

УДК 004.85:[004.738.5:339]

ДОСЛІДЖЕННЯ МЕТОДІВ ІНТЕГРАЦІЇ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ У РОЗПОДІЛЕНІ ВЕБСИСТЕМИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ДЛЯ ПРОГНОЗУВАННЯ ПОПИТУ НА ТОВАРИ

Морочковський О.М.

Науковий керівник – к.т.н., доц. Імангулова З.А.

Харківський національний університет радіоелектроніки, каф. СТ,

м. Харків, Україна тел.: +38(066)-889-68-55,

e-mail: oleksii.morochkovskyi@nure.ua

The work is focused on researching methods for integrating machine learning models into distributed e-commerce web systems to forecast product demand. It examines the architectural approaches for deploying ML models within a microservices infrastructure, ensuring high scalability and low latency. The research highlights the importance of data preprocessing and real-time synchronization between distributed components. The integration of forecasting models allows e-commerce platforms to optimize inventory management, reduce operational costs, and improve customer satisfaction. The study also discusses the selection of appropriate frameworks and communication protocols between the web system and the ML inference engine. Special attention is paid to Clean Architecture and design patterns like Repository and Dependency Injection to ensure system flexibility and testability.

В сучасних умовах, спричинених глобальними змінами у логістиці та соціально-економічними викликами, роль онлайн-сервісів та освітніх чи торговельних платформ значно зросла. Для систем електронної комерції критично важливим стає не просто надання інтерфейсу для покупок, а інтелектуальна підтримка бізнес-процесів. Прогнозування попиту дозволяє максимально покрити потреби споживачів та мінімізувати збитки від затоварення або дефіциту. Дане дослідження присвячене розробці інформаційної системи, що інтегрує методи машинного навчання (ML) для аналізу великих масивів даних у розподіленому середовищі.

Питання побудови сервісно-орієнтованих архітектур (SOA) та організації баз даних є фундаментальними для розподілених систем. Сучасні підходи вимагають не лише зберігання даних, а й їхньої оперативної обробки для отримання прогнозів у реальному часі. Використання технологій .NET дозволяє ефективно масштабувати систему, забезпечуючи стабільну роботу при великих обсягах інформації.

Для забезпечення масштабованості та легкої підтримки системи було обрано Clean Architecture. Це дозволяє розбити складну програму на модулі, що працюють узгоджено та ефективно. Структура проекту включає:

- Рівень Domain: містить сутності та бізнес-правила, що не залежать від зовнішніх сервісів чи ML-моделей.

- Рівень Application: визначає сценарії використання (Use Cases), де ініціюється виклик прогнозів.

- Рівень Infrastructure: реалізує доступ до баз даних (MSSQL Server) та інтеграцію з ML-моделями через API або бібліотеки.

Clean Architecture вагомо спрощує тестування, подальше розширення застосунку та його підтримку в майбутньому.

Інтеграція методів машинного навчання. Процес інтеграції ML-моделей у вебсистему, реалізовану на ASP.NET та мові C#, вимагає вирішення проблеми взаємодії різних технологічних стеків. В рамках дослідження виділено такі методи:

- Використання ML.NET: дозволяє розробляти та впроваджувати моделі безпосередньо в екосистемі .NET. Це усуває потребу в складних механізмах серіалізації даних між різними мовами програмування.

- Мікросервісна інтеграція (MaaS): модель розгортається як окремих контейнер (наприклад, Docker з Python/FastAPI), а вебсистема звертається до неї за прогнозом. Це забезпечує ізоляцію та можливість незалежного масштабування обчислювальних ресурсів.

Для роботи з базою даних MSSQL Server обрано Entity Framework, що гарантує високу швидкість розробки та зручність мапінгу даних для навчання моделей.

При розробці було використано Repository pattern для розділення бізнес-логіки та деталей роботи з базою даних. Він надає абстракцію, що полегшує зміну джерела даних для ML-моделей без впливу на інтерфейс користувача.

Патерн Dependency Injection (DI) дозволяє створювати залежності між об'єктами під час виконання. Це робить систему гнучкою: можна легко перемикається між різними версіями моделей прогнозування або використовувати "заглушки" (mocks) під час тестування модуля прогнозування.

Задача прогнозування попиту D для певної категорії товарів зводиться до знаходження апроксимуючої функції F , яка на основі вектора входних ознак X (історія продажів, сезонність, ціна) видає прогноз:

$$D_{t+1} = F(X_t, W) + n$$

де W — вагові коефіцієнти моделі, що оптимізуються в процесі навчання, а n — параметр, що враховує випадкові коливання ринку. Розподілена природа системи дозволяє проводити обчислення $F(X)$ на виділених серверах, не знижуючи швидкість відгуку основного інтерфейсу користувача.

Розроблена архітектура інформаційної системи на базі .NET із застосуванням Clean Architecture та паттернів проектування є ефективною відповіддю на сучасні виклики електронної комерції. Вона забезпечує стабільну роботу, високу швидкість обробки запитів та зручну інтеграцію інтелектуальних модулів прогнозування.

Список використаних джерел:

1. Пасічник В.В. Сервісно-орієнтована архітектура: основи, принципи, технології. Львів: Магнолія, 2016. 384 с.
2. Пасічник В.В., Резніченко В.А. Організація баз даних та знань. Київ: BHV, 2006. 384 с.
3. Fowler M. Patterns of Enterprise Application Architecture. Addison-Wesley Professional, 2002. 533 p.
4. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2017. 432 p.
5. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Київ, 2016. 17 с.