Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

магістра
(назва освітньо-кваліфікаційного рівня)

студента *Фоміна Юрія Володимировича*
(ПІБ)

академічної групи *121м-22з-1*
(шифр)

спеціальності *121 Інженерія програмного забезпечення*
(код і назва спеціальності)

освітньої програми *«Інженерія програмного забезпечення»*
(назва освітньої програми)

на тему: *Розробка інформаційно-аналітичної системи прогнозування задоволеності пасажирів авіакомпанією на основі моделей машинного навчання*

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
розділів кваліфікаційної роботи				
спеціальний	*доц. Приходченко С.Д.*			
Рецензент				
Нормоконтролер	*доц. Гуліна І.Г.*			

Дніпро
2023

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:
Завідувач кафедри
Програмного забезпечення комп'ютерних систем
(повна назва)

_____ М.О. Алексєєв
(підпис) (прізвище, ініціали)

« » _____ 20 _____ року

ЗАВДАННЯ
на виконання кваліфікаційної роботи магістра

спеціальності *121 Інженерія програмного забезпечення*
(код і назва спеціальності)

студенту *121м-22з-1* *Фоміну Юрію Володимировичу*
(група) (прізвище та ініціали)

Тема кваліфікаційної роботи *Розробка інформаційно-аналітичної системи прогнозування задоволеності пасажирів авіакомпанією на основі моделей машинного навчання*

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора НТУ «Дніпровська політехніка» від

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень – пасажири авіакомпаній, представлені статистичними даними.

Предмет досліджень – методи обробки статистичних даних та моделі машинного навчання на основі статистичних даних для прогнозування задоволеності пасажирів авіакомпанією.

Мета НДР – розробка інформаційно-аналітичної системи прогнозування задоволеності пасажирів авіакомпанією з використанням методів та алгоритмів машинного навчання.

Вихідні данні для проведення роботи – статистичні дані задоволеності пасажирів авіакомпанією.

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна результатів кваліфікаційної роботи полягає в удосконаленні методів прогнозування задоволеності пасажирів авіакомпанією.

Практична цінність полягає в тому, що методи та моделі запропоновані в дослідженні, дозволяють авіакомпаніям передбачити задоволеність різних категорій пасажирів якістю оказаних послуг, що дає можливість вдосконалити та оптимізувати ці послуги, а також вжити маркетингові заходи з втримання окремих груп пасажирів.

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Результати досліджень мають бути подані у вигляді, що дозволяє побачити та оцінити якість удосконаленої моделі трансляції до українського мовлення.

5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	**Строки виконання робіт (початок – кінець)**
Аналіз теми та постановка задачі	11.09.2023 – 31.09.2023
Аналіз та підготовка статистичних даних до моделювання	01.10.2023 – 31.10.2023
Розробка інформаційно-аналітичної системи прогнозування задоволеності пасажирів авіакомпанією	01.11.2023 – 30.11.2023

Завдання видав _____ *Приходченко С.Д.*
(підпис) (прізвище, ініціали)

Завдання прийняв до виконання _____ *Фомін Ю.В.*
(підпис) (прізвище, ініціали)

Дата видачі завдання: *11.09.2023 р*

Термін подання кваліфікаційної роботи до ЕК: 20.12.2023 р.

РЕФЕРАТ

Пояснювальна записка: 106 сторінок, 55 рисунка, 1 таблиця, 1 додаток, 40 джерел.

Кваліфікаційна робота присвячена розробці інформаційно-аналітичної системи прогнозування задоволеності пасажирів авіакомпанією на основі моделей машинного навчання.

Об’єкт дослідження - пасажири авіакомпаній, представлені статистичними даними.

Предмет дослідження - методи обробки статистичних даних та моделі машинного навчання на основі статистичних даних для прогнозування задоволеності пасажирів авіакомпанією.

Мета роботи - розробка інформаційно-аналітичної системи прогнозування задоволеності пасажирів авіакомпанією з використанням методів та алгоритмів машинного навчання.

Для вирішення поставлених задач використані математичні та статистичні методи машинного навчання, об'єктно-орієнтоване програмування.

Наукова новизна результатів кваліфікаційної роботи полягає в удосконаленні методів прогнозування задоволеності пасажирів авіакомпанією.

Практична цінність полягає в тому, що методи та моделі запропоновані в дослідженні, дозволяють авіакомпаніям передбачити задоволеність різних категорій пасажирів якістю оказаних послуг, що дає можливість вдосконалити та оптимізувати ці послуги, а також вжити маркетингові заходи з втримання окремих груп пасажирів.

Область застосування: розроблена інформаційно-аналітична система може бути використана авіакомпаніями для передбачення задоволеності пасажирів якістю оказаних послуг, а також для аналізу сприйняття якості послуг різними категоріями пасажирів.

Значення роботи та висновки: досліджено методи та алгоритми класифікації машинного навчання, створена інформаційно-аналітична система прогнозування задоволеності пасажирів авіакомпанією, використання якої дозволить авіакомпаніям збільшити кількість задоволених пасажирів шляхом вдосконалення та оптимізації своїх послуг.

Прогнози щодо розвитку досліджень: дослідити методи оптимізації алгоритмів машинного навчання для збільшення точності прогнозування задоволеності пасажирів авіакомпанією.

У роботі розглядається процес аналітичного прогнозування задоволеності пасажирів авіакомпанією на основі методів та алгоритмів машинного навчання. Особлива увага приділяється підготовці даних та створенню моделей прогнозування. Наведено огляд сучасних моделей машинного навчання та методику їх побудови, процес підготовки та аналіз даних щодо задоволеності пасажирів авіакомпанією, створення моделей на основі цих даних. Запропоновано найефективнішу модель машинного навчання для побудови на її основі системи прийняття рішень.

Список ключових слів: МАШИННЕ НАВЧАННЯ, ЛОГІСТИЧНА РЕГРЕСІЯ, ДИСКРИМІНАЦІЙНИЙ АНАЛІЗ, МЕТОД ОПОРНИХ ВЕКТОРІВ, ДЕРЕВА РІШЕНЬ, МЕТОД НАЙБЛИЖЧИХ СУСІДІВ, НАЇВНИЙ БАЄСІВ КЛАСИФІКАТОР, АНСАМБЛЕВІ МЕТОДИ МАШИННОГО НАВЧАННЯ, ШТУЧНІ НЕЙРОННІ МЕРЕЖІ, CROSS-VALIDATION, ACCURACY, PRECISION, RECALL, F1-SCORE, RANDOMFORESTCLASSIFIER, LGBMCLASSIFIER, XGBCLASSIFIER.

ABSTRACT

Explanatory note: 106 pages, 55 figures, 1 table, 1 appendix, 40 sources.

The thesis is devoted to the development of an information-analytical system for predicting passenger satisfaction with the airline based on machine learning models.

The object of the research is airline passengers, represented by statistical data.

The subject of the research is statistical data processing methods and statistical data-based machine learning models for predicting passenger satisfaction with the airline.

The purpose of the research is to develop an information-analytical system for predicting passenger satisfaction with the airline using machine learning methods and algorithms.

Mathematical and statistical methods of machine learning, object-oriented programming were used to solve the problems.

The scientific originality of the master's thesis lies in the improvement of methods of forecasting passenger satisfaction with the airline.

The practical value is that the methods and models proposed in the research allow airlines to predict the satisfaction of different categories of passengers with the quality of the services provided, which makes it possible to improve and optimize these services as well as take marketing measures to retain certain groups of passengers.

Scope of application: the developed information and analytical system can be used by airlines to predict passenger satisfaction with the quality of services provided, as well as to analyze the perception of service quality by different categories of passengers.

The value of the work and conclusions: machine learning classification methods and algorithms were investigated, an information and analytical system for predicting passenger satisfaction with the airline was created, the use of which will allow airlines to increase the number of satisfied passengers by improving and optimizing their services.

Research development predictions: explore methods for optimizing machine learning algorithms to increase the accuracy of predicting passenger satisfaction with an airline.

The master's thesis considers the process of analytical predicting of passenger satisfaction with the airline based on machine learning methods and algorithms. Special attention is paid to the preparation of data and the creation of predicting models. A research of modern machine learning models and the methodology of their construction, the process of preparation and analysis of data on passenger satisfaction with the airline, and the creation of models based on these data are provided. The most effective model of machine learning is proposed for building a decision-making system based on it.

List of keywords: MACHINE LEARNING, LOGISTIC REGRESSION, DISCRIMINANT ANALYSIS, SUPPORT VECTOR MACHINE, DECISION

TREE, K-NEAREST NEIGHBOR, NAIVE BAYES CLASSIFIER, ENSEMBLE MACHINE LEARNING METHODS, ARTIFICIAL NEURAL NETWORK, CROSS-VALIDATION, ACCURACY, PRECISION, RECALL, F1-SCORE, RANDOMFORESTCLASSIFIER, LGBMCLASSIFIER, XGBCLASSIFIER.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ANNs - Artificial Neural Networks;

CNN - Convolutional Neural Networks;

CRM - Customer Relationship Management;

DT - Decision Tree;

GANs - Generative Adversarial Networks;

GNNs - Graph Neural Networks;

KNN - K-Nearest Neighbor;

LDA - Linear Discriminant Analysis;

LR - Linear Regression;

MLP - Multi-Layer Perceptrons;

QDA - Quadratic Discriminant Analysis;

RDA - Regularized Discriminant Analysis;

RNN - Recurrent Neural Networks;

SLP - Single-Layer Perceptrons;

SVM - Support Vector Machines.

ЗМІСТ

ВСТУП

В сучасному світі, комерційні, та некомерційні організації, під час своєї діяльності збирають великі обсяги даних, кількість яких з кожним роком збільшується. Разом з цим, збільшується і потреба в аналізі цих даних, задля того, щоб вони приносили якомога більшу користь, з використанням результатів цього аналізу для прийняття рішень. Отримання нової інформації з наявних даних є задачею аналізу даних. У цій роботі увага зосереджена на аналітичному прогнозуванні, яке є важливим різновидом аналізу даних.

Аналітичне прогнозування являє собою процес створення та використання моделей, які передбачають події на основі наявних даних. Модель застосовується для прогнозування з метою надання інформаційної підтримки користувачу при прийнятті рішень. У повсякденному житті під прогнозом ми розуміємо передбачення того, що може або не може статися у майбутньому. У контексті аналітичного прогнозування термін "прогноз" означає присвоєння значення будь-якій невідомій змінній, як, наприклад, прогноз ціни товару, або вірогідності настання тієї чи іншої події. Прогностична модель навчається задля того, щоб створювати прогнози на підставі набору статистичних даних, а для навчання такої моделі, ми використовуємо методи машинного навчання.

Машинне навчання у практичному контексті являє собою автоматизований процес, який виконує функцію отримання шаблонів із даних. Для створення моделей в аналітичному прогнозуванні при виконанні задачі класифікації, використовуються методи машинного навчання з учителем, завдяки яким створені моделі, на основі зібраних даних, автоматично визначають взаємозв'язки між набором описових ознак та цільовою ознакою, та можуть бути використані для прогнозування.

Використання моделей машинного навчання є поширеною практикою у прогнозуванні поведінки споживачів, адже отримуємі завдяки розробленим інформаційно-аналітичним системам дані є основою для прийняття рішень з

вдосконалення та оптимізації діяльності організації задля досягнення поставлених цілей.

Метою дослідження є розробка інформаційно-аналітичної системи прогнозування задоволеності пасажирів авіакомпанією з використанням методів та алгоритмів машинного навчання.

Об’єктом дослідження є пасажири авіакомпаній, представлені статистичними даними.

Предметом дослідження є методи обробки статистичних даних та моделі машинного навчання на основі статистичних даних для прогнозування задоволеності пасажирів авіакомпанією.

Для вирішення поставлених задач використані математичні та статистичні методи машинного навчання, об'єктно-орієнтоване програмування.

Наукова новизна результатів кваліфікаційної роботи полягає в удосконаленні методів прогнозування задоволеності пасажирів авіакомпанією.

Практичне значення результатів кваліфікаційної роботи полягає в тому, що методи та моделі запропоновані в дослідженні, дозволяють авіакомпаніям передбачити задоволеність різних категорій пасажирів якістю наданих послуг, що дає можливість вдосконалити та оптимізувати ці послуги, а також вжити необхідні маркетингові заходи з втримання окремих груп пасажирів, як, наприклад, спеціальні пропозиції та акції.

Особистий внесок автора полягає в удосконаленні методів прогнозування задоволеності пасажирів авіакомпанією.

Тематика першого розділу присвячена огляду питань застосування методів машинного навчання щодо прогнозування задоволеності споживачів, огляду та аналізу даних задоволеності пасажирів авіакомпанією, а також підготовці їх до моделювання.

У другому розділі розглянуті основні методи та алгоритми машинного навчання які можуть бути застосовані до виконання задачі класифікації.

У третьому розділі проведено моделювання з метою прогнозування задоволеності пасажирів авіакомпанією на основі статистичних даних, аналіз та вдосконалення моделей машинного навчання.

РОЗДІЛ 1. ОГЛЯД ПИТАННЯ ПРОГНОЗУВАННЯ ЗАДОВОЛЕНОСТІ ПАСАЖИРІВ АВІАКОМПАНІЄЮ ЗА ДОПОМОГОЮ МЕТОДІВ МАШИННОГО НАВЧАННЯ

1.1 Застосування методів машинного навчання щодо прогнозування задоволеності споживачів

Вивчення та аналіз задоволеності споживачів є однією з найголовніших складових аналітичної функції маркетингу. Важливість цієї діяльності обумовлена тим що саме знання про сприйняття та відношення споживача до тих чи інших товарів або послуг, з одного боку вказує компаніям напрямки зміни та вдосконалення цих послуг, з іншого боку дозволяє вживати заходів з втримання окремих груп споживачів, як, наприклад, спеціальні пропозиції та акції.

Аналіз та прогнозування компанією задоволеності споживачів є частиною концепції CRM (Customer Relationship Management), реалізація якої передбачає:

- створення та оновлення бази даних споживачів;
- розробку підходів до аналізу бази даних;
- вибір методів та форм аналізу бази даних;
- прогнозування на основі бази даних;
- перевірку ефективності обраних методів аналізу та прогнозування;
- використання отриманих результатів для розвитку, зміні та оптимізації діяльності компанії щодо задоволеності споживачів.

Так, маючи достатню аналітичну інформацію про клієнта, компанія отримує можливість вчасно вжити ряд заходів як для запобігання втрати певних груп клієнтів, так і задля збільшення кількості клієнтів в цілому. Як, наприклад, запровадження системи бонусів, створення або вдосконалення програми лояльності, проведення цільових акцій тощо, що своєю чергою, призводить до зростання фінансових ресурсів компанії та її розвитку.

Одна з типових задач, що виникають в процесі передбачення поведінки споживача, є реалізація бінарної класифікації, а саме визначення належності споживача до одного з двох класів: задоволених або незадоволених. Застосування методів машинного навчання спроможне ефективно вирішити цю задачу. Оскільки передбачення задоволеності споживачів є однією з найбільш актуальних проблем в CRM, була обрана саме ця область для дослідження ефективності методів та алгоритмів машинного навчання.

1.2 Огляд та підготовка даних задоволеності пасажирів авіакомпанією

Процес прогнозування можна представити у вигляді трьох стадій, кожна з яких завершується отриманням цільовх для кожної стадії результатів:

- попередня обробка та підготовка даних;
- моделювання;
- аналіз та оцінка отриманих результатів.

У даному розділі основна увага приділяється початковому етапу аналітичного прогнозування, який полягає у попередній обробці та підготовці даних;

Для кращого розуміння проблематики проведемо короткий огляд статистичних даних задоволеності пасажирів авіарейсів авіакомпанією що надала їм послуги, на яких будемо тренувати моделі машинного навчання.

Цей набір даних складається з 103904 рядків, кожен з яких, окрім першого, містить інформацію щодо пасажира авіарейсу прошедшего опитування, та 25 стовпців з окремими ознаками які їх характеризують, а саме такі параметри як вік, стать, клас яким був здійснений авіарейс, відстань, затримка рейсу тощо. Дані були отримані з джерела [4].

1.2.1 Аналітичний цикл обробки даних щодо задоволеності пасажирів авіакомпанією.

Одним із ключових факторів при застосуванні алгоритмів машинного навчання є використання належних даних для навчання моделі. Навіть коли дані є змістовним та містять велику кількість корисної інформації, їх необхідно попередньо дослідити, щоб переконатись що інформація яка буде використовуватись, має відповідний формат та відповідає вимогам необхідним для навчання та тестування моделі.

Етап підготовки даних становить перший та найбільш часомісткий етап аналітичного прогнозування. Отримані дані можуть містити пропуски, помилки, шуми, дублікати, а також неузгодженість у типах та форматах. Обсяг зібраних даних може бути недостатнім або, навпаки, занадто великим для ефективного аналізу, особливо якщо частина інформації не впливає значущим чином на прогнозований результат. Ці недоліки слід усунути перед переходом до етапу моделювання, щоб вони не впливали на функціональність моделі. Крім того, підготовка даних також може включати в себе нормалізацію даних та розбиття на тестову і робочу вибірки. [5]

Для роботи з даними використаємо мову програмування Python та інтерактивне обчислювальне середовище Jupyter Notebook.

Опрацюємо таблиці з даними за допомогою бібліотеки pandas. Почнемо з загального огляду таблиці даних (рис. 1.1):

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	Food and drink	Online boarding	Seat comfort	Inflight entertainment
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	3	1	5	3	5	5
1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	2	3	3	1	3	1	1
2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	2	2	2	5	5	5	5
3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	5	5	5	2	2	2	2
4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3	3	3	3	4	5	5	3

Рисунок 1.1. Таблиця із початковими даними

Таблиця має 25 стовпця і близько 103904 записів. Оскільки кількість стовпців відносно велика та не зручна для візуального відображення у вигляді таблиці, розглянемо їх окремо (рис. 1.2):

```
Index(['Unnamed: 0', 'id', 'Gender', 'Customer Type', 'Age', 'Type of Travel',
       'Class', 'Flight Distance', 'Inflight wifi service',
       'Departure/Arrival time convenient', 'Ease of Online booking',
       'Gate location', 'Food and drink', 'Online boarding', 'Seat comfort',
       'Inflight entertainment', 'On-board service', 'Leg room service',
       'Baggage handling', 'Checkin service', 'Inflight service',
       'Cleanliness', 'Departure Delay in Minutes', 'Arrival Delay in Minutes',
       'satisfaction'],
      dtype='object')
```

Рисунок 1.2. Перелік ознак набору даних.

Опис ознак набору даних:

1. id: унікальний ідентифікатор опитаного;

2. Gender: стать опитаного (можливі значення ознаки: Male, Female);

3. Customer Type: лояльність опитаного до авіакомпанії (можливі значення ознаки: Loyal Customer, disloyal Customer);

4. Age: вік опитаного;

5. Type of Travel: тип подорожі (можливі значення ознаки: Personal Travel, Business travel);

6. Class: клас місця у салоні літака (можливі значення ознаки: Eco, Eco Plus, Business);

7. Flight Distance: відстань польоту (можливі значення ознаки: оцінка від 1 до 5);

8. Inflight wifi service: сервіс Wi-Fi на борту (можливі значення ознаки: оцінка від 1 до 5);

9. Departure/Arrival time convenient: зручність часу відправлення/прибуття (можливі значення ознаки: оцінка від 1 до 5);

10. Ease of Online booking: легкість онлайн-бронювання (можливі значення ознаки: оцінка від 1 до 5);

11. Gate location: розташування термінала посадки (можливі значення ознаки: оцінка від 1 до 5);

12. Food and drink: їжа та напої (можливі значення ознаки: оцінка від 1 до 5);

13. Online boarding: онлайн бронювання (можливі значення ознаки: оцінка від 1 до 5);

14. Seat comfort: комфорт сидіння (можливі значення ознаки: оцінка від 1 до 5);

15. Inflight entertainment: розваги на борту (можливі значення ознаки: оцінка від 1 до 5);

16. On-board service: обслуговування на борту (можливі значення ознаки: оцінка від 1 до 5);

17. Leg room service: місце для ніг (можливі значення ознаки: оцінка від 1 до 5);

18. Baggage handling: поводження з багажем (можливі значення ознаки: оцінка від 1 до 5);

19. Checkin service: сервіс реєстрації (можливі значення ознаки: оцінка від 1 до 5);

20. Inflight service: сервіс на борту (можливі значення ознаки: оцінка від 1 до 5);

21. Cleanliness: чистота (можливі значення ознаки: оцінка від 1 до 5);

22. Departure Delay in Minutes: затримка відправлення у хвилинах (можливі значення ознаки: оцінка від 1 до 5);

23. Arrival Delay in Minutes: затримка прибуття у хвилинах (можливі значення ознаки: оцінка від 1 до 5);

24. satisfaction: задоволеність (можливі значення ознаки: satisfied, neutral or dissatisfied).

Перевіривши типи даних у стовпцях, як показано на рис. 1.3, бачимо наявність таких типів даних, як int64 (ціле число), float64 (число з плаваючою точкою), object (об'єкт).

Перевіримо наявність відсутніх значень ознак набору даних, як показано на рис. 1.4.

Unnamed: 0	int64
id	int64
Gender	object
Customer Type	object
Age	int64
Type of Travel	object
Class	object
Flight Distance	int64
Inflight wifi service	int64
Departure/Arrival time convenient	int64
Ease of Online booking	int64
Gate location	int64
Food and drink	int64
Online boarding	int64
Seat comfort	int64
Inflight entertainment	int64
On-board service	int64
Leg room service	int64
Baggage handling	int64
Checkin service	int64
Inflight service	int64
Cleanliness	int64
Departure Delay in Minutes	int64
Arrival Delay in Minutes	float64
satisfaction	object

Рисунок 1.3. Типи данних набору даних

Unnamed: 0	0
id	0
Gender	0
Customer Type	0
Age	0
Type of Travel	0
Class	0
Flight Distance	0
Inflight wifi service	0
Departure/Arrival time convenient	0
Ease of Online booking	0
Gate location	0
Food and drink	0
Online boarding	0
Seat comfort	0
Inflight entertainment	0
On-board service	0
Leg room service	0
Baggage handling	0
Checkin service	0
Inflight service	0
Cleanliness	0
Departure Delay in Minutes	0
Arrival Delay in Minutes	310
satisfaction	0

Рисунок 1.4 – Перевірка наявності відсутніх значень ознак, отримані за допомогою команди df.isnull().sum()

Як бачимо, ознака "Arrival Delay in Minutes", має 310 відсутніх значення, заповнимо їх нулем використовуючи метод fillna().

Наступним кроком видалимо колонку "id" та "Unnamed: 0", які непотрібні для аналізу даних та тренування моделей.

Перевірка на дублікати за допомогою методу duplicated() бібліотеки pandas показала відсутність дублікатів.

Текст коду Python із дослідженням якості набору даних наведений у додатку А.

1.2.2 Основні характеристики даних задоволеності пасажирів авіакомпанією

Наступним кроком перейдемо до етапу дослідження даних. Використаємо графічну бібліотеку Python Plotly для візуалізації даних та проведемо їх аналіз.

На наведеній нижче діаграмі (рис. 1.5) представлене процентне співвідношення кількості опитаних за цільовою ознакою "satisfaction", яка має два значення, - "satisfied", тобто пасажир задоволений та "neutral or dissatisfied", тобто пасажир має нейтральне відношення або незадоволений. Як бачимо, більшість опитаних, а саме 56,7%, нейтральні або незадоволені авіакомпанією, у той час, як задоволеними залишились 43,3% пасажирів.

Задля подальшого якісного аналізу, розглянемо кожну ознаку набору даних, а саме долю кожного значення ознаки у загальній сукупності значень ознаки, та кількість задоволених та нейтральних або незадоволених осіб по кожному значенню ознаки, щоб зрозуміти яке зі значень має більший вплив на цільову змінну у порівнянні з іншими.

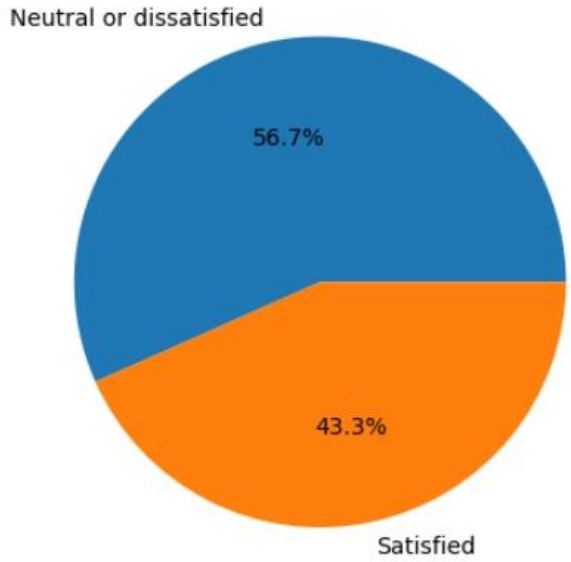


Рисунок 1.5. Діаграма співвідношення кількості опитаних за цільовою ознакою

Як бачимо на рис 1.6, розподілення опитаних за гендерною приналежністю є рівномірним, доля чоловіків у загальній сукупності складає 50,7%, а жінок 49,3%. Діаграма співвідношення значень відносно цільової змінної вказує, що доля нейтральних або незадоволених жінок трохи більша ніж доля нейтральних або незадоволених чоловіків, навіть враховуючи те, що жінок менше ніж чоловіків серед опитаних. Але виявлена особливість суттєво не впливає на цільову змінну за рахунок майже рівного розподілу чисельності жінок та чоловіків у загальній сукупності.

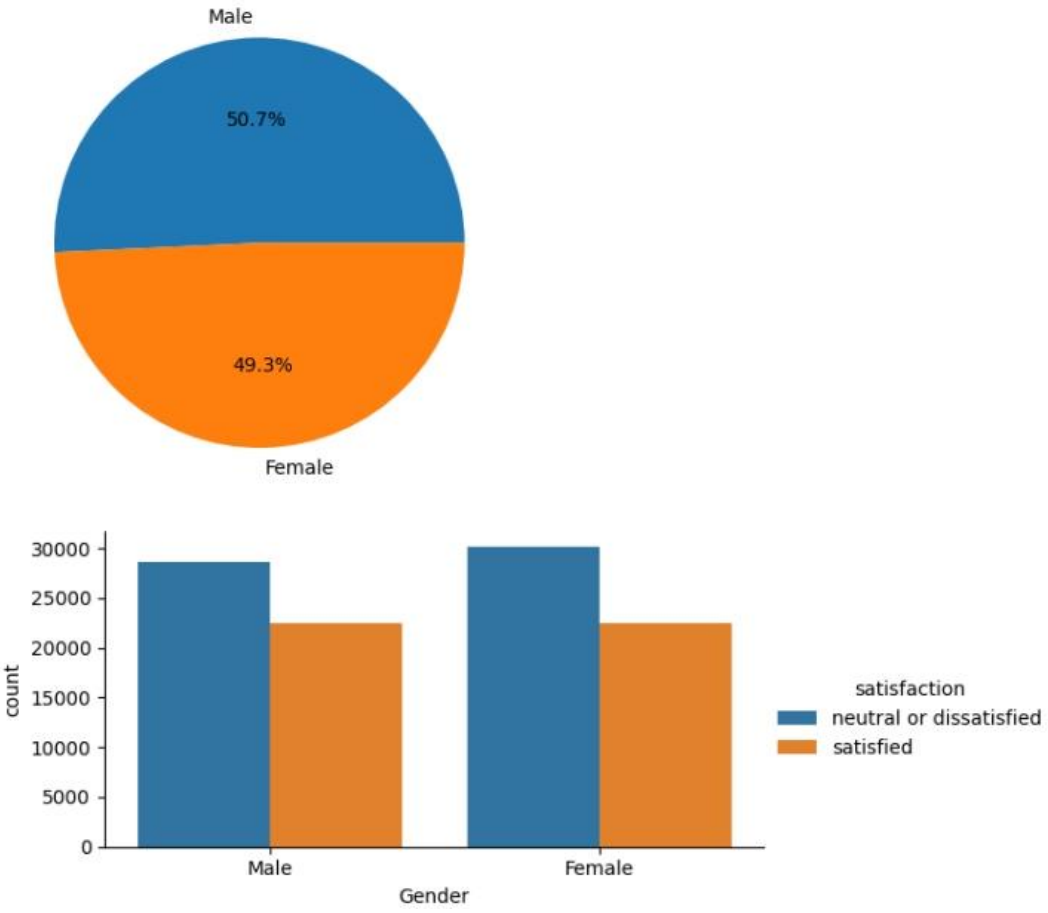


Рисунок 1.6. Діаграми розподілу значень за гендерною приналежністю.

Наступним кроком розглянемо розподіл пасажирів за ознакою Customer Type, тобто за типом пасажира, значення якої має два значення, - "Loyal Customer" та "disloyal Customer" (рис. 1.7). На круговій діаграмі розподілу бачимо, що частка лояльних пасажирів у загальній сукупності переважна більшість та складає 81,7%, з часткою нелояльних пасажирів 18,3%. Кількість незадоволених серед нелояльних пасажирів приблизно в три рази більша ніж задоволених, у той час, як серед лояльних пасажирів частки незадоволених та задоволених пасажирів відносно рівні, з невеликою перевагою незадоволених. При цьому, ступінь впливу відмінності задоволеності пасажирів за ознакою Customer Type на цільову змінну, значно зменшує переважна більшість лояльних пасажирів у загальній сукупності опитаних.

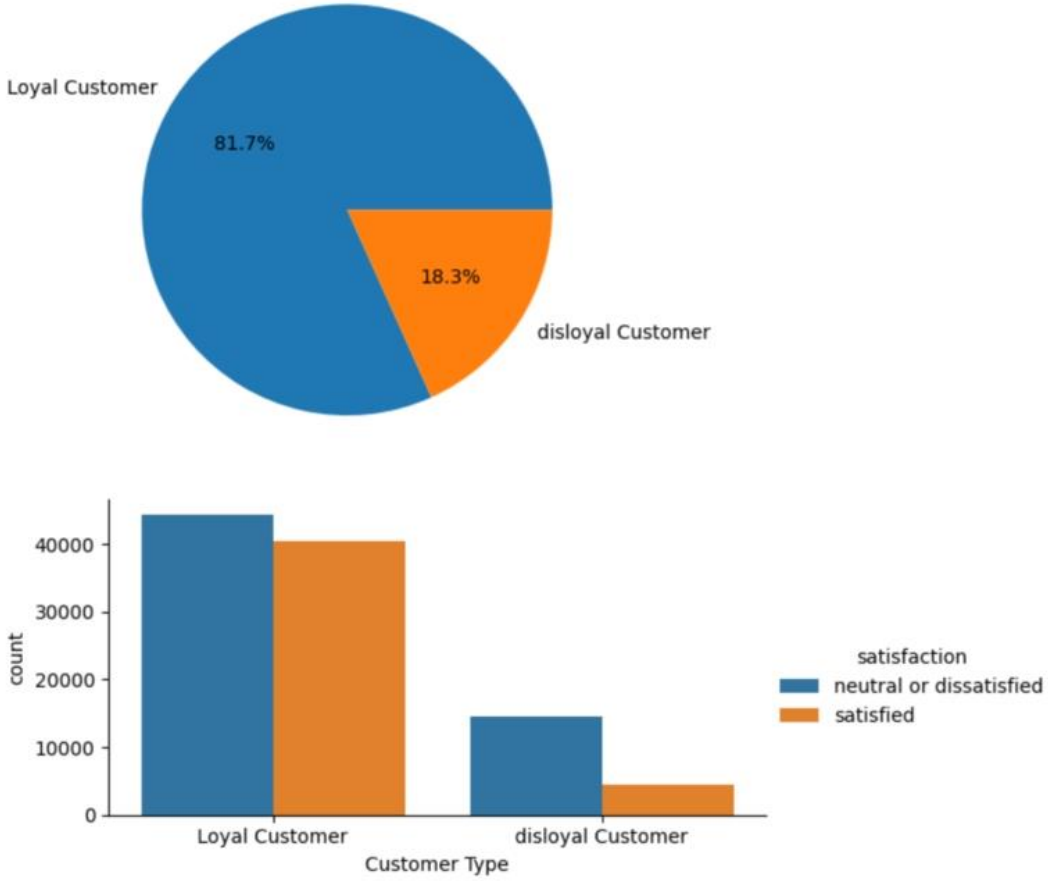


Рисунок 1.7. Діаграми розподілу значень ознаки "Customer Type".

Розглядаючи вік опитаних, бачимо що переважна більшість пасажирів знаходиться у віковому проміжку між 19 та 61 роком (рис. 1.8). Співвідношення задоволених та нейтральних або незадоволених серед осіб цього вікового проміжку різна. Так, бачимо, що серед пасажирів віком від 19 до 39 років переважає кількість нейтральних або незадоволених, у той час, як серед пасажирів віком від 40 до 61 років, навпаки. Серед пасажирів займаючих за віком меншу частину у загальній кількості опитаних, а саме від 7 до 18 років та старше 60 років, значно переважає кількість нейтральних або незадоволених осіб.

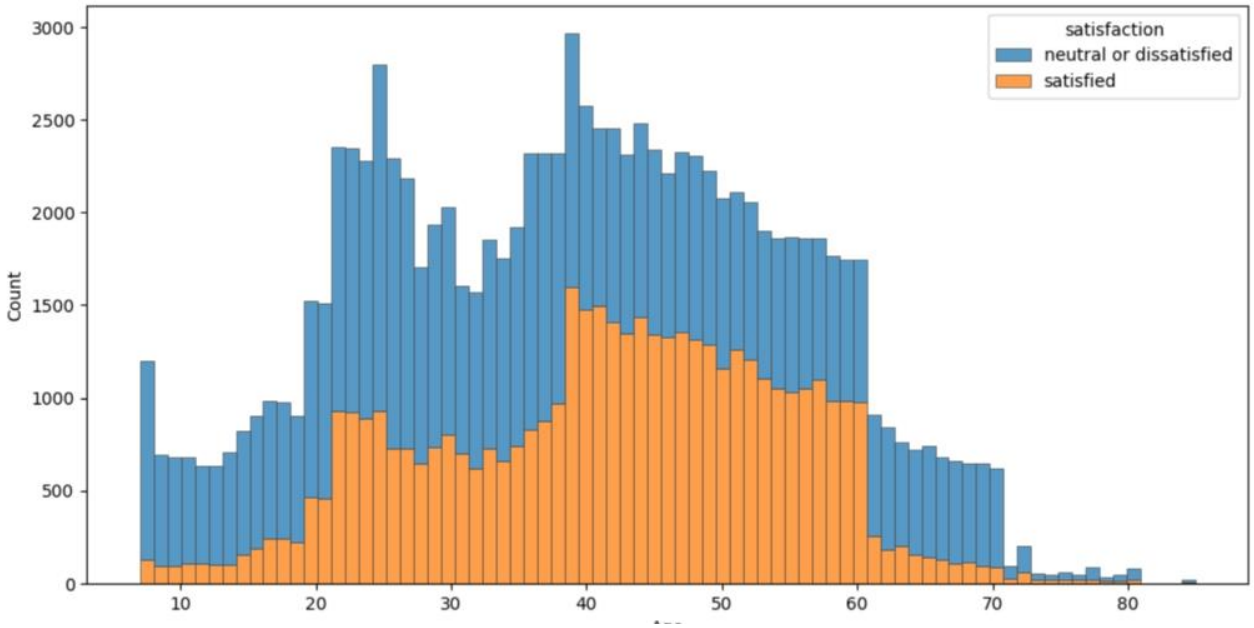


Рисунок 1.8. Діаграма розподілу опитаних за віком.

Розглядаючи ознаку "Type of Travel" (рис. 1.9), тобто тип подорожі, бачимо що більшість пасажирів подорожували в особистих справах - 69%, у той час, як частка опитаних що подорожувало по роботі складає 31%. При цьому кількість нейтральних або незадоволених опитаних серед тих хто подорожував з особистих справ приблизно в 7 разів більша ніж задоволених. У той час як серед пасажирів що подорожували у справах роботи переважає кількість задоволених опитаних.

Розглядаючи ознаку "Class" (рис. 1.10), бачимо, що більшість опитаних подорожували класами "Eco" та "Business" зі значеннями 47,8% та 45% відповідно. Переважна кількість опитаних які подорожували класом "Eco" нейтральні або незадоволені у той час, як більше ніж половина подорожуючих класом "Business" задоволені.

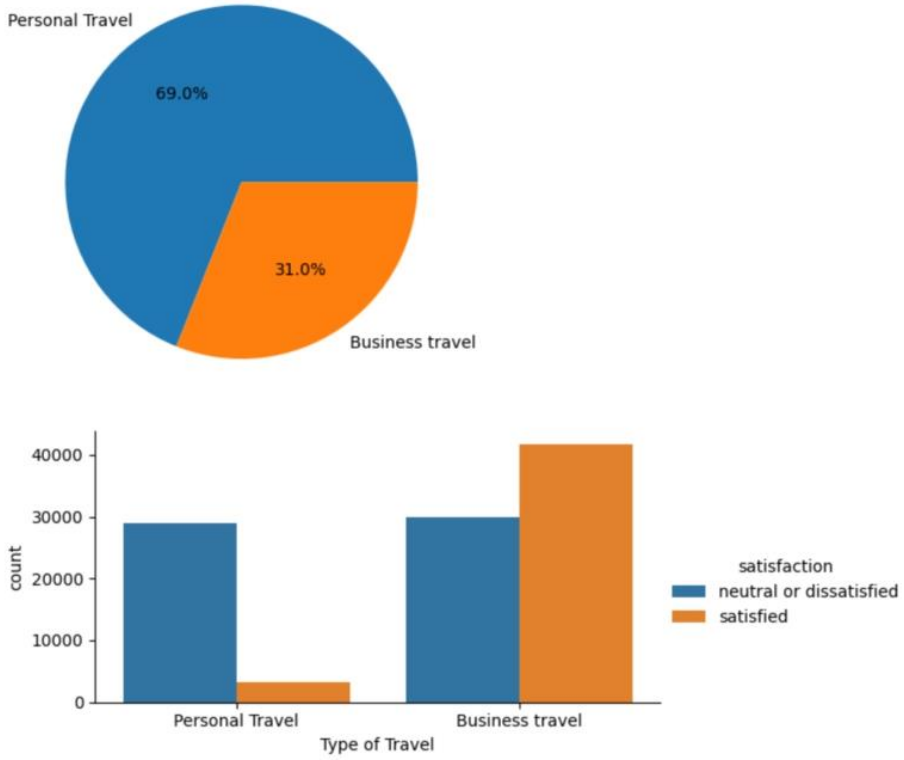


Рисунок 1.9. Діаграми розподілу значень ознаки "Type of Travel".

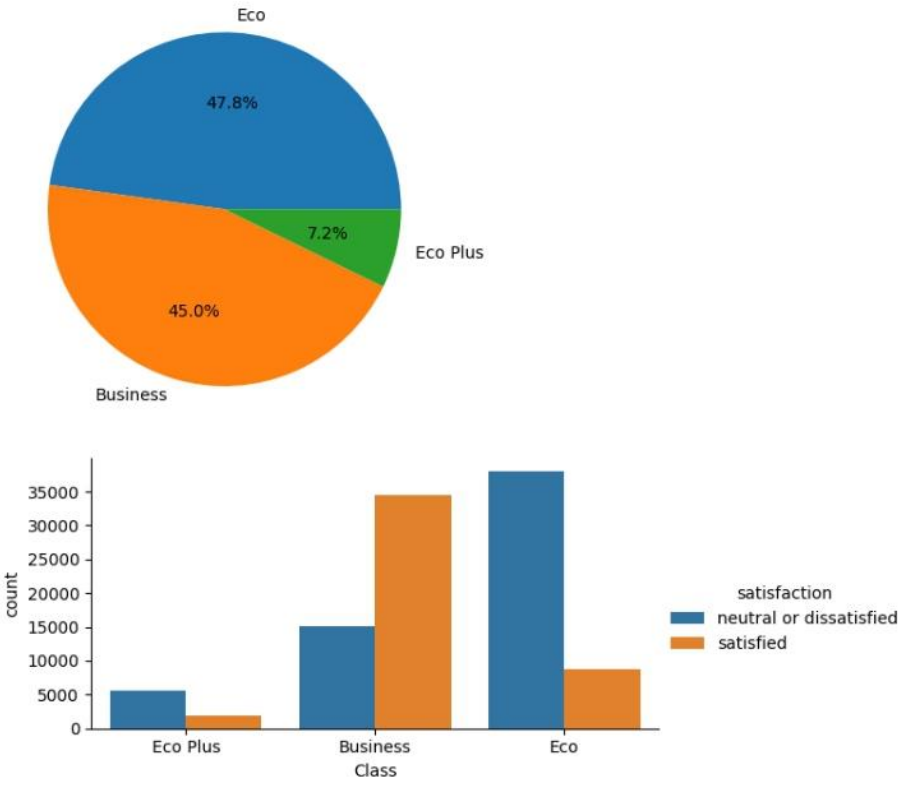


Рисунок 1.9. Діаграми розподілу значень ознаки "Class".

Аналізуючи таку ознаку як "Flight Distance", тобто дистанцію авіаперельоту, бачимо що більшість подорожей було здійснено на відстань що не перевищує 2500 миль (рис. 1.10). При цьому кількість нейтральних або незадоволених, переважає серед пасажирів що здійснили коротші подорожі, у той час, як пасажири що подорожували більш тривалий час задоволені.

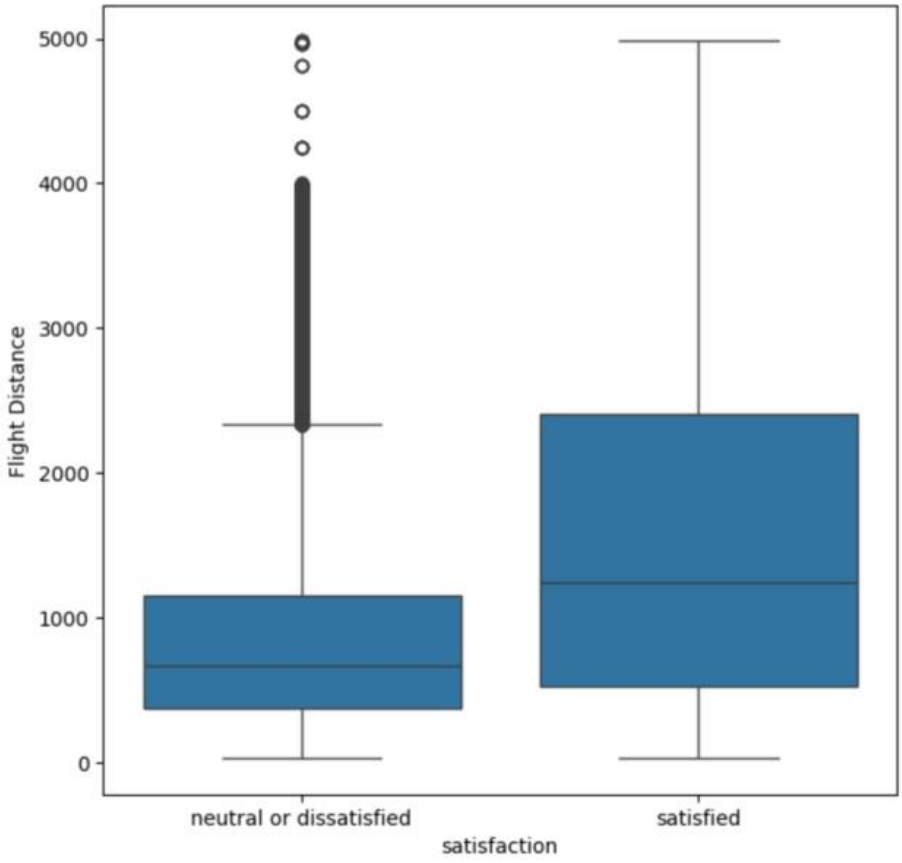
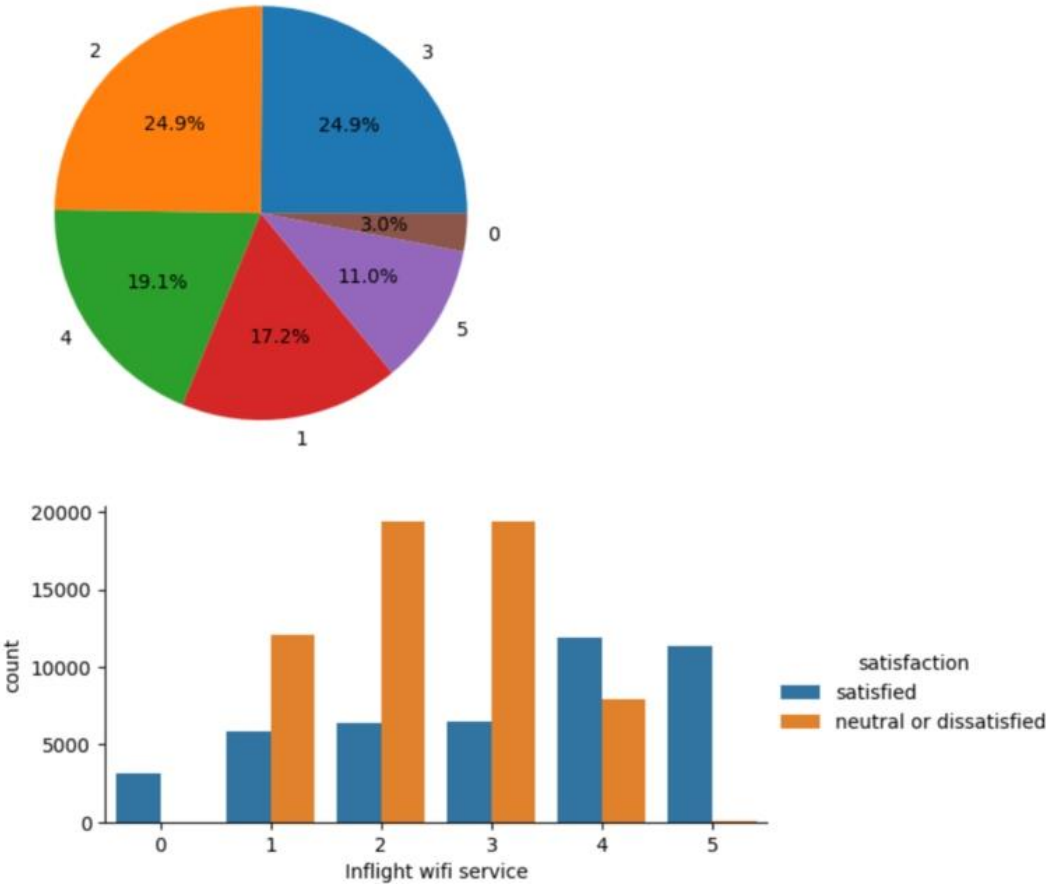


Рисунок 1.10. Розподіл цільової змінної залежно від дистанції авіаперельоту.

Розглядаючи ознаку "Inflight wifi service" (рис. 1.11), значення якої є оцінкою якості послуги "wifi" на борту літака за шкалою від 0 до 5, бачимо, що значення від 1 до 4 мають відносно рівномірне розподілення у загальній сукупності, з перевагою значень 2 та 3. У той час як максимальна оцінка 5 та мінімальна 0, являють собою меншість. При цьому, кількість задоволених пасажирів переважає серед тих, хто оцінив "Inflight wifi service" значеннями 0, 4

та 5, у той час, як пасажири з оцінкою 1, 2 та 3, які являють собою більшість всіх пасажирів переважно нейтральні або незадоволені.



у

Рисунок 1.11. Діаграми розподілу значень ознаки "Inflight wifi service".

Наступною ознакою у наборі даних є "Departure/Arrival time convenient", тобто зручність часу відправлення та прибуття авіарейсу, що оцінюється пасажирами за шкалою від 0 до 5. Суттєва перевага кількості нейтральних або незадоволених опитаних над задоволеними, як бачимо на рис. 1.12, має місце серед опитаних що надали оцінку 4 або 5, з долею цих значень у загальній сукупності значень ознаки 24,6% та 17,3% відповідно.

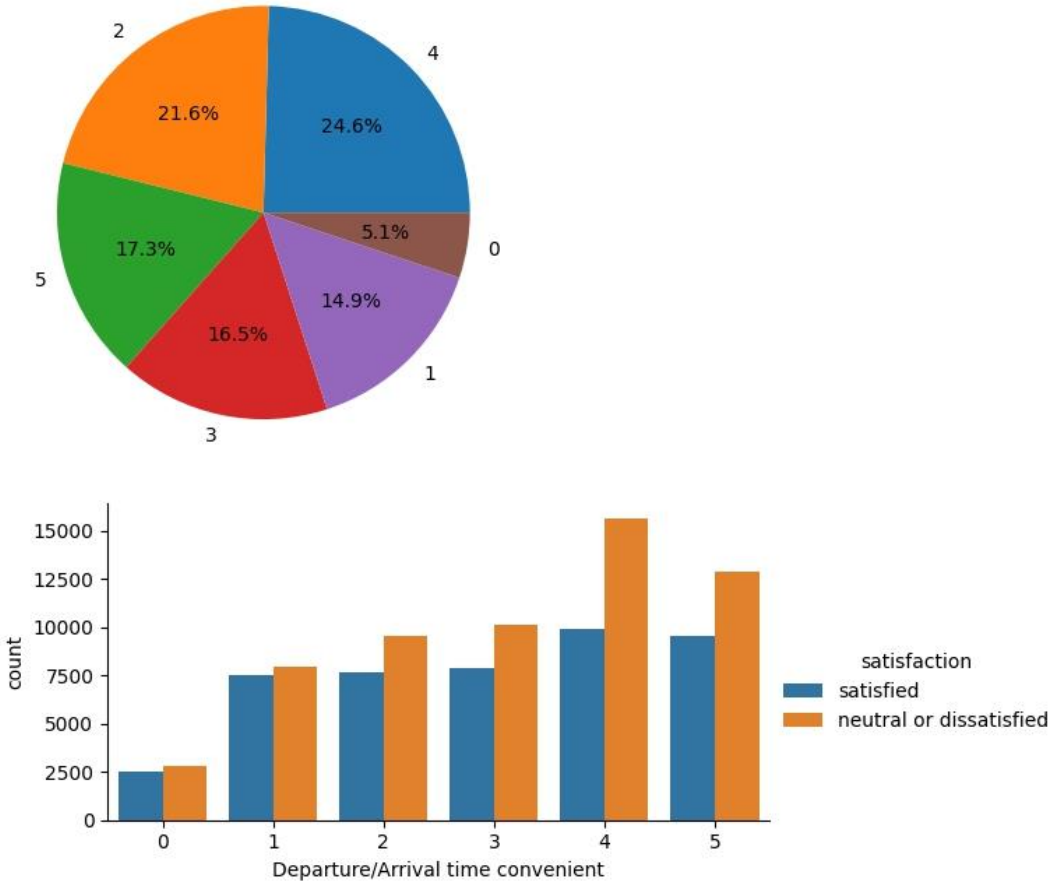


Рисунок 1.12. Діаграми розподілу значень ознаки "Departure/Arrival time convenient".

За ознакою "Ease of Online booking", тобто зручність онлайн бронювання, бачимо суттєву перевагу кількості нейтральних або незадоволених пасажирів над кількістю задоволених, які оцінили якість послуги значенням 1, 2 або 3 (рис. 1.13). Серед пасажирів які оцінили якість послуги значенням 0, 4 або 5, більшість опитаних задоволені.

Розглядаючи розподіл значень ознаки "Gate location" (рис. 1.14), бачимо перевагу кількості задоволених пасажирів лише серед тих, хто оцінив якість послуги значенням 5, по шкалі від 0 до 5. Кількість нейтральних або незадоволених найбільша серед пасажирів які оцінили послугу значенням 3 або 4, частка яких у загальній сукупності складає 23,5% та 13,4%.

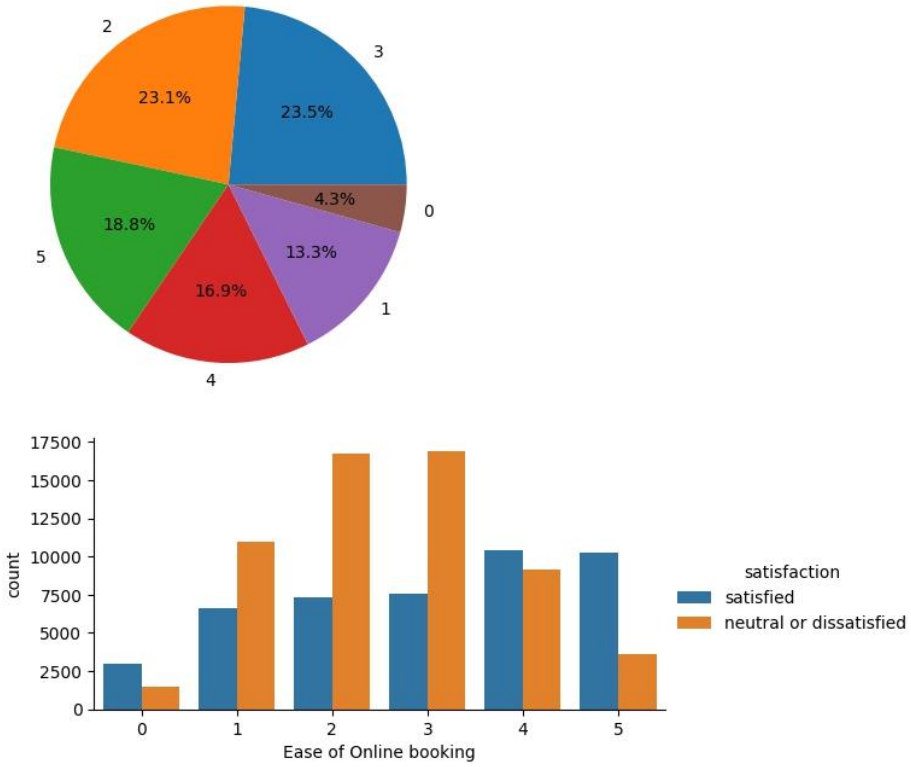


Рисунок 1.13. Діаграми розподілу значень ознаки "Ease of Online booking".

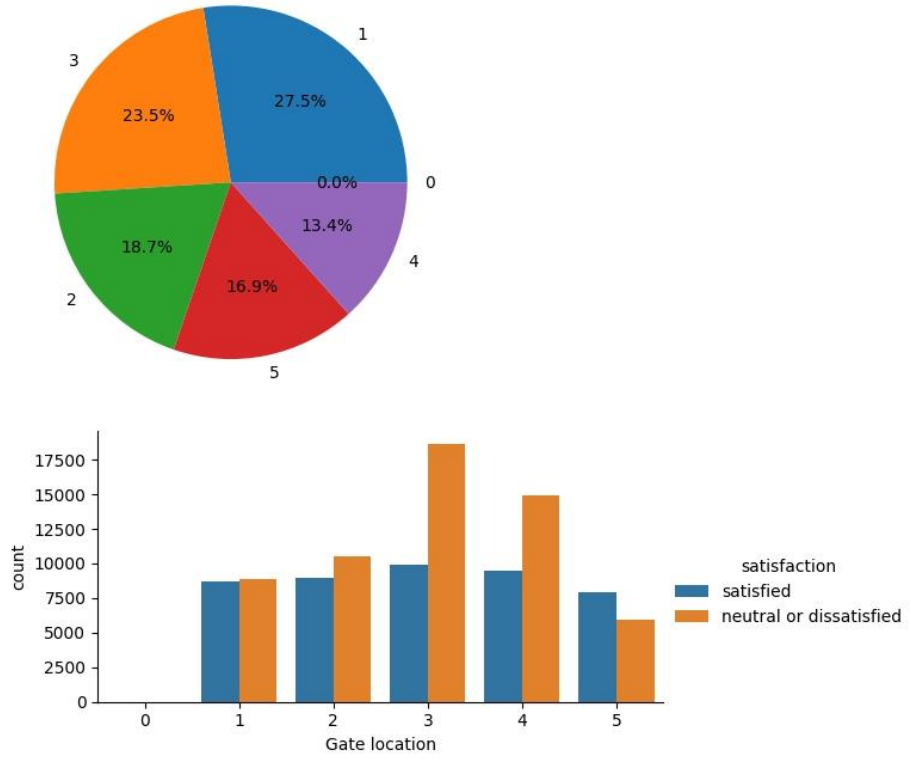


Рисунок 1.14. Діаграми розподілу значень ознаки "Gate location".

Наступним кроком розглянемо розподіл значень ознаки "Food and drink", що характеризує якість їжі та напоїв (рис. 1.15). Найменшу долю у загальній сукупності має значення ознаки 0 та 3, з показниками 0,1% та 12,4% відповідно, інші значення розподілені рівномірно. Значну перевагу кількості нейтральних або незадоволених пасажирів над задоволеними бачимо серед опитаних, які оцінили якість послуги значенням 1, але через те, що їх кількість становить 21,5% у загальній сукупності, це не має істотного впливу на загальні показники цільової змінної.

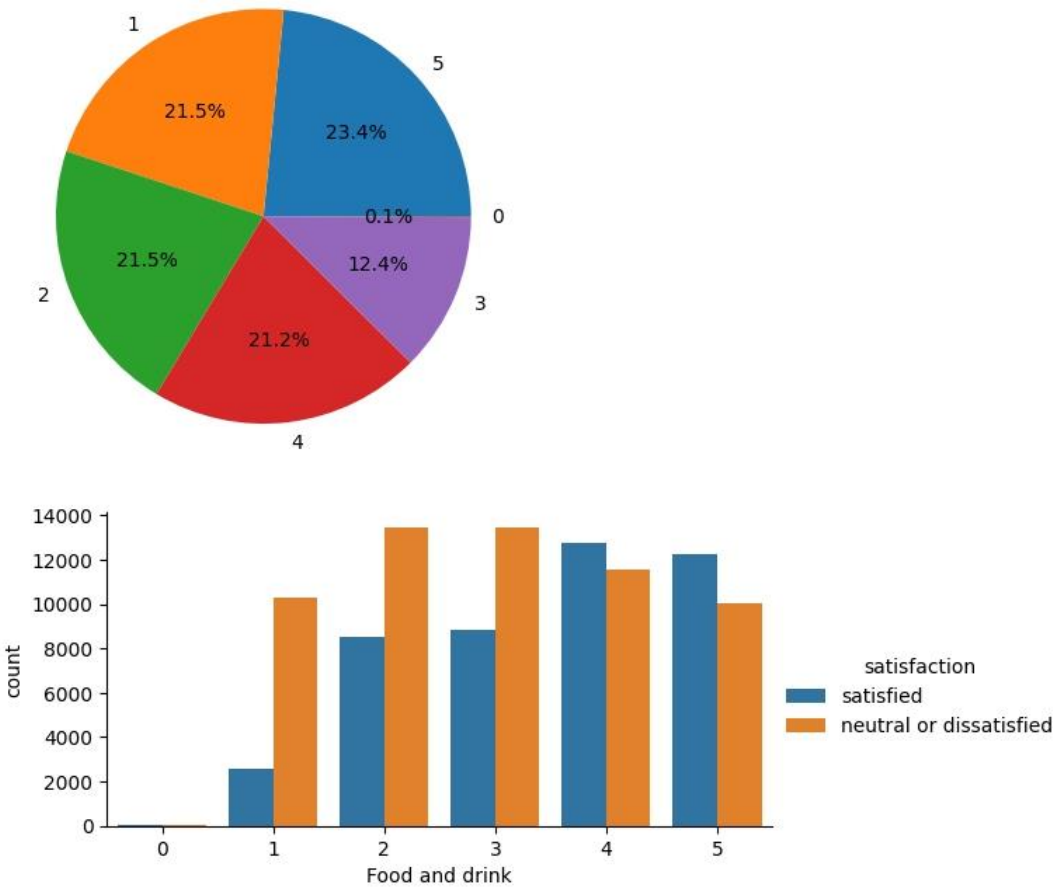


Рисунок 1.15. Діаграми розподілу значень ознаки "Food and drink".

Розглядаючи розподіл пасажирів за значеннями ознаки "Online boarding" (рис. 1.16), бачимо більшість нейтральних або незадоволених серед тих опитаних, які оцінили послугу значенням від 0 до 3 по 5-ти бальній шкалі із долею цих значень у загальній сукупності значень ознаки 68,6%. У той час як

серед пасажирів що оцінили ознаку значенням 4 або 5, суттєво переважають задоволені опитані.

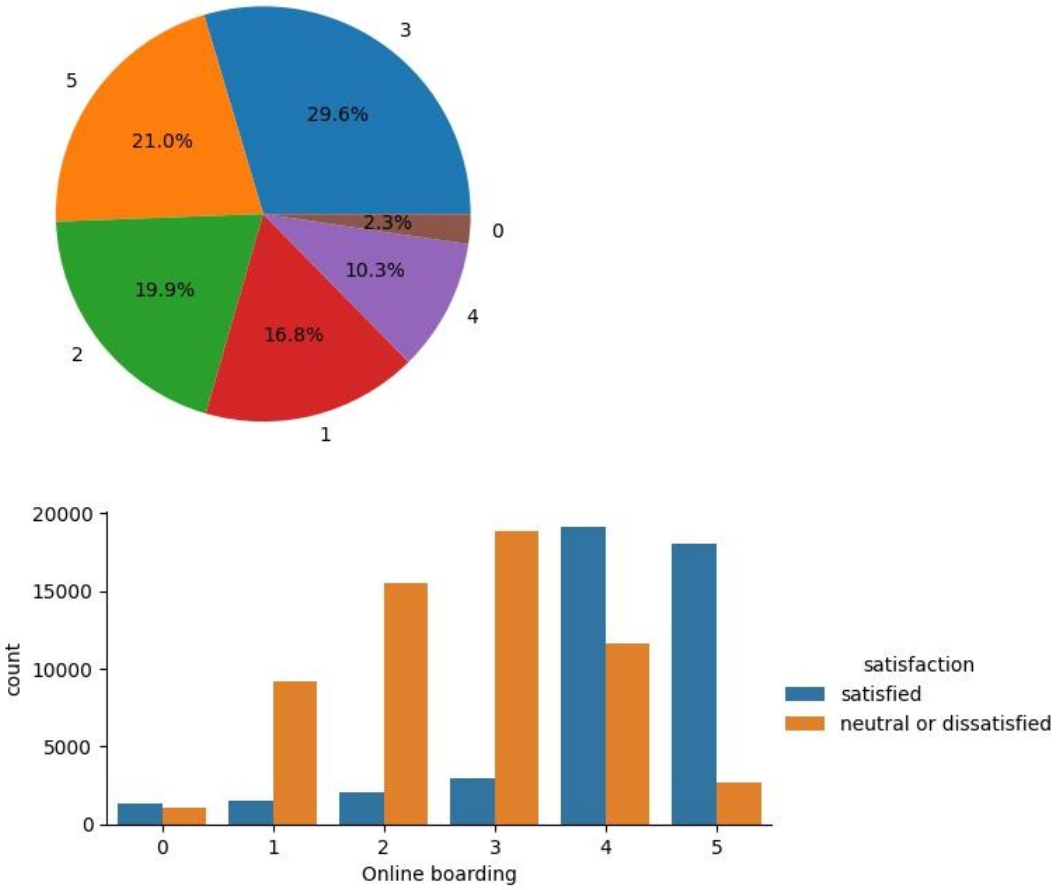


Рисунок 1.16. Діаграми розподілу значень ознаки "Online boarding".

Аналізуючи ознаку "Seat comfort", тобто зручність місця сидіння, бачимо як і у попередній ознаці помітний взаємозв'язок між оцінкою послуги та задоволеністю пасажирами авіакомпанією (рис. 1.17). Так, серед опитаних, які оцінили якісь послуги від 0 до 3, - переважна частина нейтральні або незадоволені. У той час як пасажири які оцінили ознаку значенням 4 або 5, по 5-ти бальній шкалі, - переважно задоволені, з високим рівнем задоволеності опитаних що надали оцінку 5.

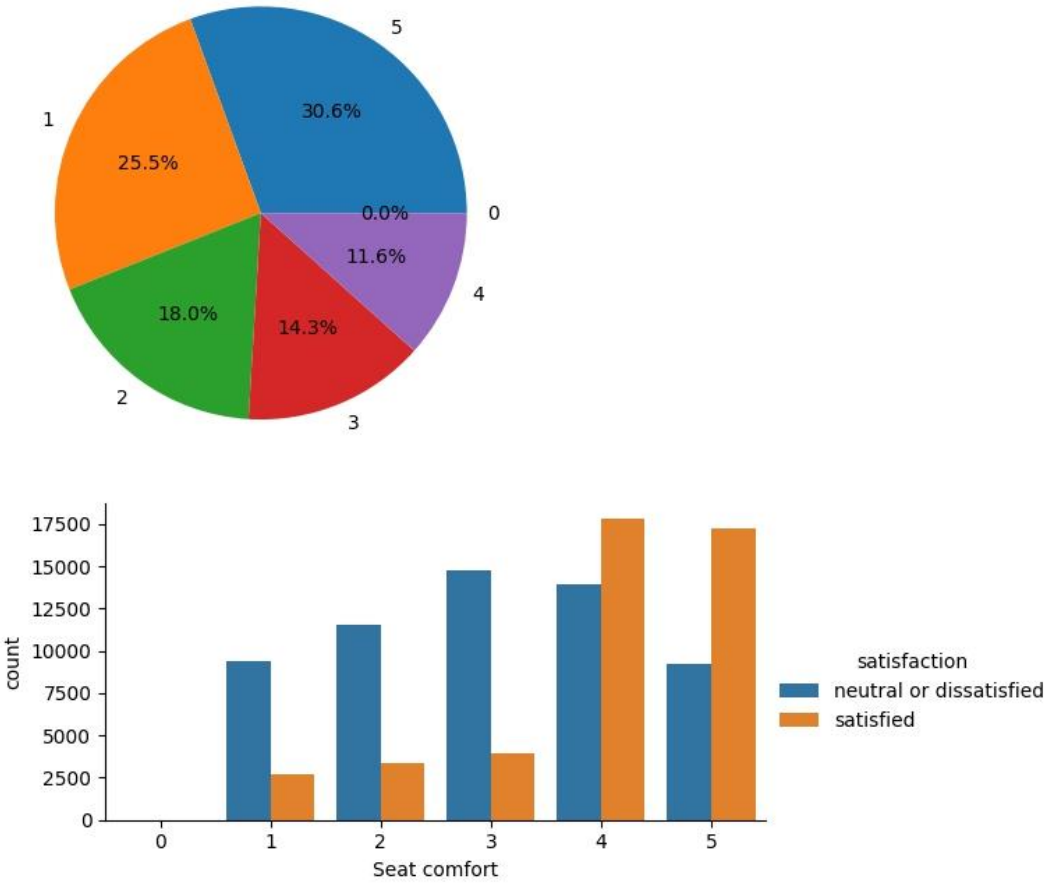


Рисунок 1.17. Діаграми розподілу значень ознаки "Seat comfort".

Наступною ознакою у наборі даних є "Inflight entertainment", тобто розваги під час польоту (рис. 1.18), аналізуючи яку, бачимо високу кореляцію між ступенем оцінки якості цієї послуги пасажирами та цільовою змінною. Так, кількість задоволених суттєво переважає серед опитаних які надали високу оцінку якості послуги, - 4 або 5.У той час як пасажири які оцінили послугу від 0 до 3, переважною більшістю нейтральні або незадоволені.

Розглядаючи ознаку "On-board service" що характеризує якість послуг на борту літака (рис. 1.19), бачимо виражену взаємозалежність між оцінкою якості послуги, та задоволеністю пасажира. Так, опитані які оцінили "On-board service" значеннями 4 та 5 переважно задоволені, з їх часткою у загальній сукупності 29,7% та 11,4%, відповідно. Переважна більшість опитаних серед тих хто надав оцінку від 0 до 4 нейтральні або не задоволені.

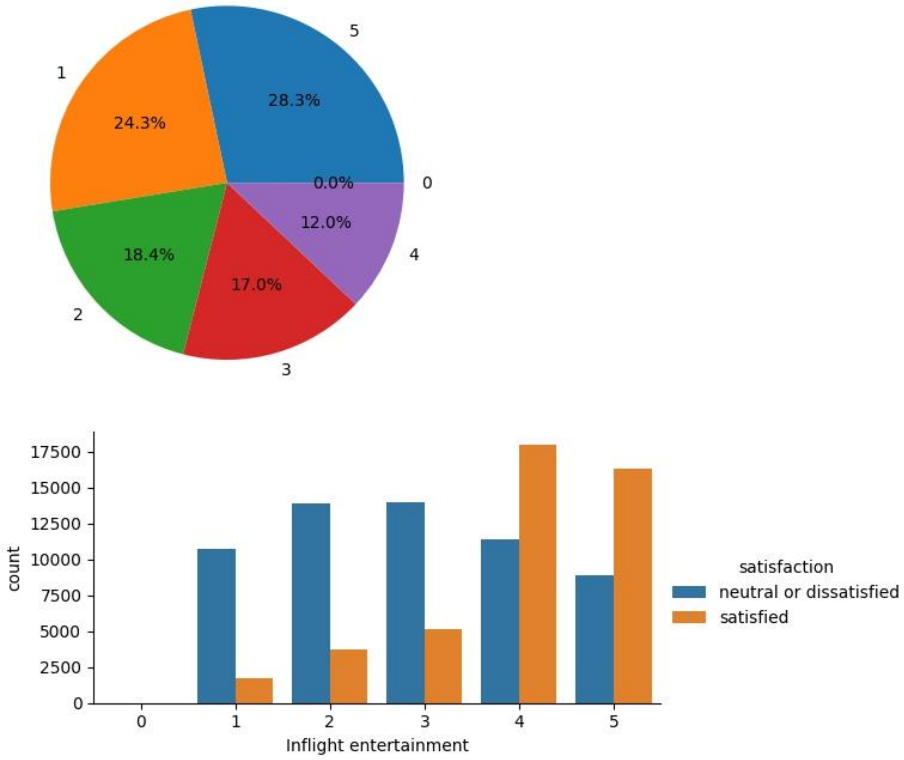


Рисунок 1.18. Діаграми розподілу значень ознаки "Inflight entertainment".

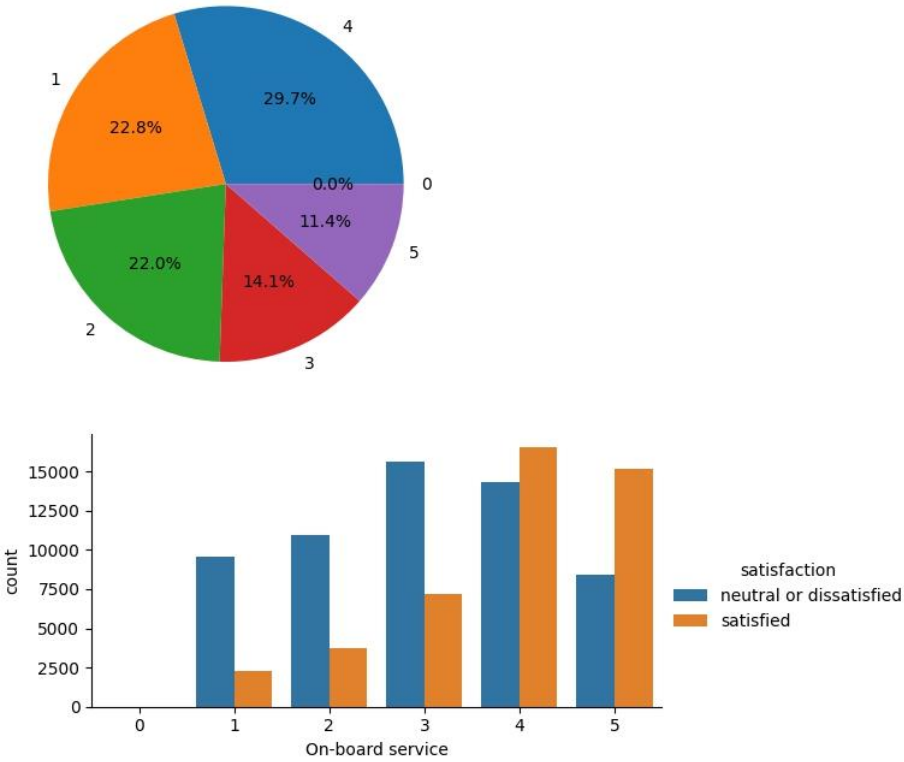


Рисунок 1.19. Діаграми розподілу значень ознаки "On-board service".

Наступною ознакою є "Leg room service" (рис. 1.20), з діаграми значень якої бачимо значну кореляцію між оцінкою якості послуги пасажирами та їх

задоволеністю авіакомпанією. Так, переважна більшість опитаних, що надали оцінку від 0 до 3 нейтральні або незадоволені, у той час, як більшість пасажирів що оцінили якість послуги значенням 4 або 5, - задоволені.

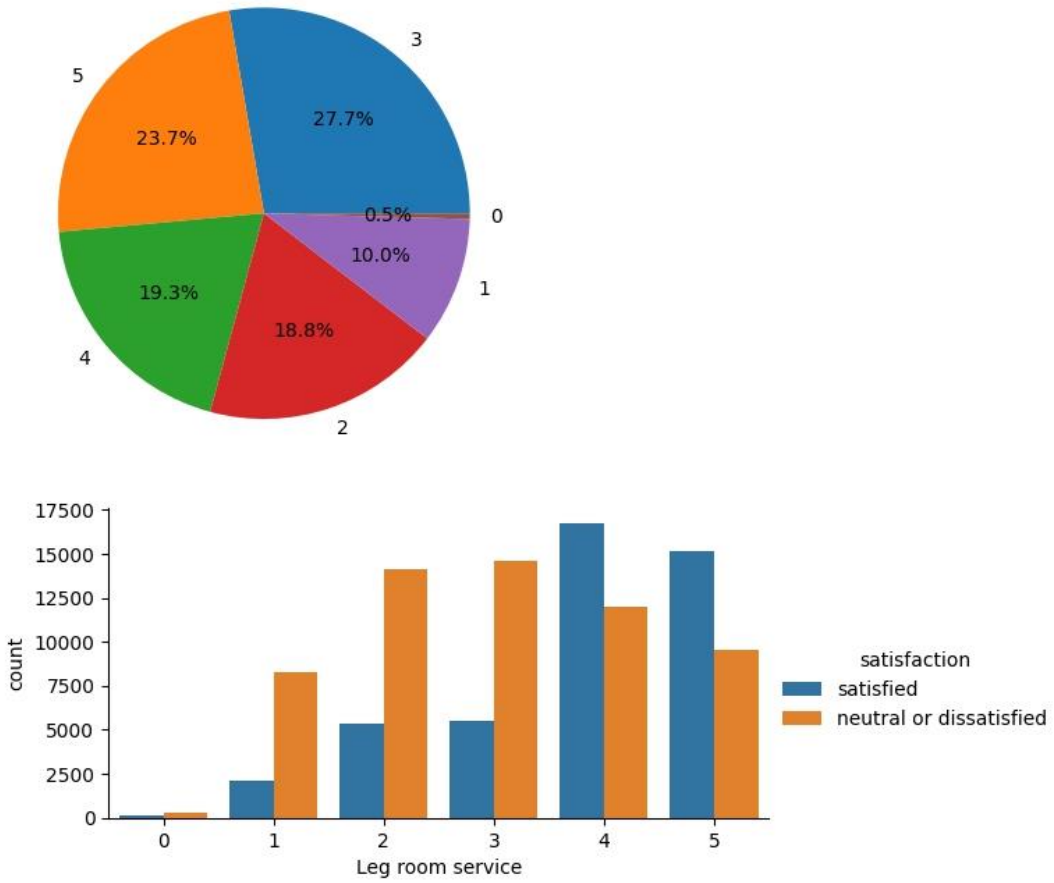


Рисунок 1.20. Діаграма розподілу значень ознаки "Leg room service".

Розглядаючи значення ознаки "Baggage handling" відносно цільової змінної (рис. 1.21), бачимо, що задоволені опитані переважають лише серед тих, хто надав оцінку якості послуги 5 по 5-ти бальній шкалі, при частці 19,9% цього значення у загальній сукупності значень ознаки. Серед тих, хто оцінив ознаку значенням від 0 до 3 суттєво переважає кількість нейтральних або незадоволених пасажирів, з майже рівним значенням відносно цільової змінної серед опитаних з оцінкою 4, доля якої складає 36% у загальному обсязі.

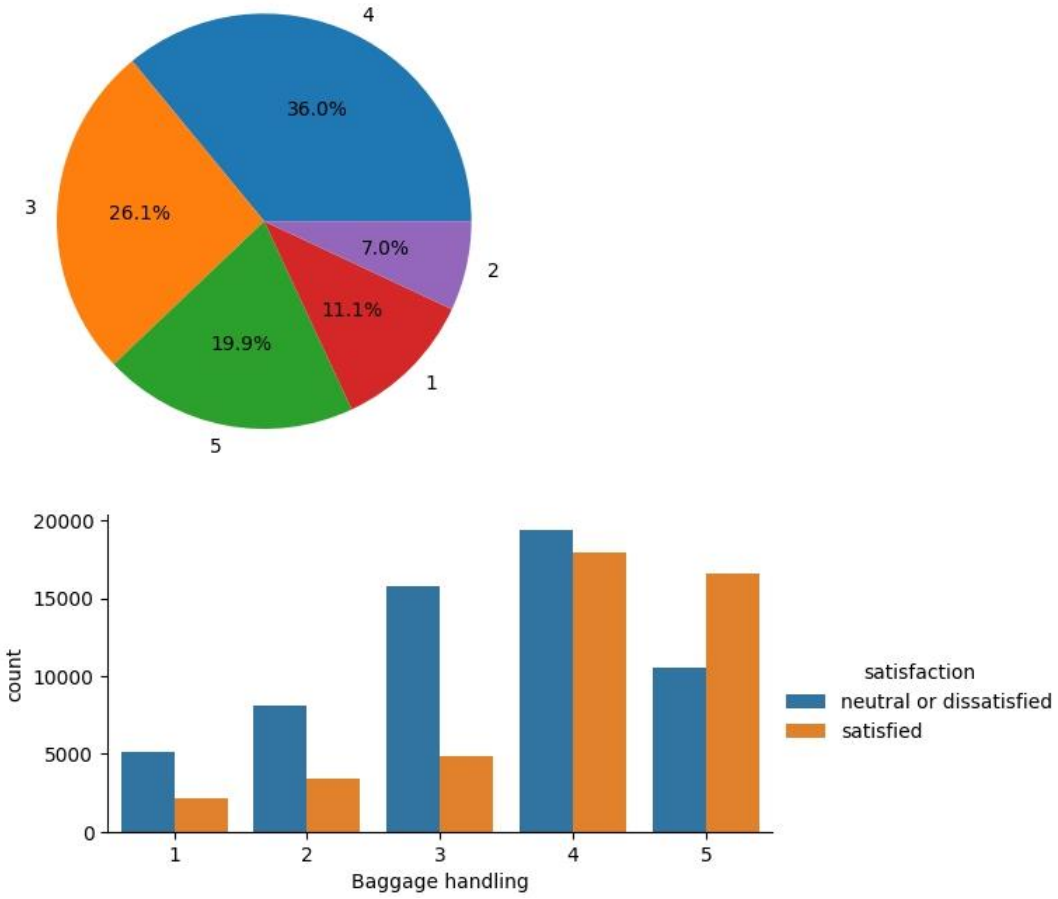


Рисунок 1.21. Діаграми розподілу значень ознаки "Baggage handling".

Далі проаналізуємо ознаку "Checkin service", діаграма залежності значень якої відносно цільової змінної (рис. 1.22) вказує на перевагу нейтральних або незадоволених пасажирів серед тих хто оцінив якість послуги від 1 до 4, з перевагою задоволених пасажирів серед надавших оцінку 5. Високу кореляцію з нейтральним або негативним відношенням пасажирів до авіакомпанії мають значення 1 та 2, з 27,4% та 12,4% частки у загальній сукупності значень відповідно.

Розглядаючи розподіл пасажирів за значеннями ознаки "Inflight service" (рис. 1.23), бачимо що опитаних, оцінивших її якість значенням 5 по 5-ти бальній шкалі більшість, а саме 36,5%, з перевагою задоволених пасажирів серед них. У той час як серед тих, хто оцінив якість послуги значенням від 1 до 3 значно переважають нейтральні або незадоволені опитані, з майже рівними показниками для значення 4.

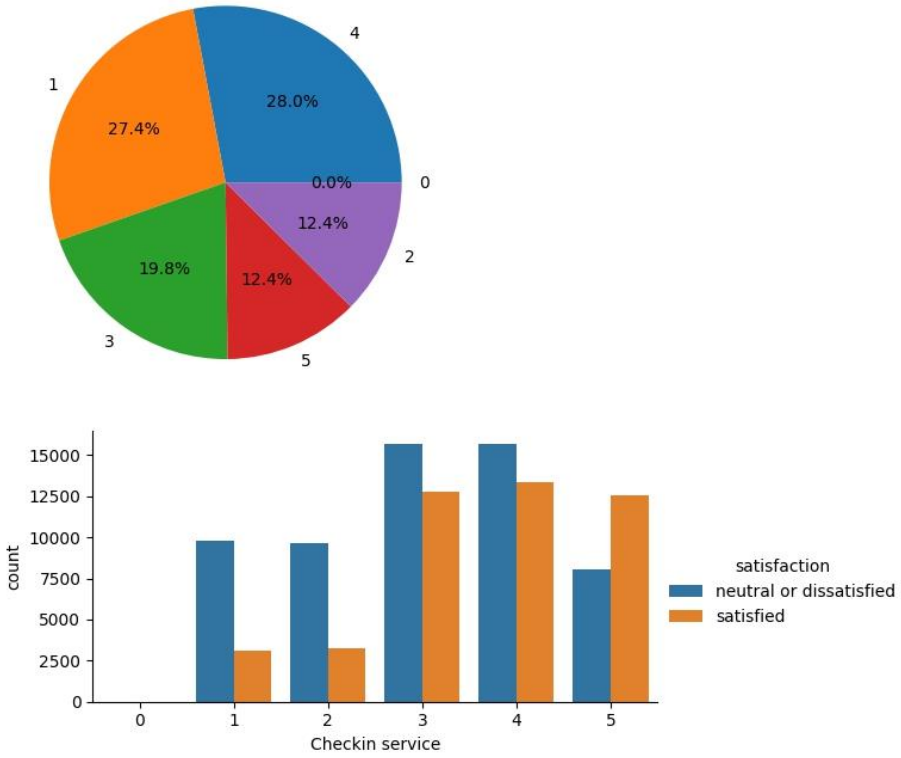


Рисунок 1.22. Діаграми розподілу значень ознаки "Checkin service".

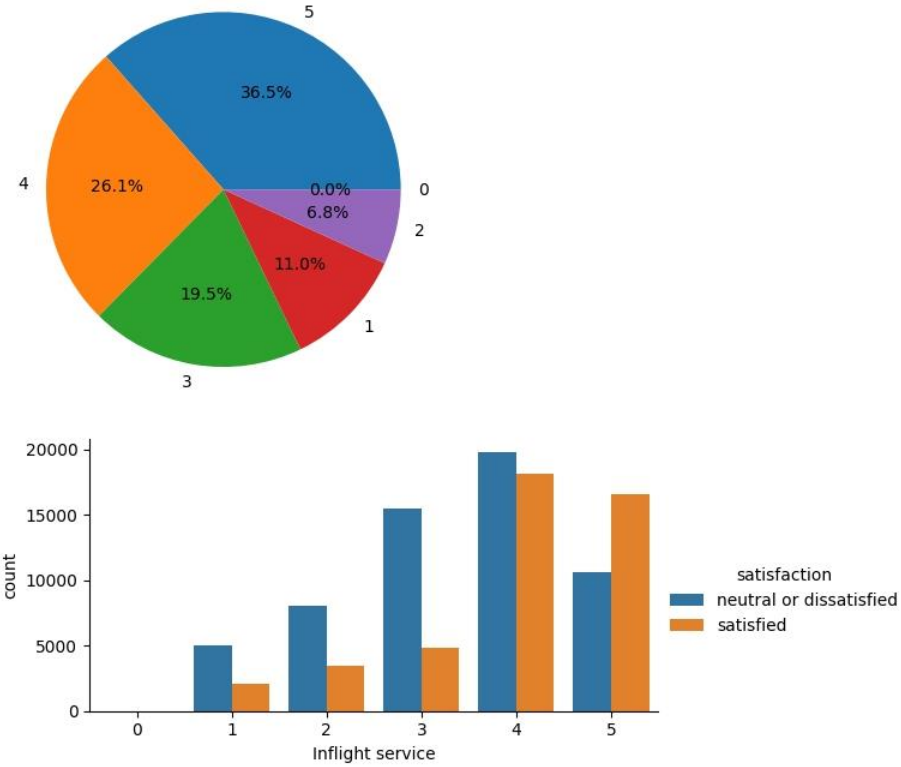


Рисунок 1.23. Діаграма розподілу значень ознаки "Inflight service".

Розглядаючи ознаку "Cleanliness" (рис. 1.24), бачимо високу взаємозалежність між значенням нейтральний або незадоволений цільової функції та значеннями ознаки 1 та 2, з долею цих значень у загальній сукупності 23,7% та 21,8%. Пасажири задоволені авіакомпанією переважають серед тих опитаних, хто оцінив якість послуги значенням 4 та 5 з майже рівними показниками для значення 4.

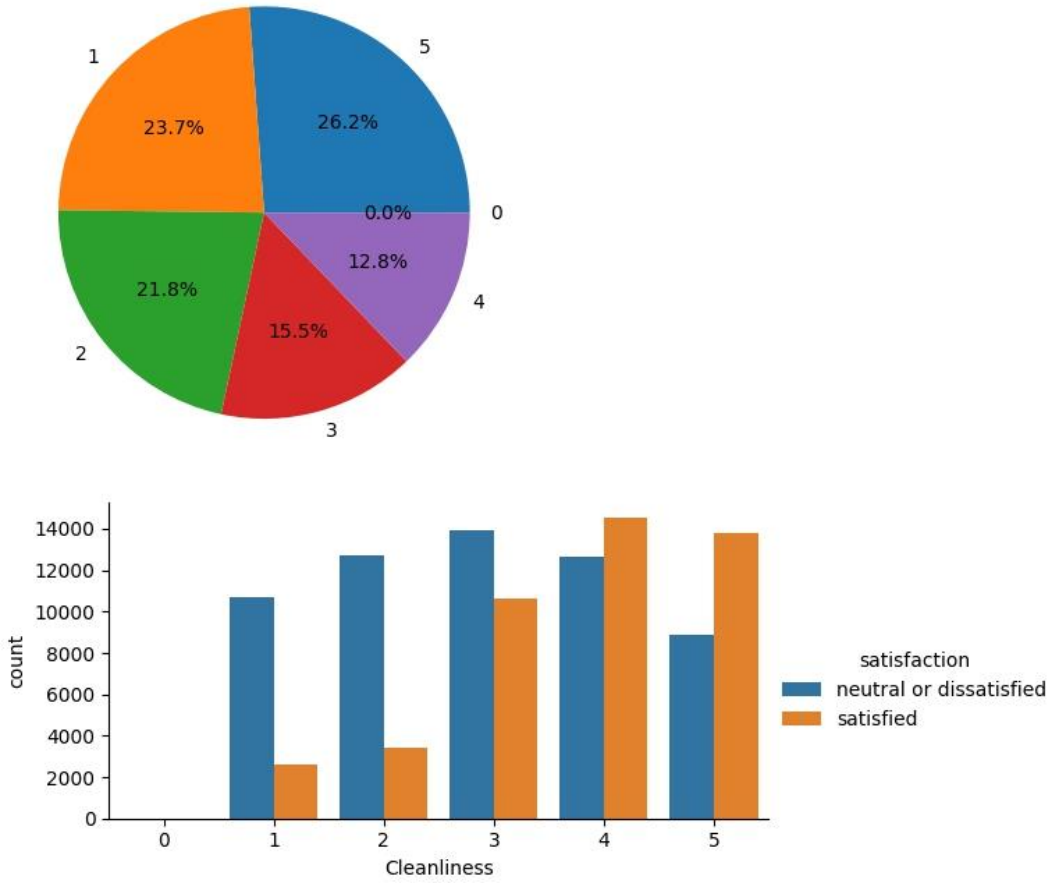


Рисунок 1.24. Діаграми розподілу значень ознаки "Cleanliness".

Розглядаючи ознаки "Departure Delay in Minutes" та "Arrival Delay in Minutes" (рис. 1.25) бачимо що щільність розподілу значень має обратнопропорційну залежність від величини значень затримки, з найбільшою щільністю у діапазоні від 0 до 600 хвилин.

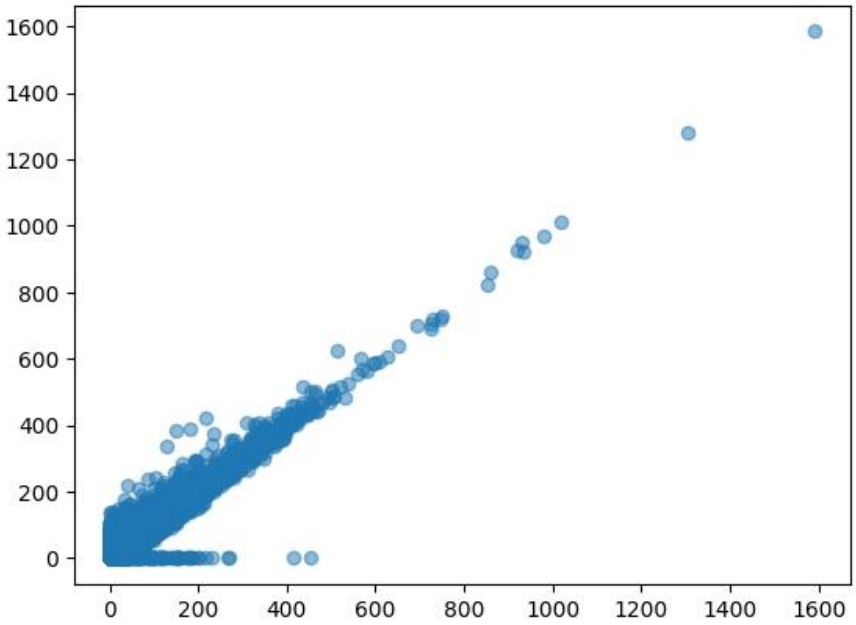


Рисунок 1.24. Графік розподілу значень ознак "Departure Delay in Minutes" та "Arrival Delay in Minutes".

Розглядаючи залежності значень ознаки "Departure Delay in Minutes" (рис. 1.25) та "Arrival Delay in Minutes" (Рис. 1.26) відносно цільової змінної, бачимо найбільшу щільність розподілу значень у діапазоні від 0 до 600 хвилин з прямопропорційною залежністю між щільністю значень, тобто наявністю затримок відправлення та прибуття та нейтральністю або незадоволеністю пасажирів.

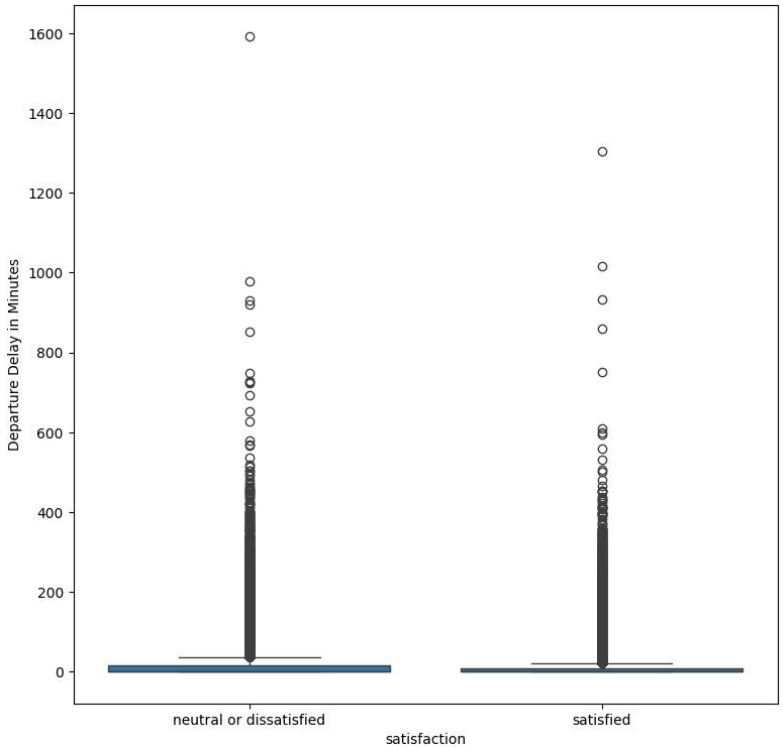


Рисунок 1.25. Розподіл цільової змінної залежно від значень ознаки "Departure Delay in Minutes".

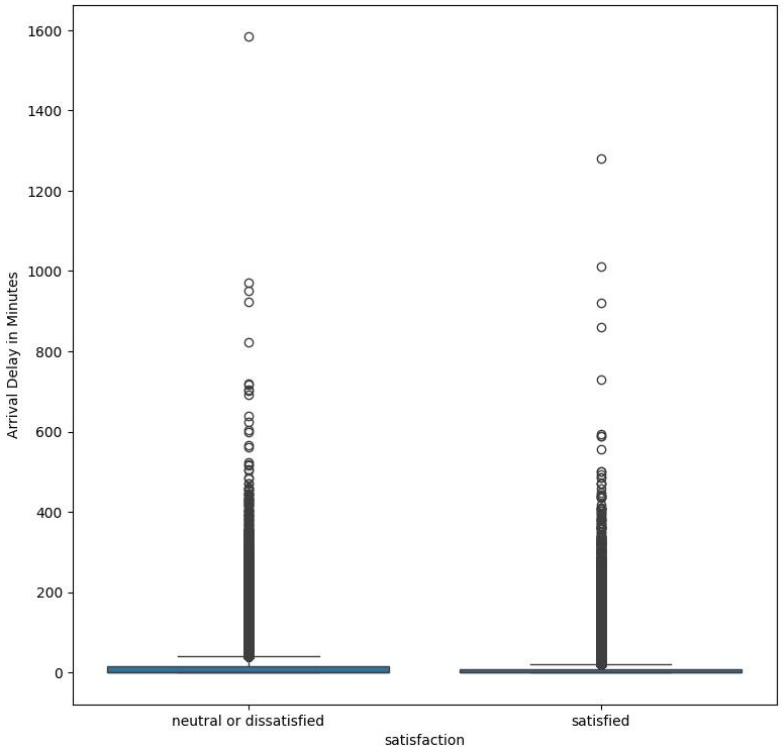


Рисунок 1.26. Розподіл цільової змінної залежно від значень ознаки "Arrival Delay in Minutes".

Розглянемо кореляцію залежності між цільовою ознакою та іншими ознаками набору даних, для цього використаємо кореляційну матрицю (рис. 1.27).

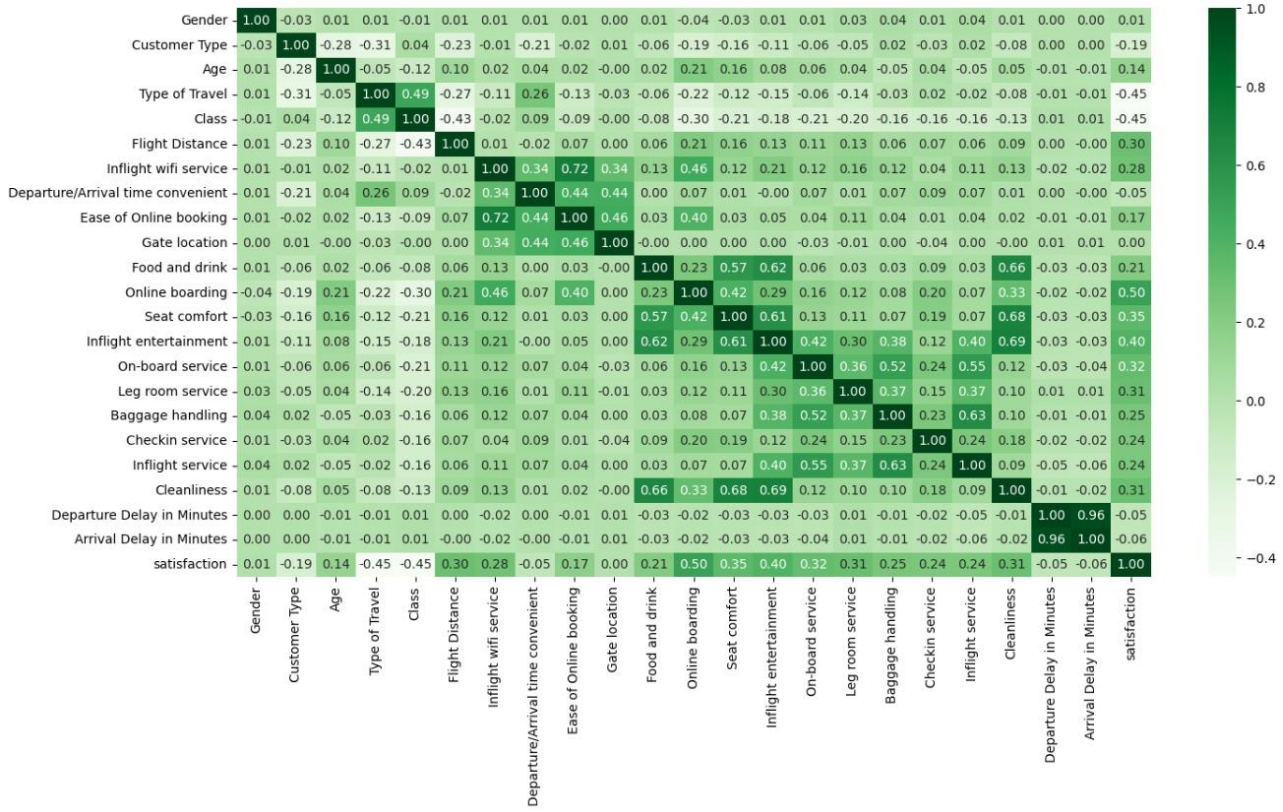


Рисунок 1.27 Кореляційна матриця ознак

Аналізуючи кореляційну матрицю ознак, бачимо помітну кореляцію між наступними ознаками, що свідчіть про схожість оцінки якості цих послуг опитаними, а саме:

- "Ease of Online booking" та "Inflight wifi service" із значенням 72%;

- "Food and drink" корелює з ознаками "Inflight entertainment" та "Seat comfort" зі значеннями 62% та 57%;

- "Cleanlines" корелює з ознаками "Food and drink", "Seat comfort" та "Inflight entertainment" зі значеннями 66%, 68% та 69% відповідно;

- "Inflight service" корелює з ознакою "Baggage handling" зі значеннями 63%.

Найвищий рівень кореляції мають ознаки ознаки "Departure Delay in Minutes" та "Arrival Delay in Minutes" зі значенням 96% що є логічним, бо затримка у відправлені призводить до затримки у прибутті.

Також бочимо, що цільова ознака "satisfaction" найбільше корелює з ознакою "Online boarding" зі значенням 50%.

Залишимо усі ознаки у складі набору даних, бо вони не повинні спричинити мультиколінеарнсть.

1.3.3 Підготовка даних щодо задоволеності пасажирів авіакомпанією до моделювання.

Перед тим як перейти до етапу моделювання, підготовимо дані. Цей процес включає в себе перевірку та уніфікацію типів даних, щоб їх можна було ефективно використати для аналізу за допомогою обраних методів та алгоритмів. Необхідно зробити дані однорідними для можливості їх використання із моделями машинного навчання.

Необхідним етапом перед моделюванням є розділення нашого набору даних на дві частини: тренувальну та тестову, що дозволить оцінити якість моделей машинного навчання.

Аналіз якості даних щодо можливості застосування для роботи моделей машинного навчання, показав що деякі ознаки мають значення відмінні від цифрового формату (рис. 1.3), тому їх потрібно перевизначити. Виконаємо це автоматично функцією бібліотеки LabelEncoder.

Інформація щодо набору даних після внесених змін перед моделюванням, представлена на рис.1. 28.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103904 entries, 0 to 103903
Data columns (total 23 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   Gender                            103904 non-null  int32
 1   Customer Type                     103904 non-null  int32
 2   Age                               103904 non-null  int64
 3   Type of Travel                    103904 non-null  int32
 4   Class                             103904 non-null  int32
 5   Flight Distance                   103904 non-null  int64
 6   Inflight wifi service             103904 non-null  int64
 7   Departure/Arrival time convenient 103904 non-null  int64
 8   Ease of Online booking            103904 non-null  int64
 9   Gate location                     103904 non-null  int64
 10  Food and drink                    103904 non-null  int64
 11  Online boarding                   103904 non-null  int64
 12  Seat comfort                      103904 non-null  int64
 13  Inflight entertainment            103904 non-null  int64
 14  On-board service                  103904 non-null  int64
 15  Leg room service                  103904 non-null  int64
 16  Baggage handling                  103904 non-null  int64
 17  Checkin service                   103904 non-null  int64
 18  Inflight service                  103904 non-null  int64
 19  Cleanliness                       103904 non-null  int64
 20  Departure Delay in Minutes        103904 non-null  int64
 21  Arrival Delay in Minutes          103904 non-null  float64
 22  satisfaction                      103904 non-null  int32
dtypes: float64(1), int32(5), int64(17)
memory usage: 16.3 MB
```

Рисунок 1.28 Характеристики підготовленого до моделювання набору даних

Таким чином, всі відсутні дані були заповнені нулями, а перевірка на дублікати за допомогою методу duplicated() бібліотеки pandas показала відсутність дублікатів. Окрім того, дані є достатньо збалансованими, для досягнення оптимальної продуктивності моделей (рис. 1.29).

	Gender	Customer Type	Age	Type of Travel	Class	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	103904.000000	103904.000000	103904.000000
mean	0.492541	0.182678	39.379706	0.310373	0.594135	1189.448375	2.729683	3.060296	2.756901
std	0.499947	0.386404	15.114964	0.462649	0.620799	997.147281	1.327829	1.525075	1.398929
min	0.000000	0.000000	7.000000	0.000000	0.000000	31.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	27.000000	0.000000	0.000000	414.000000	2.000000	2.000000	2.000000
50%	0.000000	0.000000	40.000000	0.000000	1.000000	843.000000	3.000000	3.000000	3.000000
75%	1.000000	0.000000	51.000000	1.000000	1.000000	1743.000000	4.000000	4.000000	4.000000
max	1.000000	1.000000	85.000000	1.000000	2.000000	4983.000000	5.000000	5.000000	5.000000

Рисунок 1.29 Статистичні характеристики ознак набору даних отримані за допомогою метода "describe()".

1.4 Висновки до розділу 1

Аналіз та підготовка даних є важливими складовими аналітичного прогнозування, які необхідно виконати перед тим, як застосовувати до набору даних моделі машинного навчання, бо використання непідготовлених даних внесе спотворення та зменшить передбачувальну здатність моделі.

Були розглянуті всі ознаки набору даних, а саме доля кожного значення ознаки у загальній сукупності значень ознаки, а також кількість задоволених та нейтральних або незадоволених осіб по кожному значенню ознаки, щоб зрозуміти яке зі значень має більший вплив на цільову змінну у порівнянні з іншими. У процесі аналізу виявлено, що найбільшу кореляцію у 50% з цільовою ознакою має ознака "Online boarding".

У процесі підготовки даних були виявлені та заповнені нулями відсутні значення ознаки "Arrival Delay in Minutes", видалені колонки "id" та "Unnamed: 0", які не несуть інформаційної цінності й не можуть бути використані у навчанні моделей. Перевірка на дублікати показала їх відсутність, а ознаки значення яких мали тип даних "object" були переведені у цифровий формат.

Таким чином, на етапі підготовки даних було створено навчальну вибірку, яка є збалансованою та готовою до застосування у роботі моделей машинного навчання.

1.5 Постановка задачі дослідження

Прогнозування задоволеності пасажирів має велике значення для авіакомпанії, тому що дозволяє передбачити задоволеність різних категорій пасажирів якістю наданих послуг, що дає можливість вдосконалити та оптимізувати ці послуги, а також вжити необхідні заходи з втримання окремих груп пасажирів, як, наприклад, спеціальні пропозиції та акції.

Метою дослідження є розробка інформаційно-аналітичної системи прогнозування задоволеності пасажирів авіакомпанією з використанням методів та алгоритмів машинного навчання.

Було обрано статистичні дані 103904 пасажирів авіакомпаній. Набір даних складається з 23 ознак, значення яких містять характеристику пасажирів та їх оцінку якості надання послуг. Дані були проаналізовані та підготовлені.

На наступному етапі дослідження розглянемо основні методи та алгоритми машинного навчання з виконання задачі класифікації, проведемо моделювання, застосувавши їх до нашого набору даних. Оберемо моделі які найбільше підходять для роботи з нашим набором даних і дослідимо їх детальніше, обравши серед них найефективнішу модель.

РОЗДІЛ 2. ОСНОВНІ АЛГОРИТМИ МАШИННОГО НАВЧАННЯ

2.1. Постановка задачі класифікації

Машинне навчання - це розділ штучного інтелекту, який вивчає методи, що дозволяють комп'ютерам навчатися на основі даних. Моделі машинного навчання можуть бути навчені виконувати різні завдання, такі як класифікація, регресія, розпізнавання образів, обробка природної мови тощо.

Основні підходи до машинного навчання можна розділити на три основні категорії: навчання з учителем, навчання без вчителя та навчання з підкріпленням (рис. 2.1).

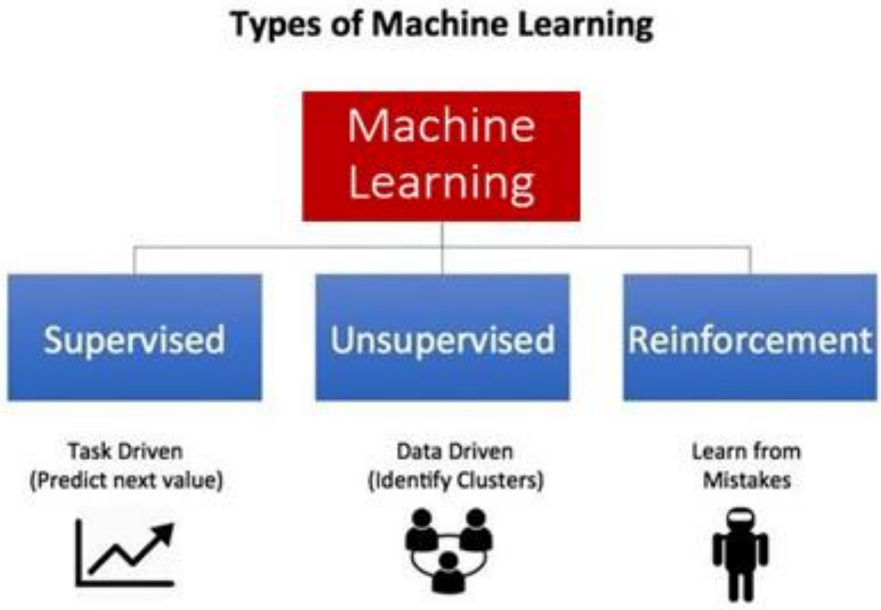


Рисунок 2.1 Основні підходи до машинного навчання [10]

Навчання з учителем або кероване навчання (англ. supervised learning) - це тип машинного навчання, в якому моделі навчаються на наборі даних, який містить помічені зразки, що використовуються для навчання моделі, як правильно ідентифікувати нові зразки.

Навчання без вчителя або некероване навчання (англ. unsupervised learning) - це тип машинного навчання, в якому моделі навчаються на непоміченому наборі даних. Ці моделі можуть бути використані для виявлення

прихованих закономірностей у даних, за допомогою, наприклад, кластерного аналізу

Навчання з підкріпленням (reinforcement learning) - це тип машинного навчання, в якому агент навчається приймати рішення в середовищі, отримуючи за них винагороду або покарання. У машинному навчанні з підкріпленням агент взаємодіє з середовищем, приймаючи послідовність дій. Після кожної дії агент отримує винагороду або покарання. Мета агента - навчитися приймати такі дії, щоб отримувати максимальну суму винагороди.

Розглянемо подробніше навчання з учителем, бо саме цей тип машинного навчання буде застосований у даній роботі для задачі бінарної класифікації.

Отже, машинне навчання з учителем є підгалуззю машинного навчання, де модель навчається на основі помічених даних, де для кожного вхідного прикладу відома його цільова величина. Головна ідея полягає в тому, щоб навчити модель відповідати правильними відповідями на основі тренувальних даних та використовувати цю інформацію для здійснення прогнозів або класифікації нових, раніше невідомих даних, тобто навчитися знаходити кореляцію між вхідними даними та вихідними даними. Після навчання модель може використовуватися для прогнозування вихідного значення для нових зразків, для яких вихідні дані невідомі.

Алгоритми машинного навчання з учителем можуть використовуватись для задач класифікації та регресії.

Класифікація у контексті машинного навчання - це задача, у якій модель навчається класифікувати зразки відносячи їх до різних категорій. Наприклад, модель машинного навчання з учителем може бути навчена класифікувати зображення тварин відповідно до їх видів.

Регресія у контексті машинного навчання - це задача у якій модель навчається прогнозувати значення величини. Наприклад, модель машинного навчання може бути навчена прогнозувати ціну на акції або, наприклад, нерухомість.

У задачах класифікації модель навчається на наборі даних, який складається з двох частин:

- вхідні дані що містять характеристики, які описують зразок. Наприклад, для класифікації зображень тварин вхідні дані можуть включати розмір, форму та колір тварини.

- вихідні дані що містять характеристики категорії до якої належить зразок. Наприклад, для класифікації зображень тварин вихідні дані можуть бути такими: "собака", "кішка", "птах" тощо.

Мета моделі машинного навчання для класифікації - навчитися знаходити кореляцію між вхідними та вихідними даними відносячи вхідні дані до класів відповідно вихідним даним. Після навчання модель може використовуватися для прогнозування класів для нових зразків, для яких клас невідомий.

Існує два основних типи класифікації, - бінарна та поліноміальна. У бінарної класифікації є два можливі значення вихідної змінної. Наприклад, модель машинного навчання може бути навчена класифікувати ознаки на "так" або "ні". У поліноміальній класифікації є більше двох можливих значень вихідної змінної, тобто, модель машинного навчання може бути навчена класифікувати ознаки на декілька класів.

Задачі класифікації мають широкий спектр застосувань у різних галузях, таких як, наприклад:

- медицина: класифікація може використовуватися для діагностики захворювань, прогнозування ризику захворювань та персоналізації лікування;

- фінанси: для виявлення шахрайства, прогнозування фінансових ринків та аналізу даних клієнтів;

- безпека: для виявлення загроз, фільтрації спаму та захисту від кібератак.

- розробка продуктів: для тестування продуктів, персоналізації досвіду користувачів та рекомендацій продуктів тощо.

З безлічі алгоритмів класифікації розглянемо найпоширеніші, а саме:

- логістичну регресія;
- дискримінантний аналіз;
- метод опорних векторів
- дерева рішень;
- метод найближчих сусідів;
- наївну модель Байєса;
- ансамблеві методи;
- штучні нейронні мережі.

2.1.1 Логістична регресія

Логістична регресія (аннгл. linear regression, LR) - це алгоритм машинного навчання, який використовується для прогнозування ймовірності того, що зразок належить до певного класу. Для моделювання ймовірності бінарної залежної змінної використовується функція логістичної кривої для перетворення прогнозованої ймовірності в значення від 0 до 1. Якщо отримана ймовірність більше за порогове значення (зазвичай 0.5), то об'єкт відноситься до класу 1, інакше - до класу 0. Основна ідея полягає в тому, щоб знаходити оптимальні ваги для лінійної комбінації вхідних факторів, які максимально добре визначають роздільну лінію між класами.

За допомогою логістичної регресії можна оцінювати вірогідність того, що подія настане для конкретного випробуваного (хворий / здоровий, повернення кредиту / дефолт і т.д.) [8].

Всі моделі регресії можуть бути представлені у вигляді формули:

$$y=F(x1,x2,\ldots,xn) \qquad (2.1)$$

У множинної лінійної регресії передбачається, що залежна змінна є лінійною функцією незалежних змінних, а саме:

$$\ln(p1-p)=b0+ b1x1+ b2x2+\cdots+ bnxn \qquad (2.2)$$

де p - залежна змінна;

$x1,x2,\ldots xn$ - незалежні змінні;

$b0,b1,\dots\ bn$ - коефіцієнти.

Тренування моделі логістичної регресії включає в себе мінімізацію функції втрат, такої як бінарна крос-ентропія, і визначення оптимальних ваг для правильної класифікації тренувальних прикладів. У якості функції в моделях бінарної класифікації можуть бути використані логістична функція та функція стандартного нормального розподілу.

Слід зазначити, що логістична регресія може бути неточною для даних, які не є лінійно розділеними, - це може призвести до того, що модель буде краще класифікувати більш поширений клас, ніж менш поширений.

Логістична регресія може бути застосована до багатьох задач класифікації, - для прогнозування того, чи має пацієнт певне захворювання на основі його симптомів, чи є електронний лист спамом на основі його вмісту, чи поверне клієнт кредит, на основі його кредитної історії тощо.

2.1.2 Дискримінантний аналіз

Дискримінантний аналіз - це статистичний метод, який використовується в машинному навчанні для визначення різниці між двома або більше групами об'єктів та визначення факторів, які найкраще відрізняють ці групи. Основна ідея дискримінантного аналізу полягає в тому, щоб знайти лінійну комбінацію змінних, яка найкраще розділяє класи.

На рис. 2.3 можна побачити приклад розташування областей прийняття рішень в лінійному дискримінантному аналізі з трьома класами.

Існує кілька різних типів дискримінантного аналізу, які можна класифікувати за різними ознаками. Один з поширених способів класифікації полягає в тому, щоб розділити дискримінантний аналіз на параметричний і непараметричний.

Параметричний дискримінантний аналіз передбачає, що розподіл ймовірностей зразків різних класів відомий. У цьому випадку дискримінантна функція може бути обчислена за допомогою статистичних методів.

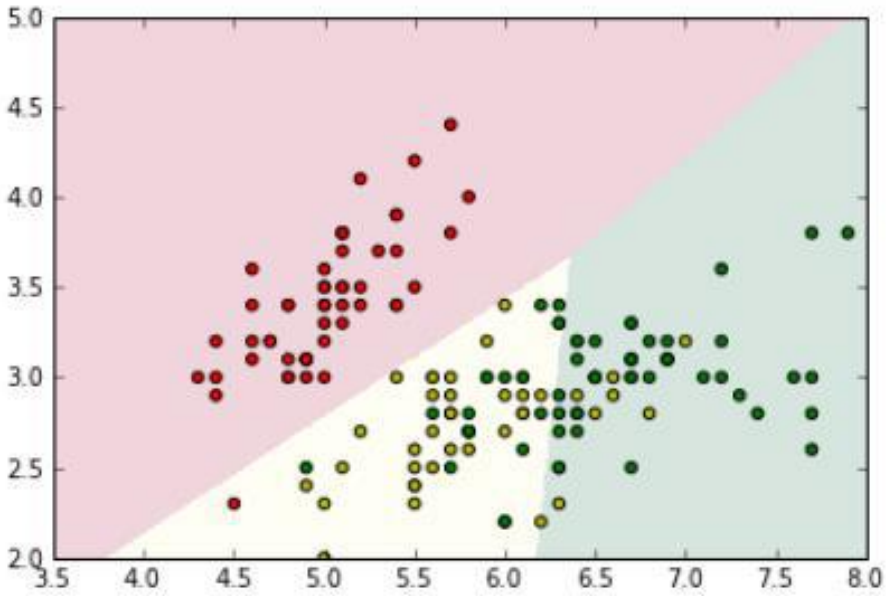


Рисунок 2.3. Приклад розташування областей прийняття рішень в лінійному дискримінантному аналізі з трьома класами.

Непараметричний дискримінантний аналіз не передбачає, що розподіл ймовірностей зразків різних класів відомий. У цьому випадку дискримінантна функція може бути обчислена за допомогою методів машинного навчання.

Існують три основні типи дискримінаційного аналізу:

- лінійний дискримінаційний аналіз (Linear Discriminant Analysis, LDA): цей метод шукає лінійну комбінацію ознак, яка максимально відокремлює класи. При цьому здійснюється проєкція даних на простір з меншою кількістю вимірів (зазвичай, на одну ось), що дозволяє легше проводити класифікацію. LDA передбачає, що ознаки мають нормальний розподіл та однакові коваріаційні матриці для всіх класів;

- квадратичний дискримінаційний аналіз (Quadratic Discriminant Analysis, QDA): у випадку QDA передбачається, що кожен клас має свою власну коваріаційну матрицю, що може робити модель більш гнучкою, але також призводити до меншої точності через більшу кількість параметрів, які потрібно оцінити;

- регуляризований дискримінаційний аналіз (Regularized Discriminant Analysis, RDA): це розширення LDA і QDA, яке використовує регуляризацію для поліпшення стабільності та уникнення перенавчання, особливо в ситуаціях з обмеженим обсягом навчальних даних.

Одним з найпоширеніших методів параметричного дискримінантного аналізу є лінійний дискримінантний. LDA передбачає, що розподіл ймовірностей зразків різних класів є нормальним. У цьому випадку дискримінантна функція LDA має вигляд:

$$f(x) = w^Tx + b \quad (2.3)$$

де w - вектор вагових коефіцієнтів моделі;

b - константа;

x - вектор характеристик зразка.

Вагові коефіцієнти w і константа b навчаються за допомогою методу максимальної правдоподібності.

Одним з найпоширеніших методів непараметричного дискримінантного аналізу є квадратичне прийняття рішень (QDA). QDA не передбачає, що розподіл ймовірностей зразків різних класів є нормальним. У цьому випадку дискримінантна функція QDA має вигляд

$$f(x) = (x - mu_1)^T sigma_1^(-1) (x - mu_1) - (x - mu_2)^T sigma_2^(-1) (x - mu_2) \quad (2.4)$$

де mu_1 і mu_2 - математичні очікування зразків класів 1 і 2 відповідно;

sigma_1 і sigma_2 - матриці коваріацій зразків класів 1 і 2 відповідно.

Дискримінантний аналіз може бути ефективним для завдань класифікації, в яких дані розділені лінійно або квадратично. При цьому може бути малоефективним для завдань класифікації, в яких дані не розділені лінійно або квадратично, а також може бути нестійким до шуму в даних.

2.1.3 Метод опорних векторів

Метод опорних векторів (Support Vector Machines, SVM) - це потужний алгоритм машинного навчання, який може бути використаний для класифікації та регресії. У задачі класифікації SVM шукає гіперплощину, що максимізує відстань між класами. Це дозволяє SVM бути ефективним для завдань класифікації, у яких дані розділені нелінійно. Така гіперплощина називається опорною. Клас нового зразка при цьому визначається як клас того боку гіперплощини, на якому знаходиться зразок (рис. 2.3).

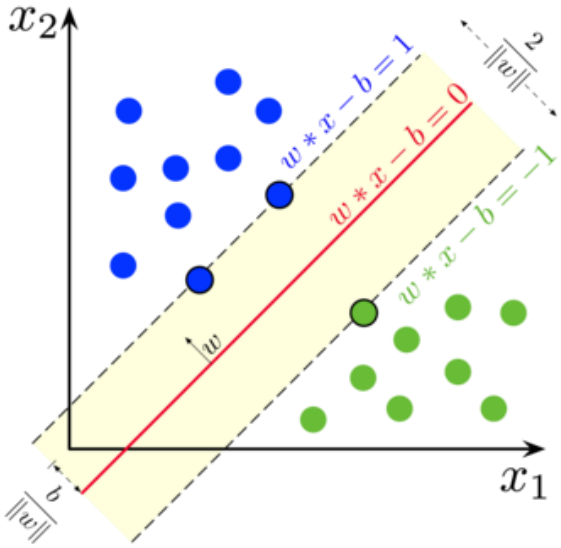


Рисунок 2.3. Максимально роздільна гіперплощина та межі для SVM бінарної класифікації.

Опорні вектори - це екземпляри даних, які лежать на межі рішення, тобто знаходяться найближче до гіперплощини (рис. 2.3). Метод приділяє особливу увагу опорним векторам, оскільки вони визначають положення опорної гіперплощини та впливають на її розташування. Кількість опорних векторів може бути різною для різних наборів даних, - якщо дані розділені лінійно, то кількість опорних векторів буде невеликою, якщо ж дані розділені нелінійно, то кількість опорних векторів може бути великою.

З математичної точки зору метод опорних векторів передбачає знаходження опорних векторів і параметрів гіперплощини, які мінімізують суму квадратів відстаней від опорних векторів до гіперплощини.

Метод опорних векторів може використовувати м'яке або жорстке розділення. В жорсткому розділенні алгоритм намагається ідеально розділити дані, але це може призвести до перенавчання. У м'якому розділенні дозволяється деяка помилка класифікації, щоб підвищити загальну здатність алгоритму до генералізації.

Метод опорних векторів використовує концепцію ядра, щоб вирішити нелінійно роздільні дані в ознаковому просторі. Ядра перетворюють дані в більш високорозмірний простір, де може бути знайдено опорна гіперплощина.

У SVM використовується параметр регуляризації "C", який контролює величину допустимих помилок класифікації. Зменшення "C" може допомогти зробити модель більш узагальненою, але й може призвести до менш точного розділення.

Метод опорних векторів ефективен до застосування у просторах великої розмірності, таких як текстові дані. З використанням ядер, SVM може розділяти дані, які не є лінійно роздільними в ознаковому просторі, м'яка регуляризація, при цьому, дозволяє підтримувати стійкість до перенавчання.

2.1.4 Дерева рішень

Дерева рішень (англ. decision tree, DT) - це метод машинного навчання, який використовується для вирішення завдань класифікації та регресії. У контексті задачі класифікації, дерево рішень розглядається як структура, що має деревоподібну форму з вузлами та гілками. Кожен вузол є розподілом на певну характеристику об'єкта, а гілки що виходять з вузлів, є результатом цього розподілу (рис. 2.4).

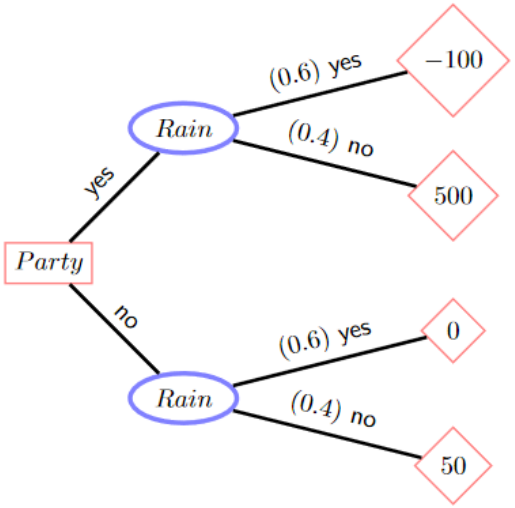


Рисунок 2.4. Приклад моделі дерева рішень.

Структура дерева рішень має наступні складові:

1. Корінь дерева, тобто початковий вузол, який представляє всі дані навчання.

2. Внутрішні вузли, що мають дочірні вузли. Кожен внутрішній вузол відповідає тесту на певну ознаку.

3. Гілки, що з'єднують вузли дерева відповідно результату тесту на певну ознаку.

4. Листя, які є вузлами без дочірніх вузлів, що представляють прогнозовані класи або значення вихідної змінної.

Основними критеріями розділення у дереві рішень є ентропія та коефіцієнт Джині.

Ентропія є мірою невизначеності у системі. У даному випадку, алгоритм буде шукати розділення, яке мінімізує ентропію.

Коефіцієнт Джині вимірює ймовірність того, що елементи, вибрані випадковим чином, будуть неправильно класифіковані. Алгоритм буде шукати розділення, яке мінімізує цей коефіцієнт.

Кількісна величина, що характеризує розгалуження значень ознаки, вимірюється за допомогою моделі ентропії Шенона, яка визначає кількісну міру неоднорідності елементів у множині:

$$H(t,D)=-\Sigma[\ P(t=i)*log2(P(t=i))\ ]i\in l \quad (2.5)$$

де $P(t=i)$ – ймовірність того, що цільова ознака t належить класу i;

l – різні класи цільової ознаки у наборі даних D.

Мірою інформативності ознаки у дереві рішень, є приріст інформації. Так як модель Шенона визначає ентропію для множини даних D відносно цільової ознаки, задля формального визначення приросту інформації, знадобляться наступні формули:

$$rem(d,D)=\Sigma|Dd=l||D|*H(t,l\in \text{рівні}(d)Dd=l) \quad (2.6)$$

Наступним кроком визначимо приріст інформації:

$$IG(d,D)= H(t,D)- rem(d,D) \quad (2.7)$$

Дерева рішень використовуються як для класифікації, так і для прогнозування неперервних величин [1].

Складові процесу побудови дерева рішень наступні:

- алгоритм вибирає ознаку, яка найкраще розділяє навчальні дані на основі критерію, такого як ентропія чи коефіцієнт Джині;

- дерево рішень розгалужується на дві або більше гілки відповідно до значень вибраної ознаки;

- процес вибору ознак та розгалуження повторюється для кожного нового вузла до тих пір, поки не будуть визначені всі класи або не буде досягнута певна умова зупинки.

Дерева рішень можуть бути дуже ефективними для завдань класифікації, тому що можуть моделювати складні нелінійні залежності в даних, що робить їх дуже ефективними для задач, де лінійні моделі можуть виявитись недостатньо ефективними.

2.1.5 Метод найближчих сусідів

Метод найближчих сусідів (англ. K-Nearest Neighbor, KNN) є одним із найпростіших та ефективніших методів у машинному навчанні,

використовується для задач класифікації та регресії. Розглянемо K-Nearest Neighbor детальніше у контексті задачі класифікації.

Метод найближчих сусідів використовує принцип "схожості": об'єкти, що подібні за характеристиками, мають високу ймовірність належати до одного класу. Коли подається новий об'єкт для класифікації, алгоритм порівнює його з набором тренувальних прикладів і визначає його клас на основі "голосування" k найближчих сусідів (рис. 2.5).

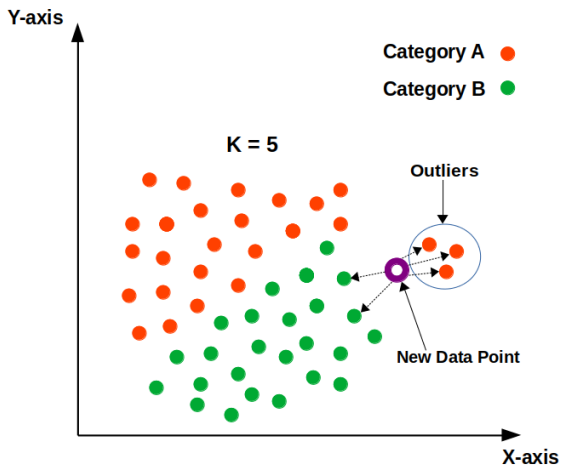


Рисунок 2.5. Приклад класифікації об'єкта за допомогою методу K-Nearest Neighbor.

У загальному вигляді алгоритм методу K-Nearest Neighbor можна записати в такий спосіб:

$$a(x) = \text{argmax}_{y \in Y} \Sigma[yi; x=y] \omega(i,x) mi=1 \qquad (2.8)$$

де ω (i; x) - задана вагова функція, яка оцінює наскільки сильно і-й сусід впливає на об'єкт x [12].

При використанні методу найближчих сусідів важливо вибрати правильну метрику відстані, що визначає відстань між двома зразками. Існує багато різних метрик відстані, які можна використовувати для методу найближчих сусідів, розглянемо деякі з найпоширеніших:

- евклідова відстань: це найпростіша метрика відстані, яка визначається як корінь квадратний з суми квадратів різниць між відповідними характеристиками двох зразків;

- манхеттенська відстань, яка визначається як сума абсолютних різниць між відповідними характеристиками двох зразків;

- косинусна відстань, що визначається як кут між векторами характеристик двох зразків.

При використанні метода найближчих сусідів, обирається кількість найближчих сусідів "k", яка буде використовуватись моделлю при роботі з даними. Зазвичай використовують непарні значення, щоб уникнути розбіжностей при голосуванні.

Для визначення метрики відстані між об'єктами, зазвичай використовується евклідова відстань, але можна використовувати інші метрики.

Метод найближчих сусідів може бути використаний для вирішення завдань класифікації, в яких дані нелінійно розділені, тобто що він може бути ефективним навіть для наборів даних, які не можуть бути розділені лінійною функцією.

KNN застосовується в різних галузях, включаючи рекомендаційні системи, розпізнавання образів та аналіз даних. Серед недоліків можна відмітити що він може бути нестійким до шуму в даних та неефективним для завдань класифікації з великими наборами даних.

Підсумовуючи можна сказати, що метод найближчих сусідів є простим, але потужним методом класифікації, який використовує концепцію схожості між прикладами для визначення класів нових екземплярів.

2.1.6 Наївний баєсів класифікатор

Наївний баєсів класифікатор - це алгоритм машинного навчання, який ґрунтується на теоремі Баєса, за допомогою якої визначається, ймовірність того, що зразок належить до певного класу, зі зміною ймовірності в залежності від характеристик зразка.

Основна ідея наївної моделі полягає в тому, що всі атрибути характеризуючі об'єкт, вважаються незалежними. Це "наївне" припущення, оскільки в реальних даних атрибути часто пов'язані, але це спрощення полегшує розрахунки. Для класифікації об'єкта, визначаються умовні ймовірності того, що об'єкт належить до кожного класу та використовується теореми Баєса для розрахунку цих ймовірностей.

Припущення про незалежність характеристик зразка є найважливішим припущенням наївної моделі Байєса. Це припущення означає, що ймовірність того, що характеристика буде мати певне значення, не залежить від значень інших характеристик. Наприклад, якщо характеристики зразка є незалежними, то ймовірність того, що електронний лист від відправника з доменом "@gmail.com" буде спамом, не залежить від того, чи є тема електронного листа "Пропозиція".

Припущення про гомогенність розподілу ймовірностей характеристик також є важливим припущенням наївної моделі Байєса. Це припущення означає, що розподіл ймовірностей характеристик є однаковим для всіх класів. Наприклад, якщо розподіл ймовірностей характеристик є гомогенним, то ймовірність того, що відправник електронного листа з доменом @gmail.com буде спамом, буде однаковою для всіх електронних листів від відправників з доменом @gmail.com.

Розглянемр математичну складову наївної моделі Баєса, - нехай є вибірка об'єктів X, що з n характеристиками, та множина класів Y. Для кожного x ∈ X обираємо клас, з максимальною ймовірністю приналежності до нього:

$$c = \text{argmax}_{y \epsilon Y} P(Y|x) \quad (2.9)$$

Перейдемо до умовних імовірностей, використовуючи формулу Баєса:

$$P(Y|x) = P(x|Y)P(Y)P(x) \quad (2.10)$$

Об'єкти з вибірки x ∈ X описуються n незалежними ознаками. Перепишемо формулу в наступному вигляді:

$$c = \text{argmax}_{y \epsilon Y} P(Y) \Pi P(xi|Y) ni = 1 \quad (2.11)$$

Таким чином, потрібно обчислити P (y) - ймовірність, приналежності об'єкта до класу y, та $P(xi|Y)$ - ймовірність приналежності об'єкта xi до класу y. Обчислення цих параметрів і описує тренування моделі. [14].

Параметри наївного баєсіва класифікатора, а саме апріорні та апостеріорні ймовірності, можуть бути навчені за допомогою методів:

- метод максимальної правдоподібності передбачає знаходження таких значень параметрів, які максимізують ймовірність того, що дані, на яких навчається модель, були отримані з моделі.

- метод баєсових ймовірностей передбачає знаходження таких значень параметрів, які мінімізують ймовірність того, що дані, на яких навчається модель, були отримані з будь-якої іншої моделі, крім наївної моделі Баєса.

Перевагою наївного баєсівського класифікатора є мала кількість даних, необхідних для навчання, оцінки параметрів і класифікації [13].

Наївна модель Баєса залишається популярним методом, особливо для завдань з аналізу тексту та обмеженими обсягами даних, де її простота та швидкодія стають значущими перевагами.

2.1.7 Ансамблеві методи

Ансамблеві методи машинного навчання - це клас алгоритмів, які об'єднують декілька базових моделей для отримання кращої прогнозної чи класифікаційної здатності, ніж може забезпечити кожен окремий алгоритм. Основні ідеї ансамблевих методів включають узгоджене використання прогнозів, що отримані від декількох моделей, та здатність ансамблю компенсувати слабкі сторони окремих алгоритмів, тобто різні моделі можуть доповнювати одна одну та компенсувати свої недоліки. Цей підхід особливо корисний, коли немає одного "ідеального" алгоритму для вирішення конкретної задачі.

Кожна базова модель тренується на підмножині даних або з використанням різних технік для введення різноманітності між моделями.

Зазвичай використовується підмножина даних, вибрана з використанням методу "bootstrap", тобто "з поверненням". Після тренування кожна базова модель робить прогноз для нових даних. Якщо це задача класифікації, то часто використовується голосування більшості або зважене голосування для визначення кінцевого класу. Потім прогнози базових моделей об'єднуються, і кінцевий результат формується за допомогою якоїсь агрегаційної стратегії. Це може бути усереднення, голосування, зважене голосування або інші методи, залежно від типу завдання та особливостей моделей.

Основні принципи створення ансамбю моделей наступні:

- різноманітність моделей: щоб ансамбль був ефективним, моделі повинні бути достатньо різноманітними, це означає, що вони мають різні особливості й помилки на різних частинах даних;

- слабкі моделі: базові моделі ансамблю можуть бути відносно простими, наприклад, неглибокі дерева чи лінійні моделі. Основна умова - вони повинні бути кращими за випадковий вибір;

- великі ансамблі: зазвичай ансамблі ефективніші, коли вони включають велику кількість моделей, поки це прийнятно з огляду на ефективність використання ресурсів.

Ансамблеві моделі можуть дозволяти аналізувати важливість кожної моделі в ансамблі та здійснювати оптимізацію за необхідності. Також може застосовуватися важливість ознак для оцінки важливості окремих ознак в задачі.

До найпопулярніших ансамблевих методів можна віднести:

- Bagging (Bootstrap Aggregating): використовує декілька копій одного й того ж алгоритму, тренуючи кожну модель на випадковій підмножині даних з поверненням. Прогнози потім узгоджуються, часто шляхом усереднення для регресії або голосування більшості для класифікації.

- Random Forests: є різновидом Bagging для рішучих дерев. Кожне дерево навчається на випадковій підмножині даних та випадковій підмножині

ознак. Під час прогнозу кожне дерево вносить свій внесок, і результат узгоджується.

- Boosting: ітеративний метод, в якому кожна наступна модель звертає увагу на помилки, зроблені попередніми моделями. AdaBoost, Gradient Boosting (GBM, XGBoost, LightGBM) - це приклади алгоритмів Boosting. Кожен новий алгоритм фокусується на об'єктах, які попередні моделі класифікували неправильно.

- Stacking (Stacked Generalization): включає в себе навчання другого рівня моделі, яка використовує прогнози першого рівня моделі як вхідні дані. Вона вивчає, як об'єднувати або зважувати прогнози, щоб покращити загальні результати.

- Voting Classifiers: голосування використовується для класифікації і полягає в отриманні прогнозів від кількох моделей та виборі класу, який набрав найбільше голосів. У випадку класифікації може використовуватися жорстке голосування (більшість голосів), м'яке голосування (врахування ваги кожного голосу), або інші варіанти

Популярним різновидом бустінгу є градієнтний бустінг, який використовує градієнт функції втрати для навчання кожної нової моделі. У градієнтному бустінгу на кожному кроці будується модель, яка апроксимує градієнт втрати відносно прогнозів поточної моделі (наприклад, середньоквадратична помилка для регресії або логарифмічна втрата для класифікації).

Основні етапи градієнтного бустінгу наступні:

1. Визначення початкової моделі: починаємо з простої моделі, яка може бути константою або простим регресором (наприклад, середнє значення цільової змінної для задачі регресії).

2. Обчислення градієнта втрат по відношенню до прогнозів поточної моделі. Градієнт представляє собою величину та напрямок, яким слід коригувати прогнози.

3. Навчання нової моделі: нова модель навчається передбачати невеликі коригування для усунення частини неузгідненостей поточної моделі. Ваги даних можуть бути змінені так, щоб найбільша увага приділялась точкам даних, де поточна модель робить більше помилок.

4. Оновлення прогнозів: прогнози оновлюються, додаючи до них вагу нової моделі, помножену на швидкість навчання ("learning rate"). Швидкість навчання контролює, наскільки сильно впливає кожна нова модель.

5. Повторення процесу: процес навчання нових моделей повторюється. Кожна наступна модель намагається скоригувати помилки попередньої, ансамблю дозволяється концентруватися на складних аспектах даних, забуваючи про більш прості аспекти.

6. Узагальнення: зазвичай градієнтний бустінг продовжується до тих пір, поки не буде досягнуто певного кількісного критерію, наприклад, досягнення заданої якості або фіксованої кількості моделей.

Ансамблеві методи є потужним методом машинного навчання, який може бути ефективним для вирішення різноманітних завдань, часто використовуються для задач класифікації. Прикладами використання ансамблевих методів можуть бути виявлення шахрайства, класифікація зображень, визначення емоцій на основі тексту тощо.

2.1.8 Штучні нейронні мережі

Штучні нейронні мережі (англ. Artificial neural networks, ANNs) моделюють роботу людського мозку використовуючи штучні нейрони для вирішення задач класифікації, регресії та інших задач. Штучний нейрон є основною будівельною одиницею нейронних мереж, він отримує вхідні сигнали, обчислюють їх вагову суму, застосовує активаційну функцію та видає вихідний сигнал (рис. 2.6)

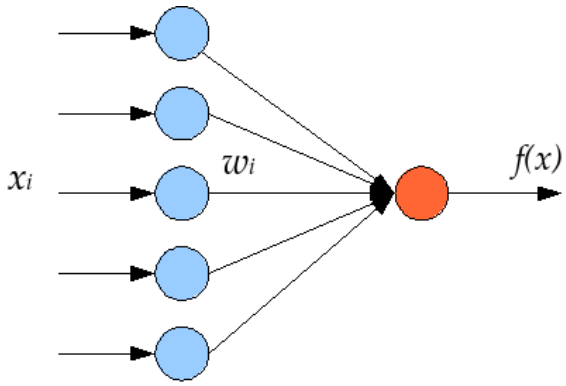


Рисунок 2.6. Схема штучного нейрона.

Нейронні мережі складаються з шарів, а саме мають вхідний шар, внутрішні (приховані) шари та вихідний шар. Вхідні дані поступають у вхідний шар нейронної мережі. Кожен з нейронів у прихованому шарі та вихідному шарі має свої ваги, які використовуються для обчислення вагової суми. До вагової суми застосовується функція активації, яка додає нелінійність та нелінійні можливості до моделі.

Під час навчання нейронної мережі параметри нейронів у мережі оновлюються таким чином, щоб зменшити похибку між прогнозами та фактичними класами зразків у наборі даних. Процес збору вхідних сигналів, обчислення вагової суми, застосування функції активації та передачі сигналів через мережу називається поширенням вперед (feedforward), яке закінчується вихідними значеннями, які можуть використовуватися для подальшого аналізу.

Для різноманітних задач застосовується функція втрат, яка оцінює, наскільки передбачені значення відповідають фактичним класам. Популярні функції втрат включають категоріальну крос-ентропію та середньоквадратичну помилку. Задача нейронної мережі - знайти такі ваги, які мінімізують функцію втрат, це робиться за допомогою алгоритмів оптимізації, таких як стохастичний градієнтний спуск.

Щоб уникнути перенавчання або недонавчання важливо налаштовувати такі гіперпараметри моделі як кількість прихованих шарів, кількість нейронів у кожному шарі та швидкість навчання, тому що вибір оптимальної архітектури

штучної нейронної мережі відповідно до набору даних є визначальним для її успішності.

Для виконання задачі класифікації штучні нейронні мережі можуть приймати різні форми, в залежності від характеристик даних, розглянемо найпоширеніші:

1. Одношарові перцептрони (Single-Layer Perceptrons, SLP): є простими нейронними мережами, які мають тільки вхідний та вихідний шар. Вони часто використовуються для бінарної класифікації, де вихідний шар складається з одного нейрона з функцією активації "sigmoid".

2. Багатошарові перцептрони (Multi-Layer Perceptrons, MLP): складаються з кількох шарів, включаючи вхідний, один або декілька прихованих та вихідний шар. Вони здатні моделювати більш складні залежності в даних. Функції активації, такі як "ReLU" або "sigmoid", використовуються для прихованих шарів, і softmax для вихідного шару для багатокласової класифікації.

3. Зворотні нейронні мережі (Recurrent Neural Networks, RNN): мають циклічні зв'язки між нейронами, що дозволяє їм враховувати часову залежність в даних. Вони можуть бути використані для класифікації послідовностей, наприклад, в аналізі текстів чи розпізнаванні мови.

4. Згорткові нейронні мережі (Convolutional Neural Networks, CNN): призначені для обробки вхідних даних зі структурою сітки, таких як зображення. Вони використовують згорткові та пулінгові шари для ефективного виявлення локальних патернів у вхідних зображеннях. Часто використовуються для класифікації зображень.

5. Довільно направлені ациклічні графи: включають архітектури, такі як Graph Neural Networks (GNNs), які можуть пристосовуватися до структурованих даних, таких як графи. Вони знаходять застосування у класифікації об'єктів зі складною топологією.

6. Автокодери: використовуються для вивчення ефективних подань даних та відновлення вхідних даних. Вони можуть бути використані для класифікації, якщо обчислені представлення виявляються інформативними.

7. Генеративні зворотні мережі (Generative Adversarial Networks, GANs): використовують пару генератора та дискримінатора для генерації нових даних, які непомітно схожі на навчальні дані. Можуть бути використані для задач класифікації, але їх головний фокус - це генерація.

Штучні нейронні мережі широко використовуються в різних областях, таких як, розпізнавання образів, мовний аналіз та синтез, класифікація тексту, автоматичний переклад мов, обробка зображень та відео тощо. Вони представляють собою потужний інструмент для рішення різноманітних задач і їхні можливості великою мірою залежать від правильного вибору архітектури та налагодження гіперпараметрів.

2.2 Висновки до розділу 2.

У цьому розділі було розглянуто основні методи та алгоритми машинного навчання, а саме: логістична регресія, дискримінаційний аналіз, метод опорних векторів, дерева рішень, метод найближчих сусідів, наївна модель Байєса, ансамблеві методи машинного навчання та штучні нейронні мережі. Були зазначені особливості, недоліки та переваги цих методів. На наступному етапі застосуємо перелічені методи до нашого набору даних та оберемо найефективніші для подальшого дослідження.

РОЗДІЛ 3. ОЦІНЮВАННЯ ЯКОСТІ РОБОТИ КЛАСИФІКАТОРІВ ТА АНАЛІЗ РЕЗУЛЬТАТІВ ПРОГНОЗУВАННЯ

У першому розділі був проведений аналіз даних щодо задоволеності пасажирів авіакомпанією та виконана їх підготовка, що дозволяє перейти до етапу моделювання. У другому розділі були розглянуті основні методи та алгоритми машинного навчання, які можуть бути використані для класифікації, тепер застосуємо їх на практиці. Для побудови моделей з метою прогнозування задоволеності пасажирів авіакомпанією застосуємо наступні методи та алгоритми машинного навчання:

- логістичну регресію;
- лінійний дискримінаційний аналіз ;
- метод опорних векторів;
- дерева рішень;
- метод найближчих сусідів;
- наївний баєсів класифікатор;
- ансамблеві методи машинного навчання, так і як "bagging", "boosting" та їх різновиди;
- багатошаровий перцептрон (MLP).

Перелічені методи та алгоритми мають різний математичний апарат та підходи до класифікації, тому застосування їх на нашому наборі даних покаже які з них найбільше підходять для прогнозування задоволеності пасажирів авіакомпаній при наявній структурі даних.

3.1 Способи оцінки якості моделі

Перед тим як перейти до моделювання, оберемо підхід до визначення якості роботи обраних методів на нашому наборі даних. Використаємо для цього перехресне затверджування (Cross-validation).

Як відомо, навчання та тестування моделі на одних і тих самих даних є методологічною помилкою та призведе до хибних результатів роботи моделі.

Щоб уникнути цього використаємо "train_test_split" - функцію з бібліотеки scikit-learn, що використовується для розділення даних на тренувальний та тестовий набори для оцінки якості моделі на незалежних даних, які не використовувалися для тренування. Це ефективний та швидкий підхід за допомогою якого можна попередньо оцінити ефективність роботи моделі на наборі даних, однак для отримання більш точного та зваженого результату щодо роботи моделей яки будуть обрані для подальшого більш детального розгляду, застосуємо функції "cross_val_predict" та "cross_val_score" модулю model_selection бібліотеки scikit-learn.

Cross-validation - це метод оцінювання моделі машинного навчання, який полягає у розбитті навчального набору даних на декілька піднаборів. Навчання та оцінка моделі виконуються кілька разів, кожен раз використовуючи різні піднабори для тренування та тестування. Основна мета крос-валідації - забезпечити більш надійну оцінку загальної ефективності моделі та уникнути перенавчання.

На рис. 3.1 представлена блок-схема типового валідаційного робочого процесу під час навчання моделей.

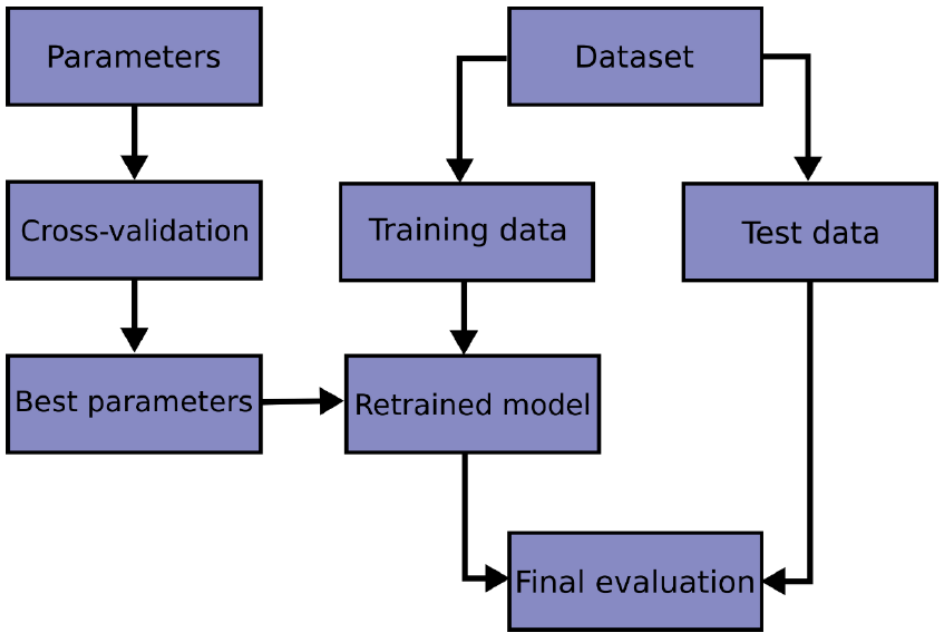


Рисунок 3.1 – Блок-схема валідаційного процесу [20]

K-Fold Cross-Validation (k-кратна перехресна перевірка) є одним з видів перехресного затверджування, який використовується для великих обсягів даних та полягає у повторному розбитті вихідної вибірки даних на k піднаборів (фолдів). Потім модель навчається на k-1 фолді, а її точність оцінюється на k-му фолді. Цей процес повторюється k разів, причому кожний фолд використовується як тестовий один раз.

Результат k-кратної перехресної перевірки - це середня точність моделі на всіх піднаборах даних.

Число k може бути будь-яким, але зазвичай воно вибирається з діапазону від 5 до 10. Більше значення k дає більш точну оцінку точності моделі, але також є більш трудомістким.

Існує кілька способів розбиття вихідної вибірки даних на піднабори (фолди). Найпоширенішим є випадковий спосіб, при цьому кожний зразок має рівну ймовірність потрапити в будь-який вибірку.

Також використовується стратифікований спосіб. При цьому вихідна вибірка даних розбивається на частини таким чином, щоб у кожному частині зберігалося відношення між різними класами. Цей спосіб особливо важливий, якщо вихідна вибірка даних містить нерівномірний розподіл класів.

Для покращення моделі машинного навчання, використаємо сітковий пошук (Grid Search). Сітковий пошук - це метод оптимізації параметрів моделі в машинному навчанні, який полягає в систематичному переборі заданого простору параметрів з метою знаходження їх найкращої комбінації. Сітковий пошук допомагає визначити оптимальні значення гіперпараметрів для моделі з точки зору її точності та ефективності.

Стандартне відхилення (Standard Deviation, STD) є статистичним показником розподілу даних, яке вимірює розсіювання значень випадкової величини відносно її математичного сподівання, тобто центру розподілу.

Стандартне відхилення STD обчислюється як квадратний корінь із дисперсії. Дисперсія, у свою чергу, обчислюється як середнє квадратичне відхилення від середнього значення.

Мале стандартне відхилення свідчить, що значення в наборі даних мало відхиляються від середнього значення, тобто дані більш концентровані. Велике стандартне відхилення, навпаки, свідчить про велику різницю між значеннями та вказує, що дані розподілені ширше.

В даній роботі для оцінки моделей машинного навчання використаємо такі метрики, як accuracy, precision та recall.

Перед тим як описати precision та recall, розглянемо матрицю помилок (рис. 3.2).

Рисунок 3.2. Матриця помилок.

Матриця помилок (Confusion Matrix) є інструментом машинного навчання, що використовується для визначення ефективності класифікаційних моделей. Вона виводить кількість правильних та неправильних прогнозів, розподілену за класами й зазвичай використовується у задачах класифікації.

Складові матриці помилок наступні:

- True Positive (TP): кількість прикладів, які правильно класифіковані як позитивні;

- True Negative (TN): кількість прикладів, які правильно класифіковані як негативні;

- False Positive (FP): кількість прикладів, які неправильно класифіковані як позитивні;

- False Negative (FN): кількість прикладів, які неправильно класифіковані як негативні.

На основі цих значень визначимо інші метрики, що характеризують якість та ефективність моделі машинного навчання.

Точність Accuracy є однією з ключових метрик у валідації класифікаційних моделей в машинному навчанні. Вона вимірює відсоток правильних прогнозів моделі серед усіх прогнозів. Формула Accuracy виглядає наступним чином:

$$accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (3.1)$$

де:

TP - кількість правильно класифікованих позитивних даних

TN - кількість правильно класифікованих негативних даних

FP - кількість помилково класифікованих позитивних даних

FN - кількість помилково класифікованих негативних даних

Accuracy є загальною мірою ефективності моделі машинного навчання з класифікації. Вона показує, скільки даних модель правильно класифікувала в цілому.

Точність Accuracy є важливою мірою ефективності для завдань у яких важливо, щоб модель правильно класифікувала якомога більше даних. Наприклад, для завдань виявлення захворювань важливо, щоб модель правильно діагностувала максимальну кількість пацієнтів з хворобою. Інтерпретація Accuracy є досить простою, - це відсоток правильних прогнозів відносно загальної кількості прогнозів. Чим вище точність, тим краще модель справляється з класифікацією.

Точність Precision вимірює відсоток правильно класифікованих позитивних прикладів серед усіх прикладів, які модель визначила як позитивні. Формально, Precision визначається за формулою:

$$precision = TP / (TP + FP) \quad (3.2)$$

де:

TP - кількість правильно класифікованих позитивних даних

FP - кількість помилково класифікованих позитивних даних

Precision показує, скільки даних з тих що модель класифікувала як позитивні, дійсно є позитивними. Точність Precision є важливою мірою ефективності для завдань у яких важливо, щоб модель не помилялася в класифікації позитивних даних.

Високий показник Precision вказує на те, що модель робить мало помилок, класифікуючи негативні приклади як позитивні. Однак Precision чутлива до кількості False Positives, і показник Precision може бути завищений, якщо FP є великим при маленькому значенні TP. При цьому, коли важливо, щоб модель правильно класифікувала якомога більше даних в цілому, більш важливою мірою ефективності є точність Accuracy.

Точність Precision використовується разом з повнотою Recall для отримання більш повної карти про ефективність моделі.

Повнота Recall, також відома як Sensitivity або True Positive Rate, вимірює відсоток істиннопозитивних прогнозів відносно всіх фактичних позитивних прикладів. Формально, Recall визначається наступним чином:

$$Recall = TP / (TP + FN) \quad (3.3)$$

де:

TP - кількість правильно класифікованих позитивних даних

FN - кількість помилково класифікованих негативних даних

Recall вказує на те, яку частину всіх позитивних прикладів модель виявила. Високий Recall означає, що модель здатна ефективно виявляти позитивні приклади, і це потрібно в ситуаціях коли важливо уникати помилок у класифікації позитивних елементів.

Показник Recall є чутливим до False Negatives і він може бути низьким, якщо FP має велику величину, а TP малу.

Повнота Recall та точність Precision є двома протилежними мірами ефективності моделей машинного навчання. Збільшення повноти зазвичай призводить до зниження точності, і навпаки.

Окрім того, слід зазначити що Recall не завжди є найкращою мірою ефективності. Наприклад, для завдань, для яких важливо, щоб модель не помилялася в класифікації негативних даних, більш важливою мірою ефективності може бути specificity, яка показує, скільки негативних даних модель правильно класифікувала.

F1-score (F-міра) є метрикою у валідації класифікаційних моделей машинного навчання, яка об'єднує точність Precision та повноту Recall в один числовий показник. Ця метрика використовується для вирішення проблеми балансу між Precision та Recall, що може виникати в ситуаціях, коли одна з цих метрик важливіша за іншу у конкретній задачі.

F1-score визначається наступним чином:

$$\text{F1-score} = 2 * \text{precision} * \text{recall} / (\text{precision} + \text{recall}) \quad (3.4)$$

Показник F1- score приймає значення між 0 та 1. Він досягає свого максимального значення, коли й точність й повнота максимальні. Це означає, що F1- score найкраще працює для ситуацій, де нам потрібен баланс між точністю та повнотою, і велике значення цього показника вказує на те, що модель добре справляється із забезпеченням якісних прогнозів зі збалансованою точністю та повнотою. Тобто F1-score є мірою того, наскільки добре модель класифікує як позитивні, так і негативні дані. Вона досягає свого максимального значення 1, коли модель класифікує всі дані правильно, і 0, коли модель класифікує все неправильно.

AUC-ROC (Area Under the Receiver Operating Characteristic Curve) є метрикою що використовується для оцінки якості класифікаційних моделей, зокрема у контексті бінарної класифікації.

ROC-крива - це графік, який відображає відношення між чутливістю (True Positive Rate) та специфічністю (False Positive Rate) при різних порогах відсічення. Цей графік показує, як змінюється здатність моделі розподіляти правильно позитивні та правильно негативні приклади при зміні порога відсічення.

AUC-ROC вимірює площу під ROC-кривою і надає кількісну оцінку якості моделі. Значення AUC-ROC може знаходитися в діапазоні від 0 до 1, де 0 вказує на повну неспроможність моделі, а 1 - на найвищу якість моделі. Ідеальна модель, це якщо ROC-крива надходить до верхнього лівого кута графіка, тобто коли AUC-ROC буде близьким до 1, що свідчить про високу якість моделі. Якщо ROC-крива апроксимується до діагоналі, тобто AUC-ROC буде близьким до 0.5, це буде свідчити про випадковий вибір класів.

Метрика AUC-ROC дозволяє зробити порівняння моделей та визначити, яка краще вирішує завдання бінарної класифікації. AUC-ROC може бути особливо корисною в ситуаціях, коли класи незбалансовані, оскільки оцінюється здатність моделі розрізняти між позитивними та негативними прикладами, незалежно від пропорцій класів.

Для відображення результатів роботи обраних моделей машинного навчання також була використана функція mean() яка застосовується для обчислення середнього значення для масивів числових даних.

3.2 Аналіз результатів моделювання

Розглянувши засоби забезпечення якості застосування та метрику оцінки моделей, перейдемо до моделювання. Для цього використаємо бібліотеку машинного навчання Scikit-learn, яка надає великий вибір моделей машинного навчання для задачі класифікації та застосуємо моделі з різними підходами до обробки даних та передбачення, використовуючи базові налаштування, щоб обрати моделі які краще за всього підходять для роботи з нашим набором даних.

Результат роботи моделей за показником точності Accuracy та стандартного відхилення значень відносно середнього значення STD, представлений на Таблиці 3.1.

Таблиця 3.1

Показники Accuracy та STD застосованих моделей машинного навчання

	Accuracy	STD
LogisticRegression	0.8731533612434436	0.49267950627668505
LinearDiscriminantAnalysis	0.8700255040662144	0.49296795799829746
Support Vector Machines	0.6684471392137048	0.45040952616158697
Linear Support Vector Machines	0.8461094268803234	0.49811453620749724
SGDClassifier	0.8305663827534767	0.4963540652898263
KNeighborsClassifier	0.7410134257254223	0.48970646256770034
GaussianNB	0.8609787786920745	0.49193678481106645
DecisionTreeClassifier	0.9469226697464029	0.49603162398636697
GradientBoostingClassifier	0.9412444059477407	0.49397253270702757
RandomForestClassifier	0.9633800105865935	0.4923425853829918
ExtraTreesClassifier	0.9612626918820076	0.4933855872626632
HistGradientBoostingClassifier	0.9617920215581541	0.49356614868729465
LGBMClassifier	0.9629950435493961	0.4924951592678845
XGBClassifier	0.962706318271498	0.49370568085427613
BaggingClassifier	0.9590491314181223	0.493258475639626
AdaBoostClassifier	0.927529955247582	0.49476593452722667
MLPClassifier	0.9282517684423272	0.4956443549297624

Використання різних підходів до обробки даних та прогнозування, які реалізовані у моделях представлених в таблиці, показали що найкращі результати отримали наступні моделі:

- DecisionTreeClassifier: класифікатор, який використовує дерево рішень яке представляє собою ієрархічну структуру, в якій кожен внутрішній вузол представляє тест для оцінки значення однієї з ознак, а кожен листок відповідає конкретному класу або значенню регресії. Модель приймає рішення, переходячи від кореневого вузла до листа в залежності від значень ознаки вхідних даних.

- GradientBoostingClassifier: класифікатор, що використовує градієнтний бустінг - ансамблевий метод, в якому будується послідовність слабких моделей (зазвичай дерев рішень) та відбувається їх об'єднання, з покращенням моделі на кожному кроці;

- Random Forest Classifier використовує ансамблевий метод машинного навчання на основі дерев рішень. Випадковий ліс (Random Forest) об'єднує

результати багатьох дерев рішень, що навчаються незалежно, для покращення стійкості та ефективності моделі.

- ExtraTreesClassifier також належить до сімейства ансамблевих методів, методу градієнтного бустінгу та випадкових лісів, проте використовує більше випадковості при формуванні кожного дерева ніж Random Forest Classifier;

- HistGradientBoostingClassifier використовує градієнтний бустінг з гістограмними методами для вирішення завдань класифікації. Головна відмінність від звичайного градієнтного бустінгу в даному випадку полягає у використанні гістограм для швидшого обчислення градієнтів та покращення ефективності.

- LGBMClassifier використовує алгоритм градієнтного бустінгу з градієнтним збільшенням світла. LightGBM - це бібліотека для градієнтного бустінгу, розроблена Microsoft, яка надає високопродуктивні та ефективні інструменти для навчання моделей на великих наборах даних.

- XGBClassifier представляє собою класифікатор, який використовує алгоритм градієнтного бустінгу з градієнтним підвищенням xgboost, що расшифровується як xtreme gradient boosting;

- BaggingClassifier є ансамблевим методом, що використовує техніку "мішків" (bagging), тобто ансамбль випадкових підвибірок даних, результати які об'єднуються для отримання кінцевого прогнозу.

Отже, бачимо що моделі які використовують дерево рішень та ансамблеві методи показали кращі результати у порівнянні з іншими методами. Ці моделі мають найвищі показники точності Accuracy та найбільше підходять для роботи з нашим набором даних.

Через те, щопоказники Accuracy перелічених моделей відрізняються незначною мірою, на них може вплинути фактор випадковості розбиття набору даних на навчальну та тестову частини. Щоб уникнути впливу цього фактору, перед тим як обрати моделі для подальшого більш детального розгляду, якість роботи перелічених моделей була перевірена з використанням методу перехресного затвердження (cross-validation). Результати показали що

найбільшу точність Accuracy у порівнянні з іншими моделями мають RandomForestClassifier, LGBMClassifier та XGBClassifier.

Таким чином, обираючи модель для прогнозування задоволеності пасажірів авіакомпанією, розглянемо більш детально три моделі з найкращими показниками точності Accuracy, а саме RandomForestClassifier, LGBMClassifier та XGBClassifier.

3.2.1 Модель Random Forest Classifier

RandomForestClassifier використовує алгоритм класифікації, який входить до класу ансамблевих методів, відомий як випадковий ліс, - Random Forest. Random Forest Classifier є ансамблем дерев рішень, що комбінує прогнози кількох дерев для покращення точності та стійкості моделі. Тобто використовується техніка, що об'єднує прогнози кількох базових моделей для отримання більш точного та стійкого результату. Кожне дерево при цьому навчається на випадковій підмножині даних та випадковій підмножині ознак, при цьому те що дерева навчаються на різних наборах даних дозволяє уникнути перенавчання.

Для тренування моделі Random Forest Classifier використовується bootstrap, під яким у даному випадку розуміється ітерація випадкової вибірки даних з повторенням, що допомагає створити різні підвибірки для кожного дерева. Прогнози кожного дерева об'єднуються шляхом голосування, - кілька дерев голосують для кожного прикладу, а клас, що набирає найбільше голосів, стає прогнозом ансамблю.

Розглянемо основні гіперпараметри RandomForestClassifier:

- n_estimators: кількість дерев рішень у лісі. Чим більше, тим краща стійкість та ефективність, але це призводить до збільшення обчислювальних витрат. Якщо параметр не вказаний, використовується значення 100.

- criterion: функція для вимірювання якості розбиття в деревах, може мати значення "gini", "entropy" та "log_loss". "gini" вимірює ймовірність

помилкової класифікації екземпляра даних, обраного випадковим чином. Для вузла з множиною даних, "gini", розбиття обчислюється як 1 мінус сума квадратів ймовірностей кожного класу. "entropy" вимірює невизначеність у системі. Для вузла з множиною даних, "entropy" обчислюється як сума ймовірностей класів, помножених на їхні логарифми з основою 2 та змінені зі знаком мінус. "log_loss" вимірює відхилення між дійсними та передбаченими ймовірностями класів. Для вузла з множиною даних "log_loss" обчислюється як сума логарифмів ймовірностей для кожного класу. Значення за замовченням даного параметру, - "gini".

- max_depth: максимальна глибина кожного дерева в лісі. Встановлення параметра обмежує глибину рекурсивного розбиття. Чим глибше дерева, тим точнішою буде модель, але вона також буде більш схильною до перенавчання. Значення за замовченням "none", - тобто дерева розгортаються, доки всі листи не міститимуть мінімальну кількість зразків min_samples_split.

- min_samples_split: мінімальна кількість зразків, необхідна для розділення вузла дерева рішень. Чим більше зразків, тим стійкішою буде модель до шуму в даних. За замовченням використовується значення 2.

- min_samples_leaf: мінімальна кількість зразків, яка повинна бути в листовому вузлі. Чим більше зразків, тим стійкішою буде модель до шуму в даних. Значення за замовченням даного параметру 1.

Проведемо дослідження роботи моделі RandomForestClassifier при різних встановлених гіперпараметрах на нашому наборі даних.

Результати роботи моделі при встановленому значенні "entropy" параметра criterion, представлені на рис. 3.2.

```
Model Accuracy: 0.962561955632549
Standard Deviation of Predictions: 0.4920587836564526

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.98      0.97     11891
           1       0.97      0.94      0.96      8890

    accuracy                           0.96     20781
   macro avg       0.96      0.96      0.96     20781
weighted avg       0.96      0.96      0.96     20781
```

Рисунок 3.2. Метрики моделі RandomForestClassifier при встановленому значенні "entropy" параметра criterion

З'ясуємо вплив, функції "log_loss" параметра criterion, на роботу моделі з нашим набором даних, результати застосування якої представлені на рис. 3.3.

```
Model Accuracy: 0.9627544391511477
Standard Deviation of Predictions: 0.49200661509673427

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.98      0.97     11891
           1       0.98      0.94      0.96      8890

    accuracy                           0.96     20781
   macro avg       0.96      0.96      0.96     20781
weighted avg       0.96      0.96      0.96     20781
```

Рисунок 3.3. Метрики моделі RandomForestClassifier при використанні функції "log_loss" параметра criterion.

Як бачимо, зміна функції вимірювання якості розбиття вузла дерева у моделі RandomForestClassifier, суттєво не вплинула на показники роботи моделі.

Дослідження шляхом сіточного пошуку впливу оптимізації інших основних гіперпараметрів моделі з метою її покращення, показали що модель з встановленими гіперпараметрами за замовченням є оптимальною, тому для

подальшого розгляду оберемо саме її. Метрики моделі RandomForestClassifier із встановленими параметрами за замовченням, представлені на рис. 3.4.

```
Model Accuracy: 0.9633800105865935
Standard Deviation of Predictions: 0.4923425853829918

Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97     11891
           1       0.97      0.94      0.96      8890

    accuracy                           0.96     20781
   macro avg       0.96      0.96      0.96     20781
weighted avg       0.96      0.96      0.96     20781
```

Рисунок 3.4. Метрики моделі RandomForestClassifier при використанні параметрів за замовченням.

Як бачимо, модель має високу точність Accuracy та Precision що свідчить про її ефективність у роботі з нашим набором даних.

Показник стандартного відхилення STD у нашому випадку вказує на досить високий рівень розсіювання значень у деяких ознаках даних, але низька доля таких прикладів у загальному обсязі даних не впливає на стабільність точності передбачення моделі. Такі заходи як збільшення кількості дерев рішень у ансамблі або зменшення їх глибини, не призводять до покращення моделі.

Використаємо метод feature_importances_ щоб дізнатись які ознаки здійснюють найбільший вплив на побудову прогнозу моделі (Рис. 3.5).

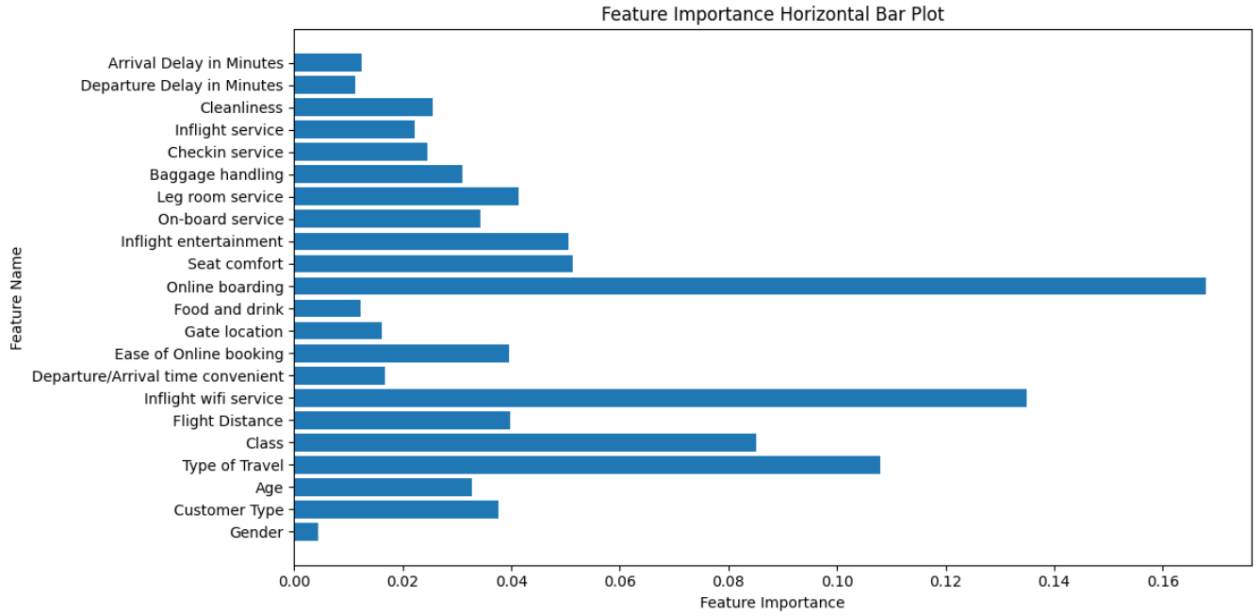


Рисунок 3.5. Гістограма важливості ознак в прогнозі моделі RandomForestClassifier

Як бачимо, найбільший вплив на передбачення цільової ознаки здійснюють оцінка пасажирами якості таких послуг як "Online boarding," "Inflight wifi service", а також ознака типу подорожі "Type of travel". Найменший вплив на передбачення здійснює ознака "Gender", тобто стать пасажира.

Крива AUC-ROC моделі представлена на рис. 3.6.

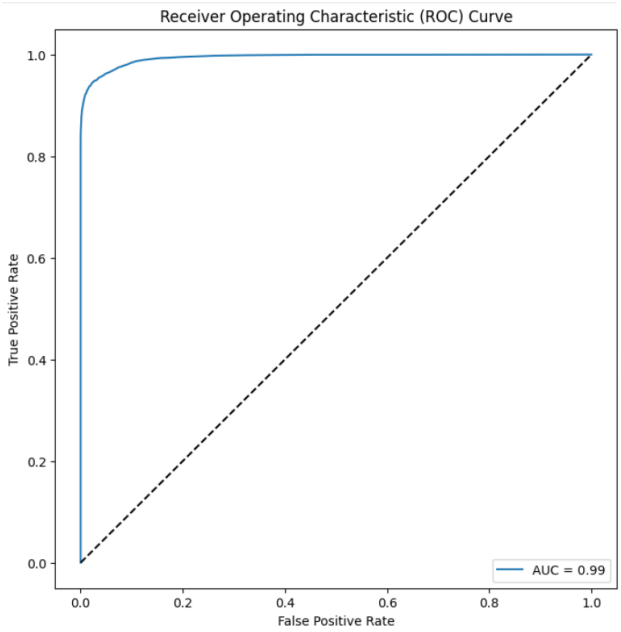


Рисунок 3.6. Крива AUC-ROC для моделі Random Forest Classifier

Показник AUC (Area Under the Curve) для кривої ROC (Receiver Operating Characteristic) має значення 0,99, це вказує на високу якість класифікації моделі, тобто 99% вірогідність правильної класифікації, що є результатом дуже високого показника True Positive Rate (TPR), як співвідношення кількості правильно класифікованих позитивних значень до загальної кількості позитивних значень та дуже низького показника False Positive Rate (FPR), як співвідношення кількості помилково класифікованих позитивних значень до загальної кількості негативних значень.

Це підтверджується матрицею помилки Confusion Matrix для моделі, яка представлена на рис. 3.7.

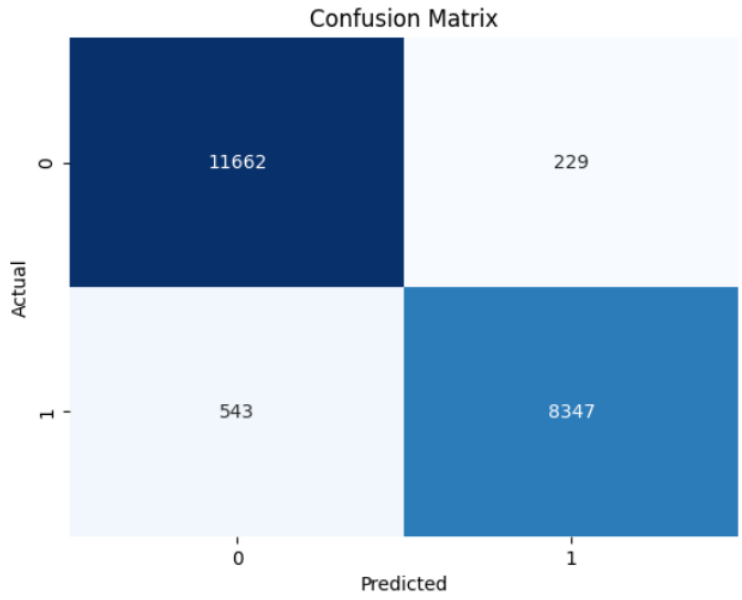


Рисунок 3.7. Матриця помилки моделі Random Forest Classifier

3.2.2 Модель LGBMClassifier

LGBMClassifier представляє собою класифікатор, який використовує алгоритм градієнтного бустінгу з градієнтним збільшенням світла (LightGBM). LightGBM - це бібліотека для градієнтного бустінгу, розроблена Microsoft, що надає високопродуктивні та ефективні інструменти для навчання моделей на великих наборах даних.

Градієнтний бустинг - це метод машинного навчання, який працює шляхом послідовного додавання нових моделей до попередньої моделі. Нові моделі додаються таким чином, щоб зменшити помилку попередньої моделі. Модель LGBMClassifier працює шляхом створення послідовності дерев рішень, які потім об'єднуються, щоб зробити прогноз.

LightGBM використовує метод на основі гістограм для створення дерев рішень, завдяки якому дані розміщуються у контейнери за допомогою гістограми розподілу, що своєю чергою використовуються для повторення, обчислення приросту та розділення даних. Алгоритм LightGBM використовує ряд специфічних функцій для зменшення розмірності, що робить його швидшим та ефективнішим.

Основні параметри LGBMClassifier:

- boosting_type: тип бустінгу, має значення за замовченням 'gbdt', що розшифровується як традиційне рішення градієнтного бустінгу. Опціональні значення 'dart' що э скороченням від "Dropouts meet Multiple Additive Regression Trees".

- n_estimators: встановлює кількість дерев рішень, які будуть створені. Чим більше дерев, тим точнішою буде модель, але вона також буде повільніша. Значення за замовченням 100.

- max_depth: максимальна глибина дерев рішень, чим глибше дерева, тим точнішою буде модель, але вона також буде більш схильною до перенавчання. За замовченням має значення "-1" що означає без обмежень.

- learning_rate: дозволяє регулювати швидкість навчання. Чим менша швидкість навчання, тим плавніше буде удосконалюватися модель, але вона також буде довше навчатися. Значення за замовченням 0,1.

- min_split_gain: мінімальне зростання, необхідне для розділення вузла дерева рішень. Чим більше мінімальне зростання, тим стійкішою буде модель до шуму в даних. Значення за замовченням 0.

Проведемо дослідження роботи моделі LGBMClassifier на нашому наборі даних при зміні типу бустінгу. Результати роботи моделі при встановленому значенні 'dart' параметра boosting_type, представлені на рис. 3.8.

```
Model Accuracy: 0.9610702083634088
Standard Deviation of Predictions: 0.4933855872626632

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.98      0.97     11772
           1       0.97      0.94      0.95      9009

    accuracy                           0.96     20781
   macro avg       0.96      0.96      0.96     20781
weighted avg       0.96      0.96      0.96     20781
```

Рисунок 3.8. Метрики моделі LGBMClassifier при встановленому значенні 'dart' параметра boosting_type

Як бачимо, зміна типу бустінгу не призвела до збільшення точності Accuracy та покращення моделі. Також, дослідження шляхом сіточного пошуку впливу оптимізації інших основних гіперпараметрів моделі з метою її покращення, показали що модель з встановленими гіперпараметрами за замовченням є оптимальною, тому для подальшого розгляду оберемо саме її. Метрики моделі LGBMClassifier із встановленими параметрами за замовченням, представлені на рис. 3.9.

```
Model Accuracy: 0.9629950435493961
Standard Deviation of Predictions: 0.4924951592678845

Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97     11871
           1       0.97      0.94      0.96      8910

    accuracy                           0.96     20781
   macro avg       0.96      0.96      0.96     20781
weighted avg       0.96      0.96      0.96     20781
```

Рисунок 3.9. Метрики моделі LGBMClassifier при використанні параметрів за замовченням.

Як бачимо, модель має високі показники точності Accuracy та Precision що свідчить про ефективність її методів класифікації у роботі з нашим набором даних.

Високий показник стандартного відхилення STD (Standard Deviation) є наслідком розсіювання значень деяких ознаках нашого набору даних. Такі заходи як збільшення кількості дерев рішень в ансамблі або зменшення глибини дерев рішень, не призводять до покращення моделі.

Застосуємо метод feature_importances_ щоб дізнатись які ознаки здійснюють найбільший вплив на побудову прогнозу моделі (рис. 3.10).

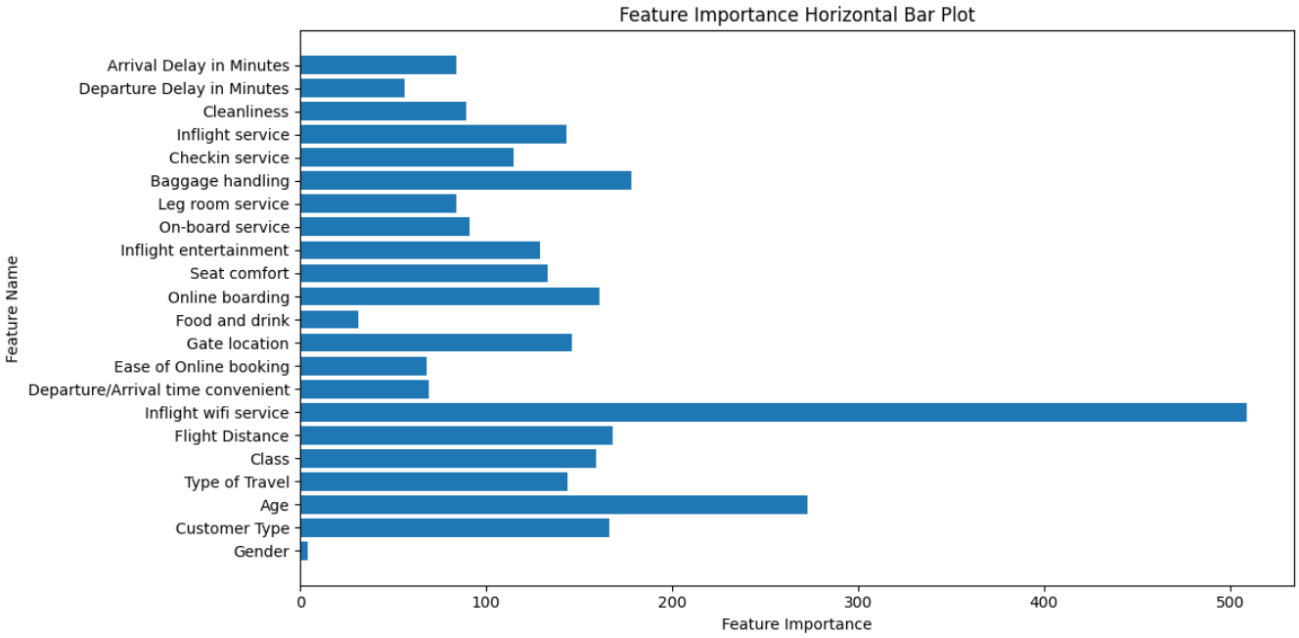


Рисунок 3.10. Гістограма важливості ознак в прогнозі моделі LGBMClassifier

Як бачимо, найбільший вплив на передбачення LGBMClassifier цільової ознаки здійснюють оцінка пасажирами якості послуги "Inflight wifi service", а також вік пасажирів, з найменшим впливом на передбачення ознаки "Gender", тобто стать пасажира.

Крива AUC-ROC моделі представлена на рис. 3.11.

Як бачимо, показник AUC для кривої ROC має значення 1, що вказує на максимально можливу для моделі передбачувальну здатність.

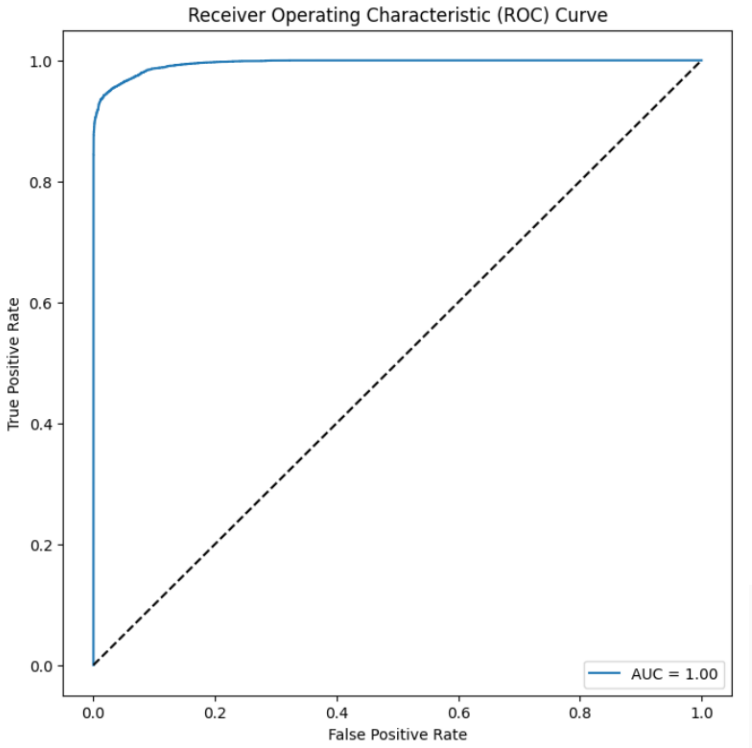


Рисунок 3.11. Крива AUC-ROC для моделі LGBMClassifier

Матриця помилки Confusion Matrix для LGBMClassifier, що представлена на рис. 3.12 свідчить про високу спроможність моделі до правильного передбачення цільової ознаки нашого набору даних з низькими показниками помилки.

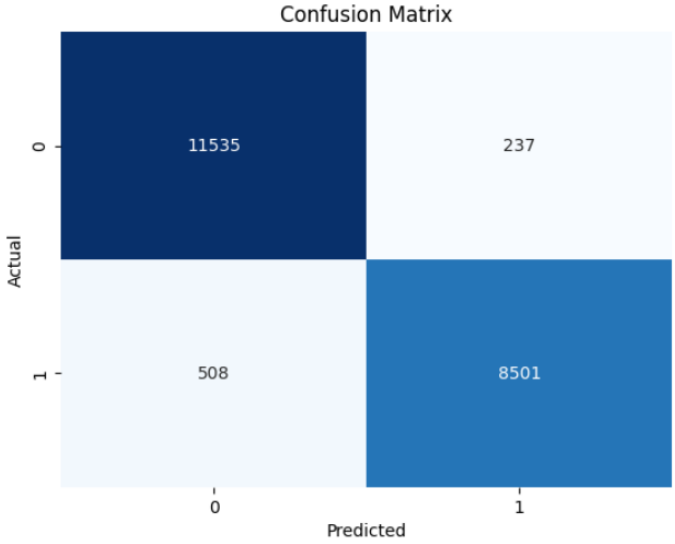


Рисунок 3.12. Матриця помилки моделі LGBMClassifier

3.2.3 Модель XGBClassifier

XGBClassifier належить до ансамблевих моделей машинного навчання та представляє собою класифікатор, що реалізує алгоритм градієнтного бустінгу з градієнтним підвищенням xgboost (eXtreme Gradient Boosting), використовуючи дерева рішень. XGBoost - це бібліотека для градієнтного бустінгу, яка широко використовується у завданнях класифікації завдяки своїй високій продуктивності та ефективності.

Модель XGBClassifier використовує логарифмічну функцію втрат, яка визначає, наскільки великою є помилка між прогнозованими і реальними значеннями, а дерева рішень підбираються таким чином, щоб мінімізувати помилку попередньої моделі. XGBClassifier підтримує регуляризацію для уникнення перенавчання яка допомагає зменшити вплив шумових або непотрібних ознак, що може бути проблемою при великій кількості ознак.

Окрім того, модель автоматично оптимізується для підвищення швидкості та зменшення використання пам'яті, що робить її зручною для обробки великих обсягів даних.

Розглянемо основні параметри XGBClassifier:

- n_estimators: встановлює кількість дерев рішень, які будуть навчатися в ансамблі. За замовченням використовується значення 100;

- max_depth: параметр що визначає максимальну глибину дерев рішень, за замовченням використовується значення 6;

- learning_rate: визначає розмір кроку, який використовується при оновленні ваг моделі після кожної ітерації навчання. Цей параметр контролює внесок кожного нового дерева у ансамбль та може впливати на здатність моделі до навчання та її стійкість до перенавчання. Значення за замовченням 0,1;

- subsample: коефіцієнт підвибору. Чим більше значення subsample, тим менше даних буде використовуватися для навчання кожного дерева рішень. За замовченням використовується значення 1;

- gamma: контролює зростання розгалуження у деревах рішень, тобто визначає мінімальне зменшення функції втрат, яке повинно відбутися, щоб зробити поділ на вузлі. Більше значення gamma призводить до більш консервативних моделей. Значення за замовченням 0.

З метою збільшення точності, проведемо дослідження роботи моделі XGBClassifier при встановленні параметра n_estimators=200 та max_depth=12, (рис. 3.13). Інші параметри моделі зі значеннями за замовченням є оптимальними для нашого набору даних.

```
Model Accuracy: 0.961214571002358
Standard Deviation of Predictions: 0.49389689009142923

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.98      0.97     11729
           1       0.97      0.94      0.95      9052

    accuracy                           0.96     20781
   macro avg       0.96      0.96      0.96     20781
weighted avg       0.96      0.96      0.96     20781
```

Рисунок 3.13. Метрики моделі XGBClassifier зі зміненими параметрами.

Як бачимо, зміни не збільшили точності моделі, дослідження оптимальних гіперпараметрів шляхом сіточного пошуку також не призвели до покращення моделі, тому для подальшого розгляду XGBClassifier оберемо модель із встановленими параметрами за замовченням, метрики застосування якої, представлені на рис. 3.14.

```
Model Accuracy: 0.962706318271498
Standard Deviation of Predictions: 0.49370568085427613

Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97     11729
           1       0.97      0.94      0.96      9052

    accuracy                           0.96     20781
   macro avg       0.96      0.96      0.96     20781
weighted avg       0.96      0.96      0.96     20781
```

Рисунок 3.14. Метрики моделі XGBClassifier при використанні параметрів за замовченням.

Як бачимо, модель має високі показники точності Accuracy та Precision що свідчить про ефективність її застосування до нашого набору даних з задачею бінарної класифікації.

Показник стандартного відхилення (STD) очікувано має такий самий рівень як і у попередніх розглянутих моделях. Такі заходи як збільшення кількості дерев рішень в ансамблі або зменшення їх глибини, не призводять до покращення моделі.

Використаємо метод feature_importances_ щоб дізнатись які ознаки здійснюють найбільший вплив на побудову прогнозу моделі (рис. 3.15).

Як бачимо, найбільший вплив на передбачення цільової ознаки здійснюють оцінка пасажирами якості послуги "Online boarding", а також тип подорожі "Type of Travel", з найменшим впливом на передбачення ознаки "Gender", тобто стать пасажира, як і в попередніх розглянутих моделях класифікації.

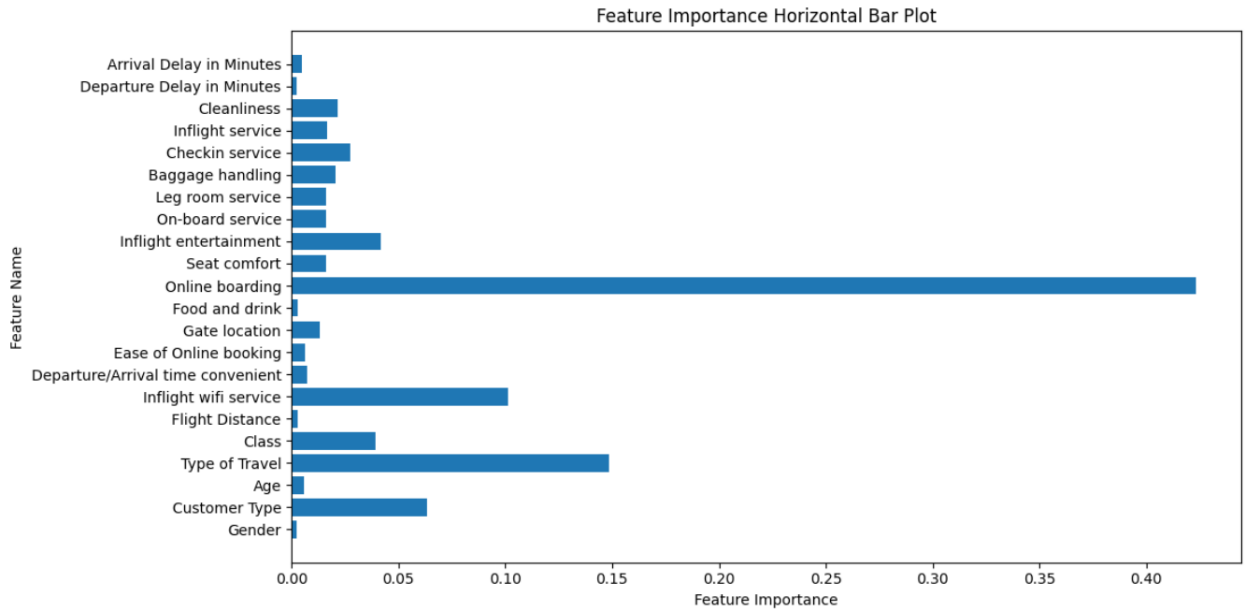


Рисунок 3.15. Гістограма важливості ознак в прогнозі моделі XGBClassifier.

Крива AUC-ROC моделі представлена на рис. 3.16, свідчить про високу передбачувальну здатність моделі, AUC = 0,99, тобто вірогідність правильної класифікації становить 99%.

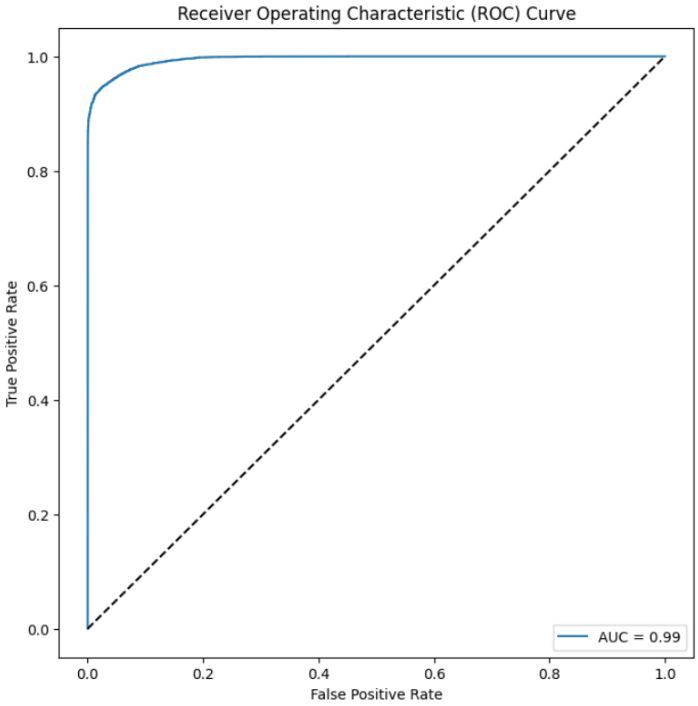


Рисунок 3.16. Крива AUC-ROC для моделі XGBClassifier

Матриця помилки Confusion Matrix для LGBMClassifier представлена на рис. 3.17 з низькими показниками помилки передбачення пояснює високий рівень показника AUC.

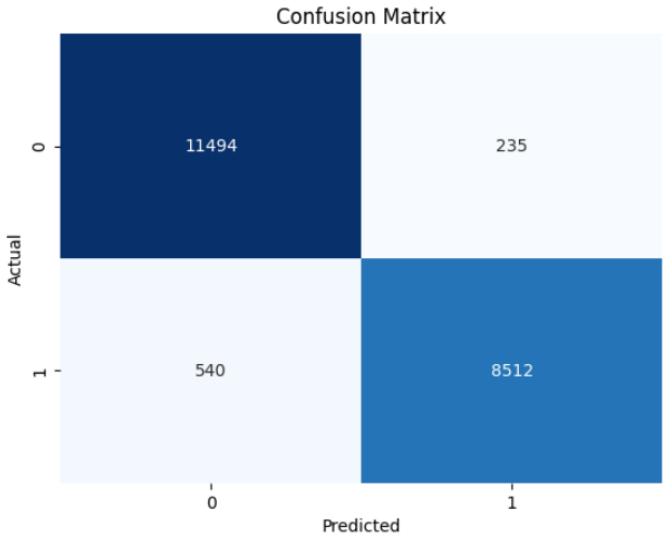


Рисунок 3.17. Матриця помилки моделі XGBClassifier

3.3 Висновки до розділу 3

В даному розділі було розглянуто 17 різних моделей машинного навчання які використовують всі основні методи та алгоритми машинного навчання для завдання класифікації. Найкращі результати у застосуванні до нашого набору даних показали моделі що застосовують ансамблеві методи з показниками точності Accuracy понад 96%. Серед цих моделей, для подальшого розгляду були обрані три моделі з найвищою точністю Accuracy, а саме RandomForestClassifier, LGBMClassifier та XGBClassifier. Аналіз показав що за показниками точності Accuracy та іншими показниками роботи у застосуванні до нашого набору даних, моделі майже не відрізняються та можуть ефективно використовуватись для передбачення задоволеності пасажирів авіакомпанією, однак найвищий результат точності передбачення показала модель RandomForestClassifier.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи, розглянута актуальність та проблематика питання прогнозування задоволеності споживачів та питань застосування методів машинного навчання щодо прогнозування задоволеності споживачів. Був проведений огляд та аналіз набору статистичних даних щодо задоволеності пасажирів авіакомпанією, у результаті якого виявлено закономірності та корисні взаємозв'язки даних, розглянуто їх особливості, проведена очистка та підготовка даних до моделювання.

Було розглянуто основні методи та алгоритми машинного навчання, зазначені особливості, недоліки та переваги цих методів. Проведено моделювання з метою прогнозування задоволеності пасажирів авіакомпанією на основі статистичних даних, аналіз та вдосконалення моделей машинного навчання. В ході дослідження було застосовано 17 різних моделей машинного навчання які використовують всі основні методи та алгоритми машинного навчання для завдання класифікації.

Для побудови моделей з метою прогнозування задоволеності пасажирів авіакомпанією були задіяні наступні методи та алгоритми машинного навчання: логістична регресія, лінійний дискримінаційний аналіз, метод опорних векторів, дерева рішень, метод найближчих сусідів, наївний баєсів класифікатор, ансамблеві методи машинного навчання, так і як "bagging", "boosting" та їх різновиди, багатошаровий персептрон.

Найкращі результати у застосуванні до набору даних показали моделі що застосовують ансамблеві методи з показниками точності Accuracy понад 96%. Серед цих моделей, для подальшого розгляду були обрані три моделі з найвищою точністю Accuracy, а саме RandomForestClassifier, LGBMClassifier та XGBClassifier. Аналіз показав що за показниками точності Accuracy та іншими показниками роботи у застосуванні до набору даних, моделі суттєво не відрізняються та можуть ефективно використовуватись для передбачення

задоволеності пасажирів авіакомпанією, однак найвищий результат точності передбачення показала модель RandomForestClassifier.

Запропонована найефективніша модель прогнозування задоволеності пасажирів авіакомпанією надає можливість авіакомпаніям у разі її використання передбачити задоволеність різних категорій пасажирів якістю наданих послуг, що дає можливість вдосконалити та оптимізувати ці послуги, а також вжити необхідні маркетингові заходи з втримання окремих груп пасажирів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kelleher J.D.. Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies / Kelleher J.D., Namee B.M, D'Arcy A. – The MIT Press, 2015. – 624 p.

2. Eremenko K. Data Science A-Z: Real-Life Data Science Exercises Included. [Електронний ресурс] - Режим доступу: https://www.udemy.com/course/datascience.

3. Карякина А. А., Мельников А. В. Сравнение моделей прогнозирования оттока клиентов интернет-провайдеров // Машинное обучение и анализ данных, 2017. Том 3, № 4. С. 250–256.

4. Airline Passenger Satisfaction [Електронний ресурс] - Режим доступу: https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction

5. Бринк Хенрик, Ричардс Джозеф, Феверолф Марк, Машинное обучение. - СПб.: Питер, 2017. -336 с.

6. Бурков Андрей. Машинное обучение без лишних слов. СПб.: Питер, 2020. - 192 с.

7. Bayesian Reasoning and Machine Learning David Barber ©2007,2008,2009,2010. [Електронний ресурс] - Режим доступу: http://web4.cs.ucl.ac.uk/staff/D.Barber/textbook/090310.pdf?roistat_visit=10865700.

8. Шакла Нишант. Машинное обучение и TensorFlow. - СПб.: Питер, 2019. - 336 с.:

9. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2015. – 400 с.

10. Машинне навчання. [Електронний ресурс] - Режим доступу: https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F

11. What are the types of machine learning? [Електронний ресурс] - Режим доступу: https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f.

12. Кероване навчання. [Електронний ресурс] - Режим доступу: https://uk.wikipedia.org/wiki/%D0%9A%D0%B5%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F

13. Гришанов К.М., Белов Ю.С. Метод классификации K-NN и его применение в распознавании / Фундаментальные проблемы науки. Сборник статей Международной научно – практической конференции 15 мая 2016г. Ч.3. Тюмень: НИЦ Аэтерна, 2016. С. 30-33.

14. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. N.Y.: Springer, 2009.

15. Albadawi S., Latif K., Kharbat F. Telecom Churn Prediction Model Using Data Mining Techniques [Bahria University Journal of Information & Communication Technologies], 2017. Vol 10. № Special Issue. P. 8–14.

16. Dziaugyte S., Mzyk M. Churn analysis — machine learning. Bloomington, 2016.

17. Cross-validation: evaluating estimator performance. [Електронний ресурс] - Режим доступу: https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation.

18. Random Forest Classifier. [Електронний ресурс] - Режим доступу: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

19. Чистяков С.П. Случайные леса: обзор // Труды Карельского научного центра РАН. 2013. №1. С. 117-136.

20. Linear and Quadratic Discriminant Analysis. [Електронний ресурс] - Режим доступу: https://scikit-learn.org/stable/modules/lda_qda.html

21. Logistic regression. [Електронний ресурс] - Режим доступу: https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

22. Support Vector Machines. [Електронний ресурс] - Режим доступу: https://scikit-learn.org/stable/modules/svm.html#support-vector-machines

23. Nearest Neighbors Classification. URL: https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-classification

20. Naive Bayes. [Електронний ресурс] - Режим доступу: https://scikit-learn.org/stable/modules/naive_bayes.html

24. Decision Trees. [Електронний ресурс] - Режим доступу: https://scikit-learn.org/stable/modules/tree.html

25. Ensembles: Gradient boosting, random forests, bagging, voting, stacking. [Електронний ресурс] - Режим доступу: https://scikit-learn.org/stable/modules/ensemble.html

26. sklearn.ensemble.RandomForestClassifier. [Електронний ресурс] - Режим доступу: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

27. Перехресне затверджування. [Електронний ресурс] - Режим доступу:https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D1%85%D1%80%D0%B5%D1%81%D0%BD%D0%B5_%D0%B7%D0%B0%D1%82%D0%B2%D0%B5%D1%80%D0%B4%D0%B6%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F

28. Стандартне відхилення. [Електронний ресурс] - Режим доступу:https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D0%B5_%D0%B2%D1%96%D0%B4%D1%85%D0%B8%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F

29. Дерева рішень. [Електронний ресурс] - Режим доступу:https://uk.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%B0_%D1%80%D1%96%D1%88%D0%B5%D0%BD%D1%8C_%D1%83_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%BC%D1%83_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%96

30. Дисперсія випадкової величини. [Електронний ресурс] - Режим доступу:https://uk.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BF%D0%B5%D1%80%D1%81%D1%96%D1%8F_%D0%B2%D0%B8%D0%BF%D0%B0%D0%B4%D0%BA%D0%BE%D0%B2%D0%BE%D1%97_%D0%B2%D0%B5%D0%BB%D0%B8%D1%87%D0%B8%D0%BD%D0%B8

31. Статистична модель. [Електронний ресурс] - Режим доступу: https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%BD%D0%B0_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C

32. Перенавчання [Електронний ресурс] - Режим доступу: https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F

33. Машинне навчання [Електронний ресурс] - Режим доступу: http://www.mmf.lnu.edu.ua/ar/1739

34. МАШИННЕ НАВЧАННЯ В ЗАДАЧАХ КІБЕРБЕЗПЕКИ [Електронний ресурс] - Режим доступу: https://learn.ztu.edu.ua/pluginfile.php/261993/mod_resource/content/1/%D0%A8%D0%86_%D0%9B-2_%D0%9C%D0%9D-1_%D0%BA%D0%BB%D0%B0%D1%81%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F_1%20%281%29.pdf

35. Машинне навчання та обробка сигналів в біомедичних електронних системах. Конспект лекцій [Електронний ресурс] - Режим доступу: https://ela.kpi.ua/bitstream/123456789/41525/1/Mashynne_navchania_Konspekt.pdf

36. АНАЛІЗ МЕТОДІВ МАШИННОГО НАВЧАННЯ З ВЧИТЕЛЕМ [Електронний ресурс] - Режим доступу: https://elartu.tntu.edu.ua/bitstream/lib/25035/2/MSNK_2018v1_Milian_N-Analysis_of_supervised_machine_51-52.pdf

37. ВСТУП В МАШИННЕ НАВЧАННЯ. ЕТАПИ ВИРІШЕННЯ ЗАВДАНЬ МАШИННОГО НАВЧАННЯ [Електронний ресурс] - Режим доступу: https://stud.com.ua/139970/informatika/osnovi_mashinnogo_navchannya

38. СЕКРЕТНІ СИЛИ МАШИННОГО НАВЧАННЯ: ОГЛЯД АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ [Електронний ресурс] - Режим доступу: https://www.zfort.com.ua/blog/cekretni-sili-mashinnogo-navchannya-oglyad-algoritmiv-mashinnogo-navchannya

39. Т.М. Басюк, В.В. .Литвин, Л.М. Захарія, Н.Е. Кунанець. Машинне навчання: Навчальний посібник призначений для студентів, що навчаються за першим (бакалаврським) рівнем вищої освіти за спеціальностями галузі знань 12 „Інформаційні технології”. - 335с.

40. МЕТОДИ КЛАСИФІКАЦІЇ МАШИННОГО НАВЧАННЯ З ВИКОРИСТАННЯМ БІБЛІОТЕКИ SCIKIT-LEARN [Електронний ресурс] - Режим доступу: http://tech.vernadskyjournals.in.ua/journals/2020/3_2020/part_1/21.pdf

Додаток А

КОД ПРОГРАМИ

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_predict, cross_val_score
from sklearn.metrics import accuracy_score, classification_report, roc_auc_score, roc_curve, confusion_matrix
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.svm import SVC
from sklearn.svm import LinearSVC
from sklearn.linear_model import SGDClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.ensemble import HistGradientBoostingClassifier
from lightgbm import LGBMClassifier
from xgboost import XGBClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.neural_network import MLPClassifier
# створюємо dataframe
df = pd.read_csv('airline-passenger-satisfaction.csv')
# додаємо опцію щоб всі стовпці були відображені
pd.set_option('display.max_columns', None)
# виводимо head після загрузки файлу
df.head()
# перевіряємо розмірність
df.shape
# перевіряємо типи даних
df.dtypes
# перевіряємо наявність відсутніх значень
df.isnull().sum()
# заповнимо відсутні значення нулями та застосовуємо зміни беспосередньо у набор даних
df['Arrival Delay in Minutes'].fillna(0, inplace=True)
df.info()
# видалимо непотрібні колонки id та Unnamed
df.drop(columns='id', inplace=True)
df.drop(columns='Unnamed: 0', inplace=True)
df.head()
# перевизначаємо тип даних object у цифровий формат
label_encoder = LabelEncoder()
columns = df.select_dtypes(include='object').columns
for column in columns:
    df[column] = label_encoder.fit_transform(df[column])
df.head()
```

```
# Перевірка наявності дублікатів
duplicates = df.duplicated()
print("Рядки з дублікатами:")
print(df[duplicates])
print("\nЗагальна кількість дублікатів:", duplicates.sum())
df.describe()
# кругова діаграма задоволеності пасажирів
plt.pie(df['satisfaction'].value_counts(), labels=df['satisfaction'].unique(), autopct='%1.1f%%')
# діаграми розподілу значень ознаки Gender
plt.pie(df['Gender'].value_counts(), labels=df['Gender'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Gender', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки Customer Type
plt.pie(df['Customer Type'].value_counts(), labels=df['Customer Type'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Customer Type', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки Age
fig, ax = plt.subplots(figsize=(12, 6))
sns.histplot(df, x="Age", hue="satisfaction", multiple="stack", edgecolor=".3", linewidth=.5, ax=ax)
plt.show()
# діаграми розподілу значень ознаки Type of Travel
plt.pie(df['Type of Travel'].value_counts(), labels=df['Type of Travel'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Type of Travel', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки Class
plt.pie(df['Class'].value_counts(), labels=df['Class'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Class', height=3,
        aspect=2, kind='count', hue='satisfaction')
# Розподіл цільової змінної залежно від дистанції авіаперельоту
fig, ax = plt.subplots(figsize=(7, 7))
sns.boxplot(x = "satisfaction", y = "Flight Distance", data = df, ax = ax)
plt.show()
# діаграми розподілу значень ознаки Inflight wifi service
plt.pie(df['Inflight wifi service'].value_counts(), labels=df['Inflight wifi service'].unique(),
autopct='%1.1f%%')
sns.catplot(data=df, x='Inflight wifi service', height=3,
        aspect=2, kind='count', hue='satisfaction')
df['Inflight wifi service'].unique()
# діаграми розподілу значень ознаки Departure/Arrival time convenient
plt.pie(df['Departure/Arrival time convenient'].value_counts(), labels=df['Departure/Arrival time
convenient'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Departure/Arrival time convenient', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки Ease of Online booking
plt.pie(df['Ease of Online booking'].value_counts(), labels=df['Ease of Online booking'].unique(),
autopct='%1.1f%%')
sns.catplot(data=df, x='Ease of Online booking', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки Gate location
plt.pie(df['Gate location'].value_counts(), labels=df['Gate location'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Gate location', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки Food and drink
plt.pie(df['Food and drink'].value_counts(), labels=df['Food and drink'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Food and drink', height=3,
        aspect=2, kind='count', hue='satisfaction')
```

```
# діаграми розподілу значень ознаки Online boarding
plt.pie(df['Online boarding'].value_counts(), labels=df['Online boarding'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Online boarding', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки Seat comfort
plt.pie(df['Seat comfort'].value_counts(), labels=df['Seat comfort'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Seat comfort', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки Inflight entertainment
plt.pie(df['Inflight entertainment'].value_counts(), labels=df['Inflight entertainment'].unique(),
autopct='%1.1f%%')
sns.catplot(data=df, x='Inflight entertainment', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки On-board service
plt.pie(df['On-board service'].value_counts(), labels=df['On-board service'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='On-board service', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки Leg room service
plt.pie(df['Leg room service'].value_counts(), labels=df['Leg room service'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Leg room service', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки Baggage handling
plt.pie(df['Baggage handling'].value_counts(), labels=df['Baggage handling'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Baggage handling', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки Checkin service
plt.pie(df['Checkin service'].value_counts(), labels=df['Checkin service'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Checkin service', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки Inflight service
plt.pie(df['Inflight service'].value_counts(), labels=df['Inflight service'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Inflight service', height=3,
        aspect=2, kind='count', hue='satisfaction')
# діаграми розподілу значень ознаки Cleanliness
plt.pie(df['Cleanliness'].value_counts(), labels=df['Cleanliness'].unique(), autopct='%1.1f%%')
sns.catplot(data=df, x='Cleanliness', height=3,
        aspect=2, kind='count', hue='satisfaction')
# кореляційна матриця ознак
plt.figure(figsize=(16, 8))
sns.heatmap(df.corr(),
        annot=True, fmt='.2f', cmap='Greens')
plt.show()
# етап моделювання
X = df.drop(columns='satisfaction')
y = df['satisfaction']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
# Створення та оцінка моделі LogisticRegression
model_lg = LogisticRegression(max_iter=100000)
model_lg.fit(X_train, y_train)
# Робимо прогнози на тестових даних
predictions = model_lg.predict(X_test)
# Оцінка точності
model_lg_accuracy = accuracy_score(y_test, predictions)

# Оцінка значення стандартного відхилення прогнозованих значень
model_lg_std = np.std(predictions)
```

```
# Виведемо значення точності та стандартного відхилення
print(f'Model Accuracy: {model_lg_accuracy}')
print(f'Standard Deviation of Predictions: {model_lg_std}')
# Створення та оцінка моделі LinearDiscriminantAnalysis
model_lda = LinearDiscriminantAnalysis()
model_lda.fit(X_train, y_train)
predictions = model_lda.predict(X_test)
model_lda_accuracy = accuracy_score(y_test, predictions)
model_lda_std = np.std(predictions)
print(f'Model Accuracy: {model_lda_accuracy}')
print(f'Standard Deviation of Predictions: {model_lda_std}')
# Створення та оцінка моделі Support Vector Machines
model_svc = SVC()
model_svc.fit(X_train, y_train)
predictions = model_svc.predict(X_test)
model_svc_accuracy = accuracy_score(y_test, predictions)
model_svc_std = np.std(predictions)
print(f'Model Accuracy: {model_svc_accuracy}')
print(f'Standard Deviation of Predictions: {model_svc_std}')
# Створення та оцінка моделі SGDClassifier
model_SGDC = SGDClassifier()
model_SGDC.fit(X_train, y_train)
predictions = model_SGDC.predict(X_test)
model_SGDC_accuracy = accuracy_score(y_test, predictions)
model_SGDC_std = np.std(predictions)
print(f'Model Accuracy: {model_SGDC_accuracy}')
print(f'Standard Deviation of Predictions: {model_SGDC_std}')
# Створення та оцінка моделі KNeighborsClassifier
model_knc = KNeighborsClassifier()
model_knc.fit(X_train, y_train)
predictions = model_knc.predict(X_test)
model_knc_accuracy = accuracy_score(y_test, predictions)
model_knc_std = np.std(predictions)
print(f'Model Accuracy: {model_knc_accuracy}')
print(f'Standard Deviation of Predictions: {model_knc_std}')
# Створення та оцінка моделі Gaussian Naive Bayes
model_gnb = GaussianNB()
model_gnb.fit(X_train, y_train)
predictions = model_gnb.predict(X_test)
model_gnb_accuracy = accuracy_score(y_test, predictions)
model_gnb_std = np.std(predictions)
print(f'Model Accuracy: {model_gnb_accuracy}')
print(f'Standard Deviation of Predictions: {model_gnb_std}')
# Створення та оцінка моделі DecisionTreeClassifier
model_dt = DecisionTreeClassifier()
model_dt.fit(X_train, y_train)
predictions = model_dt.predict(X_test)
model_dt_accuracy = accuracy_score(y_test, predictions)
model_dt_std = np.std(predictions)
print(f'Model Accuracy: {model_dt_accuracy}')
# Створення та оцінка моделі GradientBoostingClassifier
model_gb = GradientBoostingClassifier()
model_gb.fit(X_train, y_train)
predictions = model_gb.predict(X_test)
model_gb_accuracy = accuracy_score(y_test, predictions)
model_gb_std = np.std(predictions)
```

```
print(f'Model Accuracy: {model_gb_accuracy}')
print(f'Standard Deviation of Predictions: {model_gb_std}')
# Створення та оцінка моделі Random Forest
model_rf = RandomForestClassifier()
# Оцінка ефективності за допомогою крос-валідації
y_pred_cv = cross_val_predict(model_rf, X, y, cv=5)
scores = cross_val_score(model_rf, X, y, cv=5, scoring='accuracy')
# Оцінка ефективності для кінцевої моделі
model_rf_accuracy = scores.mean()
model_rf_std = np.std(y_pred_cv)
print(f'Model Accuracy: {model_rf_accuracy}')
print(f'Standard Deviation of Predictions: {model_rf_std}')
# Classification Report
class_report = classification_report(y, y_pred_cv)
print("\nClassification Report (Cross-Validated):")
print(class_report)
# Feature Importance
model_rf2.fit(X, y)
feature_importances = model_rf.feature_importances_
feature_names = X.columns
plt.figure(figsize=(12, 6))
plt.barh(feature_names, feature_importances, align='center')
plt.xlabel('Feature Importance')
plt.ylabel('Feature Name')
plt.title('Feature Importance Horizontal Bar Plot')
plt.tight_layout()
plt.show()
# AUC-ROC Curve
y_proba = cross_val_predict(model_rf, X, y, cv=5, method='predict_proba')[:, 1]
fpr, tpr, thresholds = roc_curve(y, y_proba)
roc_auc = roc_auc_score(y, y_proba)
plt.figure(figsize=(8, 8))
plt.plot(fpr, tpr, label=f'AUC = {roc_auc:.2f}')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()
# Confusion Matrix
cm = confusion_matrix(y, y_pred_cv)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
# Створення та оцінка моделі Random Forest з criterion='entropy'
model_rf2 = RandomForestClassifier(criterion='entropy')
model_rf2.fit(X_train, y_train)
predictions = model_rf2.predict(X_test)
model_rf2_accuracy = accuracy_score(y_test, predictions)
model_rf2_std = np.std(predictions)
print(f'Model Accuracy: {model_rf2_accuracy}')
print(f'Standard Deviation of Predictions: {model_rf2_std}')
class_report = classification_report(y_test, predictions)
print("\nClassification Report:")
```

```
print(class_report)
# Створення та оцінка моделі Random Forest з criterion='log_loss'
model_rf3 = RandomForestClassifier(criterion='log_loss')
model_rf3.fit(X_train, y_train)
predictions = model_rf3.predict(X_test)
model_rf3_accuracy = accuracy_score(y_test, predictions)
model_rf3_std = np.std(predictions)
print(f'Model Accuracy: {model_rf3_accuracy}')
print(f'Standard Deviation of Predictions: {model_rf3_std}')
class_report = classification_report(y_test, predictions)
print("\nClassification Report:")
print(class_report)
# Створення та оцінка моделі ExtraTreesClassifier
model_etc = ExtraTreesClassifier()
model_etc.fit(X_train, y_train)
predictions = model_etc.predict(X_test)
model_etc_accuracy = accuracy_score(y_test, predictions)
model_etc_std = np.std(predictions)
print(f'Model Accuracy: {model_etc_accuracy}')
print(f'Standard Deviation of Predictions: {model_etc_std}')
# Створення та оцінка моделі HistGradientBoostingClassifier
model_hgbc = HistGradientBoostingClassifier()
model_hgbc.fit(X_train, y_train)
predictions = model_hgbc.predict(X_test)
model_hgbc_accuracy = accuracy_score(y_test, predictions)
model_hgbc_std = np.std(predictions)
print(f'Model Accuracy: {model_hgbc_accuracy}')
print(f'Standard Deviation of Predictions: {model_hgbc_std}')
# Створення та оцінка моделі LGBMClassifier
model_lgbm = LGBMClassifier()
# Оцінка ефективності за допомогою крос-валідації
y_pred_cv = cross_val_predict(model_lgbm, X, y, cv=5)
scores = cross_val_score(model_lgbm, X, y, cv=5, scoring='accuracy')
# Оцінка ефективності для кінцевої моделі
model_lgbm_accuracy = scores.mean()
model_lgbm_std = np.std(y_pred_cv)
print(f'Model Accuracy: {model_lgbm_accuracy}')
print(f'Standard Deviation of Predictions: {model_lgbm_std}')
# Classification Report
class_report = classification_report(y, y_pred_cv)
print("\nClassification Report (Cross-Validated):")
print(class_report)
# Feature Importance
model_lgbm.fit(X, y)
feature_importances = model_lgbm.feature_importances_
feature_names = X.columns
plt.figure(figsize=(12, 6))
plt.barh(feature_names, feature_importances, align='center')
plt.xlabel('Feature Importance')
plt.ylabel('Feature Name')
plt.title('Feature Importance Horizontal Bar Plot')
plt.tight_layout()
plt.show()
# AUC-ROC Curve
y_proba = cross_val_predict(model_lgbm, X, y, cv=5, method='predict_proba')[:, 1]
fpr, tpr, thresholds = roc_curve(y, y_proba)
```

```
roc_auc = roc_auc_score(y, y_proba)
plt.figure(figsize=(8, 8))
plt.plot(fpr, tpr, label=f'AUC = {roc_auc:.2f}')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()
# Confusion Matrix
cm = confusion_matrix(y, y_pred_cv)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
# Створення та оцінка моделі LGBMClassifier з boosting_type='dart'
model_lgbm2 = LGBMClassifier(boosting_type='dart')
model_lgbm2.fit(X_train, y_train)
predictions = model_lgbm2.predict(X_test)
model_lgbm2_accuracy = accuracy_score(y_test, predictions)
model_lgbm2_std = np.std(predictions)
print(f'Model Accuracy: {model_lgbm2_accuracy}')
print(f'Standard Deviation of Predictions: {model_lgbm2_std}')
class_report = classification_report(y_test, predictions)
print("\nClassification Report:")
print(class_report)
# Створення та оцінка моделі XGBClassifier
model_xgbc = XGBClassifier()
# Оцінка ефективності за допомогою крос-валідації
y_pred_cv = cross_val_predict(model_xgbc, X, y, cv=5)
scores = cross_val_score(model_xgbc, X, y, cv=5, scoring='accuracy')
# Оцінка ефективності для кінцевої моделі
model_xgbc_accuracy = scores.mean()
model_xgbc_std = np.std(y_pred_cv)
print(f'Model Accuracy: {model_xgbc_accuracy}')
print(f'Standard Deviation of Predictions: {model_xgbc_std}')
# Classification Report
class_report = classification_report(y, y_pred_cv)
print("\nClassification Report (Cross-Validated):")
print(class_report)
# Feature Importance
model_xgbc.fit(X, y)
feature_importances = model_xgbc.feature_importances_
feature_names = X.columns
plt.figure(figsize=(12, 6))
plt.barh(feature_names, feature_importances, align='center')
plt.xlabel('Feature Importance')
plt.ylabel('Feature Name')
plt.title('Feature Importance Horizontal Bar Plot')
plt.tight_layout()
plt.show()
# AUC-ROC Curve
y_proba = cross_val_predict(model_xgbc, X, y, cv=5, method='predict_proba')[:, 1]
fpr, tpr, thresholds = roc_curve(y, y_proba)
roc_auc = roc_auc_score(y, y_proba)
```

```
plt.figure(figsize=(8, 8))
plt.plot(fpr, tpr, label=f'AUC = {roc_auc:.2f}')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()
# Confusion Matrix
cm = confusion_matrix(y, y_pred_cv)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
# Створення та оцінка моделі BaggingClassifier
model_bc = BaggingClassifier()
model_bc.fit(X_train, y_train)
predictions = model_bc.predict(X_test)
model_bc_accuracy = accuracy_score(y_test, predictions)
model_bc_std = np.std(predictions)
print(f'Model Accuracy: {model_bc_accuracy}')
print(f'Standard Deviation of Predictions: {model_bc_std}')
# Створення та оцінка моделі AdaBoostClassifier
model_ab = AdaBoostClassifier()
model_ab.fit(X_train, y_train)
predictions = model_ab.predict(X_test)
model_ab_accuracy = accuracy_score(y_test, predictions)
model_ab_std = np.std(predictions)
print(f'Model Accuracy: {model_ab_accuracy}')
print(f'Standard Deviation of Predictions: {model_ab_std}')
# Створення та оцінка моделі MLPClassifier
mlp_model = MLPClassifier()
mlp_model.fit(X_train, y_train)
predictions = mlp_model.predict(X_test)
mlp_model_accuracy = accuracy_score(y_test, predictions)
mlp_model_std = np.std(predictions)
print(f'Model Accuracy: {mlp_model_accuracy}')
print(f'Standard Deviation of Predictions: {mlp_model_std}')
```

Додаток Б

ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім’я файла	**Опис**
Пояснювальні документи	
Диплом.doc	Пояснювальна записка кваліфікаційної роботи. Документ Word.
Диплом.pdf	Пояснювальна записка кваліфікаційної роботи в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація.ppt	Презентація до магістерської роботи