

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

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА  
кваліфікаційної роботи ступеня  
магістра

(назва освітньо-кваліфікаційного рівня)

студенту	<i>Горняга Владислав Сергійович</i>
	(ПІБ)
академічної групи	<i>121М-21-1</i>
	(шифр)
спеціальності	<i>121 Інженерія програмного забезпечення</i>
	(код і назва спеціальності)
освітньої програми	<i>Інженерія програмного забезпечення</i>
на тему:	<i>Розробка програмного забезпечення додатку для управління бізнес-процесами з використанням інтелектуальних агентів</i>

*Горняга В.С.*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтин говою	інституційною	
розділів кваліфікаційної роботи				
спеціальний	<i>Проф. Удовик І.М</i>			

Рецензент	<i>Доц. Кожевніков А.В.</i>			
-----------	-----------------------------	--	--	--

Нормоконтролер	<i>Проф. Лактіонов І.С.</i>			
----------------	-----------------------------	--	--	--

Дніпро  
2022

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

---

**ЗАТВЕРДЖЕНО:**

Завідувач кафедри

Програмного забезпечення комп'ютерних систем

(повна назва)

М.О. Алексєєв

(прізвище, ініціали)

(підпис)

«        »     \_\_\_

20 22 року

### ЗАВДАННЯ

**на виконання кваліфікаційної роботи магістра**

**спеціальності**

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

**студенту**

121М-21-1  
 (група)

Горнязі Владиславу Сергійовичу  
 (прізвище та ініціали)

**Тема роботи:** Розробка програмного додатку для управління бізнес-процесами з використанням інтелектуальних агентів.

### 2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт дослідження: процес управління бізнес процесами страхової компанії.

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

Мета роботи: розробити систему для вибору найоптимальнішого страхового плану, використовуючи сучасні методи машинного навчання, знайти та обробити дані для подальшого навчання системи.

### 3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

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

У даній роботі зібрано власнубазу 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.

## 4 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок – кінець)
Аналіз існуючих рішень та постановка задачі роботи	12.09.2022-30.09.2022
Визначення підходів та методів аналізу даних, визначення критеріїв для оцінки якості моделі.	01.10.2022-31.10.2022
Розробка програмного забезпечення та тестування розробки.	01.11.2022-10.12.2022

Завдання видала

\_\_\_\_\_

(підпис)

*Удовик І.М.*

\_\_\_\_\_

(прізвище, ініціали)

Завдання прийняв до виконання

\_\_\_\_\_

(підпис)

*Горняга В.С.*

\_\_\_\_\_

(прізвище, ініціали)

Дата видачі завдання: 10.09.2022 р.Термін подання до ЕК 20.12.2022 р.

## РЕФЕРАТ

Пояснювальна записка: 60 стор., 25 рис., 2 таблиці, 2 додатка, 28 джерел.

Об'єкт дослідження: процес управління бізнес процесами страхової компанії.

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

Мета роботи: розробити систему для вибору найоптимальнішого страхового плану, використовуючи сучасні методи машинного навчання, знайти та обробити дані для подальшого навчання системи.

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. 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.

СПИСОК КЛЮЧОВИХ СЛІВ: WEB-СИСТЕМА, БАЗА ДАНИХ, ER-ДІАГРАМА, ГРАФІЧНИЙ ІНТЕРФЕЙС КОРИСТУВАЧА, АДМІНІСТРАТИВНА ПАНЕЛЬ СИСТЕМИ

## ABSTRACT

Explanatory note: 60 pages, 25 figures, 25 tables, 2 appendices, 28 sources.

Research object: the process of managing business processes of an insurance company.

Research subject: machine learning in the automation of business process management.

The purpose of the work: to develop a system for choosing the most optimal insurance plan using modern machine learning methods, to find and process data for further training of the system.

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

In this work, our own database was collected using the web scraping method, and data on life insurance and health insurance were taken from the kaggle resource.

The result of this work is a software product in the form of an application software interface that finds the most optimal insurance package out of all possible ones. The analysis was carried out from the mathematical side, the definition of the type of problem of intellectual data analysis, the study of existing methods of solving problems of this type, the study of the main criteria of the quality of work. Also, in turn, a detailed analysis of the data used to train the model was carried out, and a correlation analysis of the data used by the model was carried out. To train the model, data from the kaggle service and data from the websites of insurance companies were used.

LIST OF KEYWORDS: WEB SYSTEM, DATA BASE, ER DIAGRAM, GRAPHICAL USER INTERFACE, WEB SYSTEM ADMINISTRATIVE PANEL

## ЗМІСТ

РЕФЕРАТ.....	4
ABSTRACT.....	5
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ РОБОТИ.....	8
1.2 Актуальність розробки програмного забезпечення.....	8
1.2. Аналіз існуючих підходів до вирішення задачі автоматизації вибору страхового плану.....	12
1.3. Постановка задачі роботи.....	15
РОЗДІЛ 2. РОЗБУДОВА БІЗНЕС-ПРОЦЕСІВ ТА МЕТОДИ РОЗВ'ЯЗАННЯ ЗАДАЧ НА ОСНОВІ АНАЛІЗУ ДАНИХ.....	16
2.1. Необхідність автоматизації бізнес-процесів .....	16
2.2. Основи аналізу даних для автоматизації вибору страхового плану.....	17
2.3. Завдання аналізу даних для автоматизації вибору страхового плану.....	21
2.4 Методи та критерії якості роботи моделі.....	25
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЙОГО ЗАСТОСУВАННЯ.....	33
3.1. Обґрунтування вибору платформи та мови програмування.....	33
3.2. Збір та аналіз даних та навчання моделі .....	34
3.3. Керівництво користувача кінцевим програмним продуктом.....	38
3.4. Тестування системи.....	35
ВИСНОВКИ.....	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	44
Додаток А. Код програми.....	47
Додаток Б. Перелік файлів на диску.....	60

## 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

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ РОБОТИ

#### 1.1 Актуальність розробки програмного забезпечення.

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 однієї 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%.



Рис. 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%, автостраховання 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).

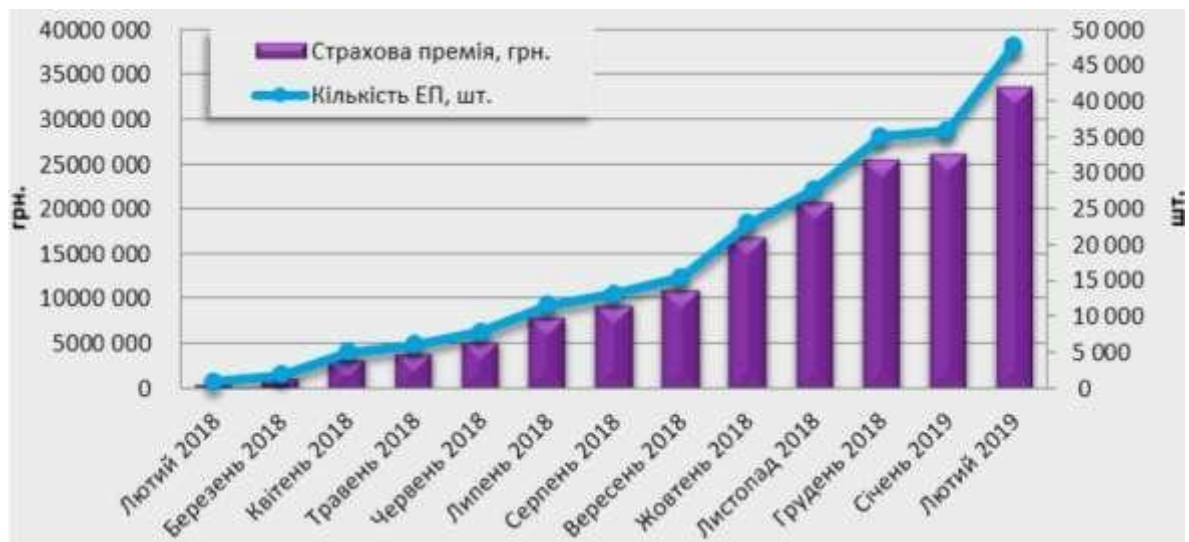


Рис. 1.2. Статистика продажів електронних договорів ОСАГО

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).

Рис. 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

Програми страхування	462.50 грн.	656.91 грн.	763.50 грн.
	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 подовження перебування		✓	✓
Дострокове 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

Рис. 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).

Рис. 1.5. Онлайн-калькулятор страхового сервісу Parasol.ua

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

Рис. 1.6. Онлайн-калькулятор страхового сервісу 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.

## РОЗДІЛ 2

### РОЗБУДОВА 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.



Рис. 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.

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, ВМІ (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.

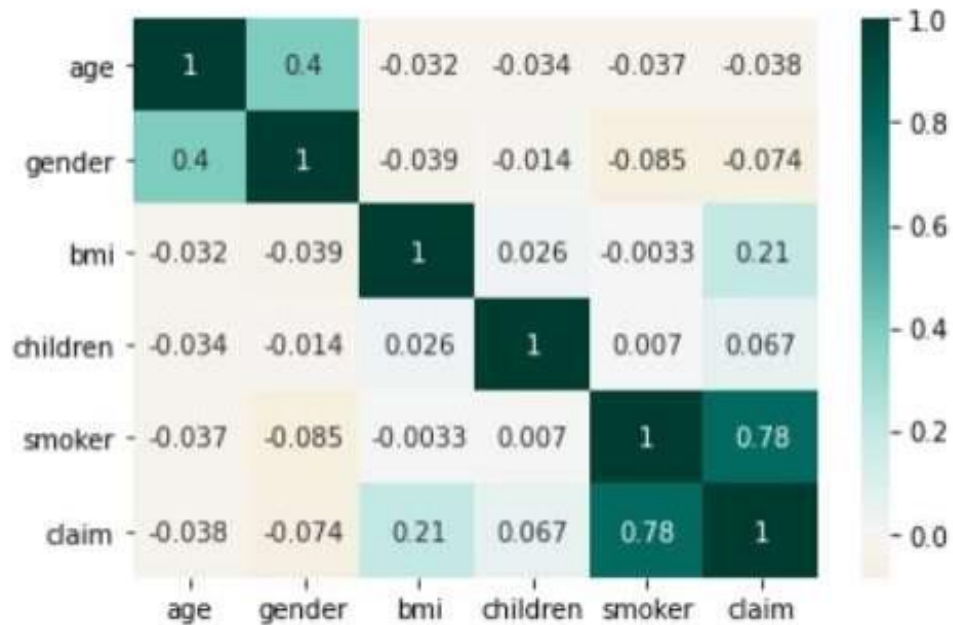


Рис. 2.2. Матриця кореляцій для датасету «Страхування життя»

Показник  $R^2$  показує якою мірою дисперсія однієї величини обумовлена впливом коефіцієнта детермінації іншою змінною. 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 вирушує 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 знаходження 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.



Рис. 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).

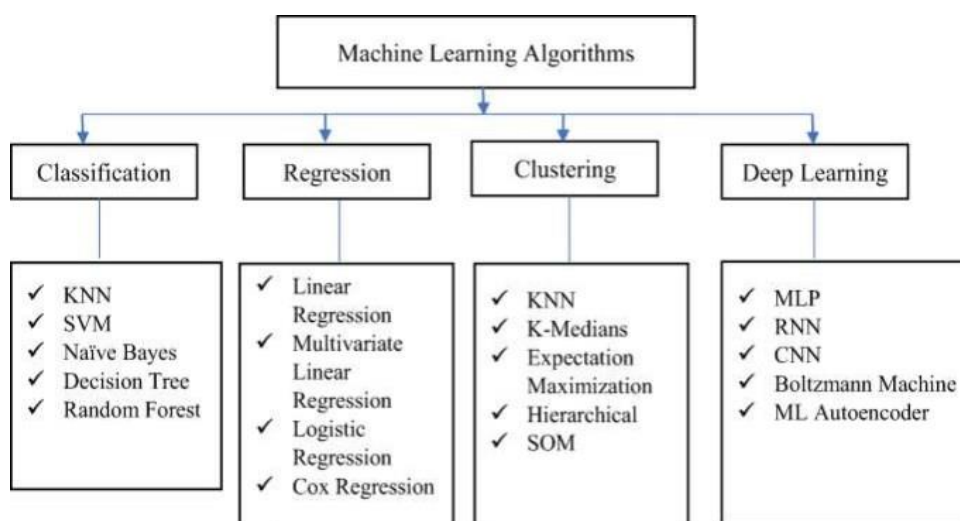


Рис. 2.4. Класифікація машинного навчання та методів



### 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.

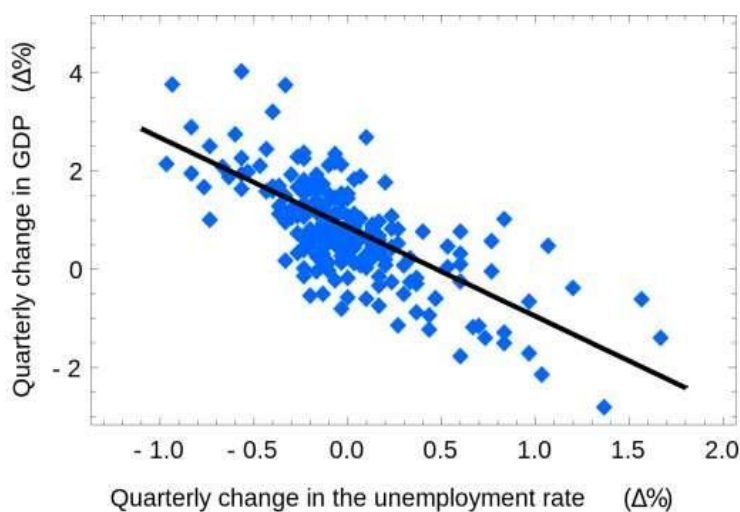


Рис. 2.5. –Приклад простої лінійної регресії

Для підбору параметрів даного рівняння зачасту 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).

Лінійна 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:

$$R^2 = \frac{\sigma_{\text{регр}}^2}{\sigma^2} \quad (2.2) \text{ у}$$

де  $\sigma_{\text{регр}}^2$  - дисперсія, що показує регресія;

$\sigma_y^2$  - дисперсія помилок.

Середньоквадратична похибка моделі регресії (MSE) розраховується за формулою:

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

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:

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

де  $y$  – залежна змінна;

$n$  – кількість елементів вибірки.

Якщо величина даного показника становить менше 0.07 то побудовано 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.

### 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.

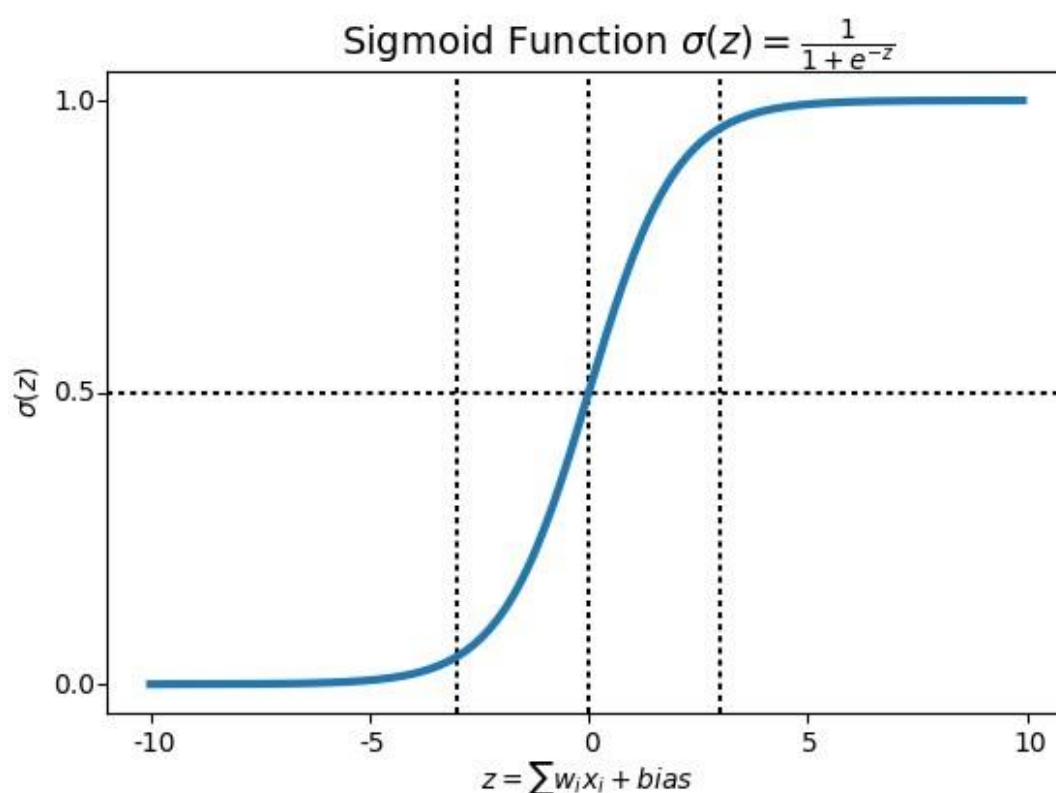


Рис. 2.6. Приклад логістичної функції

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.

### 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 задача 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 (рисунок 2.7). Таке розбиття даних відбувається щоб збільшити кількість інформації, при цьому зменшити ентропію в попередньо утворених групах за ознаками.



Рис. 2.7. Приклад структури дерева рішень

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.

## РОЗДІЛ 3

### РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЙОГО ЗАСТОСУВАННЯ

#### 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 оренди 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.

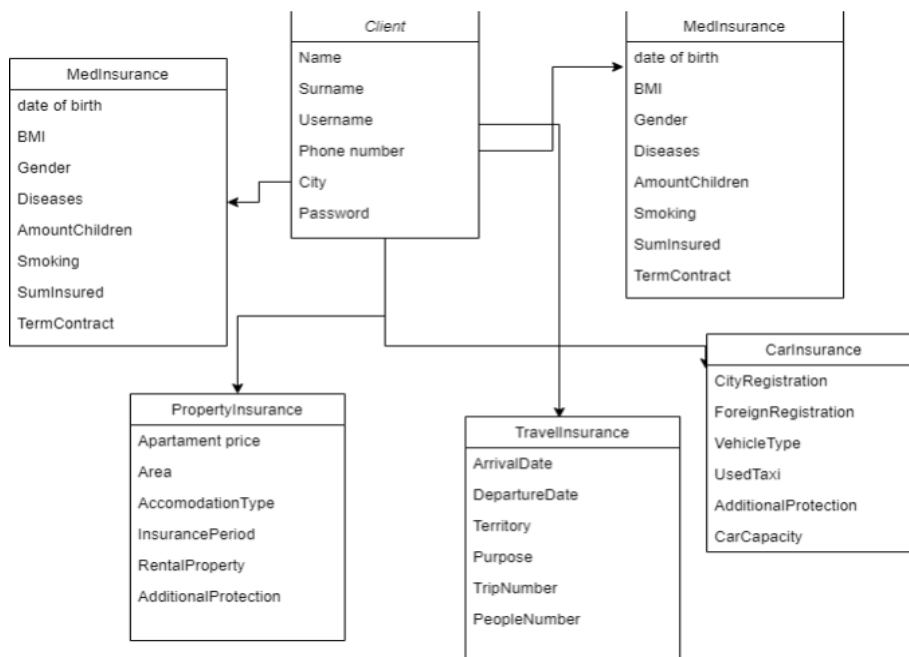


Рис. 3.1. База даних системи

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	ddditionalProtectio	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

Рис. 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.

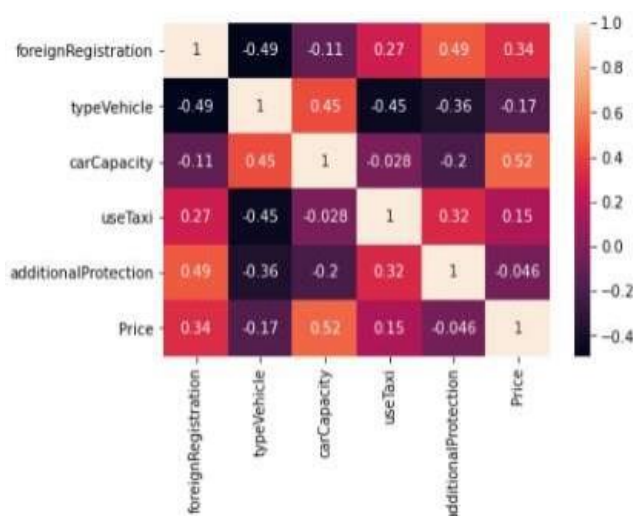


Рис. 3.3. Матриця кореляцій даних страхування рухомого майна

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  $R^2$ .

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

Рис. 3.4. Значення метрик якості для моделі

### 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”.



Рис. 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, користувачке 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 користувачке 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.

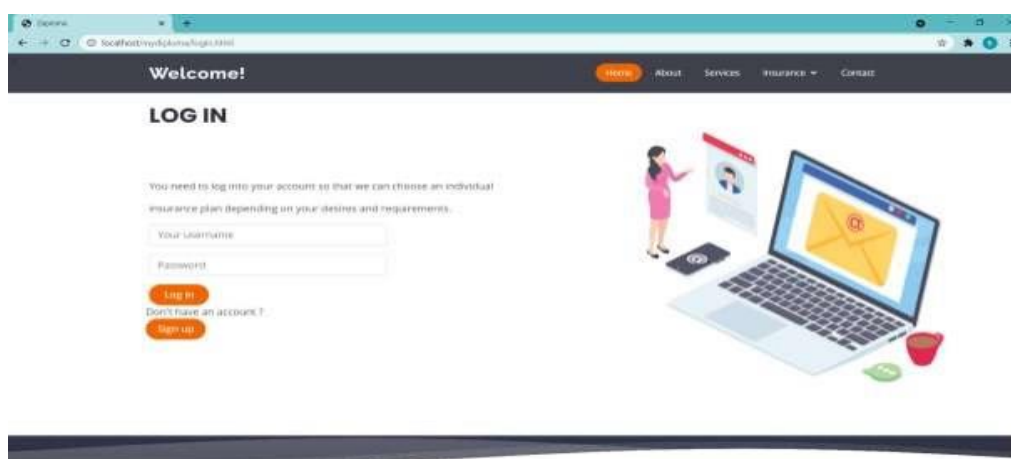


Рис. 3.6. Сторінка для входу в обліковий запис

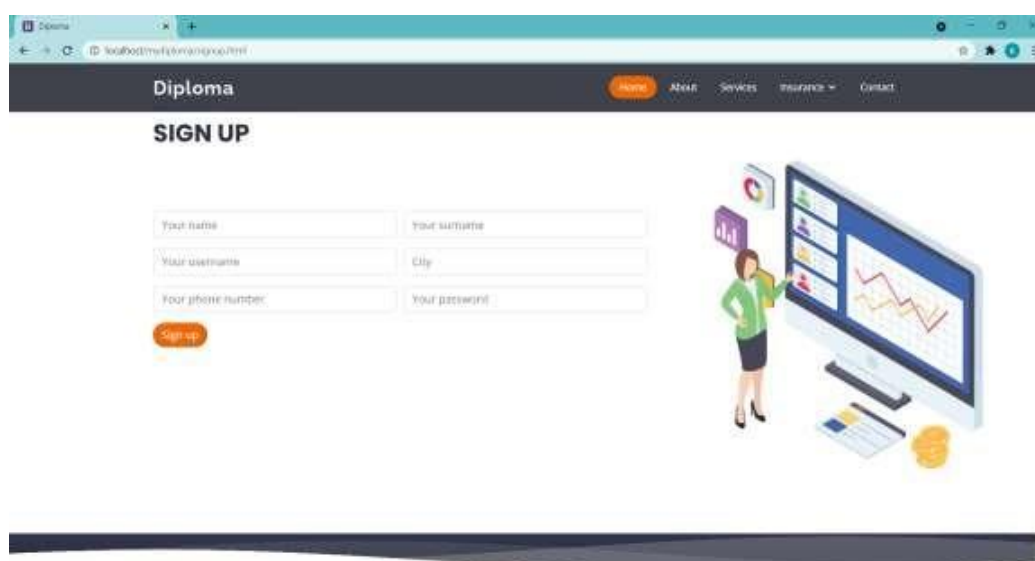


Рис. 3.7. Сторінка для створення нового облікового запису

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).

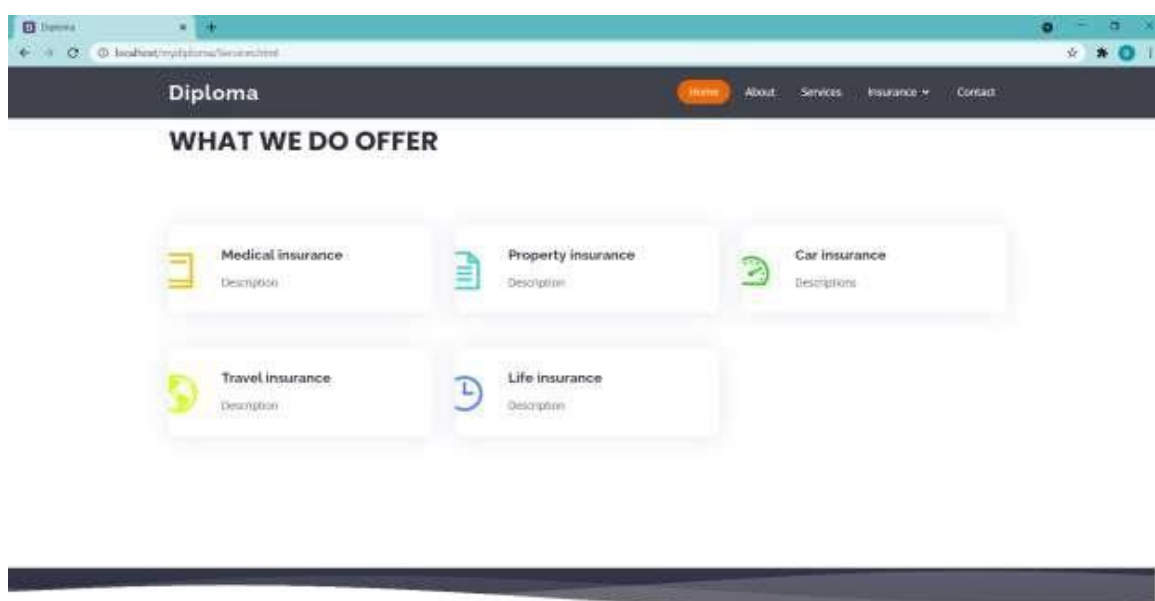


Рис. 3.8. Обрання категорії страхування

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/my diploma/insurance.html`. The page header features the 'Diploma' logo and navigation links: Home, About, Services, Insurance, and Contact. The main heading is 'ESTIMATED DATA'. The form contains the following fields:

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

A 'Submit' button is positioned at the bottom left of the form area.

Рис. 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/my diploma/insurance_property.html`. The page header features the 'Diploma' logo and navigation links: Home, About, Services, Insurance, and Contact. The main heading is 'ESTIMATED DATA'. The form contains the following fields:

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

A 'Submit' button is positioned at the bottom left of the form area.

Рис. 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 screenshot shows a web browser window with the URL 'localhost:3000/diploma/insurance.html'. The page title is 'Diploma' and the navigation menu includes 'Home', 'About', 'Services', 'Insurance', and 'Contact'. The main heading is 'ESTIMATED DATA'. Below it, there is a form with the following fields:

- Date of arrival: (calendar icon)
- Departure date: (calendar icon)
- Territory of the contract: (dropdown menu)
- The purpose of the trip: (dropdown menu)
- Number of trips: (dropdown menu)
- Number of people: (dropdown menu)
- Submit: (orange button)

Рис. 3.11. Форма для заповнення користувачем інформації про туристичне страхування

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 3.12).

Insurance plan	Classic plan	Optimal plan	Business plan
Insurance company	AIG	UNICA	INGO
Injury insurance	✓	✓	✓
Disease insurance	✓	✓	✓
Loss of life	✓	✓	✓
Unlimited online consultations with a therapist	✓	✗	✓
International medical consultation	✗	✗	✗
Price (USD)	4999	6120	7020

Рис. 3.12. Результат роботи програми

## 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 ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Trends in the Information Technology sector report by Center for Technology Innovation // 2021 URL: <https://www.brookings.edu/research/trends-in-the-information-technology-sector> (дата звернення 02.12.2022).
- 2 Logue C. Integrating IT into process manufacturing [Електронний ресурс] / Christopher Logue // 2021 URL: <https://www.isa.org/intech-home/2019/january-february/features/integrating-it-into-process-manufacturing>. (дата звернення 02.12.2022).
- 3 Kia Motors // 2021 URL: <https://www.kia.com/ua/util/news.detail.html>. (дата звернення 02.12.2021).
- 4 CAD/CAM системи // 2021 URL: <https://www.autodesk.ru/solutions/cad-cam..> (дата звернення 02.12.2022).
- 5 AutoCAD // 2021 URL: <https://www.autodesk.ru/products/autocad/overview..> (дата звернення 02.12.2022).
- 6 AutoCAD Web App // 2021 URL: <https://www.autodesk.com/products/autocad-web-app/overview> (дата звернення 02.12.2022).
- 7 Конструкторське бюро КАЙТЕК // 2021 URL: <https://caetec.ru/calconline/raschet-czilindricheskoj-pruzhiny.html>. (дата звернення 02.12.2022).
- 8 Калькулятор нержавіючого металу // 2021 URL: <https://westa.kiev.ua/calc> (дата звернення 02.12.2022).
- 9 Google Форми // 2021 URL: [https://www.google.com/intl/ru\\_ua/forms/about/](https://www.google.com/intl/ru_ua/forms/about/) (дата звернення 02.12.2022).
- 10 Open Server Panel 5.4.0 // 2021 URL: <https://ospanel.io/> (дата звернення 02.12.2022).
- 11 phpMyAdmin 4.9.6 and 5.0.3 are released // 2021 URL: <https://www.phpmyadmin.net/> (дата звернення 02.12.2022).
- 12 Що таке MySQL // 2021 URL: <https://www.hostinger.com.ua/rukovodstva/shto-takoje-mysql/> (дата звернення 02.12.2022).

- 13 HTML5 // 2021 URL: <http://htmlbook.ru/html5> (дата звернення 02.12.2021).
- 14 Основи CSS // 2021 URL: [https://developer.mozilla.org/ru/docs/Learn/Getting\\_started\\_with\\_the\\_web/CSS\\_basics](https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics) (дата звернення 05.12.2021).
- 15 JavaScript Projects for Beginners// 2021 URL: <https://www.freecodecamp.org/news/javascript-projects-for-beginners/> (дата звернення 05.12.2022).
- 16 PHP Manual // 2021 URL: <https://www.php.net/manual/en/index.php> (дата звернення 02.12.2022).
- 17 Sublimetext // 2021 URL: <https://www.sublimetext.com/> (04.12.2022).
- 18 Introduction ·Bootstrap v5.0 // 2021 URL: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
- 19 Методологія IDEF0 // 2017 URL: [https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6\\_2](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2) (дата звернення 22.11.2022).
- 20 Основи методології IDEF0 // 2018 URL: <http://easy-code.com.ua/2011/05/osnovi-metodologi%D1%97-idef1> (дата звернення 22.11.2021).
- 21 Проектування інформаційних систем // 2017 URL: <http://eprints.cdu.edu.ua/1481/1/pro.pdf> (дата звернення 22.11.2022).
- 22 Діаграма варіантів використання (USE case diagram) // 2017 URL: [https://studopedia.ru/19\\_284009\\_diagrama-variantiv-vikoristannya-USE-case-diagram.html](https://studopedia.ru/19_284009_diagrama-variantiv-vikoristannya-USE-case-diagram.html) (дата звернення 22.11.2022).
- 23 Entity Relationship Diagram (ERD) - What is an ER Diagram? // 2020 URL: <https://www.smartdraw.com/entity-relationship-diagram/> (дата звернення 22.11.2022).
- 24 E. Lavrov, O. Siryk, Y. Chybiriak, L. Danilova, V. Nahornyi and S. Vakal, "A Model for the Organization of Adaptive Dialogue Interaction "Man-Computer" Taking into Account the Requirements of Reliability and Efficiency," 2021 IEEE 4th International Conference on Advanced Information and Communication Technologies (AICT), 2021, pp. 31-35, doi: 10.1109/AICT52120.2021.9628939.

- 25 Chybiriak, Y., Baranova I., & Nikolaenko K. (2021). A method of end-to-end training in simulation modeling using FlexSim to find reserves for improving the efficiency of automated systems, offered for IT students . *Computer-Integrated Technologies: Education, Science, Production*, (42), 119-129. <https://doi.org/10.36910/6775-2524-0560-2021-42-17>
- 26 Marchenko A., Antypenko V., Vashchenko S., Fedotova N., Chybiriak Y., Krasulia A. (2021) A Complex Model of Blended Learning: Using a Project Approach to Organize the Educational Process. In: Lopata A., Gudonienė D., Butkienė R. (eds) *Information and Software Technologies. ICIST 2021. Communications in Computer and Information Science*, vol 1486. Springer, Cham. [https://doi.org/10.1007/978-3-030-88304-1\\_21](https://doi.org/10.1007/978-3-030-88304-1_21)
- 27 Chybiriak Y., Konoplianchenko I., & Marchenko A. (2020). Technological patterns and mathematical models of the synthesis of a rational sequence of product assembly . *Computer-Integrated Technologies: Education, Science, Production*, (39), 110-116. <https://doi.org/10.36910/6775-2524-0560-2020-39-19>
- 28 Nahorny, V. V., Lavrov, E. A., & Chybiriak, Y. I. (2019). Forecasting Individual Resource Of Technical Systems. *Radio Electronics, Computer Science, Control*, (1). <https://doi.org/10.15588/1607-3274-2019-1-5>

**Лістинг програми**

## ЛІСТИНГ КОДУ КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ

```

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, TravalInsuranceSerializer,
CarInsuranceSerializer, PropertyInsuranceSerializer, LifeInsuranceSerializer,
MedInsuranceSerializer
from .models import Client, TravalInsurance, 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')
class PropertyInsuranceViewSet(viewsets.ModelViewSet):
    queryset = PropertyInsurance.objects.all().order_by('apartementType')
    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):
        '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_insurance_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 = {

            '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_insurance_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
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
```

```
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_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] + 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_colomn[:,1]):
```

```
    min_error = (100 * (abs(price_colomn - 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_colomn[:,1]):
```

```
    min_error = (100 * (abs(price_colomn - linearY)) / linearY)
```

```
compare_rate(idx, idx2)
```

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

```
for i in zip(linearY, price_colomn[:,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

#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_1, X_reg_test_1, y_reg_train_1, y_reg_test_1 =
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_1,y_reg_train_1) y_reg_pred_1 =
linearReg_model.predict(X_reg_test_1)

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]

    ind1 = ind2
    return ind1

for i in zip(linearY, price_colomn[:,1]):
    min_error = (100 * (abs(price_colomn - 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_colomn[:,1]):
    min_error = (100 * (abs(price_colomn - 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_colomn[:,1]):
    min_error = (100 * (abs(price_colomn - 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] = "✗"

    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_colomn[:,1]):
    min_error = (100 * (abs(price_colomn - 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_colomn[:,1]):
    min_error = (100 * (abs(price_colomn - 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_colomn[:,1]):
    min_error = (100 * (abs(price_colomn - 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]

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)
```

```
def post_insurnce_plan_med(Xnew):
```

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

```
    data = pd.read_csv('D:\diploma 2.0\diploma\diplom\MedInsurancePlan.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_colomn[:,1]):
```

```
    min_error = (100 * (abs(price_colomn - 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_colomn[:,1]):
```

```
    min_error = (100 * (abs(price_colomn - 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_colomn[:,1]):
```

```

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
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') )

**ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ**

<b>Ім'я файла</b>	<b>Опис</b>
Пояснювальні документи	
Кваліфікаційна робота.doc	Пояснювальна записка. Документ Word.
Кваліфікаційна робота.pdf	Пояснювальна в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація.ppt	Презентація