МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ ФОРМ НАВЧАННЯ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Інтелектуальна інформаційна технологія для прогнозування відключення від послуг мобільних операторів»

за спеціальністю 122 «Комп'ютерні науки та інформаційні технології»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТдн-51с Стакан Марк Андрійович

Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою _____«__» _____2020 р.

Науковий керівник _____ к.ф.-м.н., Шовкопляс О.А.
(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії _____ Шифрін Д. М.
(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2020

Сумський державний університет
Центр заочної, дистанційної та вечірньої форм навчання
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

ЗАТВЕРДЖУЮ
Зав. секцією ІТП
_____В. В. Шендрик
«_____»_____2020 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Стакан Марк Андрійович

1 Тема роботи *Інтелектуальна інформаційна технологія для прогнозування відключення від послуг мобільних операторів*

керівник роботи *Шовкопляс Оксана Анатоліївна, к.ф.-м.н., ст.викладач*

затверджені наказом по університету від « 21 » травня 2020 р. № 0608-III

2 Строк подання студентом роботи «6» червня 2020 р.

3 Вхідні дані до роботи технічне завдання на розробку мікросервісу; навчальна вибірка анонімізованих даних клієнтів телекомунікаційної компанії

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, проєктування та вибір методів реалізації, розробка інформаційної системи для прогнозування відключень від мобільних операторів.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) постановка задачі, аналіз предметної області, масштабування проблеми, методологія побудови системи, процес розробки моделі, методи прогнозування, приклад використання системи

6 Консультанти розділів роботи:

Розділ	**Консультант**	**Підпис, дата**	
		Завдання видав	**Завдання прийняв**

7 Дата видачі завдання_____01.10.2019_____

КАЛЕНДАРНИЙ ПЛАН

Порядковий номер	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Знаходження даних для побудування системи: дані з характеристиками абонентів	01.10.2019–30.10.2019	
2	Обробка даних: обробка та усунення непотрібних даних для навчальної вибірки	01.11.2019–25.11.2019	
3	Програмування на Python: будування та валідація предиктивної моделі	26.11.2019–26.01.2020	
4	Програмування на Python: створення web-додатка на базі Flask	27.01.2020–12.03.2020	
5	Завершення роботи: проведення стилістичних виправлень додатка, тестування реалізованого функціоналу	13.03.2020–10.04.2020	
6	Здача пояснювальної записки та файлів розробленого проєкту	11.04.2020–31.05.2020	

Студент _____ Стакан М.А.

(підпис)

Керівник роботи _____ к.ф.-м.н., Шовкопляс О.А.

(підпис)

РЕФЕРАТ

Тема роботи «Інтелектуальна інформаційна технологія для прогнозування відключення від послуг мобільних операторів».

Пояснювальна записка складається із вступу, трьох основних розділів, висновку, списку використаних джерел із 20 найменувань та додатку. Загальний обсяг пояснювальної записки складає 57 сторінок, у тому числі 49 сторінок основного тексту, 2 сторінки списку використаних джерел, 6 сторінок додатку.

У першому розділі проаналізована предметна область, виявлена актуальність проблеми та її економічне обґрунтування. Також проведено огляд останніх досліджень за темою роботи та проведено пошук аналогів інтерактивного додатка, проаналізовані існуючі підходи та засоби реалізації.

У другому розділі описані мета та задачі проекту, методи і засоби для розробки моделі та обрана методологія для вирішення проблеми. Також були обрані методи прогнозування та проаналізовано їх недоліки та переваги.

У третьому розділі детально описано процес первинного візуального аналізу даних, практична реалізація обробки вихідних даних, побудови моделей машинного навчання та підбір найкращих гіперпараметрів моделі. Реалізовано REST API на базі веб-фреймворку Flask для автоматизації прогнозування нових об’єктів.

Ключові слова: ІНТЕЛЕКТУАЛЬНА СИСТЕМА, АНАЛІЗ ДАНИХ, МАШИННЕ НАВЧАННЯ, CHURN PREDICTION, АВТОМАТИЗАЦІЯ, FLASK, REST API.

Зміст

ВСТУП

Телекомунікаційний сектор став однією із основних галузей промисловості розвинених країн. Щороку провайдери телекомунікаційних послуг зазнають збитків через відтік абонентів. Річний відтік клієнтів телекомунікаційних компаній в середньому становить 25%, що немало.

Технічний прогрес і зростаюча кількість операторів підвищили рівень конкуренції. Компанії наполегливо працюють над тим, щоб вижити на цьому конкурентному ринку залежно від декількох стратегій. Для отримання більших доходів запропоновані основні стратегії: (1) придбати нових клієнтів, (2) продати існуючих клієнтів та (3) збільшити термін утримання клієнтів. Однак порівняння цих стратегій з урахуванням вартості рентабельності інвестицій кожної з них показало, що третя стратегія є найбільш профільною стратегією. Це свідчить про те, що утримання існуючого замовника послуг коштує набагато нижче, ніж залучення нового.

Для застосування третьої стратегії компаніям потрібно впливати на таке явище, відоме як «переміщення клієнта від одного постачальника до іншого». Відтік клієнтів викликає значну стурбованість у секторах послуг з високою конкуренцією. З іншого боку, прогнозування клієнтів, які, ймовірно, покинуть компанію, представлятиме потенційно велике додаткове джерело доходу, якщо це буде зроблено на ранній фазі.

Багато досліджень підтвердили, що методи машинного навчання та аналізу даних дуже ефективні для прогнозування цієї ситуації. Ця методика застосовується за допомогою методів виявлення патернів або причинно-наслідкових зв’язків у поведінці абонентів на основі історичних даних [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

На перший погляд, відтік не повинен викликати серйозну стурбованість, але чи насправді інструменти управління настільки неадекватні, а відповідні дані настільки складні для використання – на ці питання можуть дати відповідь сучасні методи аналізу даних.

Більшість клієнтів користуються кількома послугами одного провайдера, наприклад, мобільний зв'язок, інтернет і телебачення. Коли клієнт скасовує одну послугу, він також відмовляється від іншої частини послуг. Погані і хороші відгуки в соціальних мережах розходяться дуже швидко. Від друзів і знайомих, яким клієнт особисто розповідав про свої проблеми і скарги, інформація передається сотням або тисячам інших користувачів, які читають публікації незадоволених клієнтів в Instagram, Twitter, Facebook або в інших популярних соціальних мережах. Негативні відгуки роблять сильний вплив на багатьох інших клієнтів компанії.

Коли один клієнт переходить до іншого провайдера, багато клієнтів зі схожими проблемами під впливом рекомендації масово приймають рішення наслідувати його приклад [2]. Абоненти, які відмовляються від послуг оператора, під впливом сильних негативних емоцій, зазвичай охоче розповідають про причини переходу і це найважливіша інформація, яка не повинна бути випущена з уваги. На мою думку більш відкритого і щирого порадника, ніж роздратований користувач послуг вам не знайти.

Однак, таке явище, як відтік клієнтів, має й позитивний вплив. У момент, коли клієнт розриває договір, провайдер повинен записати причину відходу [3]. Незважаючи на те, що причини, іноді, можуть здаватися нераціональними, вони часто виявляють слабкості мережі і вразливість для конкурентів.

Фактично телекомунікаційні компанії, у даному випадку – оператори мобільного зв'язку, ведуть маркетингову боротьбу буквально за кожного клієнта і у цьому випадку перемагає той, хто краще аналізує дані. При правильному підході, у компанії є дані про клієнтів, які на момент аналізу вже відмовилися від послуг. Успішні провайдери знають, що попередження відтоку починається із ефективного аналізу і візуалізації та залученням IT-фахівців, якщо це необхідно [1]. Провайдери з кращими інструментами аналізу даних, які допомагають бізнес-користувачам діяти активніше, першими помічають неминучий відтік і вчасно мотивують клієнтів залишитися. Також, оперативно виявляються проблеми мережі, які стали причиною відтоку, і приймаються відповідні заходи, щоб уникнути його в майбутньому.

Предиктивна аналітика допомагає знаходити приховані причини втрати клієнтів. Навіть найпроникливіші аналітики з повними базами даних, але без ефективних інструментів прогнозування, не зможуть побачити картину цілком [2]. Предиктивна аналітика вказує на, здавалося б, випадкових клієнтів із випадковими проблемами і допомагає уникнути відтоку до того, як клієнт починає замислюватися про зміну оператора.

1.1 Дослідження актуальності проблеми

Для бізнесу, який надає телекомунікаційні послуги, важлива база клієнтів: її накопичення новими, та утримання «старих» абонентів. Кожної хвилини у цю базу приходять нові клієнти, наприклад, дізнавшись про продукт або послугу з реклами, якийсь час живуть (активно користуються продуктами) і через якийсь час перестають користуватися [2]. Такий період називають «Життєвий цикл клієнта» (англ. Customer Lifecycle) – це термін, що описує

етапи, які проходить клієнт, коли дізнається про продукт, приймає рішення про покупку, платить, використовує і стає лояльним споживачем, і в кінцевому рахунку, перестає користуватися з тих чи інших причин продуктом або послугами. Відповідно відтік – це завершальна стадія життєвого циклу клієнта, коли клієнт перестає користуватися послугами, а для бізнесу це означає, що клієнт перестав приносити прибуток і взагалі будь-яку користь.

Кожен клієнт телекомунікаційних послуг – це фізичні особи або група осіб, які вибирають той чи інший тариф або послугу спеціально під свої потреби [1]. Наприклад, якщо абонент часто подорожує – йому необхідно обрати вигідні умови роумінгу. Однією із основних потреб абонента – вигідні тарифи для дзвінків в межах України, будь-то нічні чи денні дзвінки. Також, якщо абонент має досить велику сім'ю, йому потрібно обрати відповідний сімейний тариф. Тому нам, як користувачам послуг, потрібно вибрати вигідні для нас тарифи. Загалом, змінних тут вистачає.

Також, досить великий вплив на вибір оператора послуг має фактичне місцезнаходження абонента: який сенс обирати того оператора, якщо якість надання послуг одного найкраща у Києві, а абонент фактично живе у Сумах - нехай умови тарифу такого оператора будуть хоч удвічі вигідніше, – наявність якісного надання послуг поблизу все ж буде важливим критерієм. Сюди ж можна віднести і наявність фізичних центрів обслуговування абонентів – якась частина населення або вікова група більше довіряє фізичному центру, а не додатку в смартфоні, – це теж треба враховувати [2].

Як наслідок, причин для відмови від послуг оператора (або від самого оператора) у людини може бути безліч. Наприклад, зміна роботи, де потрібно частіше користуватися послугами оператора – тариф змінився з побутового на «підвищений», який менш вигідний; переїзд в інше місто, де немає відділень або неякісний зв'язок, або за якихось причин абоненту не сподобалося спілкування з некваліфікованим операціоністом.

Проблема відтоку клієнтів є актуальною у будь-якій галузі надання телекомунікаційних, фінансових та побутових послуг – телеком, інтернет-провайдери, страхові компанії, банківський сектор, – тобто скрізь, де є клієнтська база і періодичні транзакції.

1.2 Аналіз аналогів

Сьогодні застосовується багато підходів для прогнозування відтоку абонента телекомунікаційних компаній. Більшість із цих підходів використовують машинне навчання та аналіз даних. Більшість пов’язаних робіт зосереджено на застосуванні лише одного методу обробки даних та предикативної моделі, а інші зосередили увагу на порівнянні кількох стратегій для прогнозування відтоку.

Гаврил Тодереан, доктор технічних наук у Румунському технічному університеті Клуж-Напоки, запропонував вдосконалену методологію аналізу даних для прогнозування відтоку абонентів для передплачених клієнтів, в якій використовувався набір даних про дзвінки 3333 клієнтів із 21 атрибутами (змінними), і залежний параметр, або змінна інтересу, із двома значеннями: Так / Ні, де Так означало, що абонент відмовився від послуг. Деякі атрибути включають інформацію про кількість вихідних та вихідних повідомлень та голосову пошту для кожного абонента [4]. Автор застосував алгоритм аналізу основних компонентів «PCA» для зменшення розмірності даних. Для прогнозування коефіцієнта відтоку абонентів використовувались алгоритми машинного, такі як нейронні мережі, метод опорних векторів (SVM) та Байєсівський «наївний» класифікатор. Автор використав AUC для вимірювання продуктивності алгоритмів. Гаврил запропонував модель прогнозування на основі алгоритму нейронної мережі щоб вирішити проблему

відтоку клієнтів у великій китайській телекомунікаційній компанії, яка містить близько 5,23 млн клієнтів. Метрикою оцінювання моделі був загальний коефіцієнт точності, який досяг 91,1%.

У свою чергу, Ідріс запропонував підхід, заснований на генетичному програмуванні з AdaBoost для вирішення проблеми відтоку абонентів в телекомунікаціях [5]. Модель тестували на двох стандартних наборах даних. Перший – Orange Telecom, а другий – Cell2cell, з точністю 89% для набору даних Cell2cell та 63% для другого.

Хуан та ін. вивчали проблему відтоку клієнтів на платформі великих даних [5]. Метою дослідників було довести, що великі дані значно покращують процес прогнозування відтоку залежно від обсягу, різноманітності та швидкості даних. Для цього дослідження використовувалися дані операційного відділу та департаменту підтримки бізнесу в найбільшій телекомунікаційній компанії Китаю, де використовується платформа обробки великих даних Hadoop. Був використаний та оцінений алгоритм випадкових лісів за допомогою метрики оцінювання AUC.

Різні дослідження вивчали проблему неврівноважених наборів даних, де пропорція клієнтів, які відмовились від послуг оператора була набагато менша, аніж пропорція активні класи клієнтів, оскільки це одна з основних проблем в задачах прогнозування відтоку [6–13].

Бурез та Ван ден Поель вивчали проблему неврівноваженості наборів даних у моделях прогнозування відтоку за допомогою таких алгоритмів як Gradient Boosting та метод «випадкових лісів», або Random Forest, де застосовувалися методи вибору випадкових вибірок та розширеної недостатньої вибірки для збільшення об’єму даних [7]. Метриками оцінювання були AUC та Lift-score. Результат показав, що використання методів збільшення набору даних дає значне покращення в точності прогнозування відтоку.

2 ПРОЄКТУВАННЯ ТА ВИБІР МЕТОДІВ РЕАЛІЗАЦІЇ

Для реалізації прототипу інтелектуальної системи прогнозування відтоку абонентів було використані табличні анонімізовані дані, надані компанією IBM, які знаходяться у відкритому доступі у мережі Інтернет [16].

Обрані дані зберігаються у форматі .csv, які містять репрезентативну вибірку абонентів, яка становить 7063 об’єктів та 21 атрибут для кожного з користувачів, а саме:

1. послуги, на які підписався кожен клієнт – телефон, кількість вихідних ліній, Інтернет, безпека в Інтернеті, резервне копіювання в Інтернеті, захист пристрою, технічна підтримка та трансляція телевізійних програм та кінофільмів;
2. інформація про обліковий запис клієнтів – як довго вони були клієнтами, контрактом, способом оплати, без паперовим рахунком, щомісячними платежами та загальними платежами;
3. демографічна інформація про клієнтів – стать, вік, а також чи вони мають партнерів або утриманців.

Назви використаних атрибутів, їх опис та типи даних наведені у таблиці 2.1:

Таблиця 2.1 – Опис даних

№	Назва атрибуту	Опис	Тип даних
1	CustomerId	Уникальный анонімізований ідентифікатор абонента	hash-string
2	Gender	Замовник чоловік або жінка	string
3	Senior Citizenship	Замовник є похилим	binary

		громадянином чи ні	
4	Partner	Замовника є партнер чи ні	binary
5	Dependents	Чи є у замовника утриманці чи ні	binary
6	Tenure	Кількість місяців перебування замовника з провайдером	integer
7	PhoneService	Чи має клієнт послугу сервісного обслуговування телефону чи ні	binary
8	Multiple Lines	Чи є у замовника кілька ліній чи ні	string
9	Internet Service	Інтернет-провайдер клієнта - (DSL, оптоволоконна, немає)	string
10	OnlineSecurity	Чи має клієнт послугу онлайн-безпеки в Інтернеті чи ні (Так, Ні, Немає Інтернет-послуги)	string
11	Online Backup	Чи має клієнт резервне копіювання в Інтернеті чи ні (Так, Ні, Немає Інтернет-послуги	string
12	Device Protection	Чи має клієнт послугу захисту пристрою від фізичних пошкоджень чи ні (так, ні, немає послуги Інтернету)	string
13	TechSupport	Замовник має технічну підтримку чи н	string

14	Streaming TV	Чи має клієнт потокове телебачення чи ні (так, ні, немає послуги Інтернету)	string
15	Streaming Movies	Чи дивиться клієнт потокові фільми чи ні (так, ні, немає послуги Інтернету)	string
16	Contract	Тип контракту замовника (місяць на місяць, один рік, два роки)	string
17	Paperless Billing	Чи клієнт отримує рахунки без паперовим шляхом на папері чи ні (так, ні)	binary
18	Payment Method	Спосіб оплати клієнта (електронний чек, чек по електронній пошті, банківський переказ (автоматичний), кредитна картка (автоматичний))	string
19	Monthly Charges	Сума, що стягується з замовника щомісяця	float
20	Total Charges	Загальна сума, яку заплатив замовник за всю історію співпраці з провайдером	float

Також, для кожного абонента є бінарна змінна інтересу, або незалежна змінна, яка приймає тільки два значення, де 1 означає, що абонент відмовився від послуг провайдера послуг через 1 місяць після отримання даних, 0 – абонент залишився з провайдером. Саме цей атрибут потрібно прогнозувати.

Отже, нашою задачею буде побудувати інтелектуальну систему-класифікатор для прогнозування відтоку абонента на основі даних про 7063

абонентів та 21 атрибутів, застосовуючи статистичні методи та методи машинного навчання. Метрикою оцінювання моделі буде так звана ROC-score, або метрика похибок, яка дозволяє оцінити здатність нашої системи розділяти абонентів, які відмовляються від послуг телекомунікаційної компанії, від тих хто залишиться з провайдером.

2.1 Мета та задачі

Завдання полягає у розробці програмного забезпечення, яке дозволяє зменшити кількість абонентів, які відмовляються від послуг телекомунікаційного оператора, тим самим збільшивши прибуток компанії. Завдання на розробку програмного забезпечення «Інтелектуальна інформаційна технологія для прогнозування відключення від послуг мобільних операторів» передбачає:

1. автоматизацію процесу обробки даних;
2. їх очищення від помилок та викидів (Data cleaning);
3. створення нових змінних на основі існуючих (Feature engineering);
4. пошук найбільш значущих змінних методом візуалізації (Exploratory data analysis);
5. обробки змінних (Preprocessing phase);
6. фаза моделювання та валідація моделі (Modeling phase);
7. деплоймент моделі за допомогою веб-додатку REST-API.

Усі вище описані етапи є частинами методології створення інформаційних систем машинного навчання, яка має назву CRISP-DM [15].

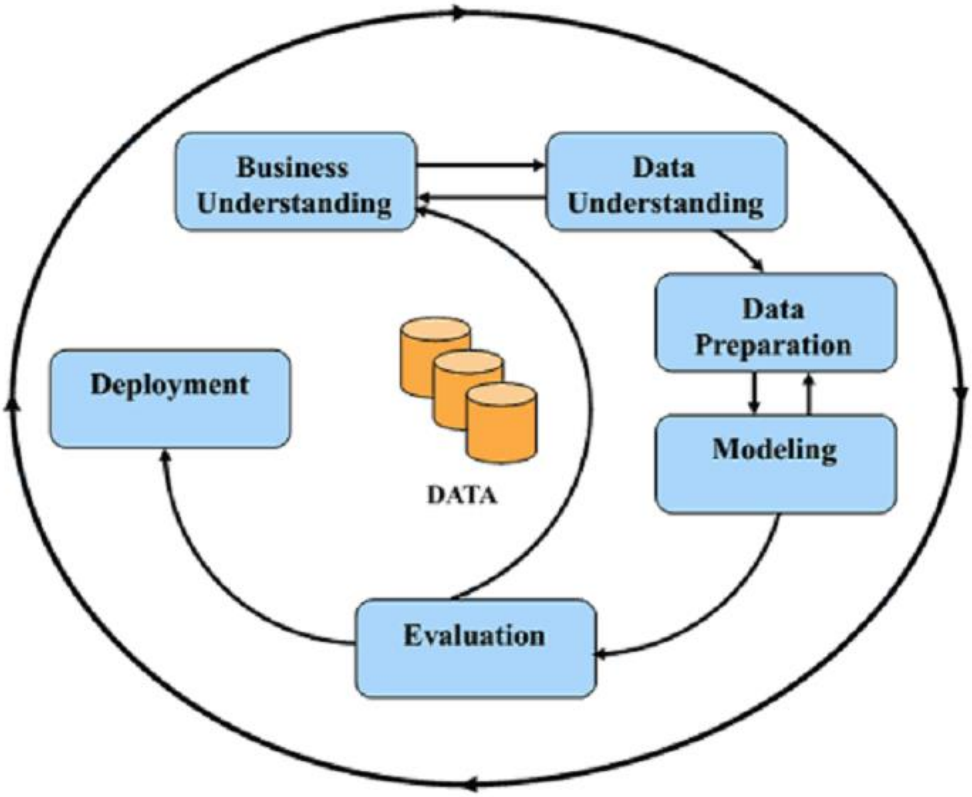


Рисунок 2.1 – діаграма процесу обробки даних CRISP-DM

CRISP-DM (Cross-Industry Standard Process for Data Mining) – міжгалузевий стандартний процес для дослідження даних, це перевірена в промисловості і найбільш поширена методологія по дослідженню даних [15].

Модель життєвого циклу дослідження даних складається з шести фаз, а стрілки позначають найбільш важливі і часті залежності між фазами [15]. Послідовність цих фаз строго не визначена. Як правило, в більшості проєктів доводиться повертатися до попередніх етапів, а потім знову рухатися вперед.

2.2 Вибір методів прогнозування віддтоку

До основних поставлених задач в розглянутій проблемі належить первинний аналіз даних та побудова прогнозуючої моделі. Існує два підходи до розв’язання цих задач.

Перший – побудова предиктивної моделі залежності чи відмовиться клієнт від послуг чи ні в залежності від набору атрибутів цього клієнта. Такий підхід вимагає дослідження предметної області та застосування отриманих результатів для створення моделі. На жаль, такий підхід вимагає значних зусиль та експертизи, а отриманий результат дослідження предметної області не можливо масштабувати для інших клієнтів.

У 20-му сторіччі були створені методи, які дозволяють розробити більш універсальний розв’язок цієї задачі, який до того ж не потребує значної експертизи у вивчення предметної області. З іншого боку, цей підхід вимагає наявності достатньої кількості даних, на основі яких ведеться пошук причинно-наслідкових зв’язків, реалізація та налаштування моделі для прогнозу. Однак, з появою інтернету, дані про клієнтів та їх поведінкові фактори почали зберігати та накопичувати. Тому наразі у великих компаній існують величезні масиви даних, за допомогою яких можна поліпшувати ефективність бізнесу та збільшувати доходи.

Такі методи носять назву машинного навчання. Машинне навчання – це розділ науки штучного інтелекту, що вмістить в собі аналіз даних. У даному випадку, об'єктом будемо називати те явище або сутність, для якого потрібно щось спрогнозувати. Відповіддю або цільовою змінною - будемо називати те, що саме потрібно спрогнозувати, тоді простір цих відповідей - це вектор всіх можливих відповідей, які необхідно аналізувати.

Об’єкти – це якась сутність реального світу, але для комп'ютера це не зрозуміло. Він не знає, що таке користувач або товар, йому потрібно

представити ці об'єкти за у виді чисел, які комп'ютер може сприймати. Ознака або атрибут– це характеристики об'єкта, а сукупність всіх d ознак називається вектором ознак цього об’єкта і з ним можна проводити векторні операції: додавати, множити на числа, тобто це елемент лінійної алгебри. Ознака може бути дійсним числом, рядком тексту, порядковим атрибутом і т. д., але все це те, що комп’ютер зможе зрозуміти.

Найбільш важливим поняттям машинного навчання є навчальна вибірка. Це та репрезентація об’єктів та їх ознак, на основі яких будується загальна закономірність. У даному випадку, ця проблема має назву навчання з учителем, тобто наша навчальна вибірка складається з пар об’єктів та їх ознак і відповідей. Окреме важливе питання – як саме зібрати навчальну вибірку, але це питання не розглядається в дипломній роботі, адже у нашому випадку розглянуто набір даних який знаходиться у відкритому доступі.

Нарешті, потрібно щось, що буде будувати припущення та допоможе розв’язувати нашу задачу. Це модель машинного навчання. По суті, це функція, яка відображає простір об'єктів у простір відповідей, приймає на вхід об'єкт x. Для тренування моделі якраз і використовується навчальну вибірку.

Враховуючи особливості поставленої задачі та наявність даних для реалізації, саме цей підхід – застосування методів машинного навчання – було обрано для вирішення задачі.

Оскільки поставленою задачею є прогнозування ймовірності відношення до одного з двох класів, то простір відповідей – множина всіх дійсних додатних чисел від 0 до 1. Задачу пошуку залежності цільової змінної від значень ознак називають задачею бінарної класифікації.

За останнє десятиліття проблема прогнозування відтоку має велику кількість прецедентів, де застосування моделей машинного навчання дозволяє ефективно поліпшувати проблему, що розглядається.

2.2.1 Лінійні моделі

Одним з найпростіших класів моделей машинного навчання є лінійні моделі. Ідея лінійних алгоритмів полягає у наступному: необхідно підібрати коефіцієнти для кожної ознаки таким чином, щоб похибка між прогнозованим та фактичним значеннями цільової змінної була мінімальною. Математично лінійний бінарний класифікатор в задачах, подібній до задачі дипломного проекту, виражається наступним чином:

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

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:

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

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

Однак існує проблема в інтерпретованості лінійних моделей. Такий тип методів не відображає особливості процесу прийняття рішень як це роблять люди. Насправді, коли людина хоче зрозуміти ту чи іншу проблему, вона буде задавати послідовність з простих питань, які в підсумку приведуть до якої-небудь відповіді.

2.2.2 Дерева ухвалення рішень

Дерева ухвалення рішень – це ще один тип методів машинного навчання який наразі грає важливу роль в машинному навчанні.

Дерево рішень – це найпотужніший і популярний інструмент для класифікації та прогнозування. Дерево рішення - це структура типу схожої з структурою фізичного дерева, де кожен внутрішній вузол позначає логічний тест на значення атрибуту, кожна гілка являє собою результат тесту, а кожен вузол листка (кінцевий вузол) містить клас об’єкта.

Дерево можна «навчати», розділивши набір даних на підмножини на основі тесту на значення атрибутів. Цей процес повторюється на кожній підмножині рекурсивно. Рекурсія завершується, коли всі підмножини у вузлі мають однакове значення цільової змінної або розділення підмножини не додають значення прогнозування. Побудова класифікатора дерева рішень не вимагає будь-яких доменних знань або налаштування параметрів, а тому є підходящим для розвідувального аналізу. Дерева рішень можуть обробляти дані високих розмірів. Загалом класифікатор дерева рішень має хорошу точність. Індукція дерева рішень - типовий індуктивний підхід для засвоєння знань щодо класифікації.

Дерева рішень у задачах класифікації та регресії вирощуються шляхом додавання вузлів запитань поступово, використовуючи приклади з навчальної вибірки для керівництва вибором питань. В ідеалі, одне просте запитання прекрасно розділило б приклади навчання на їх класи. Якщо не існує жодного

питання, яке б дало таке ідеальне розмежування, обирається питання, яке розділяє приклади максимально правильно.

Хороше запитання розділить колекцію предметів з неоднорідними мітками класів на підмножини з майже однорідними мітками, розшаровуючи дані так, що в кожному новому вузлі або ноді. Але як зрозуміти що задане питання або логічний тест є має добру розділяючу здатність між класами. Для дерев рішень два найпоширеніші методи – ентропія Шенона та індекс Джині.

Припустимо, необхідно вирішити задачу класифікації об'єктів на m класів, використовуючи набір навчальних предметів $\mathbb{E}$. Нехай p_i (i = 1,…, m) – множина елементів $\mathbb{E}$, що належать до класу i. Ентропія розподілу ймовірностей $(p_i)_{i=1}^{m}$ дає розумну міру впорядкованості об'єктів множини $\mathbb{E}$. Ентропія $-\sum_{i=1}^{m} p_i \, log(p_i)$ є найнижчою, коли один p_i дорівнює 1, а всі інші дорівнюють 0, а ентропія максимізована, коли всі p_i рівні.

Індекс Джині, інший загальний показник впорядкованості, який обчислюється за формулою

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

Знову-таки, нульове значення індексу отримано, коли множина $\mathbb{E}$ містить елементи лише одного класу.

Враховуючи міру впорядкованості $\mathcal{I}$ обирається питання, що мінімізує середньозважене середнє значення впорядкованості діючих вузлів. Тобто, якщо питання з k можливими відповідями розділяє $\mathbb{E}$ на підмножини $\mathbb{E},\mathbb{E}_1 \ldots,\mathbb{E}_k$ обирається питання для мінімізації

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

У багатьох випадках можна обрати найкраще питання, перерахувавши всі можливості. Якщо $\mathcal{I}$ - функція ентропії, то різниця між ентропією розподілу класів у початковому верхньому вузлі та цим середньозваженим середнім рівнем ентропії наступних нижчих вузлів називається посиленням інформації.

Інформаційний приріст, який виражається через розбіжність Куллбека-Лейблера, завжди має негативне значення.

Ми продовжуємо вибирати запитання рекурсивно, щоб розділити навчальні предмети на все менші підмножини, в результаті чого з'явиться дерево. Вирішальним аспектом застосування дерев рішень є обмеження складності або «глибини» дерева, щоб уникнути перенавчання. Один з прийомів – це припинити ділити гілки, коли жодне питання не збільшує чистоту підмножини більш ніж на зазначену кількість. Крім того, можна побудувати дерево, поки жоден листок не може бути подальше підрозділений. У цьому випадку, щоб уникнути перенавчання, необхідно обрізати дерево, видаливши вузли. Це може бути зроблено шляхом згортання внутрішніх вузлів у листя, якщо це знижує похибку класифікації на витриманому наборі навчальних прикладів [17].

Вирішальне дерево має наступні серйозні недоліки:

– алгоритм дуже чутливий до вхідних даних, тому здатен то перенавчання

– якість моделі дуже сильно змінюється при новій навчальній вибірці, тобто алгоритм веде себе нестабільно

– дерева ухвалення рішень не здатні до екстраполяції

2.2.3 Ансамбль моделей. Випадковий ліс (Random forest)

Хоча окремі дерева рішень можуть бути відмінними класифікаторами, підвищену точність часто можна досягти, поєднавши результати колекції дерев рішень. Ансамблі дерев рішень іноді входять до складу класифікаторів, які працюють найкраще [18]. Випадкові ліси та boosting – це дві стратегії поєднання дерев рішень.

У підході до випадкових лісів багато різних дерев рішень вирощуються за алгоритмом рандомізованої побудови дерев. Набір об’єктів за навчальної вибірки відбирається за випадкового відбору з заміщенням для отримання модифікованого навчальної вибірки, рівного розміру оригіналу, але з деякими

навчальними об’єктами, включених не один раз. Крім того, при виборі питання на кожному вузлі враховується лише невелика випадкова підмножина ознак. За допомогою цих двох модифікацій кожен запуск може призвести до дещо іншого дерева. Прогнози результуючого ансамблю дерев рішень поєднуються, приймаючи найпоширенішій прогнози з всіх випадкових дерев. Підтримання колекції хороших гіпотез, а не привласнення до одного дерева, зменшує ймовірність того, що новий об’єкт буде неправильно класифікований, якщо багатьом деревам буде присвоєно неправильний клас.

Boosting – це метод машинного навчання, який використовується для об'єднання декількох класифікаторів у сильніший класифікатор шляхом неодноразового перенавантаження ваги прикладів навчання, щоб зосередитись на найбільш проблемних [19]. На практиці прискорення часто застосовується для комбінування дерев рішень. Чергуючі дерева рішень – це узагальнення дерев рішень, що є результатом застосування варіанта прискорення для комбінування слабких класифікаторів на основі пнів, що є деревами рішень, що складаються з одного питання. У поперемінних деревах рішень рівні дерева чергуються між типовими вузлами запитання та вузлами, які містять ваги і мають довільну кількість вузлів нижчих слоїв. На відміну від стандартних дерев рішень, об’єкти можуть проходити декілька шляхів і присвоюються класам на основі ваг, з якими стикаються шляхи. Чергуючі дерева рішень можуть створювати менші та більш інтерпретовані класифікатори, ніж ті, що отримані від застосування прискорення безпосередньо до стандартних дерев рішень.

2.3 Вибір засобів реалізації

Проаналізувавши наукові статті передових спеціалістів з аналізу даних та машинного навчання, для побудування інтелектуальної системи прогнозування відтоку абонентів в роботі буде використана мова програмування Python версії 3.7 [14].

Python – інтерпретована об’єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією [6]. Розроблена в 1990 році Гвидо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та динамічним зв’язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання наявних компонентів [14]. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду.

За останні 10 років Python набрав чималої популярності серед розробників, та широко застосовується в розробках інтелектуальних систем різної складності в таких компаніях як Google, Facebook, Amazon, YouTube, Netflix та інших (рис. 2.2).

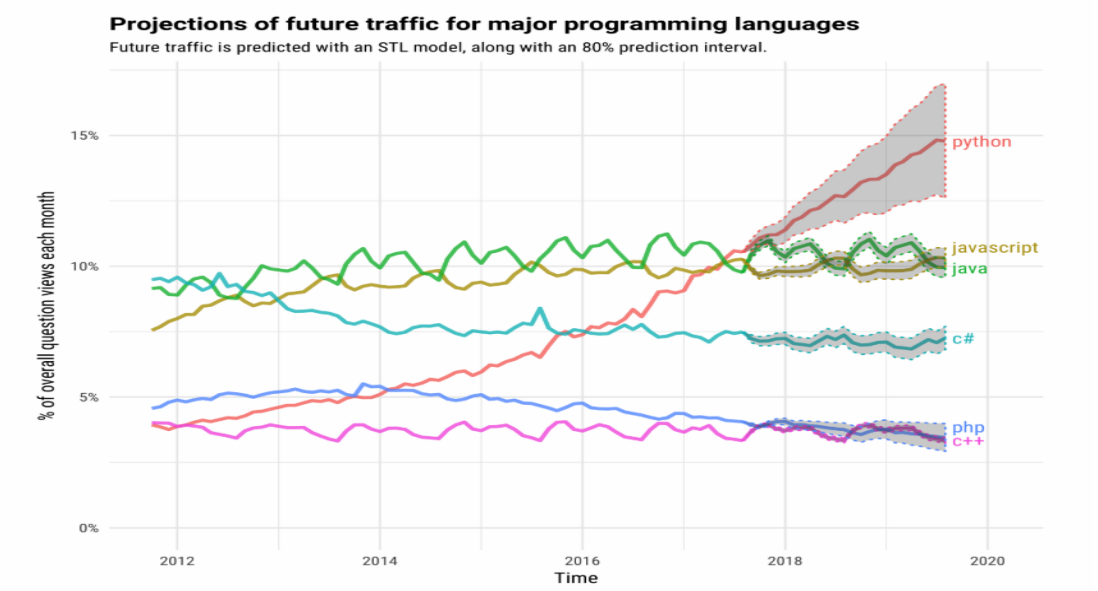


Рисунок 2.2 – Динаміка популярності Python

Обрана мова програмування містить ряд пакетів та модулів, які дозволяють використовувати функції лінійної алгебри, математичного аналізу, теорії ймовірностей та математичної статистики, що значно покращують швидкість обчислення на локальному ПК або віддаленому сервері. Для реалізації системи будуть використовуватися такі пакети в Python 3.7:

1. sklearn – для використання алгоритмів машинного навчання та їх оптимізації;
2. numpy – для швидких обчислень, використовуючи об'єкти та функції лінійної алгебри;
3. pandas – для обробки вихідних даних;
4. matplotlib/seaborn – для візуалізації даних та їх аналізу;
5. flask – для побудування REST-API для інтелектуальної системи та її хостингу на сервері.

Scikit-learn – це одна з найпопулярніших бібліотек для машинного навчання в Python. Бібліотека sklearn містить безліч ефективних інструментів для машинного навчання та статистичного моделювання, включаючи класифікацію, регресію, кластеризацію та зменшення розмірності. Sklearn використовується для побудови моделей машинного навчання. Він не повинен використовуватися для читання даних, маніпулювання та узагальнення їх.

Scikit-learn поставляється з великою кількістю функцій.

– Алгоритми машинного навчання з вчителем: починаючи від узагальнених лінійних моделей (наприклад, лінійна регресія), опорних векторних машин (SVM), дерев рішень до байєсівських методів – усі вони входять у інструментарій даної бібліотеки. Поширення алгоритмів машинного навчання є однією з головних причин високого використання наукової роботи.

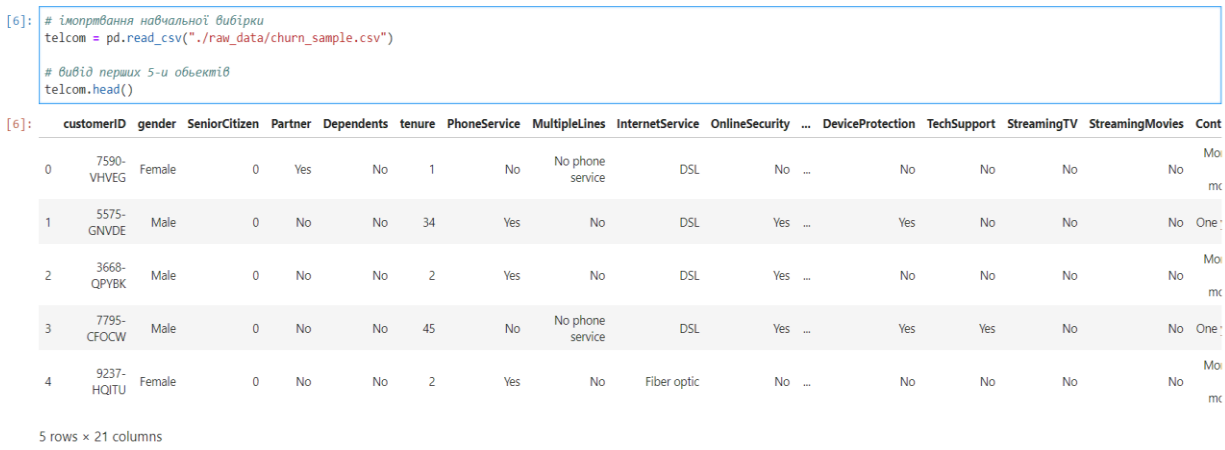
– Валідація моделі. До бібліотеки sklearn входять різні методи перевірки точності контрольованих моделей на невидимих даних.

– Алгоритми машинного навчання без учителя, до яких входять такі методи як кластеризація, факторний аналіз, аналіз основних компонентів до некерованих нейронних мереж.

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОГНОЗУВАННЯ ВІДКЛЮЧЕННЯ ВІД ПОСЛУГ МОБІЛЬНИХ ОПЕРАТОРІВ

3.1 Первинний огляд вихідних даних

У задачах прогнозування та аналізу даних дуже важливим є наявність репрезентативної вибірки об’єктів, їх атрибутів або якійсь додаткові дані, що були накоплені історично. Наявність будь-яких даних про прогнозуючий об’єкт формують «dataset», так звану навчальну вибірку, яку потрібно проаналізувати, і, якщо вибірка містить в собі цінні патерни, – використати її для побудування прогнозуючої моделі. На рисунку 3.1 показаний фрагмент навчальної вибірки.



```
[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 для інформаційної системи

Далі необхідно обстежити розмір навчальної вибірки, атрибути об’єктів, наявність пропущених значень. Для цього використаємо відповідні атрибути

об’єктів та методи бібліотеки pandas. Розмір даних та детальний опис атрибутів наведений на рисунку 3.2.

```
]: # 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 – Огляд вихідних даних

Після первинного огляду вихідних даних, можемо зробити такі висновки:

– навчальна вибірка містить 21 атрибут, один з яких – змінна інтересу – **Churn**;

– навчальна вибірка містить інформацію про 7043 об’єкта;

– на етапі первинного огляду пропущених значень не виявлено;

– деякі з атрибутів необхідно конвертувати в інший тип даних.

Наступним етапом буде так званий описовий аналіз, або Exploratory Data Analysis, завдяки якому можна більш детальніше ознайомитись з вихідними

даними, проаналізувати розподіл кожного з атрибутів та вплив цих атрибутів на інші та цільову змінну Chrun.

3.2 Описовий аналіз вихідних даних

Для того щоб більш детальніше ознайомитися з даними, необхідно візуально оцінити кожен вхідний атрибут та його вплив на цільову змінну. Для цього будемо використовувати візуальний аналіз.

Найпершим етапом буде візуалізація розподілу цільової змінної – Churn. Змінна інтересу має бінарний тип даних, де 1 – означає що об’єкт відмовився від послуг, а 0 – залишився з компанією. Діаграма наведена на рисунку 3.3.

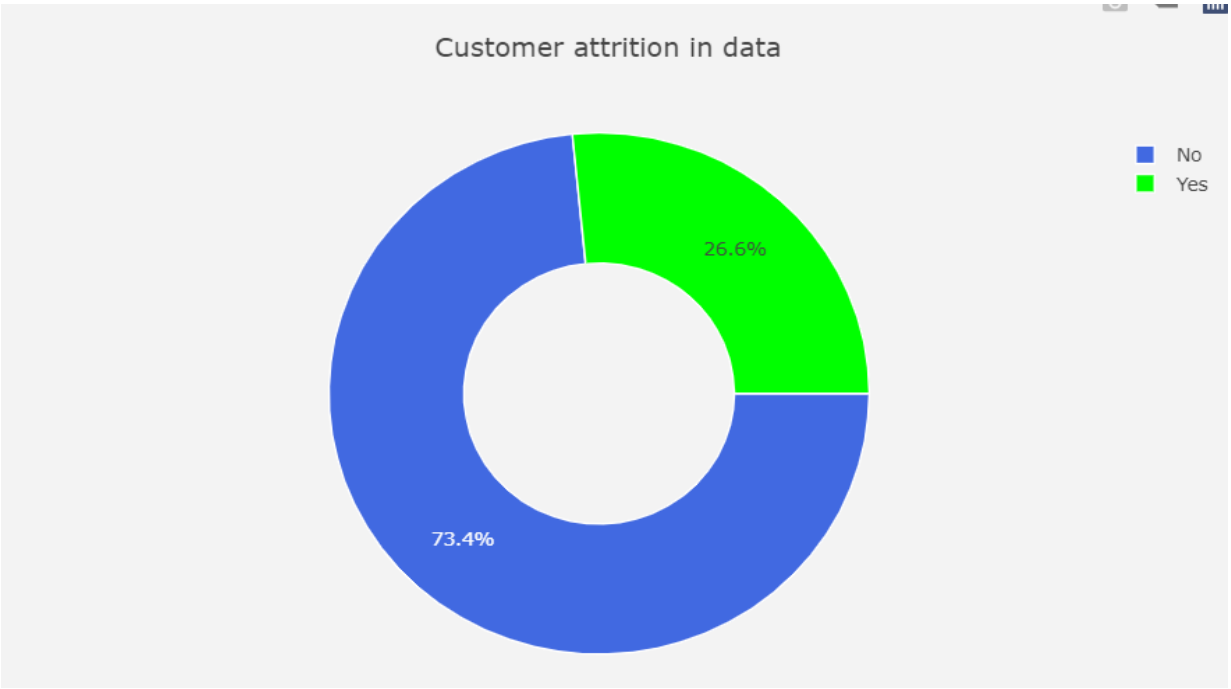


Рисунок 3.3 – Розподіл атрибуту Churn

Як видно на діаграмі, в навчальній вибірці 27% об’єктів – це ті хто відмовився від послуг. Це означає, що нам необхідно побудувати модель, на

даних з дисбалансом класів у цільовій змінній, що несе за собою деякі нюанси в моделюванні, а саме те, що модель може дуже добре запам'ятати патерни поведінки того класу, який представлений у великій кількості, та повністю ігнорувати той клас, який представлений у меншості в навчальній вибірці.

Далі необхідно побудувати графіки розподілу змінної в розрізі цільової змінної. Цей етап є одним з найголовніших, тому що дозволяє ознайомитись з даними, бізнесом, знайти якісь причино-наслідкові зв’язки. Для візуалізації категоріальних атрибутів використовувався pie-chart.

Першою із змінних є стать об’єкта – gender. Розподіл змінної зображено на рисунку 3.4

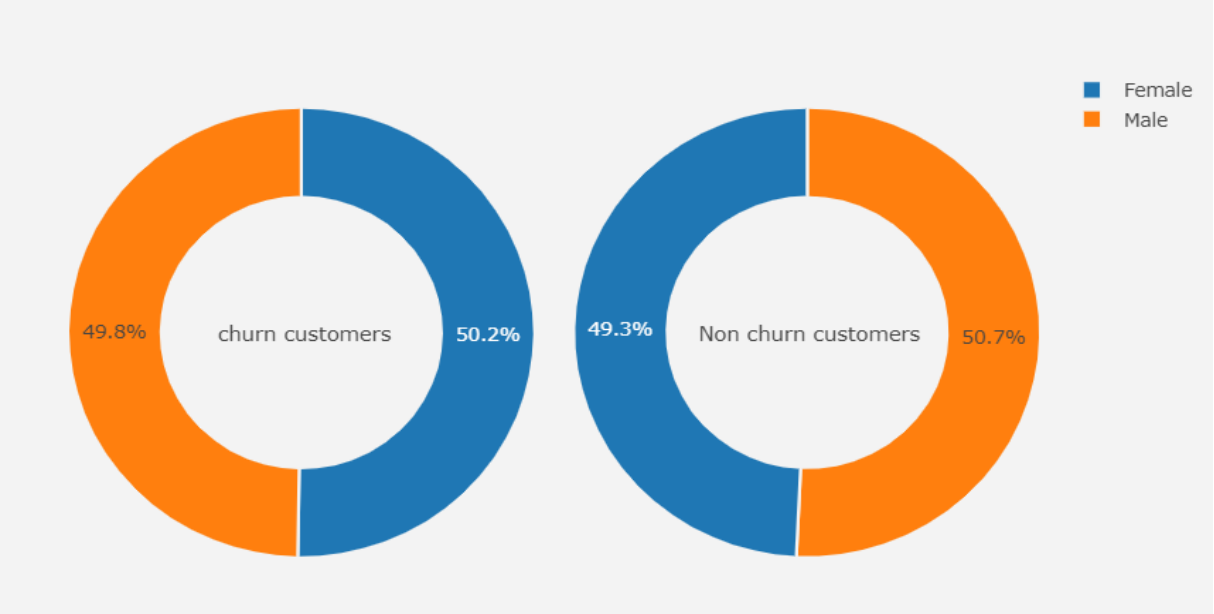


Рисунок 3.4 – Розподіл атрибуту gender в розрізі цільової змінної churn

На даному графіку наведено розподіл об’єктів з навчальної вибірки в залежності від їх статі та фактом чи відмовився той чи інший об’єкт від послуг чи ні, де кожне кільце або «пиріг» - фактор відмови від послуг, а кольори – група об’єктів. Як видно з графіку – приблизно однакові пропорції присутні в обох групах цільової змінної.

На даному етапі вкрай негативно заключати те, що такий атрибут як «стать» не впливає на вибір абонента відмовитись від послуг чи ні, адже на даному етапі порівнюється тільки дві змінні між собою. Якщо додати до цього аналізу ще одну змінну, наприклад, «наявність дітей у об’єкта» – розподіл може дуже сильно змінитися в сторону сильного впливу.

Змінна SeniorCetizenzhip. Розподіл змінної зображено на рисунку 3.5.

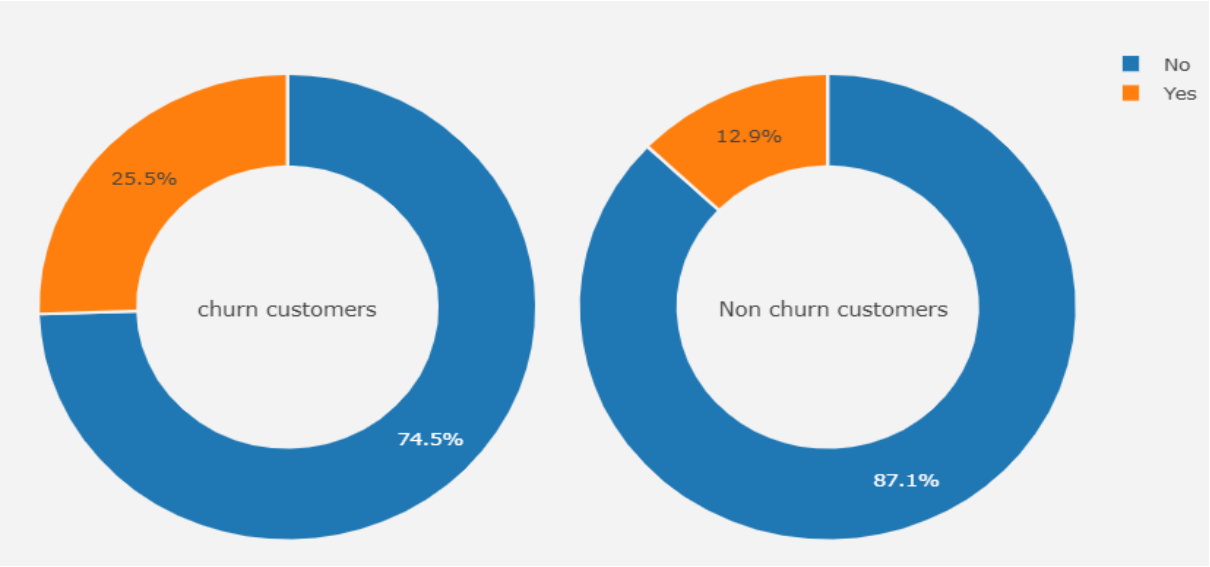


Рисунок 3.5 – Розподіл атрибуту SeniorCetizenship в розрізі цільової зміної churn

Можна спостерігати, що 25% з тих, хто відмовився – являються пенсіонерами, коли з тих, хто не відмовився – лише 13% є пенсіонерами.

Змінна Partner. Розподіл змінної зображено на рисунку 3.6.

Можна спостерігати, що 64% з тих, хто відмовився – входять до групи Partner = No, коли з тих, хто не відмовився – в групі Partner=No лише 47%. Як можна здогадатись, візуальний аналіз дозволяє зрозуміти які саме атрибути мають сильний вплив на цільовий фактор, що допомагає більш детальніше ознайомитись з проблемою та зрозуміти бізнес.

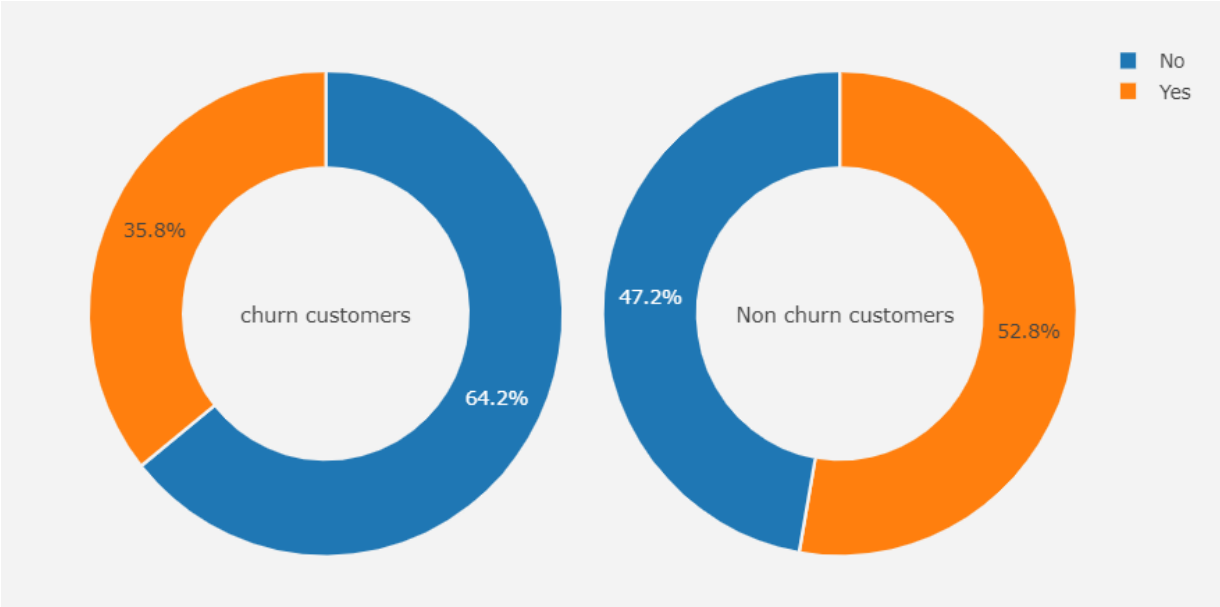


Рисунок 3.6 – Розподіл атрибуту Partner в розрізі цільової зміної Churn

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

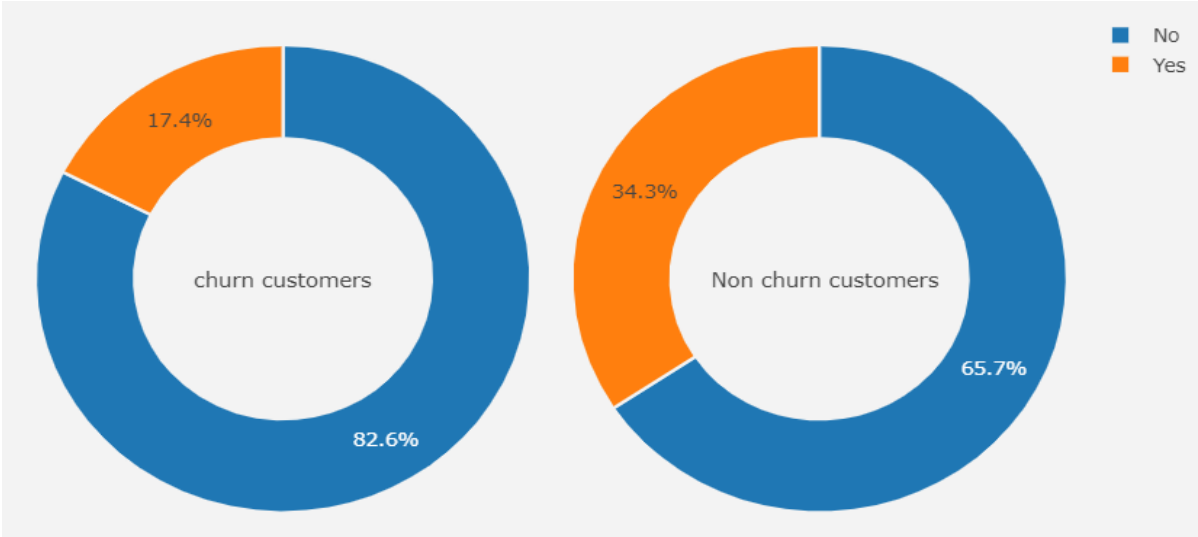


Рисунок 3.7 – Розподіл атрибуту «Has dependets» в розрізі цільової зміної Churn

Бачимо, що з тих, хто входить до класу відмовників від послуг – більшість с них мають фактор «Has dependents» = No, коли в іншій групі – навпаки.

Побудувавши графіки розподілів для кожної з категоріальних змінних та детально проаналізувавши їх було виявленно що:

– атрибут PhoneService не має впливу на цільову змінну churn при первинному аналізі;

– атрибут MultipleLines не має впливу на цільову змінну churn при первинному аналізі – в обох групах присутні однакові пропорції;

– InternetService – має вплив на те, чи відмовиться абонент від послуг чи ні, а саме ті, хто не користується фактором менш схильні відмовитись від послуг мобільного провайдера;

– ті об’єкти, у яких присутній фактор OnlineSercurity=Yes – менш схильні до відмови;

– ті об’єкти, в кого присутній фактор OnlineBackup=Yes - менш схильні до відмови;

– TechSupport – з тих об’єктів, в кого присутній даний фактор менш схильні до відтоку;

– StreamingTV та StreamingMovies атрибути не мають сильного впливу на цільову змінну Churn;

– ContractType – виходячи з візуального аналізу – ті об’єкти які платять за послуги по місячній передоплаті – дуже схильні до відключення, аніж ті, хто передплачує за послуги на рік або два;

– Такий атрибут як PaperlessBilling має досить різний розподіл між групами цільової змінної, тому вважаємо цю змінну впливовою;

– PaymentMethod – виходячи з візуального аналізу – метод оплати за послуги має сильний вплив на цільову змінну.

Далі проаналізуємо атрибути, які мають числовий розподіл – Tenure (тривалість користування послугами провайдера в місяцях) та MonthlyCharges/TotalCharges (сума місячних та всіх платіжних транзакцій об’єкта). Слід згрупувати кожне значення атрибуту в інтервальні групи чисел для більшого сприяння.

Побудуємо діаграму для атрибуту Tenure (рис. 3.8).

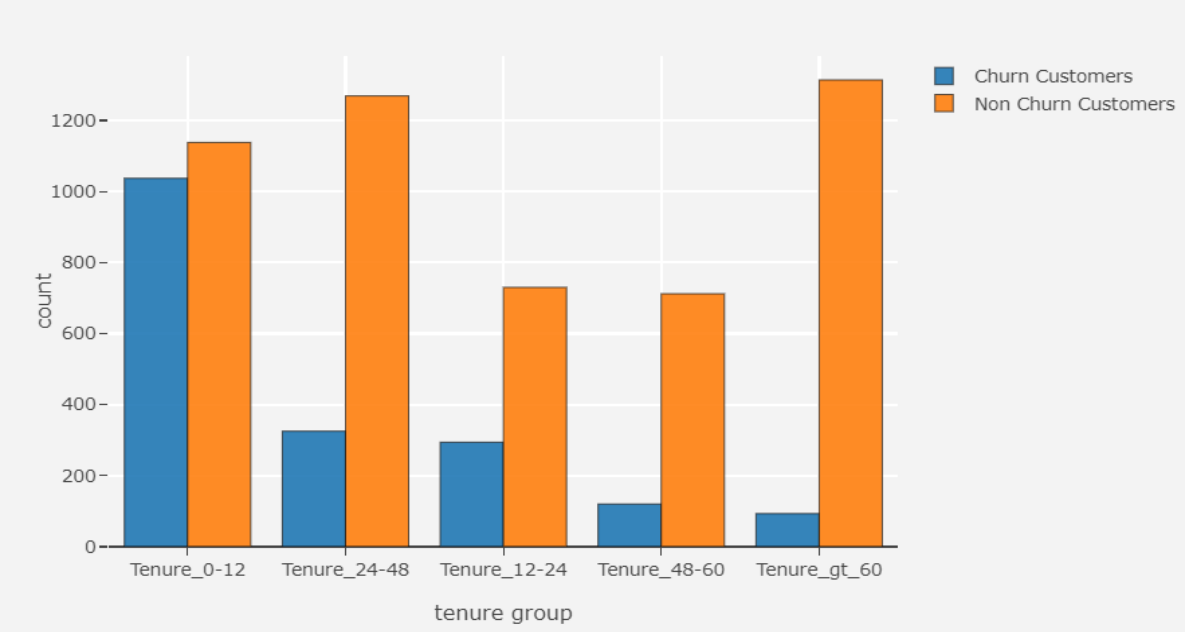


Рисунок 3.8 – Розподіл атрибуту «Tenure» в розрізі цільової змінної Churn

На графіку можна побачити, що ті об’єкти, у яких термін тривалості користування послугою до 12 місяців – більш схильні до відмови від послуг. У свою чергу, ті хто досить довго користується послугами – менш схильні до відключення. Це може бути обумовлено тим, що великий термін користування послугами викликає лояльність до компанії.

Побудуємо аналогічні діаграми для Monthly та Total charges атрибутів з застосування техніки бінінгу та групуванням по атрибуту Tenure group (рис. 3.9–3.10)

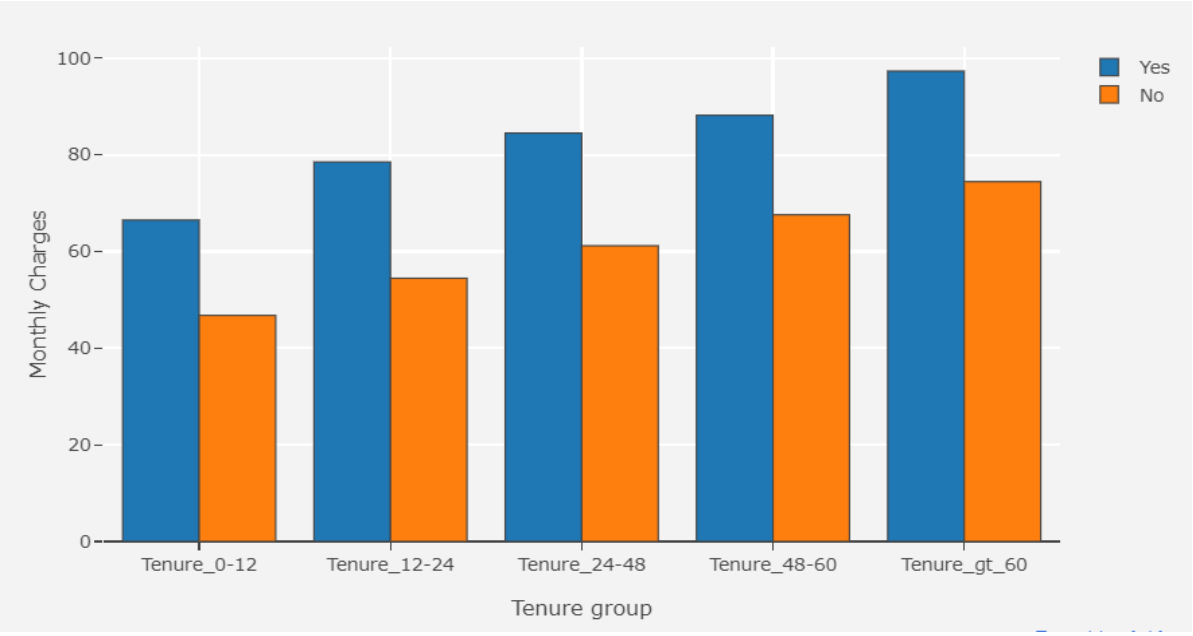


Рисунок 3.9 – Розподіл атрибуту «Monthly Charges» в розрізі цільової змінної Churn

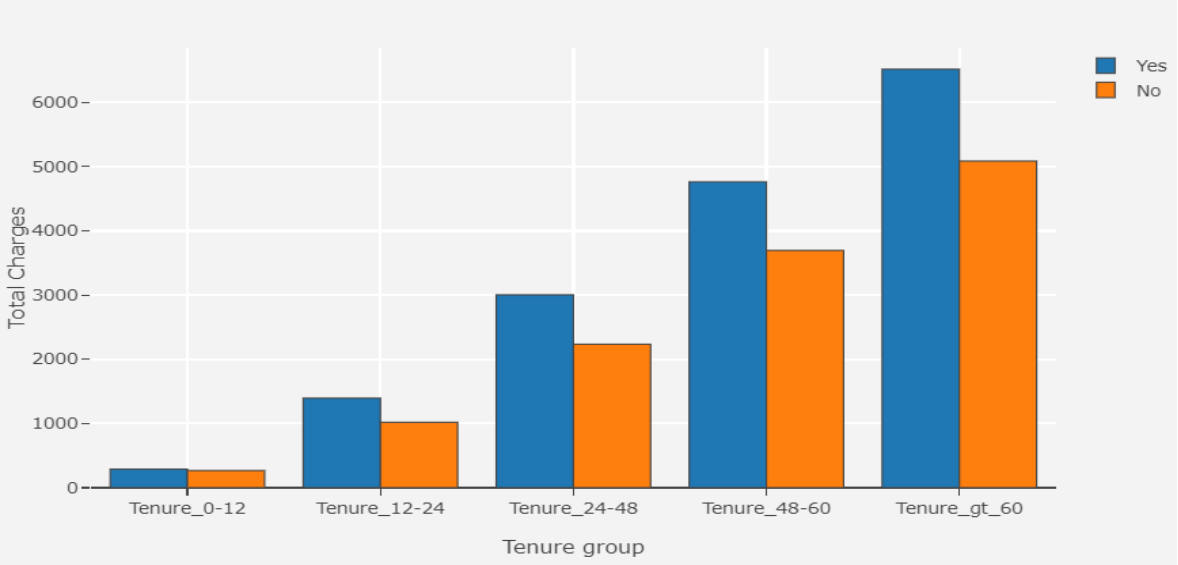


Рисунок 3.10 – Розподіл атрибуту «Total Charges» в розрізі цільової зміної Churn

На діаграмах 3.9 та 3.10 можна спостерігати, що в кожному «біні» Tenure group присутні приблизно однакові пропорції об’єктів, тобто на даному етапі

не можна вважати що атрибути Monthly та Total Charges не несуть цінного впливу на цільову змінну.

Наступним етапом необхідно проаналізувати кореляцію між атрибутами, тобто виявити ті змінні які лінійно залежні з іншими. Сильна кореляція між незалежними змінними може означати з одного боку сильний вплив на іншу змінну, а з іншого – сильно-корелюючі змінні між собою можуть вносити модель непотрібну інформацію, тим самим зменшуючи її якість. Матриця кореляцій атрибутів зображена на рисунку 3.11. Виявити позитивну, нетральну або негативну кореляцію допоможе шкала, яка зображення праворуч від матриці, де жовтий колір на шкалі означає сильну позитивну кореляцію, а темно-фіолетовий – сильну негативну кореляцію. По діагоналі матриці відображені кореляції атрибутів самих з собою, що не слід брати до уваги.

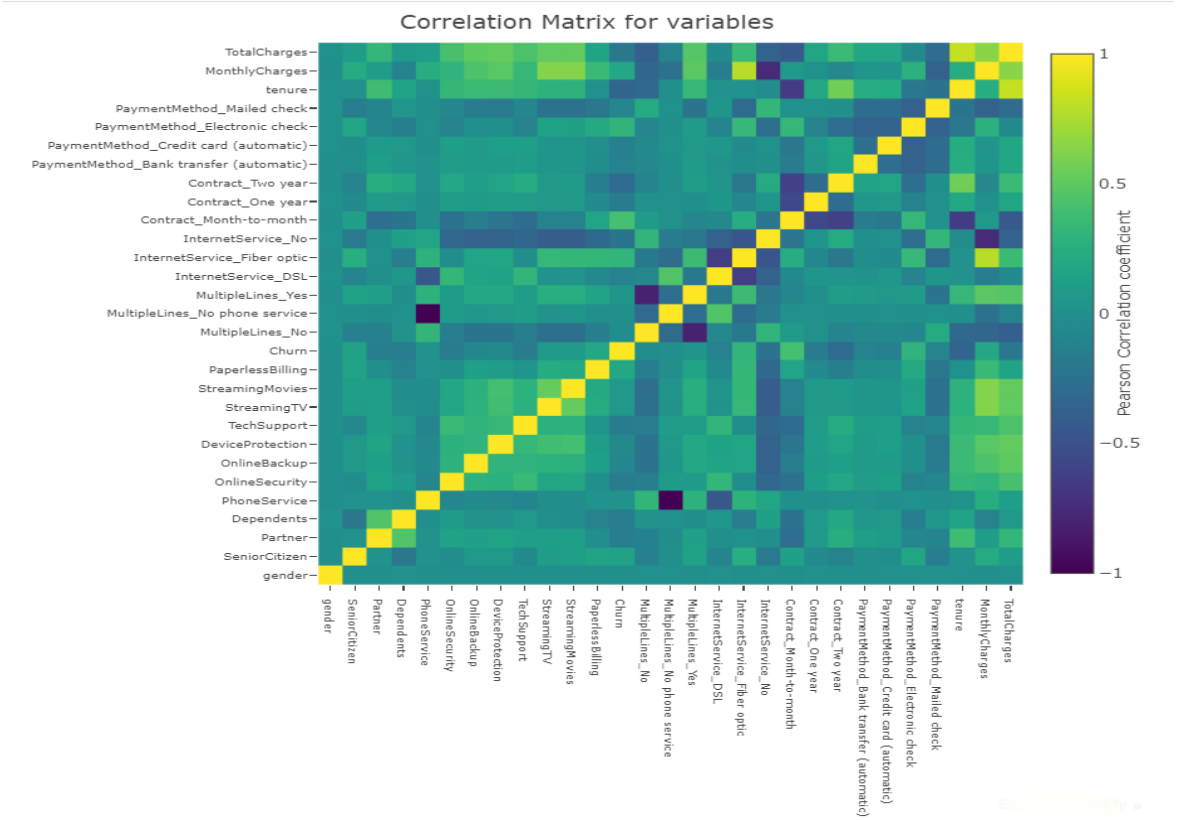


Рисунок 3.11 – Матриця кореляцій атрибутів

Детально проаналізувавши матрицю, можна зробити такі висновки.

– Багато атрибутів які означають додаткову послугу, наприклад InternetService, StreamingVideo, Streaming TV позитивно корелюють з Monthly charges, що є логічним взаємозв’язком – чим більше послуг має об’єкт тим більше буде місячний рахунок за послуги.

– Monthly charges та Total Charges досить сильно корелюють один з одним, адже один атрибут входить до складу іншого.

– Цікавим є те, що на залежну змінну чарн дуже сильно впливають такі атрибути як Contract-month-to-month, Payment method та Monthly charges.

Отже, ціллю розвідного аналізу є детальне знайомство з даними, що в свою чергу допомагає зрозуміти проблему, знайти цікаві факти, які допоможуть в створенні нових атрибутів на основі наявних (feature engineering), генерувати нові гіпотези не тільки в рамках моделювання даної проблеми, а й за її межами.

3.3 Підготовка даних та створення моделі. Вибір найкращого кандидата

Перед початком моделювання, вихідні дані необхідно обробити таким чином, щоб модель могла їх сприймати, а саме:

– всі атрибути з двома значеннями в текстовому форматі закодувати в числові;

– всі атрибути з трьома або більше значеннями в текстовому форматі переформувати в sparse-матрицю за допомогою методу One Hot Encoding;

– всі атрибути з дійсними числами нормалізувати за допомогою Z-стандартизацією.

Після переобробки вихідних данних методами класів LabelEncoder, OneHotEncoder та StandartScaler бібліотеки sklearn, дані мають таку репрезентацію, яка зображена на рисунку 3.12.

```
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 – Зразок вихідних даних після переобробки

Як було зазначено раніше, для моделювання проблеми прогнозування відключення буде використовуватися три методи: Логістична регресія, Дерево ухвалень рішень та Випадкові ліси. Але для початку нам необхідно розділити вибірку на дві групи методом випадкового відбору, де кожний об’єкт має однакову ймовірність бути обраним в одну з груп:

– train – частина даних, на яких навчається модель;

– test – частина даних, яка не буде приймати участь в навчанні моделей, на цю частину не буде ніякого впливу. На цій частині даних необхідно валідувати нашу модель та робити фінальні висновки щодо предиктивних властивостей моделі.

За стандартами спільноти Data Science – test вибірка повинна включати більше ніж 25% від обсягу всіх даних для репрезентативності результатів. Після розділу даних функцією train_test_split з параметром розміру test-вибірки в 25% було отримано 5274 об’єкта в train та 1758 в test, які були відібрані випадковим шляхом. Метрикою оцінення моделі було використано ROC AUC score та Accuracy score.

Для тренування обраних моделей імпортуємо в середовище Jupyter notebook класи LogisticRegression, DecisionTreeClassifier та RandomForestClassifier відповідно. Навчання моделей ініціалізується за допомогою методу .train() для кожного з класів моделей, а прогнозування – за допомогою методів .predict() або .predict_proba() (якщо нам потрібна саме ймовірність відношення об’єкта до певного класу.

Для прискорення навчання та порівняння предиктивної сили моделі було реалізовано функцію, яка показує якість моделі за допомогою функцій Accuracy ,ROC_AUC та матриці відповідностей, аргументами якої є:

– estimator – об’єкт моделі з заданими гіперпараметрами;

– X_train,y_train – навчальні вибірки де X* - набір атрибутів, y* - набір фактичних бінарних значень відношення до певного класу;

– X_test, y_test – тестувальні вибірки де де X* - набір атрибутів, y* - набір фактичних бінарних значень відношення до певного класу.

Після використання функції, було аналітично оцінено предиктивну здатність моделі. Детальний опис навчання LogisticRegression наведений на рисунку 3.13.

```
---------------------- 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 – Звіт з навчання LogisticRegression

Як видно з рисунку 3.13, логістична регресія показує досить перспективні результати, а саме похибка на тестувальній вибірці за метриками Accuracy та ROC AUC становить 76%. Також, якщо порівнювати похибки між train та test вибірками – можна побачити що такий феномен як перенавчання не присутній в даній моделі.

Наступним кандидатом було використано дерево ухвалень рішень – DecisionTreeClassifier. Звіт з навчання 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 – Звіт з навчання DecisionTreeClassifier

Як видно з рисунку 3.14, дерево ухвалення рішень показує поганий результат на тестовій вибірці через високу похибку, також можна побачити що навчене дерево досить сильно перенавчилось на навчальній вибірці, але це можна спростувати після підбору гіперпараметрів моделі.

Останнім кандидатом для моделі є RandomForestClassifier – метод випадкових лісів. Звіт з навчання показано 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 – Звіт з навчання 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 найкрайщих 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.

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 моделей

Назва моделі	ROC_AUC до пошуку гіперпараметрів	ROC_AUC після пошуку гіперпараметрів	Accuracy до пошуку гіперпараметрів	Accuracy після пошуку гіперпараметрів
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

3.4 Розробка Flask API для автоматизації прогнозування. Тестування мікросервісу

Коли фахівцем з аналізу даних або інженером машинного навчання розробляється проєкт із застосуванням машинного навчання за допомогою Scikit-Learn, TensorFlow, Keras, PyTorch тощо, – завжди кінцевою метою проекту є надання її доступності у виробництві, адже часто, працюючи над проектом машинного навчання, багато уваги акцентовано тільки на етапах дослідницькому аналізі даних (EDA), генерації нових атрибутів (feature engineering), налаштуванні гіпер-параметрів і т.д., але все це є лише частиною проекту.

Розгортання моделей машинного навчання або введення моделей у виробництво означає надання вашим моделям можливість бути доступним для кінцевих користувачів або систем. Однак є складність у розгортанні моделей машинного навчання. Ця частина дипломного проекту присвячена розробці, або мікросервісу з застосуванням Flask API та натренованої моделі.

Після процесінгу вихідних даних та навчання моделі на цих даних – необхідно створити так званий пайплайн предиктивної моделі, який буде містити в собі всі ці етапи. Під «пайплайном» мається на увазі об’єкт, який містить у собі послідовне застосування тих або інших об’єктів та їх функцій на вихідних даних, результат оброблених даних попереднього етапу в «пайплайні» передається до наступного етапу і т.д.

Реалізувати «пайплайн» можливе через об’єкт бібліотеки Sklearn – Pipeline та FeatureUnion, разом із застосуванням методів обробки даних відповідного типу, а саме:

– VariableSelector – персональна реалізація класу, яка дозволить відібрати атрибути відповідних типів по їх назвах.

– SimpleImputre – для заповнення пропущених або відсутніх значень статистиками для відповідного атрибуту, для числових даних – середнє арифметичне, для текстових – значення, яке найбільш частіше зустрічається.

– MultiLabelEncoder – для кодування даних типу string і трансформацію їх значень в числові. Наприклад, масив [‘Yes’,’No’,’No’,’Maybe’] – буде закодовано в масив [1, 0, 0, 2].

– StandartScaler – для трансформування числових данних в единий діапазон значень, який буде мати середнє арифметичне 1, та середньоквадратичне відхилення 0.

– RandomForestClassifier – об’єкт вже натренованої моделі, який містить в собі всі гіперпараметри. Саме цей етап відповідає за прогнозування.

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 – Реалізація Pipeline-об’єкта

Щоб використовувати створений процес обробки та прогнозування за межами робочого середовища, а саме в мікросервісі – збережемо об’єкт «пайплайну» в pickle-файл (формат .pkl) за допомогою модуля joblib, який присутній в бібліотеці sklearn.

Pickle використовується для серіалізації та десеріалізації об'єктних структур Python, також його називають «маршалінгом» або «згладжуванням». Під серіалізацією розуміється процес перетворення об'єкта в пам'яті в байт-потік, який може зберігатися на диску або надсилатися по мережі. Пізніше цей потік символів може бути повернутий і десериалізований назад до об’єкта Python. Pickling не слід плутати зі стисненням. Перше – це перетворення об'єкта з одного представлення (дані в оперативній пам’яті оперативної

пам'яті) в інше (текст на диску), а останнє – процес кодування даних з меншою кількістю бітів, щоб заощадити місце на диску.

Збережемо в pickle-фали об’єкт «пайплайну», типи вихідних даних та порядок атрибутів, в якому «пайплайн» повинен отримувати вихідні дані (рис. 3.17) – всі ці об’єкти будуть частиною мікросервісу, який буде реалізований.

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

Рисунок 3.17 – «Піклінг» об’єктів

Для реалізації мікросервісу на базі REST було використано Flask.

Flask – це web-фреймоврк. Це означає, що Flask надає вам інструменти, бібліотеки та технології, що дозволяють створювати веб-додаток. Цей веб-додаток може бути веб-сторінками, блогом, вікі або настільки ж великим, як веб-додаток календаря або комерційним веб-сайтом. Flask є частиною категорій мікро-фреймворків. Мікро-фреймворки, як правило, є мало залежними від зовнішніх бібліотек, що дозволяє досить швидко реалізовувати продукти.

Імпортуємо в середовище необхідні бібліотеки та їх модулі а саме flask, json, numpy, pandas та joblib.

Наступним кроком є створення об’єкту мікросервісу за допомогою класу Flask та завантаження pickle-файлів, які були збережені раніше.

Для реалізації мікросервісу, який дозволить автоматично прогнозувати ймовірність відтоку через вхідні данні – створимо єдину кінцеву точку, або «end point» - /predict, та зазначимо що ця кінцева точка може отримувати тільки запити медота POST. Далі зазначимо, що ця кінцева точка повинна отримувати json об’єкт з вхідними даними, які перевіряються на типи даних за

допомогою вже імпортованого pickle-об’єкта dtypes. Потім всі атрибути повинні буди впорядковані в тому порядку, в якому відбувалось тренування моделей – впорядковуємо атрибути за допомогою порядку атрибутів, який міститься pickle-об’єкті cols_order.pkl. Останнім етапом є процесінг даних та прогнозування відтоку за допомогою імпортованого «пайпалайну» та вивід результату. Релазіція мікросервісу для прогнозування відтоку наведена на рисунку 3.18.

```
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 – Реалізація REST API для автоматичного прогнозування за допомогою Flask

Вхідними даними для буде .json об’єкт, якій обов’язково повинен містити індекс (на випадок, коли з’явиться необхідність передавати декілька об’єктів). В свою чергу – кожен індекс мати в собі масив типу dictionary, який зберігає атрибути для прогнозування. Приклад вихідних даних наведено на рисунку 3.19.

```
{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 – Приклад вихідних даних для мікросервісу

Після реалізації мікросервісу його необхідно протестувати. Для цього запускаємо мікросервіс та виконуємо POST-request з вихідними даними до нашого мікросервісу.

Програмний код реалізації API збережено у файлі app.py. Ініціалізуємо запуск програми app.py через terminal командою python app.py

```
Mark@Best-Komp MINGW64 ~/Desktop/sumdu/project
$ python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\externals\joblib\__init__.py:15: FutureWarning: sklearn.externals.joblib
precated in 0.21 and will be removed in 0.23. Please import this functionality directly from joblib, which can be installed
 pip install joblib. If this warning is raised when loading pickled models, you may need to re-serialize those models with
-learn 0.21+.
  warnings.warn(msg, category=FutureWarning)
 * Restarting with stat
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\externals\joblib\__init__.py:15: FutureWarning: sklearn.externals.joblib
precated in 0.21 and will be removed in 0.23. Please import this functionality directly from joblib, which can be installed
 pip install joblib. If this warning is raised when loading pickled models, you may need to re-serialize those models with
-learn 0.21+.
  warnings.warn(msg, category=FutureWarning)
 * Debugger is active!
 * Debugger PIN: 301-376-991
 * Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
```

Рисунок 3.20 – Ініціалізація мікросервісу

Через те, що ініціалізація мікросервісу була зроблена на локальному ПК – мікросервіс доступний за адресою localhost http:///127.01.01:8080/. За замовчуванням, створений мікросервіс використовує 8080 порт. Зробимо тестовий POST request з вхідними даними, які зображені на рисунку 3.19 запит і переконаємося що все працює за планом. Зразок коду тестового запиту показано 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 – Приклад POST запиту до мікросервісу

Як видно з рисунку 3.21, – можна зробити висновок що тестування пройшло успішно. Test-case тестування мікросервісу наведено у таблиці 3.2.

Таблиця 3.2 – Test-case тестування

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

Телекомунікаційна галузь є однією з найшвидше зростаючих галузей Україні. Через конкурентоспроможні ставки різних постачальників, замовники часто схильні переключатися між ними та тим самим закликають до необхідності прогнозувати потенційні збитки і вживання відповідних дій для запобігання їх зростання. Як результат, в телекомунікаційній галузі було зроблено багато спроб прогнозувати клієнтів, що збиваються, перш ніж вони фактично залишають провайдера.

Управління відтоком клієнтів отримує все більшу увагу, оскільки утримання існуючих клієнтів є вигідним і важливим для телекомунікаційних компаній. Вартість на залучення нових клієнтів становить набагато більше, ніж витрати на утримання діючих клієнтів у бізнесі, особливо у насиченому ринку телекомунікацій.

Важливість цього виду досліджень на ринку телекомунікацій полягає у наданні допомоги компаніям збільшити прибуток, адже відомо, що прогнозування відтоку та його зменшення – одне з найбільш важливих прибуткових джерел телекомунікаційних компаній.

Ця робота присвячена проблемі прогнозування відтоку на основі даних телекомунікаційних операторів. Також, в цій роботі виявлено актуальність та значущість вирішення цієї проблеми.

Вирішення даної проблеми можливе із застосуванням мови програмування Python, яка добре себе зарекомендувала у вирішенні подібних типів задач.

На основі проведеного дослідження було обрано методологію CRISP-DM, яка складається з трьох ступенів обробки та передбачення, причому результат роботи кожного з етапів обробки може бути використаний окремо.

Поряд із побудовою прогнозуючої моделі була спроектована загальна архітектура прототипу мікросервісу, який призначений для організації зручної роботи з даними від баз даних.

Основні результати роботи оприлюднені та обговорені на міжнародній науково-технічній конференції студентів та молодих вчених «Інформатика, математика, автоматика» (ІМА – 2020) (Суми, 2020 р.) [20].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. M. Owczarczuk, “Churn models for prepaid customers in the cellular telecommunication industry using large data marts,” Expert Syst. Appl., vol. 37, no. 6, pp. 4710–4712, Jun. 2010.

2. Trevor Hastie, Robert Tibshirani, and Jerome Friedman, The Elements of Statistical Learning:Data Mining, Inference, and Prediction. Springer, 2008.

3. N. Lu, H. Lin, J. Lu, and G. Zhang, “A Customer Churn Prediction Model in Telecom Industry Using Boosting,” IEEE Trans. Ind. Inform., vol. 10, no. 2, pp. 1659–1665, May 2014.

4. 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, 2008.

5. Hui Li, Deliang Yang, Lingling Yang, Yao Lu, and Xiaola Lin, “Supervised Massive Data Analysis for Telecommunication Customer Churn Prediction,” in Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom), 2016 IEEE International Conferences on, 2016, pp. 163–169.

6. J. Burez and D. V. den Poel, “Handling class imbalance in customer churn prediction,” Expert Syst. Appl., vol. 36, no. 3, Part 1, pp. 4626–4636, 2009.

7. Nitesh V. Chawla, “Data mining for imbalanced datasets: An overview,” in Data mining and knowledge discovery handbook, Springer US, 2005, pp. 853–867.

8. Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

9. Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz, “Applying Support Vector Machines to Imbalanced Datasets,” in Machine Learning: ECML 2004, 2004, pp. 39–50.

10. 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. 2004.

11. Over-Sampling Method in Imbalanced Data Sets Learning,” in Advances in Intelligent Computing, 2005, pp. 878–887.

12. G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, “A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data,” SIGKDD Explor Newsl, vol. 6, no. 1, pp. 20–29, Jun. 2004.

13. Tomek T., “Two Modifications of CNN,” IEEE Trans. Syst. Man Commun., pp. 769–772, 1976.

14. Офіційна документація Python – https://docs.python.org/3.7/

15. Опис методології CRISP-DM https://www.sv-europe.com/crisp-dm-methodology/

16. Telco Customer Churn dataset – https://www.kaggle.com/blastchar/telco-customer-churn

17. Quinlan JR. C4.5: Programs for Machine Learning. San Mateo, CA, USA: Morgan Kaufmann Publishers; 1993.

18. Breiman L, Friedman J, Olshen R, Stone C. Classification and Regression Trees. Belmont, CA, USA: Wadsworth International Group; 1984.

19. Caruana R, Niculescu-Mizil A. An empirical comparison of supervised learning algorithms. In: Cohen WW, Moore A, editors. Machine Learning, Proceedings of the Twenty-Third International Conference; New York: ACM; 2003. pp. 161–168.

20. Стакан М. А. Інтелектуальна інформаційна технологія для прогнозування відключення від послуг мобільних операторів / М. А. Стакан, О. А. Шовкопляс // Матеріали та програма міжнародної науково-технічної конференції студентів та молодих вчених «Інформатика, математика, автоматика» (ІМА – 2020), 20–24 квітня 2020 р., м. Суми, Україна. – Суми, 2020. – С. 186. – http://elitconference.sumdu.edu.ua/proceedings/

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_t
rain,X_test,y_test)

validate(RandomForestClassifier(n_estimators = 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

pd.options.display.max_columns = None
pd.options.display.max_rows = 50
sys.path.append('./')

# імопортування навчальної вибірки
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'
    ]

    df.head()
    df = df.replace(" ", np.nan)
    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 для реалізації мікросервісу на базі Flask API

```
    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 мікросервісу

```
      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}]
```