ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Алгоритм прогнозування успішності і термінів виконання ІТ проєктів на основі застосування машинного навчання»

на здобуття освітнього ступеня магістра
зі спеціальності 122 Комп'ютерні науки
освітньо-професійної програми «Комп'ютерні науки»

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Сундучков Дмитро
(підпис)

Виконав: здобувач вищої освіти групи КНДМ-63
Дмитро Сундучков

Керівник: Петро Кравчук
доктор філософії, доцент кафедри Комп'ютерних наук

Рецензент:
науковий ступінь, вчене звання *Ім'я, ПРІЗВИЩЕ*

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Комп’ютерних наук

Ступінь вищої освіти Магістр

Спеціальність 122 Комп’ютерні науки

Освітньо-професійна програма «Комп’ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп’ютерних наук
_____ Віктор Вишнівський
«_____» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Сундучкову Дмитру Михайловичу

1. Тема кваліфікаційної роботи: «Алгоритм прогнозування успішності і термінів виконання ІТ проєктів на основі застосування машинного навчання»
керівник кваліфікаційної роботи Кравчук Петро, доктор філософії, доцент кафедри Комп'ютерних наук
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «15» жовтня 2024 р. № 320.
2. Строк подання кваліфікаційної роботи «17» грудня 2024 р.
3. Вихідні дані до кваліфікаційної роботи: сучасні технології керування ІТ проєктами, методи та алгоритми машинного навчання, документація та науково-технічна література
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   1.ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ПРОГНОЗУВАННЯ ТЕРМІНІВ ВИКОНАННЯ ІТ-ПРОЄКТІВ
   2. ВИБІР МЕТОДІВ ТА ПЛАТФОРМИ ДЛЯ ПРОГНОЗУВАННЯ ТЕРМІНІВ ВИКОНАННЯ ІТ ПРОЄКТІВ ТА ПІДГОТОВКА ДАТАСЕТІВ
   3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАДАЧІ ПРОГНОЗУВАННЯ СТРОКІВ ВИКОНАННЯ ІТ ПРОЄКТІВ З ВИКОРИСТАННЯМ РІЗНИХ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ
5. Перелік ілюстративного матеріалу:

1. Порівняльна таблиця класичних і сучасних методів прогнозування ІТ проєктів
2. Порівняльна таблиця методів (моделей) машинного навчання
3. Структура датасету
4. Структура моделі типу дерево рішень
5. Структура моделі типу ліс рішень
6. Структура моделі типу підсилені дерева
7. Таблиця аналізу результатів
8. Висновки роботи
9. Опробація результатів тези на конференцію

6. Дата видачі завдання «16» жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	16.10-23.10.2024	Виконано
3	Дослідження методів машинного навчання в прогнозуванні термінів виконання ІТ-проєктів	26.10- 01.11.2024	Виконано
4	Дослідження технологій для прогнозування прогнозування термінів виконання ІТ-проєктів з використанням машинного навчання	02.11- 04.11.2024	Виконано
5	Практична реалізація задачі прогнозування строків виконання ІТ проєктів з використанням різних моделей машинного навчання	05.11- 08.11.2024	Виконано
6	Проведення аналізу результатів прогнозування строків виконання ІТ проєктів з використанням різних моделей машинного навчання	09.11- 10.11.2024	Виконано
7	Оформлення роботи: вступ, висновки, реферат	15.11.2024	Виконано
9	Попередній захист роботи	29.11.2024	Виконано

Здобувач вищої освіти _____ Дмитро Сундучков
(підпис)

Керівник
кваліфікаційної роботи _____ Петро Кравчук
(підпис)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 89 стор., 3 табл., 24 рис., 24 джерел.

Мета роботи – прогнозування термінів виконання ІТ-проєктів за допомогою моделей машинного навчання для підвищення точності оцінки та оптимізації ресурсів.

Об'єкт дослідження – процес планування та виконання ІТ-проєктів, що включає етапи оцінки часу на розробку та виконання завдань.

Предмет дослідження – застосування методів машинного навчання для прогнозування термінів виконання задач у сфері ІТ-проєктів, зокрема використання дерев рішень, лісів рішень та підсилених дерев (Boosted Trees).

У роботі застосовано різноманітні методи, такі як аналіз літературних джерел, порівняльний аналіз існуючих методів прогнозування, побудова та оцінка моделей машинного навчання, що дозволяє комплексно оцінити ефективність застосування машинного навчання для прогнозування термінів виконання задач у ІТ-проєктах.

Проведено аналіз сучасних методів прогнозування термінів виконання задач в рамках ІТ-проєктів, а також розглянуто можливості використання методів машинного навчання для цього. Це дозволило виявити основні проблеми традиційних підходів і визначити напрямки для підвищення точності прогнозів.

Розроблено та оптимізовано моделі машинного навчання для прогнозування термінів виконання задач у ІТ-проєктах, що включають різні типи моделей: дерева рішень, Random Forest та Boosted Trees. Кожна модель була оцінена за допомогою статистичних показників, таких як Mean Absolute Error, Mean Squared Error та R Squared, що дозволяє здійснити порівняльний аналіз їх ефективності.

Проведено експериментальне дослідження, яке включає порівняння результатів побудованих моделей, оцінку їх точності та можливості інтеграції у робочі процеси управління ІТ-проєктами. Також визначено потенційні покращення

моделей для досягнення ще більшої точності прогнозів та оптимізації процесів управління проєктами.

КЛЮЧОВІ СЛОВА: ПРОГНОЗУВАННЯ ТЕРМІНІВ, ІТ-ПРОЄКТИ, МОДЕЛІ МАШИННОГО НАВЧАННЯ, ДЕРЕВА РІШЕНЬ, ЛІСИ РІШЕНЬ, ПІДСИЛЕНІ ДЕРЕВА, ПІДВИЩЕННЯ ТОЧНОСТІ ПРОГНОЗІВ

ABSTRACT

Text part of the master's qualification work: 89 pages, 3 table, 24 pictures, 24 sources.

Objective of the work – predicting the deadlines of IT projects using machine learning models to improve the accuracy of estimates and optimize resources.

Object of the research – the process of planning and executing IT projects, including the stages of estimating time for development and task execution.

Subject of the research – the application of machine learning methods for predicting task completion times in IT projects, specifically the use of decision trees, random forests, and boosted trees.

Various methods were used in the work, including literature analysis, comparative analysis of existing forecasting methods, building and evaluating machine learning models, which allows for a comprehensive assessment of the effectiveness of applying machine learning to predicting task durations in IT projects.

The study analyzes modern methods of forecasting task completion times within IT projects and explores the potential of using machine learning techniques for this purpose. This allowed us to identify the main problems with traditional approaches and define directions for improving forecast accuracy.

Machine learning models for predicting task completion times in IT projects have been developed and optimized, including various types of models: decision trees, random forests, and boosted trees. Each model was evaluated using statistical metrics such as Mean Absolute Error, Mean Squared Error, and R Squared, enabling a comparative analysis of their effectiveness.

An experimental study was conducted, including a comparison of the results of the built models, assessing their accuracy and potential for integration into IT project management processes. Additionally, potential improvements to the models were identified to achieve even greater forecasting accuracy and optimize project management processes.

KEYWORDS: PREDICTING DEADLINES, IT PROJECTS, MACHINE LEARNING MODELS, DECISION TREES, RANDOM FORESTS, BOOSTED TREES, IMPROVING FORECAST ACCURACY

ЗМІСТ

ВСТУП

У сучасному світі інформаційні технології є рушійною силою розвитку бізнесу, медицини, освіти та інших сфер суспільного життя. Впровадження IT-рішень дозволяє компаніям оптимізувати процеси, покращувати комунікації та швидше адаптуватися до змін на ринку. Однак управління IT-проєктами залишається складним завданням, особливо коли йдеться про точність планування. У дослідженні 2023 року, проведеному аналітичною компанією Standish Group, виявлено, що лише 29% IT-проєктів завершуються вчасно та в межах бюджету[1].

Основними причинами цього є наступні фактори.

Недостатня точність оцінок: на етапі планування часто бракує даних або досвіду для коректного визначення обсягу задач.

Зміни в проєкті: через динамічний характер IT-сфери вимоги можуть змінюватися, додаючи складності до планування.

Людський фактор: переоцінка можливостей, недостатній рівень комунікації в команді, а також залежність від ключових працівників.

Затримки у виконанні IT-проєктів можуть призвести до значних фінансових втрат, зриву контрактів та втрати довіри з боку клієнтів. Наприклад, у 2021 році збитки компаній від затримок проєктів у сфері електронної комерції перевищили 2,5 мільярда доларів[2].

Традиційні методи, такі як PERT (Program Evaluation and Review Technique) або CPM (Critical Path Method), використовуються для прогнозування термінів виконання задач і управління проєктами. Вони базуються на детермінованих підходах і передбачають ручний аналіз залежностей між задачами. Хоча ці підходи широко застосовуються, вони мають обмеження, серед яких:

Неможливість врахування великої кількості параметрів, таких як складність задачі, тип технології або досвід розробників.

Відсутність адаптивності до змін: традиційні підходи не встигають враховувати динамічні зміни у проєктах.

Ризик суб'єктивності: залежність від людського фактору може вплинути на точність оцінок.

Машинне навчання (ML) відкриває нові можливості для вирішення зазначених проблем. Його алгоритми здатні працювати з великими наборами даних, враховувати численні взаємозв'язки між параметрами та адаптуватися до змін у проєкті. У сфері управління IT-проєктами машинне навчання допомагає:

- автоматизувати процес оцінки складності задач;
- прогнозувати терміни виконання задач із високою точністю;
- виявляти потенційні ризики на ранніх етапах;
- оптимізувати розподіл ресурсів у команді.

Метою цієї роботи є розробка та дослідження алгоритмів машинного навчання для прогнозування термінів виконання задач і оцінки їх успішності. Основна увага приділяється використанню реальних даних для побудови моделей, здатних прогнозувати майбутні результати.

Завдання дослідження:

- Аналіз існуючих підходів до прогнозування термінів виконання задач.
- Підготовка та опис реальних даних для дослідження.
- Побудова та оцінка моделей машинного навчання (дерева рішень, Random Forest, Boosted Trees).
- Проведення порівняння моделей за точністю та ефективністю.
- Розробка рекомендацій для інтеграції моделей у робочі процеси IT-команди.

1 ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ПРОГНОЗУВАННЯ ТЕРМІНІВ ВИКОНАННЯ ІТ-ПРОЄКТІВ

1.1 Проблематика прогнозування термінів виконання ІТ-проєктів

Управління ІТ-проєктами є надзвичайно складним завданням, яке вимагає врахування багатьох факторів: технічних, економічних, організаційних та людських. Поширеною проблемою є неточність у прогнозуванні термінів виконання задач, що часто призводить до значних затримок у завершенні проєктів. Висока частка невдач у виконанні ІТ-проєктів є загальновідомим фактом, і на це є кілька ключових причин, які визначають сучасний стан цієї проблеми.

Основною проблемою є недостатня точність попередніх оцінок, оскільки традиційні методи управління проєктами часто ґрунтуються на суб’єктивних судженнях замість об’єктивних даних. Багато ІТ-компаній стикаються з ситуацією, коли планування базується на людському досвіді, який не завжди є достатньо точним для передбачення складності задач, що виконуються. Наприклад, часто недооцінюється час, необхідний для тестування та інтеграції нових технологій, що створює значні затримки. Крім того, існують складнощі з передбаченням складності завдань через величезну кількість непередбачених змін, що можуть виникнути під час реалізації проєкту, таких як зміна вимог замовника чи оновлення технологій.

Сучасні ІТ-проєкти відрізняються високим рівнем динамічності. У зв'язку з розвитком технологій і змінами на ринку, вимоги до проєктів можуть змінюватися навіть після початку виконання. Це означає, що традиційні методи планування не завжди здатні відобразити ці зміни, що призводить до значних відхилень у розрахунках. Зміни в проєкті можуть включати додавання нових функціональностей, зміну архітектури або навіть зміни в команді розробників, що створює додаткову складність у прогнозуванні термінів виконання.

Ще однією важливою причиною затримок є людський фактор. Складність задач в IT-проєктах часто недооцінюється через суб'єктивні оцінки чи оптимістичні припущення щодо ресурсів. Помилки у комунікації між учасниками проєкту, невизначеність у вимогах до кінцевого продукту, а також відсутність досвіду у вирішенні подібних завдань можуть призвести до того, що команди перевищують очікувані терміни.

Сучасні підходи до прогнозування термінів виконання в IT-сфері можуть бути розподілені на дві основні групи: традиційні методи, що ґрунтуються на точних математичних моделях, та моделі на основі даних. Перші використовують алгоритми для планування, що дозволяють визначити мінімальний час виконання задач, та передбачають певні залежності між етапами роботи. Однак ці моделі мають ряд обмежень, коли йдеться про складні проєкти з великою кількістю змінних і залежностей.

Моделі на основі даних, зокрема методи машинного навчання, здатні враховувати не тільки математичні залежності, а й більш складні патерни в даних, що дозволяє зробити прогнози більш точними і адаптованими до змін. Застосування алгоритмів машинного навчання в прогнозуванні IT-проєктів надає можливість знизити рівень невизначеності і підвищити точність прогнозів. Однак попри це, існує низка проблем, що вимагають вирішення, наприклад, недостатня кількість історичних даних для точних прогнозів або недостатня перевірка моделей на реальних кейсах.

Існуючи прогнози щодо розвитку технологій прогнозування для IT-сфери свідчать про те, що в майбутньому застосування машинного навчання та аналітики великих даних дозволить значно зменшити вплив людського фактора і змінювати підходи до планування проєктів. Однак для цього потрібно розв'язати проблеми з підготовкою даних, масштабуванням моделей та інтеграцією результатів у реальні робочі процеси.

1.2 Дослідження методів прогнозування що ґрунтуються на точних математичних моделях

Управління термінами виконання IT-проєктів завжди було важливою частиною проєктного менеджменту. Традиційні методи прогнозування, такі як PERT (Program Evaluation and Review Technique) і CPM (Critical Path Method), були розроблені в середині XX століття і залишаються популярними до сьогодні для планування й оцінки термінів виконання задач. Однак вони мають низку обмежень, що обумовлює потребу у пошуку більш ефективних методів прогнозування, зокрема з використанням машинного навчання.

Метод PERT був розроблений для планування й управління великими проектами, що включають багато невизначених і змінних факторів. Основним завданням PERT є оцінка тривалості виконання задач за допомогою трьох різних оцінок для кожної активності:

Оптимістичне значення: мінімальний час, необхідний для виконання задачі за найсприятливіших умов.

Песимістичне значення: максимальний час виконання, якщо всі умови є несприятливими.

Ймовірне значення: середнє значення часу, яке, ймовірно, буде витрачено на виконання задачі.

Цей метод дозволяє отримати більш точне уявлення про час виконання задач у порівнянні з традиційними методами, що використовують лише одну оцінку. Але він також має обмеження:

Не враховує змінні фактори: PERT не дозволяє врахувати більшість змін, які можуть виникнути під час виконання проєкту, такі як зміна вимог або непередбачені проблеми в роботі команди.

Невизначеність оцінок: Оцінки часу можуть бути сильно залежні від досвіду керівника проєкту або членів команди, що може призвести до помилок в оцінках.

Не застосовується до всіх проєктів: Метод менш ефективний для малих або стандартних задач, де тривалість виконання легко прогнозується без складних оцінок.[3]

Метод CPM застосовується для планування проєктів, в яких усі задачі мають чітко визначені терміни виконання і взаємозалежності. Він дозволяє визначити критичний шлях — найдовший шлях через мережу задач, що визначає мінімальний час, необхідний для виконання всього проєкту. Основні етапи CPM включають створення мережі задач, оцінку часу виконання кожної задачі та ідентифікацію критичних задач, на яких залежить весь проєкт.

Основні переваги CPM:

Чітка структура: дозволяє чітко визначити, які задачі є критичними для успішного завершення проєкту.

Простота: метод простий у використанні для проєктів із чітко визначеними задачами та часом їх виконання.

Однак CPM має також значні обмеження:

Ігнорування варіацій: CPM не враховує можливі зміни тривалості задач. Якщо виконання задачі займає більше або менше часу, ніж було передбачено, це може вплинути на кінцевий результат.

Неврахування змін: метод не адаптується до динамічних змін у проєкті, таких як зміна вимог замовника або технічні труднощі.

Орієнтація на великий проєкт: CPM підходить для складних проєктів, що складаються з багатьох залежних задач, але для малих проєктів цей метод часто є занадто громіздким.[4]

Методи, побудовані на точних математичних моделях мають певні обмеження.

Як PERT, так і CPM мають свої переваги, але є значні обмеження в умовах сучасних IT-проєктів. Обидва методи потребують чіткої структури та детальних оцінок для кожної задачі, що ускладнює їх застосування до проєктів із високим рівнем невизначеності та змінами в процесі виконання. Крім того, ці методи не

враховують взаємозалежності та складні патерни, що можуть виникати в реальному житті, коли одні зміни впливають на інші. Наприклад, затримка в одному етапі може вплинути на інші задачі, і це важко врахувати за допомогою традиційних методів.[5]

Ці проблеми і обмеження стають ще більш очевидними, коли йдеться про проєкти, що включають нові технології або нестандартні завдання. Тому традиційні методи часто не дають змоги здійснювати точні прогнози, особливо коли мова йде про інноваційні IT-проєкти, які вимагають більш гнучких та адаптивних підходів.

1.3 Дослідження методів машинного навчання в прогнозуванні термінів виконання IT-проєктів

Машинне навчання (ML) стає все більш популярним інструментом для вирішення складних задач прогнозування в IT-проєктах. На відміну від традиційних підходів, які базуються на математичних моделях із жорсткими припущеннями, алгоритми машинного навчання здатні адаптуватися до різних типів даних, виявляти приховані залежності та враховувати множинні фактори, які можуть впливати на результат. Це дозволяє досягати більшої точності у прогнозах і знижувати ризик невдач у проєктах.

Машинне навчання базується на аналізі історичних даних, які відображають поведінку системи у минулому. Для IT-проєктів це може включати такі характеристики, як час виконання задач, їхня складність, розподіл ресурсів, а також тривалість кожного етапу проєкту, такого як аналіз, розробка, тестування та реліз. На основі цих даних моделі ML здатні передбачати ймовірність виконання задач у межах запланованого часу та визначати ключові фактори, які впливають на результат. Однією з ключових переваг ML є можливість працювати з великими обсягами даних і враховувати складні залежності між змінними. Наприклад,

модель може виявити, що тривалість задачі залежить не лише від її складності, але й від досвіду розробника або типу технології, що використовується.[6]

Машинне навчання має низку значущих переваг. Зокрема, воно здатне автоматично адаптуватися до нових умов. Наприклад, якщо у команді змінюються процеси роботи або з’являються нові технології, модель може бути перенавчена на оновлених даних, що забезпечує її актуальність. Алгоритми ML також ефективно працюють з неоднорідними даними, які характерні для IT-проєктів, наприклад, числовими значеннями часу виконання задач або категорійними даними про тип задачі чи використовувану технологію. Крім цього, моделі можуть ідентифікувати задачі, які мають високу ймовірність затримки, і давати рекомендації щодо їх оптимізації. Також ML є надзвичайно корисним для великих проєктів, які включають сотні чи тисячі задач, оскільки традиційні методи часто виявляються непридатними в умовах такої складності.[7]

Серед найпопулярніших алгоритмів ML, які використовуються у прогнозуванні, можна виокремити дерева рішень, ансамблеві методи та нейронні мережі. Дерева рішень є простими для розуміння та візуалізації, що робить їх ідеальними для аналізу структурованих даних. Ансамблеві методи, такі як Random Forest і Boosted Trees, підвищують точність прогнозів завдяки об'єднанню результатів кількох моделей. Random Forest добре працює з великими наборами даних, тоді як Boosted Trees часто дають кращі результати для невеликих вибірок, але потребують детальнішого налаштування. Нейронні мережі, хоча і є потужним інструментом, частіше використовуються для завдань з великою кількістю нелінійних залежностей, але їхня складність і висока потреба в обчислювальних ресурсах обмежують їхнє застосування в середніх і малих IT-командах. [8]

Втім, використання ML має свої обмеження. Однією з таких є залежність від якості даних. Для побудови точної моделі потрібні великі обсяги історичних даних, які є репрезентативними для майбутніх задач. У багатьох IT-компаніях відсутність стандартизації даних або обмежений доступ до минулої інформації створює значні труднощі для впровадження ML. Іншою проблемою є складність інтерпретації

моделей. Наприклад, дерева рішень є відносно зрозумілими, тоді як ансамблеві методи або нейронні мережі часто функціонують як "чорні ящики", результати яких важко пояснити навіть фахівцям. Машинне навчання також вимагає значних обчислювальних ресурсів, особливо для навчання моделей на великих наборах даних. Окрім цього, моделі потребують регулярного оновлення, адже для збереження їхньої актуальності необхідно навчати їх на нових даних, що створює додаткове навантаження на команду.

Попри ці обмеження, машинне навчання відкриває значні можливості для покращення управління IT-проєктами. Його використання дозволяє суттєво підвищити точність планування, ідентифікувати потенційні ризики та оптимізувати розподіл ресурсів, що робить його важливим інструментом для сучасних IT-команд.[9]

Результати дослідження методів зведені в таблицю 1.1

Таблиця 1.1. Порівняльна таблиця характеристик класичних і сучасних методів прогнозування:

Критерій	Класичні методи (PERT, CPM)	Сучасні методи (машинне навчання)
Точність прогнозів	Залежить від якості вхідних даних і суб'єктивних оцінок.	Висока точність за рахунок автоматичного виявлення закономірностей у даних.
Гнучкість	Складно адаптуються до змін у проектах чи даних.	Легко адаптуються до нових даних і параметрів.
Залежність від експерта	Потребують досвіду та інтуїції менеджера для побудови моделей.	Автоматизовані; експертна думка впливає лише на підготовку даних.
Масштабованість	Обмеженість при роботі з великими обсягами даних.	Легко працюють із великими даними завдяки сучасним обчислювальним ресурсам.

Продовження таблиці 1.1:

Критерій	Класичні методи (PERT, CPM)	Сучасні методи (машинне навчання)
Час на обробку даних	Потребують значного часу на підготовку та обчислення вручну.	Автоматизовані процеси знижують час підготовки й обробки даних.
Складність впровадження	Простота у використанні, оскільки не вимагають складного програмного забезпечення.	Складніші у впровадженні через потребу в налаштуванні моделей і систем.
Критерій	Класичні методи (PERT, CPM)	Сучасні методи (машинне навчання)
Облік невизначеності	Включають фактори ризику через коефіцієнти, але можуть бути суб'єктивними.	Враховують невизначеність через аналіз усіх доступних параметрів.
Інтеграція з іншими системами	Складно інтегрувати в сучасні інформаційні системи.	Легко інтегруються в сучасні системи управління проєктами.

1.4 Перспективи застосування машинного навчання в управління IT-проєктами

Машинне навчання стало важливим інструментом у сфері управління IT-проєктами, оскільки воно дозволяє значно покращити точність прогнозів і зменшити ризики, що виникають при плануванні та реалізації проєктів. У порівнянні з традиційними методами прогнозування, що часто базуються на суб'єктивних оцінках і обмежених даних, алгоритми машинного навчання здатні враховувати безліч змінних і складних взаємозалежностей, що робить

прогнозування більш точним і надійним. Використання таких інструментів дозволяє не тільки прогнозувати час виконання задач, а й автоматизувати багато процесів, що стосуються управління проєктами, наприклад, оцінка ризиків і розподіл ресурсів.

Однією з найбільших переваг машинного навчання є здатність адаптуватися до змінних умов. Наприклад, в умовах IT-проєктів, де вимоги можуть змінюватися в процесі роботи, ML-моделі можуть оперативно оновлювати свої прогнози за допомогою нових даних. Це дозволяє більш точно передбачати тривалість виконання задач і оптимізувати розподіл ресурсів, зважаючи на актуальні умови. Крім того, алгоритми машинного навчання здатні працювати з великими обсягами даних і враховувати не лише числові показники, а й складні категоріальні характеристики, що в свою чергу підвищує точність прогнозів.[10]

Машинне навчання також дозволяє значно підвищити точність оцінки ризиків. Наприклад, за допомогою алгоритмів можна виявляти задачі, які мають підвищену ймовірність затримки, навіть на ранніх етапах реалізації проєкту. Це дає можливість керівникам проєктів коригувати плани і перерозподіляти ресурси, запобігаючи великим затримкам у майбутньому. Такі прогнози допомагають своєчасно виявляти проблемні зони і приймати необхідні заходи для покращення ефективності роботи.

Окрім цього, машинне навчання може автоматизувати процес оцінки складності задач. Традиційні методи оцінки часто базуються на досвіді експертів, що може призводити до суб'єктивних і помилкових оцінок. Алгоритми ML дозволяють значно зменшити цей вплив людського фактора. Наприклад, на основі історичних даних про схожі задачі, моделі можуть автоматично оцінювати час, необхідний для виконання нової задачі, з високою точністю. Це не тільки знижує навантаження на команду, а й робить процес планування більш об'єктивним.[11]

Використання машинного навчання у прогнозуванні IT-проєктів також дозволяє оптимізувати розподіл ресурсів. Завдяки високій точності прогнозів, можна ефективніше розподіляти завдання серед членів команди, мінімізуючи

перерви у роботі та оптимізуючи використання доступних ресурсів. Моделі машинного навчання можуть також допомогти в автоматичному формуванні графіків роботи, враховуючи як складність задач, так і доступність ресурсів у реальному часі.

Застосування машинного навчання в управлінні IT-проєктами має свої виклики. Одним із основних є необхідність високоякісних даних. Моделі, побудовані на неповних або неактуальних даних, можуть призвести до помилкових прогнозів, що в свою чергу призведе до неправильної оцінки часу виконання задач або виявлення ризиків. Для того щоб ML-моделі приносили користь, потрібно постійно оновлювати дані, що вимагає додаткових ресурсів.

Іншим важливим моментом є складність інтерпретації результатів. Деякі моделі, зокрема ансамблеві методи або нейронні мережі, можуть функціонувати як "чорні ящики", що означає, що їхні результати можуть бути важко інтерпретовані людьми, які не мають глибоких знань у машинному навчанні. Це може стати проблемою для керівників проєктів, яким необхідно пояснити результати своєї роботи іншим учасникам команди або клієнтам.

Також важливо зазначити, що для розгортання ефективної системи машинного навчання в реальних умовах необхідно забезпечити належні обчислювальні ресурси. Навчання складних моделей на великих наборах даних вимагає значної потужності процесорів і графічних карт, що може бути дорогим і потребувати спеціалізованого обладнання.

Попри всі ці виклики, перспективи застосування машинного навчання в управлінні IT-проєктами є дуже обнадійливими. Поширення технологій великого даних, автоматизації та хмарних обчислень відкривають нові можливості для впровадження машинного навчання в реальні бізнес-процеси. У майбутньому це дозволить не тільки підвищити точність прогнозів, але й значно знизити затримки у виконанні задач, зробивши проєкти більш гнучкими і менш залежними від людських помилок.[12]

2 ВИБІР МЕТОДІВ ТА ПЛАТФОРМИ ДЛЯ ПРОГНОЗУВАННЯ ТЕРМІНІВ ВИКОНАННЯ ІТ ПРОЄКТІВ ТА ПІДГОТОВКА ДАТАСЕТІВ

2.1 Дослідження методів машинного навчання в задачах прогнозування термінів виконання IT-проєктів

У цьому розділі описано, як були побудовані і навчалися моделі машинного навчання для прогнозування тривалості виконання IT-проєктів на основі вхідних параметрів. Враховуючи характер даних, були обрані методи, що дозволяють ефективно обробляти числові та категоріальні змінні, а також забезпечують хорошу точність прогнозів.

Дерево рішень було обрано першим методом для аналізу даних у цьому дослідженні. Це один із найпростіших і водночас ефективних алгоритмів машинного навчання, що дозволяє виявляти структури та закономірності у даних. Дерево працює шляхом послідовного поділу даних на основі значень вхідних параметрів, де кожен вузол дерева відповідає за перевірку певного параметра, а листові вузли — за прийняття кінцевого рішення. Базова структура дерева рішень приведена на рис. 2.1.

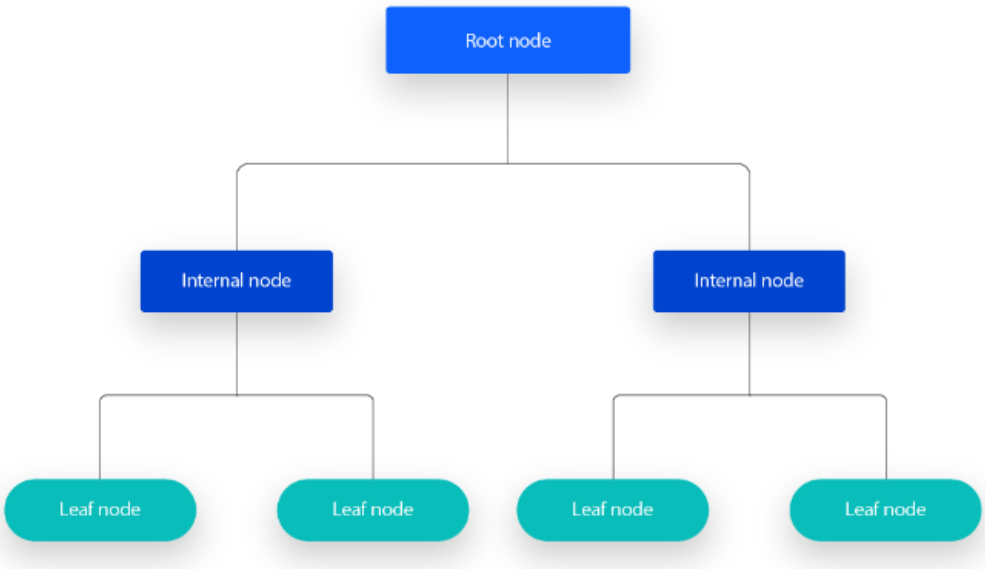


Рис. 2.1 Базова структура дерева рішень

Принцип роботи дерева рішень полягає у зменшенні невизначеності, яка оцінюється за допомогою ентропії. Ентропія, обчислювана за формулою:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \cdot \log_2 P(x_i) \qquad (2.1)$$

дає змогу оцінити рівень "чистоти" вузла. Найкращий параметр для поділу вибирається на основі інформаційного приросту:

$$IG(T, A) = H(T) - \sum_{v \in V} \frac{|T_v|}{|T|} \cdot H(T_v) \qquad (2.2)$$

де T — набір даних у вузлі, A — параметр для поділу, а V — можливі значення цього параметра.[13]

У нашому дослідженні дерево рішень було використано для побудови моделі, яка пояснює, як час виконання кожного етапу задач залежить від інших параметрів, таких як оцінка складності задачі або естімейшн. Основною перевагою методу є його інтуїтивна зрозумілість: модель легко інтерпретується та візуалізується, що є важливим для аналізу IT-проєктів.

Для уникнення переобучення було використано обмеження на глибину дерева та мінімальну кількість вузлів на кожному рівні. Це дозволило зменшити ймовірність перенавчання, коли модель надто точно підлаштовується під навчальні дані, втрачаючи здатність узагальнювати нові випадки. Наприклад, вага кожного вузла розраховувалася за формулою:

$$W(v) = \frac{|T_v|}{|T|} \qquad (2.3)$$

де $W(v)$— вага вузла v, що визначає його внесок у прогноз.

Дерева рішень було обрано для початку аналізу завдяки їхній простоті, адаптивності до різних типів даних і здатності швидко отримувати базові прогнози. Крім того, цей алгоритм є оптимальним для нашого дослідження, оскільки дозволяє легко оцінювати вплив окремих параметрів на результат, що є ключовим для аналізу задач IT-проєктів.

Дерево рішень показало себе як ефективний інструмент для дослідження залежності між параметрами задач та їх тривалістю. Це забезпечило міцну основу для подальшого застосування більш складних моделей, таких як ансамблеві методи.

Для побудови дерева рішень було використано алгоритм, що автоматично визначав найкращі параметри для побудови моделі, зокрема максимальну глибину дерева та кількість вузлів на кожному рівні. Це дозволяє моделі адаптуватися до даних і мінімізувати ймовірність переобучення, яке може виникати при занадто складних моделях.

Ліс рішень є ансамблевим методом машинного навчання, що базується на використанні кількох дерев рішень для підвищення точності прогнозів та стійкості до аномалій у даних. На відміну від окремого дерева рішень, Random Forest поєднує результати багатьох дерев, створюючи фінальний прогноз шляхом усереднення (для регресії) або голосування (для класифікації). Завдяки цьому ліс є потужним інструментом для задач із високою варіативністю даних.

Під час навчання Random Forest кожне дерево T_m будується на основі підмножини даних D_m, отриманої шляхом беггінгу — техніки випадкової вибірки з поверненням. Беггінг дозволяє зменшити кореляцію між деревами, оскільки кожне дерево отримує унікальний набір даних. Формально ця вибірка описується як $D_m \subseteq D$, де D — вихідний набір даних, а m=1,2,…,M, де M — кількість дерев у лісі.

Крім того, на кожному вузлі дерева для аналізу використовується лише випадкова підмножина параметрів $P_m \subseteq P$. Це ще більше знижує кореляцію між

деревами та сприяє підвищенню узагальнюючої здатності моделі. Всередині кожного дерева параметри для поділу обираються за формулами, аналогічними до дерев рішень. Ентропія вузла визначається як:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \cdot \log_2 P(x_i) \qquad (2.4)$$

де $P(x_i)$ — ймовірність класу хіхі у вузлі. Інформаційний приріст, що визначає оптимальний поділ, обчислюється за формулою:

$$IG(T, A) = H(T) - \sum_{v \in V} \frac{|T_v|}{|T|} \cdot H(T_v) \qquad (2.5)$$

де T — набір даних у вузлі, A — параметр для поділу, а V — можливі значення цього параметра, T_v — підмножина даних після поділу.

Головна особливість Random Forest полягає у створенні фінального прогнозу на основі результатів усіх дерев. Для регресії прогноз розраховується як середнє значення результатів дерев:

$$\hat{y} = \frac{1}{M} \sum_{m=1}^{M} T_m(x) \qquad (2.6)$$

де $T_m(x)$ — прогноз дерева T_m для вхідних даних x. Для класифікації використовується голосування, коли кожне дерево голосує за свій клас, і обирається клас із найбільшою кількістю голосів. [14]

Random Forest дозволяє ефективно вирішувати проблеми перенавчання, оскільки кожне дерево навчається на різних підмножинах даних, і фінальний прогноз є усередненим результатом усіх дерев. Такий підхід забезпечує високу

стійкість моделі до шуму у вхідних даних, підвищує точність прогнозів та узагальнюючу здатність. Загальний принцип дії лісу рішень приведено на рис. 2.2.

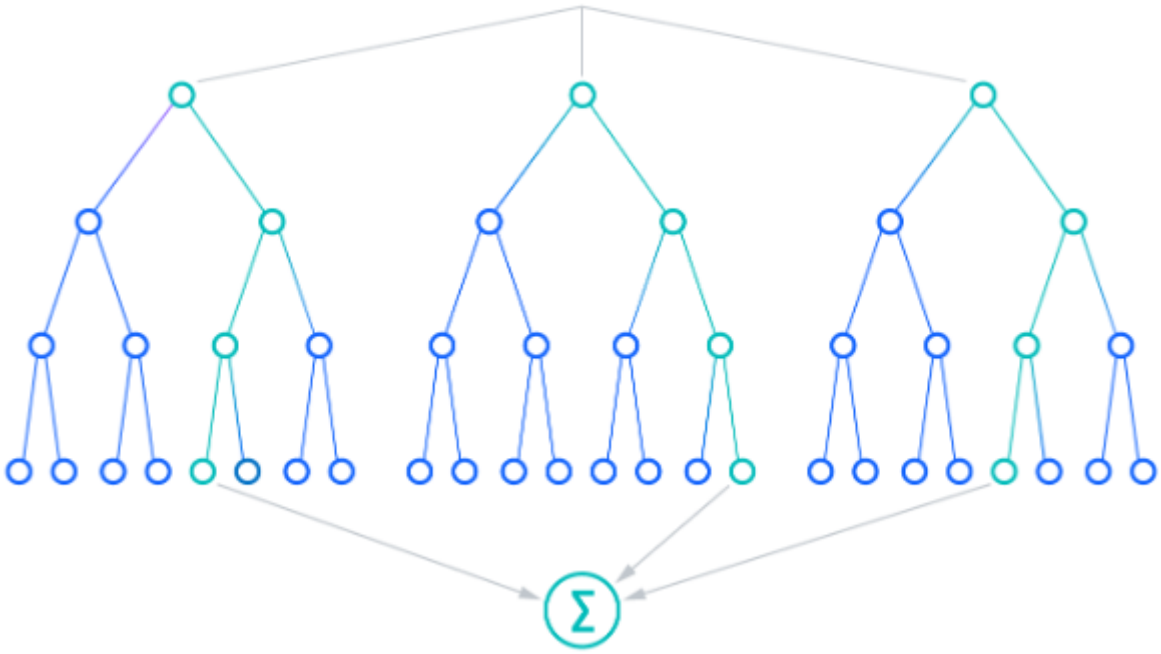


Рис. 2.2 Принцип дії лісу рішень

У нашому дослідженні Random Forest було використано для прогнозування тривалості задач IT-проєктів. Завдяки великій кількості параметрів і великим обсягам даних метод дозволив отримати точні прогнози та стабільні результати. Незважаючи на те, що всередині кожного дерева використовуються ті самі формули, що й для окремого дерева рішень, унікальні аспекти Random Forest, такі як беггінг і випадковий вибір параметрів, значно підвищують ефективність та адаптивність моделі.

Метод Boosted Trees, або підсилених дерев, є однією з найпотужніших технік ансамблевого навчання, яка дозволяє досягти високої точності за рахунок послідовного вдосконалення моделей. Основна ідея методу полягає в тому, що кожне нове дерево будується для корекції помилок попередніх. Цей підхід дозволяє приділяти більше уваги важким для прогнозування випадкам, що істотно покращує результати навіть на невеликих або зашумлених наборах даних. Загальний принцип дії методу посилених дерев приведено на рис. 2.3.

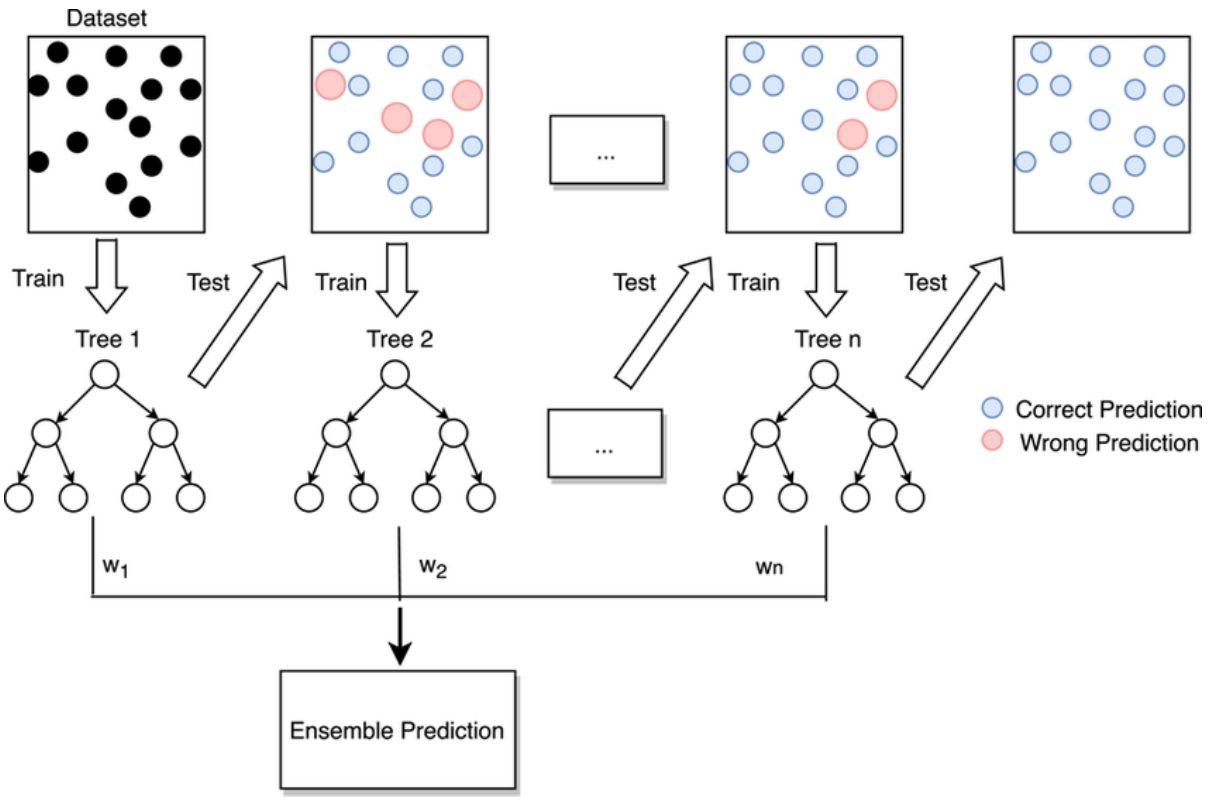


Рис. 2.3 Принцип методу посилених дерев

На першому етапі будується початкове дерево T_1, яке прогнозує результати для всього набору даних. Після цього обчислюються залишки, тобто різниця між фактичними значеннями y_i і прогнозами дерева $\hat{y}_i$. Наступне дерево T_2 створюється так, щоб зменшити ці залишки. Алгоритм повторюється, поки не буде досягнуто заданої кількості дерев або поки поліпшення моделі не стане незначним.

Формально цей процес можна описати так. Спочатку задається базовий прогноз $F_0(x)$, який зазвичай є середнім значенням для регресії або найпоширенішим класом для класифікації. На кожній ітерації m обчислюються залишки:

$$r_i^{(m)} = y_i - F_{m-1}(x_i) \qquad (2.7)$$

де $F_{m-1}(x_i)$ — прогноз моделі після $m-1$ ітерацій. Потім нове дерево T_m навчається для зменшення залишків, а модель оновлюється так:

$$F_m(x) = F_{m-1}(x) + \eta T_m(x) \qquad (2.8)$$

де η — коефіцієнт швидкості навчання (learning rate), який визначає, наскільки сильно нове дерево впливає на загальний прогноз.

Однією з ключових особливостей Boosted Trees є використання ваги для кожного прикладу. Якщо приклад важко прогнозується, його вага збільшується, щоб наступне дерево приділило йому більше уваги. Це дозволяє алгоритму адаптуватися до складних випадків і покращувати прогноз.

Для оцінки ефективності дерева на кожній ітерації використовується функція втрат $L(y, \hat{y})$, яка для регресії часто є квадратичною:

$$L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (2.9)$$

де n — кількість прикладів у наборі даних. Boosted Trees мінімізують цю функцію втрат на кожному етапі, що забезпечує поступове покращення моделі.[15]

Метод підсилених дерев показує високу ефективність при роботі з невеликими наборами даних, оскільки він може надати більше ваги рідкісним і важливим випадкам. Завдяки цьому Boosted Trees є чудовим вибором для задач, де потрібно врахувати складні залежності між параметрами. У нашому дослідженні цей метод застосовувався для прогнозування тривалості задач IT-проєктів за такими параметрами, як час аналізу та попередня оцінка.

Boosted Trees дозволяють гнучко адаптуватися до характеристик даних завдяки можливості змінювати кількість ітерацій, швидкість навчання та структуру дерев. Для уникнення перенавчання використовується техніка ранньої зупинки, яка визначає, коли подальші ітерації перестають значно покращувати результати. У порівнянні з Random Forest, метод підсилених дерев досягає вищої точності на більш складних наборах даних, але вимагає ретельного налаштування параметрів і більше часу на навчання.

Таким чином, Boosted Trees стали наступним етапом нашого дослідження після Random Forest, дозволивши не лише покращити точність прогнозів, але й ефективно працювати з невеликою вибіркою даних, які містили складні залежності та варіативність.

Автоматична оптимізація параметрів

Для кожної з моделей було застосовано автоматичну оптимізацію параметрів, що дозволяє досягти кращих результатів, не витрачаючи час на ручне налаштування кожної змінної. За допомогою цієї оптимізації було знайдено найбільш ефективні значення для таких параметрів, як кількість дерев у лісі, максимальна глибина дерев, кількість вузлів на кожному рівні дерева та інші. Це дозволяє зберігати ефективність моделей і робить процес побудови більш автоматизованим.

Оцінка якості моделей

Для оцінки якості моделей було використано кілька метрик, серед яких середня абсолютна помилка (MAE), середня квадратична помилка (MSE) і коефіцієнт детермінації (R-squared). Ці метрики дозволяють отримати точне уявлення про те, наскільки добре модель прогнозує час виконання задач і наскільки ефективно вона адаптується до різних типів даних.

2.2 Дослідження технологій для прогнозування прогнозування термінів виконання IT-проєктів з використанням машинного навчання

Різноманіття платформ для машинного навчання дозволяє ефективно адаптувати інструменти під потреби дослідника, рівень його підготовки та специфіку задач. У цьому розділі ми розглянемо кілька основних платформ, які можна використовувати для прогнозування термінів виконання IT-проєктів, а також їх функціональні можливості, переваги та обмеження. Окрему увагу приділено платформам, які забезпечують гнучкість, зручність у використанні та високу ефективність.

Однією з найбільш зручних платформ є BigML (Machine Learning as a Service). Це хмарне рішення, яке надає користувачам інтуїтивно зрозумілий інтерфейс для створення моделей без необхідності програмування. BigML підтримує різноманітні алгоритми, включаючи дерева рішень, Random Forest, Boosted Trees та методи кластеризації. Платформа також автоматизує налаштування параметрів моделей, що дозволяє досягати високої точності навіть без глибоких знань про внутрішню структуру алгоритмів. Однак, варто зазначити, що ця платформа має певні обмеження у кастомізації складних моделей, а також залежить від стабільного інтернет-з'єднання.[16]

Іншою популярною платформою є Google Colab, яка надає безкоштовне інтерактивне середовище для роботи з Python, зокрема для використання бібліотек машинного навчання, таких як Scikit-learn і TensorFlow. Однією з основних переваг Google Colab є підтримка GPU та TPU, що дозволяє значно пришвидшити навчання моделей, особливо на великих наборах даних. Ця платформа є безкоштовною та доступною онлайн, що робить її ідеальним вибором для досліджень і експериментів. Однак, для ефективної роботи з Google Colab потрібні базові знання програмування, що може бути перешкодою для початківців.[17]

RapidMiner є ще одним інструментом, що дозволяє створювати моделі без написання коду, використовуючи візуальний інтерфейс. Платформа орієнтована на бізнес-користувачів і дослідників, які хочуть швидко створювати моделі для аналізу даних. RapidMiner підтримує великий спектр алгоритмів машинного навчання і дозволяє виконувати весь процес обробки даних, від їх попередньої обробки до створення прогнозів і оцінки результатів. Однак, як і у випадку з іншими безкодовими платформами, можливості кастомізації у RapidMiner можуть бути обмеженими порівняно з використанням більш складних бібліотек, таких як TensorFlow чи PyTorch.[18]

Для автоматизації процесів машинного навчання популярною є платформа H2O.ai, яка реалізує концепцію AutoML. Вона дозволяє автоматично вибирати алгоритми, налаштовувати їх параметри та оцінювати

результати. H2O.ai є потужним інструментом для роботи з великими наборами даних і розробки складних моделей. Її здатність працювати як в локальних, так і в хмарних середовищах надає користувачам гнучкість у виборі інфраструктури. Однак інтерфейс цієї платформи може бути складним для початківців, а також вона вимагає великих обчислювальних ресурсів для деяких задач.[19]

Ще одним інструментом, що набирає популярності серед новачків, є Teachable Machine від Google. Ця платформа дозволяє створювати прості моделі машинного навчання для задач класифікації зображень, звуків або поз без необхідності написання коду. Teachable Machine ідеально підходить для експериментів з базовими задачами, такими як класифікація зображень, і дозволяє швидко отримати результат за допомогою простого інтерфейсу. Проте її функціональність обмежена в порівнянні з іншими платформами, такими як BigML або H2O.ai, і вона не підходить для складних аналітичних задач.[20]

KNIME Analytics Platform є потужним інструментом для візуального аналізу даних і створення моделей. Вона підтримує інтеграцію з іншими мовами програмування, такими як Python та R, і надає можливість працювати з різними типами даних. KNIME забезпечує повний цикл обробки даних і створення прогнозів, що робить його універсальним інструментом для аналізу. Однак, для більш складних задач, KNIME може вимагати додаткових знань програмування.[21]

Orange Data Mining є ще одним візуальним інструментом для машинного навчання, орієнтованим на початківців. Він дозволяє створювати моделі за допомогою інтерфейсу перетягування елементів (drag-and-drop), що робить процес навчання простим і доступним. Orange добре підходить для базового аналізу даних та візуалізації, але для більш складних задач обмежений у можливостях.[22]

Для великих і складних проєктів також популярним інструментом є IBM Watson Studio, яке забезпечує потужні функції для аналізу даних та створення моделей, використовуючи як графічний інтерфейс, так і код. Watson Studio дозволяє працювати з великими наборами даних і забезпечує інтеграцію з

іншими сервісами IBM. Однак ця платформа більше підходить для досвідчених користувачів і корпоративних проєктів, оскільки вимагає високого рівня кваліфікації.[23]

Microsoft Azure Machine Learning є корпоративним рішенням для створення та розгортання моделей машинного навчання. Вона надає безліч можливостей для налаштування, навчання та оптимізації моделей. Azure Machine Learning підходить для великих проєктів і організацій, які потребують високої потужності обчислень та інтеграції з іншими продуктами Microsoft. Водночас її складність та вимоги до технічного рівня користувача можуть бути перешкодою для новачків.[24]

Загалом, вибір платформи залежить від рівня підготовки користувача, специфіки задачі та доступних ресурсів. Наприклад, платформи на кшталт Teachable Machine, BigML та RapidMiner є ідеальними для початківців завдяки їхньому інтуїтивно зрозумілому інтерфейсу та мінімальній потребі в попередніх технічних знаннях. Вони дозволяють швидко створювати базові моделі без потреби у складному налаштуванні. Такі платформи підходять для задач з обмеженими даними або для швидкої валідації ідей. У той же час Google Colab та H2O.ai пропонують розширені можливості для досвідчених користувачів, забезпечуючи доступ до глибшої кастомізації моделей, підтримку роботи з великими наборами даних і потужні інструменти оптимізації параметрів. Ці платформи вимагають більш високого рівня технічної підготовки, але надають значну гнучкість у вирішенні складних задач.

Зведені порівняльні дані про можливості кожної платформи, представлені в таблиці 2.1, дозволяють наочно оцінити їх функціональність і вибрати найбільш підходящий інструмент для вирішення конкретних задач прогнозування термінів виконання IT-проєктів. У процесі вибору платформи необхідно враховувати рівень складності задачі, наявність чи відсутність великих обсягів даних, вимоги до обчислювальних ресурсів і часові обмеження. Наприклад, прості задачі аналізу з невеликими даними можуть бути вирішені на базових платформах, тоді як складні проєкти з численними взаємозв'язками між параметрами потребують потужніших

рішень із розширеними можливостями. Таким чином, правильний вибір платформи є ключовим для досягнення оптимальних результатів та ефективної реалізації проєкту.

Таблиця 2.1 Порівняння платформ для машинного навчання за функціональними параметрами

Параметри	BigML (Machine Learning as a Service)	Google Colab	RapidMiner	H2O.ai	Teachable Machine (Google)	KNIME Analytics Platform	Orange Data Mining	Watson Studio (IBM)	Azure Machine Learning (Microsoft)
Простий інтерфейс, не потребує програмування	+	+	+	-	+	+	+	-	-
Безкоштовний доступ або тестовий період	+	+	+	-	+	+	+	-	+
Підтримка базових моделей (дерева рішень, лінійна регресія)	+	+	+	+	+	+	+	+	+
Підтримка ансамблевих моделей (Random Forest, Boosted Trees)	+	+	+	+	-	+	+	+	+
Підтримка нейронних мереж	+	+	-	+	-	+	-	+	+
Підтримка AutoML (автоматичне налаштування моделей)	+	-	-	+	-	-	-	+	+
Інструменти попередньої обробки даних	+	+	+	+	-	+	+	+	+
Інтерактивна візуалізація даних та результатів	+	+	+	+	+	+	+	+	+

2.3 Функціональні можливості платформи машинного навчання BIGML

BigML — це хмарна платформа для машинного навчання, яка забезпечує широкий спектр інструментів для аналізу даних, побудови моделей та їх використання у реальних бізнес-задачах. Однією з ключових переваг платформи є простота та інтуїтивно зрозумілий інтерфейс, що дозволяє навіть новачкам ефективно працювати з даними.

Варто зазначити, що BigML підтримує широкий спектр моделей, включаючи дерева рішень, ліси рішень (Random Forest), підсилені дерева (Boosted Trees), а також кластеризацію, аналіз аномалій та глибокі нейронні мережі (Deepnets). Це робить платформу універсальним інструментом для розв’язання різноманітних задач прогнозування, класифікації та виявлення аномалій.

Ще однією важливою особливістю є автоматизація процесів. Платформа надає функцію автооптимізації, яка дозволяє автоматично підбирати параметри моделей для досягнення найкращих результатів. Це значно спрощує процес побудови моделей, особливо для тих користувачів, які не мають глибоких технічних знань у сфері машинного навчання.

Також варто відзначити потужні інструменти візуалізації, які пропонує BigML. Користувачі отримують можливість у зручному графічному вигляді аналізувати моделі, переглядати розподіл даних та результати оцінки моделей. Це сприяє кращому розумінню того, як модель працює, і допомагає знаходити можливі точки для її вдосконалення.

Інша важлива особливість платформи — підтримка різноманітних форматів даних. BigML дозволяє імпортувати дані у форматах CSV, Excel, JSON та інших, що значно полегшує інтеграцію з існуючими системами. Це забезпечує гнучкість у роботі з даними, незалежно від їх джерела.

Платформа також пропонує API, який дає змогу інтегрувати створені моделі у зовнішні системи або автоматизувати процеси аналізу даних. Це робить BigML

привабливим вибором для компаній, які прагнуть впроваджувати машинне навчання у свої бізнес-процеси.

Процес роботи з платформою складається з кількох етапів. Першим кроком є завантаження даних, після чого BigML автоматично аналізує датасет, визначаючи типи даних та можливі проблеми, такі як пропущені значення. Наступним кроком є формування моделі. Користувач може налаштувати параметри вручну або скористатися функцією автооптимізації. Після цього необхідно провести оцінку моделі. Платформа автоматично розділяє дані на навчальний та тестовий набори і надає метрики, такі як R-squared, MAE та MSE. Останнім етапом є застосування моделі для прогнозування або її інтеграція через API.

BigML має свої переваги та обмеження. До основних переваг належать простота використання, потужна автоматизація процесів та широкий набір інструментів для роботи з даними. Проте існують і певні обмеження: у безкоштовній версії доступ до деяких функцій обмежений, а також платформа може мати труднощі з обробкою дуже великих наборів даних у хмарному середовищі.[16]

Загалом, BigML є потужним і гнучким інструментом для машинного навчання, який ідеально підходить як для новачків, так і для досвідчених користувачів. У межах даного дослідження платформа була ефективно використана для побудови моделей, що дозволило досягти високої точності прогнозування термінів виконання IT-проєктів.

2.4 Формування датасетів для прогнозування термінів виконання IT-проєктів

Для цього дослідження були використані реальні дані з тікетингової системи, що стосуються виконання IT-проєктів. Дані містять інформацію про час виконання окремих етапів задач, таких як аналіз, розробка, тестування та реліз. Всі задачі були

оцінені за часом виконання на цих етапах, що є основною інформацією для подальшого аналізу та прогнозування.

Дані, що використовуються в дослідженні, мають такі основні параметри:

Час виконання етапів: кожен етап задачі (аналіз, розробка, тестування, реліз) має свій окремий час виконання, що є важливим для оцінки загальної тривалості задач.

Дані для дослідження містять тільки часи виконання етапів, що дають змогу передбачити реальний час виконання задач на основі цих параметрів. Для обробки даних застосовуються стандартні методи попередньої обробки, включаючи нормалізацію числових значень, що дозволяє зменшити варіації в значеннях та привести їх до одного масштабу для подальшого використання в моделях машинного навчання. Загальна структура датасету приведена на рис.2.4.

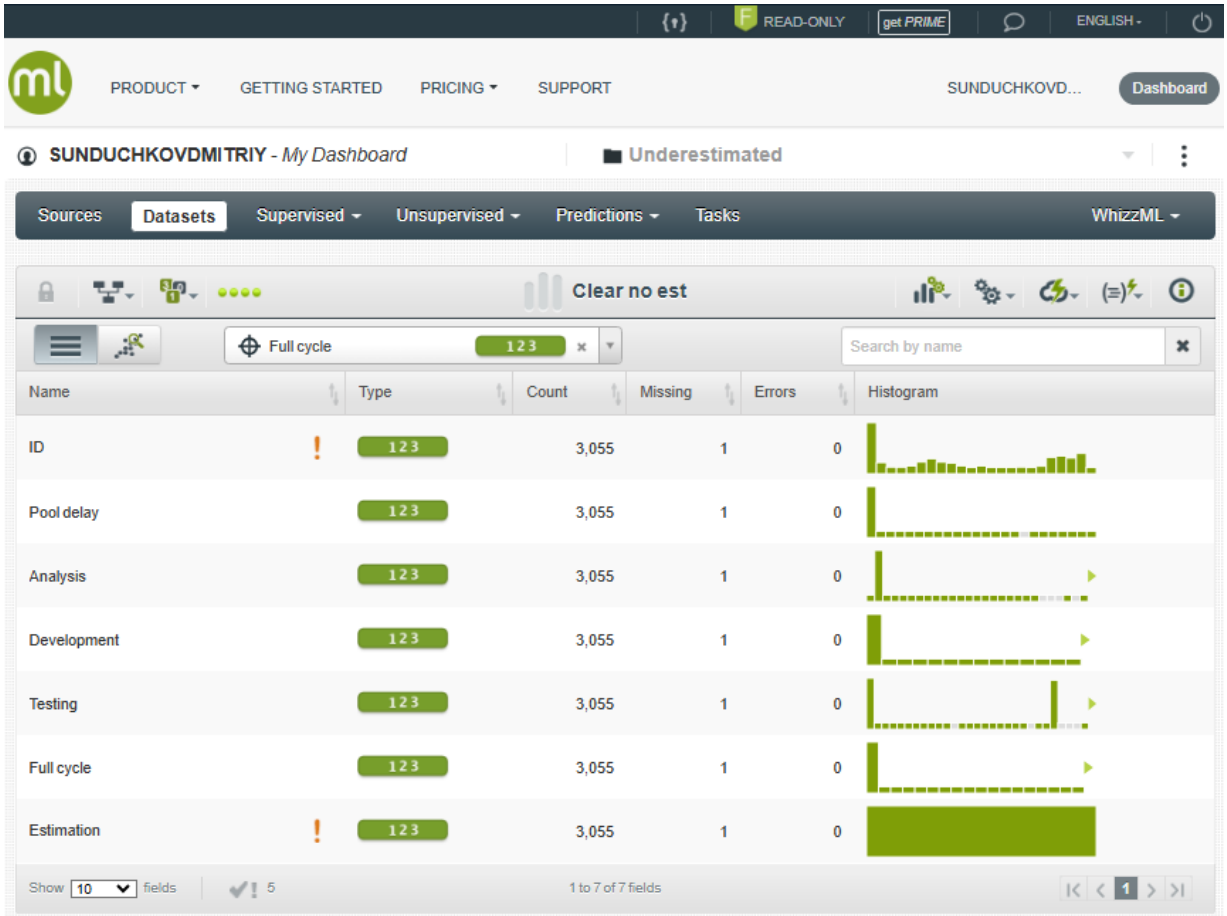


Рис. 2.4 Структура датасету

Попередня обробка даних є важливим етапом перед навчанням моделей машинного навчання. У цьому дослідженні були проведені основні кроки, які

забезпечують правильну підготовку даних до аналізу та покращують точність результатів. Тренд розподілу залежності часу розробки від тривалості аналізу приведено на рис. 2.5.

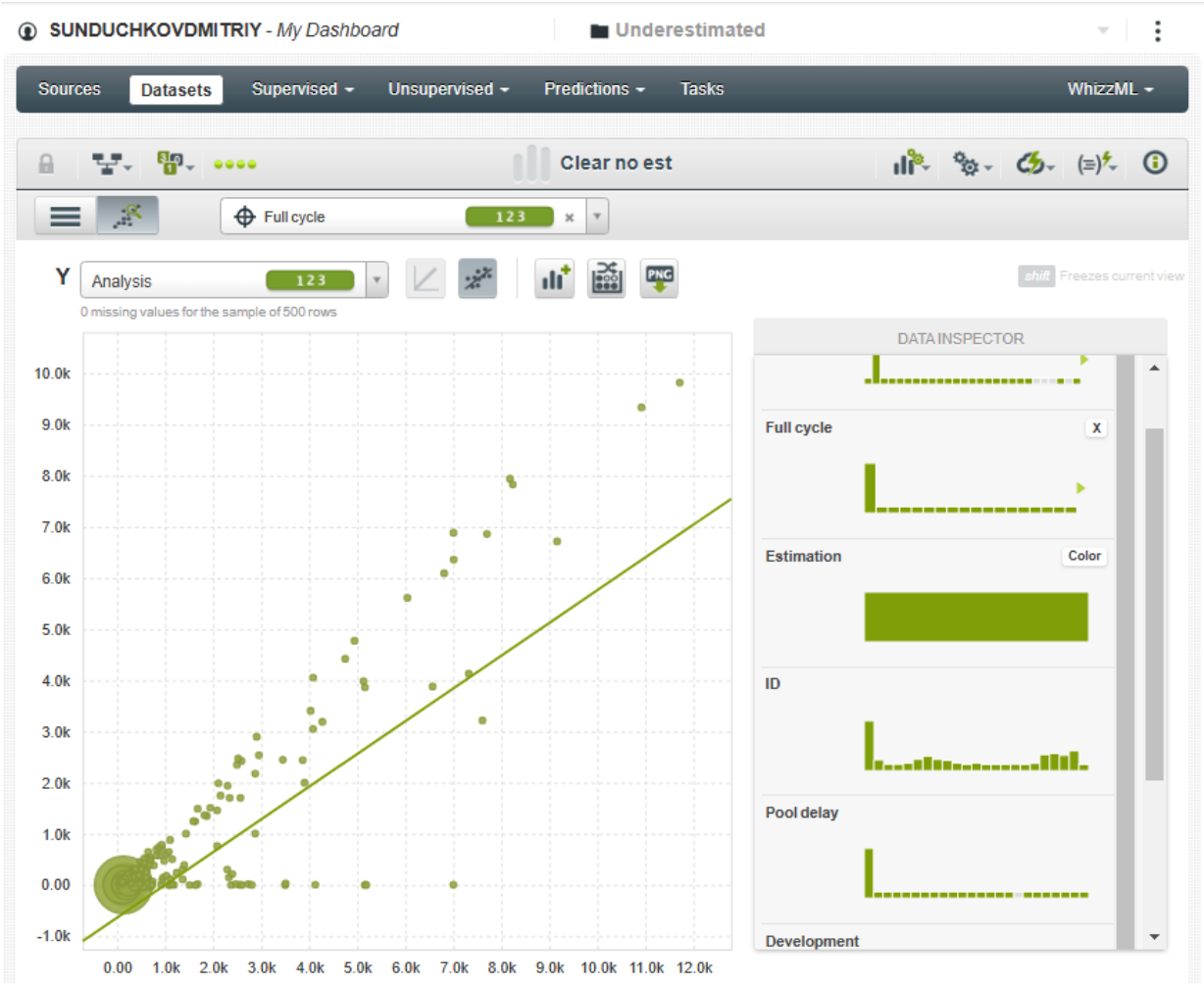


Рис. 2.5 Тренд розподілу залежності часу розробки від тривалості аналізу

Оскільки в даних не було пропущених значень, основним завданням було нормалізувати числові значення. Час виконання кожного етапу (аналіз, розробка, тестування, реліз) може мати різні масштаби, тому було застосовано нормалізацію числових значень, щоб привести їх до одного діапазону і уникнути домінування одних параметрів над іншими під час навчання моделі.

Що стосується поділу даних, вони були розбиті на тренувальний та тестовий набори. Зазвичай для таких досліджень використовується стандартне співвідношення 70/30 або 80/20, де більша частина даних іде на тренування, а менша — на тестування, щоб оцінити точність моделі на нових, не бачених даних.

Також були враховані дані лише для тих задач, де було визначено час виконання кожного етапу, оскільки лише ці записи мали достатньо повну інформацію для навчання моделей. Інші записи були виключені з аналізу, оскільки вони не давали змоги здійснити точні прогнози. Результуючі датасети приведені на рис. 2.6.

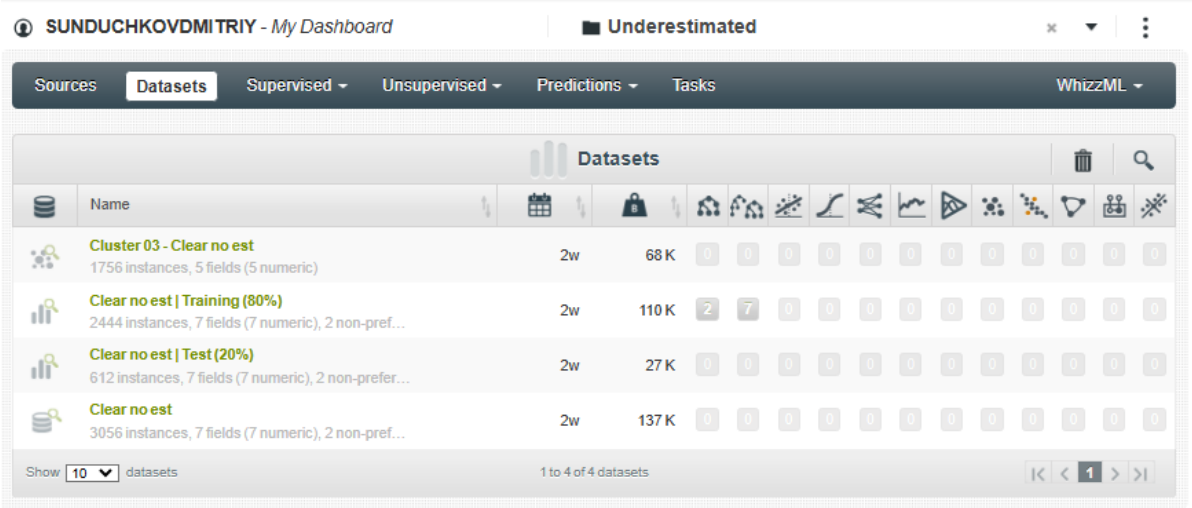


Рис. 2.6 Результуючі розподілені датасети

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАДАЧІ ПРОГНОЗУВАННЯ СТРОКІВ ВИКОНАННЯ ІТ ПРОЄКТІВ З ВИКОРИСТАННЯМ РІЗНИХ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ

3.1 Практична реалізація задачі прогнозування строків виконання ІТ проектів з використанням дерев рішень

Дерево рішень — це один з найпростіших та найпоширеніших алгоритмів машинного навчання, що застосовується для вирішення задач регресії та класифікації. В основі дерева рішень лежить принцип побудови моделі у вигляді дерева, де кожен внутрішній вузол представляє собою перевірку певної ознаки, а кожен листовий вузол містить прогноз. Алгоритм намагається максимізувати інформацію на кожному розбитті, щоб отримати найбільш значущі рішення для подальших етапів.

Процес побудови дерева рішень передбачає пошук найбільш інформативних ознак для кожного вузла, використовуючи критерії для вибору найкращих точок розбиття. Це дозволяє моделі поділити дані на підмножини таким чином, щоб різниця між категоріями або значеннями прогнозу була мінімальною.

Однією з ключових характеристик дерева рішень є його глибина та поріг вузлів:

Глибина дерева визначає, скільки рівнів має дерево, а надмірна глибина може призвести до переобучення.

Поріг вузлів (Node Threshold) контролює мінімальну кількість записів, які повинні бути в вузлі, щоб він був поділений далі.

Дерево рішень будується поетапно, починаючи з кореня (найбільш загальна характеристика), і на кожному кроці модель робить розбиття за найкращими ознаками. Після того, як дерево будується, проводиться оцінка моделі на тестових даних, щоб визначити її точність, скоригувати параметри (якщо потрібно) і забезпечити оптимальні прогнози.

Однак однією з проблем, яка може виникнути при побудові дерева рішень, є переобучення (overfitting), коли модель стає занадто складною і починає враховувати навіть незначні особливості навчальної вибірки, що знижує її здатність до узагальнення на нових даних. Щоб запобігти цьому, застосовують статистичне обрізання (pruning).

Обрізання дерева полягає у видаленні тих гілок дерева, які дають незначне покращення в точності прогнози, але при цьому значно збільшують складність моделі. Обрізання допомагає зробити модель простішою, зменшити її складність і покращити її здатність до узагальнення, зберігаючи високу точність на тестових даних.

Побудова першого дерева рішень.

Параметри моделі:

Depth threshold: 512

Node threshold: 512

Missing splits: No

Statistical pruning: Smart pruning

Перше дерево рішень було побудоване за стандартними налаштуваннями без додаткової оптимізації параметрів. Це дозволило отримати початковий варіант моделі для аналізу, але також призвело до помилок, оскільки модель була занадто простою для складності даних. Результати побудови показано на рис. 3.1.

Рис. 3.1 Вигляд першого дерева рішень

Оцінка результатів:

Середня абсолютна помилка (MAE): 85.36

Середня квадратична помилка (MSE): 169,944.67

Коефіцієнт детермінації (R-squared): 0.95

Перше дерево рішень показало дуже високі результати, з R-squared = 0.95, що свідчить про високу точність моделі для прогнозування часу виконання задач. Середня абсолютна помилка (MAE) та середня квадратична помилка

(MSE) були на рівні 85.36 та 169,944.67 відповідно, що показує добру здатність моделі до прогнозування, але ще є простір для покращення точності. Враховуючи високий R-squared, можна зробити висновок, що перше дерево добре відображає основні закономірності в даних, однак з більшою кількістю налаштувань та оптимізації параметрів можна досягти ще кращих результатів. Результати оцінки моделі показані на рис. 3.2.

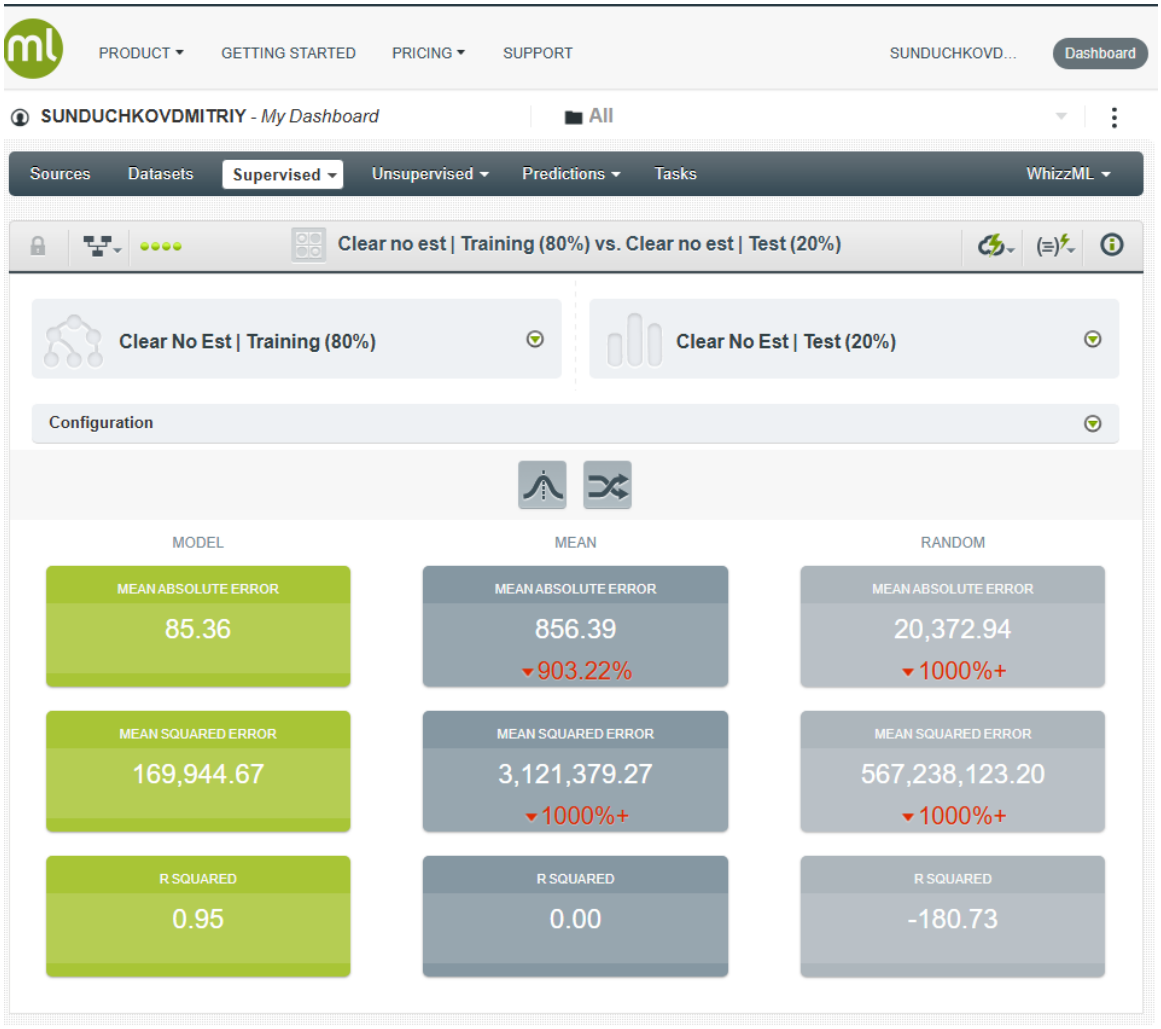


Рис. 3.2. Результати оцінки першого дерева рішень

Побудова другого дерева рішень (auto-optimized)

Параметри моделі:

Model candidates: 128

Depth threshold: 512

Node threshold: 1,008

Missing splits: No

Statistical pruning: No

Для другого дерева рішень були використані параметри автоматичної оптимізації, що включають 128 кандидатів для моделі. Поріг вузлів (Node Threshold) був змінений на значення 1,008, що дозволило зменшити кількість даних на кожному вузлі і зробити модель більш детальною, але менш схильною до переобучення. Глибина дерева була залишена на значенні 512, але статистичне обрізання не було застосоване для досягнення більшої гнучкості моделі. Результати побудови показано на рис. 3.3.

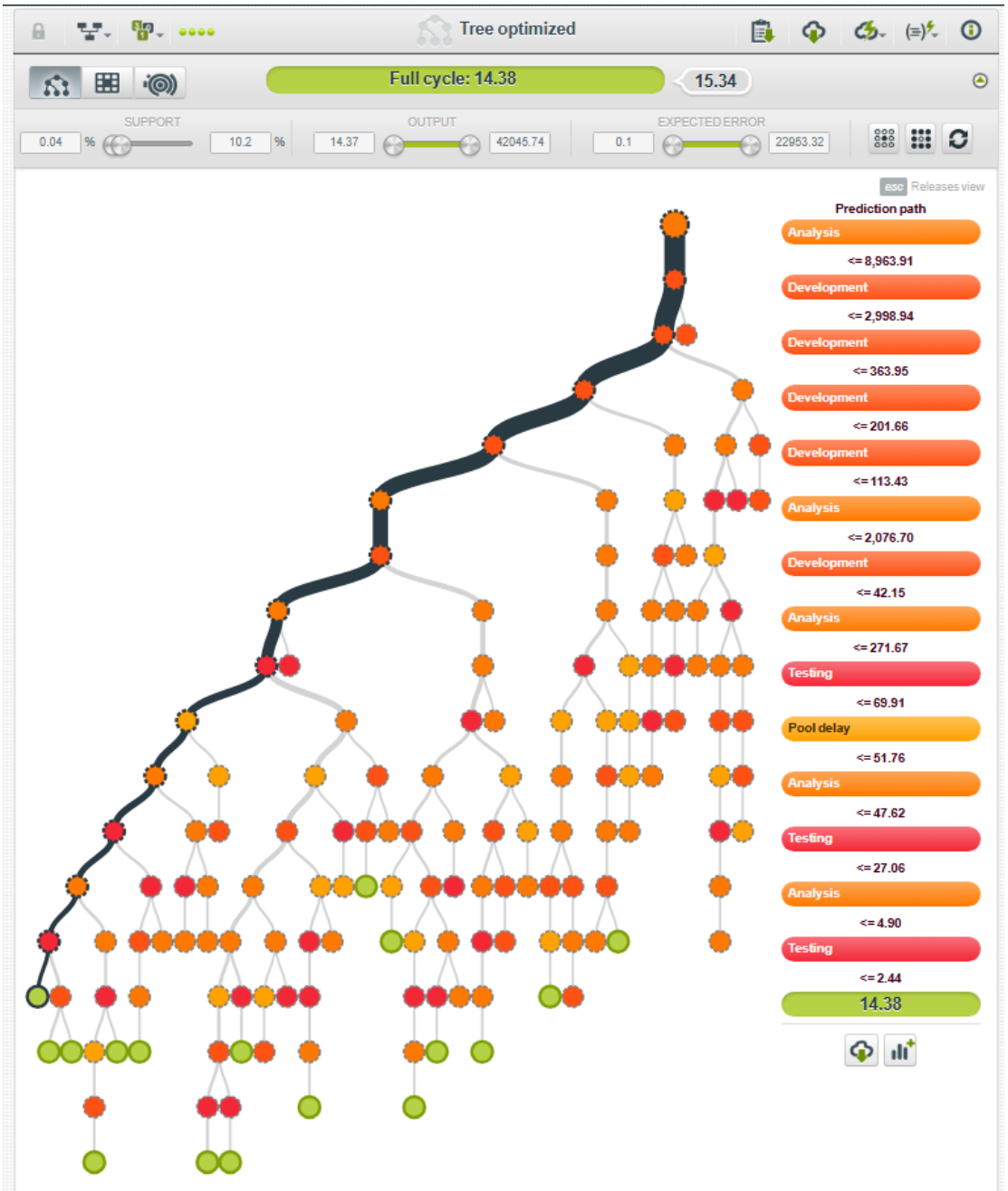


Рис. 3.3. Вигляд другого дерева рішень

Оцінка результатів:

Середня абсолютна помилка (MAE): 123.02

Середня квадратична помилка (MSE): 289,484.41

Коефіцієнт детермінації (R-squared): 0.91

Для другого дерева рішень, яке було автоматично оптимізоване, результати оцінки показали зниження точності: R-squared = 0.91, що свідчить про погіршення якості прогнозування в порівнянні з першим деревом. Середня абсолютна помилка (MAE) і середня квадратична помилка (MSE) збільшилися до 123.02 і 289,484.41 відповідно, що може свідчити про надмірну варіативність в даних, яку модель не змогла ефективно врахувати. Вимкнення статистичного обрізання, разом з більшим порогом вузлів, призвело до меншої стабільності моделі. Це означає, що хоча автоматична оптимізація і покращила деякі параметри, вона не забезпечила істотного поліпшення у порівнянні з першим деревом. Результати оцінки моделі показані на рис. 3.4.

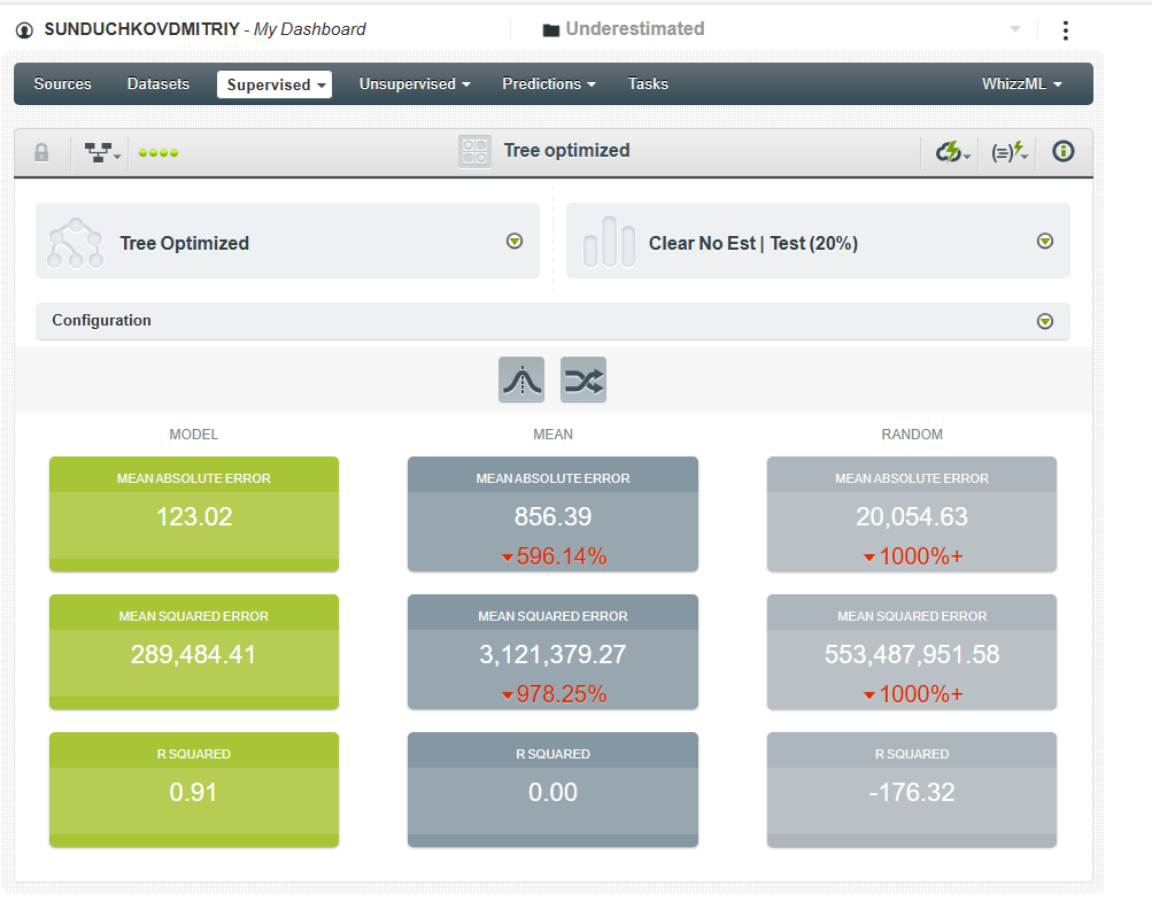


Рис. 3.4. Результати оцінки другого дерева рішень

Побудова третього дерева рішень

Параметри моделі:

Depth threshold: 512

Node threshold: 100

Missing splits: No

Statistical pruning: Smart pruning

Третє дерево було побудоване з іншими параметрами для досягнення кращої точності. Node Threshold був знижений до 100, що дозволило моделі працювати з меншою кількістю даних на кожному вузлі, дозволяючи отримати більшу деталізацію для кожної задачі. Також було застосовано статистичне обрізання для запобігання переобученню, що дозволило зберегти точність на тестовій вибірці, зменшуючи ризик перенавчання на навчальних даних. Результати побудови показано на рис. 3.5.

Рис. 3.5. Вигляд третього дерева рішень

Оцінка результатів:

Середня абсолютна помилка (MAE): 113.90

Середня квадратична помилка (MSE): 172,101.29

Коефіцієнт детермінації (R-squared): 0.94

Третє дерево рішень, яке було побудоване з оптимізованими параметрами, також показало гірші результати порівняно з першим деревом, з R-squared = 0.94. Це свідчить про те, що збільшення глибини дерева та застосування статистичного обрізання не призвели до значного покращення точності. Середня абсолютна помилка (MAE) і середня квадратична помилка (MSE) становили 113.90 та 172,101.29 відповідно. Хоча ці результати і були кращими, ніж для другого дерева, вони все ще не досягли рівня точності, як у першій моделі. Це може вказувати на те, що додаткові налаштування не завжди призводять до значного поліпшення в разі відсутності додаткових даних або ознак. Результати оцінки моделі показані на рис. 3.6.

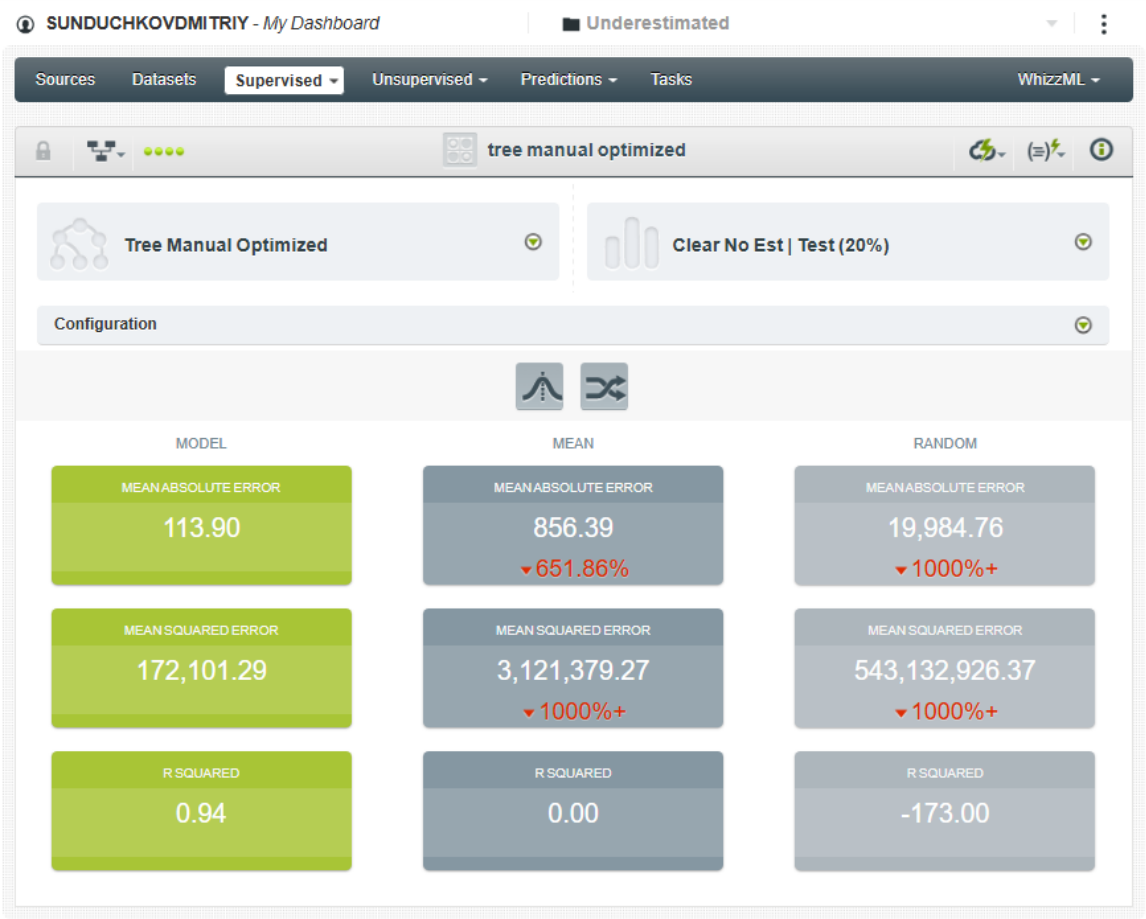


Рис. 3.6. Результати оцінки третього дерева рішень

Висновки по деревам рішень.

Перше дерево показало дуже високий R-squared = 0.95, що підтверджує його ефективність для прогнозування часу виконання задач.

Друге дерево, побудоване з автоматично оптимізованими параметрами, дало гірші результати, з R-squared = 0.91, через вимкнене статистичне обрізання та більший поріг вузлів.

Третє дерево продемонструвало незначне погіршення точності порівняно з першим, хоча R-squared = 0.94 все ще є прийнятним для моделі.

Таким чином, перше дерево рішень виявилось найбільш ефективним для прогнозування часу виконання задач. Для подальших поліпшень було прийнято рішення дослідити ансамблеві методи, такі як Random Forest та Boosted Trees, для досягнення ще кращих результатів.

3.2 Практична реалізація задачі прогнозування строків виконання IT проектів з використанням лісу рішень (Random Forest)

Random Forest (ліс рішень) — це ансамблевий метод машинного навчання, який створює велику кількість незалежних дерев рішень і об'єднує їх прогнози для підвищення точності. На відміну від Boosted Trees, де кожне нове дерево будується на основі помилок попередніх, дерева в Random Forest будуються незалежно одне від одного. Цей підхід дозволяє моделі ефективно уникати переобучення і бути стійкою до варіативності в даних.

Кожне дерево в лісі рішень працює аналогічно тим, які ми будували раніше, але з деякими додатковими особливостями:

Bagging (Bootstrap Aggregating): Для кожного дерева вибирається випадкова підмножина даних із заміною, що дозволяє зменшити кореляцію між деревами.

Random Feature Selection: На кожному вузлі дерева використовується випадковий підмножина ознак для розбиття. Це ще більше зменшує кореляцію і підвищує різноманітність дерев.

Фінальне рішення у Random Forest приймається шляхом усереднення прогнозів для задач регресії або голосування більшості для задач класифікації. Це робить Random Forest дуже стійким до шуму в даних, оскільки вплив окремих неточних дерев мінімізується в результаті комбінування.

Random Forest також надає можливість автоматично налаштовувати параметри, такі як кількість дерев, максимальна кількість вузлів та глибина дерев. Ці параметри можна змінювати вручну для покращення точності і швидкодії, залежно від специфіки задачі.

Побудова першого лісу рішень

Параметри моделі:

Models (n_estimators): 10

Node threshold: 512

Missing splits: No

Statistical pruning: Smart pruning

Процес побудови: Перший ліс рішень був побудований з 10 деревами. Кожне дерево мало Node Threshold = 512, і для зменшення переобучення було застосовано статистичне обрізання (Smart pruning). Це дозволило створити більш стабільну модель порівняно з окремими деревами. Результати побудови показано на рис. 3.7.

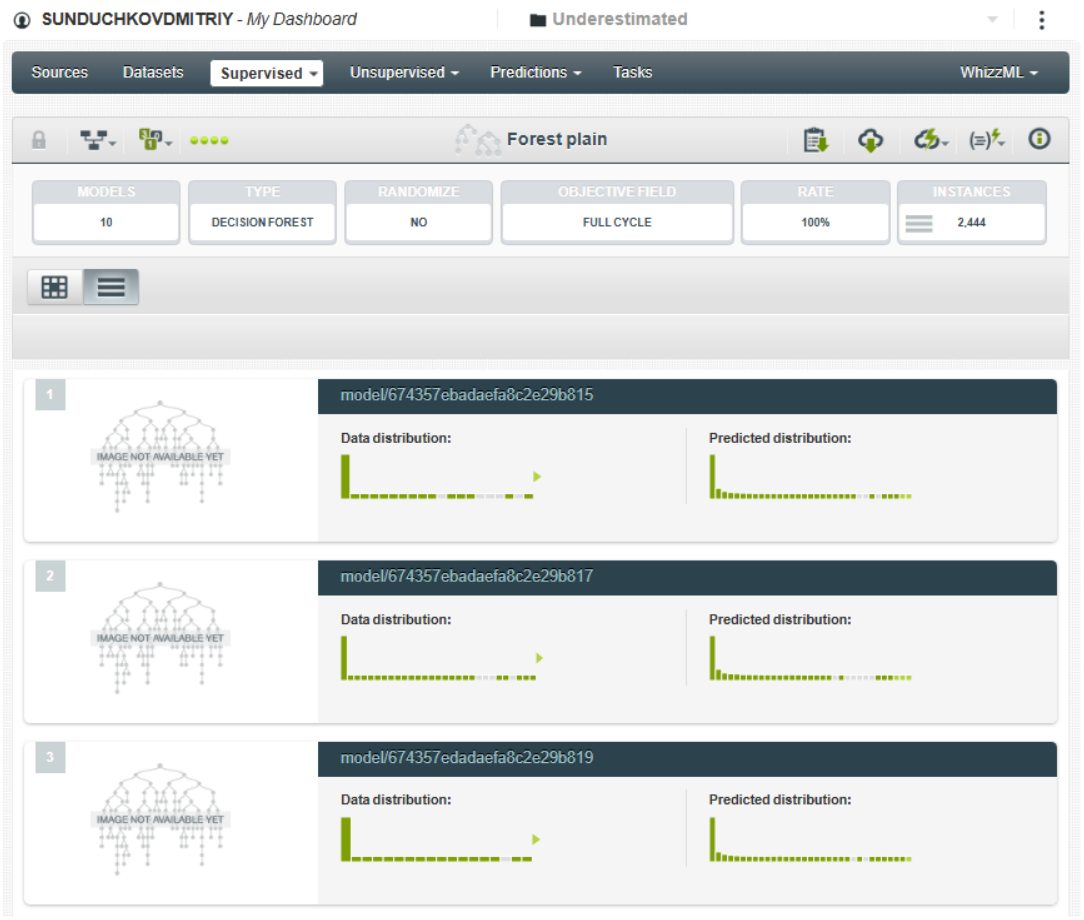


Рис. 3.7. Вигляд першого лісу рішень

Оцінка результатів:

Середня абсолютна помилка (MAE): 53.60

Середня квадратична помилка (MSE): 90,444.80

Коефіцієнт детермінації (R-squared): 0.97

Результати оцінки моделі показано на рис. 3.8.

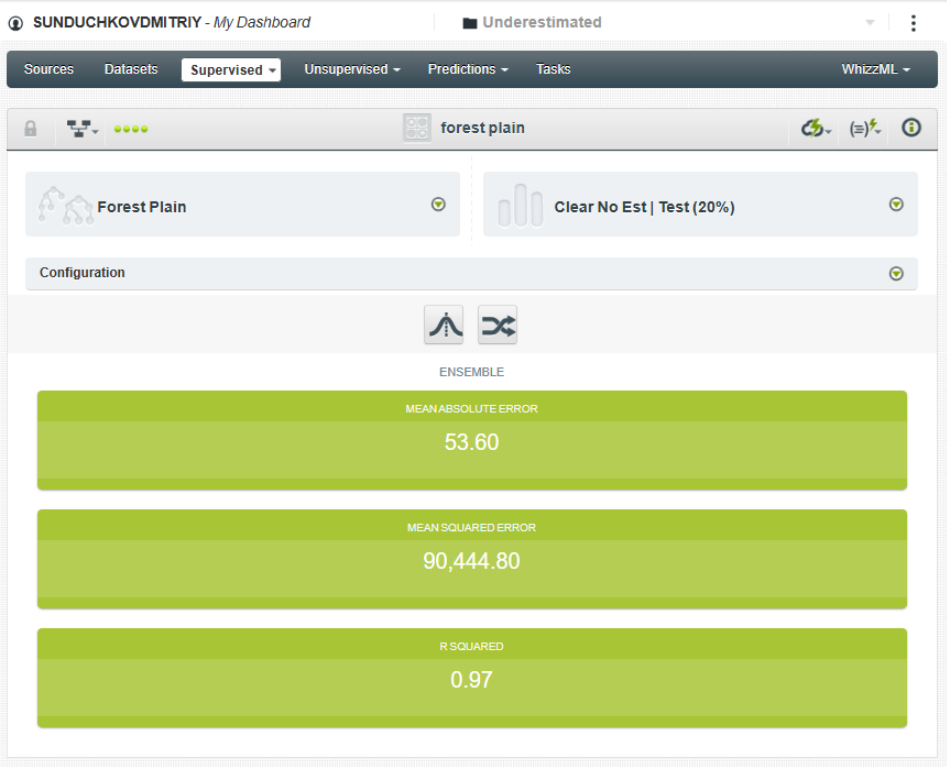


Рис. 3.8. Результати оцінки першого лісу рішень

Перший ліс рішень показав високу точність з R-squared = 0.97, що свідчить про успішне прогнозування часу виконання задач. MAE і MSE також знаходяться на хорошому рівні, що підтверджує ефективність моделі при прогнозуванні. Це дерево рішень значно покращило результат порівняно з одиничними деревами.

Побудова другого лісу рішень

Параметри моделі:

Models (n_estimators): 100

Node threshold: 100

Missing splits: No

Statistical pruning: Smart pruning

Процес побудови:

Для другого лісу рішень було використано 100 дерев. Поріг вузлів був знижений до 100, що дозволило моделі працювати з більш детальними даними на кожному вузлі. Також застосовано статистичне обрізання для запобігання переобученню і покращення стабільності моделі. Результати побудови показано на рис. 3.9.

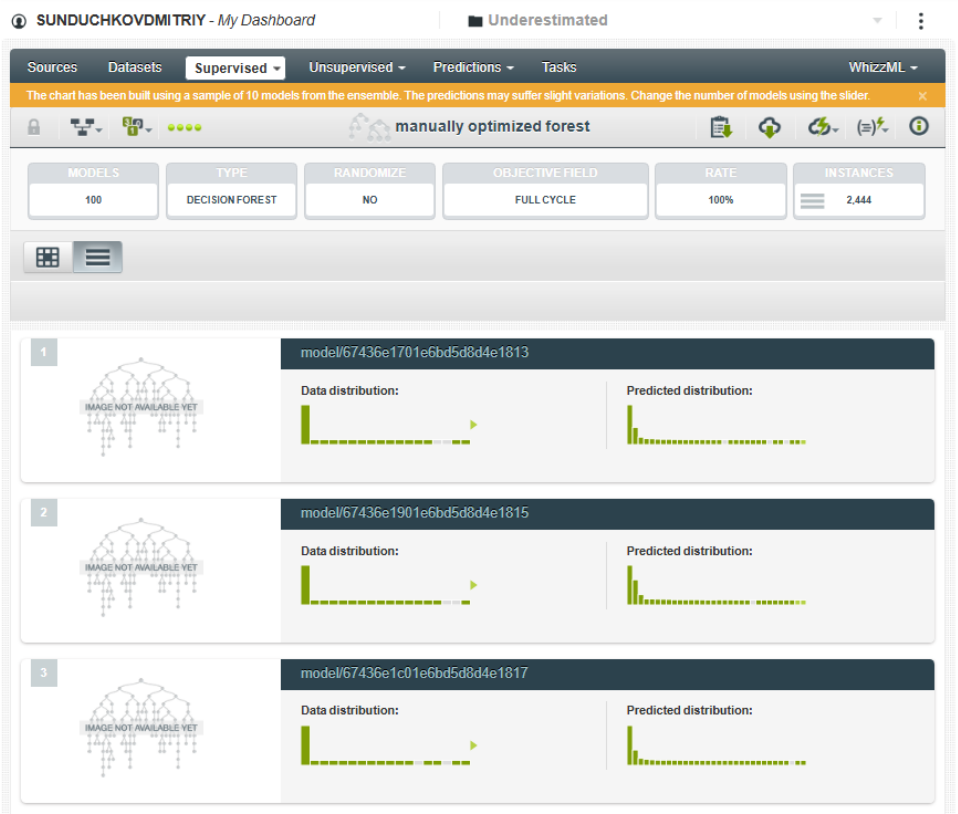


Рис. 3.9. Вигляд другого лісу рішень

Оцінка результатів:

Середня абсолютна помилка (MAE): 75.08

Середня квадратична помилка (MSE): 92,279.90

Коефіцієнт детермінації (R-squared): 0.97

Результати оцінки моделі показано на рис. 3.10.

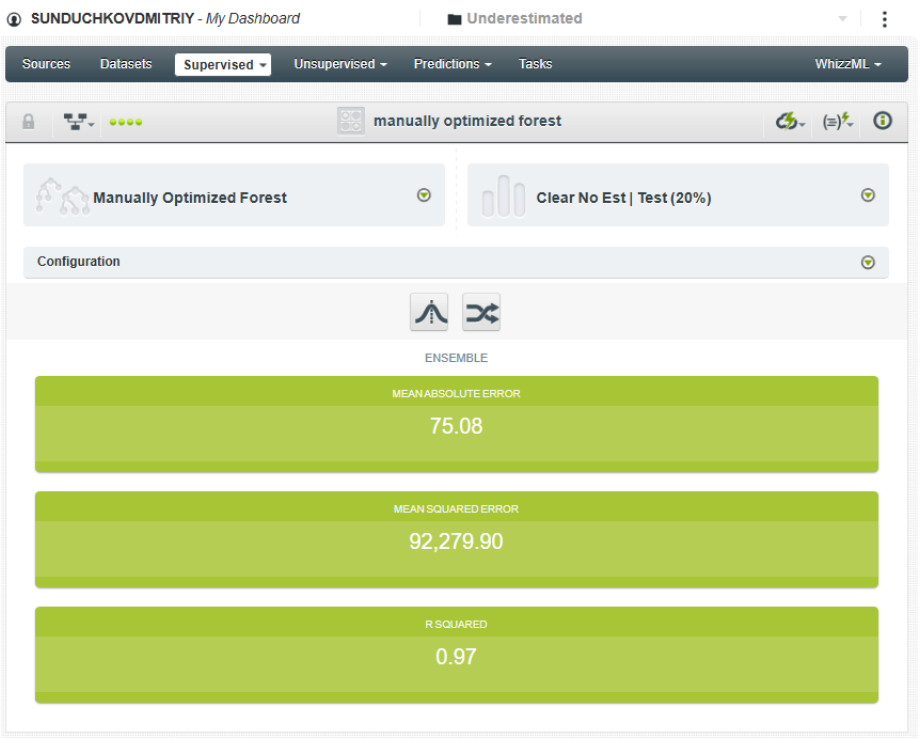


Рис. 3.10. Результати оцінки другого лісу рішень

Другий ліс рішень показав майже ідентичні результати з першим, з R-squared = 0.97, що вказує на високу точність моделі. Зменшення порогу вузлів до 100 не дало значного поліпшення, але модель стала більш стабільною при роботі з великим числом дерев.

Побудова третього лісу рішень: 256 дерев

Параметри моделі:

Models: 256

Node threshold: 512

Missing splits: No

Statistical pruning: Smart pruning

Процес побудови:

Цей ліс рішень був створений з використанням 256 дерев в ансамблі та порогу вузлів 512, що забезпечило більшу кількість дерев у порівнянні з попередніми моделями. Параметр Node threshold було встановлено на рівні 512, що дозволило створити дерева середньої складності без надмірного перенавчання. Вибір Smart

pruning дозволив зменшити складність дерев, одночасно зберігаючи високу точність прогнозів.

Цей ліс був побудований з більшою кількістю дерев, ніж попередні два (де було по 10 та 100 дерев), що дозволило значно знизити варіативність результатів і підвищити точність прогнози. Missing splits було залишено на значенні No, що означає, що пропущені значення не використовувалися під час побудови дерев, тим самим уникнувши складнощів при обробці неповних даних. Результати побудови показано на рис. 3.11.

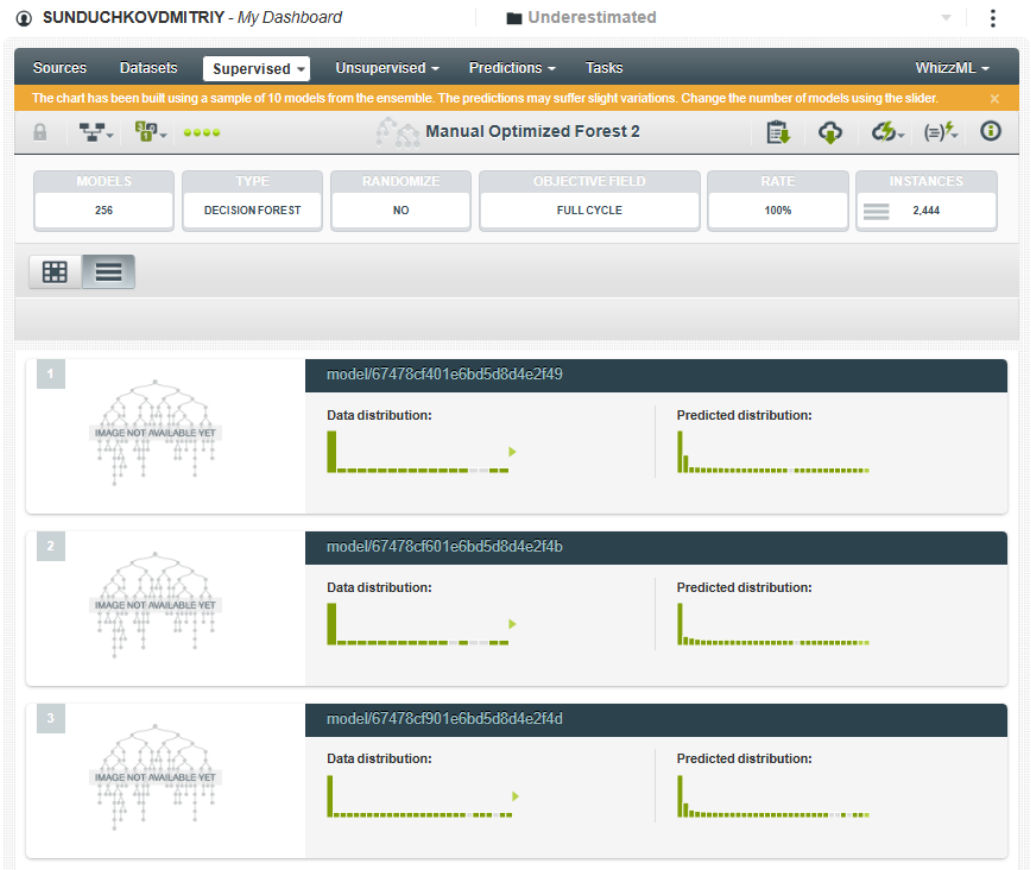


Рис. 3.11. Вигляд лісу рішень (Random Forest) з 256 дерев

Оцінка результатів:

Mean Absolute Error (MAE): 51.42

Mean Squared Error (MSE): 79,582.78

R-squared (R^2): 0.97

Результати оцінки моделі показано на рис. 3.12.

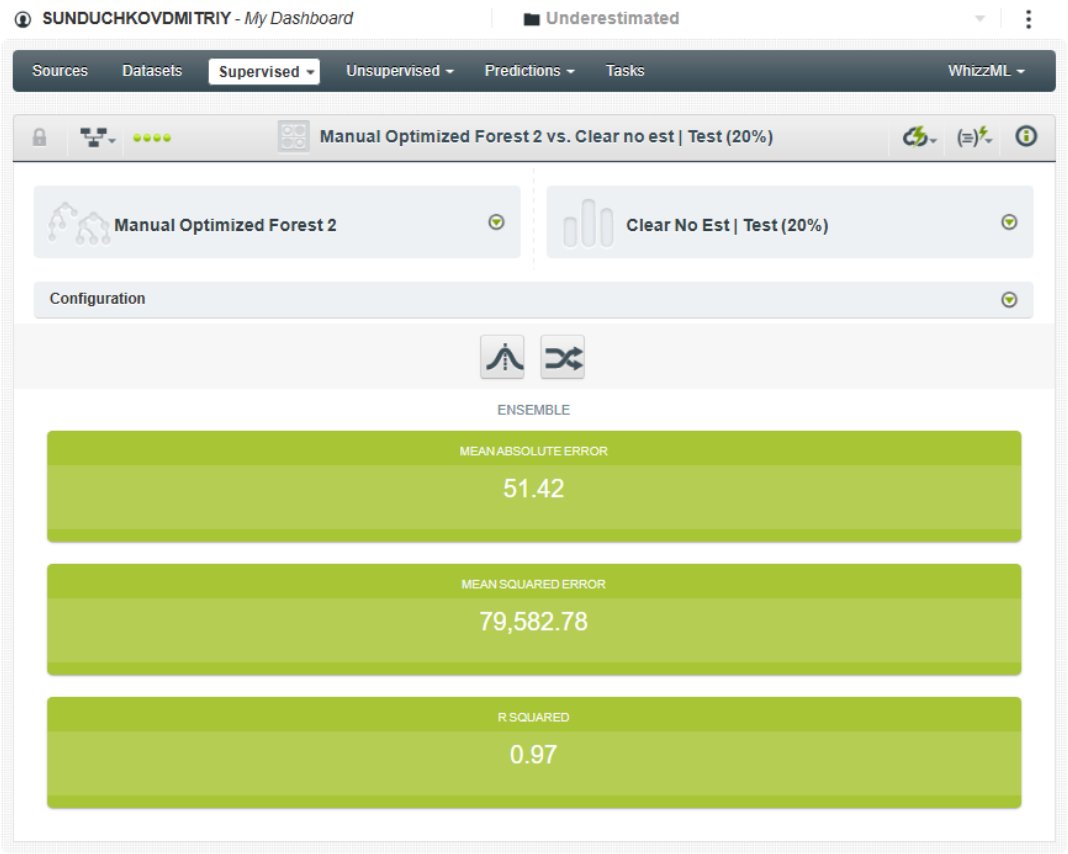


Рис. 3.12. Результати оцінки лісу рішень (Random Forest) з 256 дерев

Цей ліс рішень показав дуже хороші результати з R-squared = 0.97, що свідчить про високу точність моделі. Параметри Mean Absolute Error (MAE) та Mean Squared Error (MSE) виявилися значно кращими, ніж у попередніх моделях з меншими кількостями дерев, що підтверджує здатність моделі досягати більш точної та стабільної прогнози при збільшенні кількості дерев.

Порівняно з Лісом 1 та Лісом 2, цей ліс показав покращення в MAE і MSE, що вказує на перевагу збільшення кількості дерев для досягнення кращих результатів.

R-squared залишився на стабільно високому рівні (0.97), що вказує на сильну здатність моделі до узагальнення.

Це демонструє, що для підвищення точності прогнозів варто збільшувати кількість дерев у ансамблі, що знижує варіативність і покращує узагальнення.

Висновки по лісах рішень

У результаті побудови та оцінки трьох моделей Random Forest, було досягнуто дуже хороших результатів у точності прогнозування термінів виконання задач.

Перший ліс з 10 деревами показав R-squared = 0.97, що свідчить про високу точність моделі. Однак, наявність лише 10 дерев обмежувала здатність моделі до узагальнення, що призвело до деяких обмежень у точності прогнозів. Ця модель дала добрі результати, але можна було досягти більшої стабільності, збільшивши кількість дерев.

Другий ліс з 100 деревами показав подібні результати (також R-squared = 0.97), але при цьому злегка гірші показники Mean Absolute Error (MAE) та Mean Squared Error (MSE) у порівнянні з першим лісом. Збільшення кількості дерев допомогло зменшити варіативність результатів і покращити узагальнення, хоча і не призвело до суттєвих покращень точності.

Третій ліс з 256 деревами продемонстрував найкращі результати серед усіх трьох моделей. Він мав R-squared = 0.97, що аналогічно до попередніх лісів, але показав найкращі показники Mean Absolute Error (MAE) та Mean Squared Error (MSE). Це вказує на те, що збільшення кількості дерев до 256 значно покращує точність прогнозів і стабільність моделі.

Random Forest з більшою кількістю дерев демонструє стабільно високі результати в прогнозуванні, знижуючи помилки і варіативність прогнозів.

Збільшення кількості дерев до 256 дозволило покращити точність моделі, що показує перевагу використання великої кількості дерев для зменшення варіативності результатів.

Усі три моделі показали R-squared = 0.97, що свідчить про їх високу точність, але додавання більше дерев допомогло значно знизити MAE і MSE.

Ці результати підтверджують ефективність Random Forest для прогнозування термінів виконання задач в умовах великих і складних даних. Для подальших покращень можна експериментувати з параметрами, такими як кількість дерев, поріг вузлів та методи обрізання, для досягнення ще кращих результатів.

3.3 Практична реалізація задачі прогнозування строків виконання IT проектів з використанням підсилених дерев (Boosted Trees)

Boosted Trees — це ансамблевий метод машинного навчання, який поступово будує модель, додаючи нові дерева для корекції помилок попередніх. На відміну від лісу рішень, де кожне дерево будується незалежно і результати узагальнюються голосуванням, Boosted Trees додає кожне нове дерево з урахуванням помилок, зроблених попередніми деревами. Це дозволяє ефективно враховувати складні взаємозв'язки в даних і поступово покращувати точність.

Кожне дерево в Boosted Trees працює аналогічно тим, які ми будували на початку: воно використовує розбиття вузлів і параметри, характерні для дерев рішень. Проте, на відміну від окремих дерев, ці моделі створюються послідовно, причому кожне нове дерево додається з метою мінімізації залишкових помилок від попередніх дерев. Після завершення ітерацій модель використовує підсилене зважене голосування, де прогноз кожного дерева комбінується з урахуванням його ваги, визначеної на основі його внеску до зниження загальної помилки.

У Boosted Trees кінцеве рішення приймається шляхом сумування прогнозів усіх дерев у ансамблі, але з урахуванням того, як добре кожне дерево скоригувало попередні помилки. Це відрізняється від методу голосування, що використовується в лісі рішень. Застосування ранньої зупинки (Early Stopping) дозволяє завершити побудову, як тільки покращення точності перестають бути суттєвими, що забезпечує баланс між точністю моделі та її складністю.

Побудова першої моделі Boosted Trees

Параметри моделі:

Models: 55

Iterations: 64

Node threshold: 512

Learning rate: 10%

Missing splits: No

Early stopping: Out of Bag

Процес побудови:

Під час побудови моделі використовувалася функція ранньої зупинки (Early Stopping), яка зупинила процес після 55 дерев із запланованих 64 ітерацій. На кожному кроці оцінювалася якість прогнозів на основі out-of-bag samples — підмножини даних, які не використовувалися для навчання в поточній ітерації. Якщо модель більше не покращувала результати на цих даних, навчання зупинялося. Це дозволило уникнути перенавчання і водночас зберегти високу якість прогнозів. Результати побудови показано на рис. 3.13.

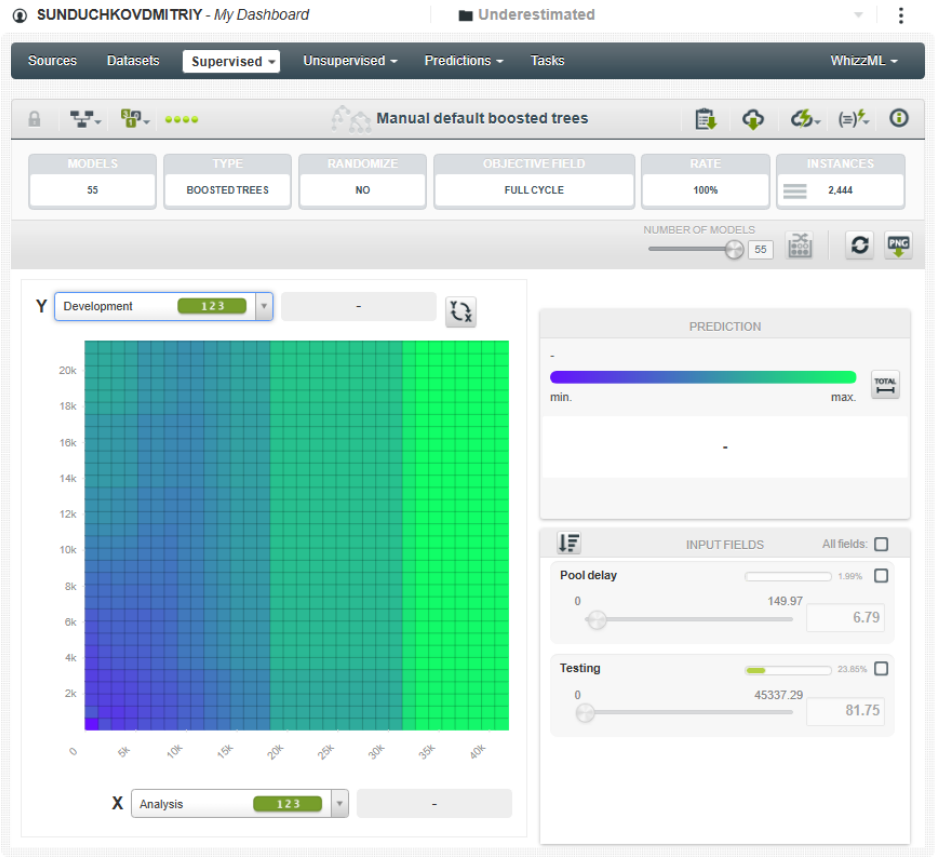


Рис. 3.13. Вигляд першої моделі Boosted Trees

Оцінка результатів:

Mean Absolute Error (MAE): 41.77

Mean Squared Error (MSE): 57,902.41

R-squared: 0.98

Результати оцінки моделі показані на рис. 3.14.

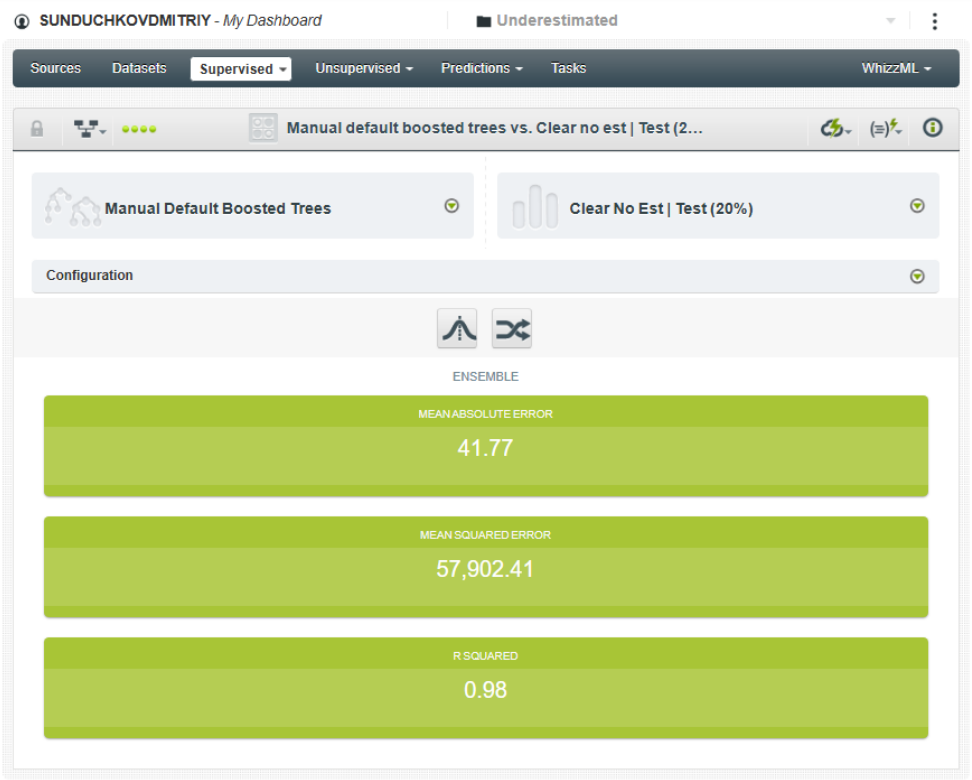


Рис. 3.14. Результати оцінки першої моделі Boosted Trees

Перша модель Boosted Trees показала високі результати з R-squared = 0.98, що свідчить про дуже високу точність моделі. Mean Absolute Error (MAE) та Mean Squared Error (MSE) значно знизилися порівняно з лісом рішень. Це підтверджує, що метод підсилення ефективно зменшує помилки, поступово вдосконалюючи прогноз. Використання ранньої зупинки не лише скоротило час навчання, але й запобігло перенавчанню, дозволивши зупинитися на оптимальній кількості дерев для досягнення найкращого результату.

Побудова другої моделі Boosted Trees (auto-optimized)

Параметри моделі:

Training duration: 5

Ensemble candidates: 128

Models: 15

Node threshold: 1,923

Iterations: 15

Процес побудови:

Модель була побудована з використанням автооптимізації, де деякі параметри, такі як Training duration та Ensemble candidates, були задані вручну. Інші параметри, зокрема Models, Node threshold та Iterations, були автоматично розраховані системою в процесі оптимізації. Система вибрала оптимальні значення для цих параметрів на основі тестових даних, щоб досягти найкращих результатів.

Під час навчання використовувалася рання зупинка (Early Stopping), яка зупинила процес після 15 ітерацій, коли модель більше не демонструвала суттєвого покращення на тестових даних. Це дозволило заощадити час та ресурси, зберігши високу точність моделі. Результати побудови показано на рис. 3.15.

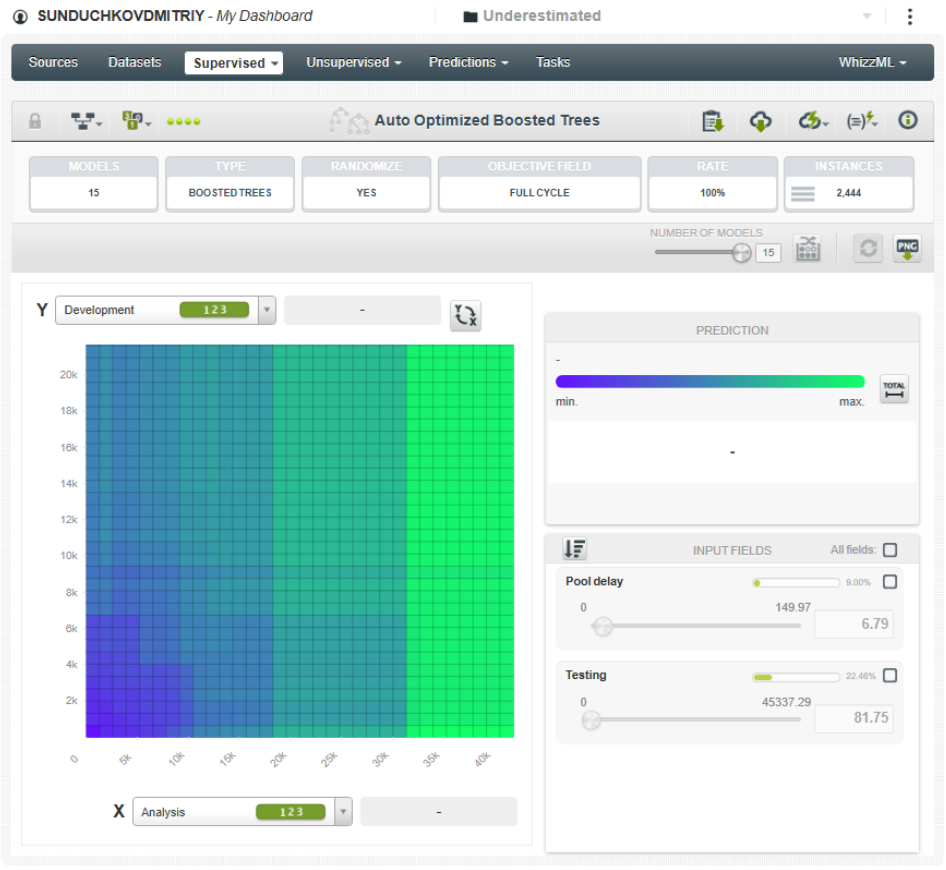


Рис. 3.15. Вигляд автооптимізованого Boosted Trees

Оцінка результатів:

Mean Absolute Error (MAE): 70.54

Mean Squared Error (MSE): 171,018.72

R-squared (R^2): 0.95

Результати оцінки моделі показано на рис. 3.16.

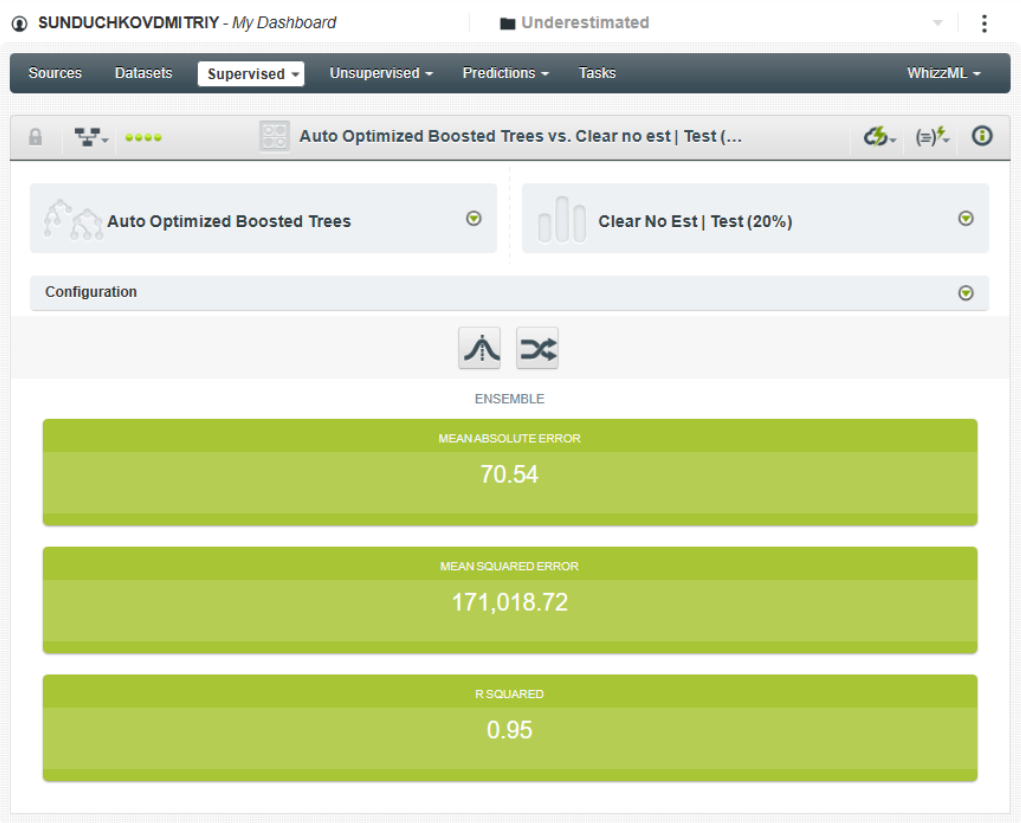


Рис. 3.16. Результати оцінки автооптимізованого Boosted Trees

Автооптимізоване дерево показало результат R-squared = 0.95, що є дуже хорошим результатом, однак трохи нижчим за R-squared = 0.98 першої моделі Boosted Trees. Відносно Mean Absolute Error (MAE) та Mean Squared Error (MSE), нова модель показала трохи більші значення, що свідчить про те, що, хоча автоматична оптимізація значно покращила точність, вона все ж не змогла досягти таких самих результатів, як вручну налаштовані параметри першої моделі.

Завдяки ранній зупинці та швидкості навчання, модель змогла досягти хороших результатів без перенавчання, що є суттєвою перевагою, оскільки цей процес зазвичай займає менше часу. Однак порівняння з першою моделлю показує, що в деяких випадках додаткові налаштування вручну можуть призвести до більш точних прогнозів, хоча автоматична оптимізація й не залишає значних недоліків.

Побудова третьої моделі Boosted Trees

Параметри моделі:

Models: 50

Node threshold: 100

Iterations: 256

Missing splits: No

Early stopping: Early out of bag

Learning rate: 10%

Модель була побудована з використанням покращеної версії Boosted Trees, де кількість дерев збільшена до 50, а ітерації доведені до 256 для досягнення найкращих результатів. Використання ранньої зупинки дозволило уникнути перенавчання і скоротити час навчання, зупинивши його, коли модель більше не демонструвала істотного покращення на out-of-bag samples. Результати побудови показано на рис. 3.17.

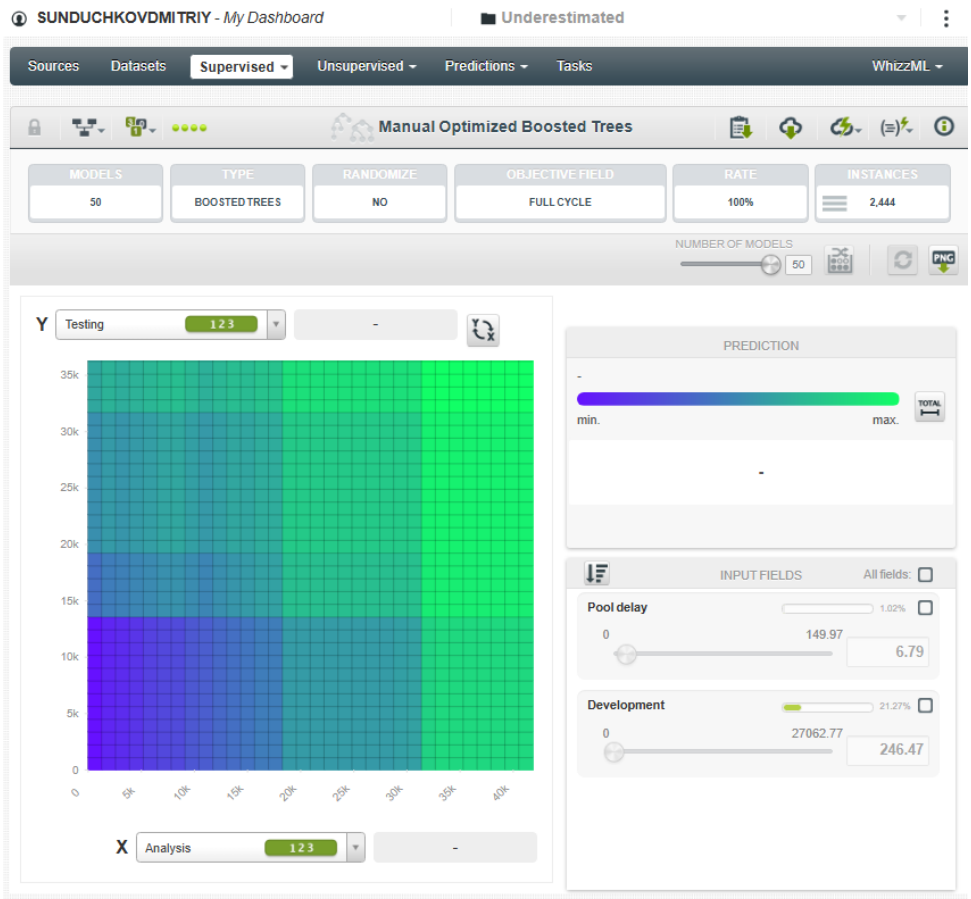


Рис. 3.17. Вигляд покращеного Boosted Trees

Оцінка результатів:

Mean Absolute Error (MAE): 54.28

Mean Squared Error (MSE): 33,484.17

R-squared (R^2): 0.99

Результати оцінки моделі показано на рис. 3.18.

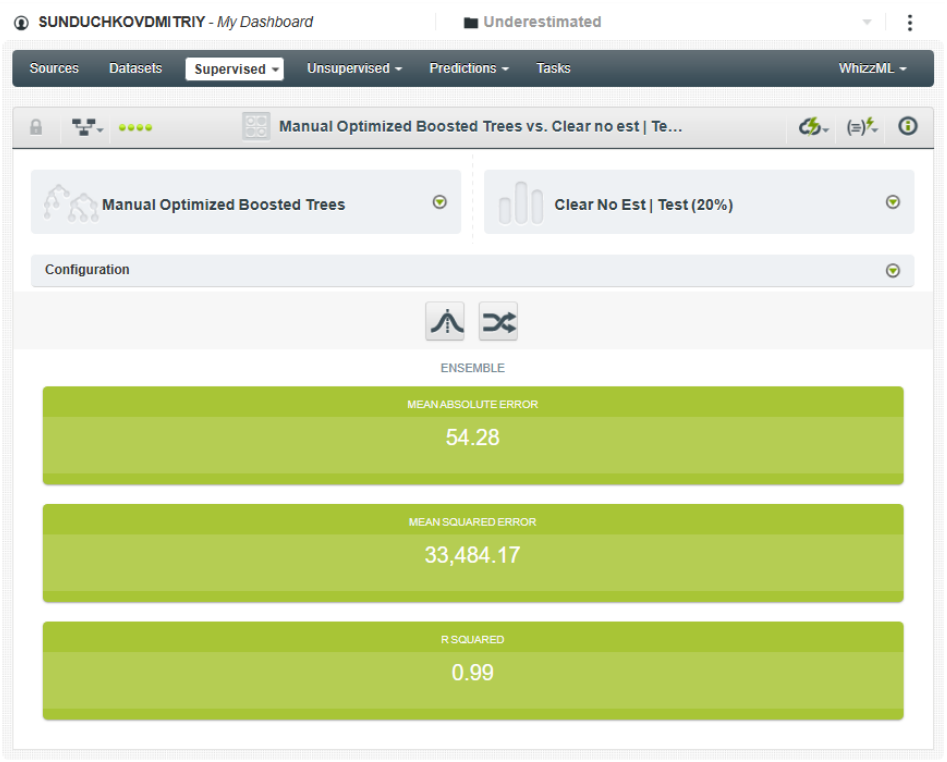


Рис. 3.18. Результати оцінки покращеного Boosted Trees

Ця модель продемонструвала чудові результати, досягнувши R-squared = 0.99, що свідчить про надзвичайно високу точність прогнозів. Mean Absolute Error (MAE) і Mean Squared Error (MSE) залишаються дуже низькими, що свідчить про хорошу узагальнювальну здатність моделі. Важливою перевагою цієї моделі є рання зупинка, яка дозволила оптимізувати час навчання і уникнути перенавчання при досягненні найкращого результату.

Висновки по підсилених деревах

У результаті побудови та оцінки трьох моделей Boosted Trees було отримано значні покращення у точності прогнозування порівняно з іншими типами моделей, такими як Random Forest.

Перша модель Boosted Trees (вручну налаштовані параметри) показала дуже високі результати з R-squared = 0.98, що свідчить про дуже точні прогнози. Вона

продемонструвала найкращі результати серед усіх моделей Boosted Trees за Mean Absolute Error (MAE) та Mean Squared Error (MSE).

Автооптимізоване дерево Boosted Trees досягло R-squared = 0.95, що також є хорошим результатом, але трохи поступається результатам вручну налаштованої моделі. Однак це дозволило заощадити час і ресурси, завдяки автоматичному налаштуванню параметрів, без потреби в додаткових налаштуваннях вручну.

Покращена модель Boosted Trees, з більшою кількістю дерев та ітерацій, показала R-squared = 0.99, що є найкращим результатом серед трьох моделей. Це свідчить про дуже високу точність прогнозів. Завдяки ранній зупинці (Early Stopping), модель змогла уникнути перенавчання і досягти оптимального результату за менший час.

Моделі Boosted Trees, як вручну налаштовані, так і автооптимізовані, продемонстрували значно кращі результати в порівнянні з іншими моделями, зокрема, Random Forest. Mean Absolute Error (MAE) та Mean Squared Error (MSE) були низькими, що свідчить про високу точність прогнозів.

Рання зупинка (Early Stopping) в обох моделях допомогла уникнути перенавчання, дозволивши моделі зупинитися на оптимальному етапі навчання, зберігаючи високу точність та економлячи час.

Швидкість навчання (Learning Rate) в моделі з автооптимізацією була налаштована на 43.5%, що дозволило швидко адаптувати модель до даних, але із збереженням стабільності.

Загалом, використання Boosted Trees дозволяє досягти високої точності прогнозів. Перша модель, створена вручну, показала кращі результати, однак автооптимізовані моделі також демонструють хорошу ефективність, заощаджуючи час на налаштуваннях. Покращена модель Boosted Trees продемонструвала найкращі результати серед усіх трьох моделей, що робить її найефективнішою для вирішення задач прогнозування термінів виконання задач.

У майбутньому, для досягнення ще кращих результатів, можна розглянути можливість змінювати learning rate, node threshold, iterations, а також використовувати більшу кількість дерев в ансамблі.

3.4 Аналіз результатів

У цьому розділі проведено детальний аналіз хронології дослідження та прийнятих рішень на кожному етапі. Для порівняння моделей використовувалися ключові метрики: R-squared (R^2), Mean Absolute Error (MAE) та Mean Squared Error (MSE), які дозволяють оцінити точність та ефективність прогнозів.

Дослідження розпочалося зі створення моделі дерева рішень з базовими параметрами. Перша модель показала хороші результати: $R^2 = 0.95$, але деякі показники, зокрема MAE та MSE, вказували на можливість покращення.

На наступному етапі було вирішено покращити параметри моделі, зокрема node threshold. Це дозволило побудувати другу модель дерева рішень, яка продемонструвала трохи кращі показники точності. Однак, навіть з оптимізованими параметрами, модель мала обмеження у точності прогнозування.

Третє дерево рішень було створено з використанням функції автооптимізації. Ця модель виявилася менш ефективною за інші дерева, що свідчить про те, що автоматичний вибір параметрів не завжди забезпечує оптимальні результати.

Після аналізу результатів дерев рішень було прийнято рішення перейти до побудови лісу рішень, що дозволяє об’єднати результати кількох дерев для покращення прогнозів.

Перша модель лісу рішень складалася з 10 дерев і показала $R^2 = 0.97$, що свідчить про суттєве покращення точності у порівнянні з деревами рішень. Проте було помітно, що збільшення кількості дерев може ще більше знизити помилки прогнозування.

На наступному етапі було побудовано другий ліс рішень зі 100 деревами. Ця модель зберегла високу точність ($R^2 = 0.97$), але MAE та MSE виявилися трохи вищими, ніж у першого лісу.

Остання модель лісу рішень з 256 деревами показала найкращі результати серед лісів, значно знизивши помилки прогнозування (MAE = 51.42, MSE = 79,582.78), при цьому зберігши $R^2 = 0.97$.

Після аналізу результатів лісів було прийнято рішення спробувати Boosted Trees, що дозволяють підсилювати результати кожної наступної ітерації.

Перша модель Boosted Trees, створена вручну, показала дуже високий результат $R^2 = 0.98$ з мінімальними помилками. Це продемонструвало перевагу підсилених дерев у точності прогнозів.

Друга модель була створена за допомогою автооптимізації, але її результати виявилися гіршими ($R^2 = 0.95$). Це свідчить, що автоматичне налаштування не завжди забезпечує оптимальні результати.

Третя модель Boosted Trees, з покращеними параметрами, досягла найкращого результату серед усіх моделей: $R^2 = 0.99$, що свідчить про найвищу точність прогнозів. Зведені дані щодо оцінки всіх моделей представлені в таблиці 3.1.

Таблиця 3.1. Порівняння результатів моделей

Модель	R-squared	Mean Absolute Error (MAE)	Mean Squared Error (MSE)
Перше дерево рішень	0.95	85.36	169,944.67
Друге дерево рішень	0.94	113.90	172,101.29
Третє дерево рішень	0.94	123.02	289,484.41
Перший ліс рішень (Random Forest, 10 дерев)	0.97	53.60	90,444.80

Продовження таблиці 3.1.

Модель	R-squared	Mean Absolute Error (MAE)	Mean Squared Error (MSE)
Другий ліс рішень (Random Forest, 100 дерев)	0.97	75.08	92,279.90
Третій ліс рішень (Random Forest, 256 дерев)	0.97	51.42	79,582.78
Перша модель Boosted Trees	0.98	41.77	57,902.41
Автооптимізовані Boosted Trees	0.95	70.54	171,018.72
Покращена модель Boosted Trees (вручну налаштована)	0.99	54.28	33,484.17

Реальне значення і користь

Реальні результати, отримані в результаті побудови та оцінки моделей, мають практичну цінність для прогнозування термінів виконання IT-проєктів. Порівняння точності різних моделей, зокрема дерев рішень, Random Forest і Boosted Trees, продемонструвало високу ефективність у прогнозуванні тривалості задач на основі даних, таких як час аналізу та попередні оцінки складності. Це дозволяє точніше планувати проєкти, визначати можливі затримки та оптимізувати ресурси. Різні моделі продемонстрували свої переваги, зокрема Random Forest та Boosted Trees, які знизили рівень помилок і підвищили точність прогнозів порівняно з одиничними деревами рішень.

Моделі машинного навчання, побудовані в рамках цього дослідження, можуть бути інтегровані в робочі процеси команд, що займаються управлінням IT-проєктами. Завдяки автоматизації процесу прогнозування, організації часу та розподілу ресурсів, керівники проєктів можуть отримати точніші та оперативніші оцінки термінів виконання задач. Це дозволить швидше виявляти потенційні проблеми в графіку і здійснювати коригування на ранніх етапах реалізації проєкту.

Інтеграція цих моделей у систему управління проєктами дозволяє зменшити ризики затримок і підвищити ефективність командної роботи.

Побудовані моделі можна інтегрувати в існуючі системи планування та моніторингу IT-проєктів, що дозволить автоматизувати процес оцінки термінів виконання задач. Такі моделі можуть бути вбудовані в програмне забезпечення для управління проєктами, що дозволить керівникам та командам отримувати рекомендації на основі аналізу даних про етапи виконання задач. Інтеграція з системами управління задачами дозволяє автоматично надавати прогнози на основі актуальних даних, значно знижуючи людський фактор і покращуючи точність планування.

Потенційні напрямки покращення моделей включають підвищення їх точності за допомогою більш детальної обробки даних, використання більш складних моделей машинного навчання, таких як нейронні мережі та глибинне навчання. Можна також експериментувати з новими методами попередньої обробки даних, що дозволить підвищити якість прогнозів. Окрім цього, можна вдосконалити існуючі моделі шляхом оптимізації параметрів, збільшення обсягу навчальних даних та використання більш специфічних характеристик задач для точнішого прогнозування. Подальше дослідження також може зосереджуватись на адаптації моделей для специфічних типів IT-проєктів та тестуванні їх на різноманітних реальних наборах даних.

ВИСНОВКИ

У межах дослідження було проведено детальний аналіз методів прогнозування термінів виконання ІТ-проєктів, що включав як теоретичну, так і практичну частини. Розроблений алгоритм на основі машинного навчання дозволив суттєво підвищити точність прогнозів, що є вагомим внеском у забезпечення успішного виконання ІТ-проєктів.

На першому етапі дослідження було проведено аналіз проблематики прогнозування термінів виконання ІТ-проєктів. Було виявлено, що сучасні методи, такі як PERT і CPM, хоч і широко застосовуються, мають обмеження, пов'язані з точністю оцінки в умовах невизначеності. У той же час, використання машинного навчання виявилося перспективним підходом, який дозволяє враховувати взаємозв'язки між багатьма параметрами проекту.

Детальний аналіз сучасних методів машинного навчання показав, що моделі, засновані на деревах рішень, лісах рішень та підсилених деревах, є одними з найбільш ефективних для задач прогнозування. Вони забезпечують високу точність і гнучкість при роботі з великими масивами даних. Крім того, ці моделі здатні адаптуватися до різних типів даних, що робить їх універсальними.

Було досліджено основні технологічні платформи для реалізації задач машинного навчання, такі як Google Colab, RapidMiner, H2O та BigML. Серед них платформа BigML була обрана для практичної частини дослідження через її зручний інтерфейс, широкий набір інструментів для роботи з моделями і можливість автоматизації процесу побудови моделей.

Особлива увага була приділена підготовці даних для моделювання. На етапі формування датасетів виконано очищення даних, нормалізацію та їх розбиття на навчальну і тестову вибірки. Це забезпечило якісну базу для побудови та оцінки моделей.

У практичній частині було реалізовано побудову моделей для прогнозування термінів виконання ІТ-проєктів. Спочатку було створено три моделі дерев рішень. Перше дерево показало базові результати, після чого проведено оптимізацію

параметрів для покращення точності. Третя модель із автооптимізацією дозволила автоматично підібрати параметри, однак її точність виявилася нижчою порівняно з моделями, що були налаштовані вручну.

Далі було побудовано три моделі лісів рішень (Random Forest). Вони забезпечили стабільно високі результати, а збільшення кількості дерев дозволило знизити рівень помилок прогнозування. Це підтвердило ефективність ансамблевих методів для задач прогнозування.

Найкращі результати були досягнуті за допомогою Boosted Trees. Перша модель підсилених дерев, створена вручну, показала високу точність. Однак покращена модель, в якій було оптимізовано параметри, досягла R-squared = 0.99, що свідчить про майже ідеальну відповідність прогнозів реальним даним. Це підтвердило, що Boosted Trees є найефективнішим підходом серед усіх розглянутих моделей.

На завершення було проведено порівняльний аналіз усіх моделей, що дозволило зробити висновки про їх ефективність та області застосування.

Дослідження продемонструвало, що машинне навчання є потужним інструментом для прогнозування термінів виконання IT-проєктів. Зокрема:

Boosted Trees показали найвищу точність, що дозволяє рекомендувати їх для застосування у проектах з високими вимогами до прогнозування.

Random Forest забезпечили стабільні результати, що робить їх зручними для задач, де важлива збалансованість між точністю та швидкістю обчислень.

Дерева рішень є простим і ефективним підходом для початкового аналізу, однак поступаються ансамблевим методам у точності.

Подальші дослідження можуть бути спрямовані на автоматизацію процесу налаштування параметрів моделей та інтеграцію з сучасними інструментами управління проектами. Запропонований підхід має великий потенціал для розвитку і може бути використаний як у невеликих командах, так і у великих організаціях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Standish Group Report, 2015.
URL: https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf
2. “The Cost of IT Project Delays: Analysis of Financial Losses,” Journal of IT Management, 2021.
3. A Three-Point Estimating Technique: PERT
URL: https://projectmanagementacademy.net/resources/blog/a-three-point-estimating-technique-pert/
4. Critical path method
URL: https://en.wikipedia.org/wiki/Critical_path_method
5. Critical path method: How to use CPM for project management
URL: https://asana.com/resources/critical-path-method
6. Machine Learning, ML
URL: https://www.it.ua/knowledge-base/technology-innovation/machine-learning
7. Що таке машинне навчання: як працює та де використовується
URL: https://gigacloud.ua/blog/navchannja/scho-take-mashinne-navchannja-jak-pracjue-ta-de-vikoristovuetsja
8. Kanwal Mehreen. 10 Machine Learning Algorithms Explained Using Real-World Analogies
URL: https://machinelearningmastery.com/10-machine-learning-algorithms-explained-using-real-world-analogies/
9. В.М. Терещенко, А.Д. Бугайов. АЛГОРИТМИ МАШИННОГО НАВЧАННЯ У КОНТЕКСТІ ВЕЛИКИХ ДАНИХ. Штучний інтелект, 2018, №3
URL: http://dspace.nbuv.gov.ua/bitstream/handle/123456789/162446/09-Tereshchenko.pdf?sequence=1
10. Dennis Kayser. ML in Project Management: Classification & Regression Problems with Text
URL: https://www.forecast.app/blog/machine-learning-in-project-management

11. Jiwat Ram. Why should project management professionals know about Machine Learning?
URL: https://ipma.world/why-should-project-management-professionals-know-about-machine-learning/
12. Ash Aujla. How machine learning will change the world of project management
URL: https://birdviewpsa.com/blog/how-machine-learning-will-change-the-world-of-project-management/
13. What is a decision tree?
URL: https://www.ibm.com/topics/decision-trees
14. What is random forest?
URL: https://www.ibm.com/topics/random-forest
15. Introduction to Boosted Trees
URL: https://xgboost.readthedocs.io/en/stable/tutorials/model.html
16. Machine Learning made beautifully simple for everyone
URL: https://bigml.com/features
17. Машинне навчання
URL: https://colab.research.google.com
18. Altair® RapidMiner®
URL: https://altair.com/altair-rapidminer
19. H2O Driverless AI
URL: https://h2o.ai/platform/ai-cloud/make/h2o-driverless-ai/
20. Teachable Machine by Google
URL: https://teachablemachine.withgoogle.com/
21. Statistics and Machine Learning with KNIME
URL: https://www.knime.com/key-capabilities/statistics-machine-learning
22. Orange Data Mining Documentation
URL: https://orangedatamining.com/docs/
23. IBM Watson Studio
URL: https://www.ibm.com/products/watson-studio

24. Azure Machine Learning

URL: https://azure.microsoft.com/en-us/products/machine-learning

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

ДИПЛОМНА РОБОТА
на ступінь вищої освіти магістр
із спеціальності 122 Комп'ютерні науки

Алгоритм прогнозування успішності і термінів виконання IT проєктів на основі застосування машинного навчання

Виконав: студент 6 курсу, групи КНДМ-63
Сундучков Дмитро
Керівник: доцент кафедри Комп'ютерних наук
Петро Кравчук

Київ - 2024

ЗАГАЛЬНА ХАРАКТЕРИСТИКА ДИПЛОМНОЇ РОБОТИ

Тема	Алгоритм прогнозування успішності і термінів виконання IT проєктів на основі застосування машинного навчання
Мета дослідження	прогнозування термінів виконання IT-проєктів за допомогою моделей машинного навчання для підвищення точності оцінки та оптимізації ресурсів
Наукове завдання	практичне використання моделей машинного навчання для прогнозування термінів виконання задач у сфері IT-проєктів
Об'єкт дослідження	процес планування та виконання IT-проєктів, що включає етапи оцінки часу на розробку та виконання завдань
Предмет дослідження	застосування методів машинного навчання для прогнозування термінів виконання задач у сфері IT-проєктів, зокрема використання дерев рішень, лісів рішень та підсилених дерев

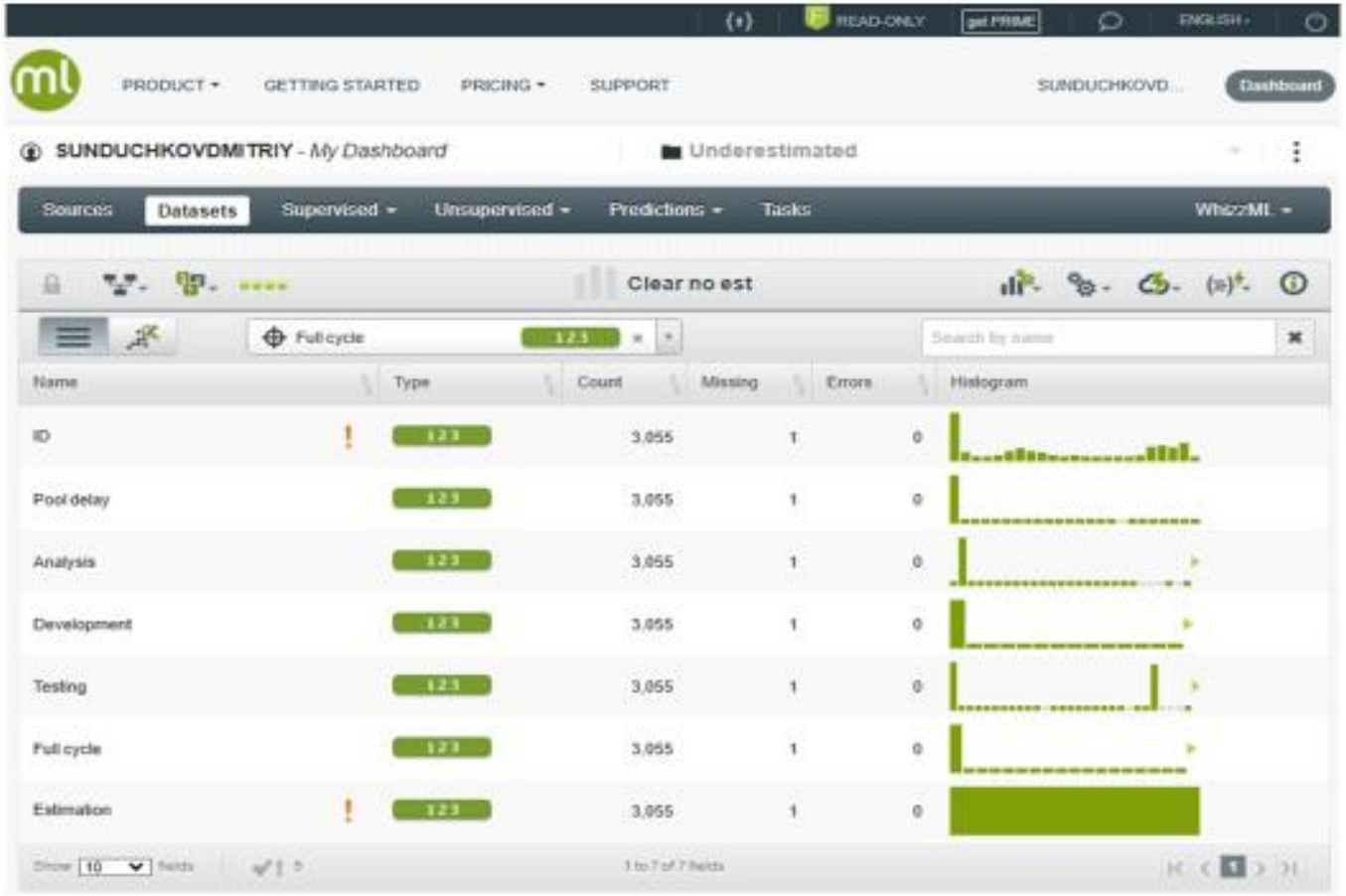
Порівняльна таблиця класичних і сучасних методів прогнозування ІТ проєктів

Критерій	Класичні методи (PERT, CPM)	Сучасні методи (машинне навчання)
Точність прогнозів	Залежить від якості вхідних даних і суб'єктивних оцінок.	Висока точність за рахунок автоматичного виявлення закономірностей у даних.
Гнучкість	Складно адаптуються до змін у проектах чи даних.	Легко адаптуються до нових даних і параметрів.
Залежність від експерта	Потребують досвіду та інтуїції менеджера для побудови моделей.	Автоматизовані; експертна думка впливає лише на підготовку даних.
Масштабованість	Обмеженість при роботі з великими обсягами даних.	Легко працюють із великими даними завдяки сучасним обчислювальним ресурсам.
Час на обробку даних	Потребують значного часу на підготовку та обчислення вручну.	Автоматизовані процеси знижують час підготовки й обробки даних.
Складність впровадження	Простота у використанні, оскільки не вимагають складного програмного забезпечення.	Складніші у впровадженні через потребу в налаштуванні моделей і систем.

Порівняльна таблиця методів (моделей) машинного навчання

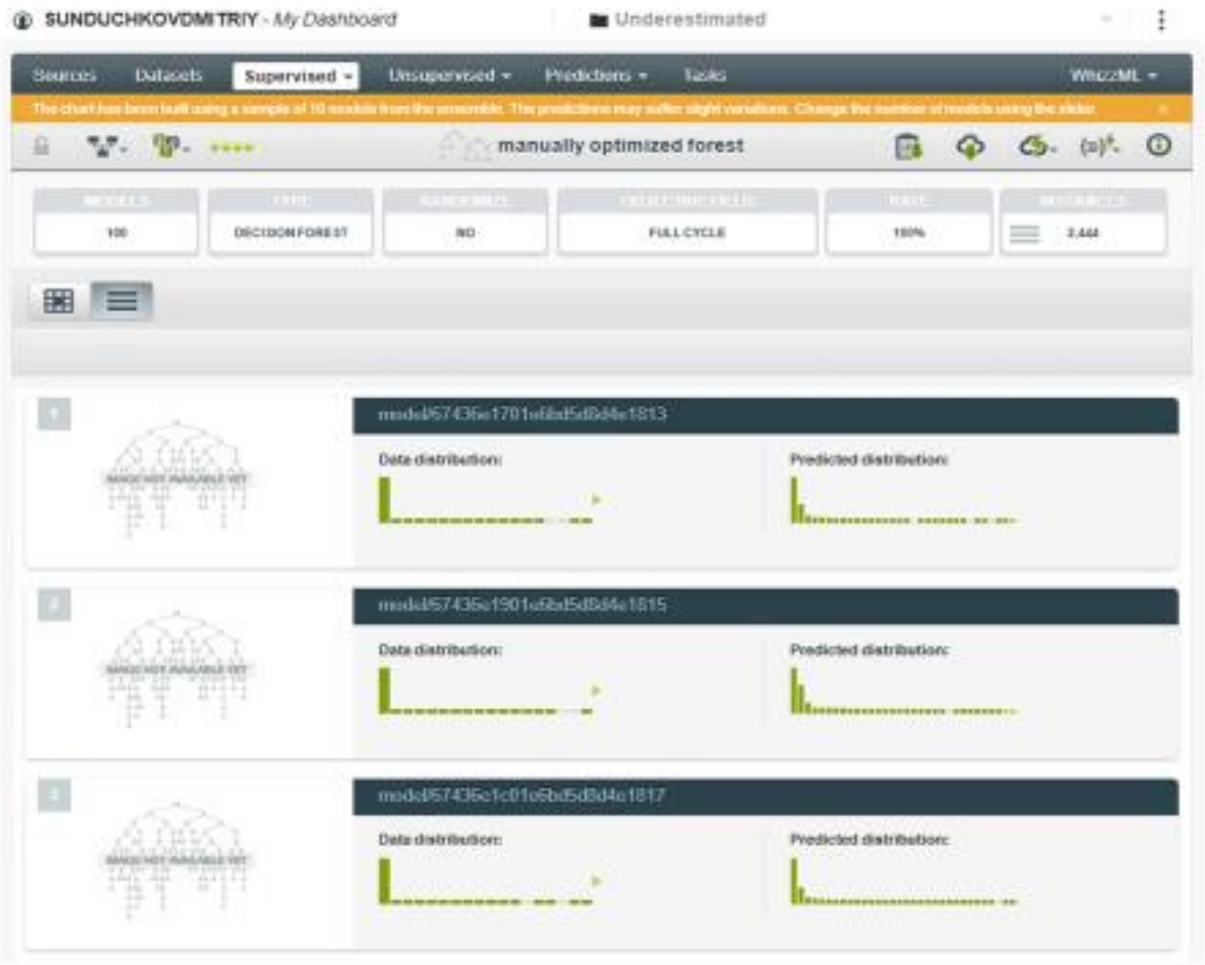
Параметр	Дерево рішень	Ліс рішень (Random Forest)	Підсилені дерева (Boosted Trees)
Складність моделі	Низька, проста у побудові	Середня, ансамбль кількох дерев	Висока, кожне дерево коригує помилки попередніх
Простота інтерпретації	Висока, легко зрозуміти структуру рішення	Середня, кожне дерево складно розібрати окремо	Низька, складно пояснити внесок кожного дерева
Стійкість до переобучення	Низька, схильна до переобучення	Висока, завдяки беггінгу	Середня, можливе переобучення без налаштування
Точність прогнозів	Середня	Висока, завдяки зниженню варіативності	Висока, за рахунок поступового вдосконалення
Чутливість до шуму	Висока, може реагувати на аномалії	Низька, шум усереднюється між деревами	Середня, модель може надмірно підлаштовуватися
Швидкість навчання	Висока, модель будується швидко	Середня, потребує навчання кількох дерев	Низька, кожне дерево додається поступово
Ефективність на великих даних	Обмежена, через обмеження одного дерева	Висока, завдяки паралельному навчанню	Середня, через поступове навчання дерев
Основна перевага	Простота та інтерпретація	Надійність і точність	Точність навіть на складних наборах даних
Основний недолік	Схильність до переобучення	Відносна складність і ресурсозатратність	Складність налаштування параметрів

Структура датасету

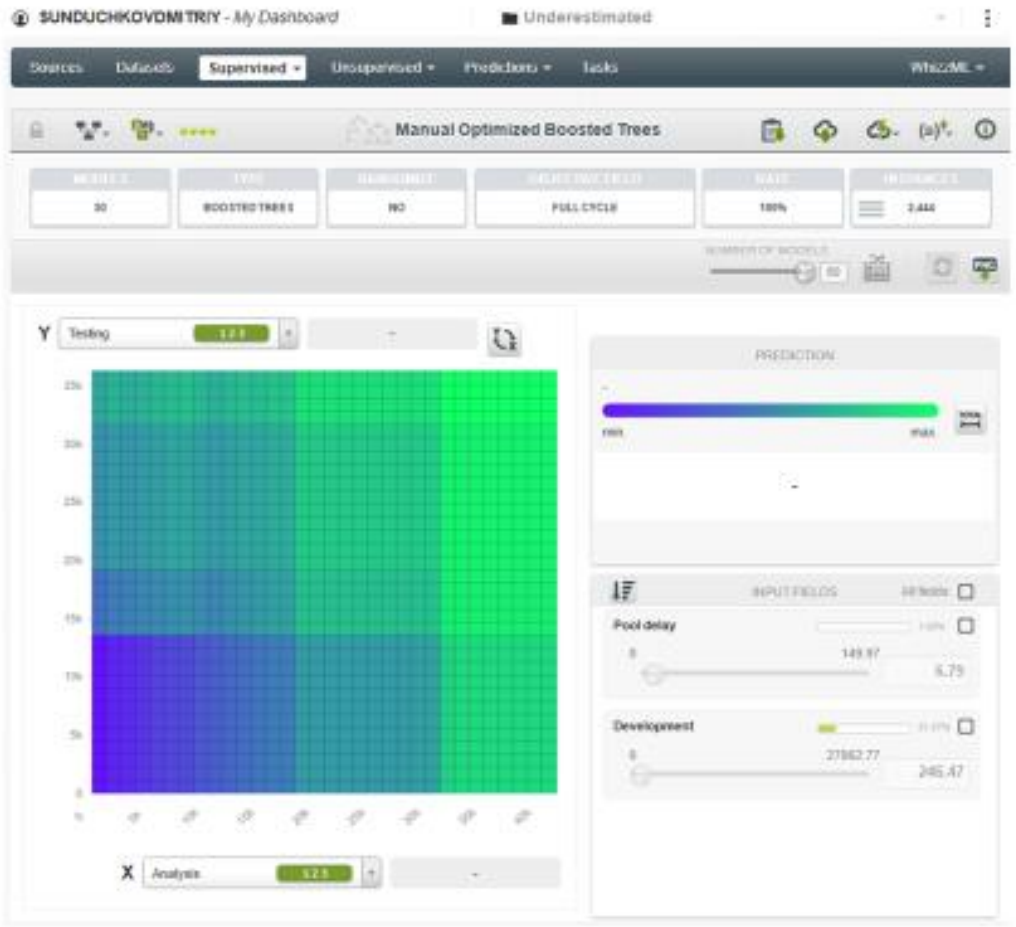


Структура моделі типу дерево рішень

Структура моделі типу ліс рішень



Структура моделі типу підсилені дерева



Таблиця аналізу результатів

Модель	R-squared	Mean Absolute Error (MAE)	Mean Squared Error (MSE)
Перше дерево рішень	0.95	85.36	169,944.67
Друге дерево рішень	0.94	113.90	172,101.29
Третє дерево рішень	0.94	123.02	289,484.41
Перший ліс рішень (Random Forest, 10 дерев)	0.97	53.60	90,444.80
Другий ліс рішень (Random Forest, 100 дерев)	0.97	75.08	92,279.90
Третій ліс рішень (Random Forest, 256 дерев)	0.97	51.42	79,582.78
Перша модель Boosted Trees	0.98	41.77	57,902.41
Автооптимізовані Boosted Trees	0.95	70.54	171,018.72
Покращена модель Boosted Trees (вручну налаштована)	0.99	54.28	33,484.17

Висновки роботи

1. Досліджено класичні методи прогнозування термінів виконання IT-проєктів (PERT, CPM) та виявлено їх обмеження в умовах сучасних проєктів.

2. Аналізовано сучасні методи машинного навчання, зокрема дерева рішень, Random Forest та Boosted Trees, для прогнозування термінів виконання задач.

3. Розроблено та оптимізовано моделі машинного навчання для прогнозування термінів виконання задач у IT-проєктах на основі реальних даних.

4. Проведено порівняння точності прогнозів між різними моделями та виявлено переваги Random Forest та Boosted Trees.

5. Інтегровано результати моделей у робочі процеси управління IT-проєктами для підвищення точності планування та оптимізації ресурсів.