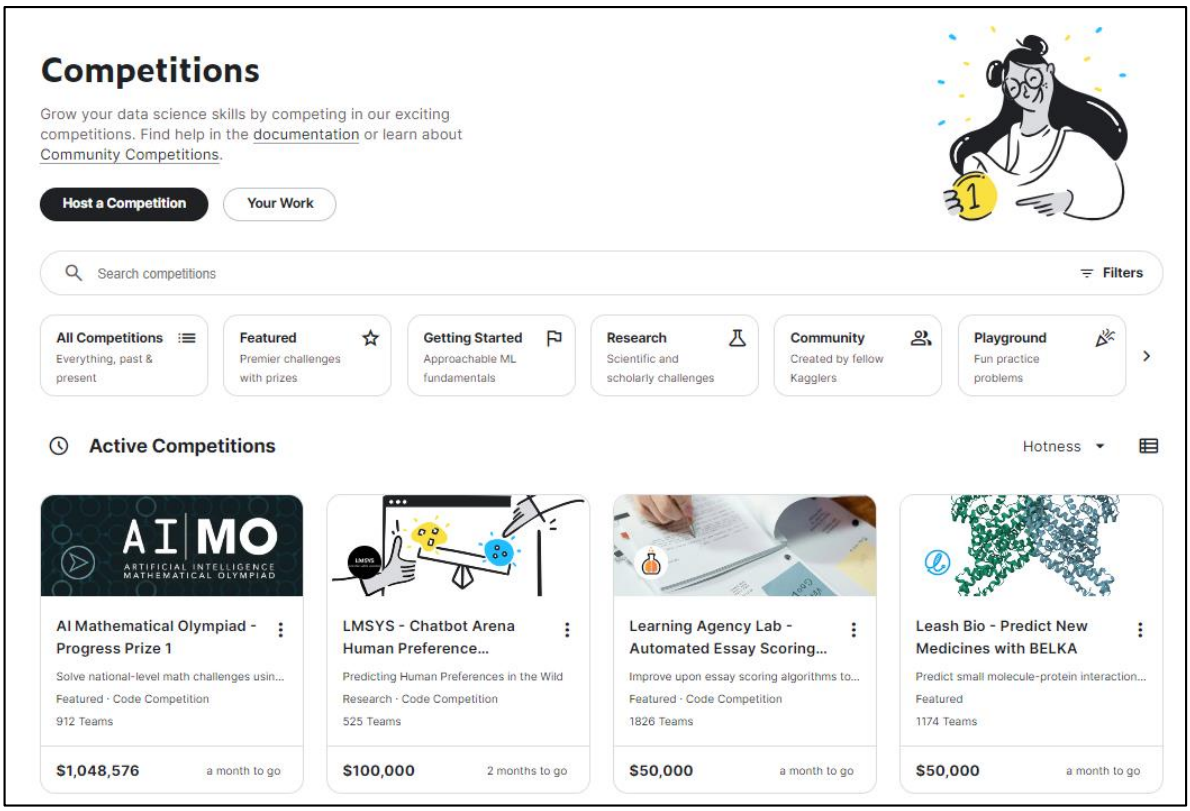

Рисунок 2.2 – Сторінка зі змаганнями в Kaggle

Python був обраний як мова програмування, оскільки вона має найбільший вибір інструментів та бібліотек для створення та використання моделей машинного навчання.

Python є високорівневою, інтерпретованою мовою загального призначення, що була розроблена Guido van Rossum і вперше випущена у 1991 році. Вона характеризується простим синтаксисом, що полегшує вивчення

новачкам, і підтримує об'єктно-орієнтоване, імперативне та функціональне програмування.

Основні особливості Python включають динамічне визначення типів, автоматичне управління пам'яттю та широкий набір функцій у стандартній бібліотеці. Мова активно використовується в різних галузях програмування, таких як веб-розробка, аналіз даних, штучний інтелект, машинне навчання та багато інших.

Розгалужена стандартна бібліотека Python містить широкий асортимент готових модулів для вирішення різноманітних завдань, що спрощує процес розробки та дозволяє створювати функціональні програми швидше. Довіра до Python у заслуговує його успіху, оскільки мову активно використовують у різних галузях, починаючи від науки й закінчуючи великими корпоративними системами.

Здатність використовувати Python для розробки прототипів і проведення експериментів допомагає скоротити час, необхідний для створення та вдосконалення моделей машинного навчання. Python також легко інтегрується з іншими популярними мовами програмування та інструментами, що робить його вибором для великих та складних проектів у сфері штучного інтелекту та аналізу даних [9].

Оскільки це дослідження включає в себе завдання класифікації, розумним вибором буде використання таких моделей машинного навчання як Naive Bayes, Decision Tree, Random Forest, Extra Trees, KNN, Logistic Regression, AdaBoost, GradientBoost та LGBM для досягнення цілей дослідження.

2.2 Модель машинного навчання «Decision Tree»

Модель Дерево рішень (Decision Tree) використовується для прийняття рішень на основі умовних правил. Це структура, подібна до дерева, де кожен вузол є тестом, що характеризує певну властивість вхідних даних, а кожне розгалуження вказує на можливий результат цього тесту.

Основна концепція дерева рішень полягає в рекурсивному поділі набору даних на підмножини з урахуванням атрибутів, що мають найбільший вплив на прогнозовану змінну. Процес навчання дерева включає вибір оптимальних атрибутів для поділу та побудову дерева досягненням критерію зупинки, такого як максимальна глибина дерева або мінімальна кількість вибірок у вузлі [10].

На рисунку 2.3 показано принцип роботи алгоритму.

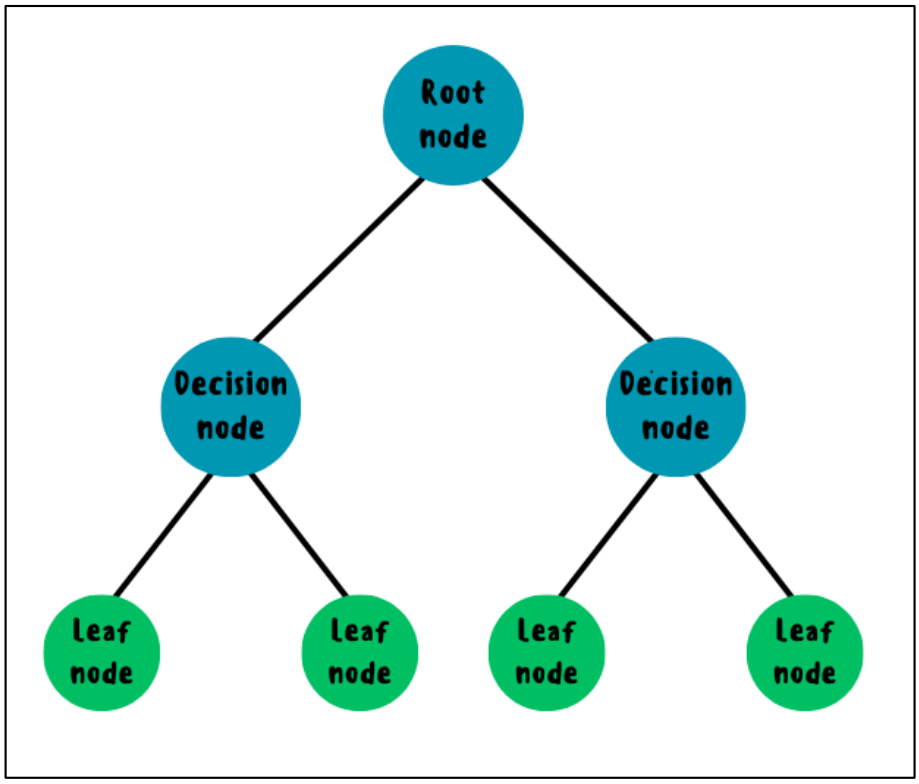

Рисунок 2.3 – Приклад роботи алгоритму Decision Tree

Важливо відзначити, що модель Decision Tree допомагає уникнути проблеми перенавчання завдяки вбудованому механізму обмеження глибини дерева та інших параметрів. Це дозволяє підтримувати стабільність моделі та уникнути втрати узагальнювальної здатності на нових даних.

Крім того, модель Дерево рішень є дуже гнучким алгоритмом, оскільки може легко адаптуватися до змін у вхідних даних та умовах завдань. Ця гнучкість робить його ефективним в різних сферах, від медицини до фінансів, де важливо миттєво реагувати на зміни у даних та отримувати точні прогнози.

Використання моделі Decision Tree наведено на рисунку 2.4.

```
# Навчання моделі Decision Tree
decision_tree = DecisionTreeClassifier(criterion='entropy', min_samples_split=6, max_depth=6, rand
om_state=0)
decision_tree.fit(X_train_imputed, y_train)
previsoes = decision_tree.predict(X_test_imputed)

# Передбачення на тестовому наборі
test_predictions = decision_tree.predict(X_test_imputed)

# Передбачення на тренувальному наборі
train_predictions = decision_tree.predict(X_train_imputed)
```

Рисунок 2.4 – Використання моделі Decision Tree

2.3 Модель машинного навчання «Random Forest»

Random Forest – це модель машинного навчання, яка використовується для прийняття рішень на основі умовних правил. Це ансамбль дерев рішень, де кожне дерево є випадковим підмножиною вхідних даних, а прийняття рішення здійснюється шляхом голосування або середнього значення виходів кожного дерева.

Основна ідея Random Forest полягає у тому, що кожне дерево будується на основі випадкового підмножини набору даних, і воно вирішує задачу класифікації або регресії. Під час тренування кожне дерево в Random Forest використовується для прийняття рішення на основі підмножини атрибутів, що мають найбільший вплив на прогнозовану змінну.

Random Forest вирішує проблему перенавчання завдяки випадковості вибору даних для кожного дерева та ансамблюванню результатів декількох дерев. Це дозволяє підтримувати стабільність моделі і уникнути втрати узагальнювальної здатності на нових даних [11].

На рисунку 2.5 зображено принцип роботи алгоритму.

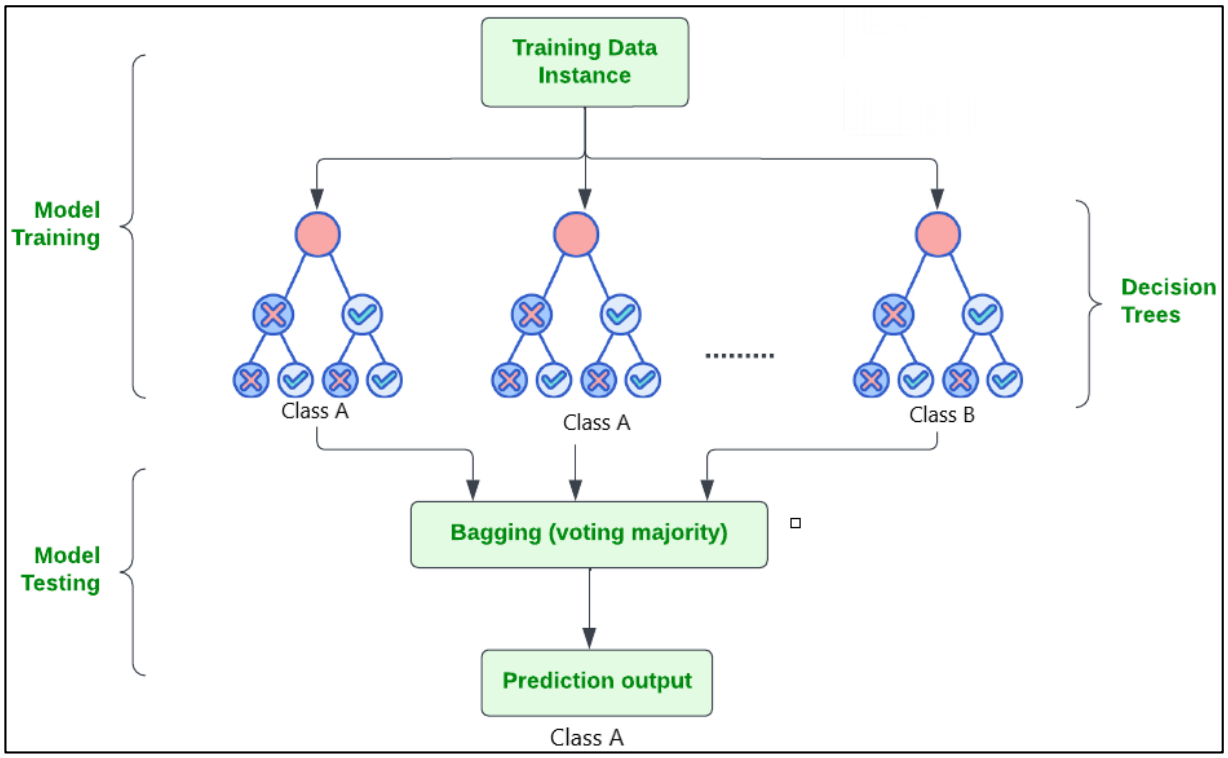

Рисунок 2.5 – Алгоритм Random Forest

Використання моделі Random Forest наведено на рисунку 2.6.

```
# Навчання моделі Random Forest
random_forest = RandomForestClassifier(n_estimators=100, min_samples_split=5, max_depth=5, criteri
on='gini', random_state=0)
random_forest.fit(X_train_imputed, y_train)
previsoes = random_forest.predict(X_test_imputed)

# Передбачення на тестовому наборі
test_predictions = random_forest.predict(X_test_imputed)

# Передбачення на тренувальному наборі
train_predictions = random_forest.predict(X_train_imputed)
```

Рисунок 2.6 – Використання моделі Random Forest

2.4 Модель машинного навчання «GradientBoost»

Gradient Boosting Classifier (GBC) – це метод машинного навчання, що входить до категорії ансамблевих алгоритмів. Основна концепція полягає у послідовному створенні слабких класифікаторів, зазвичай дерев рішень, та поетапному коригуванні їх прогнозів для підвищення точності моделі.

GradientBoost використовує градієнтні методи оптимізації для мінімізації функції втрат.

Під час навчання GBC будує новий класифікатор, який коригує помилки, допущені попередніми класифікаторами у послідовності. Цей новий класифікатор додається до моделі з вагами, що залежать від того, наскільки ефективно він виправляє помилки. Важливою характеристикою є використання параметра кроку (learning rate), який регулює внесок кожного нового класифікатора.

GradientBoost широко використовується для розв'язання завдань класифікації та регресії і відомий своєю високою точністю. Проте слід зауважити, що цей метод може схильний до перенавчання, і оптимальна настройка гіперпараметрів грає ключову роль у досягненні найкращих результатів продуктивності моделі.

GradientBoost допомагає створити потужну прогнозну модель шляхом поєднання багатьох слабких моделей. На кожному кроці модель фокусується на тих прикладах, які попередні моделі класифікували недостатньо точно, та намагається покращити їх прогнози. Цей процес повторюється ітеративно, що призводить до створення потужної моделі з високою точністю.

Однією з ключових переваг градієнтного бустінгу є його здатність автоматично виявляти важливі функції для класифікації або регресії, що робить його ефективним для роботи з складними даними. Крім того, він добре пристосовується до великої кількості даних та різноманітних завдань машинного навчання.

Важливо пам'ятати, що градієнтний бустінг вимагає значних обчислювальних ресурсів і тривалого процесу навчання, але при належній настройці параметрів він може забезпечити вражаючі результати в різних галузях застосування, таких як фінанси, медицина або аналіз даних [12].

Використання моделі GradientBoost наведено на рисунку 2.7.

```
# Train the GradientBoostingClassifier
grad_boost = GradientBoostingClassifier(n_estimators=300, learning_rate=0.01, random_state=0)
grad_boost.fit(X_train_imputed, y_train)

# Predict on the test set
previsoes = grad_boost.predict(X_test_imputed)

# Calculate accuracy
train_accuracy8 = grad_boost.score(X_train_imputed, y_train)
test_accuracy8 = grad_boost.score(X_test_imputed, y_test)
print("Train Accuracy:", train_accuracy8)
print("Test Accuracy:", test_accuracy8)
```

Рисунок 2.7 – Використання моделі Random Forest

2.5 Модель машинного навчання «KNN»

Метод K-найближчих сусідів (KNN) є одним з простих і ефективних алгоритмів машинного навчання, особливо в задачах класифікації та регресії. Основна ідея полягає в тому, щоб передбачати клас чи значення для нового прикладу, основуючись на його найближчих сусідах у просторі ознак.

Однією з переваг KNN є його простота та інтуїтивність. Алгоритм не потребує складних математичних обчислень або великої кількості параметрів для налаштування, що робить його відмінним вибором для початківців у машинному навчанні.

Крім того, KNN може працювати добре з нелінійними даними та не чутливий до форми розподілу даних. Він також може використовуватися для класифікації або регресії в задачах з різними типами даних, такими як текстові, зображення або числові.

Іншою важливою перевагою KNN є його можливість пристосовуватися до змін у наборі даних без необхідності повторної навчання моделі. Це робить його корисним в ситуаціях, де дані регулярно оновлюються або додаються нові приклади.

Однак, важливо враховувати, що KNN може бути обчислювально витратним для великих наборів даних, особливо якщо кількість признаків велика. Також він чутливий до вибору параметра k - кількості найближчих

сусідів, і неправильне вибране значення може призвести до погіршення точності моделі.

Загалом, KNN є потужним інструментом, який може бути ефективним в різних ситуаціях, забезпечуючи хорошу точність і простоту використання [13].

На рисунку 2.8 зображено роботу алгоритму.

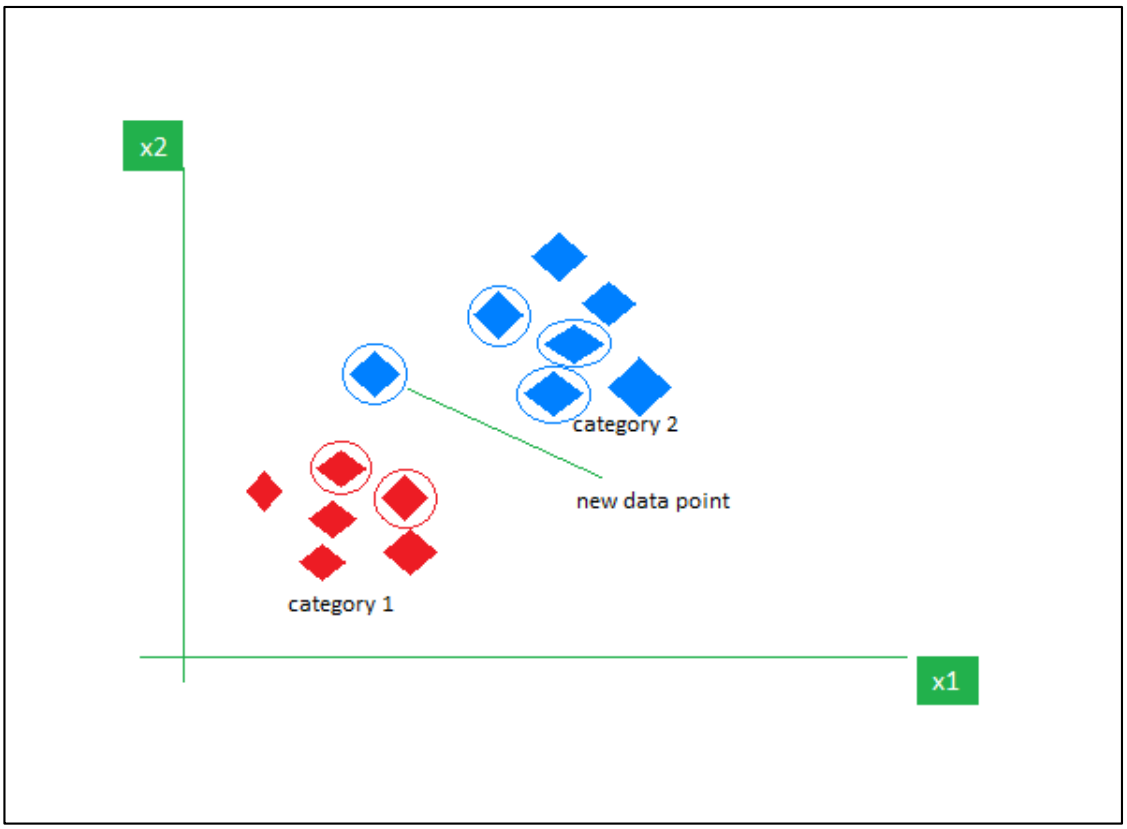

Рисунок 2.8 – Візуалізація роботи алгоритму

Використання моделі KNN наведено на рисунку 2.9.

```
# Навчання моделі KNN
knn = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2)
knn.fit(X_train_imputed, y_train)
previsoes = knn.predict(X_test_imputed)

# Передбачення на тестовому наборі
test_predictions = knn.predict(X_test_imputed)

# Передбачення на тренувальному наборі
train_predictions = knn.predict(X_train_imputed)

# Обчислення точності для тренувального та тестового наборів
train_accuracy5 = accuracy_score(y_train, train_predictions)
test_accuracy5 = accuracy_score(y_test, test_predictions)
```

Рисунок 2.9 – Використання моделі KNN

2.6 Висновки

Для вирішення поставленого завдання була обрана платформа Kaggle, відома своїм великим співтовариством фахівців, доступом до потужних обчислювальних ресурсів та інтегрованими інструментами для обробки даних.

Платформа також надає можливість брати участь у змаганнях, що сприяє обміну ідеями та розвитку навичок у галузі аналізу даних. Використання Kaggle було визнано доцільним завдяки її стабільності та надійності.

Python був обраний як основна мова програмування через широкий вибір інструментів та бібліотек для створення моделей машинного навчання. Його простий синтаксис і багатофункціональність дозволяють ефективно розробляти прототипи та проводити експерименти.

Для виконання завдань класифікації були обрані різні моделі машинного навчання: Naive Bayes, Decision Tree, Random Forest, Extra Trees, KNN, Logistic Regression, AdaBoost, GradientBoost та LGBM.

Обрано кілька моделей для експериментального порівняння їх ефективності в рішенні задачі класифікації на основі їх унікальних характеристик і здатностей..

3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПЕРЕДБАЧЕННЯ РІВНІВ ЗАДОВОЛЕНОСТІ ПАСАЖИРІВ АВІАКОМПАНІЯМИ

3.1 Аналіз обраних бібліотек та побудова схеми алгоритму

Першим кроком до виконання завдання було розроблено ряд дій, які необхідно виконати для успішного виконання завдання.

На рисунку 3.1 показана схема алгоритму, використаного в подальшому для передбачення рівнів задоволеності пасажирів авіакомпаніями.

Наведена схема алгоритму містить наступні кроки:

– Імпорт бібліотек;
– Імпорт набору даних;
– Розвідувальний аналіз;
– Обробка даних;
– Побудова моделей;
– Оцінка та порівняння моделей

У разі незадовільної точності результатів необхідно провести додаткову обробку даних, змінити параметри та повернутися до етапу побудови моделей.

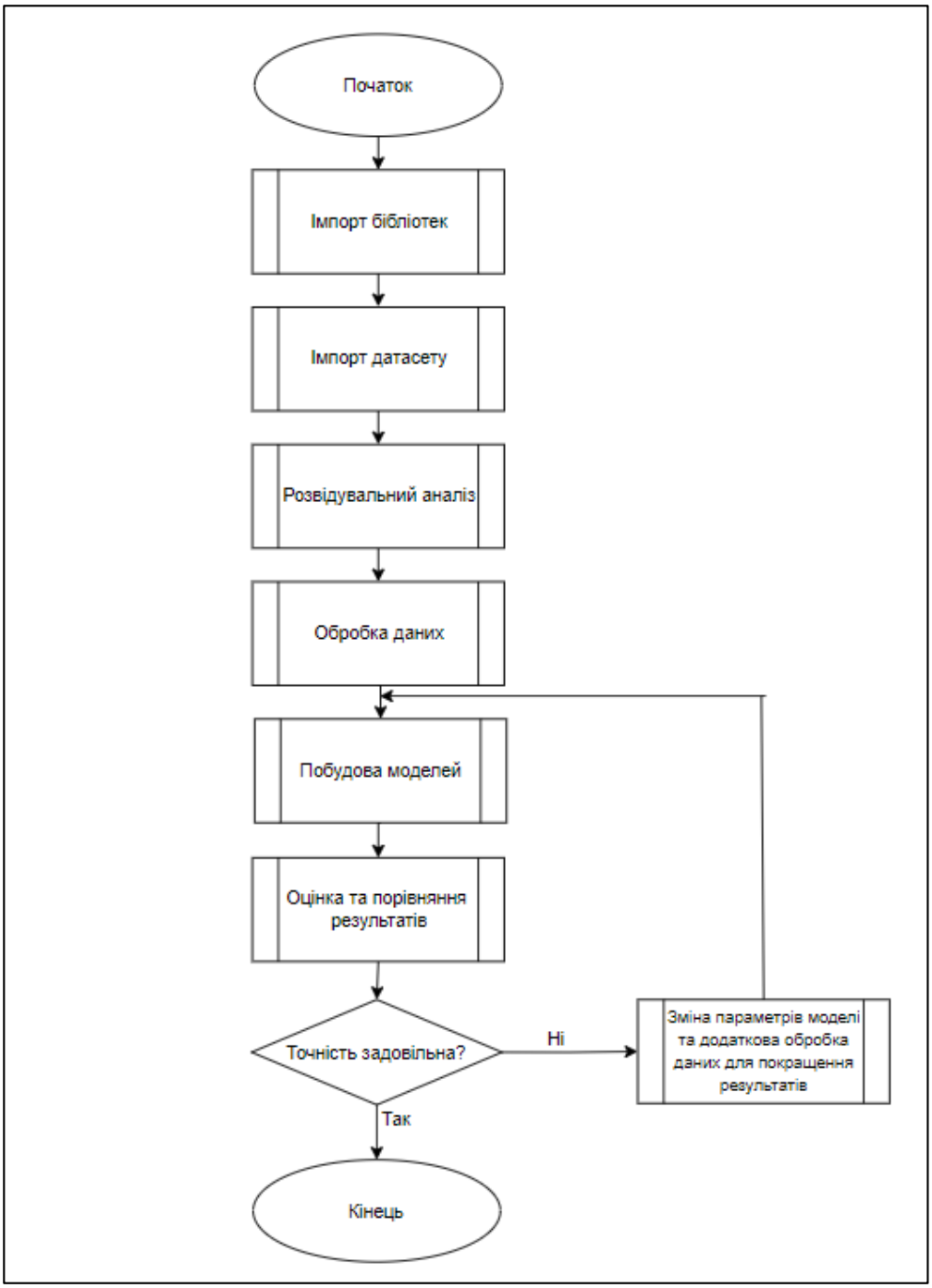

Рисунок 3.1 – Схема алгоритму

Для початку виконання поставленої задачі було створено новий ноутбук в Kaggle під назвою «Airline_BD», та використано датасет «Airline Passenger Satisfaction». Наступним кроком було обрано та імпортовано певний перелік бібліотек, які будуть використовуватися для проведення аналізу даних, це допоможе ефективно виконати усі поставлені задачі.

На рисунку 3.2 зображено перелік використаних бібліотек, які забезпечують широкі можливості для вивчення та обробки даних.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import seaborn as sns
import plotly.express as px
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from yellowbrick.classifier import ConfusionMatrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
from sklearn.preprocessing import LabelEncoder
from sklearn.impute import SimpleImputer
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.impute import SimpleImputer
from sklearn.model_selection import RandomizedSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import train_test_split, cross_val_score, learning_curve
import joblib


import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

Рисунок 3.2 – Список використаних бібліотек

Модуль plotly.express (зазвичай імпортований як px) містить функції, які можуть створювати цілі фігури одночасно, і називається Plotly Express або PX.

Plotly Express є вбудованою частиною бібліотеки plotly і є рекомендованою відправною точкою для створення найбільш поширених фігур. Кожна функція.

Plotly Express використовує внутрішньо об’єкти графіка та повертає екземпляр plotly.graph_objects.Figure [14].

Pandas — це бібліотека Python, яка використовується для роботи з наборами даних.

Вона має функції для аналізу, очищення, дослідження та маніпулювання даними.

Назва «Pandas» містить посилання як на «Панельні дані», так і на «Аналіз даних Python» і була створена Весом МакКінні в 2008 році. Pandas дозволяє нам аналізувати великі дані та робити висновки на основі статистичних теорій.

Pandas може очистити брудні набори даних і зробити їх читабельними та актуальними [15].

NumPy (Numerical Python) — це бібліотека для мови програмування Python, яка підтримує багатовимірні масиви та матриці, а також пропонує безліч математичних функцій для роботи з ними.

NumPy є фундаментальною бібліотекою для наукових обчислень з Python, часто використовується як основа для створення інших бібліотек, таких як SciPy, Pandas, Matplotlib і багато інших. NumPy швидким і гнучким інструментом для роботи з багатовимірними масивами. Масиви можуть мати довільну кількість вимірів (осей) і дозволяють ефективно маніпулювати великими наборами даних.

Бібліотека включає широкий спектр математичних функцій для операцій з масивами, таких як арифметичні операції, функції для роботи з лінійною алгеброю, статистичні функції, генератори випадкових чисел та багато інших [16].

3.2 Підготовка даних та розвідувальний аналіз

Першим кроком у підготовці даних було імпортовано датасет під назвою «Airline Passenger Satisfaction». Після цього було здійснено зчитування даних з файлів "train.csv" та "test.csv", які знаходяться у вказаному датасеті. Далі, виведено повідомлення про успішне завершення операції зчитування.

Після завершення зчитування даних, виводиться результат про успішне завершення операції зчитування для кожного з файлів: "train.csv" та "test.csv".

Також було виведено перших п'ять записів з набору даних "train.csv", що дозволяє ознайомитися з їхнім вмістом та структурою. Це допомагає отримати загальне уявлення про дані та їхню організацію перед подальшим аналізом.

На рисунку 3.3 зображено статистичний опис кількісних (числових) стовпців для тренувального набору даних.

	Unnamed: 0	id	Age	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking
count	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.0000
mean	51951.500000	64924.210502	39.379706	1189.448375	2.729683	3.060296	2.756901
std	29994.645522	37463.812252	15.114964	997.147281	1.327829	1.525075	1.398929
min	0.000000	1.000000	7.000000	31.000000	0.000000	0.000000	0.000000
25%	25975.750000	32533.750000	27.000000	414.000000	2.000000	2.000000	2.000000
50%	51951.500000	64856.500000	40.000000	843.000000	3.000000	3.000000	3.000000
75%	77927.250000	97368.250000	51.000000	1743.000000	4.000000	4.000000	4.000000
max	103903.000000	129880.000000	85.000000	4983.000000	5.000000	5.000000	5.000000

Рисунок 3.3 – Статистичний огляд даних набору train

Результат виводу цього методу містить основні статистичні характеристики кожного числового стовпця, такі як кількість, середнє значення, стандартне відхилення, мінімальне та максимальне значення, а також квартилі (25%, 50%, 75%).

Наприклад, для стовпця "Age" ми бачимо, що середнє значення віку пасажирів складає близько 39 років, стандартне відхилення - приблизно 15 років, мінімальний вік – 7 років, максимальний вік – 85 років, тощо. Такий опис дозволяє зрозуміти розподіл та характеристики числових змінних у датасеті.

Розвідувальний аналіз – це процес аналізу наборів даних для узагальнення їхніх основних характеристик, часто з використанням статистичної графіки та інших методів візуалізації даних. Основна мета EDA полягає в тому, щоб зрозуміти дані та їх базову структуру, розкрити закономірності, виявити аномалії та сформулювати гіпотези, які можуть бути досліджені далі. EDA є важливим початковим кроком у будь-якому проекті аналізу даних, оскільки він надає розуміння даних, які можуть керувати подальшими процесами моделювання та прийняття рішень [17].

Першим кроком було побудовано кореляційну матрицю для дослідження зв’язків між числовими та закодованими категоріальними ознаками (рис 3.4).

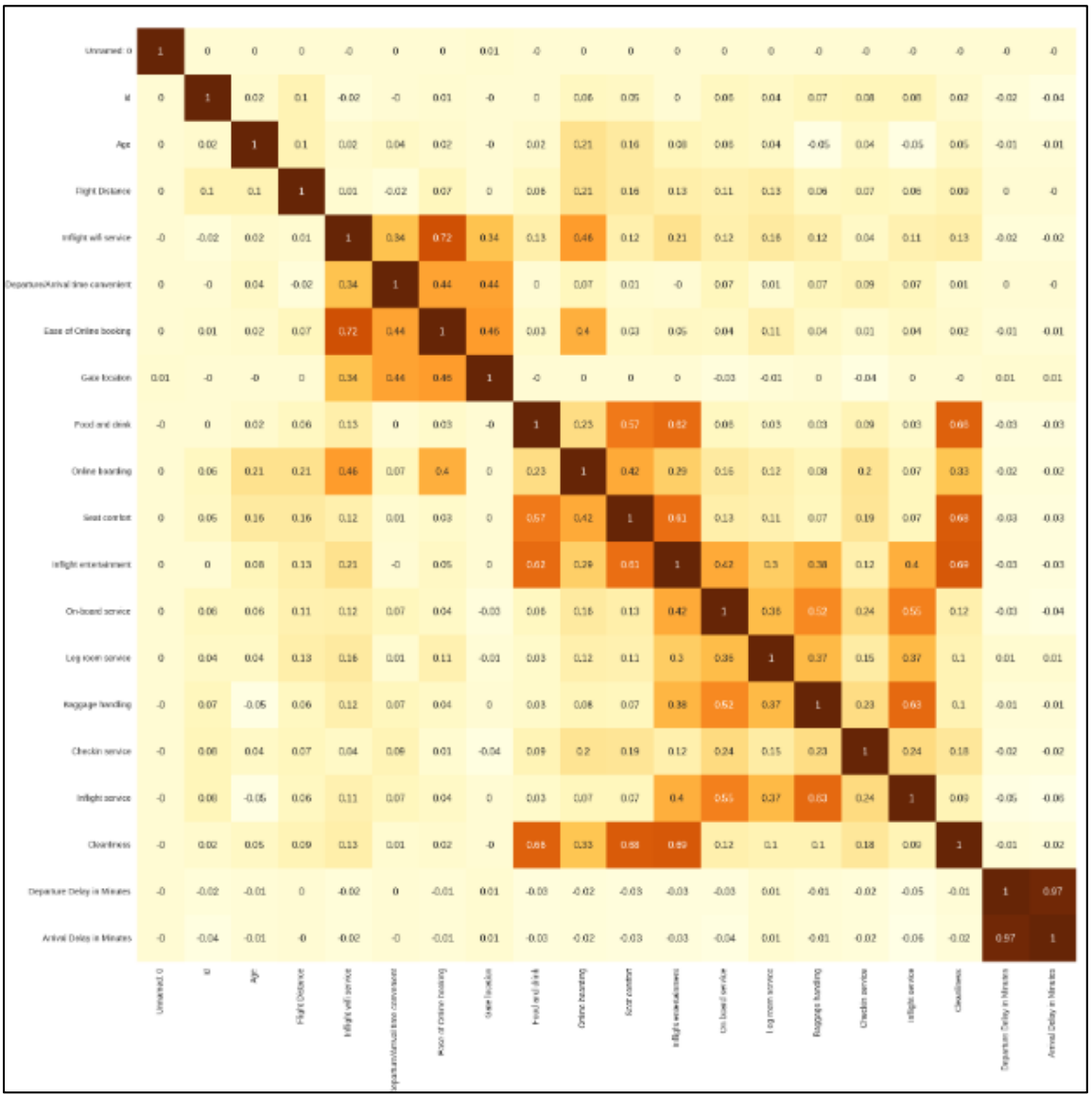

Рисунок 3.4 – Кореляційна матриця

Існує багато сильних кореляцій між змінними. "Затримка відправлення в хвилинах" сильно корелює із "Затримкою прибуття в хвилинах". Ця висока кореляція свідчить про те, що затримки у відправленні, як правило, відображаються у відповідних затримках після прибуття. "Ease_of_Online_booking" дуже корелює з "Inflight_wifi_service".

Це може вказувати на те, що пасажири, які знаходять зручним бронювання онлайн, часто оцінюють також високо і якість Wi-Fi під час польоту.

Сильні кореляції між змінними підтверджують взаємозалежність різних аспектів задоволеності пасажирів авіакомпаній, що дозволяє робити більш точні прогнози на основі аналізу взаємозв'язків цих рівнів.

Також було б корисно побачити, відображення трьох стовпчастих графіків, що показують розподіл пасажирів авіакомпанії за гендером, типом клієнтів і віком, що дозволяє візуально оцінити розподіл цих змінних у наборі даних (рис 3.5).

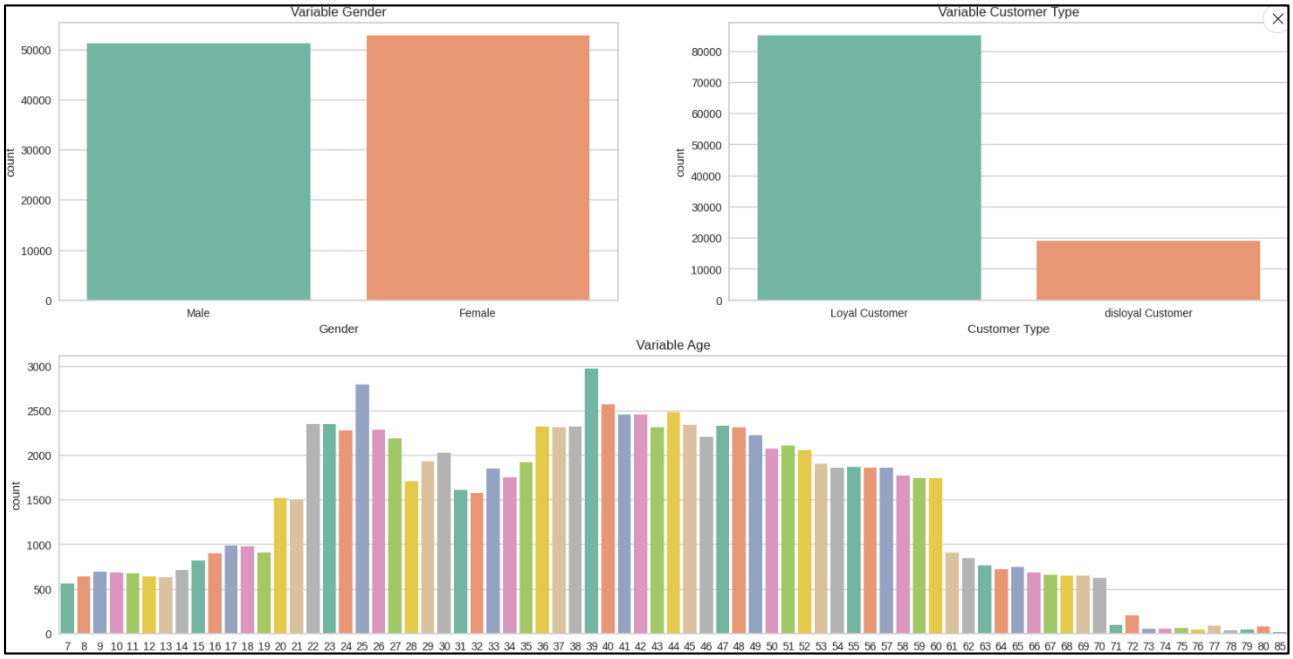

Рисунок 3.5 – Розподіл пасажирів авіакомпанії

Проведено розподіли за типом подорожі (Type of Travel) – стовпчастий графік показує, скільки пасажирів подорожує з різними цілями наприклад, бізнес або відпочинок, за класом (Class) – стовпчастий графік показує, скільки пасажирів подорожує у різних класах (наприклад, економ, бізнес), за відстанню польоту (Flight Distance) – стовпчастий графік показує розподіл відстаней польотів, дозволяючи оцінити, які відстані є найбільш поширеними.

Отримані числові значення для Flight Distance надають додаткову інформацію про варіативність та центральну тенденцію цієї змінної (рис 3.6).

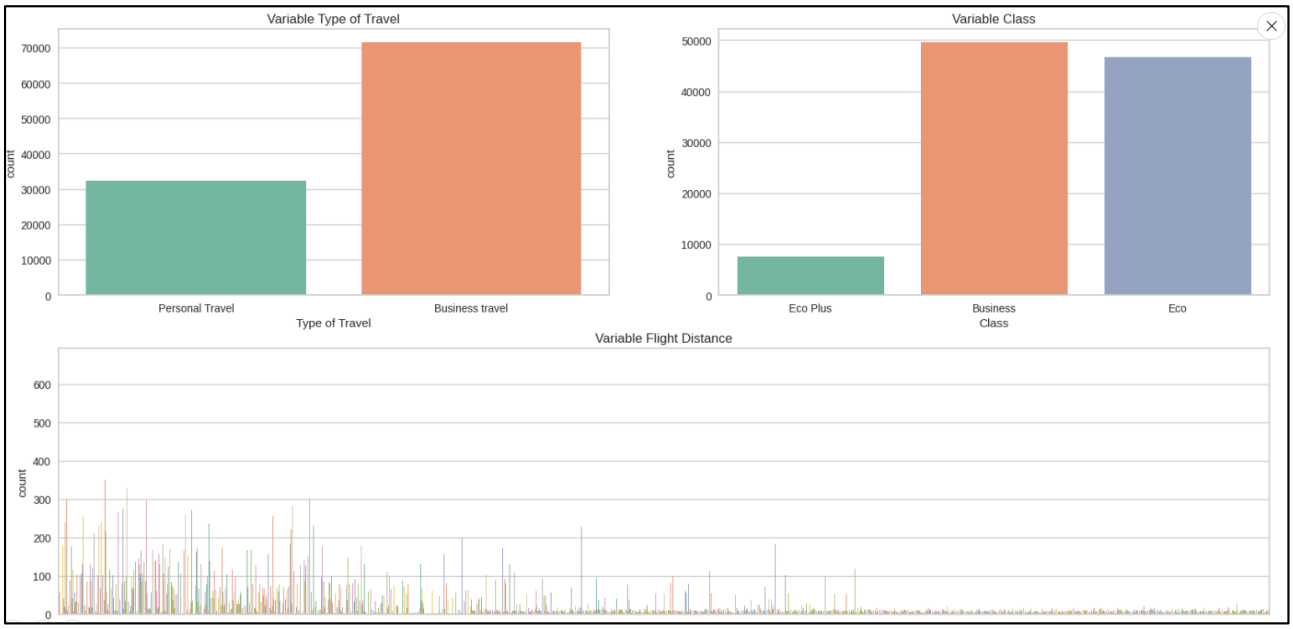

Рисунок 3.6 – Розподіл за типом, класом та відстанню

Аналіз отриманих результатів: Мінімальна відстань польоту у вибірці становить 31 милю. Це вказує на те, що у вибірці є дуже короткі рейси. Середня відстань польоту становить приблизно 1189.45 миль. Це дає уявлення про середню довжину рейсів у наборі даних. Максимальна відстань польоту у вибірці становить 4983 милі. Це вказує на те, що у вибірці є і довгі міжконтинентальні рейси.

Побудовано графіки, що дозволяють швидко оцінити розподіл затримок і визначити їх частоту за різних тривалостей. Наприклад, можна встановити, що більшість затримок складають менше 15-20 хвилин, але також можна побачити, чи є значні затримки, наприклад, більше 30 хвилин, які можуть вплинути на подорож пасажирів. Такий аналіз допомагає авіакомпаніям вдосконалювати процеси управління та покращувати рівень обслуговування, щоб зменшити затримки і підвищити задоволеність пасажирів (рис 3.7).

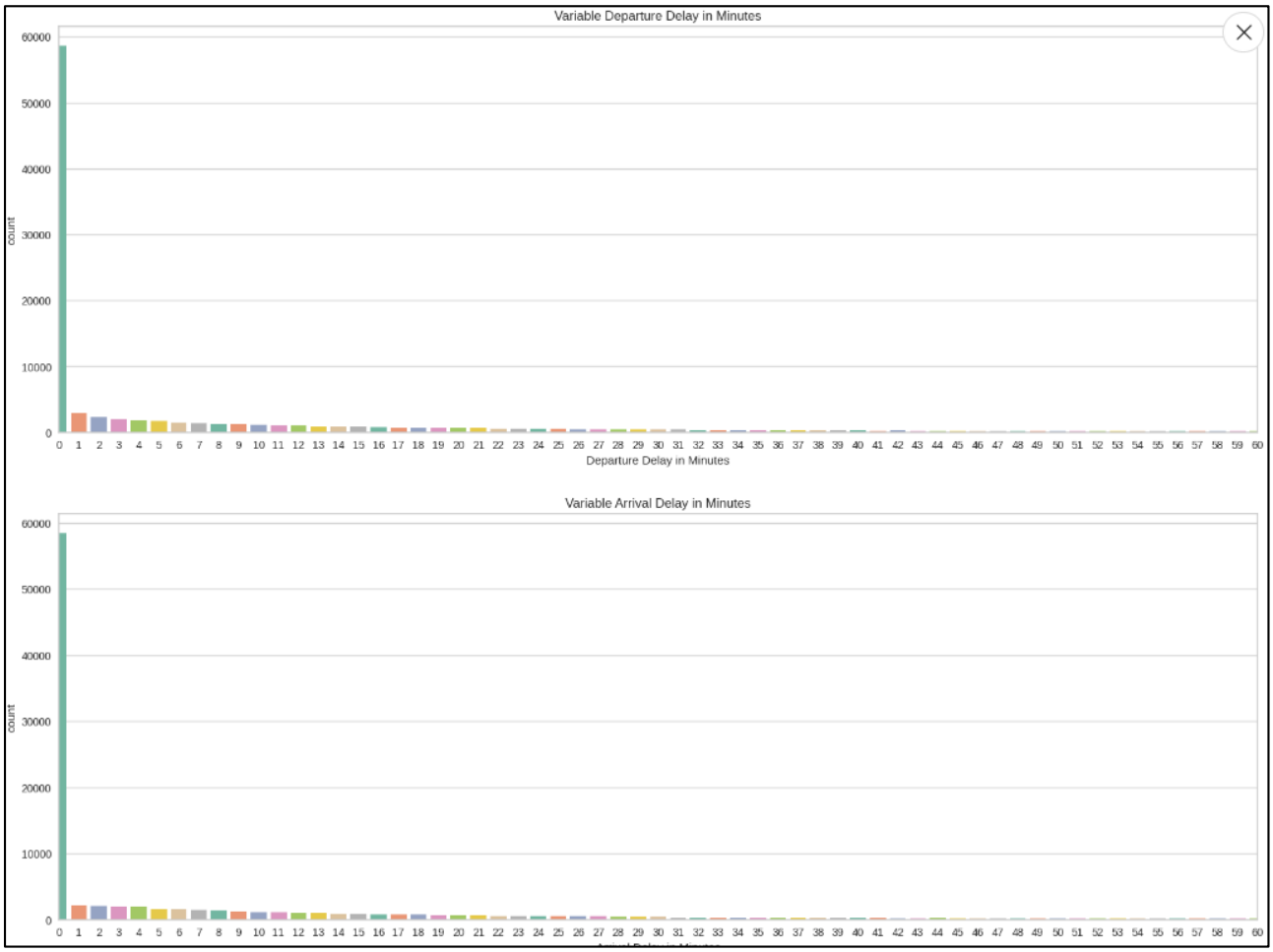

Рисунок 3.7 – Розподіл затримок

Цей аналіз затримок вильоту та прибуття важливий для декількох аспектів управління авіакомпанією та покращення обслуговування пасажирів:

– Оцінка ефективності: Розподіл затримок допомагає авіакомпаніям оцінити ефективність своїх процесів управління та виявити можливі проблемні ситуації. Наприклад, висока кількість затримок може вказувати на потребу в оптимізації графіків польотів або вдосконаленні процесів реєстрації пасажирів;

– Планування ресурсів: Аналіз затримок допомагає в плануванні використання ресурсів авіакомпанії, таких як обслуговування пасажирів, персонал та пальне. Це дозволяє керівництву авіакомпанії раціонально розподіляти ресурси та запобігати серйозним простоям;

– Покращення задоволеності клієнтів: Шляхом аналізу затримок і вжиття заходів для їх зменшення авіакомпанії можуть покращити задоволеність клієнтів. Мінімізація затримок сприяє поліпшенню досвіду польоту пасажирів,

що в свою чергу може позитивно позначитися на репутації авіакомпанії та забезпечити її конкурентоспроможність на ринку.

– Безпека і оперативність: Аналіз затримок допомагає в ідентифікації можливих проблем в системі безпеки та оперативності польотів. Відстеження частоти та тривалості затримок може допомогти у виявленні тенденцій та удосконаленні процесів для підвищення безпеки та оперативності.

Проаналізовано взаємозв'язок між рівнями задоволення та змінними "стать" та "вік", використовуючи стовпчасті діаграми (рис 3.8).

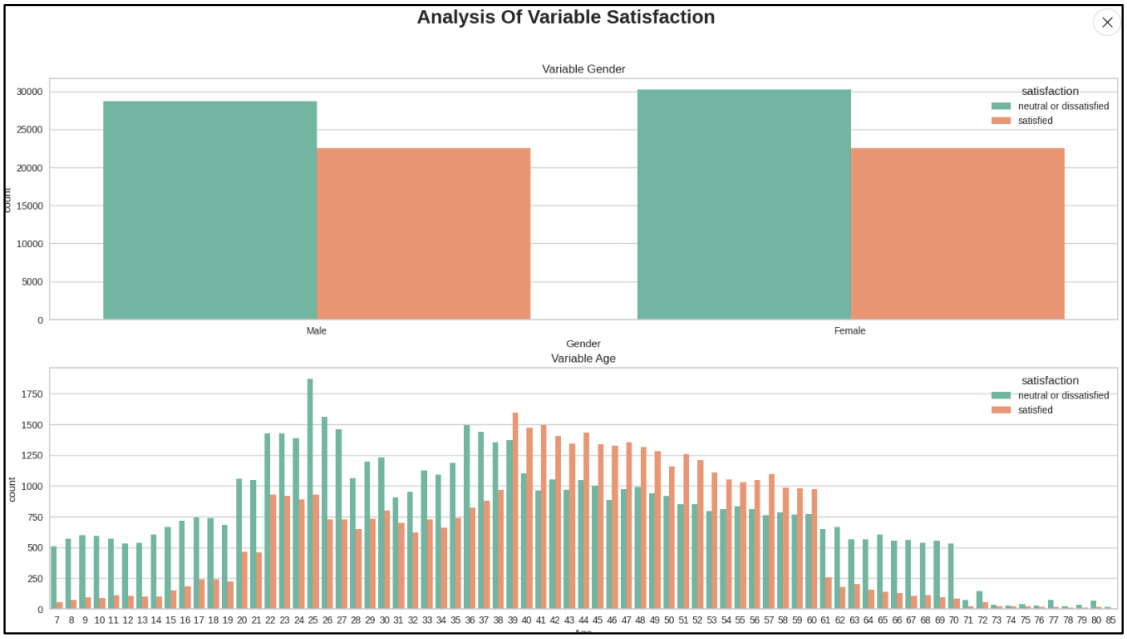

Рисунок 3.8 – Взаємозв'язок між рівнями задоволення та змінними "стать" та "вік"

З графіків випливає, що чоловіки мають вищий рівень задоволення порівняно з жінками, а також, що із зростанням віку людей збільшується їхня задоволеність.

Проведено аналіз, як рівень задоволення залежить від класу та типу подорожі (рис 3.9).

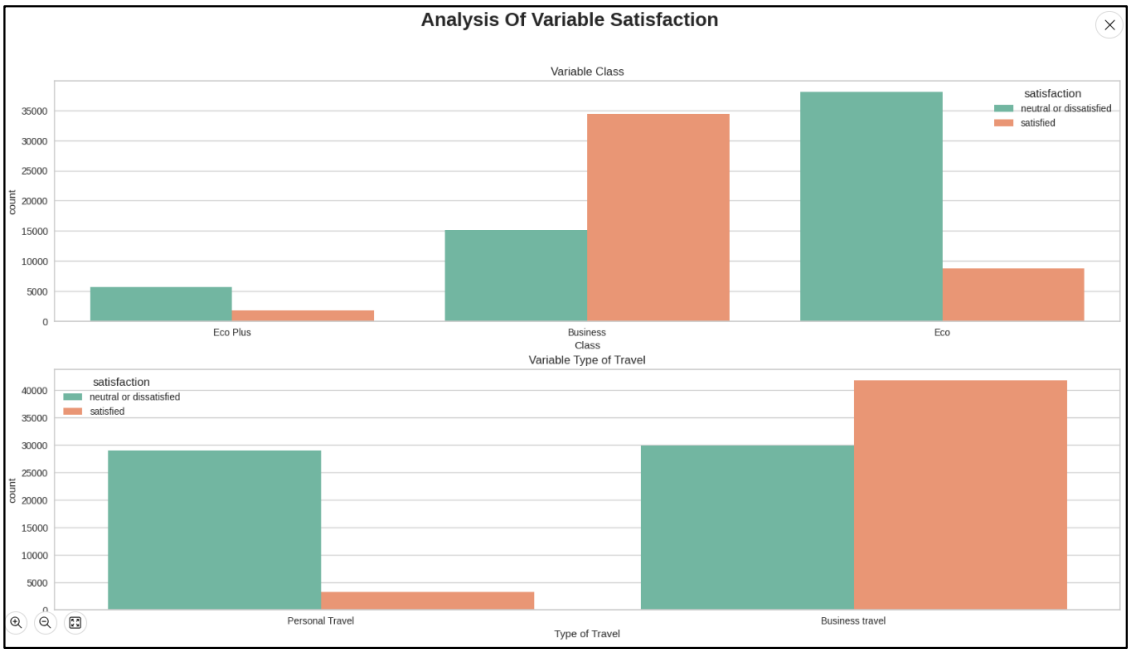

Рисунок 3.9 – Розподіл за класом та типом подорожі

Аналіз показав, що більшість пасажирів, особливо ті, хто подорожує бізнес-класом чи у рамках ділових подорожей, виявляють високий рівень задоволення. Це має важливе значення для авіакомпаній з кількох причин:

– Поліпшення обслуговування клієнтів: Розуміння того, які категорії пасажирів найбільш задоволені, дозволяє авіакомпаніям зосередитися на поліпшенні обслуговування в цих сегментах. Наприклад, якщо пасажири бізнес-класу особливо задоволені, авіакомпанія може використовувати ці результати для посилення своїх сильних сторін у цьому класі;

– Розробка маркетингових стратегій: Знання того, що бізнес-клас і ділові подорожі отримують найвищий рівень задоволення, дозволяє маркетинговим командам створювати цільові кампанії, спрямовані на залучення більшої кількості клієнтів у ці сегменти;

– Підвищення лояльності клієнтів: Виявлення факторів, що сприяють високому рівню задоволеності, дозволяє компаніям створювати програми лояльності та пропозиції, які зміцнюють довіру та прихильність клієнтів;

– Покращення конкурентоспроможності: Авіакомпанія, яка постійно аналізує та вдосконалює свій сервіс на основі зворотного зв’язку від клієнтів, може зміцнити свою позицію на ринку, випереджаючи конкурентів.

Таким чином, аналіз задоволеності клієнтів залежно від класу і типу подорожі є важливим інструментом для підвищення ефективності обслуговування, стратегічного планування і розвитку бізнесу.

3.3 Використання моделей машинного навчання для аналізу даних

Першим кроком було виконано додаткову обробку даних, включаючи видалення непотрібних стовпців, кодування категоріальних змінних та масштабування числових ознак (рис 3.10).

```
test = test.drop('id', axis = 1)

hot = pd.get_dummies(test[['Gender', 'Customer Type', 'Type of Travel', 'Class']])
test = pd.concat([test, hot], axis = 1)
test = test.drop(['Gender', 'Customer Type', 'Type of Travel', 'Class'], axis = 1)

X = test.drop('satisfaction', axis = 1)
X = X.values
y = test['satisfaction']

scaler = StandardScaler()
X_standard = scaler.fit_transform(X)

X_train,X_test, y_train, y_test = train_test_split(X_standard, y, test_size = 0.3, random_state =
0)
```

Рисунок 3.10 – Обробка даних

Далі, виконано налаштування моделі дерева рішень за допомогою випадкового пошуку параметрів. Після заміни пропущених значень у тренувальному та тестовому наборах даних на медіани, модель була навчена з використанням різних комбінацій гіперпараметрів, таких як глибина дерева, критерій вибору поділу та мінімальна кількість зразків для поділу вузла.

Найкращі параметри моделі, які були визначені через рандомізований пошук, включають мінімальну кількість зразків для поділу (Min Split) рівну 2, максимальну глибину (Max Depth) рівну 11 і критерій вибору поділу (Criterion) - "entropy" (рис. 3.11).

```
imputer = SimpleImputer(strategy='median')
X_train_imputed = imputer.fit_transform(X_train)
X_test_imputed = imputer.transform(X_test)

parameters = {'max_depth': [3, 4, 5, 6, 7, 9, 11],
              'min_samples_split': [2, 3, 4, 5, 6, 7],
              'criterion': ['entropy', 'gini']
             }

model = DecisionTreeClassifier()

gridDecisionTree = RandomizedSearchCV(model, parameters, cv=3, n_jobs=-1)
gridDecisionTree.fit(X_train_imputed, y_train)

print('Min Split: ', gridDecisionTree.best_estimator_.min_samples_split)
print('Max Depth: ', gridDecisionTree.best_estimator_.max_depth)
print('Criterion: ', gridDecisionTree.best_estimator_.criterion)
print('Score: ', gridDecisionTree.best_score_)
```

```
Min Split:  2
Max Depth:  11
Criterion:  entropy
Score:  0.9358741681790684
```

Рисунок 3.11 – Приклад коду та результат

Здійснено обробку пропущених значень у тренувальному та тестовому наборах даних за допомогою медіанного заповнення. Після цього проведено навчання моделі класифікації дерева рішень з використанням певних гіперпараметрів, зокрема критерію 'entropy', мінімальної кількості зразків для поділу 6 та максимальної глибини дерева 6. В результаті отримані прогнози міток класів для тренувального та тестового наборів даних (рис. 3.12).

```
# Обробка пропущених значень у тренувальних та тестових даних
X_train_imputed = imputer.fit_transform(X_train)
X_test_imputed = imputer.transform(X_test)

# Навчання моделі Decision Tree
decision_tree = DecisionTreeClassifier(criterion='entropy', min_samples_split=6, max_depth=6, rand
om_state=0)
decision_tree.fit(X_train_imputed, y_train)
previsoes = decision_tree.predict(X_test_imputed)

# Передбачення на тестовому наборі
test_predictions = decision_tree.predict(X_test_imputed)

# Передбачення на тренувальному наборі
train_predictions = decision_tree.predict(X_train_imputed)

# Обчислення точності для тренувального та тестового наборів
train_accuracy2 = accuracy_score(y_train, train_predictions)
test_accuracy2 = accuracy_score(y_test, test_predictions)

print("Train Accuracy:", train_accuracy2)
print("Test Accuracy:", test_accuracy2)

# Побудова матриці конфузії для тестового набору
cm_matrix = confusion_matrix(y_test, test_predictions)
```

Рисунок 3.12 – Приклад коду

Після цього була проведена оцінка точності моделі для тренувального та тестового наборів даних, яка становить 91.85% та 91.38% відповідно. Для оцінки класифікації моделі на тестовому наборі даних була побудована матриця конфузії (рис. 3.13).

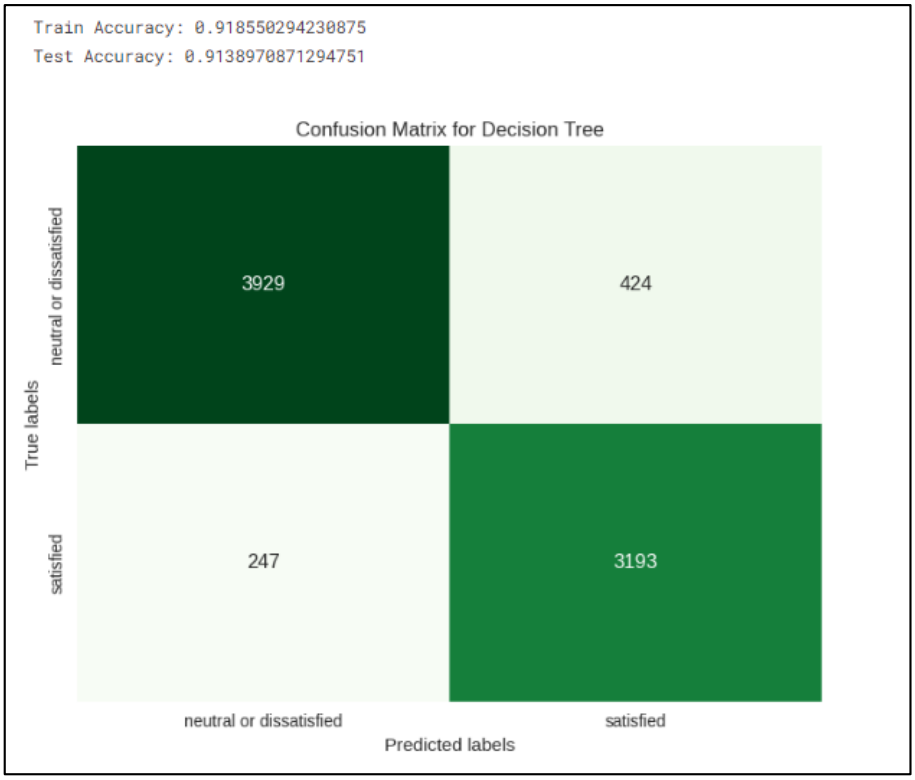

Рисунок 3.13 – Матриця конфузії для моделі Decision Tree

Також було отримано звіт про класифікацію, який містить метрики precision, recall, f1-score та підтримку (support) для двох класів: "neutral or dissatisfied" і "satisfied". Результати показують, що модель має високу точність в передбаченні обох класів, проте, відмічається невелике зниження в точності для класу "satisfied". Загальна точність моделі на тестовому наборі даних складає 91%. Середнє значення precision, recall та f1-score для обох класів також становить 91%, що свідчить про добру збалансованість моделі (рис. 3.14).

	precision	recall	f1-score	support
neutral or dissatisfied	0.94	0.90	0.92	4353
satisfied	0.88	0.93	0.90	3440
accuracy			0.91	7793
macro avg	0.91	0.92	0.91	7793
weighted avg	0.92	0.91	0.91	7793

Рисунок 3.14 – Класифікаційний звіт моделі рішень

Відсортовано список важливості ознак моделі рішень. Результат показує важливість кожної ознаки у класифікації задоволеності клієнтів. Наприклад, "Online boarding" має найбільший вплив на модель, в той час як "Class_Eco" не має жодного впливу (рис. 3.15).

```
columns = test.drop('satisfaction', axis = 1).columns
feature_imp = pd.Series(decision_tree.feature_importances_, index = columns).sort_values(ascending
= False)
feature_imp
```

```
Online boarding                    0.400757
Inflight wifi service              0.233725
Type of Travel_Business travel     0.154129
Inflight entertainment             0.063268
Class_Business                     0.039611
Customer Type_Loyal Customer       0.029963
Customer Type_disloyal Customer    0.021490
Checkin service                    0.013409
Inflight service                   0.010281
Type of Travel_Personal Travel     0.008106
Seat comfort                       0.007516
Gate location                      0.005679
Baggage handling                   0.003190
Arrival Delay in Minutes           0.002956
Age                                0.002389
Leg room service                   0.001360
Ease of Online booking             0.001146
Departure Delay in Minutes         0.000766
Flight Distance                    0.000262
Class_Eco                          0.000000
Unnamed: 0                         0.000000
Gender_Male                        0.000000
Gender_Female                      0.000000
Cleanliness                        0.000000
On-board service                   0.000000
Food and drink                     0.000000
Departure/Arrival time convenient  0.000000
Class_Eco Plus                     0.000000
dtype: float64
```

Рисунок 3.15 – Приклад коду та його результат

Проведено налаштування моделі RandomForestClassifier за допомогою рандомізованого пошуку по заданим гіперпараметрам. Після обробки пропущених значень у тренувальному та тестовому наборах даних за допомогою медіанного заповнення, модель навчається і оцінюється на тренувальному наборі. Результати показують найкращі параметри для моделі, включаючи використану алгоритм ('gini'), отриману точність (94.85%), мінімальну кількість зразків для поділу (2) та максимальну глибину (11). Фрагмент коду зображено на рисунку 3.16.

```
parameters = {'max_depth': [3, 4, 5, 6, 7, 9, 11],
              'min_samples_split': [2, 3, 4, 5, 6, 7],
              'criterion': ['entropy', 'gini']
             }

imputer = SimpleImputer(strategy='median')
X_train_imputed = imputer.fit_transform(X_train)
X_test_imputed = imputer.transform(X_test)

model = RandomForestClassifier()
gridRandomForest = RandomizedSearchCV(model, parameters, cv=5, n_jobs=-1)
gridRandomForest.fit(X_train_imputed, y_train)

print('Algorithm: ', gridRandomForest.best_estimator_.criterion)
print('Score: ', gridRandomForest.best_score_)
print('Min Split: ', gridRandomForest.best_estimator_.min_samples_split)
print('Max Nvl: ', gridRandomForest.best_estimator_.max_depth)
```

Рисунок 3.16 – Фрагмент коду

Наступним кроком було проведено навчання моделі Random Forest та оцінено її точність на тренувальному та тестовому наборах даних. Результати показують, що модель має високу точність на обох наборах: 92.23% на тренувальному та 91.63% на тестовому.

Далі, будується матриця конфузії для оцінки класифікації моделі на тестовому наборі даних (рис. 3.17).

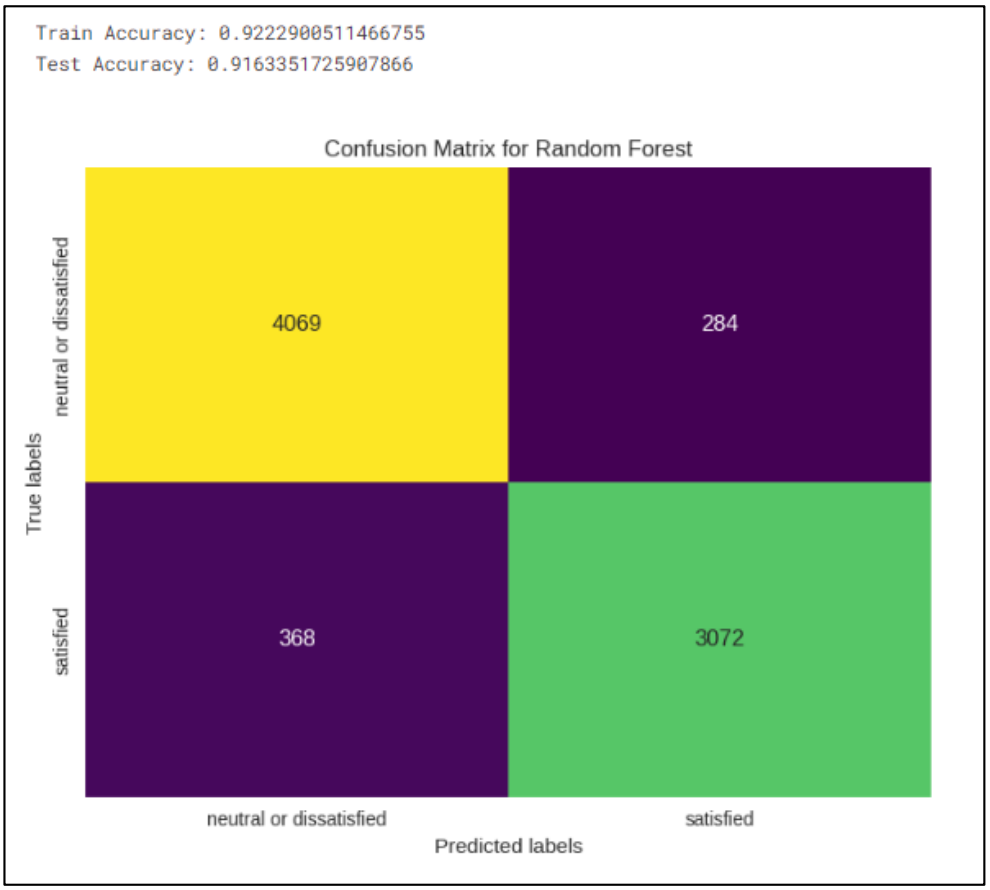

Рисунок 3.17 – Матриця конфузії для моделі Random Forest

Наступним кроком було використано SimpleImputer для заповнення відсутніх значень медіаною у тренувальному та тестовому наборах даних. Далі, ініціалізовано та навчено модель KNN з використанням навчальних даних. Після цього модель використовується для передбачення класів для тренувального та тестового наборів даних. Точність моделі обчислюється на основі фактичних та передбачених класів, а також побудована матриця конфузії для оцінки результатів класифікації на тестовому наборі даних.

На рисунку 3.18 зображено матрицю конфузії для моделі KNeighbors.

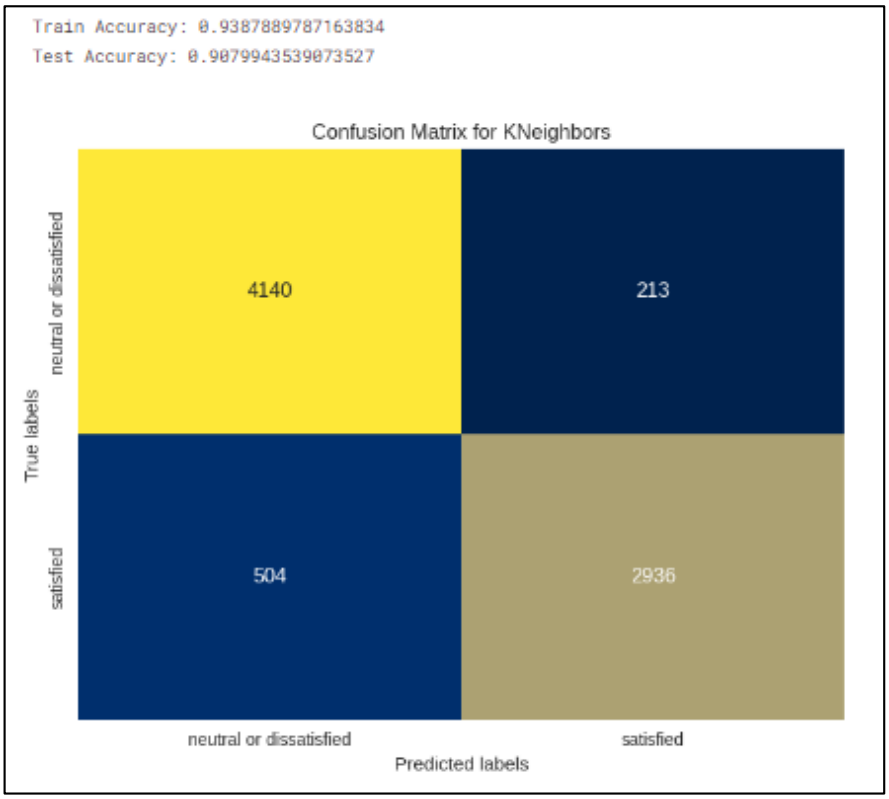

Рисунок 3.18 – Матриця конфузії для моделі KNeighbors

Точність моделі на тренувальному наборі вище, ніж на тестовому, що може вказувати на перенавчання моделі. Однак, в цілому, модель демонструє добрі показники точності на обох наборах даних.

На рисунку 3.19 зображено звіт про класифікацію для моделі KNeighbors.

	precision	recall	f1-score	support
neutral or dissatisfied	0.89	0.95	0.92	4353
satisfied	0.93	0.85	0.89	3440
accuracy			0.91	7793
macro avg	0.91	0.90	0.91	7793
weighted avg	0.91	0.91	0.91	7793

Рисунок 3.19 – Звіт про класифікацію

Результати моделі KNN показують дуже високу точність загалом (приблизно 91%). Модель має високі показники як для класу "нейтральні або незадоволені" (точність 0.89, повнота 0.95, F1-score 0.92), так і для класу

"задоволені" (точність 0.93, повнота 0.85, F1-score 0.89). Це означає, що модель добре впоралася з класифікацією обох класів, але її здатність правильно визначати "задоволені" відгуки може бути трохи нижчою.

Далі було використано SimpleImputer зі стратегією median для заповнення відсутніх значень медіаною у тренувальному (X_train) та тестовому (X_test) наборах даних. Після цього ініціалізовано та навчено модель AdaBoostClassifier з параметрами, які вказуються в конструкторі: n_estimators=500 (кількість базових моделей у комітеті) та learning_rate=0.02 (швидкість навчання). Модель використовується для передбачення класів для тренувального та тестового наборів даних. Точність моделі обчислюється на тренувальному та тестовому наборах даних за допомогою функції accuracy_score.

На рисунку 3.20 побудовано матрицю конфузії для оцінки результатів класифікації на тестовому наборі даних, яка відображає фактичні та передбачені класи.

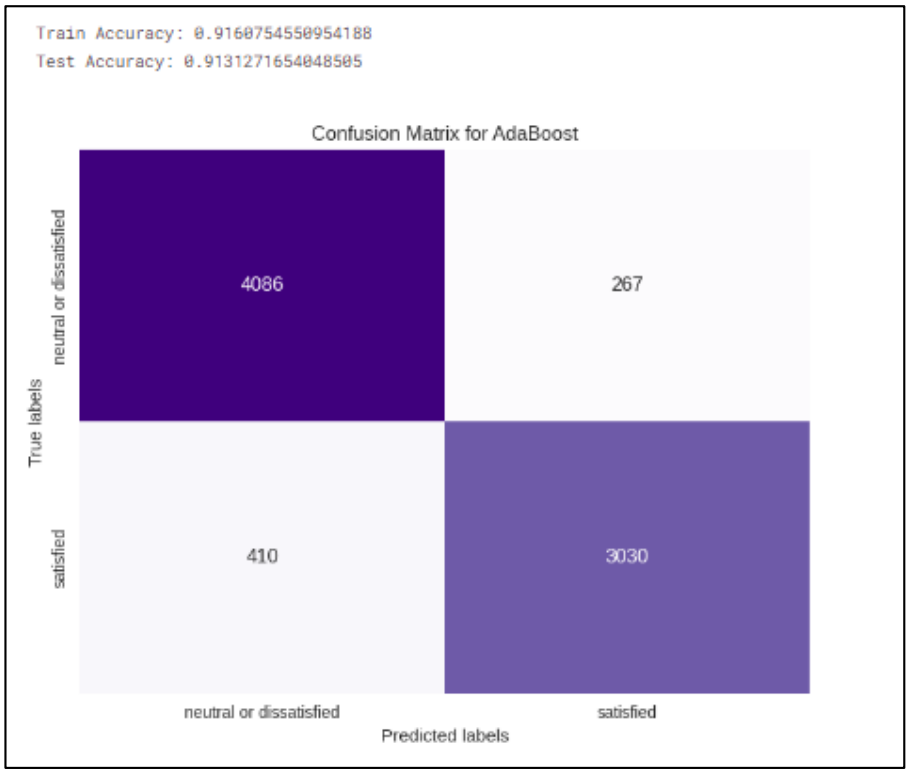

Рисунок 3.20 – Матриця конфузії для моделі AdaBoost

Модель AdaBoost демонструє досить високу точність як на тренувальному наборі даних (близько 91.61%), так і на тестовому наборі даних (близько 91.31%).

Це свідчить про те, що модель ефективно навчилася і добре узагальнюється на нових даних. Таке невелике зниження точності на тестовому наборі порівняно з тренувальним може бути пов'язане з незначною перенавчаністю моделі або невеликою кількістю даних у випадку тестування. У цілому, ці значення свідчать про те, що модель добре працює як на навчальних, так і на тестових даних.

На рисунку 3.21 зображено класифікаційний звіт для моделі AdaBoost.

	precision	recall	f1-score	support
neutral or dissatisfied	0.91	0.94	0.92	4353
satisfied	0.92	0.88	0.90	3440
accuracy			0.91	7793
macro avg	0.91	0.91	0.91	7793
weighted avg	0.91	0.91	0.91	7793

Рисунок 3.21 – Класифікаційний звіт моделі AdaBoost

Результати моделі AdaBoost також показують дуже високу точність загалом (приблизно 91%). Модель має дуже близькі показники для обох класів: для "нейтральних або незадоволених" відгуків точність 0.91, повнота 0.94 і F1-score 0.92, а для "задоволених" відгуків - точність 0.92, повнота 0.88 і F1-score 0.90. Це означає, що модель AdaBoost добре впоралася з класифікацією обох класів, і вона має дуже подібні показники до моделі KNN.

Для здійснення передбачення для моделі GradientBoost спочатку було заповнено значення відсутніх даних за допомогою SimpleImputer зі стратегією mean для тренувального (X_train) та тестового (X_test) наборів даних. Після цього була навчена модель GradientBoostingClassifier з параметрами: n_estimators=300 (кількість дерев у комітеті) та learning_rate=0.01 (швидкість навчання) з використанням тренувального набору.

Модель застосовувалась для передбачення класів для тестового набору даних. Також було обчислено точність моделі на тренувальному та тестовому наборах даних.

На рисунку 3.22 побудовано матрицю конфузії.

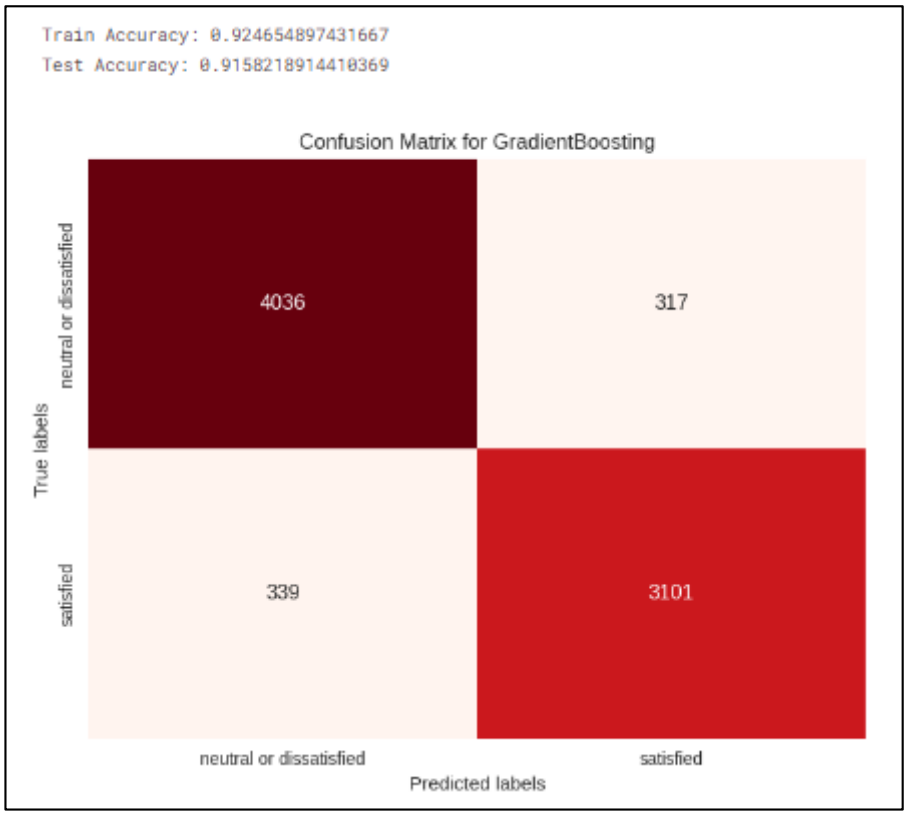

Рисунок 3.22 – Матриця конфузії для моделі GradientBoost

Результати передбачення вказують, що модель показує високу точність як на тренувальних, так і на тестових даних, що свідчить про її ефективність в передбаченні класів.

На рисунку 3.23 зображено звіт класифікації моделі GradientBoost.

	precision	recall	f1-score	support
neutral or dissatisfied	0.92	0.93	0.92	4353
satisfied	0.91	0.90	0.90	3440
accuracy			0.92	7793
macro avg	0.91	0.91	0.91	7793
weighted avg	0.92	0.92	0.92	7793

Рисунок 3.23 – Звіт класифікації моделі GradientBoost

Результати моделі GradientBoost також показують дуже високу точність загалом (приблизно 92%). Модель має дуже близькі показники для обох класів:

для "нейтральних або незадоволених" відгуків точність 0.92, повнота 0.93 і F1-score 0.92, а для "задоволених" відгуків - точність 0.91, повнота 0.90 і F1-score 0.90. Це означає, що модель GradientBoost також добре впоралася з класифікацією обох класів, і вона має подібні показники до моделей KNN та AdaBoost.

Також було навчено модель Naive Bayes (клас GaussianNB з sklearn) на оброблених тренувальних даних та здійснено передбачення для тестових та тренувальних наборів даних за допомогою навченої моделі. Обчислено точність моделі для тренувального та тестового наборів даних (train_accuracy1 та test_accuracy1) і виводить ці значення.

На рисунку 3.24 зображено фрагмент коду навчання моделі Naive Bayes.

```
import numpy as np
import pandas as pd
from sklearn.impute import SimpleImputer
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt


# X_train, X_test, y_train, y_test повинні бути визначені раніш

imputer = SimpleImputer(strategy='median')

# Обробка пропущених значень у тренувальних та тестових даних
X_train_imputed = imputer.fit_transform(X_train)
X_test_imputed = imputer.transform(X_test)

# Навчання моделі Naive Bayes
naive_bayes = GaussianNB()
naive_bayes.fit(X_train_imputed, y_train)
previsoes = naive_bayes.predict(X_test_imputed)

# Передбачення на тестовому наборі
test_predictions = naive_bayes.predict(X_test_imputed)
```

Рисунок 3.24 – Навчання моделі Naive Bayes

Створено матрицю конфузії для тестового набору і візуалізовано її за допомогою бібліотеки seaborn, що дозволяє оцінити якість класифікації.

На рисунку 3.25 зображено матрицю конфузії для моделі Naive Bayes.

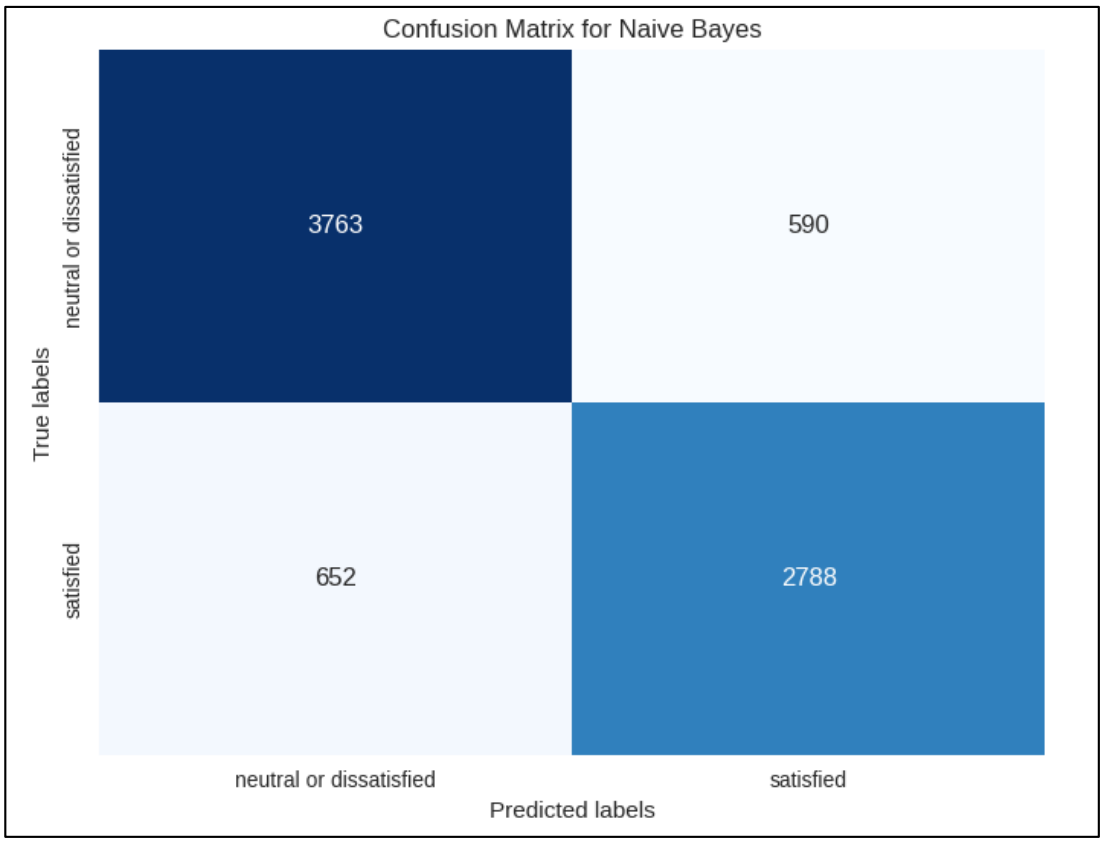

Рисунок 3.25 – Матриця конфузії для моделі Naive Bayes

Результати моделі Naive Bayes на даному наборі даних виглядають так:

– Точність на тренувальному наборі (Train Accuracy): 0.843;
– Точність на тестовому наборі (Test Accuracy): 0.841.

Наступним кроком було обчислено та виведено звіт класифікації за допомогою алгоритму Gaussian (рис. 3.26).

	precision	recall	f1-score	support
neutral or dissatisfied	0.85	0.86	0.86	4353
satisfied	0.83	0.81	0.82	3440
accuracy			0.84	7793
macro avg	0.84	0.84	0.84	7793
weighted avg	0.84	0.84	0.84	7793

Рисунок 3.26 – Звіт класифікації

Отримане значення score_naive_gaussian дорівнює 0.8406262030026947, що означає, що модель Naive Bayes досягла досить високої точності у класифікації тестового набору даних.

Для здійснення передбачення для моделі LGBM спочатку було використано SimpleImputer зі стратегією "mean" для заповнення відсутніх значень середнім значенням у тренувальному (X_train) та тестовому (X_test) наборах даних. Навчання моделі LGBMClassifier відбулося з використанням конструктора, в якому було вказано параметри моделі. Після цього модель було використано для передбачення класів для тестового набору даних. Обчислення точності проводилося за допомогою методу score для тренувального та тестового наборів даних. На завершення була побудована матриця конфузії для оцінки результатів класифікації на тестовому наборі даних.

На рисунку 3.27 зображено матрицю конфузії для моделі LGBM.

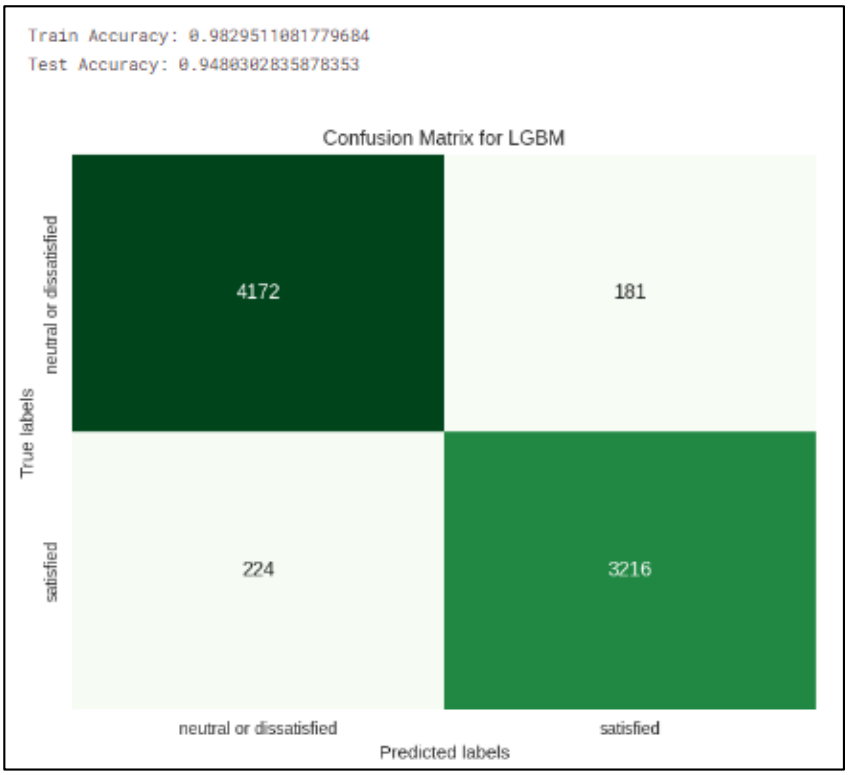

Рисунок 3.27 – Матриця конфузії для моделі LGBM

Отримані результати показують високу точність моделі на якості навчального та тестового наборів даних. Така висока точність на обох наборах свідчить про те, що модель добре узагальнює свої знання на нові дані, а не просто "запам'ятовує" тренувальні дані. Різниця між точністю на тренувальному та тестовому наборах є невеликою, що означає відсутність перенавчання.

Оскільки точність на тестовому наборі даних також досить висока (близько 94.80%), можна стверджувати, що модель відмінно працює на нових, раніше невикористаних даних. Такі результати є дуже задовільними і свідчать про ефективність моделі LGBMClassifier у вирішенні задачі класифікації для даного набору даних.

На рисунку 3.28 зображено звіт класифікації моделі LGBM.

	precision	recall	f1-score	support
neutral or dissatisfied	0.95	0.96	0.95	4353
satisfied	0.95	0.93	0.94	3440
accuracy			0.95	7793
macro avg	0.95	0.95	0.95	7793
weighted avg	0.95	0.95	0.95	7793

Рисунок 3.28 – Звіт класифікації моделі LGBM

Результати цієї моделі показують дуже високі показники точності (95%) і здатності правильно визначати обидва класи. Точність для обох класів є дуже високою: 95% для класу "нейтральні або незадоволені" та 95% для класу "задоволені". Повнота також досить висока: 96% для "нейтральні або незадоволені" та 93% для "задоволені". F1-score для обох класів також становить 0.95.

Це свідчить про те, що модель має дуже добре показники як для точності, так і для повноти в класифікації обох класів. Високий рівень точності та повноти означає, що модель ефективно впоралася з класифікацією відгуків як "нейтральні або незадоволені", так і "задоволені".

На рисунку 3.29 зображено загальну таблицю з результатами усіх восьми моделей.

	Model	Train Precision	Test Precision
0	Naive Bayes	0.843150	0.840626
1	Decision Tree	0.918550	0.913897
2	Random Forest	0.922290	0.916335
3	Extra Trees	0.926855	0.918003
4	KNN	0.938789	0.907994
5	Logistic Regression	0.872023	0.868215
6	AdaBoost	0.916075	0.913127
7	GradientBoost	0.924655	0.915822
8	LGBM	0.982951	0.948030

Рисунок 3.29 – Загальні результати

3.4 Висновки

У програмі було розроблено передбачення для восьми різних моделей машинного навчання, включаючи Naive Bayes, Decision Tree, Random Forest, Extra Trees, KNN, Logistic Regression, AdaBoost та LGBM. У звіті, для прикладу, наведено результати лише п’яти моделей, а саме: AdaBoost, Decision Tree, GradientBoost, LGBM та KNeighbors.

Загальний аналіз результатів показує, що модель LGBM продемонструвала найвищу точність прогнозування за метрикою F1_Score на тренувальних даних з результатом 0,98 та тестувальних даних з результатом 0,94 що свідчить про ефективність моделі у вирішенні задачі класифікації для даного набору даних.

Серед моделей, які показали стабільно високі показники як на тренувальному, так і на тестовому наборі, можна виділити Decision Tree, Random Forest, Extra Trees, AdaBoost та GradientBoost. Ці моделі демонструють гарні результати з точністю близькою до 90% як на тренувальному, так і на тестовому наборі даних.

ВИСНОВКИ

Бакалаврська дипломна робота присвячена інформаційній технології передбачення рівнів задоволеності пасажирів авіакомпаніями.

Здійснено огляд існуючих систем, що дало можливість отримати об'єктивне уявлення про те, які підходи та технології вже використовуються в області передбачення рівнів задоволеності пасажирів авіакомпаніями.

Здійснено підготовку даних для подальшого аналізу було важливим кроком у дослідженні, адже цей процес включав збір та очищення даних, щоб мати змогу ефективно використовувати їх для побудови моделей та проведення аналізу.

Розвідувальний аналіз даних дозволив отримати детальне розуміння щодо особливостей даних та їх взаємозв'язків. Цей етап допоміг усвідомити, які фактори можуть впливати на задоволеність пасажирів авіакомпаніями та визначити ключові залежності.

Розроблено інформаційну технологію на основі восьми різних моделей машинного навчання, включаючи Naive Bayes, Decision Tree, Random Forest, Extra Trees, KNN, Logistic Regression, AdaBoost та LGBM. Кожна модель була налаштована і протестована для забезпечення максимальної точності передбачення задоволеності пасажирів.

Після аналізу результатів побудованих моделей машинного навчання було проведено оцінку і виявлено, що модель LGBM продемонструвала найвищу точність прогнозування за метрикою F1_Score на тренувальних даних з результатом 0,98 та тестувальних даних з результатом 0,94 що свідчить про ефективність моделі у вирішенні задачі класифікації для даного набору даних.

За результатами роботи було опубліковано тези на тему «Інформаційна технологія передбачення рівнів задоволеності пасажирів авіакомпаніями» на міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (м. Вінниця, 2023-2024 рр.) [1].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Судець А. О., Козачко О. М. Інформаційна технологія передбачення рівнів задоволеності для пасажирів авіакомпаніями. *Міжнародна науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи»*. (Вінниця, 2023-2024 рр). URL: https://conferences.vntu.edu.ua/index.php/mn/mn2024/author/submission/21234.

2. Qiang Li, Ranzhe Jing, Xihua Zhu, Determinants of travel satisfaction for commercial airlines: A data mining approach, Engineering Applications of Artificial Intelligence, Volume 133, Part F, 2024. DOI: https://doi.org/10.1016/j.engappai.2024.108597.

3. Які українські авіакомпанії працюють за кордоном, 2024. [Електронний ресурс]. URL: https://thepage.ua/ua/news/yaki-ukrayinski-aviakompaniyi-pracyuyut-za-kordonom-v-2024-roci.

4. World Airline Awards | SKYTRAX, 2023-2024. [Електронний ресурс]. URL: https://www.worldairlineawards.com/.

5. AirlineRatings, 2024. URL: https://www.airlineratings.com/.

6. Gosegu, Kaggle Notebook «Airline Passenger Satisfaction» Updated 12 hours ago. URL: https://www.kaggle.com/code/seguride/airline-passenger-satisfaction.

7. Suchintikasarkar, Kaggle Notebook «Will the Passenger be Satisfied?| ACC 94%» Updated 8 days ago. URL: https://www.kaggle.comn]\5/code/suchintikasarkar/will-the-passenger-be-satisfied-acc-94#Decision-Tree.

8. Kaggle. URL: https://www.kaggle.com/.

9. Python Introduction. URL: https://www.w3schools.com/python/python_intro.asp.

10. What is a Decision Tree | IBM. URL: https://www.ibm.com/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a,internal%20nodes%20and%20leaf%20nodes.

11. Random Forest Algorithm in Machine Learning . URL: https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/.

12. Gradient Boosting Algorithm: A Complete Guide for Beginners Learning . URL: https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/#:~:text=our%20test%20data.-,What%20is%20Gradient%20Boosting%3F,%2C%20typically%20decision%20trees%2C%20sequentially.

13. K-Nearest Neighbor (KNN) Algorithm. URL: https://www.ibm.com/topics/knn#:~:text=What%20is%20the%20KNN%20algorithm,of%20an%20individual%20data%20point .

14. Plotly Express in Python. URL: https://plotly.com/python/plotly-express/

15. What is Pandas? URL: https://www.w3schools.com/python/pandas/pandas_intro.asp.

16. NumPy [Електронний ресурс]. URL: https://numpy.org/.

17. What is Exploratory Data Analysis? [Електронний ресурс]. URL: https://www.geeksforgeeks.org/what-is-exploratory-data-analysis/ .

18. Мокін В. Б., Дратований М. В. Наука про дані: машинне навчання та інтелектуальний аналіз даних. Вінниця: ВНТУ, 2024. – 258 с. [Електронний ресурс]. URL: https://docs.vntu.edu.ua/card.php?id=8163 .

19. Exploratory Data Analysis in Python, 2022. [Електронний ресурс]. URL: https://towardsdatascience.com/exploratory-data-analysis-in-python-a-step-by-step-process-d0dfa6bf94ee .

20. Anna Sudets, Kaggle Notebook «Airline_BD» Updated 11 days ago. [Електронний ресурс]. URL: https://www.kaggle.com/code/annasudets/airline-bd .

Додаток А

(обов’язковий)

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
“____”_____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на бакалаврську дипломну роботу

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ РІВНІВ ЗАДОВОЛЕНОСТІ ПАСАЖИРІВ АВІАКОМПАНІЯМИ

08-34.БДР.016.00.000 ТЗ

Керівник: к.т.н., доц.
_____ Олексій КОЗАЧКО
«__» _____ 2024 р.
Розробила студентка гр. 2ІСТ-20б
_____ Анна СУДЕЦЬ
«__» _____ 2024 р.

Вінниця 2024

1. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____2024р., та індивідуальне завдання на БДР, затверджене протоколом №__ засідання кафедри САІТ від «__» _____ 2024р.

2. Джерела розробки:

1) Qiang Li, Ranzhe Jing, Xihua Zhu, Determinants of travel satisfaction for commercial airlines: A data mining approach, Engineering Applications of Artificial Intelligence, Volume 133, Part F, 2024. DOI: https://doi.org/10.1016/j.engappai.2024.108597;

2) Мокін В. Б., Дратований М. В. Наука про дані: машинне навчання та інтелектуальний аналіз даних. Вінниця: ВНТУ, 2024. – 258 с. URL: https://docs.vntu.edu.ua/card.php?id=8163.

3. Мета і призначення роботи.

Метою дослідження є підвищення точності передбачення рівнів задоволеності пасажирів авіакомпаніями, шляхом створення інформаційної технології та використання методів машинного навчання.

4. Вихідні дані для проведення робіт:

Датасет Kaggle «Airline Passenger Satisfaction» з даними для передбачення рівнів задоволеності пасажирів авіакомпаніями.

5. Методи дослідження:
   - Підготовка даних;
   - Розвідувальний аналіз;
   - Моделі машинного навчання для передбачення даних;

6. Етапи роботи і терміни їх виконання:

a) Характеристика об’єкту досліджень _____ – _____
b) Аналіз методів машинного навчання _____ – _____
c) Проведення розвідувального аналізу _____ – _____
d) Розроблення інформаційної технології _____ – _____
e) Оформлення матеріалів до захисту БДР _____ – _____

7. Очікувані результати та порядок реалізації

Розроблення інформаційної технології передбачення рівнів задоволеності пасажирів авіакомпаніями.

8. Вимоги до розробленої документації

Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання бакалаврських кваліфікаційних робіт для студентів спеціальностей: 124 «Системний аналіз», 126 «Інформаційні системи та технології» (освітня програма «Прикладні інформаційні технології»)».

9. Порядок приймання роботи

Публічний захист «___» _____ 2024 р.
Початок розробки «___» _____ 2024 р.
Граничні терміни виконання БДР «___» _____ 2024 р.

Розробила студентка групи 2ІСТ-20б _____ Анна СУДЕЦЬ

Додаток Б

(обов'язковий)

Протокол перевірки бакалаврської дипломної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія рівнів задоволеності пасажирів авіакомпаніями»
Тип роботи: бакалаврська дипломна робота
Підрозділ: кафедра САІТ

Показники звіту подібності Unicheck

Оригінальність 91,83 % Схожість 8,17 %

Аналіз звіту подібності (відмітити потрібне)

• Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
o Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
o Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ (підпис) Сергій ЖУКОВ

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ (підпис) Анна СУДЕЦЬ

Керівник роботи _____ (підпис) Олексій КОЗАЧКО

Додаток В

(довідниковий)

Фрагмент лістингу програми

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import seaborn as sns
import plotly.express as px
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from yellowbrick.classifier import ConfusionMatrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
from sklearn.preprocessing import LabelEncoder
from sklearn.impute import SimpleImputer
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.impute import SimpleImputer
from sklearn.model_selection import RandomizedSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import train_test_split, cross_val_score, learning_cur
ve
import joblib


import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
directory = "/kaggle/input/airline-passenger-satisfaction/"
feature_tables = ['train.csv', 'test.csv']

df_train_path = directory + feature_tables[0]
df_test_path = directory + feature_tables[1]

print(f'Reading csv from {df_train_path}...')
train = pd.read_csv(df_train_path)
print('...Complete')

print(f'Reading csv from {df_test_path}...')
test = pd.read_csv(df_test_path)
print('...Complete')

corr = train.corr(numeric_only=True).round(2)
plt.figure(figsize=(25, 20))
```

```
sns.heatmap(corr, annot=True, cmap='YlOrBr')
plt.show()

plt.figure(figsize=(20, 10))
gs = gridspec.GridSpec(2, 2, height_ratios=[1, 1])

plt.subplot(gs[0, 0])
plt.gca().set_title('Variable Gender')
sns.countplot(x='Gender', palette='Set2', data=train)

plt.subplot(gs[0, 1])
plt.gca().set_title('Variable Customer Type')
sns.countplot(x='Customer Type', palette='Set2', data=train)

plt.subplot(gs[1, :])
plt.gca().set_title('Variable Age')
sns.countplot(x='Age', palette='Set2', data=train)

plt.show()

plt.figure(figsize=(20, 15))
gs = gridspec.GridSpec(3, 2, height_ratios=[1, 1, 1])

plt.subplot(gs[0, 0])
plt.gca().set_title('Variable Type of Travel')
sns.countplot(x='Type of Travel', palette='Set2', data=train)

plt.subplot(gs[0, 1])
plt.gca().set_title('Variable Class')
sns.countplot(x='Class', palette='Set2', data=train)

plt.subplot(gs[1, :])
plt.gca().set_title('Variable Flight Distance')
sns.countplot(x='Flight Distance', palette='Set2', data=train)

plt.show()

min_flight_distance = np.min(train['Flight Distance'])
mean_flight_distance = np.mean(train['Flight Distance'])
max_flight_distance = np.max(train['Flight Distance'])

print(f"Min Flight Distance: {min_flight_distance}")
print(f"Mean Flight Distance: {mean_flight_distance:.2f}")
print(f"Max Flight Distance: {max_flight_distance}")

train['Departure Delay in Minutes'].fillna(0, inplace=True)
train['Arrival Delay in Minutes'].fillna(0, inplace=True)

train['Departure Delay in Minutes'] = train['Departure Delay in Minutes'].astype(in
t)
train['Arrival Delay in Minutes'] = train['Arrival Delay in Minutes'].astype(int)

plt.figure(figsize=(20, 15))
gs = gridspec.GridSpec(2, 1, height_ratios=[1, 1])

plt.subplot(gs[0, :])
plt.gca().set_title('Variable Departure Delay in Minutes')
sns.countplot(x='Departure Delay in Minutes', palette='Set2', data=train)
plt.gca().set_xlim([0, 60])
```

```
plt.subplot(gs[1, :])
plt.gca().set_title('Variable Arrival Delay in Minutes')
sns.countplot(x='Arrival Delay in Minutes', palette='Set2', data=train)
plt.gca().set_xlim([0, 60])

plt.show()

plt.figure(figsize = (20, 10))
plt.suptitle("Analysis Of Variable Satisfaction",fontweight="bold", fontsize=20)

plt.subplot(2, 1, 1)
plt.gca().set_title('Variable Gender')
sns.countplot(x = 'Gender', hue = 'satisfaction', palette = 'Set2', data = train)

plt.subplot(2, 1, 2)
plt.gca().set_title('Variable Age')
sns.countplot(x = 'Age', hue = 'satisfaction', palette = 'Set2', data = train)

plt.figure(figsize = (20, 10))
plt.suptitle("Analysis Of Variable Satisfaction",fontweight="bold", fontsize=20)

plt.subplot(2, 1, 1)
plt.gca().set_title('Variable Class')
sns.countplot(x = 'Class', hue = 'satisfaction', palette = 'Set2', data = train)

plt.subplot(2, 1, 2)
plt.gca().set_title('Variable Type of Travel')
sns.countplot(x = 'Type of Travel', hue = 'satisfaction', palette = 'Set2', data =
train)

train['Distance Group'] = pd.cut(train['Flight Distance'], bins=8)

plt.figure(figsize=(20, 10))
plt.suptitle("Analysis Of Variable Satisfaction", fontweight="bold", fontsize=20)

plt.subplot(2, 1, 1)
plt.gca().set_title('Variable Flight Distances')
sns.countplot(x='Distance Group', hue='satisfaction', palette='Set2', data=train)

plt.show()

test = test.drop('id', axis = 1)

hot = pd.get_dummies(test[['Gender', 'Customer Type', 'Type of Travel', 'Class']])
test = pd.concat([test, hot], axis = 1)
test = test.drop(['Gender', 'Customer Type', 'Type of Travel', 'Class'], axis = 1)

X = test.drop('satisfaction', axis = 1)
X = X.values
y = test['satisfaction']

scaler = StandardScaler()
X_standard = scaler.fit_transform(X)

X_train,X_test, y_train, y_test = train_test_split(X_standard, y, test_size = 0.3, rand
om_state = 0)

import numpy as np
import pandas as pd
from sklearn.impute import SimpleImputer
```

```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Ваші дані
# X_train, X_test, y_train, y_test повинні бути визначені раніше

imputer = SimpleImputer(strategy='median')

# Обробка пропущених значень у тренувальних та тестових даних
X_train_imputed = imputer.fit_transform(X_train)
X_test_imputed = imputer.transform(X_test)

# Навчання моделі Naive Bayes
naive_bayes = GaussianNB()
naive_bayes.fit(X_train_imputed, y_train)
previsoes = naive_bayes.predict(X_test_imputed)

# Передбачення на тестовому наборі
test_predictions = naive_bayes.predict(X_test_imputed)

# Передбачення на тренувальному наборі
train_predictions = naive_bayes.predict(X_train_imputed)

# Обчислення точності для тренувального та тестового наборів
train_accuracy1 = accuracy_score(y_train, train_predictions)
test_accuracy1 = accuracy_score(y_test, test_predictions)

print("Train Accuracy:", train_accuracy1)
print("Test Accuracy:", test_accuracy1)

# Побудова матриці конфузії для тестового набору
cm = confusion_matrix(y_test, test_predictions)

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, cmap='Blues', fmt='g', cbar=False)
tick_labels = ['neutral or dissatisfied', 'satisfied']
plt.xticks(ticks=[0.5, 1.5], labels=tick_labels)
plt.yticks(ticks=[0.5, 1.5], labels=tick_labels)
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix for Naive Bayes')
plt.show()

classification_naive_gaussian = (classification_report(y_test, previsoes))
print(classification_naive_gaussian)

score_naive_gaussian = 0.8406262030026947

imputer = SimpleImputer(strategy='median')
X_train_imputed = imputer.fit_transform(X_train)
X_test_imputed = imputer.transform(X_test)

parameters = {'max_depth': [3, 4, 5, 6, 7, 9, 11],
              'min_samples_split': [2, 3, 4, 5, 6, 7],
              'criterion': ['entropy', 'gini']
             }

model = DecisionTreeClassifier()
```

```
gridDecisionTree = RandomizedSearchCV(model, parameters, cv=3, n_jobs=-1)
gridDecisionTree.fit(X_train_imputed, y_train)

print('Min Split: ', gridDecisionTree.best_estimator_.min_samples_split)
print('Max Depth: ', gridDecisionTree.best_estimator_.max_depth)
print('Criterion: ', gridDecisionTree.best_estimator_.criterion)
print('Score: ', gridDecisionTree.best_score_)

import numpy as np
import pandas as pd
from sklearn.impute import SimpleImputer
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Ваші дані (визначені раніше)
# X_train, X_test, y_train, y_test повинні бути визначені раніше

imputer = SimpleImputer(strategy='median')

# Обробка пропущених значень у тренувальних та тестових даних
X_train_imputed = imputer.fit_transform(X_train)
X_test_imputed = imputer.transform(X_test)

# Навчання моделі Decision Tree
decision_tree = DecisionTreeClassifier(criterion='entropy', min_samples_split=6, ma
x_depth=6, random_state=0)
decision_tree.fit(X_train_imputed, y_train)
previsoes = decision_tree.predict(X_test_imputed)

# Передбачення на тестовому наборі
test_predictions = decision_tree.predict(X_test_imputed)

# Передбачення на тренувальному наборі
train_predictions = decision_tree.predict(X_train_imputed)

# Обчислення точності для тренувального та тестового наборів
train_accuracy2 = accuracy_score(y_train, train_predictions)
test_accuracy2 = accuracy_score(y_test, test_predictions)

print("Train Accuracy:", train_accuracy2)
print("Test Accuracy:", test_accuracy2)

# Побудова матриці конфузії для тестового набору
cm_matrix = confusion_matrix(y_test, test_predictions)

plt.figure(figsize=(8, 6))
sns.heatmap(cm_matrix, annot=True, cmap='Greens', fmt='g', cbar=False)
tick_labels = ['neutral or dissatisfied', 'satisfied']
plt.xticks(ticks=[0.5, 1.5], labels=tick_labels)
plt.yticks(ticks=[0.5, 1.5], labels=tick_labels)
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix for Decision Tree')
plt.show()

classification_decision = (classification_report(y_test, previsoes))
print(classification_decision)
```

Додаток Г

(обов’язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ РІВНІВ ЗАДОВОЛЕНОСТІ ПАСАЖИРІВ АВІАКОМПАНІЯМИ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2024 р.

Вінниця 2024

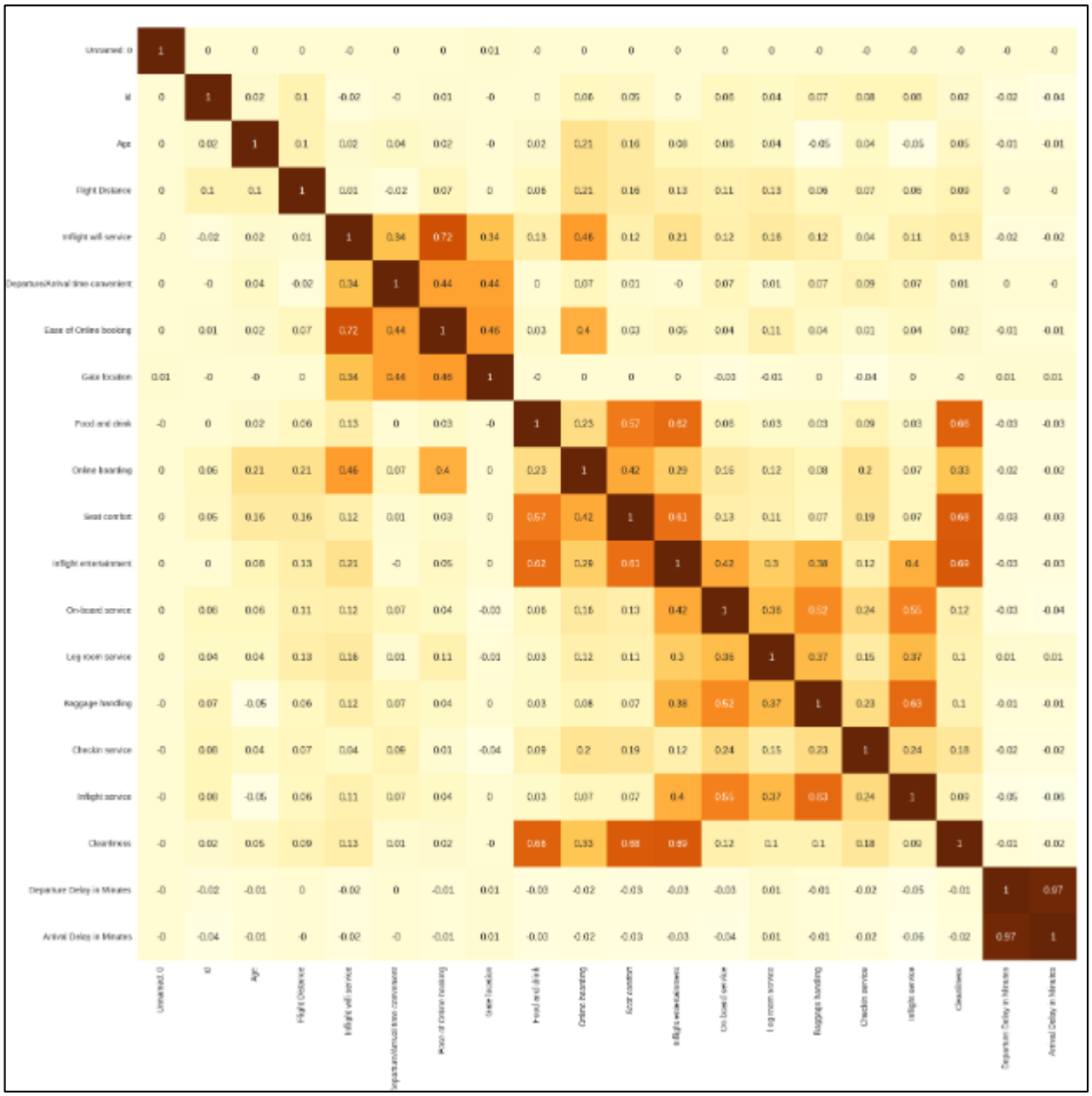

Рисунок Г.1 – Кореляційна матриця показників.

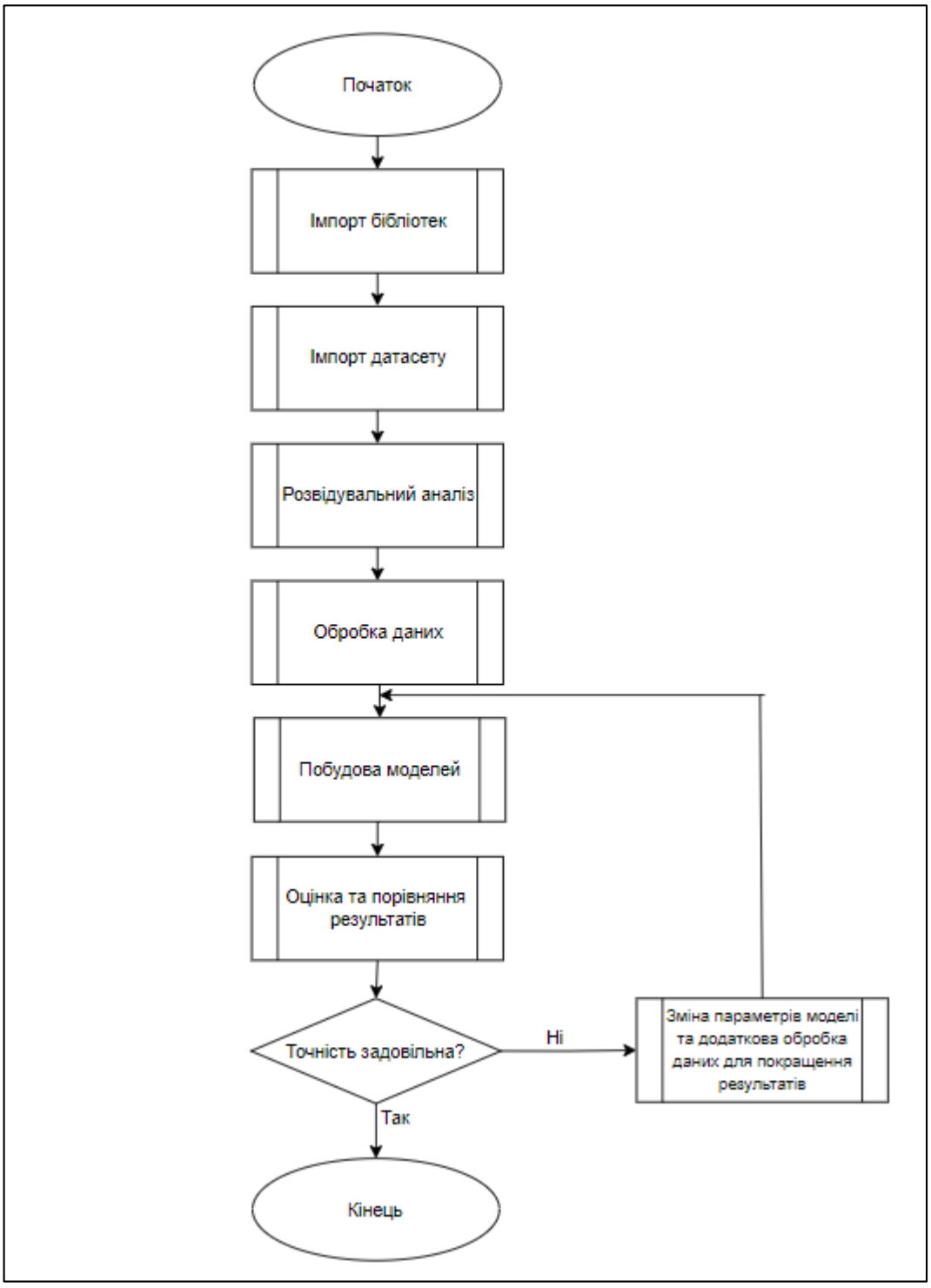

Рисунок Г.2 – Схема алгоритму

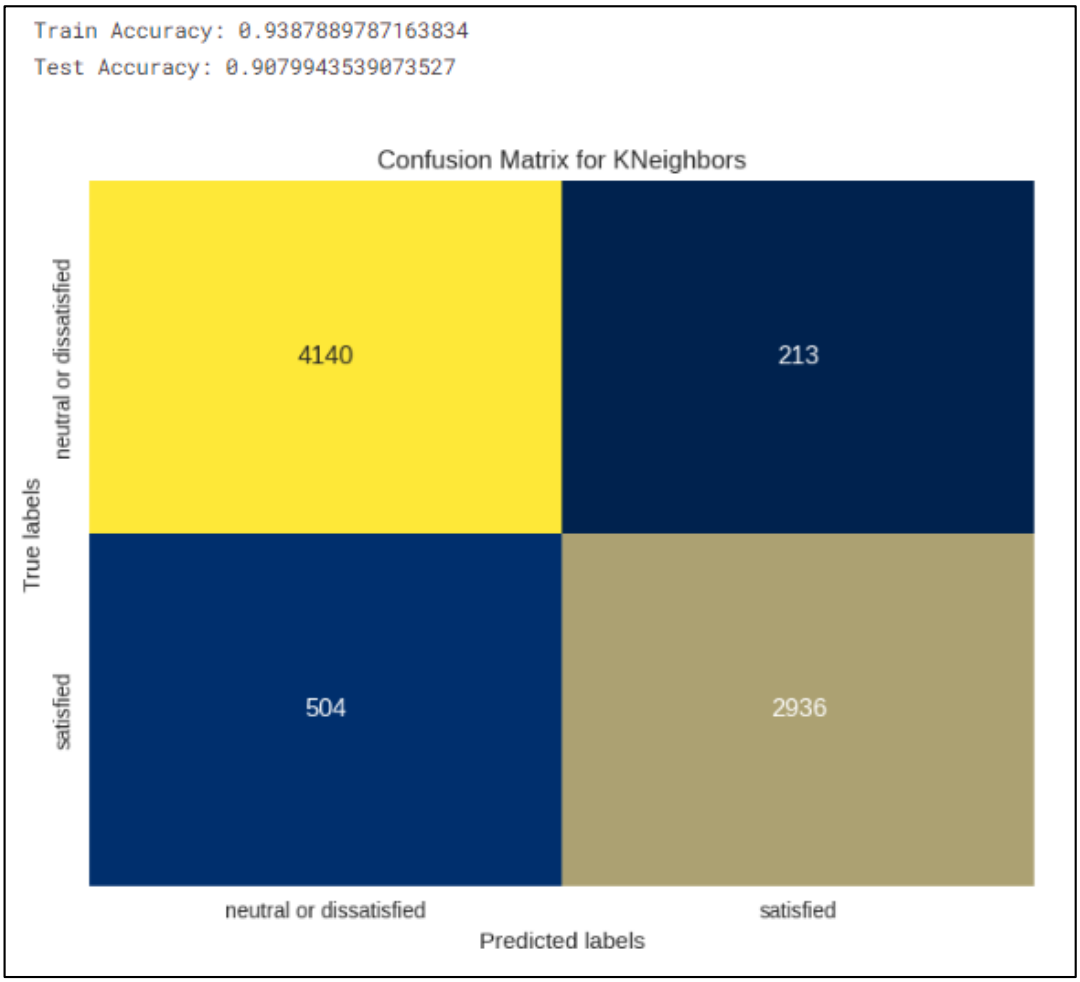

Рисунок Г.3 – Матриця плутанини моделі «KNeighbors».

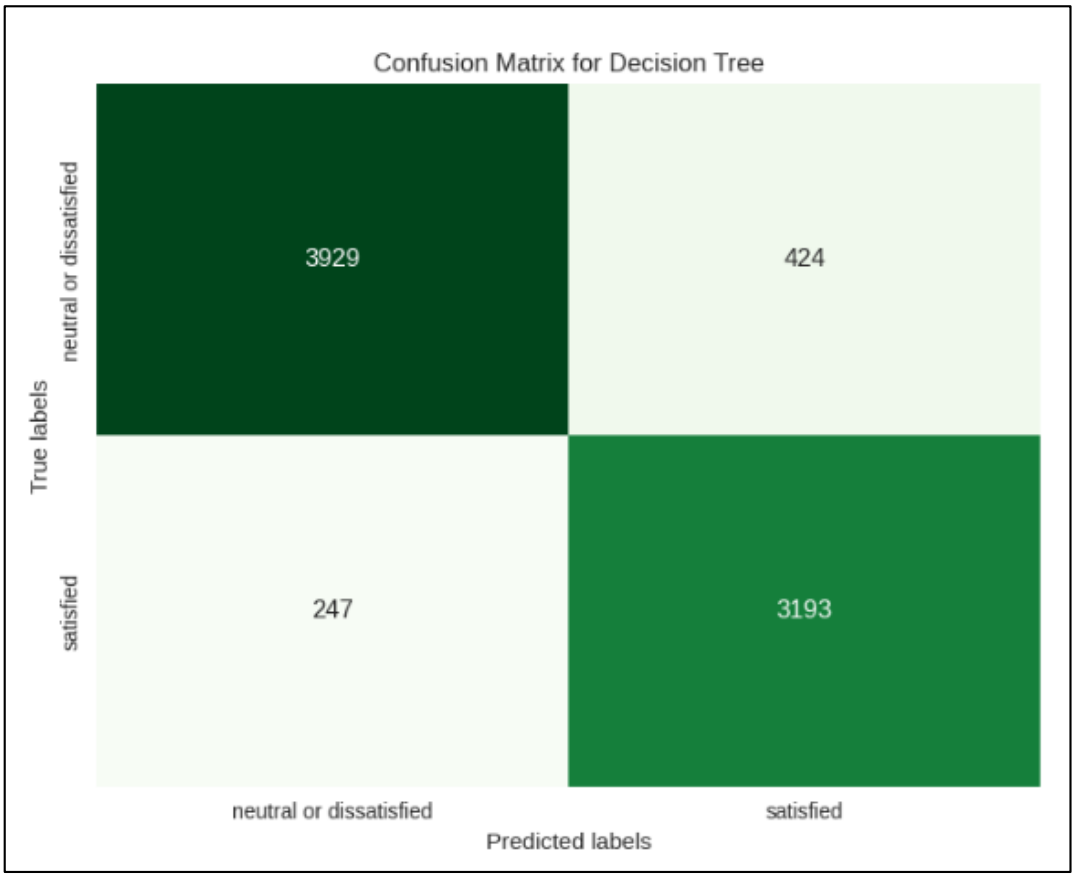

Рисунок Г.4 – Матриця плутанини моделі «Decision Tree».

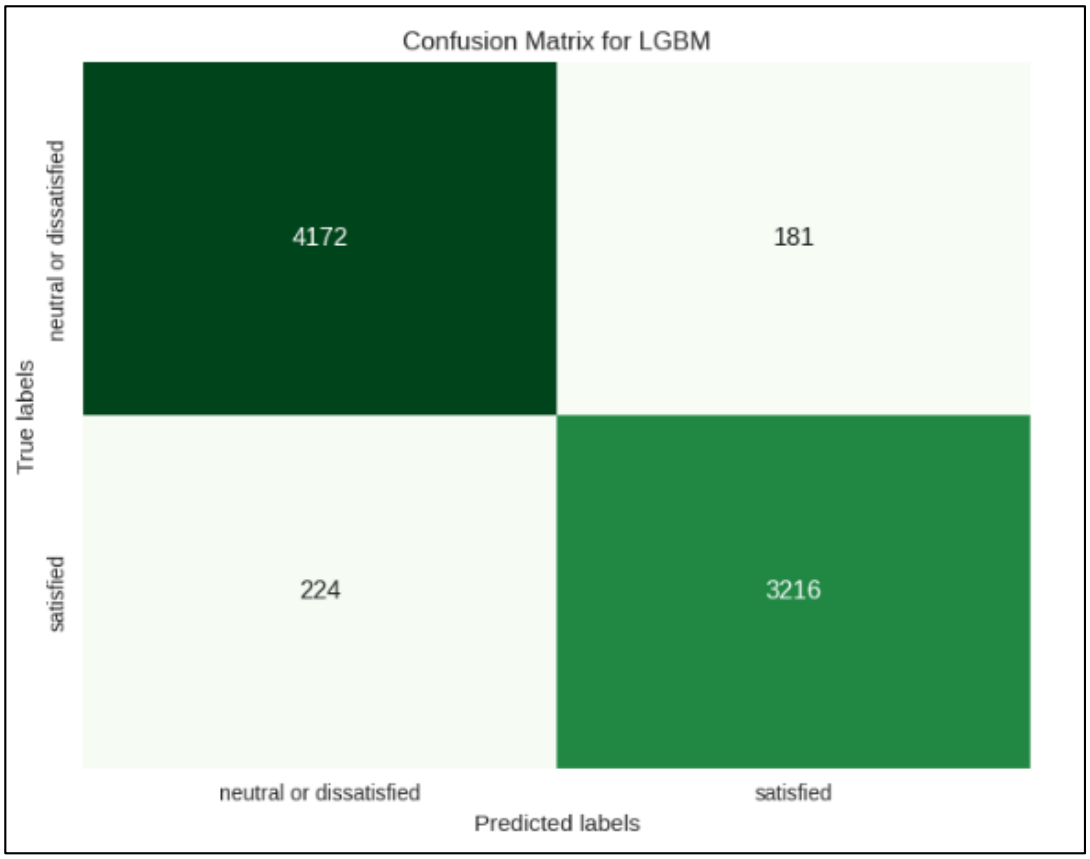

Рисунок Г.5 – Матриця плутанини моделі «LGBM».

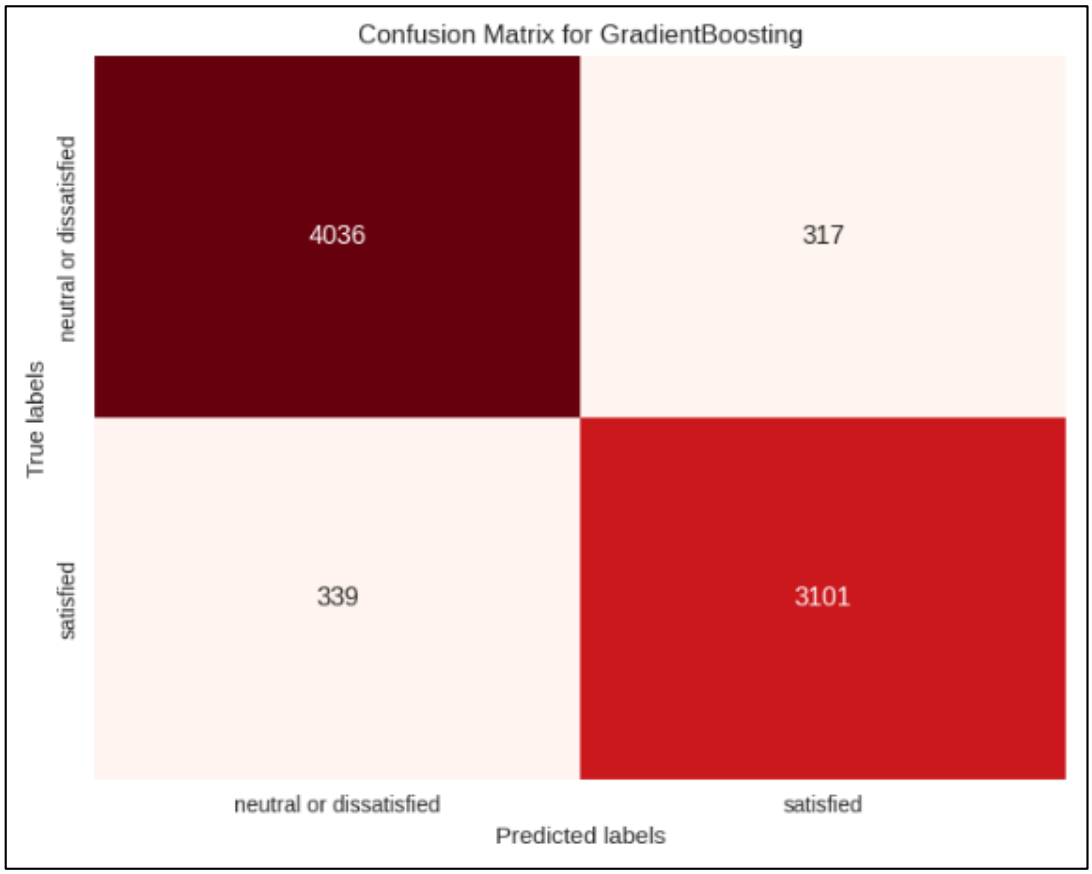

Рисунок Г.6 – Матриця плутанини моделі «GradientBoost».

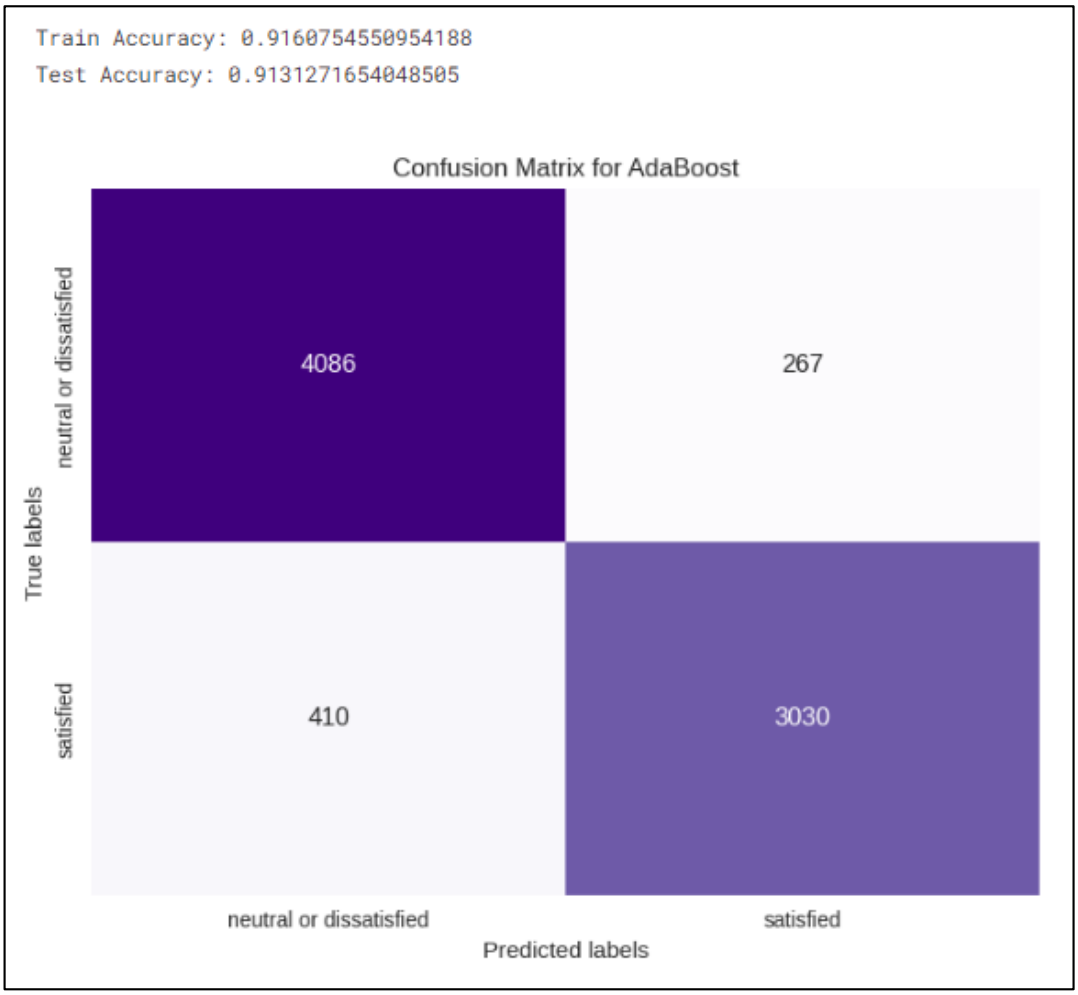

Рисунок Г.7 – Матриця плутанини моделі «AdaBoost».

	Model	Train Precision	Test Precision
0	Naive Bayes	0.843150	0.840626
1	Decision Tree	0.918550	0.913897
2	Random Forest	0.922290	0.916335
3	Extra Trees	0.926855	0.918003
4	KNN	0.938789	0.907994
5	Logistic Regression	0.872023	0.868215
6	AdaBoost	0.916075	0.913127
7	GradientBoost	0.924655	0.915822
8	LGBM	0.982951	0.948030

Рисунок Г.8 – Таблиця із загальними результатами.

	precision	recall	f1-score	support
neutral or dissatisfied	0.95	0.96	0.95	4353
satisfied	0.95	0.93	0.94	3440
accuracy			0.95	7793
macro avg	0.95	0.95	0.95	7793
weighted avg	0.95	0.95	0.95	7793

Рисунок Г.9 – Класифікаційний звіт моделі LGBM

	precision	recall	f1-score	support
neutral or dissatisfied	0.92	0.93	0.92	4353
satisfied	0.91	0.90	0.90	3440
accuracy			0.92	7793
macro avg	0.91	0.91	0.91	7793
weighted avg	0.92	0.92	0.92	7793

Рисунок Г.10 – Класифікаційний звіт моделі GradientBoost