**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Сумський державний університет**
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»
В.о. завідувача кафедри
Ігор ШЕЛЕХОВ
(підпис)
18 грудня 2023 р.

# КВАЛІФІКАЦІЙНА РОБОТА
**на здобуття освітнього ступеня магістр**

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Інформаційна технологія прогностичного моделювання надання послуг мобільного зв'язку»
здобувача групи ІН.м - 23 Стакана Марка Андрійовича

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.
Марк СТАКАН
(підпис)

Керівник,
старший викладач,
канд. фіз.-мат. наук Оксана ШОВКОПЛЯС
(підпис)

**Суми – 2023**

**Сумський державний університет**

Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

_____ Ігор ШЕЛЕХОВ

(підпис)

# ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

**на здобуття освітнього ступеня магістра**

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»

здобувача групи ІН.м-23 Стакана Марка Андрійович

1. Тема роботи: «Інформаційна технологія прогностичного моделювання надання послуг мобільного зв'язку»

затверджую наказом по СумДУ від *«06» грудня 2023 року № 1412-VI*

2. Термін здачі здобувачем кваліфікаційної роботи *до 18 грудня 2023 року*

3. Вихідні дані до кваліфікаційної роботи_____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) *1) Аналіз проблеми предметної області, постановка й формування завдань дослідження. 2) Огляд технологій, що використовуються для прогнозування відключень від мобільних операторів. 3) Розроблення інформаційної системи прогностичного моделювання надання послуг мобільного зв'язку. 4) Аналіз результатів. 5) Оформлення пояснювальної записки до кваліфікаційної роботи.*

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти до проєкту (роботи), із зазначенням розділів проєкту, що стосується їх

| Розділ | Консультант | Підпис, дата | |
|---|---|---|---|
| | | Завдання видав | Завдання прийняв |
| | | | |

7. Дата видачі завдання «____» _____ 20 ___ р.

Завдання прийняв до виконання _____ Керівник _____

(підпис) (підпис)

# КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назва етапів кваліфікаційної роботи | Термін виконання | Примітка |
|---|---|---|---|
| *1* | *Аналіз проблеми предметної області, постановка та формування завдань дослідження* | 07.11-13.11.23 | |
| *2* | *Огляд технологій, що використовуються для проєктування технології прогнозування утримання клієнтів операторів мобільного зв'язку* | 14.11-20.11.23 | |
| *3* | *Розроблення системи прогностичного моделювання надання послуг мобільного зв'язку* | 20.11-04.12.23 | |
| *4* | *Аналіз отриманих результатів* | 04.11-06.12.23 | |
| *5* | *Оформлення пояснювальної записки до кваліфікаційної роботи* | 07.11-17.12.23 | |

Здобувач вищої освіти _____ Керівник _____

(підпис) (підпис)

# АНОТАЦІЯ

**Записка**: 63 стор., 23 рис., 3 таб., 1 додаток, 27 використаних джерел.

**Обґрунтування актуальності теми роботи -** В центрі уваги даного дослідження знаходиться актуальна проблема відключення від послуг мобільних операторів, що в сучасних умовах є ключовою для їхнього ефективного функціонування. Стійка конкуренція та зміни у споживчих уподобаннях вимагають від операторів вдосконалювати стратегії для утримання і залучення клієнтів, а інтелектуальні інформаційні технології стають важливим інструментом в цьому контексті. Результати цього дослідження не лише розкривають інтелектуальний підхід до вирішення проблеми відключення від послуг мобільних операторів, але й надають практичні рекомендації для застосування розроблених технологій в бізнес-сфері.

**Об’єкт дослідження –** процес проєктування системи виявлення потенційно-схильних до відмови від послуг абонентів

**Мета роботи** – створення інформаційної технології проектування системи автоматизації виявлення клієнтів-відмовників та запобігання їх міграції до конкурентів.

**Методи дослідження** – аналіз сфери бізнесу телекомунікаційних послуг як такого, економічне обґрунтування виявлення незадоволених клієнтів, порівняння математичних методів, які можуть бути використані для автоматизації процесу виявлення потенційно-схильних до відмови від послуг абонентів.

**Результати** – створено інформаційну технологію проектування системи виявлення потенційно-схильних до відмови від послуг абонентів. Виконана робота включала пошук необхідної інформації відповідно до теми роботи, використання та аналіз методів які використовуються у веб-сервісах по підписці, проведення аналізу математичних моделей, обрані інфраструктурні програмні рішення, плагіни, та інтерфейси програмування для написання додатку, використання новітніх технологій Back-End розробки, такі як Python, Bash, Sklearn, Flask, та Rest API розгортання додатку та його тестування.

CHURN DETECTION,  DATA MINING, DATA SCIENCE, SKLEARN, FLASK, EXPLORATORY DAT ANALYSIS, PYTHON, FLASK, REST API

# Зміст

# ВСТУП

**Актуальність**. Галузь телекомунікацій стала ключовим сегментом у економіках розвинутих країн. Кожен рік, телекомунікаційні провайдери стикаються з фінансовими втратами через те, що втрачають частину своїх абонентів. Середній рівень відтоку клієнтів у цій галузі досягає 25% річно, що є значним показником.

**Об'єкт дослідження** – процес проєктування системи виявлення потенційно-схильних до відмови від послуг абонентів.

**Предмет дослідження** – стратегії залучення та утримання клієнтів у галузі телекомунікацій.

**Гіпотеза**. Подовження терміну обслуговування клієнтів є найбільш вигідною стратегією для телекомунікаційних компаній, оскільки вартість утримання існуючого клієнта значно нижча, ніж залучення нового.

**Новизна**. Робота вивчає проблематику відтоку клієнтів у телекомунікаційній галузі, вносячи вклад у розвиток стратегій утримання та залучення клієнтів у висококонкурентному середовищі. Застосування сучасних методів машинного навчання та аналізу даних для прогнозування відтоку клієнтів у телекомунікаційній галузі дозволяє підвищити точність прогнозів та оптимізувати витрати на утримання клієнтської бази.

**Апробація матеріалів роботи**. Основні результати роботи оприлюднені та оговорені на Міжнародній науково-технічній конференції студентів та молодих вчених «Інформатика, математика, автоматика» (ІМА – 2020), (Суми, 20–24 квітня 2020 р.).

**Структура**. Дана кваліфікаційна робота магістра складається зі вступу, інформаційно-аналітичного огляду предметної області щодо актуальності проблеми та вивчення стратегій для утримання і залучення клієнтів; аналітичної частини щодо створення функціональних вимог, огляду інструментів для

розробки та бази даних; практичної реалізації інформаційної технології; висновків; списку використаних джерел та додатку.

**Зв'язок роботи з науковою темою**. Кваліфікаційна робота виконана на кафедрі комп'ютерних наук та пов'язана з виконанням науково-дослідної роботи № 0118U006971 «Методи, математичні моделі та інформаційні технології аналізу і синтезу інфокомунікаційних систем» (2018-2023).

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Насамперед, відтік клієнтів може здатися не надто тривожним явищем, однак важливо розглядати, чи достатньо ефективні наявні інструменти управління, і чи не є дані, які використовуються, занадто складними. Саме тут на допомогу приходять сучасні методи аналізу даних, які можуть надати відповіді на ці запитання.

Традиційно, більшість клієнтів користується декількома послугами одного і того ж провайдера, такими як мобільний зв'язок, інтернет та кабельне телебачення. Якщо клієнт припиняє користуватися однією з послуг, це часто призводить до відмови від інших послуг цього ж провайдера. Це явище підкреслює важливість інтегрованого підходу до управління відносинами з клієнтами.

Вплив соціальних мереж у цьому контексті не можна недооцінювати. Позитивні та негативні відгуки розповсюджуються у соціальних медіа неймовірно швидко. Коли клієнт ділиться своїми негативними враженнями в Instagram, Twitter, Facebook або інших популярних мережах, це може вплинути на сприйняття та рішення тисяч інших потенційних користувачів [1-2].

Також, коли один клієнт переходить до іншого провайдера, це може стати причиною масового відходу клієнтів, які мають схожі проблеми та вирішують слідувати його прикладу. Інформація, яку розповідають клієнти про свої причини переходу до іншого оператора, має величезне значення, адже вона може виявити ключові аспекти невдоволення клієнтів.

Хоча відмітимо, що відтік клієнтів може бути не тільки викликом, але й цінним джерелом інформації для компанії. Коли клієнт припиняє співпрацю, це може бути чудовою можливістю для компанії дізнатися про слабкі сторони своїх послуг або про потенційні загрози від конкурентів. Таким чином, важливо не просто фіксувати факт відходу клієнта, але й аналізувати його причини, навіть якщо вони здаються незначними або нелогічними, оскільки вони можуть вказувати на

фундаментальні проблеми у бізнесі [2].

У сучасному світі телекомунікацій, особливо в сфері мобільного зв’язку, компанії вступають у жорстку маркетингову конкуренцію за залучення та утримання кожного клієнта. У цій боротьбі перевагу здобуває той, хто має більш розвинені здібності до аналізу даних. Ключ до успіху лежить у володінні інформацією про клієнтів, які вже припинили користуватися послугами компанії, та у використанні цих даних для попередження майбутнього відтоку.

Лідери ринку розуміють, що для ефективного попередження відтоку клієнтів необхідно не лише аналізувати й візуалізувати дані, але й залучати IT-експертів для розробки відповідних стратегій [1]. Компанії з найсучаснішими інструментами аналізу даних, здатні швидко виявляти сигнали про неминучий відтік та оперативно реагувати на них, що дозволяє мотивувати клієнтів залишатися. Крім того, виявлення та усунення проблем у мережі, які можуть стати причиною відходу клієнтів, є важливим елементом стратегії запобігання відтоку.

Предиктивна аналітика відіграє ключову роль у розкритті прихованих факторів, що спричиняють втрату клієнтів. Навіть компанії з детальною базою даних можуть не здатні отримати повну картину без ефективних прогнозуючих інструментів [3]. Ця технологія виявляє потенційно незадоволених клієнтів, які можуть здаватися випадковими, і допомагає вирішити їх проблеми до того, як вони вирішать змінити оператора [4].

Важливо також враховувати вплив цифрових медіа та соціальних мереж. Сучасний клієнт надзвичайно з'єднаний і впливовий, і його думка може швидко поширюватися через соціальні мережі, впливаючи на сприйняття бренду серед широкого кола потенційних клієнтів. Тому аналіз відгуків і настроїв у соціальних медіа повинен стати неодмінною частиною стратегії компанії, спрямованої на збереження клієнтської бази.

## 1.1 Аналіз актуальності проблеми

Розглядаючи телекомунікаційний бізнес, стає зрозуміло, що стратегічне управління клієнтською базою є життєво важливим. Воно включає в себе не лише залучення нових клієнтів, але й збереження вже існуючих, що є особливо важливим у сучасному конкурентному ринку [7]. Зростання клієнтської бази відбувається завдяки постійному потоку нових абонентів, які приваблюються за допомогою рекламних стратегій, спеціальних пропозицій та рекомендацій від інших клієнтів. Клієнти перебувають у динамічній взаємодії з послугами, активно їх використовуючи, але іноді вирішують припинити користування з різних причин. Цей процес, відомий як «Життєвий цикл клієнта», охоплює весь шлях від відкриття до відмови від послуг, що являє собою важливий елемент аналізу для бізнесу [3].

Кожен споживач в сфері телекомунікацій є унікальним, зі своїми персональними вподобаннями та потребами. Вибір послуг та тарифів залежить від різноманітних чинників. Наприклад, для деяких важливими є тарифи для міжнародних подорожей, тоді як інші можуть шукати оптимальні умови для внутрішніх дзвінків чи великих сімейних пакетів [3]. Постійні зміни в тарифах та послугах дають можливість адаптуватися до потреб кожного клієнта, що є ключовим у підтримці задоволеності споживачів.

Місцеположення клієнта відіграє значну роль у його виборі телекомунікаційного провайдера. Якість послуг може суттєво варіюватися в залежності від регіону, що робить важливим знайти провайдера, який надає високоякісний сервіс у відповідній місцевості. На додаток до якості послуг, фізична доступність сервісних центрів також є важливою, оскільки певна частина клієнтів віддає перевагу особистому обслуговуванню, ніж дистанційному.

Відмова від послуг може бути обумовлена різними чинниками: переїздом в інше місце, зміною умов роботи, які вимагають іншого тарифу, або незадоволенням сервісом або обслуговуванням. Це не лише проблема телекомунікаційної сфери, а й багатьох інших галузей, де існує клієнтська база та відбуваються регулярні

транзакції.

Розвиток бізнесу в сучасному світі вимагає не тільки високотехнологічних інновацій, але й глибокого розуміння потреб клієнтів. Життєвий цикл клієнта у цій галузі стає дедалі складнішим і багатограннішим, особливо з урахуванням різноманітності споживчих звичок та вимог. Важливо, щоб компанії не лише реагували на поточні потреби своїх абонентів, але й антиципували майбутні тенденції та зміни у споживчому поведінці. Це може включати в себе адаптацію до зростаючого попиту на мобільні дані, розробку більш гнучких та персоналізованих тарифних планів, а також забезпечення високої якості обслуговування та підтримки клієнтів [5, 6].

Ще одним ключовим аспектом є інтеграція цифрових технологій у взаємодію з клієнтами. Оскільки все більше людей використовують цифрові канали для вирішення своїх повсякденних потреб, компанії повинні використовувати ці технології для поліпшення досвіду клієнтів. Наприклад, використання штучного інтелекту для персоналізації пропозицій та чат-ботів для швидшої та ефективнішої відповіді на запити клієнтів може значно покращити їх задоволеність [8].

Враховуючи велику конкуренцію у сфері телекомунікацій, компаніям необхідно неперервно інновувати та пропонувати виняткову цінність, щоб зберегти своїх клієнтів та привабити нових. Це означає постійне вдосконалення продуктів та послуг, а також створення унікального та запам’ятовуваного досвіду користування. У такому динамічному та швидкозмінному середовищі, здатність адаптуватися та розвиватися є ключовою для успіху та зростання будь-якого телекомунікаційного бізнесу.

## 1.2 Дослідження аналогів

Сучасний світ телекомунікацій вирує викликами, одним з яких є прогнозування відтоку абонентів. Різноманітні методики, більшість з яких засновані на технологіях машинного навчання та аналізі даних, активно використовуються у цій галузі. Часто дослідники фокусуються на одному конкретному методі обробки даних та прогнозувальній моделі, тоді як інші зосереджуються на аналізі та порівнянні декількох стратегій для кращого розуміння відтоку абонентів [10].

Знаковим прикладом є робота Гаврила Тодереана, доктора технічних наук у Румунському технічному університеті Клуж-Напоки. Він розробив удосконалену методологію аналізу даних, яка була спрямована на прогнозування відтоку передплачених абонентів [9]. В основі його підходу лежав аналіз даних 3333 клієнтів із 21 атрибутом [15]. Гаврил використовував метод аналізу основних компонентів (PCA) для оптимізації обсягу даних і різні алгоритми машинного навчання, такі як нейронні мережі, SVM та Байєсівський класифікатор для прогнозування. Цікавим є той факт, що він застосував цю модель для вирішення проблеми відтоку в одній з великих китайських телекомунікаційних компаній, де вона показала високу точність – 91,1%.

Паралельно, Ідріс запропонував інноваційний підхід, заснований на генетичному програмуванні з використанням AdaBoost. Ця модель була протестована на двох відомих наборах даних – Orange Telecom та Cell2cell, демонструючи точність 89% для Cell2cell та 63% для Orange Telecom [11].

Хуан та його колеги дослідили використання великих даних у прогнозуванні відтоку абонентів. Вони сконцентрувалися на перевагах великих даних, враховуючи їхній обсяг, різноманітність і швидкість обробки. У своєму дослідженні вони використовували дані відділу операцій та бізнес-підтримки в одній з найбільших китайських телекомунікаційних компаній, використовуючи платформу Hadoop [8]. Методом вибору став алгоритм випадкових лісів, ефективність якого

була оцінена за допомогою AUC.

Також важливо згадати дослідження, присвячені вивченню неврівноважених наборів даних. Це особливо актуально, коли пропорція клієнтів, які залишають телекомунікаційні послуги, є значно меншою в порівнянні з активними абонентами. Такі задачі потребують особливого підходу, оскільки вони представляють великий виклик для прогнозувальних моделей.

Бурез та Ван ден Поель займалися проблемою неврівноваженості даних, використовуючи алгоритми, такі як Gradient Boosting і Random Forest. Метриками були AUC та Lift-score [12, 18] . Їхній підхід включав методи вибору випадкових вибірок та розширення недостатньої вибірки для покращення прогнозування. Застосування цих методів значно підвищило точність моделей, що демонструє ефективність цього підходу у вирішенні задачі неврівноважених даних.

Прогнозування відтоку абонентів у сфері телекомунікацій вимагає глибокого розуміння поведінки споживачів та вміння правильно інтерпретувати великі обсяги даних. Цей процес не тільки виявляє невдоволених клієнтів, але й надає можливість для вдосконалення послуг та збільшення лояльності клієнтів.

На наш погляд, використання машинного навчання та аналітики даних у цій сфері відкриває нові горизонти. Алгоритми, засновані на машинному навчанні, здатні виявляти складні взаємозв'язки в даних, які можуть бути неочевидними для людського ока. Це особливо важливо в умовах стрімкого розвитку технологій та зміни споживацьких звичок. З іншого боку, критично важливим є розуміння того, що алгоритми та моделі не можуть бути повністю автономними. Вони потребують постійного оновлення та налаштування, враховуючи нові тенденції та зміни на ринку [7]. У підсумку, прогнозування відтоку абонентів – це не лише технічне завдання, але й стратегічний інструмент, який дозволяє компаніям бути більш клієнтоорієнтованими та конкурентоспроможними у швидкозмінному цифровому світі.

Також варто відзначити, що успіх моделі прогнозування відтоку абонентів не

залежить лише від точності прогнозу, але й від ефективності впровадження змін на основі отриманих даних. Це вимагає гнучкості бізнес-процесів та відкритості до інновацій. Крім того, важливо пам'ятати про етичні аспекти аналізу даних. Збір та аналіз інформації про клієнтів повинен відбуватися з повагою до їхньої конфіденційності та приватності.

### 1.3 Постановка задачі

Метою проєкту є створення прототипу інтелектуальної системи, яка здатна прогнозувати відмову від послуг абонентів мобільних операторів на основі аналізу історичних даних.

Щоб досягти цієї мети, необхідно реалізувати такі задачі:

1) Дослідити предметну область та проблеми:

– ретельно вивчити функціонування мобільних операторів та їх бізнес-моделей;

– проаналізувати історичні дані, ідентифікувати чинники, що можуть призвести до відмови від послуг;

– визначити степінь актуальності проблеми для бізнесу.

2) Проаналізувати аналоги:

– провести пошук та оцінити існуючі рішення для прогнозування відмови від послуг;

– вивчити їх переваги та недоліки;

– створити інформаційну технологію на основі оптимального аналога або комбінації рішень для використання в нашому проєкті;

3) Розробити модель машинного навчання:

– виконати ітеративне вдосконалення параметрів та оптимізацію алгоритмів для підвищення точності прогнозів;

– узагальнити результати та прийняти остаточні рішення щодо конфігурації системи.

3) Підготувати та очистити історичні дані:

– виділити ключові параметри та фактори, які впливають на відмову від послуг;

– реалізувати та навчити модель, можливо, з використанням бібліотеки scikit-learn або TensorFlow;

– реалізувати прототип REST API.

4) Розробити мікросервіс для інтеграції розробленої моделі:

– визначити структуру REST API для зручної взаємодії з системою;

– протестувати прототип, включаючи валідацію та навчання моделі в реальному часі;

– оцінити ефективність моделі та сервісу.

Цей план дозволить систематично вирішити завдання проєкту, забезпечуючи створення працездатного прототипу для прогнозування відмови від послуг мобільних операторів, який можна адаптувати до реальних бізнес-проблем.

# 2 ПРОЄКТУВАННЯ ТА ВИБІР МЕТОДІВ РЕАЛІЗАЦІ

Для створення прототипу розумної системи, яка передбачає відтік абонентів, були використані анонімізовані табличні дані, доступні від IBM у відкритому доступі в Інтернеті. Ці дані представлені у форматі .csv і включають репрезентативну вибірку із 7063 абонентів, кожен з яких характеризується 21 атрибутом. Основні атрибути включають:

1) Перелік послуг, на які підписані клієнти. Це включає в себе телефонні послуги, кількість ліній, доступ до Інтернету, інтернет-безпеку, резервне копіювання в Інтернеті, захист обладнання, технічну підтримку та підписку на телебачення та стрімінг фільмів.
2) Інформація про обліковий запис кожного клієнта. Тут містяться дані про тривалість використання послуг, тип контракту, спосіб оплати, наявність безпаперового рахунку, щомісячні платежі та загальна сума платежів.
3) Демографічні дані абонентів, включаючи стать, вік, наявність партнера або утриманців.

Детальні назви та описи використаних атрибутів, а також типи даних, наведено в таблиці 2.1.

Крім того, для кожного з абонентів визначено ключову бінарну змінну, яка відображає їхнє рішення залишити або продовжувати користуватися послугами провайдера. Ця змінна має лише два можливі значення: "1" позначає відмову абонента від послуг провайдера протягом місяця після збору даних, тоді як "0" означає, що абонент залишився з провайдером. Цей атрибут є основним для прогнозування.

Нашою головною метою є розробка та впровадження системи-класифікатора, здатної прогнозувати відтік абонентів. Система буде аналізувати дані 7063 абонентів і 21 різних атрибутів, використовуючи при цьому методи статистичного аналізу та машинного навчання. Для оцінки ефективності моделі буде використано

ROC-score, метрику, яка вимірює здатність системи точно розрізняти між абонентами, які залишають телекомунікаційну компанію, та тими, хто продовжує користуватися її послугами.

Таблиця 2.1 – Опис використаних даних

| № | Назва атрибуту | Опис |
|---|---|---|
| 1 | CustomerId | Уникальный анонімізований ідентифікатор абонента |
| 2 | Gender | Замовник чоловік або жінка |
| 3 | Senior Citizenship | Замовник є похилим громадянином чи ні |
| 4 | Partner | Замовника є партнер чи ні |
| 5 | Dependents | Чи є у замовника утриманці чи ні |
| 6 | Tenure | Кількість місяців перебування замовника з провайдером |
| 7 | PhoneService | Чи має клієнт послугу сервісного обслуговування телефону чи ні |
| 8 | Multiple Lines | Чи є у замовника кілька ліній чи ні |
| 9 | Internet Service | Інтернет-провайдер клієнта - (DSL, оптоволоконна, немає) |
| 10 | OnlineSecurity | Чи має клієнт послугу онлайн- безпеки в Інтернеті чи ні |
| 11 | Online Backup | Чи має клієнт резервне копіювання в Інтернеті чи ні |
| 12 | Device Protection | Чи має клієнт послугу захисту пристрою від фізичних пошкоджень чи ні |
| 13 | TechSupport | Замовник має технічну підтримку чи н |
| 14 | Streaming TV | Чи має клієнт потокове телебачення чи ні |
| 15 | Streaming Movies | Чи дивиться клієнт потокові фільми чи ні |
| 16 | Contract | Тип контракту замовника (місяць на місяць, один рік, два роки) |
| 17 | Paperless Billing | Чи клієнт отримує рахунки без паперовим шляхом на папері чи ні |
| 18 | Payment Method | Спосіб оплати клієнта |
| 19 | Monthly Charges | Сума, що стягується з замовника щомісяця |
| 20 | Total Charges | Загальна сума, яку заплатив замовник за всю |

## 2.1 Мета та задачі

Цей проєкт зосереджений на розробці програмного забезпечення, метою якого є зниження кількості абонентів, що відмовляються від послуг мобільного оператора, сприяючи тим самим зростанню прибутків компанії. Проєкт «Розумні інформаційні технології для прогнозування відмови від послуг мобільних операторів» включає в себе кілька ключових завдань:

1) Автоматизація процедури обробки даних для ефективності та точності.
2) Виправлення помилок і вилучення невідповідних даних (Data cleaning).
3) Розробка нових змінних на базі існуючих даних (Feature engineering).
4) Визначення найважливіших змінних за допомогою технік візуалізації даних (Exploratory data analysis).
5) Попередня обробка змінних для підготовки до аналізу (Preprocessing phase).
6) Етап моделювання та перевірки ефективності моделі (Modeling phase).
7) Розгортання моделі за допомогою веб-додатку з REST-API інтерфейсом.

Ці етапи становлять основу CRISP-DM, методології, яка широко використовується в розробці інформаційних систем на основі машинного навчання.

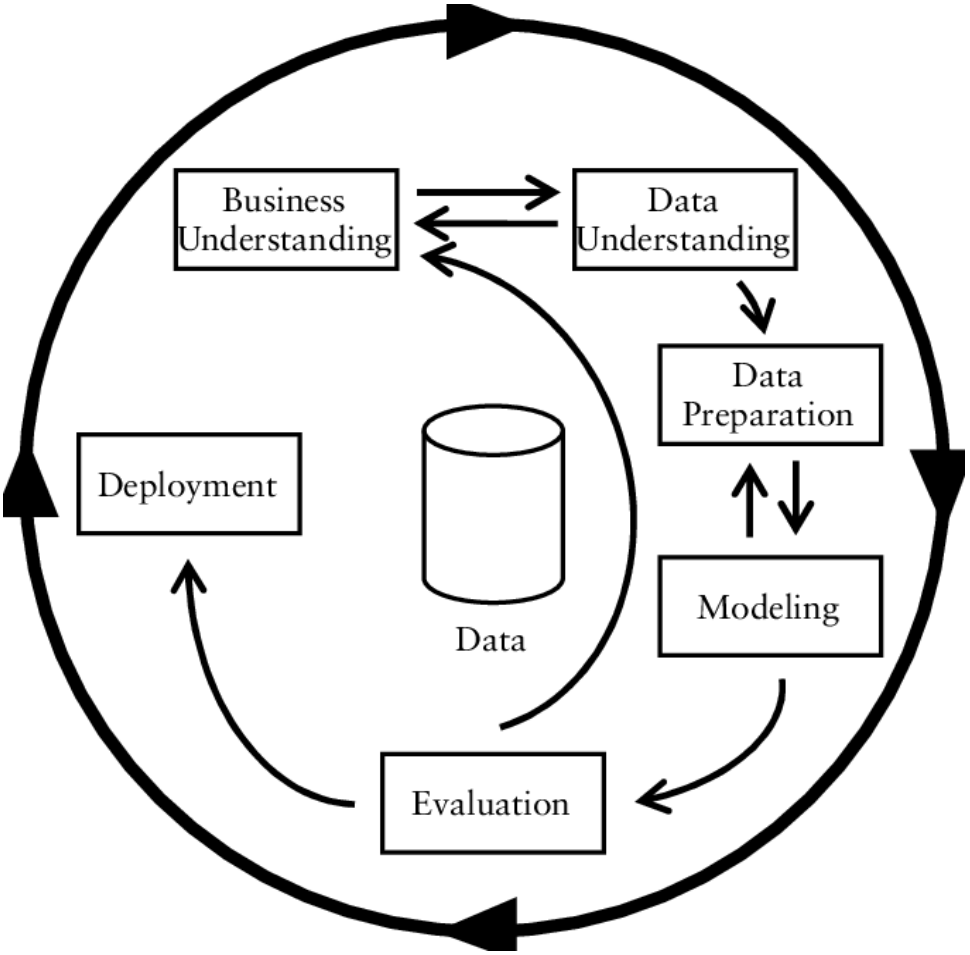



Рисунок 2.1 – Діаграма процесу розробки даних CRISP-DM

CRISP-DM, або Міжгалузевий стандартний процес для дата-майнінгу, є визнаною та широко застосовуваною методологією у сфері аналізу даних. Ця модель, яка довела свою ефективність у промисловому застосуванні, містить шість ключових фаз [19]. Важливо зазначити, що хоча між цими фазами існують основні зв'язки, показані стрілками, послідовність їхнього виконання не є незмінною. В більшості випадків проекти вимагають повернення до раніше пройдених етапів та наступного продовження руху вперед, що робить процес гнучким і адаптивним.

## 2.2 Огляд моделей прогнозування відтоку

Вирішення проблеми прогнозування відтоку абонентів у телекомунікаціях вимагає комплексного підходу, який поєднує первинний аналіз даних та розробку прогнозуючої моделі. Для цього існують два основних методи рішення.

Перший підхід полягає у створенні моделі, яка передбачає, чи відмовиться клієнт від послуг оператора на основі аналізу набору його атрибутів. Цей метод вимагає глибокого дослідження специфіки сфери та розробки моделі, заснованої на отриманих результатах. Проте такий підхід потребує значних зусиль та експертної компетенції, а результати, отримані після дослідження, часто не можуть бути масштабовані для використання з іншими клієнтами.

Другий підхід, який з'явився у 20 столітті, пропонує більш універсальне рішення цієї задачі без необхідності глибокої експертизи у предметній області. Цей підхід базується на наявності великої кількості даних, які дозволяють виявити причинно-наслідкові зв'язки та налаштувати модель для точного прогнозування. З розвитком інтернету компанії почали активно зберігати та акумулювати дані про своїх клієнтів, включаючи поведінкові фактори, що створює величезні бази даних, які можна використовувати для покращення бізнес-ефективності та збільшення доходів.

Ці методи відомі під назвою машинного навчання, яке є частиною науки про штучний інтелект та охоплює аналіз даних. У машинному навчанні «об’єкт»

визначається як сутність або явище, для якого потрібно зробити прогноз. «Цільова змінна» – це те, що потрібно прогнозувати, а «простір відповідей» включає всі можливі результати, які аналізуються [15].

Для комп’ютера об’єкти реального світу не завжди зрозумілі, тому важливо перетворити ці об’єкти на числові дані. «Ознака» або «атрибут» визначається як характеристика об’єкта, а «вектор ознак» – це сукупність усіх ознак об’єкта, з якими можна проводити векторні операції [16]. Ознаки можуть бути представлені як дійсні числа, текстові рядки, порядкові атрибути тощо, але всі вони повинні бути зрозумілими для комп’ютера.

Ключовим елементом у машинному навчанні є навчальна вибірка, яка представляє об'єкти та їх ознаки, на основі яких формується загальна закономірність. У цьому випадку мова йде про навчання з учителем, де навчальна вибірка складається з пар «об’єкт-ознаки» та «відповідь». Важливим є питання збору такої вибірки, але в даному контексті, використовується вже існуючий набір даних.

Наступним кроком є використання моделі машинного навчання, яка функціонує як функція, що відображає простір об'єктів у простір відповідей, приймаючи на вхід об'єкт X. Тренування моделі відбувається на основі навчальної вибірки.

Для вирішення поставленої задачі було обрано підхід, заснований на методах машинного навчання, оскільки це відповідає особливостям задачі та доступності даних. Завдання полягає у визначенні ймовірності віднесення до одного з двох класів, де простір відповідей - це множина всіх дійсних чисел від 0 до 1. Це є задачею бінарної класифікації, де встановлюється залежність цільової змінної від значень ознак.

За останнє десятиліття, задача прогнозування відтоку клієнтів виявилася дуже актуальною у багатьох сферах [17]. Застосування моделей машинного навчання в цій сфері демонструє значний потенціал для ефективного вирішення поставленої

задачі, дозволяючи компаніям оптимізувати свої стратегії збереження клієнтів і, відповідно, збільшувати свої доходи.

У контексті використання машинного навчання для прогнозування відтоку абонентів, важливо також звернути увагу на постійно зростаючий обсяг даних та їх різноманітність. Це створює унікальні можливості для виявлення більш тонких і складних шаблонів у поведінці споживачів. Однак, разом з цим зростає й складність моделей, що може призвести до перенавчання або неправильної інтерпретації даних. Тому важливо збалансувати складність моделі з її інтерпретованістю та здатністю до узагальнення. Також не можна ігнорувати етичні аспекти збору та обробки даних, оскільки це безпосередньо стосується приватності та конфіденційності інформації про клієнтів. У цілому, застосування машинного навчання в цій сфері відкриває нові перспективи для розвитку бізнесу, проте вимагає відповідального та обдуманого підходу.

### 2.2.1 Linear models

Лінійні моделі вважаються одними з найбільш базових у сфері машинного навчання. Суть лінійних алгоритмів криється в виборі таких коефіцієнтів для кожного атрибуту, які дозволять знизити розбіжності між передбачуваними та реальними даними цільового параметра до мінімуму. Це особливо актуально в задачах класифікації. Цікаво, що попри свою простоту, лінійні моделі часто є відправною точкою для більш складних аналітичних проектів, оскільки вони надають чітке розуміння основних тенденцій у даних. У таких випадках математична формула лінійного бінарного класифікатора приймає вигляд:

$$f(X) = \frac{1}{1+e^{-h(X)}}$$

де $h(X)$ – лінійна функція:

$$h(X) = w_0 + \sum_{i=1}^{d} w_i X_i,$$

в якій $w_0$ – нормалізуюча константа, $X_i$ – атрибути, а $w_i$ – їх коефіцієнти або ваги.

Якщо додати $(d+1)$-шу ознаку, яка на кожному об'єкті приймає значення 1, лінійний алгоритм можна буде записати в більш компактній векторній формі:

$$h(x) = \sum_{i=1}^{d+1} w_i X_i = \langle w, X \rangle,$$

де використовується позначення $\langle w, x \rangle$ для скалярного добутку двох векторів.

Похибка лінійного класифікатора (loss) та ваги $w_i$, які мінімізують ту саму похибку loss, обчислюються методом оптимізації градієнтного спуску, де в свою чергу використовується метод maximum likelihood estimation (MLE), який можна представити у вигляді функції (оскільки тепер J залежить від вектора, а не від функції) помилок:

$$J(w, X) = -\frac{1}{n}\sum_{i=1}^{n} y_i \cdot \log(\widehat{y_i}) + (1 - y_i) \cdot \log(1 - \widehat{y_i}) \rightarrow min,$$

де $\widehat{y_i}$ – прогнозована ймовірність для $i$-го об'єкта, та $y_i$ – фактичний клас $i$-го об'єкта.

Втім, у лінійних моделях існує певний недолік стосовно їх застосування. Ці методи не відтворюють складність людського процесу прийняття рішень. Людина, намагаючись розібратися в проблемі, часто вдається до серії простих запитань, які крок за кроком ведуть до вирішення. Цей аспект відсутній у лінійних моделях, що робить їх менш інтуїтивно зрозумілими, порівняно з тим, як люди звикли аналізувати та вирішувати проблеми.

### 2.2.2 Decision tree

Дерева рішень виступають як ключовий інструмент у сфері машинного навчання, представляючи унікальну методологію у цій галузі. Цей інструмент є надзвичайно потужним та широко використовуваним у задачах класифікації та прогнозування. Структура дерева рішень нагадує вигляд звичайного дерева, де кожен внутрішній вузол відображає певний логічний тест на атрибути, гілки – це можливі результати цих тестів, а листкові вузли (кінцеві вузли) представляють класифікацію об’єктів [23].

Процес 'навчання' дерева рішень полягає в рекурсивному розділенні даних на підмножини засноване на атрибутах. Ця процедура триває, поки не буде досягнуто однорідності в усіх підмножинах у вузлі, або додаткове розділення не покращує прогностичні здібності. Особливістю дерев рішень є те, що вони не вимагають специфічних знань про домен чи налаштування параметрів, що робить їх ідеальними для початкового аналізу. Вони ефективно працюють з даними, що мають великі розміри та часто демонструють високу точність у класифікації. Індукція дерева рішень є типовим прикладом індуктивного підходу в навчанні для визначення класифікаційних закономірностей.

Корисним аспектом у використанні дерев рішень є їх здатність візуалізувати рішення, що робить їх не тільки потужними аналітичними інструментами, але й доступними для неспеціалістів для розуміння логіки, яка лежить в основі прийнятих моделлю рішень.

Дерева рішень, які застосовуються в класифікації та регресії, розвиваються через поетапне додавання вузлів, що формують запитання. Ці запитання визначаються на основі даних з навчального набору [23]. В ідеальних умовах, одне ефективне запитання б здатне ідеально розділити навчальні приклади за класами. Однак, у випадках, коли таке ідеальне розмежування неможливе, вибирається питання, що найбільш точно розділяє приклади.

Ефективне запитання має здатність розбивати набір даних з різноманітними класами на підмножини з більш однорідними мітками класів. Таке розділення допомагає організувати дані таким чином, що кожен новий вузол містить більш консистентну інформацію. Виникає питання, як оцінити ефективність запитань або логічних тестів у розділенні класів? У деревах рішень для цього використовуються два основні методи: ентропія Шенона і індекс Джині.

Ентропія Шенона вимірює ступінь невизначеності або розсіяності в даних, допомагаючи оцінити, наскільки добре питання розділяє класи. З іншого боку, індекс Джині визначає ймовірність того, що випадково обраний елемент буде неправильно класифікований. Обидва ці методи допомагають у виборі найбільш ефективних запитань для створення роботизованих, але точних моделей у деревах рішень.

У задачі класифікації, де метою є розподіл об'єктів у m різних класів, використовуючи певний набір навчальних матеріалів $\mathbb{E}$ ми можемо ідентифікувати $p_i$ ($i = 1,\ldots, m$) як групи елементів $\mathbb{E}$, що належать до і-го класу. Важливим аспектом тут є ентропія, яка служить мірою розподілу ймовірностей об’єктів у цих множинах. Низька ентропія спостерігається, коли одна ймовірність дорівнює 1, тоді як інші – 0. Навпаки, максимальна ентропія досягається, коли всі ймовірності є однаковими. Ще одним важливим показником є Індекс Джині, який використовується для оцінки впорядкованості і розраховується за формулою:

$$1 - \sum_{i=1}^{m} p_i^2$$

Цікаво, що обидва ці показники відіграють ключову роль у визначенні ефективності класифікаційних моделей, допомагаючи розробникам зрозуміти, наскільки добре модель може розпізнавати та розділяти різні класи.

Коли в аналізі даних виникає ситуація, де індекс набуває нульового значення, це вказує на те, що всі елементи в множині $\mathbb{E}$ належать до одного класу. Цей факт важливий у контексті визначення ефективності питань у класифікаційних деревах.

Вибір оптимального питання заснований на мінімізації середньозваженого індексу впорядкованості. Це означає, що з усіх можливих питань, які поділяють дані на підмножини, необхідно вибрати те, яке призводить до найбільш ефективного розділення. Такий підхід допомагає забезпечити більш точну і виразну класифікацію, оптимізуючи процес вибору питань на основі їхньої ефективності у розподілі даних на однорідніші групи.

$$\sum_{j=1}^{k} \frac{|\mathbb{E}_j|}{|\mathbb{E}|} \mathcal{I}(\mathbb{E}_j)$$

У більшості ситуацій, для вибору оптимального рішення ефективним є перелік усіх можливих варіантів. В контексті теорії інформації, де - це ентропійна функція, посилення інформації визначається як різниця між ентропією початкового розподілу класів у верхівковому вузлі та середньозваженим значенням ентропії в нижніх вузлах. Цей інформаційний приріст, вимірюваний за допомогою розбіжності Куллбека-Лейблера, завжди має від’ємний показник.

Процес вибору питань для подальшого розділення навчальних даних на дрібніші підмножини продовжується рекурсивно, утворюючи в результаті структуру дерева [12]. Однак, ключовим моментом в застосуванні дерев рішень є контроль за їх складністю або «глибиною» для запобігання перенавчанню. Один з методів досягнення цього - припинення поділу гілок, коли подальші запитання не підвищують чистоту підмножин більше, ніж на заданий поріг. Також можливо формувати дерево до моменту, коли подальше розбиття листків стає неможливим. У такому випадку, для уникнення перенавчання, використовується обрізання дерева, яке полягає у видаленні вузлів шляхом трансформації внутрішніх вузлів у листя, якщо це зменшує помилку класифікації на валідаційній вибірці [15].

Варто зазначити, що алгоритми вирішальних дерев мають кілька значних недоліків:

– висока чутливість до вхідних даних, що призводить до ризику перенавчання;

– стабільність алгоритму є низькою, оскільки якість моделі може істотно змінюватися з новою навчальною вибіркою;

– дерева рішень не здатні до екстраполяції, що обмежує їх можливості у прогнозуванні даних, які виходять за рамки тренувальної вибірки.

### 2.2.3 Ensembles. Random forest

Індивідуальні дерева рішень можуть бути ефективними у класифікації, але часто точніші результати отримуються, коли використовується комбінація багатьох дерев. Такі ансамблі дерев рішень часто є частиною найбільш ефективних класифікаторів. В рамках ансамблевих методів, таких як випадкові ліси та boosting, дерева рішень об'єднуються за допомогою спеціальних стратегій.

У випадкових лісах, різноманітні дерева рішень розвиваються через процес, що ґрунтується на випадковому відборі. Для кожного дерева використовується вибірка даних, що формується випадковим чином з повною навчальною вибіркою, дозволяючи повторне використання деяких даних. Під час формування питань для вузлів дерева, вибирається лише частина доступних ознак. Ці дві характеристики забезпечують унікальність кожного дерева у ансамблі. Рішення ансамблю базується на найчастішому прогнозі серед усіх дерев. Це зменшує ризик неправильної класифікації нових даних, оскільки помилка одного дерева не є вирішальною для загального результату.

Такий підхід дещо аналогічний до групового рішення у людських командах, де різні точки зору і досвід членів команди вносять свій вклад у більш точне і зважене рішення [24]. Це підкреслює значення різноманітності та колаборації як у машинному навчанні, так і в людській взаємодії.

Boosting є передовою методикою машинного навчання, що займається поєднанням декількох слабких класифікаторів в один сильний. Цей процес включає повторне налаштування ваг навчальних прикладів, акцентуючи на тих, що викликають найбільші труднощі, для підвищення ефективності [19]. У реальному

світі boosting часто використовується для створення потужних комбінацій з дерев рішень.

Особливим варіантом є чергуючі дерева рішень, які являють собою розширення звичайних дерев рішень. Вони створюються шляхом застосування специфічної техніки boosting, що об'єднує слабкі класифікатори, базуючись на так званих "пнях", що є базовими деревами рішень із одним вузлом запитання. У чергуючих деревах рівні вузлів чергуються між стандартними вузлами запитань і спеціалізованими вузлами, що містять ваги і можуть мати різну кількість нижчих вузлів. Відмінність таких дерев від традиційних полягає у тому, що об'єкти можуть слідувати кількома шляхами і відносяться до певних класів на основі ваг, які вони зустрічають на цих шляхах. Ця методологія дозволяє створювати класифікатори, які є не тільки більш компактними, але й легшими для інтерпретації в порівнянні з класифікаторами, згенерованими за допомогою прямого застосування boosting до стандартних дерев рішень.

Така методика, як boosting, може бути ефективно використана в різноманітних галузях, від фінансового моделювання до медичної діагностики, демонструючи величезний потенціал у вирішенні складних завдань класифікації.

### 2.3 Вибір засобів реалізації

Після детального аналізу наукових публікацій лідерів у галузі аналітики даних та машинного навчання, було вирішено, що для створення передової системи прогнозування відтоку клієнтів, оптимальним вибором буде використання мови програмування Python версії 3.7. Ця мова відома своєю високорівневою об’єктно-орієнтованою структурою, інтерпретованістю та строгою динамічною типізацією, що робить її ідеальною для швидкого розвитку програмного забезпечення та інтеграції з існуючими компонентами.

Створена у 1990 році Гвидо ван Россумом, Python забезпечує високий рівень структур даних, що, у поєднанні з динамічною семантикою, забезпечує

ефективність у розвитку програмного забезпечення. Однією з ключових особливостей Python є підтримка модулів та пакетів, що значно спрощує модульність та дозволяє повторно використовувати код, що є важливим для проектів, пов’язаних з машинним навчанням та аналізом даних.

За останнє десятиліття, Python став надзвичайно популярним серед програмістів. Його широко використовують у створенні складних інтелектуальних систем у провідних технологічних компаніях, таких як Google, Facebook, Amazon, YouTube, Netflix, демонструючи його універсальність і потужність. (рис. 2.2).

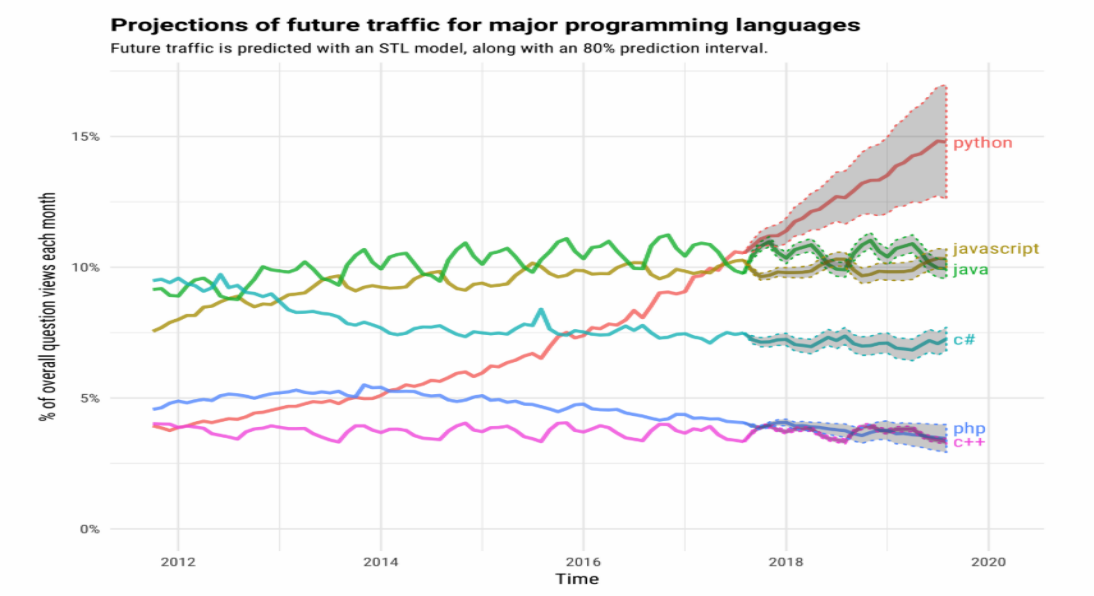



Рисунок 2.2 – Популярність мови Python

Мова програмування, яку ми обрали, включає в себе багатий набір модулів та пакетів. Ці інструменти надають можливість використовувати функції лінійної алгебри, чисельного аналізу, теорії імовірностей та статистики, що істотно підвищує швидкість виконання обчислень, як на персональному комп'ютері, так і на віддалених серверах. У проекті з використанням Python 3.7 будуть задіяні такі пакети:

1) sklearn: ця бібліотека є ключовою для застосування алгоритмів машинного навчання і їх покращення;

2) numpy: забезпечує високошвидкісні обчислення, використовуючи інструменти лінійної алгебри;

3) pandas: використовується для ефективної обробки та аналізу даних;

4) matplotlib та seaborn: ці бібліотеки відіграють важливу роль у візуалізації даних, що сприяє кращому розумінню та аналізу;

5) flask: ідеальний для створення REST-API систем, дозволяє легко розгортати інтелектуальні системи на серверах.

Sklearn, або Scikit-learn, є однією з найбільш популярних бібліотек для машинного навчання у Python. Ця бібліотека включає в себе широкий спектр інструментів для статистичного моделювання та машинного навчання, включаючи, але не обмежуючись, класифікацією, регресією, кластеризацією та зменшенням розмірності. Важливо відзначити, що sklearn оптимізований для моделювання, але не для безпосередньої роботи з даними, такої як їхнє читання, маніпулювання ними чи узагальнення.

Scikit-learn, знаменита бібліотека Python, вражає своєю різноманітністю функцій. Вона включає розгалужений набір алгоритмів машинного навчання з учителем, від лінійних моделей, таких як лінійна регресія, до складніших технік, таких як опорні векторні машини (SVM), дерева рішень, а також байєсівські методи. Широкий спектр цих алгоритмів є ключовим фактором її високої популярності у наукових дослідженнях.

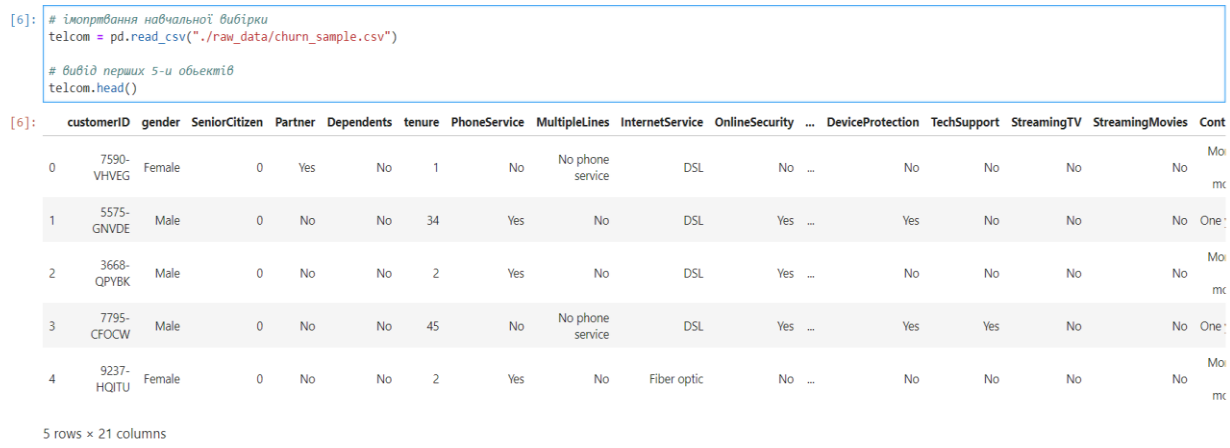
Ще одна важлива особливість Scikit-learn – це валідація моделей. Бібліотека пропонує різноманітні методики для перевірки точності навчених моделей на невідомих даних, що дозволяє отримати більш надійні результати.

Крім того, вона включає алгоритми машинного навчання без вчителя, такі як кластеризація, факторний аналіз, аналіз основних компонентів, а також некеровані нейронні мережі. Ці методи знаходять широке застосування в різноманітних дослідженнях, від маркетингових аналізів до вивчення геноміки, підтверджуючи внесок Scikit-learn у різні сфери науки та технологій.

# 3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОСТИЧНОГО МОДЕЛЮВАННЯ НАДАННЯ ПОСЛУГ МОБІЛЬНОГО ЗВ'ЯЗКУ

## 3.1 Первинний огляд вихідних даних

При роботі з прогнозуванням та аналізом даних, ключовим аспектом є наявність репрезентативного набору даних, що включає в себе вибірку об'єктів із їх характеристиками або додатковими історично зібраними даними. Ці дані складають "dataset" або навчальну вибірку, яка вимагає аналізу, а в разі наявності в ній значимих шаблонів чи патернів, може бути використана для створення прогнозуючих моделей. На рисунку 3.1 представлено фрагмент такої навчальної вибірки, який демонструє приклад даних для інформаційної системи.



```
[6]: # імопртвання навчальної вибірки
     telcom = pd.read_csv("./raw_data/churn_sample.csv")

     # вивід перших 5-и обьектів
     telcom.head()
```

[6]:

| | customerID | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines | InternetService | OnlineSecurity | ... | DeviceProtection | TechSupport | StreamingTV | StreamingMovies | Cont |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 7590-VHVEG | Female | 0 | Yes | No | 1 | No | No phone service | DSL | No | ... | No | No | No | No | Mo mc |
| 1 | 5575-GNVDE | Male | 0 | No | No | 34 | Yes | No | DSL | Yes | ... | Yes | No | No | No | One |
| 2 | 3668-QPYBK | Male | 0 | No | No | 2 | Yes | No | DSL | Yes | ... | No | No | No | No | Mo mc |
| 3 | 7795-CFOCW | Male | 0 | No | No | 45 | No | No phone service | DSL | Yes | ... | Yes | Yes | No | No | One |
| 4 | 9237-HQITU | Female | 0 | No | No | 2 | Yes | No | Fiber optic | No | ... | No | No | No | No | Mo mc |

5 rows × 21 columns

Рисунок 3.1 – Фрагмент навчальної вибірки

Надалі, важливим етапом є дослідження розміру навчальної вибірки, вивчення атрибутів об'єктів та перевірка на наявність пропущених значень [13]. Для цього ми використовуємо спеціальні атрибути та методи бібліотеки pandas. Усі ці дані, включаючи розмір вибірки та детальний опис атрибутів, можна знайти на рисунку 3.2. Цей аналіз є надзвичайно важливим, оскільки він дозволяє оцінити повноту та якість даних, які будуть використані для подальшого моделювання.

```
]: # розмір навчальної вибірки
   telcom.shape

]: (7043, 21)

]: # інформація щодо атрибутів
   telcom.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null object
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null object
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
TotalCharges        7043 non-null object
Churn               7043 non-null object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

Рисунок 3.2 – Вхідні дані

Після первісного огляду даних, ми можемо підвести наступні підсумки:

– у навчальній вибірці представлено 21 атрибут, серед яких ключовим є цільова змінна "Churn";

– ця вибірка включає інформацію про 7043 різних об’єктів;

– при першому аналізі відсутність пропущених значень не було виявлено;

– деякі атрибути потребують зміни типу даних для оптимальної обробки.

Далі нас очікує проведення описового аналізу даних (Exploratory Data Analysis - EDA). EDA є критичним етапом, який надає інсайти та напрямки для подальшого більш детального аналізу, сприяючи ефективнішому моделюванню та прогнозуванню [13]. Цей процес дозволить нам глибше погрузитися в аналіз вихідних даних, вивчити розподіл кожного атрибуту та оцінити їхній вплив на інші змінні та, зокрема, на цільову змінну Churn.

### 3.2 Описовий аналіз вихідних даних

Щоб здійснити глибший аналіз даних, критично важливо провести візуальний оцінювання кожного атрибуту та його впливу на цільову змінну. Це досягається через методи візуального аналізу даних. Першим кроком у цьому процесі є візуалізація розподілу ключової змінної, Churn. Ця змінна є бінарною: значення 1 вказує на відмову клієнта від послуг, тоді як 0 означає, що клієнт продовжує співпрацю з компанією. Відповідна діаграма представлена на рисунку 3.3.

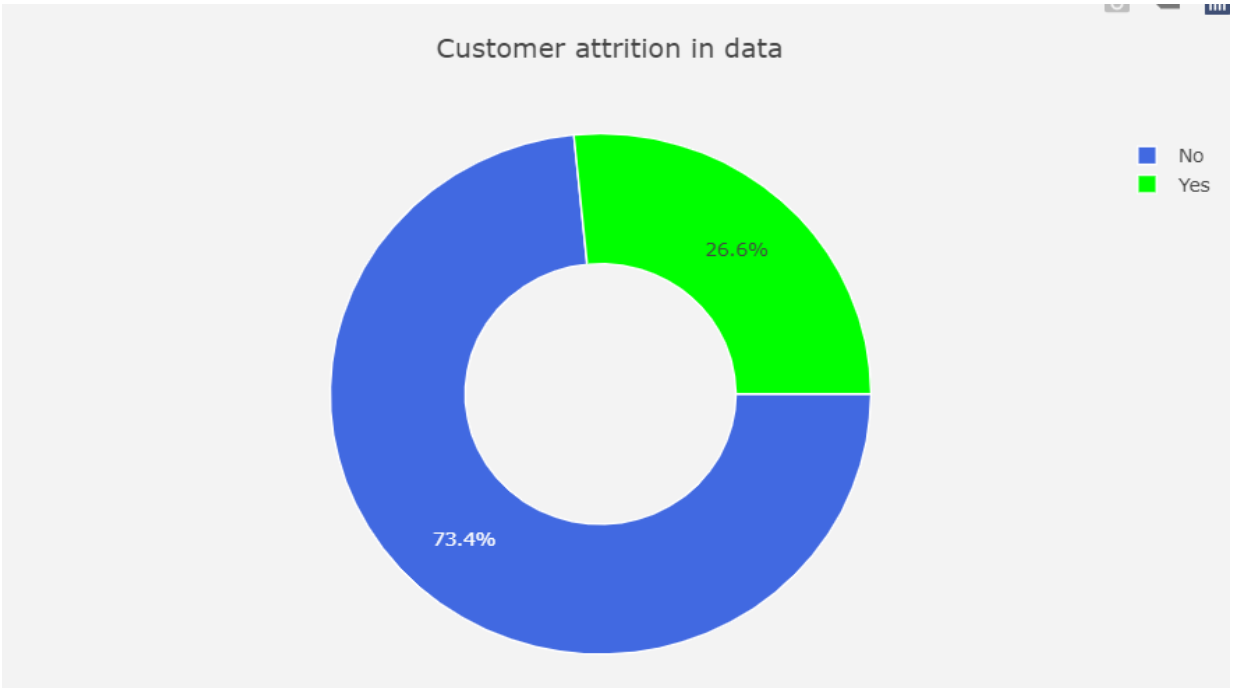



Рисунок 3.3 – Візуалізація розподілу атрибуту Churn

З діаграми видно, що в нашій вибірці близько 27% об'єктів припадає на клієнтів, які відмовились від послуг. Це вказує на необхідність розробки моделі, яка враховує дисбаланс класів у цільовій змінній. Такий дисбаланс може вплинути на моделювання, оскільки існує ризик, що модель надмірно "запам'ятає" патерни класу, який має перевагу у вибірці, та не врахує клас, що представлений у меншості. Тому, під час розробки, важливо враховувати цей аспект, щоб забезпечити баланс та точність передбачень моделі. Оптимальний підхід полягає у використанні

спеціальних технік для балансування класів або вдосконалення алгоритмів, що може підвищити загальну ефективність прогнозування.

Продовжуючи аналіз, наступним кроком є створення графіків, що ілюструють розподіл змінних з урахуванням цільової змінної. Цей процес є вирішальним у розумінні даних і бізнес-контексту, дозволяючи виявити потенційні причинно-наслідкові зв'язки [13]. Для візуалізації категорійних атрибутів ми використовуємо кругові діаграми (pie-charts).

Однією з ключових змінних є стать клієнта (gender). Відповідний розподіл представлено на рисунку 3.4.

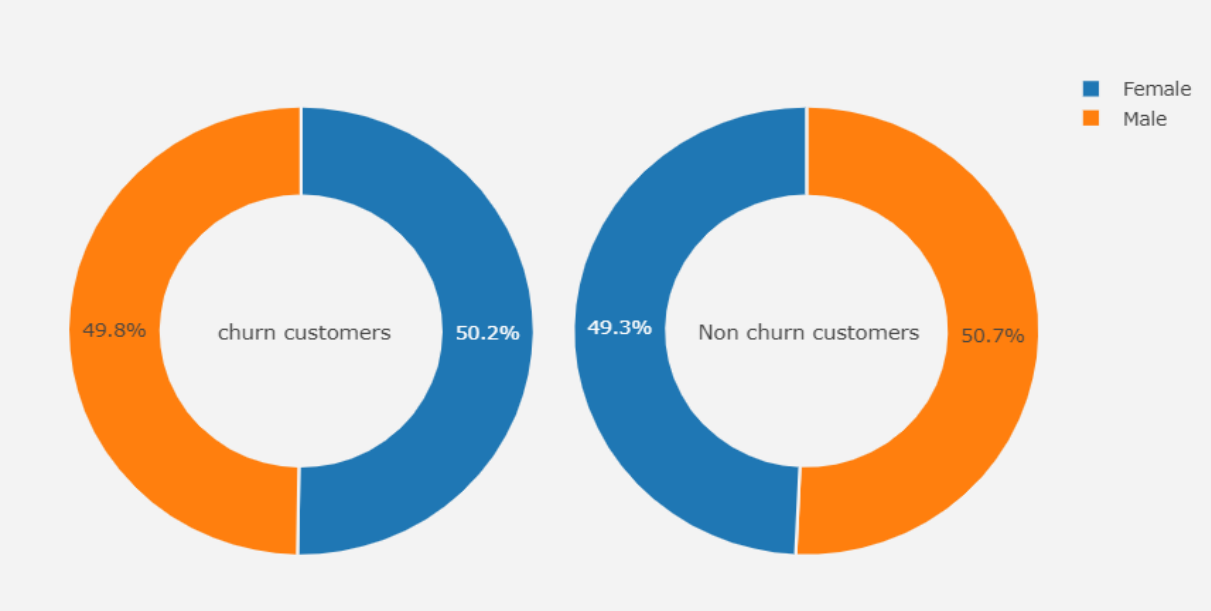



Рисунок 3.4 – Аналіз змінної gender в контексті змінної churn

На графіку представлено, як відрізняється відмова від послуг серед різних статей. Кожен «шматок пирога» вказує на відсоток відмов, а кольори відображають гендерну групу. З графіка видно, що пропорції між гендерними групами є приблизно однаковими в контексті відмови від послуг.

На цьому етапі важливо не робити поспішних висновків щодо впливу такого атрибуту як стать на рішення клієнта про відмову від послуг. Аналізуючи лише дві змінні, ми можемо пропустити інші важливі фактори. Наприклад, включення

третьої змінної, такої як «наявність дітей», може істотно змінити картину і показати, що стать може мати значний вплив у певному контексті. Таким чином, багатофакторний аналіз є ключовим для отримання більш точної картини впливу різних атрибутів на цільову змінну.

Змінна SeniorCetizenzhip. Розподіл змінної зображено на рисунку 3.5.

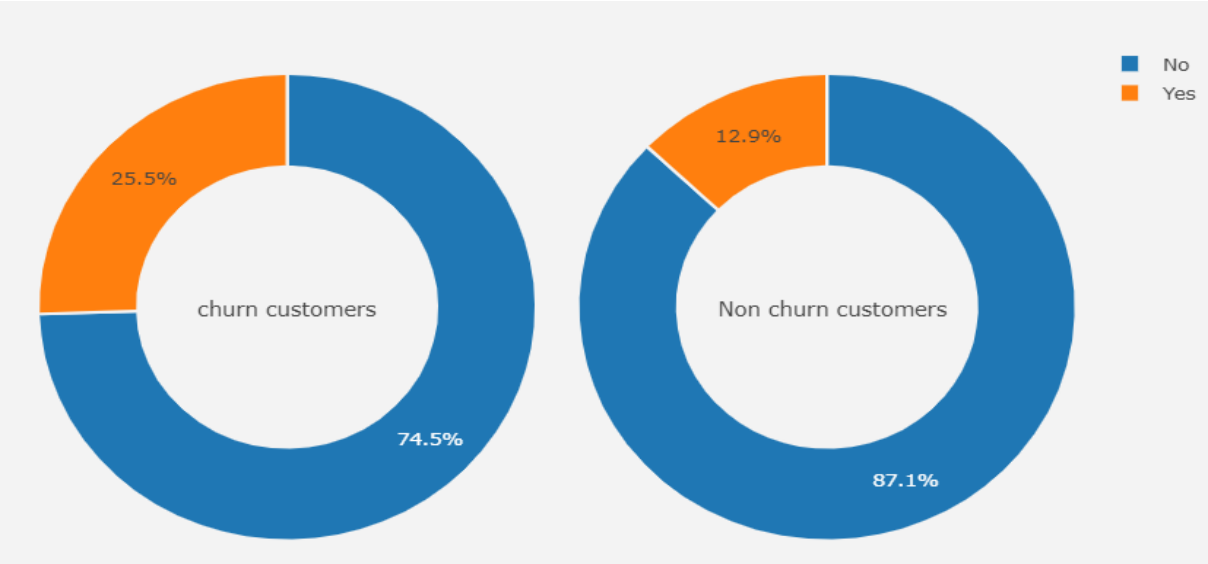



Рисунок 3.5 – Аналіз змінної SeniorCetizenship в розрізі churn

Спостерігаючи дані, ми бачимо, що серед тих, хто припинив користуватися послугами, 25% становлять пенсіонери, у той час як серед лояльних клієнтів пенсіонери складають лише 13%.

Що стосується змінної "Partner", деталі якої відображені на рисунку 3.6, ми виявили, що 64% клієнтів, які відмовилися, не мали партнера (Partner = No), у порівнянні з 47% серед тих, хто залишився з послугами.

Ці спостереження надзвичайно цінні, оскільки візуальний аналіз дозволяє виявити ключові атрибути, що мають значний вплив на цільову змінну. Вони допомагають глибше проникнути в суть проблеми та краще зрозуміти бізнес-контекст. Особливо важливим є виявлення таких зразків у поведінці клієнтів, як вплив сімейного статусу або віку на лояльність, що може бути використано для

розробки більш цілеспрямованих маркетингових стратегій та поліпшення якості обслуговування.

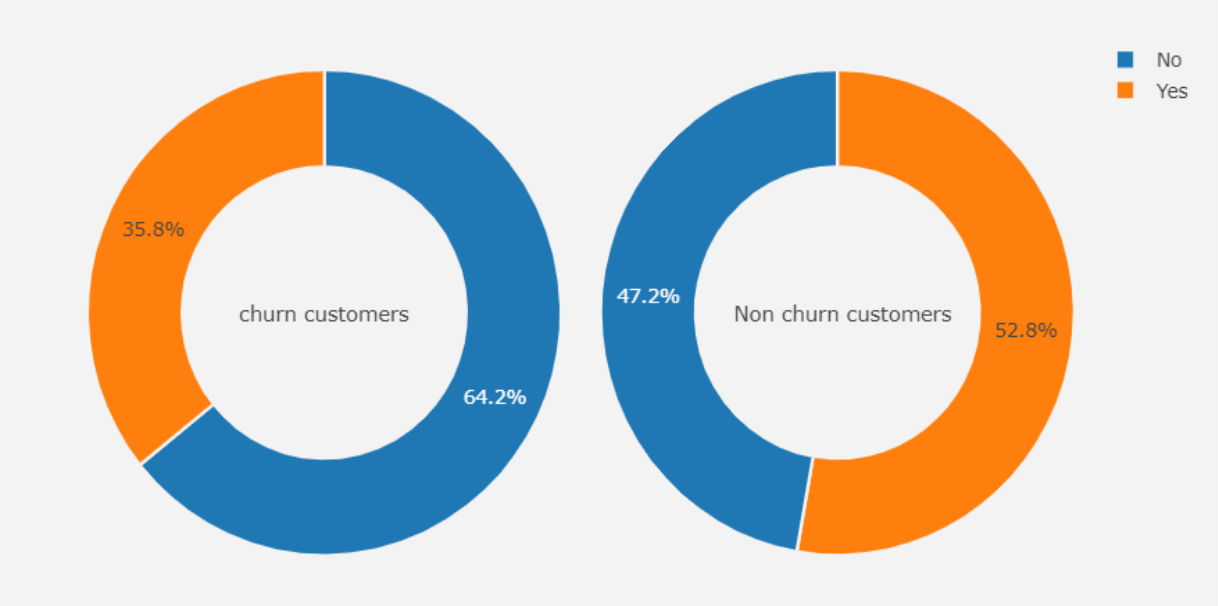



Рисунок 3.6 – Аналіз змінної Partner в розрізі атрибутуChurn

Побудуємо діаграму для змінної «Has dependepts» (рис. 3.7).

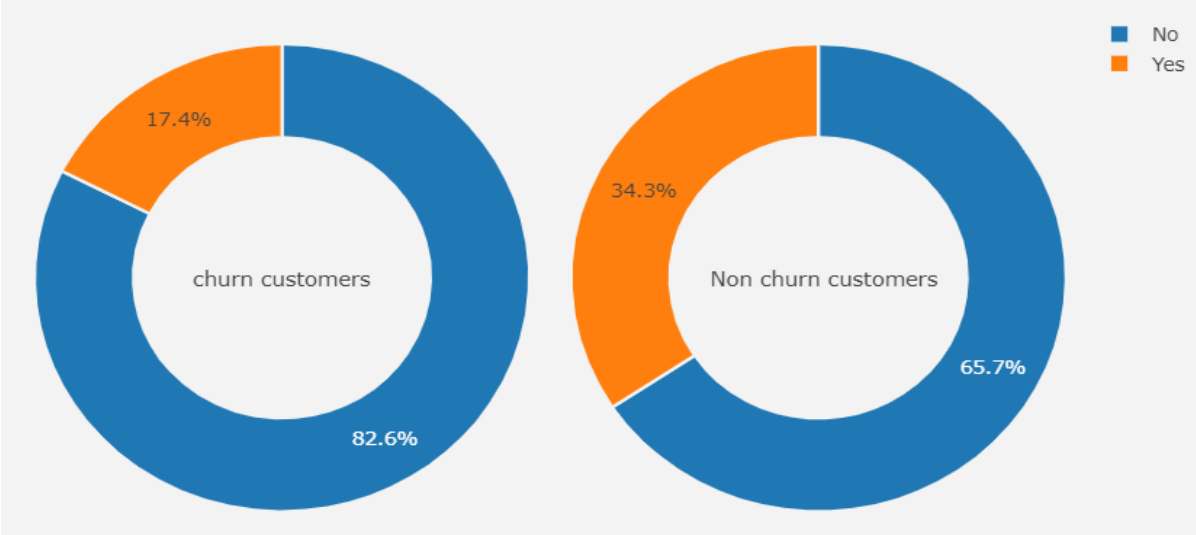



Рисунок 3.7 – Аналіз змінної Has dependets в розрізі Churn

Аналізуючи клас клієнтів, які припинили користуватися послугами, ми виявили, що більшість з них не мали залежних («Has dependents» = No), на відміну

від тих, хто залишився з послугою.

Після створення графіків розподілу для кожного з категоріальних атрибутів та їх ретельного аналізу, було виявлено, що:

- Атрибут «PhoneService» при первинному аналізі не впливає на цільову змінну Churn;
- Атрибут «MultipleLines» також не має помітного впливу на Churn, оскільки пропорції в обох групах схожі;
- «InternetService» виявився важливим фактором: клієнти, які не користуються цією послугою, менш схильні до відмови;
- Клієнти з активним фактором «OnlineSecurity=Yes» та «OnlineBackup=Yes» менш схильні до відтоку;
- Атрибути «StreamingTV» та «StreamingMovies» не мають значного впливу на Churn;
- «TechSupport»: наявність цієї послуги зменшує ймовірність відмови від послуг;
- «ContractType»: клієнти з місячною оплатою більш схильні до відтоку, порівняно з тими, хто оплачує послуги на рік чи два;
- «PaperlessBilling» показав значні різниці у розподілі між групами, що свідчить про його вплив;
- «PaymentMethod»: спосіб оплати значно впливає на рішення про відмову від послуг.

Наступним кроком є аналіз атрибутів із числовими даними: «Tenure» (тривалість користування послугами) та «MonthlyCharges/TotalCharges» (місячні та загальні витрати клієнта). Краще згрупувати ці дані у відповідні числові інтервали, щоб полегшити аналіз та зробити його більш наочним. Цей крок важливий, оскільки він допоможе визначити, як довготривалість взаємодії клієнта з провайдером та величина їх витрат впливають на їхнє рішення залишити послуги.

Побудуємо діаграму розподілу для атрибуту Tenure (рис. 3.8).

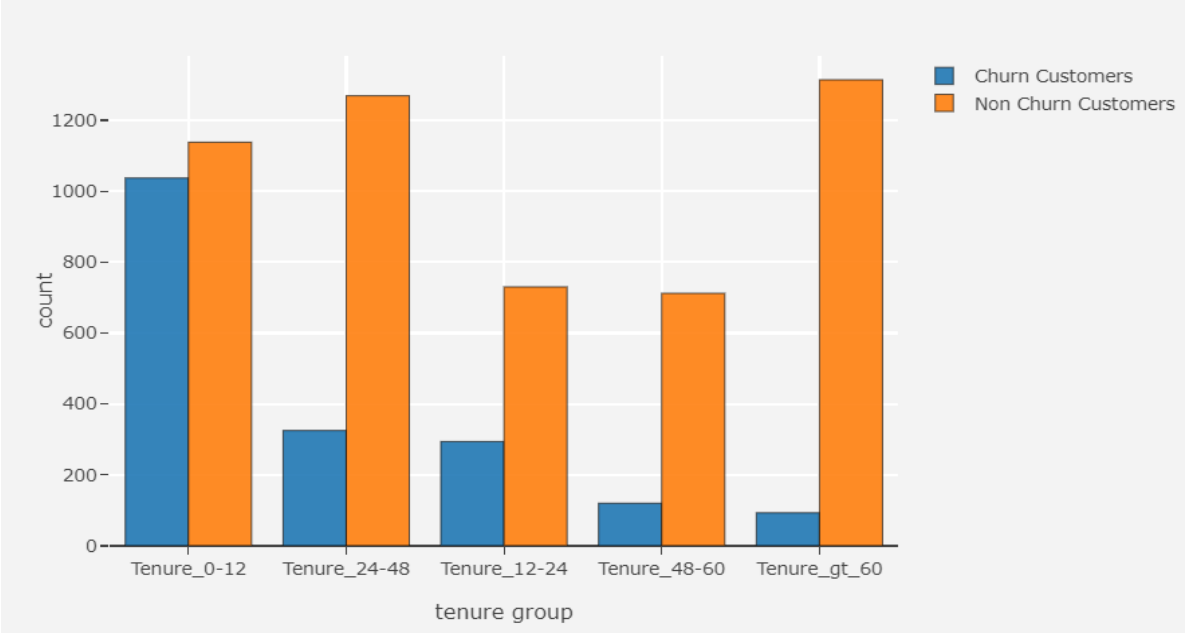



Рисунок 3.8 – Аналіз змінної Tenure в розрізі атрибуту Churn

З аналізу графіка видно, що користувачі, які використовують послуги менше ніж 12 місяців, частіше відмовляються від них. Навпаки, ті, хто користується послугами довший час, рідше приймають рішення про розірвання зв'язку з компанією. Це може бути пов'язано з підвищеною лояльністю до компанії, яка розвивається з часом.

Далі ми створимо подібні діаграми для атрибутів "MonthlyCharges" та "TotalCharges", використовуючи методику бінінгу, а також згрупуємо дані за атрибутом "Tenure group". Ці діаграми, зображені на рисунках 3.9 та 3.10, допоможуть нам краще зрозуміти, як величина щомісячних та загальних витрат впливає на рішення клієнтів щодо відмови від послуг. Цей аналіз є важливим, оскільки він може вказувати на залежність рівня задоволеності клієнтів від вартості послуг, які вони сплачують, і, отже, на потенційні можливості для оптимізації ціноутворення або підвищення якості обслуговування.

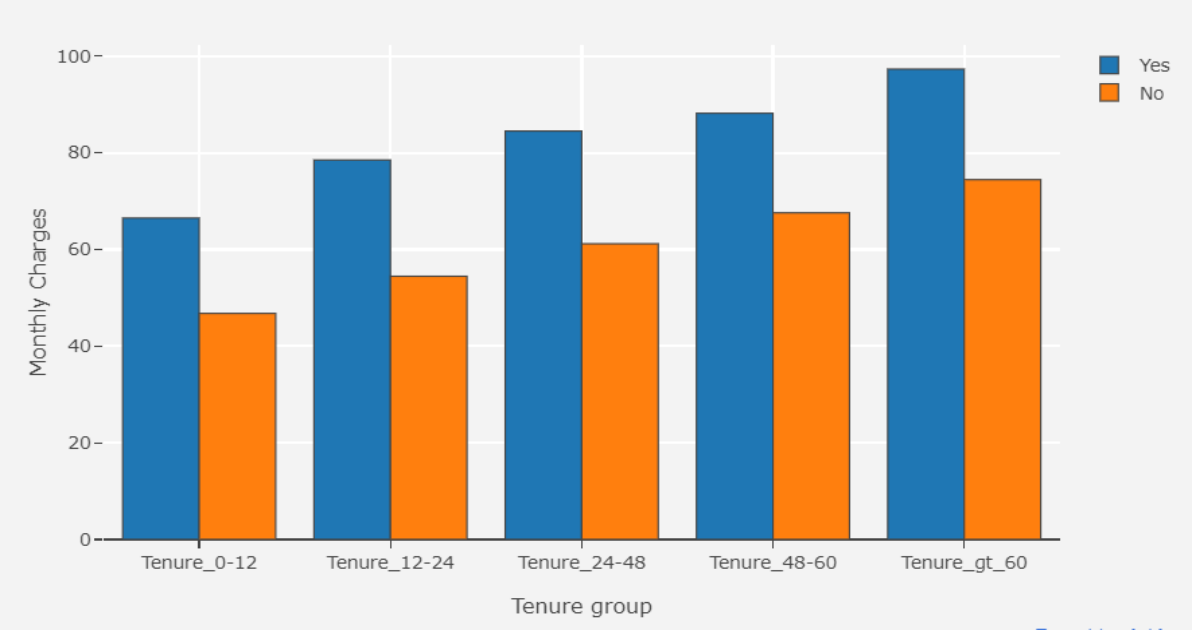



Рисунок 3.9 – Аналіз змінної Monthly Charges» в розрізі атрибуту Churn

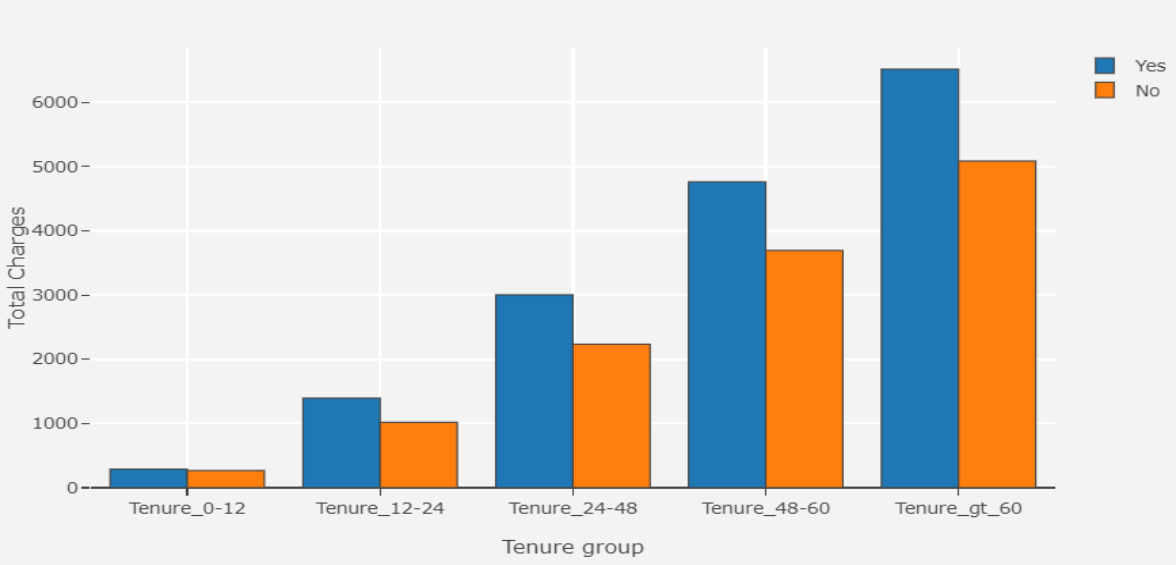



Рисунок 3.10 – Аналіз змінної Total Charges в розріз атрибуту Churn

На рисунках 3.9 та 3.10, де представлені розподіли за «бінами» групи Tenure, ми бачимо, що пропорції клієнтів у кожній групі є приблизно однаковими. Це вказує на те, що на даному етапі атрибути "Monthly Charges" та "Total Charges" не здаються значно впливовими на цільову змінну.

Наступний крок полягає у вивченні кореляцій між різними атрибутами. Це дозволяє визначити, які змінні є лінійно залежними одна від одної. Наявність

сильної кореляції може вказувати як на сильний вплив однієї змінної на іншу, так і на можливість, що сильно корелюючі змінні можуть вносити зайву або повторювану інформацію в модель, що може знижувати її точність [14]. Матриця кореляцій атрибутів представлена на рисунку 3.11. Для визначення характеру кореляції (позитивної, нейтральної або негативної) використовується колірна шкала, де жовтий колір символізує сильну позитивну кореляцію, а темно-фіолетовий – сильну негативну. На діагоналі матриці знаходяться кореляції атрибутів самі з собою, які не беруться до уваги при аналізі. Важливо відзначити, що аналіз кореляцій є ключовим для розуміння взаємозв'язків між різними змінними та може допомогти виявити приховані закономірності у поведінці клієнтів, що є цінним для покращення моделей прогнозування.

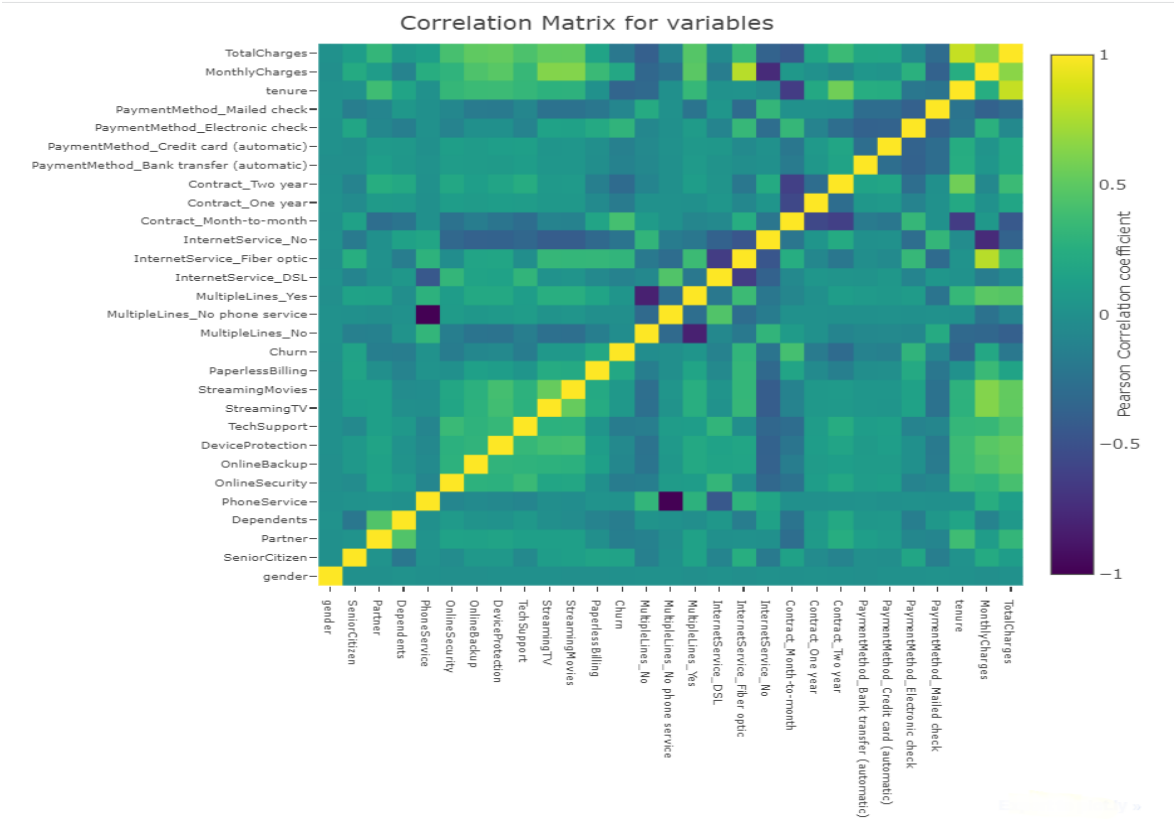



Рисунок 3.11 – Кореляційний аналіз атрибутів

Після ретельного вивчення матриці кореляцій, ми можемо зробити декілька ключових висновків:

- Виявлено, що атрибути, пов'язані з додатковими послугами, як-от "InternetService", "StreamingVideo", "StreamingTV", мають позитивну кореляцію з "Monthly Charges". Це логічно, оскільки чим більше послуг клієнт використовує, тим вищими будуть щомісячні платежі.
- "Monthly Charges" та "Total Charges" також мають сильну кореляцію, що очікувано, адже один атрибут є частиною іншого.
- Особливо важливим є той факт, що на цільову змінну "Churn" істотно впливають такі фактори, як тип контракту на місяць ("Contract-month-to-month"), метод оплати ("Payment Method") та "Monthly Charges".

Таким чином, ціллю проведення розвідувального аналізу даних є глибоке розуміння інформації, що, в свою чергу, сприяє кращому усвідомленню проблематики. Він допомагає виявити інтересні факти, які можуть бути використані для створення нових атрибутів (feature engineering) та формування нових гіпотез. Це не лише сприяє розробці ефективних моделей для даної проблеми, але й відкриває можливості для дослідження в інших сферах. Також цей аналіз може надати цінні інсайти для бізнесу щодо удосконалення стратегій збереження клієнтів та оптимізації послуг.

### 3.3 Підготовка даних та створення моделі. Вибір найкращої моделі

Перед тим, як розпочати процес моделювання, необхідно підготувати вхідні дані таким чином, щоб вони були придатні для обробки моделлю. Для цього треба виконати кілька ключових кроків:

- Атрибути, які мають лише два текстові значення, слід перетворити у числовий формат. Це можна зробити за допомогою кодування, наприклад, призначивши одному значенню 0, а іншому - 1.
- Атрибути з трьома чи більше текстовими значеннями потрібно перетворити в розріджену (sparse) матрицю, використовуючи метод One Hot Encoding. Це дозволить моделі ефективніше обробляти категорійні

дані.

– Для атрибутів із дійсними числовими значеннями слід застосувати Z-стандартизацію, що допоможе нормалізувати дані та зробити їх порівнянними на різних масштабах.

Завершивши переобробку вихідних даних за допомогою інструментів, таких як LabelEncoder, OneHotEncoder та StandardScaler від бібліотеки sklearn, ми отримуємо дані, готові для використання у моделюванні. Візуалізацію цих оброблених даних можна побачити на рисунку 3.12. Цей крок є критично важливим для точності та ефективності подальшого моделювання, оскільки він забезпечує високу якість вхідних даних, що є фундаментом для будь-якої успішної аналітичної моделі [24].

```
telcom.head()
```

| gender | SeniorCitizen | Partner | Dependents | PhoneService | OnlineSecurity | OnlineBackup | DeviceProtection | TechSupport | ... | PaymentMethod_Electronic check | PaymentMethod_Mailed check | tenure_group_Tenure_0-12 | tenure_group_Tenure_1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | ... | 1 | 0 | 1 | |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | ... | 0 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | ... | 0 | 1 | 1 | |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | ... | 0 | 0 | 0 | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | ... | 1 | 0 | 1 | |

Рисунок 3.12 – Вигляд даних після переобробки

Раніше ми визначили, що для прогнозування відмови від послуг використовуватимемо три методи: логістична регресія, дерево рішень та випадкові ліси. Але перед початком моделювання, необхідно розділити наші дані на дві частини за допомогою методу випадкового відбору, де кожен об'єкт має рівну ймовірність бути включеним до однієї з груп:

– "train" – частина даних для навчання моделі;

– "test" – частина, яка не бере участі в навчанні, і на якій відбувається валідація моделі та здійснення остаточних висновків про її прогностичні властивості.

Згідно зі стандартами спільноти Data Science, вибірка для тестування повинна становити понад 25% від загального обсягу даних, щоб забезпечити репрезентативність результатів. Використовуючи функцію train_test_split з параметром на 25% для тестової вибірки, ми отримали 5274 об'єкти для навчання та 1758 для тестування, відібраних випадковим чином. Для оцінки моделі ми використовуємо метрики ROC AUC score та Accuracy score.

Для тренування обраних моделей ми імпортуємо у середовище Jupyter Notebook класи LogisticRegression, DecisionTreeClassifier та RandomForestClassifier. Навчання ініціюється за допомогою методу .train() для кожної моделі, а прогнозування – за допомогою .predict() або .predict_proba(), якщо потрібна ймовірність віднесення об'єкта до певного класу.

Щоб оптимізувати процес навчання та оцінювання прогностичної сили моделі, було розроблено функцію, яка демонструє якість моделі через метрики Accuracy, ROC_AUC та матрицю плутанини. Ця функція приймає на вхід об'єкт моделі з визначеними гіперпараметрами, навчальні та тестові вибірки.

Використовуючи цю функцію, ми отримали аналітичну оцінку прогностичної спроможності моделі. Детальний опис процесу навчання моделі логістичної регресії представлено на рисунку 3.13. Цей підхід дає можливість не тільки оцінити ефективність кожної моделі окремо, але й порівняти їх між собою, що є важливим для вибору найкращої стратегії прогнозування у конкретному дослідженні.

```
---------------------- TRAINING REPORT ------------------------

Accuracy score: train sample=0.750 vs test sample=0.753
ROC_AUC score: train sample=0.768 vs test sample=0.761

==================== Classification report ====================
              precision    recall  f1-score   support

           0       0.90      0.74      0.82      1291
           1       0.52      0.78      0.63       467

    accuracy                           0.75      1758
   macro avg       0.71      0.76      0.72      1758
weighted avg       0.80      0.75      0.76      1758


==================== Confusion matrix ==========================
[[959 332]
 [103 364]]
```

Рисунок 3.13 – LogReg

На рисунку 3.13 видно, що логістична регресія показує дуже непогані результати. Зокрема, показники точності (Accuracy) та площі під ROC-кривою (ROC AUC) для тестової вибірки становлять відмінний рівень у 76%. Крім того, варто зауважити, що при порівнянні помилок між тренувальною та тестовою вибірками немає вираженого ефекту перенавчання, що свідчить про стабільність моделі.

Далі в аналізі було розглянуто використання DecisionTreeClassifier (дерева ухвалень рішень). Ця модель має свої переваги та особливості, і її використання може бути обґрунтованим залежно від конкретної задачі. Звіт з навчання для цієї моделі представлено на рисунку 3.14.

```
---------------------- TRAINING REPORT ------------------------

Accuracy score: train sample=0.998 vs test sample=0.741
ROC_AUC score: train sample=0.999 vs test sample=0.663

==================== Classification report ====================
              precision    recall  f1-score   support

           0       0.82      0.83      0.82      1291
           1       0.51      0.49      0.50       467

    accuracy                           0.74      1758
   macro avg       0.67      0.66      0.66      1758
weighted avg       0.74      0.74      0.74      1758


==================== Confusion matrix ========================
[[1072  219]
 [ 236  231]]
```

Рисунок 3.14 – DecisionTreeClassifier

Згідно з інформацією зображеною на таблиці зображеної на рисунку 3.14, дерево ухвалень рішень показало не надто задовільний результат під час тестування через високий рівень похибки. Крім того, очевидно, що навчена модель дуже сильно перенавчилася на тренувальній вибірці. Проте, слід зазначити, що ця ситуація може змінитися після підбору оптимальних гіперпараметрів моделі.

Останнім варіантом для моделі розглядається використання RandomForestClassifier, який є методом випадкових лісів. Варто відзначити, що цей метод може мати свої переваги та особливості, і його використання може бути обґрунтованим, залежно від контексту завдання. Подробиці навчання цієї моделі представлені на рисунку 3.15.

```
---------------------- TRAINING REPORT ------------------------

Accuracy score: train sample=0.749 vs test sample=0.742
ROC_AUC score: train sample=0.777 vs test sample=0.761

==================== Classification report ====================
              precision    recall  f1-score   support

           0       0.91      0.72      0.80      1291
           1       0.51      0.80      0.62       467

    accuracy                           0.74      1758
   macro avg       0.71      0.76      0.71      1758
weighted avg       0.80      0.74      0.76      1758


==================== Confusion matrix ==========================
[[930 361]
 [ 93 374]]
```

Рисунок 3.15 – RandomForestClassifier

Детально проаналізувавши звіт, можна побачити, що метод випадкових лісів показує найкращий результат по метрикам оцінення Accuracy та ROC AUC без поправки гіперпараметрів моделі: 74.2% Accuracy та 76% ROC AUC та досить високий показник F1 роблять RandomForestClassifier найліпшим кандидатом. Також, проаналізувавши матрицю відношень можна зробити висновок, що RFC краще знаходить об’єкти класу Churn, аніж це робить LogisticRegression.

Наступним етапом в створенні прогнозуючої моделі є підбір найкращих гіперпараметрів моделі. Для цього імпортуємо у середовище об’єкт під назвою RandomizedSearchCV. Даний клас дозволяє реалізувати підбір гіперпараметрів для кожної моделі методом грубого перебору, т.з. brute force, або відбором параметрів з розподілу ймовірностей.

Основними гіперпараметрами для Логістичної регресії в реалізації пакету sklearn, які знайдені в роботі:

– C – так званий коефіцієнт регуляризації

– Penalty – метод регуляризації, можливі аргументи l2 та l1, які означають Ridge або Lasso методи регуляризації.

Для дерева ухвалень рішень та моделі методу випадкових лісів були обрані такі гіперпараметри як:

– Max_depth – максимально допустима глибина дерева, один з найважливіших гіперпараметрів, який впливає на якість на стабільність моделі;

– Max_features – кількість атрибутів, які присутні в дереві;

– Min_samples_split – допустимий мінімум кількості об’єктів в ноді дерева/дерев для подальшого розділу;

– Min_samples_leaf – мінімально допустима кількість об’єктів в фінальноми листі дерева/дерев;

– N_estimators – тільки для RandomForestClassifier, кількість дерев ухвалень рішень, які надалі будуть агрегуватись у фінальну відповідь [27].

Після пошуку найбільш оптимальних параметрів фаворитом залишається RandomForestClassifier. Порівняння моделей до та після пошуку оптимальних гіперпараметрів застосованих на тестовій вибірці наведено у таблиці 3.1.

Таблиця 3.1 – Порівняльна таблиця навчених моделей

| Назва моделі | ROC_AUC **before** GS | ROC_AUC **after** GS | Accuracy **before** GS | Accuracy **after** GS |
|---|---|---|---|---|
| LogisticRegression | 0.761 | 0.764 | 0.751 | 0.755 |
| DecisionTree | 0.998 | 0.67 | 0.741 | 0.75 |
| RandomForest | 0.761 | 0.77 | 0.742 | 0.79 |

Як видно з таблиці, після налаштування гіперпараметрів RandomForestClassifier виявився найкращим варіантом, демонструючи найвищу точність (Accuracy) та площу під ROC-кривою (ROC AUC) порівняно з іншими моделями.

## 3.4 Розроблення веб-додатка для автоматизації прогнозування. Тестування додатка

Коли спеціаліст із аналізу даних або інженер із машинного навчання приступає до створення проекту, що включає використання таких інструментів, як Scikit-Learn, TensorFlow, Keras, PyTorch та інші, головною задачею є забезпечення його функціональності у реальних умовах виробництва. Часто, протягом роботи над таким проектом, основна увага приділяється дослідницькому аналізу даних (EDA), створенню нових атрибутів (feature engineering), налаштуванню гіперпараметрів тощо. Все це, хоч і є важливими етапами, не становить цілісного проекту.

Імплементація моделей машинного навчання в реальні умови виробництва – це процес надання можливості вашим моделям бути використаними кінцевими користувачами або системами. Проте, існує певний комплекс завдань, пов’язаних із розгортанням цих моделей. Цей аспект проекту часто включає розробку мікросервісу з використанням Flask API і вже навченої моделі.

Ключовим елементом є створення так званого пайплайну прогнозувальної моделі. Під цим терміном розуміється структура, яка включає в себе кілька послідовних кроків: кожен з них бере на вхід дані, оброблені на попередньому етапі, і передає результати далі[25]. Це означає, що кожен етап пайплайну допомагає удосконалити та адаптувати оброблення даних, щоб модель працювала якнайефективніше.

Важливо зазначити, що успіх проекту машинного навчання не лише в тому, як добре модель навчена, а й у здатності цієї моделі адаптуватися та виконувати свої функції в реальних умовах виробництва. Це потребує глибокого розуміння не тільки теоретичних аспектів машинного навчання, але й практичних навичок в реалізації моделей у виробничому середовищі.

Створення "пайплайну" в області машинного навчання можна ефективно здійснити за допомогою інструментів, які пропонує бібліотека Sklearn, зокрема класів Pipeline та FeatureUnion. Це дозволяє систематизувати та оптимізувати

процес обробки даних. Ось кілька ключових компонентів такого пайплайну:

VariableSelector: Це спеціалізований клас, який дозволяє відбирати специфічні атрибути з даних за їх назвами. Це забезпечує вищий рівень контролю над тим, які дані використовуються в моделі.

SimpleImputer: Використовується для заповнення пропущених значень у даних. Для числових атрибутів використовується середнє арифметичне, тоді як для текстових – найбільш часто зустрічаємо значення. Це допомагає зменшити вплив відсутності даних на якість моделі.

MultiLabelEncoder: Важливий для перетворення текстових даних у числові. Наприклад, перетворення масиву значень типу ['Yes', 'No', 'Maybe'] в числову форму, таку як [1, 0, 2], полегшує обробку таких даних моделлю.

StandardScaler: Нормалізує числові дані так, щоб вони мали середнє значення 0 та стандартне відхилення 1. Це допомагає уникнути проблем зі зміщенням даних та підвищує точність моделі.

RandomForestClassifier: Цей клас використовується для включення вже натренованої моделі з усіма необхідними гіперпараметрами. Це ключовий етап, який відповідає за прогнозування.

Важливим моментом є те, що такий пайплайн дозволяє створювати більш структуровані та ефективні моделі машинного навчання, забезпечуючи краще розуміння та контроль над процесом обробки даних. Використання такого підходу може значно підвищити якість прогнозувань та ефективність проектів у галузі машинного навчання.

Реалізація пайплайну наведена на рисунку 3.16.

```
[77]: pipe = Pipeline([
          ("features", FeatureUnion([
              ('categorical', make_pipeline(VariableSelector(names = cat_features), MultiColumnLabelEncoder())),
              ('numeric', make_pipeline(VariableSelector(names = num_features), SimpleImputer(strategy='mean'), StandardScaler()))
          ])
          ),
          ('prediction', grid.best_estimator_)
      ]
      )
```

Рисунок 3.16 – Пайплайн об’єкт моделі

Для забезпечення можливості використання нашої розробленої системи обробки та аналізу даних за межами традиційного робочого середовища, зокрема в рамках мікросервісу, ми вдаємося до збереження об'єкту "пайплайну" у вигляді pickle-файлу (з розширенням .pkl). Це виконується за допомогою модуля joblib, що є частиною бібліотеки sklearn.

Pickle - це техніка, що застосовується для серіалізації та десеріалізації структур об'єктів Python. Серіалізація, відома також як «маршалінг» або «згладжування», передбачає перетворення об’єкта з форми, придатної для зберігання в пам'яті, у потік байтів, який можна зберегти на диску або передати через мережу. Пізніше цей потік байтів може бути відновлений (десеріалізований) до початкового об'єкта Python[26]. Цікаво, що «pickling» не можна плутати зі стисненням даних; в той час як стиснення зменшує об’єм даних для економії місця на диску, серіалізація перетворює формат об'єкта для його зберігання або передачі.

У контексті нашого проекту, ми зберігаємо в pickle-файлах не тільки сам об'єкт "пайплайну", але й типи вихідних даних та послідовність атрибутів, які "пайплайн" повинен обробляти. Це забезпечує, що всі ключові елементи процесу обробки даних будуть інтегровані в мікросервіс, забезпечуючи його ефективність і точність, як зображено на рисунку 3.17. Такий підхід дозволяє легко відтворювати та масштабувати аналітичні процеси, роблячи їх доступними у широкому спектрі застосувань.

```
14 joblib.dump('model.pkl')
15 joblib.dump('dtypes.pkl')
16 joblib.dump('cols_order.pkl')
```

Рисунок 3.17 – Збереження і компресія об’єктів моделі

Використання Flask для створення мікросервісу заснованого на REST архітектурі є доцільним вибором при розробці ефективного веб-додатку. Flask є легковаговим веб-фреймворком, що забезпечує необхідні інструменти, бібліотеки та технології для створення різноманітних веб-додатків - від простих веб-сторінок

до складних веб-додатків, як-от календарі або комерційні сайти. Flask відноситься до категорії мікро-фреймворків, що означає мінімальну залежність від сторонніх бібліотек та здатність до швидкої реалізації проектів.

У процесі створення мікросервісу ми імпортували ключові бібліотеки та їх модулі, такі як flask, json, numpy, pandas та joblib. Після цього було створено екземпляр мікросервісу за допомогою класу Flask та завантаження раніше збережених pickle-файлів.

Ключовим аспектом нашого мікросервісу є створення окремої кінцевої точки, або «end point», з назвою /predict, яка приймає лише POST-запити. Ця кінцева точка розроблена для прийому json об'єктів з вхідними даними, які перевіряються на відповідність типам даних за допомогою pickle-об'єкта dtypes. Необхідно також впорядкувати атрибути в тому порядку, який використовувався при тренуванні моделей, використовуючи порядок, визначений у pickle-файлі cols_order.pkl.

Останній етап полягає у обробці даних та прогнозуванні потенційного відтоку за допомогою імпортованого "пайплайну" та виведенні результату. Цей процес, важливий для прогнозування відтоку, детально представлений на рисунку 3.18. Такий підхід забезпечує гнучкість та ефективність у роботі мікросервісу, дозволяючи йому ефективно використовувати аналітичні моделі для прийняття обґрунтованих рішень.

Для нашого мікросервісу основою входу є .json об'єкт, який має включати унікальний індекс для кожного набору даних. Це особливо важливо у випадку передачі декількох об'єктів одночасно. Кожен такий індекс містить масив у форматі словника (dictionary), що зберігає атрибути, необхідні для процесу прогнозування. Приклад формату вхідних даних можна знайти на рисунку 3.19.

```python
import json
import pandas as pd
import numpy as np
from flask import Flask, jsonify, request
from sklearn.externals import joblib

app = Flask(__name__)
clf = joblib.load('model.pkl')
dtypes = joblib.load('dtypes.pkl')
cols_order = joblib.load('cols_order.pkl')

@app.route('/predict', methods=['POST'])
def predict():

    json_ = request.json
    df_input = pd.DataFrame.from_dict(json_, orient='index')
    df_input = df_input[cols_order]
    df_input = df_input.astype(dtypes)
    df_input = df_input.replace(" ", np.nan)


    predictions = clf.predict_proba(df_input)[:, 1]
    idx = df_input.index
    df_output = {'index': idx, 'probability_to_churn': predictions}


    return jsonify(pd.DataFrame(df_output).to_dict(orient='records'))

if __name__ == '__main__':
    app.run(port=8080, debug=True)
```

Рисунок 3.18 – API для автоматичного прогнозування за допомогою Flask


```
{2: {'gender': 'Male',
  'SeniorCitizen': '0',
  'Partner': 'No',
  'Dependents': 'No',
  'tenure': '2',
  'PhoneService': 'Yes',
  'MultipleLines': 'No',
  'InternetService': 'DSL',
  'OnlineSecurity': 'Yes',
  'OnlineBackup': 'Yes',
  'DeviceProtection': 'No',
  'TechSupport': 'No',
  'StreamingTV': 'No',
  'StreamingMovies': 'No',
  'Contract': 'Month-to-month',
  'PaperlessBilling': 'Yes',
  'PaymentMethod': 'Mailed check',
  'MonthlyCharges': '53.85',
  'TotalCharges': '108.15'}}
```


Рисунок 3.19 – Відповідь у форматі JSON від мікросервісу

Тестування мікросервісу є наступним кроком після його реалізації. Для цього ми маємо запустити мікросервіс, а потім виконати POST-запит з відповідними

вхідними даними. Це дозволить перевірити функціональність та відповідність мікросервісу встановленим вимогам.

Код, що відповідає за реалізацію API, знаходиться у файлі app.py. Для ініціації та запуску цього коду використовується команда 'python app.py', введена в командному рядку терміналу. Цей процес запуску є стандартною процедурою в роботі з Python-додатками, забезпечуючи легкість та ефективність у розгортанні та тестуванні веб-додатків. Зразок ініціалізації коду показано на рисунку 3.20.

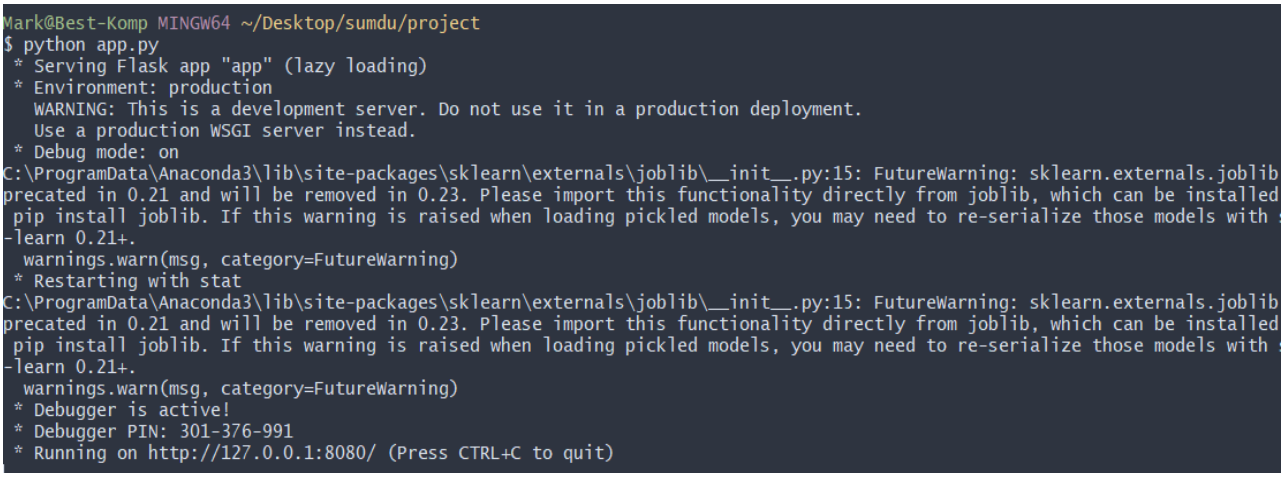



Рисунок 3.20 – Запуск мікросервісу

Оскільки наш мікросервіс був запущений локально, він доступний за адресою localhost, конкретно http://127.0.0.1:8080/. Використання порту 8080 є стандартною практикою для таких сервісів. Для перевірки роботи мікросервісу ми здійснили тестовий POST-запит з вхідними даними, що відображені на рисунку 3.19, щоб переконатися у коректності його функціонування. Зразок коду для тестового запиту представлений на рисунку 3.21.

```
import requests
import json
```

```
url = 'http://127.0.0.1:8080'
route = '/predict'
```

```
response = requests.post(url+route, json = data2)
print(r.status_code)
```

```
200
```

```
print(json.loads(r.content))
```

```
[{'index': '2', 'probability_to_churn': 0.678650531321351}]
```

Рисунок 3.21 – Запит до мікросервісу

З аналізу отриманих результатів, які відображені на рисунку 3.21, можна зробити висновок про успішність тестування. Це підтверджується також і в таблиці 3.2, де наведено детальний опис test-case для перевірки мікросервісу. Такий підхід дозволяє не лише перевірити функціональність мікросервісу, але й забезпечує упевненість у його надійності та стабільності, особливо важливо це для локально запущених сервісів, що працюють в тестовому режимі.

**Таблиця 3.2 – Базове тестування сервісу**

| Дія | Очікуваний результат | Результати тестування |
|---|---|---|
| Ініціалізація мікросервісу app.py | App.py повинен бути успішно запущено на локальному ПК за адресою http:///127.01.01:8080/ | passed |
| POST-запит з вихідними даними до точки доступу http:///127.01.01:8080/predict | Відповідь від мікросервісу має містити 200-ий код; Відповідь повинна містити json об’єкт з двома клюми – індекс та ймовірність | passed |

# ВИСНОВКИ

Телекомунікаційний сектор України демонструє вражаючі темпи росту, стаючи однією з найдинамічніших галузей країни. У цьому секторі спостерігається сильна конкуренція серед провайдерів, що призводить до того, що споживачі часто міняють своїх постачальників послуг, стимулюючи компанії до розробки стратегій для мінімізації потенційних збитків. Важливим аспектом у цьому контексті є прогнозування відтоку клієнтів, що дозволяє оперативно реагувати на можливу втрату клієнтів.

Зосередження уваги на стратегіях утримання існуючих клієнтів набуває все більшого значення для телекомунікаційних компаній. Відомо, що залучення нових клієнтів вимагає значно більших фінансових витрат, ніж збереження лояльності поточних. У такому конкурентному ринку, як телекомунікації, зниження відтоку клієнтів може стати ключовим фактором зростання прибутків компанії.

Ця дослідницька робота зосереджена на аналізі відтоку клієнтів на основі даних телекомунікаційних операторів, висвітлюючи актуальність та значимість цієї проблеми для галузі. Розв’язання цього завдання здійснюється за допомогою програмування на Python, мови, яка ефективно використовується для подібних аналітичних завдань.

У кваліфікаційній роботі реалізовані такі задачі:

1) досліджена предметна область;

2) проаналізовані аналоги;

3) розроблена модель машинного навчання;

4) підготовлені та очищені історичні дані;

5) розроблений мікросервіс для інтеграції розробленої моделі.

У дослідженні була обрана методологія CRISP-DM, яка включає в себе етапи обробки та аналізу даних, при цьому результати кожного етапу можуть використовуватися окремо для різних цілей. Паралельно з розробкою прогнозуючої

моделі була розроблена архітектура прототипу мікросервісу, який служить для ефективної роботи з даними, інтегрованими з різних баз даних, підвищуючи таким чином ефективність і точність бізнес-процесів у телекомунікаційній галузі.

Цей дослідницький підхід є не лише внеском у розвиток телекомунікаційного сектору України, але й важливим кроком у напрямку оптимізації стратегій утримання клієнтів в умовах динамічного ринкового середовища.

Основні результати роботи оприлюднені та обговорені на міжнародній науково-технічній конференції студентів та молодих вчених «Інформатика, математика, автоматика» (ІМА – 2020) (Суми, 2020 р.).

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. The Next Decade of Telecommunications Artificial Intelligence / Ye Ouyang, Lilei Wang, Aidong Yang / 2023 / pp. 1-11
2. Strategic Customer Service: Managing the Customer Experience to Increase Positive Word of Mouth, Build Loyalty, and Maximize Profits / 2019 / by John A. Goodman (2019) / pp. 132-140
3. Digital Customer Service: Transforming Customer Experience for an On-Screen World / 2021 / Rick Delisi / pp. 215-223
4. Creating Customer Loyalty: Build Lasting Loyalty Using Customer Experience Management / Kogan Page, Chris Duffy / 2020 / pp. 76-84
5. "Keep Your Customers: How to Stop Customer Turnover, Improve Retention and Get Lucrative, Long-Term Loyalty / Ali Cudby / 2021 / pp. 192-200
6. "Customer Relationship Management: Concepts and Technologies" / Francis Buttle, Stan Maklan / 2019 / pp. 104-112
7. A Greedy Approach for Offering to Telecom Subscribers / Piyush Kanti Bhunre, Tanmay Sen, Arijit Sarkar / 2023 / pp.30-36
8. "Churn Management in Telecommunications: Why Customer Retention Is Not Enough" / L. Caldara, D. Raffo / 2022 / pp. 143-152
9. "Machine Learning for Churn Prediction: An Application Guide to Predicting Customer Churn using Machine Learning" / A. Jaokar / 2023 / pp. 431-440
10. "Customer and Business Analytics: Applied Data Mining for Business Decision Making Using R" / D. S. Putler, R. E. Krider / 2021 / pp. 198-206
11. Customer churn prediction in telecom using machine learning and social network analysis in big data platform / Abdelrahim Kasem Ahmad, Assef Jafar, Kadan Aljoumaa / 2020 / pp. 57-88
12. A churn prediction dataset from the telecom sector: a new benchmark for uplift modeling / Théo Verhelst, Denis Mercier, Jeevan Shrestha, Gianluca Bontempi / 2023 / pp.1-8

13. "Pattern Recognition and Machine Learning" / C. M. Bishop / (2019) / pp. 72-80
14. Python for Machine Learning. 3rd Edition / S. Raschka, V. Mirjalili / (2021) / pp. 241-251
15. Probabilistic Machine Learning: Advanced Topics / K. P. Murphy / 2023 / pp. 123-131
16. Clifton Phua, Damminda Alahakoon, and Vincent Lee, “Minority Report in Fraud Detection: Classification of Skewed Data,” SIGKDD Explor Newsl, vol. 6, no. 1, pp. 50–59, Jun. 2019.
17. Over-Sampling Method in Imbalanced Data Sets Learning, in Advances in Intelligent Computing, 2021, pp. 878–887.
18. K. Coussement and D. Van den Poel, “Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques,” Expert Syst. Appl., vol. 34, no. 1, pp. 313–327, 2022.
19. "Introduction to Machine Learning with Python: A Guide for Data Scientists" / Andreas C. Müller, Sarah Guido / (2019) / pp. 102-110
20. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 3rd Edition / Aurélien Géron / (2022) / pp. 215-223
21. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, 3rd Edition 3rd ed. Edition / Sebastian Raschka, Vahid Mirjalili / (2021) / pp. 321-329
22. "Applied Machine Learning: Algorithms and Case Studies" / Kelleher, John D., Mac Namee, Brian, D'Arcy, Aoife / (2018) / pp. 442-450
23. "Building Machine Learning Powered Applications: Going from Idea to Product" / Emmanuel Ameisen / (2020) / pp. 591-599
24. Python Data Science Handbook: Essential Tools for Working with Data 1st Edition / Jake VanderPlasJake VanderPlas / 2021 / pp. 176-184
25. Flask Web Development: Developing Web Applications with Python / Miguel Grinberg / 2022 / pp. 127-136

26. Learning Test-Driven Development: A Polyglot Guide to Writing Uncluttered Code. 1st Edition / Saleem Siddiqui / 2022 / pp. 222-235
27. Стакан М.А. Інтелектуальна інформаційна технологія для прогнозування відключення від послуг мобільних операторів: робота на здобуття кваліфікаційного рівня бакалавр: спец. 122 - комп’ютерні науки, освітньо-професійна програма «Інформаційні технології проектування» / наук. кер. О.А. Шовкопляс. Суми: Сумський державний університет, 2020. 85 с.

# ДОДАТОК А

## Код для обробки вихідних даних

```
import numpy as np
import pandas as pd
import itertools
import warnings
warnings.filterwarnings("ignore")

from sklearn.preprocessing import StandardScaler

data = pd.read_csv("./raw_data/churn_sample.csv")

Customer_idcol = ['customerID']
target = ["Churn"]
categorical_cols   = data.nunique()[data.nunique() < 6].keys().tolist()
cat_ categorical = [x for x in cat_cols if x not in t target arget_col]
num_cols    = [x for x in data.columns if x not in cat_ cols    + target
_col + Id_col]
binary_cols   = data.nunique()[data.nunique() == 2].keys().tolist()
multi_cols = [i for i in cat categorical_cols if i not in binary_cols]

 le = LabelEncoder()
for i in bin_cols :
    telcom[i] = le.fit_transform(data[i])

#Duplicating columns for multi value columns
telcom = pd.get_dummies(data = data,columns = multi_cols )

#Scaling Numerical columns
std = StandardScaler()
scaled = std.fit_transform(data[num_cols])
scaled = pd.DataFrame(scaled,columns=num_cols)

telcom = data.drop(columns = num_cols,axis = 1)
 telcom = data.merge(scaled,left_index=True,right_index=True,how = "left")
```

# Код для порівняння моделей

```
from sklearn.model_selection import train_test_split, GridSearchCV, StratifiedShuffleSplit, cross_val_score
        from sklearn.linear_model import LogisticRegression
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.metrics import confusion_matrix, accuracy_score, classification_report from sklearn.metrics
import roc_auc_score,roc_curve, f1_score

        features = data.drop(['customerID','Churn'],axis = 1).columns
        X = data[features]
        y = data['Churn']
        train_test_split(X ,y, test_size = .25 ,random_state = 111,stratify = y)
        def validate(estimator,X_train,y_train,X_test, y_test):
            model = estimator
            model.fit(X_train, y_train)
            y_hat_proba = model.predict_proba(X_test)[:,1] > 0.5
            y_hat_train = model.predict(X_train)
                          accuracy_score_train,accuracy_score_test    =    accuracy_score(y_train,y_hat_train),
accuracy_score(y_test, y_hat_proba)
            roc_auc_train,roc_auc_test = roc_auc_score(y_train,y_hat_train), roc_auc_score(y_test, y_hat_proba)
            print("---------------------- TRAINING REPORT ------------------------\n")
                        print(f'Accuracy    score:    train    sample={accuracy_score_train:.3f}    vs    test
sample={accuracy_score_test:.3f}')
            print(f'ROC_AUC score: train sample={roc_auc_train:.3f} vs test sample={roc_auc_test:.3f}')
            print("\n=================== Classification report ====================")
            print(classification_report(y_test,y_hat_proba))
            print("\n=================== Confusion matrix ==========================")
            print(confusion_matrix(y_test,y_hat_proba))
        validate(LogisticRegression(class_weight='balanced'),X_train,y_train,X_test,y_test)

        validate(DecisionTreeClassifier(class_weight='balanced'),X_train,y_train,X_test,y_test)

        validate(RandomForestClassifier(n_estimators              =              500,              max_depth=5,
class_weight='balanced'),X_train,y_train,X_test,y_test)

500, max_depth=5, class_weight='balanced'),X_train,y_train,X_test,y_test)
```

# Код пошуку найкращих параметрів моделі

```
import pandas as pd
import numpy as np
import sys
from transformers import *
from sklearn.base import BaseEstimator, TransformerMixin
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline,make_pipeline, FeatureUnion
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split,RandomizedSearchCV,StratifiedKFold
from sklearn.metrics import accuracy_score,roc_auc_score,classification_report,confusion_matrix
from sklearn.externals import joblib

df = pd.read_csv("./raw_data/churn_sample.csv")

cat_features = ['gender', 'SeniorCitizen', 'Partner', 'Dependents',
     'PhoneService', 'MultipleLines', 'InternetService',
     'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
     'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling','PaymentMethod']
num_features = ['tenure','MonthlyCharges','TotalCharges'
]
X = df.drop(['customerID','Churn'],1)
y = df['Churn'].map({'Yes':1,'No':0})

pipe = Pipeline([("features", FeatureUnion([('categorical', make_pipeline(VariableSelector(names =
cat_features), MultiColumnLabelEncoder())),
('numeric',make_pipeline(VariableSelector(names=num_features),SimpleImputer(),StandardScaler()))])])
X_t = pipe.fit(X).transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_t ,y, test_size = .25 ,random_state = 111,stratify
= y)
cv = StratifiedKFold(n_splits=3,shuffle=True,random_state=111)
params = {
    'n_estimators':[100,250,500],
    'max_depth':[2,5,7,10,12],
    'max_features':['auto','sqrt'],
    'min_samples_split':[2,5,10],
    'min_samples_leaf':[1,2,5,10,12],
    'class_weight':['balanced',None]
}
grid=RandomizedSearchCV(RandomForestClassifier(),params,scoring='f1',cv=cv,
verbose=1).fit(X_train,y_train)
pipe=Pipeline([("features",FeatureUnion([('categorical',make_pipeline(VariableSelector(names
=cat_features),MultiColumnLabelEncoder())),('numeric',make_pipeline(VariableSelector(names
=num_features),SimpleImputer(),StandardScaler()))])),('prediction', grid.best_estimator_)])
pipe.fit(X, y)
joblib.dump(pipe,'./model/model.pkl')
```

## Код мікросервісу

```
import json
import pandas as pd
import numpy as np
from flask import Flask, jsonify, request
from sklearn.externals import joblib

app = Flask(__name__)
clf = joblib.load('./model/model.pkl')
dtypes = joblib.load('./model/dtypes.pkl')
cols_order = joblib.load('./model/cols_order.pkl')

@app.route('/predict', methods=['POST'])
def predict():

    json_ = request.json
    df_input = pd.DataFrame.from_dict(json_, orient='index')
    df_input = df_input[cols_order]
    df_input = df_input.astype(dtypes)
    df_input = df_input.replace(" ", np.nan)


    predictions = clf.predict_proba(df_input)[:, 1]
    idx = df_input.index
    df_output = {'index': idx, 'probability_to_churn': predictions}


    return jsonify(pd.DataFrame(df_output).to_dict(orient='records'))

if __name__ == '__main__':
    app.run(port=8080, debug=True)
```

## Код для тестування осервісу

```
      import requests
      import json

url = 'http://127.0.0.1:8080'
      route = '/predict'
      data = "{2: {'gender': 'Male',
        'SeniorCitizen': '0',
        'Partner': 'No',
        'Dependents': 'No',
        'tenure': '2',
        'PhoneService': 'Yes',
        'MultipleLines': 'No',
        'InternetService': 'DSL',
        'OnlineSecurity': 'Yes',
        'OnlineBackup': 'Yes',
        'DeviceProtection': 'No',
        'TechSupport': 'No',
        'StreamingTV': 'No',
        'StreamingMovies': 'No',
        'Contract': 'Month-to-month',
        'PaperlessBilling': 'Yes',
        'PaymentMethod': 'Mailed check',
        'MonthlyCharges': '53.85',
        'TotalCharges': '108.15'}}"

      response = requests.post(url+route, json = data2)
      print(response.status_code)
      #  200

print(json.loads(response.content))
      # [{'index': '2', 'probability_to_churn': 0.678650531321351}]
```