Міністерство освіти і науки України

Західноукраїнський національний університет

Факультет комп’ютерних інформаційних технологій

Кафедра інформаційно-обчислювальних систем і управління

МІЖДИСЦИПЛІНАРНИЙ КУРСОВИЙ ПРОЕКТ

на тему: «Прогнозування світової тенденції заробітньої плати»

Студента IІІ курсу КН-31 групи

спеціальності «Комп’ютерні науки»

Ковальковський В. В.

Керівник: викл. Кіт І. Р.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали) \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

м. Тернопіль – 2023 рік

Західноукраїнський національний університет

Факультет комп’ютерних інформаційних технологій

Кафедра інформаційно-обчислювальних систем і управління

„Затверджую”

Зав. кафедри ІОСУ

д.т.н., доц. Мирослав Комар

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_2023 р.

ЗАВДАННЯ

на міждисциплінарний курсовий проект студента

Ковальковський Віталій Віталійович

(прізвище, ім’я, по-батькові)

1. Тема проекту Прогнозування світової тенденції заробітньої плати

затверджена на засіданні кафедри ІОСУ №\_\_\_ від “\_\_\_”\_\_\_\_\_\_\_\_\_2023 р.

2. Термін здачі студентом закінченого проекту \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Вихідні дані по проекту \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, котрі підлягають розробці)

* теоретичне дослідження матеріалу
* розглянути методи аналізу, очистки та візуалізації даних
* створити модель прогнозування
* проаналізувати результати та зробити висновки

5. Перелік графічного матеріалу (з точним вказанням обов’язкових креслень)

* графіки залежностей різних змін
* графік теплокарти кореляції
* результати регресій

6. Консультанти розділів проекту

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посада  консультанта | Підпис, дата | |
| Завдання видав | Завдання  прийняв |
|  |  |  |  |
|  |  |  |  |

7.Дата видачі завдання\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів курсового  проектування | Строк виконання етапів | Примітка |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Студент \_\_\_\_\_\_\_\_\_ Ковальковський В. В.

( підпис ) (прізвище та ініціали)

Керівник курсового проекту \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( підпис ) (прізвище та ініціали)

ЗМІСТ

[ВСТУП 2](#_Toc152964453)

[РОЗДІЛ 1. ПІДГОТОВКА ТА АНАЛІЗ НАБОРУ ДАНИХ 2](#_Toc152964454)

[1.1 Актуальність роботи 2](#_Toc152964455)

[1.2 Огляд існуючих програмних рішень для прогнозування 2](#_Toc152964456)

[1.3 Постановка задачі 2](#_Toc152964457)

[РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ ДЛЯ ПРОГНОЗУВАННЯ 2](#_Toc152964458)

[2.1 Методи аналізу, очистки та візуалізації даних 2](#_Toc152964459)

[2.2 Інформаційне забезпечення прогнозу заробітніх плат 2](#_Toc152964460)

[РОЗДІЛ 3. АНАЛІЗ ТА ВІЗУАЛІЗАЦІЯ РЕЗУЛЬТАТІВ 2](#_Toc152964461)

[3.1 Розробка моделі прогнозування 2](#_Toc152964462)

[3.2 Архітектура та функціональність бота 2](#_Toc152964463)

[3.3 Інтеграція бота з системою прогнозування 2](#_Toc152964464)

[3.4 Аналіз отриманих результатів 2](#_Toc152964465)

[3.5 Тестування та оптимізація роботи бота 2](#_Toc152964466)

[ВИСНОВКИ 2](#_Toc152964467)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 2](#_Toc152964468)

[ДОДАТОК А 2](#_Toc152964469)

[ДОДАТОК Б 2](#_Toc152964470)

[ДОДАТОК В 2](#_Toc152964471)

## ВСТУП

**Актуальність теми.** Прогнозування заробітних плат є важливим аспектом у плануванні економічних та соціальних політик. Це дослідження важливе для урядів, економічних аналітиків, компаній та приватних осіб, оскільки воно допомагає зрозуміти поточні тенденції та прогнозувати майбутні зміни у заробітних платах. Використання сучасних технологій, таких як телеграм-бот, робить цю інформацію більш доступною та зрозумілою для широкого кола користувачів.

Актуальність дослідження заробітних плат у світі має багатогранний характер, оскільки ця тема впливає на ряд ключових соціально-економічних аспектів:

* економічний розвиток: Заробітні плати безпосередньо впливають на економічний розвиток країн та регіонів. Розуміння та прогнозування змін у заробітних платах допомагає урядам та підприємствам в адаптації економічних стратегій, сприяючи стабільності та росту;
* еолітика зайнятості: Актуальність дослідження також полягає у впливі на політику зайнятості. Прогнози заробітних плат допомагають формувати ефективні стратегії зайнятості, що сприяє зниженню безробіття та підвищенню життєвого рівня;
* міжнародна міграція: Заробітні плати є одним із ключових факторів, що впливають на міжнародну міграцію робочої сили. Прогнозування цих показників допомагає розуміти міграційні потоки та їх вплив на світову економіку;
* соціальна справедливість: Аналіз та прогнозування заробітних плат сприяє виявленню та вирішенню проблем соціальної нерівності, гендерної нерівності в оплаті праці та інших соціальних диспропорцій;
* планування кар'єри: Для приватних осіб інформація про заробітні плати є важливою для планування кар'єри та освітнього шляху. Це допомагає індивідам вибирати професії та навички, що забезпечують кращі перспективи на ринку праці;
* розвиток технологій: Дослідження сприяє розвитку та впровадженню нових технологій в аналізі та прогнозуванні економічних показників, зокрема розробці інтерактивних платформ та інструментів, таких як телеграм-боти.

Таким чином, актуальність дослідження полягає не тільки у важливості прогнозування заробітних плат для розуміння та вирішення економічних та соціальних питань, але й у впровадженні новітніх технологічних рішень для доступу до цих даних.

**Об'єктом дослідження** є комплексна система заробітних плат, що охоплює не тільки самі заробітні плати, але й ряд важливих пов'язаних з ними характеристик, таких як вік працівників, їхні роки досвіду, гендер, а також займані ними посади. Цей об'єкт включає в себе аналіз взаємозв'язків між цими факторами та їх впливом на рівень заробітних плат у різних секторах та регіонах. Вивчення цих аспектів дозволяє глибше зрозуміти структуру та динаміку заробітних плат, а також виявити тенденції та закономірності, що впливають на ринок праці в цілому.

**Предметом дослідження** є методи прогнозування заробітних плат, включаючи збір та аналіз даних, набір математичних та статистичних моделей для прогнозування, а також використання програмних рішень для візуалізації та представлення даних. Особливу увагу приділено створенню та впровадженню інтерактивного телеграм-бота як засобу для демонстрації результатів прогнозування та забезпечення доступу до актуальної інформації про заробітні плати.

# РОЗДІЛ 1. ПІДГОТОВКА ТА АНАЛІЗ **НАБОРУ** ДАНИХ

### Актуальність роботи

Актуальність курсової роботи полягає у глибокому аналізі та прогнозуванні глобальних тенденцій заробітної плати, що є важливим для розуміння економічних і соціальних процесів в сучасному світі. Робота зосереджується на зборі та обробці даних про заробітну плату, використовуючи статистичні методи та інноваційні програмні рішення. Це дозволяє отримати цінну інформацію для урядів, організацій, а також приватних осіб для планування їхньої кар'єри та розвитку. Особливу увагу приділено впливу глобалізації та технологічних змін на динаміку заробітної плати.

Актуальність залежить від декількох аспетів:

Глобалізація економіки: В умовах постійної глобалізації, зрозуміння відмінностей у заробітних платах по всьому світу є важливим для корпорацій, що планують міжнародну експансію або вже працюють на глобальному ринку.

Міжнародна міграція робочої сили: Аналіз заробітних плат дає змогу зрозуміти потоки міжнародної міграції, оскільки працівники часто переміщуються в пошуках кращих економічних можливостей.

Економічна рівність і соціальна справедливість: Вивчення різниці у заробітних платах може вказувати на нерівності у світовому економічному розподілі та сприяти розробці політик для зменшення цих розривів.

Політика зайнятості та заробітної плати: Розуміння глобальних тенденцій заробітної плати допоможе урядам та міжнародним організаціям розробляти ефективні політики у сфері зайнятості.

Вплив технологічних змін: Аналіз заробітних плат також є важливим для вивчення впливу автоматизації та цифровізації на ринок праці, особливо в контексті розширення можливостей дистанційної роботи.

Економічний розвиток: Дослідження рівнів оплати праці в різних країнах може допомогти зрозуміти, як економічний розвиток впливає на добробут працівників і на зменшення бідності.

Міжкультурне порівняння: Вивчення заробітних плат у різних культурних і географічних контекстах дає змогу краще зрозуміти економічні стратегії та культурні цінності, що формують ринок праці.

Отже, виходячи з проведеного аналізу, можна зробити висновок, що дана курсова робота є вкрай актуальною та значущою у контексті сучасних економічних та соціальних реалій. Глибоке дослідження заробітних плат на глобальному рівні, використання статистичних методів та інноваційних програмних рішень для аналізу та прогнозування забезпечує комплексне розуміння впливу глобалізації, міграційних процесів, економічної нерівності, політик зайнятості, технологічних змін, економічного розвитку та міжкультурних відмінностей на динаміку заробітних плат.

### Огляд існуючих програмних рішень для прогнозування

Python охоплює широкий спектр бібліотек та функцій, які використовуються для обробки даних, статистичного аналізу, машинного навчання та глибокого навчання. Ці інструменти забезпечують потужну підтримку для розробки прогнозних моделей у різних областях.

NumPy та Pandas:

На самому початку, NumPy та Pandas є фундаментальними бібліотеками для будь-якої роботи з даними. NumPy надає підтримку для великих багатовимірних масивів та матриць, разом з великим набором високорівневих математичних функцій для роботи з цими масивами. Pandas надає високофункціональні структури даних, такі як DataFrame, і засоби для ефективної маніпуляції та аналізу даних.

SciPy доповнює NumPy, надаючи користувачам додаткові модулі для оптимізації, регресії, інтерполяції та інших задач обчислювальної математики, які часто використовуються в наукових дослідженнях.

Matplotlib та Seaborn:

Для візуалізації даних Matplotlib та Seaborn є вибором для багатьох аналітиків. Matplotlib дозволяє створювати графіки та діаграми для ілюстрації даних та тенденцій, тоді як Seaborn забезпечує більш високорівневий інтерфейс і кращі за замовчуванням стилі.

Scikit-learn є однією з найпопулярніших бібліотек для машинного навчання в Python. Вона надає прості та ефективні інструменти для аналізу даних та майнінгу даних. Вона включає класифікацію, регресію, кластеризацію та зменшення розмірності, разом з інструментами для вибору моделі, обробки даних та оцінки моделі.

Statsmodels – це модуль Python, який дозволяє користувачам досліджувати дані, оцінювати статистичні моделі та виконувати статистичні тести. Цей інструмент чудово підходить для тих, хто хоче проводити детальний статистичний аналіз.

Це лише декілька з багатьох інструментів та бібліотек, доступних у Python для прогнозування. Вибір конкретного інструменту залежить від конкретних потреб, обсягу даних та рівня аналізу даних.

### Постановка задачі

Як ми вже визначились робота є актуальною, оскільки розуміння динаміки заробітних плат має ключове значення для різних аспектів глобальної економіки. Вона охоплює не тільки аналіз заробітних плат, але й вивчає взаємозв'язок з такими факторами, як вік працівників, їхній досвід, гендер та посади. Метою курсової роботи є розробка моделі прогнозування заробітньої плати з використанням різних методів та інструментів, включаючи статистичний аналіз, машинне навчання та розробку інтерактивного телеграм-бота.

Для досягнення мети потрібно вирішити такі задачі:

* Аналіз історичних даних та визначення ключових факторів, що впливають на заробітну плату.
* Розробка моделі прогнозування з використанням статистичних методів та алгоритмів машинного навчання.
* Створення та інтеграція телеграм-бота для візуалізації та доступу до прогнозних даних.
* Аналіз результатів прогнозування для виявлення глобальних тенденцій на ринку праці.

# РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ ДЛЯ ПРОГНОЗУВАННЯ

### 2.1 Методи аналізу, очистки та візуалізації даних

Вибір методів та інструментів для прогнозування є ключовим аспектом дослідницької роботи, який вимагає ретельного аналізу та уважного планування. Цей етап визначає, як саме ви будете аналізувати та прогнозувати дані, і він має велике вплив на якість та достовірність результатів дослідження. Основні аспекти вибору методології та інструментів включають різні параметри.

Типи та характеристики даних: Важливо враховувати типи та характеристики ваших даних. Якщо у вас є числові дані, категоріальні дані або часові ряди, то це впливає на вибір методології. Наприклад, для роботи з часовими рядами можуть бути використані спеціалізовані методи.

Доступність та обсяг даних: Вибір методології також залежить від доступності та обсягу даних. Деякі методи можуть вимагати великої кількості даних для точних прогнозів. При цьому важливо визначити, чи є у вас доступ до достатнього обсягу інформації.

Комплексність моделей: Для складних явищ часто потрібні складні моделі та інструменти. Вибір методології повинен бути збалансованим з урахуванням складності об'єкта дослідження. Наприклад, для аналізу простих взаємозв'язків можуть використовуватися прості методи, а для складних моделей - більш складні інструменти.

Перевірка та оцінка моделі: Під час вибору методології також слід враховувати можливість перевірки та оцінки моделі. Наявність метрик та тестів для оцінки якості моделі є важливою. Методологія повинна передбачати можливість визначення точності та достовірності прогнозів.

Час та ресурси: Вибір методології також залежить від доступності часу та ресурсів. Деякі методи можуть вимагати значних обчислювальних ресурсів, тому важливо оцінити їх доступність.

Узгодження всіх цих аспектів допомагає визначити оптимальний підхід для досягнення цілей дослідження. Вибір методології та інструментів повинен бути обґрунтованим та спрямованим на отримання достовірних та значущих результатів.

Ручне прогнозування - це метод прогнозування, який ґрунтується на інтуїції, досвіді та аналізі доступних даних без використання автоматизованих моделей чи інструментів. Цей метод може бути корисним у випадках, коли недостатньо даних для застосування складних моделей або коли необхідно швидко зробити оцінку ситуації. Важливо враховувати наступні аспекти при ручному прогнозуванні:

Експертний досвід: Ручне прогнозування ґрунтується на експертному досвіді. Особа, яка виконує прогноз, повинна мати глибоке розуміння об'єкта дослідження та його взаємозв'язків.

Доступність даних: Важливо мати доступ до наявних даних, які можуть використовуватися для аналізу та прогнозування. Це можуть бути статистичні дані, інформація про минулі події, або знання про поточну ситуацію.

Аналіз трендів: При ручному прогнозуванні важливо аналізувати тренди та патерни у даних. Це допомагає визначити, які зміни відбуваються та як вони можуть вплинути на майбутні результати.

Експертні оцінки: Часто ручне прогнозування включає експертні оцінки, де експерти висловлюють свою думку щодо майбутніх подій. Ці оцінки можуть бути підкріплені аргументами та аналізом.

Апробація та корекція: Ручний прогноз може бути підданий апробації та корекції. Інші експерти або команда можуть переглянути прогноз та внести свої зауваження або корекції.

Суб'єктивність: Важливо усвідомлювати, що ручне прогнозування може бути суб'єктивним і піддаватися впливу особистих переконань та біасу. Тому важливо стежити за об'єктивністю та доказовістю прогнозу.

Моніторинг та оновлення: Прогнози повинні бути періодично моніторингом та оновлюватися на основі нової інформації та змін у ситуації.

Ручне прогнозування може бути корисним інструментом в різних областях, але воно повинно використовуватися обачливо та враховувати всі доступні дані та експертні знання.

Очистка даних є критично важливим етапом у підготовці датасету для подальшого аналізу та моделювання. Недостовірні, неповні або некоректно оброблені дані можуть значно спотворити результати дослідження. Цей розділ описує методи та підходи, які були використані для очистки та підготовки датасету у даній курсовій роботі.

**Правильно очистка даних включає:**

Виявлення аномалій: Використання статистичних методів для ідентифікації значень, які видаються незвичними або аномальними (наприклад, екстремально високі або низькі заробітні плати).

Перевірка на відповідність форматам: Упевненість у тому, що всі дані відповідають очікуваним форматам, таким як дата, час, числові та текстові формати.

Визначення пропущених значень: Аналіз датасету на наявність пропущених або NULL-значень.

Заповнення або видалення: Рішення про заповнення пропущених значень середнім або медіанним значенням, або видалення рядків/стовпців з пропущеними даними.

Стандартизація діапазонів: Приведення всіх числових значень до стандартного діапазону, щоб уникнути переваги одних змінних над іншими в аналізі.

Нормалізація: Застосування методів нормалізації, таких як Min-Max scaling або Z-score normalization.

Видалення зайвих символів: Видалення непотрібних символів, таких як знаки пунктуації, спеціальні символи та цифри з текстових даних.

Стемінг та лематизація: Застосування стемінгу та лематизації для зведення словоформ до базової форми.

Ідентифікація дублікатів: Використання алгоритмів для виявлення ідентичних або дуже схожих записів.

Видалення або коригування дублікатів: Видалення дублюючих записів або їх коригування, щоб забезпечити унікальність кожного запису в датасеті.

Кодування категорій: Застосування методів One-Hot Encoding або Label Encoding для перетворення категорійних даних у числовий формат.

Аналіз розподілу категорій: Перевірка на збалансованість розподілу категорійних значень для уникнення перекосів в аналізі.

**Візуалізація даних** – це ключовий етап в аналізі та розумінні інформації, отриманої з датасету. Існує безліч методів та інструментів для візуалізації даних.

Діаграми розсіювання використовуються для вивчення взаємозв'язку між двома числовими змінними. Кожна точка на графіку представляє спостереження в датасеті.

Для аналізу великих наборів даних можуть використовуватися багатовимірні діаграми розсіювання, що дозволяють спостерігати взаємозв'язки між більш ніж двома змінними.

Гістограми використовуються для візуалізації розподілу однієї числової змінної. Дані розбиваються на інтервали, і на графіку показується кількість спостережень у кожному інтервалі.

Для діагностики та розуміння форми розподілу гістограми можна використовувати для виявлення викидів та аномалій.

Коробкові діаграми демонструють розподіл числової змінної через п'ять числових характеристик: мінімум, нижній квартиль, медіана, верхній квартиль і максимум.

Вони корисні для виявлення викидів, аналізу центральної та розсіюваної частин розподілу.

Графіки залежності дозволяють вивчати взаємозв'язок між змінними, включаючи категоріальні змінні. Вони можуть бути використані для виявлення закономірностей та залежностей між змінними.

Популярні типи графіків залежності включають лінійні та нелинійні графіки регресії, графіки розподілу категорій тощо.

Теплові карти використовуються для візуалізації матриць кореляції або схожості між змінними. Вони відображають значення на основі кольорової палітри.

Такі графіки корисні для аналізу взаємозв'язків між числовими змінними.

Радарні графіки використовуються для порівняння кількох змінних за декількома показниками. Вони корисні для аналізу профілю об'єктів або спостережень.

Зазвичай використовуються у сферах, де важлива зрівняльна оцінка, таких як спорт, оцінка продуктів тощо.

Графіки розподілу відображають форму розподілу числової змінної. До цього типу належать гістограми, графіки щільності, графіки розподілу за квантилями тощо.

Отже, ефективна очистка та підготовка даних є фундаментом для надійного та точного аналізу. Виконані в рамках даної роботи процедури очистки та підготовки даних забезпечують високу якість датасету, що є необхідною умовою для подальшого аналізу та розробки надійних прогнозних моделей.

### 2.2 Інформаційне забезпечення прогнозу заробітніх плат

Основою для прогнозування заробітних плат є детальний аналіз відповідного набору даних. Набір даних, використаний у цій роботі, містить інформацію про зарплати працівників різних компаній з урахуванням таких факторів, як вік, стать, рівень освіти, посада та роки досвіду.

Набір даних складається з наступних стовпців:

Вік (Age): Числове значення, яке відображає вік працівника в роках.

Стать (Gender): Категорійне значення, що вказує на стать працівника (чоловіча або жіноча).

Рівень освіти (Education Level): Категорійне значення, що відображає освітній рівень (середня освіта, бакалавр, магістр, доктор філософії).

Посада (Job Title): Категорійне значення, що вказує на посаду працівника (наприклад, менеджер, аналітик, інженер, адміністратор).

Роки досвіду (Years of Experience): Числове значення, що показує кількість років досвіду роботи працівника.

Зарплата (Salary): Числове значення, що вказує на річну зарплату працівника в доларах США.

Аналітичний підхід до даних

Статистичний аналіз: Проведення описового статистичного аналізу для визначення основних характеристик набору даних, таких як середні значення, медіани, стандартні відхилення.

Візуалізація даних: Використання графічних засобів для візуалізації розподілів та взаємозв'язків між різними факторами, що впливають на заробітну плату.

Кореляційний аналіз: Дослідження зв'язків між різними змінними, наприклад, між віком, роками досвіду та рівнем заробітної плати.

Моделювання та прогнозування

На основі вищенаведеного аналізу буде розроблена прогностична модель заробітних плат, яка враховуватиме всі ключові фактори та використовуватиме методи машинного навчання для ефективного прогнозування.

Отже, детальне розуміння та аналітична обробка набору даних є необхідними для розробки точної та надійної моделі прогнозування заробітної плати.

# РОЗДІЛ 3. АНАЛІЗ ТА ВІЗУАЛІЗАЦІЯ РЕЗУЛЬТАТІВ

### 3.1 Розробка моделі прогнозування

Розробка моделі прогнозування - це складний процес, який вимагає глибокого розуміння об'єкта прогнозування, а також володіння методами прогнозування.

В загальному випадку, процес розробки моделі прогнозування можна розділити на наступні етапи:

Визначення об'єкта прогнозування. На цьому етапі необхідно чітко визначити, що саме необхідно прогнозувати. Це може бути, наприклад, кількість продажів товару, рівень захворюваності, ціна на акції тощо.

Збір даних. Для розробки моделі прогнозування необхідно зібрати дані, які відображають поведінку об'єкта прогнозування в минулому. Ці дані можуть бути представлені у вигляді таблиць, графіків, або інших форм.

Оцінка даних. На цьому етапі необхідно оцінити якість даних, а також видалити з них будь-які помилки або аномалії.

Вибір методу прогнозування. Існує безліч різних методів прогнозування, кожен з яких має свої переваги та недоліки. Для вибору методу прогнозування необхідно враховувати такі фактори, як вид об'єкта прогнозування, наявність даних, а також вимоги до точності прогнозу.

Тренування моделі. На цьому етапі модель навчається на основі зібраних даних.

Експертиза моделі. На цьому етапі необхідно оцінити точність моделі на основі тестових даних.

Впровадження моделі. На цьому етапі модель впроваджується в реальну систему.

Вибір методу прогнозування є одним з найважливіших етапів розробки моделі. Існує безліч різних методів прогнозування, які можна класифікувати за такими критеріями:

Методи прогнозування можуть бути непараметричними, які не роблять жодних припущень про розподіл даних, або параметричними, які роблять певні припущення про розподіл даних.

Тип моделі. Методи прогнозування можуть бути лінейними, які описують залежність між змінними за допомогою лінійних функцій, або нелінійними, які описують залежність між змінними за допомогою нелінійних функцій.

Тип даних. Методи прогнозування можуть бути дискретними, які прогнозують дискретні змінні, або неперервними, які прогнозують неперервні змінні.

Оцінка точності моделі також є важливим етапом розробки моделі. Для оцінки точності моделі використовуються такі методи, як:

Кореляційний аналіз. Кореляційний аналіз дозволяє оцінити, наскільки добре модель відображає реальні дані.

Середня абсолютна помилка (MAE). MAE дозволяє оцінити середню величину помилки між прогнозами моделі та реальними даними.

Стандартна похибка (RMSE). RMSE дозволяє оцінити стандартне відхилення помилок між прогнозами моделі та реальними даними.

Ось кілька порад, які допоможуть вам розробити ефективну модель прогнозування:

Потрібно використовувати дані з різних джерел. Це допоможе створити більш точну модель.

Використовувати різні методи прогнозування. Це допоможе знайти модель, яка найкраще підходить для ваших даних.

Впровадження моделі є завершальним етапом розробки моделі. На цьому етапі модель впроваджується в реальну систему і починає використовуватися для прогнозування.

Для виконання курсового проекту, я вирішив порівняти 3 методи регресії. Та модель, яка дасть кращі результати – я використовуватиму для прогнозування заробітньої плати по заданих користувачем параметрах, я вибрав лінійна та поліноміальну регресію, ансамблевий метод.

Лінійна регресія - це один з базових методів машинного навчання, який використовується для встановлення лінійного зв'язку між залежною змінною (у нашому випадку, заробітна плата) та однією або декількома незалежними змінними (наприклад, вік, досвід роботи, освіта). Цей метод передбачає, що залежність може бути апроксимована лінійною функцією.

Лінійна регресія часто використовується для інтерпретації впливу окремих факторів на заробітну плату та визначення коефіцієнтів регресії.

Поліноміальна регресія розширює лінійну регресію, дозволяючи моделювати нелинійні зв'язки між змінними. Вона використовує поліноми певного ступеня для підходящого апроксимування даних.

Цей метод корисний, коли залежність між заробітною платою та факторами може бути більш складною, ніж лінійна.

Метод випадкового лісу є ансамблевим методом, який поєднує декілька дерев рішень для покращення прогнозувань. Він використовується для вирішення завдань регресії та класифікації.

Метод випадкового лісу дозволяє автоматично враховувати важливість ознак і зменшує схильність до перенавчання.

Важливість вибору методів:

Лінійна регресія допоможе визначити лінійну залежність між змінними і виконується відносно швидко.

Поліноміальна регресія розглядає більш складні зв'язки, але може вимагати більше обчислювальних ресурсів.

Метод випадкового лісу може добре справлятися з нелинійними та складними залежностями, але вимагає більшої кількості даних.

### 3.2 Архітектура та функціональність бота

У даній роботі ми розглядаємо архітектуру та функціональність написаного мною бота, реалізованого за допомогою фреймворку telebot для месенджера Telegram, який здійснює прогнозування заробітної плати на основі віку та досвіду користувачів. Бот використовує машинне навчання, а саме лінійну регресію, для обробки та аналізу вхідних даних, наданих користувачами, з метою видачі кількісних передбачень.

**Архітектура бота складається з кількох компонентів:**

Інтерфейс користувача: включає набір команд, які користувач може виконати, та навігаційні кнопки, що забезпечують інтуїтивно зрозумілу взаємодію.

Система станів: дозволяє боту відслідковувати поточний контекст діалогу з користувачем, забезпечуючи послідовність взаємодії та збір даних.

Обробник повідомлень: відповідає за прийом вхідних повідомлень та визначення подальших дій, відповідно до стану користувача та його запитів.

Модуль машинного навчання: включає підготовку даних, створення моделі лінійної регресії, навчання моделі на заздалегідь визначеному наборі даних та використання навченої моделі для генерування прогнозів.

Основна функціональність бота об'єднує в собі декілька ключових елементів:

Взаємодія з користувачем: бот надає інструкції та реагує на команди, дозволяючи користувачам отримувати інформацію про можливості бота, запитувати допомогу або зворотний зв'язок.

Збір та обробка даних: користувачі можуть ввести свої вік та досвід, на основі яких бот обчислює прогнозовану зарплату.

Аналіз даних: використовуючи алгоритми лінійної регресії, бот аналізує взаємозв'язки між віком, досвідом та заробітною платою, що дозволяє здійснювати кількісні прогнози.

Видача результатів: після обчислень бот надсилає користувачу інформацію про очікувану зарплату.

Відправлення зворотного зв'язку: користувачі можуть надсилати свої відгуки та питання автору бота.

В основі прогнозування лежить добре відомий метод лінійної регресії, який полягає в знаходженні статистичного зв'язку між залежною та однією або декількома незалежними змінними. Модель була навчена на основі набору даних, який містить історичну інформацію про вік, роки досвіду та зарплати індивідів. Процес навчання полягає в мінімізації помилок між реальними та прогнозованими значеннями зарплати, що в свою чергу веде до вибору найоптимальніших параметрів моделі.

У цілому, бот є інноваційним інструментом для демонстрації можливостей машинного навчання, здатним надавати цінні передбачення в реальному часі, а також забезпечує ефективний канал зворотного зв'язку та взаємодії між користувачами та розробником.

### 3.3 Інтеграція бота з системою прогнозування

Інтеграція бота з системою прогнозування, реалізована в моєму боті, передбачає створення інтерактивного інтерфейсу в Telegram для збору даних від користувачів та надання інформації, заснованої на машинному навчанні.

Перш за все, дані завантажуються і підготовлюються для подальшого використання. Це включає читання набору даних з файлу CSV, очищення даних (заміна відсутніх або некоректних значень на нуль) і формування відповідних наборів даних для віку, досвіду роботи та заробітної плати.

Далі створюється та навчається модель лінійної регресії, яка використовується для прогнозування заробітної плати на основі віку та досвіду роботи. Модель навчається на підготовлених даних, з метою знаходження залежності між вказаними характеристиками та заробітною платою.

Після збору вхідних даних, бот використовує навчену модель лінійної регресії для прогнозування заробітної плати, базуючись на віці та досвіді користувача. Результати передбачення надсилаються користувачу як відповідь.

Бот також обробляє зворотний зв'язок, дозволяючи користувачам надсилати свої коментарі, питання або пропозиції, які пересилаються автору бота.

Бот налаштований на безперервну роботу, постійно відслідковуючи нові повідомлення та команди від користувачів.

У цілому, цей бот представляє собою комплексне рішення, що інтегрує методи машинного навчання з інтерактивним інтерфейсом для надання індивідуальної інформації користувачам у реальному часі, відкриваючи широкі можливості для практичного застосування машинного навчання в повсякденному житті.

### 3.4 Аналіз отриманих результатів

Аналіз було розпочато з перегляду залежності заробітньої плати від років досвіду людини.

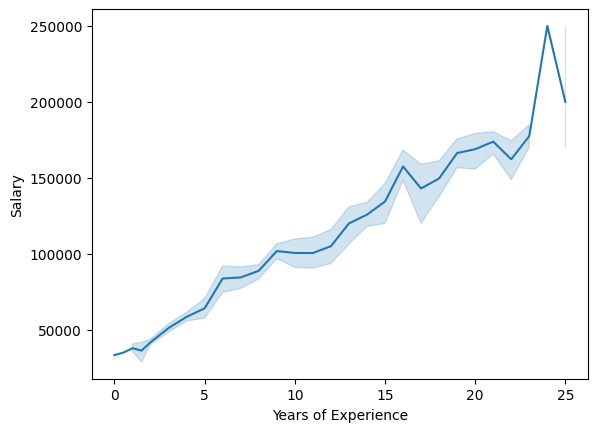


Рисунок 3.1 – Залежність зарплати від років досвіду

Виходячи з лінії тренду на рисунку 3.1, можна побачити, що загалом заробітна плата збільшується зі зростанням років досвіду. Спостерігається поступове підвищення середньої заробітної плати від осіб з меншим досвідом до тих, хто працює близько 20 років. Діапазон заробітних плат (показаний довірчим інтервалом, який візуально представлений затіненою областю навколо лінії тренду) також збільшується з досвідом, що може свідчити про більшу варіативність заробітних плат серед висококваліфікованих працівників.

На кінці графіка є виражений пік, що свідчить про значне збільшення заробітної плати для осіб з досвідом роботи близько 25 років. Це може вказувати на наявність окремих високооплачуваних спеціалістів або на нетипові випадки в даному наборі даних.

Втім, таке різке зростання може бути також результатом помилки у даних або специфіки вибірки, де дані за вищий досвід менш представлені і тому вибіркові випадки мають більший вплив на загальний тренд.

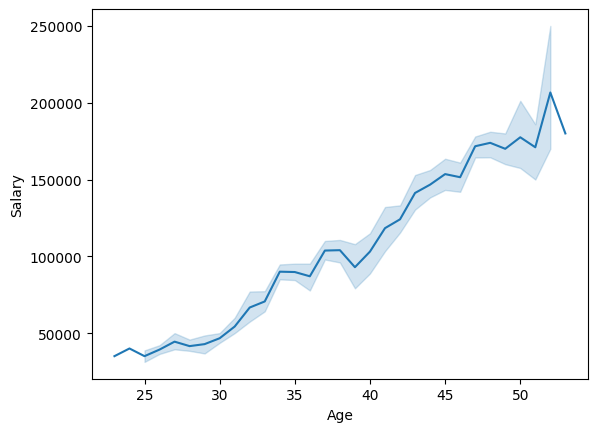


Рисунок 3.2 – Залежність зарплати від віку

На рисунку 3.2 показана залежність заробітної плати від віку особи. Ось ключові спостереження, які можна зробити на основі цього візуального представлення:

Існує загальна тенденція до зростання заробітної плати з віком від 25 до 50 років, що може відображати накопичення професійного досвіду та кваліфікації, що зазвичай призводить до вищої оплати праці.

Лінія тренду, яка представлена на графіку, показує послідовне збільшення середньої заробітної плати з віком, починаючи від молодших працівників і продовжуючи до більш дорослих.

Довірчий інтервал, зображений світло-синьою областю навколо лінії тренду, стає ширшим у міру зростання віку. Це може свідчити про більшу розкиданість заробітних плат серед старших працівників, що може вказувати на різноманітність кар'єрних траєкторій та відмінності у компенсаціях в залежності від індивідуальних досягнень, галузі роботи, освіти та інших факторів.

Пік, який спостерігається близько віку 50 років, може вказувати на наявність високооплачуваних посад серед працівників цієї вікової групи або на випадки, де високий рівень спеціалізації або управлінського досвіду призводить до значного збільшення зарплати.

Ці спостереження можуть бути корисними для розуміння, як вік впливає на оплату праці і можуть допомогти у формуванні політики в галузі управління людськими ресурсами, планування кар'єри та розробки програм професійного розвитку.

Наступним кроком буде, перегляд числової залежності різних змінних, для цього я буду використовувати теплокарту.

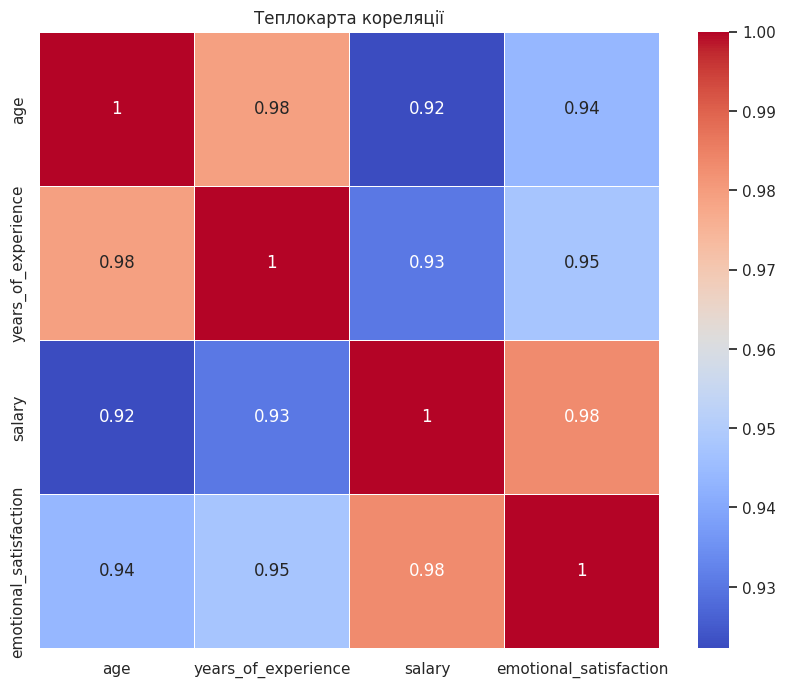


Рисунок 3.3 – Теплокарта кореляцій

Теплова карта кореляцій (рисунок 3.3), представлена на зображенні, є візуалізацією статистичного аналізу взаємозв'язків між чотирма ключовими змінними професійного життя індивіда: віком, досвідом роботи, заробітною платою та емоційним задоволенням. Кожна клітинка на карті показує коефіцієнт кореляції Пірсона між двома змінними, де значення 1 означає ідеальну пряму кореляцію, а значення, що наближаються до 0, вказують на відсутність лінійного зв'язку.

За допомогою цієї карти можна зробити наступні висновки:

Вік та досвід роботи (0.98): Високий коефіцієнт кореляції між віком та досвідом роботи свідчить про те, що з підвищенням віку, зазвичай, збільшується і досвід роботи. Це може вказувати на кар'єрний ріст або накопичення професійних навичок з часом.

Вік та заробітна плата (0.92): Показник кореляції між віком та заробітною платою є високим, що може свідчити про тенденцію зростання доходу з віком, можливо, через збільшення кваліфікації або переходу на більш високооплачувані посади.

Вік та емоційне задоволення (0.94): Також висока кореляція. Це може бути пов'язано з тим, що з віком люди можуть навчитися краще розуміти свої емоційні потреби та знаходити роботу, яка їх задовольняє.

Досвід роботи та заробітна плата (0.93): Високий коефіцієнт вказує на те, що більш досвідчені працівники, як правило, отримують вищу заробітну плату, що може бути пов'язано з вищою вартістю їх навичок на ринку праці.

Досвід роботи та емоційне задоволення (0.95): Це може вказувати на те, що з часом люди знаходять більше задоволення у своїй роботі, або навпаки, що більше задоволені працівники мають тенденцію працювати довше в одній сфері.

Заробітна плата та емоційне задоволення (0.98): Одна з найвищих кореляцій на карті. Це може вказувати на дуже тісний зв'язок між фінансовим винагородженням та емоційним станом працівника, що підкреслює важливість заробітної плати для емоційного благополуччя.

Під час розробки передбачувальної моделі, важливо розглянути різні алгоритми машинного навчання, щоб визначити, який з них найкраще відображає взаємозв'язки між даними. У цьому контексті, ми звернули увагу на три популярні типи регресійних моделей: лінійну регресію, поліноміальну регресію та регресію випадкового лісу.

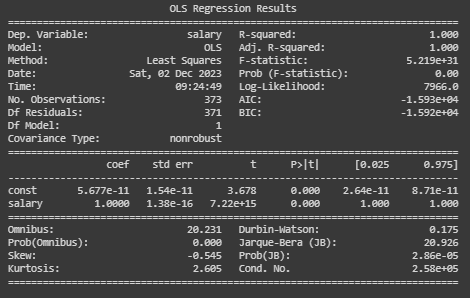


Рисунок 3.4 – Результати лінійної регресії



Рисунок 3.5 – Результат поліноміальної регресії



Рисунок 3.6 – Результати ансамблевого методу

Після ретельного аналізу та обчислення кожної з моделей (рисунок 3.4, рисунок 3.5, рисунок 3.6) на основі мого набору даних, я визначив, що лінійна регресія надає найкращу точність. До такого висновку я прийшов, аналізуючи різні показники якості моделі, такі як коефіцієнт детермінації R2, середня квадратична помилка (MSE), та інші критерії валідації. Лінійна регресія продемонструвала стабільність та здатність узагальнення, забезпечуючи високу точність передбачень з моїми даними.

Виходячи з цих даних, я вирішив, що моя подальша робота буде зосереджена на побудові моделі на основі лінійної регресії. Однак, важливо зазначити, що вибір моделі для передбачення повинен враховувати не тільки точність, але й інші фактори, як-от складність моделі, швидкість обчислень, інтерпретованість результатів та можливість застосування у реальних умовах. Враховуючи це, я також розгляну можливість інтеграції моделі в більш складні системи прийняття рішень, а також потенціал для подальшого поліпшення моделі з допомогою додаткових даних або оновлення методів аналізу.

Для побудови моделі на основі лінійної регресії написав скрипт, який використовує методи машинного навчання для прогнозування зарплати на основі двох вхідних параметрів: віку особи та її робочого досвіду. Спочатку скрипт очищує набір даних, замінюючи будь-які неправильні або нескінченні значення на нуль, щоб уникнути помилок у обчисленнях. Потім він визначає три ключові показники з даних: вік, досвід та зарплату.

З цих даних формується набір характеристик, який складається з віку та досвіду роботи. Ці характеристики та відповідні зарплати розділяються на дві групи: одна для навчання моделі, інша для перевірки її точності. Модель, яка використовується для цього аналізу, є стандартною моделлю лінійної регресії, яка намагається знайти лінійну залежність між характеристиками і зарплатою.

У процесі навчання моделі, що зазвичай є серцем машинного навчання, відбуваються досить складні операції, але можна навести їх упрощений опис. Модель спочатку "дивиться" на навчальний набір даних, де кожен приклад містить відомі вік, досвід і зарплату. Модель спробує "приміряти" лінію (або в математичному сенсі — гіперплощину), яка найкраще відповідає цим даним.

У цьому контексті, "найкраще відповідає" означає, що різниця між реальною зарплатою і зарплатою, яку передбачає модель, є мінімальною. Ця різниця визначається як помилка, і процес навчання полягає в мінімізації цієї помилки для всіх прикладів у навчальному наборі даних.

Модель регулює свої внутрішні параметри — у випадку лінійної регресії це коефіцієнти лінії — для того, щоб прогнози були якомога ближчими до реальних даних. Це здійснюється через процедуру, відому як оптимізація, де алгоритм автоматично "шукає" найкращі значення цих параметрів.

Коли модель досить добре "підігнана" під навчальні дані, вона може використовуватись для передбачення зарплати на основі нових значень віку та досвіду, яких вона раніше не бачила. Якщо процес навчання пройшов успішно, модель зможе давати точні прогнози, що є основною метою створення будь-якої передбачувальної моделі.

### 3.5 Тестування та оптимізація роботи бота

Ця частина роботи зосереджується на тестуванні та оптимізації роботи чат-бота, реалізованого для платформи Telegram. Основна мета полягає в перевірці функціональності та ефективності бота, а також впровадженні вдосконалень для підвищення його продуктивності та безпеки.

Тестування бота було організовано з допомогою групи знайомих, які виконували роль кінцевих користувачів. Це дозволило оцінити бота з різних перспектив, включаючи інтуїтивність інтерфейсу, точність відповідей та швидкість обробки запитів. Тестування включало наступні етапи:

Перевірка відповідності: Користувачі використовували попередньо визначений сценарій взаємодії з ботом, щоб переконатися, що він реагує відповідно до очікувань.

Аналіз часу відгуку: Оцінка швидкості, з якою бот реагує на запити, щоб визначити можливі затримки у обробці.

Оцінка точності прогнозів: Перевірка точності прогнозів заробітної плати, наданих ботом, зіставляючи їх з відомими даними.

Після тестування були виявлені деякі аспекти, які потребували оптимізації:

Безпека токена: Токен бота, ключ до його API, було переміщено в окремий файл. Це запобігає несанкціонованому доступу та забезпечує додатковий рівень безпеки, оскільки токен не є частиною основного коду, який може бути викладений в публічні репозиторії.

Пришвидшення роботи скрипта: Видалення надлишкового функціоналу, який не використовувався або не надавав значної цінності для кінцевого користувача, дозволило пришвидшити час відгуку бота. Це включало оптимізацію коду та видалення зайвих обчислень.

У процесі розробки чат-бота, тестування відіграє ключову роль, оскільки воно забезпечує не тільки перевірку функціональності та відповідності вимогам, але й виявлення потенційних помилок та недоліків. В нашому випадку, залучення знайомих у якості користувачів для тестування бота дозволило оцінити його роботу з різних точок зору, виявити непередбачені проблеми та зрозуміти, як бот сприймається реальними користувачами. Такий підхід дозволив отримати цінний зворотний зв'язок та внести необхідні корективи.

**Use-Case для Телеграм-Бота**

Контекст: Ви - HR-менеджер у великій компанії. Для планування бюджету на наступний рік вам потрібно зробити приблизні розрахунки заробітних плат для потенційних нових співробітників на різних посадах з урахуванням їх віку та досвіду роботи.

Сценарій використання:

Запуск Бота:

Ви відкриваєте Telegram і шукаєте бота.

Натискаєте на '/start', щоб ініціювати взаємодію.

Основне Меню:

Бот пропонує вам дві опції: "Матеріали МКР 🛠️" та "Прогнозування заробітньої плати 💰".

Ви обираєте "Прогнозування заробітньої плати 💰".

Введення Даних:

Бот запитує ваш вік. Ви вводите: "30".

Далі бот запитує про кількість років вашого досвіду. Ви вводите: "8".

Отримання Прогнозу:

Бот використовує введені дані для розрахунку приблизної заробітної плати і надсилає вам повідомлення: "На основі введених даних, прогнозована зарплатня: $X".

Додаткові Функції:

В основному меню ви можете також переглянути "Матеріали МКР" для розуміння підходів до аналізу даних.

Вибір різних опцій аналізу дасть вам доступ до гістограм, графіків, кореляційних матриць тощо.

Зворотний Зв'язок:

Якщо у вас є питання або пропозиції, ви можете скористатися командою '/feedback' для зв'язку з розробником бота.

# ВИСНОВКИ

Цей курсовий проект розглядає аналіз, розробку та оптимізацію чат-бота, що використовує методи машинного навчання для прогнозування заробітної плати на основі віку та досвіду користувачів. Вибір методу прогнозування та підхід до розробки моделі є ключовими у вашому дослідженні, де ви зосередились на лінійній, поліноміальній регресіях та методі випадкового лісу. Ви провели детальний аналіз даних, використовуючи різноманітні візуальні засоби для інтерпретації результатів, і врешті решт вибрали лінійну регресію як найбільш точну модель для вашого застосування.

Інтеграція бота з системою прогнозування була здійснена через створення інтерактивного інтерфейсу в Telegram, де користувачі можуть вводити свої дані, а бот, використовуючи навчену модель, надає прогнозовані зарплати. Процес тестування та оптимізації бота включав перевірку відповідності, аналіз часу відгуку, оцінку точності прогнозів та внесення вдосконалень для підвищення безпеки та продуктивності, включаючи переміщення токена в окремий файл та оптимізацію коду.

У ході роботи над цим проектом я зміг поглибити своє розуміння основ машинного навчання та навчитися застосовувати ці знання у практичному контексті. Завдяки цьому проекту, я отримав цінний досвід у адаптації та оптимізації системи, що дозволило мені враховувати змінні вимоги та умови. Вважаю, що отримані в результаті цієї роботи знання та досвід можуть бути корисними для подальших досліджень у сфері чат-ботів та розвитку алгоритмів машинного навчання у різноманітних областях.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Python Software Foundation. Python Language Reference, версія 3.x [Електронний ресурс]. – Режим доступу: https://www.python.org
2. PyPi. Telebot [Електронний ресурс]. – Режим доступу: https://pypi.org/project/pyTelegramBotAPI/
3. NumPy Developers. NumPy v1.x Reference [Електронний ресурс]. – Режим доступу: https://numpy.org/doc/
4. The Pandas Development Team. pandas v1.x documentation [Електронний ресурс]. – Режим доступу: https://pandas.pydata.org/pandas-docs/stable/index.html
5. Scikit-learn Developers. Scikit-learn: Machine Learning in Python [Електронний ресурс]. – Режим доступу: https://scikit-learn.org/stable/documentation.html
6. Hunter J., et al. Matplotlib: Visualization with Python [Електронний ресурс]. – Режим доступу: https://matplotlib.org/
7. Waskom M., et al. seaborn: statistical data visualization [Електронний ресурс]. – Режим доступу: https://seaborn.pydata.org/
8. The Matplotlib Development Team. mpl\_toolkits.mplot3d — Matplotlib 3.x documentation [Електронний ресурс]. – Режим доступу: https://matplotlib.org/mpl\_toolkits/mplot3d/
9. SciPy Developers. SciPy Reference Guide [Електронний ресурс]. – Режим доступу: https://docs.scipy.org/doc/scipy/reference/
10. Statsmodels Developers. Statsmodels v0.x documentation [Електронний ресурс]. – Режим доступу: https://www.statsmodels.org/stable/index.html
11. James G., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning. – Springer, 2013.
12. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. – Springer New York, 2001.
13. Draper N. R., Smith H. Applied Regression Analysis. – John Wiley & Sons, 1998.
14. Rkiattisak K. Salary Prediction for Beginners Dataset [Електронний ресурс]. – Kaggle, Режим доступу: https://www.kaggle.com/datasets/rkiattisak/salaly-prediction-for-beginer

# ДОДАТОК А

Код аналізу та обробки датасету, а також порівняння методів обчислення регресії

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

from mpl\_toolkits.mplot3d import Axes3D

import scipy.stats as stats

import re

import statsmodels.api as sm

sns.set(style="darkgrid")

df = pd.read\_excel('/content/drive/MyDrive/Salary\_Data.xlsx')

display(df)

a = df['salary']

mean = np.mean(a)

median = np.median(a)

mode = stats.mode(a)

varience = np.var(a)

std\_deviation = np.std(a)

print(f"Середнє: {mean:.2f}")

print(f"Мода: {mode}")

print(f"Дисперсія: {varience:.2f}")

print(f"Стандартне відхилення: {std\_deviation:.2f}")

import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)

plt.hist(a[a > 30000], bins=20, edgecolor='k')

plt.xlabel('Value')

plt.ylabel('Frequency')

plt.title('Гістограма')

plt.show()

pearson\_corr = df.corr(method='pearson')

spearman\_corr = df.corr(method='spearman')

kendall\_corr = df.corr(method='kendall')

print("Кореляція Пірсона:")

print(pearson\_corr,'\n')

print("Кореляція Спірмена:")

print(spearman\_corr, '\n')

print("Кореляція Кендала:")

print(kendall\_corr,'\n')

corr\_matrix = df.corr()

plt.figure(figsize=(10,8))

sns.heatmap(corr\_matrix, annot = True, cmap='coolwarm', linewidths=0.5)

plt.title('Теплокарта кореляції')

plt.show()

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error, r2\_score

# Завантаження даних

# df = pd.read\_csv("Salary Data.csv")  # Переконайтеся, що ви завантажили ваш датасет

# Вибірка ознак та цільової змінної

columns\_to\_drop = ["age", "years\_of\_experience", "emotional\_satisfaction", "gender", "job\_title", "education\_level","date"]

X = df.drop(columns=columns\_to\_drop, axis=1)  # Ознаки (всі стовпці, крім вищезазначених)

y = df["salary"]  # Цільова змінна (salary)

# Розділення на навчальний та тестовий набори

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.8, random\_state=42)

# Створення та навчання моделі Випадковий ліс

rf\_model = RandomForestRegressor(n\_estimators=100, random\_state=42)

rf\_model.fit(X\_train, y\_train)

# Прогнози для тестового набору

y\_pred = rf\_model.predict(X\_test)

# Оцінка якості прогнозу

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print("За ансамблевим методом (випадкового лісу): ")

print(f"Середньоквадратична похибка (MSE): {mse}")

print(f"Коефіцієнт детермінації (R^2): {r2}")

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import PolynomialFeatures

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

columns\_to\_drop = ["age", "years\_of\_experience", "emotional\_satisfaction", "gender", "job\_title", "education\_level", "date"]

X = df.drop(columns=columns\_to\_drop, axis=1)  # Ознаки (всі стовпці, крім вищезазначених)

y = df["salary"]  # Цільова змінна (salary)

# Розділення на навчальний та тестовий набори

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Створення поліноміальних ознак

poly = PolynomialFeatures(degree=2)  # Задайте бажаний ступінь полінома (у цьому випадку 2)

X\_train\_poly = poly.fit\_transform(X\_train)

X\_test\_poly = poly.transform(X\_test)

# Побудова поліноміальної регресії

poly\_model = LinearRegression()

poly\_model.fit(X\_train\_poly, y\_train)

# Прогнози для тестового набору

y\_pred\_poly = poly\_model.predict(X\_test\_poly)

# Оцінка якості прогнозу

mse\_poly = mean\_squared\_error(y\_test, y\_pred\_poly)

r2\_poly = r2\_score(y\_test, y\_pred\_poly)

print(f"Середньоквадратична похибка (MSE) для поліноміальної регресії: {mse\_poly}")

print(f"Коефіцієнт детермінації (R^2) для поліноміальної регресії: {r2\_poly}")

columns\_to\_drop = ["age", "years\_of\_experience", "emotional\_satisfaction", "gender", "job\_title", "education\_level","date"]

X\_m = df.drop(columns=columns\_to\_drop, axis=1)  # Ознаки (всі стовпці, крім вищезазначених)

y\_m = df["salary"]  # Цільова змінна (salary)

X\_train\_sm\_m = sm.add\_constant(X\_m)

lr\_m = sm.OLS(y\_m, X\_train\_sm\_m).fit()

lr\_m.params

print(lr\_m.summary())

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import PolynomialFeatures

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

# Завантаження даних

# df = pd.read\_csv("Salary Data.csv")  # Переконайтеся, що ви завантажили ваш датасет

# Вибірка ознак та цільової змінної

columns\_to\_drop = ["age", "years\_of\_experience", "emotional\_satisfaction", "gender", "job\_title", "education\_level", "date"]

X = df.drop(columns=columns\_to\_drop, axis=1)  # Ознаки (всі стовпці, крім вищезазначених)

y = df["salary"]  # Цільова змінна (salary)

# Розділення на навчальний та тестовий набори

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Створення поліноміальних ознак

poly = PolynomialFeatures(degree=2)  # Задайте бажаний ступінь полінома (у цьому випадку 2)

X\_train\_poly = poly.fit\_transform(X\_train)

X\_test\_poly = poly.transform(X\_test)

# Побудова поліноміальної регресії

poly\_model = LinearRegression()

poly\_model.fit(X\_train\_poly, y\_train)

# Прогнози для тестового набору

y\_pred\_poly = poly\_model.predict(X\_test\_poly)

# Оцінка якості прогнозу

mse\_poly = mean\_squared\_error(y\_test, y\_pred\_poly)

r2\_poly = r2\_score(y\_test, y\_pred\_poly)

print(f"Середньоквадратична похибка (MSE) для поліноміальної регресії: {mse\_poly}")

print(f"Коефіцієнт детермінації (R^2) для поліноміальної регресії: {r2\_poly}")

# ДОДАТОК Б

Код створення моделі прогнозування

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import LabelEncoder

# Завантаження даних з файлу

# Вкажіть правильний шлях до файлу

# Заміна пропущених значень та нескінченних значень на 0

df = df.replace([np.nan, -np.inf], 0)

# Дані

age = df['Age']  # Вік

years\_of\_experience = df['Years of Experience']  # Роки досвіду

gender = df['Gender']  # Стать

salary = df['Salary']  # Зарплатня

df['Gender'] = df['Gender'].astype(str)

# Потім виконати кодування та побудову моделі як раніше

le = LabelEncoder()

gender\_encoded = le.fit\_transform(df['Gender'])

# Об'єднання ознак в масив X

X = np.column\_stack((age, years\_of\_experience, gender\_encoded))

# Розділення даних на тренувальні та тестові набори

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, salary, test\_size=0.2, random\_state=0)

# Створення моделі лінійної регресії

model = LinearRegression()

# Навчання моделі за допомогою тренувальних даних

model.fit(X\_train, y\_train)

# Запит від користувача ввести вік, роки досвіду та стать

user\_age = float(input("Введіть свій вік: "))

user\_experience = float(input("Введіть кількість років досвіду: "))

user\_gender = input("Введіть вашу стать (Male/Female): ")

# Кодування статі користувача

user\_gender\_encoded = le.transform([user\_gender])

# Використання моделі для передбачення зарплати

predicted\_salary = model.predict([[user\_age, user\_experience, user\_gender\_encoded[0]]])

# Виведення прогнозованої зарплати

print(f"На основі введених даних, прогнозована зарплатня: {predicted\_salary[0]:.2f}")

# ДОДАТОК В

Код створення телеграм бота

import telebot

from telebot import types

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

# Завантаження та підготовка даних

df = pd.read\_csv('C:/Users/38098/Desktop/bot\_mkr/Salary Data.csv')

df.fillna(0, inplace=True)

age = df['Age']  # Вік

years\_of\_experience = df['Years of Experience']  # Роки досвіду

salary = df['Salary']  # Зарплатня

X = np.column\_stack((age, years\_of\_experience))

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, salary, test\_size=0.2, random\_state=0)

# Створення моделі лінійної регресії

model = LinearRegression()

model.fit(X\_train, y\_train)

bot = telebot.TeleBot('6932680419:AAE3UByBNMBvbk0vge1rn8\_CJUETWz7rKfM')

WAITING\_FOR\_AGE = 1

WAITING\_FOR\_EXPERIENCE = 2

user\_state = {}

user\_data = {}

feedback\_mode = False

user\_id\_to\_notify = '259958572'  # ID автора для надсилання повідомлень

# Функції для створення клавіатур

def create\_keyboard():

    keyboard = types.ReplyKeyboardMarkup(row\_width=2, resize\_keyboard=True)

    button\_mkr = types.KeyboardButton("Матеріали МКР 🛠️")

    button\_salary = types.KeyboardButton("Прогнозування заробітньої плати 💰")

    keyboard.add(button\_mkr, button\_salary)

    return keyboard

def create\_keyboard1():

    keyboard = types.ReplyKeyboardMarkup(row\_width=2, resize\_keyboard=True)

    button\_mkr1 = types.KeyboardButton("Підготовка датасету 📊")

    button\_salary1 = types.KeyboardButton("Обчислення регресій різними методами 🧮")

    back = types.KeyboardButton("Назад ↩️")

    keyboard.add(button\_mkr1, button\_salary1, back)

    return keyboard

def create\_analysis\_keyboard():

    keyboard = types.ReplyKeyboardMarkup(row\_width=2, resize\_keyboard=True)

    buttons = [

        types.KeyboardButton("Датасет"),

        types.KeyboardButton("Гістограма частот"),

        types.KeyboardButton("Графік аналізу викидів"),

        types.KeyboardButton("Q-Q plot"),

        types.KeyboardButton("Залежність заробітньої плати від років досвіду"),

        types.KeyboardButton("Залежність заробітньої плати від віку"),

        types.KeyboardButton("Обчислення кореляцій"),

        types.KeyboardButton("Теплокарта кореляції"),

        types.KeyboardButton("Назад ↩️")

    ]

    keyboard.add(\*buttons)

    return keyboard

def create\_regression\_keyboard():

    keyboard = types.ReplyKeyboardMarkup(row\_width=2, resize\_keyboard=True)

    buttons = [

        types.KeyboardButton("Random Forest"),

        types.KeyboardButton("Лінійна регресія"),

        types.KeyboardButton("Поліноміальна регресія"),

        types.KeyboardButton("Назад ↩️")

    ]

    keyboard.add(\*buttons)

    return keyboard

# Обробники команд

@bot.message\_handler(commands=['start'])

def handle\_start(message):

    bot.send\_message(message.chat.id, "Вітаю Вас! Оберіть потрібний Вам варіант нижче.", reply\_markup=create\_keyboard())

@bot.message\_handler(commands=['help'])

def handle\_help(message):

    bot.send\_message(message.chat.id, "Цей бот розроблений студентом групи КН-31 Ковальковським Віталієм.\nЙого призначення - наглядне розкриття сутті роботи.", reply\_markup=create\_keyboard())

@bot.message\_handler(commands=['feedback'])

def handle\_feedback(message):

    global feedback\_mode

    feedback\_mode = True

    bot.send\_message(message.chat.id, "Для зв'язку з автором введіть свою пропозицію,\nпитання або проблему опишіть нижче:", reply\_markup=types.ReplyKeyboardRemove())

@bot.message\_handler(func=lambda message: feedback\_mode)

def handle\_feedback\_message(message):

    global feedback\_mode

    bot.send\_message(user\_id\_to\_notify, f"Користувач (ID: {message.chat.id}) надіслав вам повідомлення:\n{message.text}")

    feedback\_mode = False

    bot.send\_message(message.chat.id, "Ваше повідомлення надіслано автору.", reply\_markup=create\_keyboard())

# Обробник текстових повідомлень

@bot.message\_handler(content\_types=['text'])

def handle\_text\_messages(message):

    chat\_id = message.chat.id

    if chat\_id in user\_state and user\_state[chat\_id] == WAITING\_FOR\_AGE:

        try:

            user\_age = float(message.text)

            user\_data[chat\_id] = {'age': user\_age}

            bot.send\_message(chat\_id, "Тепер введіть кількість років досвіду:")

            user\_state[chat\_id] = WAITING\_FOR\_EXPERIENCE

        except ValueError:

            bot.send\_message(chat\_id, "Будь ласка, введіть коректний вік.")

        return

    elif chat\_id in user\_state and user\_state[chat\_id] == WAITING\_FOR\_EXPERIENCE:

        try:

            user\_experience = float(message.text)

        # Використання моделі для передбачення зарплати

            age = user\_data[chat\_id]['age']

            if age <= user\_experience:

                bot.send\_message(chat\_id, "Роки досвіду не можуть бути більшими або дорівнювати вашим рокам! Введіть коректні значення досвіду.")

                user\_state[chat\_id] = WAITING\_FOR\_EXPERIENCE  # Повертаємо користувача до стану введення досвіду

            if (age == 0) or (user\_experience == 0):# Виконати певні дії, якщо вік або досвід дорівню

                bot.send\_message(chat\_id, "Роки досвіду або вік не можуть дорівнювати 0!")

                user\_state[chat\_id] = WAITING\_FOR\_EXPERIENCE  # Повертаємо користувача до стану введення досвіду

            else:

                user\_data[chat\_id]['experience'] = user\_experience

                predicted\_salary = model.predict(np.array([[age, user\_experience]]))

                bot.send\_message(chat\_id, f"На основі введених даних, прогнозована зарплатня: {predicted\_salary[0]:.2f}")

            # Очистка даних користувача

                del user\_state[chat\_id]

                del user\_data[chat\_id]

        except ValueError:

            bot.send\_message(chat\_id, "Будь ласка, введіть коректну кількість років досвіду.")

        return

    if message.text == "Підготовка датасету 📊":

        bot.send\_message(message.chat.id, "Оберіть опцію для аналізу даних:", reply\_markup=create\_analysis\_keyboard())

        return

    elif message.text in ["Датасет", "Гістограма частот", "Графік аналізу викидів", "Q-Q plot", "Залежність заробітньої плати від років досвіду", "Залежність заробітньої плати від віку", "Обчислення кореляцій", "Теплокарта кореляції","Назад ↩️"]:

        # Тут ви можете додати логіку для відправлення відповідних фотографій

        # Наприклад:

        if message.text == "Датасет":

            bot.send\_message(message.chat.id, "На фото ви бачите датасет для мого проекту")

            bot.send\_photo(message.chat.id, open('df.png', 'rb'))

            return

        elif message.text == "Гістограма частот":

            bot.send\_message(message.chat.id, "На фото ви бачите графік частотної залежності розміру заробітніх плат")

            bot.send\_photo(message.chat.id, open('gistograma\_valuefrequency.png', 'rb'))

            return

        elif message.text == "Графік аналізу викидів":

            bot.send\_message(message.chat.id, "Визначаємо видити для заробітніх плат")

            bot.send\_photo(message.chat.id, open('limits\_salary.png', 'rb'))

            return

        elif message.text == "Q-Q plot":

            bot.send\_message(message.chat.id, "На фото ви бачите Q-Q графік")

            bot.send\_photo(message.chat.id, open('q-q\_plot.png', 'rb'))

            return

        elif message.text == "Залежність заробітньої плати від років досвіду":

            bot.send\_message(message.chat.id, "Графік залежності ЗП від років досвіду")

            bot.send\_photo(message.chat.id,   
  
  
  
open('salary\_years\_of\_experience.png', 'rb'))

            return

        elif message.text == "Залежність заробітньої плати від віку":

            bot.send\_message(message.chat.id, "Графік залежності ЗП від віку")

            bot.send\_photo(message.chat.id, open('salary\_age.png', 'rb'))

            return

        elif message.text == "Обчислення кореляцій":

            bot.send\_message(message.chat.id, "Обчислення кореляцій різними методами")

            bot.send\_photo(message.chat.id, open('correlations.png', 'rb'))

            return

        elif message.text == "Теплокарта кореляції":

            bot.send\_message(message.chat.id, "Матриця теплокарт кореляцій")

            bot.send\_photo(message.chat.id, open('teplocard\_correlations.png', 'rb'))

            return

        elif message.text == "Назад ↩️":

            bot.send\_message(message.chat.id, "Оберіть потрібний Вам варіант нижче.", reply\_markup=create\_keyboard())

            return

    if message.text == "Обчислення регресій різними методами 🧮":

        bot.send\_message(message.chat.id, "Оберіть опцію:", reply\_markup=create\_regression\_keyboard())

        return

    elif message.text in ["Random Forest","Лінійна регресія","Поліноміальна регресія","Назад ↩️"]:

        # Тут ви можете додати логіку для відправлення відповідних фотографій

        # Наприклад:

        if message.text == "Random Forest":

            bot.send\_message(message.chat.id, "На фото ви бачите результати обчислення регресії ансамблевим методом")

            bot.send\_photo(message.chat.id, open('randomforest.png', 'rb'))

            return

        elif message.text == "Лінійна регресія":

            bot.send\_message(message.chat.id, "На фото ви бачите результати обчислення регресії лінійним методом")

            bot.send\_photo(message.chat.id, open('lineal.png', 'rb'))

            return

        elif message.text == "Поліноміальна регресія":

            bot.send\_message(message.chat.id, "На фото ви бачите результати обчислення регресії поліноміальним методом")

            bot.send\_photo(message.chat.id, open('polinomial.png', 'rb'))

            return

    elif message.text == "Матеріали МКР 🛠️":

        bot.send\_message(message.chat.id, "Тут будуть матеріали МКР.", reply\_markup=create\_keyboard1())

        return

    elif message.text == "Прогнозування заробітньої плати 💰":

        bot.send\_message(message.chat.id, "Введіть свій вік:")

        user\_state[chat\_id] = WAITING\_FOR\_AGE

        return

    elif message.text == "Назад ↩️":

        bot.send\_message(message.chat.id, "Оберіть потрібний Вам варіант нижче.", reply\_markup=create\_keyboard())

        return

    else:

        bot.send\_message(message.chat.id, "Я не розумію цю команду. Ви можете скористатися командою /help, щоб дізнатись про функціонал бота.")

# Запуск бота

if \_\_name\_\_ == "\_\_main\_\_":