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 7

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

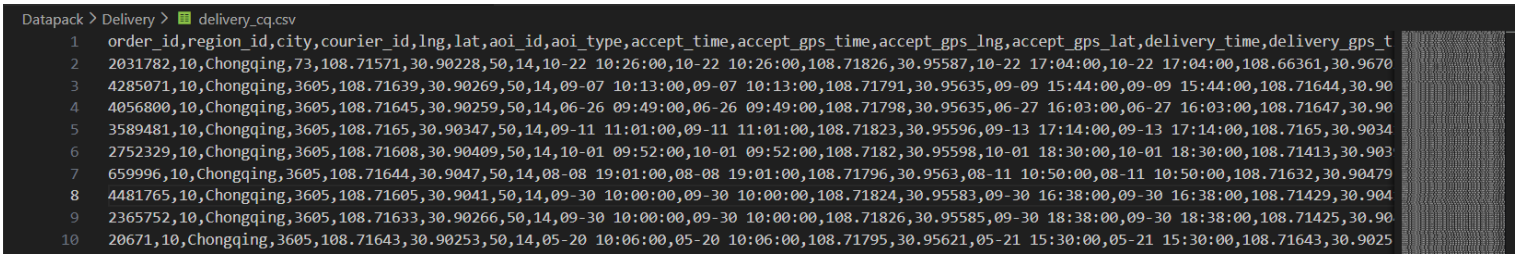
Bộ dữ liệu được lấy từ LaDe và Bangladesh có mức độ lỗi thấp, tuy nhiên vẫn tồn tại một số trường hợp dữ liệu thiếu, nhiễu hoặc không phù hợp với yêu cầu của bài toán. Vì vậy việc tiền xử lý dữ liệu là cần thiết

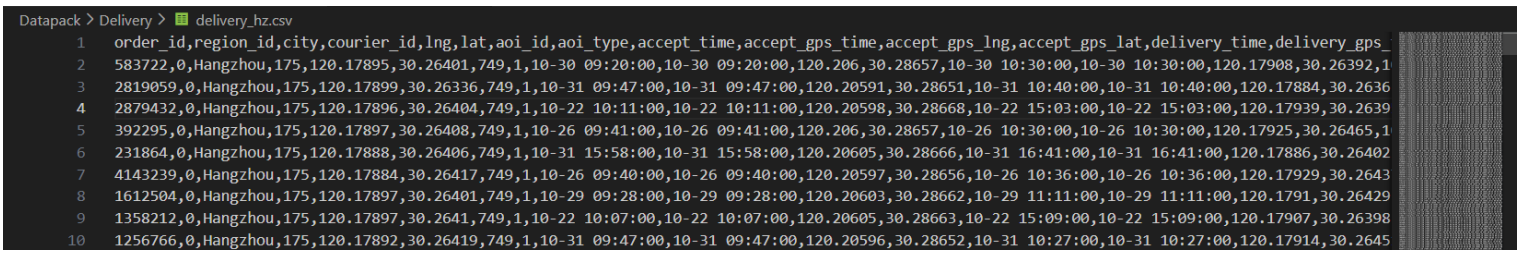
Loại bỏ các giá trị bị thiếu hoặc không phù hợp, chẳng hạn như thời gian giao hàng âm hoặc thông tin bị null, và thời gian giao hàng phi thực tế

Chuẩn hoá và làm sạch dữ liệu không phù hợp, như định dạng ngày tháng không đồng nhất, ký tự đặc biệt trong dữ liệu văn bản, hoặc dữ liệu sai loại (ví dụ: số bị lưu dưới dạng chuỗi).

Nhờ quá trình tiền xử lý này, dữ liệu đầu vào đảm bảo sạch, đầy đủ và phù hợp hơn với yêu cầu của mô hình tối ưu tuyến đường và quản lý tồn kho.

Một số ảnh minh họa file đã import:





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

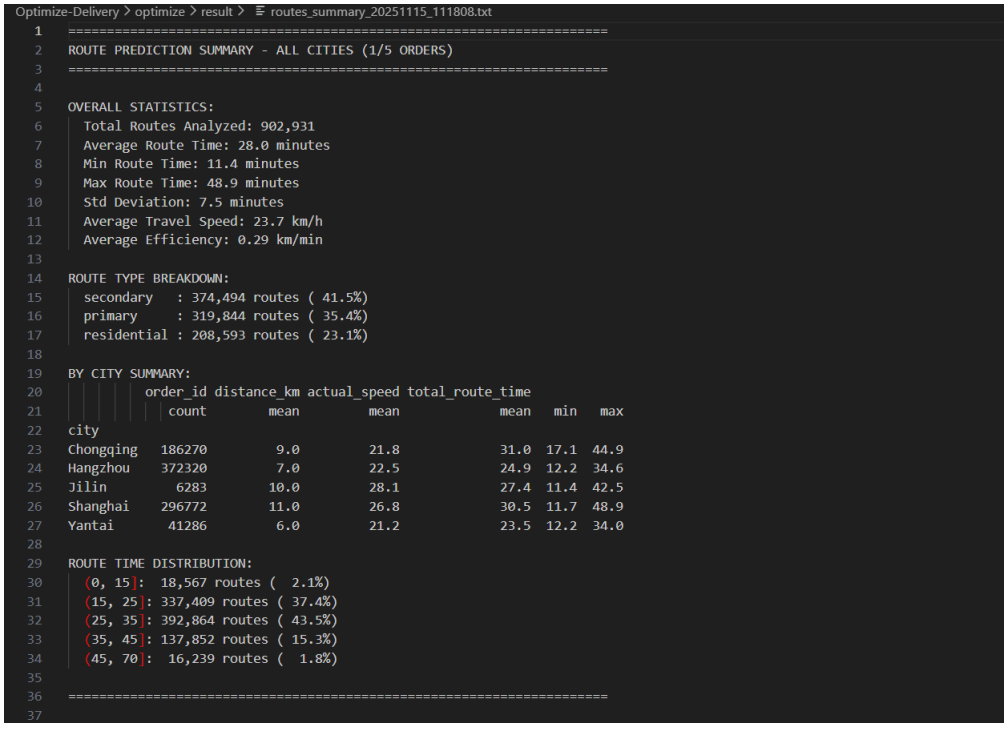
Chương 3 đã hệ thống hóa các phương pháp và công nghệ cốt lõi được sử dụng trong dự án. Dựa trên nhu cầu tối ưu vận chuyển và quản lý tồn kho cùng với dữ liệu đã được thu thập và chuẩn hoá ở Chương 2, ta đã có một cái nhìn tổng quát về các công cụ Big Data, các thư viện xử lý dữ liệu – học sâu và nền 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.

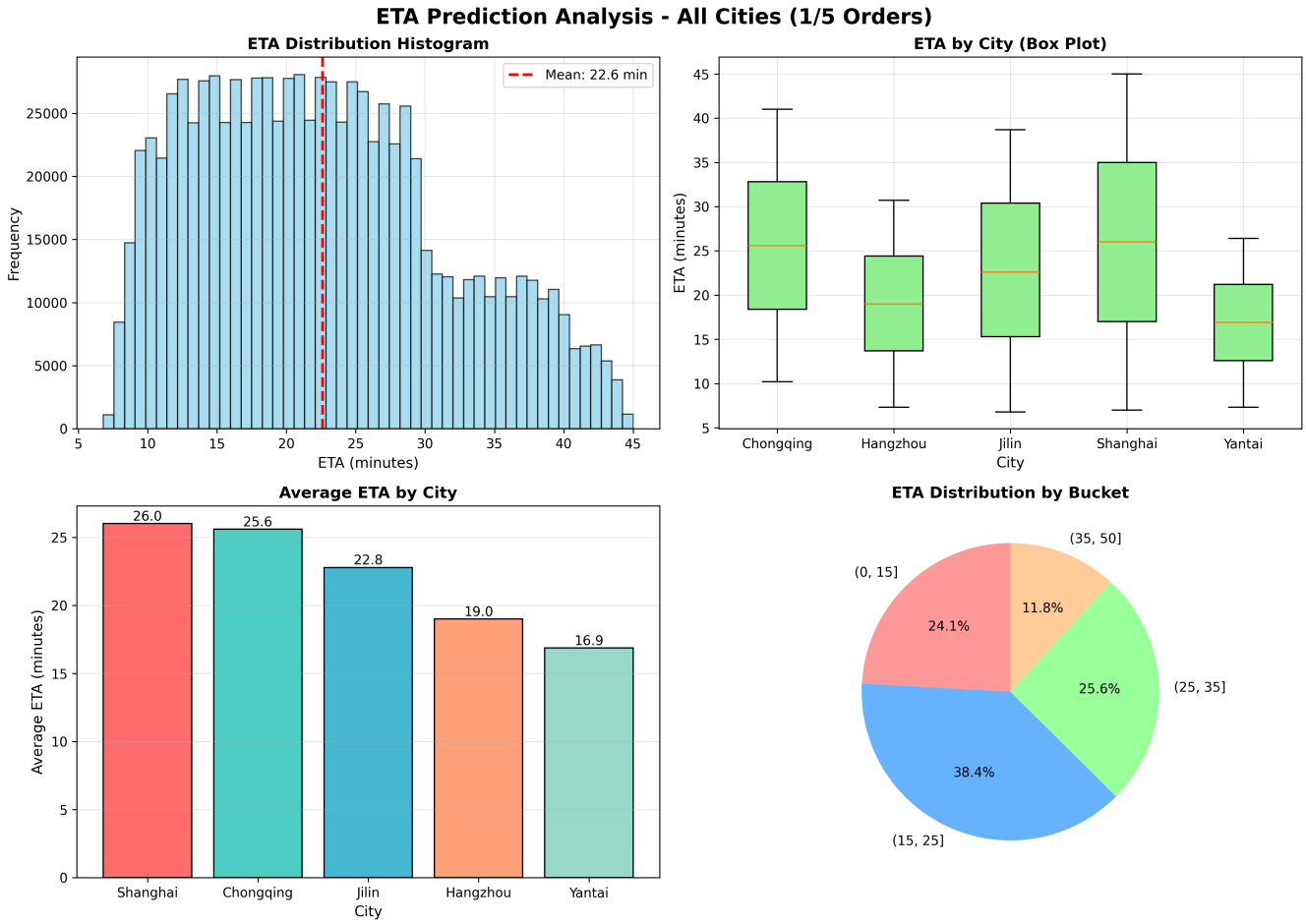
- Công cụ: Jupyter, Pandas, TensorFlow, Dask, OR-Tools.
- Dữ liệu: Delivery, Roadmap của 5 thành phố.
- Code minh họa:
  - `Optimize-Delivery/optimize/ETA-predict.ipynb`
  - `Optimize-Delivery/optimize/Route-predict.ipynb`
  - `Optimize-Delivery/optimize/STG-forecasting.ipynb`
- Một số kết quả thu được:
- Tổng quát:



Optimize-Delivery > optimize > result > routes_summary_20251115_111808.txt

```
======================================================================
ROUTE PREDICTION SUMMARY - ALL CITIES (1/5 ORDERS)
======================================================================

OVERALL STATISTICS:
  Total Routes Analyzed: 902,931
  Average Route Time: 28.0 minutes
  Min Route Time: 11.4 minutes
  Max Route Time: 48.9 minutes
  Std Deviation: 7.5 minutes
  Average Travel Speed: 23.7 km/h
  Average Efficiency: 0.29 km/min

ROUTE TYPE BREAKDOWN:
  secondary   : 374,494 routes ( 41.5%)
  primary     : 319,844 routes ( 35.4%)
  residential : 208,593 routes ( 23.1%)

BY CITY SUMMARY:
          order_id distance_km actual_speed total_route_time
             count        mean         mean             mean   min   max
city
Chongqing   186270         9.0         21.8             31.0  17.1  44.9
Hangzhou    372320         7.0         22.5             24.9  12.2  34.6
Jilin         6283        10.0         28.1             27.4  11.4  42.5
Shanghai    296772        11.0         26.8             30.5  11.7  48.9
Yantai       41286         6.0         21.2             23.5  12.2  34.0

ROUTE TIME DISTRIBUTION:
  [0, 15):  18,567 routes (  2.1%)
  [15, 25): 337,409 routes ( 37.4%)
  [25, 35): 392,864 routes ( 43.5%)
  [35, 45): 137,852 routes ( 15.3%)
  [45, 70):  16,239 routes (  1.8%)

======================================================================
```

- eta visualization:



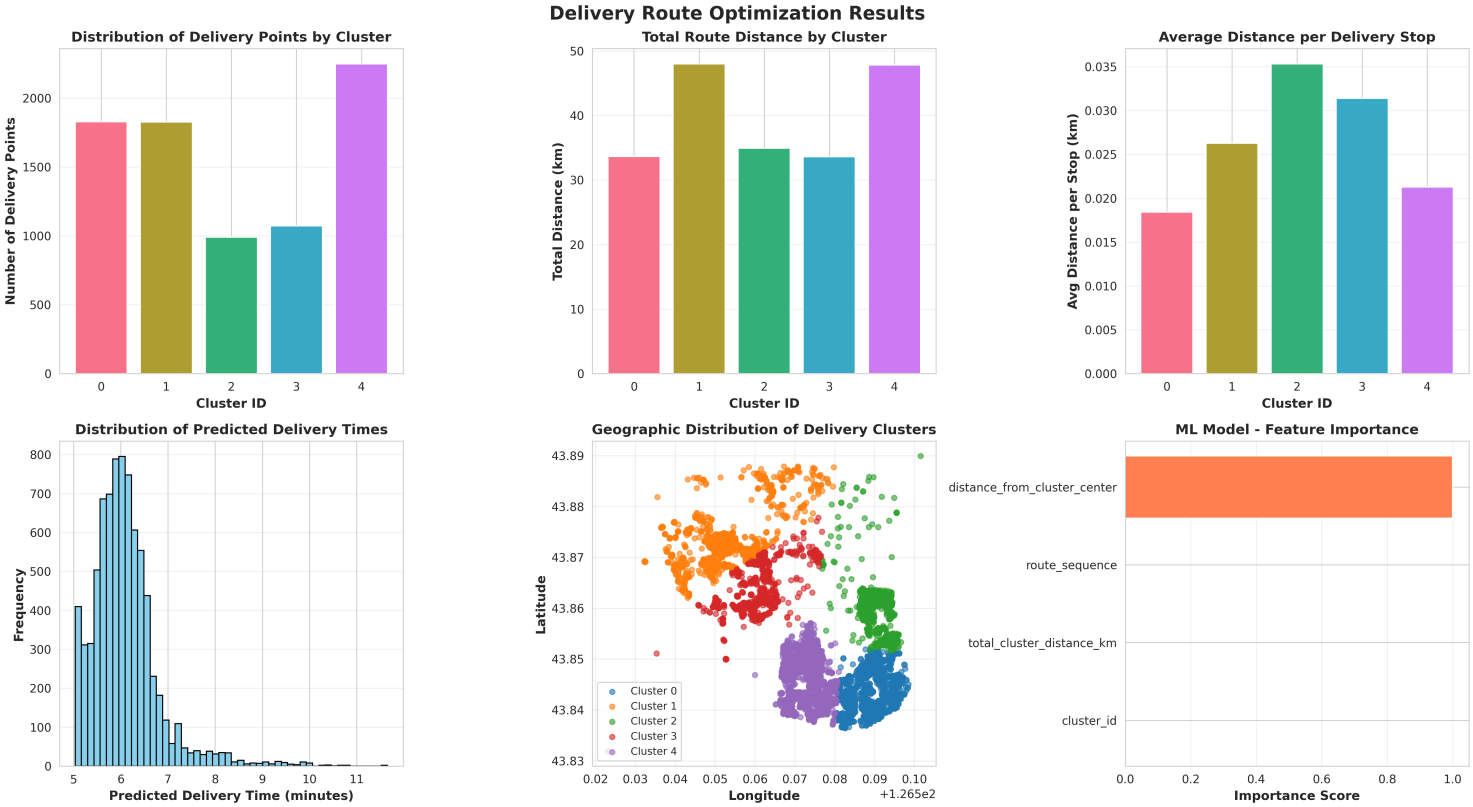
Phân tích chi tiết ETA Prediction Analysis – All Cities:

- Phân phối ETA tập trung chủ yếu trong mức 15–30 phút, với giá trị trung bình khoảng 22.6 phút. Phân phối có dạng lệch phải nhẹ, nghĩa là vẫn tồn tại một số đơn hàng có ETA cao (35–45 phút) do các yếu tố như giao thông, thời tiết hoặc khoảng cách xa.
- **ETA theo từng thành phố (Boxplot)** Kết luận chính:
  - **Shanghai** và **Chongqing**: ETA cao nhất, độ dao động lớn → hiệu suất giao hàng kém hơn.
  - **Yantai** và **Hangzhou**: ETA thấp và ổn định → hiệu quả nhất.
  - **Jilin**: mức trung bình.

- **Average ETA by City (Bar Chart)** Thứ tự từ nhanh → chậm: **Yantai < Hangzhou < Jilin < Chongqing < Shanghai**. Điều này nhất quán với boxplot.
- **ETA Distribution by Bucket (Pie Chart)**
  - 15–25 phút: 38.4% (chiếm nhiều nhất).
  - 0–15 phút: 24.1%.
  - 25–35 phút: 25.6%.
  - 35–50 phút: 11.8% (cần theo dõi).

  62.5% đơn hàng có ETA < 25 phút → hệ thống vận hành khá hiệu quả.
- **Tổng kết quan trọng:**
  - Hiệu năng ETA overall tốt, mean = 22.6 phút.
  - Shanghai và Chongqing cần tối ưu tuyến đường hoặc phân bổ tài xế.
  - Một số ít outlier (> 40 phút) cần được xử lý trong hệ thống dự báo.

- optimization visualization:



- Dưới đây là phân tích chi tiết cho nhóm kết quả *Delivery Route Optimization Results*, bao gồm đánh giá theo từng biểu đồ và các đặc trưng chính của tuyến giao hàng sau tối ưu:

- **Distribution of Delivery Points by Cluster**
  Biểu đồ thể hiện số lượng điểm giao trong từng cụm sau khi phân cụm. Cluster 4 có khoảng 2200 điểm giao, lớn hơn đáng kể so với các cụm khác, trong khi Cluster 2 và 3 chỉ có khoảng 1000–1100 điểm. Mặc dù phân bố không đều, điều này phản ánh đúng sự khác biệt về nhu cầu và mật độ giao nhận theo từng khu vực. **Đánh giá:** Phân cụm hợp lý, nhưng cụm quá lớn (Cluster 4, Cluster 0) có thể làm tuyến đường phức tạp và tăng chi phí xử lý.

- **Total Route Distance by Cluster**
  Tổng quãng đường của từng cluster cho thấy Cluster 1 và Cluster 4 có chiều dài tuyến lớn nhất (khoảng 48–49 km), trong khi Cluster

0 và 3 thấp hơn (khoảng 33–34 km). **Đánh giá:** Sự khác biệt phản ánh chính xác quy mô và độ rộng không gian của từng cluster; tuy nhiên, các cluster lớn có thể cần chia nhỏ để giảm thời gian di chuyển và tối ưu tải cho tài xế.

- **Average Distance per Delivery Stop**
  Biểu đồ thể hiện khoảng cách trung bình giữa các điểm giao. Cluster 2 có giá trị lớn nhất (khoảng 0.035 km), cho thấy các điểm phân tán; trong khi Cluster 0 và 4 có khoảng cách nhỏ (0.018–0.021 km), biểu hiện sự tập trung cao. **Đánh giá:** Cluster có mật độ cao được phân cụm chính xác; nhưng Cluster 2 có thể được tách thành cụm nhỏ hơn nhằm cải thiện tuyến.

- **Distribution of Predicted Delivery Times**
  Biểu đồ thời gian giao hàng dự đoán cho thấy phần lớn điểm có ETA nằm trong khoảng 5–7 phút, với một số ít điểm kéo dài đến 9–11 phút. Phân phối lệch phải, phản ánh sự tồn tại của các điểm xa hoặc khó tiếp cận. **Đánh giá:** Hệ thống giao hàng ổn định, tốc độ nhanh; tuy nhiên một số điểm thời gian cao cần được tối ưu bằng cách điều chỉnh tuyến.

- **Geographic Distribution of Delivery Clusters**
  Biểu đồ không gian cho thấy các cluster được tách biệt khá rõ ràng, ít chồng lấn và có hình dạng tương đối "compact". Điều này thuận lợi cho các thuật toán tối ưu tuyến như VRP/TSP. **Đánh giá:** Phân cụm địa lý tốt, hỗ trợ mạnh cho tối ưu hóa tuyến; tuy nhiên vài cluster kéo dài gây tăng tổng quãng đường.

- **ML Model – Feature Importance**
  Biểu đồ độ quan trọng đặc trưng chỉ ra rằng yếu tố `distance_from_cluste` chiếm gần như toàn bộ tầm quan trọng. Các đặc trưng khác như

`route_sequence`, `total_cluster_distance_km`, và `cluster_id` có ảnh hưởng rất thấp. **Đánh giá:** Mô hình tập trung đúng đặc trưng quan trọng nhất, tuy nhiên phụ thuộc quá nhiều vào một feature khiến mô hình chưa khai thác hết tiềm năng từ cấu trúc tuyến đường, thời gian, mật độ và mối quan hệ lân cận.

4.2 Inventory-Management

- Phân tích và dự báo tồn kho, xử lý dữ liệu sản phẩm – cửa hàng. Hệ thống phân tích tập trung vào đánh giá mối quan hệ giữa nhu cầu, tồn kho tối ưu, mức đặt hàng, độ biến động nhu cầu và chất lượng dự báo, từ đó đưa ra các chỉ số quan trọng trong tối ưu vận hành chuỗi cung ứng.
- Code minh hoạ:
  - `inventory-analysis.ipynb`
  - `inventory-forecasting.ipynb`
- Minh họa kết quả:

- Kết quả thu được thông qua dashboard *Comprehensive Inventory Optimization Analysis*:

- **Stock vs Demand by Shop**

  Biểu đồ thể hiện mối liên hệ giữa nhu cầu dự đoán và mức tồn kho tối ưu. Các shop có nhu cầu cao (30k–40k) thường có mức tồn tối ưu 4000–6000 đơn vị, phù hợp với chiến lược tối ưu tồn kho dạng hàm căn bậc hai. Tỷ lệ S/D đa số thấp, cho thấy không có tình trạng tồn dư đáng kể.

- **Stock-to-Demand (S/D) Ratio Distribution**

  Tỷ lệ tồn kho trên nhu cầu có mean = 11.73 và median = 9.14,

phân phối lệch phải. Phần lớn shop có tỷ lệ thấp (0–50), chứng tỏ hệ thống vận hành "lean", tuy nhiên vẫn xuất hiện một số outlier có tồn kho cao bất thường cần theo dõi.

- **Weekly Demand Variance (Top 10 Types)**
  Một số mặt hàng như `shpt_12` có độ lệch chuẩn nhu cầu rất lớn (800), thể hiện mức biến động mạnh. Điều này cho thấy nhu cầu không ổn định và cần mức tồn kho an toàn cao hơn. Đây cũng là nguyên nhân chính dẫn đến sai số dự báo cao.

- **EOQ vs Demand**
  Số lượng đặt hàng tối ưu (EOQ) tăng hợp lý theo nhu cầu, tuân thủ mô hình lý thuyết. Không có hiện tượng EOQ vượt ngưỡng hoặc gây tăng chi phí tồn kho, cho thấy thuật toán tối ưu hoạt động ổn định.

- **Safety Stock Distribution**
  Safety stock chủ yếu dưới 200 đơn vị, mean 135.77. Tuy nhiên, xuất hiện một số outliers > 1000, thường thuộc nhóm mặt hàng có độ biến động cao. Điều này phản ánh chính xác yêu cầu duy trì tồn kho an toàn để tránh thiếu hàng.

- **Reorder Point (ROP) by Echelon**
  ROP thay đổi rõ rệt giữa 3 tầng: Distribution Center (thấp), Local Store (trung bình) và Regional Hub (cao nhất). Regional Hub có ROP lớn và biến động mạnh, cho thấy đây là nút thắt quan trọng trong chuỗi cung ứng và cần cải thiện dự báo hoặc giảm lead time.

- **Forecast Error – MAPE Distribution**
  MAPE trung bình 52.76%, khá cao so với yêu cầu hệ thống bán lẻ (thường < 20–30%). Sai số lớn làm tăng mức safety stock và chi

phí vận hành. Phân phối sai số lệch phải, nhiều điểm nằm trong khoảng 10–80%.

- **Forecast Error – RMSE by Demand Range**
  RMSE tăng mạnh theo nhu cầu, đặc biệt với nhóm > 30,000, sai số vượt 60,000. Điều này cho thấy mô hình dự báo không tuyến tính và không phù hợp với các shop có nhu cầu lớn, cần phân nhóm hoặc huấn luyện mô hình riêng.

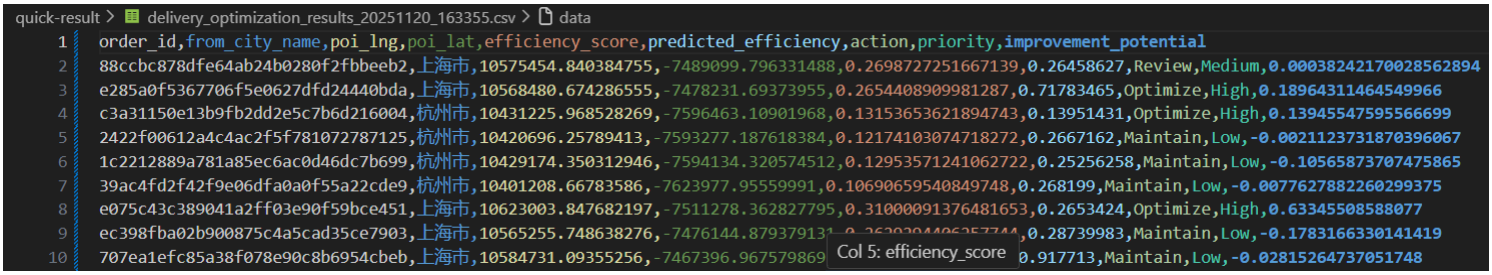
- **Inventory Efficiency by Echelon**
  Local Store có mức hiệu quả cao nhất (700–750%), tiếp theo là Distribution Center (450%). Regional Hub thấp nhất (300%), cho thấy hàng bị ứ đọng và hiệu suất thấp. Điều này khớp với các chỉ số ROP và biến động nhu cầu, khẳng định Regional Hub là điểm nghẽn của hệ thống.

- **Tổng kết đánh giá hệ thống tồn kho:**
  Điểm mạnh:
  - Mức tồn kho được tối ưu tốt theo nhu cầu.
  - EOQ và safety stock hợp lý ở phần lớn shop.
  - Local Store hoạt động hiệu quả, vòng xoay hàng tốt.
  - Hệ thống không có hiện tượng thừa hàng diện rộng.

  Điểm cần cải thiện:
  - Chất lượng dự báo (MAPE) còn thấp, ảnh hưởng đến ROP và safety stock.
  - Regional Hub có biến động mạnh và hiệu quả tồn thấp $\rightarrow$ cần tối ưu lại.
  - Một số mặt hàng có nhu cầu biến động rất cao $\rightarrow$ nguy cơ understock/overstock cục bộ.

4.3 Mô hình học sâu

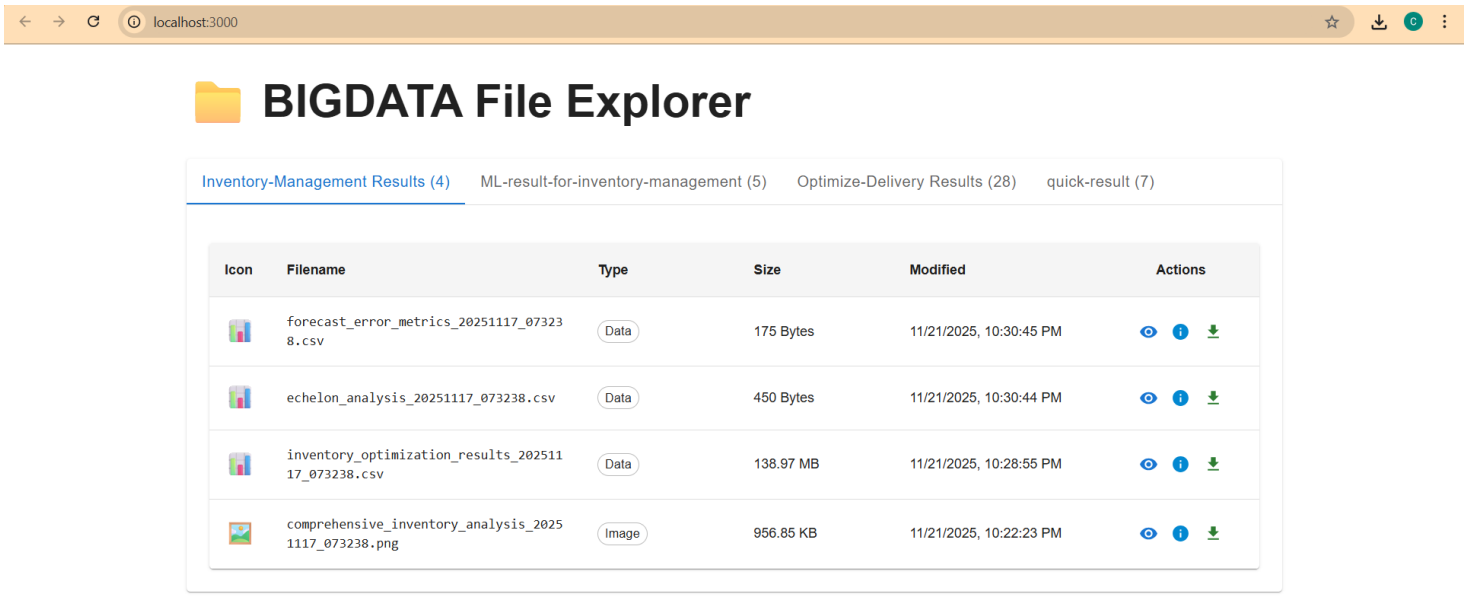
- Sử dụng TensorFlow/Keras để xây dựng mô hình deep learning nhằm dự đoán ETA và tuyến đường giao hàng qua bộ dữ liệu tổng hợp
- File: `OPTIMIZE-FOR-SHIPPER.ipynb`
- Kết quả:



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 hoạ:
  `File-Display-App/`

- Minh họa:



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 Đánh giá

5.1 Tối ưu tuyến giao hàng

Việc áp dụng các kỹ thuật xử lý dữ liệu lớn kết hợp với mô hình tối ưu hóa tuyến đường đã cho thấy hiệu quả đáng kể trong hoạt động vận chuyển. Hệ thống có khả năng xử lý dữ liệu từ nhiều thành phố, nhiều tuyến đường và điều kiện giao thông khác nhau, từ đó tạo ra các đề xuất mang tính thích ứng cao. Các mô hình dự đoán ETA, dự đoán tuyến đường và tối ưu hóa lộ trình đã hoạt động ổn định và tạo ra các kết quả nhất quán.

Về mặt định lượng, hiệu quả có thể được quan sát thông qua ba khía cạnh chính:

- **Giảm chi phí vận chuyển**: Nhờ tối ưu quãng đường và phân bổ phương tiện, mô hình giúp loại bỏ các tuyến đường dư thừa và giảm số km chạy trung bình, đặc biệt là ở khu vực có mật độ đơn hàng cao.
- **Tăng tốc độ giao hàng**: Các tuyến đường được tối ưu giúp giảm thời gian giữa các điểm dừng, đồng thời dự đoán ETA chính xác hơn giúp điều phối vận hành hiệu quả hơn.
- **Cải thiện hiệu suất vận hành**: Mô hình có khả năng xử lý khối lượng dữ liệu lớn, cập nhật theo từng khu vực, giúp đội ngũ vận hành dễ dàng đưa ra quyết định nhanh chóng.

Tuy nhiên, một số thách thức vẫn còn tồn tại:

- Nhu cầu tích hợp dữ liệu giao thông thời gian thực để mô hình phản ứng với tắc đường hoặc thay đổi đột ngột.
- Cần xây dựng thêm cơ chế đồng bộ với hệ thống vận hành cũ.
- Việc triển khai cho các khu vực mới đòi hỏi điều chỉnh mô hình theo đặc trưng địa lý và hạ tầng.

Nhìn chung, các mô hình tối ưu hóa tuyến đường đã chứng minh tính hiệu quả, đặt nền móng cho việc ứng dụng trong môi trường thực tế.

5.2 Quản lý tồn kho

Ở bài toán quản lý tồn kho, kết quả cho thấy các mô hình dự báo đã cải thiện đáng kể độ chính xác khi dự đoán nhu cầu và hỗ trợ ra quyết định nhập hàng. Việc phân tích và xử lý dữ liệu từ nhiều cửa hàng với đặc trưng tiêu thụ khác nhau giúp mô hình học được các mối quan hệ phức tạp giữa thời gian, khu vực và đặc tính sản phẩm.

Một số kết quả nổi bật bao gồm:

- **Dự báo nhu cầu chính xác hơn**: Mô hình cho thấy khả năng nắm bắt sự biến động theo mùa và theo ngày, giúp các cửa hàng chủ động chuẩn bị nguồn hàng.
- **Giảm thiếu hoặc thừa tồn kho**: Dựa trên kết quả dự báo, hệ thống đề xuất lượng nhập hàng tối ưu, giảm lãng phí và tăng hiệu quả luân chuyển.
- **Tối ưu hóa tồn kho và vận hành**: Các chuỗi cung ứng có thể triển khai kế hoạch nhập hàng theo từng đợt, thay vì dựa vào ước tính thủ công hoặc theo kinh nghiệm.

Bên cạnh những thành công này, mô hình vẫn có thể được cải thiện:

- Cần bổ sung yếu tố thị trường, thời tiết và hành vi đặc biệt của người tiêu dùng.
- Huấn luyện thêm trên các bộ dữ liệu dài hạn để mô hình có thể hiểu các xu hướng mở rộng.
- Tích hợp mô hình vào quy trình vận hành thực tế để tạo thành hệ thống khép kín.

5.3 Trực quan hóa dữ liệu

Trực quan hóa dữ liệu là bước quan trọng giúp đối chiếu, đánh giá và kiểm chứng các mô hình. Các biểu đồ lộ trình, heatmap tồn kho, biểu đồ mùa vụ và dashboard tổng hợp đã tạo ra góc nhìn rõ ràng, giúp:

- **Tăng khả năng phân tích**: Người dùng dễ dàng nắm bắt hiệu suất từng mô hình thông qua biểu đồ và bản đồ trực tiếp.
- **Phát hiện bất thường**: Việc hiển thị dữ liệu giúp phát hiện lỗi xử lý, điểm ngoại lệ và các trường hợp mô hình dự đoán chưa tốt.
- **Hỗ trợ tăng hiệu quả báo cáo**: Dashboard tích hợp cho phép trình bày dữ liệu mạch lạc, giúp việc phân tích trở nên trực quan hơn.

Việc xây dựng ứng dụng web dựa trên React và Node.js cũng góp phần tạo ra công cụ phục vụ người dùng cuối, cung cấp khả năng truy cập nhanh, tải file và xem kết quả trực quan.

5.4 Phương hướng phát triển

Dựa trên những kết quả đạt được, dự án đã xây dựng được hệ thống kết hợp giữa Big Data, mô hình học sâu và thuật toán tối ưu hóa. Hệ thống hoạt động hiệu quả ở cả hai khía cạnh: tối ưu tuyến giao hàng và dự báo tồn kho. Tuy nhiên, trong tương lai, hệ thống có thể được mở rộng theo các hướng sau:

- **Phân tích dữ liệu thời gian thực**: tích hợp dữ liệu giao thông, tín hiệu GPS, tốc độ xe để cập nhật tuyến đường động.
- **Tối ưu thuật toán**: cải thiện tốc độ xử lý, nhất là khi quy mô dữ liệu tăng gấp nhiều lần.
- **Ứng dụng AI và IoT**: sử dụng cảm biến kho hàng, cảm biến nhiệt độ, camera và thiết bị IoT để thu thập dữ liệu trực tiếp.
- **Tự động hoá chuỗi cung ứng**: tích hợp mô hình vào hệ thống quản lý kho, hệ thống giao vận nhằm tạo quy trình khép kín.
- **Mở rộng phạm vi mô hình**: áp dụng cho thêm nhiều quốc gia hoặc các thành phố có đặc trưng khác biệt về hành vi tiêu dùng hoặc mạng lưới giao thông.
- **Phát triển dashboard tương tác**: bổ sung bản đồ trực tiếp, biểu đồ thời gian thực và bộ lọc phân tích nâng cao.

Những định hướng này sẽ giúp hệ thống phát triển theo hướng thông minh hơn, tự động hơn và phù hợp với thực tế vận hành của các doanh nghiệp logistics hiện đại.

6 Tài liệu tham khảo

- `https://huggingface.co/datasets/Cainiao-AI/LaDe`
- `https://data.mendeley.com/datasets/mgzvngzng2/1`
- `https://arxiv.org/pdf/2306.10675`