BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG

PHÂN HIỆU TRƯỜNG ĐẠI HỌC THỦY LỢI

BỘ MÔN CÔNG NGHỆ THÔNG TIN

TÊN ĐỀ TÀI

DỰ ĐOÁN THỜI GIAN GIAO HÀNG

Giảng Viên:	Vũ Thị Hạnh
Sinh Viên Thực Hiện:	2351267277 Lê Văn Quang 2351267264 Trần Mạnh Hùng
Lớp:	S26-65TTNT

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1. GIỚI THIỆU

1.1. Bối cảnh

Trong thời đại thương mại điện tử phát triển mạnh mẽ, việc dự đoán chính xác thời gian giao hàng đóng vai trò quan trọng trong việc nâng cao trải nghiệm khách hàng và tối ưu hóa quy trình logistics. Khách hàng ngày càng mong đợi sự minh bạch và chính xác về thời gian nhận hàng, trong khi các doanh nghiệp cần tối ưu hóa chi phí vận chuyển và quản lý kho bãi hiệu quả.

1.2. Tầm quan trọng

Dự đoán thời gian giao hàng chính xác mang lại nhiều lợi ích:

- Nâng cao trải nghiệm khách hàng: Cung cấp thông tin chính xác về thời gian giao hàng
- Tối ưu hóa logistics: Phân bổ nguồn lực vận chuyển hiệu quả
- Giảm chi phí: Tránh tình trạng giao hàng trễ và phải bồi thường
- Cải thiện uy tín: Xây dựng lòng tin với khách hàng thông qua cam kết đúng hạn

1.3. Nguồn dữ liệu

Dự án sử dụng bộ dữ liệu "E-commerce Order Dataset" từ Kaggle, bao gồm thông tin chi tiết về đơn hàng, sản phẩm, khách hàng, người bán và thanh toán từ một nền tảng thương mại điện tử.

2. MỤC TIÊU VÀ BÀI TOÁN ĐẶT RA

2.1. Mục tiêu chính

Xây dựng hệ thống học máy để:
1. Phân loại trạng thái giao hàng (Classification): Dự đoán đơn hàng sẽ được giao đúng hạn, trễ, hay cực kỳ trễ
2. Dự đoán thời gian giao hàng (Regression): Ước tính số ngày giao hàng cụ thể

2.2. Bài toán cụ thể

Bài toán 1 - Classification (Giai đoạn 1):

- Input: Thông tin đơn hàng (sản phẩm, khách hàng, người bán, thanh toán)
- Output: Phân loại 3 lớp
  - Lớp 0: Bình thường (giao đúng hạn hoặc sớm)
  - Lớp 1: Trễ (giao sau thời gian ước tính nhưng < 60 ngày)

- Lớp 2: Cực kỳ trễ (> 60 ngày)

Bài toán 2 - Regression (Giai đoạn 2):

- Input: Thông tin đơn hàng
- Output: Số ngày giao hàng dự kiến (giới hạn tối đa 60 ngày)

2.3. Phương pháp tiếp cận

Sử dụng pipeline 2 giai đoạn:
1. Giai đoạn 1: Phân loại bằng CatBoost Classifier
2. Giai đoạn 2: Dự đoán thời gian bằng CatBoost/XGBoost Regressor

3. MÔ TẢ DỮ LIỆU VÀ CÁC BƯỚC TIỀN XỬ LÝ

3.1. Mô tả dữ liệu

Bộ dữ liệu gồm 5 bảng chính:

1. df_Orders.csv: Thông tin đơn hàng
   - order_id, customer_id, order_status
   - order_purchase_timestamp, order_delivered_timestamp
   - order_estimated_delivery_date

2. df_OrderItems.csv: Chi tiết sản phẩm trong đơn hàng
   - order_id, product_id, seller_id
   - price, shipping_charges

3. df_Products.csv: Thông tin sản phẩm
   - product_id, product_category_name
   - product_weight_g, product_length_cm, product_height_cm, product_width_cm

4. df_Customers.csv: Thông tin khách hàng
   - customer_id, customer_zip_code_prefix
   - customer_city, customer_state

5. df_Payments.csv: Thông tin thanh toán
   - order_id, payment_type, payment_value
   - payment_installments

Kích thước dữ liệu sau xử lý: 87,427 đơn hàng với 45 đặc trưng

3.2. Phân tích dữ liệu khám phá (EDA)

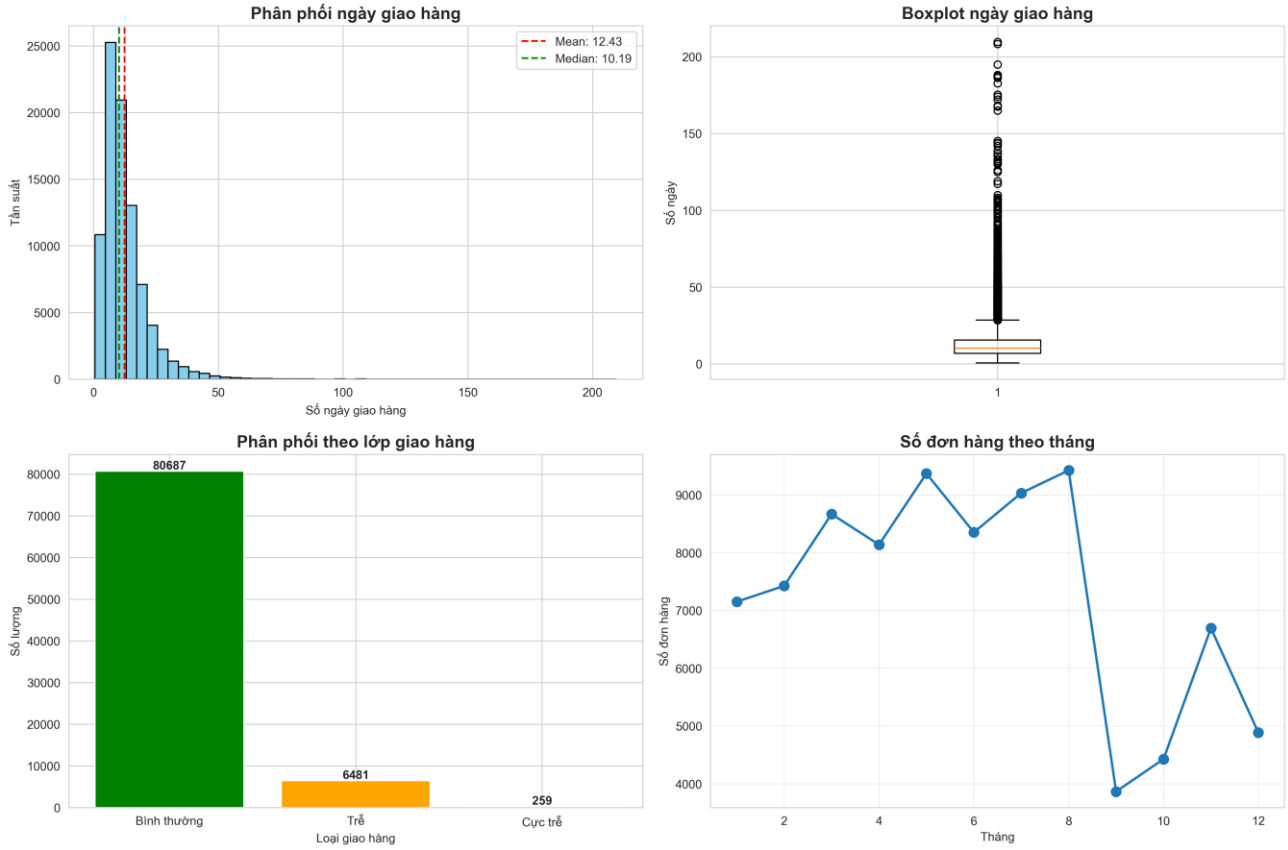
3.2.1. Phân phối thời gian giao hàng

Thống kê mô tả:

Trung bình: 12.43 ngày
Trung vị: 10.19 ngày
Độ lệch chuẩn: 9.27 ngày
Min: 0.53 ngày
Max: 209.63 ngày

Phân vị:
50%: 10.19 ngày
75%: 15.46 ngày
90%: 22.82 ngày
95%: 28.97 ngày
99%: 45.50 ngày

Ngoại lai:
Số đơn hàng > 60 ngày: 259 (0.30%)
Số đơn hàng > 90 ngày: 56 (0.06%)



Hình 1: Phân tích phân phối lệch phải, outliers, mất cân bằng lớp

3.2.2. Phân phối theo lớp

Lớp 0: Bình thường (Tỷ lệ: 92.29%, số lượng: 80.688)

Lớp 1: Trễ (Tỷ lệ 7.41%, Số lượng 6.480)

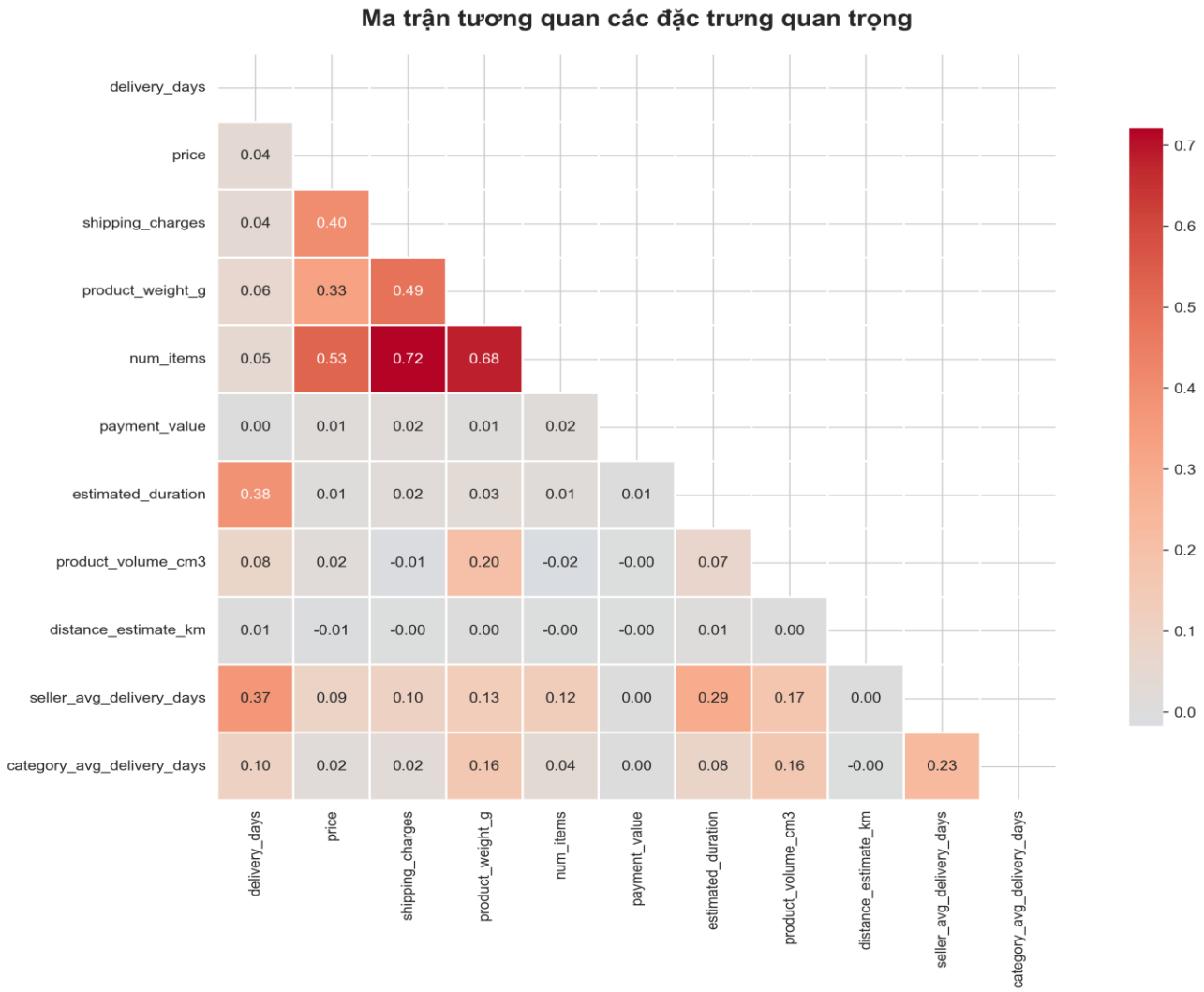
Lớp 2 cực kỳ trễ (Tỷ lệ 0.30%, Số lượng: 259

Nhận xét: Dữ liệu mất cân bằng nghiệm trọng, với lớp thiểu số (Cực kỳ trễ) chỉ chiếm 0.30%

3.2.3. Tương quan với mục tiêu

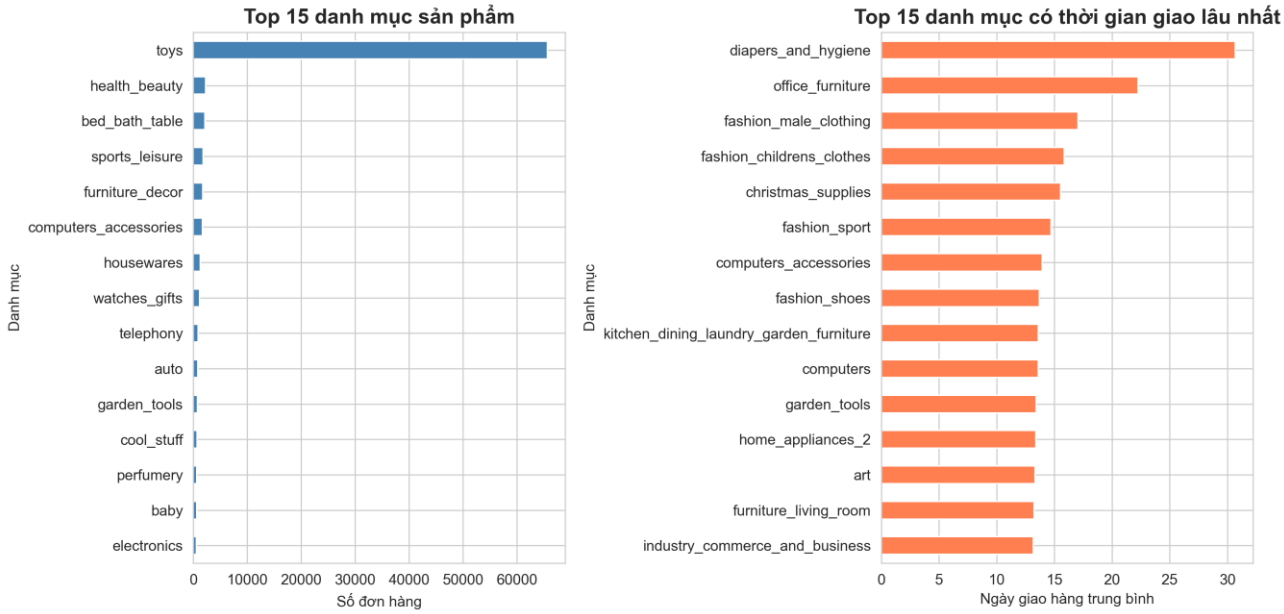
Top 10 đặc trưng có tương quan cao nhất với `delivery_days`:

1. delivery_days_capped: 0.974
2. delivery_class: 0.637
3. delay_vs_estimated: 0.586
4. is_extreme: 0.416
5. estimated_duration: 0.384
6. seller_avg_delivery_days: 0.367
7. seller_delivery_std: 0.224
8. is_holiday_season: 0.112
9. category_avg_delivery_days: 0.097



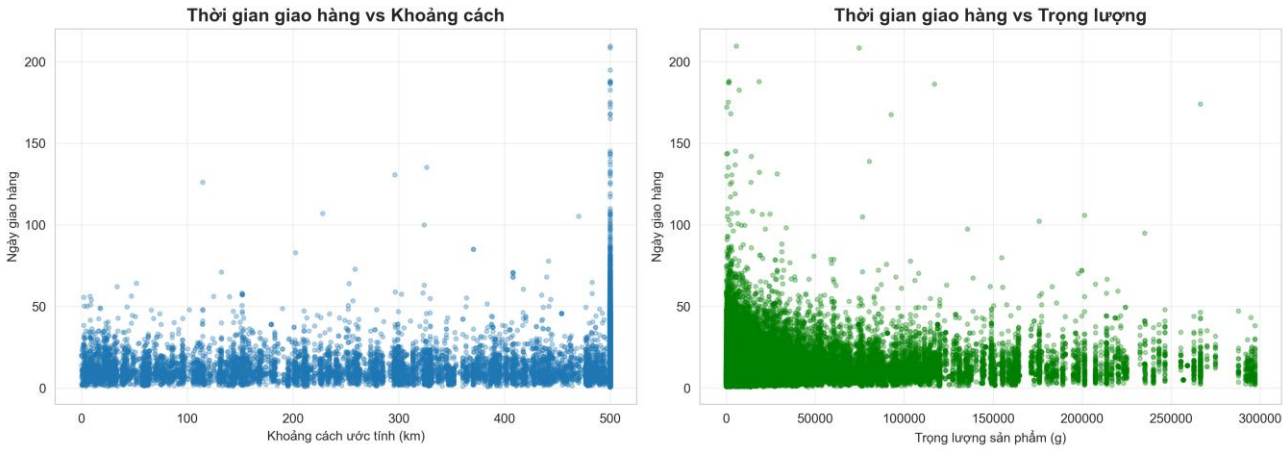
Hình 2: Phân tích tương quan features

3.2.4. Phân tích theo danh mục sản phẩm



Hình 3: So sánh danh mục phổ biến vs thời gian giao

3.2.5. Phân tích theo khoảng cách và trọng lượng



Hình 4: Quan hệ khoảng cách/trọng lượng với thời gian giao

3.3. Các bước tiền xử lý

Bước 1: Gộp dữ liệu (Data Merging)
- Gộp 5 bảng dữ liệu thành 1 DataFrame thống nhất
- Sử dụng `merge` với các khóa phù hợp (order_id, customer_id, product_id)
- Tổng hợp thông tin items theo từng đơn hàng (sum, max, count)

Bước 2: Làm sạch dữ liệu (Data Cleaning)
- Chuyển đổi cột ngày tháng sang định dạng datetime
- Loại bỏ đơn hàng chưa giao (missing delivery_timestamp)

- Loại bỏ giá trị delivery_days ≤ 0 (không hợp lệ)
- Loại bỏ bản ghi trùng lặp
- Xử lý missing values:
  - Đặc trưng số: điền bằng median
  - Đặc trưng phân loại: điền bằng 'unknown'

Bước 3: Tạo biến mục tiêu (Target Engineering)
- **delivery_days:** Số ngày từ lúc đặt hàng đến khi giao
- **delivery_class:** Phân loại 3 lớp (0: Bình thường, 1: Trễ, 2: Cực kỳ trễ)
- **delivery_days_capped:** Giới hạn tối đa 60 ngày cho regression

Bước 4: Tạo đặc trưng cơ bản (Basic Feature Engineering)
- Đặc trưng thời gian:**
  - purchase_month, purchase_dow, purchase_hour
  - is_weekend, is_month_end, is_holiday_season
  - purchase_quarter

- Đặc trưng sản phẩm:
  - product_volume_cm3 = length × height × width
  - shipping_per_weight = shipping_charges / weight
  - num_items: số lượng sản phẩm trong đơn hàng

- Đặc trưng logistics:
  - estimated_duration: thời gian ước tính của hệ thống
  - delay_vs_estimated: độ trễ so với ước tính

Bước 5: Tạo đặc trưng nâng cao (Advanced Feature Engineering)
Ước tính khoảng cách địa lý:
- Tính khoảng cách dựa trên mã bưu chính khách hàng
- distance_estimate_km = |customer_zip - median_zip| / 10
- Giới hạn tối đa 500km

Lịch sử hiệu suất người bán:
- seller_avg_delivery_days: thời gian giao hàng trung bình
- seller_order_count: số lượng đơn hàng
- seller_delivery_std: độ lệch chuẩn thời gian giao

Lịch sử danh mục sản phẩm:
- category_avg_delivery_days: thời gian giao trung bình theo danh mục
- category_delivery_std: độ lệch chuẩn theo danh mục

Lịch sử khách hàng:
- customer_order_count: số lần mua hàng của khách

CHƯƠNG 2: MÔ HÌNH HỌC MÁY SỬ DỤNG

2.1. CatBoost Classifier (Giai đoạn 1 - Phân loại)

2.1.1. Nguyên lý hoạt động

CatBoost (Categorical Boosting) là thuật toán gradient boosting được phát triển bởi Yandex, đặc biệt hiệu quả với dữ liệu có nhiều biến phân loại.

Nguyên lý chính:
- Gradient Boosting: Xây dựng mô hình theo cách tuần tự, mỗi cây mới học từ sai số của cây trước
- Ordered Boosting: Giảm overfitting bằng cách sử dụng thứ tự ngẫu nhiên khác nhau cho mỗi cây
- Categorical Features Handling: Xử lý tự động biến phân loại mà không cần one-hot encoding
- Symmetric Trees: Sử dụng cây đối xứng để tăng tốc độ và giảm overfitting

2.1.2. Lý do lựa chọn

1. Xử lý tốt dữ liệu mất cân bằng: Hỗ trợ class weights
2. Xử lý tự động categorical features: Không cần encoding thủ công
3. Hiệu suất cao: Độ chính xác tốt với ít hyperparameter tuning
4. Chống overfitting: Built-in regularization và ordered boosting
5. Hỗ trợ multi-class classification: Phù hợp với bài toán 3 lớp

2.1.3. Hyperparameter Tuning

Các siêu tham số chính:

Code python:

```
CatBoostClassifier(
    iterations=2000,            # Số cây quyết định
    learning_rate=0.03,         # Tốc độ học (giảm để ổn định)
    depth=10,                   # Độ sâu cây (tăng để bắt pattern phức tạp)
    l2_leaf_reg=5,              # L2 regularization
    border_count=254,           # Số bins cho numerical features
    loss_function='MultiClass',  # Hàm loss cho multi-class
    eval_metric='TotalF1',      # Metric đánh giá
    cat_features=[...],         # Danh sách categorical features
```

```
    class_weights=[1.0, 6.74, 225.25],  # Trọng số lớp tùy chỉnh
    early_stopping_rounds=150,    # Dừng sớm nếu không cải thiện
    random_seed=42
)
```

Giải thích các tham số quan trọng:

- iterations=2000:** Tăng từ 1000 để mô hình học tốt hơn
- learning_rate=0.03:** Giảm từ 0.05 để hội tụ ổn định hơn
- depth=10:** Tăng từ 8 để bắt các mẫu phức tạp hơn
- l2_leaf_reg=5:** Thêm regularization để chống overfitting
- class_weights:** Xử lý imbalanced data
  - Lớp 0 (Bình thường): 1.0 (baseline)
  - Lớp 1 (Trễ): 6.74× (tăng 1.5× so với balanced weight)
  - Lớp 2 (Cực trễ): 225.25× (tăng 2.0× so với balanced weight)

Quá trình tuning:
1. Bắt đầu với tham số mặc định
2. Tăng iterations và giảm learning_rate để cải thiện
3. Tăng depth để bắt pattern phức tạp
4. Thêm regularization (l2_leaf_reg) để chống overfitting
5. Điều chỉnh class_weights để cải thiện recall cho lớp thiểu số
6. Sử dụng early_stopping để tránh overfitting

2.2. CatBoost Regressor (Giai đoạn 2 - Hồi quy)

2.2.1. Nguyên lý hoạt động

Tương tự CatBoost Classifier nhưng được tối ưu cho bài toán hồi quy:
- Dự đoán giá trị liên tục (số ngày giao hàng)
- Sử dụng loss function phù hợp với regression (MAE)
- Tối ưu hóa để giảm sai số dự đoán

2.2.2. Lý do lựa chọn

1. Hiệu suất cao: Đạt $R^2 = 0.4546$, MAE = 3.79 ngày
2. Xử lý tốt categorical features**
3. Robust với outliers: Sử dụng MAE loss
4. Tốc độ huấn luyện nhanh

2.2.3. Hyperparameter Tuning

Code python
```
CatBoostRegressor(
    iterations=2500,              # Tăng để học tốt hơn
    learning_rate=0.03,           # Giảm để ổn định
```

```
    depth=10,                  # Tăng để bắt pattern phức tạp
    l2_leaf_reg=3,             # Regularization
    subsample=0.8,             # Random sampling 80%
    border_count=254,
    loss_function='MAE',       # Tối ưu MAE
    early_stopping_rounds=150,
    random_seed=42
)
```

Mục tiêu tuning:
- Tăng R^2 từ 0.39 lên 0.45+
- Giảm MAE từ 4.02 xuống 3.79

2.3. XGBoost Regressor (Giai đoạn 2 - So sánh)

2.3.1. Nguyên lý hoạt động

XGBoost (Extreme Gradient Boosting) là thuật toán gradient boosting được tối ưu hóa cao:
- Regularized Learning: L1 và L2 regularization
- Tree Pruning: Cắt tỉa cây từ dưới lên (depth-first)
- Parallel Processing: Xây dựng cây song song
- Cache Optimization: Tối ưu sử dụng bộ nhớ

2.3.2. Lý do lựa chọn

1. So sánh với CatBoost: Đánh giá mô hình nào tốt hơn
2. Phổ biến trong industry: Được sử dụng rộng rãi
3. Hiệu suất tốt: Thường đạt kết quả cao trong competitions
4. Hỗ trợ categorical features: Từ phiên bản mới

2.3.3. Hyperparameter Tuning

Code python
```
XGBRegressor(
    n_estimators=2500,          # Số cây
    learning_rate=0.03,         # Tốc độ học
    max_depth=10,               # Độ sâu cây
    min_child_weight=3,         # Chống overfitting
    subsample=0.8,              # Random sampling
    colsample_bytree=0.8,       # Random feature sampling
    gamma=0.1,                  # Minimum loss reduction
    reg_alpha=0.1,              # L1 regularization
    reg_lambda=1.0,             # L2 regularization
    objective='reg:absoluteerror', # Tối ưu MAE
    enable_categorical=True,    # Hỗ trợ categorical
```

```
    early_stopping_rounds=150
)
```

Kỹ thuật regularization:
- L1 (reg_alpha=0.1): Feature selection
- L2 (reg_lambda=1.0): Giảm độ lớn weights
- min_child_weight=3: Ngăn cây quá chi tiết
- gamma=0.1: Chỉ split khi gain > 0.1

2.4. Kỹ thuật xử lý Imbalanced Data

2.4.1. Vấn đề

Phân phối lớp:
- Lớp 0: 92.29% (đa số)
- Lớp 1: 7.41%
- Lớp 2: 0.30% (thiểu số nghiêm trọng)

2.4.2. Giải pháp: Custom Class Weights

Công thức tính:

base_weight = total_samples / (n_classes × class_count)
custom_weight = base_weight × multiplier

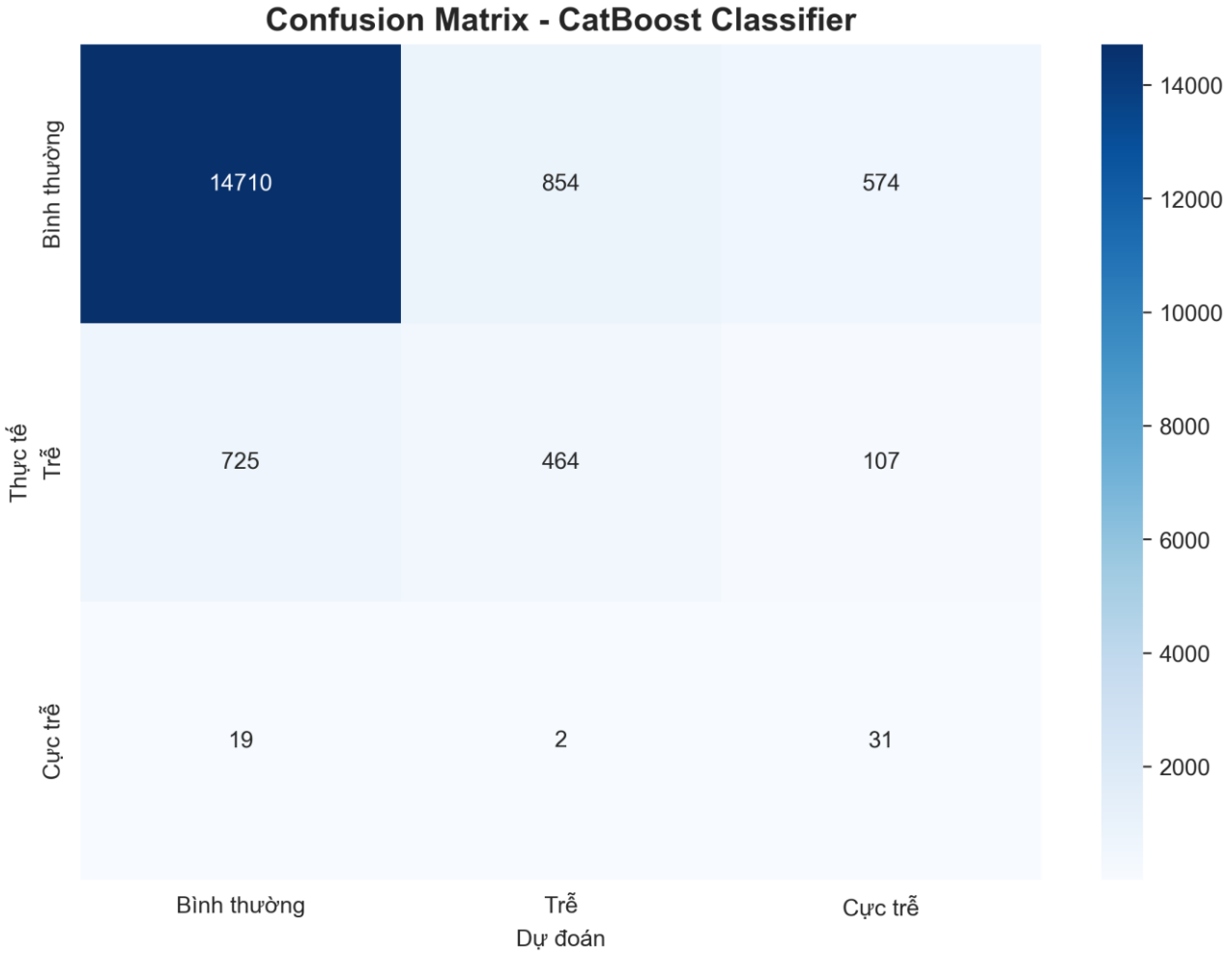
Áp dụng:
- Lớp 0: weight = 1.0 (baseline)
- Lớp 1: weight = base_weight × 1.5 = 6.74
- Lớp 2: weight = base_weight × 2.0 = 225.25

Hiệu quả:
- Tăng recall cho lớp thiểu số
- Lớp 2 (Cực trễ): recall = 0.60 (tốt cho lớp 0.30%)

3. KẾT QUẢ VÀ ĐÁNH GIÁ MÔ HÌNH

3.1. Kết quả Giai đoạn 1: CatBoost Classifier

Kết quả trên tập Test



Hình 5: Phân tích chi tiết ma trận nhầm lẫn, recall/precision từng lớp

	precision	recall	f1-score	support
0	0.95	0.91	0.93	16138
1	0.35	0.36	0.35	1296
2	0.04	0.60	0.08	52
accuracy			0.87	17486
macro avg	0.45	0.62	0.46	17486
weighted avg	0.90	0.87	0.89	17486

Recall trung bình (Macro): 0.6219

Hình 6: Báo cáo phân loại

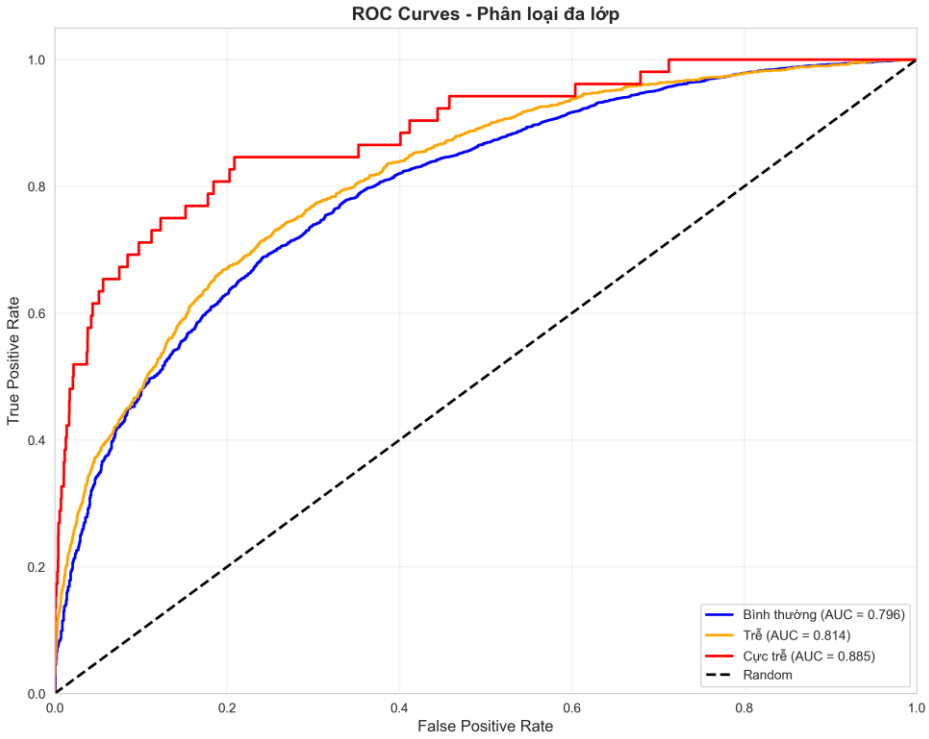
Phân tích:

- Lớp 0 (Bình thường): Precision và Recall cao (0.95 và 0.91) - Dự đoán rất tốt

- Lớp 1 (Trễ): Precision và Recall trung bình (0.35 và 0.36) - Cần cải thiện
- Lớp 2 (Cực trễ): Precision thấp (0.04) nhưng Recall cao (0.60) - Phát hiện được 60% trường hợp cực trễ, quan trọng cho business

3.1. ROC-AUC Score

ROC-AUC (One-vs-Rest, Weighted): 0.7980
Chi tiết theo lớp:
- Bình thường: 0.7964
- Trễ: 0.8145
- Cực trễ: 0.8849



Hình 7: ROC-AUC, so sánh 3 lớp

3.1. Cross-Validation (5-Fold Stratified)

Kết quả từng fold:
- Fold 1: Acc=0.9120, F1=0.5304
- Fold 2: Acc=0.9098, F1=0.5653
- Fold 3: Acc=0.9102, F1=0.5674
- Fold 4: Acc=0.9099, F1=0.5429
- Fold 5: Acc=0.9087, F1=0.5351

Nhận xét: Kết quả ổn định qua các fold , mô hình không bị overfitting.

3.2. Kết quả Giai đoạn 2: Regression Models:

CatBoost Regressor:

Kết quả trên tập Test:

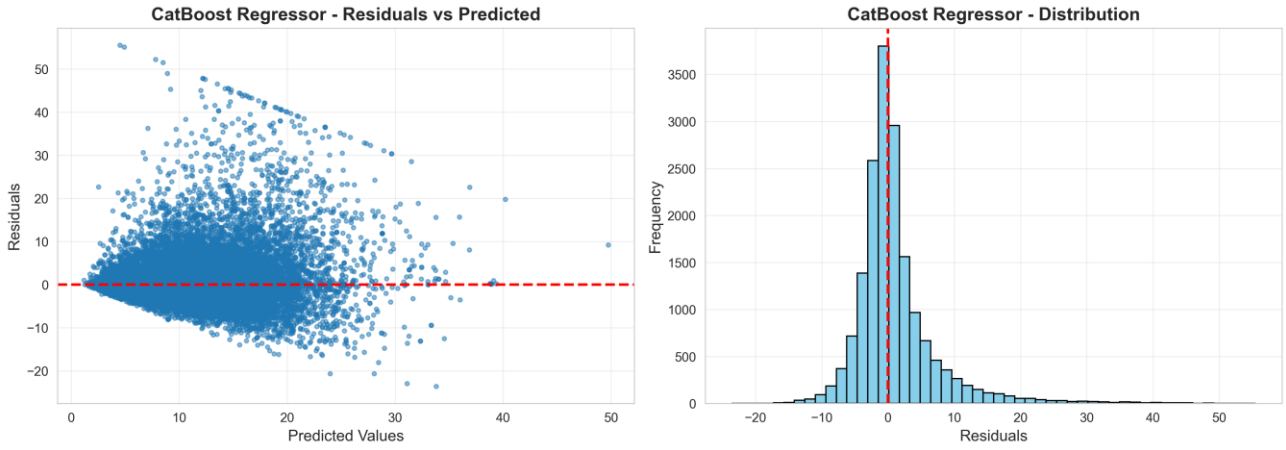
- MAE: 3.7856 ngày
- MSE: 40.1029
- RMSE: 6.3327 ngày
- R^2: 0.4546

Cross-Validation (5-Fold):

- MAE: 3.9018 ± 0.0444 ngày
- R^2: 0.4340 ± 0.0070

Kết quả từng fold:

- Fold 1: MAE=3.8835, R^2=0.4325
- Fold 2: MAE=3.9116, R^2=0.4361
- Fold 3: MAE=3.9427, R^2=0.4257
- Fold 4: MAE=3.9456, R^2=0.4294
- Fold 5: MAE=3.8254, R^2=0.4463



Hình 8: CatBoost Regressor

XGBoost Regressor:

Kết quả trên tập Test:

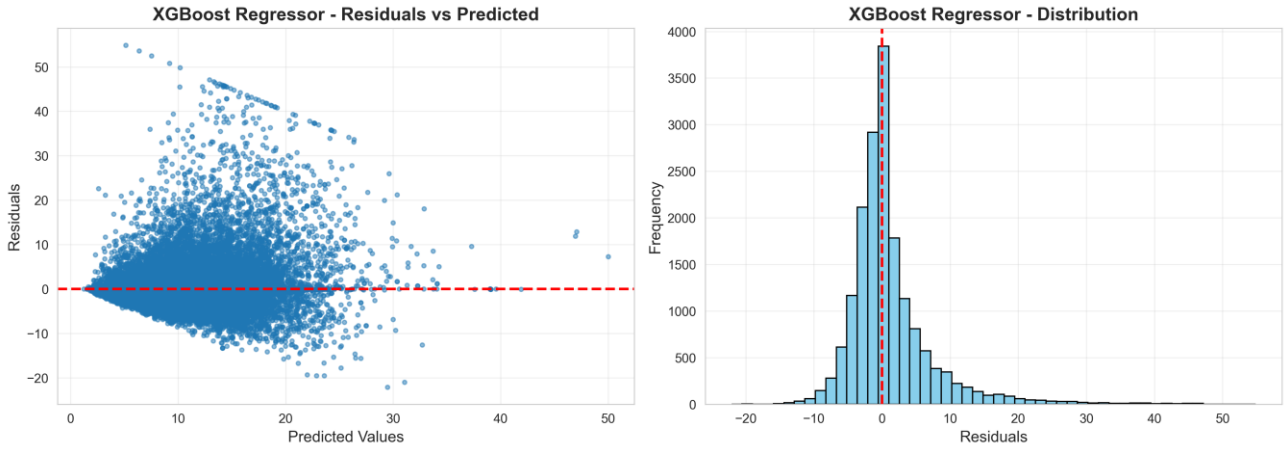
- MAE: 3.8877 ngày
- MSE: 42.7080
- RMSE: 6.5351 ngày
- R^2: 0.4192

Cross-Validation (5-Fold):

- MAE: 4.0555 ± 0.0522 ngày
- R^2: 0.3817 ± 0.0070

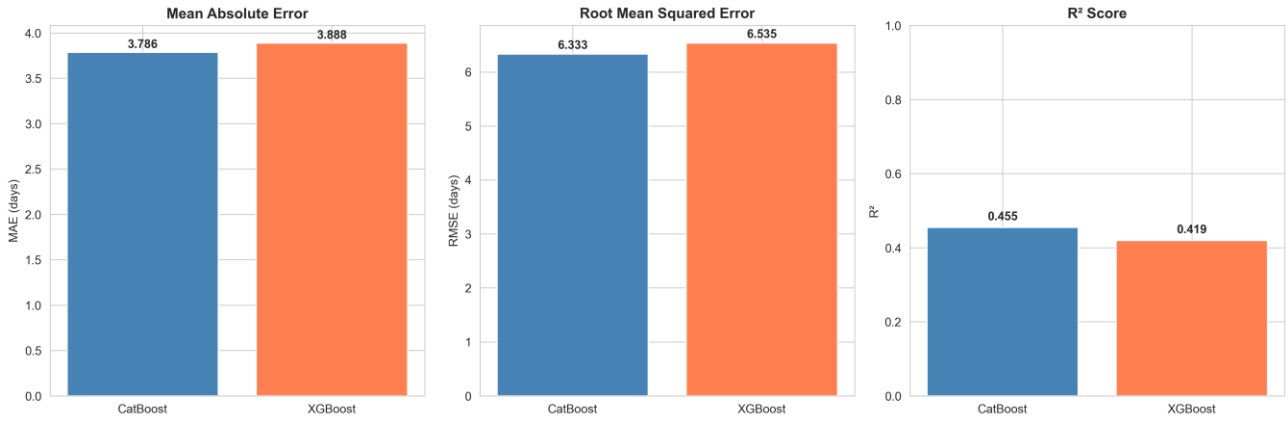
Kết quả từng fold:

- Fold 1: MAE=4.0213, R^2=0.3875
- Fold 2: MAE=4.0821, R^2=0.3807
- Fold 3: MAE=4.0949, R^2=0.3748
- Fold 4: MAE=4.1092, R^2=0.3738
- Fold 5: MAE=3.9700, R^2=0.3917



Hình 9: XGBoost Regressor

3.3. So sánh Mô hình Hồi Quy



Hình 10: So sánh metrics giữa 2 mô hình

Nhận xét:

- CatBoost Regressor tốt hơn XGBoost trên tất cả các metrics
- CatBoost có MAE thấp hơn ~ 0.1 ngày (2.6% improvement)
- CatBoost có R^2 cao hơn 0.035 (8.4% improvement)
- Mô hình được đề xuất: CatBoost Regressor

3.5 Đánh giá tổng thể

Điểm mạnh:

1. Classifier:
   - Accuracy cao: 87%
   - ROC-AUC tốt: 0.80
   - Recall cao cho lớp Cực trễ: 0.60 (quan trọng cho business)
   - Kết quả ổn định qua cross-validation

2. Regressor:
   - MAE thấp: 3.79 ngày (sai số trung bình chỉ ~3.8 ngày)
   - R^2 chấp nhận được: 0.45 (giải thích được 45% variance)
   - CatBoost vượt trội XGBoost

3. Feature Engineering:

- Các đặc trưng nâng cao (seller history, category stats, distance) có tác động lớn
- Correlation cao với target

CHƯƠNG 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

3. 1. Kết luận

Dự án đã thành công xây dựng hệ thống dự đoán giao hàng logistics với:

1. Mô hình phân loại (CatBoost Classifier):

- Accuracy: 87%
- ROC-AUC: 0.80
- Phát hiện được 60% trường hợp giao hàng cực trễ

2. Mô hình hồi quy (CatBoost Regressor):

- MAE: 3.79 ngày
- R^2: 0.45
- Dự đoán chính xác thời gian giao hàng với sai số trung bình dưới 4 ngày

3. Kỹ thuật áp dụng:

- Feature engineering phức tạp (distance, seller/category history, time features)
- Hyperparameter tuning chi tiết
- Xử lý imbalanced data bằng custom class weights
- Cross-validation để đảm bảo tính tổng quát
- So sánh nhiều mô hình (CatBoost vs XGBoost)

Ý nghĩa thực tiễn:

- Giúp doanh nghiệp dự đoán chính xác thời gian giao hàng

- Cải thiện trải nghiệm khách hàng
- Tối ưu hóa quy trình logistics
- Giảm chi phí do giao hàng trễ

3. 2. Hướng phát triển

Cải thiện mô hình:

1. Thêm features:
   - Thông tin thời tiết
   - Tình trạng giao thông
   - Vị trí kho hàng chính xác
   - Lịch sử chi tiết của shipper
2. Thử các mô hình khác:
   - LightGBM
   - Neural Networks (Deep Learning)
   - Ensemble methods (stacking, blending)
3. Xử lý imbalanced data tốt hơn:
   - SMOTE (Synthetic Minority Over-sampling)
   - ADASYN
   - Cost-sensitive learning nâng cao
4. Hyperparameter optimization:
   - Bayesian Optimization
   - Optuna
   - Grid Search chi tiết hơn

Tài Liệu Tham Khảo

1. V. H. Tiệp, *Machine Learning cơ bản.* https://www.scribd.com/document/396532696/Machine-Learning-C%C6%A1-B%E1%BA%A3n?utm_sourc
2. Nguyễn Quang Hoan (2022). *Học Máy (Machine Learning)*. Trường Đại học Thủy Lợi. https://www.studocu.vn/vn/document/dai-hoc-thuy-loi/hoc-may-nang-cao/482022bg-hoanhoc-may-thuy-loi/43470267?utm_source
3. Machine Learning cơ bản – giới thiệu thuật toán. https://machinelearningcoban.com/?utm_source