

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу**

До захисту допущено
Завідувач кафедри
_____ Оксана ТИМОЩУК
«07» 06 2021 р.

Дипломна робота

**на здобуття ступеня бакалавра
за освітньо-професійною програмою «Системний аналіз і управління»
спеціальності 124 "Системний аналіз"
на тему: «Інтелектуальні агенти для системи автоматизації управління
бізнес-процесами»**

Виконала:

Студентка IV курсу, групи КА-74

Люта Валерія Олександрівна _____

Керівник:

доцент кафедри ММСА

канд. фіз.- мат. наук Алла Петрівна Яковлева _____

Консультант з економічного розділу:

доцент кафедри ТПЕ

к.е.н., доцент Надія Василівна Рощина _____

Консультант з нормоконтролю:

доцент кафедри ММСА

к.т.н., доцент Анатолій Єпіфанович Коваленко _____

Рецензент:

доцент кафедри КТК

д.т.н. доцент Корнага Ярослав Ігорович _____

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

Студентка _____

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 124 «Системний аналіз»

Освітньо-професійна програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Оксана ТИМОЩУК

« ____ » _____ 20__ р

ЗАВДАННЯ

на дипломну роботу студентці

Лютій Валерії Олександрівні

1. Тема роботи «Інтелектуальні агенти для системи автоматизації управління бізнес-процесами», керівник роботи Яковлева Алла Петрівна, доцент кафедри ММСА, кандидат фізико-математичних наук, затверджені наказом по університету від «26» травня 2021 р. №1344-с.

2. Термін подання студентом роботи: 07.06.2021 р.

3. Вихідні дані до роботи: дані про вибір страхових пакетів, клієнтами страхових компаній, дані про ціни та умови страхових пакетів. Дані взяті з ресурсу Kaggle та загальнодоступні дані з сайтів українських страхових компаній.

4. Зміст роботи дослідження теоретичних основ автоматизації управління бізнес-процесами, розроблення системи для вибору.

5. Перелік ілюстративного матеріалу(із зазначенням плакатів, презентацій тощо): Слайди №2,3,4,5,6,7,8,9,10,11,12,13,14,15.

6. Консультанти розділів роботи*

* Якщо визначені консультанти. Консультантом не може бути зазначено керівника дипломної роботи.

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Рощина Н.В., доцент кафедри ТПЕ		

7. Дата видачі завдання 08.02.2021р

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Отримання завдання	19.04.2021	Виконано
2.	Збір інформації	26.04.2021	Виконано
3.	Аналізи існуючих підходів	29.05.2021	Виконано
4.	Проектування сервісу	02.05.2021	Виконано
5.	Підбір тестових даних	10.05.2021	Виконано
6.	Розробка програмного продукту	19.05.2021	Виконано
7.	Тестування програмного продукту	22.05.2021	Виконано
8.	Оформлення дипломної роботи	24.05.2021	Виконано

Студент

Валерія ЛЮТА

Керівник

Алла ЯКОВЛЕВА

РЕФЕРАТ

Дипломна робота: 122 с., 6 табл., 33 рис., 3 додатки, 19 джерел.

ІНТЕЛЕКТУАЛЬНІ АГЕНТИ, БІЗНЕС-ПРОЦЕСИ, УПРАВЛІННЯ
БІЗНЕС-ПРОЦЕСАМИ, ЛІНІЙНА РЕГРЕСІЯ, МОДЕЛЬ.

Тема: Інтелектуальні агенти для системи автоматизації управління
бізнес-процесами.

Мета роботи: розібратися в теоретичних основах автоматизації
управління бізнес-процесами, розробити систему для вибору
найоптимальнішого страхового плану, використовуючи сучасні методи
машинного навчання, знайти та обробити дані для подальшого навчання
системи.

Об'єкт дослідження: дані про користувача та його переваги у виборі
страхування.

Предмет дослідження: машинне навчання у автоматизації управління
бізнес-процесами.

Дослідження відбувається за допомогою методів машинного навчання,
що вирішуються задачі регресії – лінійна регресія, Random Forest.

У даній роботі зібрано власну базу даних методом веб-скрейпінгу і дані
про страхування життя та медичне страхування взяті з ресурсу kaggle.

Результатом даної роботи є програмний продукт у вигляді прикладного
програмного інтерфейсу, який знаходить найоптимальніший страховий пакет
з усіх можливих.

ABSTRACT

Thesis: 122 p., 6 tabl., 33 fig., 3 appendices, 19 sources.

INTELLECTUAL AGENTS, BUSINESS PROCESSES, BUSINESS PROCESS MANAGEMENT, LINEAR REGRESSION, MODEL.

Theme : Intelligent agents for business process management systems.

Purpose of the work to understand the theoretical foundations of business process automation, to develop a system for selecting the best insurance plan, using modern methods of machine learning, to find and process data for further training of the system.

Object of research: data about the user and his advantages in choosing insurance.

Subject of research: machine learning in automation of business process management.

The research is carried out using machine learning methods that solve regression problems - linear regression, Random Forest.

This work collects its own database by web scraping and data on life insurance and health insurance taken from the resource kaggle.

The result of this work is a software product in the form of an application software interface, which finds the best insurance package of all possible.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 ОГЛЯД БІЗНЕС-ПРОЦЕСУ АВТОМАТИЗАЦІЇ ВИБОРУ СТРАХОВОГО ПЛАНУ	10
1.1 Актуальність задачі автоматизації вибору страхового плану	10
1.2 Аналіз існуючих підходів до вирішення задачі автоматизації вибору страхового плану	14
1.3 Постановка задачі.....	18
1.4 Висновки до розділу 1	19
РОЗДІЛ 2 АВТОМАТИЗАЦІЯ БІЗНЕС-ПРОЦЕСІВ ТА МЕТОДИ РОЗВ’ЯЗАННЯ ЗАДАЧ НА ОСНОВІ АНАЛІЗУ ДАНИХ	20
2.1 Необхідність автоматизації бізнес-процесів	20
2.2 Основи аналізу даних для автоматизації вибору страхового плану	21
2.2.1 Аналіз первинних даних та виявлення аномалій	23
2.3 Завдання аналізу даних для автоматизації вибору страхового плану ...	25
2.4 Методи та критерії якості роботи моделі.....	29
2.4.1 Лінійна регресія	30
2.4.2 Логістична регресія	33
2.4.3. Древа рішень	36
2.4.4. Випадковий ліс (Random forest).....	38
2.5 Висновки до розділу 2	39
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ ДЛЯ ВИБОРУ ОПТИМАЛЬНОГО СТРАХОВОГО ПЛАНУ.....	40
3.1 Обґрунтування вибору платформи та мови програмування	40

3.2. Збір та аналіз даних та навчання моделі	41
3.3 Керівництво користувача кінцевим програмним продуктом	47
3.4 Висновки до розділу 3	55
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ	56
4.1 Постановка задачі проектування	56
4.2 Обґрунтування функцій та параметрів програмного продукту	57
4.3 Економічний аналіз варіантів розробки	65
4.4 Вибір кращого варіанта ПП техніко-економічного рівня	70
ВИСНОВКИ	72
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	73
ДОДАТОК А	75
ДОДАТОК Б.....	83
ДОДАТОК В.....	104

ВСТУП

Для будь-якої держави основним критерієм є забезпечення добробуту усім її громадянам та сталого економічного зростання. Для розвинених країн характерно використання сучасних технологій у сфері послуг та виробництва, потужна та стабільна сфера соціального захисту та підвищення загального рівня життя населення. Звичайно, що в умовах стабільного економічного розвитку кожен громадянин держави може робити довгострокові фінансові планування задля забезпечення стабільності та впевненості у майбутньому. Важливу частину у плануванні такої стабільності відіграє страхування. Страхування може зменшити негативні наслідки соціальних ризиків, наслідки стихійних лих, знизить втрату доходу внаслідок втрати працездатною особою роботи. Також страхування допоможе забезпечити гідну пенсію та реалізацію планів у майбутньому. Також страхові фонди ми можемо розглядати як одне з джерел інвестицій в економічне зростання держав.

В нинішньому світі спостерігається високі темпи розвитку послуг страхових компаній. Зрозуміло, що найбільшою перевагою для клієнтів таких компаній є ефективне використання механізмів страхового захисту. Керування діяльністю страхової компанії є доволі складним процесом. Якість та ефективність керування основною діяльністю впливають на фінансову стабільність компаній.

На жаль, існуючі програмні продукти в більшій частині пропонують програмні продукти для ведення страхової діяльності. Такі продукти не можуть використовуватися для автоматизації керування бізнес-процесами, також вони не проводять аналіз даних та не є ефективними для управління самими процесами.

Темою даної роботи було обрано створення програмного продукту для автоматизації керуванням бізнес-процесами, а саме управлінням страхового підприємства з допомогою використання нейронних мереж для підбору

найоптимальніших страхових тарифів відповідно до наданих клієнтом особистих даних. Такий програмний продукт допоможе зберегти кошти страхових компаній, прискорення процесу обробки даних, виключення рутинної роботи, можливість автоматичного створення звітності, контроль та аналіз даних.

РОЗДІЛ 1 ОГЛЯД БІЗНЕС-ПРОЦЕСУ АВТОМАТИЗАЦІЇ ВИБОРУ СТРАХОВОГО ПЛАНУ

1.1 Актуальність задачі автоматизації вибору страхового плану

З огляду на пандемію короно-вірусної хвороби, всесвітні кризи чи інші катастрофи з якими постійно стикається людство можливість страхування стає скоріше необхідністю і потребою. Щодня у житті людини відбуваються події на, які вона не має змоги вплинути, вони завдають матеріальні збитки і тим самим несуть загрозу. Страхування допомагає підприємствам або клієнтам мати впевненість у майбутньому та мінімізувати втрати.

Страхування – це вид цивільно-правових відносин, який захищає майнові інтереси юридичних осіб та громадян в наслідок певних подій, які зазначені у договорі страхування, за рахунок грошових фондів, які сформовані страховими внесками, страховими преміями. Зазначимо, що страхування являється одною з складових фінансової системи будь-якої держави. Виділяються наступні основні види страхування, такі як:

1. Соціальне страхування;
2. Особисте страхування;
3. Майнове страхування;
4. Страхування відповідальності;
5. Страхування підприємницьких ризиків.

Детальніше розглянемо кожен з вищезазначених видів страхування.

Соціальне страхування забезпечує громадянам виплати за рахунок коштів підприємств, організацій та держави, поєднує інтереси конкретної особи, колективу та суспільства у використанні коштів. Суб'єктами такого виду страхування є: держава, профспілки, організації, громадяни, а форма допомоги зазвичай – пільги, виплати. Прикладом соціального страхування

можна назвати виплати по безробіттю, пенсійне страхування, виплати у зв'язку зі втратою працездатності, тощо.

Особисте страхування має багато спільного з соціальним страхуванням. Проте головна відмінність полягає у тому що, джерелом формування страхових фондів для особистого страхування є індивідуальні доходи. Необхідність такого страхування зумовлена погіршенням стану екології, підвищений ступінь ризику, важкою фізичною роботою чи зростанням частки похилих людей у населенні. До особистого страхування відносять страхування життя, медичне страхування, страхування від нещасних випадків і тощо. Страхування робітників та службовців в обов'язковому порядку не є особистим страхуванням, а є скоріше соціальним.

Майнове страхування має досить багато різних проявів, до нього відносять страхування всіх видів транспорту чи майна, страхування фінансових ризиків. Внаслідок настання страхового випадку страхувальник має отримати компенсацію шкоди за своє майно. Фізична особа може здійснити страхування свого особистого майна, таке як: свій особистий транспорт, свою нерухомість, домашнє майно. Юридична особа може застрахувати майно підприємства: рухоме та нерухоме майно, виробниче обладнання, необхідні інструменти.

Страхування відповідальності здійснюється для покриття фінансових втрат у разі виникнення, у результаті виконання службових обов'язків, шкоди працівнику. У результаті настання страхового випадку з працівником роботодавець несе фінансову відповідальність.

Страхування фінансових ризиків запобігає ризику втрати прибутку чи не отримання доходу через низку причин.

Під час короно-вірусної пандемії багато секторів бізнесу постраждало в зв'язку з неможливістю переведення всіх бізнес-процесів у онлайн режим. Неможливість працювати персоналу ресторанів, торгових центрів, маленьких підприємств завдало великі матеріальні втрати. Страхові компанії робили

виплати для своїх клієнтів, що допомогло багатьом мати фінансову стабільність. Також під час пандемії зріс попит для медичного страхування та страхування життя. Страхові компанії допомагають отримати належну, своєчасну та кваліфіковану допомогу у важкий час для медичної системи. Туристичне страхування допомагає отримати допомогу будь-якого характеру перебуваючи закордоном. З огляду на минулий 2020 рік, можемо сказати що страхові компанії користувалися неабияким попитом. За даними одної з найбільших українських страхових компаній АТ «Страхова компанія «ІНГО» можна спостерігати позитивну динаміку у рості показників (рисунок 1.1). Проте окрім зростання показників спостерігаємо, що ріст зобов'язань перед клієнтами також зріс на 17%.



Рисунок 1.1 – Порівняння показників страхової компанії 2019 та 2020 років

За даними страхових компаній чисті премії з медичного страхування зросли на 34%, страхування від ризиків збільшилися на 13,5%, страхування майна збільшилось на 9,2%, страхування життя на 18.4%, автострахування також зросло на 17,8%. Отже, бачимо що пандемія вплинула на багато різних

сфер життя та на страхування. Окрім цього, була необхідність перевести процеси підбору страхового плану в онлайн режим, що набагато спростило цей процес і зменшило час для клієнта.

Не так давно з'явилась нова форма охорони майнових та особистих інтересів страхованих – онлайн-страхування.

Онлайн-страхування – це процес складання поліса страхування з використанням мережі інтернет на сайті страхової компанії, що включає в себе розрахунок страхового тарифу та суми, яка повинна бути сплачена, підписання поліса. У такій формі страхування має низку таких переваг:

1. Спрощення процесу страхування та можливість порівняння страхових компаній, тарифів;
2. Скорочення часу страхувальника завдяки можливості зробити замовлення не покидаючи приміщення;
3. Автоматичний розрахунок вартості;
4. Можливість та простота способу оплати (можна обрати банківський переказ, електронні гроші, готівковий розрахунок);
5. Зменшення людського фактору, мінімізація можливих помилок.

У всьому світі останні роки активно зростаються продажі послуг та товарів в мережі інтернет. Біля 70% європейців вже зараз користуються онлайн-страхуванням. Українці також поступово переходять на користування онлайн послугами страхування (рисунок 1.2).

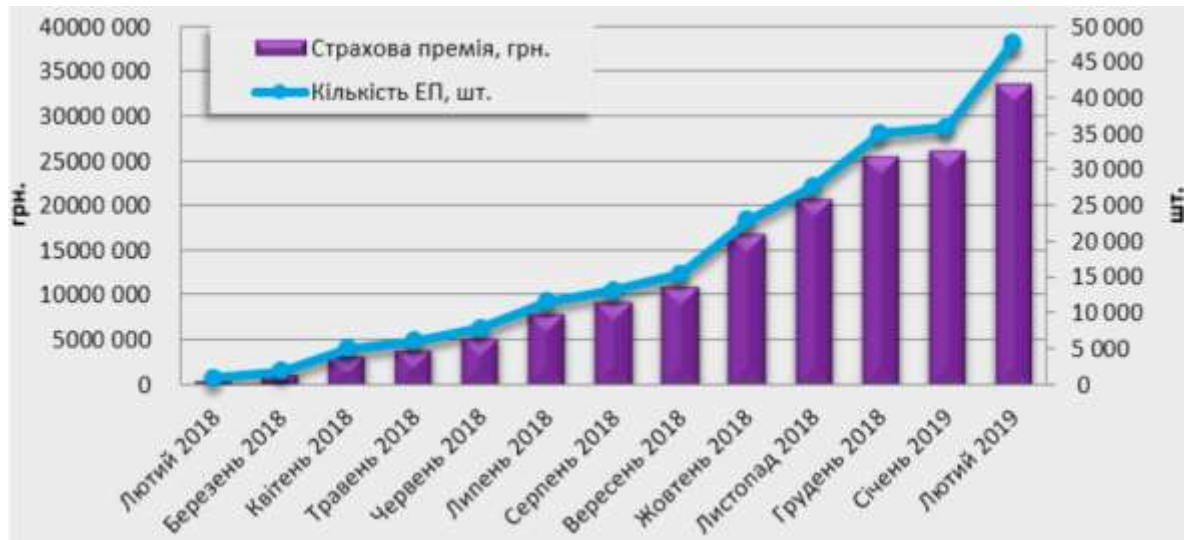


Рисунок 1.2 – Статистика продажів електронних договорів ОСАГО

Отже, з розвитком онлайн-страхування існує потреба у найкращому і найшвидшому підборі страхових тарифів з урахуванням потреб користувача. Тому оцінка всіх можливих факторів є запорукою швидкого та надійного вибору страхового полісу. Так як вищезгадана сфера зараз розвивається з великою швидкістю тема даної роботи є актуальною та допоможе зекономити сили та ресурси на самостійний пошук і вибір найкращого варіанту.

1.2 Аналіз існуючих підходів до вирішення задачі автоматизації вибору страхового плану

Насамперед, варто зазначити, що готових рішень по автоматизації вибору страхових полісів різних видів на ринку в даний час немає. Проте, є схожі рішення вирішення даного питання, які будуть розглянуті надалі.

Існує ряд прикладів з використанням онлайн-калькуляторів для розрахування вартості різних пакетів від різних страхових компаній, як українських так і зарубіжних. Для прикладу розглянемо програмний продукт відомої страхової компанії ARX (рисунок 1.3).

Розрахунок Умови страхування Персональні дані Сплата

Дані для розрахунку

Кількість поїздок: Одна Декілька ?

Дата виїзду:

Дата повернення:

Територія дії договору: Весь світ, окрім США та Канади

Кількість туристів: Одні

Мета поїздки: Робота (фізична праця)

ДАЛІ

Рисунок 1.3 – Онлайн-калькулятор розрахунку страхового пакету компанії
ARX

Дійсно, онлайн-страхування є доволі швидким та зрозумілим для клієнта. За декілька кроків, користувач отримає перелік програм страхування компанії ARX з детальним описом усіх опцій та умов страхування (рисунок 1.4).

Оберіть програму страхування

Оберіть страхову суму в EUR на одну Застраховану особу

30 000

	462.50 грн.	656.91 грн.	763.50 грн.
Програми страхування	Класична	Оптимальна	Преміальна
❖ Основний пакет	✓	✓	✓
❖ Надання невідкладної медичної допомоги у зв'язку зі захворюваннями, що пов'язані з особливо небезпечними інфекціями (в тому числі covid-19)	✓	✓	✓
❖ Надзвичайне продовження перебування		✓	✓
❖ Дostroкове повернення		✓	✓
❖ Відшкодування при затримці авіарейсу		✓	✓
❖ Відшкодування при затримці багажу		✓	✓
❖ Невідкладна гінекологічна допомога до 30-го тижня вагітності		✓	✓
❖ Надзвичайний приїзд третьої особи			✓
❖ Організація послуг адвоката			✓
❖ Виплата при втраті особистих документів			✓
❖ Організація доставки застрахованої особи до місця призначення при поломці авто			✓
	462.50 грн.	656.91 грн.	763.50 грн.
	Обрати	Обрати	Обрати

Рисунок 1.4 – Розрахунок програм страхування компанії ARX

Подібний калькулятор є зручний для користувачів, які вирішили застрахувати своє майно у даній компанії. Проте якщо клієнт ще обирає програму страхування, порівнює ціни та умови, йому необхідно витратити свій

час на заповнення анкет на сайтах різних компаній. Також варто зазначити, що вказаний сайт надає можливість розрахувати поліс лише для одного виду – туристичне страхування.

Окрім онлайн-калькуляторів, які можна побачити на сторінках страхових компаній, існують також сервіси для порівняння цін між різними фірмами. Для прикладу розглянемо страховий сервіс – Parasol.ua (рисунок 1.5).

Электронное
ОСАГО

Больше никаких бумажек,
полис на e-mail за 5 минут

Калькулятор

Тип транспортного средства
Легковой автомобиль

Объем ☐ Есть льгота
1601 — 2000 см3

Город прописки ☐ Иностранная
Одесса

Одесса, Львов, Харьков, Днепр, Киев

Сравнить цены

[Вернуться назад](#)

Рисунок 1.5 – Онлайн-калькулятор страхового сервісу Parasol.ua

Вищевказаний приклад також має свої недоліки, адже результатом такого калькулятору є перелік полісів та ціна за пакет (рисунок 1.6).

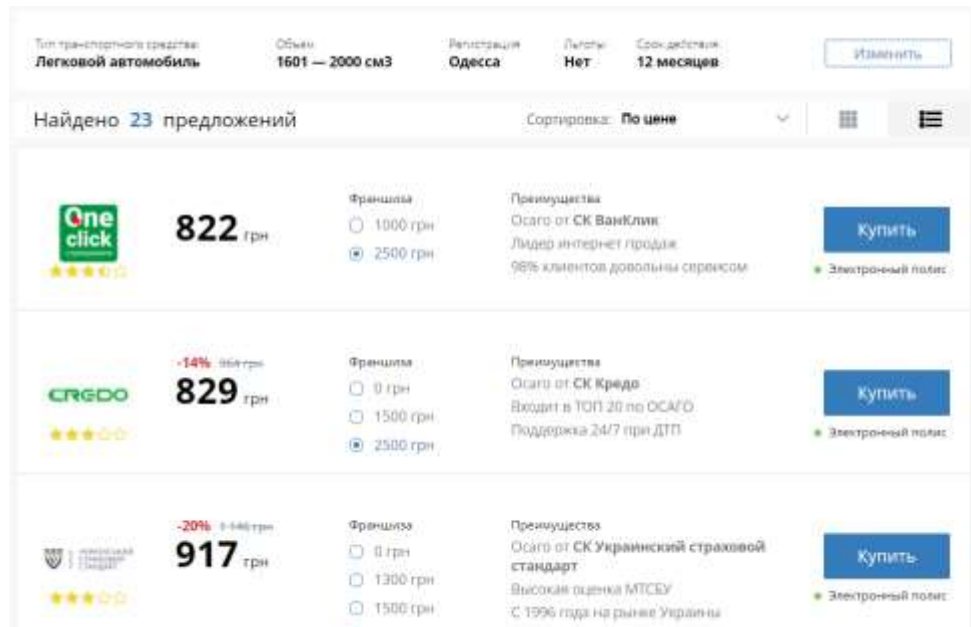


Рисунок 1.6 – Онлайн-калькулятор страхового сервису Parasol.ua

Тобто користувач для обрання страхового тарифу повинен все одно шукати його умови на сторінках компаній. Даний калькулятор не підходить клієнтам, яким важливий перелік умов обраного ними страхового пакету.

1.3 Постановка задачі

Після проведення аналізу предметної області задачі дослідження, було розглянуто вже існуючі підходи вирішення даної проблеми. На основі цієї інформації можна провести таку формалізацію постановки даної задачі:

1. Дослідження відомих підходів до вирішення задачі вибору найоптимальнішого страхового пакету;
2. Аналіз даних про клієнтів та їх вибір страхових пакетів;
3. Розробка алгоритму вирішення даної задачі;
4. Проаналізувати результати;
5. Розробити програмний продукт з зручним для користувача інтерфейсом.

1.4 Висновки до розділу 1

У даному розділі розглянута актуальність автоматизації підбору страхового полісу, розглянуто існуючі підходи до вирішення цієї задачі, приведені приклади проблем з якими зараз стикаються клієнти онлайн-страхування. У результаті проведення аналізу був сформований напрям роботи.

Окремим пунктом розділу було виділено формалізацію задачі, завдяки чому були описані напрямки та пріоритети роботи над дослідженням.

РОЗДІЛ 2 АВТОМАТИЗАЦІЯ БІЗНЕС-ПРОЦЕСІВ ТА МЕТОДИ РОЗВ'ЯЗАННЯ ЗАДАЧ НА ОСНОВІ АНАЛІЗУ ДАНИХ

2.1 Необхідність автоматизації бізнес-процесів

Управління бізнес-процесами – це спосіб управління підприємством, діяльність якого можна розглядати як множину пов'язаних між собою процесів, які спрямовані на реалізацію кінцевого продукту, який має цінність для клієнта і є джерелом доходу підприємства. Головною різницею від інших способів управління є те, що результат роботи обраного способу визначається за спільним результатом, який отриманий у процесі виконання усіх частин роботи, а не за якісною характеристикою роботи окремих функцій. Управління бізнес-процесами досягається за допомогою зменшення взаємозв'язків між функціональними підрозділами.

Отже, можемо сказати, що основною метою BPM (business process management) є збільшення ефективності та оптимізація системи управління.

На початку свого розвитку, поняття автоматизація відносилось більше до виробничої діяльності. Адже з розвитком підприємств збільшувався перелік можливих процесів. Стало очевидним, те що зі збільшенням організації збільшується і кількість працівників, відповідно зростає можливість помилки людського фактору. Зараз під автоматизацією можна розглядати реалізацію та впровадження сучасних технологій, які допоможуть зменшити людський фактор, спростити ведення бізнес-процесів реалізацію виконання однотипних задач відповідного чіткого алгоритму. Автоматизація управління збільшує загальну ефективність всього підприємства.

Існує низка переваг у використанні технологій в управлінні бізнес - процесами:

1. Зменшення рутинної ручної праці;
2. Пришвидшення процесу обробки даних;

3. Єдиний формат ведення звітності;
4. Контролювання та швидкий аналіз даних;
5. Зменшення людських помилок;
6. Зменшення витрат завдяки підвищенню ефективності;

Отже, дана опція зможе допомогти зменшити витрати підприємства та підвищити ефективність. Програмний продукт виконує багато процесів замість працівників і зменшує кількість помилок людей, що в подальшому мінімізує збитки. Впровадження автоматизації бізнес-процесів допомагає бути конкурентоспроможними в наш час.

2.2 Основи аналізу даних для автоматизації вибору страхового плану

Інтелектуальний аналіз даних - це процес узагальнення та обробки даних, аналіз їх з різних ракурсів і являє собою процес аналізу та узагальнення даних з різних точок зору і конвертація всіх даних в корисну для аналізу інформацію. Ми можемо використати як дані цифри, тексти, факти, графіки. В основному, інтелектуальний аналіз даних використовується для знайдення певних закономірностей та кореляцій, виділення та класифікація потрібної інформації (рисунок 2.1). Задачі інтелектуального аналізу даних можуть мати різну характеристику, бути задачами прогнозу та описуючими. Зазвичай, такі задачі тісно пов'язані з машинним навчанням. Вони можуть бути задачами кластеризації, класифікації, аналіз тенденцій та асоціювання.

Кластеризація – це процес ділення вибірки, яка складається з певних даних на підмножини, які також часто називають кластерами. Кожен кластер містить в собі об'єкти, які схожі один на один, а об'єкти які віднесли в різні кластери дуже різні між собою.

Класифікації – це метод аналізу даних, який оцінює належність об'єктів до певних класів, що залежить від певних.

Асоціювання - це визначення нових зв'язків пов'язаних об'єктів. Об'єднання таких об'єктів називається асоціативним правилом - наприклад, якщо один вигляд множини об'єктів в базі даних дуже схожі з іншим набором об'єктів, то такі дві множини об'єктів називають асоційованими.

Аналіз тенденцій – це виявлення нових кореляцій, загальних властивостей даних, які є змінними в залежності від часу. Інтелектуальний аналіз даних знаходить існуючі взаємозв'язки в даних.

Аналіз відхилення – виявлення аномальних та найбільш нестандартних даних.

Процес створення моделі інтелектуального аналізу має такі етапи:

1. Первинний аналіз даних;
2. Перевірка усіх даних на повноту, та виділення аномалій;
3. Аналіз та факторів впливу, виявлення кореляції;
4. Виділення винятків;
5. Виділення та виявлення основних категорій;
6. Аналіз сценаріїв;
7. Прогнозування.

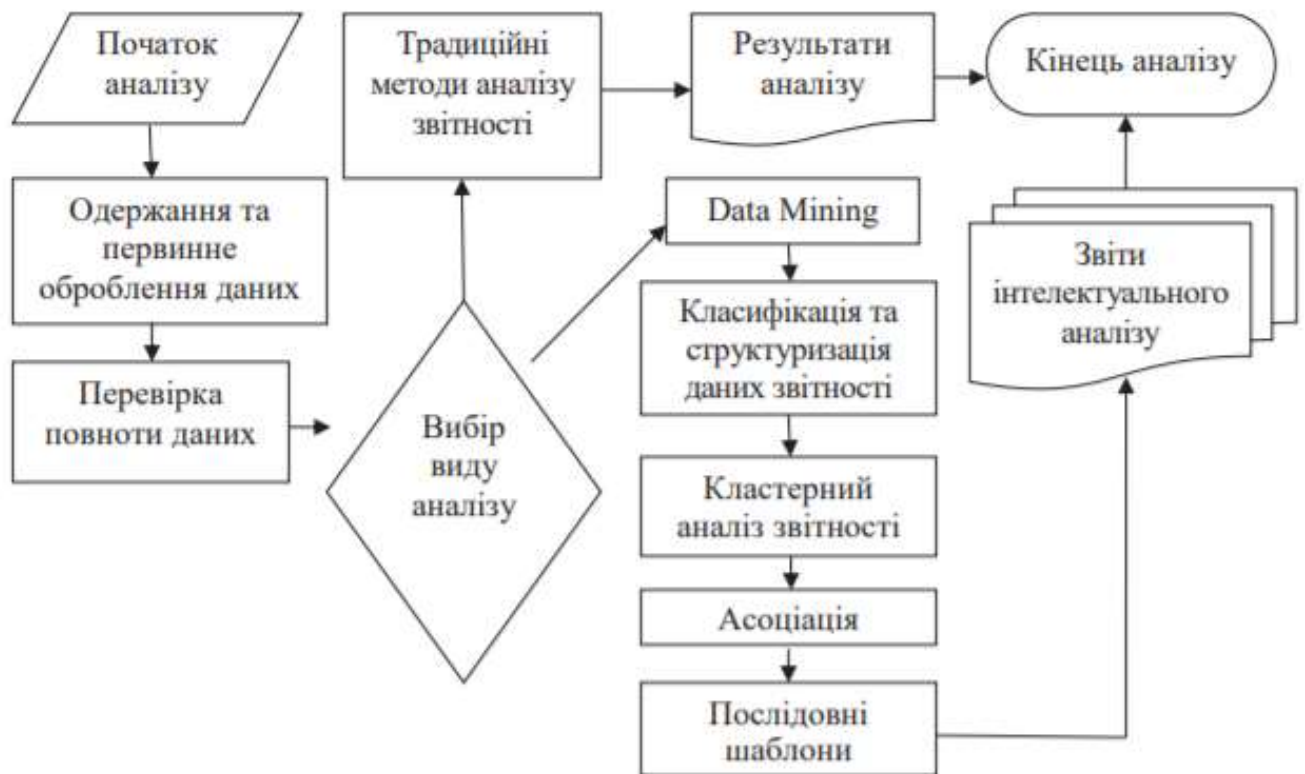


Рисунок 2.1 – Алгоритм аналізу даних методами інтелектуального аналізу даних

Часто для графічного зображення даних, які ми використовуємо для навчання, використовується візуалізація. Вона дозволяє побачити в процесі аналізу аномальні дані, розгледіти основні тенденції та структури. Для вирішення задачі візуалізації даних використовують графічні методи. Іноді візуалізацію розглядають як метод аналізу даних - Visual Mining.

У процесі інтелектуального аналізу даних досліджується велика кількість об'єктів. Вони можуть бути представлені у вигляді таблиці.

2.2.1 Аналіз первинних даних та виявлення аномалій

Для оцінки зв'язку між випадковими величинами у базі даних використовуються стандартні статистичні показники: коефіцієнт кореляції, коефіцієнт розсіювання, коефіцієнти детермінації.

Кореляцією називають статистичний взаємозв'язок випадкових величин. Кореляція може бути як додатною чи від'ємною, так і рівною нулю. Якщо кореляція додатна – між випадковими величинами є прямо пропорційний зв'язок (зі збільшенням одної величини збільшується інша). Якщо кореляція від'ємна – то зі збільшенням одної величини, інша зменшується. Між незалежними величинами кореляція відсутня.

На практиці для дослідження кореляції використовують коефіцієнт кореляції r_{xy} , що описує взаємозв'язок двох величин X та Y . Він може мати значення від -1 до 1 включно. Відповідно від'ємний коефіцієнт кореляції - від'ємна кореляція, в свою чергу додатний коефіцієнт кореляції відповідає додатній кореляції. Якщо значення коефіцієнту кореляції приблизно дорівнює нулю, то говорять, що з великою ймовірністю величини не пов'язані між собою.

Розглянемо датасет, який буде використаний в даній роботі для навчання моделі. Згаданий датасет включає в собі дані про клієнта, який звертався за допомогою до страхової компанії з метою страхування свого життя. Цей датасет містить в собі таку інформацію про користувача: вік, стать, BMI (Body mass index), наявність у клієнта звички палити, кількість дітей та виплати.

Як можна помітити (рисунок 2.2), люди з підвищеним індексом маси тіла та люди, які палять частіше отримують страхові виплати. Інші дані мають слабку кореляцію з кількістю виплат.

Також для дослідження взаємозв'язків використовують діаграму розсіювання. Такий тип графіків дозволяє візуально побачити кореляцію або її відсутність. Наприклад, (рисунок 2.2) можна побачити дві кількісні величини з коефіцієнтом кореляції, близьким до нуля, наприклад куріння та кількість дітей. Отже, скоріш за все зміна, яка відповідає за куріння не пояснюється впливом кількості дітей.

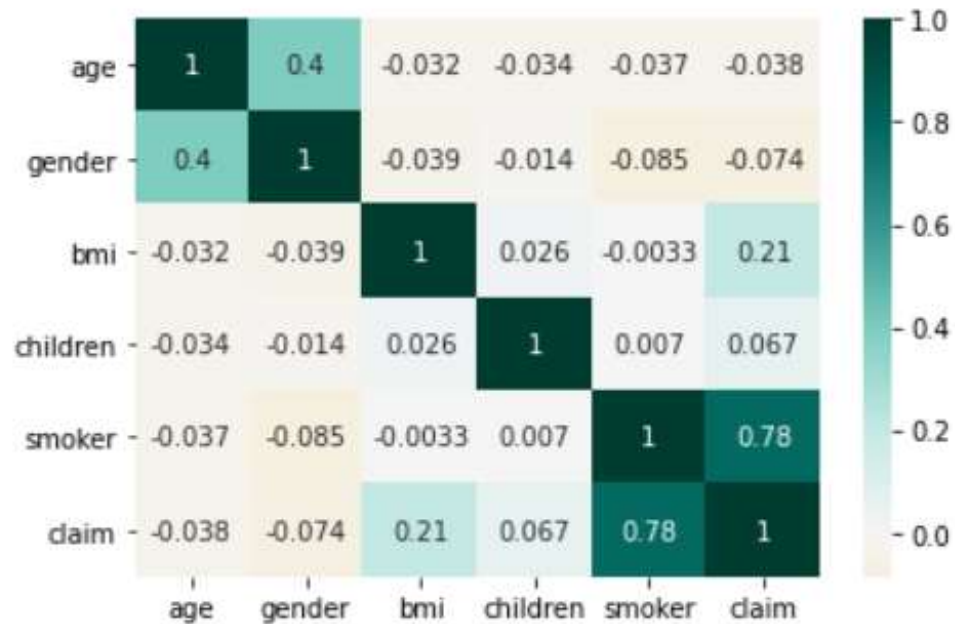


Рисунок 2.2 – Матриця кореляцій для датасету «Страхування життя»

Показник R^2 показує якою мірою дисперсія однієї величини обумовлена впливом коефіцієнта детермінації іншою змінної. З математичної статистики, знаємо що R^2 дорівнює квадрату коефіцієнта кореляції. Даний показник приймає значення від 0 до 1 включно.

При передбаченні значення величини, значення коефіцієнту детермінації допомагає зробити оцінку наскільки якісно було проведено передбачення. Чим значення R^2 ближче до одиниці, тим ближче наші передбачені значення до реальних.

2.3 Завдання аналізу даних для автоматизації вибору страхового плану

Важливим завданням інтелектуального аналізу даних є пошук нетривіальних шаблонів (рисунок 2.3). Тобто знайдені моделлю шаблони мають показувати неочевидні регулярності в датасеті, або приховані регулярності.

Як основні завдання ми можемо виокремити:

- Кластеризацію (Clustering);
- Класифікація (Classification);
- Регресія (Regression);
- Асоціація (Association);
- Прогнозування (Forecasting);
- Визначення відхилень та викидів (Deviation Detection);
- Зменшення розмірності (Dimensionality reduction).

Найчастіше використовуються у вирішенні задач кластеризації як метод k-серединіх, ієрархічний метод, та метод DBSCAN (Density-based spatial clustering of applications with noise) .

Метод кластеризації вирішує такі задач як розподілення клієнтів супермаркету на групи по середньому розміру чеку, виділення сегменту ринку за показником про послугу або визначення групи ризику пацієнтів за заданою ознакою.

Класифікацію можна представити як задачу розпізнавання образів, адже для кожного об'єкта знаходяться свої характерні ознаки. В класифікації текстів – такими ознаками є слова, словосполучення, які є у даному тексті. Такі ознаки виділяються у таблицю, яка називається інформаційною матрицею. Кожен рядок даної таблиці розглядають як окремий клас, окремі елементи рядка розглядають як ознаки цього класу, за чисельним значенням цих елементів відбувається процес навчання системи. Як тільки навчання було завершено, з урахуванням усіх вагових коефіцієнтів можна встановити клас нового тексту.

Найчастіше використовуються у вирішенні задач класифікації методи найближчого та k-найближчого сусіда, дерев рішень, нейронні мережі, байєсовські мережі.

Класифікація вирішує ряд важливих задач таких як задачі кредитного скорингу, класифікація повідомлень, що надходять до електронної скриньки (спам чи ні) та багато інших.

Під час вирішення задачі регресії встановлюється залежність між одною чи декількома змінними. Ключовим завданням є передбачення певних числових значень. Тобто потрібно побудувати алгоритм, який по вектору ознак повертає би оцінку значення регресії, довірчий інтервал чи апостеріорне розподілення на множині значень регресійної змінної.

Найчастіше використовуються у вирішенні задач регресії метод опорних векторів, лінійна регресія, логістична регресія, множинна регресія, регресія дерева рішень та інші методи.

Регресія вирішує такі задачі: ціна на квартиру в залежності від таких параметрів як відстань від метро, розмір квартири, рік побудови будинку та інші, дохід магазину на наступний місяць, враховуючи дані за минулі місяці, якість вина при тестуваннях та інших.

Під час вирішення задачі асоціації знаходяться закономірності між пов'язаними явищами чи подіями в наборі даних. Пошук закономірностей здійснюється на основі декількох подій, що відбуваються одночасно.

Найчастіше у вирішення задач такого типу – пошуку асоціативних правил, використовують алгоритм Apriori.

Типовим прикладом використання даної задачі можемо назвати аналіз покупок, тобто необхідно знайти залежності між транзакціями. При розгляданні аналізу покупок доцільно використовувати для знаходження асоціацій – вміст кошика в супермаркеті. Основною метою є знайдення трендів серед усієї кількості транзакцій, що показують поведінку покупців. Ця інформація використовується для регулювання кількості продукції, розміщення товарів, проведення нових рекламних кампаній.

Для вирішення задачі прогнозування будуються на базі особливостей і характеристик історичних даних. Прогнозування направлено на визначення основних характеристик конкретного об'єкта або процесу на основі ретроспективних даних (даних, що використовують для аналізу об'єкта в

минулому). Отже, зрозуміло, що рішення задачі прогнозування вимагає навчання на деякому наборі даних.

Як приклад задач прогнозування можемо назвати: прогнозування попиту на певний товар чи послуг, прогнозування врожайності агрокультури, прогнозування ризику виникнення технічних неполадок чи надзвичайних ситуацій.

Даний аналіз використовується майже для будь-яких задач і пов'язаний з очисткою даних. Цей аналіз є пошуком рідких даних, об'єктів чи логік, які є нетиповими для аналізуючого бізнес-процесу або моделі, яка його описує.

Задачею такого аналізу є визначення аномалій в даних, формування правил, які пояснюють причини з'явлення нестандартних ситуацій.

Популярними методами вирішення таких задач є метод опорних векторів, виявлення викидів на базі кластерного аналізу, штучні нейронні мережі та метод k-найближчих сусідів.

До таких задач ми можемо відвести пошук та визначення злочинних дій з кредитними картами чи страховими полюсами, або ж фрод, виявлення клієнтів, яким можна запропонувати сервіс на інших умовах.

Для вирішення задачі зменшення розмірності використовують скорочення великої кількості випадкових змінних — ознак до меншого, зазвичай до 3 ознак, для зручності їх візуального подання, стиснення даних. Отже, це процес можна поділити на видалення змінних та обрання більш важливих. Найчастіше обирають такі стратегії для стиснення даних як фільтрування даних, обгортання чи вкладення.

Методи, які зазвичай використовують для зменшення розмірностей, це метод головних компонент, метод лінійного роздільного аналізу, метод розкладення невід'ємних матриць та метод k-найближчих сусідів.

До задач такого виду можемо відвести пошук подібності у потоках відео, виділення головних ДНК даних.

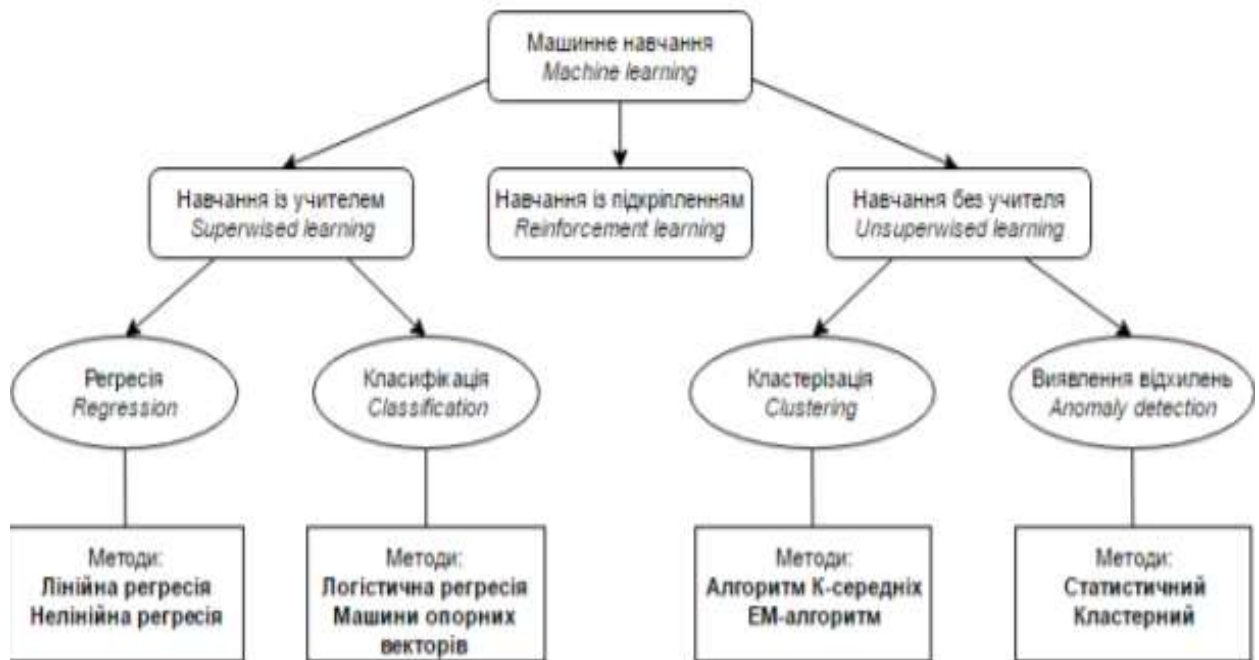


Рисунок 2.3 – Базова класифікація задач машинного навчання

Окрім основних завдань існують ще задачі, які передують вищезгаданим задачам такі як: оцінювання об'єктів (Estimation), аналіз зв'язків (Link Analysis).

Автоматизація вибору страхового плану є складною задачею, яка включає в собі задачу регресії – знаходження найкращої ціни для страхового пакету в залежності від вхідних даних про клієнта.

2.4 Методи та критерії якості роботи моделі

Розглянемо детальніше методи вирішення задач регресії за допомогою машинного навчання та розглянемо критерії якості роботи моделі (рисунок 2.4).

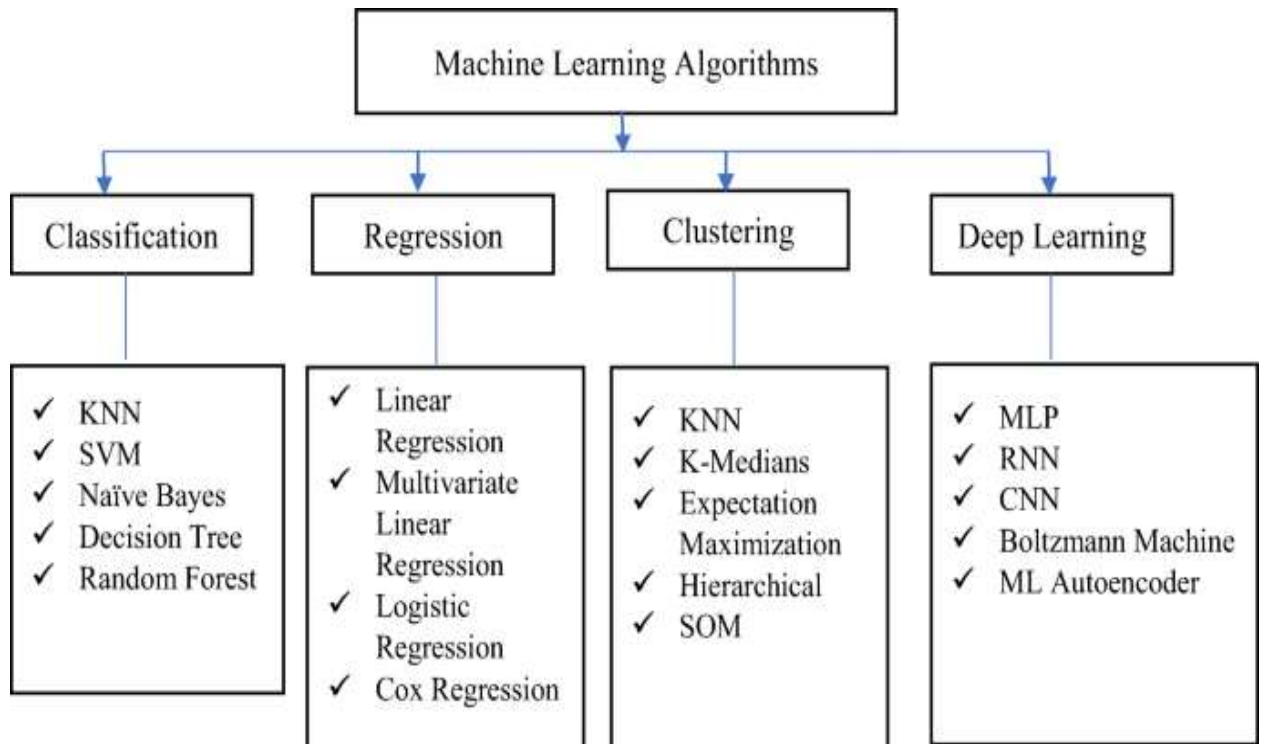


Рисунок 2.4 – Класифікація машинного навчання та методів

Нагадаємо, що сама по собі задача регресії подібна до задачі класифікації, але головною відмінністю є те, що вихід задачі регресії має безперервну та нескінченну область. Часто це керується контрольованим навчанням.

2.4.1 Лінійна регресія

Лінійна регресія часто використовують як основний використовуваний тип прогнозного аналізу. Ідея даного методу полягає в дослідженні основних двох пунктів:

- як набір величин справляється з прогнозуванням результатної (залежної) змінної;
- які змінні є найбільш впливовими на прогноз досліджуваної змінної і як ці змінні впливають на зміну результату.

Оцінки регресії показують і пояснюють зв'язок між змінними, а саме між залежною величиною та декількома незалежними величинами. Якщо ми

маємо залежність цільової випадкової величини та множини параметрів X_i , ми можемо говорити про множинну лінійну регресію. Множинну лінійну регресію можна подати у вигляді рівняння виду

$$Y = a + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_n \cdot x_n, \quad (2.1)$$

де X_i - параметри;

b_i – параметри (коефіцієнти);

a - це константа рівняння.

Метод лінійної регресії полягає у підбиранні таких параметрів, що дозволяють використовувати рівняння (2.1) для побудови прогнозування цільової випадкової величини. Для підбору параметрів даного рівняння зачасто використовують саме метод найменших квадратів. Так як ми працюємо з моделлю параметризованою і в процес намагаєтесь підібрати таку модель, що дозволить отримати найточніші прогнози (рисунок 2.5).

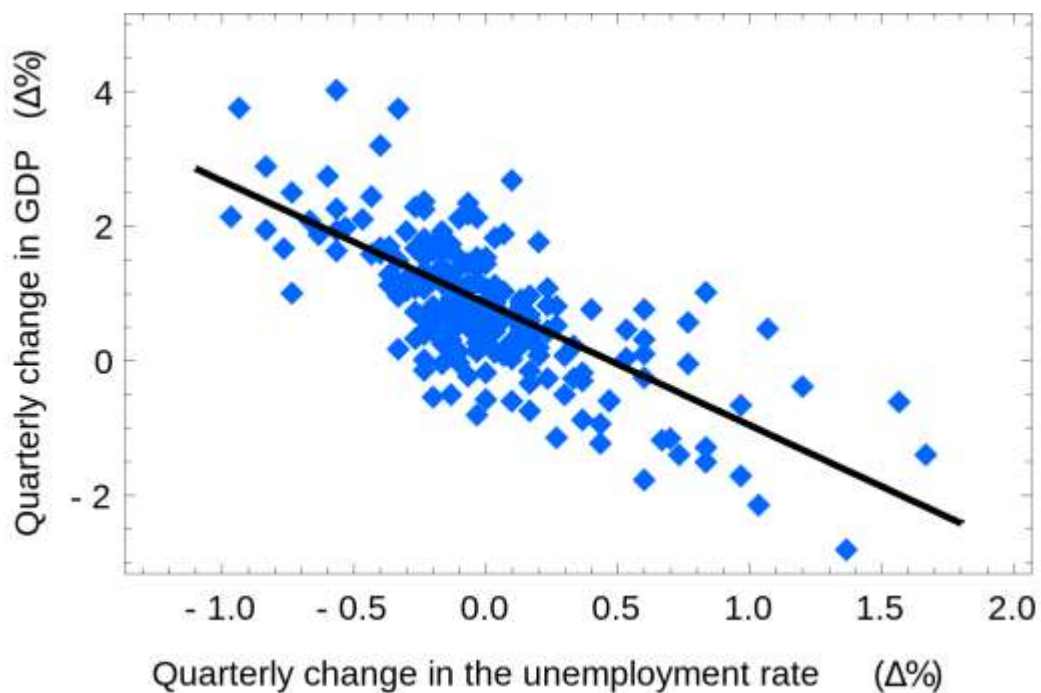


Рисунок 2.5 – Приклад простої лінійної регресії

Лінійная регресія для оптимізації та підбору параметрів рівняння користується методом найменших квадратів. Сама лінійна регресія при роботі

з нелінійними даними, має проблеми при роботі з нелінійними даними чи датасетом, які мають велику кількість залежностей. Отже, при роботі з лінійною регресією постає ряд проблем таких як мультиколінеарність, що не дозволяє використовувати модель з бажаною ефективністю.

Критеріями якості лінійної регресії є такі показники як коефіцієнт детермінації, середньоквадратична похибка моделі регресії, показник середньої похибки апроксимації, крива ROC.

Крива ROC - це графік, який зображає ефективність бінарної системи класифікаторів. ROC-аналіз оцінює та зображує графічне відображення роботи системи.. В основі методу лежить побудова ROC-кривої (ROC - receiver operating characteristic - робоча характеристика приймача) в свою чергу дана крива слугує гарним графічним методом аналізу, а коефіцієнт площі під кривою використовується для кількісної оцінки моделі.

Коефіцієнт детермінації (R^2) – найчастіше використовується для оцінки якості моделі. Даний показник обраховується за формулою:

$$R^2 = \frac{\sigma_{\text{perp}}^2}{\sigma_y^2}, \quad (2.2)$$

де σ_{perp}^2 - дисперсія, що показує регресія;

σ_y^2 - дисперсія помилок.

Середньоквадратична похибка моделі регресії (MSE) розраховується за формулою:

$$MSE = \sqrt{\frac{\sum_{i=1}^n x_i^2}{n - h}}, \quad (2.3)$$

де x_i – i -ий елемент вибірки;

h – кількість параметрів;

n - кількість елементів вибірки.

Якщо показник MSE є меншим за показник середньоквадратичного відхилення залежної змінної від модельного значення – модель вважають якісною регресійною моделлю.

Показник середньої похибки апроксимації розраховується за формулою:

$$\overline{A_y} = \frac{1}{n} \sum_{i=1}^n \frac{|y - \tilde{y}_i|}{y_i}, \quad (2.4)$$

де y – залежна змінна;

n – кількість елементів вибірки.

Якщо величина даного показника становить менше 0.07 то побудовану модель регресії вважають хорошою, а максимально припустимим відхиленням є 0.15.

Оскільки вартість страхового пакету не оцінюється лише за одним параметром, доцільно використати критерії оцінки якості моделі такі як: середньоквадратичну похибку та показник середньої похибки апроксимації, а також коефіцієнт детермінації, який перевіряє адекватність (відповідність) побудованої регресійної моделі дійсності.

2.4.2 Логістична регресія

Логістична регресія як метод машинного навчання використовує для передбачення логістичну функцію. Ідея логістичної регресії полягає у розділенні множини значень на області, які відповідають класам, що залежить від вимірності регресії. Фактично логістична регресія є алгоритмом класифікації, який відносить кожен об'єкт до одного з можливих класів.

Логістична функція, яку ще називають сигмоїдою, визначає дану регресію. Логістична регресія часто використовують як класичний метод для вирішення задач регресії або класифікації. Цей метод активно використовують

і його можна представити у вигляді одношарової нейронної мережі, де функцією активації є сигмоїда, вагами є коефіцієнти регресії, а логістичної регресії, а вага поляризації - константа регресійного рівняння. Одношарова нейронна мережа може вирішувати питання логістичного ділення. Завдяки чому можемо зробити висновок, що моделювання нелінійних залежностей неможливе для логістичної регресії. Для оцінки якості такої моделі використовують ROC-аналіз, а для обрахування коефіцієнтів логістичної регресії зазвичай використовують градієнтні методи: метод сполучених градієнтів, методи змінної метрики і інші.

Фактично логістична регресія використовується скоріше для оцінки ймовірності прийняття конкретного значення, а ніж для передбачення значення залежної змінної. Цю ймовірність можна обчислити за формулою:

$$P(y = 0) = 1 - p_i, \quad (2.5)$$

$$P(y = 1) = p_i, \quad (2.6)$$

А сама модель, яка визначає ці ймовірності, називається логістичної регресійної моделлю і має вигляд:

$$\log\left(\frac{p_i}{1 - p_i}\right) = a + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_n \cdot x_n, \quad (2.7)$$

де X_i - параметри;

b_i – параметри (коефіцієнти);

a - це константа рівняння;

p_i - ймовірність прийняття значення.

Бачимо, що права частина цього рівняння схожа до рівняння лінійної регресії (2.1), а ліва частина має вигляд вихідної змінної і в ній є логарифм

відношення ймовірностей двох взаємо виключних подій, формули (2.5) та (2.6). Цю функцію також називають логіт-перетворенням.

При роботі з логістичною регресією ми налаштовуємо відповідні параметри. При роботі з даною логістичною-регресією є некорельовані та нормального розподілу помилок нашої класифікації. Також рівняння, що буде підганятися під нашу поверхню, вже не матиме такого простого вигляду.

Для вирішення задач регресії за допомогою логістичної регресії використовується функція максимальної правдоподібності (рисунок 2.6). Отже, бачимо що оцінка коефіцієнтів рівняння регресії можна звести до максимізації ймовірності появи вибірок. Тому можемо спостерігати низький відсоток правильних передбачень.

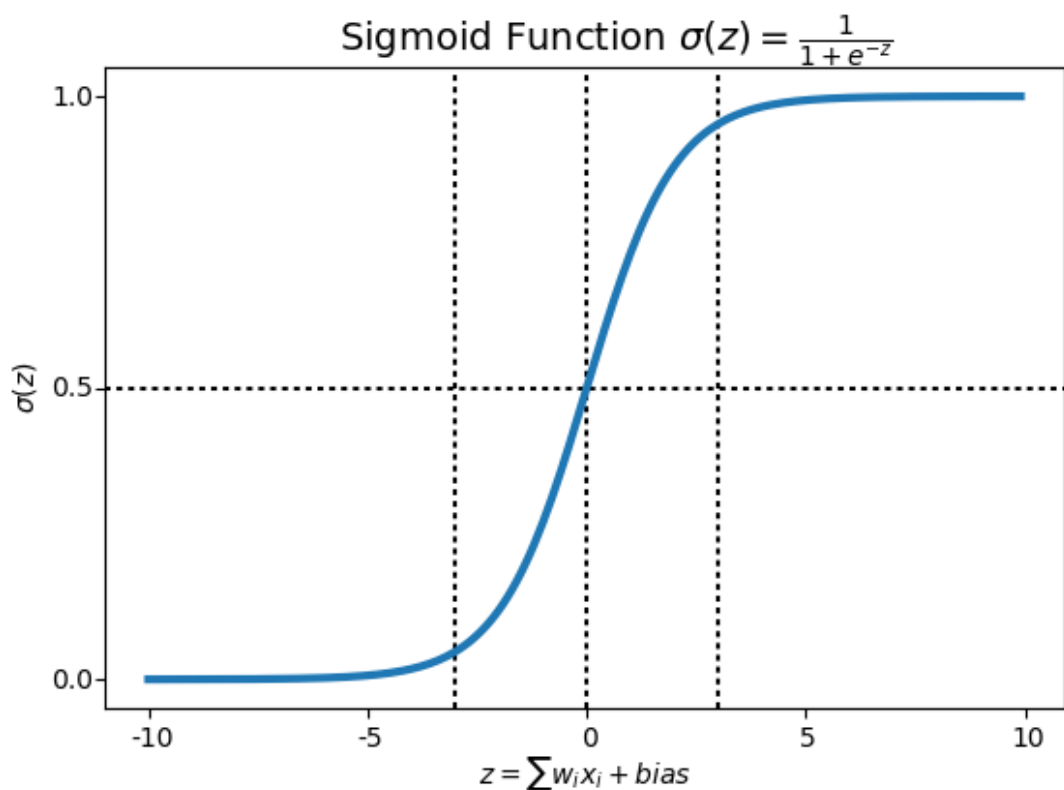


Рисунок 2.6 – Приклад логістичної функції

Можемо помітити, що логістична регресія може бути застосована для оцінки ймовірностей визначення класу об'єкта. Також цей тип регресії дає

можливість виявити значення параметрів та отримати інформацію на рахунок їх важливості, що є дуже важливим для методів машинного навчання для роботи задачі регресії.

Для критерій якості логістичної регресії можна використовувати такі критерії, що і для лінійної регресії.

Переваги даного методу полягають в тому, що моделі регресії є швидкими і можуть працювати з великим набором даних, коефіцієнти перед ознаками можуть інтерпретуватися.

Як недолік вони погано працюють для задач в яких відповідь нелінійна та складна.

2.4.3. Дерева рішень

Метод дерев рішень (decision trees) називають одним з популярних методів розв'язання таких задач як класифікація, регресія та прогнозування. Якщо цільова величина повинна приймати дискретні значення, за допомогою методу дерева рішень можна розв'язати задачу класифікації. Якщо ж цільова величина може приймати також і неперервні значення, то дерево рішень повинно дослідити залежність між залежними і незалежними величинами і розв'язувати задачу регресії. Також результатом роботи може бути і визначення значення категоріальної змінної.

Основа алгоритму дерев рішень в тому, що під час роботи на кожному кроці відбувається ділення по одному з критеріїв оцінки даних (рисунки 2.7). Таке розбиття даних відбувається щоб збільшити кількість інформації, при цьому зменшити ентропію в попередньо утворених групах за ознаками.

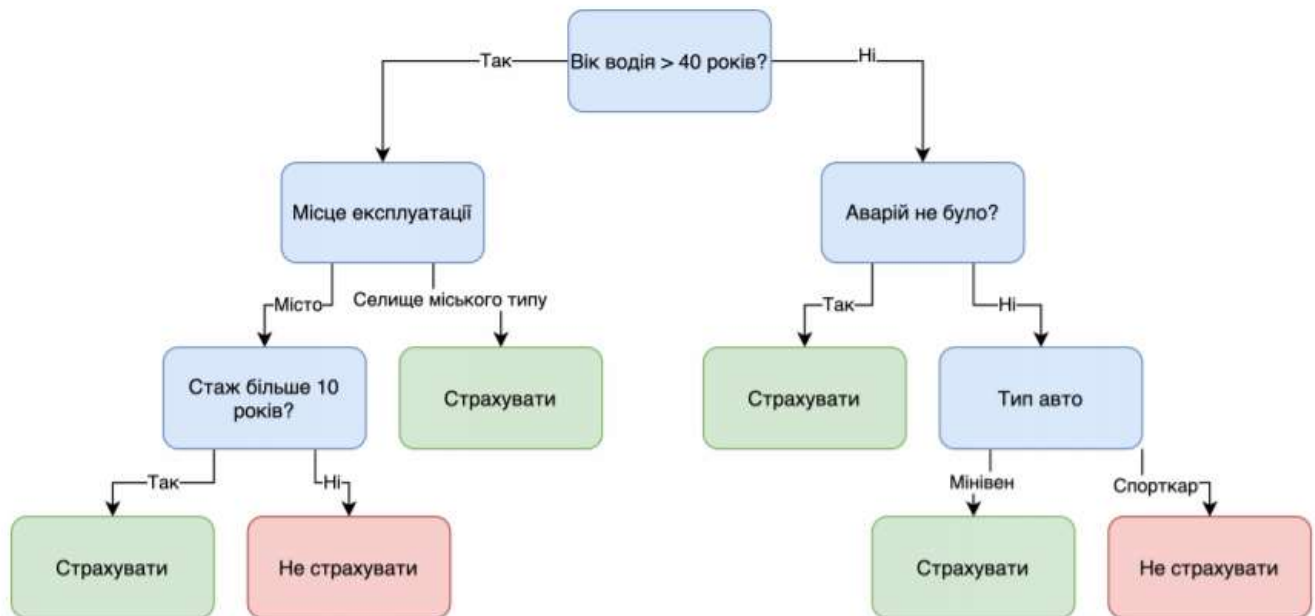


Рисунок 2.7 – Приклад структури дерева рішень

Саму ентропію ми можемо обрахувати за наступною формулою :

$$S = - \sum_{i=1}^N p_i \log_2 p_i, \quad (2.8)$$

де p_i – співвідношення i -го значення до загальної кількості прикладів, що потрапили у дане розбиття.

В свою чергу приріст інформації обчислюється наступним чином:

$$IG(Q) = S_0 - \sum_{i=1}^q \frac{N_i}{N} S_i, \quad (2.9)$$

де N_i – число елементів вибірки, у яких ознака Q має i -те значення.

В випадку задачі регресії критерієм якості стає дисперсія навколо середнього:

$$D = \frac{1}{\ell} \sum_{i=1}^{\ell} \left(y_i - \frac{1}{\ell} \sum_{i=1}^{\ell} y_i \right)^2, \quad (2.10)$$

де ℓ – кількість об'єктів в листку;

y_i – значення цільової ознаки.

Перевагами даного методу є інтуїтивність та легкість інтерпретації користувачем, точність методу, обробка великої кількості вхідних змінних без їх видалення, оцінка важливості змінної в класифікації, забезпечення ефективності методу, надання методів збалансування похибки в незбалансованих наборах даних, також даний метод є стійким до викидів.

В свою чергу є певні недоліки, такі як схильність до перенавчання.

2.4.4. Випадковий ліс (Random forest)

Random forest – метод навчання для задач класифікації та регресії, який будує великої кількості дерев прийняття рішень під час тренування моделі і створюють певну моду всередині класів або прогноз для побудованих дерев.

Метод випадкового лісу можна назвати поєднанням предикторів дерев, в яких кожне дерево залежить від значення випадкового вектора, відібраного незалежно, з однаковим розподілом для всіх дерев у лісі. Основний принцип полягає в тому, що група слабких дерев може скластися так, щоб сформувати сильне дерево.

Поодинокі дерева з одним рішенням зачасту мають велику дисперсію чи велику зміщеність. В свою чергу випадкові ліси намагаються пом'якшити проблеми з дисперсією та упередженістю. Зважаючи на те, що у випадкових лісах є кілька параметрів, які можна налаштувати, і їх можна просто використовувати з налаштуваннями параметрів за замовчуванням, вони є простим інструментом для використання без наявності моделі або для створення розумної моделі швидко та ефективно.

Перевагами даного методу є нечутливість до масштабування, здатність працювати паралельно, обробляють дані з великою кількістю параметрів та ознак, однаково добре обробляють дискретні та неперервні дані.

Недоліки даного методу полягають в схильності до перенавчання, і розмірі моделей, тобто якщо датасети мають велику кількість шумів, для зберігання моделі треба використовувати багато пам'яті.

2.5 Висновки до розділу 2

У даному розділі були представлені основні задачі та приведена аргументація автоматизації управління бізнес-процесами, проведено аналіз задачі регресії і розглянуто основні методи для вирішення даної задачі такі як: лінійна регресія, логістична регресія, дерева рішень та випадковий ліс.

Також у цьому розділі були сформовані і визначені основні критерії якості роботи даної системи автоматизації управління бізнес-процесами. Був проведений основний аналіз методів розв'язання задачі регресії.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ ДЛЯ ВИБОРУ ОПТИМАЛЬНОГО СТРАХОВОГО ПЛАНУ

3.1 Обґрунтування вибору платформи та мови програмування

Обрана задача автоматизації вибору найоптимальнішого страхового плану потребує методів інтелектуального аналізу даних та методів машинного навчання.

Для розробки функціональної частини програмного продукту, а також роботи з даними було обрано використовувати мову програмування Python та вбудовану бібліотеку машинного навчання `scikit-learn` (`sklearn`).

Мова програмування Python має низку переваг таких як висока продуктивність розробки, логічність та лаконічність, Python інтерпретована мова, тому дає можливість запускати програмний продукт на будь-яких операційних системах. Також великою перевагою є те, що дана мова має велику якісну базу бібліотек. Також мова програмування Python є найпопулярнішою мовою для машинного навчання та інтелектуального аналізу даних.

Однією з основних бібліотек для роботи з даними є `scikit-learn`. Ця бібліотека більшою мірою використовується в роботі, адже має велику кількість алгоритмів для навчання з учителем і без. У згаданій бібліотеці також є основні алгоритми машинного навчання: лінійної регресії та її модифікацій, метод опорних векторів, дерев рішень і випадкових лісів і ін. Отже, вказана бібліотека має можливість вирішити задачу регресії.

Ще однією перевагою у виборі інструментів Python стала швидкість роботи, значно переважає над інструментами написаними мовою Java.

Кінцевий програмний продукт має вигляд прикладного програмного інтерфейсу (API). Вибір був зроблений через те, що API може бути використаний на будь-якому веб-сайті.

Для створення API був використаний Django REST Framework. Взаємодія між клієнтським додатком та серверною частиною забезпечується за допомогою RESTful архітектури та HTTP запитам. REST (Representational state transfer) – це архітектура мережевих протоколів, які надають доступ до інформаційних ресурсів. GET, POST, PUT, DELETE – HTTP запити, які необхідні для створення RESTful сервісів. Цей веб-фреймворк використовується разом з Django фреймворком та надає весь необхідний функціонал для побудови якісного програмного продукту. Django був обраний через його популярність, наявність детальної документації та всього необхідного функціоналу.

Для інтерфейсу програмного продукту, а саме, веб-продукту були використано HTML+CSS+JavaScript. Для розробки веб-додатку був використаний фреймворк Bootstrap. Головною перевагою даного фреймворку є можливість створювати якісні макети, швидкість веб-розробки, кросплатформеність та наявність нових компонентів, завдяки яким був розширений арсенал візуальних складових веб-сторінки.

3.2. Збір та аналіз даних та навчання моделі

Збір інформації для бази даних був зроблений завдяки веб-скрейпінгу. Веб-скрейпінг – це спеціальна технологія, яка допомагає автоматично отримувати необхідну інформацію з веб-сайтів. У даній роботі функцію пошуку даних для подальшого аналізу робить Python-бібліотека «requests». Дана бібліотека підтримує HTTP – запити і допомагає вивантажити і завантажити дані на веб-сервіс, також вона підтримує проху.

За допомогою вищезгаданої бібліотеки та API сайтів страхових компаній було досліджено та зібрано список характеристик для кожного виду страхування. Для навчання моделі регресії було використано дані з

спеціального ресурсу kaggle. Вся зібрана інформація та сформовані датасети мають формат csv.

База даних складається з 6 таблиць зв'язаних між собою (рисунок 3.1).

Головна таблиця Client містить в собі важливу інформацію про клієнта: ім'я клієнта, прізвище клієнта, нікнейм клієнта в локальній системі, місто проживання клієнта, номер телефону, пароль. Також таблиця Client має поле id, що заповнюється автоматично.

Таблиця CarInsurance містить поля: місто реєстрації автомобіля, наявність іноземної реєстрації, тип рухомого майна, об'єм автомобіля, використання в цілях перевезення пасажирів, наявність додаткового захисту. Також таблиця CarInsurance має зв'язок з таблицею Client.

Таблиця PropertyInsurance містить поля: ціна нерухомого майна, площа помешкання, тип помешкання, період страхування, використання помешкання в цілях аренди житла, наявність додаткового захисту. Також таблиця PropertyInsurance має зв'язок з таблицею Client.

Таблиця MedInsurance містить поля: дата народження клієнта, стать клієнта, значення коефіцієнту маси тіла, наявність хвороби діабет або раку, кількість дітей, куріння, сума страхування, період страхування. Також таблиця MedInsurance має зв'язок з таблицею Client.

Таблиця TravelInsurance містить поля: дата відправлення та прибуття, територія на якій діє страхування, причина візиту, кількість поїздок за обраний період, кількість людей зазначених в контракті. Також таблиця TravelInsurance має зв'язок з таблицею Client

Таблиця LifeInsurance містить поля дата народження клієнта, стать клієнта, значення коефіцієнту маси тіла, наявність хвороби діабет або раку, кількість дітей, куріння, сума страхування, період страхування. Також таблиця LifeInsurance має зв'язок з таблицею Client.

Таблиці CarInsurancePlan, PropertyInsurancePlan, MedInsurancePlan, LifeInsurancePlan, TravelInsurancePlan мають подібну структуру. Наприклад,

таблиця PropertyInsurancePlan має поля: ціна страхового плану, перелік обмежень та умов страхового плану.

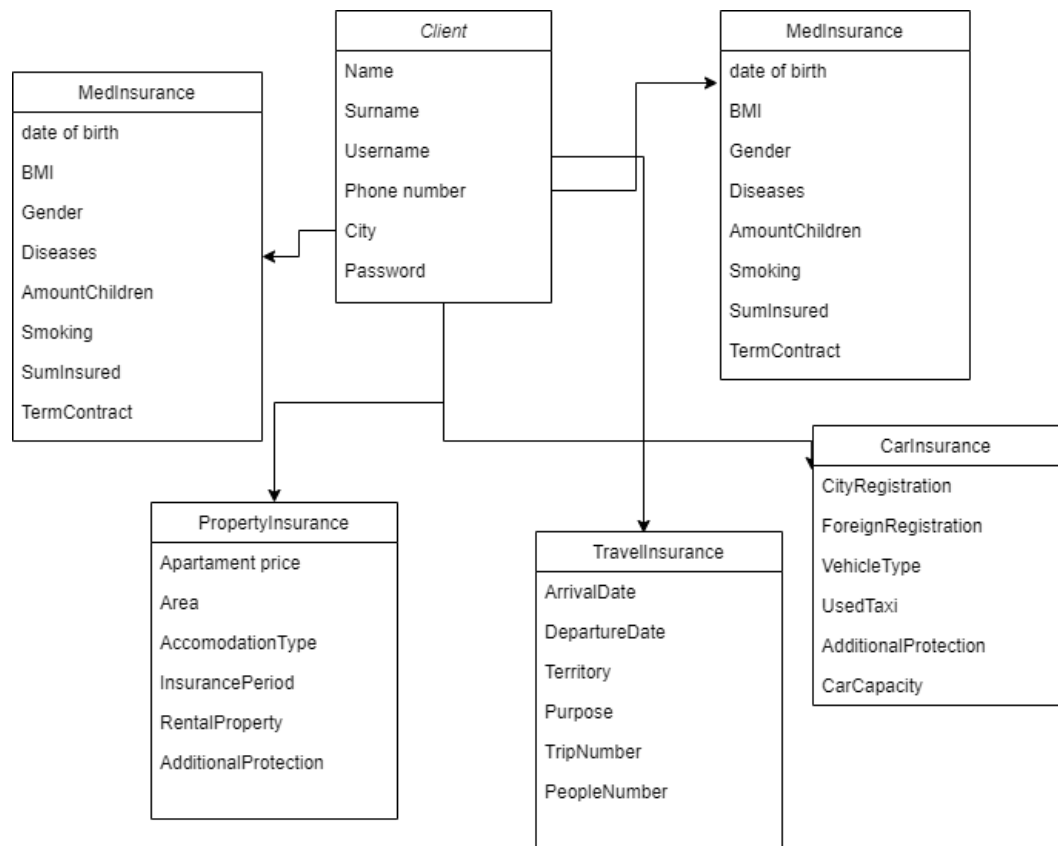


Рисунок 3.1 – База даних системи

Розглянемо навчання моделі на прикладі страхування рухомого майна. Датасетом для навчання моделі є csv файл з даними про машину: місто реєстрації машини, тип машини, об'єм машини, чи використовували даний автомобіль у якості таксі, чи має машина додатковий захист (наприклад відеореєстратор) та ціна страхового пакету (рисунок 3.2).

cityRegistration	foreignRegistration	typeVehicle	carCapacity	useTaxi	additionalProtection	Price
Poland	Yes	Passanger car	1 500,00	No	No	1350
Lviv	No	Passanger car	2 500,00	Yes	No	2120
Poland	No	Motorcycle	1 400,00	No	No	459
Kyiv	No	Electric car	1 350,00	No	No	1303
Rivne	No	Electric car	1 500,00	No	No	773
Kharkiv	No	Electric car	1 600,00	No	No	947
Poltava	No	Motorcycle	1 200,00	No	No	581
Chech	Yes	Passanger car	1 100,00	No	No	1499
Dnipro	No	Passanger car	1 900,00	Yes	No	2107
Vinnitsa	No	Passanger bus	3 800,00	No	No	1095
Kyiv	No	Passanger bus	3 500,00	No	No	1846
Odessa	Yes	Passanger car	1 500,00	Yes	Yes	651
Rivne	No	Electric car	1 600,00	No	No	588
Kyiv	No	Electric car	1 340,00	No	No	1358
Poltava	No	Truck	2 800,00	No	No	292
Kyiv	No	Passanger car	2 730,00	Yes	No	2907
Poltava	No	Truck	2 600,00	No	No	264
Dnipro	No	Passanger car	1 800,00	Yes	No	2654
Kyiv	No	Trailer	3 200,00	No	No	2895
Kyiv	No	Passanger bus	3 100,00	No	No	2391
Lviv	No	Passanger car	3 000,00	Yes	No	2748
Poland	No	Passanger car	1 600,00	No	No	2113

Рисунок 3.2 – Датасет для навчання моделі страхування рухомого майна

Дані, які використовуються для аналізу та навчання моделі повинні пройти очистку від аномальних даних або викидів. Також були видалені стрічки, які містили пропущені дані такі як «тип машини», «об'єм машини», «ціна». Якщо запис був пропущений в таких колонках «іноземна реєстрація», «додатковий захист» чи «використання як таксі», тоді в пусті клітинки було записано «No».

Так як датасет містить не тільки числові значення, а і поля типу `char`, такі поля ми будемо дискретизувати. Для полів «іноземна реєстрація», «використання як таксі», «додатковий захист» значення «No» замінимо на 0, а «Yes» на 1. Для поля «тип машини» категорії машини замінимо на відповідні значення:

- пасажирський автомобіль міняємо на значення 1;
- електромобіль міняємо на значення 2;

- мотоцикл міняємо на значення 3;
- автобус/ маршрутне таксі міняємо на значення 4;
- вантажівку замінимо на значення 5;
- трейлер замінимо на значення 6.

Після дискретизації та класифікації даних набагато простіше будувати модель для вирішення задачі регресії.

Побудуємо матрицю кореляцій (рисунок 3.3), яка демонструє зв'язки між величинами і їх вплив один на одного.

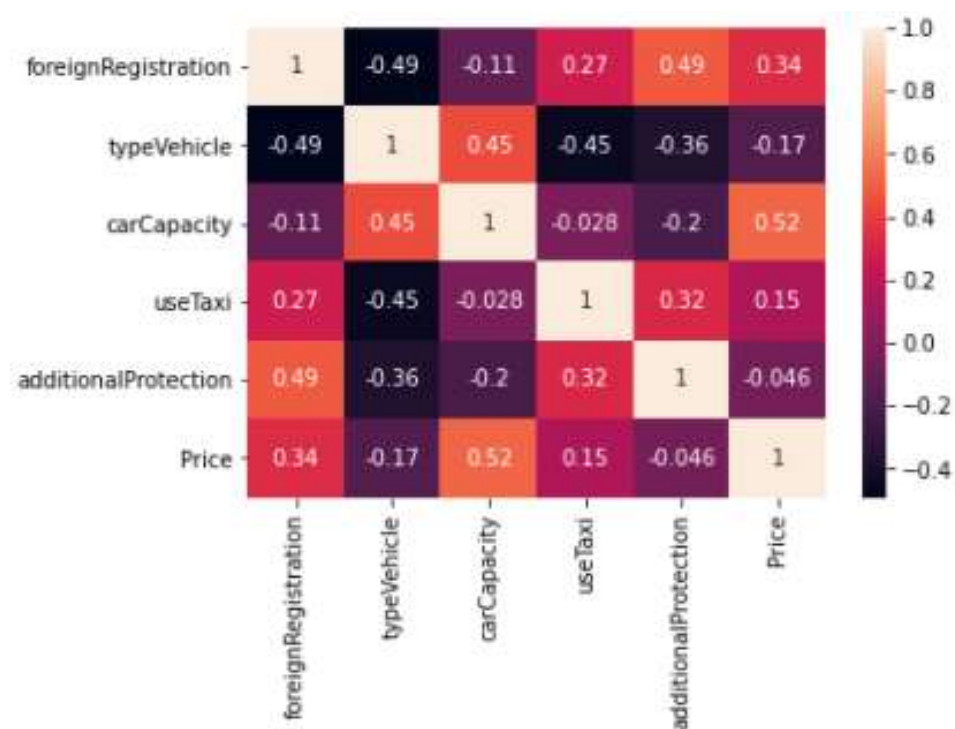


Рисунок 3.3 – Матриця кореляцій даних страхування рухомого майна

З матриці кореляцій бачимо, що такі параметри як використання як таксі, іноземна реєстрація зменшують ціну на страхування автомобіля, в той час як з розміром рухомого майна, додатковим захистом збільшується ціна страхового пакету.

Аналогічний аналіз даних був проведений для туристичного страхування, медичного страхування, страхування життя та страхування нерухомості.

За допомогою вбудованих python бібліотек побудуємо модель декількома методами: лінійна регресія, метод випадкового лісу.

```
from sklearn import metrics
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score,
max_error
#Linear Regression
linearReg_model = LinearRegression()
linearReg_model.fit(X_reg_train,y_reg_train)
y_reg_pred1 = linearReg_model.predict(X_reg_test)

#Random Forest Regression
rf_reg_model = RandomForestRegressor()
rf_reg_model.fit(X_reg_train,y_reg_train)
y_reg_pred3 = rf_reg_model.predict(X_reg_test)
```

За допомогою основних метрик регресії оцінимо роботу моделі і оцінимо проблеми машинного навчання.

Основними метриками для задачі регресії є:

- Середня абсолютна похибка або MAE

Вона являє собою суму абсолютних відхилень між реальними та прогнозованими даними, що дає зрозуміти наскільки прогнозоване значення відрізняється від фактичного.

- Середня квадратична похибка або MSE

Схожа на середню абсолютну похибку проте вона дає загальне уявлення про величину помилки.

- Коефіцієнт детермінації або R^2

Метрика показує достовірність відповідного набору прогнозу фактичним значенням.

```
metrics_df=pd.DataFrame(columns=['Model', 'R2', 'MAE', 'MSE'])
```

```
def eval_metrics(actual, pred, model_name):
    mae = mean_absolute_error(actual, pred)
    mse = mean_squared_error(actual, pred)
```

```

r2 = r2_score(actual, pred)
acc = sqrt(r2)
global metrics_df
metrics_df=metrics_df.append({'Model': model_name, 'R2':r2, 'MAE':mae, 'MSE':mse}, ignore_index=True)

```

Для даної моделі продемонструємо метрики якості оцінки моделі (рисунок 3.4).

	Model	R2	MAE	MSE	Accuracy
0	Linear Regression	0.517561	606.976839	565863.541412	0.719417
1	Random Forests Regression	0.759192	392.282673	282448.805790	0.871316

Рисунок 3.4 – Значення метрик якості для моделі

Відповідно бачимо, що лінійна регресія показала себе гірше за метод випадкового лісу.

3.3 Керівництво користувача кінцевим програмним продуктом

Для вирішення даної задачі було вирішено розробити веб-додаток, адже веб-застосунки є простими в керуванні, не потребують скачування користувачем і доступні на будь-якому приладі. Браузерна версія має зрозуміли інтерфейс для користувача додатком.

Зображена перша сторінка (рисунок 3.5), яку бачить користувач при першому відкритті додатку. Можна побачити усі види сервісів, та зайти в свій обліковий запис натиснувши кнопку “Log in”.

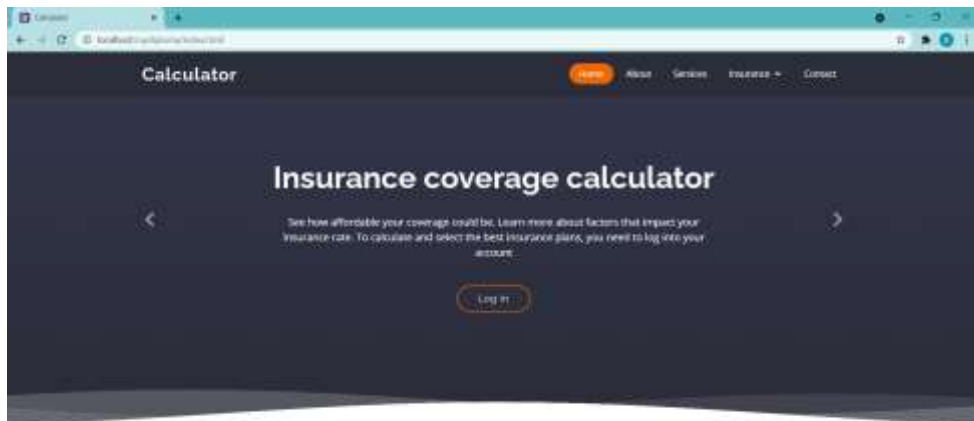


Рисунок 3.5 – Головна сторінка веб-додатку

При відкритті сторінки входу в обліковий запис користувач повинен ввести своє користувацьке ім'я та особистий пароль до сторінки. Якщо клієнт вперше використовує даний застосунок, він може натиснути кнопку “Sign up” для реєстрації всередині системи. Реєстрація всередині додатку потрібна для додаткової інформації про клієнта, що використовується при аналізі даних та підборі найоптимальнішого страхового пакету.

Натиснувши кнопку “Sign up” з'явиться поле для введення інформації про клієнта такої як: ім'я та прізвище, користувацьке ім'я всередині системи, місто проживання клієнта, номер телефону та створення паролю. Всі поля перевіряються на коректність вводу (рисунок 3.7).

При відкритті сторінки входу в обліковий запис (рисунок 3.6) користувач повинен ввести своє користувацьке ім'я та особистий пароль до сторінки. Якщо клієнт вперше використовує даний застосунок, він може натиснути кнопку “Sign up” для реєстрації всередині системи. Реєстрація всередині додатку потрібна для додаткової інформації про клієнта, що використовується при аналізі даних та підборі найоптимальнішого страхового пакету.

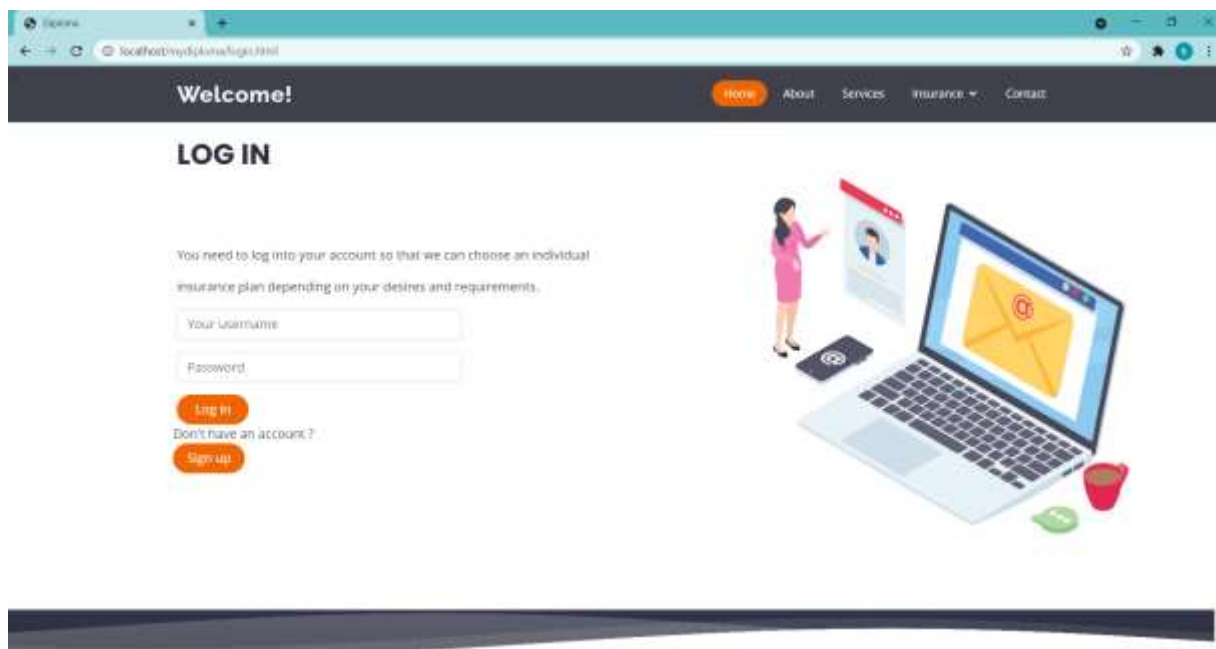


Рисунок 3.6 – Сторінка для входу в обліковий запис

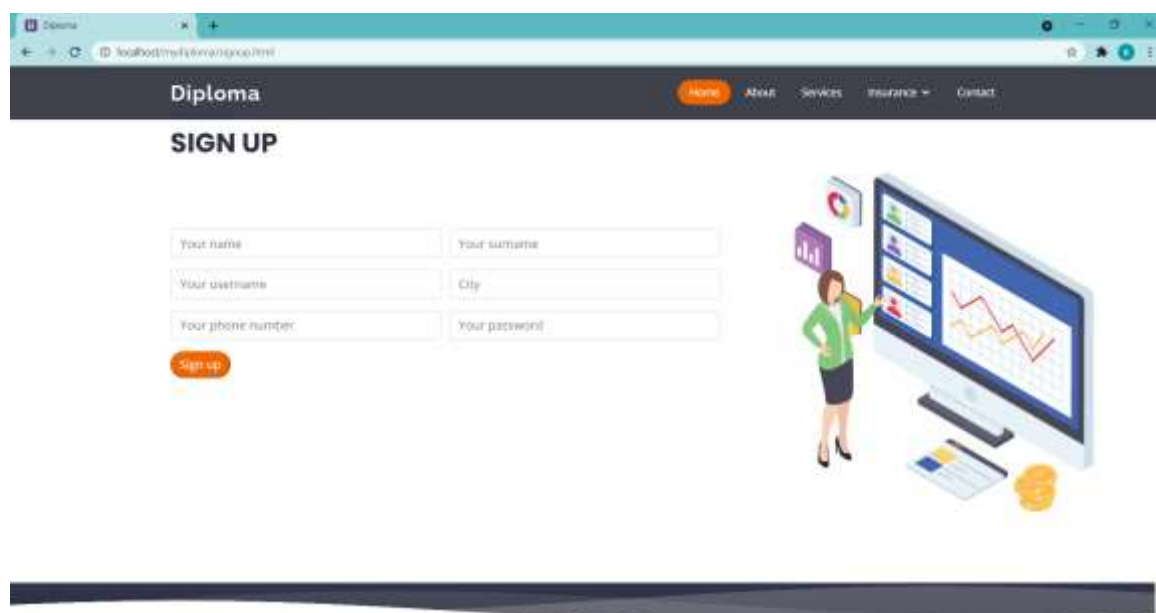


Рисунок 3.7 – Сторінка для створення нового облікового запису

Після коректного входження в систему користувач може побачити перелік можливих страхувань та підбору найоптимальнішого страхового пакету (рисунок 3.8). В даному застосунку обрахувати ціну та вибрати найоптимальнішу можна за такими категоріями як:

- Медичне страхування (Medical insurance);

- Страхування майна (Property insurance);
- Страхування автомобіля (Car insurance);
- Туристичне страхування (Travel insurance);
- Страхування життя (Life insurance).

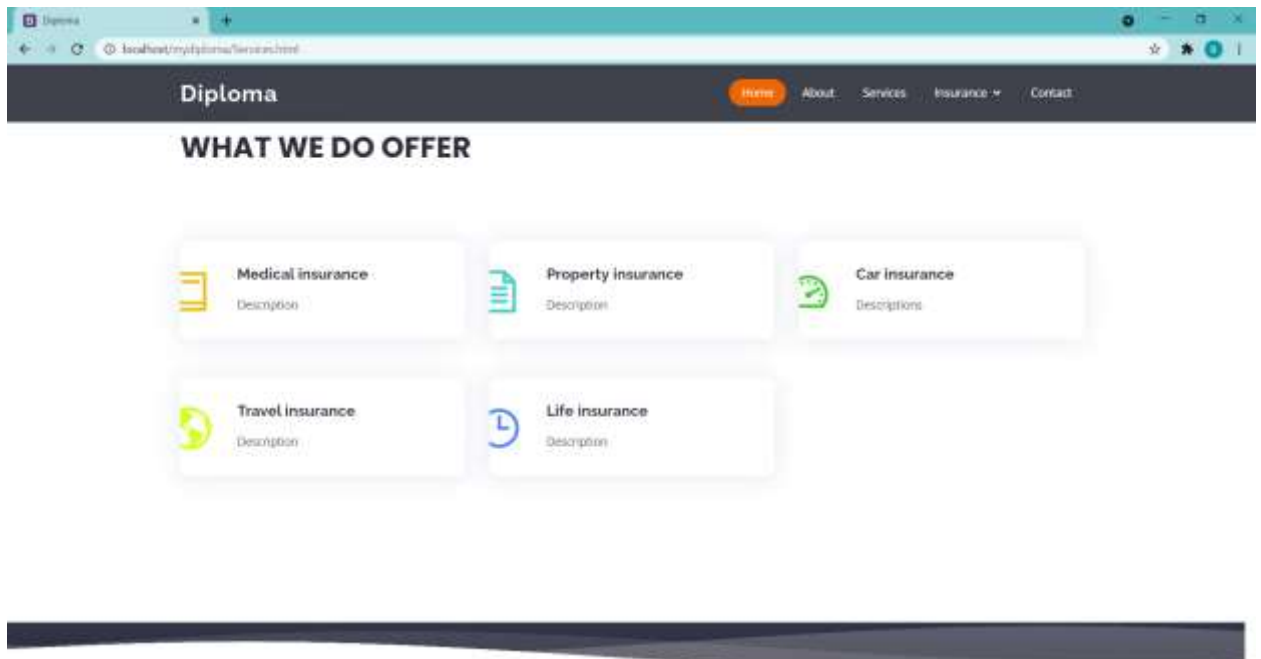


Рисунок 3.8 – Обрання категорії страхування

Залежно від обраної категорії будуть відкриті різні форми з відповідними полями, які потрібно заповнити для подальшого аналізу.

Для медичного страхування потрібно ввести дані про дату народження клієнта, його стать, коефіцієнт маси тіла, відмітити наявність або відсутність таких хвороб як діабет та будь-який вид раку також необхідно вказати чи клієнт палить, його кількість дітей, період контракту та суму відшкодування. Для медичного страхування та страхування життя форми є однаковими (рисунок 3.9).

The screenshot shows a web browser window with the URL `localhost:5555/diploma/insurance_med.html`. The page has a dark header with the 'Diploma' logo and navigation links: Home, About, Services, Insurance, and Contact. The main content area is titled 'ESTIMATED DATA' and contains a form with the following fields:

- Date of birth:
- Choose your gender:
- BMI (Body mass index):
- Diseases such as diabetes or cancer:
- Amount of children:
- Smoking:
- Term insured:
- Term of the contract:

A blue 'Submit' button is located at the bottom of the form.

Рисунок 3.9 – Форма для заповнення користувачем інформації про медичне страхування та страхування життя

Для страхування нерухомості потрібно ввести такі параметри як ціна апартаментів, площа апартаментів, вказати тип помешкання, період контракту страхування, вказати чи було помешкання здано в оренду, та чи є додатковий захист помешкання такий як камери відеоспостереження (рисунок 3.10).

The screenshot shows a web browser window with the URL `localhost:5555/diploma/insurance_property.html`. The page has a dark header with the 'Diploma' logo and navigation links: Home, About, Services, Insurance, and Contact. The main content area is titled 'ESTIMATED DATA' and contains a form with the following fields:

- Apartment price (USD):
- Apartment area (m2):
- Type of accommodation:
- Insurance period:
- Rental property:
- Additional protection:

A blue 'Submit' button is located at the bottom of the form.

Рисунок 3.10 – Форма для заповнення користувачем інформації про страхування нерухомості

Для туристичного страхування клієнт має вказати дати прибуття та відправлення закордон, територію на якій він буде перебувати, причину поїздки закордон, кількість поїздок та кількість людей, яких потрібно застрахувати (рисунок 3.11).

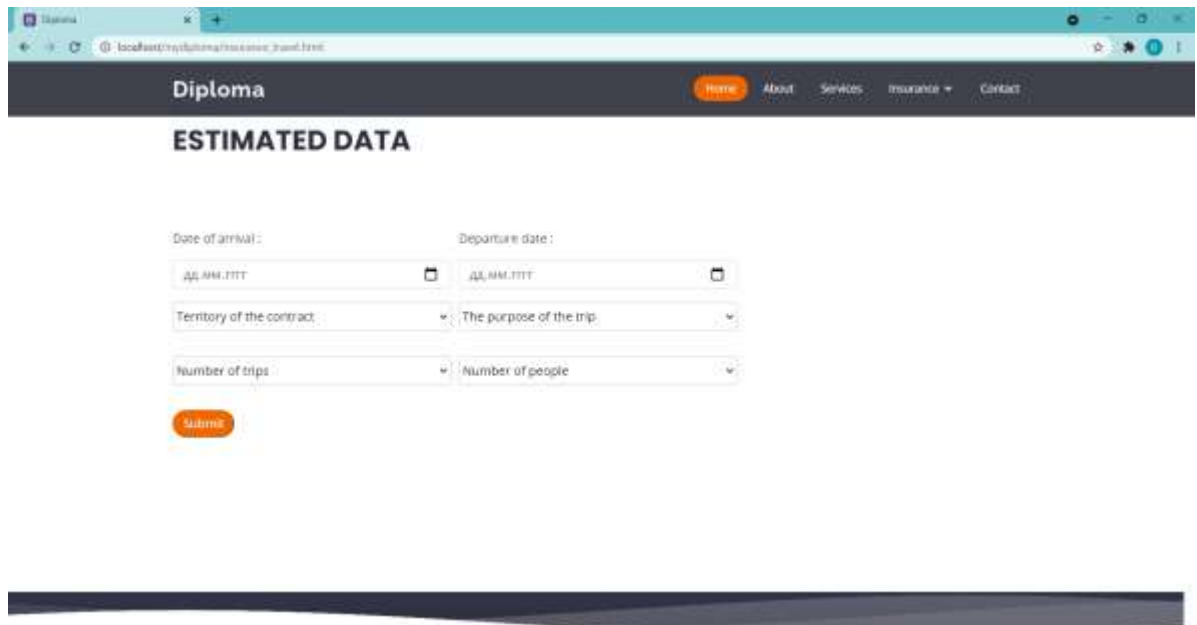
The image shows a web browser window with the URL 'localhost:3000/diploma/insurance/travel.html'. The website has a dark header with the 'Diploma' logo and navigation links: Home, About, Services, Insurance, and Contact. Below the header, the section 'ESTIMATED DATA' is displayed. The form contains several input fields: 'Date of arrival' and 'Departure date' (both with date pickers), 'Territory of the contract' and 'The purpose of the trip' (both dropdown menus), and 'Number of trips' and 'Number of people' (both dropdown menus). A blue 'Submit' button is located at the bottom of the form.

Рисунок 3.11 – Форма для заповнення користувачем інформації про туристичне страхування

Для оцінки страхування автомобіля необхідно вказати місто реєстрації автомобіля, чи зареєстрована вона закордоном, тип автомобіля чи засобу пересування, об'єм автомобіля, чи використовується воно як таксі чи для інших перевезень (рисунок 3.12).

Diploma Home About Services Insurance Contact

ESTIMATED DATA

City of registration: Foreign registration: Vehicle type: Car capacity: Used as a taxi: Additional protection: Submit

Рисунок 3.12 – Форма для заповнення користувачем інформації про страхування автомобіля

Після заповнення будь-якої з форм як результат роботи програми можна буде побачити три пакети: найдешевший пакет, найоптимальніший – ціна відповідає якості та тарифу, тариф з найбільшою кількістю страхових покриттів та функцій (рисунок 3.13).

Diploma Home About Services Insurance Contact

Insurance plan	Classic plan	Optimal plan	Business plan
Insurance company	AFG	UNICA	WAGO
Heavy insurance	✓	✓	✓
Disease insurance	✓	✓	✓
Loss of life	✓	✓	✓
Unlimited online consultations with a therapist	✓	✗	✓
International medical consultation	✗	✗	✗
Price (UAH)	4999	6120	7020

Рисунок 3.13 – Результат роботи програми

Отже, результатом роботи програми є підбір трьох пакетів по заданим параметрам. Для кожного з пакетів вказуються складові та умови страхування, ціна пакету.

Також веб-застосунок має сторінку About, де можна детально прочитати в чому особливість додатка і навіщо він потрібен (рисунок 3.14). Сторінка Contact (рисунок 3.15), дає можливість написати лист розробнику з пропозицією щодо співпраці та отримати контакти.

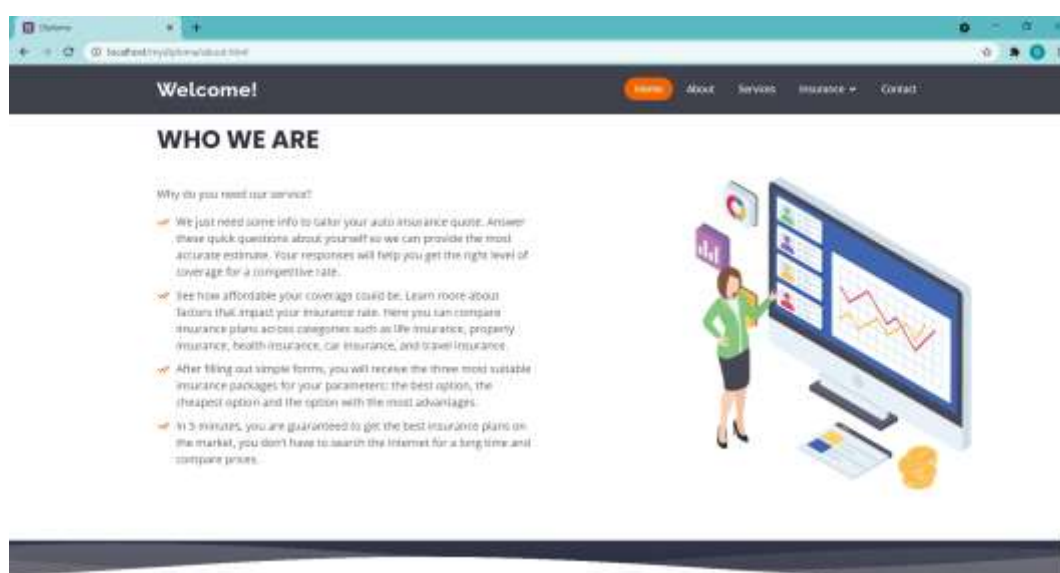


Рисунок 3.14 – Інформація про додаток

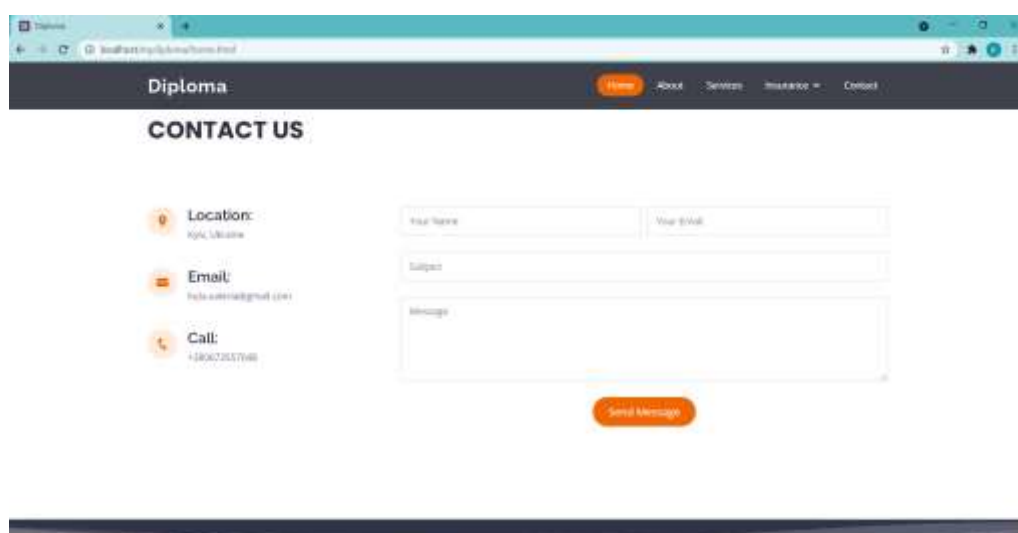


Рисунок 3.15 – Інформація про розробника та контакти

3.4 Висновки до розділу 3

У даному розділі було представлено кінцевий розроблений програмний продукт, обґрунтовано вибір мови програмування, відповідних фреймворків та бібліотек. Було наведено етапи дослідження та аналізу даних вибірки і даних наданих користувачем. Також було представлено результати роботи різних методів вирішення задачі регресії, представлені відповідні метрики і обрано найкращий метод вирішення даної задачі – метод лінійної регресії.

Окрім того, у даному розділі наведено детальну інструкцію для користувача програмного продукту. Наведений інтерфейс застосунку та структура бази даних програми.

Для вдосконалення системи у майбутньому було проведено результат виконаної роботи та продемонстровані метрики якості моделей. Було проведено аналіз майбутніх дій для вдосконалення і покращення метрик якості моделі.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ

4.1 Постановка задачі проектування

Задача полягає у проектуванні програмного продукту для автоматизації вибору страхового пакету користувача, в залежності від його потреб та бажань. Даний програмний продукт виконаний у вигляді веб-застосунку, щоб користувачі мали вільний доступ до нього та могли скористатися у будь-якій ситуації. Зазначений програмний продукт автоматично робить оцінку для різних страхових випадків, таких як страхування життя, медичне страхування, туристичне страхування, страхування нерухомого майна та страхування засобів пересування таких як мотоцикл, автомобіль та інші.

Перевагою даного програмного продукту також є його унікальність, адже проведені дослідження не виявили додатків такого типу у вільному доступі.

У даному розділі буде оцінено головні функціональні характеристики програмного продукту, який допомагає користувачу вибрати найоптимальніший страховий пакет, що представлений на ринку. Сам функціонально-вартісний аналіз допомагає обрахувати реальну вартість отриманого програмного продукту, мінімізувати витрати з боку виробника та виявити найбільш ефективний варіант відношення між витратою на виробництво та реалізацію продукту та його споживчою ціною.

Таким чином, ми можемо виділити основні вимоги до поставленої задачі автоматизації вибору страхового пакету клієнтом:

- програмний продукт повинен бути простий і зрозумілий у користуванні;
- програмний продукт має підтримуватися на різних операційних системах;

- програмний продукт повинен мати високу швидкість роботи;
- програмний продукт повинен працювати для різних даних.

4.2 Обґрунтування функцій та параметрів програмного продукту

Головна функція F_0 – розробка програмного продукту, який вирішує задачу автоматизації підбору найоптимальнішого страхового пакету та буде його модель. Беручи за основу цю функцію, можна виділити наступні:

F1 – архітектура програмного додатку:

- а) прикладний програмний інтерфейс;
- б) веб-застосунок;
- в) мобільний застосунок.

F2 – збір бази даних:

- а) веб-скрейпінг;
- б) використання відкритих баз даних;
- в) співробітництво зі страховими компаніями.

F3 – вибір мови програмування:

- а) Python;
- б) C++.

F4 – алгоритми навчання моделей:

- а) лінійна регресія;
- б) випадковий ліс.

Виходячи з наведених вище варіантів будуюмо морфологічну карту (рисунок 4.1).

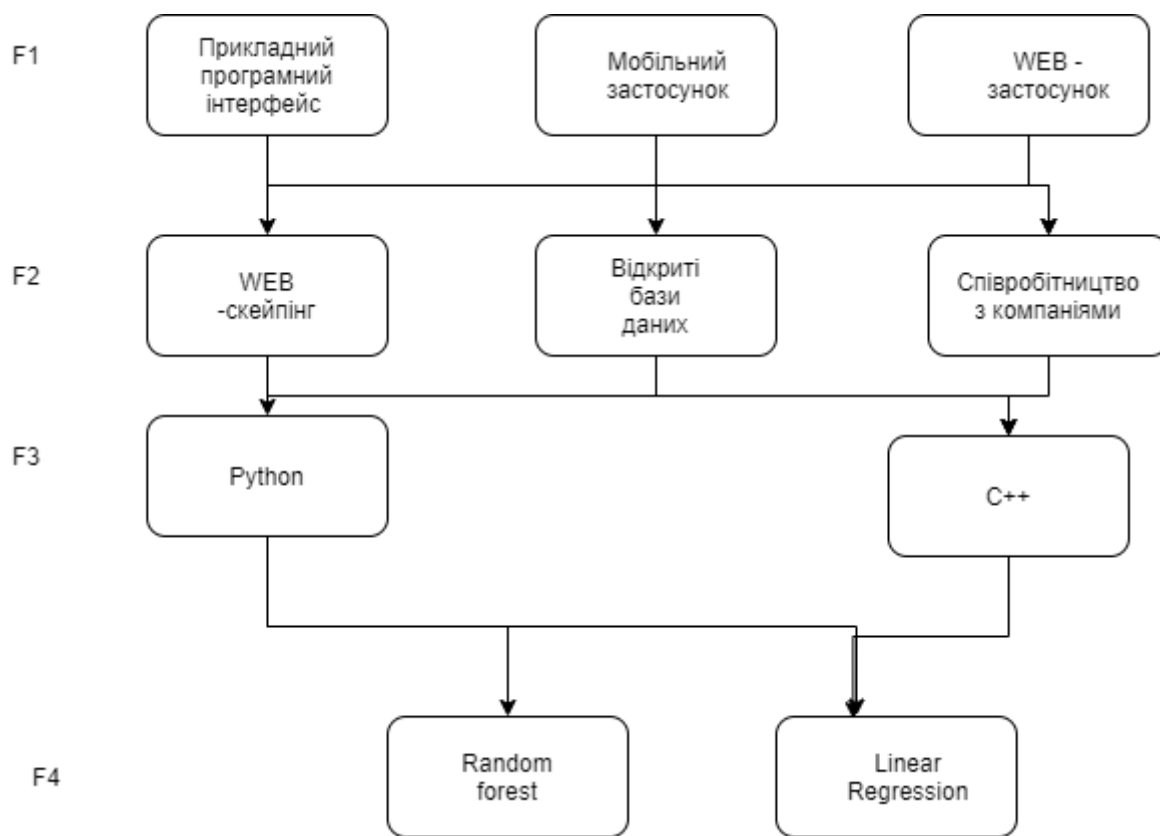


Рисунок 4.1 – Морфологічна карта

Спираючись на карту була побудована позитивно-негативна матриця (таблиця 4.1).

Таблиця 4.1– Позитивно-негативна матриця

Основна функція	Варіант реалізації	Переваги	Недоліки
F1	А	Легкість реалізації, можливість інтеграції на інших веб-ресурсах	Неможливість використовувати як окремий самостійний веб-сайт
	Б	Можливість використовувати як	Складність реалізації, великі затрати на рекламу,

		окремий самостійний додаток	втрачається аудиторія потенційних клієнтів
	В	Можливість використовувати як окремий самостійний web - застосунок	Складність реалізації, великі затрати на рекламу, втрачається аудиторія потенційних клієнтів
F2	А	Актуальність даних, незалежність від компаній, достатній об'єм даних	Складність реалізації
	Б	Легкість реалізації	Неактуальність даних, малий об'єм доступних даних
	В	Актуальність даних, можливість працювати з якісними даними	Складність реалізації
F3	А	Швидка розробка програми, доступність бібліотек, кросплатформеність	Низька швидкість роботи, особливо, якщо потрібно обробляти велику кількість даних
	Б	Швидкість роботи	Недоступність бібліотек і складність розробки
F4	А	Легкість реалізації	Необхідність якісно обробляти дані для навчання моделі, не найкращий показник точності

	Б	Велика точність, добре справляється з викидами	Складність реалізації
--	---	--	-----------------------

Для характеристики прототипу програмного додатку використовуємо параметри X1 – X4 (рисунки 4.2 – 4.5). На основі даних, що представлені у літературі, визначаємо мінімальні, середні отримуванні та максимально допустимі значення(таблиця 4.2).

Таблиця 4.2 – Система параметрів додатку

Найменування параметру	Позначення параметру	Значення параметру		
		Мінімальне	Середнє	Максимальне
Час розробки, людина*год	X1	65	160	370
Час збору бази даних, людина*год	X2	5	20	45
Час навчання моделі, с	X3	50	120	210
Величина об'єму пам'яті для збереження даних, Мб	X4	2	8	16

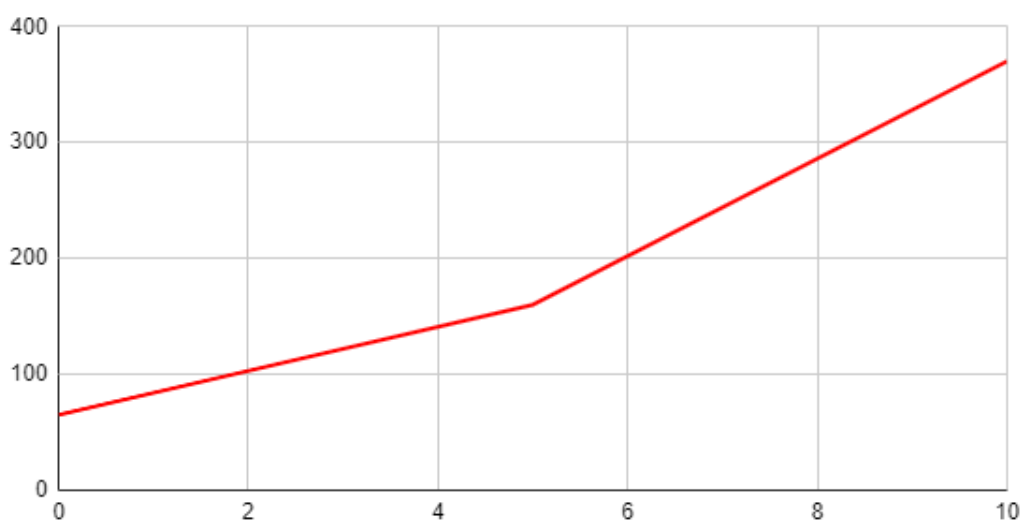


Рисунок 4.2 – Значення параметра X1

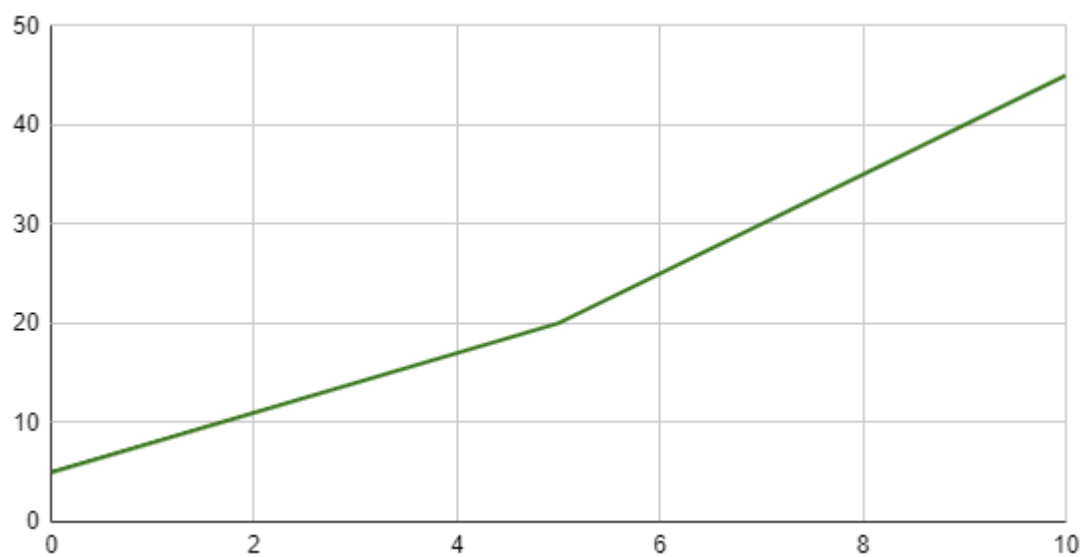


Рисунок 4.3 – Значення параметра X2

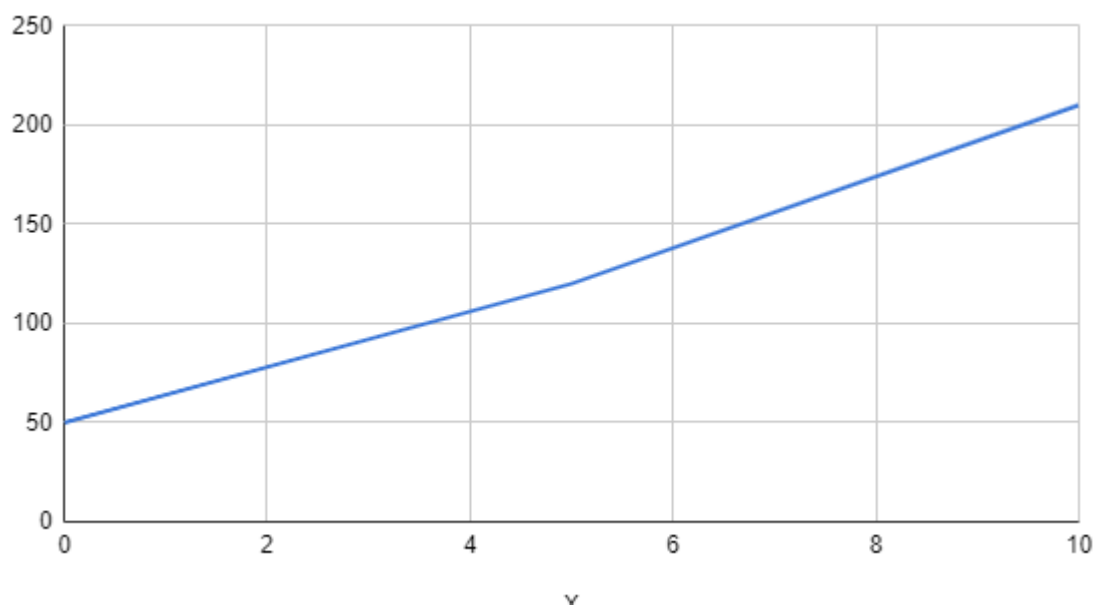


Рисунок 4.4 – Значення параметра X3

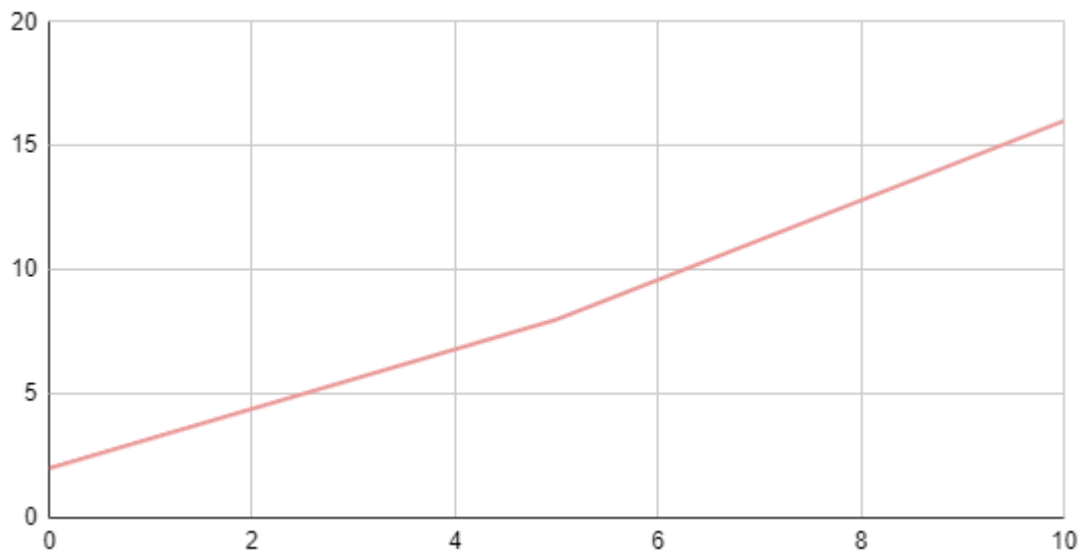


Рисунок 4.5 – Значення параметра X4

Вагомість параметрів оцінюється за допомогою методів попарного зрівняння. Ранги варіюються від 1 до 5, де 1 – найменший ранг, 5 – найбільший. Результати наведені в таблицях 4.3-4.4.

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70, \quad (4.1)$$

де N – число експертів;

n – кількість параметрів.

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 17,5, \quad (4.2)$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

Таблиця 4.4 – Попарне зрівняння параметрів

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 та X2	>	<	<	>	>	>	>	>	1.5
X1 та X3	>	>	>	>	>	>	>	>	1.5
X1 та X4	>	>	>	>	>	>	>	>	1.5
X2 та X3	>	>	>	>	<	>	>	>	1.5
X2 та X4	>	>	>	>	>	>	>	>	1.5
X3 та X4	<	>	>	>	>	<	<	>	1.5

Як видно з таблиці 4.5, різниця значень коефіцієнтів вагомості не перевищує 5% після другої ітерації, тому більшої кількості ітерацій не потрібно.

Таблиця 4.5 – Розрахунок вагомості параметрів

Параметри	Параметри X_i				Перший крок		Другий крок	
	X1	X2	X3	X4	b_i	K_{bi}	b_i	K_{bi}
X1	1	1,5	1,5	1,5	5.5	0,344	21.5	0,361
X2	0,5	1	1,5	1,5	4.5	0,281	16.25	0,275
X3	0,5	0,5	1	1,5	3.5	0,219	12.5	0,207
X4	0,5	0,5	0,5	1	2.5	0,156	9.25	0,157
Загалом:					16	1	59	1

На основі порівняльного аналізу варіантів реалізації функцій за їх перевагами та недоліками ми обрали наступні два варіанти реалізації:

1. $F1(b) \Rightarrow F2(a) \Rightarrow F3(a) \Rightarrow F4(a)$

$$2. \quad F1(b) \Rightarrow F2(a) \Rightarrow F3(a) \Rightarrow F4(b)$$

Таблиця 4.6 – Таблиця коефіцієнтів якості функцій

Основна функція	Варіант реалізації	Абсолютне значення параметру	Бальна оцінка параметру	Коефіцієнт вагомості параметру	Коефіцієнт якості
F1	в)X1	60	9.83	0.361	3.55
F2	а)X2	15	5	0.275	1.38
F3	а)X3	120	6.36	0.207	1.32
F4	а)X4	12	2.5	0.157	0.39
	б)X4	4	7.5	0.157	1.17

Обрахуємо коефіцієнти якості кожного з варіантів розробки:

$$K_{я1} = 3.55 + 1.38 + 1.32 + 0.39 = 6.64$$

$$K_{я2} = 3.55 + 1.38 + 1.32 + 1.17 = 7.42$$

Оскільки варіант 2 має більший коефіцієнт якості, він є найкращим.

4.3 Економічний аналіз варіантів розробки

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості. Всі варіанти включають в себе два окремих завдання:

1. Розробка проекту програмного продукту;
2. Розробка програмної оболонки.

При цьому два варіанти мають різні додаткові завдання:

4.1. Вирішення задачі регресії методом випадкового лісу (для першого варіанту);

4.2. Вирішення задачі регресії методом лінійної регресії (для другого варіанту).

За ступенем новизни завдання 1 відноситься до групи А, завдання 2, відноситься до групи Б. За складністю алгоритми, які використовуються в завданні 1 належать до 1 групи, завдання 2 – до 3 групи.

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру ступеню новизни А та групи складності алгоритму 1, трудомісткість дорівнює: $T_p = 90$ людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання: $K_n = 1.7$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх семи завдань рівний 1: $K_{ск} = 1$. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{ст} = 0.8$. Тоді загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 90 * 1.7 * 0.8 = 122.4 \text{ (людино-днів)}.$$

Проведемо аналогічні розрахунки для інших завдань. Для другого завдання (використовується алгоритм третьої групи складності, ступінь новизни Б), тобто $T_p = 19$ людино-днів, $K_n = 0.9$, $K_{ск} = 1$, $K_{ст} = 0.8$. Отже:

$$T_2 = 19 * 0.9 * 0.8 = 13.32 \text{ (людино-днів)}.$$

Аналогічно для завдання 4.1 та 4.2:

$$T_3 = 27 * 0.9 * 0.8 = 19.44$$

$T_p = 12$ людино-днів, $K_n = 0.6$, $K_{ск} = 1$, $K_{ст} = 0.8$

$$T_4 = 12 * 0.6 * 0.8 = 5.76$$

Визначимо повну трудомісткість варіантів (людино-днів):

$$T_1 = 122,4 + 19,44 + 19,44 = 161,28$$

$$T_2 = 122,4 + 19,44 + 5,76 = 147,6$$

Найбільш трудомістким завданням є 1. Далі вважається, що робочий день складає 8 годин, в тиждні п'ять робочих днів. В розробці беруть участь два програміста з окладом 14500 грн та аналітик з окладом 15000 грн. Визначимо середню заробітну плату за годину:

$$C_q = \frac{2 * 14500 + 15000}{3 * 21 * 8} = 87,30$$

Заробітна розробників за варіантами становить:

$$C_{зп} = 87,30 * 8 * 161,28 = 112\,637,95$$

$$C_{зп} = 87,30 * 8 * 147,6 = 103\,083,84$$

Відрахування на єдиний соціальний внесок становить 22%:

$$C_{вд} = 112\,637,95 * 0,22 = 24\,780,14$$

$$C_{вд} = 103\,083,84 * 0,22 = 22\,678,44$$

Визначимо витрати на оплату однієї машино-години. Так як одна машина обслуговує двох програмістів з окладом 14500 грн. з коефіцієнтом зайнятості 0.6 та одного аналітика з окладом 15000, то для трьох машин отримаємо:

$$C_r = 12 * 14500 * 2 * 0,6 + 12 * 15000 * 0,6 = 316\,800 \text{ (грн)}$$

Враховуючи додаткову заробітну плату 40%

$$C_{3П} = 316\,800 * (1 + 0,4) = 443\,520 \text{ (грн)}$$

Відрахування на соціальне страхування 22%

$$C_{ВІД} = 443\,520 * 0,22 = 97\,574,40 \text{ (грн)}$$

Розрахуємо амортизаційні підрахунки (амортизація 25%, вартість ЕОМ 24000 грн)

$$C_A = K_{ТМ} * K_A * Ц_{ПР} = 1,15 * 0,25 * 24000 = 6900 \text{ (грн)}$$

Розрахуємо витрати на ремонт та профілактику:

$$C_P = K_{ТМ} * Ц_{ПР} * K_P = 1,15 * 24000 * 0,05 = 1380 \text{ (грн)}$$

Розрахуємо ефективний годинний фонд часу ПК за рік:

$$T_{ЕФ} = (365 - 104 - 11 - 16) * 8 * 0,85 = 1591,20 \text{ год}$$

Розрахуємо витрати на електроенергію

$$C_{ЕЛ} = 1591,20 * 0,6 * 0,85 * 3,51 = 2\,848,40 \text{ грн}$$

Накладні витрати рівні:

$$C_H = 24000 * 0,67 = 16\,080 \text{ (грн)}$$

Отже, експлуатаційні витрати:

$$C_{\text{ЕКС}} = 443\,520 + 97\,574,40 + 6900 + 1380 + 2848,40 + 16080 = 568\,302,80 \text{ (грн)}$$

Собівартість однієї машино-години буде:

$$C_{\text{М-Г}} = \frac{568\,302,80}{1591,20} = 357,15 \text{ грн/год.}$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, то витрати на оплату машинного часу в залежності від обраного варіанту, складають:

$$C_{\text{М}} = 357,15 * 8 * 161,28 = 460\,809,22 \text{ грн.}$$

$$C_{\text{М}} = 357,15 * 8 * 147,6 = 421\,722,72 \text{ грн.}$$

Накладні витрати складають 67% від заробітної плати:

$$C_{\text{Н}} = 112\,637,95 * 0,67 = 75\,467,43 \text{ (грн)}$$

$$C_{\text{Н}} = 103\,083,84 * 0,67 = 69\,066,17 \text{ (грн)}$$

Отже, вартість розробки програмного продукту за варіантами становить:

$$C_{\text{ПП}} = 112\,637,95 + 24\,780,14 + 460\,809,22 + 75\,467,43 = 673\,694,74 \text{ (грн)}$$

$$C_{\text{ПП}} = 103\,083,84 + 22\,678,44 + 421\,722,72 + 69\,066,17 = 616\,551,17 \text{ (грн).}$$

4.4 Вибір кращого варіанта ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня:

$$K_{\text{ТЕР}1} = \frac{6.64}{616\,551,17} = 1,07 * 10^{-5}$$

$$K_{\text{ТЕР}2} = \frac{7.41}{616\,551,17} = 1,20 * 10^{-5}$$

Як бачимо, найбільш ефективним є другий варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{\text{ТЕР}2} = 1,20 * 10^{-5}$.

Отже, враховуючи всі проведені дослідження, можна зауважити, що варіант реалізації №2 є більш оптимальний, так як має вищі якісно-економічні показники. Його коефіцієнт техніко-економічного рівня складає $1,20 * 10^{-5}$, в той час як коефіцієнт для варіанту розробки №1 складає $1,07 * 10^{-5}$.

Розробка цього варіанту передбачає такі обов'язкові завдання як:

- мова програмування – Python;
- використання і збір даних за допомогою web-скейпінгу;
- розробки web-застосунку.

Серед завдань між якими ставився вибір в даному варіанті реалізовані такі завдання:

1. Розробка проекту програмного продукту;
2. Розробка програмної оболонки;
3. Вирішення задачі регресії за допомогою лінійної регресії

Даний варіант виконання програмного комплексу дає користувачу зручний інтерфейс, непоганий функціонал і швидкодію.

Отже, у цьому розділі було проведено повний функціонально-вартісний аналіз програмного продукту. Визначено основних функцій програмного продукту. Визначено параметри, які характеризують програмний продукт.

Проведено експертне оцінювання параметрів та аналіз якості варіантів реалізації функцій.

ВИСНОВКИ

В даній роботі було розглянуто використання інтелектуальних систем у автоматизації бізнес-процесів. Для прикладу реалізації таких автоматизованих системи було обрано страхову систему і бізнес-процес - визначення та підбору страховим агентом спеціального індивідуального страхового пакету для клієнта. Було проведено дослідження на попит послуг страхових компаній в Україні, аналіз сучасних рішень даної задачі, визначення основних зв'язків між даними клієнта та ціною на страховий пакет.

Також для вирішення даної задачі було проведено аналіз з математичного боку, визначення типу задачі інтелектуального аналізу даних, дослідження існуючих методів вирішення задач такого типу, дослідження основних критерій якості роботи. Також в свою чергу було проведено детальний аналіз даних, що використовувалось для навчання моделі, було проведено кореляційний аналіз даних, що використовує модель. Для навчання моделі було використано дані з сервісу kaggle та дані з веб-сайтів страхових компаній.

Результатом даної роботи є повноцінний програмний продукт для визначення найоптимальнішого страхового пакету для користувача. Створений продукт має вигляд WEB – застосунку, що працює за допомогою попередньо навченої моделі задачі регресії.

Було проаналізовано роботу системи, дослідження характеристик її роботи таких як точність, швидкість. Завдяки цьому було визначено наступні дії щодо вдосконалення системи підбору оптимального страхового пакету.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Тырышкина, Е.С. Социальное страхование: современное состояние и возможности развития сб. науч. тр. – Минск, 2019. – С. 127.
2. Жилінська Н. Н. Мировой опыт развития рынка интернет-страхования. Минск, 2019. [Електронний ресурс]: - Режим доступу: http://edoc.bseu.by:8080/bitstream/edoc/85260/1/Zhilinskaya_59_63.pdf
3. MachineLearning: професійний інформаційно-аналітичний ресурс, присвячений машинному навчанню, розпізнанню образів та інтелектуальному аналізу даних [Електронний ресурс]: MachineLearning.ru. — Режим доступу: <http://www.machinelearning.ru>
4. Голубев Ю. Ф. Нейросетевые методы в мехатронике. — М.: Изд-во Моск. унта, 2007. — 157 с. — ISBN 978-5-211-05434-9.
5. Чубукова І.О. Data Mining — Режим доступу: http://lnfm1.sai.msu.ru/~rastor/Books/ChubukovaData_Mining.pdf (дата звернення 20.02.2020)
6. Могильний С. Б. Машинне навчання з використанням мікрокомп'ютерів. – Київ, 2019. – 226 с.
7. Django Framework Documentation. URL: <https://docs.djangoproject.com/en/3.0/> (Last accessed: 18.04.2020).
8. Django REST Framework Documentation. URL: <https://www.django-rest-framework.org/> (Last accessed: 18.04.2020).
9. Scikit-learn: Machine Learning in Python: 2019.
10. Шеер А.-В. Бизнес-процессы. Основные понятия. Теория. Методы. – СПб.: Весть-Метатехнология, 2000. – 340 с.
11. Стандарт ISO 8402:1994. – http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?cs number=20115
12. Шеер А.-В. Моделирование бизнес-процессов. – СПб.: Весть-Метатехнология, 2000. – 242 с.

13. Маслобоев А.В., Шишаев М.Г. Функциональная структура и особенности реализации распределенной мультиагентной системы информационной поддержки инноваций // Информационные ресурсы России. – 2008. – № 5(105) – С. 23–27.
14. Маслобоев А.В., Шишаев М.Г. Мультиагентная система интеграции распределенных информационных ресурсов инноваций // Программные продукты и системы. – 2007. – № 4(92) – С. 30–32.
15. Добрецов С.В., Шестаков С.М. Планирование действий в искусственном интеллекте // Вестн. Академии техн. творчества «ДЕМИУРГ», 1998 – № 1. – С. 32–46.
16. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2000. – 273 с.
17. Осипов Г.С. Приобретение знаний интеллектуальными системами. – М.: Наука. Физматлит, 1997. – 142 с.
18. Хорошевский В.Ф. Методы и средства проектирования и реализации мультиагентных систем // Проблемы искусственного интеллекта: Материалы сем., Москва 2 июня 1999 г., ИПУ РАН. – 15 с.
19. Muller J.P. A Cooperation Model for Autonomous Agents // Proc. of the Third Inter. Workshop on Agent Theories, Architectures, and Languages (ATAL96), Budapest, Hungary, 1996. – P. 135–147.

ДОДАТОК А

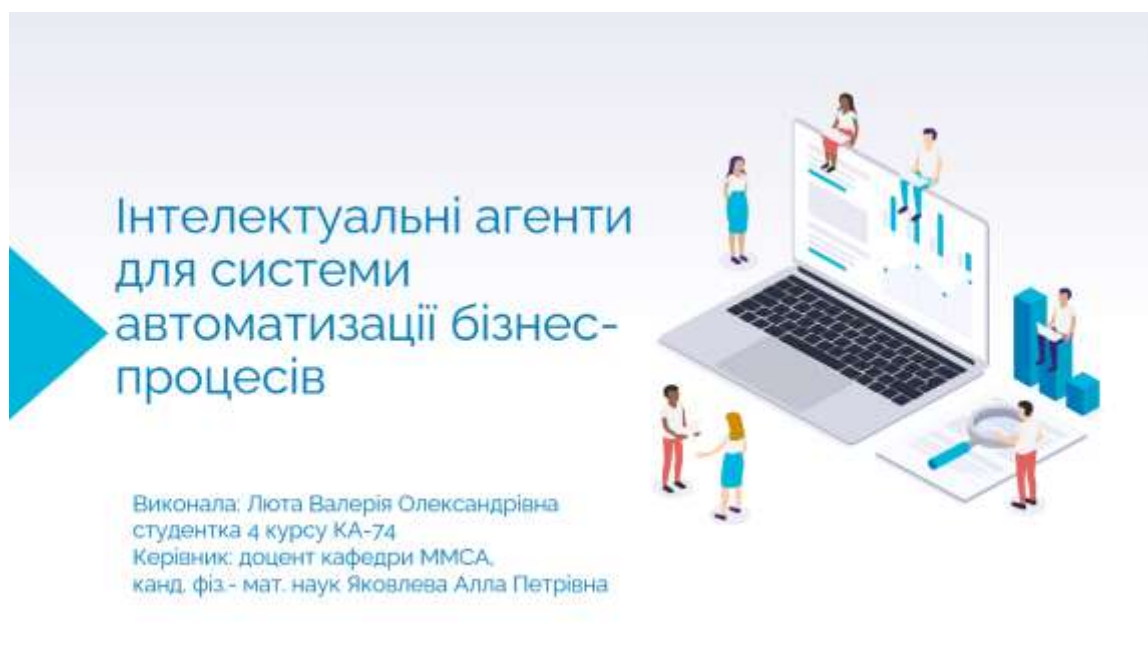


Рисунок А.1 – Слайд №1



Рисунок А.2 – Слайд №2



Рисунок А.3 – Слайд №3



Рисунок А.4 – Слайд №4



Рисунок А.5 – Слайд №5



Рисунок А.6 – Слайд №6

Основні методи та задач машинного навчання



Рисунок А.7 – Слайд №7

Лінійна Регресія

- Загальна лінійна регресійна модель має вигляд:

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + u,$$

де y – залежна пояснювальна змінна, $(x_1, x_2 \dots x_k)$ – незалежні змінні, u – випадкова похибка, розподіл якої залежить від незалежних змінних

- Математичне сподівання залежної змінної:

$$E(y) = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + u$$

де вектор $(\beta_0, \beta_1, \dots, \beta_k)$ – невідомий і задачею регресії є визначити ці параметри завдяки значенням y та $(x_1, x_2 \dots x_k)$.



Рисунок А.8 – Слайд №8

Random Forest

- Основа алгоритму дерев рішень в тому, що під час роботи на кожному кроці відбувається ділення по одному з критеріїв оцінки даних. Таке розбиття даних відбувається щоб збільшити кількість інформації, при цьому зменшити ентропію в попередньо утворених групах. ознаками д полягає в тому, що на кожному кроці відбувається розбиття по одній з ознак даних.

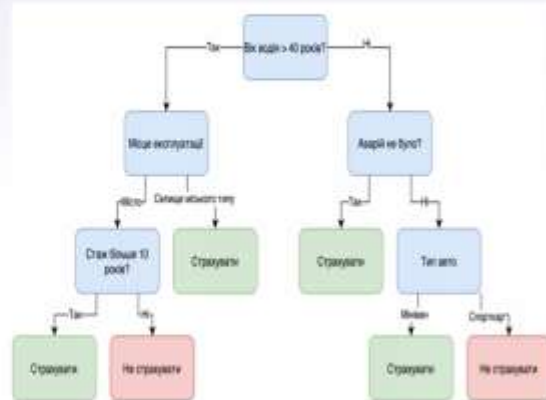


Рисунок А.9 – Слайд №9

Порівняння методів машинного навчання

Основними метриками для задачі регресії є:

- Середня абсолютна похибка або MAE (Являє собою суму абсолютних відхилень між реальними та прогнозованими даними, що дає зрозуміти наскільки прогнозоване значення відрізняється від фактичного.)
- Середня квадратична похибка або MSE (дає загальне уявлення про величину помилки).
- Коефіцієнт детермінації (Метрика показує достовірність відповідного набору прогнозу фактичним значенням)

	Model	R2	MAE	MSE
0	Linear Regression	0.517561	606.976839	565863.541412
1	Random Forests Regression	0.759192	392.282673	282448.805790

Рисунок А.10 – Слайд №10

Вибір технологій

- Для розробки функціональної частини програмного продукту, а також роботи з даними було обрано використовувати мову програмування Python та вбудовану бібліотеку машинного навчання scikit-learn (sklearn).
- Для інтерфейсу програмного продукту, а саме, веб-продукту були використано HTML+CSS+JavaScript. Для розробки веб-додатку був використаний фреймворк Bootstrap.
- Для створення API був використаний Django REST Framework. Взаємодія між клієнтським додатком та серверною частиною забезпечується за допомогою RESTful архітектури та HTTP запитам.



11

Рисунок А.11 – Слайд №11



Рисунок А.12 – Слайд №12

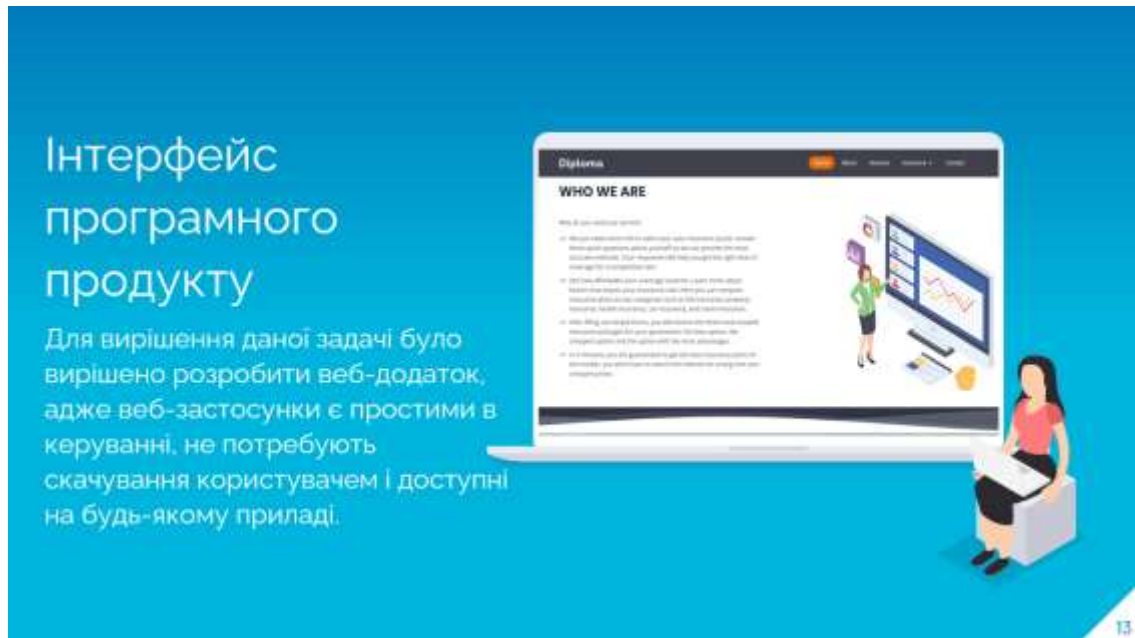


Рисунок А.13 – Слайд №13

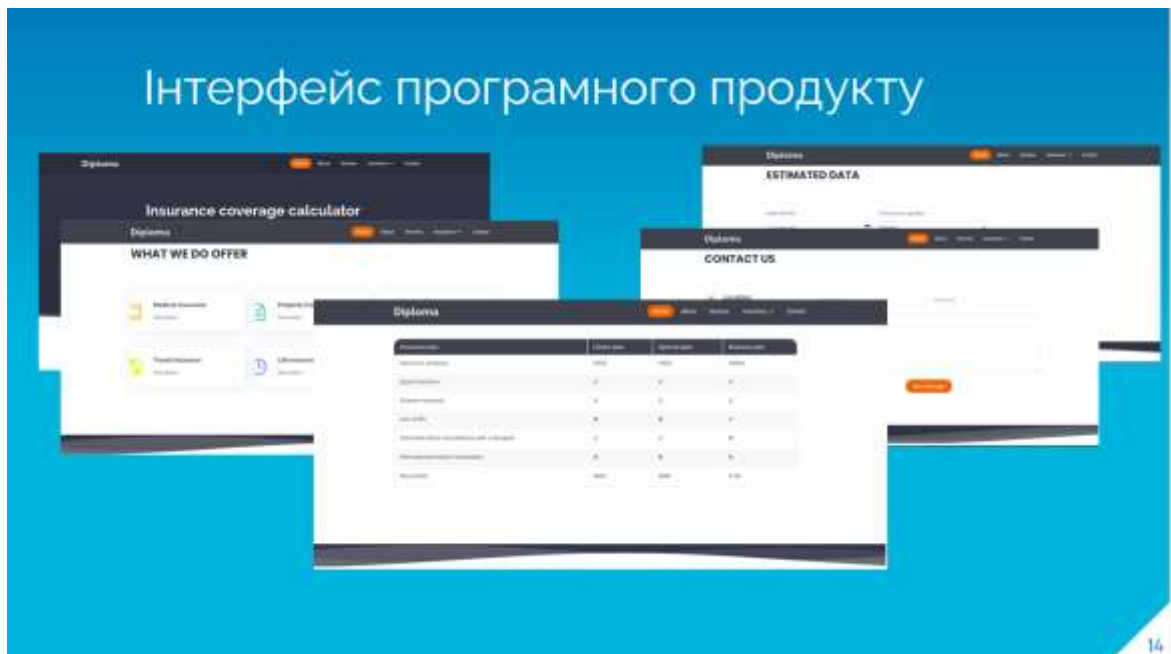


Рисунок А.14 – Слайд №14

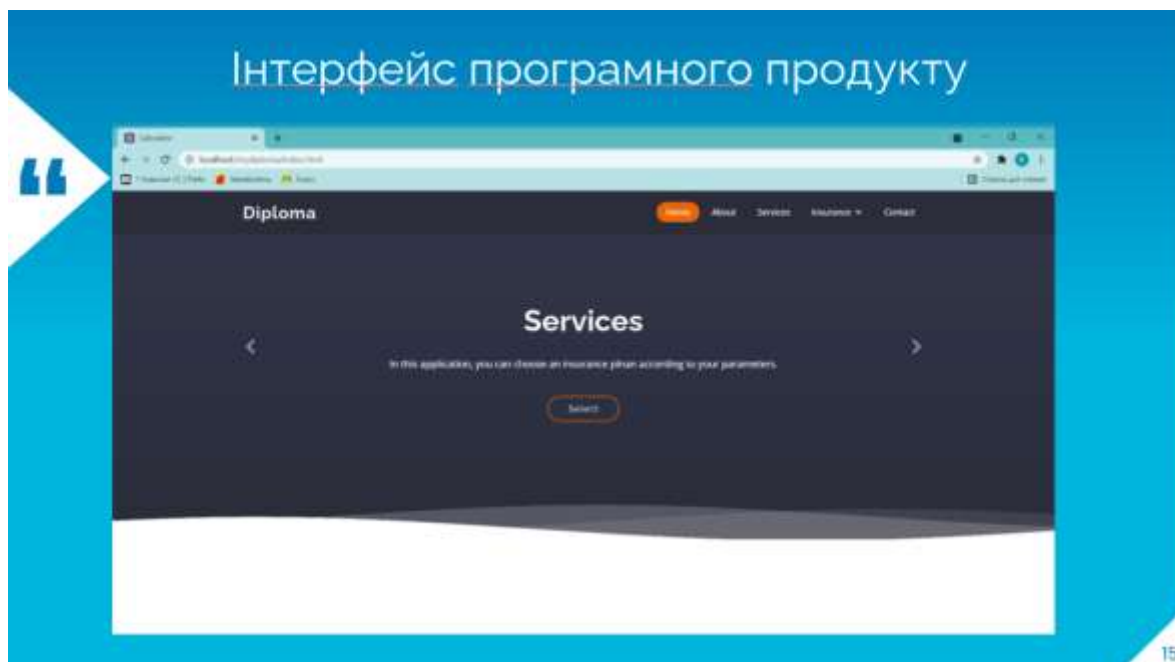


Рисунок А.15 – Слайд №15

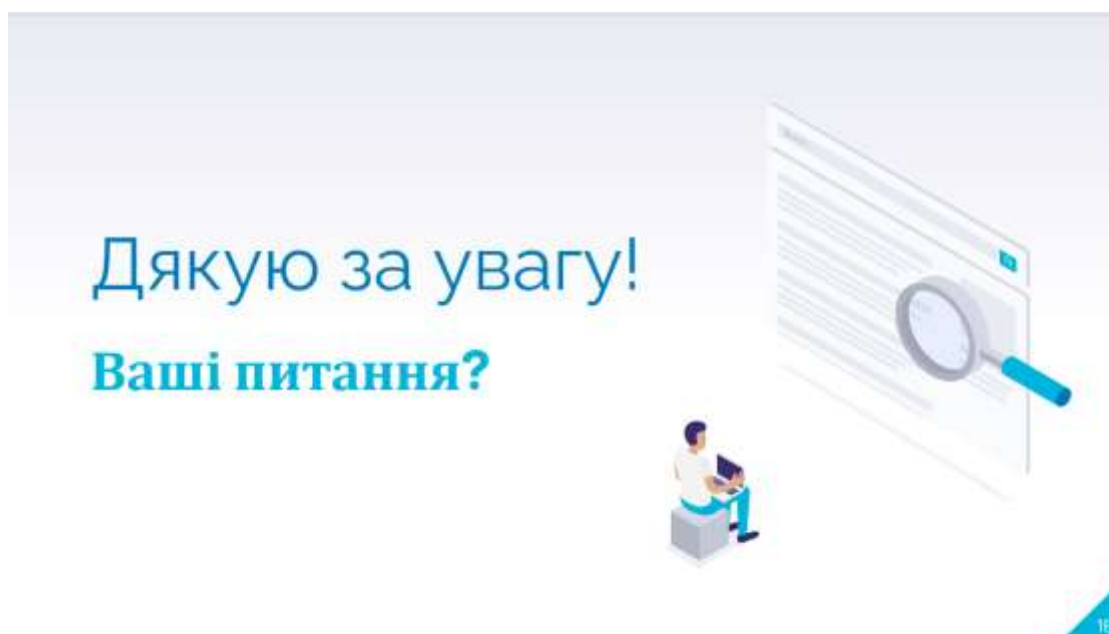


Рисунок А.16 – Слайд №16

ДОДАТОК Б

Views.py

```
import pickle
```

```
from datetime import date, datetime
```

```
import pandas as pd
```

```
from django.http import request
```

```
import json
```

```
from rest_framework import viewsets
```

```
from rest_framework.views import APIView
```

```
from rest_framework.response import Response
```

```
from rest_framework import status
```

```
from .regression import post_insurnce_plan, post_insurnce_plan_life,  
post_insurnce_plan_med
```

```
from .serializers import ClientSerializer, TravallInsuranceSerializer,  
CarInsuranceSerializer, PropertyInsuranceSerializer, LifeInsuranceSerializer,  
MedInsuranceSerializer
```

```
from .models import Client, TravallInsurance, CarInsurance, PropertyInsurance,  
LifeInsurance, MedInsurance
```

```
from django.views.decorators.csrf import csrf_exempt
```

```
from django.utils.decorators import method_decorator
```

```
@method_decorator(csrf_exempt, name='dispatch')
```

```
class ClientViewSet(viewsets.ModelViewSet):
```

```
    queryset = Client.objects.all().order_by('name')
```

```
    serializer_class = ClientSerializer
```

```
class PropertyInsuranceViewSet(viewsets.ModelViewSet):
    queryset = PropertyInsurance.objects.all().order_by('apartmentType')
    serializer_class = PropertyInsuranceSerializer
```

```
class CarInsuranceView(APIView):
    def post(self, request, *args, **kwargs):
        data = {
            'cityRegistration': request.data.get('cityRegistration'),
            'foreignRegistration': request.data.get('foreignRegistration'),
            'typeVehicle': request.data.get('typeVehicle'),
            'carCapacity': request.data.get('carCapacity'),
            'useTaxi': request.data.get('useTaxi'),
            'additionalProtection': request.data.get('additionalProtection')
        }
        serializer = CarInsuranceSerializer(data=data)
        if serializer.is_valid():
            serializer.save()
            result = post_insurnce_plan([[request.data.get('foreignRegistration'),
request.data.get('typeVehicle'), request.data.get('carCapacity'),
request.data.get('useTaxi'), request.data.get('additionalProtection')]])
            return Response(result, status=status.HTTP_201_CREATED)

        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

```
class LifeInsuranceView(APIView):
```

```
    def post(self, request):
        data = {
```

```

        'dateofBirth': request.data.get('dateofBirth'),
        'gender': request.data.get('gender'),
        'bmi': request.data.get('bmi'),
        'diabet': request.data.get('diabet'),
        'children': request.data.get('children'),
        'smoking': request.data.get('smoking'),
        'sumInsurance': request.data.get('sumInsurance'),
        'insurancePeriod': request.data.get('insurancePeriod')
    }
    serializer = LifeInsuranceSerializer(data=data)
    if serializer.is_valid():
        serializer.save()
        born = datetime.strptime(request.data.get('dateofBirth'), '%Y-%m-%d')
        today = date.today()
        age = today.year - born.year - ((today.month, today.day) < (born.month,
born.day))
        result = post_insurnce_plan_life([[age, request.data.get('gender'),
                                         request.data.get('bmi'), request.data.get('diabet'),
                                         request.data.get('children'),
request.data.get('smoking')]])
        return Response(result, status=status.HTTP_201_CREATED)

    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

class MedInsuranceView(APIView):

    def post(self, request):
        data = {
            'dateofBirth': request.data.get('dateofBirth'),

```

```

'gender': request.data.get('gender'),
'bmi': request.data.get('bmi'),
'diabet': request.data.get('diabet'),
'children': request.data.get('children'),
'smoking': request.data.get('smoking'),
'sumInsurance': request.data.get('sumInsurance'),
'insurancePeriod': request.data.get('insurancePeriod')
}
print(data)
serializer = MedInsuranceSerializer(data=data)
if serializer.is_valid():
    serializer.save()
    born = datetime.strptime(request.data.get('dateofBirth'), '%Y-%m-%d')
    today = date.today()
    age = today.year - born.year - ((today.month, today.day) < (born.month,
born.day))
    result = post_insurnce_plan_med([[age, request.data.get('gender'),
                                     request.data.get('bmi'), request.data.get('diabet'),
                                     request.data.get('children'),
request.data.get('smoking')]])
    return Response(result, status=status.HTTP_201_CREATED)

return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

```

Regression.py

from math import sqrt

import pandas as pd

```

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score,
max_error

df2 = pd.read_csv('D:\diploma 2.0\diploma\diplom\CarIns.csv')
df2 = df2.drop(columns=['cityRegistration'])
nyc_model = df2
nyc_model_x = nyc_model.drop(columns=['Price'])
nyc_model_y = nyc_model['Price'].values
X_reg_train, X_reg_test, y_reg_train, y_reg_test =
train_test_split(nyc_model_x,nyc_model_y , test_size=0.3, random_state=4)

#Linear Regression
linearReg_model = LinearRegression()
linearReg_model.fit(X_reg_train,y_reg_train)
y_reg_pred1 = linearReg_model.predict(X_reg_test)

metrics_df = pd.DataFrame(columns=['Model', 'R2', 'MAE', 'MSE'])

def eval_metrics(actual, pred, model_name):
    mae = mean_absolute_error(actual, pred)
    mse = mean_squared_error(actual, pred)
    r2 = r2_score(actual, pred)
    me = max_error(actual, pred)
    acc = sqrt(r2)
    global metrics_df
    metrics_df = metrics_df.append({'Model': model_name, 'R2': r2, 'MAE': mae,
'MSE': mse, 'Max error': me, 'Accuracy':acc }, ignore_index=True)

```

```

eval_metrics(y_reg_test, y_reg_pred1, 'Linear Regression')

def post_insurnce_plan(Xnew):
    print(Xnew)
    linearY = linearReg_model.predict(Xnew)
    data = pd.read_csv('D:\diploma 2.0\diploma\diplom\CarInsurancePlan.csv')
    price_column = data['Price'].values

    def compare_rate(ind1, ind2):
        sum3 = data.values[ind1][2] + data.values[ind1][3] + data.values[ind1][4] +
data.values[ind1][5] + \
            data.values[ind1][6] + data.values[ind1][7]
        sum1 = data.values[ind2][2] + data.values[ind2][3] + data.values[ind2][4] +
data.values[ind2][5] + \
            data.values[ind2][6] + \
            data.values[ind2][7]
        if sum1 >= sum3:
            ind1 = ind2
        return ind1

    for i in zip(linearY, price_column[:,1]):
        min_error = (100 * (abs(price_column - linearY * 0.9)) / linearY)
        val2, idx2 = min((val2, idx2) for (idx2, val2) in enumerate(min_error))
        print("The cheapest option :", data.values[idx2])

    for i in zip(linearY, price_column[:,1]):
        min_error = (100 * (abs(price_column - linearY)) / linearY)
        val, idx = min((val, idx) for (idx, val) in enumerate(min_error))

```

```

compare_rate(idx, idx2)
print("The most optimal option :", data.values[idx])

for i in zip(linearY, price_column[:,1]):
    min_error = (100 * (abs(price_column - linearY * 1.3)) / linearY)
    val1, idx1 = min((val1, idx1) for (idx1, val1) in enumerate(min_error))
    compare_rate(idx, idx1)
    print("The best option :", data.values[idx1])
    bestOption = data.values[idx1]
    optimalOption = data.values[idx]
    cheapestOption = data.values[idx2]
    for i in range(0, 8):
        if bestOption[i] == 1:
            bestOption[i] = "✓"
        if bestOption[i] == 0:
            bestOption[i] = "✗"
        if optimalOption[i] == 1:
            optimalOption[i] = '✓'
        if optimalOption[i] == 0:
            optimalOption[i] = '✗'
        if cheapestOption[i] == 1:
            cheapestOption[i] = '✓'
        if cheapestOption[i] == 0:
            cheapestOption[i] = '✗'
    result = cheapestOption, optimalOption, bestOption
    return result

```

```
#Xnew=[['1', '1', '1', '1', '1']]
```

```
#post_insurnce_plan(Xnew)
```

```
df2_life = pd.read_csv('D:\diploma 2.0\diploma\diplom\LifeIns.csv')
```

```
nyc_model_life = df2_life
```

```
nyc_model_x_life = nyc_model_life.drop(columns=['claim'])
```

```
nyc_model_y_life = nyc_model_life['claim'].values
```

```
X_reg_train_l,      X_reg_test_l,      y_reg_train_l,      y_reg_test_l      =  
train_test_split(nyc_model_x_life,nyc_model_y_life      ,      test_size=0.3,  
random_state=4)
```

```
#Linear Regression
```

```
linearReg_model = LinearRegression()
```

```
linearReg_model.fit(X_reg_train_l,y_reg_train_l)
```

```
y_reg_pred_l = linearReg_model.predict(X_reg_test_l)
```

```
def post_insurnce_plan_life(Xnew):
```

```
    linearY = linearReg_model.predict(Xnew)
```

```
    data = pd.read_csv('D:\diploma 2.0\diploma\diplom\LifeInsurancePlan.csv')
```

```
    price_colomn = data['Price'].values
```

```
    def compare_rate(ind1, ind2):
```

```
        sum3 = data.values[ind1][2] + data.values[ind1][3] + data.values[ind1][4] +  
data.values[ind1][5] + \
```

```
            data.values[ind1][6]
```

```
        sum1 = data.values[ind2][2] + data.values[ind2][3] + data.values[ind2][4] +  
data.values[ind2][5] + \
```

```
            data.values[ind2][6]
```

```
        if sum1 >= sum3:
```

```

    ind1 = ind2
    return ind1

```

```

for i in zip(linearY, price_column[:,1]):
    min_error = (100 * (abs(price_column - linearY * 0.9)) / linearY)
    val2, idx2 = min((val2, idx2) for (idx2, val2) in enumerate(min_error))
    print("The cheapest option :", data.values[idx2])

```

```

for i in zip(linearY, price_column[:,1]):
    min_error = (100 * (abs(price_column - linearY)) / linearY)
    val, idx = min((val, idx) for (idx, val) in enumerate(min_error))
    compare_rate(idx, idx2)
    print("The most optimal option :", data.values[idx])

```

```

for i in zip(linearY, price_column[:,1]):
    min_error = (100 * (abs(price_column - linearY * 1.3)) / linearY)
    val1, idx1 = min((val1, idx1) for (idx1, val1) in enumerate(min_error))
    compare_rate(idx, idx1)
    print("The best option :", data.values[idx1])
    bestOption = data.values[idx1]
    optimalOption = data.values[idx]
    cheapestOption = data.values[idx2]
    for i in range(0, 7):
        if bestOption[i] == 1:
            bestOption[i] = "✓"
        if bestOption[i] == 0:
            bestOption[i] = "✗"
        if optimalOption[i] == 1:

```

```

        optimalOption[i] = '✓'
    if optimalOption[i] == 0:
        optimalOption[i] = '✗'
    if cheapestOption[i] == 1:
        cheapestOption[i] = '✓'
    if cheapestOption[i] == 0:
        cheapestOption[i] = '✗'
result = cheapestOption, optimalOption, bestOption
return result

```

```

df2_life = pd.read_csv('D:\diploma 2.0\diploma\diplom\LifeIns.csv')
nyc_model_life = df2_life
nyc_model_x_life = nyc_model_life.drop(columns=['claim'])
nyc_model_y_life = nyc_model_life['claim'].values
X_reg_train_l, X_reg_test_l, y_reg_train_l, y_reg_test_l =
train_test_split(nyc_model_x_life,nyc_model_y_life, test_size=0.3,
random_state=4)
#Linear Regression
linearReg_model = LinearRegression()
linearReg_model.fit(X_reg_train_l,y_reg_train_l)
y_reg_pred_l = linearReg_model.predict(X_reg_test_l)

def post_insurnce_plan_life(Xnew):
    linearY = linearReg_model.predict(Xnew)
    data = pd.read_csv('D:\diploma 2.0\diploma\diplom\LifeInsurancePlan.csv')
    price_colomn = data['Price'].values

def compare_rate(ind1, ind2):

```

```

    sum3 = data.values[ind1][2] + data.values[ind1][3] + data.values[ind1][4] +
data.values[ind1][5] + \
    data.values[ind1][6]
    sum1 = data.values[ind2][2] + data.values[ind2][3] + data.values[ind2][4] +
data.values[ind2][5] + \
    data.values[ind2][6]
    if sum1 >= sum3:
        ind1 = ind2
    return ind1

```

```

for i in zip(linearY, price_column[:,1]):
    min_error = (100 * (abs(price_column - linearY * 0.9)) / linearY)
val2, idx2 = min((val2, idx2) for (idx2, val2) in enumerate(min_error))
print("The cheapest option :", data.values[idx2])

```

```

for i in zip(linearY, price_column[:,1]):
    min_error = (100 * (abs(price_column - linearY)) / linearY)
val, idx = min((val, idx) for (idx, val) in enumerate(min_error))
compare_rate(idx, idx2)
print("The most optimal option :", data.values[idx])

```

```

for i in zip(linearY, price_column[:,1]):
    min_error = (100 * (abs(price_column - linearY * 1.3)) / linearY)
val1, idx1 = min((val1, idx1) for (idx1, val1) in enumerate(min_error))
compare_rate(idx, idx1)
print("The best option :", data.values[idx1])
bestOption = data.values[idx1]
optimalOption = data.values[idx]
cheapestOption = data.values[idx2]

```

```

for i in range(0, 7):
    if bestOption[i] == 1:
        bestOption[i] = "✓"
    if bestOption[i] == 0:
        bestOption[i] = "✗"
    if optimalOption[i] == 1:
        optimalOption[i] = '✓'
    if optimalOption[i] == 0:
        optimalOption[i] = '✗'
    if cheapestOption[i] == 1:
        cheapestOption[i] = '✓'
    if cheapestOption[i] == 0:
        cheapestOption[i] = '✗'
result = cheapestOption, optimalOption, bestOption
return result

```

```

df2_med = pd.read_csv('D:\diploma 2.0\diploma\diplom\LifeIns.csv')
nyc_model_med = df2_med
nyc_model_x_med = nyc_model_med.drop(columns=['claim'])
nyc_model_y_med = nyc_model_med['claim'].values
X_reg_train_m, X_reg_test_m, y_reg_train_m, y_reg_test_m =
train_test_split(nyc_model_x_med, nyc_model_y_med, test_size=0.3,
random_state=4)
#Linear Regression
linearReg_model = LinearRegression()
linearReg_model.fit(X_reg_train_m, y_reg_train_m)
y_reg_pred_m = linearReg_model.predict(X_reg_test_m)

```

```

def post_insurnce_plan_med(Xnew):
    linearY = linearReg_model.predict(Xnew)
    data = pd.read_csv('D:\diploma 2.0\diploma\diplom\MedInsurancePlan.csv')
    price_column = data['Price'].values

    def compare_rate(ind1, ind2):
        sum3 = data.values[ind1][2] + data.values[ind1][3] + data.values[ind1][4] +
data.values[ind1][5] + \
            data.values[ind1][6]
        sum1 = data.values[ind2][2] + data.values[ind2][3] + data.values[ind2][4] +
data.values[ind2][5] + \
            data.values[ind2][6]
        if sum1 >= sum3:
            ind1 = ind2
        return ind1

    for i in zip(linearY, price_column[:,1]):
        min_error = (100 * (abs(price_column - linearY * 0.9)) / linearY)
        val2, idx2 = min((val2, idx2) for (idx2, val2) in enumerate(min_error))
        print("The cheapest option :", data.values[idx2])

    for i in zip(linearY, price_column[:,1]):
        min_error = (100 * (abs(price_column - linearY)) / linearY)
        val, idx = min((val, idx) for (idx, val) in enumerate(min_error))
        compare_rate(idx, idx2)
        print("The most optimal option :", data.values[idx])

    for i in zip(linearY, price_column[:,1]):
        min_error = (100 * (abs(price_column - linearY * 1.3)) / linearY)

```

```

val1, idx1 = min((val1, idx1) for (idx1, val1) in enumerate(min_error))
compare_rate(idx, idx1)
print("The best option :", data.values[idx1])
bestOption = data.values[idx1]
optimalOption = data.values[idx]
cheapestOption = data.values[idx2]
for i in range(0, 7):
    if bestOption[i] == 1:
        bestOption[i] = "✓"
    if bestOption[i] == 0:
        bestOption[i] = "✗"
    if optimalOption[i] == 1:
        optimalOption[i] = '✓'
    if optimalOption[i] == 0:
        optimalOption[i] = '✗'
    if cheapestOption[i] == 1:
        cheapestOption[i] = '✓'
    if cheapestOption[i] == 0:
        cheapestOption[i] = '✗'
result = cheapestOption, optimalOption, bestOption
return result

```

Models.py

```
from datetime import date
```

```
from django.db import models
```

```
TERRITORY_CHOICES = (  
    (1, u'All country'),  
    (2, u'All country except the USA and Canada')  
)
```

```
TRIPS_CHOICES = (  
    (1, u'One-time trip'),  
    (2, u'Several trips')  
)
```

```
PEOPLE_CHOICES = (  
    (1, u'1'),  
    (2, u'2'),  
    (3, u'3'),  
    (4, u'4'),  
    (5, u'5')  
)
```

```
PURPOSE_CHOICES =(  
    (1, u'Tourism/Education/Intellectual work'),  
    (2, u'Sport'),  
    (3, u'Physical labor')  
)
```

```
REGISTRATION_CHOICES =(  
    (1, u'No'),  
    (2, u'Yes')  
)
```

```
DIABETES_CHOICES =(  
(1, u'Yes'),  
(2, u'No')  
)
```

```
SMOKE_CHOICES =(  
(1, u'Yes'),  
(2, u'No')  
)
```

```
VEHICLE_CHOICES =(  
(1, u'Passanger car'),  
(2, u'Electric car'),  
(3, u'Motorcycle'),  
(4, u'Passanger bus'),  
(5, u'Truck'),  
(6, u'Trailer')  
)
```

```
CAPACITY_CHOICES =(  
(1, u'less than 1600 sm2'),  
(2, u'1601-2000 sm2'),  
(3, u'2001-3000 sm2'),  
(4, u'more than 3000 sm2')  
)
```

```
TAXI_CHOICES =(  
(1, u'No'),  
(2, u'Yes')
```

)

PROTECTION_CHOICES =(

(1, u'No'),

(2, u'Yes')

)

APARTEMENT_CHOICES =(

(1, u'Flat'),

(2, u'Cottage'),

(3, u'Penthouse'),

(4, u'Semi-detached house')

)

PERIOD_CHOICES =(

(1, u'3 months'),

(2, u'6 months'),

(3, u'12 months')

)

RENTAL_CHOICES =(

(1, u'No'),

(2, u'Yes')

)

GENDER_CHOICES =(

(1, u'Female'),

(2, u'Male'),

(3, u'Gender neutral')

)

```
SUM_CHOICES =(
(1, u'10 000 UAH'),
(2, u'20 000 UAH'),
(3, u'30 000 UAH'),
(4, u'40 000 UAH'),
(5, u'60 000 UAH')
)
```

```
class Client(models.Model):
    name = models.CharField(max_length=60)
    surname = models.CharField(max_length=60)
    username = models.CharField(max_length=60)
    city = models.CharField(max_length=60)
    phonenumber = models.CharField(max_length=60)
    password = models.CharField(max_length=60)
# id = models.IntegerField(primary_key=True, unique=True)
    def __str__(self):
        return self.name
```

```
class TravalInsurance(models.Model):
    dataArrival = models.DateField()
    dataDeparture = models.DateField()
    territoryContract = models.IntegerField(u'Territory of the contract',
choices=TERRITORY_CHOICES, default=1)
```

```

    tripPurpose = models.IntegerField(u'The purpose of the trip',
choices=PURPOSE_CHOICES, default=1)
    numberTrips = models.IntegerField(u'Number of trips',
choices=TRIPS_CHOICES, default=1)
    numberPeople = models.IntegerField(u'Number of people',
choices=PEOPLE_CHOICES, default=1)
    #clientId=models.ForeignKey('Client', on_delete=models.CASCADE,
default=1)
    def __int__(self):
        return self.tripPurpose

```

```

class CarInsurance(models.Model):

```

```

    cityRegistration = models.CharField(max_length=60)
    foreignRegistration = models.IntegerField(u'Foreign registration',
choices=REGISTRATION_CHOICES, default=1)
    typeVehicle = models.IntegerField(u'Vehicle type',
choices=VEHICLE_CHOICES, default=1)
    carCapacity = models.IntegerField(u'Car capacity',
choices=CAPACITY_CHOICES, default=1)
    useTaxi = models.IntegerField(u'Used as a taxi', choices=TAXI_CHOICES,
default=1)
    additionalProtection = models.IntegerField(u'Additional protection',
choices=PROTECTION_CHOICES, default=1)
    #clientId = models.ForeignKey('Client', on_delete=models.CASCADE,
default=1)
    def __int__(self):
        return self.typeVehicle

```

```

class PropertyInsurance(models.Model):

```

```

apartementPrice = models.IntegerField()
apartementArea = models.IntegerField()
apartementType = models.IntegerField(u'Type of accommodation',
choices=APARTEMENT_CHOICES, default=1)
insurancePeriod = models.IntegerField(u'Insurance period',
choices=PERIOD_CHOICES, default=1)
rentalProperty = models.IntegerField(u'Rental property',
choices=RENTAL_CHOICES, default=1)
additionalProtection = models.IntegerField(u'Additional protection',
choices=PROTECTION_CHOICES, default=1)
#clientId = models.ForeignKey('Client', on_delete=models.CASCADE,
default=1)
def __int__(self):
    return self.apartementType

class MedInsurance(models.Model):
    dateofBirth = models.DateField()
    gender = models.IntegerField(u'Gender', choices=GENDER_CHOICES,
default=3)
    bmi = models.FloatField(u'bmi',default=22)
    diabet = models.IntegerField(u'Disease
diabetes',choices=DIABETES_CHOICES, default=2)
    children = models.IntegerField(u'children', default=1)
    smoking = models.IntegerField(u'Smoking', choices= SMOKE_CHOICES,
default=2)
    sumInsurance = models.IntegerField(u'Sum insurance',
choices=SUM_CHOICES, default=1)
    insurancePeriod = models.IntegerField(u'Insurance period',
choices=PERIOD_CHOICES, default=1)

```

```

    #clientId    =    models.ForeignKey('Client',    on_delete=models.CASCADE,
default=1)
    def __int__(self):
        return self.gender

class LifeInsurance(models.Model):
    dateofBirth = models.DateField()
    gender    =    models.IntegerField(u'Gender',    choices=GENDER_CHOICES,
default=3)
    bmi = models.FloatField(u'bmi',default=22)
    diabet          =          models.IntegerField(u'Disease          diabetes',
choices=DIABETES_CHOICES, default=2)
    children = models.IntegerField(u'children', default=1)
    smoking  =  models.IntegerField(u'Smoking',  choices=  SMOKE_CHOICES,
default=2)
    sumInsurance          =          models.IntegerField(u'Sum          insurance',
choices=SUM_CHOICES, default=1)
    insurancePeriod      =      models.IntegerField(u'Insurance      period',
choices=PERIOD_CHOICES, default=1)
    #clientId    =    models.ForeignKey('Client',    on_delete=models.CASCADE,
default=1)
    def __int__(self):
        return self.gender

```

ДОДАТОК В

InsuranseCar.html

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Diploma</title>
  <meta content="" name="description">
  <meta content="" name="keywords">
  <link href="assets/img/favicon.png" rel="icon">
  <link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">
  <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,
600i,700,700i|Raleway:300,300i,400,400i,500,500i,600,600i,700,700i|Poppins:300
,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">
  <link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="assets/vendor/icofont/icofont.min.css" rel="stylesheet">
  <link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
  <link href="assets/vendor/animate.css/animate.min.css" rel="stylesheet">
  <link href="assets/vendor/remixicon/remixicon.css" rel="stylesheet">
  <link          href="assets/vendor/line-awesome/css/line-awesome.min.css"
rel="stylesheet">
  <link href="assets/vendor/venobox/venobox.css" rel="stylesheet">
  <link          href="assets/vendor/owl.carousel/assets/owl.carousel.min.css"
rel="stylesheet">

```

```

<link href="assets/vendor/aos/aos.css" rel="stylesheet">
<link href="assets/css/style.css" rel="stylesheet">
<style>
.btn-circle {
    width: 90px;
    height: 25px;
    border-radius: 50px;
    vertical-align: middle;
    text-align: center;
    display:table-cell;
    background-color: #ef6603;
    border-color: #ef6603;
    font-size: 16px;
}
</style>
</head>
<body>
<header id="header" class="fixed-top d-flex align-items-center header-transparent
" style="background-color: rgba(42, 44, 57, 0.9);">
    <div class="container d-flex align-items-center">

        <div class="logo mr-auto">
            <h1 class="text-light"><a href="Services.html">Diploma</a></h1>
        </div>

        <nav class="nav-menu d-none d-lg-block">
            <ul>
                <li class="active"><a href="index.html">Home</a></li>
                <li><a href="about.html">About</a></li>

```

```

<li><a href="Services.html">Services</a></li>
<li class="drop-down"><a href="">Insurance</a>
  <ul>
    <li><a href="insurance_med.html">Medical insurance</a></li>
    <li><a href="insurance_property.html">Property insurance</a></li>
    <li><a href="insurance_car.html">Car insurance</a></li>
    <li><a href="insurance_travel.html">Travel insurance</a></li>
    <li><a href="insurance_life.html">Life insurance</a></li>
  </ul>
</li>
<li><a href="home.html">Contact</a></li>

</ul>
</nav>

</div>
</header>
<section id="contact" class="contact">
  <div class="container">

    <div class="section-title" data-aos="zoom-out">
      <h2>Welcome!</h2>
      <p>Estimated data</p>
    </div>

    <div class="row mt-5">
      <div class="col-lg-8 mt-5 mt-lg-0" data-aos="fade-left">
        <form class="forms" id="test" method="post" role="form">
          <div class="form-row">
            <div class="col-md-6 form-group">

```

```

        <input type="text" name="cityRegistration" class="form-control"
id="cityRegistration" placeholder="City of registration :" data-msg="Please enter
correct date" required />

        <div class="validate"></div>
    </div>

    <div class="col-md-6 form-group">
        <select class="form-select-lg mb-3" id="foreignRegistration"
name="foreignRegistration" style=" border-radius: 5px; height:38px; width:360px;
border-color: #9a9caba6">
            <option selected> Foreign registration :</option>
            <option value="1">No</option>
            <option value="2">Yes</option>
        </select>
    </div>
</div>

<div class="form-row">
    <div class="col-md-6 form-group">
        <select class="form-select-lg mb-3" id="typeVehicle"
name="typeVehicle" style=" border-radius: 5px; height:38px; width:360px; border-
color: #9a9caba6">
            <option selected>Vehicle type :</option>
            <option value="1">Passanger car</option>
            <option value="2">Electric car</option>
            <option value="3">Motorcycle</option>
            <option value="4">Passanger bus</option>
            <option value="5">Truck</option>
            <option value="6">
        </option>
    </select>

```

```

</div>
<div class="col-md-6 form-group">
  <select      class="form-select-lg      mb-3"      id="carCapacity"
name="carCapacity" style=" border-radius: 5px; height:38px; width:360px; border-
color: #9a9caba6">
    <option selected> Car capacity :</option>
    <option value="1">less than 1600 sm2</option>
    <option value="2">1601-2000 sm2</option>
    <option value="3">2001-3000 sm2</option>
    <option value="4">more than 3000 sm2</option>
  </select>
</div>
</div>
<div class="form-row">
  <div class="col-md-6 form-group">
    <select class="form-select-lg mb-3" id="useTaxi" name="useTaxi"
style=" border-radius: 5px; height:38px; width:360px; border-color: #9a9caba6">
      <option selected> Used as a taxi :</option>
      <option value="1">No</option>
      <option value="2">Yes</option>
    </select>
  </div>
  <div class="col-md-6 form-group">
    <select      class="form-select-lg      mb-3"      id="additionalProtection"
name="additionalProtection" style=" border-radius: 5px; height:38px; width:360px;
border-color: #9a9caba6">
      <option selected>Additional protection: </option>
      <option value="1">No</option>
      <option value="2">Yes</option>
    </select>
  </div>
</div>

```

```

        </select>
    </div>
</div>

    <button type='submit' style=" width: 80px; height: 35px; border-radius:
50px; vertical-align: middle; text-align: center; display:table-cell; color: white;
background-color: #ef6603; border-color: #ef6603; font-size: 16px;"> Submit
</button>
</div>
</div>
</form>
</section>
<footer>
    <section id="hero" style="flex-shrink: 0;">
        <svg      class="hero-waves"      xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"   viewBox="0 24 150 28  "
preserveAspectRatio="none">
            <defs>
                <path id="wave-path" d="M-160 44c30 0 58-18 88-18s 58 18 88 18 58-18
88-18 58 18 88 18 v44h-352z">
            </defs>
            <g class="wave1">
                <use xlink:href="#wave-path" x="50" y="3" fill="rgba(255,255,255, .1)">
            </g>
            <g class="wave2">
                <use xlink:href="#wave-path" x="50" y="0" fill="rgba(255,255,255, .2)">
            </g>
            <g class="wave3">
                <use xlink:href="#wave-path" x="50" y="9" fill="#fff">
            </g>

```

```

    </svg>
</section>
</footer>
<script src="assets/vendor/jquery/jquery.min.js"></script>
<script src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="assets/vendor/jquery.easing/jquery.easing.min.js"></script>
<script src="assets/vendor/php-email-form/validate.js"></script>
<script src="assets/vendor/isotope-layout/isotope.pkgd.min.js"></script>
<script src="assets/vendor/venobox/venobox.min.js"></script>
<script src="assets/vendor/owl.carousel/owl.carousel.min.js"></script>
<script src="assets/vendor/aos/aos.js"></script>
<script src="assets/js/main.js"></script>
<script type="text/javascript">
    $(document).ready(function () {
        $("#test").submit(function (event) {

            var data = $("#test").serializeArray();
            $.ajax({
                type: "POST",
                url: "http://127.0.0.1:8000/CarInsurance/",
                dataType: 'json',
                contentType: 'application/json',
                data: JSON.stringify(getFormData(data)),
                error: function (request, error) {
                    console.log(JSON.stringify(getFormData(data)));
                    console.log( errorThrown );
                },
                success: function(data){
                    console.log("DATA POSTED SUCCESSFULLY" + data);
                }
            });
        });
    });

```

```

        debugger;
        localStorage.setItem("data1",JSON.stringify(data[0]));
        localStorage.setItem("data2",JSON.stringify(data[1]));
        localStorage.setItem("data3",JSON.stringify(data[2]));
        document.location.href = "CarResults.html";
    }
});
return false;
});
function getFormData(data) {
    var unindexed_array = data;
    var indexed_array = {};

    $.map(unindexed_array, function(n, i) {
        indexed_array[n['name']] = n['value'];
    });

    return indexed_array;
}

});
</script>
</body>

</html>
ResultsCar.html
<!DOCTYPE html>
<html lang="en">

```

```

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Diploma</title>
  <meta content="" name="description">
  <meta content="" name="keywords">
  <link href="assets/img/favicon.png" rel="icon">
  <link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">
  <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,
600i,700,700i|Raleway:300,300i,400,400i,500,500i,600,600i,700,700i|Poppins:300
,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">
  <link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="assets/vendor/icomfont/icomfont.min.css" rel="stylesheet">
  <link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
  <link href="assets/vendor/animate.css/animate.min.css" rel="stylesheet">
  <link href="assets/vendor/remixicon/remixicon.css" rel="stylesheet">
  <link          href="assets/vendor/line-awesome/css/line-awesome.min.css"
rel="stylesheet">
  <link href="assets/vendor/venobox/venobox.css" rel="stylesheet">
  <link          href="assets/vendor/owl.carousel/assets/owl.carousel.min.css"
rel="stylesheet">
  <link href="assets/vendor/aos/aos.css" rel="stylesheet">
  <link href="assets/css/style.css" rel="stylesheet">
<style>
.btn-circle {
  width: 90px;
  height: 25px;

```

```
border-radius: 50px;
vertical-align: middle;
text-align: center;
display: table-cell;
background-color: #ef6603;
border-color: #ef6603;
font-size: 16px;
}
.table {
width: 70%;
height: 55%;
position: absolute;
bottom: 200px;
right: 250px;
vertical-align: middle;
border: none;
margin-bottom: 20px;
border-collapse: separate;
}
.table thead th {
font-weight: normal;
text-align: left;
border: none;
padding: 10px 15px;
background: rgba(42, 44, 57, 0.9);
color: white;
font-size: 15px;
border-top: 1px solid #ddd;
}
```

```
.table tr th:first-child, .table tr td:first-child {  
    border-left: 1px solid #ddd;  
}  
.table tr th:last-child, .table tr td:last-child {  
    border-right: 1px solid #ddd;  
}  
.table thead tr th:first-child {  
    border-radius: 20px 0 0 0;  
}  
.table thead tr th:last-child {  
    border-radius: 0 20px 0 0;  
}  
.table tbody td {  
    text-align: left;  
    border: none;  
    padding: 10px 15px;  
    font-size: 14px;  
    vertical-align: top;  
}  
.table tbody tr:nth-child(even) {  
    background: #F8F8F8;  
}  
.table tbody tr:last-child td {  
    border-bottom: 1px solid #ddd;  
}  
.table tbody tr:last-child td:first-child {  
    border-radius: 0 0 0 20px;  
}  
.table tbody tr:last-child td:last-child {
```

```

border-radius: 0 0 20px 0;
}
</style>
</head>
<body>
<header id="header" class="fixed-top d-flex align-items-center header-transparent
" style="background-color: rgba(42, 44, 57, 0.9);">
  <div class="container d-flex align-items-center">

    <div class="logo mr-auto">
      <h1 class="text-light"><a href="Services.html">Diploma</a></h1>
    </div>

    <nav class="nav-menu d-none d-lg-block">
      <ul>
        <li class="active"><a href="index.html">Home</a></li>
        <li><a href="about.html">About</a></li>
        <li><a href="Services.html">Services</a></li>
        <li class="drop-down"><a href="">Insurance</a>
          <ul>
            <li><a href="insurance_med.html">Medical insurance</a></li>
            <li><a href="insurance_property.html">Property insurance</a></li>
            <li><a href="insurance_car.html">Car insurance</a></li>
            <li><a href="insurance_travel.html">Travel insurance</a></li>
            <li><a href="insurance_life.html">Life insurance</a></li>
          </ul>
        </li>
        <li><a href="home.html">Contact</a></li>

```

```
</ul>
```

```
</nav>
```

```
</div>
```

```
</header>
```

```
<section id="pricing" class="pricing">
```

```
<table class="table" id="myTable">
```

```
<thead name="head" id="head">
```

```
<tr>
```

```
<th> Insurance plan </th>
```

```
<th> Classic plan </th>
```

```
<th> Optimal plan </th>
```

```
<th> Business plan </th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<td> Insurance company </td>
```

```
<td> </td>
```

```
<td></td>
```

```
<td> </td>
```

```
</tr>
```

```
<tr>
```

```
<td>Property liability coverage up to UAH 130.000</td>
```

```
<td></td>
```

```
<td></td>
```

```
<td></td>
```

```
<tr>
```

```
<td>24/7 support in case of an accident</td>
```

Organization of medical care
Evacuation in case of breakage to the nearest service station
Payment for treatment in case of an accident
Possibility of additional coverage

```

        <td>Price (UAH) </td>
        <td></td>
        <td></td>
        <td></td>
    </tr>
</tbody>
</table>
</table>
</section>
<footer>
    <section id="hero" style="flex-shrink: 0; position: absolute; bottom: 10px;
right: 15px; ">
        <svg      class="hero-waves"      xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"  viewBox="0 24 150 28  "
preserveAspectRatio="none">
            <defs>
                <path id="wave-path" d="M-160 44c30 0 58-18 88-18s 58 18 88 18 58-18
88-18 58 18 88 18 v44h-352z">
            </defs>
            <g class="wave1">
                <use xlink:href="#wave-path" x="50" y="3" fill="rgba(255,255,255, .1)">
            </g>
            <g class="wave2">
                <use xlink:href="#wave-path" x="50" y="0" fill="rgba(255,255,255, .2)">
            </g>
            <g class="wave3">
                <use xlink:href="#wave-path" x="50" y="9" fill="#fff">
            </g>
        </svg>

```

```

</section><!-- End Hero -->

</footer>

<script src="assets/vendor/jquery/jquery.min.js"></script>
<script src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="assets/vendor/jquery.easing/jquery.easing.min.js"></script>
<script src="assets/vendor/isotope-layout/isotope.pkgd.min.js"></script>
<script src="assets/vendor/venobox/venobox.min.js"></script>
<script src="assets/vendor/owl.carousel/owl.carousel.min.js"></script>
<script src="assets/vendor/aos/aos.js"></script>
<script src="assets/js/main.js"></script>
<script type="text/javascript">
    debugger
    var data1 = JSON.parse(localStorage.getItem("data1"))
    var data2 = JSON.parse(localStorage.getItem("data2"))
    var data3 = JSON.parse(localStorage.getItem("data3"))
    var cell = document.getElementById("myTable").rows[1].cells[1];
    var newText = document.createTextNode(data1[0]);
    cell.appendChild(newText);
    var data = localStorage.getItem("data")
    var cell = document.getElementById("myTable").rows[1].cells[2];
    var newText = document.createTextNode(data2[0]);
    cell.appendChild(newText);
    var data = localStorage.getItem("data")
    var cell = document.getElementById("myTable").rows[1].cells[3];
    var newText = document.createTextNode(data3[0]);
    cell.appendChild(newText);
    var data = localStorage.getItem("data")
    var cell = document.getElementById("myTable").rows[2].cells[1];
    var newText = document.createTextNode(data1[2]);

```

```
cell.appendChild(newText);
var data = localStorage.getItem("data")
var cell = document.getElementById("myTable").rows[2].cells[2];
var newText = document.createTextNode(data2[2]);
cell.appendChild(newText);
var data = localStorage.getItem("data")
var cell = document.getElementById("myTable").rows[2].cells[3];
var newText = document.createTextNode(data3[2]);
cell.appendChild(newText);
var data = localStorage.getItem("data")
var cell = document.getElementById("myTable").rows[3].cells[1];
var newText = document.createTextNode(data1[3]);
cell.appendChild(newText);
var data = localStorage.getItem("data")
var cell = document.getElementById("myTable").rows[3].cells[2];
var newText = document.createTextNode(data2[3]);
cell.appendChild(newText);
var cell = document.getElementById("myTable").rows[3].cells[3];
var newText = document.createTextNode(data3[3]);
cell.appendChild(newText);
var data = localStorage.getItem("data")
var cell = document.getElementById("myTable").rows[4].cells[1];
var newText = document.createTextNode(data1[4]);
cell.appendChild(newText);
var data = localStorage.getItem("data")
var cell = document.getElementById("myTable").rows[4].cells[2];
var newText = document.createTextNode(data2[4]);
cell.appendChild(newText);
var data = localStorage.getItem("data")
```

```
var cell = document.getElementById("myTable").rows[4].cells[3];
var newText = document.createTextNode(data3[4]);
cell.appendChild(newText);
var data = localStorage.getItem("data")
var cell = document.getElementById("myTable").rows[5].cells[1];
var newText = document.createTextNode(data1[5]);
cell.appendChild(newText);
var data = localStorage.getItem("data")
var cell = document.getElementById("myTable").rows[5].cells[2];
var newText = document.createTextNode(data2[5]);
cell.appendChild(newText);
var data = localStorage.getItem("data")
var cell = document.getElementById("myTable").rows[5].cells[3];
var newText = document.createTextNode(data3[5]);
cell.appendChild(newText);
var data = localStorage.getItem("data")
var cell = document.getElementById("myTable").rows[6].cells[1];
var newText = document.createTextNode(data1[6]);
cell.appendChild(newText);
var cell = document.getElementById("myTable").rows[6].cells[2];
var newText = document.createTextNode(data2[6]);
cell.appendChild(newText);
var data = localStorage.getItem("data")
var cell = document.getElementById("myTable").rows[6].cells[3];
var newText = document.createTextNode(data3[6]);
cell.appendChild(newText);
var data = localStorage.getItem("data")
var cell = document.getElementById("myTable").rows[7].cells[1];
var newText = document.createTextNode(data1[7]);
```

```

    cell.appendChild(newText);
    var data = localStorage.getItem("data")
    var cell = document.getElementById("myTable").rows[7].cells[2];
    var newText = document.createTextNode(data2[7]);
    cell.appendChild(newText);
    var data = localStorage.getItem("data")
    var cell = document.getElementById("myTable").rows[7].cells[3];
    var newText = document.createTextNode(data3[7]);
    cell.appendChild(newText);
    var data = localStorage.getItem("data")
    var cell = document.getElementById("myTable").rows[8].cells[1];
    var newText = document.createTextNode(data1[1]);
    cell.appendChild(newText);
    var cell = document.getElementById("myTable").rows[8].cells[2];
    var newText = document.createTextNode(data2[1]);
    cell.appendChild(newText);
    var cell = document.getElementById("myTable").rows[8].cells[3];
    var newText = document.createTextNode(data3[1]);
    cell.appendChild(newText);
</script>
</body>
</body>
</html>

```