МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
В.о. завідувача кафедри інтелектуальних інформаційних систем, канд. техн. наук, доцент
_____ Є. В. Сіденко
«____»_____2023 року

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПРОГНОЗУВАННЯ МЕДИЧНИХ ВИДАТКІВ ДЛЯ СТРАХОВОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Спеціальність 122 «Комп'ютерні науки»

122 – МКР – 601.1710225

Виконав студент 6-го курсу, групи 601
_____В.А Пронін
«__16__» лютого 2023 р.
Керівник канд. техн. наук, доцент
_____ І.О. Калініна
«__16__» лютого 2023 р.

Миколаїв – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Освітньо-кваліфікаційний рівень **магістр**
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри інтелектуальних інформаційних систем, канд. техн. наук, доцент
_____ Є. В. Сіденко
«____» _____ 20__ р.

З А В Д А Н Н Я
на магістерську кваліфікаційну роботу

Проніну Валентину Андрійовичу

1. Тема магістерської кваліфікаційної роботи «Інтелектуальна система прогнозування медичних видатків для страхового забезпечення на основі методів машинного навчання».

Керівник роботи Калініна Ірина Олександрівна, кандидат техн. наук, доцент

Затв. наказом Ректора ЧНУ ім. Петра Могили від «03» листопада 2022 р. № 199

2. Строк подання студентом роботи ___ лютого _____ 20_23_ р.

3. Вхідні (початкові) дані до роботи: загальні відомості машинного навчання, прогнозування та інтелектуальних систем.

Очікуваний результат роботи: створення системи прогнозування видатків на основі вибіркової бази даних.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

5. Перелік графічного матеріалу: презентація, рисунки, таблиці.

6. Завдання до спеціальної частини: Охорона праці та безпека у надзвичайних ситуаціях.

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Григор'єва Л. І., д. б. н., професор	
Методична частина	Калініна І. О. , доцент к. т. н	

Керівник роботи _канд. техн. наук, доцент, Калініна І.О.

(наук. ступінь, вчене звання, прізвище та ініціали)

_____

(підпис)

Завдання прийнято до виконання ___Пронін В.А

(прізвище та ініціали)

_____

(підпис)

Дата видачі завдання «_____» _____ 20__ р.

КАЛЕНДАРНИЙ ПЛАН

виконання магістерської кваліфікаційної роботи

Тема: «Інтелектуальна система прогнозування медичних видатків для страхового забезпечення на основі методів машинного навчання»

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Визначення керівника і теми МКР. Подання заяви на затвердження теми МКР	01.09.2022	20.10.2022	Виконано
2.	Отримання завдання на виконання МКР	21.10.2022	10.11.2022	Виконано
3.	Складання календарного плану	11.11.2022	15.11.2022	Виконано
4.	Огляд літератури за темою дослідження	18.11.2022	27.11.2022	Виконано
5.	Проходження переддипломної практики, збір та аналіз матеріалів до МКР	28.11.2022	18.12.2022	Виконано
6.	Аналіз предметної області та розробка технічного завдання.	20.12.2022	22.12.2022	Виконано
7.	Програмна реалізація системи прогнозування	15.01.2022	15.01.2023	Виконано
8.	Робота над розділами фаховох частини	16.01.2023	24.12.2023	Виконано
9.	Попередній захист МКР	27.01.2023	01.02.2023	Виконано
10.	Обговорення отриманих результатів з керівником та попередній захист МКР	03.02.2023	3.02.2023	Виконано
11.	Корегування роботи за результатами попереднього захисту	4.02.2023	6.02.2023	Виконано
12.	Остаточне оформлення пояснювальної записки та слайдів доповіді до зихисту	7.02.2023	9.02.2023	Виконано
13.	Подання рецензенту та рецензування МКР	9.02.2023	12.02.2023	Виконано
14.	Подання МКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	15.02.2023	16.02.2023	Виконано
15.	Захист МКР перед ЕК	23.02.2023	23.02.2023	Виконано

Розробив студент Пронін В.А. _____
(прізвище та ініціали) (*підпис*)

Керівник роботи канд.тех. наук, доцент Калініна І.О. _____
(наук. ступінь, вчене звання, прізвище та ініціали) *(підпис)*

«_12_» ___листопада__ 2022 р.

АНОТАЦІЯ

до магістерської кваліфікаційної роботи
студента групи 601 ЧНУ ім. Петра Могили

Проніна Валентина Андрійовича

на тему: **“ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПРОГНОЗУВАННЯ МЕДИЧНИХ ВИДАТКІВ ДЛЯ СТРАХОВОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ”**

Актуальність даного дослідження полягає у необхідності підвищення якості прогнозування медичних видатків за рахунок методів машинного навчання. Таким чином страхові компанії матимуть змогу більш точно контролювати сплати за лікування різних груп населення, та проводити швидкий аналіз різного спектру людей на основі затверджених даних.

Об’єктом дослідження є процес прогнозування видатків для страхового забезпечення.

Предметом дослідження є методи прогнозування на основі машинного навчання.

Метою є підвищення якості прогнозування медичних видатків для страхового забезпечення за рахунок використання методів машинного навчання.

В результаті виконання роботи було розроблено систему прогнозування для збору даних, аналізу та інтерпретації, яка на основі цих даних реалізує описову статистику та візуалізацію залежностей між ознаками.

Дана робота складається з шести розділів. Кожен розділ відповідно присвячений: аналізу предметної області, математичним моделям і методам, використаним у магістерській роботі, розробці і візуалізації системи, аналізу отриманих результатів, охороні праці, методичній частині магістерської роботи. Загальний обсяг роботи – 82 сторінок. Магістерська кваліфікаційна робота містить додаток 1, рисунків − 26 , посилання на літературних джерел − 43.

***Ключові слова**: система, машинне навчання, метод, інтерпретатор, прогнозування.*

ABSTRACT

to the master's qualification work by the student of the group 601 of Petro Mohyla Black Sea National University

Pronin Valentyn

“INTELLIGENT MEDICAL EXPENDITURE FORECASTING SYSTEM FOR INSURANCE BASED ON MACHINE LEARNING METHODS”

A relevance of this study lies in the need to improve the quality of forecasting insurance costs due to machine learning methods. In this way, insurance companies will be able to more accurately monitor payments for the treatment of different types of population, and conduct rapid analysis of a diverse range of people based on validated data, enabling insurance companies to spend less time and money on developing predictive models.

An object of research is an intelligent system of forecasting expenses for insurance provision.

A subject of the research is forecasting methods based on machine learning.

A purpose of this research of this work is to improve the quality of forecasting medical expenses for insurance coverage through the use of machine learning methods. The study uses patient data to estimate average health care costs for different segments of the population.

As a result of the work, a forecasting system was developed for data collection, analysis and interpretation, which, based on these data, implements descriptive statistics and visualization of dependencies between features.

This work consists of six sections. Each section is respectively devoted to: analysis of the subject area, mathematical models and methods used in the master's work, development and visualization of the system, analysis of the obtained results, labor protection, methodical part of the master's work. The total amount of work - ?? pages. Master's qualification work contains 82 app, 43 drawings 26, and link to 43 sources.

Keywords: *system, machine learning, method, interpreter, forecast*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ІС – Інтелектуальна система

МН – Машинне Навчання

SL – Supervised learning

BMI – Body Mass Index

CSV – Comma-Separated Values

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему:

«ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПРОГНОЗУВАННЯ МЕДИЧНИХ ВИДАТКІВ ДЛЯ СТРАХОВОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ»

Спеціальність 122 «Системний аналіз»

122 – МКР – 601.21710225

Виконав студент 6-го курсу, групи 601
_____ В.А. Пронін
«16» лютого 2023 р.

Керівник: канд. техн. наук, доцент
_____І.О. Калініна
«16» лютого 2023 р.

Миколаїв – 2023

ВСТУП

Для того, щоб медична страхова компанія мала змогу заробляти гроші, вона повинна збирати щорічні внески більше, ніж витрачає на медичне обслуговування своїх бенефіціарів. Таким чином, страховикам необхідно витрачати багато часу та грошей на розробку моделей, які точно прогнозують медичні витрати населення, яке застраховане.

Витрати на лікування важко оцінити, оскільки є рідкісні випадки захворювань. Проте існує статистика проблем із здоров'ям, які є більш частими серед населення. Або випадки, які є більш поширеними для певної групи людей. Наприклад, рак легенів більш ймовірний серед курців, ніж людей, які ведуть здоровий спосіб життя, а захворювання серця можуть бути більш ймовірними серед людей, які мають таку проблему як ожиріння. У групу ризику також входять люди зі шкідливими звичками та низькою стресостійкістю.

Метою цієї роботи є підвищення якості прогнозування медичних видатків для страхового забезпечення за рахунок використання методів машинного навчання. В дослідженні використовуються дані пацієнтів для оцінки середніх витрат на медичне обслуговування для різних сегментів населення. Ці оцінки можна використовувати для створення страхових таблиць, які встановлюють вищу або нижчу суму щорічних внесків залежно від очікуваних витрат на лікування.

1 ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ ТА МАШИННЕ НАВЧАННЯ, ЇХ МЕТОДИ ТА КЛАСИФИКАЦІЇ

1.1 Поняття інтелектуальної системи, її види на класифікації

Інтелектуальна система (ІС, англ. intelligent system) − це технічна або програмна система, здатна вирішувати завдання, що традиційно вважаються творчими, що належать конкретної предметної області, знання про яку зберігаються в пам'яті такої системи. Структура інтелектуальної системи включає три основні блоки − базу знань, механізм виведення рішень та інтелектуальний інтерфейс.

Інтелектуальні системи вивчаються групою наук, які об'єднуються під назвою «штучний інтелект».

У технологіях прийняття рішень інтелектуальна система – це інформаційно-обчислювальна система з інтелектуальною підтримкою, яка вирішує завдання без участі людини – особи, яка приймає рішення (ЛПР), на відміну від інтелектуалізованої системи, в якій оператор є:

- види інтелектуальних систем;
- інтелектуальна інформаційна система;
- експертна система;
- розрахунково-логічні системи;
- гібридна інтелектуальна система;
- рефлекторна інтелектуальна система.

До розрахунково-логічним системам відносять системи, здатні вирішувати управлінські та проектні завдання за декларативними описами умов. При цьому користувач має можливість контролювати у режимі діалогу всі стадії обчислювального процесу. Дані системи здатні автоматично будувати математичну модель задачі та автоматично синтезувати обчислювальні алгоритми формулювання завдання intelligent systems in production processes. Ці властивості

реалізуються завдяки наявності бази знань у вигляді функціональної семантичної системи та компонентів дедуктивного виведення та планування. Використання інтелектуальних систем для розпізнавання тексту на зображенні.

1.2 Машинне навчання систем

Нейронна система математична модель, а також її програмне або апаратне втілення, побудована за принципом організації та функціонування біологічних нейронних систем − систем нервових клітин живого організму. Це поняття виникло щодо процесів, які у мозку, і за спробі змоделювати ці процеси. Першою такою спробою були нейронні системи У. Маккалока та У. Піттса. Після розробки алгоритмів навчання одержувані моделі почали використовувати у досить простих (особливо порівняно з процесорами, які у практичних цілях: у завданнях прогнозування, для розпізнавання образів, у завданнях управління та ін.

ІНС є системою з'єднаних і взаємодіючих між собою простих процесорів (штучних нейронів). Такі процесори зазвичай персональних комп'ютерах). Кожен процесор подібної системи має справу лише з сигналами, які він періодично отримує, та сигналами, які він періодично надсилає іншим процесорам. І, тим щонайменше, будучи з'єднаними досить велику систему з керованим взаємодією, такі окремі прості процесори разом здатні виконувати досить складні завдання:

— з точки зору машинного навчання, нейронна система є окремим випадком методів розпізнавання образів, дискримінантного аналізу;

— з погляду математики навчання нейронних систем − це багатопараметричне завдання нелінійної оптимізації;

— з погляду кібернетики, нейронна система використовується у завданнях адаптивного управління та як алгоритми для робототехніки;

— з погляду розвитку обчислювальної техніки та програмування, нейронна система − спосіб вирішення проблеми ефективного паралелізму;

— з погляду штучного інтелекту, ІНС є основою філософської течії коннекціонізму та основним напрямом у структурному підході щодо вивчення можливості побудови (моделювання) природного інтелекту за допомогою комп'ютерних алгоритмів.

Нейронні системи не програмуються у звичному значенні цього слова, вони навчаються. Можливість навчання − одна з головних переваг нейронних систем перед традиційними алгоритмами. Технічно навчання полягає у знаходженні коефіцієнтів зв'язків між нейронами. У процесі навчання нейронна система здатна виявляти складні залежності між вхідними даними та вихідними, а також виконувати узагальнення. Це означає, що у разі успішного навчання система зможе повернути правильний результат на підставі даних, які були відсутні у навчальній вибірці, а також неповних та/або «зашумлених», частково спотворених даних.

Розпізнавання та класифікації.

Дані можуть виступати різними за своєю природою об'єктами: символи тексту, зображення, зразки звуків тощо.

Під час навчання системи пропонуються різні зразки образів із зазначенням, якого класу вони ставляться. Зразок, як правило, представляється як вектор значень ознак. У цьому сукупність всіх ознак має однозначно визначати клас, якого належить зразок. Якщо ознак недостатньо, система може співвіднести той самий зразок з кількома класами, що неправильно.

Після закінчення навчання системи їй можна пред'являти невідомі раніше образи та отримувати відповідь про належність до певного класу. Топологія такої системи характеризується тим, що кількість нейронів у вихідному шарі, як правило, дорівнює кількості визначених класів. При цьому встановлюється відповідність між виходом нейронної системи та класом, який він представляє. Коли системи пред'являється якийсь образ, одному з її виходів має з'явитися ознака те, що образ належить цього класу. У той самий час інших виходах може бути ознака те, що образ даному класу не належить.

Якщо на двох або більше виходах є ознака приналежності до класу, вважається, що система не впевнена у своїй відповіді.

Використовувані архітектури навчання:

- навчання з учителем;
- перцептрон;
- згорткові нейронні системи;
- навчання без вчителя;
- системи адаптивного резонансу;
- змішане навчання;
- система радіально-базових функцій.

Аналіз часових рядів.

Часовий ряд (динамічний ряд, ряд динаміки) − зібраний у різні моменти часу статистичний матеріал про значення будь-яких параметрів (у найпростішому випадку одного) досліджуваного процесу. Кожна одиниця статистичного матеріалу називається виміром чи відліком, також допустимо називати його рівнем на зазначений із ним час. У часовому ряді для кожного відліку має бути вказано час виміру або номер виміру по порядку. Тимчасовий ряд істотно відрізняється від простої вибірки даних, тому що при аналізі враховується взаємозв'язок вимірів з часом, а не лише статистичне розмаїття та статистичні характеристики вибірки

Аналіз часових рядів − сукупність математико-статистичних методів аналізу, призначених для виявлення структури часових рядів та їх прогнозування. Сюди належать, зокрема, методи регресійного аналізу. Виявлення структури часового ряду необхідне у тому, щоб побудувати математичну модель того явища, що є джерелом аналізованого часового ряду. Прогноз майбутніх значень часового ряду використовується для ефективного ухвалення рішень.

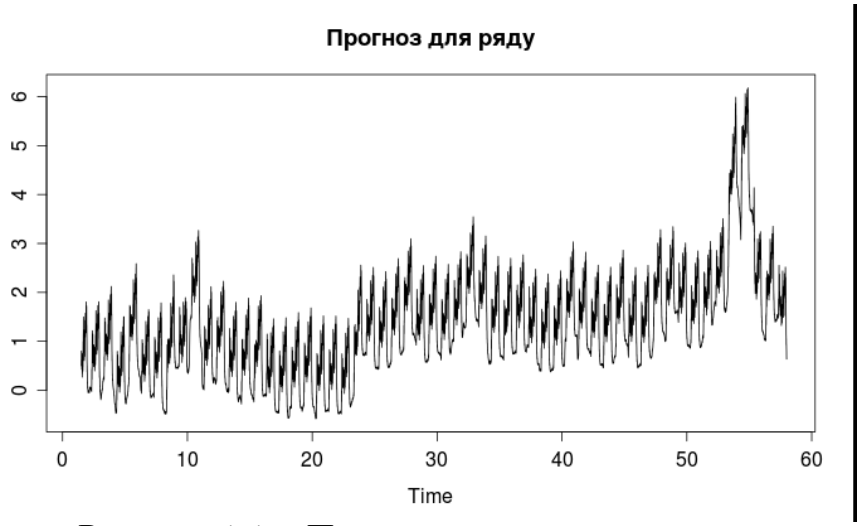


Рисунок 1.1 − Приклад прогнозу часового ряду

Приклад часового ряду.

Тимчасові ряди складаються з двох елементів:

– періоду часу, протягом якого чи станом який наводяться числові значення;

– числових значень того чи іншого показника, які називаються рівнями ряду.

Тимчасові ряди класифікуються за такими ознаками:

– за формою подання рівнів:

– ряди абсолютних показників;

– відносних показників;

– середніх величин;

– за кількістю показників, для яких визначаються рівні у кожний момент часу: одновимірні та багатовимірні часові ряди;

– за характером часового параметра: моментні та інтервальні часові ряди. У моментних часових рядах рівні характеризують значення показника за станом певні моменти часу. У інтервальних рядах рівні характеризують значення показника певні періоди часу. Важлива особливість інтервальних часових рядів абсолютних величин полягає у можливості підсумовування їх рівнів. Окремі рівні моментного ряду абсолютних величин містять елементи повторного рахунку. Це

робить безглуздим підсумовування рівнів моментних рядів;

– по відстані між датами та інтервалами часу виділяють рівновіддалені - коли дати реєстрації або закінчення періодів йдуть один за одним з рівними інтервалами і неповні (нерівновіддалені) - коли принцип рівних інтервалів не дотримується;

– по наявності пропущених значень: повні та неповні часові ряди;

– тимчасові ряди бувають детермінованими та випадковими: перші отримують на основі значень деякої невипадкової функції (ряд послідовних даних про кількість днів у місяцях); другі є результатом реалізації деякої випадкової величини.

В залежності від наявності основної тенденції виділяють стаціонарні ряди, в яких середнє значення та дисперсія постійні, та нестаціонарні, що містять основну тенденцію розвитку.

1.3 Прогнозування

Здібності системи до прогнозування безпосередньо випливають з її здатності до узагальнення та виділення прихованих залежностей між вхідними та вихідними даними. Після навчання система здатна передбачити майбутнє значення певної послідовності на основі кількох попередніх значень та (або) якихось існуючих на даний момент факторів.

Прогнозування можливе лише тоді, коли попередні зміни справді певною мірою визначають майбутні. Наприклад, прогнозування котирувань акцій на основі котирувань за минулий тиждень може виявитися успішним (а може й не виявитися), тоді як прогнозування результатів завтрашньої лотереї на основі даних за останні 50 років майже напевно не дасть жодних результатів.

Вибір даних для навчання системи та їх обробка є найскладнішим етапом розв'язання задачі. Набір даних для навчання має задовольняти кільком критеріям: Репрезентативність − дані повинні ілюструвати справжній стан речей у

предметній галузі; Несуперечність - суперечливі дані в навчальній вибірці призведуть до поганої якості навчання системи. Вихідні дані перетворюються на вид, у якому їх можна подати на входи системи. Кожен запис у файлі даних називається навчальною парою або навчальним вектором. Навчальний вектор містить по одному значенню на кожен вхід системи та, залежно від типу навчання (з учителем або без), за одним значенням для кожного виходу системи.

Навчання системи на «сирому» наборі зазвичай не дає якісних результатів. Існує низка способів покращити «сприйняття» системи. Нормування виконується, коли різні входи подаються дані різної розмірності. Наприклад, перший вхід системи подаються величини зі значеннями від нуля до одиниці, але в другий − від ста до тисячі. За відсутності нормування значення на другому вході завжди будуть істотно впливати на вихід системи, ніж значення на першому вході.

При нормуванні розмірності всіх вхідних та вихідних даних зводяться докупи; Квантування виконується над безперервними величинами, котрим виділяється кінцевий набір дискретних значень. Наприклад, квантування використовують для завдання частот звукових сигналів при розпізнаванні мови; Фільтрування виконується для «зашумлених» даних. З іншого боку, велику роль грає саме уявлення як вхідних, і вихідних даних. Припустимо, система навчається розпізнаванню літер на зображеннях і має один числовий вихід номер літери в алфавіті. У цьому випадку система отримає хибне уявлення про те, що літери з номерами 1 та 2 більш схожі, ніж літери з номерами 1 та 3, що загалом неправильно. Для того, щоб уникнути такої ситуації, використовують топологію системи з великою кількістю виходів, коли кожен вихід має власний сенс. Чим більше виходів у системи, тим більша відстань між класами і тим складніше їх сплутати.

У процесі навчання система певному порядку переглядає навчальну вибірку. Порядок перегляду може бути послідовним, випадковим тощо. Деякі системи, які навчаються без вчителя (наприклад, системи Хопфілда), переглядають вибірку лише один раз. Інші (наприклад, системи Кохонена), а

також системи, що навчаються з учителем, переглядають вибірку безліч разів, при цьому один повний прохід вибірки називається епохою навчання.

При навчанні з учителем набір вихідних даних ділять на дві частини - власне навчальну вибірку та тестові дані; принцип поділу може бути довільним. Навчальні дані подаються системи для навчання, а перевірочні використовуються для розрахунку помилки системи (перевірочні дані ніколи для навчання системи не застосовуються).

Рисунок 1.2 − Процес навчання

Таким чином, якщо на перевірочних даних помилка зменшується, система дійсно виконує узагальнення. Якщо помилка на навчальних даних продовжує зменшуватися, а помилка на тестових даних збільшується, то система перестала виконувати узагальнення і просто «запам'ятовує» навчальні дані.

Це називається перенавчанням системи або оверфіттингом. У разі навчання зазвичай припиняють. У процесі навчання можуть виявитися інші проблеми, такі як параліч або влучення системи в локальний мінімум поверхні помилок. Неможливо заздалегідь передбачити прояв тієї чи іншої проблеми, так само як і дати однозначні рекомендації щодо їх вирішення. Все вище сказане стосується лише ітераційних алгоритмів пошуку нейросистемевих рішень. Для них дійсно не

можна нічого гарантувати і не можна повністю автоматизувати навчання нейронних систем.

Навчання з учителем (англ. Supervised learning) − SL один із способів машинного навчання, в ході якого випробувана система примусово навчається за допомогою прикладів "стимул-реакція". З погляду кібернетики, одна із видів кібернетичного експерименту. Між входами та еталонними виходами (стимул-реакція) може існувати певна залежність, але вона невідома. Відома лише кінцева сукупність прецедентів − пара «стимул-реакція», яка називається навчальною вибіркою. На основі цих даних потрібно відновити залежність (побудувати модель відносин стимул-реакція, придатних для прогнозування), тобто побудувати алгоритм, здатний для будь-якого об'єкта видати досить точну відповідь. Для вимірювання точності відповідей, як і у навчанні на прикладах, може вводитися функціонал якості.

Машинне навчання без вчителя. Алгоритми навчання без вчителя навчаються на нерозмічених даних. Такі алгоритми переглядають нові дані, намагаючись встановити значущі зв'язки між вхідними та наперед визначеними вихідними даними. Вони можуть виявляти закономірності та класифікувати дані. Наприклад, алгоритми без вчителя можуть групувати статті новин з різних новинних веб-сайтів у загальні категорії, такі як спорт, кримінал і т. д. Вони можуть використовувати обробку природної мови для розуміння сенсу та емоцій у статті. У роздрібній торгівлі навчання без вчителя допоможе знайти закономірності у покупках клієнтів та надати результати аналізу даних, такі як: покупець, швидше за все, купить хліб, якщо також купить олію. Навчання без вчителя корисне для розпізнавання образів, виявлення аномалій та автоматичного групування даних за категоріями. Оскільки навчальні дані не вимагають маркування, налаштування просте. Ці алгоритми також можна використовувати для автоматичного очищення та обробки даних для подальшого моделювання. Обмеження цього у тому, що не може дати точних прогнозів. З іншого боку, він може самостійно виділяти конкретні типи вихідних даних.

Машинне навчання із частковим залученням вчителя.

Як випливає з назви, цей метод поєднує у собі навчання з учителем і без нього. Цей метод ґрунтується на використанні невеликої кількості розмічених даних та великої кількості нерозмічених даних для навчання систем. Спочатку розмічені дані використовуються для часткового навчання алгоритму машинного навчання. Після цього частково навчений алгоритм сам розмічає нерозмічені дані. Цей процес називається псевдомаркуванням. Потім модель перенавчається на результуючому наборі даних без програмування. Перевага цього методу в тому, що вам не потрібні великі обсяги даних. Це зручно при роботі з такими даними, як довгі документи, читання та маркування яких забирає надто багато часу в людини.

Навчання з підкріпленням.

Навчання з підкріпленням – це метод, у якому значення винагороди прив'язані до різних кроків, які мають пройти алгоритм. Таким чином, мета моделі – накопичити якнайбільше призових балів і зрештою досягти кінцевої мети. Більшість практичного застосування навчання з підкріпленням за останнє десятиліття була пов'язана з відеоіграми. Передові алгоритми навчання з підкріпленням досягли вражаючих результатів у класичних та сучасних іграх, часто значно перевершуючи ручні аналоги. Хоча цей метод найкраще працює у невизначених та складних середовищах даних, він рідко застосовується у бізнес-контексті. Це неефективно для чітко визначених завдань і упередженість розробників може вплинути на результати. Оскільки спеціаліст із роботи з даними розробляє нагороди, вони можуть впливати на результати.

Перевірка адекватності навчання.

Навіть у разі успішного, на перший погляд, навчання система не завжди навчається саме тому, чого від неї хотів автор. Відомий випадок, коли система навчалася розпізнаванню зображень танків за фотографіями, проте пізніше з'ясувалося, що всі танки були сфотографовані на тому самому тлі. Через війну система «навчилася» розпізнавати цей тип ландшафту, замість «навчитися»

розпізнавати танки. Таким чином, система «розуміє» не те, що від неї вимагалося, а те, що найпростіше узагальнити.

Тестування якості навчання нейросистеми необхідно проводити з прикладів, які брали участь у її навчанні. При цьому кількість тестових прикладів має бути тим більшою, чим вища якість навчання. Якщо помилки нейронної системи мають ймовірність близьку до однієї мільярдної, то й для підтвердження цієї ймовірності потрібен мільярд тестових прикладів. Виходить, що тестування добре навчених нейронних систем стає дуже важким завданням. Можливість перенавчання існує в тому, що критерій, який застосовується для тренування моделі, відрізняється від критерію, який застосовується для оцінки її ефективності. Зокрема, модель зазвичай тренують шляхом максимізації її продуктивності на якомусь наборі тренувальних даних. Проте її ефективність визначається не її продуктивністю на тренувальних даних, а її здатністю працювати добре на даних небачених. Перенавчання відбувається тоді, коли модель починає «запам'ятувати» тренувальні дані, замість того, щоб «вчитися» узагальненню з тенденції. Як крайній приклад, якщо число параметрів є таким самим, або більшим, як число спостережень, то проста модель або процес навчання може відмінно передбачати тренувальні дані, просто запам'ятовуючи їх повністю, але така модель зазвичай зазнаватиме рішучої невдачі при здійсненні передбачень про нові або небачені дані, оскільки ця проста модель взагалі не навчилася узагальнювати.

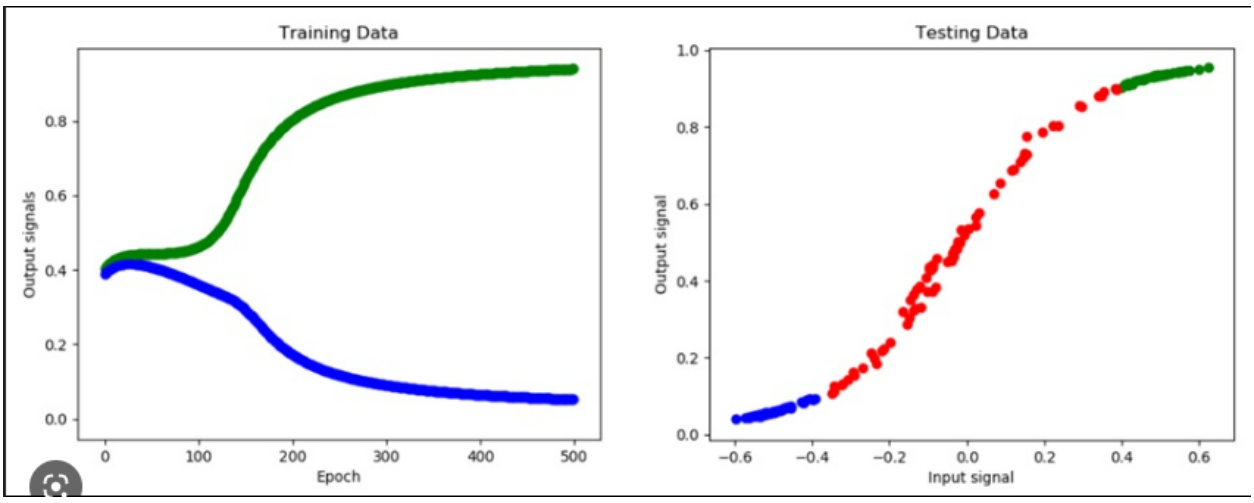


Рисунок 1.3 − Перевірка адекватності системи на прикладі повороту голови

Проблеми машинного навчання.

Головна проблема МН − перенавчання. Воно полягає в тому, що система «запам'ятовує» відповіді замість того, щоб уловлювати закономірності даних. Наука сприяла появі світ декількох методів боротьби з перенавчанням: сюди відносяться, наприклад, регуляризація, нормалізація батчів, нарощування даних та інші. Іноді перенавчена модель характеризується великими абсолютними значеннями ваги. Механізм цього явища приблизно такий: вихідні дані нерідко дуже багатовимірні (одна точка з навчальної вибірки зображується великим набором чисел), і ймовірність того, що навмання взята точка виявиться невідмінною від викиду, буде тим більше, чим більша розмірність. Замість того, щоб «вписувати» нову точку в існуючу модель, коригуючи ваги, нейросистема начебто вигадує сама собі виняток: цю точку ми класифікуємо за одними правилами, інші − за іншими. І таких точок зазвичай багато.

Очевидний спосіб боротьби з такого роду перенавчанням – регуляризація ваги. Вона полягає або у штучному обмеженні на значення ваг, або додаванні штрафу в міру помилки на етапі навчання. Такий підхід не вирішує проблему повністю, але найчастіше покращує результат.

Другий спосіб ґрунтується на обмеженні вихідного сигналу, а не значень ваг, − йдеться про нормалізацію батчів. На етапі навчання дані подаються нейросистеми пачками – батчами. Вихідні значення для них можуть бути якими завгодно, і тим їх абсолютні значення більші, ніж вище значення ваги. Якщо з кожного ми віднімемо якесь одне значення і поділимо результат на інше, однаково для всього батча, то ми збережемо якісні співвідношення (максимальне, наприклад, все одно залишиться максимальним), але вихід буде більш зручним для обробки його наступним шаром. Третій підхід працює не завжди. Як уже говорилося, перенавчена нейросистема сприймає багато точок як аномальні, які хочеться обробляти окремо. Ідея полягає в нарощуванні навчальної вибірки, щоб точки були начебто тієї ж природи, що й вихідна вибірка, але згенеровані штучно. Однак тут одразу народжується велика кількість супутніх проблем: добір

параметрів для нарощування вибірки, критичне збільшення часу навчання та інші. В окрему проблему виділяється пошук реальних аномалій в навчальній вибірці. Іноді це навіть розглядають як окреме завдання. Зображення вище демонструє ефект виключення аномального значення набору. У разі нейронних систем ситуація буде аналогічною.

Робота в середовищах, що динамічне змінюються (наприклад, у фінансових), складна для нейронних систем. Навіть якщо вам вдалося успішно натренувати систему, немає гарантій, що вона не перестане працювати у майбутньому. Фінансові ринки постійно трансформуються, тому те, що працювало вчора, може з тим самим успіхом «зламатися» сьогодні.

Тут дослідникам або доводиться тестувати різноманітні архітектури систем та вибирати з них найкращу, або використовувати динамічні нейронні системи. Останні «стежать» за змінами середовища та підлаштовують свою архітектуру відповідно до них.

Висновки до розділу 1

Машинне навчання є важливим кроком людвства до автоматизації процесів навчання. Вже сьогодні ми використовуємо в напрямах розпізнавання, прогнозування, покращення навчання та автоматизованого виробництва. Так машинне навчання чеж має свох недоліки, швидкість та точність виконання іноді потребує багато часу та ресурсів, але очевидно що внесок катого автоматизованого навчання буде окупати потрачені на нього ресурси та час.

2 МЕТОДИ ТА МОДЕЛІ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ПРОГНОЗУВАННЯ

2.1 Засоби та інструменти розробки: RStudio та мова R

RStudio − це добре інтегроване середовище розробки, визнане R. RStudio складається з панелей написів, серед іншого, ви можете перейти до панелі «Джерело», редактора сценаріїв (сценарій), щоб зберегти послідовності, панелі «Консоль».

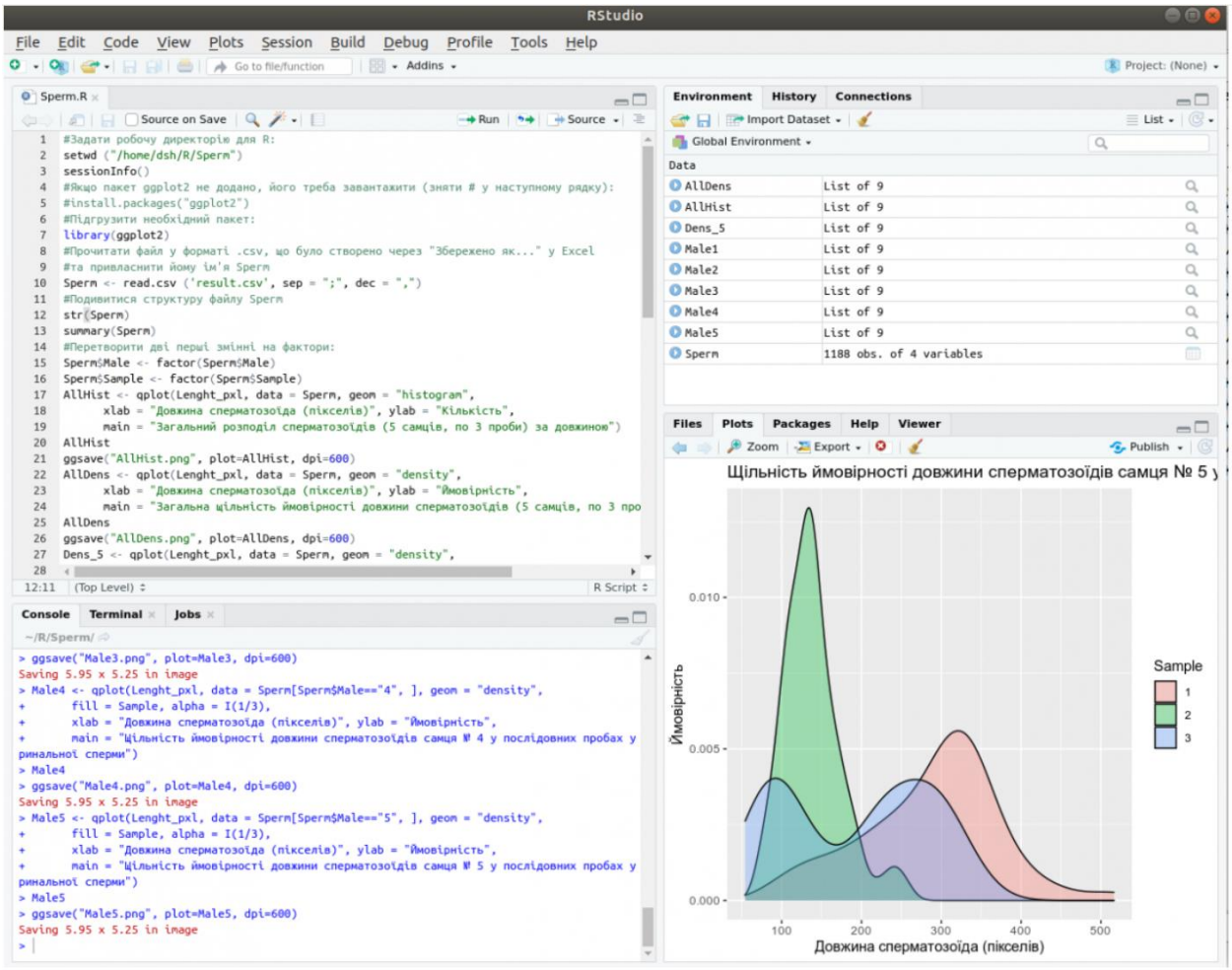


Рисунок 2.1 − Основні елементи консолі RStudio

Методи програмування в R.

Основою роботи в середовищі R є розрахунок виразів, які відокремлюють точкою з комою (;). Вирази можуть містити назви змінних або констант, оператори, виклики функцій та керуючі символи. Для назв змінних можна застосовувати літери латинського алфавіту, цифри, точку та знак підкреслювання, причому першою має бути обов'язково літера, наприклад, profit.1_bank.

Для подальшого використання результати розрахунку виразів можна присвоювати змінним за допомогою оператора <-, крім того можна використовувати =, а також функцію assign (name, value). Останній варіант зручно використовувати, коли ім'я змінної формується під час виконання програми.

Для того, щоб побачити значення деякої змінної достатньо виконати вираз, який складається лише з її назви, що аналогічно виконанню виразу print (name). Мова R є чутливою до регістру літер, тобто, змінні з назвами Profit.1Bank та profit.1bank будуть сприйматись програмою як різні змінні. Ряд назв є зарезервованими, наприклад, if є керуючою командою умовного виконання виразу. Крім того, ряд назв використовуються для базових функцій, проте присвоювання значень змінним з такою назвою, зазвичай, не приводить до помилки.

Під час роботи з логічним виразом можуть використовуватись значення TRUE та FALSE, для позначення пропущених значень використовуються NA, нескінченності − Inf, а для невизначених − NaN.

Мова програмування.

R – мова програмування для статистичної обробки даних та роботи з графікою, а також вільне програмне середовище обчислень з відкритим вихідним кодом у рамках проекту GNU. Мова створювалася як аналогічна мови S, розробленій в Bell Labs, і є її альтернативною реалізацією, хоча між мовами є суттєві відмінності, але в більшості своїй код мовою S працює в середовищі R. Спочатку R був розроблений співробітниками статистичного факультету

Оклендського університету Россом Айхекою (англ.: Ross Ihaka) і Робертом Джентлменом (англ. Robert Gentleman) (перша буква їх імен − R); мова та середовище підтримуються та розвиваються організацією R Foundation.

Широко використовується як статистичне програмне забезпечення для аналізу даних та фактично став стандартом для статистичних програм

R − мова програмування, що інтерпретується, основним способом роботи з яким є командний інтерпретатор. Мова є регістрозалежною, у плані синтаксису вона схожа, з одного боку, на функціональні мови типу Scheme, з іншого − на типові сучасні сценарні мови, з простим синтаксисом і невеликим набором основних конструкцій. Мова об'єктна: будь-який програмний об'єкт має набір атрибутів − іменований список значень, що визначають його.

Мова підтримує мінімальний набір примітивних типів даних: символьний (character), числовий (numeric), логічний (logical) та комплексний (complex). Числові змінні, крім звичайних чисел, можуть набувати спеціальних значень NaN (Not a Number − «не число») та Inf (Infinity − «нескінченність»). Нескінченність (позитивна або негативна) виходить при виході результату обчислень за межі діапазону, що надається реалізацією, NaN − при операціях з невизначеним результатом. Крім цих, є ще одне дуже важливе спеціальне значення, NA (Not Available - "не доступно"). Воно може бути використане для фіксації того факту, що відповідне значення, що бере участь у обчисленнях, з якоїсь причини не було отримано (досить звичайна у статистичних розрахунках ситуація, коли через збої в зборі даних деякі спостереження залишаються без результатів).

Значення примітивних типів можуть поєднуватися у вектори (vector), списки (list), матриці або масиви (matrix), у тому числі багатовимірні; ці комбіновані типи зберігають набори даних одного й того самого примітивного типу. Крім цього, мова містить поняття факторів (factor) − наборів категоріальних або шкальних даних, що приймають строго певний набір значень. Нарешті, можуть створюватися таблиці (data frame) − структури даних, які кожної рядки (індивіда) зберігають набір різних (і мають різні типи) параметрів (ознак).

Особливістю R є те, що операції з векторами та матрицями підтримуються на рівні самої мови, як, наприклад, APL.

Існує операція вилучення та запису даних (аналог присвоєння) «<-», а також звичайні операції роботи з даними, у тому числі арифметичні. Доступ за індексом до елементів векторів та масивів здійснюється за допомогою квадратних дужок, доступ до атрибутів списків – за допомогою оператора «$». Є мінімальний набір стандартних конструкцій імперативного програмування: умовний оператор if, цикли while і for. Вирази на R можна описувати як окремі об'єкти та обчислювати при необхідності. На цьому ж механізмі ґрунтується опис функцій. Є вбудовані в мову засоби застосування виразів та функцій до векторів та масивів.

Функції R можуть об'єднуватися в пакети − модулі, що завантажуються, які підключаються до будь-якої програми і надають об'єднані в них обчислювальні засоби. Пакети для R можуть розроблятися іншими мовами програмування, у тому числі Сі, що дозволяє, з одного боку, компенсувати обмеженість образотворчих засобів самої мови R, а з іншого − при необхідності досягти високих показників обчислювальної продуктивності.

Сама мова має досить обмежені та не надто зручні засоби опису даних, але це компенсується наявністю бібліотечних засобів, які дозволяють завантажувати у вигляді таблиць R набори даних, представлених у більшості відкритих та багатьох пропрієтарних форматах. Так, R можуть бути легко завантажені таблиці в простому текстовому форматі, таблиці Excel різних версій, дані у форматах CSV, XML і багатьох інших.

В цілому, як мова програмування, R досить проста і навіть примітивна. Його найсильніша сторона – можливість необмеженого розширення за допомогою пакетів. У базове постачання R включено основний набір пакетів, а за станом на 2019 рік доступно більше 15 316 пакетів. У R реалізовані практично всі актуальні засоби універсальних статистичних обчислень, такі як регресійний аналіз та аналіз часових рядів, а також безліч специфічних алгоритмів для вирішення вузькоспеціалізованих завдань та досліджень в окремих галузях.

Ще одна особливість мови – можливість створення якісної графіки друкарського рівня, яка може бути експортована у поширені графічні формати та використана для презентацій чи публікацій. Є готові пакети, що зв'язують R з GUI-фреймворками (наприклад, заснованими на Tcl/Tk) і дозволяють створювати спеціалізовані утиліти статистичного аналізу з графічним інтерфейсом користувача та відображенням результатів у вигляді графіків та діаграм.

2.2 Аналіз та попередня обробка даних

Матриця кореляції.

Кореляційна матриця − це просто таблиця, яка відображає коефіцієнти кореляції для різних змінних. Матриця зображує кореляцію між усіма можливими парами значень у таблиці. Це потужний інструмент для узагальнення великого набору даних, а також для виявлення та візуалізації шаблонів у наданих даних.

Кореляційна матриця складається з рядків і стовпців, які показують змінні. Кожна комірка таблиці містить коефіцієнт кореляції.

Крім того, кореляційна матриця часто використовується в поєднанні з іншими типами статистичного аналізу. Наприклад, це може бути корисним для аналізу лінійної регресії. Пам'ятайте, що моделі містять кілька незалежних змінних. У множинній лінійній регресії кореляційна матриця визначає коефіцієнти кореляції між незалежними змінними в моделі.

Існують три основні причини для обчислення матриці кореляції:

— підсумовувати велику кількість даних, де мета полягає в тому, щоб побачити закономірності. У прикладі вище спостерігається закономірність у тому, що це змінні сильно корелюють друг з одним;

— для введення до інших аналізів. Наприклад, люди зазвичай використовують матриці кореляції як вхідні дані для розвідувального факторного аналізу, факторного аналізу підтверджень, моделей структурних рівнянь і лінійної регресії при виключенні пропущених значень по парах;

— як діагностика під час перевірки інших аналізів. Наприклад, при використанні лінійної регресії велика кількість кореляцій дозволяє припустити, що оцінки лінійної регресії будуть ненадійними.

$$\begin{vmatrix} 1 & R_{12} & \dots & R_{1j} & \dots & R_{1m} \\ R_{21} & 1 & \dots & R_{2j} & \dots & R_{2m} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ R_{i1} & R_{i2} & \dots & 1 & \dots & R_{im} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ R_{m1} & R_{m2} & \dots & R_{mj} & \dots & 1 \end{vmatrix}$$

Рисунок 2.2 – Приклад кореляційної матриці

Кореляційна статистика.

У більшості матриць кореляції використовується кореляція «продукт-момент» Пірсона (r). Також часто використовуються кореляція Спірмена та Тау-Б Кендала. Обидві ці матриці є непараметричними кореляціями і менш сприйнятливими до відхилень, ніж r.

Кодування змінних.

Якщо у вас є дані опитування, вам необхідно вирішити, як кодувати дані, перш ніж обчислювати кореляції. Наприклад, якщо респондентам були запропоновані варіанти з сильно вираженою незгодою, з деякою незгодою, ні з чим не згоден, з деякою згодою та з сильно вираженою згодою, можна було б присвоїти коди 1, 2, 3, 4 і 5 відповідно (або математично еквівалентні з погляду кореляції, значення -2, -1, 0, 1 та 2). Однак можливі інші кодування, такі як -4, -1, 0, 1, 4. Зміни в кодуваннях, як правило, малоефективні, за винятком крайніх випадків.

Лікування відсутніх значень.

Дані, які ми використовуємо для обчислення кореляцій, часто містять значення, що відсутні. Це може бути тому, що ми не збирали ці дані, або тому, що не знаємо відповідей. Існують різні стратегії до роботи з пропущеними значеннями при обчисленні матриць кореляції. Зазвичай найкращою практикою є використання множинних настанов. Однак люди частіше використовують парні пропущені значення (іноді відомі як часткові кореляції). Це включає обчислення кореляції з використанням всіх пропущених даних для двох змінних. В якості альтернативи деякі використовують видалення за списком, також відоме як видалення регістру, яке використовує тільки спостереження без пропущених даних. Як парне, і регістрове видалення передбачає, що дані повністю відсутні випадковим чином. Ось чому кращим варіантом, як правило, є багаторазове поставлення.

Лінійна регресія.

Лінійна регресія − це алгоритм, який забезпечує лінійний зв’язок між незалежною змінною та залежною змінною для прогнозування результатів майбутніх подій. Це статистичний метод, який використовується в науці про дані та машинному навчанні для прогнозного аналізу.

Незалежна змінна також є прогностичною або пояснювальною змінною, яка залишається незмінною через зміну інших змінних. Однак залежна змінна змінюється разом із коливаннями незалежної змінної. Регресійна модель передбачає значення залежної змінної, яка є змінною відповіді або результату, що аналізується або вивчається.

Таким чином, лінійна регресія − це контрольований алгоритм навчання, який моделює математичний зв’язок між змінними та робить прогнози для неперервних або числових змінних, таких як продажі, зарплата, вік, ціна продукту тощо.

Цей метод аналізу є перевагою, коли в даних доступні принаймні дві змінні, як це спостерігається при прогнозуванні фондового ринку, управлінні портфелем, науковому аналізі тощо.

Ключові переваги лінійної регресії

Лінійна регресія є популярним статистичним інструментом, який використовується в науці про дані, завдяки кільком перевагам.

1. Легка реалізація.

Модель лінійної регресії обчислювально проста для реалізації, оскільки вона не вимагає великих інженерних витрат ні перед запуском моделі, ні під час її обслуговування.

2. Інтерпретованість.

На відміну від інших моделей глибокого навчання (нейронні системи), лінійна регресія є відносно простою. Як наслідок, цей алгоритм випереджає моделі чорної скриньки, які не можуть обґрунтувати, яка вхідна змінна викликає зміну вихідної змінної.

3. Масштабованість

Лінійна регресія не є важкою для обчислень і, отже, добре підходить у випадках, коли масштабування є важливим. Наприклад, модель може добре масштабуватися щодо збільшення обсягу даних (великі дані).

4. Оптимальний для онлайн налаштувань

Простота обчислення цих алгоритмів дозволяє використовувати їх в онлайн-налаштуваннях. Модель можна навчати та перенавчати з кожним новим прикладом, щоб генерувати прогнози в режимі реального часу, на відміну від нейронних систем або опорних векторних машин, які є важкими для обчислень і вимагають багато обчислювальних ресурсів і значного часу очікування для повторного навчання на новому наборі даних. Усі ці фактори роблять такі інтенсивні обчислювальні моделі дорогими та непридатними для програм реального часу.

Наведені вище функції підкреслюють, чому лінійна регресія є популярною моделлю для вирішення реальних проблем машинного навчання.

Регресійний аналіз є дуже широко використовується статистичним інструментом для встановлення моделі відносин між двома змінними. Одна з цих

змінних називається предикторною змінною, значення якої збирається в ході експериментів. Інша змінна називається змінною відповіді, значення якої отримано із змінної предиктора.

У лінійній регресії ці дві змінні пов'язані через рівняння, де показник (ступінь) обох змінних дорівнює 1. Математично лінійна залежність представляє пряму лінію, коли вона зображена у вигляді графіка.

Загальне математичне рівняння для лінійної регресії:

$$y = ax + b; \quad (2.1)$$

y – змінна відповіді; x – це передикторна змінна; a та b є константами, які називаються коефіцієнтами.

Кроки щодо створення регресії.

Простим прикладом регресії є прогнозування ваги людини, коли відоме її зростання. Для цього нам необхідно мати співвідношення між зростанням та вагою людини.

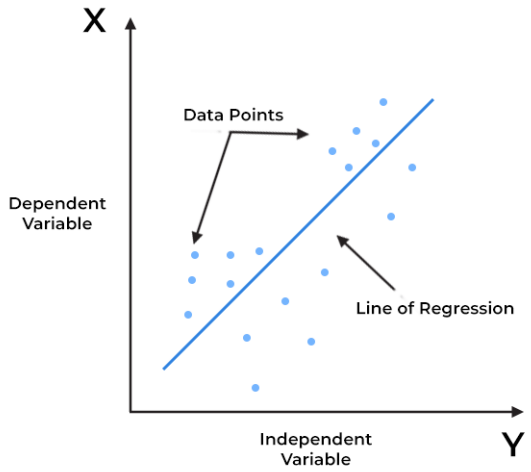
Проводять експеримент зі збору зразка значень зростання, що спостерігаються, і відповідної ваги.

Створіть модель відносин, використовуючи функції lm() у R.

Знайдіть коефіцієнти зі створеної моделі та створіть математичне рівняння, використовуючи ці

Отримайте інформацію про модель відносин, щоб дізнатися середню помилку в прогнозуванні. Також називається залишками.

Щоб передбачити вагу нових людей, використовуйте функцію предикату () у R.



Best Fit Line for a Linear Regression Model

Рисунок 2.3 – Приклад звичайної лінійної регресії

Рівняння множинної лінійної регресії.

Вищеописаний процес застосовується до простої лінійної регресії з однією ознакою або незалежною змінною. Однак регресійну модель можна використовувати для кількох функцій, розширивши рівняння для кількості змінних, доступних у наборі даних.

Рівняння для множинної лінійної регресії подібне до рівняння для простого лінійного рівняння, тобто y(x) = p0 + p1x1 плюс додаткові ваги та вхідні дані для різних ознак, які представлені p(n)x(n). Формула множинної лінійної регресії виглядатиме так:

$$y(x) = p0 + p1x1 + p2x2 + \ldots + p(n)x(n) \quad (2.2)$$

Модель машинного навчання використовує наведену вище формулу та різні значення ваги, щоб малювати лінії відповідно до розміру. Крім того, щоб визначити лінію, яка найкраще відповідає даним, модель оцінює різні вагові комбінації, які найкраще відповідають даним, і встановлює міцний зв’язок між змінними.

Крім того, поряд із функцією прогнозування регресійна модель використовує функцію вартості для оптимізації ваг (pi). Функція вартості лінійної регресії є середньоквадратичною помилкою або середньоквадратичною помилкою (MSE).

По суті, MSE вимірює середню квадратичну різницю між фактичними та прогнозованими значеннями спостереження. Результатом є вартість або оцінка, пов’язана з поточним набором ваг і, як правило, одне число. Метою тут є мінімізація MSE для підвищення точності регресійної моделі.

Типи моделей лінійної регресії включають.

1. Проста лінійна регресія

Проста лінійна регресія виявляє кореляцію між залежною змінною (вхід) і незалежною змінною (вихід). У першу чергу цей тип регресії описує наступне:

Сила зв'язку між заданими змінними.

Приклад: взаємозв’язок між рівнями забруднення та підвищенням температури.

2. Множинна лінійна регресія

Множинна лінійна регресія встановлює зв'язок між незалежними змінними (двома або більше) і відповідною залежною змінною. Тут незалежні змінні можуть бути неперервними або категоріальними. Цей тип регресії допомагає передбачити тенденції, визначити майбутні значення та передбачити наслідки змін.

Приклад: Розглянемо завдання на обчислення артеріального тиску. У цьому випадку зріст, вага та кількість фізичних вправ можна вважати незалежними змінними. Тут ми можемо використовувати множинну лінійну регресію для аналізу зв’язку між трьома незалежними змінними та однією залежною змінною, оскільки всі розглянуті змінні є кількісними.

3. Логістична регресія

Логістична регресія − також відома як логіт-модель − застосовна у випадках, коли є одна залежна змінна та кілька незалежних змінних.

Фундаментальна відмінність між множинною та логістичною регресією полягає в тому, що цільова змінна в логістичному підході є дискретною (двійкове або порядкове значення). Маючи на увазі, що залежна змінна є кінцевою або категоричною – або P, або Q (бінарна регресія), або діапазон обмежених варіантів P, Q, R або S.

Значення змінної обмежено лише двома можливими результатами лінійної регресії. Однак логістична регресія вирішує цю проблему, оскільки вона може повернути оцінку ймовірності, яка показує ймовірність будь-якої конкретної події.

Приклад: можна визначити ймовірність вибору пропозиції на вашому веб-сайті (залежна змінна). Для цілей аналізу ви можете переглянути різні характеристики відвідувачів, наприклад сайти, з яких вони прийшли, кількість відвідувань вашого сайту та активність на вашому сайті (незалежні змінні). Це може допомогти визначити ймовірність певних відвідувачів, які з більшою ймовірністю приймуть пропозицію. Як наслідок, це дає вам змогу приймати кращі рішення щодо того, просувати пропозицію на своєму сайті чи ні.

Крім того, логістична регресія широко використовується в алгоритмах машинного навчання в таких випадках, як виявлення спаму в електронних листах, прогнозування суми кредиту для клієнта тощо.

4. Порядкова регресія

Порядкова регресія включає одну залежну дихотомічну змінну та одну незалежну змінну, яка може бути порядковою або номінальною. Це полегшує взаємодію між залежними змінними з кількома впорядкованими рівнями з однією або кількома незалежними змінними.

Для залежної змінної з m категоріями буде створено рівняння (m -1). Кожне рівняння має різні точки перетину, але однакові коефіцієнти нахилу для змінних предиктора. Таким чином, порядкова регресія створює кілька рівнянь передбачення для різних категорій. У машинному навчанні порядкова регресія

стосується навчання ранжирування або ранжирувального аналізу, обчисленого за допомогою узагальненої лінійної моделі (GLM).

Приклад: розглянемо опитування, у якому респонденти мають відповісти «згоден» або «не згоден». У деяких випадках такі відповіді не допомагають, оскільки неможливо зробити остаточний висновок, що ускладнює узагальнені результати. Однак ви можете спостерігати природний порядок у категоріях, додаючи рівні до відповідей, наприклад, згоден, повністю згоден, не згоден і категорично не згоден. Таким чином, порядкова регресія допомагає передбачити залежну змінну, «впорядкувавши» кілька категорій за допомогою незалежних змінних.

5. Мультиноміальна логістична регресія

Мультиноміальна логістична регресія (MLR) виконується, коли залежна змінна номінальна з більш ніж двома рівнями. Він визначає зв’язок між однією залежною номінальною змінною та однією чи кількома незалежними змінними безперервного рівня (інтервал, відношення або дихотомічні). Тут номінальна змінна відноситься до змінної без внутрішнього порядку.

Приклад: Мультиноміальний логіт можна використовувати для моделювання вибору програм, зроблених учнями. Вибір програм, у цьому випадку, відноситься до професійно-технічної програми, спортивної програми та академічної програми. Вибір типу програми можна передбачити, враховуючи різноманітні атрибути, наприклад, наскільки добре учні вміють читати та писати з предметів, які викладаються, стать і нагороди, які вони отримали.

Тут залежною змінною є вибір програм із кількома рівнями (невпорядкованими). Для прогнозування в такому випадку використовується техніка мультиноміальної логістичної регресії.

Нелінійна регресія.

Нелінійна регресія – це форма регресійного аналізу, у якій дані підлаштовуються під модель, а потім виражаються як математична функція.

Проста лінійна регресія пов’язує дві змінні (X і Y) прямою лінією (y = mx + b), тоді як нелінійна регресія зв’язує дві змінні нелінійною (кривою) залежністю.

Мета моделі − зробити суму квадратів якомога меншою. Сума квадратів – це міра, яка відстежує, наскільки спостереження Y відрізняються від нелінійної (кривої) функції, яка використовується для прогнозування Y.

Він обчислюється шляхом спочатку знаходження різниці між підігнаною нелінійною функцією та кожною точкою Y даних у наборі. Потім кожна з цих різниць зводиться в квадрат. Нарешті, усі фігури в квадраті додаються разом. Чим менша сума цих квадратів, тим краще функція відповідає точкам даних у наборі. Нелінійна регресія використовує логарифмічні функції, тригонометричні функції, експоненціальні функції, степеневі функції, криві Лоренца, функції Гауса та інші методи підгонки.

Ключові висновки:

— нелінійна регресія відноситься до регресійного аналізу, де модель регресії відображає нелінійний зв’язок між залежними та незалежними змінними;

— вона більш точна і гнучка, ніж лінійна модель. Модель може вміщувати різноманітні криві, виводячи складні зв’язки між двома чи більше змінними;

— приклади цієї статистичної моделі включають зображення зв’язку між ціною на золото та інфляцією ІСЦ у США та настроями інвесторів і прибутковістю фондового ринку;

— його застосовують у різних дисциплінах, таких як машинне навчання, страхування, дослідження в лісовому господарстві тощо;

— як лінійна, так і нелінійна регресія прогнозують відповіді Y від змінної (або змінних) X;

— нелінійна регресія − це вигнута функція змінної X (або змінних), яка використовується для прогнозування змінної Y;

— нелінійна регресія може показати прогноз зростання з часом.

Нелінійне регресійне моделювання подібне до лінійного регресійного моделювання тим, що обидва прагнуть графічно відстежити певну реакцію набору змінних. Розробити нелінійні моделі складніше, ніж лінійні, оскільки функція створюється за допомогою серії наближень (ітерацій), які можуть виникати методом проб і помилок. Математики використовують кілька усталених методів, таких як метод Гаусса-Ньютона та метод Левенберга-Марквардта.

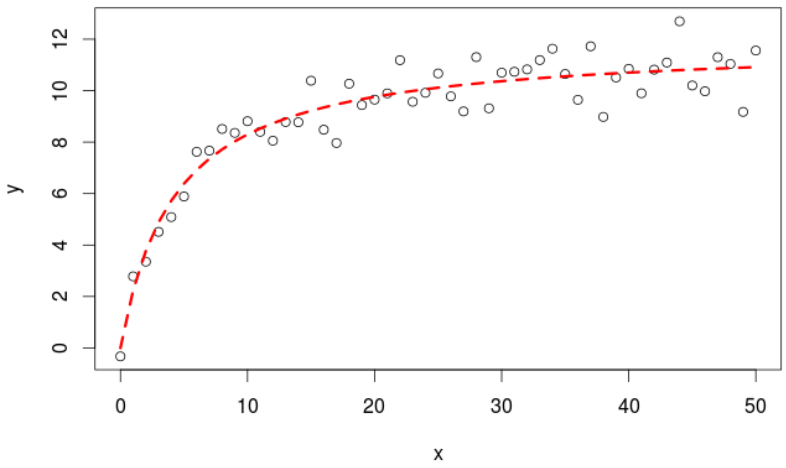


Рисунок 2.4 – Приклад нелінійної регресії

Часто регресійні моделі, які на перший погляд здаються нелінійними, насправді є лінійними. Процедуру оцінки кривої можна використовувати для визначення характеру функціональних зв’язків у ваших даних, щоб ви могли вибрати правильну регресійну модель, лінійну чи нелінійну. Моделі лінійної регресії, хоча вони зазвичай утворюють пряму лінію, можуть також формувати криві, залежно від форми рівняння лінійної регресії. Подібним чином можна використовувати алгебру для перетворення нелінійного рівняння таким чином, щоб воно імітувало лінійне рівняння − таке нелінійне рівняння називають «внутрішньолінійним».

Приклад нелінійної регресії.

Одним із прикладів використання нелінійної регресії є прогнозування зростання чисельності населення з часом. Діаграма розсіювання змін даних про

чисельність населення з часом показує, що, здається, існує зв'язок між часом і зростанням чисельності населення, але це нелінійний зв'язок, що вимагає використання нелінійної регресійної моделі. Логістична модель зростання населення може надати оцінки чисельності населення за періоди, які не вимірювалися, і прогнози майбутнього зростання населення.

Незалежні та залежні змінні, які використовуються в нелінійній регресії, повинні бути кількісними. Категориальні змінні, такі як регіон проживання чи віросповідання, слід кодувати як бінарні змінні або інші типи кількісних змінних.

Щоб отримати точні результати моделі нелінійної регресії, ви повинні переконатися, що вказана функція точно описує зв'язок між незалежними та залежними змінними. Хороші початкові значення також необхідні. Погані початкові значення можуть призвести до того, що модель не зможе збігатися, або рішення, яке буде оптимальним лише локально, а не глобально, навіть якщо ви вказали правильну функціональну форму для моделі.

У нелінійній регресії експериментальні дані зіставляються з моделлю, а математична функція, що представляє змінні (залежні та незалежні) у нелінійній криволінійній залежності, формується та оптимізується. Це прийнято як гнучку форму регресійного аналізу

Прикладами алгоритмів, які використовуються для розробки нелінійних моделей, є нелінійні найменші квадрати Левенберга-Марквардта та алгоритми Гаусса-Ньютона.

Проста нелінійна модель регресії виражається таким чином:

$$Y = f(X,\beta) + \epsilon \qquad (2.3)$$

Де:

X − вектор P предикторів

β – вектор з k параметрів

F (-) – відома функція регресії

є − член помилки

Крім того, модель також можна записати наступним чином:

$$Yi = h\,[xi(1), xi(2), \ldots, xi(m); \Theta 1, \Theta 2, \ldots, \Theta p] + Ei \qquad (2.4)$$

Де:

Yi − це відповідна змінна

h − функція

x - це вхід

Θ – параметр, що підлягає оцінці

Оскільки кожен параметр можна оцінити, щоб визначити, чи є він нелінійним чи лінійним, дана функція Yi може містити суміш нелінійних і лінійних параметрів. Розглядається функція h в моделі, оскільки її не можна записати як лінійну за параметрами. Натомість функція виводиться з теорії.

Термін «нелінійний» стосується параметрів у моделі, на відміну від незалежних змінних. Існують необмежені можливості для опису детермінованої частини моделі. Така гнучкість забезпечує хорошу основу для статистичних висновків.

Метою моделі є якомога менша мінімізація суми квадратів за допомогою ітераційних числових процедур. Найкращою оцінкою параметрів моделі є принцип найменших квадратів, який вимірює, скільки спостережень відхиляється від середнього значення набору даних. Також варто зазначити, що різниця між моделями лінійної та нелінійної регресії полягає в обчисленні методу найменших квадратів.

Висновок до розділу 2

Проєктування з використанням матриць кореляцій та лінійних, нелінійних регресій є одним ж найблільш точних підходів для створення інтелектуальниї систем які можуть прогнозувати дані на основі баз даних. А діаграми розсіювання вже використовують для прогнозів зростання численості населення з часом. Використання таких методів вже показує свою результативність у різних сферах діяльності.

В наступному розділі самі ці методи будут використані для створення інтелектуальної системи.

3 ПРОЄКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

3.1 Підключення пакетів та бібліотек

Першочергова задача після завантаження та підготовки є підключення декількох розширених ноборів інструментів (бібліотек) та пакету роботи з документацією, так як ми працюємо з базою данних

```
install.packages('cli', version='3.4.1')
install.packages(
  c(
    'purrr',
    'bruceR',
    'corrplot',
    'RColorBrewer',
    "PerformanceAnalytics",
    'tidyverse',
    'ggplot2'
  )
)
library(cli)
library(PerformanceAnalytics)
library(purrr)
library(bruceR)
library(corrplot)
library(RColorBrewer)
library(tidyverse)
library(ggplot2)
```

Після того як усі бібліотеки будут підключені, нам потрібно вказати шлях до нашої бази данних, яка містить такі данні як:

- age – вік;
- sex – стать;
- bmi – індекс маси тіла;
- children – наявність дітей;

– smoker – спиживання цигарок;
– region – регіон;
– expenses – витрати.

```
filepath<-"C:\\Users\\Valentin\\Desktop\\diploma\\insurance.csv"
data <- read.csv(filepath)
```

Ця команда відповідає за використання бази даних з назвою insurance.csv для нашого проетку .Так як у нашій базі данних є не тільки числа а й слова, наприклад вибірка статі або куріння, то нам потрібно перетворити слова в чисельні значення

```
data['sex_bin'] <- flatten_dbl(map(data$sex, function(x)
as.numeric(x=='female')))
data['smoker_bin'] <- flatten_dbl(map(data$smoker, function(x)
as.numeric(x=='yes')))
regions <-unique(data$region)
print(regions)
```

Ця команда перетворення бінарної хар-ки "стать" та “паління” до числового типу. А категоріальну ознаку "Регіон" перетворюють на сукупність бігарних ознак.

```
for (y in 1:length(regions)) {
  data[regions[y]]<-flatten_dbl(map(data$region,
function(x) as.numeric(x==regions[y])))
}
data <- data[ , !(names(data) %in% c('region','sex','smoker'))]
columns <-names(data)
for (y in 1:length(columns)) {
  data[columns[y]]<-scaler(data[columns[y]], min = 0, max = 10)
}
```

Так як у кожна категорія має велику широту значень ми скорочуємо її дня виділення кожного значення ознаки у окрему бінарну хар-ку числового типу.

Та нормалізуємо дані, до діапазону від 0 до 10, проводимо визначення усіх характеристик та їх нормалізацію. Де числовє значення матиме запит True\False/

3.2 Створення матриці кореляцій.

```
M <- cor(data)
corrplot(M, method="color")
chart.Correlation(data, histogram = TRUE, method = "pearson")
```

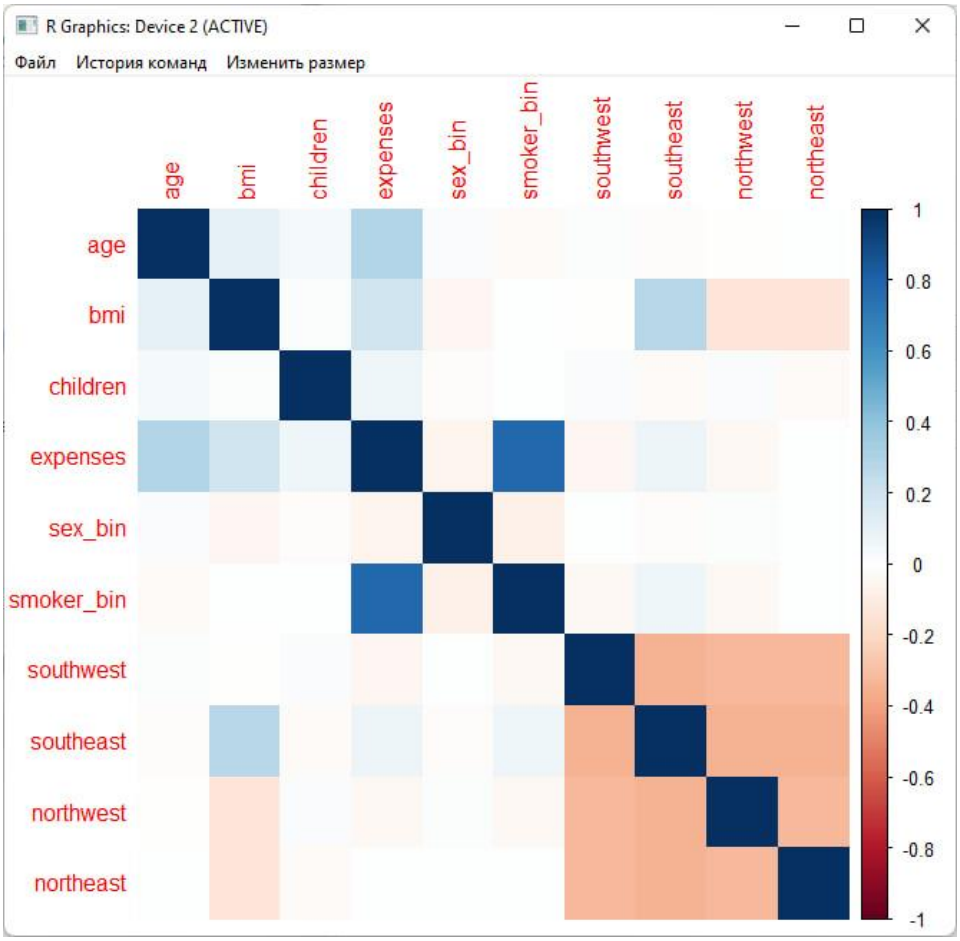


Рисунок 3.1 – Матриця кореляцій

Дана команда будує матрицю кореляцій та теплову карту за заданою матрицею. Створює до неї гістограми та аналітичні функцій розподілу за кожною характеристикою

Також створює діаграму розсіювання для кожної пари характеристик.

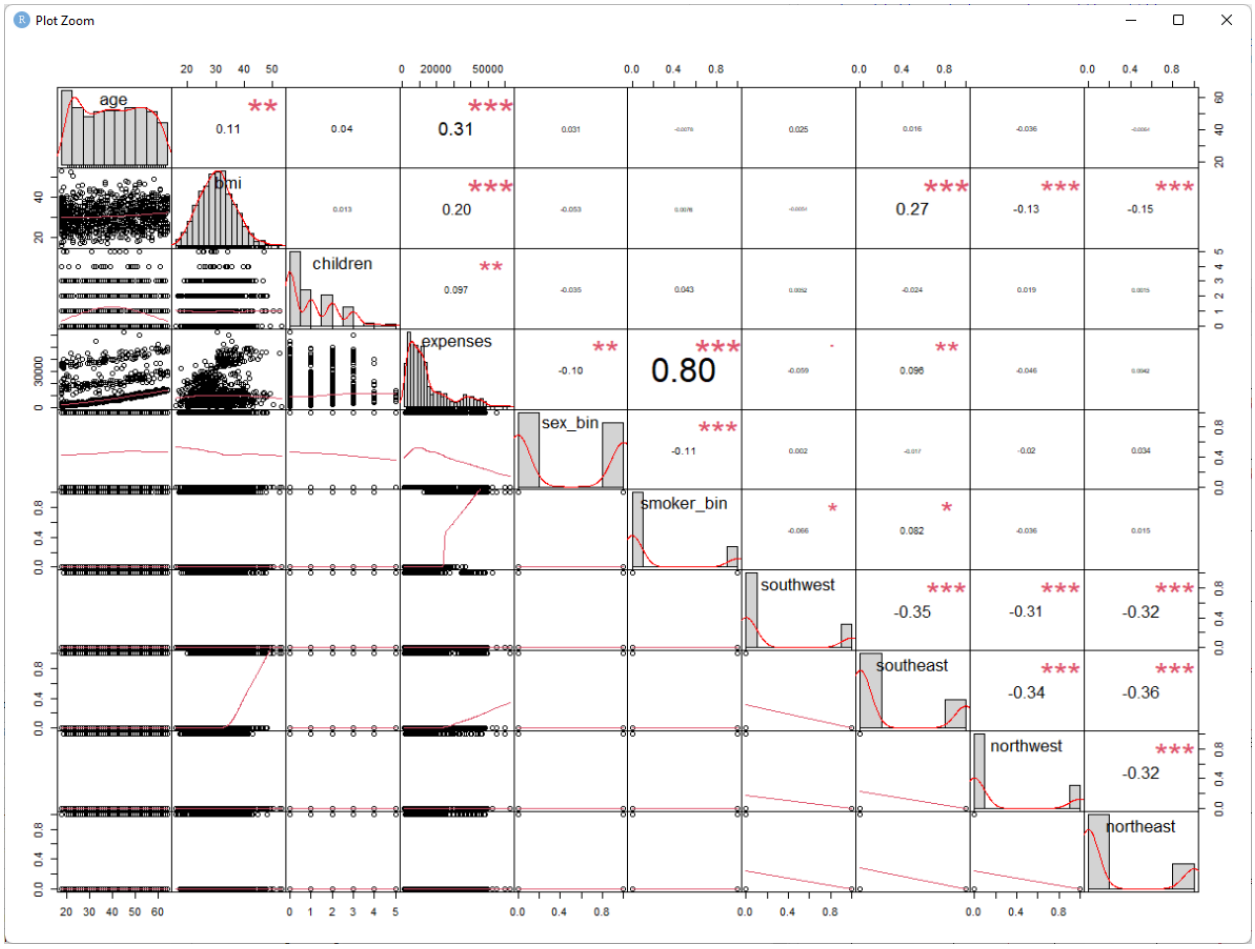


Рисунок 3.2 – Теплова карта заданої матриці кореляцій

```
linear_model <- lm(expenses ~ age + bmi +
children+sex_bin,smoker_bin+southwest+southeast+northwest+northeast,
data = data)
chart.Correlation(data, histogram = TRUE, method = "pearson")
```

Команда linear_model займається ініціалізаціалізацією та навчанням моделі лінійної регресії

```
for (y in 1:length(columns)) {
  hist(flatten_dbl(data[columns[y]]),main = paste("Histogram of" ,
columns[y]),
       xlab=columns[y])
}
columns <-c('age', 'bmi', 'expenses', 'children')
```

```
for (i in 1:length(columns)) {
  for (j in (i+1):length(columns)) {
    plot(flatten_dbl(data[columns[i]]),
flatten_dbl(data[columns[j]]),
         main=paste(columns[i], ' ', columns[j]),
         xlab=columns[i], ylab=columns[j], pch=19)
    print(paste(columns[i], ' ', columns[j]))
  }
}
```

3.3 Аналіз найбільш впливових точок кореляцій

Ця частина коду показую нам основні гистограми залезностей нашох бази данних.

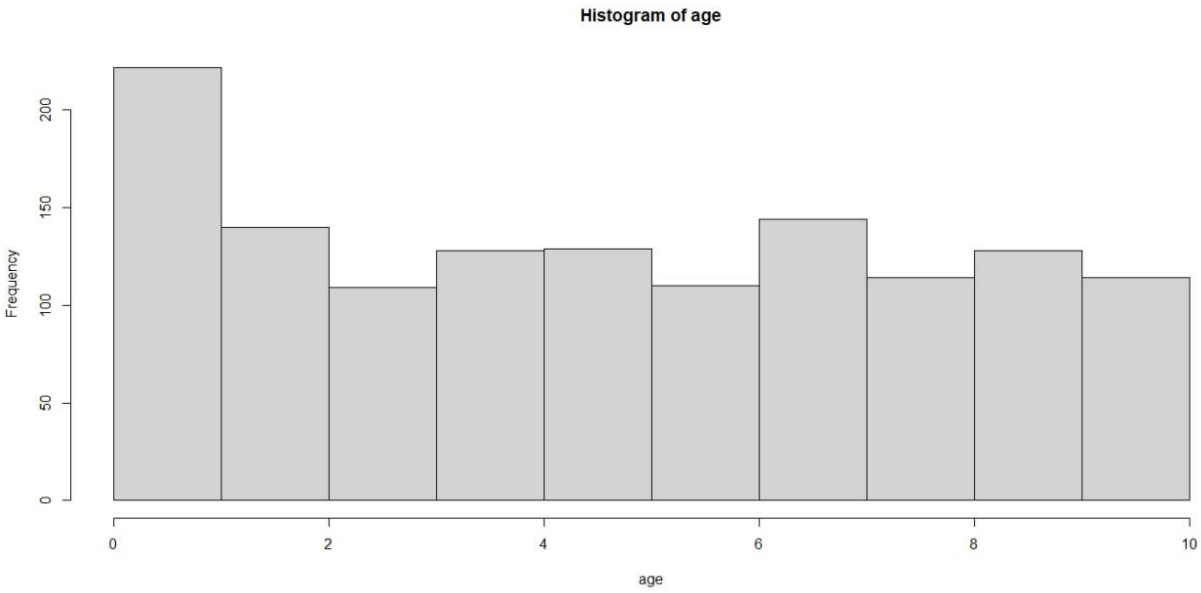


Рисунок 3.3 – Гістограма розподілу за віком

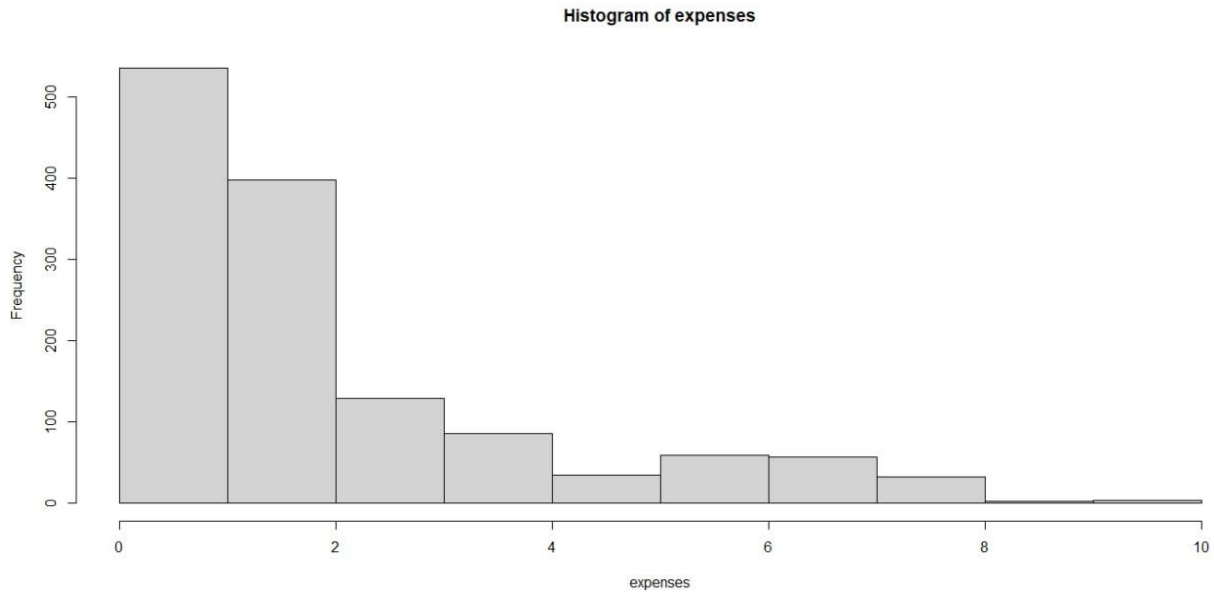


Рисунок 3.4 − Гістограма розподілу за кількістю витрат

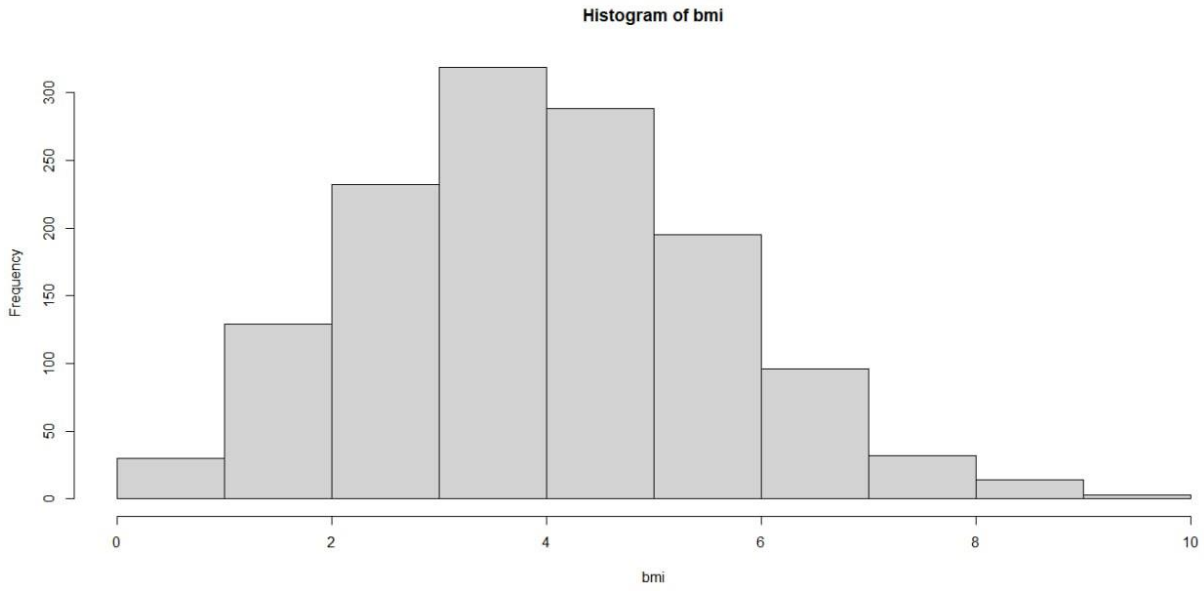


Рисунок 3.5 − Гістограма розподілу за індексом маси тіла (BMI)

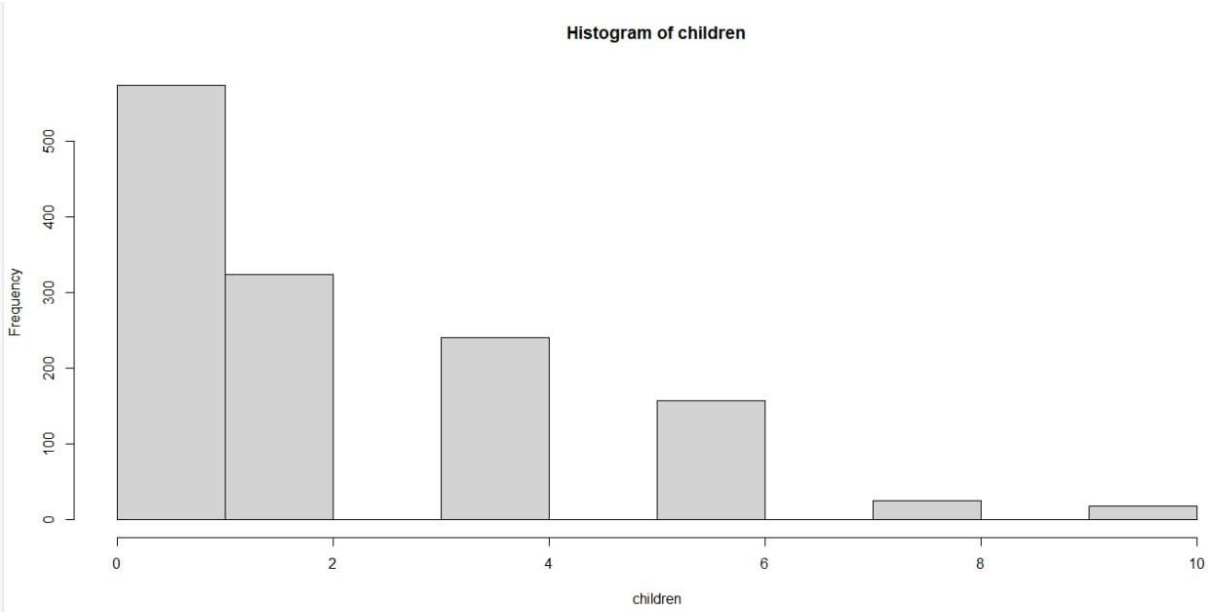


Рисунок 3.6 – Гістограма розподілу за кількістю дітей

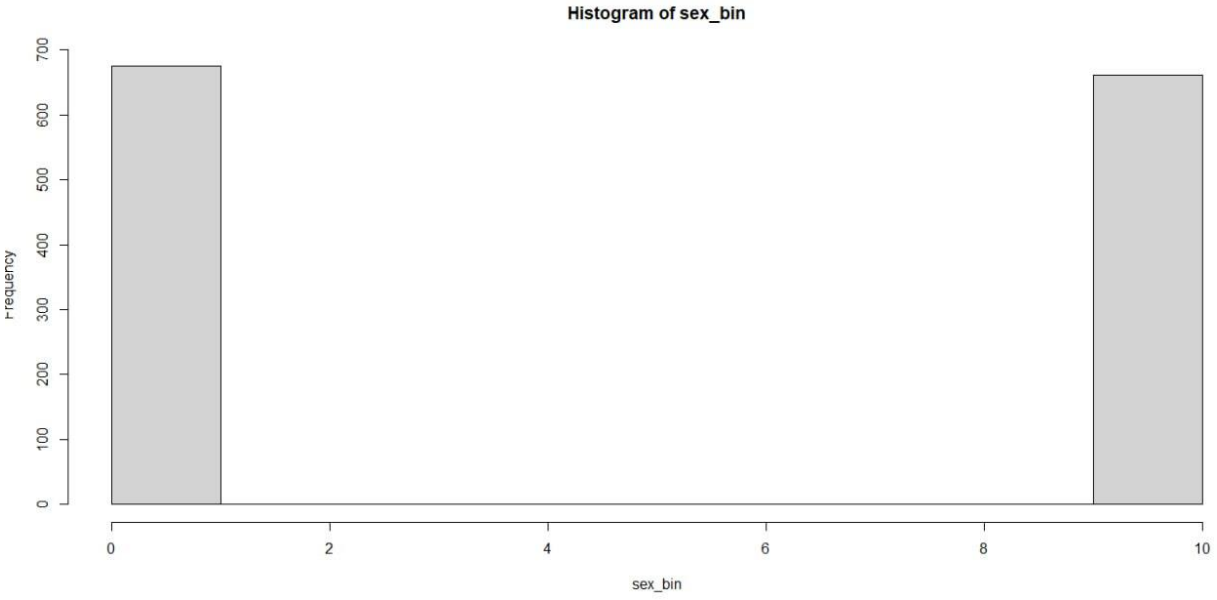


Рисунок 3.7 − Гістограма розподілу за статтю (видно, що характеристика бінарна, як і наступні 5)

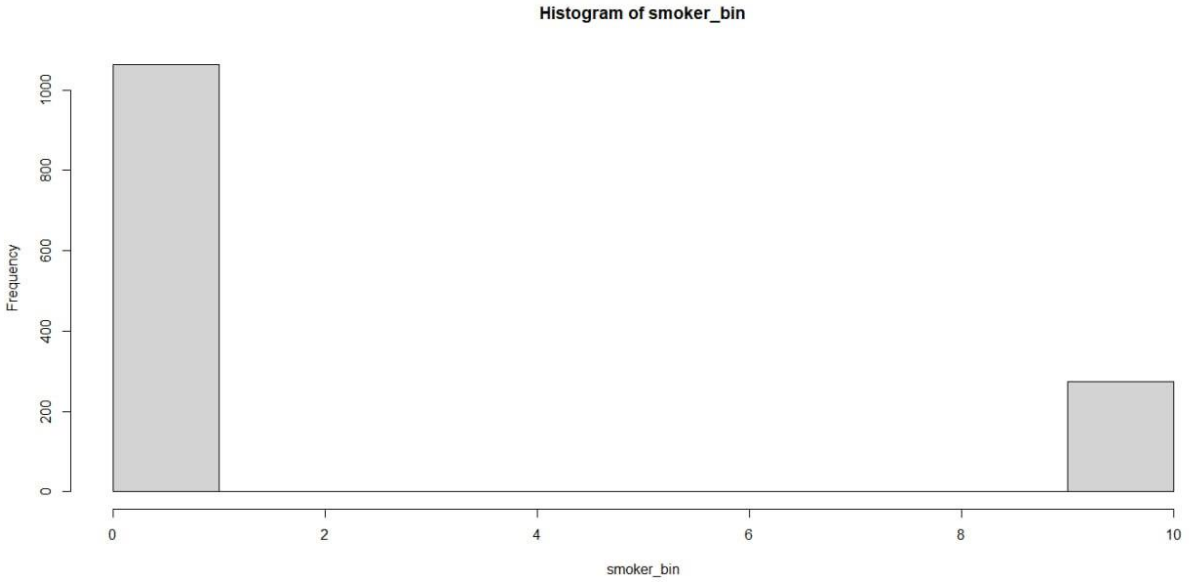


Рисунок 3.8 − Гістограма розподілу за тим, чи палить людина

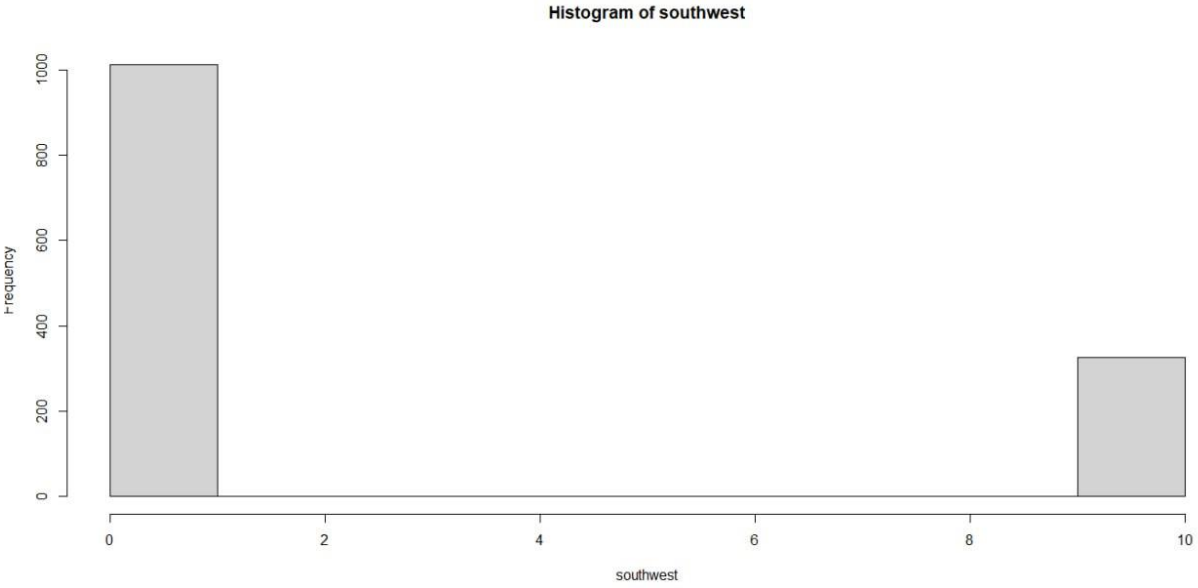


Рисунок 3.9 − Чи живе людина у регіоні"southwest"

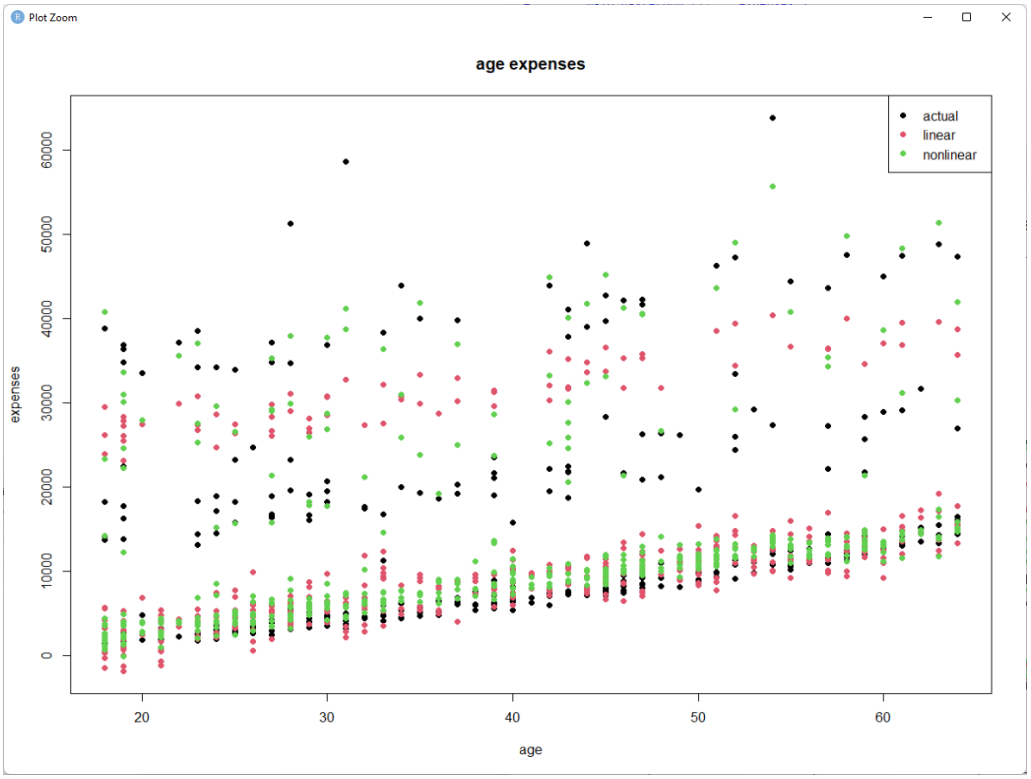


Рисунок 3.10 – Діаграма віку та витрат

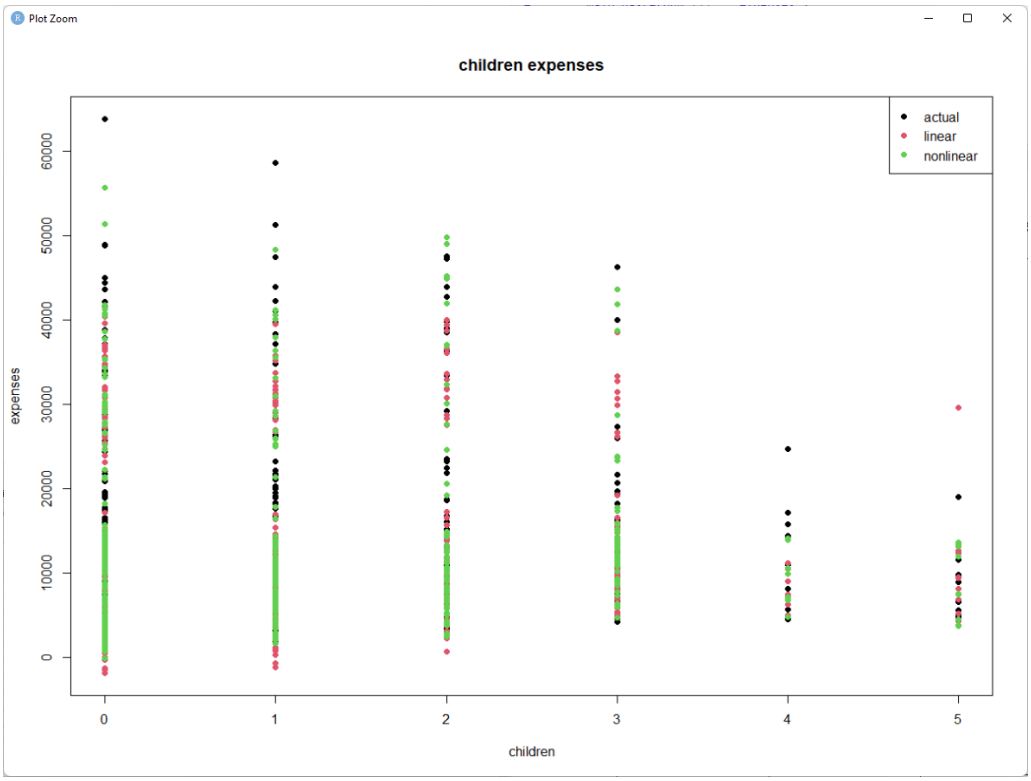


Рисунок 3.11 – Діаграма залежностей за витратами на медицину та кількістю дітей

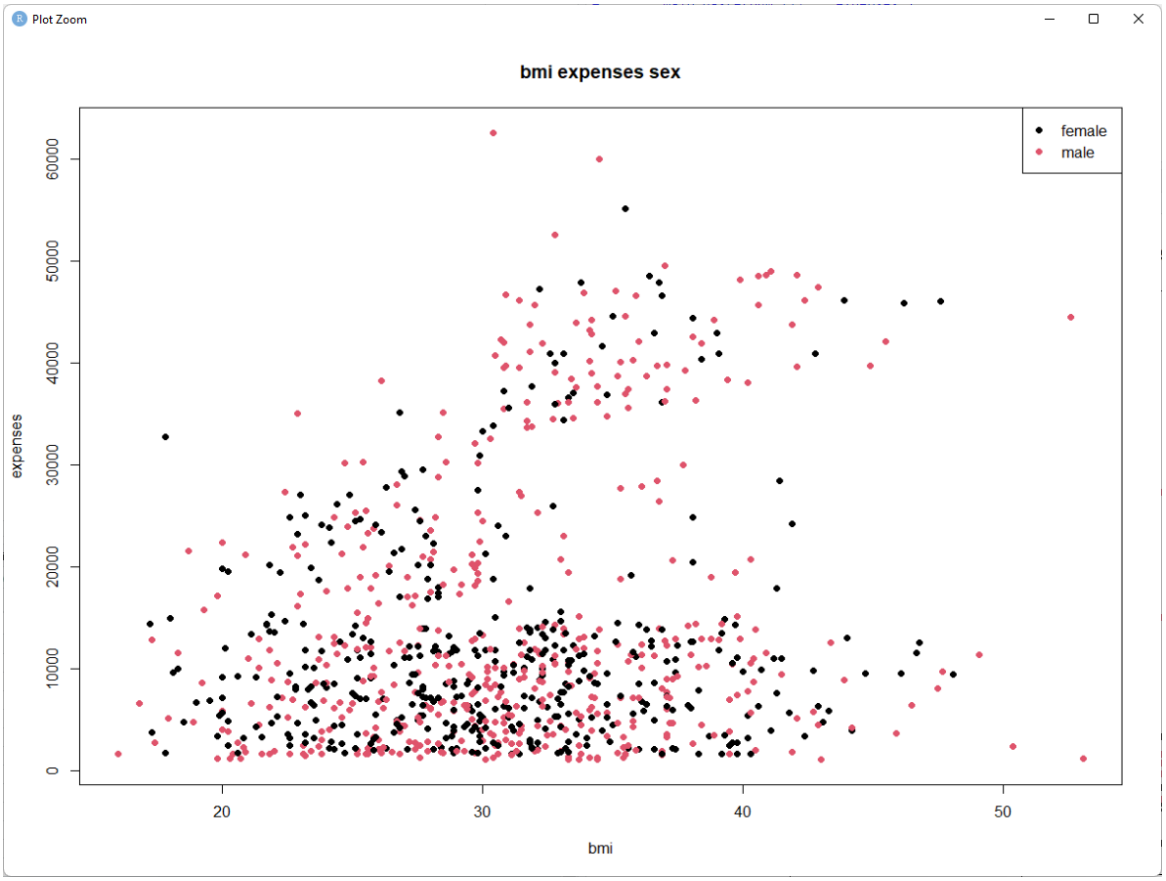


Рисунок 3.12 – Діаграма розсіювання індексу маси тіла, віку та статі (у програмі наведено усі подальщі залежності)

```
train  <- data[sample, ]

num_<-c('age', 'bmi', 'children')
cat_<-c('smoker', 'sex','region')
```

3.4 Тренування вибірки

Проводимо тренування вибірки, загалом було використано 910 варіантів. Будуємо діаграми розсіювання змінної витрат відносно інших числових параметрів.

Також розмальовуємо точки на діаграмі за кожним з категоріальних параметрів:

```
for (i in 1:length(num_)) {
  for (j in 1:length(cat_)) {
```

```
    plot(flatten_dbl(train[num_[i]]),
flatten_dbl(train['expenses']),
         main=paste(num_[i],  'expenses',cat_[j]),
         xlab=num_[i], ylab='expenses', pch=19,
         col=factor(flatten_chr(train[cat_[j]])))
    legend("topright",
           legend = levels(factor(flatten_chr(train[cat_[j]]))),
           pch = 19,
           col =
factor(levels(factor(flatten_chr(train[cat_[j]])))))

  }
```

Прибираємо необроблені змінні у наборі

```
data <- data[ , !(names(data) %in% c('region','sex','smoker'))]
```

Будуємо діаграми розсіювання змінної витрат відносно інших числових параметрів. Також розмальовуємо точки на діаграмі за кожним з категоріальних параметрів

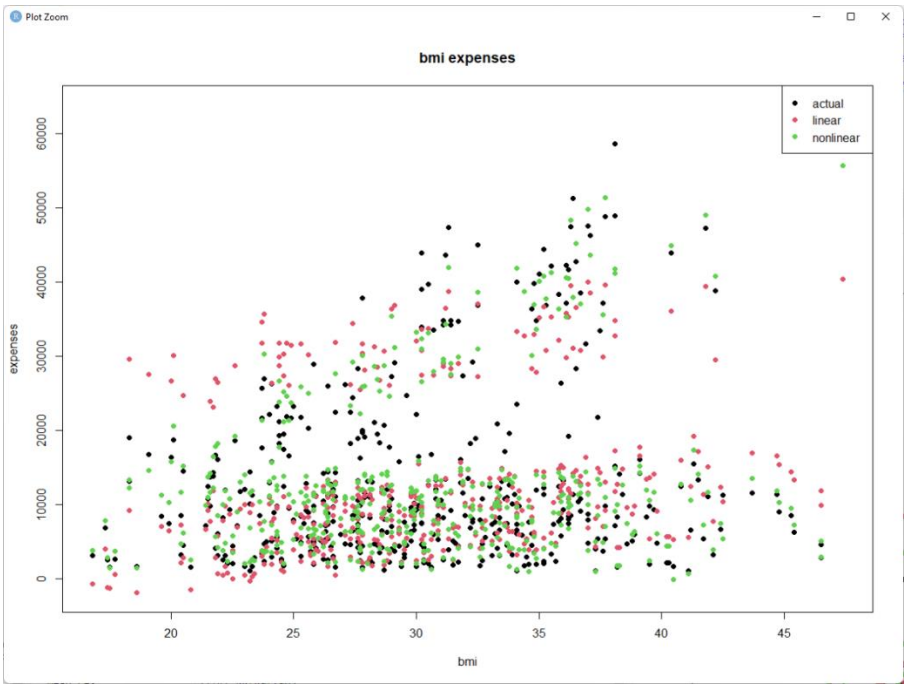


Рисунок 3.14 – Діаграма розсіювання індексу маси тіла та витратами намедицини

```
for (i in 1:length(num_)) {
```

```
  for (j in 1:length(cat_)) {
    plot(flatten_dbl(train[num_[i]]),
flatten_dbl(train['expenses']),
         main=paste(num_[i],  'expenses',cat_[j]),
         xlab=num_[i], ylab='expenses', pch=19,
         col=factor(flatten_chr(train[cat_[j]])))
    legend("topright",
           legend = levels(factor(flatten_chr(train[cat_[j]]))),
           pch = 19,
           col =
factor(levels(factor(flatten_chr(train[cat_[j]])))))

  }
```

Прибираємо необроблені змінні у наборі

```
data <- data[ , !(names(data) %in% c('region','sex','smoker'))]
```

Розбиваємо набір даних на тестову і тренувальну вибірки

```
train  <- data[sample, ]
test   <- data[!sample, ]
test_copy <- test[ , !(names(test) %in% c('expenses'))]
columns <-names(data)
```

Задаємо коєфіцієнти детермінації обох моделей (вони ж R^2. Чим ближче до одиниці, тим краще)

```
summary(linear_model)$r.squared
summary(nonlinear_model)$r.squared
```

Тестування тестовою вибіркою

```
lin_res<-predict(linear_model,test_copy)
nonlin_res<-predict(nonlinear_model,test_copy)
```

"print(R_nonlin)" рахуэмо коєфіцієнти детермінації для тестових вибірок

```
res_   <-flatten_dbl(test['expenses'])
mean_res <-mean(res_)
lin_e <-lin_res-res_
nonlin_e <-nonlin_res-res_
R_lin<-1-(
  sum(
    unlist(
```

```
      map(
        unlist(lin_e),
        function(x) (x)^2
      )
    )
  )/
    sum(
      unlist(
        map(
          unlist(lin_e),
          function(x) (x-mean_res)^2
        )
      )
    )
)
R_nonlin<-1-(
  sum(
    unlist(
      map(
        unlist(nonlin_e),
        function(x) (x)^2
      )
    )
  )/
    sum(
      unlist(
        map(
          unlist(nonlin_e),
          function(x) (x-mean_res)^2
        )
      )
    )
)
print(R_lin)
print(R_nonlin)
```

Вставляємо передбачення різних моделей та реальні значення тестової вибірки, щоб побудувати їх в одному вікні

```
res__<-test
res__['type']<-'actual'
tmp<-test_copy
tmp['expenses']<-lin_res
```

```
tmp['type']<-'linear'
res__<-rbind(res__, tmp)
tmp<-test_copy
tmp['expenses']<-nonlin_res
tmp['type']<-'nonlinear'
res__<-rbind(res__, tmp)

test_copy['expenses'] <- nonlin_res
cat_<-flatten_chr(unique(res__['type']))
```

Будуємо діаграми розсіювання змінної витрат відносно інших числових параметрів за передбаченнями різних моделей та реальними значеннями тестової вибірки

```
for (i in 1:length(num_)) {
  plot(flatten_dbl(res__[num_[i]]), flatten_dbl(res__['expenses']),
       main=paste(num_[i],  'expenses'),
       xlab=num_[i], ylab='expenses', pch=19,
       col=factor(flatten_chr(res__['type'])))
  legend("topright",
         legend = cat_,
         pch = 19,
         col = factor(cat_)
  )
}
```

Отримуємо результат.

Висновки до 3 розділу

У 3 розділі було реалізовано інтелектуальна система прогнозування яка складається з п’яти компонент. Перша компонента здійснює збір даних, аналіз та їх інтерпретацію. Друга – займається дослідженням та підготовкою даних до моделювання. Також здійснюються висновки на основі описової статистики, гістограм щільності розподілу для змінних, вивчення та візуалізації залежностей між ознаками (матриця кореляції та матриця розсіяння). Третя компонента здійснює навчання базової моделі на основі даних. Четверта компонента визначає ефективність базової моделі. П’ята здійснює прогноз на тестовому наборі даних на основі базової моделі.

ВИСНОВКИ

В рамках випускної кваліфікаційної роботи було розроблему інтелектуальну систему прогнозування медичних видатків на основі методів машинного навчання з винористанням медодів лінійної та нелінійної регресії. Система використовує тесьовий набір даних для прогнозування якісних результатів прогнозу медичних видатків серед заданої нами вибірки.

Витрати на лікування важко оцінити, оскільки є рідкісні випадки захворювань. Проте існує статистика проблем із здоров'ям, які є більш частими серед населення. Або випадки, які є більш поширеними для певної групи людей. Наприклад, рак легенів більш ймовірний серед курців, ніж людей, які ведуть здоровий спосіб життя, а захворювання серця можуть бути більш ймовірними серед людей, які мають таку проблему як ожиріння. У групу ризику також входять люди зі шкідливими звичками та низькою стресостійкістю.

Для того, щоб медична страхова компанія мала змогу заробляти гроші, вона повинна збирати щорічні внески більше, ніж витрачає на медичне обслуговування своїх бенефіціарів. Таким чином, страховикам необхідно витрачати багато часу та грошей на розробку моделей, які точно прогнозують медичні витрати населення, яке застраховане.

Використання такої системи прогнозування дозволить більш ефективно робити прогнози видатків страхових компаній та покращить комфорт для якість для користувачів таких послуг. Сьогодні машинне навчання вносить вагомий внесок у покращеня якості життя більшої частини людства, та у майбутньому стане невід'эмною та вагомою частиною користування в усіх сферах життя.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. N. Meade, T. Islam. Prediction Intervals for Growth Curve Forecasts // Journal of Forecasting. − 1995. − Т. 14, вип. 5. − С. 413−430.
2. Mitchell, Tom (1997). Machine Learning. New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892. Archived from the original on 2020-04-07.
3. R. M. Bethea, B. S. Duran, T. L. Boullion. Statistical Methods for Engineers and Scientists. − New York: Marcel Dekker, 1985.
4. Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks". Machine Learning. 20 (3): 273–297
5. Stevenson, Christopher. "Tutorial: Polynomial Regression in Excel". facultystaff.richmond.edu. Archived from the original on 2 June 2013.
6. Zhou, Victor (2019-12-20). "Machine Learning for Beginners: An Introduction to Neural Networks". Medium. Archived from the original on 2022-03-09.
7. Y. Bengio; A. Courville; P. Vincent (2013). "Representation Learning: A Review and New Perspectives". IEEE Transactions on Pattern Analysis and Machine Intelligence. 35 (8): 1798–1828
8. "Machine-learning models vulnerable to undetectable backdoors". The Register. Archived from the original on 13 May 2022.
9. Shapiro, Ehud Y. "The model inference system." Proceedings of the 7th international joint conference on Artificial intelligence-Volume 2. Morgan Kaufmann Publishers Inc., 1981.
10. Agrawal, R.; Imieliński, T.; Swami, A. (1993). "Mining association rules between sets of items in large databases". Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93. p. 207.
11. Stevenson, Christopher. "Tutorial: Polynomial Regression in Excel". facultystaff.richmond.edu. Archived from the original on 2 June 2013.
12. National Physical Laboratory (1961). "Chapter 1: Linear Equations and Matrices: Direct Methods". Modern Computing Methods. Notes on Applied Science. Vol. 16

(2nd ed.).

13. Hawkins, Douglas M. (1973). "On the Investigation of Alternative Regressions by Principal Component Analysis". Journal of the Royal Statistical Society, Series C. 22 (3): 275–286.
14. Draper, Norman R.; van Nostrand; R. Craig (1979). "Ridge Regression and James-Stein Estimation: Review and Comments". Technometrics. 21 (4): 451–466
15. Galton, Francis (1886). "Regression Towards Mediocrity in Hereditary Stature". The Journal of the Anthropological Institute of Great Britain and Ireland. 15: 246–263.
16. Lange, Kenneth L.; Little, Roderick J. A.; Taylor, Jeremy M. G. (1989). "Robust Statistical Modeling Using the t Distribution" (PDF). Journal of the American Statistical Association. 84 (408): 881–896.
17. Narula, Subhash C.; Wellington, John F. (1982). "The Minimum Sum of Absolute Errors Regression: A State of the Art Survey". International Statistical Review. 50 (3): 317–326.
18. Brillinger, David R. (1977). "The Identification of a Particular Nonlinear Time Series System". Biometrika. 64 (3): 509–515
19. Hidalgo, Bertha; Goodman, Melody (2012-11-15). "Multivariate or Multivariable Regression?". American Journal of Public Health. 103 (1): 39–40.
20. Tsao, Min (2022). "Group least squares regression for linear models with strongly correlated predictor variables". Annals of the Institute of Statistical Mathematics.
21. Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the Lasso". Journal of the Royal Statistical Society, Series B. 58 (1): 267–288.
22. Закон України “Про охорону праці” / Законодавство України про охорону праці. - К. Нова редакція 2002 р
23. Hansen L., Salamon P. Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence. San Diego, 1990. P. 993–1001.
24. Bachman, Charles W. (1973). "The Programmer as Navigator". Communications of the ACM. 16 (11): 653–658.
25. Москальова В. М. Основи охорони праці: підручник. Київ: ВД Професіонал,

2005. 666

26. Breiman, L. Bagging predictors. Machine Learning, Vol. 24 (2), pp. 123- 140, 1996
27. Загальна теорія статистики: Підручник/За ред. Р. А. Шмойлової. - 3-тє видання, перероблене. - Москва: Фінанси та Статистика, 2002. - 560 с.
28. Єлісєєва І. І., Юзбашев М. М. Загальна теорія статистики: Підручник / За ред. І. І. Єлісєєвої. - 4-те видання, перероблене та доповнене. - Москва: Фінанси та Статистика, 2002. - 480 с.
29. Карташов М. В. Імовірність, процеси, статистика. − Київ : ВПЦ Київський університет, 2007. − 504 с.
30. Hazewinkel, Michiel, ред. (2001). Correlation (in statistics). Енциклопедія математики[en]. Springer. ISBN
31. Croxton, Frederick Emory; Cowden, Dudley Johnstone; Klein, Sidney (1968) Applied General Statistics, Pitman. ISBN 9780273403159 (page 625)
32. Yule, G.U and Kendall, M.G. (1950), "An Introduction to the Theory of Statistics", 14th Edition (5th Impression 1968). Charles Griffin & Co. pp 258–270
33. Borsdorf, Rudiger; Higham, Nicholas J.; Raydan, Marcos (2010). Computing a Nearest Correlation Matrix with Factor Structure.. SIAM J. Matrix Anal. Appl. 31 (5): 2603–2622.
34. Qi, HOUDUO; Sun, DEFENG (2006). A quadratically convergent Newton method for computing the nearest correlation matrix.. SIAM J. Matrix Anal. Appl. 28 (2): 360–385.
35. Aldrich, John (1995). Correlations Genuine and Spurious in Pearson and Yule. Statistical Science 10 (4): 364–376.
36. Anscombe, Francis J. (1973). Graphs in statistical analysis. The American Statistician 27 (1): 17–21.
37. Роберт Кабаков. R у дії = R in Action. - ДМК-Прес, 2014. - 588 с.
38. Хедлі Уїкем, Гарретт Гроулмунд. Мова R у задачах науки про дані: імпорт, підготовка, обробка, візуалізація та моделювання даних = R for Data Science: Visualize, Model, Transform, Tidy, and Import Data. - Вільямс, 2017. - 592 с.
39. Норман Метлофф[en]. Мистецтво програмування на R. Занурення у великі дані =

The Art of R Programming: Tour of Statistical Software Design. - Пітер, 2019. - 416 с.

40. About Linear Regression [Електронний ресурс]. – Режим доступу : https://www.ibm.com/topics/linear-regression
41. Difference Between Linear and Nonlinear Regression [Електронний ресурс]. – Режим доступу : https://statisticsbyjim.com/regression/difference-between-linear-nonlinear-regression-models/
42. R Tutorial [Електронний ресурс]. – Режим доступу : https://www.w3schools.com/r/
43. Correlation Matrix - an overview [Електронний ресурс]. – Режим доступу : https://www.sciencedirect.com/topics/mathematics/correlation-matrix

ДОДАТОК А
Лістинг коду системи прогнозування медичних видатків

```
install.packages('cli', version='3.4.1')
install.packages(
  c(
    'purrr',
    'bruceR',
    'corrplot',
    'RColorBrewer',
    "PerformanceAnalytics",
    'tidyverse',
    'ggplot2'
  )
)
library(cli)
library(PerformanceAnalytics)
library(purrr)
library(bruceR)
library(corrplot)
library(RColorBrewer)
library(tidyverse)
library(ggplot2)

rm(list=ls())

filepath<-"C:\\Users\\Valentin\\Desktop\\diploma\\insurance.csv"
data <- read.csv(filepath)

sample <- sample(c(TRUE, FALSE), nrow(data), replace=TRUE, prob=c(0.7,0.3))

data['sex_bin'] <- flatten_dbl(map(data$sex, function(x) as.numeric(x=='female')))
data['smoker_bin'] <- flatten_dbl(map(data$smoker, function(x) as.numeric(x=='yes')))
regions <-unique(data$region)
print(regions)
for (y in 1:length(regions)) {
```

```
  data[regions[y]]<-flatten_dbl(map(data$region, function(x)
as.numeric(x==regions[y])))
}


train  <- data[sample, ]

num_<-c('age', 'bmi', 'children')
cat_<-c('smoker', 'sex','region')


for (i in 1:length(num_)) {
  for (j in 1:length(cat_)) {
    plot(flatten_dbl(train[num_[i]]), flatten_dbl(train['expenses']),
         main=paste(num_[i],  'expenses',cat_[j]),
         xlab=num_[i], ylab='expenses', pch=19,
         col=factor(flatten_chr(train[cat_[j]])))
    legend("topright",
           legend = levels(factor(flatten_chr(train[cat_[j]]))),
           pch = 19,
           col = factor(levels(factor(flatten_chr(train[cat_[j]])))))

  }
}

data <- data[ , !(names(data) %in% c('region','sex','smoker'))]

train  <- data[sample, ]
test   <- data[!sample, ]
test_copy <- test[ , !(names(test) %in% c('expenses'))]
columns <-names(data)

for (y in 1:length(columns)) {
  hist(flatten_dbl(train[columns[y]]),main = paste("Histogram of" , columns[y]),
       xlab=columns[y])
}

M <- cor(train)
```

```
corrplot(M, method="color")

chart.Correlation(train, histogram = TRUE, method = "pearson")

linear_model <- lm(expenses ~ age + bmi +
children+sex_bin+smoker_bin+southwest+southeast+northeast+northwest, data = train)

nonlinear_model <- lm(
  expenses ~
I(bmi*sex_bin*southwest)+I(bmi*sex_bin*southeast)+I(bmi*sex_bin*northwest)+I(bm
i*sex_bin*northeast)+I(bmi*sex_bin)+I(bmi*smoker_bin*southwest)+I(bmi*smoker_b
in*southeast)+I(bmi*smoker_bin*northwest)+I(bmi*smoker_bin*northeast)+I(bmi*sm
oker_bin)+I(bmi*southwest)+I(bmi*southeast)+I(bmi*northwest)+I(bmi*northeast)+b
mi+I(age*sex_bin*southwest)+I(age*sex_bin*southeast)+I(age*sex_bin*northwest)+I(
age*sex_bin*northeast)+I(age*sex_bin)+I(age*smoker_bin*southwest)+I(age*smoker_
bin*southeast)+I(age*smoker_bin*northwest)+I(age*smoker_bin*northeast)+I(age*sm
oker_bin)+I(age*southwest)+I(age*southeast)+I(age*northwest)+I(age*northeast)+age
+I(children*sex_bin*southwest)+I(children*sex_bin*southeast)+I(children*sex_bin*no
rthwest)+I(children*sex_bin*northeast)+I(children*sex_bin)+I(children*smoker_bin*s
outhwest)+I(children*smoker_bin*southeast)+I(children*smoker_bin*northwest)+I(chi
ldren*smoker_bin*northeast)+I(children*smoker_bin)+I(children*southwest)+I(childre
n*southeast)+I(children*northwest)+I(children*northeast)+children+I(sex_bin*southwe
st)+I(sex_bin*southeast)+I(sex_bin*northwest)+I(sex_bin*northeast)+sex_bin+I(smoke
r_bin*southwest)+I(smoker_bin*southeast)+I(smoker_bin*northwest)+I(smoker_bin*n
ortheast)+smoker_bin+southwest+southeast+northwest+northeast+northeast,
  data=train
)


summary(linear_model)$r.squared
summary(nonlinear_model)$r.squared

lin_res<-predict(linear_model,test_copy)
nonlin_res<-predict(nonlinear_model,test_copy)

res_   <-flatten_dbl(test['expenses'])
mean_res <-mean(res_)
lin_e <-lin_res-res_
```

```
nonlin_e <-nonlin_res-res_
R_lin<-1-(
  sum(
    unlist(
      map(
        unlist(lin_e),
        function(x) (x)^2
      )
    )
  )/
    sum(
      unlist(
        map(
          unlist(lin_e),
          function(x) (x-mean_res)^2
        )
      )
    )
)
R_nonlin<-1-(
  sum(
    unlist(
      map(
        unlist(nonlin_e),
        function(x) (x)^2
      )
    )
  )/
    sum(
      unlist(
        map(
          unlist(nonlin_e),
          function(x) (x-mean_res)^2
        )
      )
    )
)
print(R_lin)
```

```
print(R_nonlin)

res__<-test
res__['type']<-'actual'
tmp<-test_copy
tmp['expenses']<-lin_res
tmp['type']<-'linear'
res__<-rbind(res__, tmp)
tmp<-test_copy
tmp['expenses']<-nonlin_res
tmp['type']<-'nonlinear'
res__<-rbind(res__, tmp)

test_copy['expenses'] <- nonlin_res
cat_<-flatten_chr(unique(res__['type']))

for (i in 1:length(num_)) {
  plot(flatten_dbl(res__[num_[i]]), flatten_dbl(res__['expenses']),
       main=paste(num_[i],  'expenses'),
       xlab=num_[i], ylab='expenses', pch=19,
       col=factor(flatten_chr(res__['type'])))
  legend("topright",
         legend = cat_,
         pch = 19,
         col = factor(cat_)
  )
}
```