Machine learning assignment 1

Anastasia Luzinsan
a.luzinsan@innopolis.university

1 Motivation

The primary goal of predicting order preparation time is to reduce delays and optimize the delivery process. Accurate predictions can improve delivery time estimates displayed to customers during the preparation phase, enhancing customer satisfaction and enabling more efficient restaurant workflows. This task focuses on features known at the time of order creation and the start of preparation, excluding any later information. The objective is to build a model that predicts actual preparation time based on these early-stage features, thus mitigating delays and optimizing operational efficiency. Additionally, the classification task is motivated by risk factors to reassess planned preparation time, providing a better understanding of potential risks associated with delays and offering opportunities for real-time adjustments, ultimately leading to more accurate delivery forecasts.

2 Data

The data for both regression and classification tasks were derived from several interconnected tables: order_props_value, order_history, order_basket, and products. An important aspect of data preparation was handling the order_props_value table, which contained various types of values restored through a pivot process by matching them to their descriptions in order_props. Key features were extracted, including region_id, order_ready, order_start_prepare, delivery_distance, order_pickup, and profit.

The tables were merged into a single dataset using a join on store_id and order_id. During preprocessing, missing values in the price column were imputed (most were restored based on other rows with the same product), and aggregated data was processed at the order level. The target variable for regression was calculated as the difference between order_ready and order_start_prepare, termed actual_prep_time. In the classification task, the target is binary, equaling 1 if the absolute difference between planned_prep_time and actual_prep_time does not exceed 5 minutes, and 0 otherwise.

The final dataset contains 16 features and 517,610 records, including store_id, profit, delivery_distance, date_create, order_start_prepare, planned_prep_time, order_ready, order_pickup, region_id, status_id, products_count, order_price, max_price, min_price, avg_price, and unique_products_sold_by_store.

3 Exploratory Data Analysis

During exploratory data analysis (EDA), several key insights were uncovered. Many features exhibited significant outliers both globally and within individual stores. A strong correlation was observed between profit and order price. The status_id column was highly imbalanced, with 99% "F" and only 1% "C" , leading to its removal due to low informativeness.

Some date columns had incorrect chronological orders, which were corrected to ensure the logical sequence of events: date_create ≤ order_start_prepare ≤ order_ready ≤ order_pickup. This sorting was applied to rows with complete date information. Based on insights from EDA, several preprocessing steps were undertaken. Outliers were addressed using the interquartile range (IQR) method to prevent distortion of model predictions. The dataset was split into training and test sets to prevent information leakage. Missing values in date columns were imputed based on adjacent columns; if date_create was missing, it was filled using the formula order_start_prepare - mean(order_start_prepare - date_create), ensuring logical imputation within each store.

In regression target variable, records yielding null or negative values, along with those significantly deviating from planned_prep_time, were removed. The columns order_ready and order_pickup were dropped to mitigate data leakage. After analyzing the classification target, it was determined that the dataset was balanced, with 52.2% late orders and 47.8% on-time orders. Correlation analysis revealed minimal relationships with the target variable, with only slight correlation with planned_prep_time.

Features derived from date columns were transformed using sine and cosine functions to capture their cyclical nature. Remaining missing values in profit, max_price, and min_price were first imputed using the median within each store. Any remaining values were addressed using the SimpleImputer with median imputation. For encoding categorical features, the CatBoostEncoder was chosen for its effectiveness with high-cardinality variables while minimizing overfitting risks. The MinMaxScaler was applied to numeric features to maintain their relationships while normalizing the data, ensuring equal contribution to model performance.

4 Task

4.1 Regression

For the regression task, the goal is to predict the actual preparation time of orders, termed ‘actual_prep_time‘, using remaining features. The performance of regression models will be evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2). MAE offers a clear interpretation of average error, while MSE emphasizes larger

errors. R^2 indicates how well the model explains the variance in the target variable.

We will start with linear regression as a baseline model, followed by Lasso regression to identify significant features through L1 regularization. Next, we will apply Ridge regression with these features, and we will also train Random Forest Regressor and CatBoost Regressor to capture more complex relationships.

4.2 Classification

For the classification task, we aim to predict whether an order will be delivered on time based on ‘actual_prep_time‘ and ‘planned_prep_time‘. Model performance will be assessed using accuracy, precision, recall, F1 score, and ROC-AUC metrics. These metrics provide insights into overall performance and class balance, especially useful in scenarios with imbalanced classes.

As a baseline for classification, we will use logistic regression, followed by Support Vector Classifier (SVC) for non-linear decision boundaries. Finally, we will design a simple neural network architecture to explore its advantages in modeling complex relationships.

5 Results

5.1 Regression Results

The following results were obtained for different regression models:

Table 1. Performance metrics of regression models

Model	MAE	MSE	RMSE	R^2
Linear Regression	4.88	37.43	6.12	0.55
Lasso Regression	4.88	37.46	6.12	0.55
Ridge Regression	4.88	37.43	6.12	0.55
Random Forest Regressor	4.85	37.22	6.10	0.55
CatBoost Regressor	4.75	35.55	5.96	0.57

Among the regression models, the CatBoost Regressor performed the best, achieving the lowest MAE, MSE, and RMSE, as well as the highest R^2 score. This indicates that it was able to capture the underlying patterns in the data more effectively than the other models.

5.2 Classification Results

For the classification task, the following results were obtained:

The Logistic Regression model served as the baseline, yielding reasonable results, but further experiments indicated a need for improved accuracy and other performance metrics. The model demonstrated moderate precision and recall, indicating that it struggles with false positives and negatives.

Table 2. Performance metrics of classification models

Model	Acc	Recall	Prec	F1	ROC-AUC
LogReg	0.567	0.519	0.514	0.517	0.562
SVC	0.553	0.513	0.499	0.506	0.549
NeuralNet	0.54	0.08	0.48	0.14	0.5

5.3 Model Assessment and Overfitting/Underfitting

Cross-validation results indicate that while the CatBoost Regressor performed well, it is at risk of overfitting due to its complexity and tendency to capture noise in the training data. Regularization techniques in Lasso and Ridge regression helped mitigate this risk, enhancing generalization on unseen data. For classification models, the relatively balanced values of precision, recall, and F1 score suggest adequate performance, but further tuning may be necessary to improve these metrics.

6 Data Imbalance

Although the dataset remained balanced after preprocessing, techniques such as SMOTE and Borderline SMOTE were employed to evaluate their impact on model performance. The results for logistic regression using SMOTE indicates an accuracy of 0.567, with precision, recall, F1 score, and ROC AUC values of 0.514, 0.534, 0.524, and 0.564, respectively. Cross-validation results showed consistent performance with average F1 scores around 0.56 to 0.57. In comparison, the Borderline SMOTE technique yielded similar metrics, with an accuracy of 0.566 and slightly improved recall (0.535). Overall, both methods demonstrated similar effectiveness in balancing the dataset; however, the improvements in model performance were minimal, underscoring the robustness of the original balanced dataset.

7 Conclusion

The most significant factors influencing the timely preparation of orders include ‘planned_prep_time‘, ‘store_id‘, ‘order_price‘, and ‘max_price‘. The high importance of ‘planned_prep_time‘ underscores the necessity of effective advance planning, while ‘store_id‘ indicates disparities in efficiency across different stores. Price-related factors, such as ‘order_price‘ and ‘max_price‘, suggest that higher-value orders may be associated with delays, indicating a need for process optimization. Additionally, the variable ‘product_count‘ reflects the role of the number of items in an order, alongside temporal factors like ‘order_delay‘. It is recommended to optimize operations in stores with high ‘store_id‘ values, analyze delays for expensive orders, enhance logistics for complex orders, and implement more accurate preparation time planning.