PREDICTING TIME OF ORDER DELIVERY FOR FAST FOOD

Turgunboev Dadakhon
d.turgunboev@innopolis.university

1 Motivation

Timely preparation of orders is essential for both customer happiness and business success in the highly competitive food delivery industry. Precise estimation of setup time enables efficient operations, enhanced client satisfaction, and lower operating expenses. This assignment attempts to create a model that can precisely estimate order preparation time by utilizing machine learning techniques, such as regression and classification. This would allow organizations to make data-driven decisions and increase their overall efficiency.

2 Data

CLASSIFICATION

- Features:
  - order_id: Unique identifier for each order
  - DATE_CREATE: Date and time order was created
  - region_id: Region where the order was placed
  - delivery_distance: Distance for delivery
  - STATUS_ID: Order status (encoded as numeric: 'F' - 0, 'C' - 1)
  - Profit: Profit associated with the order
  - product_count: Number of items in the order
  - total_price: Total price of the order
  - Order_start_prepare: Commencement of order prep
  - Order_ready_date: Data the order will be ready
  - planned_prep_time: Planned prep time for the order
  - Extracted time Features: Year, month, day, hour, minute, second for relevant dates
- Predictor: y_train, y_test: (accurate or inaccurate prediction)

$$y\_train, y\_test = order\_ready\_date - order\_start\_prepare$$

REGRESSION

- Features:
  - order_id: Unique identifier for each order
  - DATE_CREATE: Date and time order was created
  - region_id: Region where the order was placed
  - delivery_distance: Distance for delivery
  - STATUS_ID: Order status (encoded as numeric: 'F' - 0, 'C' - 1)
  - Profit: Profit associated with the order
  - product_count: Number of items in the order
  - total_price: Total price of the order
  - Order_start_prepare: Commencement of order prep
  - Order_ready_date: Data the order will be ready
  - planned_prep_time: Planned prep time for the order
  - Extracted time Features: Year, month, day, hour, minute, second for relevant dates

3 Exploratory data analysis

The original dataset contained missing values and complex features.Some of the features needed to be converted to numeric format in order to use them in training the model. The initial task is to handle the missing values by:
Planned_prep_time: Missing values filled with mean.
DATE_CREATE: Rows with missing DATE_CREATE values were dropped, as this is likely a crucial column for time-based analysis.
order_ready_date and order_start_prepare: Missing values in these columns were filled using the mean time difference between the respective date and DATE_CREATE for each store.
order_ready_date and order_start_prepare: Missing values in these columns were filled using the mean time difference between the respective date and DATE_CREATE for each store.

Time Difference Features: The time_diff_start and time_diff_ready features were created to capture the time difference between order_start_prepare and DATE_CREATE, and between order_ready_date and DATE_CREATE respectively.

Date Feature Extraction: The extract_date_feats function extracted year, month, day, hour, minute, and second features from the datetime columns DATE_CREATE, order_ready_date, order_start_prepare, and order_pickup_date. This can capture temporal patterns and trends.
Constant Value Removal: Columns with constant values (e.g., the year component from all datetime columns) were removed to reduce dimensionality and avoid redundant features.

time_diff_start and time_diff_ready: These features were dropped after extracting the individual date components, as they would become redundant.

4 Task

5 Task

Regression Task: Estimate delivery time based on multiple features. Classification Task: Predict delivery success based on categorical and numerical inputs.

5.1 Regression

Input: A vector, $\vec{X}$ representing the characteristics of an order.
Output: A continous value, Y representing the predicted preparation time in minutes.
Estimating Function: The goal is to learn a function f(x), that maps the input features X to the predicted preparation time, Y,such that the error or cost function is minimize
f(x) = $\vec{W}\vec{X}$ + b
Where:
$\vec{X}$ = [order_id STATUS_ID store_id planned_prep_time DATE_CREATE region_id order_ready_date order_start_prepare delivery_distance profit order_pickup_date product_count total_price].
$\vec{W}$ = The vector weights of the input features
b = The bias term
The model learns optimal values of vector W, and the term b to minimize the prediction error e.g the mean square error

5.2 Classification

Input: A vector, $\vec{X}$ representing the characteristics of an order.
Output:A **binary** value, Y representing whether the planned preparation time is accurate(1) or not(0).
Estimation Function: The goal is to learn a function f(x) that maps the input features x to a probability of the planned preparation time being accurate. The function that does this mapping is the sigmoid function.
f(x) = $\vec{W}.\vec{x}$ + b
Sigmoid (f(x)) = $\frac{1}{1+e^{-f(x)}}$
Where:
$\vec{X}$ = [order_id STATUS_ID store_id planned_prep_time DATE_CREATE region_id order_ready_date order_start_prepare delivery_distance profit order_pickup_date product_count total_price].
$\vec{W}$ = The vector weights of the input features
The predicted class label is determined by thresholding the output probability

6 Results

He regression models occasionally displayed overfitting; we applied regularization techniques and cross-validation checks. The classification models demonstrated good generalization ability, with Logistic Regression showing lower recall, indicating potential improvements in capturing positive classes.

Table 1. Task 1

Model	RMSE.	MSE	MAE	R2
Linear	7.4	5.11	4.5	0.59
Lasso	8.73	76.31	6.34	0.34

Table 2. Task 2

Model	Acc.	Recall	Precision	F1-score
Logistic	0.56	0.35	0.55	0.5
SVM	0.5	0.01	0.5	0.5

7 Data Imbalance

In our dataset, we observed a significant imbalance between the classes, which can adversely affect the performance of classification algorithms. The distribution of classes led to various challenges:

Impact on Logistic Regression: The low recall (0.357) indicates that the model struggled to identify the positive class effectively, likely due to the imbalance. Precision at 0.553 suggests that while the model does make positive predictions, many of them may not be true positives.

8 Conclusion

The analysis revealed notable results from both regression and classification models applied to the dataset. While Linear Regression demonstrated a reasonable performance with an R^2 of 0.571, Lasso Regression's results indicated a potential for overfitting with lower R^2 values. Logistic Regression's accuracy was moderate at 56.5

To enhance model performance, especially in classification tasks, addressing data imbalance through techniques such as resampling, synthetic data generation, and cost-sensitive learning is essential. This will not only improve recall and precision but also provide a more reliable framework for decision-making based on predictive analytics.