МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра прикладної математики та моделювання складних систем

Допущено до захисту
В.о. завідувача кафедри ПМ та МСС
_____ доц. Ігор КОПЛИК
«____» _____ 20__р.

КОМПЛЕКСНА КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня «бакалавр»
спеціальність 113 «Прикладна математика»
освітньо-професійна програма «Прикладна математика»

тема роботи **«Моделювання тенденцій на ринку праці IT сектору України»**

Виконавець
Студентка факультету ЕлІТ
Владислава МОЛЧАН _____

Науковий керівник
к.е.н., доцент
Тетяна МАРИНИЧ _____

Суми
2020

РЕФЕРАТ

Кваліфікаційна робота: 72с., 27 рисунків, 6 таблиць, 16 джерел, 4 додатки.

Мета роботи: статистичний аналіз та моделювання заробітної плати на ринку праці ІТ сектору України.

Об'єкт дослідження: ринок праці ІТ-сектору України.

Предмет дослідження: визначення залежності заробітної плати в ІТ-секторі України від сукупності факторів, що включають: навички, рівень освіти, вік, стать, посада працівників, пропозиції на ринку.

Методи навчання: очищення та підготовка даних, статистичний аналіз, тестування гіпотез, регресійний аналіз, непараметричні моделі дерева рішень та випадкового лісу (Random Forest).

Ключові слова: ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, ІТ-РИНОК, СТАТИСТИЧНИЙ АНАЛІЗ ДАНИХ, ЛІНІЙНА РЕГРЕСІЯ, ДЕРЕВО РІШЕНЬ, МОДЕЛЮВАННЯ, ЗАРОБІТНА ПЛАТА**.**

ЗМІСТ

ВСТУП

Інформаційні технології можуть використовуватися не лише для аналізу і обробки теперішньої інформації, а і для прогнозування і маркетингового дослідження на майбутнє. Все більше країн опираються на технології і їх розробку, тому велика кількість людей залучаються у дану галузь. Великі перспективи, висока заробітна плата та комфортні умови праці, роблять цю сферу привабливою саме для молодих спеціалістів.

В Україні, в останні роки український IT-ринок збільшився вдвічі і став одним з головних напрямків експорту послуг. Якщо у світі динамічно розвиваються такі галузі як: програміст-розробник, UI/UX і Product дизайнер, системні аналітики, Big Data спеціалісти та архітектори програмного забезпечення, то в Україні найбільш популярними за даними dou.ua професіями на IT-ринку є програміст розробник, Web, системний адміністратор, тестувальник, data scientist, data engineer.

Метою цієї роботи є вивчення загальних та специфічних тенденцій на ринку праці IT-сектору України, аналіз та моделювання факторів, які впливають на рівень заробітної плати IT-спеціалістів. Для цього використано дані Державного управління статистики України ukrstat.gov.ua, дані відкритої платформи dou.ua.

Досліджено статистичні методи аналізу даних, висновувальний аналіз регресійні моделі та кластерний аналіз. Програмна реалізація здійснена у середовищі RStudio.

Відповідно до мети дослідження, основними задачами роботи є:

1. Аналіз джерел для вивчення доменної галузі - ринку IT послуг в Україні та світі.
2. Виявлення факторів, що впливають на формування попиту та пропозиції на ринку, та, відповідно на визначення з/пл.
3. Огляд методів математичного моделювання та статистичного аналізу в контексті теми дослідження.
4. Збір та підготовка даних для емпіричного дослідження.
5. Побудова моделей та підготовка висновків.

1. АНАЛІЗ ГАЛУЗІ ДОСЛІДЖЕННЯ

1.1 Тенденції розвитку IT-галузі в Україні та світі

Сьогодні всі країни, незалежно від соціально-економічного стану активно розвивають інформаційні технології та максимально діджиталізують усі процеси. Результатом є збільшення числа зайнятих людей в цій сфері. За даними , частка робітників IT галузі в Україні з кожним роком зростає на 10%, наприкінці 2019 року кількість працівників становила 220 тисяч робітників. За даними дослідження N-iX [1] вклад цієї галузі в економіку країни сягає 5 млрд. дол. США щорічно , і має стійку тенденцію до зростання.

Інформаційні потреби різних типів ростуть швидкими темпами, це розширює можливості інформаційного світу, стимулює появу нових інформаційних продуктів та розвиток усіх видів інформаційної роботи.

Загалом IT індустрія представляє собою широкобічне виробництво, починаючи від складних систем і товарів і закінчуючи контентом пов’язаним з IT. Вона складається з двох суттєво різних частин: виробництво інформаційної техніки і виробництво безпосередньо інформації. IT-послуги включають три основні сегменти: IT-аудит, IT-аутсорсинг, IT-консалтинг. На даний момент послуги сфери IT спрямовані на:

- підвищення ефективності виробництва;
- скорочення витрат компаній;
- оптимізацію технологічних та адміністративних процесів;
- зберігання та захист інформації та каналів її передачі.

Сьогодні технологіями IT індустрії, які найбільш динамічно розвиваються і мають потенціал у майбутньому, є:

- автоматизація бізнес-процесів;
- обробка великих об’ємів даних у реальному режимі часу;
- хмарні сервіси;
- технології блокчейну;

- біометричні системи;
- технології штучного інтелекту (безпілотний транспорт, розпізнавання тексту, зображень тощо);
- доповнена та віртуальна реальність (AR / VR);
- підвищення безпеки передачі та зберігання даних.

Технології майбутнього вимагають зростання центрів обробки даних (ЦОД). Сучасні ЦОД поділяють на корпоративні (обслуговують конкретну компанію) і комерційні (надають послуги всім бажаючим користувачам). Для багатьох компаній надійність безперебійного функціонування обладнання та мережевої інфраструктури стає найважливішим фактором для зростання бізнесу. Тому ринок хостингових послуг в ЦОД буде продовжувати рости. Буде спостерігатися і зростання постачальників хмарних послуг. Перевага в тому, що ви отримуєте доступ до своїх даних з будь-якої точки світу, підключившись до Інтернету.

Біометричні системи існують не одне десятиліття. Але сьогодні цей ринок отримує додаткові перспективи. Їх будуть використовувати не тільки для забезпечення безпеки, але і для поліпшення комунікацій з клієнтами. У банках для проведення транзакцій клієнтам не потрібні будуть паспорта, пластикові карти, SMS. Нові IT-технології дозволяють розширити застосування біометричних систем.

Також IT-технології змінюють автомобільний ринок. Tesla збирається запустити в 2020р таксі-безпілотники. Автономні машини відрізняє висока безпека (виключені аварії через утому і неуважність водія, суворе дотримання правил дорожнього руху).

Технології AR/VR продовжують розширювати сфери застосування. Найбільш активно використовуються в ігровій індустрії (забезпечує видовищність і повне занурення в штучний світ). AR може бути застосована в маркетингу і рекламі. Величезний потенціал у віртуальній і доповненої реальності для використання в освіті, архітектурі, будівництві та медицині. Віртуальні анатомічні атласи відтворюють зовнішні і внутрішні

характеристики людських органів і тканин, так допомагаючи навчання медичного персоналу.

Світ блокчейн не настільки сильно увійшов в наше життя, як штучний інтелект, але все зараз можна ставити його на одне місце зі звичайними фінансовими операціями. Через те, що блокчейн система дозволяє децентралізувати процес, це значно спрощує і робить його дешевим.

Середа квантових обчислень тільки зароджується. Але перспективи її воістину безмежні. Існує безліч моделей, вивчення яких стандартними аналітичними алгоритмами неможливо або недоцільно. Через це аналіз переходить на рівень кубітів - найдрібніших елементів інформації.

При зборі та аналізі даних до сих пір головну роль грає людина. Обчислювальні пристрої використовуються переважно для здійснення розрахунків і систематизації інформації. Доповнення ж комп'ютерних програм нотками штучного інтелекту дозволить створити додатки, що здатні до самонавчання зі збору та аналізу даних і це збільшить швидкість і якість обробки великих масивів даних.

Тенденції розвитку IT індустрії переходять на всебічне використання технологій у різних сферах життя людини. Саме для цього потрібно шукати більш швидкі, дешеві і зручні способи обробки даних. І головним рушієм в цьому процесі виступає штучний інтелект.

1.2. Вітчизняний ринок IT-послуг у цифрах

В Україні сектор IT-послуг активно розвивається. Якщо у 2012 р. внесок IT в економіку країни становив 0,8% ВВП України, то у 2019 р. – уже 4%. В середньому IT індустрія в останні 5 років зростала на 20-25% щорічно. IT-сектор займає другу позицію за обсягом експорту послуг в країні, який дає більше як 5 млрд доларів на рік [2]. Основними замовниками послуг є країни Європи та Росія (рис. 1). Розгорнута географічна структура зовнішньої торгівлі IT-послугами наведена у Додатку А.

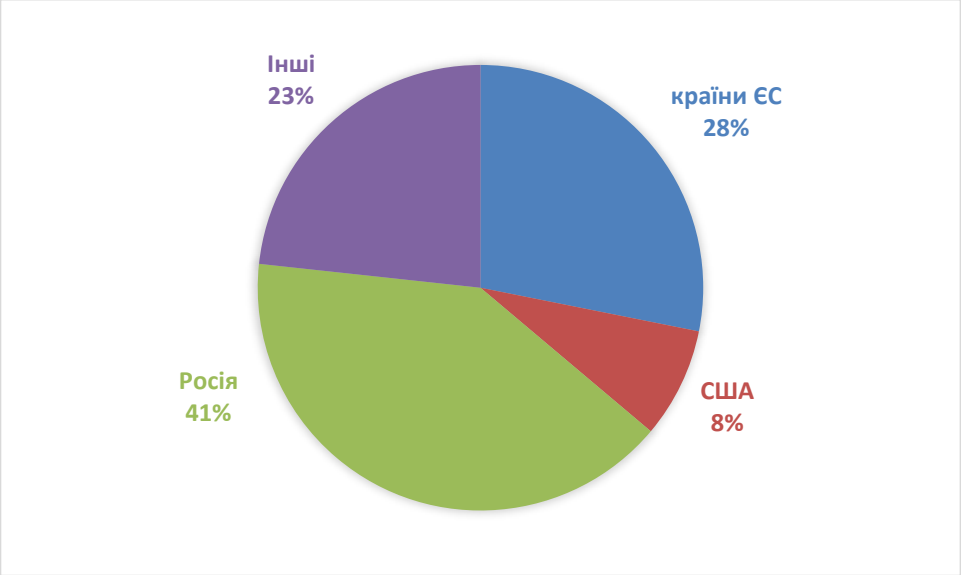

Рис. 1. – Географічна структура експорту IT-послуг у 2018 році [3].

За видами послуг, що експортуються домінують комп'ютерні послуги – 1,92 млрд дол. США у 2018 р. (79,3%), решту – $395,5 млн дол. США (16,7%) складають інформаційні послуги [3]. Найбільш популярними видами IT-проєктів в Україні є: розробка програмного забезпечення, електронна комерція, сервісне обслуговування обладнання та мобільні технології.

На внутрішньому ринку домінують послуги розробки програмного забезпечення на замовлення підприємств, організацій та установ (Таблиця 1).

Таблиця 1. – Структура внутрішнього ринку IT - послуг [4].

Види IT - послуг	Обсяг реалізованих послуг усього, млн.грн	У т.ч. реалізовано послуг, млн.грн		
		населенню	підприємствам (установам)	іншим категоріям споживачів
Комп'ютерне програмування, консультування та пов'язана з ними діяльність	12457,1	14,3	11595,5	847,3
Надання інформаційних послуг	4008,6	377,7	3400,7	230,2
Оброблення даних, розміщення інформації на веб-вузлах і пов'язана з ними діяльність; веб-портали	3226,7	271,2	2737,6	217,9
Надання інших інформаційних послуг	781,9	106,5	663,1	12,4

За даними [5], кількість людей, які працюють у сфері IT (офіційно та неофіційно) в останні 5 років зростає на 10% щорічно (рис. 2).

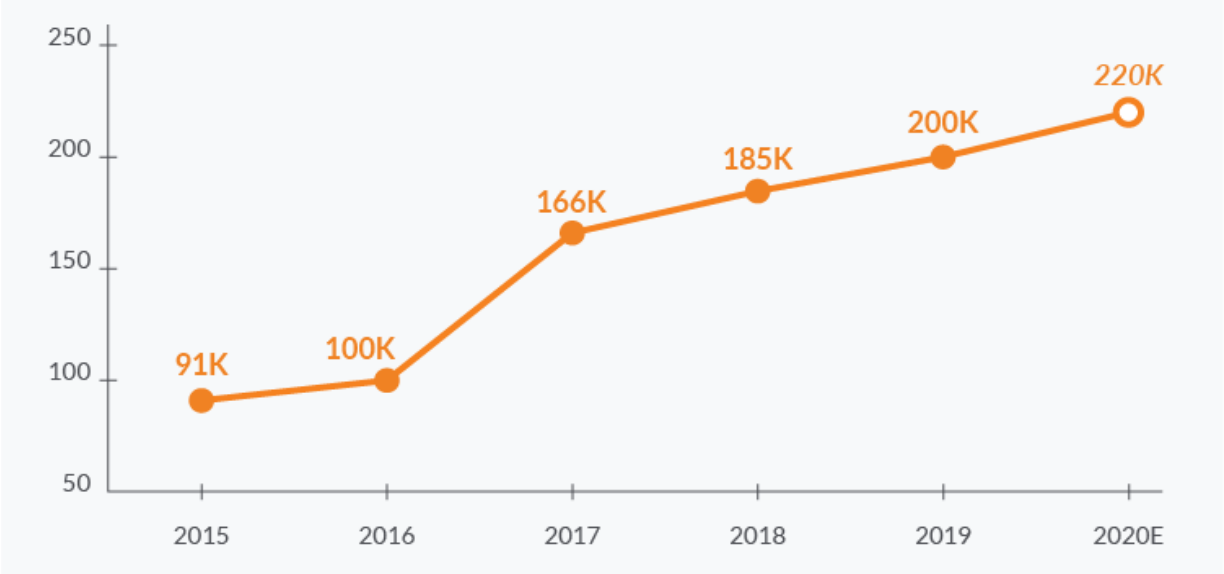

Рис. 2. – Кількість людей у сфері IT в Україні за останні 5 років [5].

Перспективи подальшого розвитку галузі залежать не тільки від бізнесу та рівня підготовки фахівців, а і від державної політики та підтримки.

Зараз діапазон заробітної плати фахівців IT компаній становить від 400 до 3000 доларів США за місяць [5], що значно перевищує рівень винагороди у інших галузях. Динаміка середньої місячної заробітної плати (у гривнях) за останні чотири роки представлена на рис. 3.

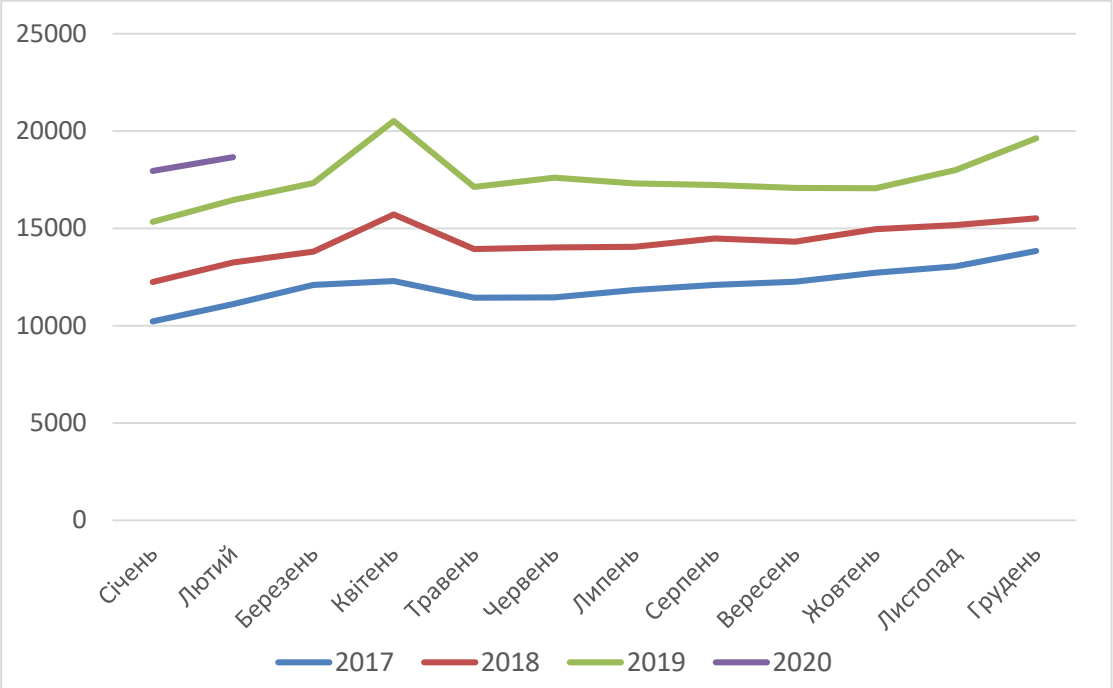

Рис. 3. – Середня заробітна плата IT-спеціалістів в Україні [3].

Серед чинників, що впливають на розмір заробітної плати, основними є тип проєктів і мова програмування, що використовується. Оскільки в Україні не велика кількість власних розробок і головним чином компанії працюють на аутсорсинг та підтримку проєктів для інших компаній, то заробітна плата буде меншою ніж у колег, які створюють нові і інноваційні проєкти.

Важливим фактором є навчання фахівця. Великі компанії дивляться не тільки на диплом, а і на наукові статті майбутнього працівника. Чим вище в рейтингу посідає університет або академія, тим вища заробітна плата чекає на студента в майбутньому. Основні навички якими повинен володіти фахівець поділяються на soft skills та hard skills. Soft skills – це, ті навички, що не пов'язані з конкретною професією, але допомагають добре виконувати свою роботу і важливі для кар'єри, такі як:

- Навички спілкування, адаптації у колективі та робота в команді;

- Здатність безперервно навчатися та до самонавчання;
- Нестандартне і адаптивне мислення;
- Здатність обробляти великі обсяги інформації та вміння виділяти головне;
- Міжкультурна компетентність - необхідно для вдалої взаємодії в будь-якій міжнародній компанії та для максимально ефективної комунікації з потенційними клієнтами.

Hard skills (технічні навички) спрямовані до певної сфери діяльності, для IT-сфери це знання мов програмування, алгоритмів, шаблонів проєктування. Хоча технічних навичок менше, але без них soft skills не значили б абсолютно нічого для роботи в IT компанії.

1.3. Чинники, що визначають рівень заробітної плати ІТ-фахівця.

З економічної теорії відомо, що рівень заробітної плати визначається як ціна на ресурс – людський капітал і формується під впливом факторів попиту і пропозиції на ринку.

За даними інтернет-порталу dou.ua [2]., наразі найбільш затребуваними спеціальностями на ринку праці є:

- програміст розробник – займається написанням коду і програмного продукту на замовлення промислових та сільськогосподарських підприємств, фінансових, державних та комунальних установ і організацій;
- web – майстер – займається розробкою або підтримкою сайтів та веб додатків;
- системний адміністратор – слідкує за роботою комп'ютерної техніки, мереж та програмного забезпечення;
- тестувальник – перевіряє якість програмного продукту, шукає помилки та дефекти;
- спеціаліст з підтримки програмного забезпечення та обладнання;
- data scientist –досліджує великі об'єми даних та вилучає корисну інформацію з них;
- data engineer – спеціаліст, який досліджує великі об'єми даних.

Майже всі професії ІТ мають схожу ієрархію: Trainee – Junior – Middle – Senior. На початку професійного шляху спеціаліст стартує з позиції стажера (trainee), яка передбачає невелику зарплатню (як для ІТ індустрії в цілому) та початкові навички для роботи. Якщо пройти цей етап, де відсіюється велика кількість працівників, то можна перейти на рівень Junior, де вам дозволять працювати самостійно над невеликими проєктами або допомагати на великих проєктах. Спеціалістам рівня Middle доручають керівництво або супроводження складних проєктів, що дає значний приріст в заробітній платі. Останнім рівнем є Senior – має велику зону відповідальності і може без перешкод керувати проєктами або бути лідером команди. Таку посаду можна

отримати лише після 5-10 років стажу, але і заробітна плата складає більше 10000$ США на місяць [6].

Популярними спеціальностями в IT зараз є: системний адміністратор, веб-майстер та веб-програміст, розробник відеоігор, тестувальник, аналітик програмного забезпечення, архітектор програмного забезпечення [7]. А найбільш затребуваними спеціальностями в 2020 році являються: UX/UI-дизайнер, Blockchain-спеціаліст, SEO-спеціаліст, спеціаліст з кібербезпеки, розробник мобільних додатків та SMM-спеціаліст [8].

2. ОГЛЯД ПІДХОДІВ ТА МЕТОДІВ ДОСЛІДЖЕННЯ

Для емпіричного підтвердження або спростування експертних висновків, а також більш глибокого розуміння проблеми та формулювання управлінських рішень і рекомендацій, часто використовують статистичний аналіз даних та моделювання.

Поширений алгоритм аналізу представлено на рис. 4.

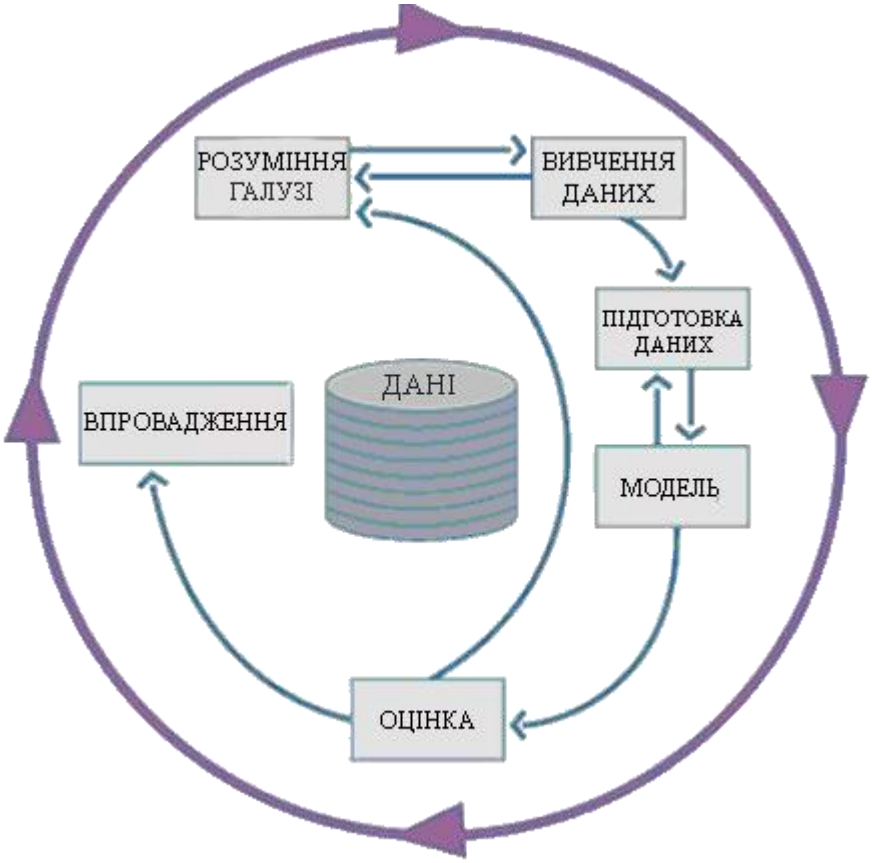

Рис. 4. - Діаграма CRISP-DM (Cross-Industry Standard Process for Data Mining).

Модель життєвого циклу дослідження даних складається з шести фаз: розуміння цілі, вивчення та розуміння даних, їх підготовка, моделювання, оцінка та впровадження.

2.1. Підготовка та очищення даних

Перед тим як приступити до аналізу даних та їх обробки, необхідно підготувати та очистити дані. Точність даних – важлива вимога аналізу, яка забезпечується шляхом недопущення, виправлення, усунення допущених помилок.

Можна виділити (умовно) такі етапи підготовки даних до аналізу:

- збір та імпорт даних;
- приведення даних до довгого формату, де кожна колонка відповідає унікальній змінній, а кожен рядок відповідає назві об’єкту;
- перевірка на повноту і очищення даних;
- заповнення пропущених значень.

Також можлива спеціальна підготовка даних, яка являє собою перетворення інформації у форму, зручну для обробки і аналізу.

Перевірка даних на правильність і логічну сумісність може здійснюватися одночасно із введенням або після його завершення, при цьому повнота даних перевіряється візуальним шляхом.

Оскільки у даних які були розглянуті багато пропущених значень, то подивимось на основні методи роботи з пропущеними даними.

Багато змінних (в багатьох соціологічних дослідженнях, що ґрунтуються на масових опитуваннях, переважна їх більшість) мають пропущені значення. Останні ведуть до зниження статистичної потужності, знижують ймовірність знаходження реальних закономірностей в даних, а також можуть бути причиною систематичних помилок . Обробка пропущених значень є досить розвиненою дослідницької областю з загальноприйнятою термінологією і безліччю рішень для різних дисциплін і конкретних досліджень.

Існує три основних типи відсутніх даних:

- Повністю випадкові (MCAR);
- Випадкові (MAP);
- Невипадкові пропуски (NMAR).

Постає питання, що робити з пропусками, виділяються декілька популярних способів:

1. Нічого не робити – деякі алгоритми добре справляються з проблемою відсутніх даних. Вони використовують найліпші дані для моделі або зовсім не враховують їх [9].

2. Використання середнього (медіанного) значення – працює шляхом визначення середнього значення не пропущених значень у стовбці, і заміною цими значеннями пропущених. Має низку мінусів такі, як: працює лише в числовими значенням, не враховує кореляцію між стовбцями, не є точним методом [9].

3. Використання найбільш популярного (моди) значення – працює з категорійними змінними, замінюючи відсутні дані на найбільш часто зустрічаючими значеннями. З мінусів не враховує кореляцію між стовбцями, що може призвести до зміщення даних [9].

4. Використання k-NN – алгоритм який використовує метод «схожості признаків» для прогнозування значень нових даних. Це означає, що новій точці присвоюється значення, засноване на тому, наскільки близько воно відповідає точкам в тренувальному наборі [9]. Це може бути дуже корисно для прогнозування пропущених значень шляхом знаходження найближчих сусідів до спостереження з відсутніми даними і подальшого поставлення їх на основі не пропущених значень в околиці. З мінусів можна виділити велику загрузку на пам’ять комп’ютера, через те, що потрібно зберігати навчальні дані в пам’яті.

5. Використання Multivariate Imputation by Chained Equation (MICE) – алгоритм працює, заповнюючи пропущені значення декілька разів. Кілька імпутацій (MI) набагато краще, ніж одна імпутація, оскільки вони краще вимірюють невизначеність відсутніх значень. Підхід ланцюгових рівнянь також є дуже гнучким і може обробляти різні змінні різних типів даних (наприклад, безперервні або двійкові), а також складності, такі як діапазони пропусків або схеми пропуску опитування [10].

6. Використання глибокого навчання – метод дуже гарно працює з категорійними даними і нечисловими функціями. Datawig – бібліотека яка вивчає дані з використанням нейронних мереж, для розрахунку відсутніх даних. З мінусів можна відмітити повільну роботу при великих об’ємах даних, робота з однією колонкою [9].

2.2. Описова статистика

Для розуміння кількісних даних, симетричності розподілу і форми, наявності аномальних спостережень використовують описову статистику, що включає показники центральної тенденції, міри варіації або мінливості даних, коефіцієнти ексцесу та асиметрії.

а) Показники центральної тенденції

Показники центральної тенденції визначають центр розподілу спостережень і включають математичне сподівання, середнє значення, медіана та мода.

Математичне сподівання ($\bar{X}$) неперервної кількісної ознаки визначають як центр тяжіння розподілу й визначають за формулою [11]:

$$\bar{X} = \int_{-\infty}^{+\infty} xf(x)dx, \quad (1)$$

де $f(x)$ – щільність розподілу.

Для дискретних випадкових величин, що виміряні у кількісних шкалах, замість математичного сподівання використовують середнє значення (середнє арифметичне [6].

$$\mu = \sum_{i=0}^{\infty} x_i\, p_i, \quad (2)$$

де x_i – значення випадкової величина, розподіленої на інтервалі $(-\infty;\ +\infty)$; p_i – ймовірність цих значень.

Вибіркове середнє (середнє арифметичне значення) – найпоширеніша статистика центральної тенденції для чисельних даних [11]:

$$\overline{X} = \frac{\sum_{i=1}^{n} x_i}{n}, \quad (3)$$

де $\overline{X}$ – середнє вибірки даних; n – кількість спостережень; x_i – значення спостережень.

Вибіркова медіана (Me) є загальною характеристикою центра розподілу, що базується на принципі симетрії, і представляє середнє спостереження, яке розподіляє набір впорядкованих даних у рівних половинах [11].

Медіана є кращою мірою центральної тенденції для вибірки даних з викидами, тому що невелика кількість аномальних значень на неї не впливають.

Вибіркова мода (Мо) – значення, яке найчастіше зустрічається в наборі даних.

Мода позначає пік(и) розподілу даних. Наявність двох і більше піків (вершин) свідчить про неоднорідність вибірки даних, про поєднання в ній груп з різними рівнями ознаки. У такому разі необхідно більш ретельно проаналізувати наявну вихідну інформацію, перегрупувати дані, виділивши однорідні групи [11].

Геометричне середнє – узагальнює дані та використовується як міра центральної тенденції для позитивно зміщеної вибірки даних (Середнє > Медіана). При цьому логарифмування використовується для мінімізації ефекту екстремальних спостережень. Для розрахунку геометричного середнього:

- візьміть натуральний логарифм усіх значень даних (ln);
- розрахуйте середнє прологарифмованих даних;
- проекспонуйте отримане значення.

б) Показники змінюваності

Міри змінюваності описують діапазон відхилень від центральної тенденції даних і включають наступні показники:

• інтервал - визначається двома значеннями (Інтервал=*Max – Min*) і не надає інформації про значення між ними. Інтервал може бути застосований, коли відома Медіана.;

• квартильний - поділяють дані на 4 рівні частини: Q1 (еквівалентний 25-му процентилю), Q2 (50-й процентиль), Q3 (75-й процентиль даних);

• міжквартильний інтервал - діапазон 50% даних = Q3 - Q1.

• дисперсія– це сума квадратів відхилень кожного спостереження від середнього, поділеного на *n-1*.

Дисперсія генеральної сукупності:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n} \quad (4)$$

Вибіркова дисперсія:

$$S^2 = \frac{\sum_{i=1}^{n}(x_i-\bar{X})^2}{n-1} \quad (5)$$

• стандартне (середньоквадратичне) відхилення– це квадратний корінь з дисперсії.

$$S = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{X})^2}{n-1}} \quad (6)$$

• коефіцієнт варіації - це показник відносної варіації вибірки даних:

$$CV = \frac{S}{\bar{X}} \times 100 \quad (7)$$

CV використовується для порівняння рівня змінності даних, представлених у різних масштабах.

Слід зазначити, що стандартні методики розрахунку показників описової статистики базується на припущенні, що розподіл вибірки є нормальним. Також відомо, що нормальний розподіл можливий лише за нескінченного набору даних, більшість же реальних вибірок даних є близько наближеними до нормального розподілу.

2.3. Висновувальний аналіз

Дослідивши описову статистику, можна переходити до індуктивного (висновувального) статистичного аналізу, який ще має назву «статистичні висновки» (Statistical Inference).

Він необхідний для того, щоб встановити властивості розподілу даних генеральної сукупності даних на підставі аналізу вибірки даних з цієї сукупності, перевірки гіпотез та статистичної значущості отриманих оцінок параметрів.

Статистичні висновки передбачають використання розрахункового значення вибірки (статистики), щоб оцінити або краще зрозуміти невідоме значення генеральної сукупності (параметра) [11].

Статистичні гіпотези розрізняються за видом припущень, що містяться в них, тому існує багато тестів, для статистичного аналізу.

Таблиця 2. – Типи статистичних тестів [11].

Тип тесту	Використання
Параметричні тести	
Кореляційний аналіз	Пошук асоціацій між змінними
Pearson correlation	Перевіряє силу зв’язку між двома неперервними змінними
Spearman correlation	Перевіряє силу зв’язку між двома змінними (не вимагає виконання припущення щодо нормального розподілу даних)
Chi-square	Тестує силу зв’язку між двома категорійними змінними
Аналіз середніх вибіркових значень: *визначення різниці між середніми*	
Paired T-test	Тестує різницю між двома пов’язаними змінними
Independent T-test	Тестує різницю між двома незалежними змінними
ANOVA	Тестує різницю між групами середніх спираючись на дисперсійний аналіз залежної змінної

Продовження таблиці 2.

Тип тесту	Використання
Регресійний аналіз: *оцінює чи зміни в одній змінній роблять внесок у прогнозування іншої змінної*	
Simple regression	Тестує як зміни у незалежній змінній роблять внесок у прогноз змін у залежній змінній
Multiple regression	Тестує як зміни у двох або більше незалежних змінних прогнозують зміни у залежній змінній
Непараметричні тести: *використовуються, коли дані не відповідають припущенням, які вимагають параметричні тести*	
Wilcoxon rank-sum test	Тестує різницю між двома незалежними змінними — бере до уваги величину та напрям різниці
Wilcoxon sign-rank test	Тестує різницю між двома пов’язаними змінними — бере до уваги величину та напрям різниці
Sign test	Тестує чи дві пов’язані змінні є відмінними — ігнорує величину різниці, бере до уваги тільки напрям змін

Статистичні тести допоможуть визначити, чи результати / відносини, які ми спостерігаємо, є реальними або випадковими. Важливо також розуміти, що результат статистичної значущості не означає, що ефект є значущим.

Після заповнення пропущених значень можна перейти до аналізу структури даних та візуалізації. Але ще потрібно правильно розділити параметри, для правильного моделювання. Методом для цього є кластерний аналіз.

Ще потрібно правильно розділити параметри, для правильного моделювання. Методом для цього є кластерний аналіз. Кластерний аналіз - це метод класифікаційного аналізу. Основна ціль методу - розбиття множини досліджуваних об'єктів та ознак на однорідні в певному сенсі класи, або кластери [12].

Це багатовимірний статистичний метод, тому очікується, що вихідні дані можуть бути чималого обсягу. Суттєва перевага кластерного аналізу в тому, що він надає можливість робити розбиття об'єктів не за однією ознакою, а за рядом ознак. До того ж, кластерний аналіз в порівнянні з більшістю математико-статистичних методів не накладає жодних обмежень на вигляд розглянутих об'єктів і дозволяє досліджувати значного розміру вихідні дані.

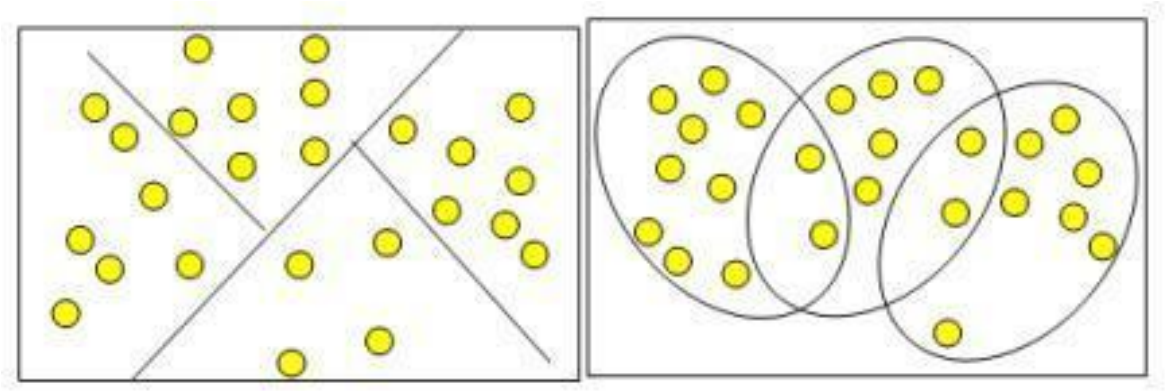

Рис. 5. – Схематичний приклад непересічних і пересічних кластерів [12].

2.4. Регресійний аналіз

До класичних методів статистичного аналізу можна віднести: кореляційний аналіз, регресійний аналіз, перевірка гіпотез, метод порівняння середніх, аналіз часових рядів та інші. Оскільки попередньо було визначено цільову змінну – рівень заробітної плати ІТ-фахівця, то найбільш прийнятним методом моделювання є регресійний аналіз.

Регресійний аналіз, який використовує обраний метод оцінки, залежну змінну і одну або кілька незалежних змінних для створення рівняння, яке оцінює значення залежної змінної. Модель регресії включає вихідні дані, наприклад R^2 і p-значення, за якими можна зрозуміти, наскільки добре модель оцінює залежну змінну [13]. Можна виділити два основних типи регресії – це лінійна та не лінійна.

$$R^2 = 1 - \frac{\sigma^2}{\sigma_y^2} \quad (8)$$

Регресія називається простою, якщо вхідна змінна одна. Однак така модель є занадто грубим наближенням дійсності, і на практиці, як правило, використовують залежності від декількох змінних.

Для лінійної регресії коефіцієнти повинні бути перевірені і задоволені при використанні методу МНК. За МНК коефіцієнти регресії визначаються як унікальні значення, які мінімізують суму квадратів помилок ε в межах вибірки даних, що досліджуються в моделі.

Також однією з необхідних умов для застосування МНК при оцінюванні параметрів моделі є гомоскедастичність, тобто сталість дисперсії випадкових залишків для кожного спостереження. Якщо дисперсія залишків змінюється для кожного спостереження, то це явище називається гетероскедастичність. При наявності гетероскедастичності оцінки параметрів моделі будуть обґрунтованими, незміщеними, але неефективними. Виконання вимоги гомоскедастичності випадкових залишків може бути перевірено візуально, на основі графіка залишків.

В лінійній регресії функція залежить від незалежних параметрів лінійно. З рисунку 6.а видно, що такий вид регресії має вигляд лінії. Якщо представити лінійну регресію у вигляді залежності x від y, то отримаємо формулу [13]:

$$y = \beta_0 + \beta_1 x + \varepsilon, \quad (9)$$

де y та x – деякі змінні, β_0 та β_1 – коефіцієнти моделі, ε – помилка.

Тоді як для не лінійної регресії функція повинна залежати не лише від суми незалежних змінних, а мати варіації множення, піднесення у степінь, та інші. Наприклад:

$$y = \beta_0 x^2 + \beta_1 x + \varepsilon \quad (10)$$

Тоді такий вид регресії буде мати вигляд кривої, як показано на рисунку 6.б.

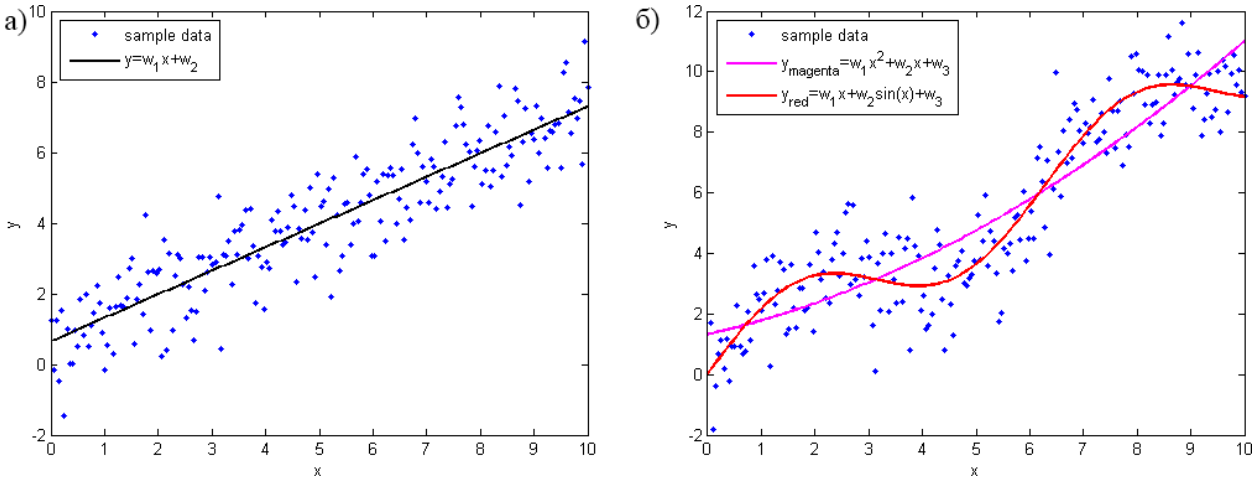

Рис.6. – а) лінійна регресія; б) не лінійна регресія [13].

Саме для мого випадку буде підходящим будувати регресійні моделі і підбирати параметри для неї, щоб використовувати її для прогнозування заробітної плати працівника.

Вибір правильної моделі регресії вимагає:

- Підготовку та очищення даних, обізнаність про їх походження;
- Дослідження описової статистики, щоб зрозуміти загальні закономірності даних та виявити проблеми з якістю даних;

- Застосування відповідних перетворень даних, якщо наявні докази нелінійних залежностей, або шум, який не є нормально розподіленим чи залежить від часу;
- Порівняння різних моделей та їх вдосконалення;
- Перевірки, чи достатньо добре задовольняються припущення певної моделі , або пропонується альтернативна модель;
- Вибору серед прийнятних моделей для заданого рівня надійності;
- Отримання корисних висновків з усього процесу.

Після побудови моделі маємо такі її показники, що необхідні для перевірки на адекватність моделі та для її подальшого опису і аналізу:

- Intercept - якщо у нас модель представлена у вигляді $\beta_0+\beta_1 x+\varepsilon$, то тоді β_0- точка перетину прямої з віссю координат, або intercept.
- R-squared - коефіцієнт детермінації показує наскільки тісним є зв'язок між факторами регресії і залежною змінною, це співвідношення пояснених сум квадратів збурень, до непоясненим. Чим ближче до 1, тим яскравіше виражена залежність.
- Adjusted R-squared - проблема з R^2 в тому, що він з будь-якого зростає з числом факторів, тому високе значення даного коефіцієнта може бути оманливим, коли в моделі присутня безліч чинників. Для того, щоб вилучити з коефіцієнта кореляції ця властивість був придуманий скоригований коефіцієнт детермінації.
- F-statistic - використовується для оцінки значущості моделі регресії в цілому, є співвідношенням зрозумілої дисперсії, до незрозумілою. Якщо модель лінійної регресії побудована вдало, то вона пояснює значну частину дисперсії, залишаючи в знаменнику малу частину. Чим більше значення параметра - тим краще.
- t-value - критерій, заснований на t розподілі Стьюдента. Значення параметра в лінійної регресії вказує на значущість чинника, прийнято вважати, що при $t > 2$ фактор є значимим для моделі.

- p-value - це ймовірність істинності нульової гіпотези, в якій мовиться, що незалежні змінні не пояснюють динаміку залежної змінної. Якщо значення p-value нижче порогового рівня (0.05 або 0.01 для найвимогливіших), то нульова гіпотеза помилкова. Чим нижче - тим краще.

2.5. Дерева рішень

Дерева рішень є одним з найбільш ефективних інструментів інтелектуального аналізу даних, які дозволяють вирішувати завдання класифікації і регресії.

Метод дерева рішень - це графічний метод автоматичного аналізу величезних масивів даних. Мета всього процесу побудови дерева прийняття рішень - створити модель, по якій можна було б класифікувати випадки і вирішувати, які значення може приймати цільова функція, маючи на вході кілька змінних. . У методиці використовується ієрархічна структурна схема, що складається з елементів двох типів - вузли (node) і листки (leaf) [14].

Процес побудови дерева рішень полягає в послідовному, рекурсивному розбитті навчальної множини на підмножини з застосуванням вирішальних правил в вузлах. Процес розбиття триває до тих пір, поки всі вузли в кінці всіх гілок не будуть листям. Оголошення вузла листом може статися природним чином (коли він буде містити єдиний об'єкт, або об'єкти тільки одного класу), або після досягнення деякого умови зупинки, що задається користувачем (наприклад, мінімально допустиму кількість прикладів у вузлі або максимальна глибина дерева).

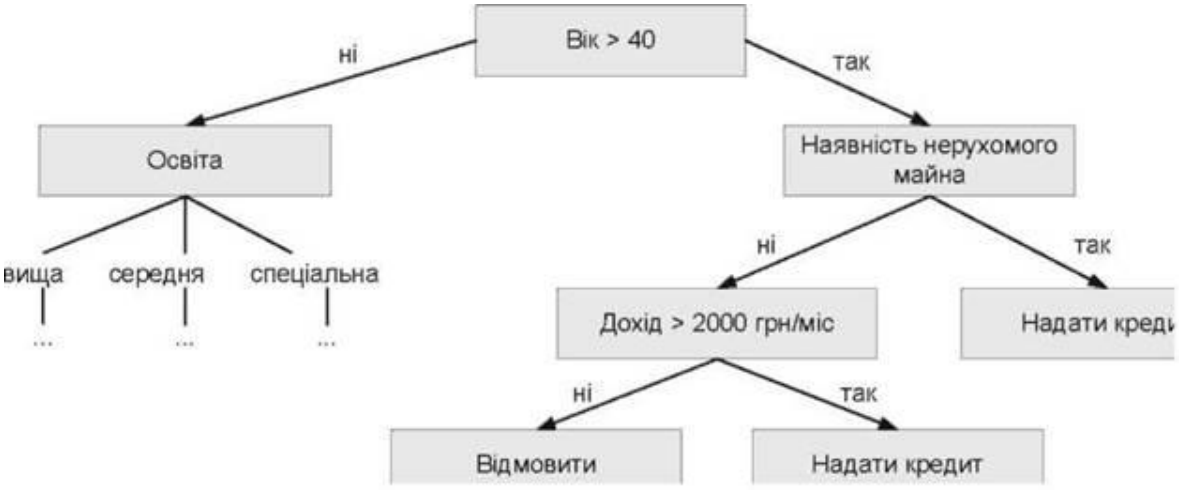

Рис. 7. – Приклад дерева [14].

На схемі верхнє положення займає кінцева мета розв'язання проблеми. Для кожного дерева рішень будується матриця. Зазвичай вводяться коефіцієнти взаємної корисності рішень, вони показують вплив ступеня важливості одних рішень на інші.

Random Forest («випадковий ліс») – алгоритм, що для отриманих даних він створює безліч дерев прийняття рішень і потім усереднює результат їх прогнозів. Важливим моментом тут є елемент випадковості у створенні кожного дерева. Алгоритм побудови дерева прийняття рішень дуже швидкий. І тому нам не складно зробити стільки дерев, скільки буде потрібно.

Алгоритм random forest має по-суті лише один параметр: розмір випадкової підмножини обраного на кожному кроці побудови дерева. Хоча цей алгоритм є доволі простим, він дає дуже хороші результати в реальних задачах, тому «випадковий ліс» є одним з часто використовуваних алгоритмів data mining.

2.6 Градієнтний спуск

Говорячи про мінімізацію деякої функції, завжди варто згадувати метод градієнтного спуску (gradient boosting). Ця техніка використовує ідею про те, що наступна модель буде вчиться на помилках попередньої.

Метод градієнтного спуску використовують у багатьох алгоритмах, де потрібно знайти екстремум функції - нейронні мережі, k-середніх, регресії.

Це простий і ефективний метод мінімізації, заснований на ітеративному обчисленні градієнта функції в точці і подальшому її зміщення в сторону, протилежну градієнту. Кожен такий крок наближає рішення до мінімуму. Суть алгоритму – це процес отримання найменшого значення помилки.

Оскільки градієнт - це вектор, який вказує на найбільше збільшення функції, негативний градієнт - це вектор, який вказує на максимальне зменшення функції. Градієнт функції обчислюється за формулою [15]:

$$\nabla F = \left(\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z}\right), \tag{11}$$

де F=f(x,y,z).

Отже, ми можемо мінімізувати функцію, ітеративно трохи рухаючись в напрямку негативного градієнта -∇F:

$$\bar{x}^{[j+1]} = \bar{x}^{[j]} - \lambda^{[j]} \nabla F\left(\bar{x}^{[j]}\right), \mathrm{j} \geq 0, \tag{12}$$

де $\lambda^{[j]}$ обирається:

- постійно, в цьому випадку метод може розходитися;
- дробовим кроком, тобто довжина кроку в процесі спуску ділиться на деяке число;
- найшвидшим спуском:

$$\lambda^{[j]} = argmin_{\lambda} F\left(\bar{x}^{[j]} - \lambda^{[j]} \nabla F\left(\bar{x}^{[j]}\right)\right) \tag{13}$$

Також нам необхідно задати початкове наближення $\bar{x}^{[0]}$ і точність розрахунку ε. Після цього ми можемо переходити до ітерацій за формулою (11). Обчислюємо допоки виконується умова:

$$\left|\bar{x}^{[j+1]} - \bar{x}^{[j]}\right| > \varepsilon \qquad (14)$$

Якщо умова не виконується, то $\bar{x} = \bar{x}^{[j+1]}$ і зупиняємося.

Існує три типових варіанти градієнтного спуску, які широко використовуються в машинному навчанні [15]:

- пакетний градієнтний спуск: використовує всю партію навчальних даних на кожному кроці. Він розраховує помилку для кожного запису і приймає середнє значення для визначення градієнта;
- стохастичний градієнтний спуск: обирає один екземпляр з навчального набору на кожному кроці і оновлює градієнт тільки на основі цього окремого запису;
- міні-пакетний градієнтний спуск: поєднує в собі концепцію пакетного і стохастичного градієнтного спуску. На кожному етапі алгоритм обчислює градієнт на основі підмножини навчального набору замість повного набору даних або тільки одного запису.

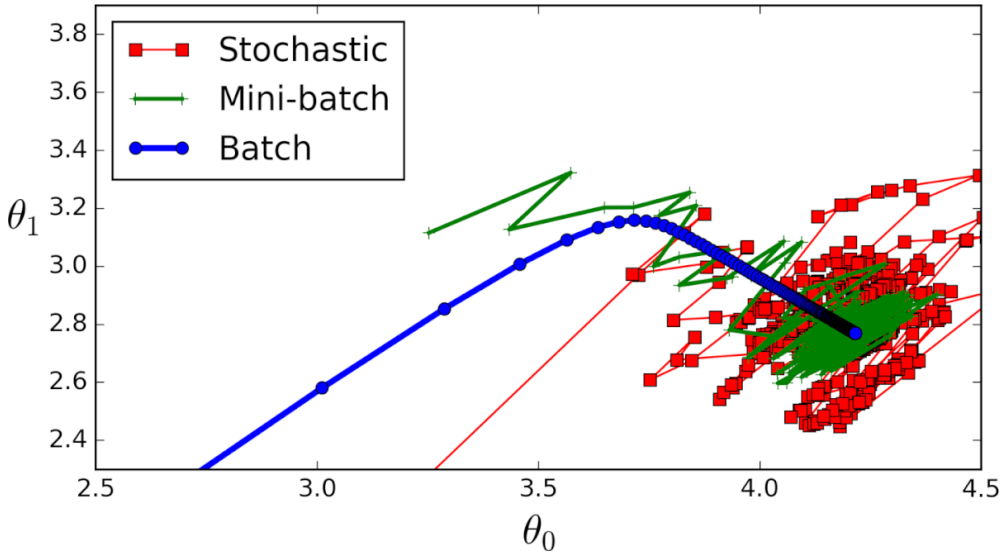

Рис. 8. - Порівняння попадання в локальні мінімуми різними варіантами градієнтного спуску [15].

2.7 Перевірка моделі

Перевірка моделі на адекватність

Важливим кроком у побудові регресійної моделі є її перевірка на адекватність. Під адекватністю вважають ступінь відповідності моделі до реального процесу, для опису якого вона вводиться.

Якість моделі лінійної регресії у цілому можна оцінити, використовуючи такі величини:

Коефіцієнт детермінації R^2 вказує наскільки отримані спостереження підтверджують модель і оцінює наскільки зміни залежної змінної Y пояснюються варіацією незалежних змінних X :

$$R^2 = \frac{TSS-RSS}{TSS} = 1 - \frac{RSS}{TSS}, \quad (15)$$

де $TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$ – загальна сума квадратів для лінійної регресії;

$=$

$\hat{y}_i, \bar{y}$ – фактична, середнє значення Y відповідно.

Якщо статистика R^2, що може приймати значення в інтервалі [0,1], близька до 1, вказує на те, що значна частина варіативності реакції була пояснена лінійною регресією, а якщо статистика R^2 біля 0 то регресія не пояснює більшість мінливості реакції. Рівень адекватності R^2 залежить від застосування.

Якщо R^2 показує силу асоціації між залежною та незалежними змінними, то F – статистика визначає чи ця залежність є статистично значущою

Загальна ***F***-статистика використовується розраховується як:

$$F = \left(\frac{RSS(R)-RSS(F)}{df_R - df_F}\right) \div \left(\frac{RSS(F)}{df_F}\right), \quad (16)$$

де $RSS(R)$ та $RSS(F)$ є сумою квадратів залишків зменшеної та повної моделей; $df_R \ ma \ df_F$ є ступенями свободи $df = n - p$ зменшеної та повної моделей; n – кількість спостережень вибірки, p – кількість змінних.

Ще використовують Т-статистику для перевірки статистичної значущості кожного чинника регресійній моделі, яка обраховує кількість стандартних відхилень $\hat{\beta}_1$ від 0:

$$t = \frac{\widehat{\beta}_1 - 0}{SE(\widehat{\beta}_1)}, \quad (17)$$

де $SE(\hat{\mu})^2 = \frac{\sigma^2}{n}$ - стандартне відхилення вибіркового розподілу, σ^2 є невідомим, але може бути оціненим за вибіркою даних; n – кількість спостережень вибірки.

Також для перевірки адекватності моделі можна зробити наступні тести та графіки:

- Тест на автокореляцію залишків;
- Тест на нормальність залишків;
- Графік розподілу залишків. Можна використати гістограму залишків, додаючи нормально розподілену криву з середнім значенням, рівним нулю і тим же стандартним відхиленням, що і залишки, або побудувати Q-Q графіку залишків, який називається графіком нормальної ймовірності;
- Графік Залишків проти залишків. Використовується для перевірки кореляції залишків;
- Діаграму розсіювання та додати лінію регресії , оцінену за методом найменших квадратів, щоб побачити, чи відповідають наближені встановлені значення фактичним даним.

Перевірка точності

Важливо також перевірити точність побудованої моделі, щоб отримані дані відповідали реальності.

Є декілька показників похибок, популярними серед них є:

- MAE - середня абсолютна похибка.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \quad (18)$$

де y_i- фактичні значення, $\hat{y}_i$- прогнозні значення, n – кількість спостережень вибірки.

- MAPE (середня абсолютна похибка у відсотках)

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\%, \quad (19)$$

де y_i- фактичні значення, $\hat{y}_i$- прогнозні значення, n – кількість спостережень вибірки.

MSE - середньоквадратична похибка, підкреслює великі похибки за рахунок зведення кожної похибки в квадрат.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \quad (20)$$

де y_i- фактичні значення, $\hat{y}_i$- прогнозні значення, n – кількість спостережень вибірки.

- RMSE – це корінь від квадрата похибки. Її легко інтерпретувати, оскільки він має ті ж одиниці, що і вихідні значення. Також оперує меншими величинами за абсолютним значенням, що може бути корисно для обчислення на комп'ютері.

$$RMSE = \sqrt{MSE} \quad (21)$$

3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1. Опис даних

В роботі були використані дані які зібрані з опитування IT спеціалістів за 2019 рік.

Через проблему того, що більшість компаній не надає повної інформації про працівників у відкритому доступі, потрібно обмежуватися статистичними даними з сайту ukrstat.gov.ua та дані з опитування респондентів. Однією з проблем даного опитування, не повна інформація за деякими параметрами, що ускладнює обробку та аналіз даних.

У наборі даних надається інформація про працівника, місце роботи, навички та інше. Завдяки цим даним є можливість аналізу та прогнозування заробітної плати працівника. Дані розділяються на декілька типів: кількісні та категорійні.

Переглянувши всі параметри, отримаємо таку картину:

- City – місто;
- Income – заробітна плата у місяць (дол.США на рік);
- Income_changes – зміни у заробітній платі за останні 12 місяців (дол.США;)
- Position – посада;
- exp – досвід (кількість років);
- current_exp – досвід на даному місці праці (кількість років);
- Programming_language - мова програмування;
- Age – вік (кількість років);
- Sex – стать;
- Education – освіта;
- University – університет;
- Student – чи є студентом на даний момент;
- English_lvl – рівень англійської;
- Compaty_size – розмір компанії (кількість людей);
- Company_type – тип компанії;

- Subject_area – область праці;

Основним показником в даних є заробітна плата – буде використовуватись в якості головної компоненти в розрахунку тенденції заробітку працівника. Також в основні компоненти можна віднести: досвід, мова програмування, освіта, рівень англійської, тип компанії та вік.

Перевіримо до яких типів даних відносяться наші дані за допомогою функції summary(data) :

```
'data.frame':   4344 obs. of  17 variables:
 $ City                : Factor w/ 24 levels "Винница","Днепр",..: 10 7 12 17 22 17 10 17 10 2 ...
 $ Income              : int  12000 12000 12000 12000 12000 11400 11200 11000 10800 10000 ...
 $ Income_changes      : int  2000 0 0 2000 0 1000 1200 5000 2800 2000 ...
 $ Position            : Factor w/ 34 levels "BI Engineer",..: 31 7 7 24 27 8 31 26 31 7 ...
 $ exp                 : num  10 3 5 7 3 9 10 8 6 3 ...
 $ current_exp         : num  3 0.25 2 7 2 3 1 0.25 4 2 ...
 $ Programming_language: Factor w/ 24 levels "1C","ABAP","APL",..: 5 NA NA NA 14 NA 5 13 5 NA ...
 $ Specialisation      : Factor w/ 4 levels "Automation QA",..: NA NA NA NA NA NA NA NA NA NA ...
 $ Age                 : int  28 29 30 30 31 28 30 30 27 26 ...
 $ Sex                 : Factor w/ 2 levels "женский","мужской": 2 1 1 2 1 1 2 2 2 1 ...
 $ Education           : Factor w/ 7 levels "Высшее","Два высших",..: 7 1 1 1 2 1 1 1 5 1 ...
 $ University          : Factor w/ 37 levels "ВНТУ","ВНУ им. Даля",..: 8 13 NA 8 8 21 15 22 16 4 ...
 $ Student             : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ English_lvl         : Factor w/ 5 levels "выше среднего",..: 1 1 5 3 3 3 3 3 1 4 ...
 $ Company_size        : Factor w/ 6 levels "до 10 человек",..: 6 4 2 1 1 1 6 1 6 4 ...
 $ Compaty_type        : Factor w/ 5 levels "Аутсорсинговая",..: 1 1 4 4 3 3 1 4 1 4 ...
 $ Subject_area        : Factor w/ 202 levels "Android","Android,Desktop Applications,Embedded",..:
```

Рис. 9. – Загальна інформація про дані.

Як бачимо дані поділяються на два типи чисельні (int та num) і категорійні (Factor).

3.2. Очищення та підготовка даних

Для кількісних даних знайдемо центральні тенденції, де можна побачити основні міри, такі як мінімальне та максимальне значення, перший та третій квантилі, медіану та середнє, а також дисперсію для основних показників:

```
     Income       Income_changes         exp           current_exp        Age       
 Min.   : 2000   Min.   :-5000.0   Min.   : 0.000   Min.   : 0.000   Min.   :18.0  
 1st Qu.: 2500   1st Qu.:  200.0   1st Qu.: 4.000   1st Qu.: 0.500   1st Qu.:26.0  
 Median : 3100   Median :  500.0   Median : 5.000   Median : 2.000   Median :30.0  
 Mean   : 3423   Mean   :  607.1   Mean   : 5.943   Mean   : 2.146   Mean   :30.1  
 3rd Qu.: 4000   3rd Qu.: 1000.0   3rd Qu.: 8.000   3rd Qu.: 3.000   3rd Qu.:33.0  
 Max.   :12000   Max.   : 5000.0   Max.   :10.000   Max.   :10.000   Max.   :59.0  
```

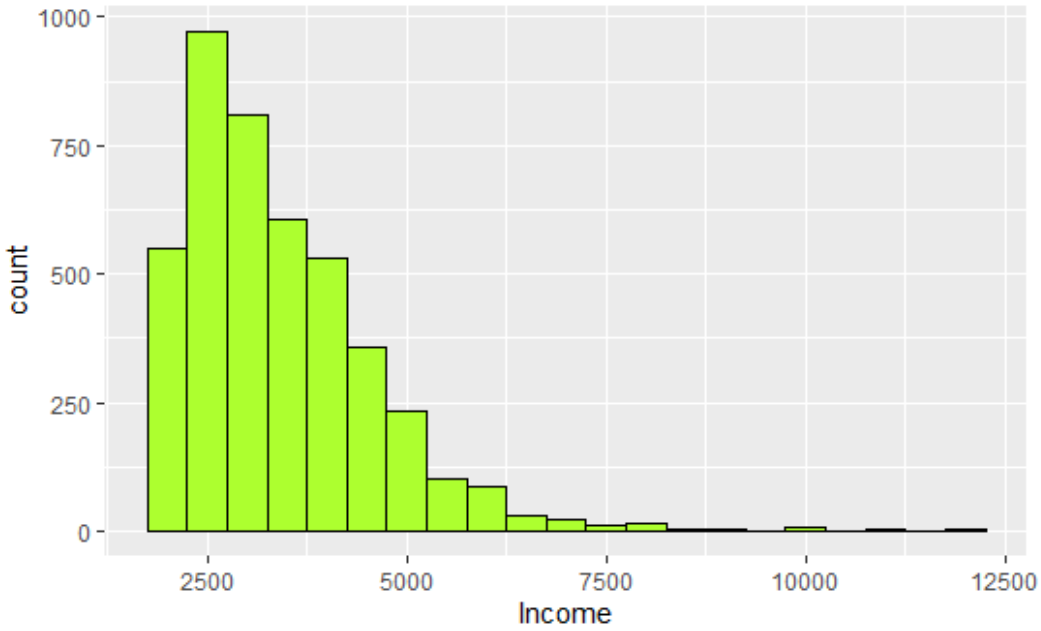

Рис. 10. – Гістограма розподілу змінної Income.

Також для кількісних даних перевіримо гіпотезу про нормальність розподілу. Для цього скористаємось Шапіро-Уілко тестом [16]:

Таблиця 3. – Результати Шапіро-Уілко тесту.

	Income	Age	Exp
W-stat	0.866	0.963	0.91
p-value	< 2.2e-16	< 2.2e-16	< 2.2e-16

Згідно з результатами таблиці 3, у всіх трьох тестах p-value < 0.05, де число 0.05 – є довірчим інтервалом, і якщо значення не попадає в нього

нульову гіпотезу відхиляємо, тобто розподіл змінних не є нормальним. Якщо подивитись на графіки рисунку 10, то побачимо, що розподіл дійсно далекий від нормального і є багато аномалій. Спробуємо привести його до більш схожого на нормальний. Для цього існує кілька способів: логарифмування, піднесення до квадрату, метод арксинуса, метод Box-Cox. Найбільш популярним методом є логарифмування, тому спробуємо прологарифмувати дані.

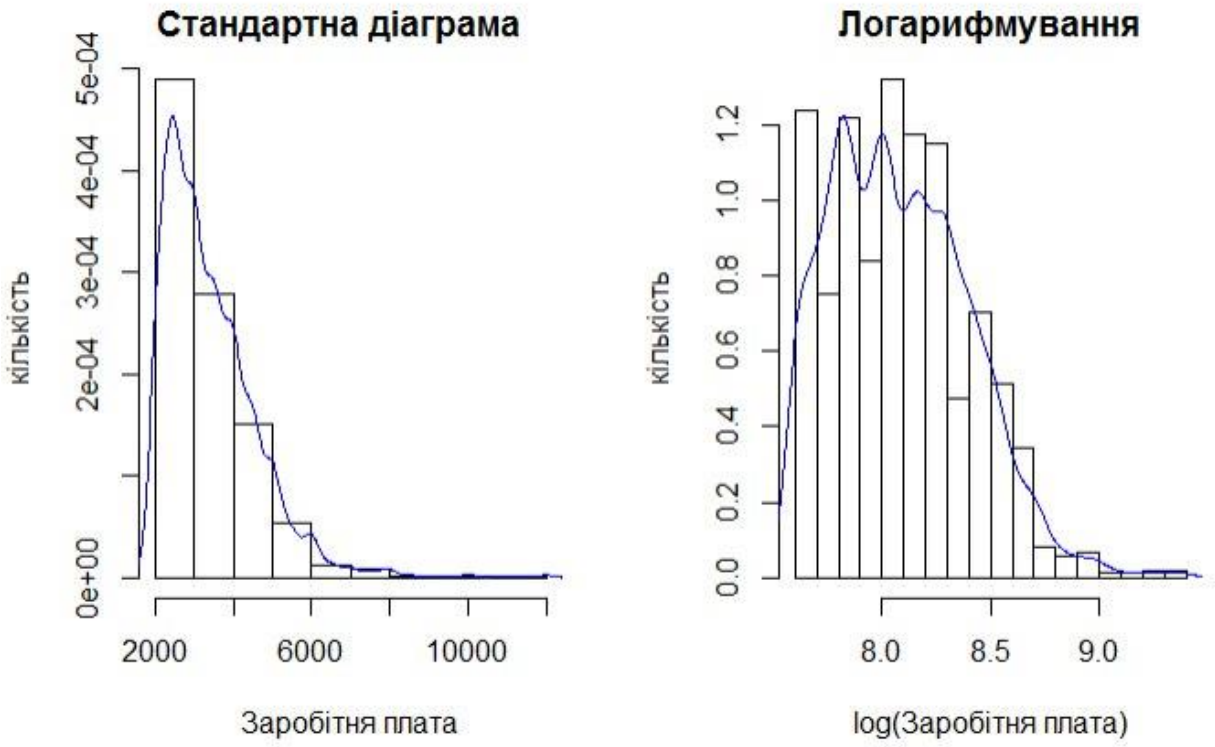

Рис. 11. – Гістограми змінної Income до і після логарифмування.

Як видно з рисунку 11 логарифмування показника заробітної плати допомогло у нашому випадку ,тому для подальших розрахунків обираємо саме його також спробуємо прибрати аномалії, щоб привести до ще більшого схожого на нормального розподілу змінної.

Для зручності візуалізації і розуміння даних, згрупуємо основні кількісні параметри даних, отримаємо три нових колонки даних, які можна віднести до категорійних змінних. Маємо таку картину:

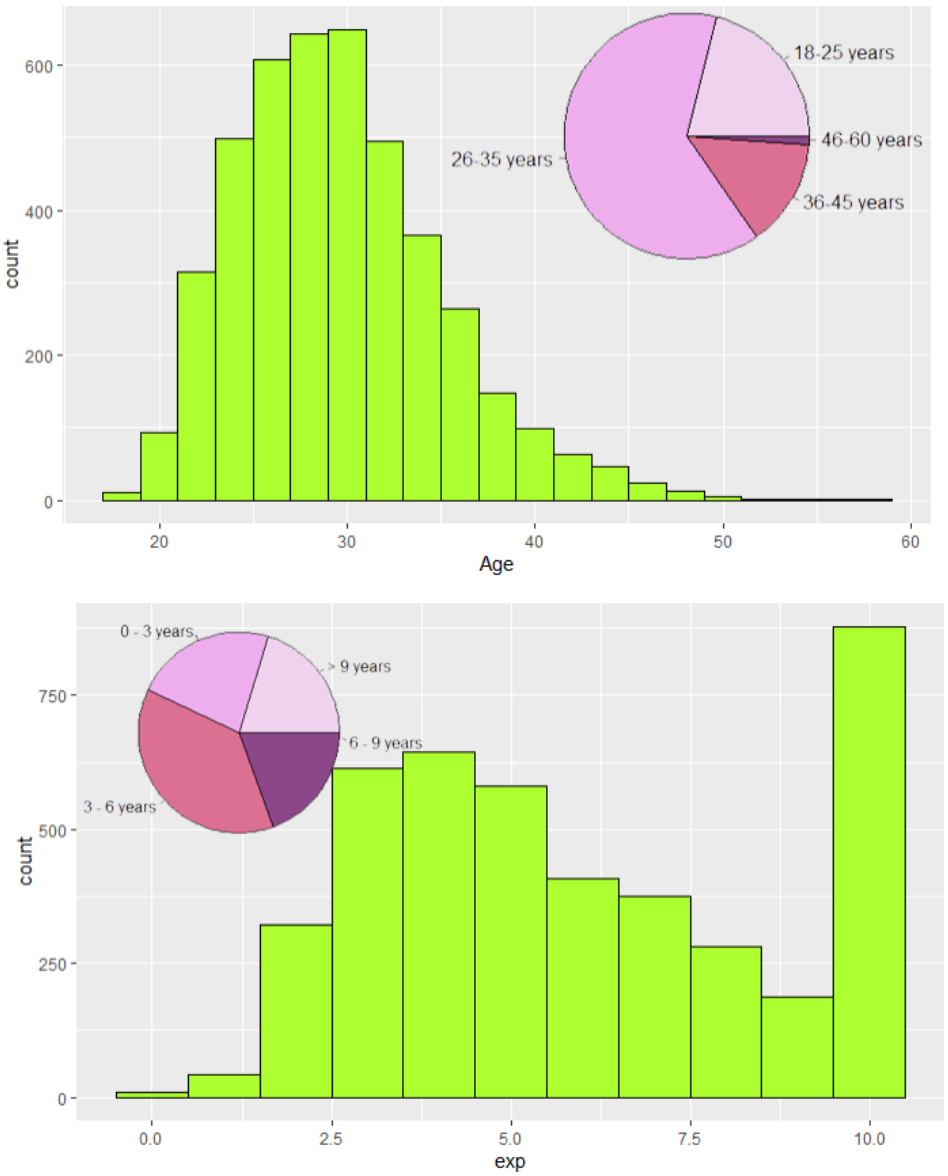

Рис. 12. – Гістограми розподілу незалежної змінних Age та Exp.

Перейдемо до категорійних даних. Для них потрібно побудувати частоту та імовірнісні таблиці. Для початку зробимо огляд цих даних:

Можна побачити скільки значень містить той або інший фактор, також можна побачити, що присутній елемент NA, що вказує на пропущені значення. Розглянемо змінну Programming_language:

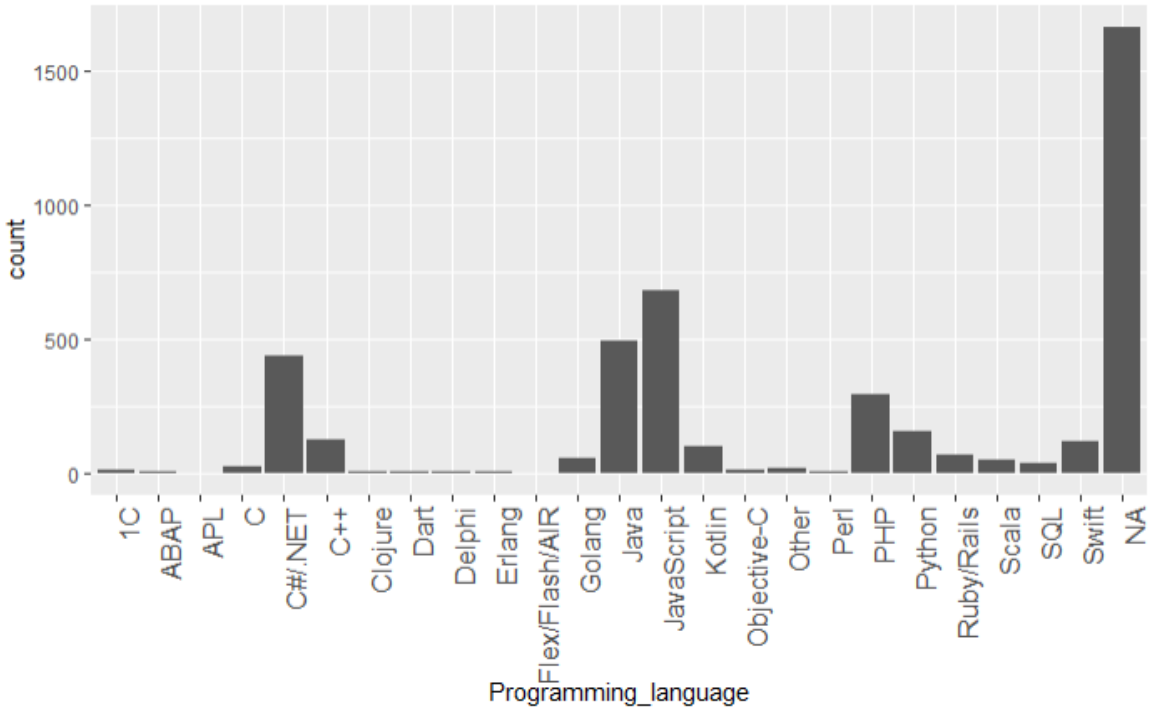

Рис. 13. – Гістограма розподілу змінної Programming_language.

З рисунку 13 бачимо значну частку займають пропущені значення, це може бути пов’язано з тим, що людина знає багато мов програмування (описова статистика та код для побудови графікі у додатку Б). Тож спробуємо прибрати пропуски та застосуємо метод MICE.

Для цього скористаємось вбудованою функцією mice().

Метод MICE використовує наступні формули для розрахунків (ітеративно):

$$\theta_1^{(t)} \sim P\left(\theta_1 | Y_1^{obs}, Y_2^{(t-1)}, \dots, Y_p^{(t-1)}\right)$$
$$Y_1^{(t)} \sim P\left(Y_1 | Y_1^{Obs}, Y_2^{(t-1)}, \dots, Y_p^{(t-1)}, \theta_1^{(t)}\right)$$
$$\dots$$
$$\theta_P^{(t)} \sim P\left(\theta_P | Y_P^{obs}, Y_1^{(t-1)}, \dots, Y_{p-1}^{(t-1)}\right)$$
$$Y_P^{(t)} \sim P\left(Y_P | Y_P^{Obs}, Y_1^{(t-1)}, \dots, Y_p^{(t-1)}, \theta_P^{(t)}\right) \quad (22)$$

1. за допомогою функції mice () моделюється набір масивів для тих значень, яких немає (за замовчуванням є п'ять таких масивів);

2. з підтримкою функцій з () для кожного з отриманих на попередньому етапі масивів, сумісного з основним масивом, застосовується необхідний статистичний метод (наприклад, лінійна регресія);

3. за допомогою функції pool () об'єднуються результати, отримані для кожного з масивів на попередньому етапі.

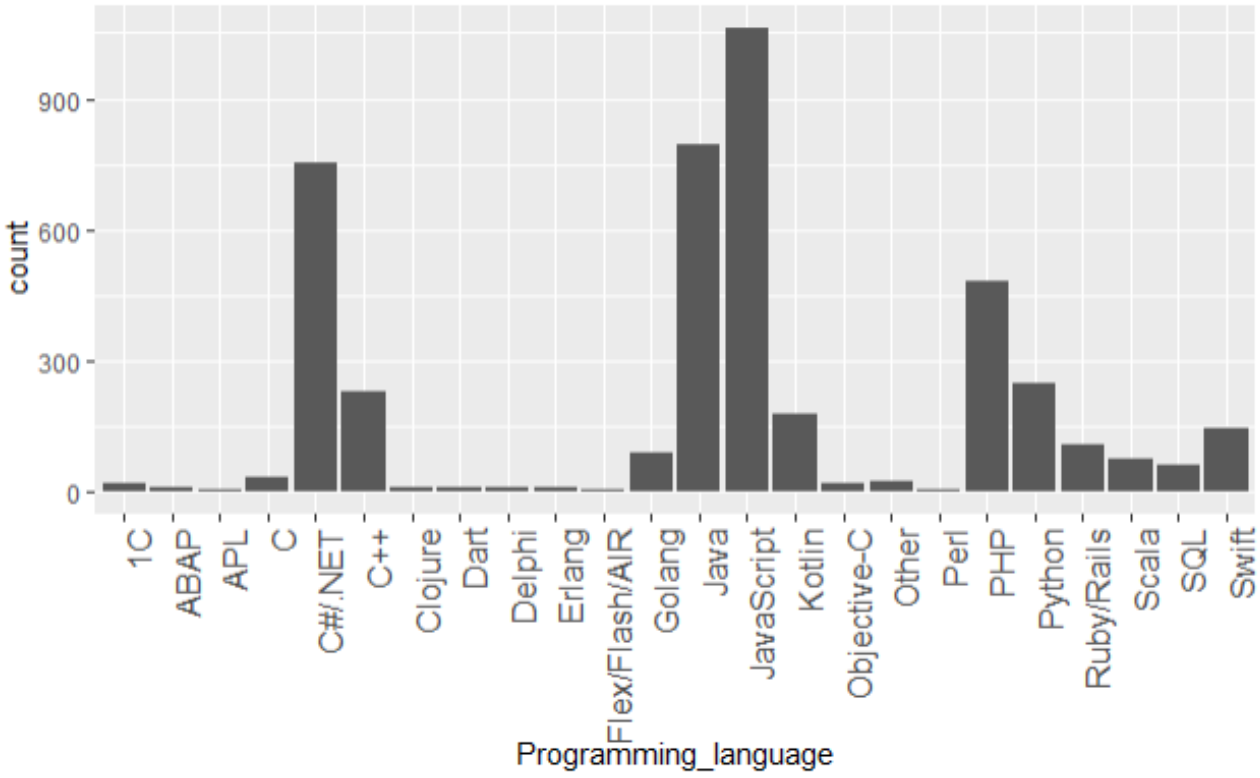

Рис.14. – Гістограма розподілу змінної Programming_language після заповнення пропущених даних.

Оскільки ми замінили невідомі значення, то деякі параметри дуже сильно змінилися, надалі подивимось наскільки це вплине на остаточний результат. В подальшому будемо працювати над обома варіантами, з пропусками і без пропусків. Потім перевіримо точність і зробимо висновок про доцільність додавання даних.

Ми звільнилися від пропусків, але ми маємо велику кількість мов з низькими показниками, тому для легшого моделювання та подальшої роботи згрупуємо усі мови програмування на декілька категорій: C-language, Java, Web, Python, Other.

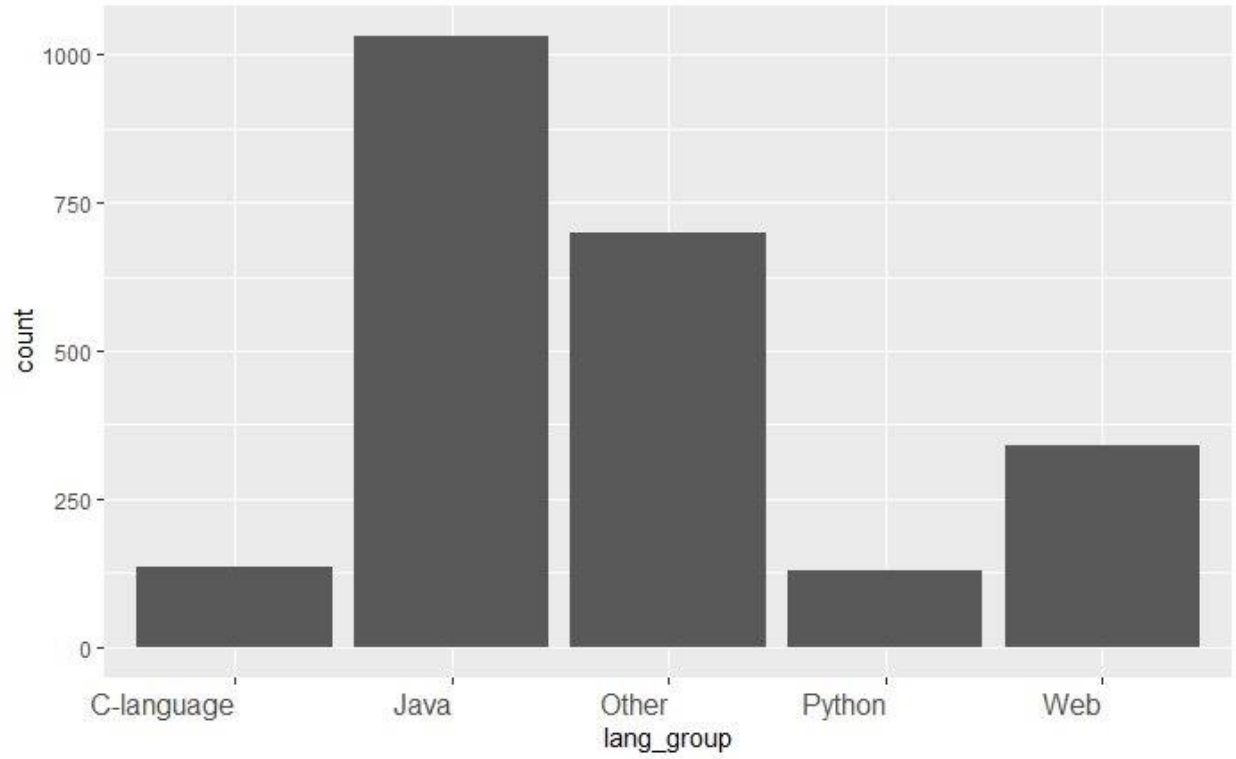

Рис. 15. – Гістограма розподілу змінної Programming_language після розподілу на категорії.

Те саме зробимо для інших категорійних змінних.

Таблиця 4. – Розподіл змінних на категорії.

Пояснювальна зміна											
Education		Age		Income		Exp		Position		Programming_lang	
Высшее	3808	18-25 years	916	10000$-1200	9	> 9 years	878	Engineer	3002	C-language	154
Специальное	536	26-35 years	2760	2000-4000	3343	0 - 3 years	988	Manager	908	Java	1171
		36-45 years	621	4000-6000	887	3 - 6 years	1633	Other	434	Other	792
		46-60 years	47	6000-8000	89	6 - 9 years	845			Python	156
				8000$-10000	16					Web	408

В подальшому змінні, що розбили на категорії будемо використовувати під іншими назвами: Position – pos_group, Programming_language – lang_group, Education – edu_group.

3.3. Статистичні тести

Через відсутність даних по soft skills, будемо надалі розглядати обмежену модель. Для вибірки, яка представлена в цій роботі, маємо навички робітника: мова програмування, позицію яку він займає, рівень англійської мови та рівень освіти.

Розглянемо влив навичок на заробітню плату. Для цього скористаємося Anova test для кількісних даних та Chi-squared test для категорійних.

```
             Df    Sum Sq   Mean Sq F value   Pr(>F)
exp           1 8.272e+08 827244887 886.016  < 2e-16 ***
Age           1 8.881e+06   8880792   9.512  0.00207 **
```

З Anova тесту бачимо, що змінні Age та exp є впливовими.

```
        Pearson's Chi-squared test

data:  df$Income and df$English_lvl
X-squared = 756.63, df = 732, p-value = 0.2566

□□□□□□□□□□□□□□□ □□ □□□□□□ □□-□□□□□□□ □□□□□ □□□□
□□□□□□□□□□□□□
        Pearson's Chi-squared test

data:  df$Income and df$pos_group
X-squared = 402.11, df = 183, p-value < 2.2e-16

□□□□□□□□□□□□□□□ □□ □□□□□□ □□-□□□□□□□ □□□□□ □□□□
□□□□□□□□□□□□□
        Pearson's Chi-squared test

data:  df$Income and df$edu_group
X-squared = 199.3, df = 183, p-value = 0.1942

□□□□□□□□□□□□□□□ □□ □□□□□□ □□-□□□□□□□ □□□□□ □□□□
□□□□□□□□□□□□□
        Pearson's Chi-squared test

data:  df$Income and df$lang_group
X-squared = 725.55, df = 732, p-value = 0.5603
```

Рис. 16. - Chi-squared test для змінних English_lvl, pos_group, edu_group, lang_group.

З рисунку 16 бачимо, що змінні English_lvl, edu_group, lang_group не сильно впливають на заробітню плату, на відміну від змінної pos_group.

Також побудуємо кореляційну матрицю між всіма показниками, щоб оцінити їх влив один на одного. Оскільки ми маємо як кількісні так і категорійні показники, то для побудови кореляційної матриці R використовує:

- для кількісних – Pearson тест, що обчислюється за формулою:

$$X_n^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}, \tag{23}$$

де O - спостережувані частоти (Observed), E – очікувані частоти (Expected).

- для категорійних – Chi-squared тест, також використовує формулу 23

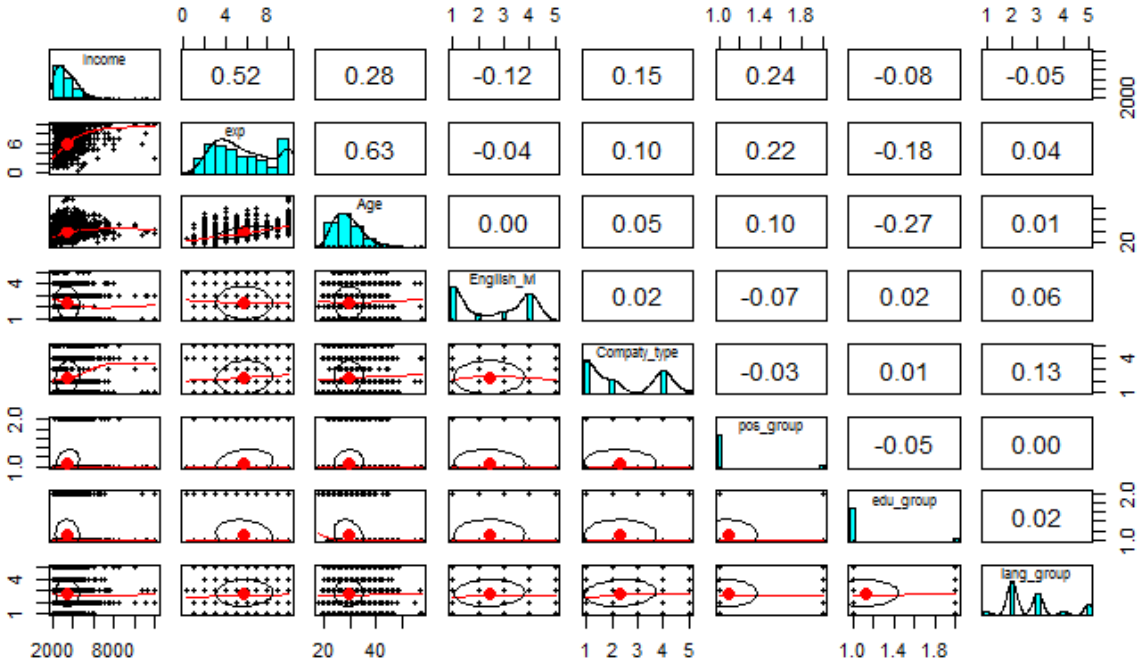

Рис. 17. – Кореляційна матриця.

Як видно з рисунку 17, найбільшу кореляцію мають між заробітною платою саме стаж, вік та позиція в фірмі. З чого можна зробити висновок, що вони мають найбільший вплив на показник заробітної плати. Також маємо велике значення кореляції для змінної exp, що вказує на мультиколініарність, тому при побудові моделі маємо це врахувати.

3.4. Побудова моделей

3.4.1 Регресійна модель

Вибір оптимальної моделі

У попередніх кроках ми визначили, що дані залежної змінної будемо використовувати прологарифмованими, а дані незалежних категорійних змінних згрупованими по категоріям.

Тепер розділимо масив даних на тренувальну і тестову вибірки. Для цього скористаємось розділенням даних, яке пропонує мова програмування R.

```
intrain<- createDataPartition(df$Income,p=0.8,list=FALSE)
set.seed(200)
training<- df[intrain,]
testing<- df[-intrain,]
```

Після цього маємо два набори даних, які поділили основний масив даних 20 на 80. Побудуємо декілька регресійних моделей та порівняємо. Скориставшись функцією lm() побудовані моделі (Таблиця 5).

Таблиця 5. – Регресійні моделі.

модель	залежна зміна	незалежні змінні
lmI1	log(Income)	Position, exp, Age, English_lvl, Company_type, edu_group
lmI2	log(Income)	Position, exp, Age, English_lvl, Company_type, edu_group, log(Age), Age*exp (ефект взаємодії)
lmI3	log(Income)	Position, exp, Age, English_lvl, Company_type, edu_group, log(Age), log(exp)
lmI4	log(Income)	Position, exp, Age, English_lvl, Company_type, edu_group, exp:Age (ефект взаємодії)
lmI5	log(Income)	Position, exp, Age, English_lvl, Company_type, edu_group, log(Age), I(exp^2)(для специфікації), exp:Age

Маючи 5 різних моделей оцінимо коефіцієнт детермінації для кожної ж них, для цього проведемо тест ANOVA, який покаже чи сильно відрізняються моделі одна від одної. ANOVA тест обраховує за формулою:

$$F = \frac{MSTR}{MSE} \quad (24)$$

Тут $MSTR = \frac{\sum_{i=1}^{k} n_i(y_i - \bar{y})}{k-1}$ – середньоквадратичне відхилення;

$MSE = \frac{\sum_{i=1}^{k} \sum_{j=1}^{n} (y_{ij} - \bar{y}_i)}{n-k}$ - середньоквадратична похибка; y_i - спостереження, n - загальна кількість спостережень, k – ступінь свободи. Перевіримо попарно моделі:

anova(lmI1, lmI2)

```
Model 1: log(Income) ~ Position + exp + Age + English_lvl + Compaty_type +
    edu_group
Model 2: log(Income) ~ Position + exp + Age + English_lvl + Compaty_type +
    edu_group + log(Age) + Age * exp
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1   2665 115.84
2   2663 114.32  2    1.5264 17.778 2.139e-08 ***
```

anova(lmI2, lmI3)

```
Model 1: log(Income) ~ Position + exp + Age + English_lvl + Compaty_type +
    edu_group + log(Age) + Age * exp
Model 2: log(Income) ~ Position + exp + Age + English_lvl + Compaty_type +
    edu_group + log(Age) + log(exp)
  Res.Df    RSS Df Sum of Sq F Pr(>F)
1   2663 114.32
2   2663 114.75  0  -0.43032
```

anova(lmI3, lmI4)

```
Model 1: log(Income) ~ Position + exp + Age + English_lvl + Compaty_type +
    edu_group + log(Age) + log(exp)
Model 2: log(Income) ~ Position + exp + Age + English_lvl + Compaty_type +
    edu_group + exp:Age
  Res.Df    RSS Df Sum of Sq F Pr(>F)
1   2663 114.75
2   2664 114.35 -1   0.39634
```

anova(lmI4, lmI5)

```
Model 1: log(Income) ~ exp + Age + English_lvl + Company_size + exp_group +
    pos_group + lang_group + exp:Age
Model 2: log(Income) ~ exp + Age + English_lvl + Company_size + exp_group +
    pos_group + lang_group + log(Age) + I(exp^2) + exp:Age
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1   2314 125.13
2   2312 124.85  2    0.2761 2.5563 0.07781 .
```

Як видно з тестів anova, при перевірці між моделями lmI2 та lmI3 коефіцієнт p показав, що ці моделі не мають значних відмінностей, всі інші порівняння також показали, що відмінностей немає, отже можна вибрати одну з цих моделей. Для свого прикладу обираю модель під номером 5:

$$\log(\text{Income}) \sim \text{Position} + \text{exp} + \text{Age} + \text{English\_lvl} + \text{Compaty\_type} + \text{edu\_group} + \log(\text{Age}) + I(\text{exp}^2) + \text{exp}:\text{Age} \quad (25)$$

```
Call:
lm(formula = log(Income) ~ . + log(Age) + I(exp^2) + exp:Age,
    data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.66457 -0.15559 -0.00666  0.13942  1.36685

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    8.0276148  0.7534263  10.655  < 2e-16 ***
exp                            0.1733338  0.0153523  11.290  < 2e-16 ***
Age                            0.0199031  0.0129820   1.533 0.125382
English_lvlниже среднего      -0.1909692  0.0192109  -9.941  < 2e-16 ***
English_lvlпродвинутый         0.0550155  0.0159874   3.441 0.000589 ***
English_lvlсредний            -0.0808257  0.0108241  -7.467 1.15e-13 ***
English_lvlэлементарный       -0.1171258  0.0511384  -2.290 0.022090 *
Compaty_typeАутстаффинговая    0.0617595  0.0132236   4.670 3.18e-06 ***
Compaty_typeДругая             0.0429106  0.0403956   1.062 0.288228
Compaty_typeПродуктовая        0.0779998  0.0113925   6.847 9.66e-12 ***
Compaty_typeСтартап            0.1112293  0.0245034   4.539 5.93e-06 ***
pos_groupManager               0.1271353  0.0177084   7.179 9.39e-13 ***
edu_groupСпециальное           0.0080389  0.0161674   0.497 0.619072
lang_groupJava                 0.0457624  0.0214598   2.132 0.033073 *
lang_groupOther                0.0256141  0.0218350   1.173 0.240888
lang_groupPython               0.0504238  0.0285539   1.766 0.077542 .
lang_groupWeb                 -0.0415705  0.0238568  -1.742 0.081554 .
log(Age)                      -0.3136626  0.3345719  -0.938 0.348597
I(exp^2)                      -0.0048430  0.0009750  -4.967 7.28e-07 ***
exp:Age                       -0.0017396  0.0005326  -3.267 0.001104 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2302 on 2314 degrees of freedom
Multiple R-squared:  0.4286,    Adjusted R-squared:  0.4239
F-statistic: 91.34 on 19 and 2314 DF,  p-value: < 2.2e-16
```

Рис. 18. – Загальний вигляд оптимальної моделі.

З отриманої моделі можна виділити коефіцієнти, які представлені в формулі (6), модель з самими значущими параметрами має вигляд:

$$\begin{aligned}\text{Log(Income)} = {} & 8.02 + 0.17*\text{exp} + 0.02*\text{Age} - 0.19*\text{English\_lvlниже среднего} + \\ & 0.05*\text{English\_lvlпродвинутый} - 0.08*\text{English\_lvlсредний} - \\ & 0.11*\text{English\_lvlэлементарный} + 0.06*\text{Compaty\_typeАутстаффинговая} - \\ & 0.04*\text{Compaty\_typeДругая} + 0.07*\text{Compaty\_typeПродуктовая} + \\ & 0.11*\text{Compaty\_typeСтартап} + 0.12*\text{pos\_groupManager} + \\ & 0.008*\text{edu\_groupСпециальное} + 0.04*\text{lang\_groupJava} + \\ & 0.02*\text{lang\_groupOther} + 0.0025*\text{lang\_groupPython} - 0.005*\text{lang\_groupWeb} \\ & - 0.31*\log(\text{Age}) - 0.005*I(\text{exp}^2) - 0.002*\text{exp}:\text{Age}\end{aligned} \quad (26)$$

При цьому коефіцієнт Intercept і буде точкою перетину осі координат. Деякі з коефіцієнтів мають від’ємні значення, що вказує на напрямок.

Далі потрібно перевірити модель на виконання умов МНК. Для початку перевіримо помилки на нормальність, для цього побудуємо графік Q-Q та проаналізувати його. Графік для аналізу показаний на рисунку 19, з нього видно, що залишки розподілені нормально, але є незначні викиди.

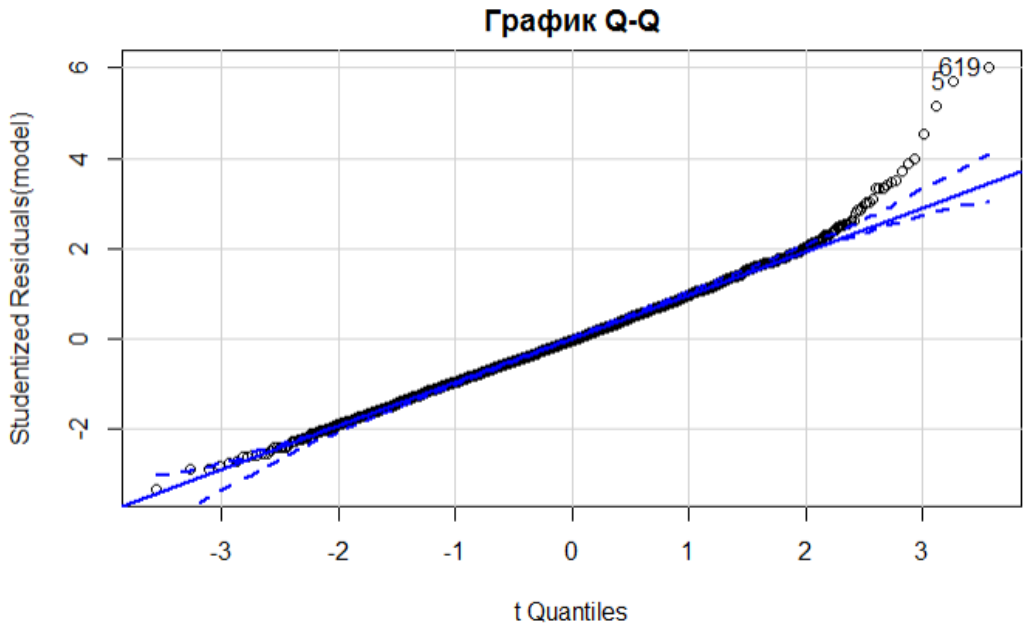

Рис. 19. – Графік для аналізу залишків на нормальність.

Наступним показником є незалежність помилок. Перевіримо його тестом Дарбіна-Уотсона. Результат якого показує, що автокореляція відсутня.

```
lag Autocorrelation D-W Statistic p-value
  1       0.6405151     0.7109586       0
Alternative hypothesis: rho != 0
```

Далі перевіримо модель на гомоскедастичність. Побудувавши графік spreadLevelPlot проаналізуємо наскільки розкидані залишки. На рисунку 20 видно, що залишки, в більшій кількості, лежать в проміжку не більше $1*e^{-2}$, це показує, що залишки не гомоскедастичні.

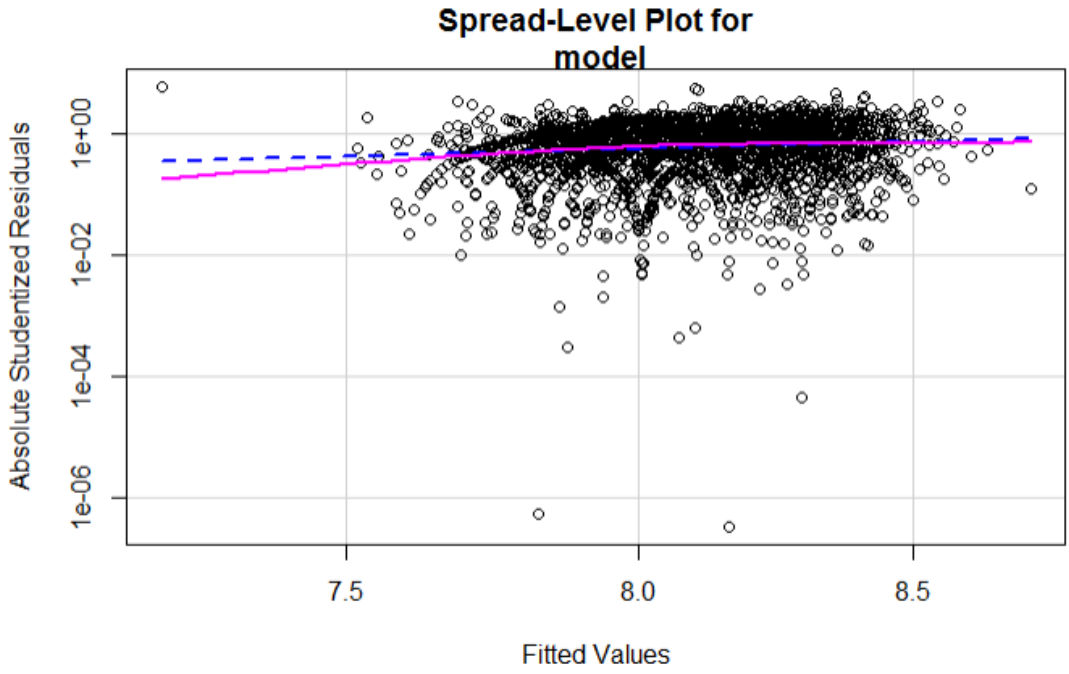

Рис. 20. – Графік розкиду залишків.

Також перевіримо модель на мультиколінеарність пояснювальних змінних. Для цього скористаюсь формулою variance inflation factor (VIF):

$$\text{VIF} = \frac{1}{1-R^2} \qquad (27)$$

Отримаємо значення VIF = 1.77, для границі в 2. Тому з впевненістю можна стверджувати, що мультиколінеарність відсутня.

Отже, підсумувавши все це, можна сказати, що умови МНК підтверджені і використання моделі є доцільним. Наступним кроком, є тестування моделі на нових, в нашому випадку, тестових даних, які ми виділили спочатку. За допомогою моделі, розрахованої вище, знайдемо значення заробітної плати для тестових значень. Отримавши значення і проекспонувавши його, отримаємо значення, та порівняємо їх з тим, що було в тестових значеннях.

Оцінка для прогнозованих значень:

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
2050    2777    3205    3255    3682    4691
```

Оцінка для реальних значень:

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 2000    2500    3200    3418    4000    7675
```

Як видно, значення середнього та медіани близькі, також можна відмітити схожі показники на першому та третьому квантилі. Та точність логістичної регресії: 0.739784946, що свідчить про гарну точність. Але через те, то деякі співробітники отримують велику кількість заробітної плати, такі як 6000 і більше, це можна було побачити на гістограмі рисунку 10. Одним з варіантів, буде не враховувати таку заробітну плату при моделюванні, через те, що при влаштуванні на нову роботу, початкова заробітна плата не буде такою аномально високою. Спробуємо побудувати модель виключивши робітників із заробітною платою більше 6000.

Візьмемо ту саму формулу для моделі, але в даних змінної Income відкинемо аномалії. Отримали модель:

```
Call:
lm(formula = log(Income) ~ . + log(Age) + I(exp^2) + exp:Age,
    data = datamin)

Residuals:
     Min       1Q   Median       3Q      Max
-0.59657 -0.14603  0.00265  0.14146  0.86424

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  8.0305267  0.6896041  11.645  < 2e-16 ***
exp                          0.1639902  0.0141316  11.604  < 2e-16 ***
Age                          0.0171977  0.0118915   1.446   0.1483
English_lvlниже среднего    -0.1727012  0.0173929  -9.929  < 2e-16 ***
English_lvlпродвинутый       0.0216569  0.0150526   1.439   0.1504
English_lvlсредний          -0.0700336  0.0098649  -7.099 1.68e-12 ***
English_lvlэлементарный     -0.0878723  0.0460587  -1.908   0.0565 .
Compaty_typeАутстаффинговая  0.0609428  0.0120709   5.049 4.81e-07 ***
Compaty_typeДругая          -0.0518994  0.0386562  -1.343   0.1795
Compaty_typeПродуктовая      0.0619502  0.0104532   5.926 3.58e-09 ***
Compaty_typeСтартап          0.0906298  0.0230537   3.931 8.71e-05 ***
pos_groupManager             0.1303351  0.0166275   7.839 7.00e-15 ***
edu_groupСпециальное        -0.0081182  0.0148755  -0.546   0.5853
lang_groupJava               0.0266428  0.0195547   1.362   0.1732
lang_groupOther              0.0081256  0.0199145   0.408   0.6833
lang_groupPython             0.0312054  0.0263030   1.186   0.2356
lang_groupWeb               -0.0516417  0.0217136  -2.378   0.0175 *
log(Age)                    -0.2837617  0.3064098  -0.926   0.3545
I(exp^2)                    -0.0053459  0.0008962  -5.965 2.83e-09 ***
exp:Age                     -0.0014362  0.0004898  -2.932   0.0034 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2072 on 2231 degrees of freedom
Multiple R-squared:  0.4236,    Adjusted R-squared:  0.4187
F-statistic: 86.31 on 19 and 2231 DF,  p-value: < 2.2e-16
```

Рис. 21. - Модель з усуненням аномалій.

Оцінка для прогнозованих значень:

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
2035    2733    3153    3208    3652    4778
```

Оцінка для реальних значень:

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
2000    2500    3045    3288    3992    5600
```

Точність цієї моделі становить 0.745535714, що в порівнянні з попередньої моделлю є більшою, тобто модель є кращою.

Тепер перевіримо модель з заповненими пропусками. Оскільки модель без аномалій не на багато, але краще, то в даних також виключимо робітників із заробітною платою більше 6000. Формула для моделі залишається та сама (24).

```
Call:
lm(formula = log(Income) ~ . + log(Age) + I(exp^2) + exp:Age,
    data = dfCompmin)

Residuals:
     Min       1Q   Median       3Q      Max
-0.70619 -0.16825 -0.00416  0.16450  0.94777

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    8.8546418  0.5812643  15.233  < 2e-16 ***
exp                            0.1314309  0.0116491  11.283  < 2e-16 ***
Age                            0.0322116  0.0100317   3.211 0.001333 **
English_lvlниже среднего      -0.1531974  0.0148266 -10.333  < 2e-16 ***
English_lvlпродвинутый         0.0015810  0.0110938   0.143 0.886683
English_lvlсредний            -0.0732393  0.0083386  -8.783  < 2e-16 ***
English_lvlэлементарный       -0.1429308  0.0340941  -4.192 2.82e-05 ***
Compaty_typeАутстаффинговая    0.0547363  0.0103999   5.263 1.49e-07 ***
Compaty_typeДругая            -0.0368783  0.0259321  -1.422 0.155069
Compaty_typeПродуктовая        0.0342742  0.0083494   4.105 4.12e-05 ***
Compaty_typeСтартап            0.0704937  0.0193741   3.639 0.000278 ***
lang_groupJava                 0.0512758  0.0152292   3.367 0.000767 ***
lang_groupOther                0.0205066  0.0155401   1.320 0.187042
lang_groupPython               0.0351672  0.0207005   1.699 0.089420 .
lang_groupWeb                  0.0126850  0.0170962   0.742 0.458143
edu_groupСпециальное           0.0027423  0.0115960   0.236 0.813070
pos_groupManager               0.0511242  0.0095470   5.355 9.02e-08 ***
pos_groupOther                 0.0415974  0.0123382   3.371 0.000755 ***
log(Age)                      -0.6319798  0.2578647  -2.451 0.014294 *
I(exp^2)                      -0.0015308  0.0006937  -2.207 0.027378 *
exp:Age                       -0.0019905  0.0004096  -4.860 1.22e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2313 on 4143 degrees of freedom
Multiple R-squared:  0.3135,    Adjusted R-squared:  0.3102
F-statistic:  94.6 on 20 and 4143 DF,  p-value: < 2.2e-16
```

Рис. 22. - Модель з усуненням пропусків та аномалій.

Оцінка для прогнозованих значень:

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
2077    2675    2985    3072    3456    4434
```

Оцінка для реальних значень:

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
2000    2100    2700    2981    3700    5900
```

Точність цієї моделі становить 0.696096654. В порівнянні з попередніми моделями точність менша, але ця модель краще описує данні, оскільки даних стало більше. Також на точність вплинуло те, що дані заповнювалися не реальними, а «випадковими».

3.4.2. Модель з використанням методу Random Forest

Побудуємо модель з використанням Random Forest. Random Forest є композицією безлічі вирішальних дерев, що дозволяє знизити проблему перенавчання і підвищити точність в порівнянні з одним деревом. Прогноз виходить в результаті агрегування відповідей безлічі дерев. Тренування дерев відбувається незалежно один від одного, що не просто вирішує проблему побудови однакових дерев на одному і тому ж наборі даних.

Основна функція для побудови такого дерева називається randomForest(), а в моєму випадку модель має вигляд:

randomForest(Income~ Position+ exp+ Age+ English_lvl+ Company_type+ edu_group ,data=dataset, ntree=50, mtry=2)

Тут ntree – кількість дерев для розрахунку, mtry - кількість проходів. Через те, що маємо достатньо велику кількість факторів, то збільшення кількості проходів сильно збільшує час розрахунку, тому оптимально залишити 2.

Завдяки формулі випадкового ліса було побудовано модель регресії Подивимось на результати розрахунків.

```
 randomForest(formula = Income ~ ., data = df, do.trace = 50,
mtry = 2, importance = TRUE, nPerm = 3)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 1248.752
                    % Var explained: 51.66
```

Варіант з регресією показав, що дерева не дуже гарно підходять для мого випадку. Як видно процент пояснених значень становить 51.66%. Нижче розглянемо дерева рішень як класифікатор. Маючи таку саме модель, отримали результат помилки майже 88%, тобто лице 12 відсотків пояснено, що є занадто поганим показником.

```
 randomForest(formula = Income ~ ., data = df, do.trace = 50,
mtry = 2, importance = TRUE, nPerm = 3)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 87.95%
```

(код програми у Додатку Г).

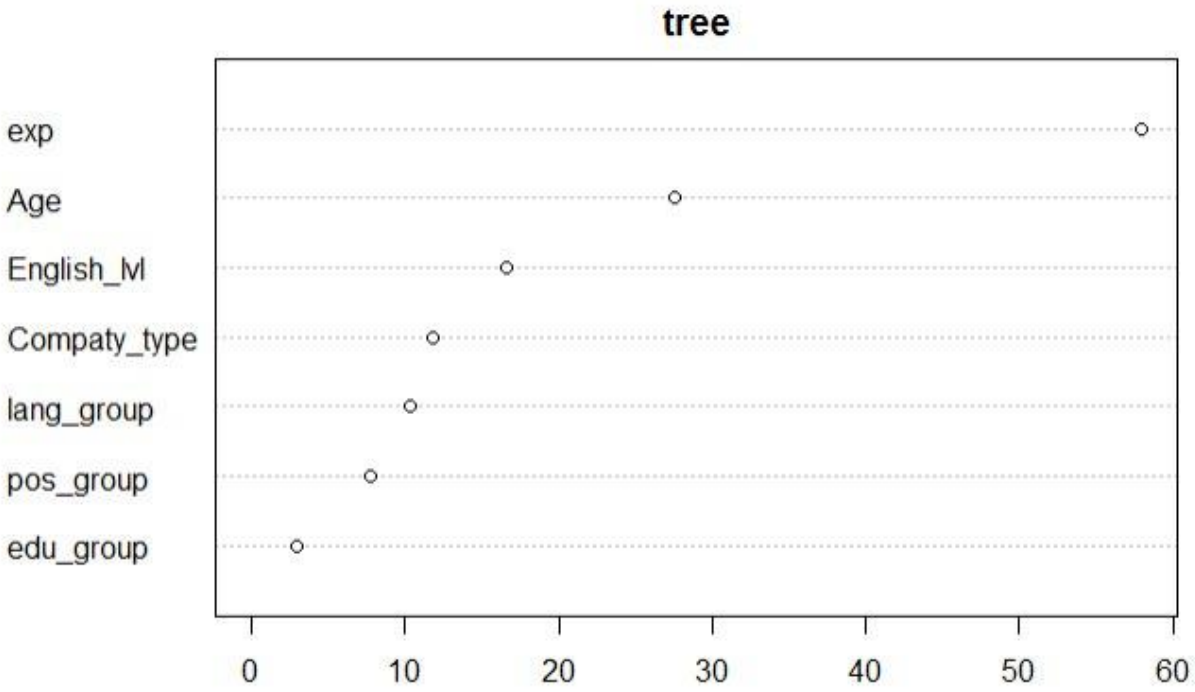

Рис. 23. - Показники важливості окремих змінних для моделі випадкового лісу.

3.4.3. Модель з використанням Gradient Boosting

Спробуємо ще один варіант прогнозування Gradient Boosting. Цей метод є схожим на дерева рішень але має значно більше переваг, тому дає можливість врахувати деякі фактори і значно збільшити точність моделі.

Для побудови моделі скористаємось бібліотекою gbm. На вхід у модель подається список, в якому знаходяться дані для моделювання та розрахунків.

Boost <- gbm(Income ~ ., data = df, distribution = "gaussian",n.trees = 1000, shrinkage = 0.01)

Розглянемо налаштування функції:

В дужках описується модель, до якої входять всі фактори, далі йде список даних, по яким буде будуватися модель. Distribution – розподіл даних, в моєму випадку це гаусівський. Кількість дерев в функції стоїть 100, але цього не вистачає для повного дослідження, тому я збільшила їх до 1000, і останнім параметром йде швидкість навчання, яка сильно впливає на точність розрахунку, але точність досягається суттєво (для прикладу я використаю shrinkage = 0.01 і shrinkage = 0.001.

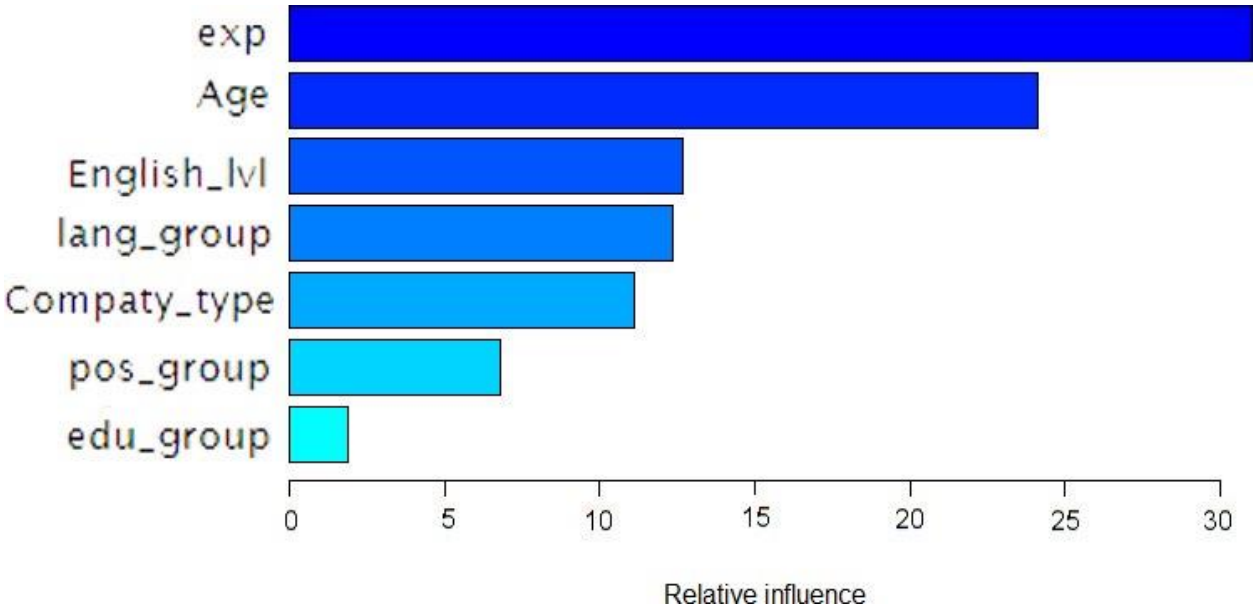

Рис. 24. – Вплив показників в моделі з Gradient Boosting (при швидкості 0.01).

Модель визначила параметр «досвід» найважливішою, на другому місці йде вік працівника.

Також можна подивитися процес навчання і коли похибка стала незначною.

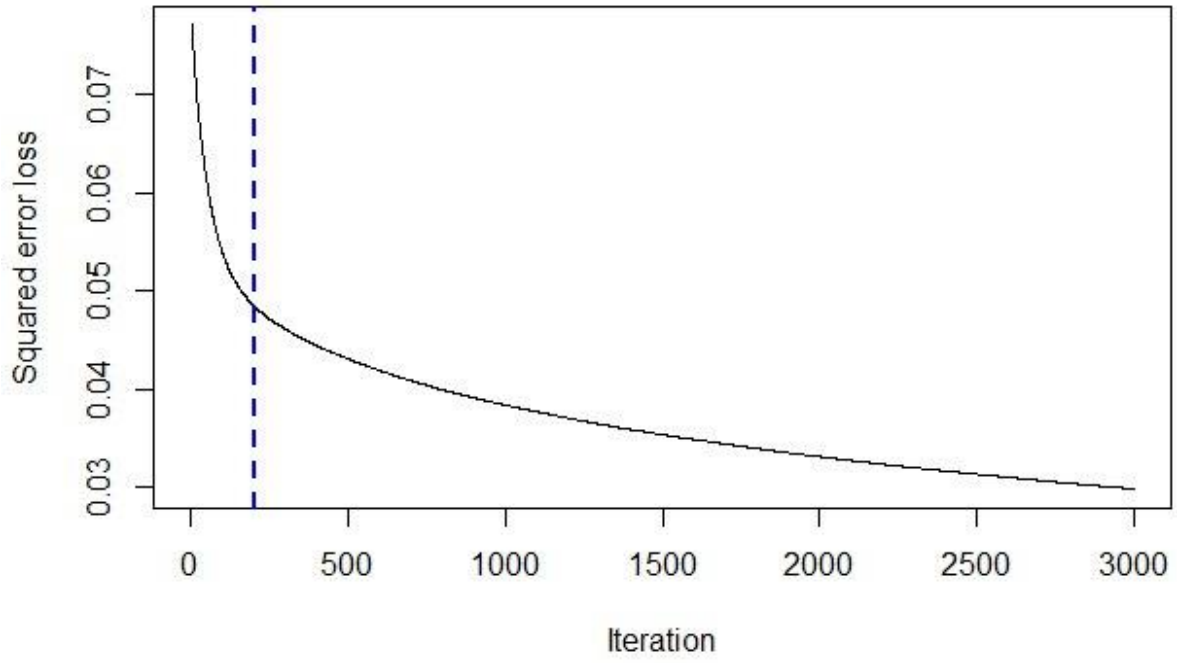

Рис. 25. – Графік швидкості навчання моделі (при швидкості 0.01).

З рисунку 25 видно, що помилка зменшилась приблизно на 250 ітерації, що свідчить про значну швидкість, тому наступні ітерації були не важливі для моделі. Після моделювання на тренінгової вибірці, перейдемо на прогнозування. Для цього зробимо прогноз на тестовій вибірці. Маємо середнє значення з тестової (3449) та прогнозованої вибірки (3566), порівнюючи їх бачимо, що різниця в середніх значеннях менша за 0.05, можна сказати, що вони схожі. Також знайдемо значення AUC = 0.709, що свідчить про те, що точних значень було знайдено близько 70%, пізніше порівняємо значення AUC для дерева рішень та для градієнта.

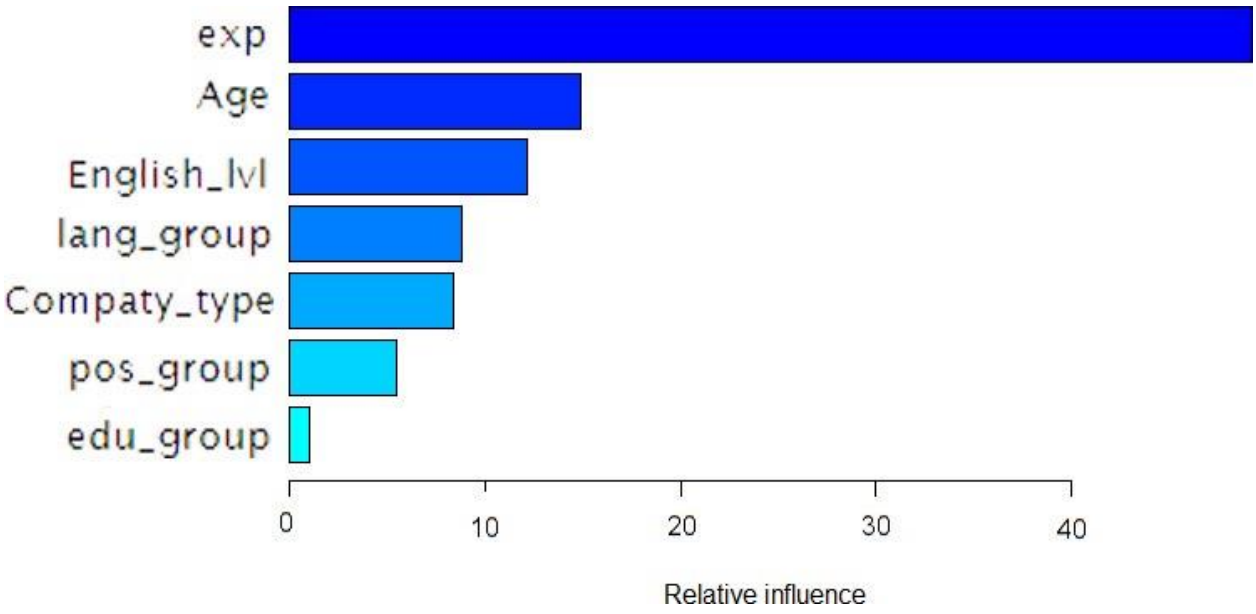

Рис. 26. – Вплив показників в моделі з Gradient Boosting (при швидкості 0.001).

Як бачимо при моделюванні змінились ваги деяких факторів, тоді подивимось на швидкість моделі і порівняємо її з попередньою. На рисунку 27 видно, що помилка зменшувалась вже на 2000 ітерації, це свідчить про більш точний підхід до моделювання.

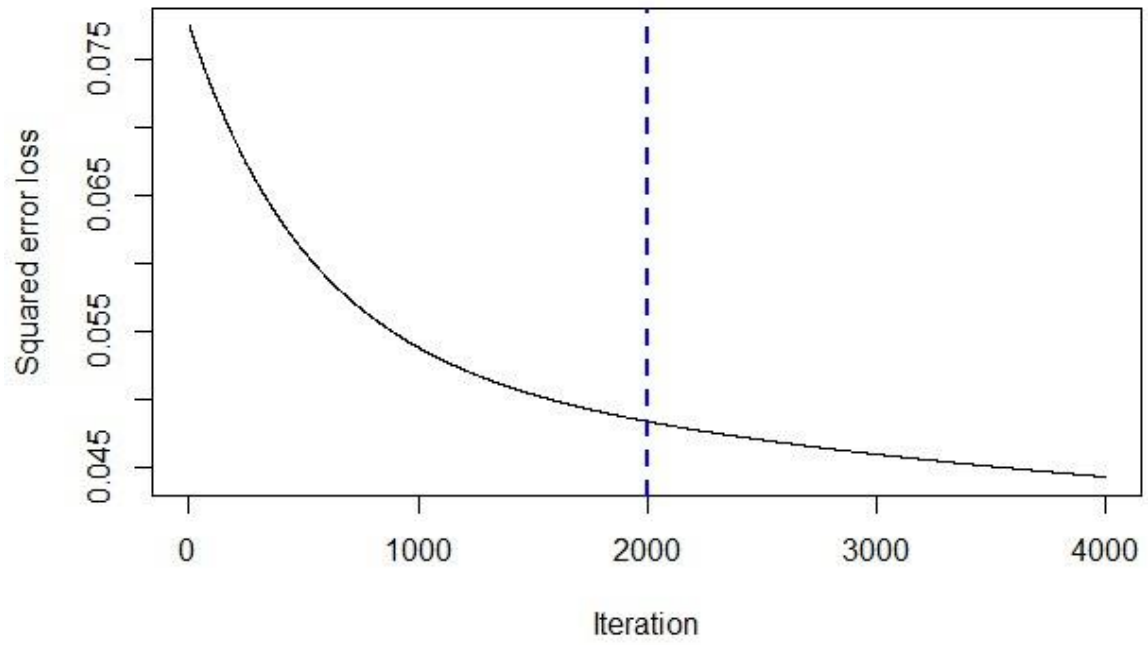

Рис. 27. – Графік швидкості навчання моделі (при швидкості 0.001).

Проведемо прогнозування на моделі та порівняємо середні значення. Нове прогнозоване значення становить 3571, що все одно дає 0.05 довірчого інтервалу при порівнянні з тестовою вибіркою. Наступним кроком буде порівняння усіх отриманих моделей.

Таблиця 6. – Порівняння побудованих моделей.

		RMSE	MAPE
Модель з пропусками та не очищена	model1	876	0,19173
Модель з пропусками та очищена(<6000)	model2	717	0,17389
Повна та очищена модель (<6000)	model3	784	0,22897
Модель з Random Forest	tree	606	0,17441
модель з Gradient Boosting (λ = 0.01)	gradient1	649	0,18732
модель з Gradient Boosting (λ = 0.001)	gradient2	706	0,20711

Дивлячись на таблицю 6, спираючись на показники RMSE та MAPE, які повинні бути мінімальними, то за оптимальну можна прийняти модель з використанням дерева прийняття рішень, оскільки показник похибки за MAPE в даному випадку є 17.4%, але модель з пропущеними значеннями та очищена від аномалій також має схожу похибку, через це було перевірено показник RMSE, що вказує на середнє відхилення від фактичних значень. Через це в сукупності всіх факторів прийнята модель з використанням дерева прийняття рішень.

ВИСНОВКИ

В даній роботі були розглянуті джерела, які стосуються сфери IT. Було досліджено IT – ринок, як в Україні та і за межами, та описано механізм формування заробітної плати. Для Української сфери були виділені основні показники, які впливають на рівень заробітної плати працівника.

Виконано пошук даних для подальшого використання у роботі. При цьому велика кількість даних була пропущена, що могло вплинути на результати моделювання в майбутньому. Тому було прийнято рішення в заповненні пропущених даних, та розгляд моделей з пропущеними та заповненими даними, щоб подивитись на доцільність заповнення пропущених даних.

Основним інструментом при моделюванні було використання статистичних методів в економіці. В якості візуалізації були подані графіки залежностей, статистичного опису та моделей. Були побудовані регресійні моделі та моделі з використанням Random Forest та Gradient Boosting, перевірені на адекватність і оцінена точність кожної з них. За допомогою різних показників помилок встановлено оптимальну модель для прогнозування заробітної плати працівника IT-сфери.

СПИСОК ЛІТЕРАТУРИ

1. Ukraine – the country that codes : IT Industry in Ukraine. 2019 Market Report. URL: https://s3-eu-west-1.amazonaws.com/new.n-ix.com/uploads/2019/09/26/Software_development_in_Ukraine_2019_2020_IT_industry_market_report.pdf (Last accessed: 09.06.2020)

2. Овчаренко Д. IT в Україні: куди ми рухаємося. *DOU.ua* : вебсайт. URL: https://dou.ua/lenta/columns/future-of-it-ukraine/ (дата звернення: 09.06.2020).

3. Державна служба статистики України : вебсайт. URL: http://ukrstat.gov.ua (дата звернення: 09.06.2020).

4. Зовнішня торгівля України послугами у 2019 році : експрес-випуск / Державна служба статистики України. 2020. URL: http://www.ukrstat.gov.ua/express/expr2020/02/16.pdf?fbclid=IwAR38B6V8aXkgkliZx_ip-S8M63nVE0jLb-XbjtYaup672NyrzbGJQ4jObiQ (дата звернення: 09.06.2020).

5. Malashniak M. Sofrware development in Ukraine 2019-2020 IT market report. *N-IX* : website. URL: https://www.n-ix.com/software-development-in-ukraine-2019-2020-market-report/ (Last accessed: 09.06.2020).

6. Відмінності junior, middle и senior розробників. *Tproger* : вебсайт. URL: https://tproger.ru/experts/junior-middle-senior-developers-differences (дата звернення: 09.06.2020).

7. Список популярних професій IT-сфери. *Навигатор поступления* : образовательный форум. URL: https://propostuplenie.ru/article/spisok-populyarnyh-professij-v-it-sfere/ (дата обращения: 09.06.2020).

8. Самые востребованные IT-профессии на 2020 год. *IT рейтинг Украины* : вебсайт. URL: https://it-rating.in.ua/samyie-vostrebovannyie-it-professii-na-2020-god (дата обращения: 09.06.2020).

9. Badr W. 6 Different Ways to Compensate for Missing Values In a Dataset (Data Imputation with examples). *Medium* : website. URL: https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779 (Last accessed: 09.06.2020).

10. Buuren S. V., Groothuis-Oudshoorn, K. Mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*. 2011. Vol. 45, Issue 3.

11. Маринич Т. О. Математичні методи в економіці : навч. матер. URL: https://elearning.sumdu.edu.ua/s/15-ktn (дата звернення: 09.06.2020).

12. Чубукова И. Задачи Data Mining. Класификация и кластеризация. *Национальный Открытый Университет «ИНТУИТ»* : вебсайт. URL: https://www.intuit.ru/studies/courses/6/6/lecture/166?page=4 (дата обращения: 09.06.2020).

13. Дрейпер Н., Смит Г. Прикладной регрессионный анализ. 3-е изд. М . : Вильямс, 2007. 912 с.

14. Дерева рішень. *Навчальні матеріали онлайн*. URL: https://pidruchniki.com/12280805/informatika/dekompozitsiya_tsiley_informatsiynoyi_tehnologiyi (дата звернення: 09.06.2020).

15. Градиентный спуск: всё, что нужно знать : базовый курс. *Neurohive* : вебсайт. URL: https://neurohive.io/ru/osnovy-data-science/gradient-descent/ (дата обращения: 09.06.2020).

16. Критерий Шапиро-Уилка. *MachineLearning.ru* : вебсайт. URL: http://www.machinelearning.ru/wiki/index.php?title=%D0%9A%D1%80%D0%B8%D1%82%D0%B5%D1%80%D0%B8%D0%B9_%D0%A8%D0%B0%D0%BF

%D0%B8%D1%80%D0%BE-%D0%A3%D0%B8%D0%BB%D0%BA%D0%B0
(дата обращения: 09.06.2020).

ДОДАТОК А

Географічна структура зовнішньої торгівлі послугами у 2018 році

	Експорт			Імпорт			Сальдо
	млн.дол. США	у % до 2018р.	у % до загаль-ного обсягу	млн.дол. США	у % до 2018р.	у % до загаль-ного обсягу	
Усього	**15237,5**	**130,9**	**100,0**	**6527,9**	**103,5**	**100,0**	**8709,6**
у тому числі							
Австрія	224,2	113,6	1,5	98,9	100,7	1,5	125,3
Бельгія	110,6	119,0	0,7	172,8	111,2	2,6	-62,2
Білорусь	141,2	123,1	0,9	127,4	106,6	2,0	13,8
Болгарія	39,2	147,3	0,3	26,0	152,0	0,4	13,2
Британські Віргінські Острови	160,7	107,1	1,1	0,9	71,5	0,0	159,8
Греція	24,2	105,1	0,2	37,9	126,6	0,6	-13,7
Грузія	42,2	117,2	0,3	52,3	132,1	0,8	-10,1
Данія	179,2	113,8	1,2	71,9	127,4	1,1	107,3
Естонія	194,3	110,3	1,3	39,9	59,4	0,6	154,4
Єгипет	75,4	106,8	0,5	188,4	135,8	2,9	-113,0
Ізраїль	235,9	116,9	1,5	53,6	107,7	0,8	182,3
Індія	121,0	118,6	0,8	9,0	104,3	0,1	112,0
Ірландія	86,8	147,4	0,6	306,0	121,2	4,7	-219,1
Іспанія	84,7	88,1	0,6	50,2	122,3	0,8	34,5
Італія	128,4	93,1	0,8	69,5	114,9	1,1	58,9
Казахстан	47,3	80,0	0,3	30,0	89,6	0,5	17,3
Канада	95,6	108,1	0,6	18,2	36,3	0,3	77,4
Китай	162,7	153,5	1,1	222,5	117,7	3,4	-59,7
Кіпр	335,3	105,4	2,2	400,9	103,7	6,1	-65,7
Латвія	50,3	73,2	0,3	47,6	120,9	0,7	2,7
Литва	56,1	117,8	0,4	40,7	141,6	0,6	15,5
Люксембург	11,5	74,9	0,1	96,3	84,0	1,5	-84,8
Мальта	154,4	129,6	1,0	161,1	112,1	2,5	-6,7
Нідерланди	231,8	119,7	1,5	142,3	104,5	2,2	89,5
Німеччина	584,3	98,8	3,8	454,0	112,6	7,0	130,3
Об'єднані Арабські Емірати	346,7	135,0	2,3	112,2	77,0	1,7	234,5
Польща	403,1	115,1	2,6	222,6	113,5	3,4	180,4
Республіка Корея	50,4	97,9	0,3	15,0	67,9	0,2	35,5
Республіка Молдова	49,2	106,0	0,3	29,0	92,3	0,4	20,2
Російська Федерація	6181,6	185,3	40,6	288,3	63,7	4,4	5893,4
Румунія	115,0	123,6	0,8	37,8	95,0	0,6	77,2
Словаччина	51,3	115,2	0,3	89,0	75,2	1,4	-37,6
Сполучене Королівство Великої Британії та Північної Ірландії	583,5	101,8	3,8	586,5	114,6	9,0	-3,0
США	1219,4	120,5	8,0	495,6	107,5	7,6	723,8
Туреччина	193,7	108,7	1,3	552,9	135,0	8,5	-359,2
Угорщина	211,7	106,4	1,4	56,3	129,6	0,9	155,4
Фінляндія	30,0	98,2	0,2	30,3	89,0	0,5	-0,4
Франція	178,8	116,1	1,2	161,3	132,1	2,5	17,5
Чехія	91,8	109,7	0,6	59,9	106,6	0,9	32,0
Швейцарія	972,7	108,6	6,4	215,7	94,8	3,3	757,0
Швеція	93,4	91,9	0,6	48,9	36,2	0,7	44,5
Довідково:							
Країни ЄС	4288,0	107,7	28,1	3563,8	106,2	54,6	724,2

ДОДАТОК Б

Підключення бази даних та візуальна статистика

```
data <- read.csv('dec2019mini.csv',na.strings=c(""))
summary(data)
str(data)
sapply(data, function(data) sum(is.na(data)))
data$Специализация <- NULL
data$N <- NULL

ggplot(data, aes(x = Income)) +
  geom_histogram(binwidth = 500, colour = "black", fill = "greenyellow")

ggplot(data, aes(x = Age)) +
  geom_histogram(binwidth = 2, colour = "black", fill = "greenyellow")
tab <- table(data$age_group)
perc <- tab/sum(tab) * 100
colors <- c('thistle2', 'plum2', 'palevioletred', 'orchid4', 'purple3', "blue",
"green")
pie(perc, col = colors,main = "Вік")

ggplot(data, aes(x = exp)) +
  geom_histogram(binwidth = 1, colour = "black", fill = "greenyellow")
tab <- table(data$exp_group)
perc <- tab/sum(tab) * 100
colors <- c('thistle2', 'plum2', 'palevioletred', 'orchid4', 'purple3', "blue",
"green")
pie(perc, col = colors,main = "Досвід")

tab <- table(data$Sex)
perc <- tab/sum(tab) * 100
colors <- c('thistle2', 'blue', 'palevioletred', 'orchid4', 'purple3', "blue",
"green")
pie(perc, col = colors,main = "Стать")

tab <- table(data$Education)
perc <- tab/sum(tab) * 100
colors <- c('thistle2', 'plum2', 'palevioletred', 'orchid4', 'purple3', "blue",
"green")
```

```
pie(perc, col = colors,main = "Освіта")

ggplot(data, aes(x =
Programming_language),vertical)+geom_bar()+theme(axis.text.x =
element_text( size=12, angle=90, hjust=1))


qqnorm(data$Income)
qqline(data$Income)
qqnorm(data$Age)
qqline(data$Age)
qqnorm(data$exp)
qqline(data$exp)
shapiro.test(data$Income)
shapiro.test(data$Age)
shapiro.test(data$exp)
```

ДОДАТОК В

Очищення та заповнення порожніх значень.

```
library(mice)
md.pattern(data)
library(VIM)
aggr_plot <- aggr(data, col=c('navyblue','red'), numbers=TRUE,
sortVars=TRUE, labels=names(data), cex.axis=.7, gap=3, ylab=c("Histogram of
missing data","Pattern"))
tempData <- mice(data,m=1,maxit=5,meth='pmm',seed=50)

completeData <- complete(tempData)
aggr_plot <- aggr(completeData, col=c('navyblue','red'), numbers=TRUE,
sortVars=TRUE, labels=names(data), cex.axis=.7, gap=3, ylab=c("Histogram of
missing data","Pattern"))
completeData$exp_group <- sapply(completeData$exp,group_exp)
completeData$exp_group <- as.factor(completeData$exp_group)
completeData$age_group <- sapply(completeData$Age,group_age)
completeData$age_group <- as.factor(completeData$age_group)
completeData$income_group <-
sapply(completeData$Income,group_income)
completeData$income_group <- as.factor(completeData$income_group)

data$Position <- as.factor(mapvalues(data$Position,
                                     from=c("Director of Engineering / Program
Director"),
                                     to=c("Program Director")))
data$Position <- as.factor(mapvalues(data$Position,
                                     from=c("Senior Project Manager / Program
Manager"),
                                     to=c("Program Manager")))
group_age <- function(age){
   if (age >= 18 & age <= 25){
      return('18-25 years')
   }else if(age > 25 & age <= 35){
      return('26-35 years')
   }else if (age > 35 & age <= 45){
      return('36-45 years')
```

```
    }else if (age > 45 & age <= 60){
      return('46-60 years')
    }else if (age > 60){
      return('> 60 years')
    }
  }
  data$age_group <- sapply(data$Age,group_age)
  data$age_group <- as.factor(data$age_group)
  group_lang <- function(Programming_language){
   if (Programming_language == 'Java' || Programming_language ==
  'JavaScript' || Programming_language == 'PHP' || Programming_language ==
  'C#'){
      return('Popular')
   }else if(Programming_language == 'C++' || Programming_language == 'C'
  || Programming_language == 'Golang' || Programming_language == 'Kotlin'
  ||Programming_language == 'Python' || Programming_language == 'SQL' ||
  Programming_language == 'Swift'){
      return('SemiPopular')
   }else {
      return('NonPopular')
   }
}
dfremove$lang_group                                                    <-
  sapply(dfremove$Programming_language,group_lang)
dfremove$lanf_group <- as.factor(dfremove$lang_group)
group_edu <- function(Education){
   if (Education == 'Два высших' || Education == 'Высшее' || Education ==
  'Кандидат'){
      return('Высшее')
   }else {
      return('Специальное')
   }
```

```
}
dfremove$edu_group <- sapply(dfremove$Education,group_edu)
dfremove$edu_group <- as.factor(dfremove$edu_group)
group_income <- function(income){
  if (income >= 2000 & income <= 4000){
    return('2000$-4000$')
  }else if(income > 4000 & income <= 6000){
    return('4000$-6000$')
  }else if (income > 6000 & income <= 8000){
    return('6000$-8000$')
  }else if (income > 8000 & income <= 10000){
    return('8000$-10000$')
  }else if (income > 10000 & income <= 12000){
    return('10000$-12000$')
  }
}
data$income_group <- sapply(data$Income,group_income)
data$income_group <- as.factor(data$income_group)
group_exp <- function(exp){
  if (exp >= 0 & exp <= 3){
    return('0 - 3 years')
  }else if(exp > 3 & exp <= 6){
    return('3 - 6 years')
  }else if (exp > 6 & exp <= 9){
    return('6 - 9 years')
  }else if (exp > 9){
    return('> 9 years')
  }
data$exp_group <- sapply(data$exp,group_exp)
data$exp_group <- as.factor(data$exp_group)
```

ДОДАТОК Г

Побудова Моделей для прогнозування

```
df <- dfremove[c(1,2,3,5,7,8,11)]
dfComp <- completeData[c(1,2,3,5,7,8,11)]
datamin <- dfremove[c(1,2,3,5,7,8,11)]
datamin <- datamin[which(datamin$Income < 5000),]
df <- na.omit(df)
intrain<- createDataPartition(df$Income,p=0.8,list=FALSE)
set.seed(200)
training<- df[intrain,]
testing<- df[-intrain,]
lmI1 <- lm(log(Income) ~ ., data = df)
lmI2 <- lm(log(Income) ~ .+log(Age)+Age*exp, data = df)
lmI3 <- lm(log(Income) ~ .+log(Age)+log(exp), data = df)
lmI4 <- lm(log(Income) ~ .+exp:Age, data = df)
lmI5 <- lm(log(Income) ~ . +log(Age)+I(exp)+exp:Age, data = df)
summary(lmI2)

plot(density(data$Income), col = "red", lwd = 2)
plot(density(log(data$Income)), col = "red", lwd = 2)

par(mflow = c(2,2))
plot(model)
model <- lm(log(Income) ~ .+exp:Age, data = df)

fitted.results <- exp(predict(model, newdata=testing))
summary(fitted.results)
summary(testing$Income)
ft <- fitted.results-testing$Income
```

```
fitted.results1 <- ifelse(abs(ft) < 800,1,0)
table(fitted.results1)

misClasificError <- mean(fitted.results1 != 1)
print(paste('Logistic Regression Accuracy',1-misClasificError))
vif(model)
r2 <- summary(model)$r.squared
vif <- 1/(1-r2)
vif

anova(lmI1, lmI2)
anova(lmI2, lmI3)
anova(lmI3, lmI4)
anova(lmI4, lmI5)

dfComp <- completeData[c(2,4,5,9,16,23)]
dfCompmin <- dfComp[which(dfComp$Income < 5000),]
datacomp <- na.omit(dfCompmin)
intrain<- createDataPartition(datacomp$Income,p=0.8,list=FALSE)
set.seed(200)
training<- datacomp[intrain,]
testing<- datacomp[-intrain,]

model <- lm(log(Income) ~ .+exp:Age, data = datacomp)
pre <- exp(predict(model, newdata=testing))
auc(testing$Income, pre)

fitted.results <- exp(predict(model, newdata=testing))
summary(fitted.results)
```

```
summary(testing$Income)
ft <- fitted.results-testing$Income

fitted.results1 <- ifelse(abs(ft) < 800,1,0)
table(fitted.results1)

misClasificError <- mean(fitted.results1 != 1)
print(paste('Logistic Regression Accuracy',1-misClasificError))

psych::pairs.panels(df)

rf <- randomForest(log(Income) ~ ., data = df, do.trace=50, mtry=2,
importance=TRUE, nPerm=3)

tree <- randomForest(Income ~ ., data = df, do.trace=50, mtry=2,
importance=TRUE, nPerm=3)
rf
tree
predicte <- predict(tree, testing)
tab <- table(predicte, testing$Income)
print(sum(diag(tab))/sum(tab))
auc(testing$Income, predicte)

library(gbm)
require(MASS)

Boost <- gbm(Income ~ ., data = df, distribution = "gaussian",n.trees =
30000, shrinkage = 0.0001, interaction.depth = 30, keep.data = TRUE)
Boost
summary(Boost)
```

```
boostpred <- predict.gbm(Boost, newdata = testing, n.trees = 100000, type
= "link")
gbm.perf(Boost)
head(boostpred)

caret::RMSE(boostpred, testing$Income)
auc(testing$Income, boostpred)

x <- mean(testing$Income)
y <- caret::RMSE(boostpred, testing$Income)
x
y

print(x/y*100)
test <- testing
test$pred <- boostpred
test$old <- testing$Income
```