Одеський національний університет імені І. І. Мечникова
Факультет математики, фізики та інформаційних технологій
Кафедра оптимального керування і економічної кібернетики

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

«Прогнозування результатів вступної кампанії до закладів вищої освіти на основі моделей машинного навчання»

«Forecasting the results of the admissions campaign to higher education institutions based on machine learning models»

Виконав: здобувач денної форми навчання
спеціальності 113 Прикладна математика
Освітня програма «Прикладна математика»
Чачко Натан Леонідович

Керівник: канд. фіз.-мат. наук, доц. Страхов Є. М. _____
Рецензент: канд. техн. наук, доц. Мороз В.В.

Рекомендовано до захисту:

Протокол засідання кафедри

№ ___ від _____ 2024 р.

Завідувач кафедри

_____ _____

Захищено на засіданні ЕК № _____

Протокол № ___ від _____ 2024 р.

Оцінка _____ / _____ / _____

Голова ЕК

_____ _____

Одеса — 2024 р.

ЗМІСТ

ВСТУП

Сучасне освітнє середовище в Україні стоїть перед викликами забезпечення доступності та якості вищої освіти для молоді. Система вступу до закладів вищої освіти через «широкий конкурс» [1] відіграє важливу роль у забезпеченні цих цілей. Аналіз цієї системи є актуальним, оскільки від нього залежить розподіл бюджетних місць та можливість молоді отримати освіту у відповідних галузях.

Метою даної дипломної роботи є дослідження системи «широкого конкурсу» у вступній кампанії до закладів вищої освіти в Україні, зокрема побудова та порівняння математичних моделей машинного навчання для передбачення кількості бюджетних місць для конкретних спеціальностей в університетах.

Об'єктом дослідження є система вищої освіти в Україні, яка включає в себе різноманітні аспекти організації вступної кампанії та розподілу бюджетних місць.

Предметом дослідження є система «широкого конкурсу» у вступній кампанії до закладів вищої освіти в Україні.

Для досягнення поставленої мети використовуватимуться статистичний аналіз результатів широкого конкурсу за 2018-2023 роки та побудова моделей машинного навчання. Зокрема, у роботі будуть розглянуті лінійні моделі, дерева рішень та градієнтний бустінг для прогнозування кількості бюджетних місць для різних спеціальностей у вищих навчальних закладах.

Основними результатами роботи будуть:

- Глибший аналіз та розуміння системи «широкого конкурсу» у вступній кампанії до закладів вищої освіти в Україні.
- Розробка та порівняння математичних моделей машинного навчання для прогнозування кількості бюджетних місць на різні спеціальності.
- Зменшення кількості змінних у моделі та створення практичного додатка для прогнозування результатів вступної кампанії.

РОЗДІЛ 1

СИСТЕМА «ШИРОКОГО КОНКУРСУ»

1.1 Загальні відомості

Міністерство освіти і науки України прийняло рішення відмовитися від ручного розподілу державного замовлення у вищій освіті і перейти до нової системи, яка базується на принципі «бюджетні місця йдуть за кращими вступниками» та формульному розподілі державного замовлення.

Основна ідея «широкого конкурсу» [1] полягає в тому, що бюджетні місця не надаються безпосередньо закладам вищої освіти, а розподіляються від держави громадянам, які здобувають їх на конкурсній основі. Це означає, що абітурієнти, які закінчили 11 класів, отримують рекомендації до бажаних вишів завдяки автоматичному розподілу бюджетних місць. Кількість державних місць, які отримує кожен вищий навчальний заклад, залежить від конкурсних балів абітурієнтів. Таким чином, виші, до яких подавали заяви абітурієнти з вищими конкурсними балами, отримують більшу кількість державних місць.

1.2 Опис алгоритму

1) Етап А.

   Перший крок.

   Кожному розрахунковому конкурсу пропонується перелік вступників, для яких цей розрахунковий конкурс має найвищу пріоритетність. Кожний розрахунковий конкурс включає до списку очікування кращих за власним рейтинговим списком вступників із запропонованих вступників у кількості, що не перевищує обсягу розрахункового конкурсу, а решті відмовляє.

   Кожна група субконкурсів кожного конкурсу (субконкурс А, та/або субконкурс Б, та/або субконкурс ББ і субконкурс В) перевіряється на перевищення максимального (загального) обсягу державного чи

регіонального замовлення конкурсу. У разі перевищення визначається відповідна кількість вступників із субконкурсу В з нижчими позиціями в рейтинговому списку вступників, які отримують відмову.

Кожний широкий конкурс перевіряється на перевищення суперобсягу державного замовлення. У разі перевищення за об'єднаним списком очікування визначається відповідна кількість вступників (не із субконкурсів А, і не із субконкурсів Б, і не із субконкурсів ББ) з найменшими значеннями конкурсного бала (за рівних конкурсних балів — з урахуванням пункту 2 розділу IX Умов прийому), які також отримують відмову.

К-ий крок (К $>$ 1).

На наступних кроках кожний вступник, який на цей момент не внесений до списку очікування жодного розрахункового конкурсу, пропонується тому розрахунковому конкурсу, який має для нього найвищу пріоритетність (крім тих, де він уже отримав відмову).

Кожний розрахунковий конкурс об'єднує наявний у нього список очікування та отриману пропозицію, формує новий список очікування за власним рейтинговим списком вступників у кількості, що не перевищує обсягу розрахункового конкурсу, а решті відмовляє.

Кожна група субконкурсів кожного конкурсу (субконкурс А, та/або субконкурс Б, та/або субконкурс ББ і субконкурс В) перевіряється на перевищення максимального (загального) обсягу державного або регіонального замовлення конкурсу. У разі перевищення визначається відповідна кількість вступників із субконкурсу В з нижчими позиціями в його рейтинговому списку вступників, які отримують відмову.

Кожний широкий конкурс перевіряється на перевищення суперобсягу державного замовлення. У разі перевищення за об'єднаним списком очікування визначається відповідна кількість вступників (не із субконкурсів А, і не із субконкурсів Б, і не із субконкурсів ББ) з найменшими значеннями конкурсного бала (за рівних конкурсних балів — з урахуванням пункту 2 розділу IX Умов прийому), які також отримують відмову.

Етап А вважається виконаним, коли вичерпується перелік пропо-

зицій вступників до розрахункових конкурсів, які не перебувають у списках очікування та не отримали відмови за всіма розрахунковими конкурсами.

2) Етап Б.

Якщо наявні конкурси, в яких кількість вступників у списку очікування менше ніж мінімальний обсяг державного або регіонального замовлення, такі конкурси анулюються, а вступники з їх списків очікування виключаються, отримують відмову і помічаються як такі, що допущені до етапу В.

Фіналіст розрахункового конкурсу — вступник з найнижчим положенням у рейтинговому списку розрахункового конкурсу, включений до списку очікування, після завершення етапу А.

Фіналіст широкого конкурсу — вступник, крім вступників із субконкурсів А, Б та ББ, з найнижчим положенням у широкому рейтинговому списку широкого конкурсу, включений до списку очікування, після завершення етапу А .

3) Етап В

К-й крок (К >= 1).

Кожний допущений до етапу В вступник, який на цей момент не внесений до списку очікування жодного розрахункового конкурсу, пропонується тому розрахунковому конкурсу, який має для нього найвищу пріоритетність (крім тих, де він уже отримав відмову в межах етапів А та В). Кожний розрахунковий конкурс включає до свого списку очікування вступників з отриманих пропозицій.

Кожен розрахунковий конкурс перевіряється на перевищення обсягу розрахункового конкурсу. У разі перевищення визначається відповідна кількість вступників, допущених до етапу В, з нижчими позиціями в рейтинговому списку вступників, які отримують відмову, за винятком тих вступників, чия позиція в рейтинговому списку розрахункового конкурсу вища за позицію фіналіста розрахункового конкурсу.

Кожна група субконкурсів кожного конкурсу (субконкурс А, та/або субконкурс Б, та/або субконкурс ББ і субконкурс В) перевіряється

на перевищення максимального обсягу державного або регіонального замовлення конкурсу. У разі перевищення визначається відповідна кількість вступників, допущених до етапу В, із субконкурсу В з нижчими позиціями в його рейтинговому списку вступників, які отримують відмову, за винятком тих вступників, чия позиція вища за позицію фіналіста розрахункового конкурсу відповідного субконкурсу В.

Кожний широкий конкурс перевіряється на перевищення суперобсягу державного замовлення і в разі перевищення за об'єднаним списком очікування визначається відповідна кількість вступників (не із субконкурсів А, не із субконкурсів Б, не із субконкурсів ББ), допущених до етапу В, з найменшими значеннями конкурсного бала (за рівних конкурсних балів — з урахуванням пункту 2 розділу IX Умов прийому), які також отримують відмову, за винятком тих вступників, чия позиція в широкому рейтинговому списку вища за позицію фіналіста широкого конкурсу.

Етап В вважається виконаним, коли вичерпується перелік пропозицій вступників (допущених до етапу В) до розрахункових конкурсів, які не перебувають у списках очікування та не отримали відмови за всіма розрахунковими конкурсами.

Вступники, які на цей момент залишились у списках очікування, одержують рекомендацію до зарахування. Кількість вступників, що одержали рекомендацію, визначає кількість рекомендованих за кожним конкурсом вступників.

РОЗДІЛ 2

ЗБІР ТА ОПИС НАБОРУ ДАНИХ

2.1 Етап збору даних

Першим і одним з найвідповідальніших етапів дослідницької роботи є збір даних. Від якості і кількості досліджуваних даних залежать результати аналізу і точність математичної моделі для прогнозування розподілу бюджетних місць у майбутніх вступних кампаніях. В цій роботі будуть розглядатися статистичні дані вступних кампаній з 2018-го по 2023-й роки включно.

Частину даних вдалося знайти у вигляді Excel-таблиць у відкритому доступі [2], але вони не містили деяких важливих параметрів, тому було прийняте рішення зібрати їх додатково. Необхідні дані можна знайти на тих самих сайтах, з яких були завантажені Excel-таблиці, але доступні вони лише при перегляді інформації про конкурсну пропозіцію у браузері.

Звісно, що збирати такий об'єм даних (а це близько 5-ти тисяч рядків за кожен рік) вручну прийшлося би дуже довго. Тому спеціально для цієї задачі був розроблений додаток мовою Python з використанням бібліотеки Selenium, за допомогою якого відбувався парсінг сайтів з необхідною інформацією.

Оскільки ці сайти мають деякі відмінності у своїй структурі, насправді довелося написати скрипт для парсінгу кожного сайту окремо, отже в результаті вийшло 5 окремих додатків [3].

Одна з найбільших проблем, з якою довелося зіткнутися під час збору необхідних даних, — це блокування веб-сайтами ботів. Коли з одного місця надходить дуже багато запитів за короткий час, система блокує доступ до сайту. Тому було необхідно встановити штучну затримку між запитами, щоб максимально імітувати поведінку людини.

2.2 Опис отриманого набору даних

В результаті збору даних була отримана таблиця [3], яка містить 31 різний стовпчик або параметр, відомий як «фіча», що може впливати на результати вступної кампанії та розподіл бюджетних місць в закладах вищої освіти:

1) uni_code — код університету. Є у кожного ЗВО країни.
2) spec_num — номер спеціальності.
3) Спеціальність — назва спеціальності.
4) specialization — назва спеціалізації (якщо є).
5) form — форма навчання: денна або заочна.
6) Орган управління — назва органу, якому підпорядковується ЗВО.
7) Назва закладу.
8) Усього подано заяв — загальна кількість заяв, що були подані на дану пропозицію.
9) Подано заяв на бюджет — кількість заяв, що були подані на бюджет.
10) Допущено до конкурсу — кількість допущених до конкурсу абітурієнтів.
11) Середній пріоритет допущених.
12) Усього рекомендовано.
13) Середній пріоритет рекомендованих.
14) Суперобсяг — кількість місць на відкриті конкурсні пропозиції, які складають широку конкурсну пропозицію, на які може бути надано рекомендацію для зарахування на місця державного замовлення
15) Фіксований обсяг — обсяг конкурсної пропозиції із заздалегідь визначеною кількістю бюджетних місць.
16) Рекомендовано за співбесідою.
17) Рекомендовано за квотою-2.

    Квота 2 — це частина від максимального обсягу бюджетних місць у закладах вищої освіти, яка може бути використана для прийому вступників на основі повної загальної середньої освіти або ступеня молодшого бакалавра, місце проживання яких зареєстровано на тимчасово окупованій території, території населених пунктів на лінії зіткнення та адміністративній межі або які переселилися з неї.

18) УСЬОГО — ітогова кількість рекомендованих на бюджет. Цей показник і буде нашою цільвою змінною (таргетом).
19) На загальних підставах — кількість рекомендованих на основі результатів широкого конкурсу (без квот).
20) Середній пріорітет рекомендованих на загальних підставах.
21) квота-1 — це визначена частина максимального обсягу бюджетних місць, яка може бути використана для прийому вступників наступних категорій:
    - які мають право на вступ на основі індивідуальної усної співбесіди (крім осіб, які мають право на квоту 2);
    - дітей-сиріт, дітей, позбавлених батьківського піклування, осіб з їх числа.
22) Мін. Бал (на загальних підставах) — мінімальний конкурсний бал серед рекомендованих на загальних підставах.
23) Сер. Бал (на загальних підставах) — середній конкурсний бал рекомендованих на загальних підставах.
24) Макс. Бал (на загальних підставах) — максимальний конкурсний бал серед рекомендованих на загальних підставах.
25) Рік — розглядаються дані вступних кампаній за 2018-2023 роки.
26) Макс. обсяг держзамовлення — максимальна можлива кількість бюджетних місць, на які можуть бути зараховані абітурієнти. Це означає, що кількість осіб зарахованих на бюджет до відповідного закладу освіти може бути меншою, ніж визначений максимальний обсяг державного замовлення.
27) СЕР — середній конкурсний бал усіх поданих заяв.
28) МІН — мінімальний конкурсний бал серед усіх поданих заяв.
29) МАКС — максимальний конкурсний бал серед усіх поданих заяв.
30) Ліцензійний обсяг — максимальна кількість студентів, які можуть бути зараховані до університету на даний напрям підготовки. Включає в себе місця державного замовлення та місця за кошти фізичних чи юридичних осіб.
31) Регіональний коефіцієнт — це коефіцієнт, на який остаточно множиться конкурсний бал. Залежить від регіону, де розташований ЗВО.

РОЗДІЛ 3

СТАТИСТИЧНИЙ АНАЛІЗ ДАНИХ

3.1 Попередня обробка даних

Спочатку потрібно об'єднати стовпчики, які несуть інформацію про спеціальність. Серед спеціальностей зустрічаються різні спеціалізації, тому унікальною спеціальністю будемо вважати комбінацію стовпчиків `spec_num`, `Спеціальність` та `specialization`.

`Назва закладу` несе таку ж інформацію як і `uni_code`. Знаючи один з цих параметрів, можна завжди однозначно визначити інший. Тому видаляємо стовпчик `Назва закладу`.

Пропозиції з фіксованими обсягами нас не цікавлять, тому стовпчик `Фіксований обсяг` видаляємо.

В датасеті присутні стовпчики, які дуже сильно корелюють з цільовою змінною і несуть практично ту ж саму інформацію: `на загальних підставах`, `Усього рекомендовано` і рідко та мало відрізняються. Ці дані становляться відомими також після розподілу бюджетних місць, тому їх ми не враховуємо при аналізі та побудові моделей.

Наступним кроком видалимо ті рядки, в яких не вистачає усіх даних.

У нашому датасеті є декілька категоріальних змінних: назва спеціальності, органу управління, номер університету та рік. Можна було б застосувати до них метод `pandas.get_dummies()`, щоб перетворити категоріальні змінні на фіктивні змінні, які являють собою числові змінні, що використовуються для представлення категоріальних даних. Але в нашому випадку такий підхід би значно збільшив розмір датасету (аж до 416 стовпчиків), що не є дуже добре. Тому будемо застовувати так званий Target encoding, який не вимагає створення додаткових стовпчиків.

3.1.1 Target encoding

Target encoding передбачає заміну категоріальної ознаки середнім цільовим значенням усіх точок даних, що належать до категорії.

Однією з проблем цільового кодування є перенавчання. Деякі також називають це витоком цільової змінної в одну з фіч (Leakage of target). У цих випадках модель із цільовим кодуванням погано узагальнює нові дані. Зменшити перенавчання при цільовому кодуванні можна за допомогою згладжування.

Одним із популярних методів згладжування є використання комбінації таргету для категорії та глобального цільового середнього для кожної точки даних. Ця техніка особливо корисна для вирішення ситуацій, коли для деяких категорій дуже мало даних.

Адитивне згладжування

$$\mu = \frac{n \times \bar{x} + m \times w}{n + m}, \tag{3.1}$$

де

- μ — середнє, яке ми намагаємося обчислити (те, яке замінить наші категоріальні значення)
- n — кількість елементів у групі
- $\bar{x}$ — передбачуване середнє
- m — ваговий коефіцієнт, який застосовується для загального середнього значення
- w — загальне середнє значення

У цій формулі m є єдиним параметром, який потрібно встановити. Ідея полягає в тому, що чим вищий m, тим більше ми покладаємося на загальне середнє w. Якщо m дорівнює 0, тоді отримаємо емпіричне середнє, яке дорівнює:

$$\mu = \frac{n \times \bar{x} + 0 \times w}{n + 0} = \frac{n \times \bar{x}}{n} = \bar{x} \tag{3.2}$$

Іншими словами, в такому випадку згладжування не відбувається.

3.2 Визначеня типу розподілу змінних

Спочатку подивимося на розподіл цільової змінної:

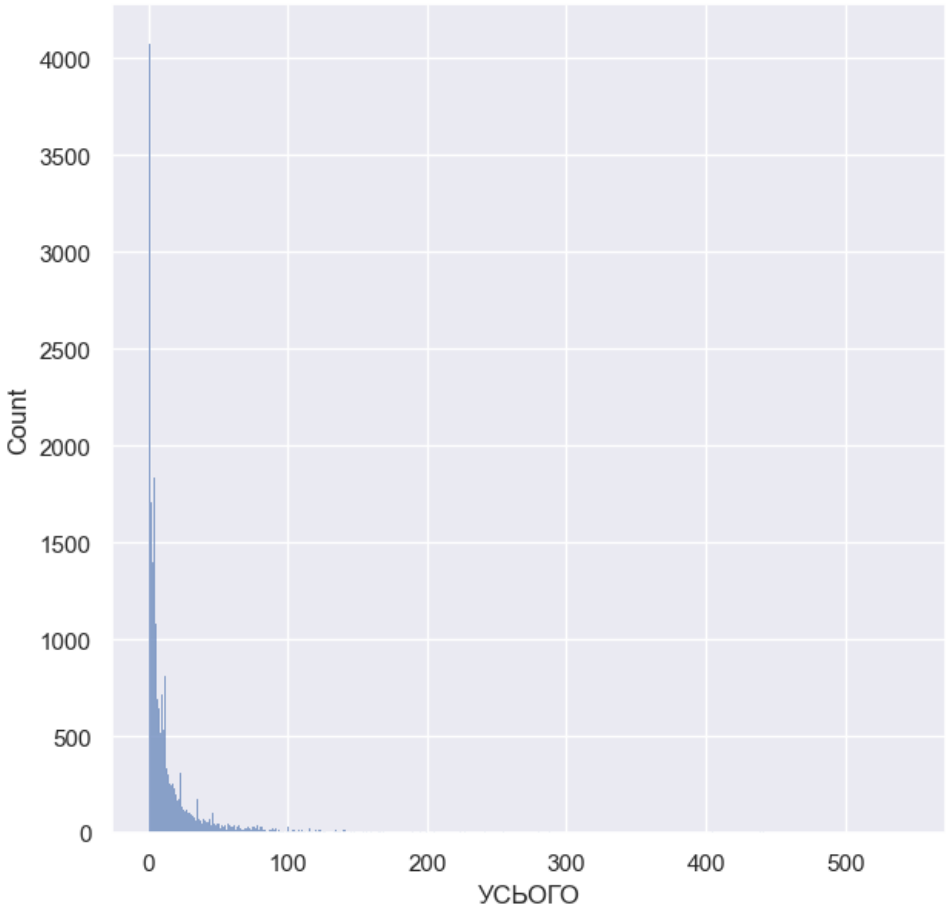


Рис. 3.1. Розподіл змінної “УСЬОГО”

Як бачимо, цей розподіл дуже далекий від нормального.

Для перевірки нормальності розподілу решти змінних скористуємося тестами Д’Агостіно-Пірсона та Шапіро-Уілка [3]. Обидва тести вказують на те, що нульову гіпотезу (H_0: вибіркові значення походять з нормального розподілу) можна відкинути.

Тому для визначення тісноти зв’язку будемо користуватися ранговими коефіцієнтами Спірмена і Кендалла.

3.3 Кореляційний аналіз

Для розуміння загальної картини спочатку побудуємо кореляційну матрицю.

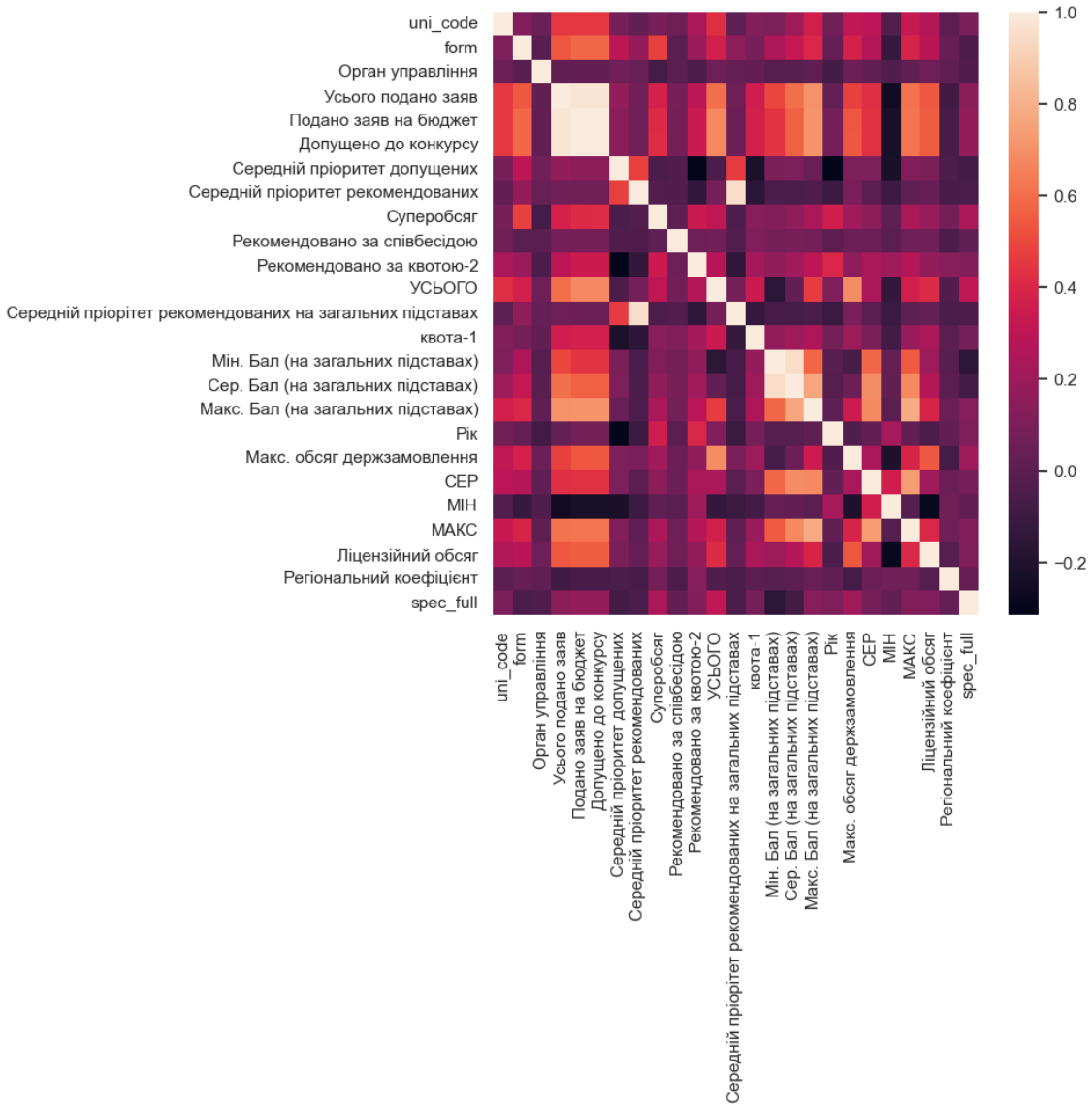


Рис. 3.2. Кореляційна матриця

Для побудови кореляційної матриці був застосован метод Спірмена. Вона симетрична. Цікавими для нас будуть найсвітліші фрагменти, які вказують на високу кореляцію ознак, що знаходяться на перетину відповідних рядків і стовпчиків. Це означає, що якісь з них можна не враховувати при побудові моделі, щоб запобігти мультиколінеарності.

Розглянемо тепер конкретні коефіцієнти кореляції факторів с цільовою змінною. Спочатку коефіцієнти кореляції Спірмена:

УСЬОГО	1.000000
Макс. обсяг держзамовлення	0.690271
Допущено до конкурсу	0.672055
Подано заяв на бюджет	0.671505
Усього подано заяв	0.602191
Макс. Бал (на загальних підставах)	0.463079
uni_code	0.424142
Ліцензійний обсяг	0.413872
form	0.372138
МАКС	0.365154
квота-1	0.338165
spec_full	0.309695
Суперобсяг	0.303935
Рекомендовано за квотою-2	0.267334
СЕР	0.237016
Рік	0.107041
Середній пріорітет рекомендованих на загальних підставах	0.078796
Середній пріоритет рекомендованих	0.078544
Рекомендовано за співбесідою	0.060364
Орган управління	0.054770
Сер. Бал (на загальних підставах)	0.022433
Регіональний коефіцієнт	-0.040122
Середній пріоритет допущених	-0.048844
МІН	-0.136833
Мін. Бал (на загальних підставах)	-0.159429

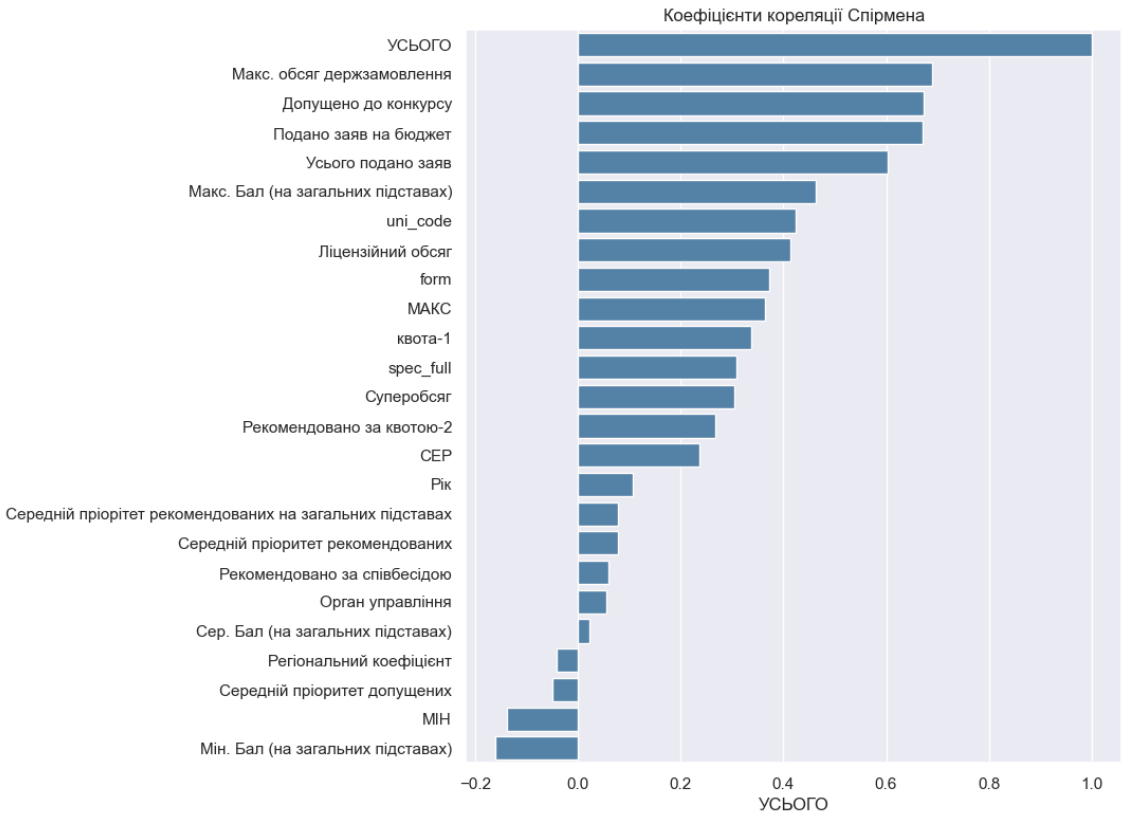


Рис. 3.3. Коефіцієнти кореляції Спірмена

Коефіцієнти кореляції Кендалла:

УСЬОГО	1.000000
Макс. обсяг держзамовлення	0.542016
Допущено до конкурсу	0.506950
Подано заяв на бюджет	0.506701
Усього подано заяв	0.446472
Макс. Бал (на загальних підставах)	0.329120
form	0.311158
uni_code	0.299814
Ліцензійний обсяг	0.297941
квота-1	0.278490
МАКС	0.257080
spec_full	0.218819
Рекомендовано за квотою-2	0.215833
Суперобсяг	0.211465
СЕР	0.164506
Рік	0.078700
Рекомендовано за співбесідою	0.050434
Середній пріоритет рекомендованих	0.049456
Середній пріорітет рекомендованих на загальних підставах	0.047884
Орган управління	0.045701
Сер. Бал (на загальних підставах)	0.016964
Регіональний коефіцієнт	-0.030969
Середній пріоритет допущених	-0.033866
МІН	-0.094701
Мін. Бал (на загальних підставах)	-0.109817

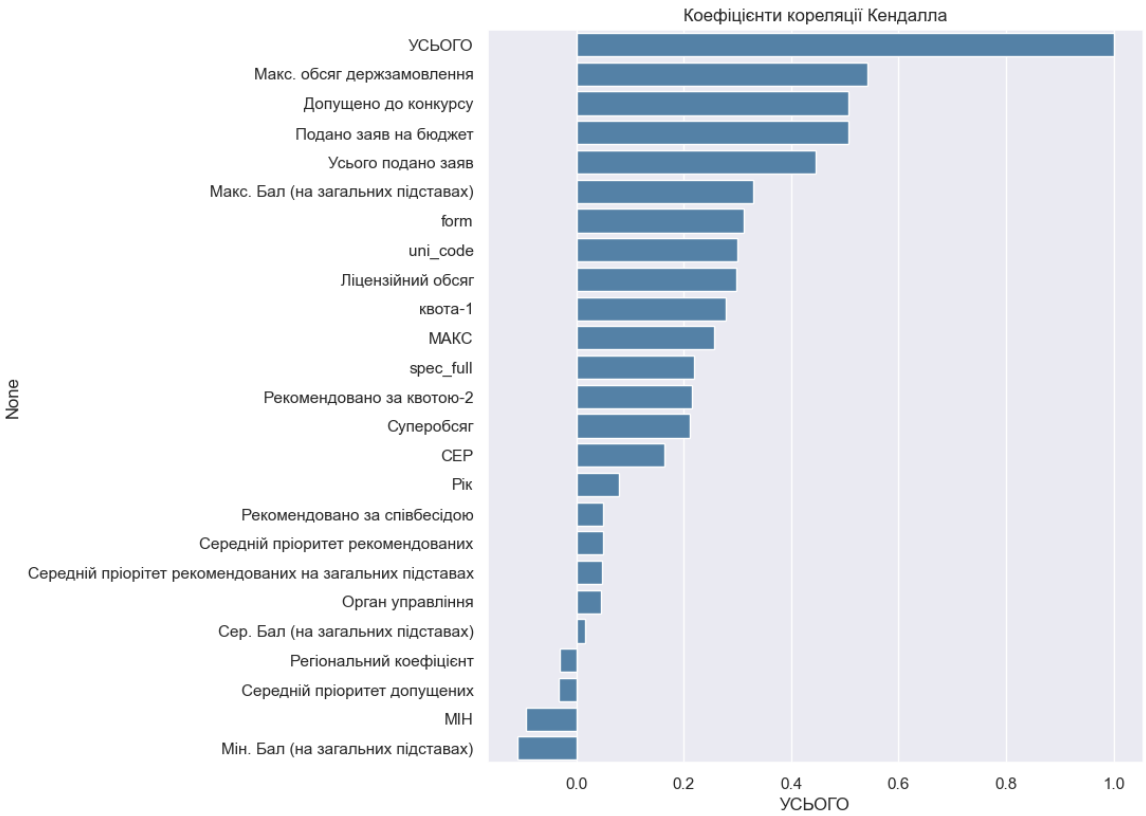


Рис. 3.4. Коефіцієнти кореляції Кендала

При порівнянні результатів можна помітити, що коефіцієнт Спірмена більший за коефіцієнт Кендалла майже для усіх параметрів. Це вказує на лінійну залежність між цими параметрами та цільовою змінною.

Розглянемо детальніше відношення кожної змінної до таргету.

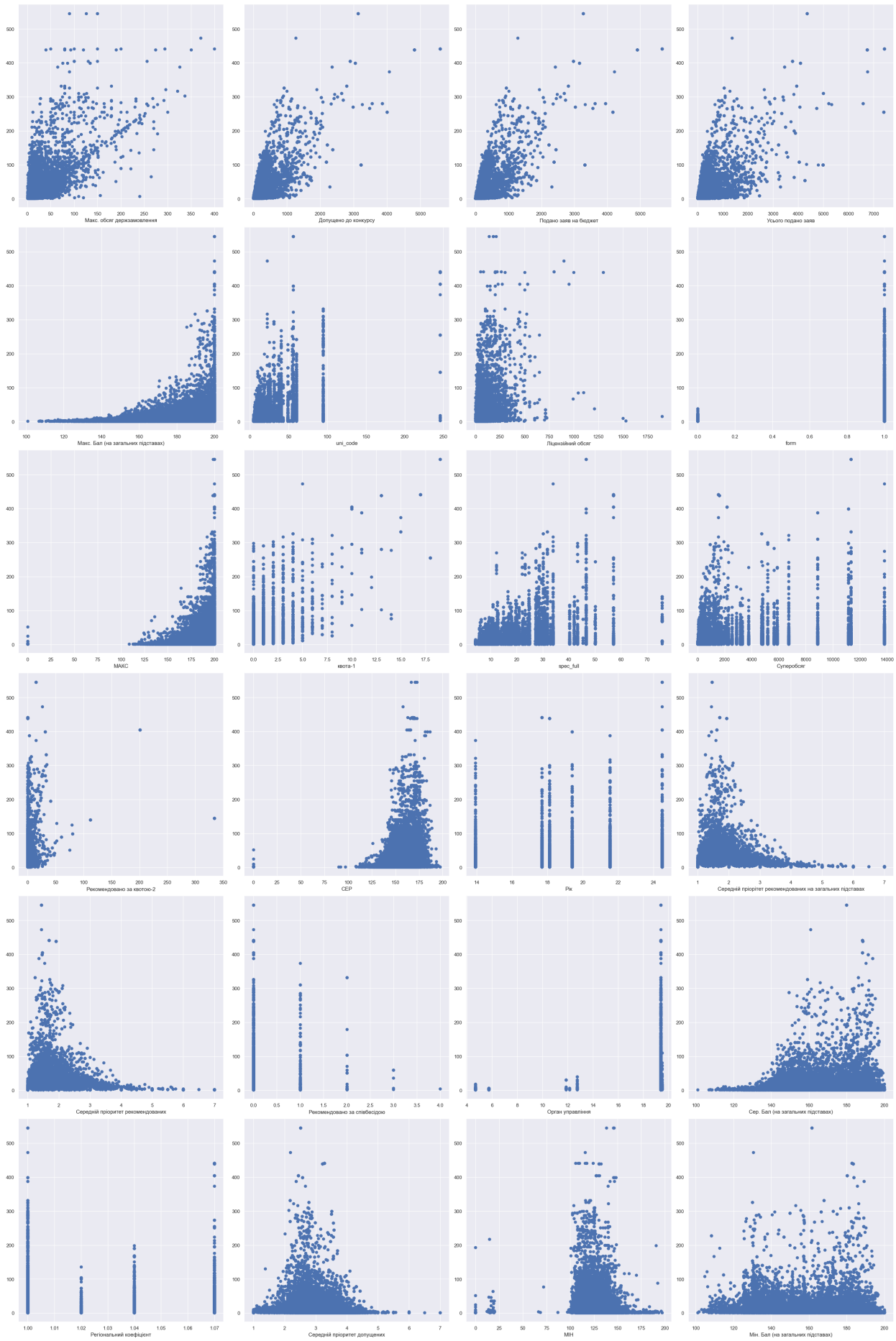


Рис. 3.5. Графіки взаємного розподілу параметрів і таргету

3.4 Коефіцієнти еластичності

Для оцінки впливу регресора на регресанд, не враховуючи одиниці їх виміру, використовується коефіцієнт еластичності. Цей коефіцієнт вказує, на скільки відсотків зміниться регресанд при збільшенні k-го регресора на один відсоток за умови, що інші фактори залишаються незмінними.

Спочатку навчимо багатофакторну лінійну регресійну модель

$$y = b_1 x_1 + b_2 x_2 + \ldots + b_n x_n + \varepsilon = \sum_{i=1}^{n} b_i x_i + \varepsilon \tag{3.3}$$

та оцінимо її якість за допомогою коефіцієнта детермінації R^2 та MSE (середньоквадратичної помилки):

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - y_i'\right)^2, \tag{3.4}$$

де

- y_i — значення цільової змінної;
- y_i' — передбачені моделлю значення цільової змінної.

Отримали результат:

```
MSE: 275.79128785392237
R2 score: 0.8222631534996667
```

Далі, використовуючи коефіцієнти моделі, розрахуємо коефіцієнти еластичності:

$$\widehat{\varepsilon}_k = \widehat{\beta}_k \frac{x_k^*}{y^*} \quad y^* \neq 0, \quad k = \overline{2, N}, \tag{3.5}$$

де x_k^*, y^*- значення k-го регресора і регресанда, що визначають точку регресійної функції, для якої розраховується коефіцієнт еластичності.

Результати представлені в таблиці:

Макс. Бал (на загальних підставах)	4.433053
Допущено до конкурсу	2.152362
СЕР	1.609303
Макс. обсяг держзамовлення	0.453709
uni_code	0.176948
Рік	0.172405
Середній пріорітет рекомендованих на загальних підставах	0.128261
spec_full	0.115771
Орган управління	0.096325
Суперобсяг	0.046732
form	0.038425
квота-1	0.033652
Рекомендовано за співбесідою	0.000843
Рекомендовано за квотою-2	-0.003014
Середній пріоритет рекомендованих	-0.004048
МІН	-0.105161
Ліцензійний обсяг	-0.138845
Усього подано заяв	-0.408145
Середній пріоритет допущених	-0.456461
Регіональний коефіцієнт	-0.793661
Подано заяв на бюджет	-1.228256
Сер. Бал (на загальних підставах)	-1.631084
МАКС	-1.819768
Мін. Бал (на загальних підставах)	-2.715525

Табл. 3.1. Коефіцієнти еластичності

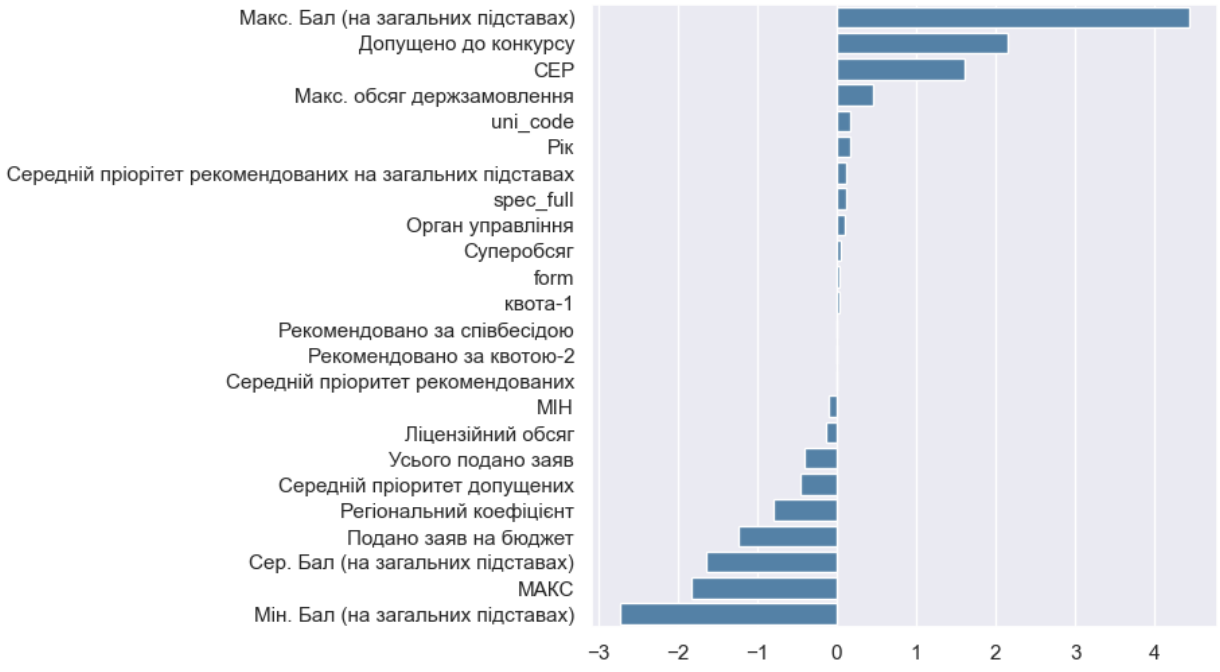


Рис. 3.6. Коефіцієнти еластичності

3.5 Часткові коефіцієнти детермінації

Розрахуємо ще одну ознаку впливовості регресора у моделі — часткові коефіцієнти детермінації. Вони показують, на яку величину зменшиться коефіцієнт детермінації, якщо якусь фічу виключити з моделі. Чим більший відповідний ΔR_k^2, тим більш впливовим є у моделі k-й регресор. Частковий коефіцієнт детермінації розраховується за формулою:

$$\Delta R_k^2 = \Delta R_{x_k}^2 = \frac{1 - R^2}{T - N} \left(\frac{\widehat{\beta}_k}{\widehat{\sigma}_{\widehat{\beta}_k}} \right)^2 \tag{3.6}$$

Результати представлені в таблиці:

Макс. обсяг держзамовлення	4.898698e-02
Усього подано заяв	8.330619e-03
uni_code	5.188016e-03
Ліцензійний обсяг	5.057401e-03
Макс. Бал (на загальних підставах)	4.791235e-03
Допущено до конкурсу	1.956863e-03
Середній пріоритет допущених	1.581942e-03
МАКС	1.569290e-03
Мін. Бал (на загальних підставах)	1.361797e-03
spec_full	1.076050e-03
Суперобсяг	8.479845e-04
СЕР	7.231165e-04
Подано заяв на бюджет	6.121923e-04
квота-1	5.227105e-04
Регіональний коефіцієнт	3.731534e-04
Рік	1.865689e-04
Сер. Бал (на загальних підставах)	1.548083e-04
Рекомендовано за квотою-2	7.794680e-05
Середній пріорітет рекомендованих на загальних підставах	7.595020e-05
form	4.765436e-05
МІН	1.571509e-05
Рекомендовано за співбесідою	1.515148e-05
Орган управління	9.994696e-06
Середній пріоритет рекомендованих	6.527172e-08

І візуалізація результатів:

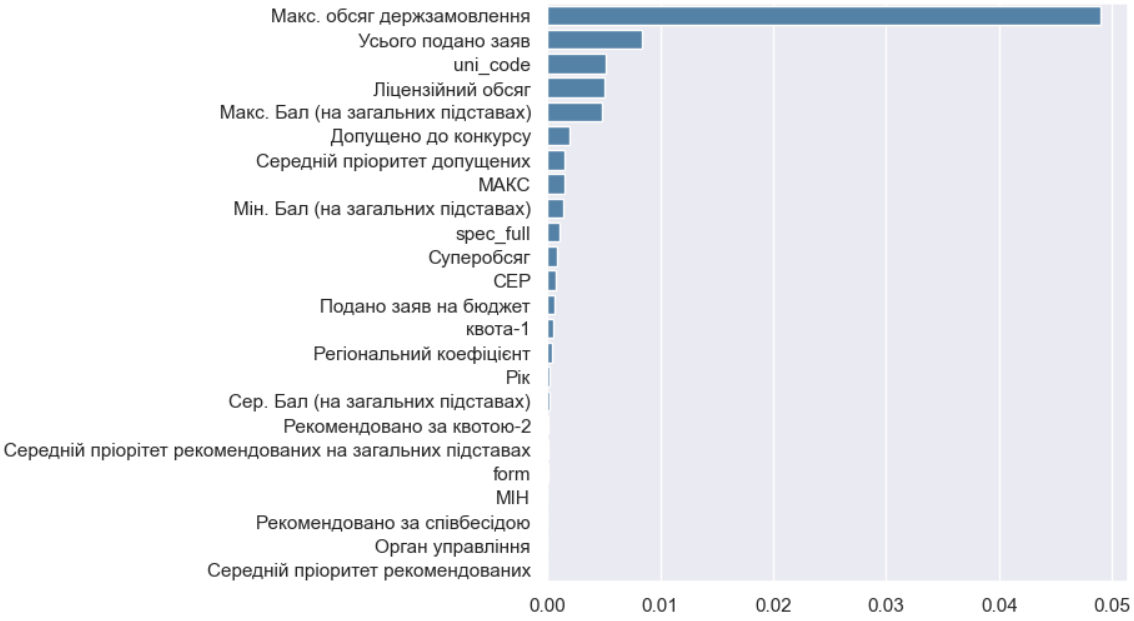


Рис. 3.7. Часткові коефіцієнти детермінації

РОЗДІЛ 4

ПОБУДОВА ТА НАЛАШТУВАННЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ

4.1 Лінійні моделі

Для того, щоб побудувати модель машинного навчання, ми використовуємо функцію втрат: чим менша функція втрат, тим краще. Звичайна модель лінійної регресії була побудована в попередньому розділі при розрахунку коефіцієнтів еластичності та часткових коефіцієнтів детермінації. Зараз розглянемо дві інші моделі: `Lasso` і `Ridge`.

Вони відрізняються від звичайної регресії тільки наявністю штрафу у функції втрат:

$$J_{LASSO} = \sum_i (y_i - \hat{y})^2 + \alpha \sum_i |w_i| \tag{4.1}$$

$$J_{RIDGE} = \sum_i (y_i - \hat{y})^2 + \alpha \sum_i w_i^2 \tag{4.2}$$

де α — гіперпараметр. Чим більша α, тим сильніше модель штрафується за величину коефіцієнтів і їхню кількість. Якщо α занулити, ми отримаємо звичайну функцію втрат методом найменших квадратів, відповідно — звичайну регресію. Тобто в `Lasso` і `Ridge` модель намагається знайти баланс між гарним передбаченням, що підходить під тренувальні дані, і не надто великою складністю моделі, коли ми використовуємо не всі фічі і не робимо коефіцієнти дуже великими. Зрозуміло, що чим довший вектор коефіцієнтів (тобто чим більше в ньому ми розглядаємо фічей) і чим більшими є ці коефіцієнти за модулем, тим сильніше штрафуватиметься модель.

Суттєва відмінність регресії `Lasso` від `Ridge` у тому, що `Lasso` зануляє коефіцієнти. Тобто буквально перед якимись фічами вона ставить 0 і в моделі вони не розглядаються. `Ridge` же може коефіцієнт сильно зменшити, але не занулити.

4.1.1 Підбір гіперпараметрів, регуляризація та відсіювання маловпливових ознак

Як уже було сказано, у формулах функції втрат в `Lasso` і `Ridge` регресіях присутній гіперпараметр α, який ми можемо налаштовувати вручну. Побудуємо спочатку `Lasso`-регресію та подивимось, які фічі вона вважає невпливовими. Для визначення найкращого значення параметра α для моделі `Lasso` скористуємося `LassoCV`. Будемо шукати найкраще α серед 500 значень на проміжку [0.001; 5].

В результаті отримали найкраще значення $\alpha = 0.04107214428857715$ і відповідні значення коефіцієнтів:

Середній пріорітет рекомендованих на загальних підставах	1.119885
квота-1	1.037890
Макс. обсяг держзамовлення	0.530953
Макс. Бал (на загальних підставах)	0.477673
form	0.262565
Допущено до конкурсу	0.257500
СЕР	0.196068
uni_code	0.175880
Рік	0.173821
spec_full	0.110427
Орган управління	0.052310
Суперобсяг	0.000410
Рекомендовано за співбесідою	0.000000
Середній пріоритет рекомендованих	0.000000
Регіональний коефіцієнт	-0.000000
МІН	-0.017047
Усього подано заяв	-0.031476
Ліцензійний обсяг	-0.049814
Рекомендовано за квотою-2	-0.051814
Подано заяв на бюджет	-0.143111
МАКС	-0.187122
Сер. Бал (на загальних підставах)	-0.187981
Мін. Бал (на загальних підставах)	-0.327073
Середній пріоритет допущених	-2.717804

Табл. 4.1. Значення коефіцієнтів в моделі Lasso

Ті параметри, напроти яких стоїть 0, найменше впливають на таргет.

`Найменш важливі фічі: 'Середній пріоритет рекомендованих', 'Рекомендовано за співбесідою', 'Регіональний коефіцієнт'.`

Побудуємо модель `Ridge` з урахуванням отриманих результатів (тобто без маловпливових фіч) та одразу знайдемо оптимальний параметр α за допомогою `RidgeCV`. Найкраще значення для α будемо шукати серед 2000 значень на проміжку [0.01; 1000].

Отримали наступні результати:

Середній пріорітет рекомендованих на загальних підставах	1.151872
квота-1	1.075106
form	0.613948
Макс. обсяг держзамовлення	0.530231
Макс. Бал (на загальних підставах)	0.477355
Допущено до конкурсу	0.260255
СЕР	0.200119
Рік	0.178662
uni_code	0.176887
spec_full	0.112935
Орган управління	0.083278
Суперобсяг	0.000398
МІН	-0.017940
Усього подано заяв	-0.031437
Ліцензійний обсяг	-0.049782
Рекомендовано за квотою-2	-0.053281
Подано заяв на бюджет	-0.146005
МАКС	-0.190538
Сер. Бал (на загальних підставах)	-0.192412
Мін. Бал (на загальних підставах)	-0.324491
Середній пріоритет допущених	-2.771674
Мін. Бал (на загальних підставах)	-0.327073
Середній пріоритет допущених	-2.717804

Табл. 4.2. Значення коефіцієнтів в моделі Ridge

при $\alpha = 262.138444422211105$.

4.1.2 Результати навчання моделей

Порівняємо якість побудованих моделей за допомогою MSE та R^2:

```
Linreg:  MSE = 275.79128785392237   R2 = 0.8222631534996667
Lasso:   MSE = 275.89762664590637   R2 = 0.8221946222502028
Ridge:   MSE = 275.97643746072777   R2 = 0.8221438317201412
```

Як бачимо, різниця несуттєва, але звичайна лінійна регресія показує трохи кращі результати.

Результати цього розділу були апробовані на міжнародній науково-практичній конференції «Інформаційні технології і автоматизація – 2023» та опубліковані у вигляді тез [4].

4.2 Дерево рішень

Дерево рішень — це модель, яка використовує деревоподібну структуру для прогнозування. Воно нагадує дерево, а процес прийняття рішень відбувається шляхом задавання серії питань. Дерево рішень розбиває дані на кілька наборів, кожен з яких поділяється на піднабори для досягнення рішення.

- Дерева рішень можна використовувати як для навчання з учителем, так і для навчання без учителя.
- Вони допускають як категоріальні, так і числові дані.
- Дерева рішень можуть обробляти пропущені значення настільки ж легко, як і звичайні значення змінної.
- Вони допомагають вибирати найважливіші змінні, оцінюючи зниження нечистоти для кожної змінної.
- Дерева рішень можуть виявляти складні залежності між змінними і добре працюють у випадках, коли неможливо побудувати одну лінійну залежність між цільовою змінною і змінними-ознаками.
- Вони не вимагають нормалізації даних і добре справляються з мультиколінеарністю (взаємозалежністю змінних).

Алгоритм побудови дерева рішень:

1) **Визначення проблеми**: Початковий етап, де визначається проблема або завдання, для якого потрібно знайти рішення.
2) **Визначення варіантів**: Всі можливі варіанти дій або рішень ідентифікуються та додаються до дерева рішень як гілки.
3) **Оцінка наслідків**: Для кожного варіанту дії оцінюються можливі наслідки, які можуть виникнути.
4) **Вибір оптимального шляху**: Шлях, який має найкращі наслідки або є найбільш прийнятним, обирається в якості рішення.

При побудові дерева рішень використовують певні алгоритми для визначення розбиття вузла на два або більше підвузли. Серед найпоширеніших алгоритмів можна виділити:

- **Ентропія (Entropy)**: Ентропія вимірює ступінь неупорядкованості або нечистоти в наборі даних. Вона розраховується для кожного розбиття в дереві і вибирається розбиття з найменшою ентропією або найбільшою чистотою.
- **Індекс Джині (Gini Index)**: вимірює ймовірність неправильної класифікації елемента, якщо вибрати класифікацію випадковим чином.
- **Помилка класифікації (Misclassification Error)**: простий метод, який обчислює частку неправильно класифікованих прикладів в підмножині даних. Вибирається розбиття з найменшою помилкою класифікації.
- **Інформаційний приріст (Information Gain)**: вимірює, наскільки добре дана ознака розділяє тренувальні приклади за їхнім класифікаційним показником. Вибирається розбиття з найбільшим приростом інформації.
- **Середнє квадратичне відхилення (Mean Squared Error, MSE)**: вимірює середньоквадратичну помилку між фактичними та передбаченими значеннями. Вибирається розбиття з найменшим MSE.
- **Середнє абсолютне відхилення (Mean Absolute Error, MAE)**: вимірює середню абсолютну помилку між фактичними та передбаченими значеннями. Вибирається розбиття з найменшим MAE.

4.2.1 Підбір гіперпараметрів

У файлі `decision_tree.ipynb` [3] представлений код для пошуку оптимальних гіперпараметрів моделі регресійного дерева. Ці параметри включають максимальну глибину дерева, мінімальну кількість зразків у листовому вузлі, максимальну кількість ознак для розгляду на кожному розбитті та максимальну кількість листових вузлів. Пошук оптимальних значень параметрів моделі виконується, оцінюючи модель з різними комбінаціями параметрів на тестових даних. Оцінка здійснюється за допомогою середньоквадратичної помилки (MSE), що вказує на точність моделі. У результаті отримали найкращу модель:

```
DecisionTreeRegressor(random_state=1, max_depth=19, max_features=7)
MSE: 93.80876088723575  R2 score: 0.9395438722377902
```

Цей результат уже значно кращий за той, що показували лінійні моделі!

Також модель `DecisionTreeRegressor` дозволяє подивитися на важливість кожного з параметрів (фічей) завдяки атрибуту `feature_importances_`:

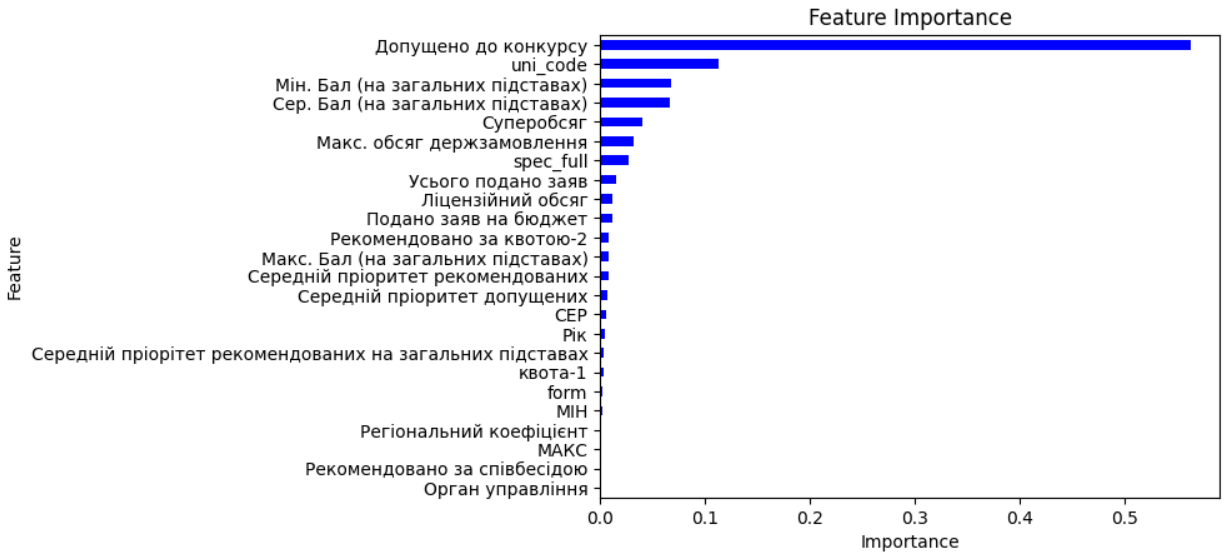


Рис. 4.1. Важливість фічей для дерева рішень

4.3 Випадковий ліс

Випадковий ліс є потужним алгоритмом машинного навчання, який використовує комбінацію багатьох дерев рішень для прогнозування. Він поєднує поняття багатьох дерев рішень у єдиний модельний ансамбль для отримання кращих прогнозів.

- Випадковий ліс може використовуватись як для задач регресії, так і для класифікації.
- Він використовує метод «багатократного дерева рішень» (bagging), що дозволяє будувати кілька дерев рішень на підмножинах даних та комбінує їх прогнози для отримання кінцевого результату.
- Кожне дерево в випадковому лісі побудоване на випадковій підмножині даних, а також випадково вибраних ознак.
- Застосування випадковості дозволяє зменшити кореляцію між деревами та забезпечує кращу узагальнюючу здатність моделі.
- Випадковий ліс може автоматично обробляти відсутні дані та розріджені дані.
- Він ефективно працює з великими наборами даних та багатовимірними просторами ознак.

Алгоритм побудови випадкового лісу:

1) **Вибір випадкової підмножини даних**: З кожного з наборів даних вибирається випадкова підмножина, яка буде використовуватись для побудови окремого дерева рішень.
2) **Побудова дерев рішень**: На вибраній підмножині даних будується дерево рішень за допомогою алгоритму, такого як ID3, C4.5 або CART.
3) **Повторення кроків 1-2**: Кроки 1 і 2 повторюються кілька разів, щоб побудувати багато дерев рішень.
4) **Прогнозування**: Прогнози кожного дерева комбінуються, наприклад, шляхом усереднення (для задач регресії) або голосування більшості (для задач класифікації), щоб отримати кінцевий результат.

ID3, C4.5 і CART є популярними алгоритмами для побудови дерев рішень в задачах класифікації та регресії.

- **ID3 (Iterative Dichotomiser 3)**: Цей алгоритм використовується для побудови дерев рішень у задачах класифікації. Він працює шляхом рекурсивного розбиття набору даних на більші та менші підмножини на основі значень ознак. ID3 вибирає ознаку, яка найкраще розділяє набір даних, використовуючи метрику ентропії або інші критерії.
- **C4.5**: Цей алгоритм є розширенням ID3 і також використовується для побудови дерев рішень у задачах класифікації. Однак C4.5 використовує інформаційний приріст (information gain) як критерій вибору ознаки для розділення вузла. Крім того, C4.5 може обробляти атрибути з відсутніми значеннями.
- **CART (Classification and Regression Trees)**: Цей алгоритм може використовуватися як для задач класифікації, так і для регресії. CART використовує критерій Джині для вибору ознаки для розділення вузла при класифікації та середнє квадратичне відхилення (MSE) при регресії. CART будує бінарне дерево, де кожен внутрішній вузол розділяється на дві дочірні гілки.

Випадковий ліс є потужним і гнучким методом машинного навчання, який широко використовується в багатьох областях, включаючи прогнозування, класифікацію, візуалізацію даних та виявлення аномалій.

У випадку використання випадкового лісу для регресії, кожне дерево в лісі використовується для прогнозування числової величини. Кінцевий прогноз отримується шляхом усереднення прогнозів всіх дерев.

Випадковий ліс має кілька переваг:

- Він є ефективним алгоритмом з малою кількістю гіперпараметрів, що спрощує його використання та налаштування.
- Добре працює з великими наборами даних, оскільки може бути паралельно здійснений над багатьма деревами.
- Може автоматично обробляти відсутні дані та розріджені дані, що зменшує необхідність у попередній обробці даних.
- Має високу узагальнюючу здатність та знижує ризик перенавчання.

Деякі з недоліків випадкового лісу включають:

- Він може бути витратним за ресурсами, особливо якщо кількість дерев в лісі велика.
- Він може мати погану відтворюваність, оскільки випадковість в процесі побудови дерев може призводити до різних результатів при кожному запуску.
- Він може бути схильним до перенавчання, якщо кількість дерев в лісі дуже велика.
- Він може бути складним для інтерпретації, оскільки важко зрозуміти, як саме приймаються рішеня.

4.3.1 Підбір гіперпараметрів

У наведеному в файлі `random_forest.ipynb` [3] коді спочатку навчається модель випадкового лісу регресії з параметрами за замовчуванням. І навіть з цими параметрами за замовчуванням модель показує результат майже в два рази кращий за результат одного дерева рішень:

```
MSE: 53.05728123282293  R2 score: 0.9613631159140079
```

Потім за допомогою `GridSearchCV` виконується пошук оптимальних гіперпараметрів для моделі, таких як максимальна глибина дерев, кількість дерев у лісі та максимальна кількість ознак для розгляду при розбитті. Після знаходження найкращих параметрів модель знову навчається і оцінюється на тестових даних. Остаточно, отримали найкращу модель:

```
RandomForestRegressor(random_state=1, n_jobs=-1,
n_estimators=151, max_depth=16, max_features=13)
MSE: 47.67195155832302  R2 score: 0.9652607765396858
```

4.4 Gradient Boosting

Бустинг — це метод ансамблю в машинному навчанні, який покращує точність прогнозування моделей шляхом комбінування декількох слабких моделей в одну сильну. Основна ідея полягає в тому, що кожна наступна модель намагається виправити помилки своїх попередників.

Основні види бустингу:

- **Адаптивний бустинг (AdaBoost)** - це рання модель бустингу, яка адаптується і коригує класифікатори. Він ефективний для завдань класифікації, але менш ефективний при кореляції між ознаками.
- **Градієнтний бустинг (GB)** - це метод послідовного навчання, який оптимізує функцію втрат. Він підходить для завдань класифікації та регресії і дає більш точні результати, ніж AdaBoost.
- **Екстремальний градієнтний бустинг (XGBoost)** - це покращена версія градієнтного бустингу, яка фокусується на швидкості обчислень та масштабах моделі. Він ефективний для роботи з великими даними.
- **LightGBM (Light Gradient Boosting Machine)** - це безкоштовний та відкритий для загального користування розподілений фреймворк для машинного навчання, який було розроблено компанією Microsoft. Він базується на алгоритмах дерев рішень і використовується для ранжування, класифікації та інших завдань машинного навчання.

Основні переваги бустингу:

- **Простота реалізації**: бустинг має прості для розуміння та інтерпретації алгоритми, здатні вчитися на своїх помилках. Ці алгоритми не вимагають попередньої обробки даних, а також мають вбудовані процедури обробки відсутніх значень. Крім того, у більшість мов вбудовані бібліотеки для реалізації алгоритмів бустингу з безліччю параметрів, що дозволяють точно задавати продуктивність.
- **Зменшення упередженості**: упередженість (bias) — це наявність невизначеності або неточності в результатах машинного навчання.

Алгоритми бустингу об'єднують кілька слабких моделей у послідовний метод, ітеративно покращуючи спостереження. Такий підхід допомагає зменшити високу упередженість, яка поширена в моделях машинного навчання.

- **Ефективність алгоритмів**: алгоритми бустингу фокусуються на елементах, що підвищують точність прогнозування під час навчання. Вони здатні зменшувати кількість атрибутів даних та ефективно обробляти великі набори.

Основні недоліки бустингу:

- **Чутливість до викидів у даних**: моделі бустингу чутливі до викидів або значень, що відрізняються від інших даних у датасеті. Викиди можуть суттєво спотворювати результати, оскільки кожна модель намагається виправити помилки попередньої.
- **Реалізація в режимі реального часу**: оскільки цей алгоритм складніший за інші процеси, при реалізації бустингу в режимі реального часу можуть виникнути труднощі. Бустинг проявляє високу адаптивність, тому можна використовувати різноманітні параметри моделі, безпосередньо впливаючі на її продуктивність.

4.4.1 GradientBoostingRegressor (Scikit-learn)

Градієнтний бустинг (GB) схожий на AdaBoost: він також є методом послідовного навчання. Різниця між AdaBoost та GB в тому, що GB не призначає неправильно класифікованим елементам більшу вагу. Замість цього програмне забезпечення GB оптимізує функцію втрат через послідовне генерування базових моделей, в результаті чого поточна базова модель завжди стає ефективнішою попередньої. На відміну від AdaBoost, метод GB намагається відразу генерувати точні результати, а не виправляти помилки. З цієї причини метод GB дає більш точні результати. Градієнтний бустинг підходить і для завдань класифікації, і для регресії.

В бібліотеці Scikit-learn GradientBoostingRegressor має кілька параметрів, які можна налаштовувати, включаючи кількість базових моделей (де-

рев рішень), швидкість навчання (learning rate), глибину дерев (max_depth) та інші. Ці параметри дозволяють контролювати процес навчання і підгонку моделі під конкретні дані.

Під час оцінки моделі з параметрами за замовчуванням, отримано наступні результати:

```
MSE: 95.50899046748748 R2 score: 0.9384481398588886
```

Після цього виконується підбір оптимальних гіперпараметрів за допомогою GridSearchCV. Найкращі знайдені параметри та результати оцінки моделі з цими параметрами:

```
{'learning_rate': 0.1, 'max_depth': 5, 'max_features': 12,
 'n_estimators': 5000, 'random_state': 1}
MSE: 31.840338648765577  R2 score: 0.9794801299672244
```

Ці результати показують покращення якості моделі після підбору оптимальних гіперпараметрів. Також є можливість вивести важливість фічей:

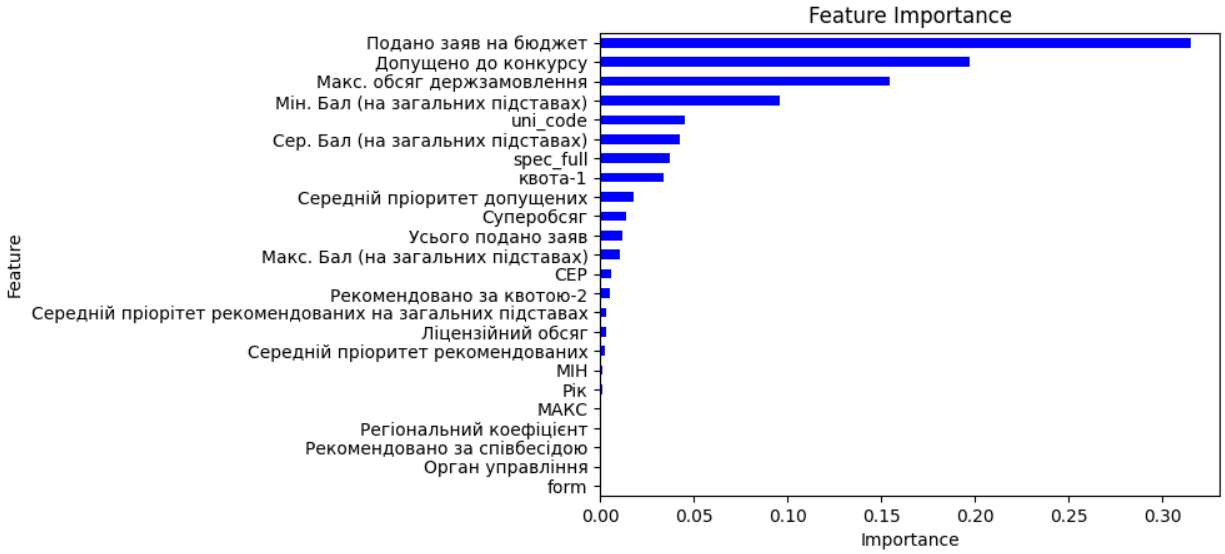


Рис. 4.2. Важливість фічей для GradientBoostingRegressor

4.4.2 XGBRegressor (XGBoost)

Екстремальний градієнтний бустинг (XGBoost) різними способами покращує градієнтний бустинг, фокусуючись на швидкості обчислень

та масштабах моделі. XGBoost розроблений для ефективної багатоядерної паралельної обробки прямо під час навчання. Цей алгоритм бустингу є ефективним інструментом для роботи з великими даними. Ключовими особливостями XGBoost є паралелізація, розподілені обчислення, оптимізація кешу та зовнішні обчислення.

Порівняно з традиційним градієнтним бустингом, XGBoost пропонує ряд інноваційних підходів, що сприяють покращенню швидкості та якості моделі. Однією з ключових особливостей XGBoost є його ефективність у використанні багатоядерних систем та паралельних обчислень. Це дозволяє прискорити процес навчання моделі, особливо в разі великих обсягів даних.

Додатково, XGBoost використовує оптимізацію кешу та зовнішні обчислення, що сприяє подальшому збільшенню продуктивності та швидкості алгоритму. Ці техніки роблять XGBoost ефективним інструментом для роботи з великими обсягами даних та високорозмірними моделями.

XGBRegressor також надає можливість налаштовувати різні параметри моделі, такі як глибина дерев, швидкість навчання та інші, для досягнення оптимальних результатів в конкретній задачі регресії.

Для моделі XGBRegressor проводиться аналогічний процес підбору гіперпараметрів, що і для GradientBoostingRegressor. Також виконується аналіз впливу гіперпараметрів, таких як максимальна глибина дерева та базовий рейтинг, на якість моделі. Найкращі знайдені параметри та результати оцінки моделі такі:

```
{'base_score': 0.05, 'learning_rate': 0.1, 'max_depth': 5,
 'n_estimators': 5000, 'seed': 1, 'tree_method': 'exact'}
MSE: 34.872211846994205  R2 score: 0.977526204644074
```

4.4.3 LGBMRegressor (LightGBM)

LightGBM (Light Gradient Boosting Machine) — це градієнтний бустінговий алгоритм, розроблений компанією Microsoft, який використовується для задач регресії. Він базується на деревах прийняття рішень і використовує техніку зменшення градієнта для навчання ансамблю дерев [11]. LightGBM є одним з найшвидших і ефективних реалізацій градієнтного бустінгу.

Він використовує дві нові техніки, названі Gradient-Based One-Side Sampling (GOSS) та Exclusive Feature Bundling (EFB), які дозволяють алгоритму працювати швидше, зберігаючи високий рівень точності.

LightGBM розширює алгоритм градієнтного бустингу, додаючи тип автоматичного вибору функцій, а також зосереджуючись на підсиленні прикладів з більшими градієнтами3. Це може призвести до значного прискорення навчання та покращення прогнозувальної продуктивності

Основні відмінності:

- **Швидкодія**: LightGBM працює швидше за багато інших бібліотек, таких як XGBoost або GradientBoostingRegressor, завдяки своєму унікальному алгоритму побудови дерев і оптимізації.
- **Розподілене навчання**: LightGBM підтримує розподілене навчання моделі на кількох вузлах, що дозволяє обробляти великі обсяги даних.
- **Оптимізація категоріальних ознак**: Має вбудовану підтримку для категоріальних ознак і може оптимізувати роботу з ними без необхідності попередньої обробки.

Аналогічно до попередніх моделей, LightGBM дозволяє налаштовувати гіперпараметри. Після підбору параметрів та оцінки моделі найкращі знайдені параметри та результати оцінки моделі такі:

```
{'learning_rate': 0.05, 'max_depth': 8, 'n_estimators': 5000,
 'n_jobs': -1, 'num_leaves': 11, 'random_state': 1}
MSE: 41.69977555287888      R2 score: 0.9731261032057471
```

Порівняння бустингових моделей

Порівняємо значення MSE та R2 для кожної з трьох моделей градієнтного бустингу:

Модель	MSE	R2 score
`GradientBoostingRegressor`	31.8403	0.9795
`XGBRegressor`	34.8722	0.9775
`LGBMRegressor`	41.6998	0.9731

Табл. 4.3. Порівняння оцінок точності для моделей градієнтного бустингу

Ці значення показують, що модель `GradientBoostingRegressor` має найменше значення MSE та найбільше значення R2 score, що вказує на кращу точність передбачень порівняно з `XGBRegressor` та `LGBMRegressor`. Але, враховуючи час, витрачений на навчання кожної з них, віддамо перевагу моделі `XGBRegressor`, яка показує відмінний результат значно швидше.

4.5 Зменшення розмірностей

Зменшення розмірностей є важливим кроком в обробці даних, особливо коли ми маємо великі набори даних з великою кількістю змінних. Це допомагає видалити шум, покращити ефективність алгоритмів та спростити візуалізацію даних.

Зменшення розмірностей важливе також для отримання більш глибокого розуміння даних. Воно дозволяє виявляти скриті шаблони та зв'язки між змінними, які можуть бути неочевидними в високовимірному просторі. Крім того, зменшення розмірностей може допомогти виявити основні фактори, які впливають на наші дані, що може бути корисним для прийняття рішень та прогнозування.

4.5.1 Метод головних компонент (PCA)

Метод головних компонент (PCA) є одним з найпопулярніших методів зменшення розмірностей. Він використовується для виявлення кореляційних залежностей між ознаками та витягування головних напрямків в даних.

Основна ідея PCA полягає в тому, щоб знайти нові осі (головні компоненти) у просторі даних, які описують найбільшу дисперсію даних. Ці головні компоненти впорядковані за спаданням важливості, тобто перша головна компонента описує найбільшу дисперсію в даних, друга — наступну за величиною дисперсію і так далі.

Одним із способів вимірювати, наскільки добре кожна кількість головних компонент описує вихідні дані, є пояснена дисперсія. Значення поясненої дисперсії може знаходитись в інтервалі від 0 до 1. Зазвичай воно виражається у відсотках і показує, яку частку загальної дисперсії даних вдається пояснити з використанням вибраної кількості головних компонент.

Було проаналізовано різну кількість головних компонет і знайдено їх пояснену дисперсію. Результати представлені у наступній таблиці:

Кількість компонент	Пояснена дисперсія
1	0.24964384657679153
2	0.3751630645568822
3	0.47612280022965836
4	0.5396964993229907
5	0.5948737780849066
6	0.6435855434799927
7	0.6867219626625171
8	0.7271394019224467
9	0.7651179554014387
10	0.802301956887641
11	0.8354085907824268
12	0.867700275706746
13	0.8942933937812144
14	0.9164723050349604
15	0.9379024645053927
16	0.9542679993857073
17	0.9683023742514705
18	0.9819767752996434
19	0.9918554287723037
20	0.9963564715526677
21	0.9982548492727039
22	0.999482684810372
23	0.9999921959441181
24	1.0000000000000002

Табл. 4.4. Пояснена дисперсія для кожної кількості головних компонент

Можна проаналізувати кореляцію між вихідними фічами та головними компонентами для різної кількості головних компонент.

Розглянемо найбільш показові випадки: для однієї та двох головних компонент.

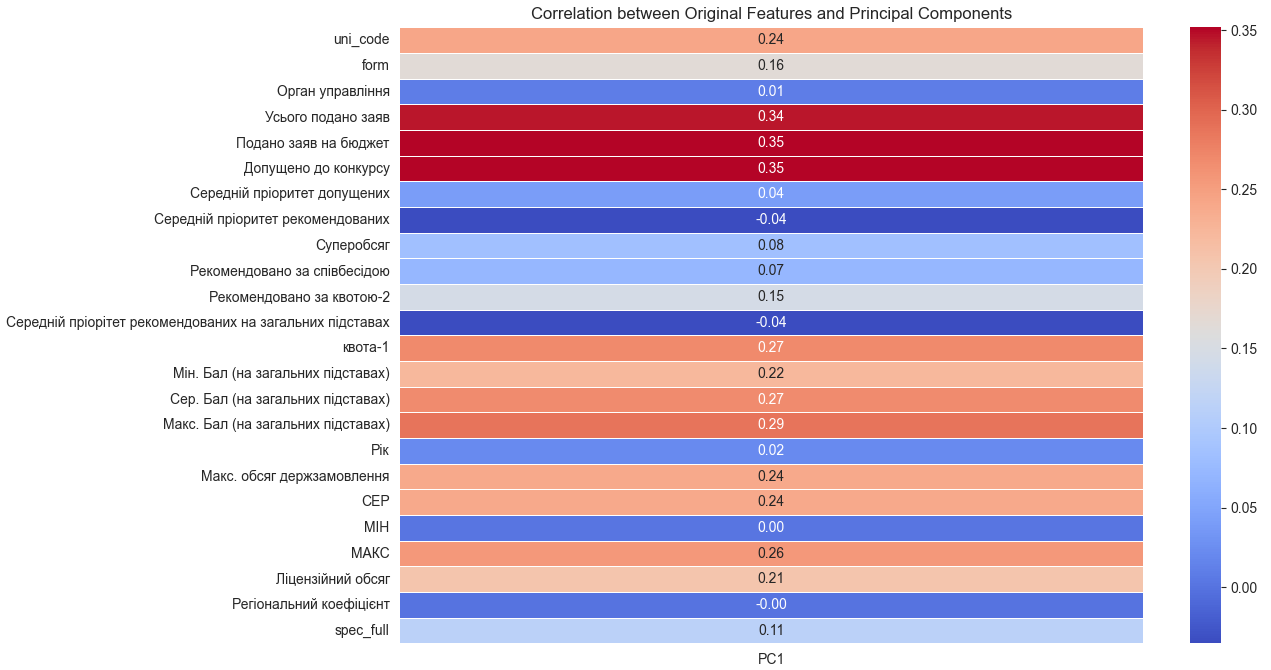


Рис. 4.3. Кореляція між фічами та однією головною компонентою

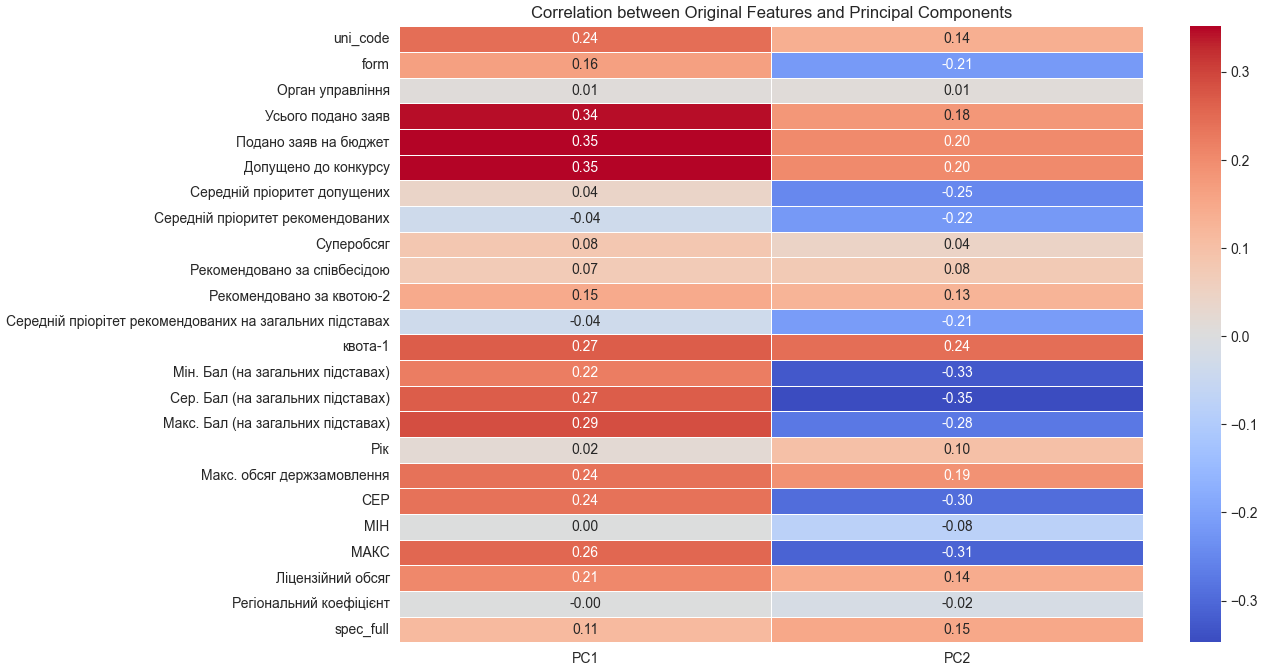


Рис. 4.4. Кореляція між фічами та двома головними компонентами

Як бачимо, при додаванні другої головної компоненти, перша залишилася незмінною, як і очікувалось. При цьому значення першої головної компоненти найбільше корелюють з такими фічами, як “Усього подано заяв”, “Подано заяв на бюджет”, “Допущено до конкурсу”, “Макс. Бал (на загальних підставах)”, “Сер. на загальних підставах” та “квота-1”. А в другій

головній компоненті спостерігається від'ємна кореляція з балами на загальних підставах, проте залишається достатньо висока кореляція з фічами "Усього подано заяв", "Подано заяв на бюджет", "Допущено до конкурсу".

Потім були побудовані моделі лінійної регресії, для навчання яких вікористовувалася різна кількість головних компонент. Порівняння оцінок якості MAE та R2 цих моделей представлена на графіку:

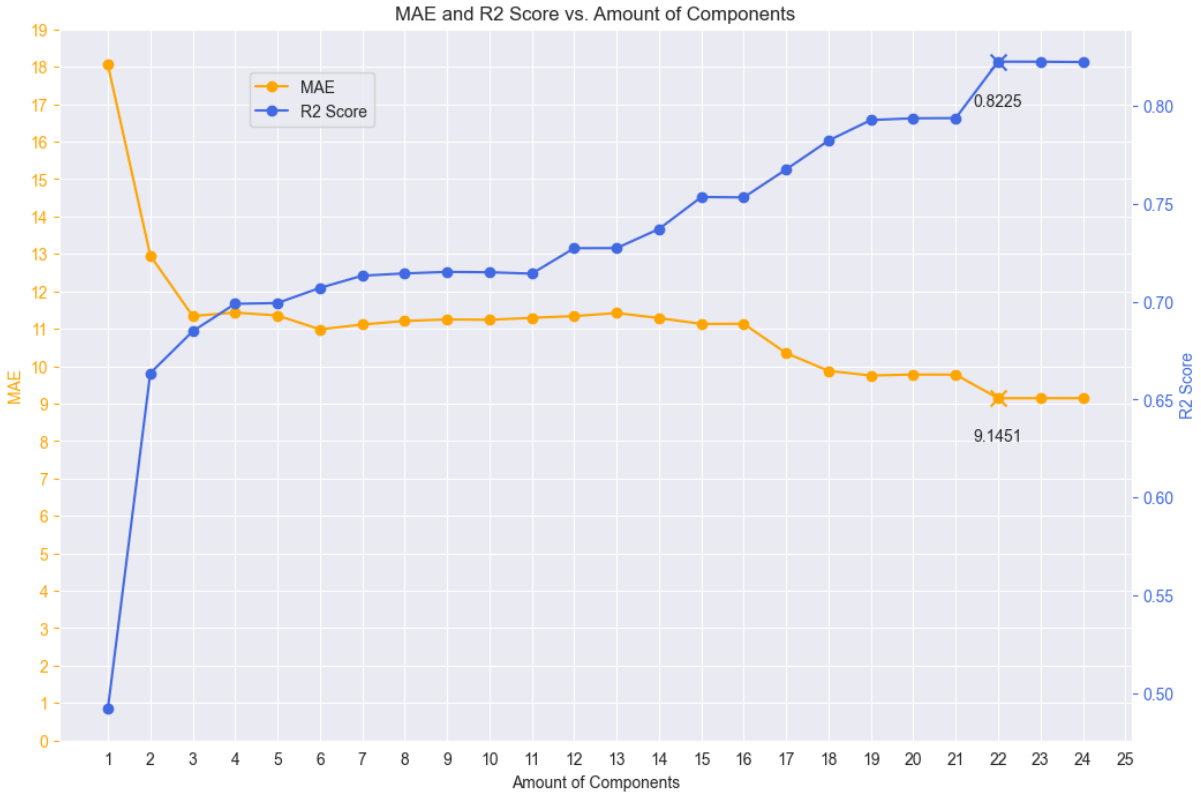


Рис. 4.5. MAE та R2 лінійної регресії від кількості головних компонент

Як можна помітити, навіть у найкращому випадку, коли використано 22 компоненти, якість моделі погіршується порівняно з навчанням на вихідних даних.

Далі було проведено аналогічне порівняння, але з використанням моделі `XGBRegressor`. Показники MSE та R2 моделей представлені на графіку:

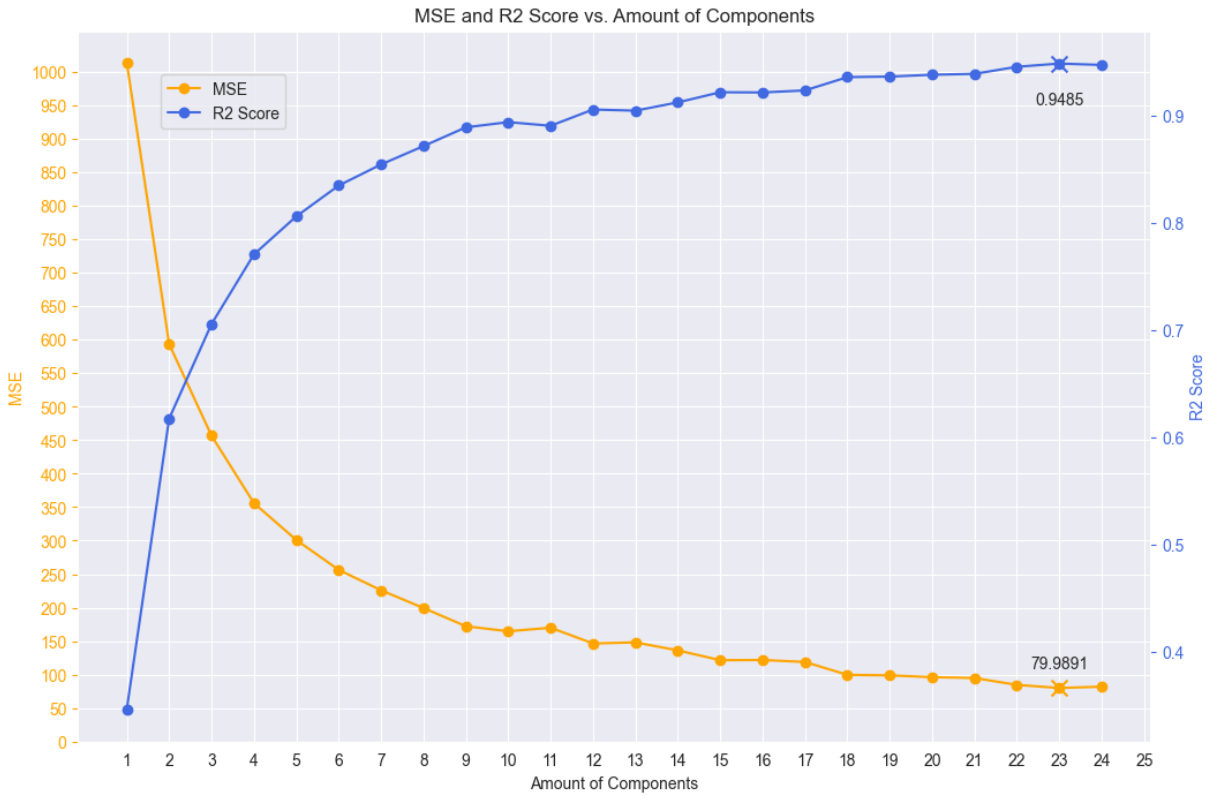


Рис. 4.6. MSE та R2 XGBRegressor від кількості головних компонент

Розкладання вихідних даних на головні компоненти не покращило якість моделі `XGBRegressor`.

Отже, на основі проведених експериментів можна зробити висновок, що використання головних компонент для розкладання вихідних даних не є ефективним підходом для прогнозування в даному контексті з використанням лінійної моделі регресії та моделі градієнтного бустінгу `XGBRegressor`.

4.5.2 Відбір найбільш впливових змінних

Відбір найбільш впливових змінних є альтернативним способом зменшення розмірності. Цей підхід полягає в визначенні набору змінних, які найефективніше відображають варіативність даних. Це можна зробити, використовуючи відбір на основі значимості змінних.

Ми будемо користуватися важливістю фічей, засновану на навчанні глибокого дерева та моделі `GradientBoostingRegressor`.

Спочатку навчимо дерево рішень з параметром `max_depth=1000` і подивимося на важливість фічей:

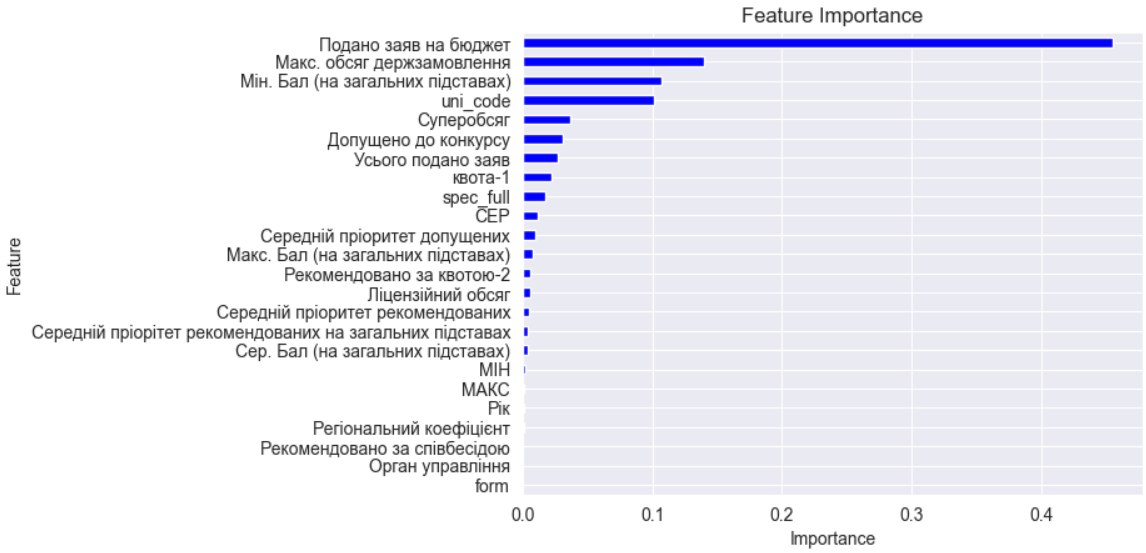


Рис. 4.7. Важливість фічей для глибокого дерева рішень

Тепер будемо додавати фічі одну за одною в порядку зменшення їх значимості, пропускати дані через наші моделі і дивитися, які результати ми отримаємо. Для наочності скористаємося метриками MSE та R2:

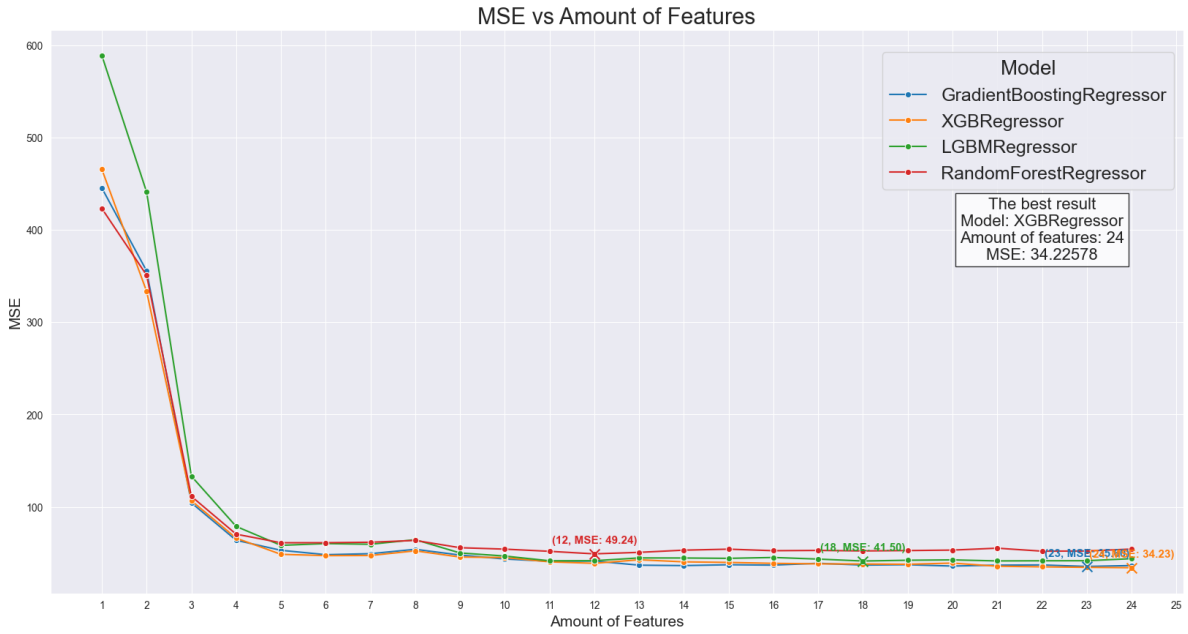


Рис. 4.8. MSE від кількості фічей (глибоке дерево рішень)

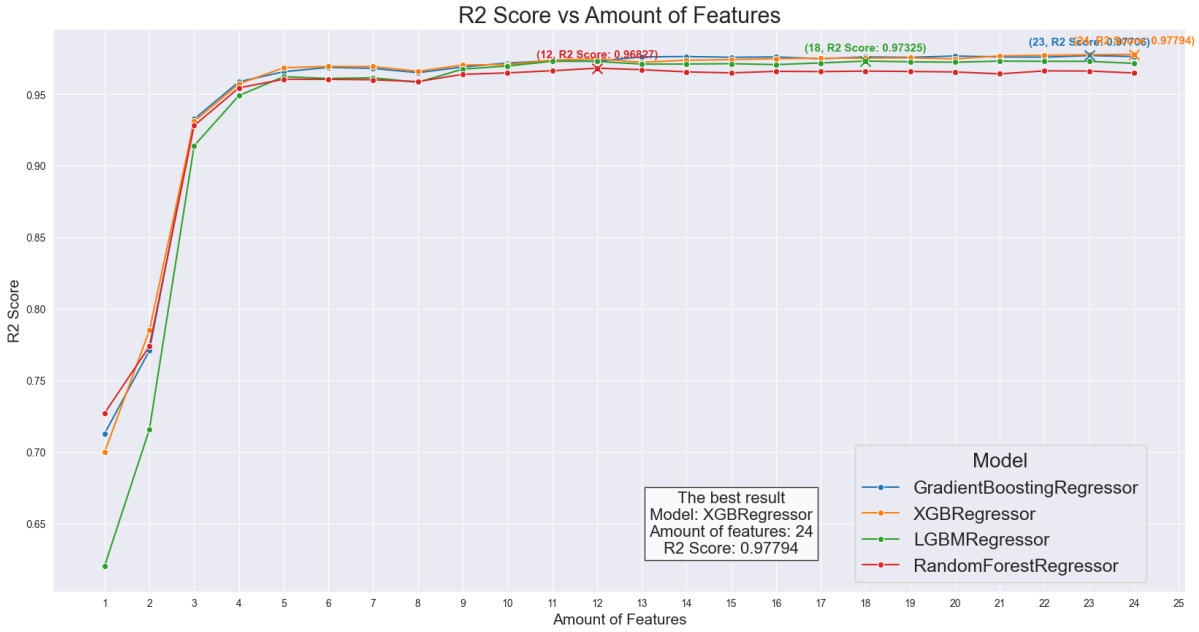


Рис. 4.9. R2 від кількості фічей (глибоке дерево рішень)

Як бачимо, MSE дуже стрімко падає, і R2 відповідно зростає при додаванні перших 5-ти фічей, а потім тримається на приблизно однаковому рівні.

Проведемо ту ж саму процедуру, але будемо додавати фічі на основі їх важливості у `GradientBoostingRegressor`.

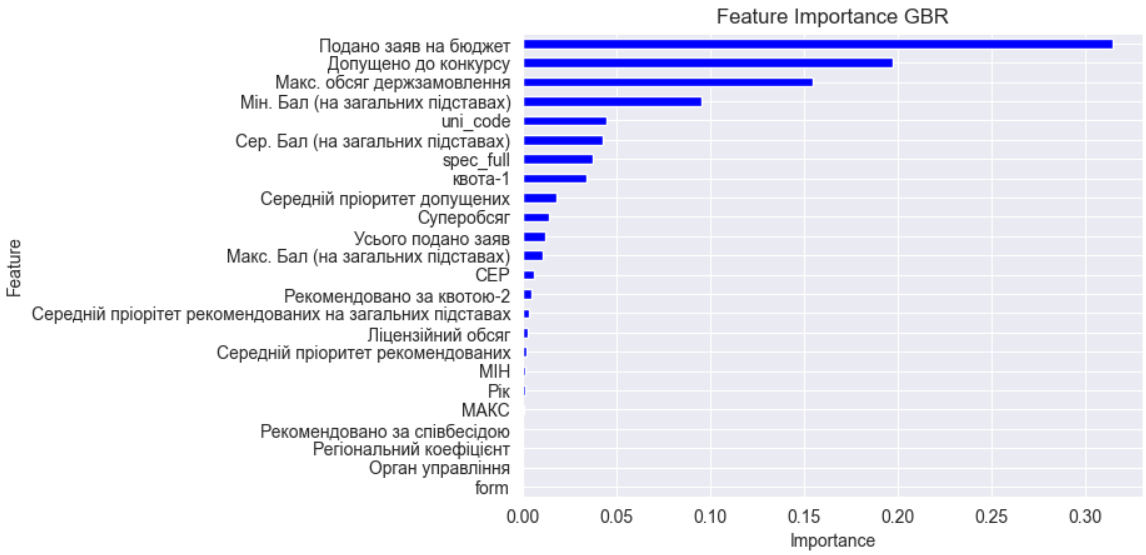


Рис. 4.10. Важливість фічей для моделі GradientBoostingRegressor

Результати представлені на графіках:

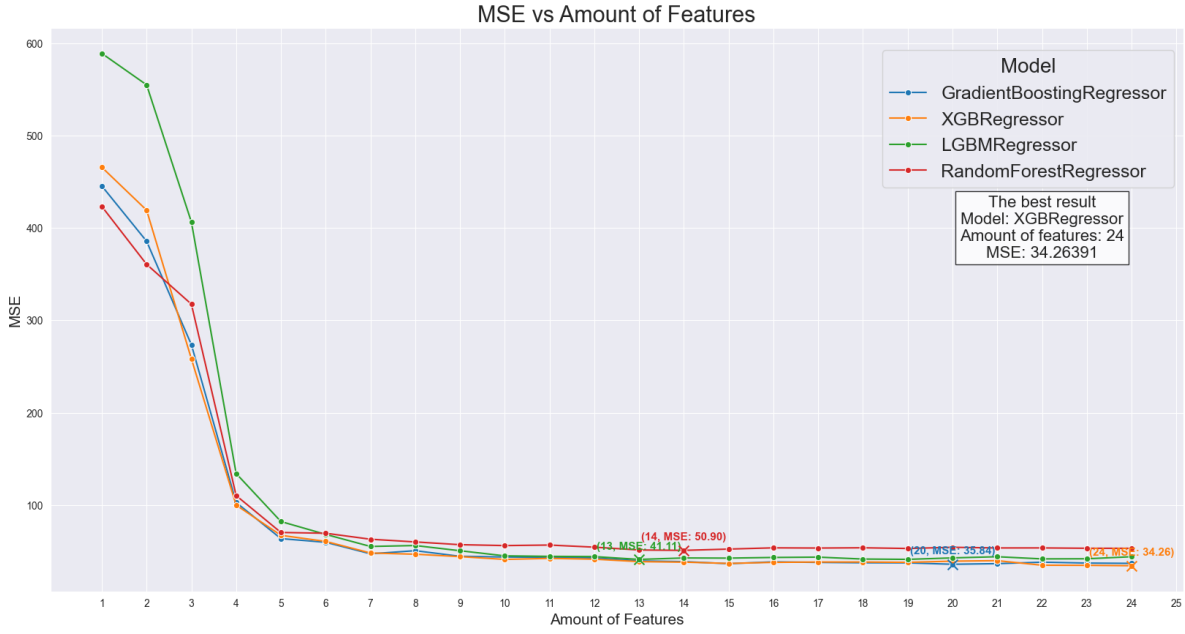


Рис. 4.11. MSE від кількості фічей (GradientBoostingRegressor)

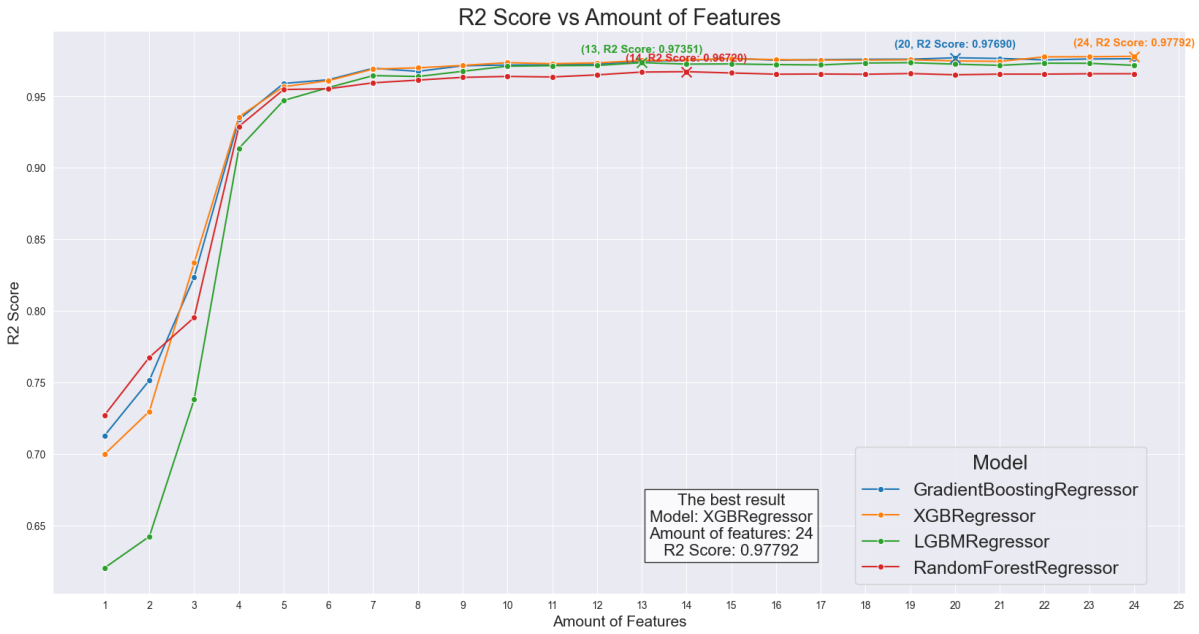


Рис. 4.12. R2 від кількості фічей (GradientBoostingRegressor)

Якщо уважно подивитись на графіки, то можна помітити, що для випадку, коли порядок фічей береться на основі глибокого дерева рішень, графік виходить більш сжатим до осі y. Це вказує на те, що дерево рішень дає більш точне уявлення про важливість фічей. Але починаючи з 5-ої фічі обидва графіки тримаються приблизно на одному рівні.

Порівняємо перші 5 фічей:

На основі дерева рішень	**На основі GradientBoostingRegressor**
Подано заяв на бюджет	Подано заяв на бюджет
Макс. обсяг держзамовлення	Допущено до конкурсу
Мін. Бал (на загальних підставах)	Макс. обсяг держзамовлення
uni_code	Мін. Бал (на загальних підставах)
Суперобсяг	uni_code

Таблиця показує, що результати в обох підходах мають деякі спільні фічі, але розташовані в різному порядку. Це може бути результатом того, що кожен метод використовує різні критерії для визначення важливості фічей. Однак при порівнянні перших фічей, все ж таки, наприклад, комбінація ***Подано заяв на бюджет***, ***Макс. обсяг держзамовлення*** та ***Мін. Бал (на загальних підставах)*** буде більш вигідною для передбачення, ніж ***Подано заяв на бюджет***, ***Допущено до конкурсу*** та ***Макс. обсяг держзамовлення***.

4.5.3 Навчання моделей з фічами, які відомі до фінальних результатів широкого конкурсу

Такі фічі, як «Усього рекомендовано», «Допущено до конкурсу», «Середній пріоритет допущених», «Сер. Бал (на загальних підставах)» і т.п. стають відомими вже після того, як стає відомо і цільову змінну. Тому для практичного використання результатів цього дослідження є сенс навчити модель, використовуючи лише ті фічі, які доступні до оприлюднення остаточних результатів вступної кампанії. Для цього будемо використовувати модель `XGBRegressor`, яка показала найкращі результати відносно якості та швидкості в попередніх експериментах.

Для початку, видалимо з датасету усі фічі, які неможливо дізнатися до закінчення вступної кампанії. З 24-х фічей залишилось 15, які перша тестова модель `XGBRegressor` розташувала у такому порядку за важливістю:

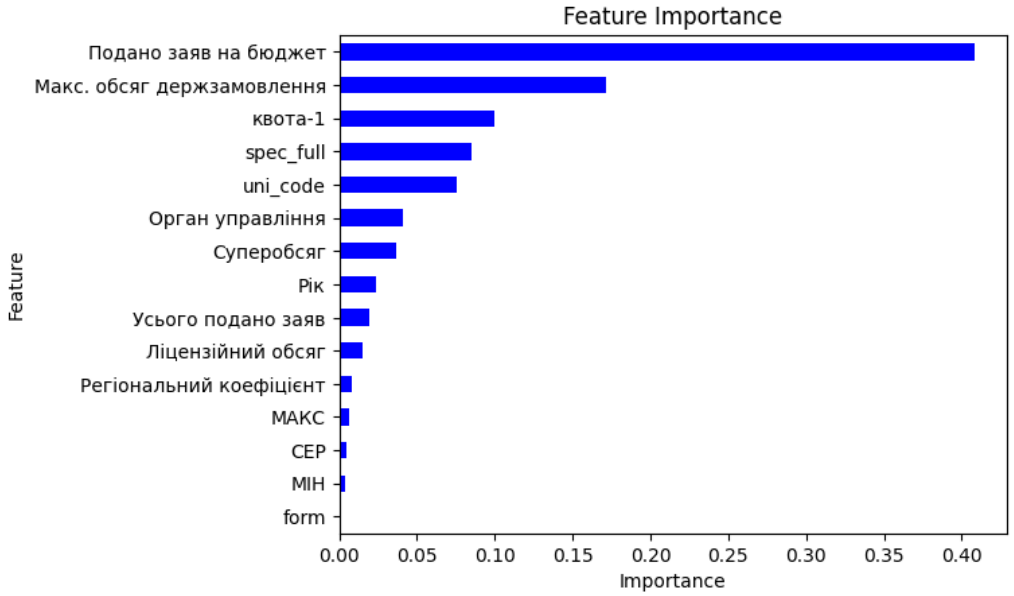


Рис. 4.13. Фічі, відомі до результатів широкого конкурсу

при наступних показниках самої моделі:

```
MSE: 58.748339457424315  R2 score: 0.9621389614097442
```

Далі підбираємо оптимальні значення гіперпараметрів для моделей з різною кількостю фічей:

- **learning_rate**: швидкість навчання, яка визначає, наскільки швидко модель адаптується до проблеми.
- **max_depth**: максимальна глибина дерева.
- **eta**: крок навчання, який використовується для зменшення ваги оновлень і забезпечення більшої стабільності.
- **lambda**: параметр L2-регуляризації, який додається до ваг для запобігання перенавчанню.
- **alpha**: параметр L1-регуляризації, який додається до ваг для запобігання перенавчанню.

Ці гіперпараметри підбираються для кожної можливої комбінації, а потім обчислюється середньоквадратична помилка (MSE). Найкращі гіперпараметри обираються на основі найменшої MSE.

В результаті експерименту було встановлено, що найкращі результати отримані за умови використання лише семи найбільш важливих фічей. Це свідчить про те, що інші ознаки не надають достатньої корисної інформації для покращення прогностичної здатності моделі. Такий підхід дозволяє спростити модель та зменшити її складність без втрати точності прогнозування. В результаті, модель стає більш інтерпретованою та менше схильною до перенавчання, що підвищує її узагальнюючу здатність.

При цьому важливість останніх фічей трохи змінилась:

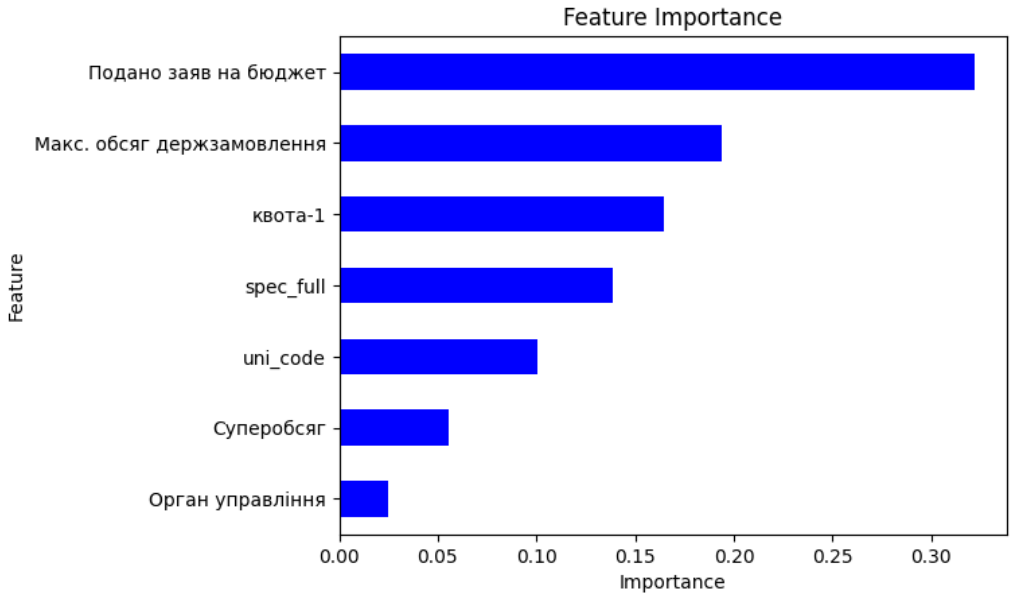


Рис. 4.14. 7 найважливіших фічей

при наступних показниках самої моделі:

```
MSE: 50.15804638833622  R2 score: 0.9676750739261849
```

Цю модель буде дуже зручно використовувати для створення додатка для прогнозування результатів вступної кампанії.

4.6 Порівняння отриманих результатів

Отже, після проведення ряду експериментів ми можемо порівняти результати та зробити деякі висновки.

Усі моделі були оцінені за допомогою таких метрик як середньоквадратична помилка (MSE) та коефіцієнт детермінації (R2). Ці метрики допомагають нам оцінити якість прогнозування моделі, її здатність узагальнювати дані та точність прогнозування.

В кінці для наочності було вирішено додати ще одну метрику — середня абсолютна помилка (MAE).

MAE — це корисна метрика для оцінки якості моделей регресії, оскільки вона надає пряме уявлення про те, наскільки близько прогнози моделі до фактичних значень. Вона вимірює середню величину помилок у прогнозах, без урахування напрямку.

На графіку наведено порівняльний аналіз усіх розглянутих моделей:

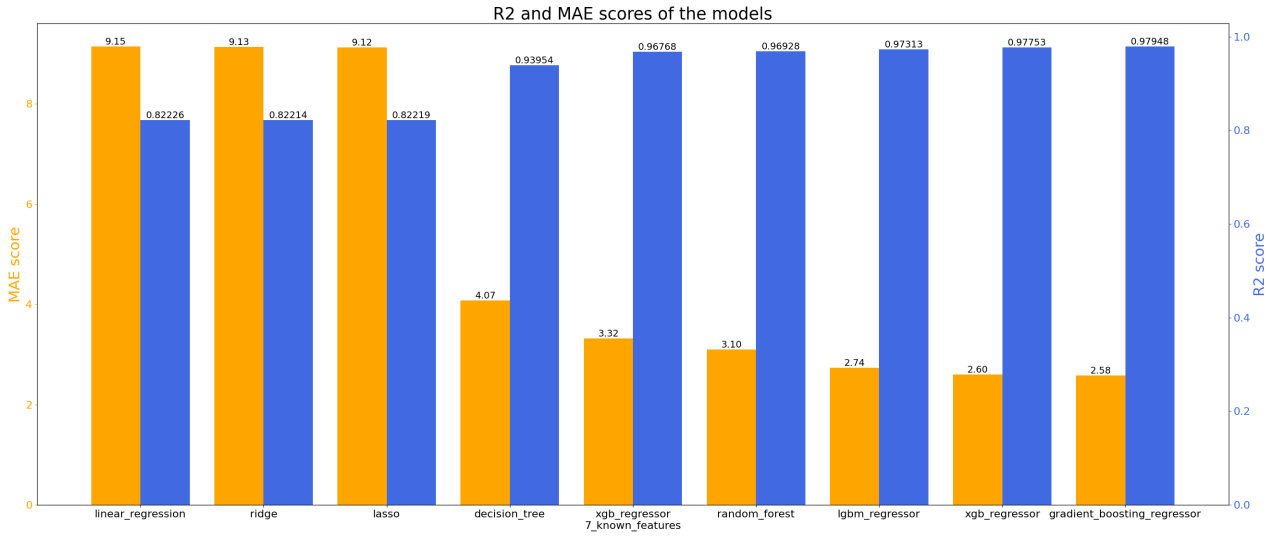


Рис. 4.15. R2 та MAE навчених моделей

Результати, представлені в цьому розділі, були апробовані на Всеукраїнській науково-технічній конференції молодих вчених, аспірантів та студентів "Стан, досягнення та перспективи інформаційних систем і технологій — 2024"[5].

РОЗДІЛ 5

СТВОРЕННЯ ДОДАТКА ДЛЯ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ ВСТУПНОЇ КАМПАНІЇ ДО ЗВО НА ОСНОВІ МОДЕЛІ XGBREGRESSOR

Для практичного використання результатів проведеного дослідження було створено спеціальний додаток (Dashboard [14]). Цей додаток дозволяє здійснювати прогнози на основі моделі XGBRegressor, використовуючи ключові параметри. Структура веб-додатка включає наступні компоненти:

- Поля вводу: Додаток має набір полів вводу, які дозволяють користувачам ввести необхідні дані для прогнозування результатів вступної кампанії.
- Випадаючі списки: Для зручності вибору, користувач може обирати університет та спеціальність з випадаючих списків, які містять доступні варіанти.
- Кнопка “Спрогнозувати”: Після введення необхідних даних користувачем, за допомогою цієї кнопки ініціюється процес прогнозування результатів вступної кампанії на основі моделі XGBRegressor.
- Виведення результатів прогнозування: Результати прогнозування відображаються користувачу у вигляді кількості бюджетних місць, які ймовірно будуть виділені на поточний рік.
- Можливість перегляду статистики за минулі роки: Додаток надає можливість користувачам переглядати статистику за минулі роки для обраного університету та спеціальності. Це дозволяє аналізувати тенденції та динаміку змін кількості бюджетних місць, поданих заяв та інших параметрів протягом років.

Зовнішній вигляд дашборду та приклад прогнозування результатів вступної кампанії можна побачити в додатку А.

5.1 Процес прогнозування результатів

Процес прогнозування результатів вступної кампанії базується на введених користувачем даних, таких як кількість поданих заяв, максимальний обсяг державного замовлення тощо. Ці дані обробляються за допомогою попередньо натренованої моделі XGBRegressor, яка була навчена на історичних даних про вступні кампанії. Після обробки введених даних, модель робить прогноз щодо кількості бюджетних місць, які можуть бути виділені на поточний рік.

5.2 Перегляд статистики за минулі роки

Додаток надає користувачам можливість переглядати статистику за минулі роки для обраного університету та спеціальності. Для цього використовуються випадаючі списки для вибору даних, а також графічні візуалізації, які показують динаміку зміни кількості бюджетних місць, поданих заяв та інших параметрів протягом років.

Загалом, цей веб-додаток забезпечує користувачам зручний інтерфейс для прогнозування результатів вступної кампанії та аналізу статистики за минулі роки, що допомагає приймати обгрунтовані рішення щодо вступу до закладів вищої освіти.

ВИСНОВКИ

В рамках роботи було проведено дослідження системи «широкого конкурсу» у вступній кампанії до закладів вищої освіти в Україні з метою аналізу та прогнозування розподілу бюджетних місць для конкретних спеціальностей в університетах.

Для досягнення цієї мети були використані методи машинного навчання та статистичного аналізу даних. Були зібрані дані результатів вступних кампаній з 2018-го по 2023-ий роки та проведено дослідження, щоб з'ясувати, які фактори найбільше впливають на кількість виділених бюджетних місць для кожної конкурсної пропозиції.

Був проведений статистичний та кореляційний аналіз, за допомогою яких були виявлені найбільш впливові фактори для поставленої задачі.

Далі, було побудовано та налаштовано різні моделі машинного навчання, такі як лінійні моделі, дерева рішень, випадковий ліс та градієнтний бустінг. Ці моделі були використані для прогнозування кількості бюджетних місць для різних спеціальностей у вищих навчальних закладах.

Було здійснено зменшення розмірностей у моделях за допомогою методу головних компонент та відбору найбільш впливових змінних.

Основними результатами роботи стали глибше розуміння системи «широкого конкурсу», розробка та порівняння математичних моделей машинного навчання для прогнозування кількості бюджетних місць, а найголовніше — створення практичного додатка для прогнозування результатів вступної кампанії до закладів вищої освіти.

Завдяки цьому додатку абітурієнти зможуть отримати більш точне уявлення про можливий розподіл бюджетних місць та прийняти відповідне рішення щодо подачі своїх заяв на обрані конкурсні пропозиції.

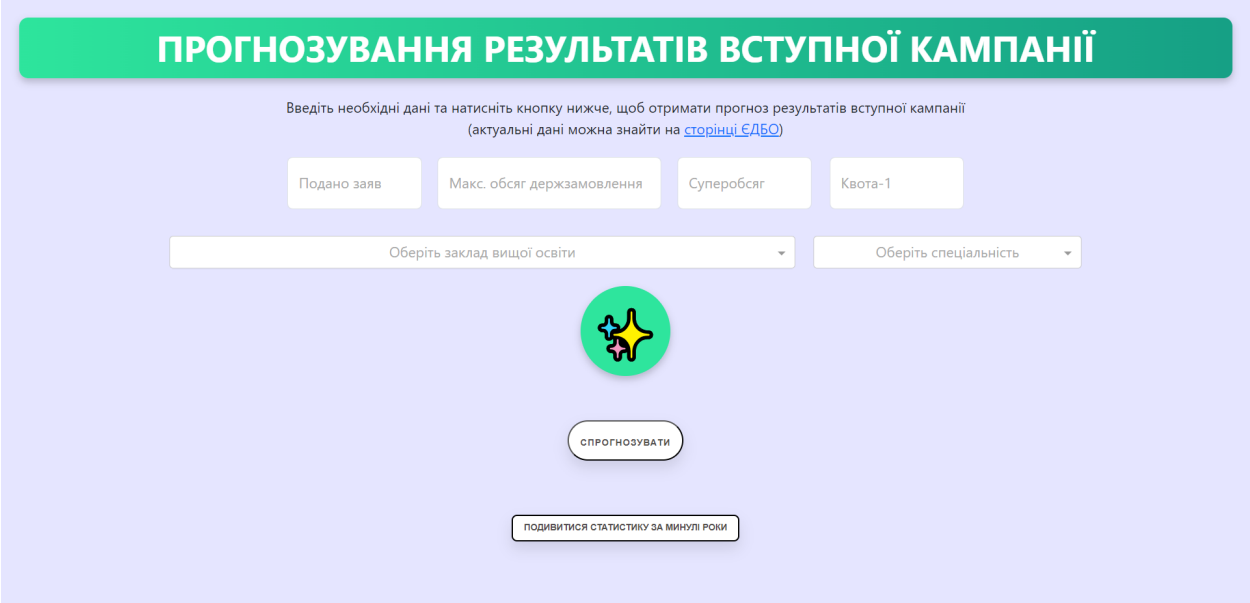
Результати, викладені в цій роботі, були представлені на міжнародній науково-практичній конференції «Інформаційні технології і автоматизація – 2023» [4] та Всеукраїнській науково-технічній конференції молодих вчених, аспірантів та студентів «Стан, досягнення та перспективи інформаційних систем і технологій — 2024» [5] і опубліковані у вигляді тез.

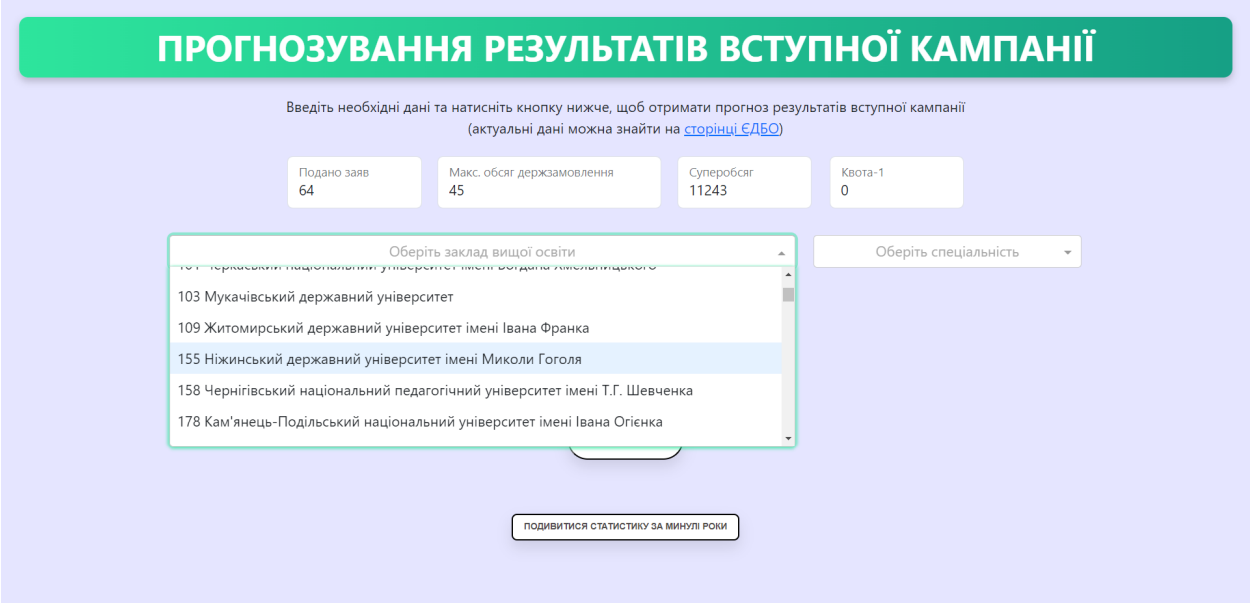
СПИСОК ЛІТЕРАТУРИ

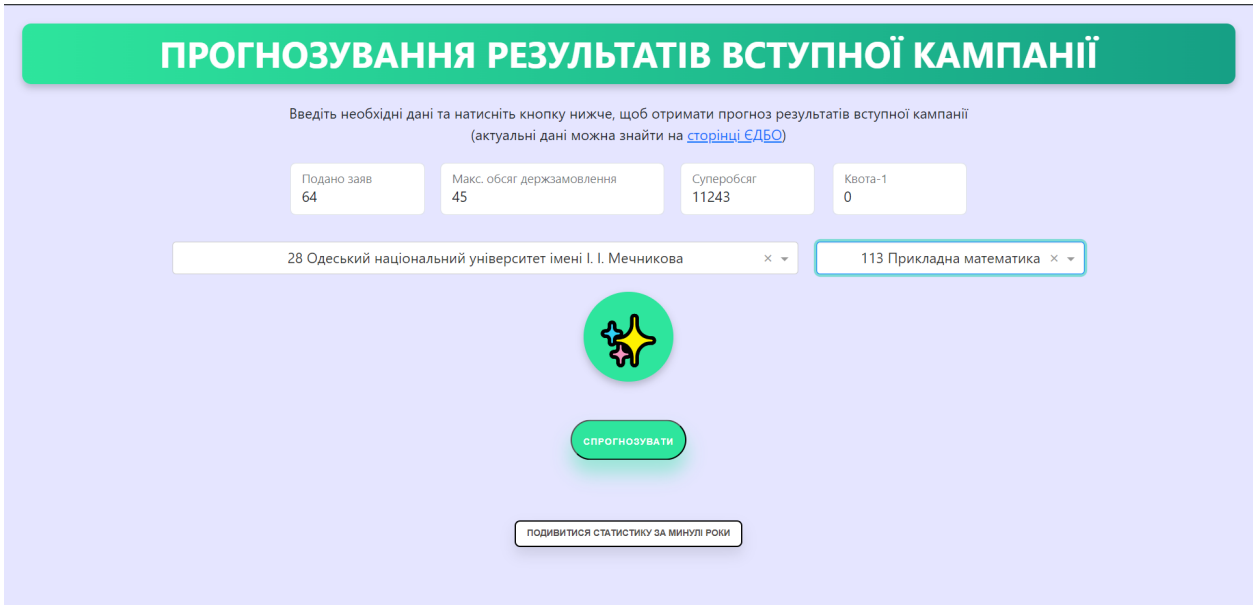
1. Додаток 6 до Умов прийому на навчання для здобуття вищої освіти в 2021 році (пункт 5 розділу IX), частина III. Визначення рекомендованих до зарахування за конкурсами.
2. Державне підприємство «Інфоресурс» Вступна кампанія: показники адресного розміщення державного замовлення за спеціальностями (спеціалізаціями) і закладами освіти: https://vstup.edbo.gov.ua/statistics/konkurs-universities/
3. Вихідний код та усі дані: https://github.com/Natanius18/diploma
4. Страхов Є.М., Чачко Н.Л. Статистичний аналіз впливу факторів на результати вступної кампанії до закладів вищої освіти // Інформаційні технології і автоматизація – 2023. Матеріали XVI міжнародної науково-практичної конференції. Одеса, 19-20 жовтня 2023 р. Одеса, Видавництво ОНТУ, 2023 р. С. 275-276.
5. Страхов Є.М., Чачко Н.Л. Прогнозування результатів вступної кампанії до закладів вищої освіти на основі моделей машинного навчання // Стан, досягнення та перспективи інформаційних систем і технологій — 2024. Матеріали XXIV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 18-19 квітня 2024 р. Одеса, Видавництво ОНТУ, 2024 р. С. 183-184.
6. Baiju Muthukadan Selenium with Python — Creative Commons Attribution-ShareAlike 4.0 International License, 2011-2018 selenium-python.readthedocs.io
7. Pandas documentation: pandas.pydata.org/docs
8. NumPy documentation: numpy.org/doc/stable
9. SciPy documentation: docs.scipy.org/doc/scipy/reference
10. Scikit-learn documentation: https://scikit-learn.org/stable/index.html
11. LightGBM documentation: https://lightgbm.readthedocs.io/en/stable/
12. Seaborn documentation: seaborn.pydata.org/index
13. Matplotlib documentation: matplotlib.org/stable/contents
14. Plotly documentation: plotly.com/

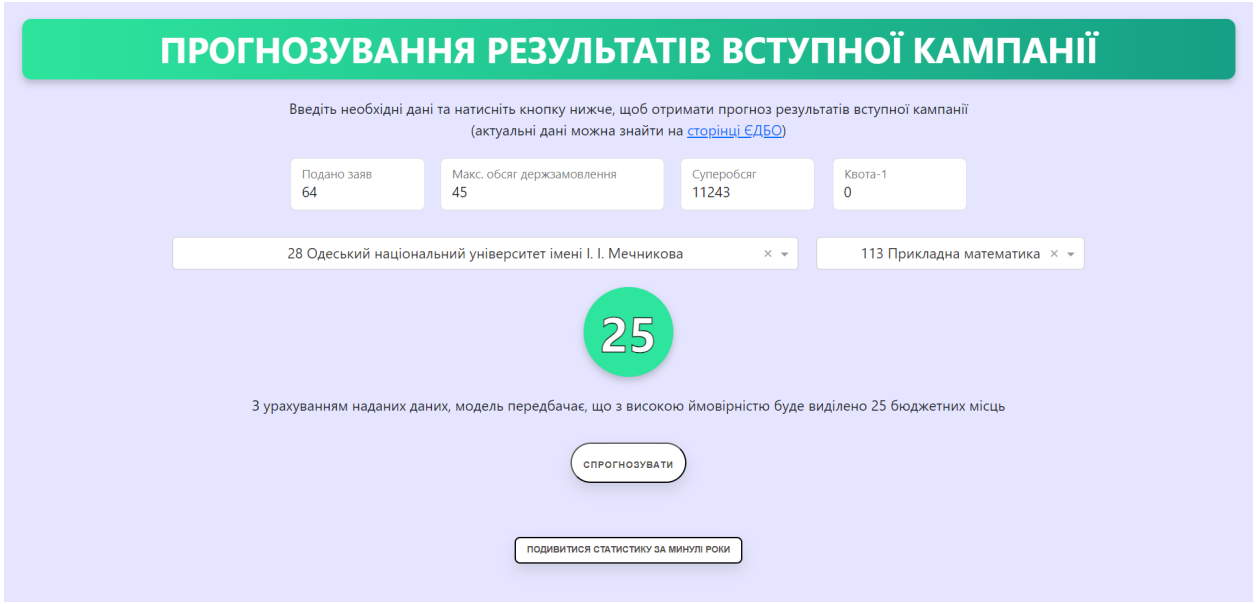
ДОДАТОК А

Зовнішній вигляд веб-додатка для прогнозування

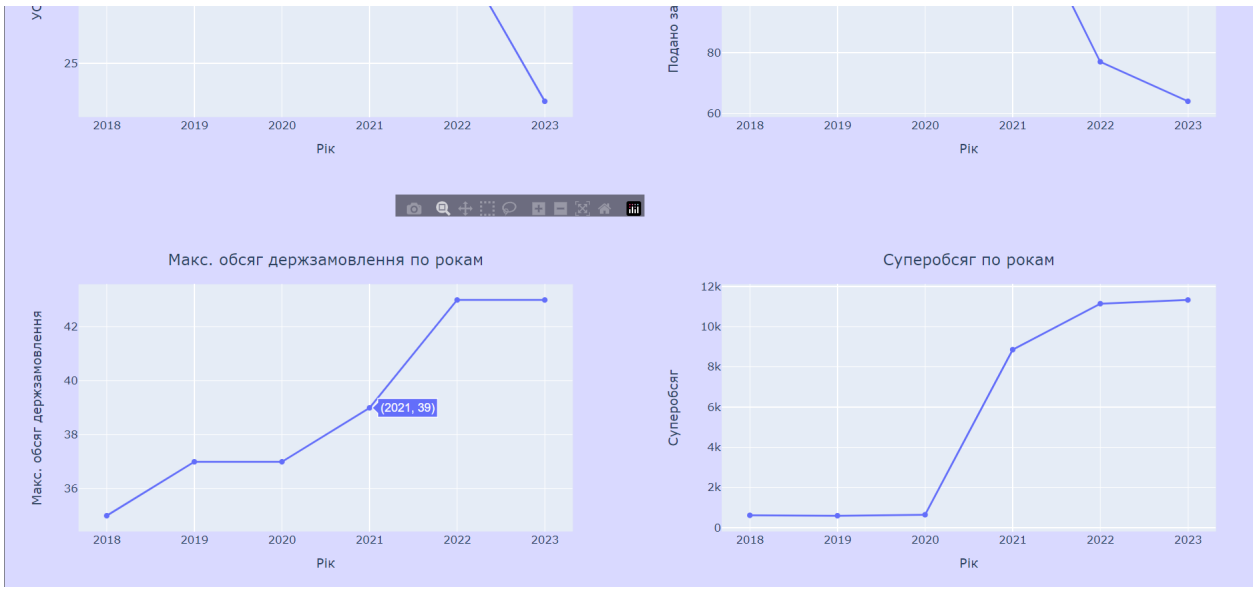





ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ ВСТУПНОЇ КАМПАНІЇ
Введіть необхідні дані та натисніть кнопку нижче, щоб отримати прогноз результатів вступної кампанії
(актуальні дані можна знайти на сторінці ЄДБО)
Подано заяв
64
Макс. обсяг держзамовлення
45
Суперобсяг
11243
Квота-1
0
28 Одеський національний університет імені І. І. Мечникова
113 Прикладна математика
СПРОГНОЗУВАТИ
ПОДИВИТИСЯ СТАТИСТИКУ ЗА МИНУЛІ РОКИ


ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ ВСТУПНОЇ КАМПАНІЇ
Введіть необхідні дані та натисніть кнопку нижче, щоб отримати прогноз результатів вступної кампанії
(актуальні дані можна знайти на сторінці ЄДБО)
Подано заяв
64
Макс. обсяг держзамовлення
45
Суперобсяг
11243
Квота-1
0
28 Одеський національний університет імені І. І. Мечникова
113 Прикладна математика
25
З урахуванням наданих даних, модель передбачає, що з високою ймовірністю буде виділено 25 бюджетних місць
СПРОГНОЗУВАТИ
ПОДИВИТИСЯ СТАТИСТИКУ ЗА МИНУЛІ РОКИ


ПРИХОВАТИ СТАТИСТИКУ
Оберіть назву закладу та спеціальність, для яких бажаєте отримати статистику:
28 Одеський національний університет імені І. І. Мечникова
113 Прикладна математика
Підсумкова кількість бюджетних місць по рокам
усього
Рік
Подано заяв на бюджет по рокам
Подано заяв на бюджет
Рік


25
2018
2019
2020
2021
2022
2023
Рік
Подано за
80
60
2018
2019
2020
2021
2022
2023
Рік

Макс. обсяг держзамовлення по рокам
Макс. обсяг держзамовлення
42
40
38
36
(2021, 39)
2018
2019
2020
2021
2022
2023
Рік
Суперобсяг по рокам
Суперобсяг
12k
10k
8k
6k
4k
2k
0
2018
2019
2020
2021
2022
2023
Рік