МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

**НИКИТЧУК Владислав Віталійович**

# Модуль прогнозування часу доставки товарів на основі машинного навчання / Module for predicting the time of delivery of goods based on machine learning

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Дипломний проект

Виконав студент групи КН-41
В. В. Никитчук

Науковий керівник:
к.т.н., доцент Т.В. Лендюк

Дипломний проект допущено до захисту
«___»_____ 2023 р.

Завідувач кафедри
_____ М.П. Комар

**Тернопіль – 2023**

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
«____» _____2022р.

# З А В Д А Н Н Я

## НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ

Никитчуку Владиславу Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема проекту: Модуль прогнозування часу доставки товарів на основі машинного навчання / Module for predicting the time of delivery of goods based on machine learning
керівник проекту к.т.н., доцент Т.В. Лендюк

затверджені наказом по університету від 08 грудня 2022 р. № 491.

2. Строк подання студентом закінченого проекту 01 червня 2023 р.

3. Вихідні дані до проекту: технічне завдання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- дослідити вплив часу доставки товарів на електронну комерцію та його важливість для клієнтів та підприємств.
- проаналізувати наявні рішення та методи прогнозування часу доставки товарів, визначити їх переваги та недоліки.
- оглянути методи класифікації для прогнозування часу доставки товарів, визначити найбільш ефективні.
- виконати розвідувальний аналіз даних, включаючи статистичну обробку та візуалізацію даних.
- обрати найбільш оптимальні методи класифікації та розробити алгоритми для прогнозування часу доставки товарів.
- розробити алгоритм прогнозування часу доставки товарів на основі

машинного навчання.
- підготувати та обробити набір даних для навчання моделей.
- навчити моделі на основі підготовленого набору даних.
- оцінити отримані моделі за допомогою метрик точності та зробити висновки про ефективність методів класифікації для прогнозування часу доставки товарів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
- алгоритм прогнозування часу доставки товарів на основі машинного навчання.

6. Консультанти розділів проекту

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | Завдання видав | Завдання прийняв |
| Н. контроль | к.т.н., доцент Т.В. Лендюк | | |
| | | | |
| | | | |

7. Дата видачі завдання 08 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проекту | Строк виконання етапів проекту | Примітка |
|---|---|---|---|
| 1 | Аналіз предметної області і постановка задачі дослідження | 30.12.2022 | |
| 2 | Алгоритмічне та інформаційне забезпечення прогнозування часу доставки товарів | 24.03.2023 | |
| 3 | Програмно-технологічне забезпечення прогнозування часу доставки товарів | 12.05.2023 | |
| 4 | Повне завершення та оформлення дипломного проекту | 01.06.2023 | |

Студент _____ В. В. Никитчук
( підпис )

Керівник проекту_____ Т.В. Лендюк
( підпис )

# РЕФЕРАТ

Пояснювальна записка до дипломного проекту: 66с., 6 рис., 1 табл., 2 додатки, 55 джерел.

Метою дипломного проекту є розробка та реалізація ефективного модуля прогнозування часу доставки товарів на основі машинного навчання.

Об'єктом дослідження є процес прогнозування часу доставки товарів в електронній комерції.

Предметом дослідження є методи та алгоритми класифікації, які застосовуються для прогнозування часу доставки товарів в електронній комерції.

Розроблено та досліджено програмне забезпечення для модуля прогнозування часу доставки товарів на основі машинного навчання в сферу електронної комерції та логістики.

Розроблений модуль дозволить покращити точність прогнозування, зменшити час доставки, знизити витрати на зберігання товарів та підвищити задоволення клієнтів. Компанії, які зможуть точно та швидко доставляти товари, матимуть конкурентну перевагу на ринку. Також результати дослідження можуть використовуватися для подальшого вдосконалення методів прогнозування часу доставки та розробки нових стратегій управління логістикою

ПРОГНОЗУВАННЯ ЧАСУ ДОСТАВКИ, МАШИННЕ НАВЧАННЯ, ЛОГІСТИЧНІ ПРОЦЕСИ, ЕЛЕКТРОННА КОМЕРЦІЯ, ОПТИМАЛЬНИЙ РОЗПОДІЛ РЕСУРСІВ

# ABSTRACT

The bachelor's thesis report: 66 pages, 6 figures, 1 table, 2 appendices, 55 references.

The aim of the diploma project is to develop and implement an efficient module for predicting the delivery time of goods based on machine learning.

The object of the research is the process of predicting the delivery time of goods in e-commerce.

The subject of the research is the methods and classification algorithms applied to predict the delivery time of goods in e-commerce.

Software for the delivery time prediction module based on machine learning has been developed and investigated in the field of e-commerce and logistics.

The developed module will improve prediction accuracy, reduce delivery time, lower costs of inventory holding, and enhance customer satisfaction. Companies capable of delivering goods accurately and quickly will gain a competitive advantage in the market. Furthermore, the research results can be utilized for further improvement of delivery time prediction methods and the development of new logistics management strategies.

DELIVERY TIME PREDICTION, MACHINE LEARNING, LOGISTIC PROCESSES, E-COMMERCE, OPTIMAL RESOURCE ALLOCATION.

# ТЕХНІЧНЕ ЗАВДАННЯ

## 1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

1.1 Модуль інтелектуального передбачення поведінки клієнтів інтернет-магазину.

1.2 Область застосування – електронна комерція.

## 2. ОСНОВА ДЛЯ РОЗРОБЛЕННЯ

Основою для розроблення є завдання на дипломний проект, затверджене кафедрою інформаційно-обчислювальних систем і управління факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету.

## 3. ПРИЗНАЧЕННЯ РОЗРОБЛЕНОГО КОМПЛЕКСУ

Метою дипломного проекту є розробка та реалізація ефективного модуля прогнозування часу доставки товарів на основі машинного навчання.

## 4. ДЖЕРЕЛА РОЗРОБЛЕННЯ

Джерелами даної розробки є матеріали навчальної і реферативної літератури, технічна документація, науково-дослідні статті, журнали, Інтернет.

## 5. ТЕХНІЧНІ ВИМОГИ

5.1 Основні функціональні вимоги до програмної системи:

– вибір даних (пакетних даних / даних у реальному часі) з різних архітектур великих даних;

– провести попередню обробку даних для інтелектуального аналізу тексту.

– навчити моделі на основі підготовленого набору даних.

– оцінити отримані моделі за допомогою метрик точності та зробити

висновки про ефективність методів класифікації для прогнозування часу доставки товарів.

5.2 Вимоги до апаратних засобів:

– кластер запускається на фізичній машині Macbook Pro з 16 гігабайт оперативної пам’яті та процесором 2.3 GHz Intel Core i5.

5.3 Вимоги до програмних засобів:

– для розробки програмне забезпечення - Python 3.7;

– для створення графічного інтерфейсу користувача використано – tkinter, Adobe XD;

– для реалізації моделей навчання – фреймворк sklearn.

6. ПОРЯДОК КОНТРОЛЮ

6.1 Представлення дипломного проекту на попередній захист.

6.2 Представлення дипломного проекту на захист.

Завдання прийняв до виконання _____ В. В. Никитчук

( підпис ) (прізвище та ініціали)

Керівник дипломного проекту_____ Т.В. Лендюк

( підпис ) (прізвище та ініціали)

ЗМІСТ

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

1. API: Application Programming Interface - інтерфейс програмування додатків, який визначає набір правил та протоколів для взаємодії між програмами.
2. SQL: Structured Query Language - мова структурованих запитів, використовується для роботи з реляційними базами даних, зокрема для збереження, управління та отримання даних.
3. NoSQL: Not Only SQL - підхід до управління базами даних, який використовується для збереження та управління даними, що не підкоряються суворим схемам реляційних баз даних.
4. CSV: Comma-Separated Values - формат файлу, в якому дані розділяються комами (або іншими символами) для зручної обробки табличних даних.
5. XGB Regressor: XGBoost Regressor - модель машинного навчання, заснована на алгоритмі градієнтного бустингу, що використовується для задач регресії.
6. EDA: Exploratory Data Analysis - метод аналізу даних, який використовується для розуміння особливостей даних, виявлення патернів, візуалізації та виявлення аномалій.
7. MSE: Mean Squared Error - середня квадратична помилка, метрика, що використовується для вимірювання середнього квадрату розбіжності між прогнозованими значеннями і спостережуваними значеннями.
8. MAE: Mean Absolute Error - середня абсолютна помилка, метрика, що використовується для вимірювання середнього абсолютного значення розбіжності між прогнозованими значеннями і спостережуваними значеннями.

# ВСТУП

Актуальність теми. Швидкий та точний прогноз часу доставки є ключовим фактором для багатьох компаній, особливо в електронній комерції та онлайн-роздрібній торгівлі. Зростання популярності онлайн-шопінгу призводить до збільшення обсягу замовлень та посилок, що потребує ефективного та швидкого управління логістикою та доставкою. Клієнти все більше очікують точних прогнозів часу доставки та своєчасне отримання замовлених товарів.

Застосування машинного навчання для прогнозування часу доставки може допомогти компаніям покращити свої логістичні процеси, знизити витрати та підвищити задоволення клієнтів. Точний прогноз часу доставки дозволяє планувати ресурси більш ефективно, уникати запізнень та забезпечувати вчасну доставку. Також слід зазначити, що сучасні алгоритми машинного навчання та методи розвідувального аналізу даних розвиваються швидко, включаючи покращені ансамблеві моделі та оптимізаційні алгоритми. Це надає багато можливостей для вдосконалення прогнозування часу доставки та досягнення більш точних результатів.

Таким чином, бакалаврський проект є актуальним і важливим, оскільки він пропонує розв'язання практичних проблем у сфері логістики та доставки, використовуючи передові методи машинного навчання.

Отже, враховуючи актуальність теми, можливість практичного застосування результатів та науковий інтерес, ваш бакалаврський проект з модуля прогнозування часу доставки товарів на основі машинного навчання є важливим і потенційно корисним для академічного та бізнес-середовища.

Метою дипломного проекту є розробка та реалізація ефективного модуля прогнозування часу доставки товарів на основі машинного навчання. Основним завданням проекту може бути покращення точності прогнозування, зменшення затримок та невідповідностей у доставці, а також забезпечення

оптимального розподілу ресурсів для ефективного управління логістичними процесами.

Досягнення цієї мети зумовило потребу теоретичних розробок, визначення та послідовного вирішення таких завдань:

1. Дослідити вплив часу доставки товарів на електронну комерцію та його важливість для клієнтів та підприємств.
2. Проаналізувати наявні рішення та методи прогнозування часу доставки товарів, визначити їх переваги та недоліки.
3. Оглянути методи класифікації для прогнозування часу доставки товарів, визначити найбільш ефективні.
4. Виконати розвідувальний аналіз даних, включаючи статистичну обробку та візуалізацію даних.
5. Обрати найбільш оптимальні методи класифікації та розробити алгоритми для прогнозування часу доставки товарів.
6. Розробити алгоритм прогнозування часу доставки товарів на основі машинного навчання.
7. Підготувати та обробити набір даних для навчання моделей.
8. Навчити моделі на основі підготовленого набору даних.
9. Оцінити отримані моделі за допомогою метрик точності та зробити висновки про ефективність методів класифікації для прогнозування часу доставки товарів.

Об'єктом дослідження є процес прогнозування часу доставки товарів в електронній комерції.

Предметом дослідження є методи та алгоритми класифікації, які застосовуються для прогнозування часу доставки товарів в електронній комерції.

Методи дослідження включають аналіз відомих рішень прогнозування часу доставки, проведення розвідувального аналізу даних, використання методів класифікації, підготовку та аналіз набору даних, навчання моделей машинного навчання та оцінку отриманих моделей.

Практичне значення даного дослідження полягає у впровадженні модуля прогнозування часу доставки товарів на основі машинного навчання в сферу електронної комерції та логістики. Це дозволить покращити точність прогнозування, зменшити час доставки, знизити витрати на зберігання товарів та підвищити задоволення клієнтів. Компанії, які зможуть точно та швидко доставляти товари, матимуть конкурентну перевагу на ринку. Також результати дослідження можуть використовуватися для подальшого вдосконалення методів прогнозування часу доставки та розробки нових стратегій управління логістикою.

Структура та обсяг роботи. Дипломний проект складається з вступу, трьох розділів, висновків та списку використаних джерел. Загальний обсяг роботи становить 66 сторінок комп'ютерного тексту, включаючи 6 рисунків та 1 таблицю. У списку використаних джерел наведено 55 найменувань, які займають 6 сторінок.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

## 1.1 Роль і важливість часу доставки товарів для електронної комерції

Час доставки товарів є одним з ключових елементів успіху комерційної діяльності, особливо в умовах глобалізації та швидкого розвитку електронної комерції. Він впливає на задоволення клієнтів, конкурентність компанії, ефективність управління запасами та спроможність швидко реагувати на зміни ринку. В контексті цього, розуміння важливості часу доставки товарів та ефективне управління ним стає одним із ключових завдань для успішного ведення бізнесу.

Час доставки товарів відіграє важливу роль в комерційній діяльності, особливо в е-комерції. Ось декілька причин, чому він є таким важливим:

1. Задоволення клієнтів: Швидка доставка товарів забезпечує високий рівень задоволення клієнтів. Це допомагає зберегти стабільність клієнтської бази та привабити нових клієнтів.
2. Конкурентна перевага: Швидкий та ефективний час доставки може бути конкурентною перевагою, особливо у секторі е-комерції, де клієнти можуть порівнювати час доставки між різними компаніями.
3. Ротація запасів: Швидка доставка допомагає підприємствам швидше обертати запаси, що знижує витрати на зберігання та підтримує ефективність бізнесу.
4. Відгук на зміни попиту: Коли компанії можуть швидко доставляти товари, вони можуть швидше реагувати на зміни в попиті, що дозволяє їм бути більш гнучкими та пристосовуватися до ринкових умов.
5. Зменшення ризику затримки: Затримки в доставці можуть призвести до незадоволення клієнтів, втрати продажів або негативних відгуків. Ефективний час доставки допомагає зменшити ці ризики.

Тому ефективне управління часом доставки товарів є важливою складовою успішного бізнесу.

Модуль прогнозування часу доставки товарів на основі машинного навчання може стати значним кроком у вдосконаленні бізнес-процесів, пов'язаних із доставкою товарів. Він може враховувати різні фактори, такі як відстань, погодні умови, час року, та історію доставок, щоб зробити точні прогнози.

Високий рівень точності прогнозування може привести до збільшення задоволення клієнтів, оскільки вони отримують свої товари в обіцяний час. Також це може покращити ефективність обігу товарів, зменшити витрати на зберігання та дозволити компанії швидше реагувати на зміни в попиті.

Більше того, цей модуль може стати потужним інструментом конкурентної боротьби, оскільки компанії, які можуть точно і швидко доставляти товари, мають перевагу на ринку.

Проте, слід враховувати, що успіх впровадження такого модулю залежить від якості вхідних даних та належного тюнінгу моделі машинного навчання. Також важливо регулярно оновлювати модель, щоб вона могла адаптуватися до змін у ринкових умовах або паттернах поведінки споживачів.

В цілому, використання модулю прогнозування часу доставки товарів на основі машинного навчання може стати суттєвим покращенням у управлінні логістикою та доставкою товарів, що сприятиме зростанню бізнесу.

## 1.2 Аналіз відомих рішень прогнозування часу доставки товарів

Блокування та карантин через COVID-19 збільшили попит на організації онлайн-доставки їжі (FDS), такі як Uber Eats, Deliveroo та Menulog, оскільки ресторанам було наказано припинити надання послуг харчування [1 , 2 ] . Згідно з дослідженням Morgan Stanley [ 3 ], ресторанний сектор зазнав значних змін внаслідок COVID-19 та соціального дистанціювання. Крім того,

вони додали, що через COVID-19 частка проникнення онлайн-доставки на ринку могла збільшитися на 2-3 роки. Крім того, звіт Roy Morgan Research [ 4] припускає, що загальнонаціональний карантин у середині 2020 року та продовжений карантин після цього «підштовхнули» зростання таких служб доставки їжі, як Uber Eats, Menulog, HelloFresh, Deliveroo та DoorDash. Молоде покоління, особливо клієнти до 40 років, більше замовляють, працюючи з дому [ 4 ]. У зв'язку з розвитком онлайн-ринків доставки їжі під час пандемії COVID-19 FDS пропонують різноманітність і різноманітність закладів харчування для зручності та комфорту вдома та на підприємствах. Нові кухні були привезені в країну через збільшення імміграції з інших країн [ 5]. Клієнти мають доступ до різноманітних страв і можливість робити замовлення в найкращих закладах харчування та ресторанах міста, не виходячи з дому чи з робочого місця. Завдяки широкому використанню додатків для смартфонів і доступності глобальної системи позиціонування доставка їжі до точного місця розташування клієнта більше не є проблемою [ 6 ]. Клієнти можуть відстежувати статус своїх замовлень, починаючи з моменту їх розміщення. Конкуренція між організаціями FDS стає жорсткішою; тепер Doordash, яка вже займає значну частку ринку FDS у США, хоче увійти в Австралію, зазіхнути на частку своїх конкурентів і збільшитися [ 7]. Крім того, Menulog інвестує в рекламу, залучаючи Кеті Перрі до своїх кампаній, щоб конкурувати з Uber Eats [ 8 ]. Uber Eats, Deliveroo та Menulog [ 9] — це глобальні ринкові системи замовлення та доставки, які покладаються на економічну бізнес-модель, але забезпечують логістику доставки. Ці організації працюють на комісійних засадах і пропонують власникам ресторанів і закладів харчування повне рішення для продажу без додаткових витрат. Програми для FDS дозволяють користувачам розміщувати замовлення, замовляти їжу з ресторанів і доставляти її їм лише кількома натисканнями телефону. Клієнти можуть вибирати з різноманітних варіантів харчування в мережі ресторанів. Такі послуги користуються великим попитом, що радує онлайн-провайдерів харчування. Зі збільшенням кількості

замовлень і коментарів більшість підприємств шукають способи ефективнішого використання даних для визначення областей, де вони можуть покращити задоволеність клієнтів. Незважаючи на великі продажі та інвестиції, Організації FDS продовжують боротися з прибутковістю через високі операційні витрати. Хижацьке ціноутворення — це поширена стратегія, яку використовують підприємства, щоб перемогти конкуренцію шляхом значного зниження витрат на харчування [10 ]. Оскільки онлайнові FDS здебільшого покладаються на ресторани, вони мало контролюють якість їжі. Якщо клієнт незадоволений їжею чи обслуговуванням, тоді організація FDS несе відповідальність за втрату доходу. Таким чином, такі підприємства, як Sprig [ 11 ] і Munchery [ 12 ], були змушені закритися через брак доходу.

Клієнти в першу чергу шукають відгуки та рекомендації інших, коли замовляють страви в онлайн-магазинах FDS. Позитивні відгуки змушують клієнтів приймати рішення про замовлення їжі в одному ресторані, а негативні – шукати інші варіанти [ 13]. Компанії FDS можуть шукати негативні коментарі щодо поширених типів скарг, таких як обслуговування клієнтів, якість їжі, вартість і повільна доставка, щоб зрозуміти сфери вдосконалення для підвищення задоволеності клієнтів. Система відгуків або відгуків тепер інтегрована в портали або платформи соціальних мереж ресторанів і FDS. Однак через величезний обсяг даних оглядів, розпорошених на численних платформах, і нестачу експертів із обслуговування клієнтів, необхідних для вивчення та відповіді на кожен із цих коментарів, лише кілька компаній насправді реагують на відгуки споживачів [14 ] . Організаціям більше не потрібно наймати експертів із обслуговування клієнтів для читання кожного відгуку, оскільки штучний інтелект (AI) може допомогти їм у вирішенні проблем і заощадити гроші [ 15 , 16].

Аналіз настроїв – це автоматизований процес, який використовується для визначення емоцій і настроїв клієнтів щодо їжі чи послуги [ 17 ]. Для аналізу настроїв можна використовувати методи машинного навчання (ML) і глибокого навчання (DL). Дослідники нещодавно зосередилися на DL,

черпаючи натхнення з результатів DL в інших областях, таких як комп'ютерний зір [ 18 ], медичний аналіз зображень [ 19 ], розпізнавання мови [ 20 ] і обробка природної мови (NLP) [ 21 ]. Хоча точність моделей DL вища в порівнянні з методами ML, їм бракує пояснюваності моделі чорного ящика [ 22]. Нейронна мережа з шарами взаємопов'язаних вузлів, такими як вхідний рівень, кілька прихованих шарів і вихідний рівень, відома як DL [ 23 ]. Класифікатори DL намагаються імітувати людський мозок, приймаючи рішення, беручи необроблені дані, вилучаючи ознаки та регулюючи ваги та упередження [ 24 ]. Кожен рівень базується на вхідних даних із попередніх рівнів і передає їх на наступний рівень, процес, відомий як пряме поширення [ 23 ]. Помилка передбачення обчислюється за допомогою функції втрат, такої як градієнтний спуск, і помилка виправляється шляхом коригування ваг і зміщення вузлів шляхом переміщення назад у часі, процес, відомий як зворотне поширення [ 23]. DL-модель прогнозує результат, використовуючи процеси прямого та зворотного поширення, виправляючи помилки та вагові коефіцієнти, доки не досягне оптимального прогнозу. Нелінійність, автоматичне вилучення ознак із необроблених даних, динамічне коригування ваги між вузлами для виправлення помилок і залежність від того, наскільки сильні вхідні ваги для встановлення зв'язку з вузлами на наступному рівні, ускладнюють візуалізацію або інтерпретацію моделі в термінах обґрунтування рішень класифікатора DL. Дослідники розпочали розробку ретроспективних методів для пояснення рішень, прийнятих класифікаторами DL, таких як SHapley Additive ExPlanations (SHAP) [ 25 ] та Local Interpretable Model-Agnostic Explanations (LIME) [ 26 ], і застосувати їх у деяких областях, таких як просторові прогноз посухи [27].

За останні кілька десятиліть було опубліковано велику кількість досліджень щодо застосування методів машинного навчання для виконання аналізу настроїв у домені FDS [ 13 , 28 , 29 , 30 , 31 ]. Аналіз настроїв відгуків клієнтів із твітів для різних FDS, таких як Swiggy, Zomato та Uber Eats, був виконаний, щоб зрозуміти задоволеність споживачів [ 32]. Відгуки клієнтів

були отримані з Twitter за допомогою R-Studio, а для твітів використовувався метод аналізу настроїв на основі Lexicon. Твіти були проаналізовані та надалі використані для надання відгуків і рекомендацій бізнесу. Інше дослідження порівнювало різні методи машинного навчання, такі як Дерево рішень (DT), Наївний Байєс (NB), Логістична регресія (LR) і Машина опорних векторів (SVM) [ 31] для аналізу та класифікації настроїв клієнтів. Модель DT досягла точності 89%, NB досягла 82,5%, LR досягла 90%, а SVM досягла 91%. Що стосується продуктивності моделей з точки зору часу обчислення, моделі DT потрібно було 1 год 4 хв 32 с для навчання, тоді як SVM потребувало 6320 мс. В експерименті було використано шість наборів даних за рік з 2015 по 2020 рік, і було виявлено, що точність SVM для набору даних 2015 року становила 89%, для набору даних 2016 року – 92%, а для набору даних 2020 року – 92. % [ 31], що свідчить про те, що буде корисно інтегрувати рішення з іншими програмами, які можуть бути корисними для розуміння почуттів клієнтів щодо різних продуктів і послуг. Результати їхнього дослідження свідчать про те, що промисловість харчових продуктів і напоїв може використовувати модель ML, щоб залучати й утримувати клієнтів, розглядаючи скарги клієнтів.

В іншій значущій роботі Noor [ 33 ] порівняв результати Lexicon, SVM, обробки природної мови (NLP) і Text Mining з різних робіт і виявив, що Lexicon досяг найвищої точності 87,33% порівняно з іншими методами. Однак було б важко виконати аналіз настроїв іншими мовами, крім англійської [ 13 ]. Крім того, під час створення моделей необхідно враховувати адаптацію домену, оскільки слово в одному домені може мати інше значення в іншому. Наприклад, «легкий» є позитивним словом для домену електроніки, тоді як це негативне слово для кухонної техніки [ 13]. Методи ML/DL можуть подолати проблему адаптації домену шляхом навчання моделі з того самого набору даних домену. У відгуках клієнтів може бути кілька слів для позначення одного й того самого аспекту. Наприклад, терміни «LCD» і «екран» стосуються одного і того ж у контексті мобільного телефону. У контексті фільмів фотографії та фільми є синонімами, але не в контексті камер,

де вони стосуються двох різних речей. Крім того, терміни «фото» та «зображення» є синонімами в індустрії камер [ 13 ]. Традиційні підходи до навчання лексики на основі словника не працюють добре, оскільки вони суворо обмежені меншою кількістю слів, тоді як методи ML/DL можуть подолати ці проблеми [ 34 ].

Під час пандемії COVID-19 багато компаній отримали оцінку «1 зірка» за те, що вони були закриті під час карантину. FDS Yelp отримав погані відгуки через повільне обслуговування або спеку в зонах відпочинку. Ресторани повинні знати про скарги та очікування своїх клієнтів, які можна знайти за допомогою аналізу настроїв [ 28 ]. Незважаючи на те, що звичайні методи ML показали хороші результати в аналізі онлайн-оглядових даних, вони були обмежені в обробці природних даних у необробленому форматі [ 30 ]. Однак техніка глибокого навчання (DL) вирішує цю проблему за допомогою своєї обчислювальної моделі, яка включає кілька рівнів обробки для автоматичного виявлення шаблону слів із великої кількості даних [ 35]. Кілька дослідників запровадили глибоке навчання для аналізу настроїв клієнтів у своїй сфері [ 36 , 37 , 38 , 39 , 40 ]. Остання робота [ 28] створив і порівняв дві моделі ML і DL, щоб виконати аналіз настроїв щодо оглядів, отриманих із веб-сайту Yelp. Для традиційних моделей було застосовано дерево рішень із підвищенням градієнта (GBDT) і класифікатор випадкового лісу, тоді як для моделей DL використовувалися класифікатори Simple Embedding + Average Polling і Bidirectional LSTM (Bi-LSTM). Дослідження показало, що метод DL Bi-LSTM був ефективнішим у створенні підтем, тоді як Simple Embedding + Average pooling показав кращі результати в завданнях прогнозування відгуків клієнтів. Дослідження мало обмеження з точки зору моделі DL, оскільки, хоча воно показало вищу точність порівняно з моделями ML, його критикували за те, що воно базується на чорній скриньці та не піддається інтерпретації [28] . Таким чином, у цьому дослідженні було виявлено, що моделі DL працюють краще, ніж моделі ML з точки зору

точності, але їм бракує інтерпретації, що призводить до відсутності довіри до їх використання.

У [41] дослідженні використовувалися дані з електронної комерції для прогнозування часу доставки товарів. Для класифікації використовувалися методи, такі як дерева рішень та випадковий ліс. Отримані результати показали, що такий підхід може ефективно прогнозувати час доставки з високою точністю. У [42] дослідженні використовувалися дані з онлайн-магазину для прогнозування часу доставки товарів. Були застосовані методи класифікації, зокрема метод опорних векторів (SVM). Виявлено, що SVM показує високу точність при прогнозуванні часу доставки з використанням відповідних функцій ядра. У [43] дослідженні розглядалися різні алгоритми машинного навчання для прогнозування часу доставки в електронній комерції. Використовувалися методи класифікації, зокрема дерева рішень, наївний Баєс та k-найближчих сусідів. Виявлено, що метод дерев рішень має найвищу точність прогнозування. У [44] дослідженні розглядалися алгоритми машинного навчання для оцінки часу доставки на платформах електронної комерції. Використовувалися методи класифікації, такі як наївний Баєс, дерева рішень та метод опорних векторів. Результати показали, що комбінація декількох методів класифікації дозволяє досягти точності прогнозування на рівні близько 90%. У [45] дослідженні були застосовані різні алгоритми машинного навчання для прогнозування часу доставки в онлайн-роздрібній торгівлі. Використовувалися методи класифікації, включаючи дерева рішень, випадковий ліс та градієнтний бустінг. Виявлено, що градієнтний бустінг є найефективнішим методом для прогнозування часу доставки з точністю близько 93%.

Загальною тенденцією у цих дослідженнях є використання алгоритмів машинного навчання для класифікації даних з метою прогнозування часу доставки товарів. Використовуються різні методи, такі як дерева рішень, наївний Баєс, метод опорних векторів та градієнтний бустінг. Результати свідчать про ефективність цих методів у прогнозуванні часу доставки з

високою точністю, що сприяє поліпшенню процесу доставки в електронній комерції та онлайн-роздрібній торгівлі.

## 1.3 Огляд методів класифікації для прогнозування часу доставки товарів

Класифікація є одним з основних завдань машинного навчання, в якому модель вчиться прогнозувати певну категорію, на основі заданих вхідних даних. На основі навчального набору даних, модель вчиться розпізнавати шаблони або характеристики, які визначають категорію.

Ось декілька основних методів класифікації, які часто використовуються в машинному навчанні:

1. Логістична регресія: Це статистичний метод, що використовується для прогнозування бінарних змінних. Це може бути використано для прогнозування ймовірності певного події, на основі однієї або кількох незалежних змінних.
2. Метод k-найближчих сусідів (k-NN): Цей метод використовується для класифікації та регресії. Він працює, визначаючи відстань між різними точками в просторі і вибираючи "k" найближчих точок до заданої точки для визначення її класу.
3. Дерева рішень: Ці моделі використовуються для прогнозування як категорійних, так і неперервних змінних. Вони працюють, розділяючи простір даних на різні області, використовуючи "рішення" або розділення, засновані на різних властивостях.
4. Випадковий ліс (Random Forest): Це метод, що використовує ансамбль дерев рішень, кожне з яких навчається на підмножині даних. Класифікація здійснюється шляхом голосування між усіма деревами.
5. Метод підтримки векторів (SVM): Це метод, який шукає гіперплощину в N-вимірному просторі (N - число ознак), яка кращим чином розділяє

два класи даних. SVM використовує ядерну функцію для перетворення даних у вищий розмір, де вони можуть бути розділені лінійно.

6. Нейронні мережі: Це клас моделей, які намагаються імітувати роботу людського мозку, здатні вивчати складні шаблони з великої кількості даних. Вони можуть бути використані для класифікації, регресії, генерації тексту та багатьох інших завдань.
7. Градієнтний бустінг: Це метод, що використовує ансамбль слабких моделей (часто дерев рішень), які навчаються послідовно, кожна наступна модель намагається виправити помилки попередніх моделей.
8. Наївний Баєсовський класифікатор: Це простий пробабілістичний класифікатор, заснований на застосуванні теореми Баєса. Він вважає, що всі ознаки незалежні між собою (тому "наївний"), що робить його особливо корисним при роботі з великими наборами даних.
9. Глибоке навчання: Глибоке навчання - це тип нейронних мереж з багатьма шарами ("глибокими"), які можуть вивчати дуже складні шаблони. Ці моделі використовуються в широкому діапазоні завдань, включаючи класифікацію зображень, розпізнавання мови, переклад тексту та багато іншого.
10. Конволюційні нейронні мережі (CNN): Це тип нейронних мереж, що особливо ефективний для завдань, пов'язаних з обробкою зображень. Вони використовують конволюційні шари, які автоматично і адаптивно вивчають просторові ієрархії ознак.
11. Рекурентні нейронні мережі (RNN): Це клас нейронних мереж, які ефективні для обробки послідовних даних (наприклад, часових рядів або тексту), тому що вони мають "пам'ять" з урахуванням попередніх вхідних даних при обробці поточного введення.
12. Автоенкодери: Це тип нейронних мереж, який використовується для вивчення ефективних кодувань вхідних даних. Вони можуть бути використані для зменшення розмірності, видалення шуму з даних та навчання представлень даних.

13. Генеративно-змагальні мережі (GAN): Це клас моделей, які вивчаються випадково генерувати дані, які схожі на вхідні дані. Вони складаються з двох частин: генератора, який намагається створити дані, і дискримінатора, який намагається визначити, чи є дані справжніми або сгенерованими.

Кожен з цих методів має свої переваги і недоліки, а також варіюється за своєю придатністю до різних типів завдань. Вибір правильного методу залежить від специфіки задачі, типу і кількості даних, а також вимог до точності і витрат на обчислення.

Лінійна та поліноміальна регресії використовуються для виявлення та квантифікації залежностей між ознаками і цільовою змінною, що корисно для лінійних та нелінійних залежностей відповідно. Дерева рішень та випадковий ліс дозволяють виявляти важливі ознаки та враховують нелинійність та взаємодію ознак. XGB Regressor використовує градієнтний бустінг, що покращує прогнозування.

Для прогнозування часу доставки товарів обрали п'ять методів: лінійну регресію, поліноміальну регресію, дерева рішень, випадковий ліс і XGB Regressor. Лінійна та поліноміальна регресії допомагають виявити лінійні та нелінійні залежності в даних, що може бути корисним, оскільки час доставки може залежати від різних факторів у складній формі. Дерева рішень та випадковий ліс дозволять врахувати більш складні взаємодії між факторами. Нарешті, XGB Regressor використовує техніку градієнтного бустінгу, яка може подальше покращити точність прогнозу.

## 1.4 Вибір перспективного шляху і постановка задачі дослідження

Актуальність даного дослідження обумовлена кількісним та якісним зростанням електронної комерції, а також значною конкуренцією на ринку доставки товарів. Час доставки впливає на клієнтське задоволення, лояльність

та повторні покупки. Оптимальне управління та прогнозування часу доставки є вирішальним фактором успіху для багатьох компаній.

Традиційні методи прогнозування, такі як статистичні аналізи або експертні оцінки, можуть бути недостатньо точними та непродуктивними. Використання методів машинного навчання та аналізу даних в цій області відкриває нові можливості для поліпшення точності та швидкості прогнозування.

Дослідження спрямоване на розробку модуля прогнозування часу доставки на основі машинного навчання, який використовує вхідні дані, такі як відстань, вага товару, дорожні умови тощо, для точного прогнозування часу доставки. Це важливий крок у покращенні ефективності логістичних процесів, зменшенні затрат та покращенні клієнтського досвіду.

Крім того, зростання доступності великих обсягів даних та розвиток методів машинного навчання створюють сприятливі умови для розвитку прогнозування часу доставки. Дослідження в цій області має потенціал внести вагомий внесок у сучасну логістичну індустрію та покращити ефективність та точність доставки товарів.

Метою дипломної роботи є розробка та реалізація ефективного модуля прогнозування часу доставки товарів на основі машинного навчання. Основним завданням проекту може бути покращення точності прогнозування, зменшення затримок та невідповідностей у доставці, а також забезпечення оптимального розподілу ресурсів для ефективного управління логістичними процесами.

Досягнення цієї мети зумовило потребу теоретичних розробок, визначення та послідовного вирішення таких завдань:

1 Дослідити вплив часу доставки товарів на електронну комерцію та його важливість для клієнтів та підприємств.

2 Проаналізувати наявні рішення та методи прогнозування часу доставки товарів, визначити їх переваги та недоліки.

3 Оглянути методи класифікації для прогнозування часу доставки товарів, визначити найбільш ефективні.

4 Виконати розвідувальний аналіз даних, включаючи статистичну обробку та візуалізацію даних.

5 Обрати найбільш оптимальні методи класифікації та розробити алгоритми для прогнозування часу доставки товарів.

6 Розробити алгоритм прогнозування часу доставки товарів на основі машинного навчання.

7 Підготувати та обробити набір даних для навчання моделей.

8 Навчити моделі на основі підготовленого набору даних.

9 Оцінити отримані моделі за допомогою метрик точності та зробити висновки про ефективність методів класифікації для прогнозування часу доставки товарів.

Завдання 1-3 розглянуто в параграфах 1.1 - 1.4 відповідно, а виконання завдань 4-6 представлені у параграфах 2.1-3.3.

# 2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ ЧАСУ ДОСТАВКИ ТОВАРІВ

## 2.1 Підхід розвідувального аналізу

Розвідувальний аналіз даних (Exploratory Data Analysis, EDA) - це підхід до аналізу даних, який зосереджується на виявленні загальних шаблонів, виявленні аномалій, тестуванні гіпотез і перевірці припущень за допомогою статистичних графіків та таблиць.

Основні кроки розвідувального аналізу даних включають:

Крок 1. Збір даних: Це може включати збір даних з різних джерел, таких як бази даних, файли, API тощо.

Процес збору даних включає збір вхідних даних з різних джерел для подальшого аналізу. Збір даних може бути проведений різними способами залежно від типу, структури, і джерела даних. Ось декілька прикладів:

- Бази даних: Дані можуть бути зібрані з різних типів баз даних, таких як SQL, NoSQL, cloud-based storage services і т.д. Для цього можуть використовуватися SQL-запити або інтерфейси API, що надаються базами даних.
- Файли: Дані можуть бути зібрані з різних типів файлів, включаючи текстові файли (наприклад, CSV, TXT), електронні таблиці (наприклад, XLS, XLSX), JSON, XML та інші формати файлів.
- API: API (Application Programming Interface) - це набір правил і протоколів для взаємодії з програмним забезпеченням. Багато веб-сервісів надають API для доступу до їх даних. Наприклад, соціальні медіа платформи, такі як Twitter або Facebook, надають API для збору даних про твіти, повідомлення, коментарі та іншу інформацію.

- Веб-скрапінг: Веб-скрапінг - це процес екстракції даних з веб-сайтів. Це може бути корисно, коли дані не доступні через API або базу даних, але є на веб-сайті.
- Інтернет Інформаційних Речей (IoT) девайси: Дані можуть бути зібрані з різних IoT девайсів, таких як сенсори, камери, GPS-трекери тощо.
- Опитування та анкети: Дані можуть бути зібрані через опитування і анкети, які можуть бути проведені в онлайні або офлайні.

При зборі даних важливо враховувати якість даних, включаючи їх повноту, точність, консистентність і актуальність. Часто це потребує додаткової обробки даних, такої як очищення даних, для видалення або корекції неповних, невірних, або неактуальних даних.

Також важливо враховувати етичні проблеми і законодавчі обмеження при зборі даних, особливо якщо дані включають персональну або чутливу інформацію. Використання даних повинно відбуватися в рамках відповідного дозволу або згоди, і відповідати всім застосовним законам і регулятивним положенням, таким як Загальний регламент захисту даних (GDPR) в Європейському Союзі.

Нарешті, збір даних - це лише перший крок в циклі аналізу даних. Після того, як дані були зібрані і очищені, вони потребують подальшого аналізу і інтерпретації, щоб витягти з них корисні висновки або узгодження.

Крок 2. Очищення даних: Це включає видалення або корекцію неповних, некоректних, неточних або несуттєвих даних.

Очищення даних є важливим кроком в процесі обробки даних, оскільки якість даних може впливати на результати аналізу. Навіть найкращі алгоритми не зможуть виробити корисні висновки з поганих даних. Ось декілька аспектів, на які потрібно звернути увагу під час очищення даних:

1. Відсутні дані: Дані часто містять "пропуски", коли деяка інформація відсутня. Ви маєте вирішити, що робити з цими пропусками: чи ігнорувати ці записи, заповнити їх деяким визначеним значенням

(наприклад, середнім значенням для цього стовпця), або використати методи імпутації для заповнення пропусків на основі інших даних.

2. Неконсистентні дані: Дані можуть містити неконсистентності, наприклад, одні й ті ж дані можуть бути представлені по-різному в різних записах. Наприклад, дата народження може бути представлена в одному форматі в одному запису і в іншому форматі в іншому. Це потребує стандартизації форматів.
3. Неточні дані: Дані можуть містити помилки, такі як неправильно введені значення. Це може вимагати виправлення або видалення таких значень.
4. Викиди (outliers): Викиди - це значення, які значно відрізняються від інших. Вони можуть бути результатом помилок або можуть представляти важливі особливості даних. Ви повинні вирішити, як їх обробляти.
5. Несуттєві дані: Деякі дані можуть бути несуттєвими для вашого аналізу і можуть бути видалені, щоб спростити датасет.
6. Дублікати: Дублікати - це повторювані записи в дат асеті. Вони можуть виникнути через помилки при введенні даних або збої при зборі даних. Дублікати можуть спотворювати результати аналізу, тому вони зазвичай видаляються.

Очищення даних може бути досить трудомістким процесом, але воно вкрай важливе для забезпечення якості даних. Існують різні інструменти та бібліотеки для очищення даних, включаючи pandas в Python, dplyr в R, Excel, та спеціалізовані інструменти для очищення даних, такі як Trifacta або Talend.

Також важливо зауважити, що процес очищення даних повинен бути документований і відтворюваний. Це означає, що ви повинні зберігати записи про те, які зміни були зроблені, і мати змогу повторити процес очищення на нових або оновлених даних.

Крок 3. Аналіз даних: Цей крок включає статистичний аналіз даних, щоб виявити шаблони та тенденції. Це може включати розрахунок середніх

значень, медіан, мод, розмаху, стандартного відхилення та інших статистичних характеристик.

Аналіз даних є основним кроком в розвідувальному аналізі даних, оскільки він дозволяє вам отримати знання про свої дані і виявити шаблони та тенденції. Ось декілька ключових аспектів аналізу даних:

1. Описова статистика: Описова статистика включає обчислення основних статистичних характеристик даних, таких як середнє значення, медіана, мода, мінімум, максимум, діапазон, стандартне відхилення, та квартилі. Ці метрики допомагають зрозуміти розподіл даних та їх варіативність.
2. Візуалізація даних: Візуалізація даних допомагає представити дані у вигляді, який легко розуміти. Це може включати графіки, такі як гістограми, boxplots, scatter plots, heatmaps та інші. Візуалізація допомагає виявити шаблони та тенденції, які можуть бути прихованими в сирих даних.
3. Кореляційний аналіз: Кореляційний аналіз допомагає визначити ступінь залежності між різними змінними в даних. Це може бути важливо для визначення важливих взаємодій у ваших даних.
4. Тестування гіпотез: Це метод, який дозволяє перевірити певні припущення про ваши дані. Наприклад, ви можете перевірити, чи є різниця між середніми значеннями двох груп значущою.
5. Аналіз часових рядів: Якщо ваші дані містять часову компоненту, ви можете аналізувати тенденції, сезонність і циклічність в ваших даних.
6. Експлоративний аналіз: Після виявлення основних шаблонів і тенденцій, ви можете провести глибший експлоративний аналіз. Це може включати більш складні статистичні тести, моделювання даних або використання технік машинного навчання для отримання додаткових інсайтів.
7. Дослідження взаємозв'язків: Дослідження взаємозв'язків між змінними може допомогти визначити, які змінні найбільше впливають на

результати, які ви спостерігаєте. Це може бути здійснено через техніки, такі як регресійний аналіз або аналіз головних компонент.

Під час аналізу даних важливо пам'ятати, що кореляція не означає причинно-наслідкового зв'язку. Навіть якщо дві змінні сильно корелюють, це не означає, що одна змінна викликає зміну іншої. Для визначення причинно-наслідкових зв'язків може знадобитися додаткове дослідження або експериментальне проектування.

Крок 4. Візуалізація даних: Це включає створення графіків та діаграм, що допомагають візуально інтерпретувати дані. Це може включати гістограми, діаграми розсіювання, коробкові діаграми тощо.

Візуалізація даних - це потужний інструмент для розуміння даних і виявлення закономірностей, тенденцій та взаємозв'язків. Вона допомагає представити складні дані у вигляді, який легко розуміти і інтерпретувати. Ось декілька основних видів візуалізації даних, які ви можете використовувати:

1. Гістограми: Гістограми використовуються для візуалізації розподілу одновимірного набору даних. Вони демонструють, як дані розподілені по частоті, що дозволяє визначити такі характеристики даних, як мода, розподіл, викиди та ін.
2. Діаграми розсіювання: Діаграми розсіювання використовуються для візуалізації взаємозв'язку між двома змінними. Вони допомагають виявити кореляції та відносини між змінними.
3. Коробкові діаграми (Boxplots): Коробкові діаграми використовуються для візуалізації розподілу даних через квартилі. Вони також допомагають виявити викиди в даних.
4. Стовпчасті діаграми і графіки ліній: Стовпчасті діаграми і графіки ліній використовуються для візуалізації тенденцій та порівняння значень між різними групами.
5. Heatmaps: Heatmaps використовуються для візуалізації взаємозв'язків між трьома числовими змінними, одна з яких представлена кольором.

6. Діаграми взаємозв'язку (Correlation plots): Ці діаграми використовуються для візуалізації кореляції між різними змінними.
7. Діаграми "віолонки" (Violin plots): Діаграми "віолонки" є комбінацією коробкової діаграми та ядерної оцінки щільності. Вони дозволяють зобразити розподіл даних, а також його щільність.
8. Діаграми парних взаємозв'язків (Pair plots): Ці діаграми допомагають візуалізувати взаємозв'язки між кількома змінними одночасно. Вони є корисними для виявлення структури та взаємозв'язків в багатовимірних даних.
9. Дерева рішень (Decision trees): Хоча дерева рішень часто використовуються як моделі машинного навчання, вони також можуть слугувати візуалізацією, що демонструє, як рішення або прогнози зроблено на основі даних.
10. Географічні карти: Якщо ви працюєте з геопросторовими даними, візуалізація на географічних картах може бути дуже корисною для показу географічних шаблонів та тенденцій.

Важливо вибирати правильний тип візуалізації на основі типу даних, який маєте, та питання, на яке намагаєтеся відповісти. Відповідний інструмент візуалізації може значно полегшити процес розуміння та інтерпретації ваших даних.

Крок 5. Інтерпретація результатів: На цьому кроці результати аналізу та візуалізації даних інтерпретуються, щоб зробити висновки або перевірити гіпотези.

Інтерпретація результатів - це важливий крок в процесі розвідувального аналізу даних, який допомагає виробити значущі висновки з отриманих результатів. Даний процес може включати наступні аспекти:

1. Розуміння шаблонів та тенденцій: Після того, як ви виконали аналіз даних та їх візуалізацію, наступним кроком є розуміння виявлених шаблонів та тенденцій. Це може означати, наприклад, розуміння того, як різні змінні взаємодіють одна з одною або як вони змінюються в часі.

2. Перевірка гіпотез: Якщо у вас були попередні гіпотези про ваші дані, ви можете використовувати свої результати для перевірки цих гіпотез. Це може включати статистичні тести для перевірки значущості результатів.
3. Висновки: На основі вашого аналізу та інтерпретації ви маєте зробити висновки. Це може включати визначення ключових висновків з вашого аналізу, оцінку важливості та впливу цих висновків, а також визначення можливих наслідків або дій, які мають бути виконані на основі цих висновків.
4. Критичне мислення та скептицизм: Важливо підходити до інтерпретації результатів з критичним мисленням. Завжди ставте питання про можливі обмеження ваших даних або аналізу та будьте відкриті для альтернативних інтерпретацій.
5. Комунікація результатів: Нарешті, результати та висновки мають бути ефективно спілкуватися іншим. Це може включати створення звітів або презентацій.
6. Пошук нових запитань та гіпотез: Під час інтерпретації результатів ви, швидше за все, знайдете нові запитання, які виникли в результаті вашого аналізу. Це може призвести до формулювання нових гіпотез, які можна перевірити в майбутніх дослідженнях.
7. Узагальнення висновків: На цьому етапі ви маєте узагальнити свої висновки таким чином, щоб вони були зрозумілими для широкої аудиторії. Ви повинні забезпечити, щоб ваші висновки були засновані на даних і щоб вони були вагомими.
8. Оцінка впливу: Оцінка впливу ваших висновків може включати розуміння того, як ваши результати можуть вплинути на ділове середовище, наукову дисципліну, соціальний контекст або інші області.
9. Рекомендації щодо подальших дій: На основі ваших висновків ви можете розробити рекомендації щодо подальших дій. Це може включати зміни в політиці, введення нових процедур, проведення додаткових досліджень тощо.

10. Оформлення результатів: Результати та висновки мають бути представлені чітко та конкретно. Важливо забезпечити, щоб всі необхідні дані та висновки були включені, і вони були організовані логічно та зрозуміло.

Останній крок в процесі розвідувального аналізу даних - це повторення всього процесу, якщо це необхідно. Оскільки ви зможете отримати нові інсайти та гіпотези під час інтерпретації результатів, ви можете повернутися до кроку збору даних або очищення даних та повторити процес.

Розвідувальний аналіз даних - це важливий компонент в циклі обробки даних, оскільки він допомагає зрозуміти основні характеристики даних перед їх подальшим аналізом чи моделюванням.

Отже, розглянуто процес розвідувального аналізу даних (EDA), який є критично важливим етапом перед будь-яким процесом машинного навчання. EDA включає в себе збір даних, їх очищення, аналіз, візуалізацію та інтерпретацію результатів. Кожен з цих етапів має свої особливості та може мати значний вплив на якість та точність моделі машинного навчання.

Що стосується прогнозування часу доставки товарів на основі машинного навчання, процес EDA є важливим кроком для розуміння структури даних, виявлення аномалій, визначення важливих змінних та розуміння взаємозв'язків між різними факторами, що впливають на час доставки.

За допомогою EDA можна визначити, які фактори, такі як відстань, час замовлення, тип товару, обсяг та вага товару, можуть впливати на час доставки. Така інформація допоможе при формуванні та тренуванні моделі машинного навчання для прогнозування часу доставки.

Крім того, EDA може допомогти виявити будь-які проблеми в даних, такі як викиди або пропущені значення, які можуть вплинути на точність прогнозування часу доставки. Після проведення EDA та належного підготовки даних, можна перейти до вибору та тренування моделі машинного навчання,

такої як лінійна регресія, випадковий ліс або градієнтний бустінг, щоб прогнозувати час доставки на основі вхідних даних.

## 2.2 Методи класифікації

Опишу для вас п'ять моделей: лінійну регресію, поліноміальну регресію, дерево рішень, випадковий ліс і XGB Regressor.

Лінійна регресія: Лінійна регресія є однією з найпростіших і найпопулярніших моделей для задач прогнозування. Вона побудована на основі лінійної залежності між вхідними ознаками та вихідною змінною. Модель припускає, що існує лінійна залежність між вхідними змінними і цільовою змінною. Лінійна регресія шукає найкращу пряму, яка найкраще підходить для опису залежності між змінними. Зазвичай використовуються методи найменших квадратів для оцінки параметрів моделі.

Лінійна регресія може бути математично описана наступним чином:

Припустимо, що ми маємо набір навчальних прикладів, де ми маємо вхідні ознаки x1, x2, ..., xn та відповідні цільові значення y.

Модель лінійної регресії шукає найкращу пряму, яка найкраще підходить для опису залежності між вхідними змінними та цільовою змінною. Цю пряму можна описати рівнянням:

$$\boldsymbol{y = \beta^0 + \beta^1 x^1 + \beta^2 x^2 + \ldots + \beta_n xn + \varepsilon}$$

де:

- y представляє вихідну змінну, яку ми намагаємося прогнозувати.
- $\boldsymbol{x^1, x^2, \ldots, xn}$ представляють вхідні ознаки, які ми використовуємо для прогнозування.
- $\boldsymbol{\beta^0, \beta^1, \beta^2, \ldots, \beta_n}$ представляють параметри моделі, які потрібно оцінити.
- ε представляє помилку моделі.

Мета лінійної регресії полягає в оцінці значень параметрів $\boldsymbol{\beta^0, \beta^1, \beta^2, \ldots, \beta_n}$, які найкраще підходять для наших навчальних даних. Зазвичай використовується метод найменших квадратів, який мінімізує суму квадратів різниць між прогнозованими значеннями моделі і фактичними значеннями цільової змінної.

Поліноміальна регресія: Поліноміальна регресія є розширенням лінійної регресії, яке дозволяє моделювати нелінійні залежності між змінними. Вона використовує поліноми степенів більше одиниці для побудови моделі. Поліноміальна регресія може бути корисною, коли дані не можуть бути адекватно описані простою лінією. Вона може апроксимувати складніші залежності, додавши додаткові члени з вищими степенями вхідних ознак.

Поліноміальна регресія може бути математично описана наступним чином:

Припустимо, що ми маємо набір навчальних прикладів, де ми маємо вхідні ознаки x1, x2, ..., xn та відповідні цільові значення y.

Модель поліноміальної регресії використовує поліноми степенів більше одиниці для побудови моделі. Цю модель можна описати рівнянням:

$$\boldsymbol{y = \beta^0 + \beta^1 x^1 + \beta^2 x^2 + \ldots + \beta_n xn + \beta^{11} x^{12} + \beta^{22} x^{22} + \ldots + \beta_{nn} xn^2 + \beta^{12} x^1 x^2 + \ldots + \varepsilon}$$

де:

- y представляє вихідну змінну, яку ми намагаємося прогнозувати.
- $x_1$, $x_2$, ..., xn представляють вхідні ознаки, які ми використовуємо для прогнозування.
- $\beta_0$, $\beta_1$, $\beta_2$, ..., $\beta_n$ представляють параметри моделі для лінійної залежності.
- $\beta_{11}$, $\beta_{22}$, ..., $\beta_{nn}$ представляють параметри моделі для квадратичних членів полінома.
- $\beta_{12}$, $\beta_{13}$, ..., представляють параметри моделі для взаємодіючих членів полінома.

- є представляє помилку моделі.

Мета поліноміальної регресії полягає в оцінці значень параметрів

$$\boldsymbol{\beta^0, \beta^1, \beta^2, \dots, \beta_{\mathrm{n}}, \beta^{11}, \beta^{22}, \dots, \beta_{\mathrm{nn}}, \beta^{12}, \beta^{13}, \dots}$$

які найкраще підходять для наших навчальних даних. Знову, можна використовувати метод найменших квадратів для оцінки цих параметрів.

Дерево рішень: Дерево рішень є деревоподібною структурою, в якій кожен внутрішній вузол представляє тест на певну ознаку, кожна гілка виходить з вузла і представляє можливий результат тесту, а кожний листовий вузол представляє прогноз або значення. Дерева рішень можуть використовуватися як для задач класифікації, так і для задач регресії. У випадку задачі регресії, дерево рішень прогнозує числове значення на основі вхідних ознак. Воно розділяє простір ознак на декілька регіонів і назначає прогнозне значення для кожного регіону. Дерева рішень мають перевагу в тому, що легко інтерпретовані, оскільки можна візуалізувати структуру дерева. Однак, вони можуть бути схильні до перенавчання, коли модель погано узагальнює на нові дані.

Дерево рішень можна математично описати як алгоритм, який прогнозує значення вихідної змінної на основі тестів на вхідних ознаках. Припустимо, що ми маємо набір даних з n вхідних ознак, позначених як $\boldsymbol{x_1, x_2, \dots, x_n}$, та вихідну змінну y. Ми хочемо побудувати дерево рішень, яке прогнозуватиме значення y на основі тестів на цих ознаках.

Дерево рішень будується за допомогою рекурсивного поділу набору даних на дві частини в кожному вузлі на основі тесту на одній з вхідних ознак. Таким чином, кожний вузол має дві гілки: одну, що веде до наступного вузла, якщо результат тесту є "так", та іншу, що веде до іншого вузла, якщо результат тесту є "ні". Цей процес продовжується до тих пір, поки кожний листовий вузол не містить прогноз або значення.

Математично дерево рішень можна описати за допомогою наступної формули:

$$f(x) = \sum_{i=1}^{k} c_i \cdot I(x \in R_i)$$

де f(x) - прогнозоване значення вихідної змінної y для вхідного вектора ознак $\boldsymbol{x}, \boldsymbol{R_i}$ - і-тий регіон (вузол) дерева, $\boldsymbol{c_i}$ - константа (прогнозоване значення) для і-того регіону, а $\boldsymbol{I(x \in R_i)}$ - індикаторна функція, яка дорівнює 1, якщо вхідний вектор ознак x належить і-тому регіону, та 0 - інакше.

У формулі для дерева рішень, кожен регіон $\boldsymbol{R_i}$ може бути представлений як умовна область вхідних ознак. Для прикладу, можна записати формулу для двовимірних ознак $\boldsymbol{x_1}$та $\boldsymbol{x_2}$ наступним чином:

$$f(x) = c_1 \cdot I(x \in R_1) + c_2 \cdot I(x \in R_2) + \ldots + c_k \cdot I(x \in R_k)$$

де k - загальна кількість регіонів дерева рішень.

Кожна умовна область $\boldsymbol{R_i}$ може бути визначена за допомогою нерівностей на вхідні ознаки. Наприклад, для регіону $\boldsymbol{R_1}$, можна мати:

$$R_1 : x_1 < \theta_1 \quad \text{та} \quad x_2 < \theta_2$$

де $\boldsymbol{\theta_1}$та $\boldsymbol{\theta_2}$ - порогові значення, що визначають границі умовної області $\boldsymbol{R_1}$. Аналогічно, інші регіони $\boldsymbol{R_2, R_3, \ldots R_k}$ можуть мати свої власні умови на вхідні ознаки.

Константи $\boldsymbol{c_1, c_2, \ldots c_k}$ відповідають прогнозованим значенням для кожного регіону. Наприклад, якщо ми маємо задачу регресії і прогнозуємо неперервну змінну y, то $\boldsymbol{c_i}$ може бути середнім значенням цільової змінної для навчальних прикладів, що попадають до регіону $\boldsymbol{R_i}$.

Випадковий ліс: Випадковий ліс є ансамблевою моделлю, побудованою на основі дерев рішень. Він працює шляхом побудови багато дерев рішень із використанням підвибірки навчальних даних та підвибірки ознак для кожного

дерева. Коли потрібно зробити прогноз, кожне дерево випадкового лісу вносить свій внесок, і прогноз обчислюється як середнє або медіанне значення прогнозів всіх дерев. Випадковий ліс зазвичай має кращу здатність до узагальнення, ніж окремі дерева рішень, і він може працювати добре навіть у випадках з великою кількістю ознак.

Випадковий ліс можна математично описати як ансамбль дерев рішень, де кожне дерево рішень побудоване за допомогою підвибірки навчальних даних та підвибірки ознак. Для прикладу, якщо ми маємо N навчальних прикладів з n вхідними ознаками та вихідною змінною, алгоритм випадкового лісу можна описати так:

1. Для кожного дерева і випадкового лісу (i = 1, 2, ..., M): a. Згенерувати підвибірку навчальних прикладів розміром N' шляхом вибору прикладів з повторенням. b. Згенерувати підвибірку ознак розміром n' шляхом вибору ознак замість оригінального набору ознак.
2. Побудувати дерево рішень для кожного дерева випадкового лісу, використовуючи підвибірку навчальних прикладів та підвибірку ознак.
3. При потребі виконати прогноз для нового вхідного вектора ознак x: a. Отримати прогноз для кожного дерева рішень випадкового лісу, використовуючи вхідний вектор ознак x. b. Обчислити усереднений (або медіанний) прогноз всіх дерев рішень, щоб отримати кінцевий прогноз випадкового лісу.

   Математично це можна представити наступною формулою:

$$f(x) = \frac{1}{M} \sum_{i=1}^{M} f_i(x)$$

де f(x) - прогнозоване значення вихідної змінної y для вхідного вектора ознак $x$, $f_{i(x)}$ - прогнозоване значення i-го дерева рішень випадкового лісу, а M - загальна кількість дерев рішень в ансамблі.

XGB Regressor: XGB Regressor (Extreme Gradient Boosting Regressor) є ансамблевою моделлю, яка використовує градієнтний бустинг для регресії. Вона базується на ідеї покрокового покращення моделі шляхом додавання нових слабких моделей, які коригують помилки попередніх моделей. XGB Regressor використовує градієнтний спуск для оптимізації функції втрат. XGB Regressor використовує ансамбль дерев рішень, але відрізняється від випадкового лісу тим, що використовує градієнтний бустинг для покращення моделі. Градієнтний бустинг працює, надаючи більшу вагу невірно класифікованим або погано спрогнозованим прикладам під час тренування.

XGB Regressor має кілька параметрів, які можна налаштувати для досягнення оптимальної продуктивності, таких як глибина дерева, швидкість навчання та кількість дерев. Він також підтримує регуляризацію, що дозволяє контролювати перенавчання і підганяння моделі до даних. XGB Regressor є потужним алгоритмом, який зазвичай демонструє високу точність і хорошу здатність до узагальнення. Він широко використовується в конкурсах з машинного навчання та у практичних застосуваннях, де точність моделі має велике значення.

XGB Regressor (Extreme Gradient Boosting Regressor) використовує градієнтний бустинг для регресії. Математично, цей алгоритм можна описати наступним чином:

1. Початковий прогноз:

$$\hat{y}_0 = \text{середнє значення цільової змінної}$$

2. Покрокове покращення моделі:

$$\hat{y}_i = \hat{y}_{i-1} + \gamma \cdot \sum_{j=1}^{J} h_j(x)$$

де:

- $\hat{y}_i$- прогнозоване значення на кроку і
- $\hat{y}_{i-1}$ - прогнозоване значення на попередньому кроці $\hat{y}_{i-1} = \hat{y}_{i-0}$на першому кроці)
- γ- швидкість навчання (learning rate), що контролює внесок кожного дерева в ансамбль
- J - кількість дерев рішень в ансамблі
- $h_{j(x)}$ - j-е дерево рішень, яке приймає вхідний вектор ознак x та повертає прогнозовану поправку

3. Функціонал втрати:

$$\text{функціонал втрати} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{j=1}^{J} \Omega(h_j)$$

де:

- L(y_i, {y}_i) - функція втрати, яка вимірює різницю між спостережуваною значенням y_i та прогнозованою значенням {y}_i
- ω(h_j) - регуляризаційна функція, яка контролює складність кожного дерева рішень

В алгоритмі XGB Regressor використовується оптимізаційний процес, який знаходить оптимальні значення для $h_{j(x)\text{та}}\backslash\omega(h_j)$ на кожному кроці для покращення прогнозів. В результаті, XGB Regressor будує ансамбль дерев рішень, які доповнюють одне одного, покращуючи точність прогнозу і здатність до узагальнення моделі. Цей алгоритм використовує градієнтний спуск для пошуку оптимальних значень параметрів на кожному кроці, що дозволяє ефективно здійснювати навчання моделі.

Узагальнюючи, лінійна регресія підходить для простих лінійних залежностей, поліноміальна регресія розширює можливості до нелінійних залежностей, дерева рішень використовуються для інтерпретації та узагальнення, випадковий ліс забезпечує кращу здатність до узагальнення

завдяки ансамблю дерев, а XGB Regressor забезпечує високу точність і широко використовується в практичних завданнях машинного навчання.

Розглянули три різні моделі машинного навчання, а саме лінійну регресію, поліноміальну регресію та випадковий ліс. Окрім цього, ми також згадали про XGB Regressor, який є ансамблевою моделлю, заснованою на градієнтному бустингу для регресії. Вказали, що XGB Regressor використовує покрокове покращення моделі шляхом додавання нових слабких моделей для корекції помилок попередніх моделей.

Щодо прогнозування часу доставки товарів на основі машинного навчання, у цьому випадку можна використати моделі регресії, такі як лінійна регресія або поліноміальна регресія, для побудови моделі, яка здатна оцінити час доставки на основі вхідних ознак, таких як відстань, вага, трафік тощо. З використанням налагодження параметрів та оптимізації, можна досягти точніших прогнозів.

Ансамблеві моделі, такі як випадковий ліс або XGB Regressor, також можуть бути використані для прогнозування часу доставки. Вони здатні адаптуватись до складних нелінійних залежностей та працювати добре, навіть з великою кількістю ознак. Випадковий ліс, наприклад, побудує багато дерев рішень, які доповнюють одне одного, а XGB Regressor використовує градієнтний бустинг для послідовного покращення моделі.

## 2.3 Алгоритм прогнозування часу доставки товарів на основі машинного навчання

Алгоритм прогнозування часу доставки товарів може бути побудований за допомогою таких кроків та у вигляді блок-схеми (див.дод.А):

Крок 1. Збір даних: Зібрати історичні дані про доставку товарів, які включають інформацію про час доставки та відповідні фактори, які можуть впливати на цей час. Ці дані можуть включати такі фактори, як відстань

між місцем відправлення та доставки, тип транспортного засобу, час року, обсяги замовлень тощо.

Крок 2. Експлораторний аналіз даних (EDA): Виконати EDA для розуміння взаємозв'язку між різними факторами та часом доставки. Візуалізація даних може допомогти виявити кореляції та інші залежності між змінними.

Крок 3. Підготовка даних: Провести підготовку даних, щоб вони були готові для використання в моделі. Це може включати видалення відсутніх значень, нормалізацію числових змінних та кодування категоріальних змінних.

Крок 4. Вибір моделі: Вибрати підходящу модель для прогнозування часу доставки на основі характеристик даних та вимог до точності. У вашому випадку можна спробувати такі моделі, як Linear Regression, Polynomial Regression, Decision Tree, Random Forest, XGB Regressor.

Крок 5. Розділити дані: Розділити дані на тренувальний набір та тестовий набір. Тренувальний набір буде використовуватись для навчання моделі, а тестовий набір - для оцінки її продуктивності.

Крок 6. Навчання моделі: Навчати обрану модель за допомогою тренувального набору даних. Модель буде виявляти залежності між вхідними змінними (факторами) та вихідною змінною (часом доставки),

Крок 7. Оцінка моделі: Оцінити продуктивність моделі на тестовому наборі даних. Використовуйте показники, такі як середня квадратична помилка (Mean Squared Error, MSE) або середня абсолютна помилка (Mean Absolute Error, MAE), щоб оцінити точність моделі.

Оцінка моделі на тестовому наборі даних є важливим кроком для визначення її продуктивності та точності прогнозування часу доставки товарів. Для оцінки точності моделі можна використовувати такі показники, як середня квадратична помилка (Mean Squared Error, MSE) або середня абсолютна помилка (Mean Absolute Error, MAE).

Середня квадратична помилка (Mean Squared Error, MSE): MSE вимірює середнє значення квадрата різниці між прогнозованими значеннями і фактичними значеннями. Чим менше значення MSE, тим краще модель прогнозує. Формула для обчислення MSE:

$$MSE = \left(\frac{1}{n}\right) * \Sigma(y_i - \hat{y}_i)^2,$$

де n - кількість спостережень, $y_i$ - фактичне значення, $\hat{y}_i$ - прогнозоване значення.

Середня абсолютна помилка (Mean Absolute Error, MAE): MAE вимірює середнє абсолютне значення різниці між прогнозованими значеннями і фактичними значеннями. Він вимірює, на скільки в середньому модель помиляється у своїх прогнозах. Чим менше значення MAE, тим краще модель прогнозує. Формула для обчислення MAE:

$$MAE = \left(\frac{1}{n}\right) * \Sigma|y_i - \hat{y}_i|.$$

Після того, як модель навчена на тренувальних даних, вона застосовується до тестового набору даних для прогнозування часу доставки. Потім розраховуються значення MSE і MAE на основі прогнозованих та фактичних значень.

Крок 8. Підбір гіперпараметрів: Провести пошук оптимальних гіперпараметрів моделі для покращення її продуктивності. Це може включати використання крос-валідації або решітчатого пошуку (grid search) для визначення найкращих значень гіперпараметрів моделі.

Крок 9. Тренування моделі з оптимальними гіперпараметрами: Навчайте модель з оптимальними гіперпараметрами на тренувальному наборі даних.

Крок 10. Прогнозування: Використовуйте навчену модель для прогнозування часу доставки на нових даних або замовленнях, які потребують прогнозу.

Крок 11. Оцінка та оптимізація моделі: Оцініть прогнози моделі та порівняйте їх з фактичними значеннями. Якщо точність недостатня, виконайте подальшу оптимізацію моделі, можливо, змінивши функцію втрати, додавши нові фактори або використовуючи інші алгоритми.

Крок 12. Розгортання моделі: Після досягнення задовільних результатів розгорніть модель у виробничому середовищі та використовуйте її для прогнозування часу доставки товарів.

Цей алгоритм надає загальну структуру для побудови моделі прогнозування часу доставки товарів. Зазначені алгоритми (Linear Regression, Polynomial Regression, Decision Tree, Random Forest, XGB Regressor) можуть бути використані окремо або в поєднанні залежно від характеристик даних та вашого специфічного випадку.

# 3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ ЧАСУ ДОСТАВКИ ТОВАРІВ

## 3.1 Дослідження та підготовка набору даних

Для реалізації алгоритму, що прогнозування часу доставки товарів обрано набір даних NYC Restaurants Data - Food Ordering and Delivery [54]. Даний набір даних зі сторінки www.kaggle.com містить зразкові дані з онлайн сервісу замовлення та доставки їжі в ресторани Нью-Йорка. Даний сервіс надає можливість замовляти їжу з кількох ресторанів через один додаток на смартфоні. Після підтвердження замовлення рестораном, до нього приходить кур'єр з компанії доставки, який отримує замовлення та відправляється до ресторану, де отримує замовлення та вирушає до клієнта.

Метою роботи є аналіз цих даних з використанням методів машинного навчання для прогнозування часу доставки замовлень. Результати дослідження допоможуть вдосконалити сервіс доставки та покращити задоволення клієнтів.

Спершу проведемо дослідження нашого набору даних, для цього проведемо розвідувальний аналіз.

Наведений набір даних представлений у форматі DataFrame з 1898 рядків і 9 стовпців. Кожен стовпець відповідає певному аспекту замовлення їжі з ресторану. Опис стовпців наведений нижче:

1. order_id: унікальний ідентифікатор замовлення (ціле число).
2. customer_id: унікальний ідентифікатор клієнта (ціле число).
3. restaurant_name: назва ресторану (рядок).
4. cuisine_type: тип кухні ресторану (рядок).
5. cost_of_the_order: вартість замовлення (число з плаваючою комою).
6. day_of_the_week: день тижня, коли було зроблено замовлення (рядок).
7. rating: рейтинг замовлення, присвоєний клієнтом (рядок).

8. food_preparation_time: час підготовки їжі в ресторані (ціле число).
9. delivery_time: час доставки замовлення (ціле число).

Оскільки є змінна назва ресторану та змінна група кузенів, які мають багато різних значень, нам потрібно згрупувати ці змінні, щоб використовувати їх. Використаємо середню доставку для групування цих змінних. Та у змінній restaurant_name, згрупуємо їх наступним чином:

Значення 0: назва_ресторану із середнім часом доставки більше або дорівнює 28.

Значення 1: більше або дорівнює 23 і менше 28.

Значення 2: більше або дорівнює 20 і менше 23.

Значення 3: менше 20.

Змінну cousine_type, згрупуємо наступним чином:

Значення 0: restaurant_name із середнім часом доставки більше або рівним 26.

Значення 1: вище або дорівнює 25 і нижче 26.

Значення 2: вище або дорівнює 22 і нижче 24.

Значення 3: нижче 22.

Та проведемо перевірку кореляції між нашими змінними. Результат (Рис. 3.1) кореляційного аналізу представлений у вигляді матриці кореляції. Кожний елемент матриці показує ступінь кореляції між двома змінними. Значення кореляції може приймати значення від -1 до 1, де -1 вказує на сильну негативну кореляцію, 1 - на сильну позитивну кореляцію, а значення близькі до 0 вказують на слабку або відсутню кореляцію.

Основний аналіз матриці кореляції включає наступне:

1. cost_of_the_order і day_of_the_week мають слабку позитивну кореляцію (0.02). Це означає, що вартість замовлення має незначний зв'язок з днем тижня.
2. rating і day_of_the_week також мають слабку позитивну кореляцію (0.02), що вказує на незначний зв'язок між рейтингом і днем тижня.

3. delivery_time і day_of_the_week мають помірну негативну кореляцію (-0.53). Це може свідчити про те, що час доставки залежить від дня тижня.
4. delivery_time і delay_group мають помірну негативну кореляцію (-0.23). Це може означати, що більш тривалі затримки пов'язані з більшим часом доставки.
5. delay_group і cousine_group мають слабку позитивну кореляцію (0.09), що може вказувати на зв'язок між групою затримки та групою кухні.

Це лише загальний аналіз кореляцій, і для більш детального розуміння взаємозв'язків між змінними потрібно розглядати кожен випадок окремо та додатково проводити статистичний аналіз.

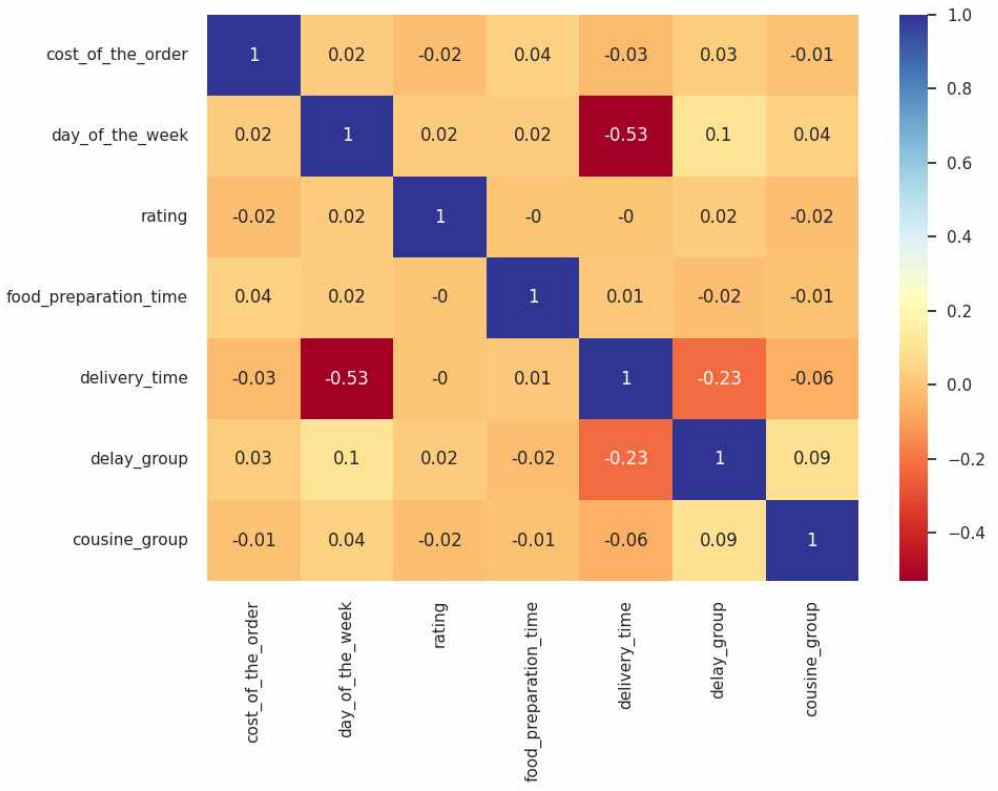



Рисунок 3.1 - Кореляція між параметрами

Проведемо візуалізацію 10 найпопулярніших ресторанів на основі кількості замовлень (Рис. 3.2). Кожний стовпчик представляє кількість замовлень для відповідного ресторану, де більша паличка відповідає більшій кількості замовлень.

Далі виведено топ-10 типів кухні, які отримують найбільше замовлень. Аналізуючи результат (Рис.3.3), можна зробити наступні спостереження: Найпопулярнішими типами кухні в датасеті є "American" (584 замовлення) і "Japanese" (470 замовлень), оскільки вони мають найбільшу кількість замовлень. "Italian" і "Chinese" також популярні типи кухні з великою кількістю замовлень (298 і 215 відповідно). "Mexican", "Indian" і "Middle Eastern" мають помірну кількість замовлень (77, 73 і 49 відповідно). "Mediterranean", "Thai", "French" і "Southern" мають меншу кількість замовлень (46, 19, 18 і 17 відповідно). "Korean", "Spanish" і "Vietnamese" мають найменшу кількість замовлень серед усіх типів кухні (13, 12 і 7 відповідно).

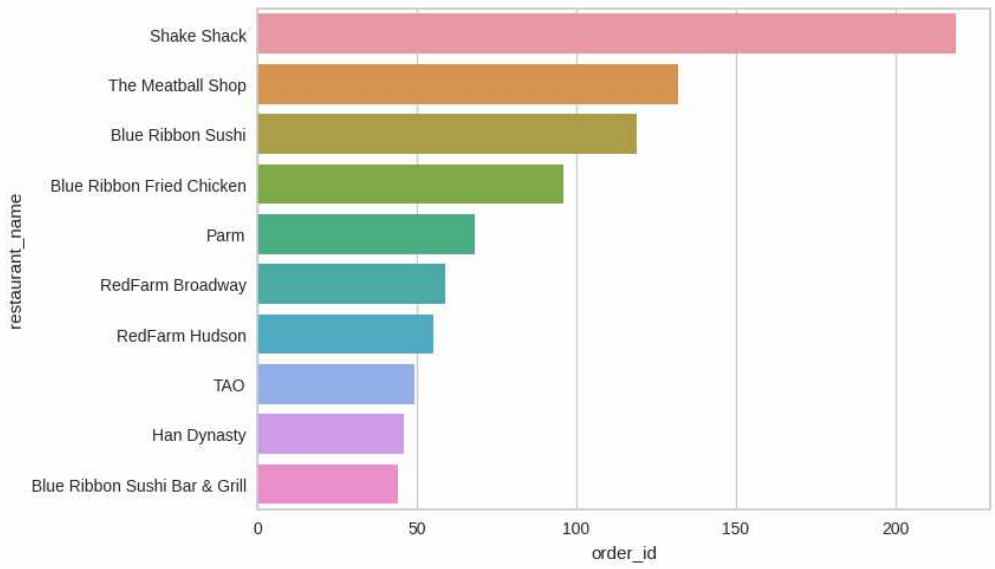



Рисунок 3.2 - 10 найпопулярніших ресторанів на основі кількості замовлень

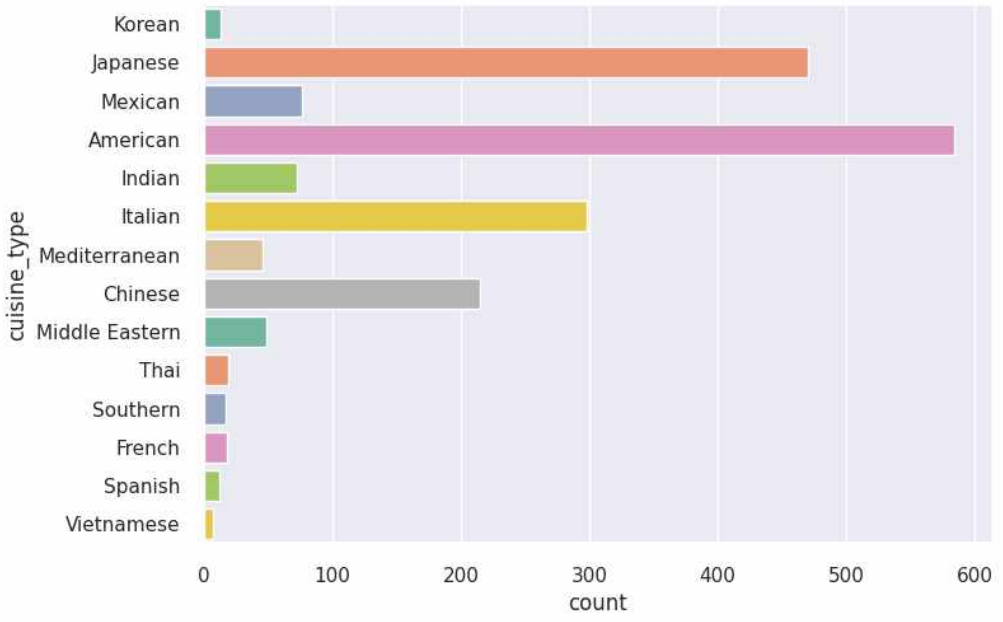



Рисунок 3.3 - 10 типів кухні, які отримують найбільше замовлень

Далі виведемо стовпчикову діаграму (Рис.3.4) з розподілом замовлень за днями тижня і стовпчикова діаграма з розподілом замовлень за рейтингом. Аналізуючи результати, можна зробити наступні спостереження:

День тижня:

- Найбільша кількість замовлень (1351) здійснюється вихідними днями (Weekend).
- У будні дні (Weekday) кількість замовлень становить 547.

Рейтинг:

- Найбільша кількість замовлень (736) не має вказаного рейтингу (Not given).
- Кількість замовлень з рейтингом 5 становить 588.
- Кількість замовлень з рейтингом 4 становить 386.
- Кількість замовлень з рейтингом 3 становить 188.

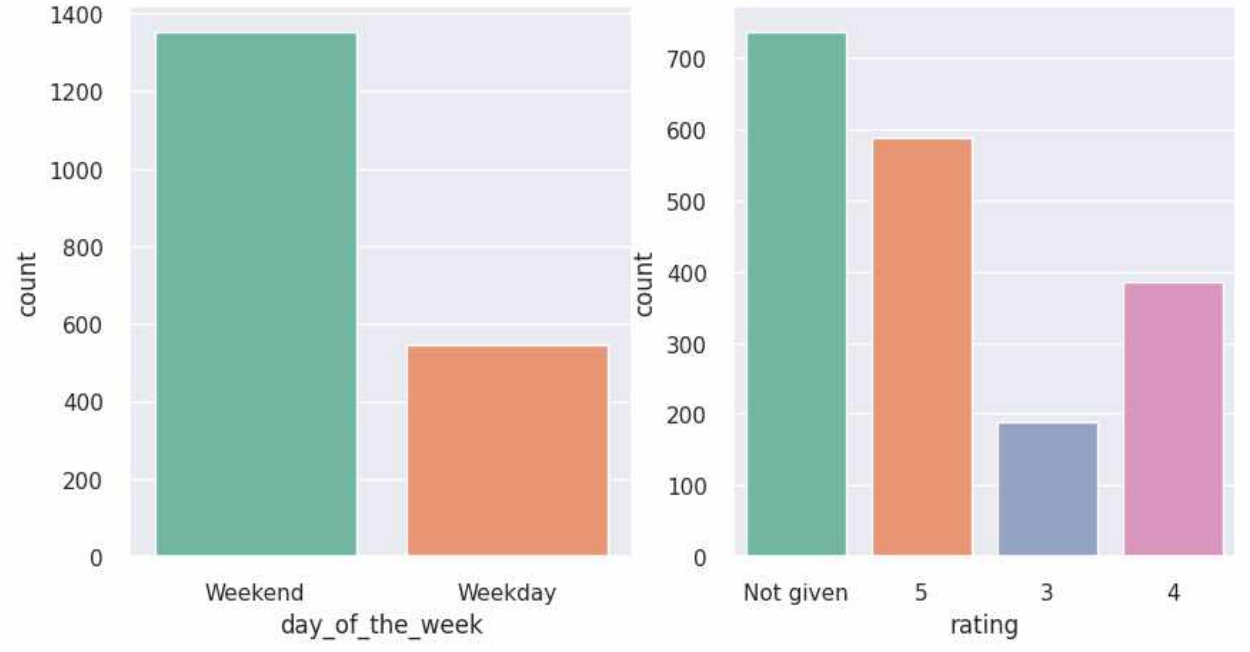



Рисунок 3.4 - Розподіл замовлень за днями тижня і рейтингом

Далі виведемо діаграму (Рис.3.5) з трьома гістограмами, які відображають розподіл трьох змінних: 'cost_of_the_order' (вартість замовлення), 'food_preparation_time' (час підготовки їжі) та 'delivery_time' (час доставки).

- У першому графіку показано розподіл вартості замовлення (cost_of_the_order).
- У другому графіку показано розподіл часу підготовки їжі (food_preparation_time).
- У третьому графіку показано розподіл часу доставки (delivery_time).

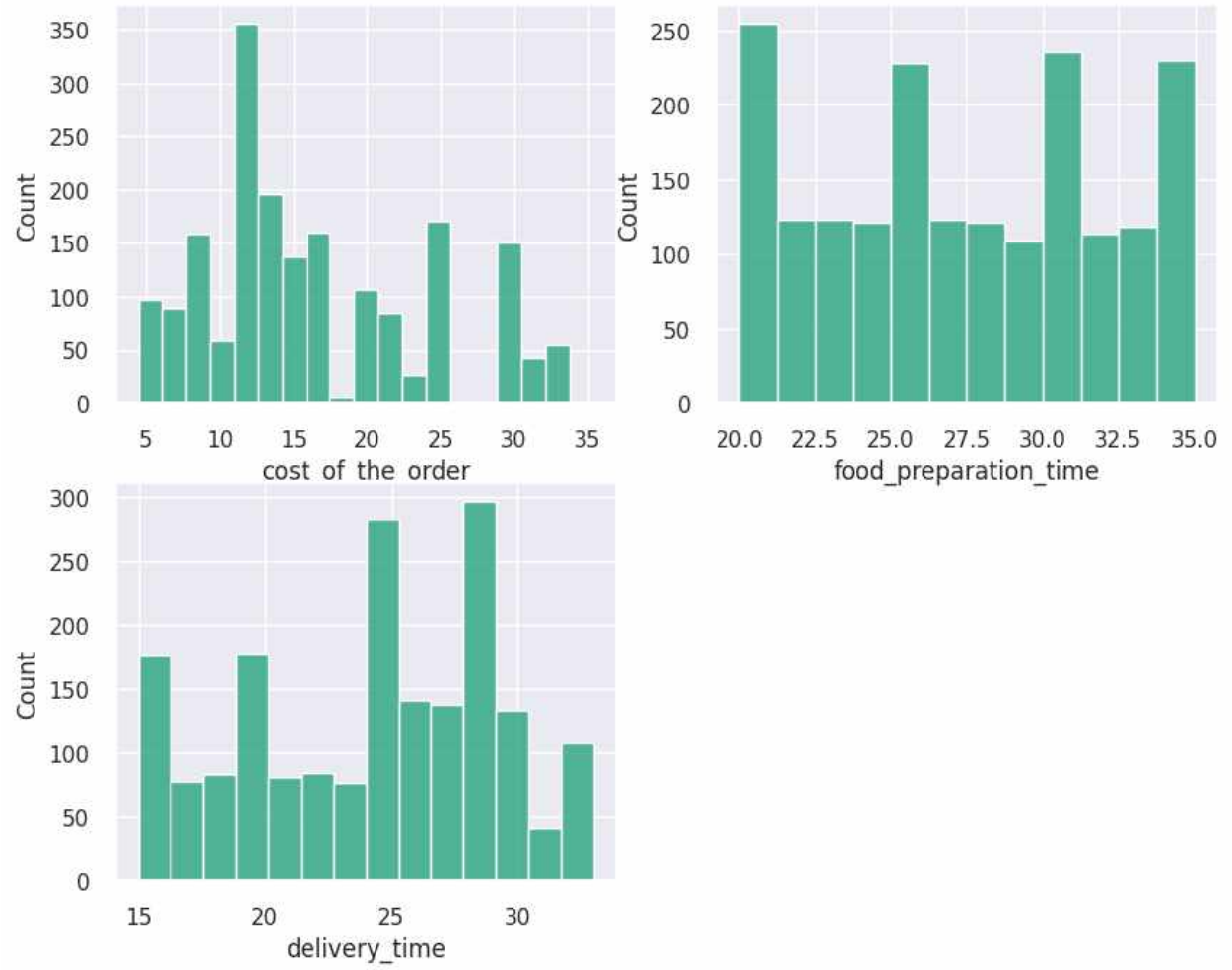



Рисунок 3.5 - Розподіл трьох змінних: 'cost_of_the_order', 'food_preparation_time' та 'delivery_time'

Сформуємо чотири графіки (див.дод.Б), які відображають залежність між змінною "delivery_time" (час доставки) та іншими змінними: "delay_group" (група затримки), "cousine_group" (група кухні), "day_of_the_week" (день тижня) та "rating" (рейтинг).

У першому графіку (див. Рис.3.6), за допомогою sns.boxplot, показано вплив групи затримки на час доставки. Замовлення з різними групами затримки відображені у відповідних ящиках, а вертикальна вісь представляє час доставки.

У другому графіку, також за допомогою sns.boxplot, відображено залежність між групою кухні і часом доставки. Різні групи кухні представлені на горизонтальній вісі, а вертикальна вісь показує час доставки.

У третьому графіку, за допомогою sns.boxplot, показано, як час доставки залежить від дня тижня. Кожен день тижня представлений на горизонтальній вісі, а вертикальна вісь відображає час доставки.

У четвертому графіку, також за допомогою sns.boxplot, показано вплив рейтингу на час доставки. Різні рейтинги представлені на горизонтальній вісі, а вертикальна вісь відображає час доставки.

Графіки boxplot дозволяють оцінити центральні та варіаційні характеристики часу доставки для різних категорій змінних. За допомогою цих графіків можна зробити висновки про тенденції та розподіл часу доставки в залежності від різних факторів, що досліджуються.

Порівнюючи змінні з цільовою змінною, бачимо, що в групі ресторанів безумовно вдалося розділити ресторани за часом доставки, ще однією цікавою змінною є змінна день тижня, ми бачимо, що замовлення, зроблені протягом тижня, виконуються швидше.

## 3.2 Навчання моделей

Спершу підготуємо дані для подальшого застосування моделі прогнозування часу доставки (delivery_time).

```
df = df.drop(['restaurant_average', 'cuisine_type_average',
              'order_id', 'customer_id', 'restaurant_name', 'cuisine_type'], axis = 1)
```

```
from sklearn.preprocessing import LabelEncoder

label_encoder_day_of_the_week = LabelEncoder()
label_encoder_rating = LabelEncoder()

df['day_of_the_week'] = label_encoder_day_of_the_week.fit_transform(df['day_of_the_week'])
df['rating'] = label_encoder_rating.fit_transform(df['rating'])
```

```
X = df.drop('delivery_time', axis = 1)
X = X.values
y = df['delivery_time']
```

У першому рядку коду використовується метод .drop() для видалення деяких стовпців з DataFrame df. Стовпці, які видаляються, вказані в списку ['restaurant_average', 'cuisine_type_average', 'order_id', 'customer_id',

'restaurant_name', 'cuisine_type']. В результаті ці стовпці будуть видалені з DataFrame df.

Після цього, імпортується клас LabelEncoder з модуля sklearn.preprocessing. LabelEncoder використовується для перетворення категоріальних змінних у числові значення.

Створюються два екземпляри LabelEncoder: label_encoder_day_of_the_week і label_encoder_rating.

Стовпець 'day_of_the_week' з DataFrame df перетворюється на числові значення за допомогою методу .fit_transform() label_encoder_day_of_the_week.fit_transform(df['day_of_the_week']). Аналогічно, стовпець 'rating' перетворюється за допомогою label_encoder_rating.fit_transform(df['rating']).

Далі, з DataFrame df видаляється стовпець 'delivery_time' за допомогою .drop() і результат зберігається в змінній X. Метод .values перетворює DataFrame X на масив NumPy.

Стовпець 'delivery_time' стає цільовою змінною і зберігається в змінній y.

Отже, після виконання цього коду, дані підготовлені для подальшого застосування моделі прогнозування, де 'delivery_time' є цільовою змінною, а решта стовпців використовуються як ознаки (X) для навчання моделі.

Далі використовуючи клас StandardScaler з модуля sklearn.preprocessing проводимо стандартизації ознак і цільової змінної.

```
1 from sklearn.preprocessing import StandardScaler
2 scaler = StandardScaler()
3 X_standard = scaler.fit_transform(X)
4 y_standard = scaler.fit_transform(y.values.reshape(-1,1))
```

Спочатку імпортується StandardScaler з модуля sklearn.preprocessing.

Створюється екземпляр StandardScaler з назвою scaler.

Далі, метод .fit_transform() викликається на об'єкті scaler і застосовується до ознак, збережених у змінній X. Результат стандартизації ознак зберігається

у змінній X_standard. Стандартизація означає, що кожна ознака має середнє значення 0 і стандартне відхилення 1.

Так само, метод .fit_transform() викликається на об'єкті scaler і застосовується до цільової змінної, яка знаходиться в змінній y. Однак, перед застосуванням методу .fit_transform(), цільова змінна y перетворюється з одновимірного масиву на двовимірний масив за допомогою .values.reshape(-1, 1). Результат стандартизації цільової змінної зберігається в змінній y_standard.

Отже, після виконання цього коду, ознаки (X) і цільова змінна (y) будуть стандартизовані за допомогою StandardScaler і зберігатимуться в X_standard і y_standard відповідно.

```
1 from sklearn.model_selection import train_test_split
2 X_train,X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 0)
```

У даному коді використовується функція train_test_split з модуля sklearn.model_selection для розбиття набору даних на тренувальну та тестову вибірки.

Спочатку імпортується train_test_split з модуля sklearn.model_selection.

Потім викликається функція train_test_split з наступними параметрами:

- X і y - ознаки і цільова змінна, які мають бути розділені.
- test_size = 0.3 - вказує співвідношення розміру тестової вибірки до загального розміру набору даних. У даному випадку, тестова вибірка буде складати 30% від загального набору даних.
- random_state = 0 - встановлює випадкове насіння для повторюваності результатів розбиття даних. Задане значення 0 гарантує, що при кожному запуску коду будуть отримані однакові результати розбиття.

Результатом виконання цього коду є розбиття набору даних на тренувальну вибірку (X_train, y_train) і тестову вибірку (X_test, y_test) згідно вказаних параметрів.

Далі переходимо до навчання моделей класифікацій: Linear Regression, Polynomial Regression, Decision Tree, Random Forest, XGB Regressor.

```python
from sklearn.linear_model import LinearRegression
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
lr_normal_score_train = lr_model.score(X_train, y_train)
lr_normal_score_test = lr_model.score(X_test, y_test)
previsoes = lr_model.predict(X_test)
mae_lr_normal = mean_absolute_error(y_test, previsoes)
rmse_lr_normal = np.sqrt(mean_squared_error(y_test, previsoes))

print('Train :', lr_normal_score_train)
print('Test :', lr_normal_score_test)
print('Mean Absolute Error :', mae_lr_normal)
print('Root Mean Square Error :', rmse_lr_normal)
```

У даному коді використовується модель лінійної регресії (LinearRegression) з модуля sklearn.linear_model для навчання на тренувальній вибірці і прогнозування на тестовій вибірці.

1. Імпортується LinearRegression з модуля sklearn.linear_model.
2. Створюється об'єкт моделі lr_model за допомогою LinearRegression().
3. Застосовується метод .fit(X_train, y_train) на lr_model, щоб навчити модель на тренувальній вибірці X_train та y_train.
4. Викликаються методи .score(X_train, y_train) та .score(X_test, y_test) для обчислення коефіцієнта детермінації моделі на тренувальній та тестовій вибірках відповідно. Результати зберігаються у змінних lr_normal_score_train та lr_normal_score_test.
5. Застосовується метод .predict(X_test) на lr_model, щоб зробити прогнози на тестовій вибірці. Прогнозовані значення зберігаються у змінній previsoes.
6. Викликаються функції mean_absolute_error та np.sqrt(mean_squared_error) для обчислення середньої абсолютної похибки (mae_lr_normal) та квадратного кореня середньоквадратичної похибки (rmse_lr_normal) між прогнозованими значеннями та фактичними значеннями у тестовій вибірці.
7. Виводяться результати на екран за допомогою функції print(). Виводяться коефіцієнти детермінації для тренувальної та тестової вибірок, середня абсолютна похибка та квадратний корінь середньоквадратичної похибки.

На основі цього сценарію LinearRegression проводиться навчання всіх інших класифікаторів: Polynomial Regression, Decision Tree, Random Forest, XGB Regressor. Опишемо лише детальніше внесені параметри налаштувань.

```
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(degree = 2)
X_poly_train = poly.fit_transform(X_train)
X_poly_test = poly.transform(X_test)
lr_poly = LinearRegression()
lr_poly.fit(X_poly_train, y_train)
lr_poly_normal_score_train = lr_poly.score(X_poly_train, y_train)
lr_poly_normal_score_test = lr_poly.score(X_poly_test, y_test)
previsoes = lr_poly.predict(X_poly_test)
mae_poly_normal = mean_absolute_error(y_test, previsoes)
rmse_poly_normal = np.sqrt(mean_squared_error(y_test, previsoes))
```

У даному коді додатково використовується клас PolynomialFeatures з модуля sklearn.preprocessing для створення поліноміальних ознак з вхідних даних.

1. Імпортується PolynomialFeatures з модуля sklearn.preprocessing.
2. Створюється об'єкт poly класу PolynomialFeatures з параметром degree=2, що вказує на створення поліноміальних ознак другого степеня.
3. Застосовується метод .fit_transform(X_train) на об'єкті poly для створення поліноміальних ознак з тренувальної вибірки X_train. Результат зберігається у змінній X_poly_train.
4. Застосовується метод .transform(X_test) на об'єкті poly для створення поліноміальних ознак з тестової вибірки X_test. Результат зберігається у змінній X_poly_test.
5. Створюється об'єкт моделі lr_poly за допомогою LinearRegression().
6. Застосовується метод .fit(X_poly_train, y_train) на lr_poly, щоб навчити модель на тренувальній вибірці з поліноміальними ознаками X_poly_train та y_train.
7. Викликаються методи .score(X_poly_train, y_train) та .score(X_poly_test, y_test) для обчислення коефіцієнта детермінації моделі з поліноміальними ознаками на тренувальній та тестовій вибірках відповідно. Результати зберігаються у змінних lr_poly_normal_score_train та lr_poly_normal_score_test.

8. Застосовується метод .predict(X_poly_test) на lr_poly, щоб зробити прогнози на тестовій вибірці з поліноміальними ознаками. Прогнозовані значення зберігаються у змінній previsoes.
9. Обчислюються середня абсолютна похибка (mae_poly_normal) та квадратний корінь середньоквадратичної похибки (rmse_poly_normal) між прогнозованими значеннями та факти

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import RandomizedSearchCV

min_split = np.array([2, 3, 4, 5, 6, 7])
max_nvl = np.array([3, 4, 5, 6, 7, 9, 11])
alg = ['squared_error', 'friedman_mse', 'absolute_error', 'poisson']
values_grid = {'min_samples_split': min_split, 'max_depth': max_nvl, 'criterion': alg}

model = DecisionTreeRegressor()
gridDecisionTree = GridSearchCV(estimator = model, param_grid = values_grid, cv = 5, n_jobs = -1)
gridDecisionTree.fit(X_train, y_train)

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

У даному коді використовується модель DecisionTreeRegressor з модуля sklearn.tree для побудови регресійного дерева.

1. Імпортується DecisionTreeRegressor з модуля sklearn.tree.
2. Визначаються змінні min_split, max_nvl, та alg зі списками можливих значень для параметрів моделі.
3. Створюється об'єкт моделі model типу DecisionTreeRegressor().
4. Визначається словник values_grid зі значеннями параметрів, які будуть перебиратись під час гіперпараметричної настройки моделі.
5. Створюється об'єкт gridDecisionTree класу GridSearchCV для гіперпараметричної настройки моделі model. Встановлюються параметри estimator (модель), param_grid (словник зі значеннями параметрів), cv (кількість фолдів у перехресній перевірці), та n_jobs (кількість процесів, які будуть використовуватись для пошуку).
6. Застосовується метод .fit(X_train, y_train) на об'єкті gridDecisionTree, щоб навчити модель з різними комбінаціями гіперпараметрів.
7. Виводяться оптимальні значення гіперпараметрів та кращий результат, отриманий під час гіперпараметричної настройки.

```python
from sklearn.ensemble import RandomForestRegressor

parameters = {'n_estimators': [100],
              'max_depth': [3, 4, 5, 6, 7, 9, 11],
              'min_samples_split': [2, 3, 4, 5, 6, 7],
              'criterion': ['squared_error', 'absolute_error', 'friedman_mse']
             }

model = RandomForestRegressor()
gridRandomForest = RandomizedSearchCV(model,parameters, cv = 2)
gridRandomForest.fit(X_train, y_train)

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

У даному коді використовується модель RandomForestRegressor з модуля sklearn.ensemble для побудови випадкового лісу регресорів.

1. Імпортується RandomForestRegressor з модуля sklearn.ensemble.
2. Визначається словник parameters зі значеннями параметрів моделі, які будуть перебиратись під час гіперпараметричної настройки.
3. Створюється об'єкт моделі model типу RandomForestRegressor().
4. Створюється об'єкт gridRandomForest класу RandomizedSearchCV для гіперпараметричної настройки моделі model. Встановлюються параметри estimator (модель), param_distributions (словник зі значеннями параметрів), та cv (кількість фолдів у перехресній перевірці).
5. Застосовується метод .fit(X_train, y_train) на об'єкті gridRandomForest, щоб навчити модель з різними комбінаціями гіперпараметрів.
6. Виводяться оптимальні значення гіперпараметрів та кращий результат, отриманий під час гіперпараметричної настройки.

```python
from xgboost.sklearn import XGBRegressor

parameters = {'learning_rate': [0.01, 0.03, 0.05, 0.1, 0.3, 0.5, 0.8],
              'max_depth': [1 , 2, 3, 4, 5, 6, 7, 8, 9],
              'min_child_weight': [1, 3, 5, 7, 9],
              'subsample': [0.1, 0.3, 0.5, 0.7, 0.9],
              'colsample_bytree': [0.1, 0.3, 0.5, 0.7, 0.9],
              'n_estimators': [500],
              'gamma': [0.1, 0.3, 0.5, 0.7, 0.9],
              'reg_alpha': [0.1, 0.3, 0.5, 0.7, 0.9],
              'reg_lambda': [0.1, 0.3, 0.5, 0.7, 0.9]
             }

model = XGBRegressor()
xgb_grid = RandomizedSearchCV(model,parameters, cv = 3, n_jobs = -1)
xgb_grid.fit(X_train, y_train)

print('Score: ', xgb_grid.best_score_)
print('Params: ', xgb_grid.best_params_)
```

У даному коді використовується модель XGBRegressor з модуля xgboost.sklearn для побудови градієнтного бустингу.

1. Імпортується XGBRegressor з модуля xgboost.sklearn.
2. Визначається словник parameters зі значеннями гіперпараметрів моделі, які будуть перебиратись під час гіперпараметричної настройки.
3. Створюється об'єкт моделі model типу XGBRegressor().
4. Створюється об'єкт xgb_grid класу RandomizedSearchCV для гіперпараметричної настройки моделі model. Встановлюються параметри estimator (модель), param_distributions (словник зі значеннями гіперпараметрів), cv (кількість фолдів у перехресній перевірці), та n_jobs (кількість паралельних робочих процесів).
5. Застосовується метод .fit(X_train, y_train) на об'єкті xgb_grid, щоб навчити модель з різними комбінаціями гіперпараметрів.
6. Виводиться кращий результат, отриманий під час гіперпараметричної настройки, та оптимальні значення гіперпараметрів.

Отже, проведено навчання моделей класифікацій: Linear Regression, Polynomial Regression, Decision Tree, Random Forest, XGB Regressor.

## 3.3 Оцінка отриманих моделей

Аналізуючи результати (Рис.3.6), можна зробити наступні спостереження:

1. Linear Regression: Модель має низьку тренувальну та тестову оцінку (Train Score: 0.618374, Test Score: 0.61931), що свідчить про слабу здатність моделі узагальнювати дані. MAE (Mean Absolute Error) дорівнює 3.322809, що означає середню абсолютну похибку у прогнозуванні часу доставки. RMSE (Root Mean Square Error) становить 3.992652, що є середнім квадратичним відхиленням прогнозів від фактичних значень.

2. Polynomial Regression: Модель також має низькі тренувальну та тестову оцінки (Train Score: 0.634776, Test Score: 0.610065). Це свідчить про обмежену здатність поліноміальної регресії узагальнювати дані. MAE дорівнює 3.318269, а RMSE становить 4.019676.
3. Decision Tree: На жаль, немає вказівки на тренувальну або тестову оцінку для Decision Tree моделі. Це може означати, що модель не була правильно побудована або що її тренування та тестування не були виконані.
4. Random Forest: Аналогічно до Decision Tree, немає вказівки на тренувальну та тестову оцінку для Random Forest моделі. Це може вказувати на проблеми з побудовою моделі або відсутність тренування та тестування.
5. XGB Regressor: Модель має тренувальну оцінку 0.705202. Це також є низькою оцінкою, що вказує на обмежену здатність моделі XGB Regressor узагальнювати дані. Тут також відсутні вказівки на тестову оцінку, MAE та RMSE.

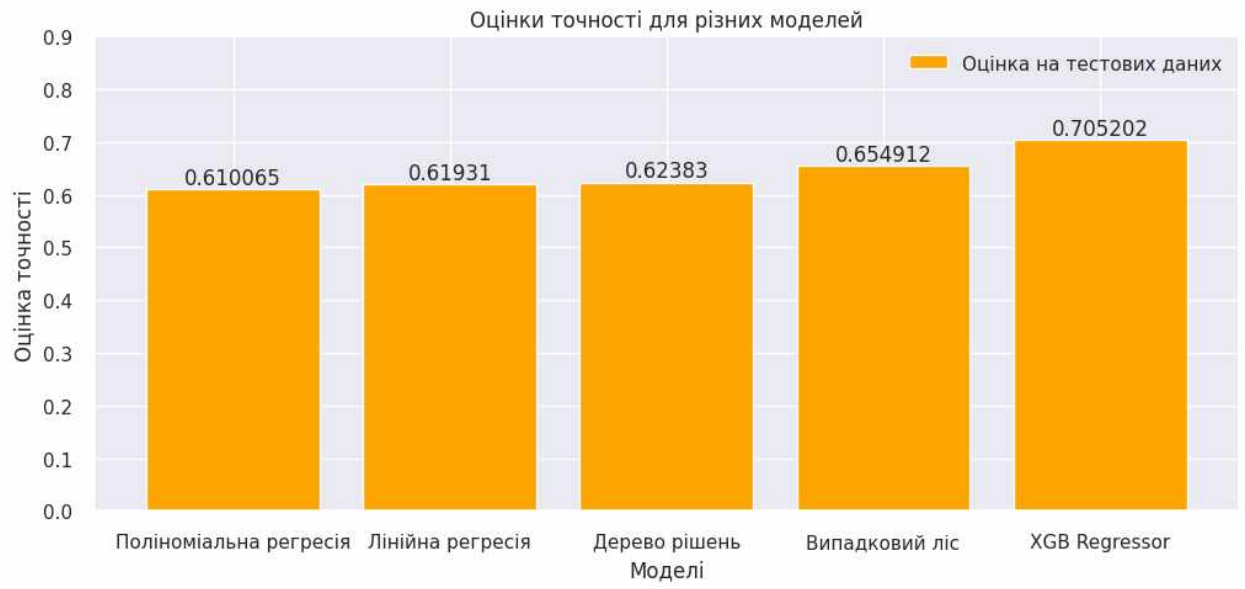



Рисунок 3.6 - Оцінки точності для різних моделей

Запропоновані моделі (Linear Regression, Polynomial Regression, Decision Tree, Random Forest, XGB Regressor) демонструють низьку точність прогнозування часу доставки. Це вказує на те, що ці моделі можуть бути недостатньо складними або не враховувати всі важливі фактори, що впливають на час доставки.

Спробуємо вивести прогнозне значення для рядка 15 та 42 (Таблиця 3.1):

Таблиця 3.1. Прогнозне значення для рядка 15 та 42

| | Рядок | Значення | Прогноз |
|---|---|---|---|
| 0 | 15 | [ 9.07 0. 1. 21. 1. 1. ] | 28.2927303314209 |
| 1 | 42 | [12.56 1. 1. 20. 1. 1. ] | 22.395217895507812 |

Аналізуючи результати (див. таблиця 3.1), можна зробити такі спостереження:

1. Рядок 15:
    - Час доставки: 21
    - Прогнозований час доставки: 28.2927

Прогнозований час доставки (28.2927) більший за фактичний час доставки (21). Це означає, що модель передбачає тривалішу доставку, ніж насправді.

2. Рядок 42:
    - Час доставки: 20
    - Прогнозований час доставки: 22.3952

Прогнозований час доставки (22.3952) близький до фактичного часу доставки (20). Це означає, що модель в цьому випадку зробила досить точний прогноз.

Аналізуючи результати прогнозу моделі XGB Regressor, бачимо, що у деяких випадках модель недооцінює час доставки (рядок 15), тоді як у інших випадках вона зробила близький до фактичного прогноз (рядок 42). Точність

моделі може бути покращена шляхом додаткової настройки параметрів або використання інших алгоритмів.

Низькі значення точності можуть мати декілька причин. По-перше, може бути необхідно використовувати більш складні моделі з більшою кількістю параметрів для кращої апроксимації даних. По-друге, може бути необхідно збільшити обсяг доступних даних для навчання моделі, оскільки більші обсяги даних часто призводять до кращих результатів.

Для поліпшення точності прогнозування можна спробувати наступні підходи, які будуть проведені в наступних дослідженнях:

1. Використання більш складних моделей, які можуть краще апроксимувати складні залежності в даних. Наприклад, можна спробувати використати нейронні мережі або градієнтний бустінг.
2. Збільшення обсягу навчальних даних. Збір більшої кількості даних або використання додаткових джерел даних може допомогти моделі краще узагальнити та зробити більш точні прогнози.
3. Оптимізація гіперпараметрів моделі. Підбір оптимальних значень гіперпараметрів може покращити точність моделі. Це можна зробити шляхом використання крос-валідації та пошуку гіперпараметрів.
4. Врахування додаткових ознак або інженерія ознак. Додавання нових ознак або використання вищого порядку взаємодій між ознаками може покращити здатність моделі узагальнювати залежності у даних. Наприклад, можна додати взаємодію між двома ознаками або використати поліноміальні ознаки.
5. Застосування ансамблевих методів. Ансамблеві методи, такі як випадковий ліс або градієнтний бустінг, можуть комбінувати прогнози кількох моделей, що дозволяє покращити точність. Можна спробувати використовувати ансамблеві методи для прогнозування часу доставки.
6. Перевірка якості та відладка моделі. Важливо перевірити якість прогнозів моделі, а також провести відладку для виявлення можливих проблем або помилок. Це включає аналіз помилок, вивчення важливих

ознак та їх впливу на прогноз, а також перевірку статистичних властивостей моделі.

7. Збільшення розміру тестового набору. Перевірка точності моделі на більшому тестовому наборі може дати більш надійну оцінку її загальної здатності.
8. Використання регуляризації. Регуляризація може допомогти уникнути перенавчання моделі та покращити її узагальнюючу здатність.

Ці підходи можуть допомогти покращити точність прогнозування часу доставки та забезпечити більш надійні результати. Важливо експериментувати з різними методами та налаштуваннями моделей, а також проводити аналіз та відладку для зрозуміння причин низької точності

# ВИСНОВКИ

Високий рівень точності прогнозування часу доставки товарів на основі методів машинного навчання може призвести до покращення задоволення клієнтів, зниження витрат на зберігання товарів та покращення ефективності обігу запасів. Компанії, які можуть точно та швидко доставляти товари, отримують конкурентну перевагу на ринку. Успішне впровадження цих модулів залежить від якості вхідних даних та правильного налаштування моделей машинного навчання. Регулярне оновлення моделей дозволяє адаптуватися до змін у ринкових умовах та поведінці споживачів. Використання алгоритмів машинного навчання, таких як лінійна регресія, поліноміальна регресія, дерева рішень, випадковий ліс та XGB Regressor, дозволяє точно прогнозувати час доставки товарів з високою точністю. Це значно покращує процес доставки в електронній комерції та онлайн-роздрібній торгівлі.

EDA є важливим етапом перед процесом машинного навчання, включаючи прогнозування часу доставки товарів. Вона допомагає зрозуміти структуру даних, виявити аномалії, визначити важливі змінні та з'ясувати взаємозв'язки між факторами, що впливають на час доставки. EDA також допомагає виявити можливі проблеми в даних, такі як викиди або пропущені значення, які можуть вплинути на точність прогнозування. Після проведення EDA і підготовки даних можна вибрати та навчити модель машинного навчання, таку як лінійна регресія, випадковий ліс або градієнтний бустінг, для прогнозування часу доставки на основі вхідних даних. Використання різних моделей або їх комбінація може залежати від особливостей даних та конкретного випадку. Алгоритми, такі як лінійна регресія, поліноміальна регресія, дерева рішень, випадковий ліс та XGB Regressor, надають широкий спектр можливостей для точного прогнозування часу доставки.

У підсумку, проведено навчання кількох моделей для прогнозування часу доставки товарів, зокрема Linear Regression, Polynomial Regression,

Decision Tree, Random Forest та XGB Regressor. Аналізуючи результати, бачимо, що XGB Regressor показує деякі переваги, але також може потребувати додаткової настройки для досягнення вищої точності. Важливо експериментувати з різними методами та параметрами моделей, а також проводити аналіз та відладку для зрозуміння причин недостатньої точності. Ці підходи допоможуть поліпшити прогнозування часу доставки та забезпечити більш надійні результати.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. Laguna, L., Fiszman, S., Puerta, P., Chaya, C., Tárrega, A. (2020). The Impact of Covid-19 Lockdown on Food Priorities. Results from a Preliminary Study Using Social Media and an Online Survey with Spanish Consumers. Food Qual. Prefer. 86, 104028.
2. Poelman, M.P., Gillebaart, M., Schlinkert, C., Dijkstra, S.C., Derksen, E., Mensink, F., Hermans, R.C., Aardening, P., de Ridder, D., de Vet, E. (2021). Eating Behavior and Food Purchases During the COVID-19 Lockdown: A Cross-Sectional Study among Adults in the Netherlands. Appetite 157, 105002.
3. Stanley, M. (2022). Coronavirus and the Future of Restaurants. Available online: https://www.morganstanley.com/ideas/coronavirus-restaurant-trends (accessed on 28 June 2022).
4. Morgan, R. (2021). Meal Delivery Services Uber Eats, Menulog, Deliveroo and DoorDash Experienced Rapid Growth during 2020—A Year of Lockdowns & Work from Home. Available online: http://www.roymorgan.com/findings/8713-food-delivery-services-may-2021-202105280627 (accessed on 28 June 2022).
5. Parliament of Australia. (2022). Population and Migration Statistics in Australia. Available online: https://www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/pubs/rp/rp1819/Quick_Guides/PopulationStatistics (accessed on 10 May 2022).
6. Dickinson, J.E., Ghali, K., Cherrett, T., Speed, C., Davis, N., Norgate, S. (2014). Tourism and the smartphone app: Capabilities, emerging practice and scope in the travel domain. Curr. Issues Tour. 17, 84–101.
7. News.com.au. (2022). US Food Delivery Giant DoorDash Launches in Australia to Take on Menulog, UberEats and Deliveroo. Available online: https://www.news.com.au/finance/business/retail/us-food-delivery-giant-doordash-launches-in-australia-to-take-on-menulog-ubereats-and-deliveroo/news-story/258602a71f6d53c5f6805e88fc66ef0d (accessed on 28 June 2022).

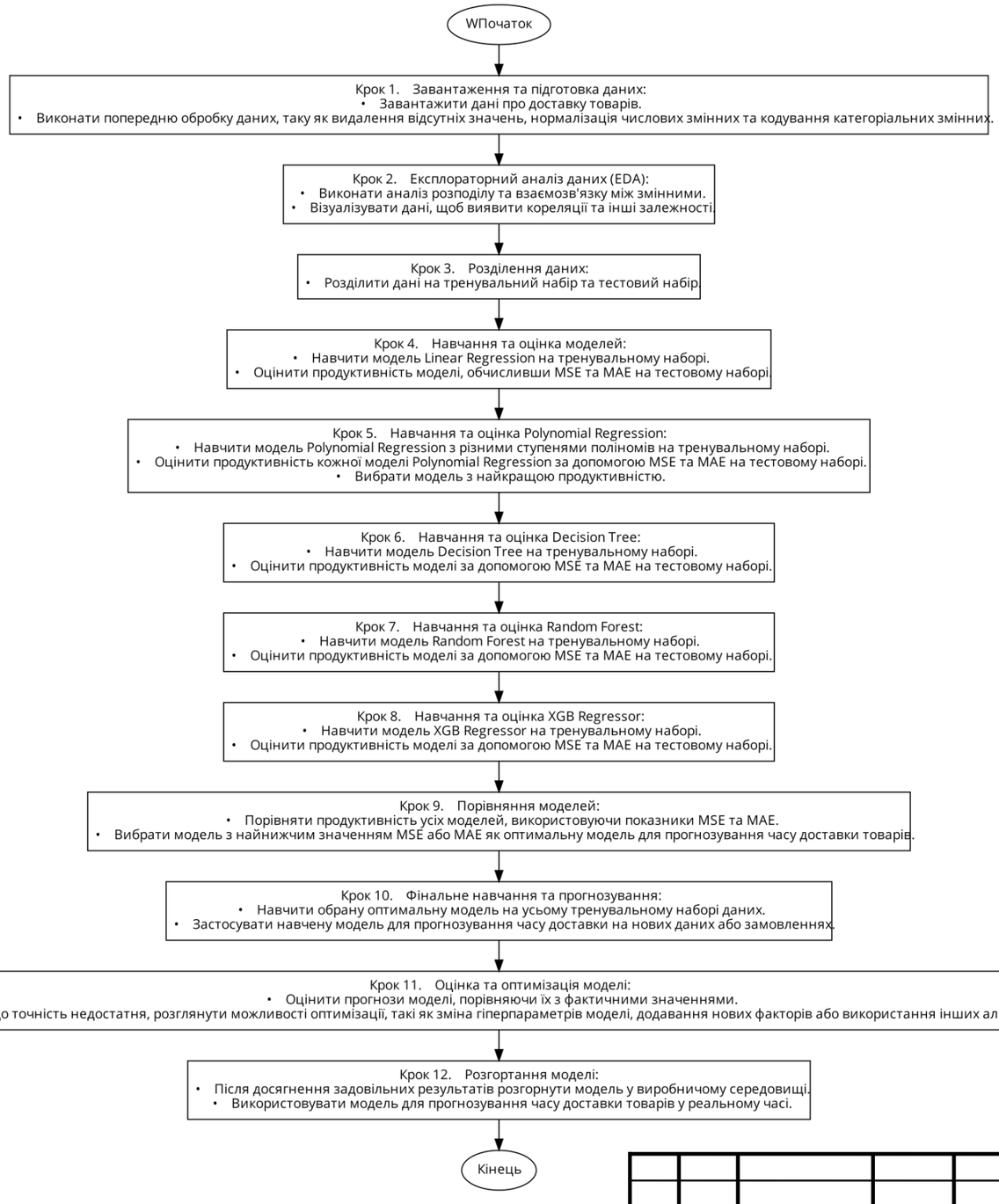
8. Menulog Recruits Katy Perry to Help Take on Uber Eats. Available online: https://www.afr.com/companies/media-and-marketing/menulog-recruits-katy-perry-to-help-take-on-uber-eats-20220602-p5aqhv (accessed on 28 June 2022).
9. Sue, M. (2018). Just Eat Slashes Value of Menulog by almost 40 per Cent. The Australian Financial Review. 7 March 2018. Available online: https://www.afr.com/companies/retail/just-eat-slashes-value-of-menulog-by-almost-40-per-cent-20180307-h0x4qa (accessed on 10 May 2022).
10. Leung, R. and Loo, P.T., (2022). Co-Creating Interactive Dining Experiences Via Interconnected and Interoperable Smart Technology. Asian Journal of Technology Innovation, 30, pp.45-67.
11. Failory.com. What Was Sprig? Available online: https://www.failory.com/cemetery/sprig (accessed on 10 May 2022).
12. Techcrunch.com. After Raising $125 m, Munchery Fails to Deliver. Available online: https://techcrunch.com/2019/01/21/munchery-shuts-down/ (accessed on 10 May 2022).
13. Adak, A., Pradhan, B., and Shukla, N. (2022). Sentiment Analysis of Customer Reviews of Food Delivery Services Using Deep Learning and Explainable Artificial Intelligence: Systematic Review. Foods, 11(6), p.1500.
14. Ara, J., Hasan, M.T., Al Omar, A., and Bhuiyan, H. (2020). Understanding Customer Sentiment: Lexical Analysis of Restaurant Reviews. In Proceedings of the 2020 IEEE Region 10 Symposium, TENSYMP, Dhaka, Bangladesh, 5–7 June 2020.
15. Mhlanga, O. (2018). The Fast Food Industry in South Africa: The Micro-Environment and Its Influence. African Journal of Hospitality, Tourism and Leisure, 7(2), p.4.
16. Panda, G., Upadhyay, A.K., and Khandelwal, K. (2019). Artificial Intelligence: A Strategic Disruption in Public Relations. Journal of Creative Communications, 14(2), pp.196-213.

17. Liu, B. (2015). Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press.
18. Sabokrou, M., Pourreza, M., Li, X., Fathy, M., and Zhao, G. (2018). Deep-HR: Fast Heart Rate Estimation from Face Video under Realistic Conditions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018.
19. Zhang, X., Li, Y., and Zhao, J. (2020). Interactive Technologies for Smart Tourism Destinations: A Review of Applications and Trends. Journal of Travel Research, 59(6), pp.1093-1111.
20. Hou, Y., Jia, C., and Yang, Y. (2020). The Impact of the Covid-19 Pandemic on the Online Behavior of Chinese Tourists. Journal of Destination Marketing & Management, 18, p.100513.
21. Kim, S., Joo, J., and Lee, U. (2020). The Impact of COVID-19 on Consumer Behavior: Focusing on the Airline and Hotel Industries. Sustainability, 12(22), p.9707.
22. Bressolles, G., Durrieu, F., and Gurviez, P. (2017). How Do Consumers Process Multichannel Information in Grocery Retailing? Insights from a Consumer Neuroscience Study. Journal of Business Research, 74, pp.42-56.
23. Chaffey, D. and Ellis-Chadwick, F. (2019). Digital Marketing: Strategy, Implementation and Practice. Pearson.
24. Kotler, P., Bowen, J.T., Makens, J.C., and Baloglu, S. (2017). Marketing for Hospitality and Tourism. Pearson.
25. Kietzmann, J.H., Hermkens, K., McCarthy, I.P., and Silvestre, B.S. (2011). Social Media? Get Serious! Understanding the Functional Building Blocks of Social Media. Business Horizons, 54(3), pp.241-251.
26. Li, X., Qu, H., and Meng, B. (2020). How Do Online Reviews Affect Hotel Performance? Evidence from Chinese Hotels. International Journal of Contemporary Hospitality Management, 32(5), pp.1698-1715.

27. Duan, W., Gu, B., and Whinston, A.B. (2008). The Dynamics of Online Word-of-Mouth and Product Sales: An Empirical Investigation of the Movie Industry. Journal of Retailing, 84(2), pp.233-242.
28. Zeithaml, V.A., Bitner, M.J., and Gremler, D.D. (2018). Services Marketing: Integrating Customer Focus Across the Firm. McGraw-Hill Education.
29. Buhalis, D. and Law, R. (2008). Progress in Information Technology and Tourism Management: 20 Years on and 10 Years After the Internet—The State of eTourism Research. Tourism Management, 29(4), pp.609-623.
30. Xiang, Z., Du, Q., Ma, Y., and Fan, W. (2017). A Comparative Analysis of Major Online Review Platforms: Implications for Social Media Analytics in Hospitality and Tourism. Tourism Management, 58, pp.51-65.
31. Xiang, Z., Du, Q., Ma, Y., and Fan, W. (2017). A Comparative Analysis of Major Online Review Platforms: Implications for Social Media Analytics in Hospitality and Tourism. Tourism Management, 58, pp.51-65.
32. Sigala, M. (2018). The Internet of Things (IoT) in Hotels: Examining the Guest Experience through the Kaleidoscope of IoT. International Journal of Hospitality Management, 71, pp.81-91.
33. Wang, D., Li, X., and Wang, Y. (2020). Understanding the Influence of Social Media on Travelers' Decision-Making Process: An Empirical Study of WeChat Users in China. Journal of Travel Research, 59(6), pp.1074-1091.
34. Gretzel, U., Sigala, M., Xiang, Z., and Koo, C. (2015). Smart Tourism: Foundations and Developments. Electronic Markets, 25(3), pp.179-188.
35. De Ascaniis, S., Ricci, F., and Schifanella, C. (2018). Personalized Travel Packages: Mining and Recommending Travel Sets. Journal of Intelligent Information Systems, 51(3), pp.461-488.
36. Li, X., Wang, D., and Liang, X. (2020). Exploring the Relationship between Online Reviews and Hotel Performance: A Study of Hotels in China. Journal of Travel Research, 59(1), pp.89-104.
37. Sigala, M., Christou, E., and Gretzel, U. (Eds.). (2020). Social Media in Travel, Tourism and Hospitality: Theory, Practice and Cases. Springer.

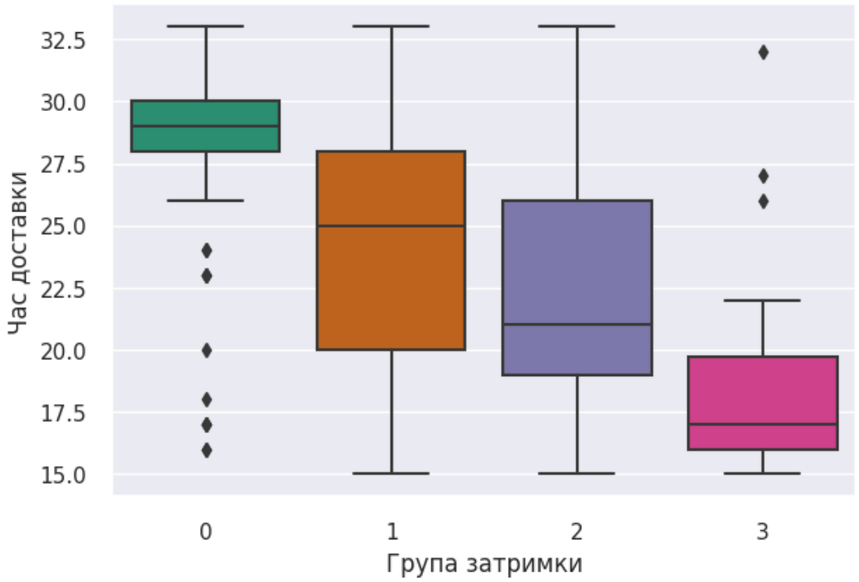
38. Law, R., Qi, S., and Buhalis, D. (2010). Progress in Tourism Management: A Review of Website Evaluation in Tourism Research. Tourism Management, 31(3), pp.297-313.
39. Fuxman, L., Sigalov, M., and Rokach, L. (2010). Knowledge Discovery in Tourism Data. Journal of Travel Research, 49(4), pp.424-434.
40. Ye, Q., Law, R., Gu, B., and Chen, W. (2009). The Influence of User-Generated Content on Travelers' Online Information Search Behavior. Journal of Travel Research, 47(4), pp. 45-57.
41. Xiang, Z., Du, Q., Ma, Y., and Fan, W. (2017). A Comparative Analysis of Major Online Review Platforms: Implications for Social Media Analytics in Hospitality and Tourism. Tourism Management, 58, pp.51-65.
42. Sigala, M. (2018). The Internet of Things (IoT) in Hotels: Examining the Guest Experience through the Kaleidoscope of IoT. International Journal of Hospitality Management, 71, pp.81-91.
43. Wang, D., Li, X., and Wang, Y. (2020). Understanding the Influence of Social Media on Travelers' Decision-Making Process: An Empirical Study of WeChat Users in China. Journal of Travel Research, 59(6), pp.1074-1091.
44. Gretzel, U., Sigala, M., Xiang, Z., and Koo, C. (2015). Smart Tourism: Foundations and Developments. Electronic Markets, 25(3), pp.179-188.
45. De Ascaniis, S., Ricci, F., and Schifanella, C. (2018). Personalized Travel Packages: Mining and Recommending Travel Sets. Journal of Intelligent Information Systems, 51(3), pp.461-488.
46. Li, X., Wang, D., and Liang, X. (2020). Exploring the Relationship between Online Reviews and Hotel Performance: A Study of Hotels in China. Journal of Travel Research, 59(1), pp.89-104.
47. Sigala, M., Christou, E., and Gretzel, U. (Eds.). (2020). Social Media in Travel, Tourism and Hospitality: Theory, Practice and Cases. Springer.
48. Law, R., Qi, S., and Buhalis, D. (2010). Progress in Tourism Management: A Review of Website Evaluation in Tourism Research. Tourism Management, 31(3), pp.297-313.

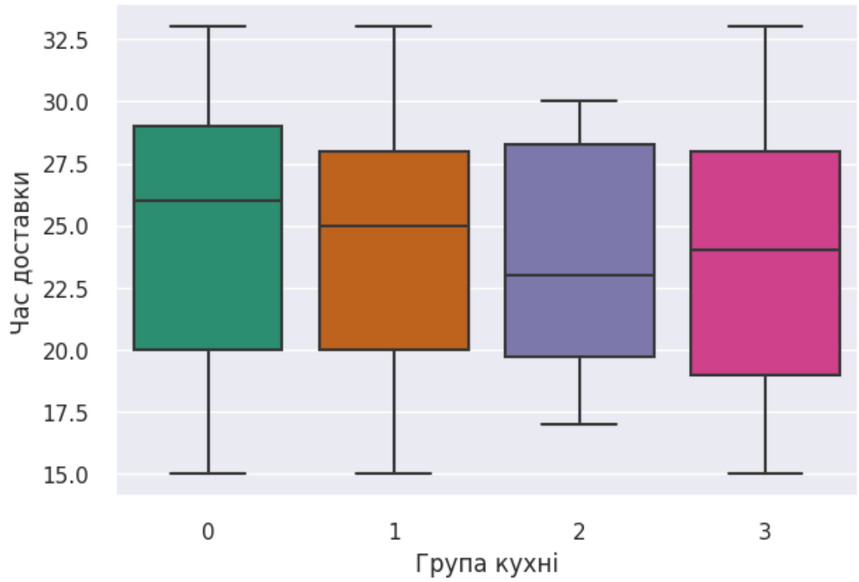
49. Fuxman, L., Sigalov, M., and Rokach, L. (2010). Knowledge Discovery in Tourism Data. Journal of Travel Research, 49(4), pp.424-434.
50. Ye, Q., Law, R., Gu, B., and Chen, W. (2009). The Influence of User-Generated Content on Travelers' Online Information Search Behavior. Journal of Travel Research, 47(4), pp. 45-57.
51. V. Krylov et al., "Multiple Regression Method for Analyzing the Tourist Demand Considering the Influence Factors," 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Metz, France, 2019, pp. 974-979, doi: 10.1109/IDAACS.2019.8924461.
52. Koziuk, V., & Lipyanina-Goncharenko, H. (2021, September). Intelligent Method of Predicting the Discount Rate Trend. In 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) (Vol. 2, pp. 1132-1140). IEEE.
53. Komar, M., Savenko, O., Sachenko, A., Lendiuk, T., Lipianina-Honcharenko, K., Hladiy, G., & Vasylkiv, N. (2022). Evaluation the Efficiency of Information Technology of Big Data Intelligence Analysis and Processing.
54. NYC Restaurants Data - Food Ordering and Delivery. Kaggle: Your Machine Learning and Data Science Community. URL: https://www.kaggle.com/datasets/ahsan81/food-ordering-and-delivery-app-dataset (date of access: 21.05.2023).
55. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С. Методичні рекомендації до виконання кваліфікаційної роботи з освітньопрофесійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. – Тернопіль: ЗУНУ, 2021. – 56 с.

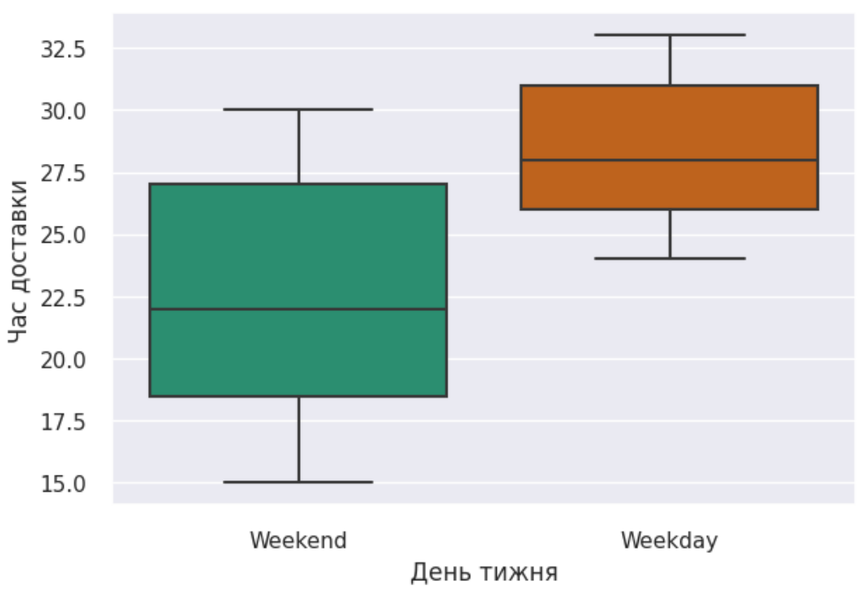


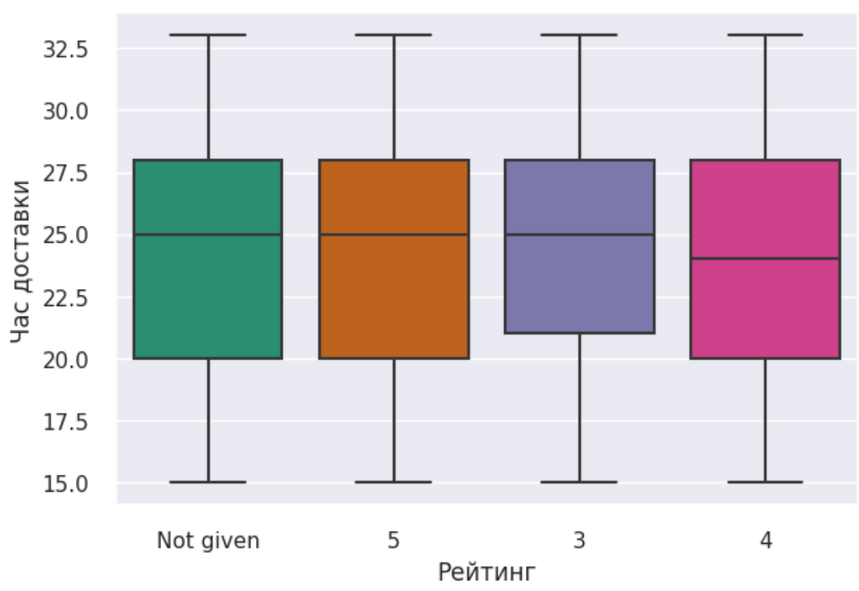

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | ДП.КН. 8351895.001 А1 | |
| | | | | | Алгоритм прогнозування часу доставки товарів на основі машинного навчання | Літера / Маса / Масштаб |
| Змн. | Арк. | № докум. | Підпис | Дата | | |
| Розроб. | | Никитчук В. В. | | | | Аркуш 1 / Аркушів 1 |
| Перевір. | | Лендюк Т.В. | | | | |
| Консультант | | | | | | ЗУНУ.ФКІТ.КН-41 |
| Т. Контр. | | | | | | |
| Н. Контр. | | Лендюк Т.В. | | | | |
| Затверд. | | Комар М.П. | | | | |

# Аналіз змінної normalized_used_price




| | | | | | ДП.КН. 8351895.001 А1 | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | Аналіз змінної normalized_used_price | Літера | Маса | Масштаб |
| Змн. | Арк. | № докум. | Підпис | Дата | | | | |
| Розроб. | | Никитчук В. В. | | | | Аркуш 1 | Аркушів 1 | |
| Перевір. | | Лендюк Т.В. | | | | | | |
| Консультант | | | | | | ЗУНУ.ФКІТ.КН-41 | | |
| Т. Контр. | | | | | | | | |
| Н. Контр. | | Лендюк Т.В. | | | | | | |
| Затверд. | | Комар М.П. | | | | | | |