

ПАСЕМКО ІГОР

Національний університет «Львівська політехніка»

<https://orcid.org/0009-0000-2055-7170>e-mail: igor.s.pasemko@lpnu.ua**ХУДИЙ АНДРІЙ**

Національний університет «Львівська Політехніка»

<https://orcid.org/0000-0003-2029-7270>e-mail: Andrii.M.Khudyi@lpnu.ua

РЕКОМЕНДАЦІЙНА СИСТЕМА ДЛЯ АБІТУРІЄНТІВ НА ОСНОВІ РЕГРЕСІЙНОЇ МОДЕЛІ АНАЛІЗУ ЗАРПЛАТ В ІТ-ГАЛУЗІ

У роботі описано процес розробки та функціонування рекомендаційної системи для абітурієнтів на основі регресійної моделі аналізу зарплат в ІТ-галузі. Абітурієнти часто орієнтуються лише на особисті уподобання або загальну популярність напрямку, не маючи достатньої інформації про фінансові перспективи. Рекомендаційна система, що аналізує реальні дані про зарплати в ІТ-компаніях, допоможе усунути цей інформаційний бар'єр та сприятиме більш усвідомленому вибору навчальної програми.

Ключові слова: рекомендаційна система, регресійна модель, зарплата.

PASEMKO IHOR**KHUDYI ANDRII**

Lviv Polytechnic National University

RECOMMENDATION SYSTEM FOR APPLICANTS BASED ON A REGRESSION MODEL OF SALARY ANALYSIS IN THE IT INDUSTRY

Recommender systems are actively developing in the field of education, helping applicants choose majors based on data analysis. Modern systems use artificial intelligence, machine learning, and big data analytics to provide personalized recommendations, allowing applicants to receive advice based on current job market trends and their own preferences.

The relevance of developing a recommendation system for applicants is due to the rapid development of the IT industry and constant changes in the labor market. Modern technologies and new specializations are emerging rapidly, and the level of salaries varies significantly depending on the demand for specialists. A recommendation system that analyzes real data on salaries in IT companies will help eliminate this information barrier and promote a more informed choice of curriculum.

The object of the study is the process of choosing an educational program by an applicant, which determines his future professional prospects. The subject of the study is the use of regression models and recommendation systems for predicting salaries and selecting optimal IT programs, which will contribute to a more rational choice of educational direction in accordance with the requirements of the labor market.

The scientific novelty of the proposed approach lies in the integration of regression salary forecasting with a recommendation system for selecting educational programs. The practical value of the project under development lies in its ability to help applicants, career counselors, and educational institutions make informed decisions about choosing an educational program. The system contributes to increasing the level of employment of graduates and their satisfaction with professional development, ensuring that the education obtained meets the current demands of the labor market.

Keywords: recommender system, regression model, salary.

Стаття надійшла до редакції / Received 03.10.2025

Прийнята до друку / Accepted 15.11.2025

Постановка проблеми та її рішення

Сфера інформаційних технологій приваблює багатьох своєю динамічністю, інноваційністю та обіцянками високих заробітних плат. Проте, саме ця різноманітність та швидкість розвитку створюють значні труднощі. Майбутні фахівці стикаються з проблемою вибору конкретної спеціалізації в ІТ, визначення найбільш затребуваних мов програмування та усвідомлення важливості рівня англійської мови для майбутнього доходу. Критичним є також розуміння реального потенціалу заробітку в обраному напрямку. Відсутність чітких, об'єктивних та актуальних відповідей на ці питання може призвести до хибного вибору освітнього шляху та, як наслідок, до розчарування у професії. Саме ця проблема невизначеності та браку достовірної інформації на етапі вибору майбутнього є основним призначенням нашої рекомендаційної системи.

Призначення системи полягає у наданні майбутнім ІТ-фахівцям – абітурієнтам, студентам перших курсів та їхнім батькам – чітких, обґрунтованих прогнозів потенційної заробітної плати та персоналізованих рекомендацій щодо найперспективніших напрямків розвитку в ІТ-сфері. Система аналізує ключові параметри, такі як прогнозований стаж, майбутня посада, необхідний рівень володіння англійською мовою, основні мови програмування та географічне розташування, щоб допомогти зробити усвідомлений вибір освітньої траєкторії.

Місце застосування системи – це, перш за все, домашні умови абітурієнтів, школи, центри профорієнтації, а також приймальні комісії та відділи маркетингу університетів, які можуть використовувати ці дані для інформування потенційних студентів.

Очікувані ефекти від впровадження системи є значними. Для абітурієнтів це перш за все ухвалення обґрунтованих рішень щодо вибору спеціальності та вищого навчального закладу, що мінімізує ризик розчарування та сприяє успішному старту кар'єри. Вони отримають чітке розуміння, які навички (наприклад,

поглиблене вивчення англійської, освоєння специфічних мов програмування) варто розвивати вже зараз, щоб максимізувати майбутній дохід.

Прогнозування заробітної плати є класичною задачею регресії, яка вимагає виявлення складних, часто нелінійних залежностей між численними вхідними ознаками (такими як посада, місто, мова програмування, рівень англійської, стаж та рік) та цільовою змінною (зарплата).

Окрім числового прогнозування, розроблювана система має важливу функцію надання дієвих кар'єрних рекомендацій. Цей метод полягає в агрегації та інтерпретації результатів регресійної моделі. Процес включає генерацію гіпотетичних сценаріїв з різними комбінаціями кар'єрних факторів на майбутній рік, подальше масове прогнозування потенційних зарплат для цих сценаріїв, а також групування та усереднення прогнозів за ключовими критеріями. Застосування додаткових фільтрів, таких як мінімальна кількість вакансій для забезпечення статистичної значущості або критерії для виділення можливостей для початківців, дозволяє трансформувати сирі числові дані в зрозумілі, обґрунтовані та практичні поради. Цей підхід є цінним, оскільки перетворює складні дані в доступні інсайти, допомагаючи абітурієнтам приймати обґрунтовані рішення щодо вибору спеціалізації та розвитку навичок.

Вибір та обґрунтування засобів вирішення завдання

Основним засобом для розробки продукту було обрано мову програмування Python. Цей вибір обґрунтований її домінуючою позицією у сфері аналізу даних та машинного навчання, завдяки багатому арсеналу високопродуктивних бібліотек та фреймворків. Читабельний синтаксис Python, велика спільнота розробників та широка доступність навчальних ресурсів значно прискорили процес розробки та налагодження системи.

Для ефективної роботи з даними та їх попередньої обробки було використано бібліотеки pandas та numpy. pandas є основним інструментом для маніпуляцій з табличними даними (DataFrame), що дозволило зручно імпортувати, очищати, трансформувати та агрегувати великі обсяги інформації про вакансії. numpy, як фундаментальна бібліотека для наукових обчислень у Python, забезпечила високоефективні операції з багатовимірними масивами, які є основою для роботи більшості бібліотек машинного навчання.

Центральне місце у реалізації машинного навчання посідає бібліотека scikit-learn. Вона надала необхідний функціонал для попередньої обробки даних, зокрема, OneHotEncoder для кодування категоріальних ознак та ColumnTransformer для створення гнучких пайплайнів попередньої обробки. Крім того, scikit-learn була використана для розділення даних на тренувальні та тестові набори (train_test_split) та для обчислення ключових метрик оцінки моделі, таких як mean_absolute_error, mean_squared_error та r2_score, що є критично важливим для валідації якості прогнозів. Можливість створення комплексних пайплайнів (Pipeline) у scikit-learn дозволила інкапсулювати всі етапи обробки даних та моделювання в єдину структуру, підвищуючи модульність та зручність використання.

Безпосередньо для реалізації обраних ансамблевих моделей було залучено бібліотеки xgboost та scikit-learn (для RandomForestRegressor). Бібліотека xgboost є автономною, високооптимізованою реалізацією алгоритму Extreme Gradient Boosting, що відома своєю швидкістю, ефективністю та здатністю досягати передових результатів у задачах регресії та класифікації. RandomForestRegressor з scikit-learn забезпечив реалізацію алгоритму випадкового лісу.

Для серіалізації та десеріалізації навчених моделей було використано бібліотеку joblib. Це дозволило ефективно зберігати навчені пайплайни моделей у файл на диску та швидко завантажувати їх назад, уникнувши необхідності повторного навчання при кожному запуску програми, що є важливим для оптимізації робочого процесу та розгортання моделі.

Форматований вивід рекомендацій у консоль був реалізований за допомогою бібліотеки tabulate. Цей засіб дозволив представити згенеровані таблиці рекомендацій у чистому та візуально привабливому форматі, значно покращивши читабельність та сприйняття інформації кінцевим користувачем.

Середовище розробки та супутні бібліотеки, такі як matplotlib (для потенційної візуалізації, якщо вона використовувалася для аналізу, навіть без прямого виводу у фінальній версії), ipykernel, jupyter_client, prompt_toolkit та інші, забезпечили комфортні умови для інтерактивної розробки, відладки та виконання коду. Використання віртуального середовища (наприклад, venv) гарантувало ізоляцію залежностей проекту, запобігаючи конфліктам версій пакетів та забезпечуючи відтворюваність програмного оточення на різних системах.

У сукупності, ці програмні засоби утворили надійний та ефективний стек технологій, що дозволив успішно реалізувати поставлені завдання прогнозування заробітної плати та формування кар'єрних рекомендацій.

Опис структури бази даних

Для розробки цього продукту було використано публічно доступні дані опитувань щодо заробітних плат українських IT-спеціалістів, зібрані спільнотою DOU.ua. Дані завантажувалися з репозиторію <https://github.com/devua/csv>, охоплюючи період з травня 2016 року по грудень 2024 року. Кожне опитування зазвичай публікувалося двічі на рік (у червні та грудні/травні та грудні).

Незважаючи на цінність даних, зібраних DOU.ua, аналіз виявив низку значних невідповідностей та проблем з якістю, які є типовими для даних, зібраних через опитування. Ці проблеми потребували інтенсивної попередньої обробки для забезпечення достовірності та придатності даних для подальшого моделювання:

- неповний часовий ряд: дані були доступні не за всі місяці в період з 2016 по 2024 рік, що створювало пропуски в часовій послідовності.
- низька якість заповнення: дані містили численні "биті" або неправильно заповнені поля. Це свідчить про використання опитувальної форми, яка, ймовірно, не мала жорстких обмежень щодо типу введених даних, що призводило до забруднення інформації.

- невідповідності у форматі та мові: окремі таблиці даних були заповнені російською мовою інші – українською або з елементами змішаної мови. Також спостерігалися різні варіанти написання одних і тих же категорій (наприклад, різні назви посад, міст, рівнів англійської), що ускладнювало їх уніфікацію.

Враховуючи виявлені проблеми з якістю та форматом даних, була розроблена комплексна система функцій для їх очищення та нормалізації. Ці функції виконували критично важливі операції, такі як:

- переклад та нормалізація назв міст та стандартизацію їхнього формату, видаляючи зайві текстові елементи;

- уніфікація рівнів англійської мови: включало приведення всіх значень до єдиного реєстру, переклад з російської, а також видалення будь-яких числових префіксів чи інших допоміжних символів.

- нормалізація стажу: перетворення різноманітних текстових представлень стажу у єдиний числовий формат (років).

- уніфікація назв посад: розроблена для стандартизації назв посад, що є одним з найскладніших аспектів обробки даних опитувань. Ця функція включає видалення дужок, об'єднання інформації з полів "посада" та "тайтл" для створення більш точних і уніфікованих назв, а також обробку специфічних випадків.

З огляду на те, що формат та якість даних відрізняються від опитування до опитування (різні роки можуть мати різні назви колонок, мову заповнення, наявність специфічних аномалій), було розроблено механізм гнучкої обробки даних.

Після індивідуальної попередньої обробки кожного річного файлу, фінальним етапом підготовки даних до моделювання є їх об'єднання в єдиний консолідований набір. Цей процес автоматизовано.

Оброблений датасет є вхідною базою для навчання та валідації моделі прогнозування. Для цілей даного дослідження та побудови моделі особливий інтерес становили наступні ознаки (стовпці):

- Місто: географічне розташування респондента.
- Заробітна плата: основна цільова змінна, що підлягає прогнозуванню.
- Посада: позиція, яку обіймає респондент (наприклад, Software Engineer, QA Engineer, Project Manager).
- Стаж: досвід роботи у роках.
- Мова програмування: основна мова програмування, яку використовує респондент.
- Рівень англійської: рівень володіння англійською мовою.
- Рік: рік проведення опитування, доданий як ознака для урахування часової динаміки.

Для організації процесу обробки даних та управління версіями даних, використано структуровану файлову систему директорій.

Опис механізмів рекомендаційної системи

Система кар'єрних рекомендацій функціонує як інструмент інтерпретації числових прогнозів заробітних плат для формування конкретних, статистично обґрунтованих кар'єрних порад. Її основним завданням є перетворення складних даних, отриманих від прогностичної моделі, у зрозумілі та дієві рекомендації для абітурієнтів та спеціалістів, що прагнуть зрозуміти потенційний дохід у різних сегментах ІТ-ринку.

Формування рекомендацій відбувається у кілька ключових етапів:

Генерація майбутніх сценаріїв: для того, щоб надати рекомендації на майбутнє, система спочатку формує гіпотетичний набір даних. Це робиться шляхом копіювання існуючого, вже об'єднаного та очищеного датасету (`merged_data.csv`). Потім значення колонки рік у цьому новоствореному датасеті встановлюється на цільовий майбутній рік (наприклад, 2025). Таким чином, створюється набір сценаріїв, що представляють різні комбінації посад, міст, мов програмування, рівнів англійської та стажу, але "перенесених" у майбутній період.

Отримання числових прогнозів: до згенерованих майбутніх сценаріїв застосовується вже навчена регресійна модель (`XGBoost` або `Random Forest`, залежно від конфігурації, обраної при запуску скрипта `main_xgb.py` або `main_rf.py`). Модель обчислює прогнозовані значення заробітної плати (прогноз_зарплати) для кожної комбінації ознак у `future_df`. Ці прогнози є числовою основою для всіх подальших рекомендацій.

Передача даних до модуля формування рекомендацій (`reporting_module.py`): отриманий датафрейм з прогнозованими зарплатами передається до спеціалізованого модуля `reports/reporting_module.py`. Цей модуль містить усю логіку для агрегації, фільтрації, ранжування та форматування даних з метою генерації різних типів кар'єрних рекомендацій.

Деталізація логіки формування рекомендацій реалізує такі механізми формування рекомендацій:

- Топ-посади з найкращими комбінаціями факторів: система групує прогнозовані дані за посадою, рівнем англійської, мовою програмування та стажем, обчислюючи середню прогнозовану зарплату для кожної унікальної комбінації. Потім для кожної унікальної посади обирається комбінація, яка демонструє найвищу прогнозовану зарплату. Результати сортуються та виводяться у вигляді таблиці, показуючи найбільш прибуткові шляхи розвитку.

- Загальні висновки за посадами: обчислюється середня прогнозована зарплата для кожної посади (незалежно від інших факторів). Це дає загальне уявлення про привабливість посад на ринку праці. Результати сортуються за спаданням середньої зарплати.

- Вплив рівня англійської на зарплату: аналізується середня прогнозована зарплата для кожного рівня володіння англійською мовою. Це дозволяє наочно продемонструвати кореляцію між рівнем англійської та потенційним доходом, підкреслюючи важливість її вивчення.

- Аналіз мов програмування для технічних ліdersьких та інженерних посад: система фільтрує дані, залишаючи лише ті, що відповідають певним ключовим словам у назвах посад (наприклад, 'software engineer', 'tech lead', 'architect'). Потім для цих спеціалізованих посад аналізується середня прогнозована зарплата та кількість вакансій для кожної мови програмування. Застосовується фільтр на мінімальну кількість спостережень (наприклад, 50), щоб забезпечити статистичну значущість. Виводяться окремі таблиці для найоплачуваніших та найпопулярніших мов.

- Важливість стажу: проводиться аналіз середньої прогнозованої зарплати для різних діапазонів стажу. Це ілюструє, як досвід роботи впливає на потенційний дохід, надаючи абітурієнтам розуміння довгострокових кар'єрних перспектив.

- Найкращі можливості для початківців (стаж до 2 років): для цього типу рекомендацій застосовується спеціальна фільтрація даних: враховуються записи, де посада містить ключові слова junior, intern, trainee, або де стаж респондента становить до 2 років, при цьому виключаються посади, що містять senior, lead тощо, щоб фокусуватися саме на початковому рівні. Далі дані агрегуються за посадою, мовою програмування та рівнем англійської, обчислюються середня прогнозована зарплата та кількість спостережень. Застосовується поріг мінімальної кількості вакансій (наприклад, 15), щоб рекомендації були релевантними. Результати сортуються за середньою зарплатою.

- Формат виводу результатів: для зручності сприйняття всі згенеровані рекомендації форматовуються у вигляді таблиць за допомогою бібліотеки tabulate, які виводяться безпосередньо у консоль.

Таким чином, механізми формування рекомендацій дозволяють системі не просто видавати числові прогнози, а перетворювати їх на конкретні, зрозумілі та дієві кар'єрні поради, що базуються на аналізі ринкових тенденцій та потенціалу зростання.

Програмне забезпечення побудовано за модульним принципом, що забезпечує гнучкість, розширюваність та легкість підтримки. На високому рівні архітектура системи може бути представлена як послідовність взаємодіючих функціональних блоків:

- Блок завантаження та попередньої обробки даних: відповідає за отримання сирих даних, їх очищення, нормалізацію та консолідацію.

- Блок навчання/завантаження моделі: здійснює процес навчання прогностичної моделі на підготовлених даних або завантажує вже навчену модель з файлової системи.

- Блок прогнозування: використовує навчену модель для генерації числових прогнозів заробітних плат на заданий період.

- Блок формування рекомендацій: інтерпретує числові прогнози, перетворюючи їх на зрозумілі та структуровані кар'єрні рекомендації.

Програмний комплекс забезпечує наступний ключовий функціонал:

- Автоматизована підготовка даних.

- Гнучке керування моделями:

- Навчання моделей.

- Збереження та завантаження.

- Примусове перенавчання.

- Прогнозування заробітних плат.

- Комплексна генерація кар'єрних рекомендацій:

- Загальні висновки щодо середніх зарплат за посадами, містами, мовами програмування та стажем.

- Аналіз впливу рівня англійської мови на дохід.

- Спеціальні рекомендації для початківців, що виділяють найперспективніші напрямки з мінімальним досвідом.

Аналіз контрольного прикладу та отриманих результатів

Для успішного запуску програмного комплексу та отримання прогнозів і рекомендацій, необхідно переконатися в наявності підготованих даних. Якщо у директорії data/processed/ відсутні готові оброблені дані, потрібно виконати первинну передобробку сирих даних за допомогою скрипту scripts/prepare_data.py. Після цього, для створення єдиного консолідованого датасету (merged_data.csv), який використовуватиметься моделями, оброблені дані необхідно об'єднати за допомогою скрипту scripts/merge_data.py.

Для запуску програмного комплексу та отримання прогнозів і рекомендацій використовується головний скрипт run.py. Цей скрипт надає користувачеві можливість обрати бажану модель для виконання завдання – Random Forest або XGBoost. Вибір здійснюється шляхом введення відповідного номера в консолі.

Окрім вибору моделі, система також передбачає параметр FORCE_TRAIN (визначений у файлах main_xgb.py та main_rf.py), який контролює процес навчання моделі. Значення FORCE_TRAIN = False вказує системі на спробу завантажити вже навчену модель, якщо вона існує. Проте, оскільки даний запуск є першим

або попередня навчена модель відсутня (наприклад, через відсутність файлу моделі у директорії model/), система автоматично ініціює процес навчання обраної моделі.

```

$ python run.py
Оберіть модель для запуску:
1 - Random Forest
2 - XGBoost
Введіть номер моделі: 1
  
```

Рис. 1. Запуск run.py

Після завершення навчання, навчена модель зберігається на диск для подальшого використання, що запобігає необхідності повторного навчання при наступних запусках системи.

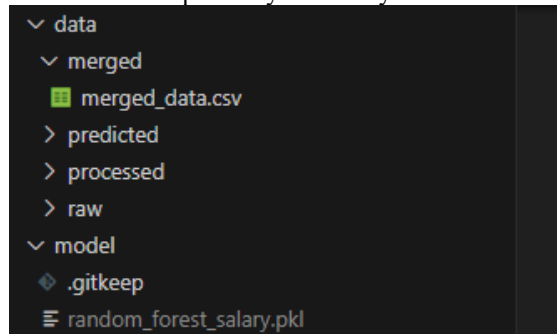


Рис. 2. Збережена модель

Після успішного завершення навчання або завантаження моделі, система переходить до етапу прогнозування заробітних плат на заданий майбутній рік (наприклад, 2025). Отримані прогнози зберігаються у файлі salary_prediction_2025_random_forest.csv або salary_prediction_2025_xgb.csv у папці data/predicted/ для подальшого аналізу та верифікації. На основі цих прогнозів генеруються детальні кар'єрні рекомендації, які виводяться у консоль у вигляді структурованих таблиць.

A1	A	B	C	D	E	F	G
1	посада	місто	стак	мова програмування	рівень англійської	рік	прогноз зарплати
2	Senior Software Engineer	Київ	5	Swift	Upper-Intermediate	2025	4191
3	Software Engineer	Київ	2	C#/.NET	Pre-Intermediate	2025	1571
4	Junior Software Engineer	Чернівці	0.5	Java	Intermediate	2025	680
5	Software Engineer	Київ	5	Java	Elementary	2025	2555
6	Junior Software Engineer	Київ	1	PHP	Pre-Intermediate	2025	779
7	Software Engineer	Дніпро	1.5	Ruby/Rails	Pre-Intermediate	2025	1140
8	System Architect	Київ	10	Java	Intermediate	2025	6051
9	Software Engineer	Київ	3	Objective-C	Intermediate	2025	2237
10	Junior Software Engineer	Київ	0	Java	Intermediate	2025	583
11	Software Engineer	Київ	3	C#/.NET	Intermediate	2025	2194
12	Software Engineer	Харків	3	Objective-C	Upper-Intermediate	2025	2049
13	Senior Software Engineer	Харків	8	PHP	Upper-Intermediate	2025	3961

Рис. 3. Збережена таблиця

В рамках контрольного запуску системи з використанням моделі Random Forest, була проведена оцінка її ефективності на тестових даних. Отримані метрики якості продемонстрували наступні значення:

Mean Absolute Error (MAE): 1107.08 USD

Root Mean Squared Error (RMSE): 1655.06 USD

R-squared (R²) Score: 0.4487

```

--- Оцінка ефективності моделі ---
Оцінка моделі на даних за 2024 рік:
Mean Absolute Error (MAE): 1107.08 USD
Root Mean Squared Error (RMSE): 1655.06 USD
R-squared (R2) Score: 0.4487

**Висновок про ефективність моделі:**
- **MAE (1107.08 USD):** В середньому, прогноз моделі відхиляється від фактичної зарплати на 1107.08 USD. Чим менше це значення, тим точніші прогнози.
- **RMSE (1655.06 USD):** Аналогічно MAE, але надає більшу вагу більшим помилкам. Значення знаходиться в тому ж масштабі, що й зарплата. Менше - краще.
- **R-squared (0.4487):** 44.87% варіації зарплати пояснюється моделлю. Це означає, що модель може пояснити 44.87% коливань зарплати на основі вхідних факторів. Чим ближче значення до 1, тим краще модель пояснює дані.
Ці метрики показують, наскільки добре модель прогнозує зарплату на даних за 2024 рік, які вона не використовувала для навчання.

Прогноз завершено. Дані збережено у data/predicted/salary_prediction_2025_random_forest.csv
  
```

Рис. 4. Оцінка ефективності моделі Random Forest

MAE (Mean Absolute Error) = 1107.08 USD. цей показник означає, що в середньому прогнозована моделлю заробітна плата відхиляється від фактичної на 1107.08 доларів США.

RMSE (Root Mean Squared Error) = 1655.06 USD: RMSE є подібним до MAE, але надає більшої ваги більшим помилкам, оскільки помилки спочатку зводяться в квадрат. Значення RMSE знаходиться в тому ж масштабі, що й цільова змінна (зарплата). Вище значення RMSE порівняно з MAE вказує на наявність більших окремих помилок, які мають значний вплив на загальну оцінку. Чим менше RMSE, тим краще.

R² (R-squared) Score = 0.4487: коефіцієнт детермінації R² показує, яка частка дисперсії цільової змінної (зарплати) пояснюється моделлю. Значення 0.4487 (або приблизно 44.87%) означає, що модель Random Forest здатна пояснити менше половини варіабельності заробітних плат на основі вхідних ознак. Хоча це вказує на певну пояснювальну силу моделі, такі значення для R² не є високими, і це свідчить про наявність значних розбіжностей між прогнозованими та фактичними зарплатами.

Отримані значення MAE (1107.08) та RMSE (1655.06) є відносно великими, що вказує на значні відхилення прогнозів моделі від реальних значень. Це може бути пов'язано з кількома факторами, які впливають на якість та повноту вихідних даних.

Можлива проблема полягає у недостатньому обсязі історичних даних або їх обмеженому діапазоні для адекватного навчання моделі на всіх можливих комбінаціях ознак (посади, міста, мови програмування, рівні англійської, стаж). Недостатня кількість спостережень для певних рідкісних комбінацій може призводити до менш точних прогнозів у цих сегментах. Опитування DOU.ua, які є джерелом даних, могли проводитися нерегулярно або мати пропущені місяці/періоди. Це створює "прогалини" в часових рядах даних, що ускладнює виявлення довгострокових тенденцій та сезонності (якщо така є) та негативно впливає на здатність моделі до узагальнення. Заробітна плата в IT-сфері залежить від багатьох неврахованих у поточному датасеті факторів, таких як розмір компанії, тип компанії (аутсорс, продукт, стартап), наявність додаткових бонусів, спеціалізація в рамках однієї посади, індивідуальні навички, сертифікації, а також поточна економічна ситуація. Ці фактори можуть значно впливати на дисперсію зарплат і, не будучи включеними до моделі, обмежують її пояснювальну силу.

Незважаючи на наявність цих відхилень, модель все ж таки демонструє певну здатність до прогнозування ($R^2 > 0$), і отримані результати можуть слугувати відправною точкою для подальшого аналізу та вдосконалення, зокрема шляхом збору більшого обсягу даних або включення додаткових релевантних ознак. Ці метрики слугуватимуть базою для порівняння з іншими моделями, зокрема з XGBoost, для вибору найбільш ефективного рішення.

Після успішного прогнозування заробітних плат на майбутній період (наприклад, 2025 рік) для всіх змодельованих сценаріїв, система переходить до етапу інтерпретації цих числових даних у конкретні, зрозумілі кар'єрні рекомендації. Цей процес відбувається у спеціалізованому модулі `reporting_module.py`, який реалізує низку механізмів для агрегації, фільтрації та ранжування прогнозів.

Сформовані рекомендації виводяться безпосередньо у консоль у вигляді читабельних таблиць, що дозволяє користувачеві швидко та ефективно сприймати ключові висновки.

Топ 10 рекомендацій за посадами з найкращими комбінаціями факторів (2025 рік):

Посада	Прогнозована ЗП (USD)	Стаж (років)	Рівень англійської	Мова програмування
System Architect	15112	15	Upper-Intermediate	Python
DevOps, SRE	12910	15	Upper-Intermediate	Python
CTO, Director of Engineering, Program Director, COO, CEO, founder	12171	6	Upper-Intermediate	Python
Senior DevOps, SRE	10947	15	Advanced	Python
Senior Software Engineer / Programmer	10595	12	Advanced	Python
Senior Software Engineer	10253	10	Intermediate	Salesforce
Team Lead	9650	15	Advanced	ABAP
Manager CTO, Director of Engineering, Program Director, COO, CEO, founder	9580	10	Advanced	Python
Tech Lead	9556	9	Intermediate	JavaScript

Рис. 5. Топ 10 рекомендацій за посадами з найкращими комбінаціями факторів

Як видно з таблиці, серед найвищих прогнозованих заробітних плат домінують позиції високого рівня відповідальності та значного досвіду, такі як System Architect, DevOps, SRE, а також керівні ролі (CTO, Director of Engineering). Це цілком очікувано, оскільки зі зростанням стажу та рівня експертизи зростає і потенційний дохід.

Ця рекомендація, хоча й очевидно вказує на лідерські та архітектурні позиції як найбільш високооплачувані, все одно надає цінне розуміння абітурієнтам та молодшим спеціалістам. Вона чітко окреслює кар'єрні траєкторії, показуючи, куди варто рухатися в майбутньому, які навички (зокрема, рівень англійської) та мови програмування (наприклад, Python) є найбільш затребуваними для досягнення високих доходів у довгостроковій перспективі.

PROBLEMS (206) OUTPUT DEBUG CONSOLE TERMINAL PORTS GIT LENS SPELL CHECKER (206)

--- Загальні висновки за посадами ---
Середні прогнозовані зарплати за посадами:

посада	прогноз_зарплати
CTO, Director of Engineering, Program Director, COO, CEO, founder	7069
Software Engineer / Programmer	6530
System Architect	6098
Manager Software Engineer / Programmer	6003
Senior DevOps, SRE	5534

Рис. 6. Середні прогнозовані зарплати за посадами

Ця таблиця підтверджує загальну тенденцію: керівні та архітектурні позиції, а також досвідчені розробники (Software Engineer / Programmer) демонструють найвищий рівень середніх заробітних плат. Це підкреслює важливість професійного зростання та розвитку до лідерських та експертних ролей для збільшення доходу в IT-сфері.

--- Вплив рівня англійської на зарплату ---
Середні прогнозовані зарплати за рівнями англійської:

рівень англійської	прогноз_зарплати
Advanced	3811
Upper-Intermediate	3229
Intermediate	2251
Pre-Intermediate	1659
Elementary	1587

****Висновок:**** Вищий рівень англійської зазвичай корелює з вищими зарплатами. Рекомендовано абітурієнтам покращувати свій рівень.

Рис. 7. Вплив рівня англійської на зарплату

Аналіз показує чітку тенденцію: вищий рівень володіння англійською мовою прямо корелює з вищими прогнозованими заробітними платами.

--- Аналіз мов програмування для технічних ліdersких та інженерних посад ---

--- Топ найоплачуваніших мов програмування ---

мова_програмування	прогноз_зарплати	кількість
Clojure	4195	51
Go	4148	333
Rust	3811	54
Scala	3741	594
Ruby	3531	395

****Висновок (зарплата):**** Найвища середня – Clojure (4195 USD, 51 вакансій)

Це може свідчити про високу спеціалізацію або складність задач, які виконуються з використанням цієї мови, що відповідно впливає на рівень оплати.

Рис. 8. Топ найоплачуваніших мов програмування

--- Топ мов програмування за популярністю ---

мова_програмування	прогноз_зарплати	кількість
JavaScript	2189	13135
Java	2678	9750
PHP	2175	7278
C#/.NET	2435	7811
Python	2632	3445

****Висновок (попит):**** Найбільший попит має JavaScript (13135 вакансій)

Це може свідчити про більшу легкість у пошуку роботи для початківців або про ширші можливості.

Рис. 9. Топ мов програмування за популярністю

Як видно з колонки "кількість", JavaScript має найбільший попит на ринку праці з 13135 вакансіями. Це свідчить про його широке застосування у веб-розробці та може вказувати на більшу легкість у пошуку першої роботи для початківців або про ширші можливості для зміни проектів у цьому напрямку.

--- Важливість стажу ---
Середні прогнозовані зарплати за стажем:

стаж	прогноз_зарплати
12	5495
13	5398
11	5353
15	5183
14	5072

****Висновок:**** Чим більший стаж, тим вища потенційна зарплата. Програми з фокусом на практиці та стажуванні можуть бути корисними.

Рис. 10. Середні прогнозовані зарплати за стажем

--- Середні прогнозовані зарплати за містами ---
Топ 10 міст з найвищими середніми прогнозованими зарплатами:

Місто	Середня ЗП (USD)
Не Україна	5829
Ужгород чи Закарпатська область	3278
Луцьк чи Волинська область	2984
Київ	2939
Львів	2723
Одеса	2399
Харків	2378
Івано-Франківськ	2332
Вінниця	2295
Житомир	2272

****Висновок:**** Найвища середня прогнозована зарплата спостерігається в місті **Не Україна** (5829 USD).

Вибір міста може суттєво впливати на потенційний дохід, тому абітурієнтам варто розглядати міста з вищими зарплатами.

Рис. 11. Середні прогнозовані зарплати за містами

--- Найкращі можливості для початківців (стаж до 2 років) ---
 Top 10 комбінацій навичок/посад для початківців (стаж до 2 років або Junior/Intern):

Посада	Мова програмування	Рівень англійської	Середня ЗП (USD)	Кількість вакансій
Software Engineer	Scala	Intermediate	2113	30
Software Engineer	Scala	Upper-Intermediate	1929	29
Software Engineer	Java	Advanced	1844	70
Software Engineer	Swift	Upper-Intermediate	1748	87
Software Engineer	Swift	Upper-Intermediate	1748	87
Software Engineer	Python	Advanced	1747	51
Software Engineer	Python	Advanced	1747	51
Software Engineer	Python	Advanced	1747	51
Software Engineer	Python	Advanced	1747	51
Software Engineer	Java	Intermediate	1734	708

Рис.12. Топ 10 комбінацій навичок/посад для початківців (стаж до 2 років або Junior/Intern)

З представленої таблиці видно, що позиція Software Engineer є домінуючою для початківців, що підкреслює її універсальність та високий попит на ринку. Хоча кількість вакансій для Scala-розробників з невеликим стажем не є великою (30 та 29 вакансій), вона демонструє найвищі прогнозовані середні зарплати (2113 та 1929 USD) для початківців. Це може вказувати на високу цінність цих спеціалістів на ринку або на меншу конкуренцію. Java, Swift, Python: ці мови також пропонують привабливі можливості для початківців, з прогнозованими зарплатами в діапазоні 1747–1844 USD та значно більшою кількістю вакансій. Це робить їх більш безпечним вибором для тих, хто прагне легше знайти першу роботу.

Для повноцінного аналізу ефективності розробленої рекомендаційної системи, після оцінки моделі Random Forest, було вирішено запустити та оцінити другу реалізовану модель – XGBoost. Цей крок дозволить порівняти їхню продуктивність та визначити, яка модель краще справляється з задачею прогнозування заробітних плат на наданому наборі даних.

```
$ python run.py
Оберіть модель для запуску:
1 - Random Forest
2 - XGBoost
Введіть номер моделі: 2
Завантажую дані для навчання/прогнозування...
Примусове навчання нової моделі (FORCE_TRAIN=True).
Починаю навчання моделі...
```

Рис.13. Запуск моделі XGBoost

PROBLEMS 223 OUTPUT DEBUG CONSOLE TERMINAL P

Оцінка моделі на тестових даних (2024 рік):
 Mean Absolute Error (MAE): 631.54 USD
 Root Mean Squared Error (RMSE): 1040.40 USD
 R-squared (R²) Score: 0.6651

Рис. 14. Оцінка ефективності моделі XGBoost

MAE (631.54 USD): в середньому, прогноз моделі відхиляється від фактичної зарплати на 631.54 USD. Це значно менша абсолютна похибка порівняно з Random Forest, що вказує на вищу точність прогнозів.

RMSE (1040.40 USD): аналогічно MAE, значення RMSE також суттєво нижче, ніж у Random Forest. Це означає, що XGBoost краще справляється з мінімізацією як середніх, так і великих помилок.

R² (R-squared) Score: 0.6651: цей коефіцієнт показує, що близько 66.51% варіації заробітної плати пояснюється моделлю XGBoost. Це значно вище, ніж у Random Forest (44.87%), що свідчить про значно кращу здатність XGBoost виявляти та моделювати залежності в даних.

Як видно, XGBoost демонструє значно кращі показники за всіма ключовими метриками. Це вказує на те, що XGBoost краще підходить для даного набору даних та задачі, забезпечуючи більш точні та надійні прогнози. Його здатність пояснювати більше варіантів у даних (вищий R²) робить його більш ефективним інструментом для розробки рекомендаційної системи. Ці результати підкреслюють важливість вибору відповідної моделі машинного навчання для досягнення оптимальної продуктивності.

Порівняно з результатами, отриманими від моделі Random Forest, прогнози XGBoost щодо найбільш високооплачуваних комбінацій виявилися значно вищими, що корелює з кращими метриками якості цієї моделі. Таблиця XGBoost включає такі високоспеціалізовані та інноваційні позиції, як Principal Computer Vision Engineer (з прогнозом 19959 USD) та Senior Data Scientist (з 18395 USD), чого не спостерігалось в топі Random Forest. Це підкреслює зростаючу цінність фахівців у сферах AI/ML та Data Science.

--- Формування рекомендацій для абітурієнтів ---
 Top 10 рекомендацій за посадами з найкращими комбінаціями факторів (2825 рік):

Посада	Прогнозована ЗП (USD)	Стаж (років)	Рівень англійської	Мова програмування
Principal Computer Vision Engineer	19959	12	Advanced	Python
Senior Data Scientist	18395	8	Advanced	Python
CEO / C-level (Chief) / Director / VP CAO	17939	3	Intermediate	SQL
Staff Software Engineer / Software Developer	17629	14	Advanced	Clojure
Head of Engineering	16657	14	Advanced	C# / .NET
System Architect	15919	13	Upper-Intermediate	Інша мова
Senior DevOps Engineer	15147	13	Advanced	Python
Lead/Team Lead Engineering Manager	15859	15	Advanced	Clojure
Head of R&D	14484	7	Advanced	Java
Tech Lead	14373	14	Advanced	Clojure

Рис. 15. Top 10 рекомендацій за посадами з найкращими комбінаціями факторів

--- Загальні висновки за посадами ---
 Середні прогнозовані зарплати за посадами:

посада	прогноз_зарплати
Principal Computer Vision Engineer	19959
CEO / C-level (Chief) / Director / VP CAO	17939
Staff SRE	12958
CEO / C-level (Chief) / Director / VP Director of Engineering	8748
Head of Engineering	8313

Рис. 16.Середні прогнозовані зарплати за посадами

--- Високий рівень англійської на зарплату ---
 Середні прогнозовані зарплати за рівнями англійської:

рівень англійської	прогноз_зарплати
Advanced	3783
Upper-Intermediate	3288
Не знає базові	3831
Intermediate	2646
Pre-Intermediate	2129
Elementary	1971

Висновок: Вищий рівень англійської завжди корелює з вищими зарплатами. Рекомендовано абітурієнтам покращувати свій рівень.

Рис. 17.Середні прогнозовані зарплати за рівнями англійської

--- Аналіз мов програмування для технічних керівників та інженерів посад ---
 Top найоплачуваніших мов програмування ---

мова_програмування	прогноз_зарплати	кількість
Clojure	6186	71
Scala	5238	661
Elixir	4885	51
Go	4679	658
Rust	4377	187

Висновок (зарплата): Найвища середня – Clojure (6186 USD, 71 вакансія)
 Це може свідчити про високу спеціалізацію або складність задач, які виконуються з використанням цієї мови, що відносно впливає на рівень оплати.

Рис. 18. Top найоплачуваніших мов програмування

Обидві моделі ідентифікують Clojure, Go, Rust та Scala як одні з найоплачуваніших мов, що свідчить про стійкість цього висновку. Однак, XGBoost додатково виділяє Elixir до топу, тоді як Random Forest включає Ruby. Це може вказувати на те, що XGBoost точніше ідентифікує специфічні високооплачувані, можливо, більш нові або вузькоспеціалізовані технології, які Random Forest міг недооцінити або не виявити через меншу складність моделі чи інші параметри навчання.

--- Top мов програмування за популярністю ---

мова_програмування	прогноз_зарплати	кількість
JavaScript	3234	15132
Java	3695	11279
PHP	3456	8375
C#/.NET	3675	7811
TypeScript	3588	4855

Висновок (попит): Найбільший попит має JavaScript (15132 вакансій)
 Це може свідчити про більшу легкість у пошуку роботи для початківців або про ширші можливості.

Рис. 19. Top мов програмування за популярністю

Для тих, хто шукає легший вхід в IT-сферу або широкий вибір вакансій, мови з високим попитом, такі як JavaScript та Java, є гарним вибором.

--- Важливість стажу ---
Середні прогнозовані зарплати за стажем:

стаж	прогноз_зарплати
14	5361
13	5150
15	5120
12	5100
11	5009

Рис. 20. Важливість стажу

--- Середні прогнозовані зарплати за містами ---
Топ 10 міст з найвищими середніми прогнозованими зарплатами:

Місто	Середня ЗП (USD)
Не Україна	3521
Київ	3212
Львів	2914
Ужгород чи Закарпатська область	2852
Вінниця	2676
Івано-Франківськ	2675
Луцьк чи Волинська область	2642
Хмельницький	2633
Чернівець	2611
Одеса	2594

Рис. 21. Топ 10 міст з найвищими середніми прогнозованими зарплатами

Хоча обидві моделі погоджуються, що "Не Україна" та великі ІТ-центри (Київ, Львів) пропонують найвищі зарплати, XGBoost, з його вищими метриками якості, надає більш консистентні та, ймовірно, точніші прогнози для ключових українських міст, ставлячи Київ та Львів вище.

--- Найкращі можливості для початківців (стаж до 2 років) ---
Топ 10 комбінацій навичок/посад для початківців (стаж до 2 років або Junior/Intern):

Посада	Мова програмування	Рівень англійської	Середня ЗП (USD)	Кількість вакансій
Software Engineer	Scala	Upper-Intermediate	4236	29
Software Engineer	Scala	Intermediate	4082	30
Software Engineer	JavaScript	Advanced	3994	127
Software Engineer	Python	Advanced	3895	51
Software Engineer	Swift	Advanced	3872	33
Software Engineer	Java	Advanced	3791	70
Software Engineer	C#/.NET	Advanced	3729	38
Software Engineer	C	Upper-Intermediate	3719	17
Software Engineer	JavaScript	Upper-Intermediate	3658	589
Software Engineer	Kotlin	Upper-Intermediate	3636	43

Рис. 22. Топ 10 комбінацій навичок/посад для початківців (стаж до 2 років або Junior/Intern)

Топ-10 XGBoost включає ширший спектр мов програмування для початківців з високою зарплатою, зокрема JavaScript (з Advanced та Upper-Intermediate англійською), C#/.NET, C та Kotlin, які не були так високо представлені або не були присутні у топі Random Forest для початківців. Це дає молодим спеціалістам більше варіантів для вибору, показуючи, що високооплачувані можливості існують у різних технологічних стеках.

Проведений порівняльний аналіз двох реалізованих моделей, Random Forest та XGBoost, чітко продемонстрував, що модель XGBoost є значно ефективнішою та точнішою для вирішення задачі прогнозування заробітних плат в ІТ-сфері. Її вищі показники за метриками MAE (631.54 USD проти 1107.08 USD), RMSE (1040.40 USD проти 1655.06 USD) та R² (0.6651 проти 0.4487) свідчать про кращу здатність моделі виявляти та моделювати складні залежності у даних, що призводить до більш надійних та реалістичних прогнозів. З огляду на це, для подальшого використання та формування кар'єрних рекомендацій ми рекомендуємо орієнтуватися саме на результати та прогнози, надані моделлю XGBoost. Це забезпечить користувачам рекомендаційної системи найбільш актуальну та точну інформацію для прийняття обґрунтованих рішень щодо свого професійного розвитку, вибору технологій та стратегічного планування кар'єри в динамічному ІТ-секторі.

Висновки

Розроблена система продемонструвала, що навіть для початківців існують можливості для досягнення значних заробітних плат за умови володіння певними технологіями (наприклад, Scala, JavaScript, Python) та високим рівнем англійської. Це розширює горизонти для молодих спеціалістів та дозволяє їм з самого початку обирати найбільш перспективні напрямки розвитку. Загалом, розроблена система є успішним прототипом, що доводить можливість використання машинного навчання для надання цінних кар'єрних порад в ІТ-галузі.

Майбутній розвиток рекомендаційної системи може включати кілька ключових напрямків для підвищення її точності, функціональності та зручності використання. Однією з головних перспектив є інтеграція більшої кількості джерел даних. Додавання інформації про конкретні фреймворки, бібліотеки, спеціалізовані доменні знання (наприклад, FinTech, HealthTech, Gaming), а також даних про розмір компанії, її

локалізацію (продуктова чи аутсорсингова), дозволить моделі враховувати ще більше нюансів, що впливають на зарплату, та надавати більш деталізовані та персоналізовані рекомендації.

Література

1. Даниленко М.С. Методи розробки рекомендаційних систем/Даниленко М.С., Колесник І.С.// Вінницький національний технічний університет. Інформаційні технології та комп'ютерна інженерія. -2021. - С. 10–16.
2. Recommender system [Електронний ресурс].- Режим доступу: https://en.wikipedia.org/wiki/Recommender_system - (дата звернення: 02.03.2025). – Назва з екрана.
3. Марченко А.В. Проектування рекомендаційних систем: навч. посібник / А.В. Марченко // Київ, вид-во КНЕУ. - 2016. - 250 с.
4. Недашківський О.М. Рекомендаційні системи: навч. посібник / О.М. Недашківський // Київ. - 2014. - 215с.
5. Т.В. Нескородєва / Експертні та рекомендаційні системи: навч. посіб. / Т.В. Нескородєва, Є.Є. Федоров, Т.В. Січко, А.Р. Нескородєва // Вінниця: ДонНУ імені Василя Стуса. - 2023. - 224 с.
6. Годовиченко М.А. / Рекомендаційні системи: моделі, виклики та можливості/ Годовиченко М. А., Горбатенко А. А.// Вісник сучасних інформаційних технологій. - 2023. - с. 308–319.

References

1. Danylenko M.S. Metody rozrobky rekomendatsiinykh system / Danylenko M.S., Kolesnyk I.S. // Vinnytskyi natsionalnyi tekhnichnyi universytet. Informatsiini tekhnolohii ta kompiuterna inzheneriia . -2021. - P. 10–16.
2. Recommender system [Elektronnyi resurs]. - Rezhym dostupu: https://en.wikipedia.org/wiki/Recommender_system - (data zvernennia: 02.03.2025). – Nazva z ekrana.
3. Marchenko A.V. Proektuvannia rekomendatsiinykh system: navch. posibnyk / A.V. Marchenko // Kyiv, vyd-vo KNEU. - 2016. - 250 p.
4. Nedashkivskyi O.M. Rekomendatsiini systemy: navch. posibnyk/ O.M. Nedashkivskyi // Kyiv. - 2014. – 215p.
5. T.V. Neskordieva / Ekspertni ta rekomendatsiini systemy: navch. posib. / T.V. Neskordieva, Ye.Ie. Fedorov, T.V. Sichko, A.R. Neskordieva // Vinnytsia: DonNU imeni Vasylia Stusa. - 2023. - 224 p.
6. Hodovychenko M.A. / Rekomendatsiini systemy: modeli, vyklyky ta mozhlyvosti/ Hodovychenko M. A., Horbatenko A. A.// Visnyk suchasnykh informatsiinykh tekhnolohii.-2023.- p. 308–319.