 an/uet5.jpg
ĐẠI HỌC CÔNG NGHỆ - ĐHQG HÀ NỘI
KHOA TRÍ TUỆ NHÂN TẠO

BÁO CÁO BÀI TẬP LỚN

Đề tài: Optimize Delivery Routes & Inventory Management

Môn học: Kỹ thuật và công nghệ dữ liệu lớn cho Trí tuệ nhân tạo

Thành viên nhóm LQT

Nông Phi Long – 23020398

Chu Thanh Tùng – 23020431

Phạm Quân – 23020418

Mở đầu

Trong bối cảnh thương mại điện tử và logistics phát triển mạnh mẽ, các doanh nghiệp phải đối mặt với áp lực tối ưu hóa quy trình vận chuyển và quản lý tồn kho để giảm chi phí, rút ngắn thời gian giao hàng và nâng cao chất lượng dịch vụ. Tuy nhiên, sự gia tăng nhanh chóng của số lượng đơn hàng, biến động nhu cầu theo thời gian và hạn chế về nguồn lực khiến việc vận hành thủ công trở nên kém hiệu quả và dễ xảy ra sai sót.

Trước thách thức đó, việc ứng dụng các phương pháp phân tích dữ liệu và thuật toán tối ưu hóa vào **Delivery Route Optimization** và **Inventory Management** trở thành yếu tố quan trọng giúp doanh nghiệp đạt được hiệu quả vận hành vượt trội.

Dự án được thực hiện nhằm xây dựng mô hình tối ưu hóa tuyến đường giao hàng, đồng thời hỗ trợ quản lý tồn kho thông minh dựa trên dữ liệu thực tế. Nhóm sử dụng các công cụ như HDFS, NFS, OR-Tools, Deep Learning và những kỹ thuật xử lý dữ liệu lớn nhằm chứng minh rằng tự động hóa có thể giúp giảm chi phí vận chuyển, rút ngắn thời gian giao hàng, hạn chế thiếu hoặc thừa tồn kho và nâng cao hiệu quả toàn bộ chuỗi cung ứng.

Mục lục

1 Giới thiệu

1.1 Tại sao tối ưu tuyến giao hàng và quản lý tồn kho lại quan trọng?

Trong bối cảnh thương mại điện tử và logistics phát triển mạnh mẽ, các doanh nghiệp phải đối mặt với áp lực tối ưu hóa quy trình vận chuyển và quản lý tồn kho. Việc ứng dụng các phương pháp phân tích dữ liệu và thuật toán tối ưu hóa vào hai lĩnh vực này giúp doanh nghiệp:

- Giảm chi phí vận chuyển.
- Rút ngắn thời gian giao hàng.
- Hạn chế sai sót trong quy trình.
- Nâng cao hiệu quả toàn bộ chuỗi cung ứng.

1.2 Mục tiêu của dự án

- Xây dựng mô hình tối ưu hóa tuyến giao hàng.
- Dự báo tồn kho và nhu cầu của cửa hàng.
- Tối ưu chi phí vận chuyển.
- Cải thiện chất lượng dịch vụ và hiệu quả vận hành.

1.3 Phạm vi của dự án

Dự án tập trung vào hai phần chính:

1. **Delivery Route Optimization:** tối ưu lộ trình cho các phương tiện vận chuyển.
2. **Inventory Management:** phân tích dữ liệu tồn kho và dự báo nhu cầu.

1.4 Tổng kết chương 1

Việc áp dụng Big Data cho phép thu thập, lưu trữ và phân tích một lượng lớn dữ liệu phát sinh từ hoạt động vận chuyển và tồn kho theo thời gian thực. Bên cạnh đó, các mô hình học sâu hỗ trợ trích xuất đặc trưng, dự báo nhu cầu và nhận diện các mẫu hành vi phức tạp mà các phương pháp truyền thống thường khó xử lý. Chúng đã chứng minh được hiệu quả rõ rệt trong cả hai khía cạnh: tối ưu hóa vận chuyển và quản lý tồn kho. Những công nghệ này giúp doanh nghiệp giảm chi phí, rút ngắn thời gian xử lý và nâng cao hiệu suất toàn chuỗi cung ứng, tạo tiền đề vững chắc cho các chương tiếp theo của dự án.

2 Thu thập và tiền xử lý dữ liệu

2.1 Nguồn dữ liệu

- **Optimize delivery routes:**

<https://huggingface.co/datasets/Cainiao-AI/LaDe>

- **Inventory management:**

<https://data.mendeley.com/datasets/mgzvngzng2/1>

Bộ dữ liệu gồm thông tin giao hàng, tuyến đường, bản đồ địa lý, dữ liệu sản phẩm, cửa hàng và lịch sử bán hàng.

2.2 Tiền xử lý dữ liệu

2.2.1 Xử lý dữ liệu thiếu và lỗi

Bộ dữ liệu từ LaDe và Bangladesh có rất ít lỗi, nên có thể import trực tiếp vào mô hình.

Một số ảnh minh họa file đã import:

big_data/b1.png

big_data/b2.png

big_data/b3.png

2.2.2 Upload, đọc và lưu trữ trên HDFS/NFS

Dữ liệu được nhập và lưu trữ trên hệ thống phân tán HDFS hoặc NFS để đảm bảo khả năng mở rộng và truy cập nhanh chóng.

Dữ liệu được lưu trữ trên hệ thống phân tán qua:

- Dask để đọc/ghi dữ liệu HDFS/NFS.
- Phải cấu trúc thư mục rõ ràng để quản lý.
- Upload dữ liệu:
 - Code cho việc upload dữ liệu lên HDFS/NFS:
BIGDATA/upload_to_hdfs.py
 - cấu hình shell script để upload dữ liệu lên HDFS/NFS:
BIGDATA/upload_to_hdfs.sh

Cấu trúc thư mục HDFS sau khi upload:

```
|-- [DIR] Datapack
    |-- [DIR] Delivery
        |-- (4.52 MB) delivery_jl.csv
        |-- (30.07 MB) delivery_yt.csv
        |-- (138.35 MB) delivery_cq.csv
```

```
    |-- (217.50 MB) delivery_sh.csv
    |-- (273.60 MB) delivery_hz.csv
|-- [DIR] Inventory
    |-- (4.26 KB) product_info.csv
    |-- (84.77 MB) product_target_for_shop.csv
    |-- (111.55 MB) shop_info_with_geo.csv
|-- [DIR] Pickup
    |-- (41.77 MB) pickup_jl.csv
    |-- (181.38 MB) pickup_yt.csv
    |-- (181.70 MB) pickup_cq.csv
    |-- (217.76 MB) pickup_sh.csv
    |-- (320.26 MB) pickup_hz.csv
|-- [DIR] Roadmap
    |-- (221.01 MB) roads.csv
|-- [DIR] output
    |-- (1.35 MB) combined_all_data.csv
```

2.3 Tổng kết chương 2

Chương 2 đã trình bày quá trình thu thập dữ liệu từ các nguồn khác nhau và thực hiện các bước tiền xử lý cần thiết để đảm bảo dữ liệu sẵn sàng cho mô hình. Dữ liệu được làm sạch, kiểm tra lỗi và tổ chức lại trước khi được lưu trữ trên hệ thống phân tán HDFS/NFS.

3 Các phương pháp được sử dụng

3.1 Hệ thống Big Data và xử lý phân tán

- **Hadoop (HDFS, YARN):** là hệ thống phổ biến trong lưu trữ và xử lý dữ liệu lớn, cung cấp khả năng mở rộng, độ tin cậy cao và hỗ trợ xử lý song song thông qua mô hình MapReduce. Hadoop cũng sở

hữu khả năng chịu lỗi tốt, đảm bảo dữ liệu luôn sẵn sàng trong môi trường phân tán.

- **Kiến trúc HDFS** bao gồm các thành phần chính:
 - *NameNode*: quản lý hệ thống tệp và metadata.
 - *DataNode*: lưu trữ các khối dữ liệu.
 - *Secondary NameNode*: hỗ trợ sao lưu metadata và giảm tải cho NameNode.
- **NFS (Network File System)**: cho phép chia sẻ và truy cập tệp tin qua mạng, giúp các hệ thống và ứng dụng làm việc với dữ liệu một cách linh hoạt và hiệu quả.
- **Thiết lập NFS**: cấu hình và triển khai hệ thống NFS được thực hiện thông qua:
 - `BIGDATA/set_up_NFS.py`
 - `BIGDATA/NFS.md`
 - `BIGDATA/mount_nfs_archives.sh`
- **Dask**: là thư viện Python mã nguồn mở hỗ trợ xử lý dữ liệu phân tán và song song, cho phép mở rộng quy mô xử lý vượt quá giới hạn bộ nhớ của một máy đơn lẻ.

3.2 Xử lý dữ liệu và Deep Learning

- **Pandas, NumPy**: được sử dụng trong quá trình tiền xử lý và phân tích dữ liệu, cung cấp các công cụ mạnh mẽ để thao tác với dữ liệu dạng bảng và mảng, hỗ trợ làm sạch, chuẩn hóa và biến đổi dữ liệu.
- **Scikit-learn**: hỗ trợ các bước chuẩn hóa dữ liệu, mã hóa biến, chia tập dữ liệu và xây dựng các mô hình máy học cơ bản trước khi đưa vào mô hình học sâu.

- **TensorFlow / Keras:** dùng để xây dựng và huấn luyện các mô hình deep learning phục vụ bài toán dự đoán tuyến đường, thời gian giao hàng (ETA) và các tác vụ phân tích nâng cao khác.
- **OR-Tools:** thư viện tối ưu hóa của Google, được áp dụng để giải quyết các bài toán tối ưu lộ trình (VRP), phân bổ phương tiện và các bài toán vận tải phức tạp.
- **Matplotlib, Jupyter Notebook:** hỗ trợ trực quan hóa dữ liệu, hiển thị kết quả mô hình và thực hiện phân tích tương tác trong quá trình nghiên cứu.

3.3 Web Application

- **Frontend:**
 - **React + Material-UI:** được sử dụng để xây dựng giao diện người dùng hiện đại, trực quan và dễ sử dụng.
 - **Axios:** hỗ trợ gửi yêu cầu API tới backend để lấy dữ liệu, cập nhật thông tin và hiển thị kết quả trên giao diện.

- **Backend:**

- **Node.js + Express:** dùng để xây dựng API, xử lý yêu cầu từ frontend, đọc dữ liệu từ các tệp CSV và truyền kết quả lại cho giao diện web.

- **Triển khai Web:**

- **Docker + Docker Compose:** được sử dụng để đóng gói ứng dụng frontend và backend vào các container riêng biệt, giúp việc triển khai, quản lý và vận hành hệ thống trên nhiều môi trường trở nên dễ dàng và đồng nhất.

3.4 Tổng kết chương 3

Chương 3 đã hệ thống hóa các phương pháp và công nghệ cốt lõi được sử dụng trong dự án. Dựa trên nhu cầu tối ưu vận chuyển và quản lý tồn kho cùng với dữ liệu đã được thu thập và chuẩn hóa ở Chương 2, ta đã có một cái nhìn tổng quát về các công cụ Big Data, các thư viện xử lý dữ liệu – học sâu và nền tảng web phục vụ trực quan hóa kết quả. Những phương pháp này đóng vai trò nền tảng kỹ thuật, đảm bảo dữ liệu được xử lý hiệu quả, mô hình được triển khai chính xác và hệ thống có khả năng vận hành tốt.

4 Các Module Chính và Triển khai

4.1 Optimize-Delivery


- Dự đoán ETA, dự đoán tuyến đường, tối ưu tuyến, dự báo nhu cầu, sử dụng thư viện OR-Tools của Google để giải quyết bài toán tối ưu hóa tuyến đường giao hàng phức tạp.
- Công cụ: Jupyter, Pandas, TensorFlow, Dask, OR-Tools.

- Dữ liệu: Delivery, Roadmap của 5 thành phố.
- Code minh họa:
 - `Optimize-Delivery/optimize/ETA-predict.ipynb`
 - `Optimize-Delivery/optimize/Route-predict.ipynb`
 - `Optimize-Delivery/optimize/STG-forecasting.ipynb`
- Một số kết quả thu được:
 - decomposition summary :




big_data/b8.png


– Tổng quát:

big_data/b12.png


– decomposition analysis:

big_data/decomposition_analysis.png


– eta visualization:

big_data/eta_visualization_20251115_110240.png


– forecast 7days trend:

big_data/forecast_7days_trend.png


– optimization visualization:

optimization_visualization.png

– routes visualization:

routes_visualization_20251115_111802.png


– seasonality heatmap:

seasonality_heatmap.png

4.2 Inventory-Management

- Phân tích và dự báo tồn kho, xử lý dữ liệu sản phẩm – cửa hàng.
- Code minh họa:
 - inventory-analysis.ipynb
 - inventory-forecasting.ipynb

- Kết quả thu được:

comprehensive_inventory_analysis_20251117_073238.png

4.3 Mô hình học sâu

- Sử dụng TensorFlow/Keras để xây dựng mô hình deep learning nhằm dự đoán ETA và tuyến đường giao hàng qua bộ dữ liệu tổng hợp
- File: OPTIMIZE-FOR-SHIPPER.ipynb

- Kết quả:

b13.png

b14.png

4.4 File Display App

- Giao diện web xem kết quả, file CSV/JSON/PNG/JPG, và tải file.
- Frontend: React + MUI
- Backend: Node.js + Express
- Code minh họa:
File-Display-App/

- Minh họa:

b15.png

4.5 Datapipeline

- Thu thập, xử lý, lưu trữ dữ liệu trên HDFS/NFS.
- Công cụ: Python, Pandas, Dask, HDFS/NFS.
- Import và kết hợp dữ liệu.
- Lưu trữ dữ liệu đã xử lý.
- Sử dụng mô hình để tối ưu tuyến giao hàng và quản lý tồn kho.
- Kết hợp và xuất kết quả cho web app.

4.6 Tổng kết chương 4

Chương 4 đã trình bày quá trình triển khai các mô hình và hệ thống thực nghiệm dựa trên dữ liệu đã được xử lý và các phương pháp đã mô tả ở Chương 3. Các mô-đun gồm tối ưu tuyến giao hàng, dự báo tồn kho, mô hình deep learning và ứng dụng web đã được xây dựng và kiểm thử trên dữ liệu thực tế của nhiều thành phố. Bên cạnh đó, hệ thống datapipeline giúp kết nối toàn bộ quy trình từ xử lý dữ liệu, huấn luyện mô hình đến trực quan hóa kết quả. Những thành phần này tạo nên một quy trình hoàn chỉnh, sẵn sàng hỗ trợ cho việc đánh giá và tối ưu hóa trong chương tiếp theo.

5 Kết quả và đánh giá

Việc tối ưu tuyến giao hàng bằng những kỹ thuật xử lý dữ liệu lớn đã giúp:

5.1 Tối ưu tuyến giao hàng

- Giảm chi phí vận chuyển.
- Tăng tốc độ giao hàng.
- Cải thiện hiệu suất vận hành.

Nhìn chung, việc tối ưu hóa tuyến giao hàng đã mang lại những cải thiện tương đương về hiệu suất vận chuyển, giảm chi phí và nâng cao trải nghiệm khách hàng. Tuy nhiên, vẫn còn một số thách thức cần giải quyết trong tương lai, bao gồm việc xử lý dữ liệu thời gian thực, tích hợp với các hệ thống hiện có và mở rộng mô hình để áp dụng cho các khu vực địa lý khác nhau.

5.2 Quản lý tồn kho

- Dự báo nhu cầu chính xác hơn.
- Giảm thiểu/thừa hàng.
- Tối ưu số lượng nhập hàng.

Việc quản lý tồn kho đã giúp cải thiện độ chính xác trong dự báo nhu cầu, giảm thiểu tình trạng thiếu hoặc thừa hàng tồn kho, và tối ưu hóa quy trình nhập hàng và phân phối. Tuy nhiên, vẫn cần tiếp tục cải thiện mô hình dự báo và tích hợp các yếu tố bên ngoài như xu hướng thị trường và hành vi khách hàng để nâng cao hiệu quả quản lý tồn kho trong tương lai.

5.3 Trực quan hóa dữ liệu

Trực quan hóa giúp dữ liệu được đánh giá 1 cách chính xác và có cái nhìn tổng quát về hiệu năng của mô hình đồng thời triển khai web để xem dữ liệu là 1 cách hiệu quả để tóm tắt kết quả đạt được trong dự án.

5.4 Phương hướng phát triển

Nhóm đã thành công trong việc xây dựng và triển khai các mô hình tối ưu hóa tuyến giao hàng và quản lý tồn kho, sử dụng các công cụ và kỹ thuật xử lý dữ liệu lớn như HDFS, NFS, Dask, cùng với các mô hình deep learning. Ngoài ra, nhóm cũng đã phát triển một ứng dụng web để trực quan hóa và truy cập kết quả một cách dễ dàng. Bên cạnh đó nhóm cũng đang hướng tới mục tiêu:

- Phân tích dữ liệu thời gian thực.
- Tối ưu hóa thuật toán.
- Ứng dụng AI và IoT.

6 Tài liệu tham khảo

- <https://huggingface.co/datasets/Cainiao-AI/LaDe>
- <https://data.mendeley.com/datasets/mgzvngzng2/1>