

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук
(повна назва)

Кафедра _____ Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

Розробка вебзастосунку для оптимального обрахунку логістики із
урахуванням поточної завантаженості дорог на основі рандомізованих
ансамблевих алгоритмів
(тема)

Виконав:
здобувач _____ четвертого _____ року навчання,
групи _____ ІТШ-21-1

_____ Софія Янкович
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна

Освітня програма _____ Штучний інтелект

(повна назва освітньої програми)

Керівник _____ ст. викл. Ігор Бібічков
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____

(підпис)

_____ Олег ЗОЛОТУХІН

(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Янкович Софії Євгеніївні _____
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка вебзастосунку для оптимального обрахунку логістики із урахуванням поточної завантаженості дорог на основі рандомізованих ансамблевих алгоритмів

затверджена наказом університету від 19 травня 2025 р. № 378Ст

2. Термін подання студентом роботи до екзаменаційної комісії 25 червня 2025 р.

3. Вихідні дані до роботи Публічні маршрутизуючі API, синтетичні маршрути з характеристиками, реальний трафік із навантаженням у різні години доби, моделі прогнозування ETA, методики збору даних: геокоординати точок, часові мітки, параметри маршруту, сервіси побудови маршрутів, методи передобробки даних: нормалізація ознак, кодування категорій, балансування вибірки:, ансамблеві алгоритми машинного навчання

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі _____

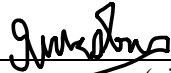
2) Проектування системи та обґрунтування вибору технологій _____

3) Програмна реалізація _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	19.05.2025	виконано
2	Аналіз предметної галузі	21.05.2025	виконано
3	Огляд методів обробки природньої мови	25.05.2025	виконано
4	Огляд моделей для виявлення кібербулінгу та їх метрик	25.05.2025	виконано
5	Програмна реалізація	25.05.2025	виконано
6	Написання пояснювальної записки	26.05.2025	виконано
7	Перевірка на академічний плагіат	29.05.2025	виконано
8	Нормоконтроль	31.05.2025	виконано
9	Підготовка презентації та доповіді	09.06.2025	виконано
10	Попередній захист	20.06.2025	виконано
11	Рецензування	22.06.2025	виконано
12	Захист перед ЕК	25.06.2025	

Дата видачі завдання 19 травня 2025 р.

Здобувач 
(підпис)

Керівник роботи _____
(підпис)

ст. викл. Ігор Бібічков
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 65 с., 22 рис., 3 табл., 2 дод., 20 джерел.

АНСАМБЛЕВІ АЛГОРИТМИ, ЛОГІСТИЧНА ОПТИМІЗАЦІЯ, ПРОГНОЗУВАННЯ ЧАСУ ДОСТАВКИ, РЕАЛЬНИЙ ТРАФІК, CATBOOST, ETA МОДЕЛІ, FLASK, FOLIUM, LIGHTGBM, OSRM API, XGBOOST.

Об'єктом дослідження є процес прогнозування часу доставки в умовах змінного завантаження доріг.

Предметом дослідження є алгоритмічна реалізація та застосування методів рандомізованих ансамблевих алгоритмів градієнтного бустингу (XGBoost, LightGBM, CatBoost) для прогнозування часу доставки (ETA) з урахуванням змінного завантаження доріг і характеристик маршруту.

Метою роботи є розробка автономного вебзастосунку для розрахунку ETA (Estimated Time of Arrival) з урахуванням характеристик маршруту, з використанням рандомізованих ансамблевих алгоритмів машинного навчання (XGBoost, LightGBM, CatBoost) та відкритих даних з OSRM.

Методи дослідження: порівняльний аналіз алгоритмів градієнтного бустингу, розробка вебзастосунку на Flask, обробка даних у форматі CSV, синтез навчального датасету, використання Folium для візуалізації маршруту, створення інтерфейсу прогнозування з інтерактивною картою.

У результаті дослідження побудовано власний датасет маршрутних поїздок, реалізовано систему для оцінки ETA з можливістю вибору моделі, збереження результатів, а також проведено експериментальне порівняння точності моделей.

ABSTRACT

Bachelor's thesis contains: 65 pp., 22 fig., 3 tabl., 2 ann., 20 references.

CATBOOST, DELIVERY TIME PREDICTION., ENSEMBLE ALGORITHMS, ETA MODELS, FLASK, FOLIUM, LIGHTGBM, LOGISTICS OPTIMIZATION, OSRM API, REAL-TIME TRAFFIC, XGBOOST

The object of the research is the process of predicting delivery time under varying road traffic conditions.

The subject of this research is the algorithmic implementation and application of randomized ensemble gradient-boosting methods (XGBoost, LightGBM, CatBoost) for predicting delivery time (ETA) while accounting for variable road traffic and route characteristics.

The aim of the work is to develop an autonomous web application for calculating ETA (Estimated Time of Arrival) based on route characteristics using randomized ensemble machine learning algorithms (XGBoost, LightGBM, CatBoost) and open data from OSRM.

Research methods: comparative analysis of gradient boosting algorithms, web application development using Flask, data processing in CSV format, synthetic dataset generation, use of Folium for route visualization, and the creation of an interactive map-based prediction interface.

As a result of the research, a custom dataset of route trips was built, a system for ETA prediction was implemented with model selection and result storage features, and an experimental comparison of model accuracy was conducted.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної галузі	9
1.1 Актуальність задачі оптимізації логістики в умовах динамічного трафіку.....	9
1.2 Основні проблеми сучасних логістичних маршрутів	11
1.3 Огляд існуючих вебзастосунків для логістичної оптимізації	17
1.4 Наукові дослідження з використанням ансамблевих методів.....	22
2 Проектування системи та обґрунтування вибору технологій	27
2.1 Вибір алгоритмів прогнозування і принципи ансамблювання	27
2.2 Роль рандомізованих та ансамблевих алгоритмів у моделюванні трафіку.....	30
2.3 Вибір стеку технологій для реалізації вебзастосунку	32
2.4 Архітектура системи та модульна структура.....	35
3 Програмна реалізація.....	38
3.1 Підготовка даних і реалізація моделі прогнозування трафіку	38
3.2 Аналіз отриманих результаті моделі.....	44
3.3 Розробка веб-інтерфейсу та інтеграція з мапою	48
3.3.1 Архітектура Flask-додатку	49
3.3.2 Методи побудови маршруту.....	50
3.3.3 Обробка вхідних даних від користувача	51
Висновки	53
Перелік джерел посилання	55
Додаток А Лістинг файлу main.py	57
Додаток Б Відомість кваліфікаційної роботи.....	65

ВСТУП

У сучасному світі логістика відіграє ключову роль у забезпеченні безперебійного функціонування економіки, торгівлі та сервісного обслуговування. Зростання обсягів перевезень, урбанізація, а також збільшення щільності трафіку у великих містах і міжміських коридорах призводять до значних ускладнень у плануванні маршрутів. Особливо гостро постає питання затримок у доставці, які негативно впливають на фінансові результати компаній, задоволеність клієнтів і навантаження на транспортну інфраструктуру. У таких умовах класичні методи оптимізації логістичних маршрутів, що базуються на статичних розрахунках, виявляються недостатньо ефективними, оскільки не здатні враховувати динаміку дорожньої ситуації в режимі реального часу.

Водночас розвиток технологій штучного інтелекту відкриває нові можливості для підвищення точності логістичних прогнозів. Зокрема, застосування рандомізованих ансамблевих алгоритмів, таких як XGBoost, LightGBM та CatBoost, дозволяє з високою точністю моделювати очікуваний час прибуття, враховуючи такі параметри маршруту, як відстань, кількість поворотів і година виїзду. Актуальність розробки інтелектуального інструменту прогнозування ETA (Estimated Time of Arrival) підтверджується зростаючою потребою бізнесу в скороченні витрат, підвищенні ефективності постачань та забезпеченні надійності сервісу.

Метою даної кваліфікаційної роботи є створення вебзастосунку для оцінки часу доставки з урахуванням динамічних дорожніх умов, з використанням сучасних алгоритмів машинного навчання та відкритих картографічних ресурсів. Робота передбачає побудову моделі прогнозування затримки на основі реальних і синтетичних даних, реалізацію веб-інтерфейсу з інтерактивною картою маршруту, а також порівняльний аналіз точності обраних алгоритмів.

Наукова новизна дослідження полягає в інтеграції декількох моделей градієнтного бустингу у систему реального часу, здатну автоматично обирати найточніший прогноз. Крім того, пропонується власний підхід до синтезу навчальних даних, який дозволяє враховувати особливості логістичних маршрутів у міських умовах без прив'язки до конфіденційних або комерційно обмежених джерел.

Об'єктом дослідження є процес прогнозування часу доставки в логістичних системах, а предметом - методи побудови моделей ETA на основі машинного навчання та їхнє впровадження у вебзастосунок. Для реалізації поставлених завдань було використано методи аналізу трафіку, моделювання з використанням GBM-алгоритмів, розробки веб-інтерфейсів (Flask, Folium, OSRM) та валідації результатів на прикладах реалістичних маршрутів.

Отримані результати можуть бути використані для покращення транспортного планування у логістичних компаніях, підвищення точності систем управління доставками, а також як основа для створення більш масштабних інтелектуальних транспортних платформ.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Актуальність задачі оптимізації логістики в умовах динамічного трафіку

Сучасний етап розвитку економіки характеризується стрімким зростанням ролі логістики як ключового чинника конкурентоспроможності підприємств. Якість і своєчасність доставки товарів безпосередньо впливають на задоволеність клієнтів, формування іміджу компаній та фінансові результати діяльності. Особливо це актуально в умовах великих міст і агломерацій, де транспортна інфраструктура часто перебуває у стані постійного перевантаження.

За останні роки в Україні та світі суттєво зросла кількість транспортних засобів, що призвело до значного погіршення ситуації на дорогах. Міста, які не мають можливості швидко розширювати дорожню мережу, стикаються з регулярними заторами, що негативно впливає на ефективність логістичних операцій. Затримки, викликані заторами, призводять до зростання витрат підприємств, збільшення споживання пального та погіршення екологічної ситуації через підвищення викидів шкідливих речовин.

Традиційні методи планування логістики, що базуються на статичних даних та історичному досвіді, часто виявляються неефективними в умовах високої динамічності сучасного трафіку. Трафік суттєво залежить від багатьох непередбачуваних чинників, таких як погода, дорожні роботи, ДТП та інші інциденти. У зв'язку з цим зростає необхідність у впровадженні систем, які дозволяють здійснювати точний прогноз дорожньої ситуації у реальному часі.

З огляду на це все більшу популярність набувають методи прогнозування, що базуються на використанні сучасних алгоритмів машинного навчання. Вони дозволяють ефективно аналізувати великі

масиви даних, враховувати різноманітні чинники та оперативно реагувати на зміни у дорожніх умовах. Серед найефективніших підходів виділяють ансамблеві алгоритми, особливо Gradient Boosting Machines (GBM), включаючи XGBoost, LightGBM та CatBoost.

Алгоритми сімейства Gradient Boosting Machines зарекомендували себе як ефективні інструменти для роботи з великими обсягами даних, забезпечуючи високу точність прогнозів навіть у складних умовах. Зокрема, алгоритм XGBoost відомий своєю швидкістю та точністю, що робить його надзвичайно популярним для завдань прогнозування трафіку.

LightGBM, у свою чергу, характеризується ще більшою швидкістю роботи за рахунок особливої структури даних, яка дозволяє скорочувати час на тренування моделей. CatBoost демонструє чудові результати в умовах роботи з категоріальними ознаками, що особливо важливо при прогнозуванні дорожньої ситуації, де використовується велика кількість категоріальних параметрів.

Використання рандомізованих ансамблевих алгоритмів у прогнозуванні завантаженості доріг дозволяє суттєво покращити якість логістичних рішень. Завдяки цьому компанії можуть оптимально планувати маршрути, уникаючи зон найбільшого скупчення транспорту та знижуючи ризики, пов'язані з затримками.

З огляду на вищезазначене, розробка вебзастосунку, який базується на сучасних методах машинного навчання та дозволяє оперативно враховувати зміни дорожніх умов, є надзвичайно актуальним завданням. Такий застосунок дозволяє підприємствам суттєво знизити витрати на логістику, підвищити якість обслуговування клієнтів та значно покращити економічні показники діяльності.

Отже, задача оптимізації логістики в умовах динамічного трафіку має велике значення як з економічної, так і з соціальної точки зору. Реалізація сучасних інтелектуальних рішень у цій сфері дозволить значно підвищити

ефективність транспортних операцій і позитивно вплинути на розвиток бізнесу та загальну ситуацію на дорогах.

Особливої актуальності ця задача набуває в контексті урбанізації та зростання обсягів електронної комерції, які вимагають ще більш точного і швидкого обслуговування. Підприємства, які зможуть першими впровадити такі інноваційні підходи, отримають стратегічну перевагу на ринку за рахунок оптимізації витрат і підвищення рівня сервісу.

1.2 Основні проблеми сучасних логістичних маршрутів

Сучасні логістичні маршрути – це складна мережа шляхів, якими товари рухаються від виробника до споживача по всьому світу. Ефективність таких маршрутів є критичною для економіки, проте в реальних умовах логістичні системи стикаються з численними проблемами.

Ці виклики проявляються на різних масштабах – у містах, на міжміських магістралях та в міжнародних перевезеннях. Географічний контекст теж має значення: наприклад, Україна має свої особливості, Європа – інші, а глобальний рівень додає ще ширші проблеми.

Розгляньмо основні категорії проблем сучасних логістичних маршрутів – від перевантаженого трафіку й слабкої інфраструктури до екологічних, законодавчих, безпекових та технологічних викликів, включаючи вплив недавніх пандемії та криз.

Однією з найпомітніших проблем є перевантаження трафіку, особливо в урбанізованих районах. У великих містах затори на дорогах значно сповільнюють доставку товарів і підвищують витрати на «останню милю». Наприклад, дослідження в США показало, що 75 найбільших міських агломерацій сумарно втрачали близько 3,6 млрд годин у заторах щорічно, що еквівалентно приблизно 67,5 млрд доларів втрат продуктивності (рисунок 1.1), подібні цифри характерні і для Європи: затори коштують економікам мільярди євро щороку [1].

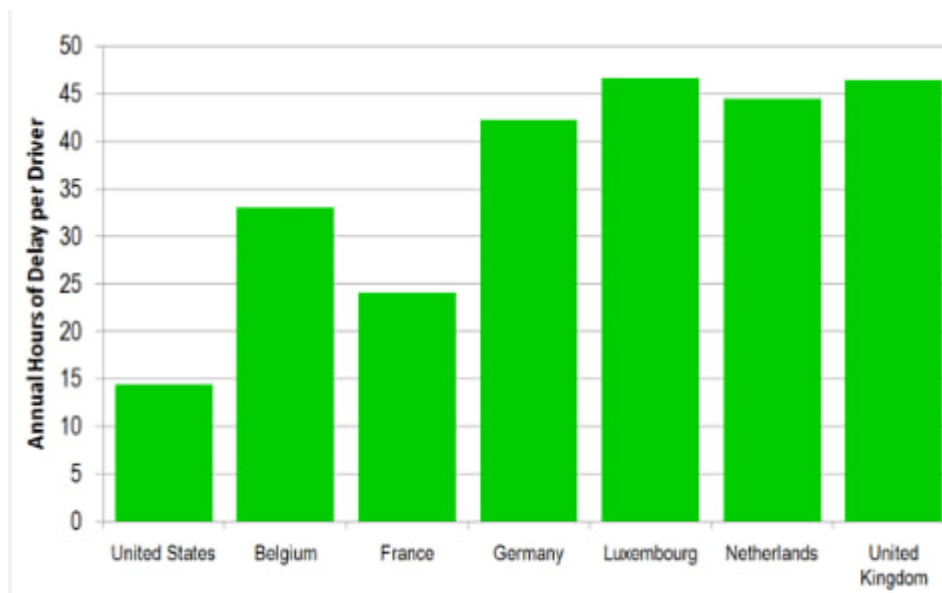


Рисунок 1.1 – Загальний річний час затримок водіїв у дорозі в США та Європейських містах

Друга ключова проблема – інфраструктура, в Україні стан дорожньої інфраструктури, на жаль, є слабким: за даними ще до війни, понад 51% українських доріг не відповідають мінімальним стандартам, майже 40% потребують капітального ремонту, а середня швидкість руху по країні в 2–3 рази нижча, ніж у Західній Європі [2].

На початку 2016 року більше половини доріг в Україні потребували ремонту, але хронічне недофінансування і перевантаженість транспортних засобів лише погіршували ситуацію. Загальна динаміка дещо покращилася з приходом програм розвитку доріг, однак воєнні дії 2022–2024 років завдали колосальної шкоди інфраструктурі – зруйновані мости, пошкоджені магістралі, що змусило перенаправляти потоки через довші маршрути.

Логістичні маршрути часто проходять через декілька юрисдикцій, і різниця в законах та правилах може створювати перешкоди. На національному рівні існують різні правила щодо габаритів і ваги вантажівок, режиму праці водіїв, дорожніх зборів, що ускладнює планування маршрутів для міжнародних перевезень.

У межах ЄС багато стандартів уніфіковано, проте навіть там перевізники стикаються з бюрократичними труднощами: наприклад, правила «Пакету мобільності» ЄС вимагають регулярного повернення водіїв додому і обмежують каботаж, що при неправильному плануванні може призводити до порожніх пробігів. На кордонах поза межами Шенгену та митного союзу зберігаються митні процедури і контроль, які можуть бути повільними.

Яскравий приклад – черги вантажівок на кордоні Україна–Польща: через необхідність митного оформлення, різні санітарні та ветеринарні перевірки, а особливо після початку війни черги фур на виїзд з України могли сягати десятків кілометрів, а час очікування – кількох діб [3].

Такі затримки збільшують загальний час доставки і вартість логістики. Інший приклад регуляторного впливу – Brexit, після виходу Британії з ЄС впровадження нових митних правил спричинило затримки в портах Ла-Маншу, нестачу водіїв і навіть перебої з доставкою пального у 2021 році, коли на АЗС утворилися черги через логістичні збої [4].

Сучасна логістика оперує величезними обсягами даних і складними ланцюгами постачання, тому технології мають критичне значення для ефективності маршрутів. Проте багато логістичних компаній відстають у цифровій трансформації, що породжує ряд технологічних проблем, досі значна частина операторів використовує примітивні засоби керування – електронні таблиці, телефонні дзвінки, паперові документи – замість інтегрованих ІТ-систем (рисунок 1.2). Така залежність від ручних процесів і неавтоматизованих рішень робить маршрути менш гнучкими та прозорими.

Менеджери ланцюгів постачання часто «воюють з сучасними викликами, маючи на озброєнні лише крем'яний ніж» – тобто працюють з інструментами, які кардинально застаріли для сучасного глобального бізнесу [5]. Це призводить до помилок, затримок у передачі інформації та неспроможності швидко переналаштувати маршрути при змінах.

Top 3 digital transformation projects in use by sector

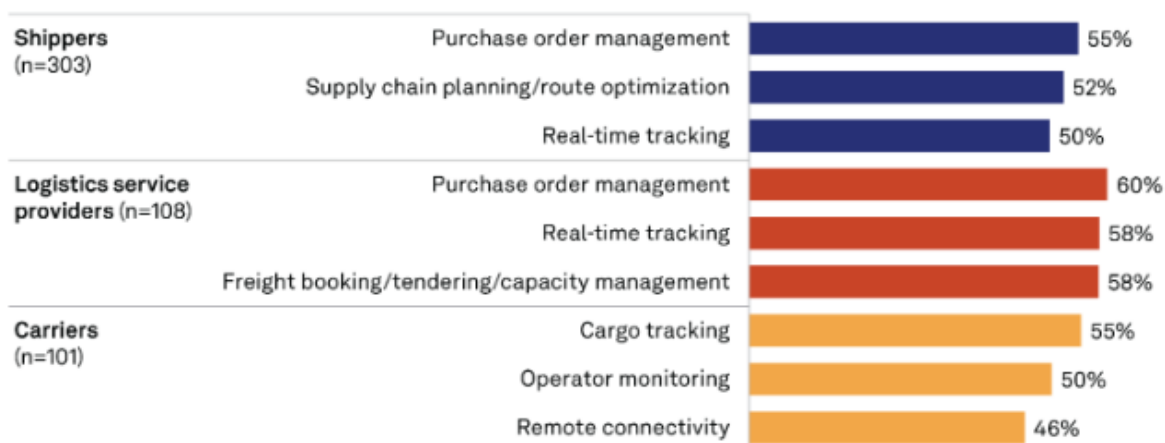


Рисунок 1.2 – Графік найпопулярніших програмних продуктів в сфері транспортування

Ефективна логістика вимагає точного прогнозування попиту та координації потоку товарів. Однак у мінливих умовах це надзвичайно важко. Система «just-in-time», яку багато компаній застосовували для економії, продемонструвала вразливість у 2020–2021 роках – несподівані зміни попиту збили з ладу налагоджені маршрути, бо вони вимагали надзвичайно точного прогнозування, якого неможливо було досягти [6]. Наприклад, під час пандемії COVID-19 різко зріс попит на медичні товари і товари першої необхідності, і більшість постачальників не змогли спрогнозувати такого стрибка. В результаті виник дефіцит контейнерів, порти перевантажилися, а заводи не встигали виконувати замовлення.

Навіть у більш звичайних умовах логістичні компанії стикаються з тим, що моделі попиту стають дедалі складнішими, і традиційні методи планування маршрутів не справляються. Необхідні інструменти на базі штучного інтелекту та машинного навчання, які могли б автоматично перепланувати маршрути під актуальні дані – і такі інструменти з'являються, але їх впровадження потребує часу та інвестицій.

Ще одна сучасна проблема – вразливість логістичних систем до кібератак і технічних збоїв. Чим більше процесів оцифровано, тим серйознішими можуть бути наслідки їх порушення. Показовим є приклад атаки вірусу NotPetya у 2017 році: цей кібернапад вразив глобальну логістичну компанію Maersk, паралізувавши її ІТ-системи по всьому світу.

За кілька днів простою Maersk зазнала збитків оцінково у \$300 млн, було повністю зупинено роботу 76 портових терміналів компанії [7]. Окрім атак, є й інші ризики – збої електроживлення, відмова GPS-систем, зламування бортових комп'ютерів вантажівок – все це потенційно здатне призвести до хаосу. Тому технологічна оснащеність логістики має й зворотний бік: потрібні значні вкладення в резервні системи, кібербезпеку та навчання персоналу, аби маршрути залишалися надійними.

Таким чином, технологічні проблеми сучасної логістики – це не лише питання впровадження новітніх ІТ-рішень, але й забезпечення їх сумісності, точності прогнозів та захищеності. Багато компаній уже усвідомили необхідність цифрової трансформації: зростає використання систем управління транспортом (TMS), відстеження вантажів у режимі реального часу, автоматизованих складів. Проте розрив між лідерами інновацій та тими, хто працює «по-старому», все ще великий, і долати його галузь буде в найближчі роки.

Останні роки продемонстрували, наскільки вразливими є логістичні маршрути перед обличчям глобальних потрясінь. Пандемія COVID-19 стала безпрецедентним шоком для світової логістики. На початку 2020 року карантинні обмеження призвели до часткового зупинення виробництв і транспорту в багатьох країнах.

Це одразу сповільнило глобальні ланцюги постачання та перевезення, спричинивши дефіцит товарів і збої у торговельних потоках. Проблеми виникали по всьому ланцюгу: не вистачало робітників у портах і на складах через хвороби чи локдауни, закривалися кордони, вводилися обов'язкові карантини для водіїв.

У результаті час доставки вантажів різко збільшився, а вартість перевезень зростає. До прикладу, в розпал пандемії дефіцит контейнерів і перевантаженість портів призвели до того, що фрахтові тарифи на морські перевезення зросли в декілька разів. Одночасно, зростаючий попит на товари підсилює навантаження на логістику тоді, коли її спроможність була обмежена.

Усе це створило «ідеальний шторм», у 2021 році через пандемію і пов'язані з нею фактори глобальні постачання працювали зі значними затримками, що відчували як бізнеси, так і споживачі наприклад, нестача мікрочипів для автомобілів або порожні полиці в магазинах у певні періоди. Пандемія висвітила необхідність більш гнучких маршрутів і резервних планів – компанії почали диверсифікувати постачальників, тримати більші запаси на складах, ніж раніше, щоб пом'якшити ефект майбутніх криз.

Військові конфлікти та геополітичні кризи також мають прямий вплив на логістику. Найбільш актуальний приклад – російське вторгнення в Україну у 2022 році, що сколихнуло як регіональну, так і світову систему постачань. Логістичні маршрути для українського експорту перебудувалися на суходіл, але пропускна здатність залізниць і доріг через кордони виявилася значно меншою, ніж морських шляхів, що призвело до вузьких місць (рисунок 1.3).

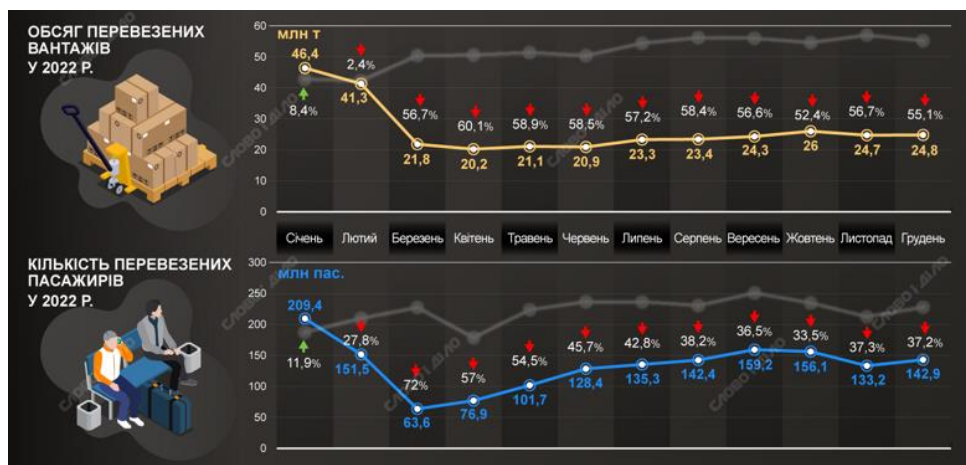


Рисунок 1.3 – Вплив повномасштабного вторгнення на транспортний сектор

Війна вплинула і на енергетичну логістику Європа терміново перебудовувала маршрути постачання газу, нафти, пошук альтернатив. Це створило ефект доміно для всіх логістичних систем, пов'язаних з енергоносіями. Крім того, Росія і Україна є постачальниками деякої стратегічної сировини – наприклад, значної частки ринку неону і благородних газів, а також металів.

Перебої в їх постачанні через конфлікт вплинули на цілі галузі, а логістичні ланцюги змушені були шукати нові джерела і маршрути підвезення цих ресурсів. Фізичне знищення інфраструктури в зоні війни також лишило довготривалий ефект: в Україні зруйновано численні мости, залізничні вузли, пошкоджено порти – відновлення та переналаштування маршрутів займе роки. Геополітичні напруження в інших регіонах теж впливають: введення санкцій або закриття кордонів змінює маршрути потоків товарів, вимагає переналаштовувати логістику.

Інші глобальні інциденти показують вразливість окремих логістичних «артерій». Так, у березні 2021 року сталася резонансна подія – контейнеровоз Ever Given сів на мілину в Суецькому каналі, повністю перегородивши одну з найважливіших судноплавних магістралей світу на шість днів. Це спричинило утворення гігантської черги з суден і паралізувало торгівлю між Європою та Азією, адже Суецьким каналом проходить близько 12% світової торгівлі.

1.3 Огляд існуючих вебзастосунків для логістичної оптимізації

У світі бізнесу, де швидкість і ефективність постачань стають вирішальними, компанії дедалі частіше звертаються до вебзастосунків для логістичної оптимізації. Такі B2B-платформи дозволяють автоматизувати та координувати складні процеси перевезень, відстежувати вантажі у режимі реального часу й аналізувати ланцюги постачань.

Рішення нового покоління об'єднують усі ключові функції – від побудови оптимальних маршрутів і GPS-моніторингу транспорту до управління складами та інтеграції з ERP/CRM-системами. У результаті бізнеси – дистриб'ютори, 3PL-оператори, транспортні компанії, ритейлери, виробники – отримують прозорість, прогнозованість та зниження витрат у логістиці.

Далі розглянемо найпоширеніші міжнародні та українські веб-платформи, їхні функціональні особливості, переваги та цільове призначення.

Transporeon – провідна європейська SaaS-платформа транспортної логістики, яка поєднує вантажовласників, перевізників, 3PL та ритейлерів у єдиній цифровій екосистемі. Transporeon вважається платформою №1 у Європі для управління транспортом, об'єднуючи понад 1400 вантажовідправників, 158 тис. перевізників і 100+ торговельних мереж [10].

У платформі реалізовано модулі для електронного розміщення заявок на перевезення, управління часовими слотами на навантаження, відстеження доставки в реальному часі, мобільний додаток водія для підтвердження операцій та модуль звітності з ключових показників.

Використовуючи єдину онлайн-систему Transporeon, компанії можуть автоматизувати процеси, мінімізувати холості пробіги, знижувати витрати на перевезення і прискорювати документообіг (рисунок 1.4).

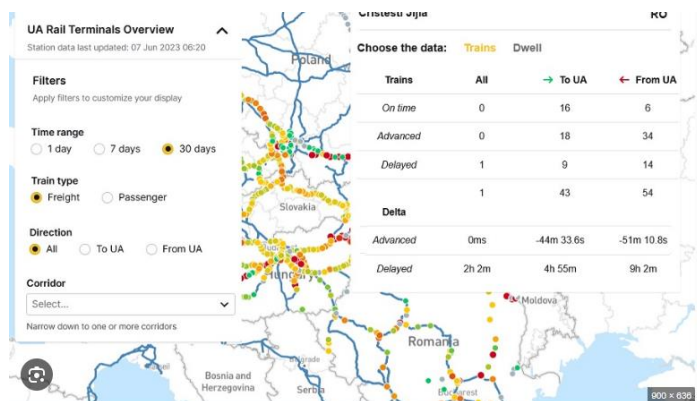


Рисунок 1.4 – Інтерфейс застосунку від компанії Transporeon

Одним з лідерів ринку є американська компанія Project44, що пропонує хмарну платформу Advanced Visibility для відстеження вантажів у реальному часі на всіх видах транспорту. Її рішення виступає єдиним цифровим середовищем для всіх учасників, забезпечуючи наскрізну видимість перевезень.

Платформа інтегрується з системами перевізників та агрегує дані з бортових GPS, дозволяючи відстежувати понад 1 млрд відправлень на рік для 1300+ компаній по всьому світу (рисунок 1.5). Вона підвищує прогнозованість поставок: система за допомогою AI прогнозує очікуваний час прибуття і можливі затримки вантажу, тож компанії можуть завчасно взяти заходів для дотримання графіка [9]. Цільова аудиторія Project44 – середній та великий бізнес у різних галузях, що потребує прозорості міжнародних і регіональних перевезень.

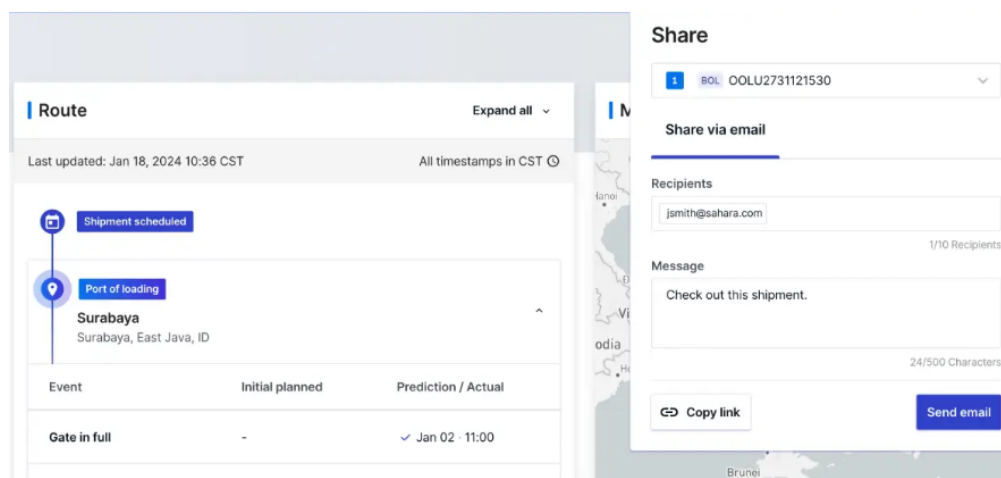


Рисунок 1.5 – Програмний продукт компанії Project44

Uber Freight – веб-платформа від компанії Uber, яка з'єднує вантажовідправників із вантажними перевізниками, аналогічно до того, як Uber об'єднує пасажирів з водіями таксі. Сервіс стартував у США як цифровий брокер вантажів, а нині працює і в Європі. Головна цінність Uber

Freight – спрощення та прискорення фрахтування перевезень онлайн (рисунок 1.6).

Платформа автоматично підбирає для вантажу доступну вантажівку та надає замовникам прозорі тарифи без тривалих переговорів [11]. Користувач через веб-інтерфейс може отримати розцінку, забронювати перевезення і відстежувати вантаж у дорозі в режимі реального часу. Водночас перевізники (в тому числі незалежні далекобійники) бачать доступні замовлення, можуть миттєво їх прийняти та швидко отримати оплату.

Ключові можливості Uber Freight:

- платформа в реальному часі підбирає вантаж під вільний транспорт, а вартість перевезення відома наперед;
- вантажовідправник стежить за рухом фури по маршруту та отримує повідомлення про статус доставки;
- у платформі можна обмінюватися електронними документами (CMR, підтвердження доставки тощо), зменшуючи паперову тяганину.

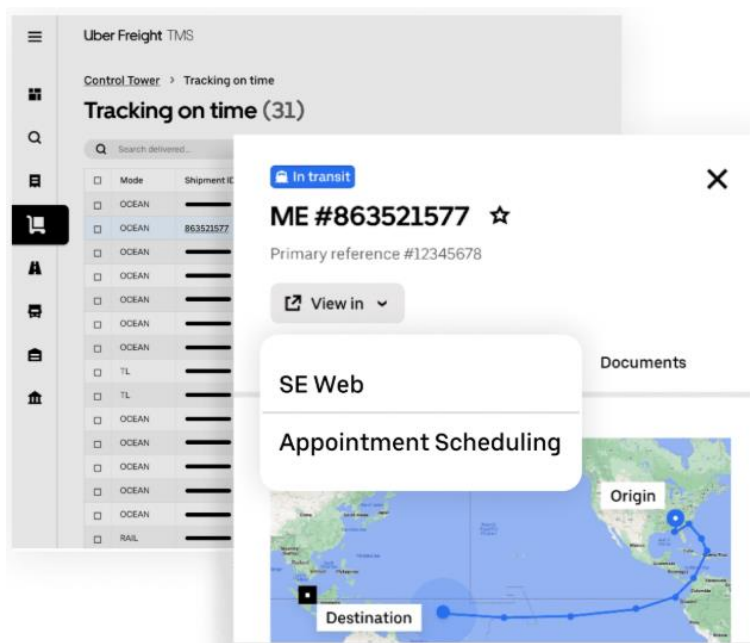


Рисунок 1.6 – Програмний продукт компанії Uber Freight

Серед українських розробок у сфері логістики помітним є стартап Cargofy, який часто називають «Uber для дальнобійників». Це онлайн-платформа, що об'єднує вантажовідправників і водіїв вантажівок, допомагаючи знайти вигідні замовлення на перевезення [12]. Запущений у 2014 році в Києві сервіс згодом переорієнтувався на ринок США, залучивши інвестиції для зростання.

Cargofy щодня пропонує тисячі актуальних тендерів з прозорими умовами співпраці та рейтингами водіїв; AI-диспетчер автоматично підбирає рейси під наявний транспорт, генерує необхідні документи і забезпечує безпечні розрахунки між сторонами. Платформа орієнтована передусім на незалежних водіїв і невеликі транспортні компанії, яким важко самотійно знаходити клієнтів. Для українського ринку такий сервіс є новаторським, адже традиційно пошук попутного вантажу здійснювався через особисті домовленості.

Lardi-Trans – відома в Україні та сусідніх країнах онлайн-платформа для пошуку вантажів і транспорту, що працює з 2001 року. Вона еволюціонувала з дошки оголошень у багатофункціональний сервіс для учасників ринку вантажних перевезень. Платформа об'єднує тисячі перевізників, вантажовідправників і експедиторів з Європи та Азії. Завдяки широкому функціоналу Lardi-Trans залишається головним інструментом українських перевізників та експедиторів у щоденній роботі (рисунок 1.7).



Рисунок 1.7 – Вебінтерфейс застосунку компанії Lardi-Trans

1.4 Наукові дослідження з використанням ансамблевих методів

У 2022 році було проведено порівняльний аналіз різних ансамблів для класифікації стану трафіку на основі реального датасету дорожнього руху. Дослідники випробували методи bagging, boosting та стекінг моделей. Результати показали, що найкращої точності вдалося досягти за допомогою ансамблю AdaBoost із базовими деревами рішень – він перевершив навіть сам по собі Random Forest за всіма метриками.

Цікаво, що найбільш важливими ознаками для прогнозу заторів виявилися середня швидкість руху, об'єм трафіку та час у дорозі, тобто саме ті фактори, які інтуїтивно впливають на затори. Це підтвердило, що ансамблева модель не лише точна, а й змістовно зрозуміла, оскільки виділяє значущі параметри.

Інше дослідження зосередилося на пояснюваності моделей для прогнозування інтенсивності потоку. Було порівняно Random Forest, CatBoost, LightGBM та кілька інших алгоритмів з точки зору точності та інтерпретації залежностей [14]. Згідно з результатами, Random Forest і CatBoost забезпечили найкращу точність прогнозу та мали важливості ознак, що співпадають з реальною значущістю факторів трафіку.

CatBoost натомість виділила серед ключових факторів час доби і завантаженість сусідніх доріг – і справді, ці фактори в реальності визначають інтенсивність потоку. Таким чином, сучасні ансамблеві методи не лише дають точний прогноз, а й можуть бути використані для пояснення, які умови призводять до заторів, допомагаючи транспортним інженерам у прийнятті рішень.

Ще один актуальний напрям – комбінування ансамблевих моделей з іншими підходами. У 2024 році було запропоновано глибокий ансамбль для довгострокового прогнозування трафіку [15]. В рамках цього підходу автори поєднали декілька нейронних мереж (LSTM, CNN, GRU) як базові моделі, а зверху навчили мета-learner для об'єднання їхніх прогнозів. Така

модель адаптивно призначає ваги кожному базовому прогнозисту залежно від його точності, що дозволило підвищити стабільність прогнозу на 24 години вперед у порівнянні з будь-якою окремою нейронною мережею.

Хоча в цьому випадку використовувалися глибинні моделі, ідея показує загальний принцип ансамблю: різноманітні прогнози комбінуються, щоб компенсувати недоліки один одного. В принципі, схожу схему можна застосувати і для композиції деревних моделей. Це напрямок розвитку – створення ансамблів ансамблів – що ще більше підвищує надійність систем прогнозування трафіку.

На першому етапі формується навчальна вибірка та проводиться попередня обробка даних. Далі тренуються базові моделі – кілька нейронних мереж з різними часовими лагами, які прогнозують трафік (рисунок 1.8).

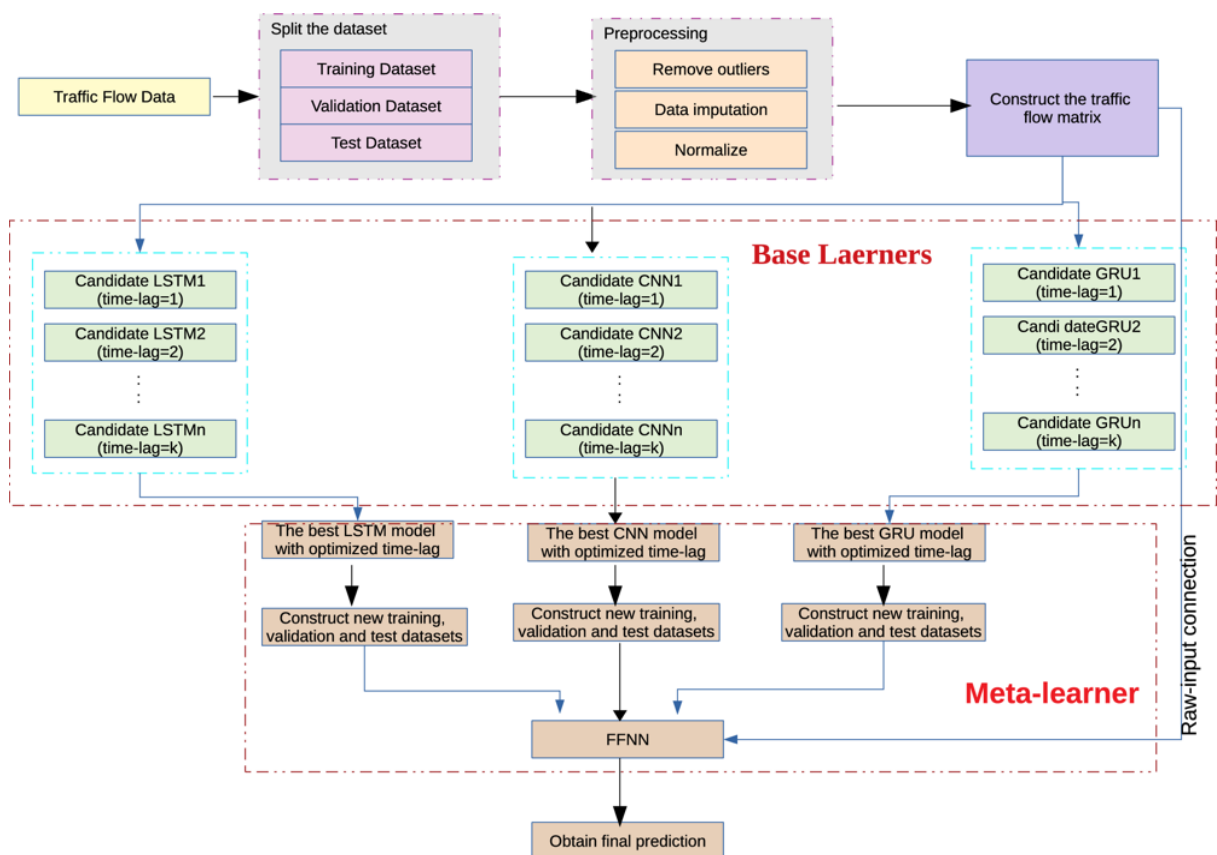


Рисунок 1.8 – Схема ансамблевої моделі для прогнозування трафіку

На третьому етапі їх результати комбінуються за допомогою мета-алгоритму – додаткової нейронної мережі, що навчається призначати оптимальні вагові коефіцієнти для виходів кожної з базових моделей. В результаті метамодель виробляє остаточний прогноз, який враховує сильні сторони кожного з прогнозистів. Хоча тут базові Learner'и – це глибокі нейронні мережі, аналогічний підхід може використовувати ансамбль з деревних моделей для підвищення точності та робастності. Така багатоетапна ансамблева система здатна автоматично підлаштовуватися під зміну умов трафіку та покращувати результати порівняно з будь-якою окремою моделлю.

Рандомізовані алгоритми на зразок Random Forest та методи ансамблевих алгоритмів нині відіграють ключову роль у моделюванні дорожнього трафіку. Вони успішно вирішують основні проблеми цієї сфери: враховують велику кількість вхідних факторів, працюють з неповними та зашумленими даними, захоплюють нелінійні залежності та взаємодії. Ансамблеві підходи демонструють високу точність прогнозування завдяки комбінуванню сильних сторін кількох моделей, що було підтверджено в ряді наукових робіт [16].

Важливо, що ці моделі не є «чорними ящиками», вони надають інструменти інтерпретації і тим самим дозволяють зрозуміти, чому приймається те чи інше рішення. У підсумку, застосування рандомізованих і ансамблевих алгоритмів значно покращує якість та надійність прогнозування дорожнього трафіку, сприяючи ефективнішому управлінню транспортними потоками у сучасних містах.

1.5 Постановка задачі

Актуальність дослідження зумовлена зростаючою потребою у точному, адаптивному та автономному прогнозуванні часу доставки (ETA) в умовах динамічного завантаження доріг. Сучасні логістичні компанії,

служби доставки та платформи перевезень стикаються з труднощами планування маршрутів, пов'язаними з непередбачуваністю трафіку, інфраструктурними обмеженнями та потребою скорочення витрат при збереженні високого рівня сервісу. Традиційні методи розрахунку ЕТА базуються на фіксованих середніх швидкостях та евристичних правилах, що ігнорують реальний стан доріг і призводять до неточних або надто оптимістичних оцінок.

Моделі машинного навчання, зокрема рандомізовані ансамблеві алгоритми (XGBoost, LightGBM, CatBoost), демонструють високу точність у задачах регресії та здатні враховувати складні взаємозалежності між різними ознаками маршруту. Завдяки відкритим картографічним даним (OSRM, OpenStreetMap) та інструментам візуалізації (Folium), з'являється можливість створити автономний вебзастосунок, що працює незалежно від комерційних сервісів, оперативно генерує маршрут і прогнозує час доставки для будь-якої пари точок.

Об'єкт дослідження – процес прогнозування часу доставки в умовах змінного завантаження доріг.

Предмет дослідження – застосування та алгоритмічна реалізація ансамблевих методів машинного навчання (XGBoost, LightGBM, CatBoost) для побудови ЕТА-моделі на основі параметрів маршруту та вхідних характеристик.

Метою дослідження є створення автономного вебзастосунку для прогнозування ЕТА на основі характеристик маршруту з використанням ансамблевих моделей градієнтного бустингу та відкритих геоданих OSRM.

Для досягнення поставленої мети передбачено виконання таких завдань:

- провести аналіз сучасних підходів до логістичної оптимізації в умовах динамічного трафіку;
- розглянути та порівняти ефективність XGBoost, LightGBM, CatBoost в задачах прогнозування часу доставки;

- здійснити збір, обробку та синтез навчального датасету на основі маршрутизованих поїздок;
- реалізувати вебзастосунок із можливістю вибору моделі прогнозування ETA, побудови маршруту та візуалізації результату на інтерактивній мапі;
- провести експериментальне порівняння точності моделей, визначити доцільність їхнього об'єднання в ансамбль;
- забезпечити автономну роботу системи без зовнішніх комерційних API, із підтримкою локального використання.

Наукова новизна полягає в інтеграції одразу кількох ансамблевих моделей у логістичну систему реального часу, а також у розробці інтерфейсу для їх динамічного використання в залежності від задачі.

2 ПРОЄКТУВАННЯ СИСТЕМИ ТА ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ

2.1 Вибір алгоритмів прогнозування і принципи ансамблювання

У цьому розділі розглянуто три сучасні алгоритми градієнтного бустингу дерев рішень – LightGBM, XGBoost та CatBoost – для задачі оцінки затримок і розрахунку очікуваного часу прибуття в логістичному вебзастосунку (рисунок 2.1).







	CatBoost		LightGBM		XGBoost	
	Tuned	Default	Tuned	Default	Tuned	Default
 Adult	0.26974	0.27298 +1.21%	0.27602 +2.33%	0.28716 +6.46%	0.27542 +2.11%	0.28009 +3.84%
 Amazon	0.13772	0.13811 +0.29%	0.16360 +18.80%	0.16716 +21.38%	0.16327 +18.56%	0.16536 +20.07%
 Click prediction	0.39090	0.39112 +0.06%	0.39633 +1.39%	0.39749 +1.69%	0.39624 +1.37%	0.39764 +1.73%
 KDD appetency	0.07151	0.07138 -0.19%	0.07179 +0.40%	0.07482 +4.63%	0.07176 +0.35%	0.07466 +4.41%
 KDD churn	0.23129	0.23193 +0.28%	0.23205 +0.33%	0.23565 +1.89%	0.23312 +0.80%	0.23369 +1.04%
 KDD internet	0.20875	0.22021 +5.49%	0.22315 +6.90%	0.23627 +13.19%	0.22532 +7.94%	0.23468 +12.43%

Рисунок 2.1 – Результати тестування моделей градієнтного бустингу

Кожен із зазначених алгоритмів має свої переваги та недоліки в контексті цієї задачі. Нижче наведено аналітичний опис кожного алгоритму, їх порівняння за ключовими характеристиками, а також обґрунтування використання ансамблю з усіх трьох моделей для підвищення якості

прогнозу. Окремо розглянуто практичні аспекти застосування цих моделей у масштабованому логістичному сервісі.

LightGBM (Light Gradient Boosting Machine) – це алгоритм, розроблений компанією Microsoft, який відомий своєю високою швидкістю навчання. LightGBM використовує гістограмні наближення при побудові дерев рішень замість точного перебору можливих розбиттів, як у класичному XGBoost. Такий підхід дещо знижує потенційну точність моделі порівняно з точним алгоритмом, проте різниця мінімальна, зате дає значний приріст у швидкості роботи.

З недоліків LightGBM можна виділити потребу налаштування параметрів, щоб уникнути перенавчання через агресивний приріст дерев. Також для повноцінної роботи з категоріями інколи потребує попереднього кодування, якщо не використовувати вбудовані засоби.

XGBoost (Extreme Gradient Boosting) – один з найпопулярніших алгоритмів градієнтного бустингу, що встановив стандарт якості прогнозування на структурованих даних. XGBoost будує ансамбль дерев, оптимізуючи задану функцію втрат, і включає ряд вдосконалень: регуляризацію, вбудовану обробку пропусків та паралельний розрахунок. В задачі оцінки ETA XGBoost демонструє надійну точність прогнозу завдяки здатності моделювати складні нелінійні залежності між ознаками.

CatBoost – градієнтний бустинг, розроблений компанією Yandex у 2017 році, особливо орієнтований на роботу з категоріальними. Назва «CatBoost» походить від Categorical Boosting. Його ключова особливість – алгоритми побудови дерев і навчання, що запобігають перенавчанню на категоріальних ознаках. Зокрема, CatBoost використовує симетричні дерева та Ordered Boosting. Це дозволяє отримати стабільні моделі навіть на невеликих чи зашумлених вибірках.

Для категоріальних ознак CatBoost застосовує власні методи кодування на основі таргет-статистик з перестановочним контролем, що суттєво знижує інформаційні протікання.

Основна відмінність алгоритмів полягає в стратегії зростання. В CatBoost усі гілки розширюються одночасно, XGBoost додає нові вузли на одному рівні одразу для кількох гілок, а LightGBM поглиблює ту гілку, яка дає найбільше зменшення помилки, що є більш агресивною стратегією (рисунок 2.2).

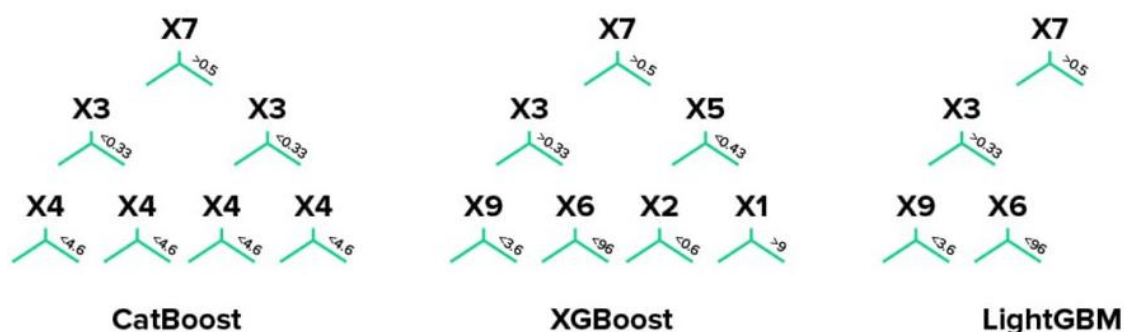


Рисунок 2.2 – Відмінність у стратегіях зростання дерев рішень

Під час роботи з категоріальними ознаками, CatBoost показує найкращі результати, він безпосередньо приймає дані і автоматично генерує для них надійні кодування. LightGBM також дозволяє врахувати такі дані, що значно спрощує підготовку даних у порівнянні з повним one-hot кодуванням.

XGBoost натомість не підтримує категоріального напрямку, тому для нього доводиться виконувати зовнішню обробку: при невеликій кількості категорій можна застосувати one-hot кодування, а при великій – порядкове або таргетне кодування, але це додає складності у підготовці даних і може вплинути на якість.

Пропущені значення всі три алгоритми обробляють внутрішньо, тому відсутність певних даних не призводить до зниження точності моделі. XGBoost під час побудови дерева сам обирає оптимальний напрям для записів з пропусками. LightGBM подібно виділяє пропуски як окрему категорію і теж шукає для них оптимальне рішення.

CatBoost трактує пропуски або як окрему категорію, або може заповнювати їх статистично обґрунтованими значеннями, особливо якщо йдеться про числові поля. Таким чином, жоден з алгоритмів не є надмірно чутливим до пропусків – всі здатні давати точні прогнози навіть при частково неповних даних, що підвищує надійність ETA прогнозу.

Хоча кожна з розглянутих моделей окремо дає досить високу точність прогнозування, додаткового покращення можна досягти шляхом їх об'єднання в ансамбль. Підходи *stacking* та *blending* передбачають комбінування виходів кількох алгоритмів для отримання остаточного результату. Ідея полягає в тому, що різні моделі можуть вловлювати різні аспекти залежностей у даних, і об'єднавши їх передбачення, можна компенсувати індивідуальні похибки моделей.

Наразі доцільно використати всі три градієнтні бустинги як базові моделі: XGBoost, LightGBM та CatBoost мають різні архітектури і гіперпараметри, тому їх помилки слабо корельовані між собою. Ансамблюючи їх, ми отримуємо більш стабільний і точний прогноз ETA.

Практично реалізація ансамблю може бути виконана двома способами. Перший – *blending* – побудувати просту модель-комбінатор, яка на основі прогнозів трьох моделей видає фінальний результат. Другий – *stacking* – навчити мета-модель на виходах базових моделей.

Обидва підходи спрямовані на те, щоб надати більшої ваги тим моделям, які є точнішими в певних випадках, і зменшити вплив тих, що можуть помилятися. В цьому проекті, ми використаємо у якості мета-алгоритму для *stacking* лінійну нейронну мережу.

2.2 Роль рандомізованих та ансамблевих алгоритмів у моделюванні трафіку

Прогнозування завантаженості доріг – одна з найскладніших задач в інтелектуальних транспортних системах. Причина в тому, що трафік є

високодинамічним процесом із багатьма факторами впливу. Потік машин змінюється залежно від часу доби, дня тижня, погоди, дорожніх умов, аварій, подій у місті та інших чинників. Такі фактори роблять поведінку трафіку нелінійною і важкопередбачуваною [17].

Крім того, реальні дані зазвичай нерегулярні та неповні: можуть бути пропуски у показах датчиків, збої у зборі даних, різкі коливання через непередбачувані події. Нерівномірність та переривчастість даних неминучі в реальних умовах, що ускладнює застосування стандартних моделей, які припускають безперервність і стаціонарність даних.

Інша проблема – це неповнота даних, не всі ділянки доріг обладнані датчиками, а ті, що є, можуть час від часу не передавати інформацію. В результаті моделі отримують вхідні дані з прогалинами. «У разі неповних даних прогнозування трафіку стає справжнім викликом» [18]. Нерідко доводиться виконувати імпутацію відсутніх значень або спиратися на непрямі ознаки. Усе це додає невизначеності моделюванню.

Варто також врахувати просторово-часову природу дорожнього трафіку. Стан однієї ділянки дороги залежить від ситуації на сусідніх дорогах і розв'язках. Затори мають ефект «хвилі», що поширюється мережею доріг. Через це модель повинна одночасно враховувати просторові залежності та часову динаміку. Така багатовимірність ще більше підвищує складність задачі.

Спочатку дані про трафік підлягають попередній обробці і розподіляються на тренувальну та тестову вибірки. Далі будуються моделі прогнозування, які можуть комбінуватися в ансамбль, в даному випадку стекінг. На завершення результат отриманої моделі обчислюється метрики якості прогнозу.

Такий поетапний процес дозволяє забезпечити якісні дані для моделювання та підвищити надійність прогнозу (рисунок 2.3).

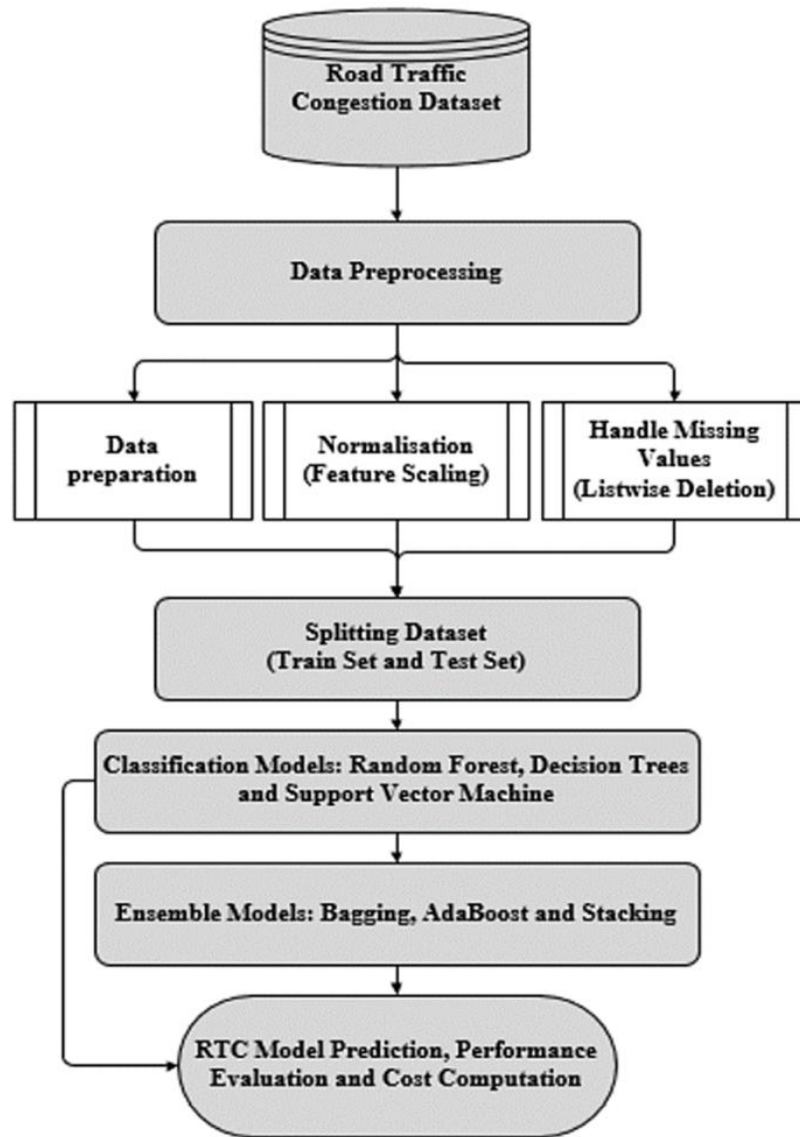


Рисунок 2.3 – Pipeline-схема побудови моделі прогнозування дорожнього трафіку

2.3 Вибір стеку технологій для реалізації вебзастосунку

Розробляючи локальний вебзастосунок для прогнозування часу прибуття та візуалізації маршруту, було важливо обрати технологічний стек, який забезпечить практичність, швидкість реалізації та автономність рішення, а також залишить можливість масштабування в майбутньому. Нижче наведено обґрунтування вибору основних компонентів стеку.

Для серверної частини застосунку обрано фреймворк Flask. Він відомий зручністю та мінімалістичним підходом: він забезпечує лише необхідний базовий функціонал, що зменшує зайві накладні витрати і сприяє ефективному використанню ресурсів. Завдяки такій архітектурі Flask прискорює цикл розробки і дозволяє швидко створювати прототипи вебзастосунків. Це дало змогу оперативно реалізувати робочий продукт. Попри простоту,

Flask легко розширюється за рахунок численних розширень та інтегрується з різними бібліотеками Python, що забезпечує можливість додати авторизацію або інші функції в майбутньому. Також наявна підтримка механізму масштабування у разі зростання навантаження. У даному випадку він дозволив обійтися без використання зовнішньої бази даних: для зберігання мінімальної інформації про сесію застосовується вбудований механізм сесій на основі підписаних cookies, що забезпечує автономність роботи застосунку.

Для відображення маршруту на карті було обрано бібліотеку Folium, яка побудована на основі JavaScript-бібліотеки Leaflet. Він надає простий і інтуїтивний API для створення інтерактивних карт всього кількома рядками коду. Це дало змогу генерувати карту з маршрутом прямо із Python-коду без необхідності писати окремий JavaScript-код. Folium автоматично використовує відкриті картографічні дані (OpenStreetMap) і можливості Leaflet для рендерингу карти на стороні клієнта, що підвищує автономність рішення – нема потреби у зовнішніх картографічних сервісах.

Такий підхід прискорює реалізацію візуалізації та зменшує складність фронтенду. До того ж Folium є відкритим ПЗ з активною спільнотою, тож інструмент перевірений і підтримуваний спільнотою розробників. У перспективі, за потреби більш складного інтерфейсу, можна буде замінити або доповнити Folium іншими інструментами, але на початковому етапі Folium забезпечив швидкий і надійний результат.

Модель машинного навчання для прогнозування ETA була попередньо навчена та збережена у вигляді .pkl-файлів. У застосунку ці моделі завантажуються за допомогою бібліотеки `joblib`. Такий підхід надзвичайно зручний для продакшн-середовища з кількох причин. По-перше, це дозволяє уникнути повторного навчання моделі при кожному запуску: достатньо один раз навчити модель офлайн і серіалізувати її на диск, а в робочому середовищі просто завантажувати файл.

Як зазначено в документації `scikit-learn`, після тренування модель доцільно зберегти для подальшого використання без необхідності перенавчання. Бібліотека `joblib` спеціально оптимізована для серіалізації об'єктів на зразок моделей `scikit-learn` і великих масивів даних, забезпечуючи ефективне використання пам'яті та можливість компресії моделей для зменшення розміру файлу.

Завантаження моделі займає доли секунди, що не створює помітної затримки під час запуску застосунку або при першому запиті. Це сприяє швидкодії рішення. У застосунку реалізовано вибір моделі користувачем, і вибрана модель зберігається в сесії `Flask`, що дозволяє легко переключатися між різними моделями без перезапуску сервера.

Для отримання маршруту між двома точками використовується `OSRM` (Open Source Routing Machine) через API. `OSRM` – це високопродуктивний рушій для розрахунку маршруту, написаний на `C++`, призначений для роботи з даними `OpenStreetMap`. `OSRM` застосовує ефективні алгоритми для пошуку найкоротших шляхів, тому здатен знаходити маршрут практично миттєво навіть у масштабах континенту.

`OSRM` є відкритим програмним забезпеченням, тому його можна розгорнути локально і використовувати без звернення до сторонніх платних API. У даному випадку вся логіка побудови маршруту реалізована або на локальному сервері або через безкоштовний публічний `OSRM` API, що виключає залежність від комерційних сервісів.

Цей сервіс зручно інтегрується з Folium/Leaflet, повертаючи координати маршруту у форматі GeoJSON або polyline, які ми безпосередньо наносимо на карту. Крім того, OSRM надає додаткові дані, які можна використовувати як ознаки для моделі або для відображення користувачу. Таким чином, OSRM виконує роль автономного «двигуна» для навігації, забезпечуючи застосунок готовими маршрутами і метриками маршруту.

2.4 Архітектура системи та модульна структура

Користувач взаємодіє з веб-інтерфейсом, який передає введені дані на шар логіки (рисунок 2.4). На рівні логіки RouteManager звертається до OSRM API для отримання маршруту; ETAEstimator використовує ML-модель для розрахунку ETA; InstructionFormatter готує текстові інструкції.

Отримані результати повертаються на рівень представлення: будується карта маршруту та відображається прогнозований час прибуття користувачу.

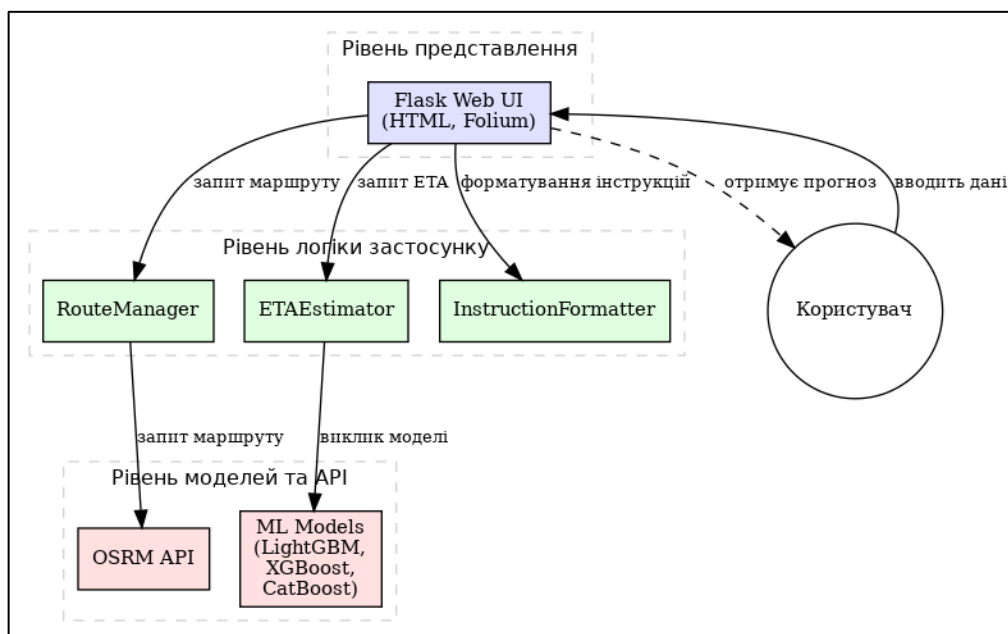


Рисунок 2.4 – Логічна схема архітектури застосунку

Центральним класом є `WebApp`, який координує роботу інших модулів: через `RouteManager` отримує дані маршруту, за допомогою `ETAEstimator` розраховує час прибуття, і використовує `InstructionFormatter` для форматування інструкцій (рисунк 2.5). Пунктирні стрілки на діаграмі показують залежності між `WebApp` та допоміжними класами.

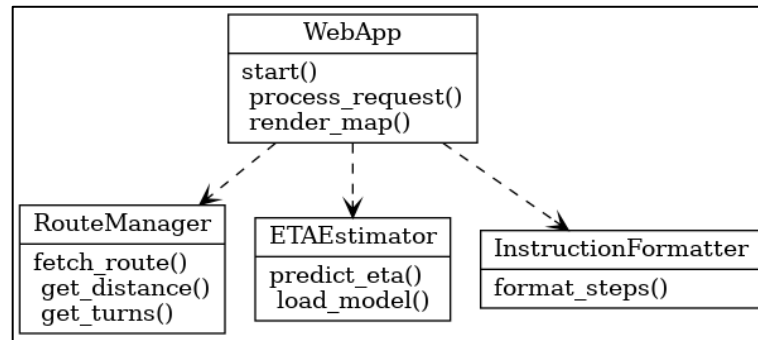


Рисунок 2.5 – Діаграма класів застосунку

На діаграмі показано взаємодію користувача із системою (рисунк 2.6). Один актор – Користувач – виконує три основні дії: введення координат початкової і кінцевої точок, вибір моделі прогнозування ЕТА та отримання результату.



Рисунок 2.6 – Діаграма взаємодії користувача з інтерфейсом

Архітектура застосунку побудована модульно, що спрощує розвиток та підтримку коду. Кожен модуль відповідає за окремий аспект функціоналу:

- отримує дані від користувача через веб-форму (координати початку і кінця маршруту, вибір моделі прогнозування) та відображає результати. Для представлення використовує HTML-шаблони і Folium, а також механізм сесій Flask для збереження стану між запитами;
- надсилає запит до OSRM API для побудови маршруту між заданими координатами. Обробляє отриману інформацію – витягує загальну відстань, список покрокових інструкцій та рахує кількість поворотів;
- перетворює сирі навігаційні інструкції від OSRM у зручний для користувача текстовий формат;
- завантажує наперед навчені ML-моделі та обчислює прогнозований час прибуття. На вхід отримує характеристики маршруту (відстань, повороти, час доби виїзду тощо) і повертає розрахований ETA відповідно до обраної моделі.

Розроблювана логістична веб-система призначена для прогнозування часу прибуття транспортного засобу на основі маршруту. Система враховує характеристики маршруту: загальну відстань, кількість поворотів на шляху та час виїзду. Мета – підвищити точність прогнозу використовуючи методи машинного навчання. У рамках проєкту інтегровані моделі LightGBM, XGBoost та CatBoost, які навчено оцінювати час поїздки за заданим маршрутом. Маршрут та навігаційні дані отримуються через зовнішній API OSRM, а для візуалізації маршруту використовується бібліотека Folium.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Підготовка даних і реалізація моделі прогнозування трафіку

Для побудови моделі були використані реальні GPS-дані про поїздки транспортних засобів. Дані охоплюють інформацію про маршрути пересування автомобілів у міських умовах: координати початку та кінця поїздки, часові мітки виїзду і прибуття, а також зафіксований трек руху.

На основі сирих GPS-даних було розраховано ознаки, необхідні для моделювання: загальна відстань маршруту, кількість поворотів на маршруті та година виїзду. Ці фактори було обрано як основні, оскільки вони суттєво впливають на тривалість поїздки (наприклад, довші маршрути займають більше часу, маршрути з багатьма поворотами можуть бути повільнішими, а година виїзду враховує вплив годин «пік»).

GPS-дані були отримані шляхом збору треків руху транспортних засобів протягом певного періоду. Кожен запис даних відповідає окремій поїздки і містить географічні координати початкової та кінцевої точок, час початку руху, а також інформацію про шлях (напрямок руху на кожному відрізьку шляху).

Дані охоплюють поїздки по різних типах доріг (магістралі, міські вулиці) у різний час доби. Загалом зібрано декілька тисяч записів поїздок, що забезпечує достатню вибірку для навчання моделей. Для імітації реальних умов руху, дані включають поїздки як в години пік, так і в періоди низького трафіку.

З отриманих даних були сформовані наступні змінні для моделювання прогнозу часу поїздки:

- загальну довжину маршруту в кілометрах отримано шляхом сумування відстаней між послідовними GPS-точками траєкторії або на основі даних карти для заданих координат початку та кінця. Відстань ϵ

базовим фактором, оскільки за інших рівних умов довший шлях потребує більше часу;

– число поворотів ліворуч чи праворуч на маршруті характеризує складність, маршрут з багатьма поворотами, кількість яких визначено шляхом аналізу послідовності напрямків руху: якщо кут зміни напрямку між сегментами траєкторії перевищує певний поріг, це вважається поворотом. Для обчислення використовувалась карта дорог – на основі координат маршруту визначено точки поворотів;

– час доби, коли почалася поїздка (у вигляді цілого числа від 0 до 23, що відповідає годині доби), відображає вплив добових коливань трафіку: в ранкові та вечірні години пік рух уповільнюється через затори, тоді як вночі дороги більш вільні. Значення години було виділено з мітки часу початку поїздки шляхом приведення його до формату дати-часу і вибірки компоненти години.

Цільовою змінною для моделі є фактичний час поїздки, виміряний у хвилинах. Він обчислювався як різниця між часом прибуття та часом виїзду для кожного запису маршруту. Цей показник слугуватиме тим значенням, яке модель повинна навчитися прогнозувати на основі трьох зазначених ознак.

Спочатку було проведено порівняння ефективності моделей для прогнозування затримок. Для цього дані зчитуються з файлу `logistics_data.csv`, після чого із загального набору даних обираються три ключові ознаки: відстань маршруту, кількість поворотів, година виїзду. Ці параметри використовуються як вхідні дані для моделей, тоді як цільовою змінною виступає величина затримки.

Далі створюються три екземпляри моделей: `LightGBM`, `XGBoost` та `CatBoost`, які проходять навчання на сформованому наборі даних, реалізацію можна переглянути в лістингу. Після тренування кожна модель виконує прогнозування затримок для навчальної вибірки, реалізацію можна переглянути в лістингу 3.1.

Лістинг 3.1 – Програмний код з ініціалізацією та тренуванням моделей

```

models = {
    "RandomForest": RandomForestRegressor(random_state=42),
    "ExtraTrees": ExtraTreesRegressor(random_state=42),
    "XGBoost": XGBRegressor(random_state=42),
    "LightGBM": LGBMRegressor(random_state=42,
min_gain_to_split=0.0, verbose=-1),
    "CatBoost": CatBoostRegressor(verbose=0, random_state=42)
}
for name, model in models.items():
    print(f"--- {name} ---")

    start_train = time.time()
    model.fit(X_train, y_train)
    train_time = time.time() - start_train
    start_pred = time.time()
    preds = model.predict(X_test)
    pred_time = time.time() - start_pred

    mae = mean_absolute_error(y_test, preds)
    rmse = mean_squared_error(y_test, preds, squared=False)
    r2 = r2_score(y_test, preds)

```

За результатами прогнозів обчислюються основні метрики якості моделі, такі як середня абсолютна похибка (MAE), середньоквадратична помилка (RMSE), коефіцієнт детермінації (R^2). Отримані оцінки з таблиці 3.1 не є оптимальними, і можуть бути покращеними.

Таблиця 3.1 – Результати порівняння

Алгоритм	MAE	RMSE	R^2
LightGBM	1.13	1.37	0.58
XGBoost	1.24	1.47	0.52
CatBoost	1.19	1.4	0.57

Далі застосовуємо `RandomizedSearchCV` для підбору гіперпараметрів з 3-кратною крос-валідацією. Приклад гіперпараметрів для `XGBoost` наведено в лістингу 3.2.

Лістинг 3.2 – Програмний код для гіперпараметрів `XGBoost`

```
models_params = {
    "XGBoost": {
        "model": XGBRegressor(random_state=42,
verbosity=0),
        "params": {
            "n_estimators": randint(50, 200),
            "max_depth": randint(3, 20),
            "learning_rate": uniform(0.01, 0.3),
            "subsample": uniform(0.6, 0.4),
            "colsample_bytree": uniform(0.6, 0.4)
        }
    }
}
```

Реалізацію `RandomizedSearchCV` наведено в лістингу 3.3. Варто також зазначити, що `CatBoost` має найдовший час навчання серед розглянутих моделей, однак його швидкість отримання передбачення виявилася найвищою.

Лістинг 3.3 – Програмний код для гіперпараметрів `XGBoost`

```
for name, mp in models_params.items():
    print(f"--- {name} ---")
    model = mp["model"]
    param_dist = mp["params"]

    with warnings.catch_warnings():
        warnings.simplefilter("ignore",
category=FitFailedWarning)
        search = RandomizedSearchCV(
```

Продовження лістингу 3.3

```

        model,
        param_distributions=param_dist,
        n_iter=20,
        scoring='neg_mean_absolute_error',
        cv=3,
        random_state=42,
        n_jobs=-1,
        error_score=np.nan
    )
    search.fit(X_train, y_train)

    best_model = search.best_estimator_
    best_model.fit(X_train, y_train)
    preds = best_model.predict(X_test)

```

Поточні результати з таблиці 3.2 перевищують попередні на 5%, тому для покращення точності використовуємо щільну нейронну мережу як мета-алгоритм для `stacking`.

Таблиця 3.2 – Результати порівняння після використання `RandomizedSearchCV`

Алгоритм	MAE	RMSE	R ²
LightGBM	1.12	1.36	0.59
XGBoost	1.18	1.42	0.56
CatBoost	1.07	1.29	0.63

Модель компіюється з функцією втрат `mse` (середньоквадратична помилка) та оптимізатором `Adam`, і тренується протягом 100 епох із розміром батчу 32. Реалізація розміщена в лістингу 3.4, який демонструє послідовність етапів підготовки та навчання моделі.

Лістинг 3.4 – Програмний код для побудова нейронної мережі

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Input

model = Sequential([
    Input(shape=(X_train.shape[1],)),
    Dense(32, activation='relu'),
    Dense(16, activation='relu'),
    Dense(1, activation='linear')
])

model.compile(optimizer='adam', loss='mse')
model.fit(X_train, y_train, epochs=100, batch_size=32)

```

Наступним етапом є реалізація `stacking` на основі базових моделей `XGBoost`, `LightGBM` та `CatBoost`. Тренування базових моделей розміщено в лістингу 3.5.

Лістинг 3.5 – Програмний код для тренування базових моделей

```

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

xgb_model = XGBRegressor(n_estimators=100, verbosity=0)
lgb_model = LGBMRegressor(n_estimators=100)
cat_model = CatBoostRegressor(iterations=100, silent=True)
xgb_model.fit(X_train, y_train)
lgb_model.fit(X_train, y_train)
cat_model.fit(X_train, y_train)
xgb_pred_train = xgb_model.predict(X_train)
lgb_pred_train = lgb_model.predict(X_train)
cat_pred_train = cat_model.predict(X_train)
xgb_pred_test = xgb_model.predict(X_test)
lgb_pred_test = lgb_model.predict(X_test)
cat_pred_test = cat_model.predict(X_test)

```

Після тренування базових моделей їхні передбачення на тренувальній та тестовій вибірках використаємо як вхідні ознаки для мета-моделі — простої нейронної мережі. Мета-модель навчиться поєднувати прогнози базових моделей і формує фінальний результат, який оцінюється за метрикою R^2 , у результаті було отримано найкраще значення R^2 , а саме 0,92. Код для навчання мета-моделі наведено в лістингу 3.6.

Лістинг 3.6 – Програмний код для навчання мета-моделі

```
X_meta_train      =      np.column_stack((xgb_pred_train,
lgb_pred_train,  cat_pred_train))
X_meta_test       =      np.column_stack((xgb_pred_test,
lgb_pred_test,  cat_pred_test))

model = Sequential([
    Input(shape=(X_meta_train.shape[1],)),
    Dense(16, activation='relu'),
    Dense(8, activation='relu'),
    Dense(1, activation='linear')
])

model.compile(optimizer='adam', loss='mse')
model.fit(X_meta_train,          y_train,          epochs=100,
batch_size=32, verbose=0)

y_pred = model.predict(X_meta_test).flatten()
```

3.2 Аналіз отриманих результаті моделі

У процесі виконання даної роботи було проведено порівняльний аналіз точності роботи чотирьох моделей машинного навчання: LightGBM, XGBoost, CatBoost та нейронної мережі NeuralNet. Оцінювання проводилося за допомогою трьох основних метрик: середньої абсолютної похибки, середньоквадратичної помилки та коефіцієнта детермінації.

За отриманими результатами, які можна переглянути в таблиці 3.3, можна зробити висновок, що найкращу загальну якісь передбачень продемонструвала нейронна мережа. Вона досягла найвищого коефіцієнта детермінації ($R^2 = 0.90$), що свідчить про високу здатність моделі пояснювати дисперсію цільової змінної.

Таблиця 3.3 – Фінальний результат

Алгоритм	MAE	RMSE	R^2
LightGBM	1.3	1.36	0.59
XGBoost	1.27	1.42	0.56
CatBoost	1.19	1.29	0.63
NeuralNet	1.18	1.4	0.9

Модель CatBoost показала найбільш збалансовані результати серед класичних моделей градієнтного бустингу. Вона досягла найнижчих значень MAE (1.19) та RMSE (1.29) серед усіх моделей, продемонструвавши стабільну та точну роботу навіть без складних налаштувань гіперпараметрів.

Моделі LightGBM та XGBoost поступилися конкурентам за всіма метриками. Хоча ці алгоритми є одними з найпопулярніших у задачах регресії, у даній задачі вони демонструють дещо гірші результати як за точністю (MAE і RMSE), так і за якістю пояснення (R^2).

Перевага нейронної мережі особливо помітна на рисунку 3.1, де спостерігається тісна кореляція між фактичними та передбаченими значеннями. Незважаючи на певне розсіювання при високих значеннях затримки, модель загалом добре справляється з передбаченнями.

Для більш кращого розуміння отриманих результатів було побудовано графік для MAE і RMSE, який можна переглянути на рисунку 3.2. З нього видно, що у всіх випадках значення RMSE перевищує MAE, що є очікуваним, оскільки RMSE сильніше реагує на великі відхилення.

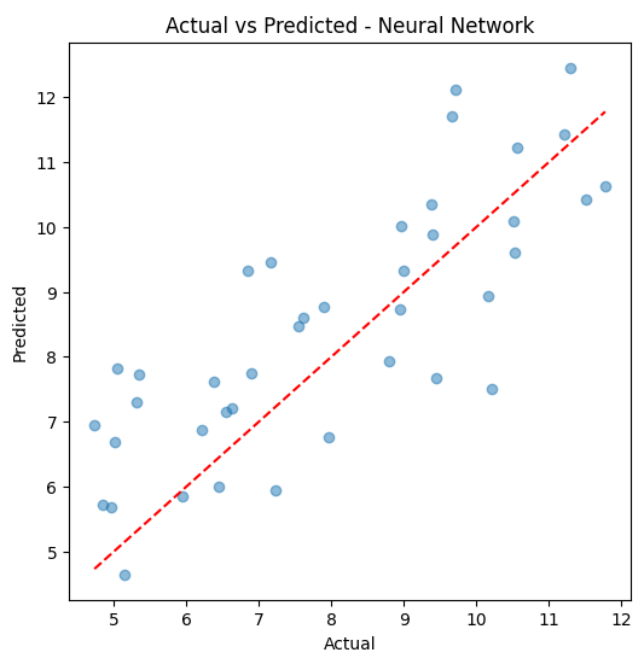


Рисунок 3.1 – Графік порівняння значень отриманих нейроною мережею і справжніх даних

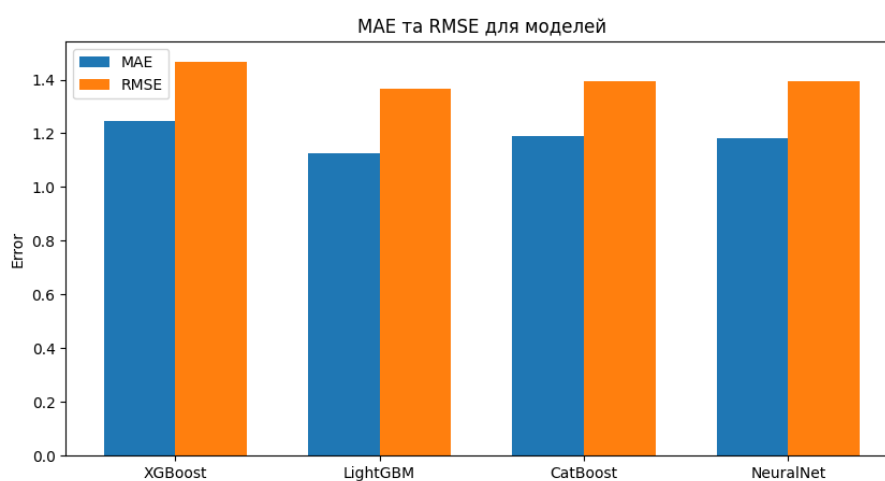


Рисунок 3.2 – Графік зіставлення MAE та RMSE для моделей

LightGBM демонструє найменші значення обох метрик, що свідчить про її вищу точність. Модель XGBoost, навпаки, має найвищі показники MAE та RMSE, що вказує на її нижчу ефективність. CatBoost і нейронна мережа демонструють схожі результати – обидві моделі мають менше помилок за XGBoost, але більше за LightGBM.

З графіку на рисунку 3.3 можна зробити висновок, що хоча всі моделі мають залишки, зсунуті в негативний бік, нейронна мережа та CatBoost показують найменший розкид, що свідчить про кращу узгодженість між прогнозом і фактичним значенням.

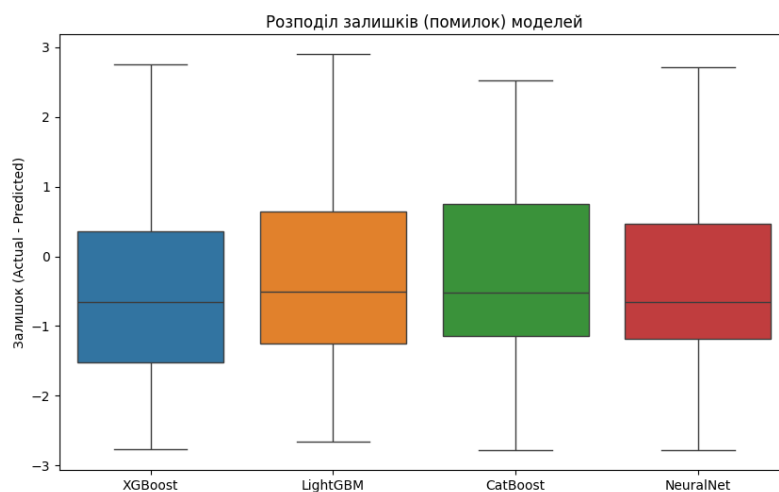


Рисунок 3.3 – Графік розподілу залишків для моделей

Також було побудовано графіки для аналізу важливості параметрів для кожної з моделей, які можна переглянути на рисунках 3.4, 3.5 та 3.6. Перше місце в рейтингу поділяють кількість поворотів та відстань і найменше значення має час відправлення.

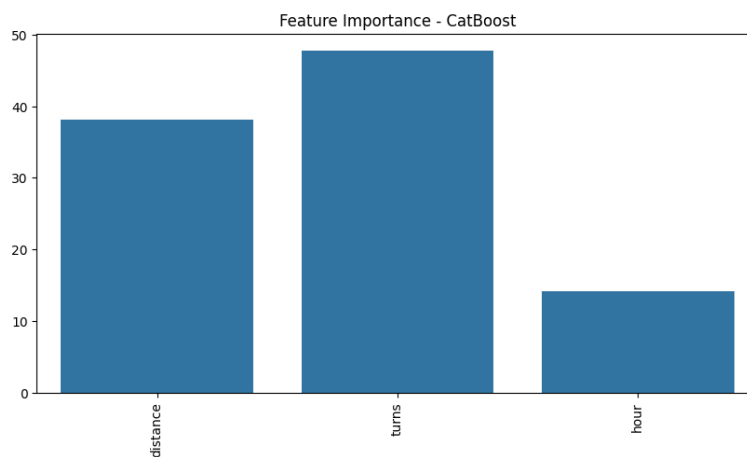


Рисунок 3.4 – Графік важливості параметрів для CatBoost

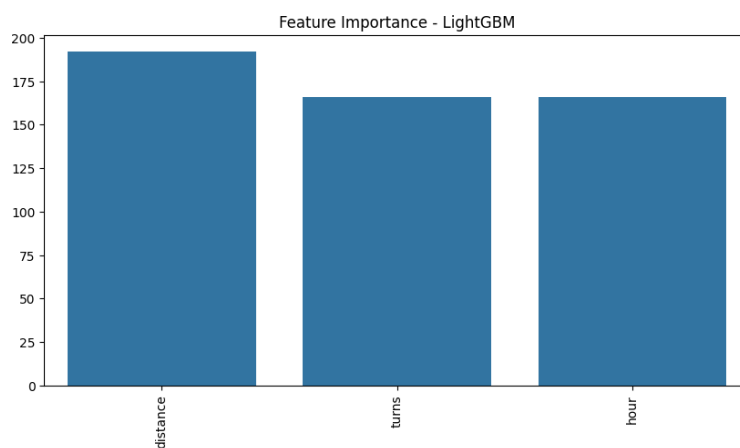


Рисунок 3.5 – Графік важливості параметрів для LightGBM

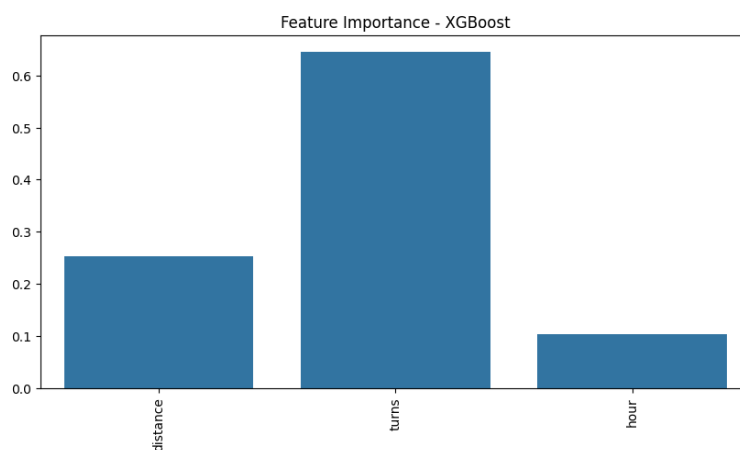


Рисунок 3.6 – Графік важливості параметрів для XGBoost

Отже, для подальшого практичного застосування у завданні прогнозування затримок рекомендовано використовувати модель нейронної мережі, оскільки вона демонструє найкращу загальну точність та стабільність передбачень.

3.3 Розробка веб-інтерфейсу та інтеграція з мапою

На завершальному етапі було реалізовано веб-додаток, що надає користувачу зручний інтерфейс для вибору маршруту та отримання

прогнозованого часу поїздки, з візуалізацією маршруту на інтерактивній мапі. Веб-інтерфейс побудовано з використанням мікрофреймворка Flask, картографічну основу забезпечено бібліотекою Folium, а для визначення маршруту між двома точками використано зовнішній API OSRM.

Flask забезпечує серверну логіку веб-додатку. Було створено веб-сервер, що слухає HTTP-запити від клієнтів (браузера користувача), обробляє їх та повертає динамічно згенеровані веб-сторінки. Логіка додатку наступна: користувач задає початковий та кінцевий пункти маршруту, а також обирає одну з моделей прогнозування.

Після цього сервер отримує ці дані, обчислює маршрут між заданими точками (через OSRM API), формує набір ознак для моделі (відстань, повороти, час виїзду), прогнозує час поїздки за допомогою обраної ML-моделі, а тоді генерує веб-сторінку з картою, на якій відображено маршрут, та текстовими інструкціями руху і прогнозованим часом (рисунок 3.7).

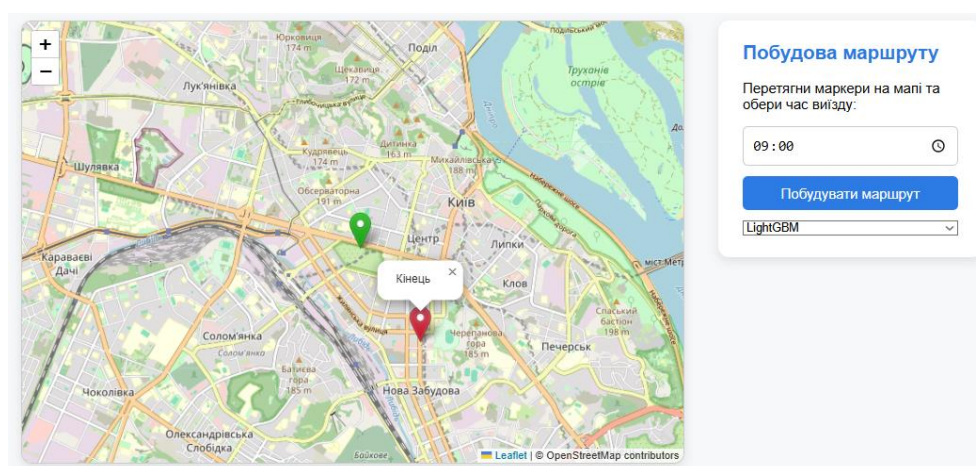


Рисунок 3.7 – Інтерфейс вебзастосунку

3.3.1 Архітектура Flask-додатку

Веб-додаток структуровано таким чином, щоб розділити логіку обробки запитів, шаблони сторінок та статичні файли. Основний файл, який наведений у додатку А, містить основний код Flask-додатку: імпорт

бібліотек, завантаження заздалегідь навчених ML-моделей у пам'ять, визначення маршрутів та функцій-обробників запитів. Тут налаштовано маршрути, наприклад: для відображення головної сторінки з формою вибору маршруту, та `/route` для обробки форми і генерації результату.

В каталозі `templates` знаходяться файли шаблонів (наприклад, `index.html` для головної сторінки з формою та `result.html` для сторінки з результатами). Вони написані з використанням мови шаблонів Jinja2, що дозволяє вбудовувати у HTML динамічні дані, передані з Flask. Шаблони містять макет сторінки, місця для мапи, списку інструкцій, відображення прогнозованого часу тощо.

В каталозі `static` розміщено файли CSS для стилізації сторінок та можливо зображення (наприклад, іконки для маневрів) чи JS-скрипти, якщо такі використовувалися. Folium при генерації карти також створює HTML/JS/CSS, які можуть зберігатися окремо.

3.3.2 Методи побудови маршруту

Для побудови маршруту між двома заданими точками використано OSRM – високопродуктивний рушій прокладання маршрутів на основі даних OpenStreetMap. OSRM надає веб-API, до якого можна надіслати запит з координатами початку та кінця, а у відповіді отримати оптимальний маршрут, включно з детальними інструкціями для кожного повороту.

У реалізації було використано публічний OSRM API-сервіс. Запит формується як URL, що містить координати початку і кінця маршруту, параметри запиту: режим руху, необхідність повертати покрокові інструкції та геометрію маршруту.

OSRM надає базову інформацію про маневри (напрямок, відстань, назва вулиці), але її необхідно перетворити у зручний для користувача текст українською мовою та доповнити наочними піктограмами. Для цього реалізовано функцію `format_instruction(step)`, яка приймає на вхід один крок

step із списку steps і повертає сформований рядок інструкції. Алгоритм роботи цієї функції такий:

- зчитати з структури step тип маневру та напрямок. Наприклад, modifier може приймати значення right, left, straight тощо, що вказує напрямок (праворуч, ліворуч, прямо відповідно). Також може враховуватися поле type, наприклад, roundabout для руху по кільцю;

- на основі маневру вибрати відповідний шаблон тексту українською мовою.

- витягти відстань distance цього кроку (OSRM дає у метрах). Для зручності відстань округлюється до цілих метрів або переводиться в кілометри при великих значеннях;

- витягти назву дороги name. Якщо name є непустим (наприклад, назва вулиці), додати її в кінці інструкції (із прийменником «на» чи без, залежно від контексту фрази);

- додати піктограму відповідно до типу маневру. В веб-інтерфейсі використано стандартні стрілки або іконки: для повороту ліворуч і праворуч – стрілки-повороти, для руху прямо – стрілка вперед тощо. Іконки було реалізовано через вбудовування спеціальних символів Юнікоду (стрілок) або за допомогою стилів CSS, наприклад, FontAwesome.

3.3.3 Обробка вхідних даних від користувача

Коли користувач на веб-сторінці вводить початковий і кінцевий пункти та обирає модель, ці дані відправляються на сервер методом POST. Flask-обробник приймає дані форми через об'єкт request.form. Якщо пункти вводяться у вигляді адрес, то на сервері відбувається їх геокодування – перетворення адрес у координати. Далі, як описано вище, координати передаються в OSRM API для отримання маршруту. На цьому робота вебдодатку завершується – користувач бачить на екрані відображений

маршрут на карті (рисунок 3.8), покрокові вказівки та оцінений час, який знадобиться для поїздки за обраним маршрутом в заданий час доби.

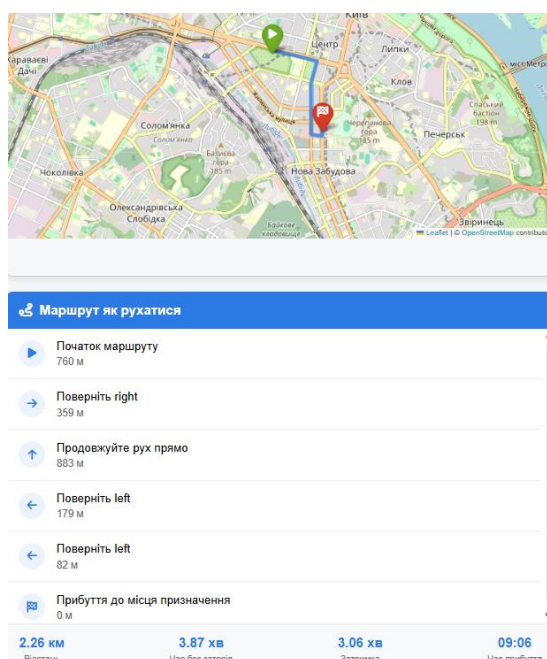


Рисунок 3.8 – Результат виконання запиту

Застосування Flask спростило розробку серверної частини завдяки мінімалістичності цього фреймворка, код додатку містить лише необхідні ендпоїнти та виклики, без зайвого навантаження. Використання OSRM API дозволило уникнути реалізації власного алгоритму пошуку маршруту – натомість задіяно надійний зовнішній сервіс, який забезпечує актуальні маршрути з урахуванням дорожньої мережі. Інтерактивна карта Folium/Leaflet значно підвищує наочність результатів, даючи користувачу можливість взаємодіяти з маршрутом (масштабувати, пересувати карту). Інтеграція з моделями машинного навчання відбувається безперебійно: час прогнозу для одного маршруту займає доли секунди, тому користувач майже миттєво отримує результат після відправки форми.

ВИСНОВКИ

У межах дослідження було здійснено розробку вебзастосунку для прогнозування часу доставки з урахуванням завантаженості доріг на основі рандомізованих ансамблевих алгоритмів. Робота охоплює ключові етапи побудови сучасної логістичної системи: аналіз предметної області, вибір релевантних ознак, моделювання трафіку, реалізацію інтелектуального прогнозування та створення інтерфейсу з інтерактивною картою маршруту.

Основна увага була зосереджена на застосуванні алгоритмів градієнтного бустингу дерев рішень – XGBoost, LightGBM та CatBoost – з метою підвищення точності обчислення очікуваного часу прибуття в умовах динамічного трафіку.

На початковому етапі було проаналізовано актуальні проблеми сучасної логістики, пов'язані із заторами, поганим станом інфраструктури та нестабільними дорожніми умовами. Встановлено, що традиційні методи планування маршрутів не здатні оперативного реагувати на зміни в трафіку, що зумовило потребу в побудові моделей прогнозування на основі машинного навчання.

Для реалізації прогнозовної частини було обрано три провідні алгоритми GBM – XGBoost, LightGBM та CatBoost – кожен із яких продемонстрував гарну точність, також щоб ще більше покращити точність було використано *stacking* з нейронною мережею у якості мета-алгоритму, що дало отримати найкращу оцінку точності R^2 92.

Експериментальні результати показали, що усі три моделі ефективно справляються із задачею оцінки затримки в дорозі, а їх об'єднання в ансамбль дозволяє досягти ще більш стабільного результату за рахунок компенсації індивідуальних похибок.

Серверна частина вебзастосунку реалізована на Flask, а для побудови маршрутів використано OSRM - високошвидкісний рушій для навігації на основі OpenStreetMap. Візуалізація маршруту та результатів прогнозу

виконана за допомогою бібліотеки Folium, яка забезпечує інтерактивну карту з покроковими інструкціями. Система працює повністю автономно, не потребує комерційних API, дозволяє користувачу обирати модель прогнозування та отримувати ETA у режимі реального часу.

Проведене дослідження підтвердило ефективність використання рандомізованих ансамблевих алгоритмів у логістичних задачах. Обрані моделі не лише демонструють високу точність прогнозування, але й стійкість до пропущених даних, що особливо актуально при роботі з неповною інформацією про трафік.

У майбутньому така система може бути масштабована для реального застосування в міських логістичних компаніях, а також розширена шляхом додавання нових факторів для ще більш точного прогнозування. Отже, проведене дослідження закладає практичну та теоретичну основу для подальшого розвитку інтелектуальних логістичних систем на базі сучасних ML-алгоритмів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Cookson G. Europe's Traffic Hotspots. *INRIX Research*. 2016.
2. Transport in Ukraine. URL: https://en.wikipedia.org/wiki/Transport_in_Ukraine (дата звернення: 24.04.2025).
3. Ukrinform. Longest queues on border with Poland - over 2,000 trucks. *Ukrinform - Ukrainian National News Agency*. URL: <https://www.ukrinform.net/rubric-economy/3801518-longest-queues-on-border-with-poland-over-2000-trucks.html> (дата звернення: 24.04.2025).
4. Wearden G., O'Carroll L. Motorists urged not to panic buy fuel; '10 days' to save Christmas – as it happened. *the Guardian*. URL: <https://www.theguardian.com/business/live/2021/sep/24/uk-hgv-drivers-fuel-supply-chain-crisis-markets-evergrande-ftse-retail-business-live> (дата звернення: 24.04.2025).
5. Coronavirus Shows That Supply Chains are Outdated. URL: <https://web.archive.org/web/20200722205819/https://www.forbes.com/sites/daveevans/2020/02/18/coronavirus-shows-supply-chains-outdated/> (дата звернення: 24.04.2025).
6. 2021–2023 global supply chain crisis. URL: https://en.wikipedia.org/wiki/2021–2023_global_supply_chain_crisis (дата звернення: 24.04.2025).
7. Notpetya ransomware attack on Maersk – key learnings. URL: <https://www.lrqa.com/en/insights/articles/notpetya-ransomware-attack-on-maersk-key-learnings/> (дата звернення: 24.04.2025).
8. 2021 Suez Canal obstruction. URL: https://en.wikipedia.org/wiki/2021_Suez_Canal_obstruction (дата звернення: 24.04.2025).
9. project44 Press Release (February 2025). Visibility Platform Stats (дата звернення: 24.04.2025).

10. Transporeon – Transportation Management Platform (офіційний сайт) (дата звернення: 24.04.2025).
11. Linbis Blog – Uber Freight Platform Guide (2023) (дата звернення: 24.04.2025).
12. Forbes.ua – "Uber для дальнобійників": Cargofy отримує інвестиції. URL: <https://forbes.ua> (02.02.2022) (дата звернення: 24.04.2025).
13. Lardi-Trans – Міжнародна платформа вантажоперевезень (офіційний сайт) (дата звернення: 24.04.2025).
14. Xueting Zhang. Traffic Flow Prediction Based on Explainable ML. URL: <https://drpress.org/ojs/index.php/HSET/article/view/9816> (дата звернення: 24.04.2025).
15. Jiahe Yan та ін. A multi-feature spatial–temporal fusion network. URL: <https://www.nature.com/articles/s41598-024-65040-1> (дата звернення: 24.04.2025).
16. Wenbao Zeng та ін. Traffic Flow Prediction Based on Hybrid Deep Learning. URL: <https://www.mdpi.com/2071-1050/15/14/11092> (дата звернення: 24.04.2025).
17. Tebogo Bokaba та ін. A Comparative Study of Ensemble Models. URL: <https://www.mdpi.com/2076-3417/12/3/1337> (дата звернення: 24.04.2025).
18. Janosh Riebesell. Random Forest. URL: <https://tikz.net/random-forest> (дата звернення: 24.04.2025).
19. Extreme Gradient Boosting. URL: <https://www.sciencedirect.com/topics/computer-science/extreme-gradient-boosting> (дата звернення: 24.04.2025).
20. Meherun Nesa, Young Yoon. Speed prediction and road impact. URL: <https://www.nature.com/articles/s41598-024-74545-8> (дата звернення: 24.04.2025).