

<https://doi.org/10.31891/2307-5732-2026-361-10>

УДК 004.8

**ВЕРБОВСЬКИЙ ОЛЕКСАНДР**

Державний університет «Житомирська політехніка»

<https://orcid.org/0009-0006-2912-4092>

e-mail: [ipzm241\\_voyu@student.ztu.edu.ua](mailto:ipzm241_voyu@student.ztu.edu.ua)

**ЛОКТИКОВА ТАМАРА**

Державний університет «Житомирська політехніка»

<https://orcid.org/0000-0002-3525-0179>

e-mail: [dfikt\\_ltn@ztu.edu.ua](mailto:dfikt_ltn@ztu.edu.ua)

**КУШНІР НАДІЯ**

Державний університет «Житомирська політехніка»

<https://orcid.org/0000-0002-0797-3687>

e-mail: [kipz\\_kno@ztu.edu.ua](mailto:kipz_kno@ztu.edu.ua)

**ЛИСОГОР ЮРІЙ**

Державний університет «Житомирська політехніка»

<https://orcid.org/0000-0003-1194-2813>

e-mail: [lysogor@ztu.edu.ua](mailto:lysogor@ztu.edu.ua)

## ДОСЛІДЖЕННЯ ПРИНЦИПІВ ПОБУДОВИ ТА ПРОЄКТУВАННЯ ІНТЕРАКТИВНОЇ ПЛАТФОРМИ З ВИКОРИСТАННЯМ МЕТОДІВ ТА АЛГОРИТМІВ ПРОГНОЗУВАННЯ ДАНИХ

Автоматизація процесів є одним з основних напрямів у розвитку сучасних технологій. Вона дозволяє значно підвищити ефективність і знизити ризики, забезпечуючи швидкість і якість виконання завдань різного роду. У статті пропонується дослідження розробки платформи оренди різноманітних речей з механізмом прогнозування ймовірності виникнення суперечок. Платформа побудована з використанням мультисерверної архітектури та чітким використанням певних технологій для вирішення конкретних задач. Особливу увагу приділено огляду й аналізу можливих моделей прогнозування ймовірності виникнення суперечок. Було розглянуто кілька методів машинного навчання, а саме – логістичної регресії у поєднанні з попереднім перетворенням тексту в числові вектори та алгоритм випадкового лісу, оскільки цей метод може працювати з різними наборами числових даних. Також було проаналізовано можливість використання нейронних мереж, адже вони здатні виявляти складні нелінійні залежності в даних, що є корисним у такому проєкті. У результаті було прийнято рішення використовувати алгоритм випадкового лісу як основний для тренування моделі прогнозування даних, оскільки він не потребує великої кількості даних та значних ресурсів порівняно з нейронними мережами. До цього, проєкт має функціонал, який дозволяє адміністраторам створювати кастомні моделі на основі обраних полів. На додачу, щоби при тренуванні не отримати неочікуваних результатів, платформа включає в себе механізм рекомендацій. Він використовує  $\chi^2$  тест для виявлення залежностей між категоріальними змінними та матрицю кореляції для визначення лінійних зв'язків між числовими даними, що і дозволяє відбирати найбільш релевантні поля. Також у дослідженні наведено структуру бази даних проєкту та описано використання полів ключових таблиць для покращення точності тренувань. У результаті було створено платформу, яка спрощує визначення кращого опонента для початку замовлення.

**Ключові слова:** автоматизація, алгоритм випадкового лісу, веб-платформа, оренда, суперечки.

**VERBOVSKYI OLEKSANDR, LOKTIKOVA TAMARA, KUSHNIR NADIA, LYSOGOR IURII**

Zhytomyr Polytechnic State University

## RESEARCH ON THE PRINCIPLES OF BUILDING AND DESIGNING AN INTERACTIVE PLATFORM USING DATA PREDICTION METHODS AND ALGORITHMS

Automation of processes is one of the main directions in the development of modern technologies. It significantly increases efficiency and reduces risks by ensuring the speed and quality of task execution of various types. This article presents a study on the development of a rental platform for various items that includes a mechanism for predicting the probability of disputes. The platform is built using a multi-server architecture with a clear application of specific technologies to solve particular tasks. Special attention is given to the review and analysis of possible models for predicting the probability of disputes. Several machine learning methods were considered, namely logistic regression combined with prior text-to-vector transformation, and the random forest algorithm, as this method can work with diverse sets of numerical data. The potential use of neural networks was also analyzed, as they are capable of detecting complex nonlinear dependencies in data, which is beneficial for this type of project. As a result, it was decided to use the random forest algorithm as the main model for training data prediction, since it does not require large amounts of data or significant computational resources compared to neural networks. In addition, the project includes functionality that allows administrators to create custom models based on selected fields. Moreover, to avoid unexpected results during training, the platform includes a recommendation mechanism. It uses the  $\chi^2$  test to identify dependencies between categorical variables and a correlation matrix to determine linear relationships between numerical data, allowing the selection of the most relevant fields. The study also presents the structure of the project's database and describes the use of key table fields to improve training accuracy. As a result, a platform was created that simplifies the identification of the most suitable opponent for initiating an order.

**Keywords:** automation, disputes, random forest algorithm, rental, web platform.

Стаття надійшла до редакції / Received 22.12.2025

Прийнята до друку / Accepted 11.01.2026

Опубліковано / Published 29.01.2026



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Вербовський Олександр, Локтікова Тамара, Кушнір Надія, Лисогор Юрій

### Постановка проблеми

За своєю природою люди завжди хочуть отримувати максимальний можливий зиск за мінімальних витрат. Доказом цього є періоди Чорної п'ятниці, під час яких продажі як офлайн, так і онлайн магазинів зазвичай зростають на 30-40%. Однак, деякі покупки, навіть зі знижкою, все одно можуть бути для людей занадто вартісними та недоцільними. Гарними прикладами можуть слугувати придбання будівельних інструментів, які потрібні для побудови власного будинку, туристичного знаряддя, яке зазвичай використовується лише для приємного проведення часу в період відпустки та навіть – тематичних костюмів, які одягне дитина в садочок на свято.

Рішенням цих дилем стали сервіси для оренди, на яких за незначну ціну будь-хто може стати власником як ексклюзивної книги, так і топового спортивного автомобіля на необхідний період. Відносини такого виду поміж клієнтом та постачальником є надзвичайно вигідними для обох, адже користувач не платить повну ціну за предмет і не думає потім, як від нього найефективніше позбутися, а власник якісного інструменту може використовувати його для власних потреб, коли забажає, а коли той йому не потрібен – отримувати пасивний прибуток.

Однак, як і в будь-яких інших торговельних механізмах, у період оренди досить часто виникають суперечки, які не лише потребують часу опонентів, а й дуже втомлюють їх, що призводить до страху знову потрапити у подібну ситуацію, і є причиною відмови від таких застосунків. Уже існують механізми для зниження ризиків виникнення таких ситуацій, наприклад, рейтингова система та коментарі, однак, вони вимагають від користувачів часу на збір та аналіз інформації, крім того, визначення ймовірності ризику теж повністю здійснюється користувачами, що все одно може призвести до неправильної оцінки ситуації та прийняття ненайкращого рішення, особливо у випадках, коли відгуки про опонентів є штучно створеними або взагалі відсутніми.

Тому є необхідність у розробці механізму для автоматизованої оцінки ризику виникнення суперечок між власником та орендарем, адже це не лише дозволить покращити досвід користувачів таких платформ, а й зменшити навантаження їхніх адміністраторів, до обов'язків яких належить вирішення критичних ситуацій, що у свою чергу дозволить працівникам проявляти більше творчості і покращувати платформи для залучення ще більшої кількості клієнтів.

### Аналіз останніх досліджень

Під час розробки платформи було здійснено детальний аналіз та порівняння з існуючими аналогами, які мають схоже призначення, функціонал та орієнтовані на ту саму цільову аудиторію. Однак, через відсутність реалізації прогнозування ймовірності завершення замовлень без інцидентів на більшості з цих платформ, було вирішено розширити аналіз і включити до нього також сервіси іншої тематики, які містять подібні механізми.

До першого переліку належать застосунки Yoodlize та Pickle. На основі них було визначено основний функціонал розробленої платформи та його дизайн, включно з основними сторінками, основними процесами платформи та передбаченням зручності застосунку. До другого списку належать PandaDoc, Salesforce Einstein та Gong. Ці платформи є прикладами важливості інтеграції алгоритму визначення ймовірності успішності угод.

Першою розглянутою платформою є Yoodlize [1]. Це американська платформа, призначена для оренди предметів широкого спектра – від важкої техніки, спрямованої на виконання будівельних робіт, до настільних ігор. Завдяки ній клієнти можуть як користуватися речами без витрачання повної суми на їх купівлю, так і заробляти на предметах, які просто припадають пилом десь у шафі чи в коморі.

Інтерфейс цієї платформи орієнтований на користувачів будь-якого віку, а процес створення оголошення або бронювання займає всього лише кілька хвилин. Комунікація між сторонами угоди відбувається через вбудований чат, у якому можна обговорити деталі, час та місце передачі речей.

Оплата є дуже важливим етапом будь-якої подібної платформи. Дана платформа не є винятком і теж має цей механізм. Усе відбувається безпосередньо через платформу. Кошти резервуються на період оренди й потрапляють до власника лише після закриття офери. Якщо ж виникають якісь проблеми, то платформа може заморозити виплату до моменту розв'язання конфліктної ситуації. Також Yoodlize забезпечує страхове покриття орендованого предмета, що додатково захищає власника у випадку пошкодження або втрати речі.

Після повернення предмета власнику та повного завершення замовлення, обидві сторони можуть залишити відгук і оцінку, на основі чого і формується репутація кожного учасника спільноти. Завдяки цьому платформа заохочує добросовісну поведінку, а система рейтингів допомагає новим користувачам зорієнтуватися у виборі орендаря або орендодавця. Однак, як і було згадано раніше, цей механізм змушує користувачів самотужки аналізувати оцінки, відгуки та загальну інформацію про опонента, що досить часто призводить до складних суперечок, які псують позитивне враження користувачів від платформи.

Наступною платформою стала ще одна американська платформа для оренди одягу – Pickle [2]. Ця платформа орієнтована здебільшого на модний одяг та аксесуари. На ній користувачі можуть орендувати топові речі, які коштують сотні доларів, буквально за десять-двадцять відсотків від їхньої ціни при купівлі.

Візуальна частина застосунку вирізняється сучасним мінімалістичним дизайном, за допомогою якого орендарі можуть у декілька кліків створити нове оголошення для здачі в оренду речі, а замовники, так само за мінімальну кількість кроків, – знайти її, переглянути інформацію та зробити запит на оренду. Дана платформа, як і Yoodlize, також має вбудований чат, який можна активувати відразу при перегляді оголошення, дозволяючи миттєво обговорити деталі та задати питання власнику.

Оплата на Pickle відбувається за допомогою стороннього сервісу Stripe. Транзакції є максимально прозорими для обох користувачів, а кошти також списуються, щойно розпочинається оренда, і є замороженими

на рахунках платформи до завершення замовлення. Якщо ж виникають якісь проблеми, то адміністратори підключаються до обговорення і здійснюють фінансові операції вручну.

По завершенню оренди, користувачі отримують змогу залишити оцінки, на основі яких і будуються враження інших користувачів щодо опонентів, однак, як і у випадку з Yoodlize, суперечок на платформі виникає доволі багато. Підхід платформи до вирішення суперечок завдяки ручному втручанню команди підтримки, хоча й дозволяє враховувати контекст, часто виявляється повільним та непрозорим для обох сторін, що суттєво знижує довіру до платформи у складних ситуаціях.

Ще одним недоліком, з яким стикаються користувачі у випадках пошкодження речей, є доволі низька компенсація, що досить часто не покриває і половини ціни. Таким чином, власники менше би розчарувались задачею власних речей в оренду та рідше втрачали інтерес до платформи, якби система мала механізм прогнозування ризику і попереджала їх щодо можливих проблем ще до початку оренди.

Далі було розглянуто платформу для автоматизації документообігу – PandaDoc [3]. Ця платформа допомагає компаніям створювати, надсилати та підписувати різні типи документів, наприклад, комерційні пропозиції, угоди та контракти. Платформа спрощує робочий процес, надаючи інструменти для створення інтерактивних документів із шаблонів, можливості спільної роботи та відстеження активності в реальному часі.

На ній наявна можливість прогнозування успіху. Платформа аналізує дані про взаємодію з документами, наприклад, час перегляду, частоту відкриття та дії, які здійснюються одержувачем. Використовуючи ці дані, платформа надає аналітику щодо ймовірності укладення угоди, допомагаючи сейлз-менеджерам приймати обґрунтовані рішення. Це дозволяє компаніям розуміти поведінку потенційних клієнтів та оптимізувати свої пропозиції для підвищення прибутків. Оцінка ймовірності відображається у вигляді звітів і дашбордів.

Однак, API розглядуваної платформи не можливо інтегрувати у платформу оренди речей, оскільки її функціонал сфокусований на прогнозуванні успішності угод саме на основі взаємодії з документами, а не на аналізі різних факторів, пов'язаних саме з користувачами, їхніми діями та даними про попередні оренди.

Наступним аналогом було розглянуто рішення, що використовує платформа CRM Salesforce [4], а саме – Salesforce Einstein. Воно розроблене на основі штучного інтелекту та надає можливість аналізувати поведінку клієнтів і прогнозувати результати угод. Платформа використовує дані історії, а саме електронні листи, дзвінки, активність у соціальних мережах і демографічні показники, щоби створювати рекомендації для продажів та маркетингу. Це допомагає командам ефективніше залучати клієнтів, підвищувати продуктивність і прогнозувати майбутні доходи.

Ймовірність успіху кожної угоди у вигляді відсотка формується на основі машинного навчання та аналізу великих обсягів даних. Функціонал включає прогнозування обсягів продажів, автоматичне визначення пріоритетних лідів та можливостей, а також виявлення ризиків у процесі укладення угод. Компанії можуть використовувати ці прогнози для зменшення ризиків і вдосконалення стратегії залучення клієнтів.

Однак, не зважаючи на потужність цього інструменту для аналізу поведінки клієнтів і прогнозування успіху угод у сфері продажів, він не підходить для платформи оренди речей, оскільки орієнтований на бізнес-процеси та взаємодію з потенційними клієнтами, а не на оцінку ризиків, пов'язаних саме із орендою, для якої важливо враховувати специфіку передачі, стану речей, поведінку користувачів у контексті використання предметів, а також автоматизацію вирішення суперечок, що суттєво виходить за межі функцій проаналізованої системи.

Ще одним проаналізованим застосунком стала платформа для управління продажами Gong [5]. Вона фіксує та аналізує взаємодію команд із клієнтами через дзвінки, електронну пошту та відеозустрічі. Платформа допомагає компаніям зрозуміти, які аспекти комунікації найбільш ефективні для закриття угод, визначаючи найкращі стратегії взаємодії з клієнтами. Gong використовує штучний інтелект для розпізнавання мовлення та аналізу тональності, що дозволяє визначити настрій і зацікавленість клієнта.

Щодо прогнозування успіху, платформа аналізує розмови та дії клієнтів, виявляючи патерни, що корелюються із успішними продажами. Система надає детальні звіти про те, які фрази чи теми сприяють укладенню угод, а також підкреслює потенційні ризики у спілкуванні. Це дозволяє відділам продажів вчасно вносити корективи у свої стратегії та підвищувати ймовірність успіху. Безпосередньо ймовірність успішності угоди відображається на спеціальних дашбордах, прив'язаних до замовлень.

Однак, ця платформа теж не підходить для прогнозування ризиків виникнення суперечок при оренді, через свою вузьку спеціалізацію на продажах і взаємодії з клієнтами в бізнес-контексті, адже вона не має функціоналу для враховування характеристик користувачів, предметів, місць передачі та повернення речей.

Таким чином, аналіз цих платформ дозволив визначити ключовий функціонал розроблюваної платформи для оренди одягу: пошук та вибір замовлення, подання оферу, створення замовлення, прив'язування PayPal-акаунта до профілю користувача для отримання коштів, наявність головної сторінки з кількома секціями для пошуку за категоріями і не тільки, сторінки опису платформи, сторінки нових замовлень, внутрішнього чату та можливість взаємодії з замовленнями в ньому, а також можливість створення суперечки і залучення адміністратора до неї.

А також, у результаті проведеного аналізу було визначено основні напрями в галузі прогнозування даних. Серед них – текстовий аналіз, наприклад, аналіз текстової частини опису пропозиції, замовлення, його назви або листувань у внутрішньому чаті, а також аналіз параметрів, а саме кількості успішно виконаних замовлень виконавцем, кількості успішно оплачених завдань замовником, кількості коментарів у рейтингах користувачів, тривалості виконання замовлення та його ціни.

### Формулювання цілей

Основна ціль полягає у дослідженні розробки інноваційної платформи для оренди одягу та інтеграції методів прогнозування даних у неї, з метою підвищення безпеки взаємодії між користувачами, зниження ризиків виникнення суперечок та покращення досвіду користувачів.

### Викладення основного матеріалу

На початку розробки було здійснено вибір технологій для вирішення конкретних задач. На цьому етапі виявилось, що найкращим рішенням для поточної розробки буде використання багатосерверної архітектури, оскільки система має складатися мінімум з трьох взаємопов'язаних компонентів і це дозволить в майбутньому легше інтегрувати нові розробки та унікальні сервіси без прямої залежності від ядра застосунку.

Таким чином, першим сервером є сервер-ядро, призначений для взаємодії з базою даних, опрацювання та перевірки оплат, а також роботи зі сторонніми API, які потребують саме бекенд-інтеграції через різні обмеження безпеки, наприклад, API для роботи з координатами на реальних картах [6]. Для розробки цього компонента платформи застосована Node.js, оскільки це сучасна платформа, що досить добре зарекомендувала себе у веб-розробці і має велику кількість цікавих інструментів [7]. Крім того, Node.js використовує асинхронність та має можливість виконання скриптів за допомогою stop-завдань, що дозволило розробити логіку фінансових виплат виконавцям. Для виконання таких операцій використано платформу PayPal, адже вона дуже зручна для керування фінансовими операціями. Також платформа дозволяє інтегрувати скрипти для авторизації і таким чином прив'язувати гаманці до акаунта платформи.

Для збереження даних використано базу PostgreSQL [8, 9], адже в порівнянні з MySQL дана СУБД швидше працює з великою кількістю даних, що є корисним на етапі тренування моделі прогнозування ризиків на основі реальних даних. Крім того, PostgreSQL підтримує складні типи даних, а саме JSON, який дозволяє зберігати напівструктуровані дані з можливістю швидкого пошуку та індексації по них. Це особливо корисно для зберігання гнучких, змінних структур даних, які важко формалізувати у звичайних реляційних таблицях, і може бути використано для майбутнього підвищення гнучкості сервісу прогнозування ризиків і дозволить перевикористовувати його на інших проектах, незалежно від структури самої бази даних проекту.

Другим сервером є клієнтський, який відповідає за візуалізацію даних та зручну взаємодію з платформою. Для його розробки використано Next.js, адже цей фреймворк виконує збирання сторінок дуже швидко, завдяки можливості серверного рендерингу [10, 11]. Крім того, він виконує автоматичне розбиття коду на менші частини, що надає можливість завантажувати лише необхідний для зміни код. У свою чергу, це дозволяє зменшити обсяг завантажених даних, що також позитивно впливає на швидкість перерендерингу сторінок.

Третій сервер орієнтований на роботу з моделями машинного навчання, здійснює навчання моделей та формує прогнози щодо можливих ризиків виникнення суперечок під час оренди. Даний компонент платформи розроблений за допомогою високорівневої мови програмування Python [12], адже ця мова має можливості для реалізації високоякісних моделей прогнозування даних.

Однією з них є бібліотека sklearn, яка надає широкий набір інструментів для прогнозування даних. Також у мові Python розроблено бібліотеку pandas, завдяки якій можна ефективно обробляти та аналізувати великі обсяги даних. Це стало корисним доповненням до раніше згаданої sklearn, адже у парі ці бібліотеки можуть швидко та якісно натренувати модель, при цьому інструменти дозволяють обмежуватись невеликим обсягом коду, що також зменшує ймовірності помилок та витрати часу на тестування.

Ще однією перевагою мови Python є потужні інструменти для роботи з HTTP-протоколами. Завдяки таким бібліотекам як Flask і FastAPI можна легко створювати безпечні API, які забезпечують зручний доступ до нейронних мереж та інших моделей машинного навчання для сторонніх серверів. Також потрібно згадати бібліотеку psycopg2 у мові Python, адже це одна з найпопулярніших бібліотек для роботи з базою PostgreSQL, що стало дуже корисним на етапі збору даних для тренування моделі, оскільки витягування даних відбувалося безпосередньо з робочого простору, дозволяючи уникнути будь-яких помилок, пов'язаних з пошкодженням даних.

На рис. 1 зображено загальну схему взаємодії між серверами, а також зі сховищем, базою даних та зовнішніми API.

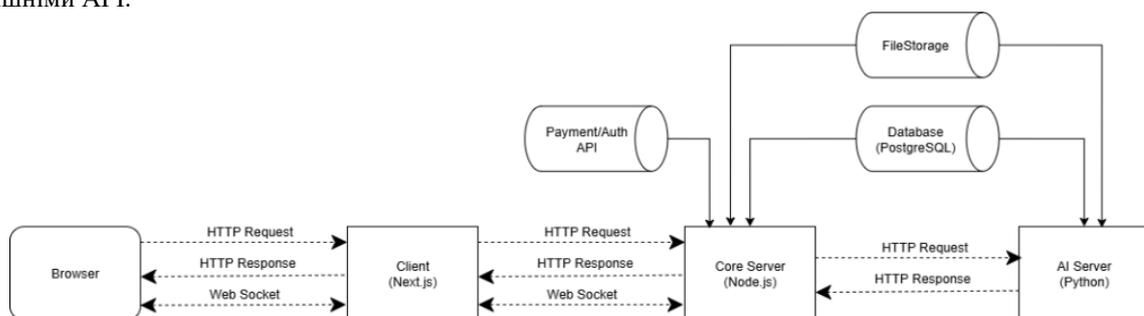


Рис. 1. Схема взаємодії компонентів платформи

Наступним етапом проектування платформи стало визначення найкращого методу для прогнозування виникнення суперечок. На сьогодні існує велика кількість методів прогнозування даних. Одні з них призначені

для виявлення залежностей між числовими величинами, інші – для класифікації, а деякі здатні аналізувати текст і знаходити закономірності навіть у ньому. Через це для реалізації механізму є багато різних підходів, методів та алгоритмів, тому важливим кроком є визначення найкращого методу для визначення ризику виникнення суперечки під час оренди. На основі аналізу існуючих застосунків-конкурентів було визначено три можливі підходи, а саме – аналіз людського мовлення, аналіз числових та категоріальних значень, а також використання нейронних мереж.

Першим розглянутим підходом стало прогнозування ймовірності виникнення суперечки на основі текстових даних за допомогою логістичної регресії [13]. Це дуже простий у реалізації метод, який найкраще підходить для задач, де потрібно спрогнозувати, чи відбудеться певна подія чи ні. Однак, логістична регресія не може працювати безпосередньо з текстовими даними, тому на етапі попередньої обробки текст необхідно перетворити у числовий вектор. Для цього використовуються спеціальні методи векторизації, наприклад, *bag-of-words*, що створює вектор, у якому кожна позиція відповідає певному слову зі словника, а значення – кількості його вживань у тексті.

Після підготовки векторизованих даних відбувається тренування із застосуванням логістичної регресії. Модель підбирає лінійну комбінацію ознак, яка дозволяє з максимальною точністю розрізнити випадки виникнення суперечок та успішного завершення замовлень. Під час навчання кожному слову або словосполученню присвоюється вага, яка впливає на загальну оцінку ймовірності. Для визначення ймовірності виникнення суперечки у розроблюваному проєкті можна було би тренувати моделі окремо для описів оголошень, їхніх назв, відгуків на замовника та власника, а також повідомлень з чатів, а потім отримати середнє значення прогнозованих даних.

Однією з переваг цього методу є те, що після тренування можна оцінити, які слова найсильніше впливають на прогноз і де саме. Наприклад, якщо в описі присутні конструкції по типу “пошкоджено”, “неповна комплектація” або “не відповідає опису”, то модель може автоматично підвищити ймовірність виникнення суперечки. Це дозволяє розробляти функціонал, який є доступним для преміум-користувачів і надаватиме підказки користувачам, коли вони будуть створювати оголошення або вести комунікацію з замовниками.

Однак, не зважаючи на всі переваги та цікаві можливі реалізації, даний метод не підходить для проєкту саме на ранньому етапі, оскільки для тренування подібних моделей потрібний досить великий обсяг даних, які мають бути збалансованими, очищеними та з чітким визначенням наявності суперечки. Крім того, обробка текстів за допомогою таких моделей потребує суттєвих обчислювальних ресурсів сервера, що також є недоцільним на перших етапах реалізації платформи.

Ще одним потужним і популярним підходом, який можна застосувати для розв’язання цієї задачі, є використання нейронних мереж. Метод глибокого навчання, який використовує багатoshарові штучні нейрони для навчання складних нелінійних функцій, дуже подібний до роботи людського мозку. Основна ідея нейронних мереж полягає в тому, що кожен нейрон у мережі отримує вхідні дані, обробляє їх за допомогою певної функції активації і передає результат наступному шару. Навчання зазвичай здійснюється за допомогою алгоритмів оптимізації, наприклад, градієнтного спуску, які дозволяють поступово коригувати ваги нейронів для зменшення помилки між передбаченнями мережі та фактичними результатами.

Основною перевагою є те, що моделі, побудовані за допомогою нейронних мереж, зазвичай є дуже гнучкими та ефективними для задач, які вимагають вивчення складних структур у великих наборах даних, наприклад, виявлення об’єктів на зображеннях, автоматичний переклад тексту, прогнозування на основі часових рядів та інші.

Проте цей метод, як і попередній, потребує великої кількості даних, що є проблемним при розробці невеликих проєктів, а також потребує великої кількості обчислювальних ресурсів. У зв’язку з цим доцільним стало зосередитись на простіших, менш ресурсоемних підходах, які дозволяють швидко отримати базові результати, водночас формуючи базу даних для подальшого навчання більш складних моделей.

Тому далі розглянемо дерева рішень – це один з найбільш поширених алгоритмів для класифікації та регресії. Для реалізації моделі будується загальне дерево, у якому формується послідовність дій залежно від даних на основі певної ознаки [14]. Листками в цих деревах є кінцеві прогнози. При кожному новому випадку будується нова гілка разом з листком, що робить такий метод дуже гнучким і здатним працювати з новими кейсами у майбутньому. Таку модель легко тестувати, адже достатньо отримати зображення дерева, щоби переглянути хід прогнозу.

Однак такі дерева також схильні до перенавчання. Дерево може стати занадто глибоким і підлаштовуватиметься під тренувальні дані, навіть якщо маленька зміна призведе до перебудови значної частини. У подальшому це може почати призводити до помилкових відповідей.

Але варто зауважити, що даний алгоритм може бути вдосконаленим. В ансамблевому методі випадкового лісу комбінується багато дерев рішень. Це призводить до отримання більш стабільного та точного прогнозу, адже кожне дерево створюється на основі випадково обраних підмножин даних. Фінальний результат можна отримати не лише у вигляді відповіді так/ні, 0 або 1, true/false, а й у відсотках, завдяки усередненню прогнозів усіх побудованих дерев [15].

Таким чином, на основі проведених огляду й аналізу різних методів і алгоритмів прогнозування даних, було прийнято рішення щодо реалізації прогнозів виникнення суперечок за допомогою методу випадкового лісу. Адже цей метод здатний забезпечувати високу точність прогнозів без використання великого обсягу даних та значних обчислювальних ресурсів.



(name), який здають в оренду, його опис (description), координати пункту (lat, lng), де користувач зможе забрати орендований предмет, відстань (radius), на яку власник може відійти від пункту видачі, щоби зменшити замовнику шлях, ідентифікатор власника предмету (owner\_id), ціна за день оренди (price) та ідентифікатор категорії (category\_id), до якої належить річ. Для прогнозування виникнення суперечок важливими є такі поля: категорія, ціна та ідентифікатор власника, оскільки, як згадувалось раніше, на основі ідентифікаторів можна отримати додаткову інформацію з інших таблиць, наприклад, про відгуки власника.

Далі розглянемо найважливішу таблицю проекту, а саме – замовлення (orders). У цій таблиці важливими є такі поля: звичайний статус (status), статус скасування замовлення (cancel\_status), податки користувачів (renter\_fee, owner\_fee), ідентифікатор замовника (renter\_id), ідентифікатор оголошення (listing\_id), ціна оренди за день (price), яку погодили користувачі, дати початку (start\_time) та кінця замовлень (finish\_time). Також на платформі є можливість коригування ціни та періоду замовлення. Для цього існує окрема таблиця запитів оновлення замовлень (order\_update\_requests). Тому таблиця замовлень містить ще інформацію про початково запропоновані терміни оренди замовником (prev\_start\_time, prev\_finish\_time) та початкову ціну оренди за день (prev\_price). Для моделі прогнозування важливими є ціна за день, тривалість замовлення та посилання на інструмент. Також варто згадати про таблицю запитів на оновлення замовлень. Потрібно також аналізувати, чи були внесені правки відносно початкової пропозиції, різницю між початково запропонованою ціною та узгодженою, датами та самою тривалістю, яка мала бути і на яку домовились. Також потрібно отримувати інформацію про те, чи закінчилося замовлення суперечкою, що отримується за допомогою приєднання таблиці суперечок (disputes).

Таблицями, на основі яких будуються рейтинги користувачів, є таблиці з коментарями власників (owner\_comments) та коментарями орендарів (renter\_comments). Вони дуже схожі загальною структурою, містять описи (description), ідентифікатори (id), особисті відгуки від опонента (leave\_feedback), підтвердження адміністраторів (approved), статус, що відображає потребу підтвердження адміністратором (waiting\_admin), описи причин відхилення коментарів (reject\_description), час створення (created\_at) та ідентифікатор замовлення, на яке залишили відгук (order\_id). З цих таблиць вираховуються середні значення оцінок по кожному з критеріїв, що впливає на оцінку ймовірності суперечок і є потрібними для тренування моделі.

Також на платформі є чати, саме для них і було створено таблиці зв'язків до чатів (chat\_relations), чатів (chats), повідомлень (chat\_messages) та контентів повідомлень (chat\_message\_contents). У подальшому, за більшої кількості даних на платформі, можливо буде використовувати дані з цих чатів для тренування нейронної мережі, яка замінить поточну логіку тренування моделі прогнозування виникнення суперечок.

Останньою розглядуваною таблицею буде архів моделей передбачень суперечок (dispute\_prediction\_models). За допомогою цієї таблиці адміністратори можуть перемикатися поміж моделями, які вони тренують зі своєї панелі. Такий підхід дозволяє експериментувати з різними полями, що передаються на сервер з прогнозуванням, і уникнути ситуацій, за яких користувачам відобразиться неправильна інформація, наприклад, якщо у новій моделі точність буде дуже низькою, адже можна буде перемкнутись на стару модель, або взагалі, при ініціалізації тренування, не вказувати, що нова модель буде використовуватись при оцінці ризиків.

Після створення бази даних, розробки моделі прогнозування та її навчання, було виконано інтеграцію відповідного модуля у загальну архітектуру платформи. Для цього було побудовано структуру API-ендпоінтів, які забезпечують передачу параметрів замовлення до сервера, де й визначається прогноз та зводиться до числового значення.

На рис. 4 зображено реалізацію відображення ймовірності виникнення суперечки до створення замовлення.

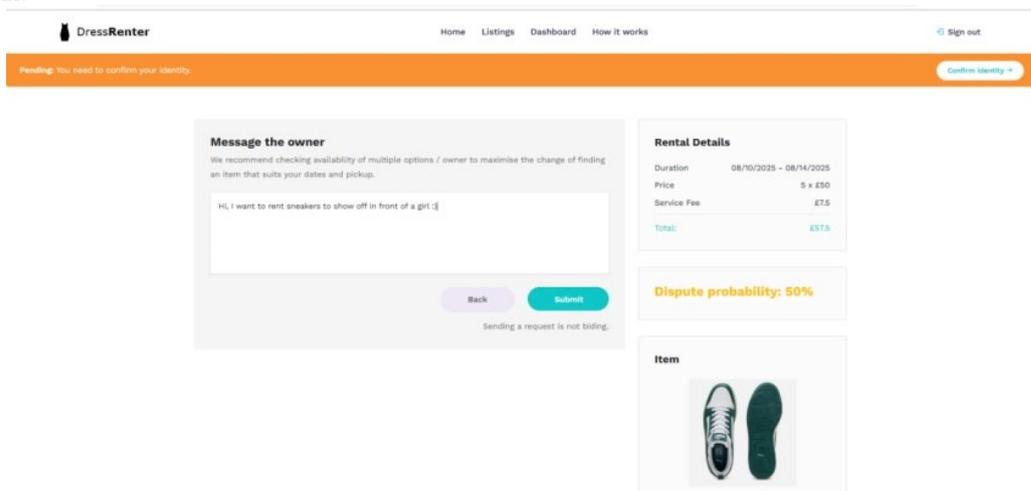


Рис. 4. Відображення ймовірності виникнення суперечки до створення замовлення

На рис. 5 зображено реалізацію відображення ймовірності виникнення суперечки на етапі підтвердження замовлення власником.

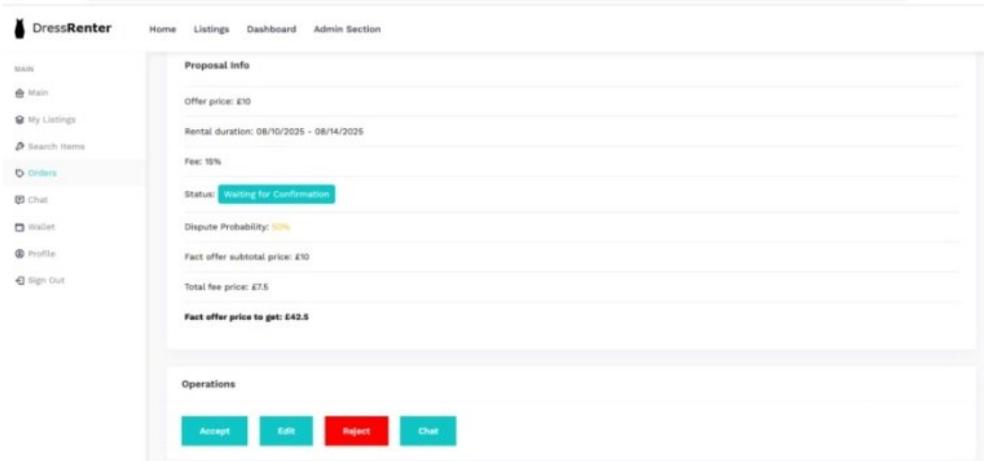


Рис. 5. Відображення ймовірності виникнення суперечки на етапі підтвердження замовлення власником

Щодо реалізації, то варто ще раз наголосити, що адміністратори мають змогу самостійно визначати, які поля та формули використовувати для тренування моделі, а щоби точно визначати, які поля варто використати для тренування моделі, сервер, призначений для тренування, має спеціальний API, який дозволяє визначити важливість кожного поля. Цей прогноз визначається за допомогою попереднього статистичного аналізу вхідних даних, а саме за допомогою аналізу кореляції для числових ознак, що дозволяє визначити ступінь лінійного зв'язку між кожним числовим полем та змінною, а також  $\chi^2$  тесту для визначення залежності категоріальних ознак, який дозволяє виявити статистично значущі залежності між окремими значеннями полів та фактом затримки.

Результати цього аналізу автоматично обробляються системою, і за встановленими пороговими значеннями система формує список рекомендованих ознак. Якщо таких ознак виявляється недостатньо, то система доповнює список найбільшими за ступенем залежності до досягнення мінімального вказаного обсягу для побудови моделі. Таким чином, рекомендаційна система покращує досвід адміністраторів і дозволяє тренувати моделі навіть без спеціальних знань та аналізів графіків.

Після проведення кількох тестових тренувань було виявлено, що точність моделей прогнозування виникнення суперечок значною мірою залежить від кількості та якості використаних вхідних параметрів. Найкращу точність значенням 84% продемонструвала модель, для тренування якої використано дані, запропоновані рекомендаційною системою. Ключовими стали такі характеристики: чи здає замовник речі, чи бере власник в оренду інші речі, а також важливою є категорія оголошення. Натомість моделі з більшою кількістю логічних ознак показували лише до 60% точності. Це свідчить про те, що для підвищення ефективності прогнозування доцільно орієнтуватися на рекомендаційну систему, яка враховує поведінкові та контекстні зв'язки, а не лише кількісні характеристики. На рис. 6 зображено результати тренування кількох тестових моделей.

ID	Status	Active	Fields	Accuracy	Training Percent	Date
#0	Finished	Active	listing_category_id, owner_is_renter, renter_is_owner	84%	100%	09/15/2025
#3	Finished	Not Active	renter_average_rating, owner_average_rating, price_per_day, duration_days, listing_category_id, count_owner_completed_orders, count_renter_finished_orders, count_order_update_requests, owner_is_renter, renter_is_owner	51%	100%	09/15/2025
#7	Finished	Not Active	renter_average_rating, owner_average_rating, listing_category_id, count_owner_completed_orders, count_renter_finished_orders	56%	100%	09/08/2025

Рис. 6. Порівняльні результати тренування тестових моделей прогнозування суперечок між користувачами

## Висновки

У дослідженні було розглянуто комплексну розробку інноваційної платформи для оренди одягу, зосередженої не лише на базовому функціоналі взаємодії між користувачами, а й на підвищенні їхнього досвіду за рахунок впровадження механізму прогнозування виникнення суперечок.

Було здійснено огляд й аналіз існуючих платформ у галузі оренди та окремо систем прогнозування, на основі яких було виокремлено найкращі практики та виявлено суттєві недоліки сучасних рішень.

Було вибрано з обґрунтуванням стек технологій, до якого ввійшли платформа Node.js для обробки логіки, фреймворк Next.js для розробки клієнтської частини, база PostgreSQL для зберігання даних і мова програмування Python для реалізації алгоритму машинного навчання. Така мультисерверна архітектура дозволяє ефективно масштабувати платформу, розширювати її функціонал та легко інтегрувати нові сервіси.

Окрему увагу було приділено вибору методу прогнозування виникнення суперечок. На основі порівняння різних підходів, методів і алгоритмів – від логістичної регресії до нейронних мереж – було визначено, що найкращим рішенням на початковому етапі розробки є використання методу випадкового лісу (Random Forest), адже він поєднує у собі високу точність, здатність працювати з невеликими вибірками даних та низьку чутливість до перенавчання. Крім того, завдяки цьому методу платформа дозволяє оцінити важливість кожного з полів у формуванні прогнозу, що дозволить у майбутньому покращити користувацький інтерфейс та розробити додатковий інтерактив.

Також було описано структуру бази даних, яка підтримує як основні функції платформи, призначені саме для роботи користувачів та адміністраторів між собою, так і додаткові, для контролю та тренування моделей для оцінки ймовірності виникнення суперечок.

У результаті було створено сучасну гнучку систему, яка не лише забезпечує ефективну оренду одягу, а й автоматично аналізує потенційні ризики виникнення суперечок. На подальше, платформа має потенціал для масштабування на інші сфери оренди та інтеграції з більш складними моделями, включаючи аналіз текстів, поведінку користувачів і систему рекомендацій.

## References

1. Yoodlize [Electronic resource]. – Resource access mode: <https://www.yoodlize.com>.
2. Pickle [Electronic resource] – Resource access mode: <https://www.shoponpickle.com>.
3. PandaDoc [Electronic resource] – Resource access mode: <https://www.pandadoc.com>.
4. CRM Salesforce [Electronic resource] – Resource access mode: <https://www.salesforce.com/crm>.
5. Gong [Electronic resource] – Resource access mode: <https://www.gong.io>.
6. Singh U. REST API Framework: Designing and Developing Web Services [Electronic resource] / U. Singh // International Research Journal of Engineering and Technology. – 2021. – Vol. 8. – pp. 815–817. – Resource access mode: <https://doi.org/10.13140/RG.2.2.24791.69288>.
7. Freeman A. Mastering Node.js Web Development: Go on a comprehensive journey from the fundamentals to advanced web development with Node.js / A. Freeman. – [N. p.]: Packt Publishing, 2024. – 778 p.
8. Zapata J. G. MySQL vs PostgreSQL: A Comparative Analysis of Relational Database Management Systems (RDBMS) Technologies Response Time in Web-based E-commerce [Electronic resource] / J. G. Zapata. – 2024. – Resource access mode: <https://doi.org/10.13140/RG.2.2.24791.69288>.
9. Vadlamani V. PostgreSQL Skills Development on Cloud [Electronic resource] / V. Vadlamani. – Berkeley, CA: Apress, 2024. – 646 p. – Resource access mode: [https://doi.org/10.1007/979-8-8688-0817-3\\_1](https://doi.org/10.1007/979-8-8688-0817-3_1).
10. Halili F. Web Services: A Comparison of Soap and Rest Services [Electronic resource] / F. Halili, E. Ramadani // Modern Applied Science. – 2018. – Vol. 12, № 3. – 175 p. – Resource access mode: <https://doi.org/10.5539/mas.v12n3p175>.
11. Patel V. Analyzing the Impact of Next.JS on Site Performance and SEO [Electronic resource] / V. Patel // International Journal of Computer Applications Technology and Research. – 2023. – Vol. 12, № 10. – pp. 24–27. – Resource access mode: <https://doi.org/10.7753/IJCATR1210.1004>.
12. Di Meglio S. Starting a New REST API project? A Performance Benchmark of Frameworks and Execution Environments / S. Di Meglio // CEUR Workshop Proceedings. – 2023. – Vol. 3543. – pp. 109–127.
13. Das A. Logistic regression [Electronic resource] / Abhik Das // Encyclopedia of the Sciences of Learning. – 2012. – C. 3680–3682. – Resource access mode: [https://doi.org/10.1007/978-94-007-0753-5\\_1689](https://doi.org/10.1007/978-94-007-0753-5_1689).
14. Yang F. An Extended Idea about Decision Trees [Electronic resource] / F. Yang // 2019 International Conference on Computational Science and Computational Intelligence (CSCI). – 2019. – pp. 349–354. – Resource access mode: <https://doi.org/10.1109/CSCI49370.2019.00068>.
15. Identification of Brain Stroke using Boosted Random Forest [Electronic resource] / V. Sapra // 2022 International Conference on Advances in Computing, Communication and Materials (ICACCM). – 2022. – pp. 1–5. – Resource access mode: <https://doi.org/10.1109/ICACCM56405.2022.10009527>.