COMP1816 - Machine Learning Coursework Report

Md Abdul Raihan Tanzim - 001341954
Word Count: No more than 3500 words (edit here)

1. Introduction

This report analyses a historical food delivery dataset (1,000 orders; 18 columns) to support three operational objectives: (i) understand order patterns via exploratory analysis, (ii) predict delivery time at the moment an order is placed, and (iii) predict whether a complaint will occur, again using only creation-time information to avoid leakage. Following the coursework specification, First summarise key data properties and relationships, then segment orders with clustering and present representative orders for each segment, and finally build and evaluate supervised machine learning models for regression and classification.

Summary of the data and baseline models. Delivery times average 47.27 minutes (median 44.27) and exhibit a pronounced right tail (max 133.43), indicating occasional severe delays. Complaints occur in 24.4% of orders, so complaint prediction is an imbalanced classification problem. Unsupervised structure is weak: K-means selected $k = 2$ by silhouette, but the silhouette score is low (0.148). For delivery-time regression, a tuned Random Forest achieved MAE $= 12.57$ minutes and $R^2 = 0.208$ on the test set. For complaint prediction, a tuned Balanced Random Forest achieved ROC-AUC $= 0.692$, PR-AUC $= 0.482$, and F1 $= 0.452$ (threshold 0.5).

2. Data Exploration

2.1. EDA

2.1.1. Dataset Overview and Structure

The dataset contains 1,000 rows and 18 columns. Features include market/area, store category, order protocol, basket features (items and prices), supply–demand indicators (on-shift partners, busy partners, outstanding orders), and creation-time context (day, holiday flag, `created_at`). Two derived targets were created:

`delivery_mins`: time between `created_at` and `actual_delivery_time` (minutes), with a midnight-crossing correction when the delivery completion occurs after midnight.

`complaint_flag`: binary complaint indicator derived from the `complaint` field, treating missing complaint values as `NO`.

These design choices follow the requirement to clearly define targets and to use only creation-time information for prediction.

Table 1. Dataset summary statistics

Property	Value
Rows	1000
Columns	18
Complaint rate (`complaint_flag=1`)	0.244
Delivery minutes mean (SD)	47.27 (17.06)
Delivery minutes median (IQR)	44.27 (35.42–55.78)
Delivery minutes min / max	16.12 / 133.43

Table 1 provides a high-level snapshot of the dataset and the two modelling targets. The data comprise 1,000 orders described by 18 variables, with complaints observed for 24.4% of orders (`complaint_flag=1`), confirming that complaint

prediction is an imbalanced classification problem. Delivery duration (`delivery_mins`) has mean 47.27 minutes (SD 17.06) and a median of 44.27 minutes with an interquartile range of 35.42–55.78, indicating a right-skewed distribution where occasional delays increase the mean. The minimum and maximum delivery times (16.12 and 133.43 minutes) further highlight the presence of rare but severe delays.

Figure 1 summarises the dataset's relational structure. `ORDER` is the central table and links to three reference tables via foreign keys: `MARKET` (`market_id`), `STORE_CATEGORY` (`category_id`), and `ORDER_PROTOCOL` (`protocol_id`). Each order produces a corresponding record in `DELIVERY` (joined on `order_id`), which contains the realised delivery completion time (`actual_delivery_time`) and operational supply–demand indicators (e.g., `total_onshift_partners`, `total_busy_partners`, `total_outstanding_orders`). Complaints are stored in `COMPLAINT` and are linked back to orders through `order_id`; because not every order has a complaint record, complaint status is optional at the schema level and is therefore represented using the derived binary target `complaint_flag`.

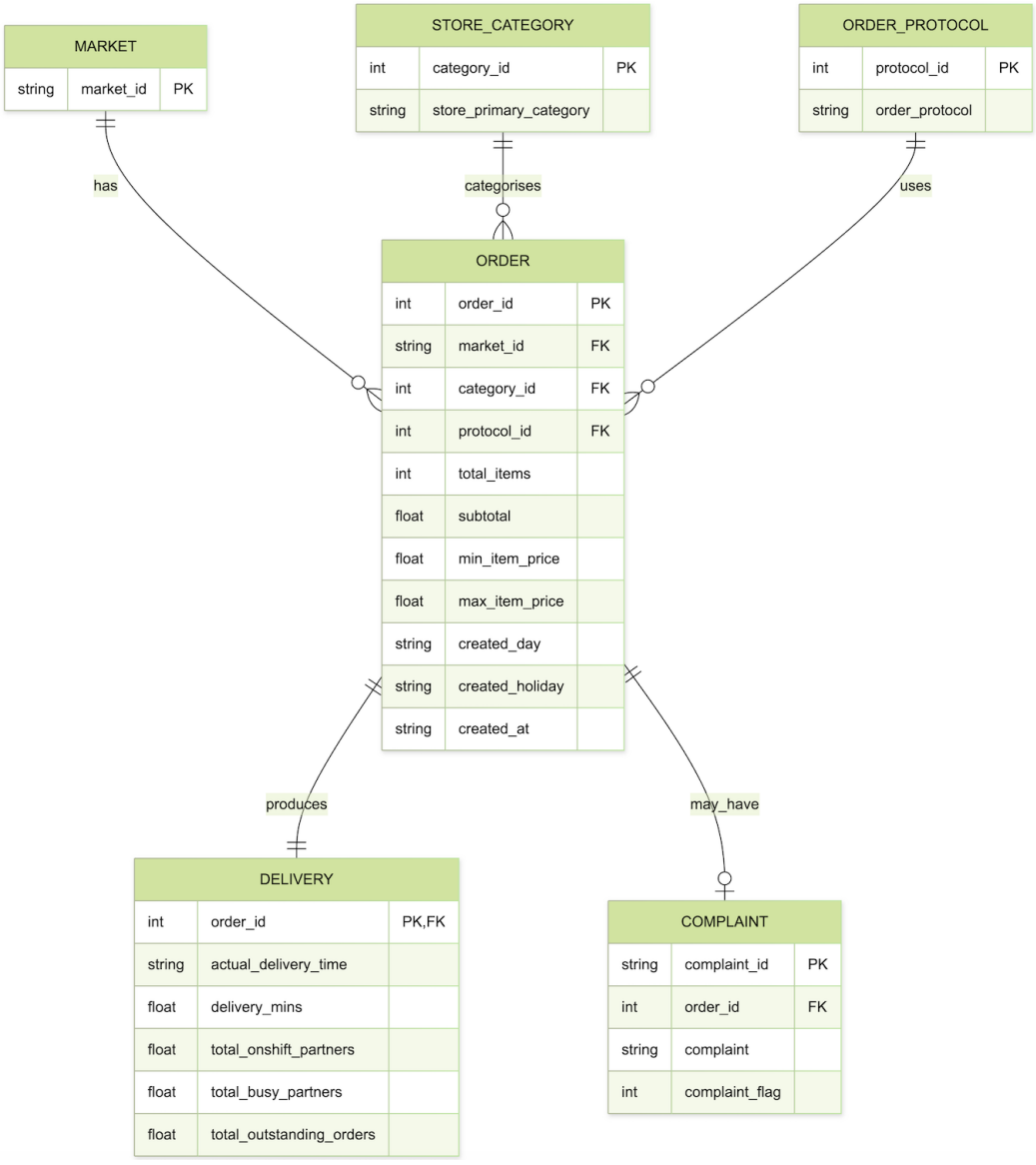

Figure 1. Conceptual ERD for the Food Delivery Dataset (Normalised Schema)

2.1.2. Missing Data and Practical Implications

As shown in Figure 2, missingness is concentrated in the complaint-related fields (`complaint` and `complaint_id`), suggesting that complaint records are only populated for a subset of orders. Therefore, missing `complaint` values are treated as `NO` when constructing the binary target, while the smaller amounts of missingness in operational variables are handled using median/mode imputation to avoid discarding observations.

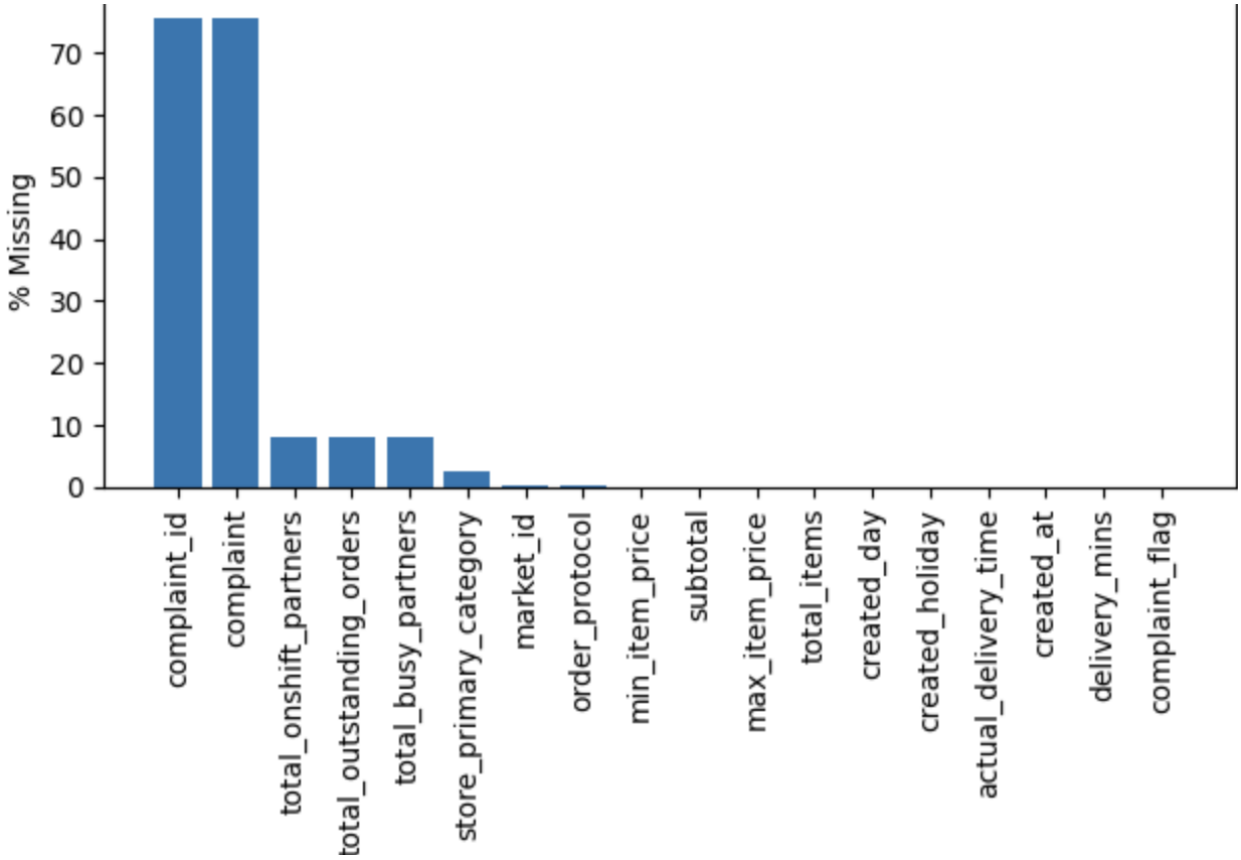

Figure 2. Percentage of missing values per feature in the Food Delivery dataset.

Complaint-related fields (`complaint`, `complaint_id`) have the highest missingness ($\approx 75.6\%$), while partner workload variables have moderate missingness ($\approx 8\%$).

Table 2 summarises the extent of missingness in key variables. The complaint-related fields (`complaint_id` and `complaint`) are missing for 75.6% of orders, consistent with complaints being recorded only for a subset of transactions. In contrast, operational supply–demand indicators (`total_onshift_partners`, `total_busy_partners`, `total_outstanding_orders`) exhibit modest missingness (8.0%), and `store_primary_category` has minimal missingness (2.6%), suggesting that simple imputation strategies are unlikely to distort the overall feature distributions.

A key operational interpretation is that complaint records appear to exist only for a subset of orders. Treating missing complaint entries as `NO` complaint is plausible because the derived complaint rate equals 24.4%, which matches the share of non-missing complaint fields. Nevertheless, this assumption may be violated if some complaints were unrecorded; in that case, the target label contains noise, which reduces the best achievable predictive performance.

For the remaining missing values, median imputation for numeric variables and most-frequent imputation for categorical variables is reasonable because (i) missingness is modest (approximately 8% for the partner metrics) and (ii) these features are used broadly across the modelling pipeline, so a simple, stable strategy supports consistent model comparison.

Table 2. Missing-value summary for key variables

Variable	Missing (%)
complaint_id	75.6
complaint	75.6
total_onshift_partners	8.0
total_busy_partners	8.0
total_outstanding_orders	8.0
store_primary_category	2.6

2.1.3. Distribution of Delivery Times

Figure 3 shows that delivery times are concentrated around the mid-range (roughly 30–60 minutes) with a pronounced right tail extending beyond 100 minutes. This implies the presence of occasional extreme delays, which are operationally important and can inflate error measures that penalise large deviations. Consequently, reporting both MAE (typical error in minutes) and RMSE (more sensitive to rare large errors) is appropriate for evaluating regression models on delivery_mins. The distribution is unimodal with a right tail, indicating that while most deliveries occur around typical times, a smaller number of orders experience substantially longer delays.

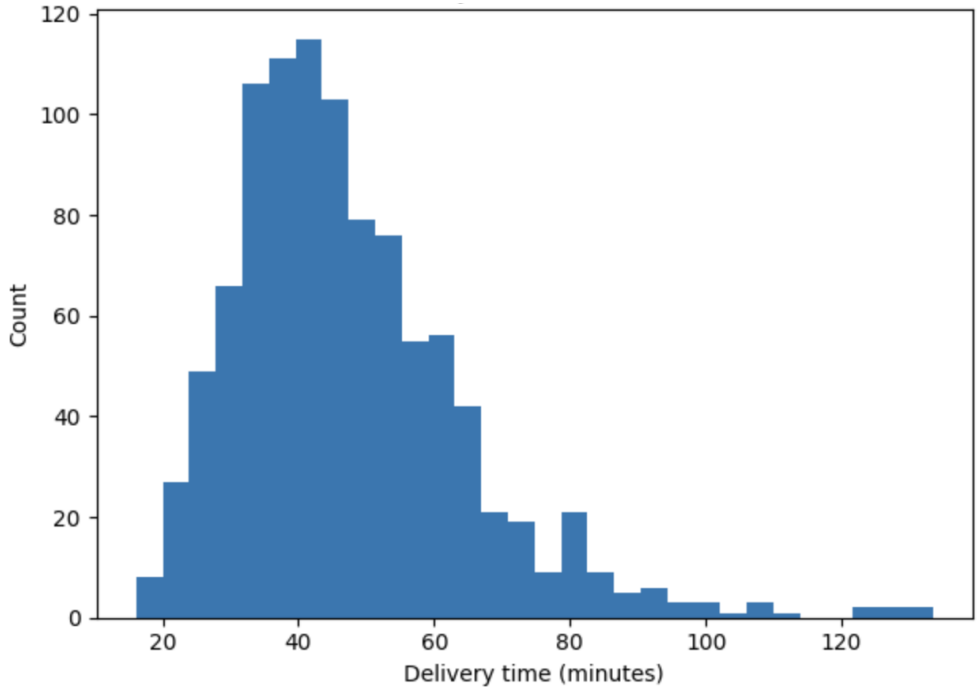

Figure 3. Histogram of the delivery-time target delivery_mins.

Figure 4 explores whether more expensive orders tend to take longer to deliver. The plot does not indicate a clear linear association between subtotal and delivery_mins; instead, delivery times remain widely spread across most subtotal values. Notably, high-subtotal orders exhibit increased dispersion, which is consistent with heteroscedasticity and occasional extreme outcomes. This motivates the use of non-linear models (e.g., tree ensembles) that can capture interaction effects and are less sensitive to outliers than purely linear approaches.

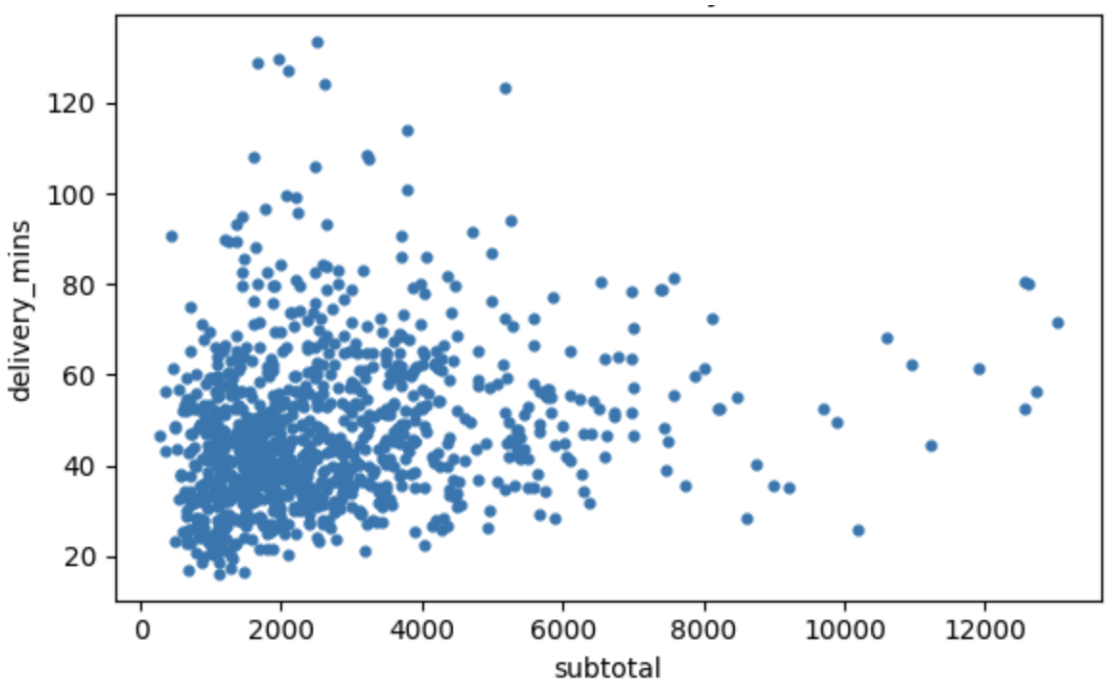

Figure 4. Scatter plot of `subtotal` (order value) versus `delivery_mins`.

The relationship is not strongly linear; however, higher subtotals show greater variability in delivery time, suggesting heteroscedasticity and the presence of outliers.

Figure 5, several operational and pricing features exhibit strong right-skew (e.g., `subtotal`, `total_outstanding_orders`), meaning that a small number of orders occur under extreme values. The target `delivery_mins` is approximately unimodal but also has a noticeable right tail, consistent with rare but severe delivery delays. These distributional properties motivate the use of robust imputation (median for numeric variables) and favour non-linear models (e.g., tree ensembles) that can capture asymmetric and interaction-driven effects without requiring strict normality assumptions.

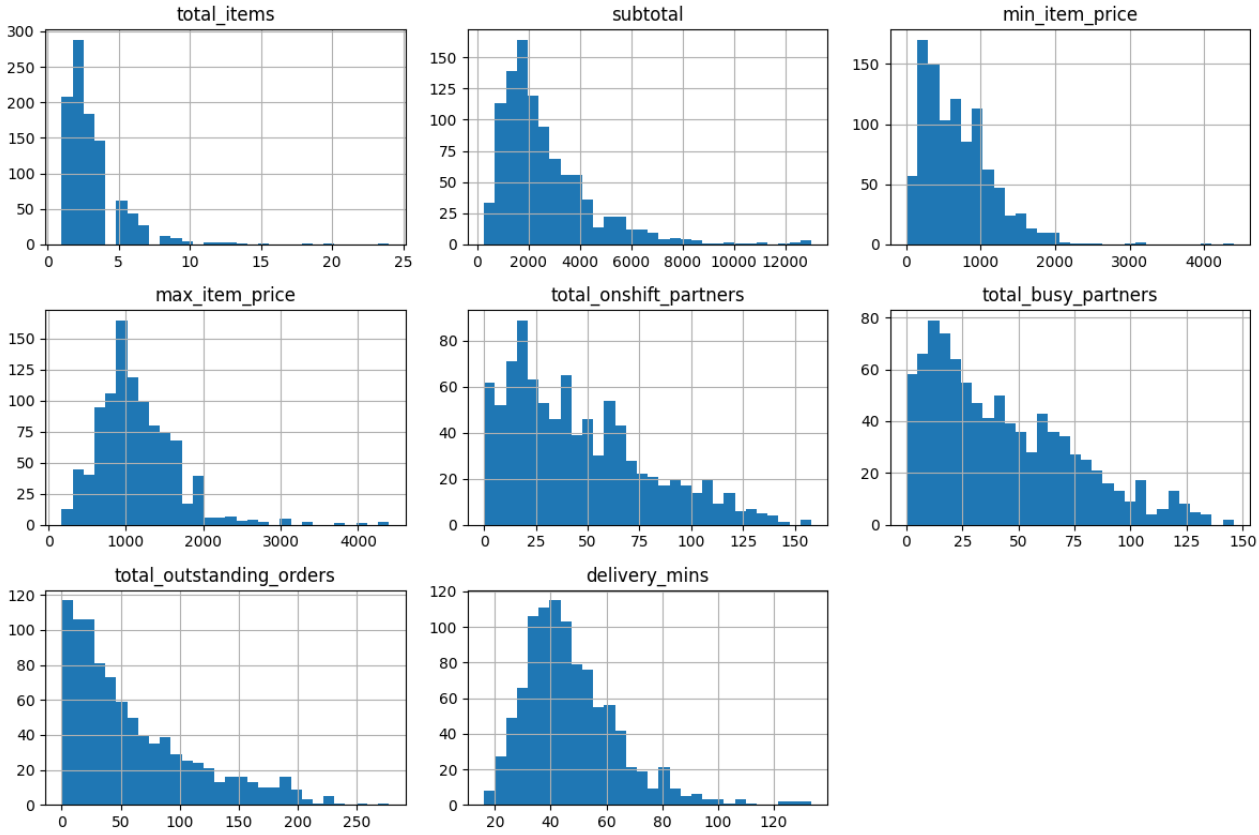

Figure 5. Histograms of key numeric variables in the Food Delivery dataset.

Including basket size/value (e.g., `total_items`, `subtotal`), price variables, courier capacity indicators, outstanding orders, and the regression target `delivery_mins`. Most variables are positively skewed with long right tails, indicating occasional extreme order values and high-demand conditions.

Table 3 summarises the distribution of delivery duration (`delivery_mins`). The mean delivery time is 47.27 minutes (SD 17.06), while the median is lower at 44.27 minutes, indicating a right-skewed distribution. The interquartile range spans from 35.42 minutes (Q1) to 55.78 minutes (Q3), suggesting that the middle 50% of deliveries fall within a relatively tight window. However, the maximum delivery time of 133.43 minutes compared with a minimum of 16.12 minutes highlights occasional extreme delays, consistent with a long right tail.

Table 3. Summary statistics for `delivery_mins`

Statistic	**Value (minutes)**
Mean	47.27
Standard deviation	17.06
Minimum	16.12
First quartile (Q1)	35.42
Median	44.27
Third quartile (Q3)	55.78
Maximum	133.43

The long upper tail implies that large-error penalties matter, so RMSE is informative alongside MAE for regression evaluation (RMSE emphasises rare but operationally expensive extreme delays).

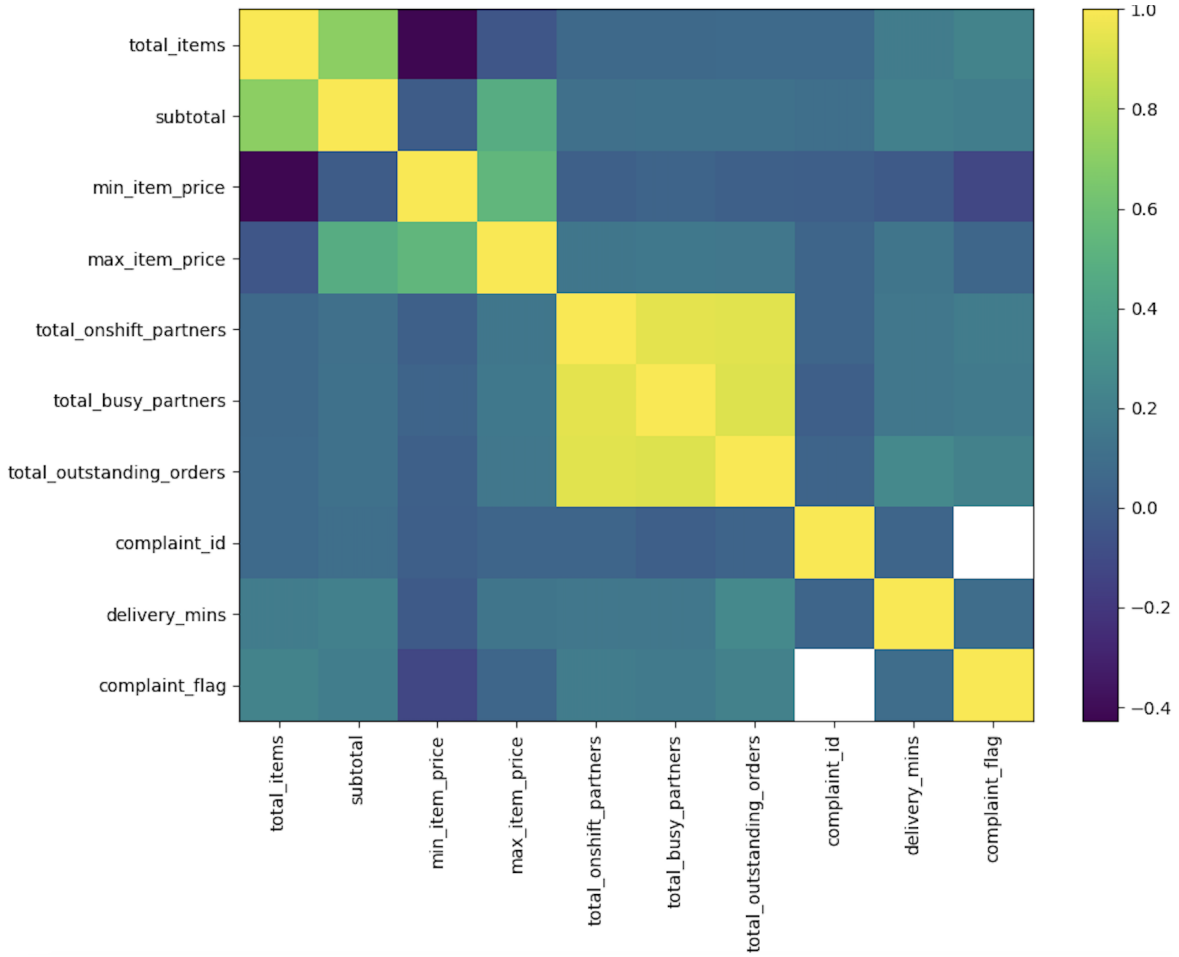

Figure 6. Correlation heatmap for numeric variables in the Food Delivery dataset.

Figure 6 summarises linear relationships between numeric features. The most prominent pattern is the high positive correlation between workforce and congestion indicators (`total_onshift_partners`, `total_busy_partners`, and `total_outstanding_orders`), which is expected because these variables describe related aspects of system load. In contrast, `delivery_mins` and `complaint_flag` show relatively weak pairwise correlations with any single variable, suggesting that delivery time and complaints are likely driven by non-linear effects and interactions (e.g., time-of-day combined with congestion), rather than a single dominant linear predictor. This supports the later use of engineered ratio features (such as `busy_ratio` and `outstanding_per_partner`) and non-linear models (tree ensembles) that can capture interaction structure more effectively than purely linear methods. Strong positive correlations are visible among operational capacity variables (e.g., `total_onshift_partners`, `total_busy_partners`, and `total_outstanding_orders`), while delivery time and complaint indicators show weaker linear correlations with individual predictors.

2.1.4. Complaint Prevalence and Imbalance

Complaint counts comprise 756 non-complaints and 244 complaints, giving a prevalence of 0.244. This class imbalance makes accuracy a poor primary metric: a naive classifier that predicts *no complaint* for every order would achieve 75.6% accuracy, yet provide little operational value. Instead, threshold-robust metrics such as ROC-AUC and PR-AUC are more appropriate because they evaluate ranking performance across all possible decision thresholds. In particular, PR-AUC is especially informative in imbalanced settings where the positive class (complaints) is the minority.

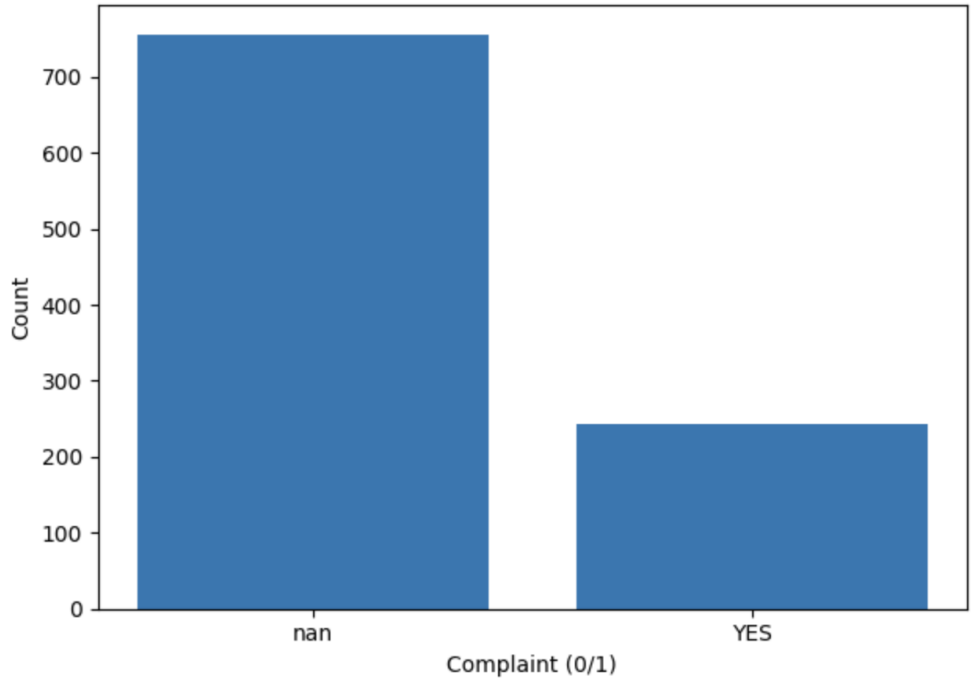

Figure 7. Distribution of the complaint field in the dataset.

As shown in Figure 7, complaint labels are not fully populated: the majority of orders have missing (`NaN`) values and only a minority are marked `YES`. Since the dataset appears to only record complaint details when a complaint occurs, missing values are treated as `NO` when creating `complaint_flag`. This results in an imbalanced classification problem (approximately 24.4% positive class), so metrics such as PR-AUC and F1 score are more informative than accuracy, and class-balancing methods are justified. Most records have missing (`NaN`) complaint entries, while a smaller subset is labelled `YES`. This motivates constructing a binary target `complaint_flag` by treating missing complaints as `NO`, and using imbalance-aware evaluation metrics for classification.

2.1.5. Relationships and Modelling Implications

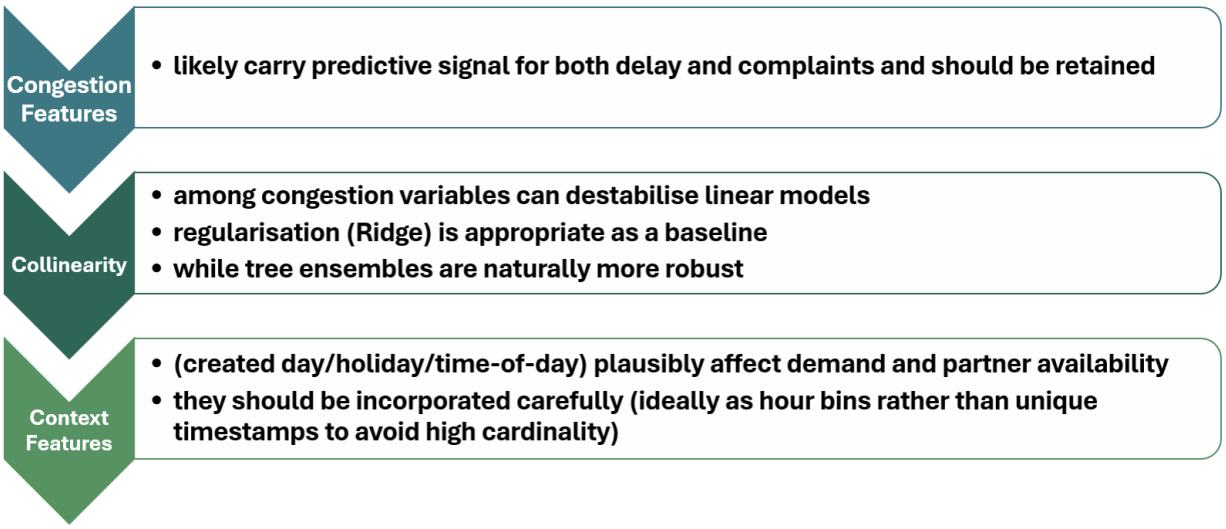

Figure 8. Conceptual summary of feature groups and modelling implications.

As summarised in Figure 8, congestion indicators (e.g., partner availability and outstanding orders) are expected to influence both delay and complaint risk and should therefore be retained. However, these variables are also strongly correlated, which can inflate variance in coefficient-based linear models; consequently, regularised linear baselines (Ridge/logistic regression with class weights) are appropriate. In contrast, tree ensembles are less sensitive to collinearity and can capture non-linear interactions between congestion and temporal context features, motivating their use as the primary models in Tasks B and C. From an operational perspective, the feature set suggests several plausible drivers of both delivery duration and complaint risk:

- **Demand intensity.** `total_outstanding_orders` and the workload ratio

$$\texttt{busy\_ratio} = \frac{\texttt{total\_busy\_partners}}{\texttt{total\_onshift\_partners}},$$

  where higher workload is expected to increase delivery times and, indirectly, complaint probability.

- **Supply availability.** Available capacity measured by

$$\texttt{free\_partners} = \texttt{total\_onshift\_partners} - \texttt{total\_busy\_partners},$$

  where greater capacity is expected to reduce delivery times.

- **Basket complexity and value.** Basket size and price characteristics (e.g., `total_items`, `subtotal`, `price_range`, `avg_item_price`) may proxy preparation complexity; larger or higher-variance orders may take longer and may increase complaint risk when delays occur.

- **Temporal effects.** Day-of-week and time-of-day patterns capture systematic variation such as peak demand periods (e.g., rush hours) versus low-demand periods (e.g., overnight).

These hypotheses motivate the feature engineering introduced later, including ratio features and cyclical encodings for time variables, since the underlying effects are expected to be non-linear and periodic.

2.2. Clustering

2.2.1. Method Selection and Justification (K-Means)

K-means clustering was applied to engineered and preprocessed creation-time features. This method is suitable because the dataset is moderate in size ($n = 1{,}000$), preprocessing yields a feature space with many numeric signals (after encoding and scaling), and K-means is computationally efficient for discovering broad, coarse-grained "order types".

Formally, K-means partitions observations into k clusters by minimising the within-cluster sum of squares (inertia):

$$\min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in C_j} \left\|\mathbf{x}_i - \boldsymbol{\mu}_j\right\|_2^2, \tag{1}$$

where $\boldsymbol{\mu}_j$ denotes the centroid of cluster C_j.

Why not choose a different clustering method? Given these trade-offs and the goal of an interpretable segmentation with representative examples, **K-means** is a defensible baseline.

Why not hierarchical clustering? After one-hot encoding expands the feature space, hierarchical clustering becomes harder to interpret: distance-based merges are less meaningful and results can vary substantially depending on the linkage criterion (e.g., single, complete, average), making justification less straightforward in this setting.

Why not DBSCAN? Although DBSCAN can identify non-convex clusters, it requires careful tuning of density parameters and is sensitive to distance concentration in high-dimensional encoded spaces. With mixed categorical and numeric information expanded via one-hot encoding, DBSCAN often labels a large fraction of observations as noise.

2.2.2. Selecting the Number of Clusters

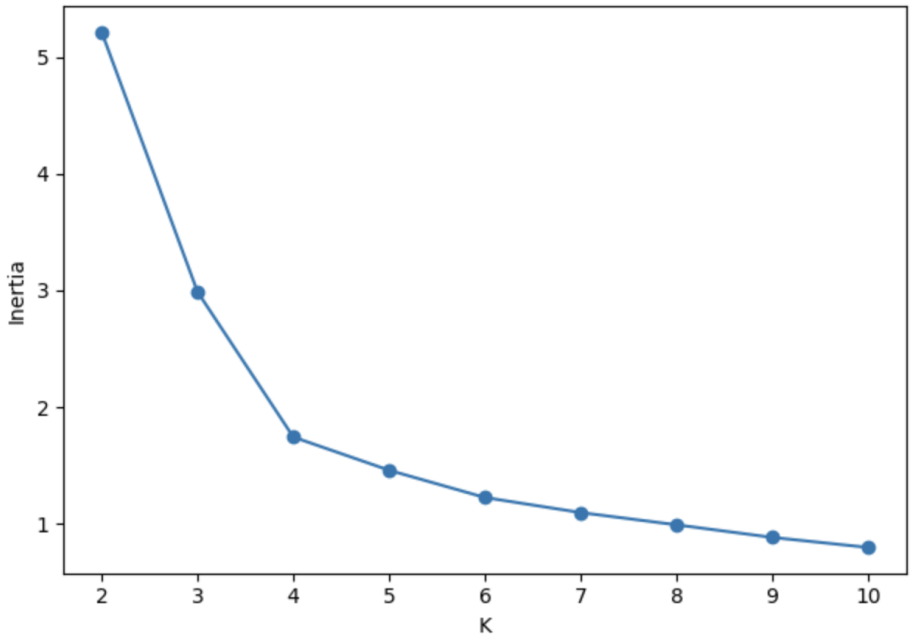

Figure 9. Elbow plot for K-Means clustering showing inertia (within-cluster sum of squares) as a function of the number of clusters K.

Figure 9 provides an "elbow" diagnostic for choosing K. Inertia decreases rapidly from $K = 2$ to $K = 4$ and then improves more gradually, suggesting that additional clusters beyond a small K offer limited reduction in within-cluster variance. In this work, the final choice of K is guided primarily by the silhouette score (which directly measures cluster separation), with the elbow plot used as supporting evidence.

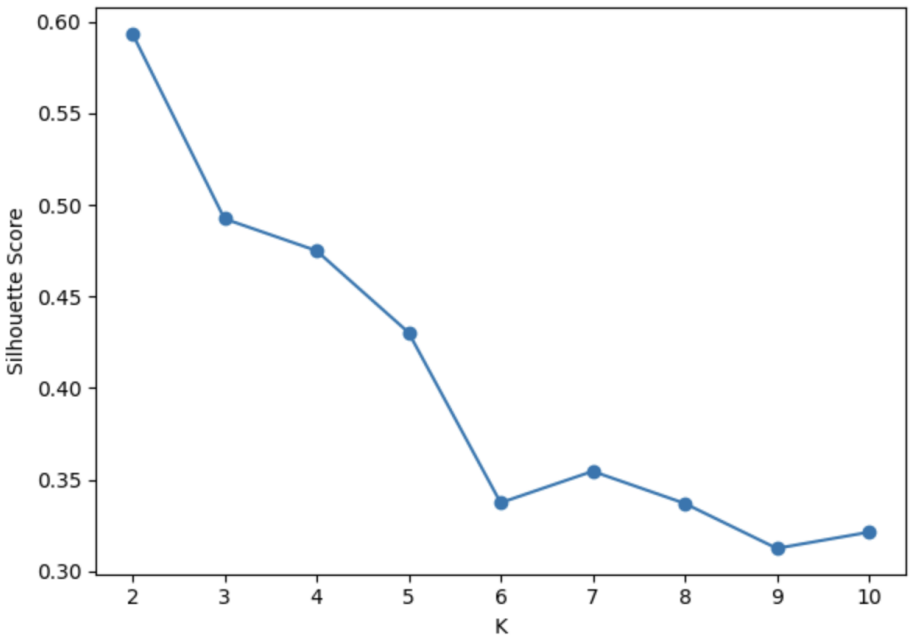

Figure 10. Silhouette score as a function of the number of clusters K for K-Means.

Figure 10 shows that the silhouette score is highest at $K = 2$, implying that a two-cluster solution provides the most distinct partition of the data among the tested values. As K increases beyond 2, the silhouette score declines, indicating that additional clusters mainly subdivide existing groups rather than uncovering genuinely well-separated new structure. Therefore, $K = 2$ is selected as the final number of clusters, supported by this separation-based diagnostic (and complemented by the elbow/inertia analysis).

Cluster separation was assessed using the silhouette score. For a point i,

$$a(i) = \frac{1}{|C_{c_i}| - 1} \sum_{\substack{j:\, c_j = c_i \\ j \neq i}} \|x_i - x_j\|, \tag{2}$$

$$b(i) = \min_{\ell \neq c_i} \frac{1}{|C_\ell|} \sum_{j:\, c_j = \ell} \|x_i - x_j\|, \tag{3}$$

and the silhouette is:

$$s(i) \;=\; \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \tag{4}$$

Higher values indicate more cohesive and better-separated clusters.

2.2.3. Preprocessing for Clustering

K-means assigns observations using Euclidean distance, computed on the preprocessed feature vector $\mathbf{z}_i$:

$$d(\mathbf{z}_i, \mathbf{z}_\ell) = \|\mathbf{z}_i - \mathbf{z}_\ell\|_2.$$

Accordingly, missing values were imputed, categorical variables were one-hot encoded, and numeric variables were standardised. This standardisation step is crucial; without it, high-variance features (e.g., `subtotal`) would dominate $\|\cdot\|_2$ and reduce the influence of operational capacity indicators.

2.2.4. Selecting the Number of Clusters

The number of clusters was selected by evaluating $k \in \{2, \ldots, 10\}$ using the silhouette score. For each observation i, the silhouette value is

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i),\, b(i)\}}, \tag{5}$$

where $a(i)$ is the mean distance from i to other points in the same cluster, and $b(i)$ is the mean distance from i to points in the nearest neighbouring cluster.

Across the tested values, the best silhouette score occurred at $k = 2$ with $s = 0.148$, and scores decreased as k increased. Although $k = 2$ is optimal under this criterion, the absolute value 0.148 is low, indicating substantial overlap between clusters and weak separation in the available feature space. This suggests that (i) creation-time features may not naturally form strongly discrete "order types," and/or (ii) one-hot encoding and mixed-type variables in a high-dimensional space reduce the distinctness of distance-based separation.

2.2.5. Cluster Results and Representative Features

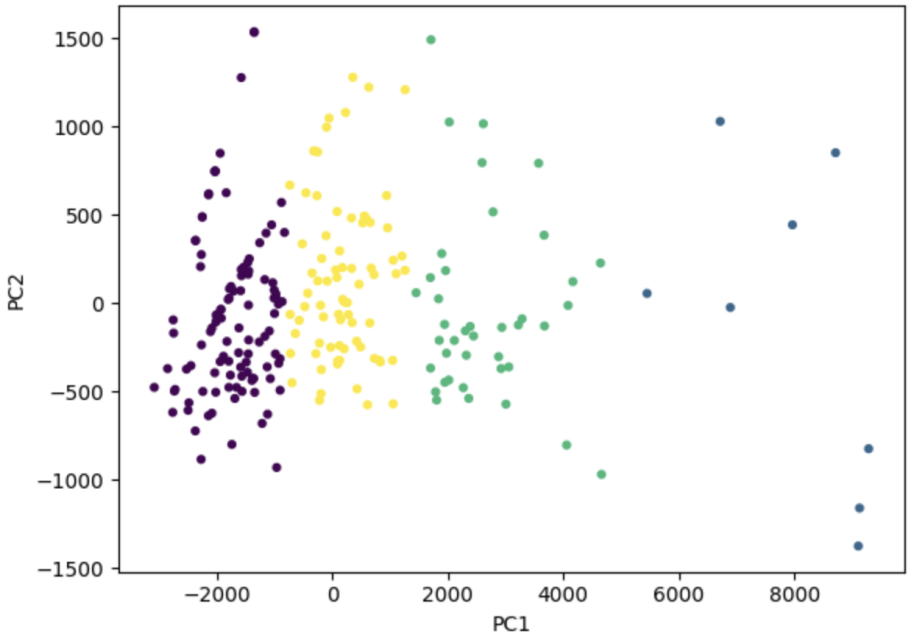

Figure 11. K-Means clusters visualised in a two-dimensional PCA projection (PC1 vs PC2).

Figure 11 provides a visual interpretation of the clustering outcome by projecting the preprocessed feature space into two principal components. Although some separation is visible between groups, overlap remains, indicating that clusters are not perfectly distinct in low-dimensional space. This is consistent with the relatively modest silhouette scores and suggests that the derived clusters should be interpreted as broad operational segments rather than sharply separated order types.

Table 4. Representative centroid-nearest orders for $k = 2$.

Attribute	**Cluster 0 (Row 780)**	**Cluster 1 (Row 617)**
Area	AREA B	AREA A
Store category	mediterranean	japanese
Order protocol	3	3
Items	4	3
Subtotal	2696	2300
Partners (onshift / busy)	46 / 38	64 / 62
Outstanding orders	46	61
Temporal context	Tue 20:28	Fri 03:30

A plausible operational distinction suggested by Table 4 is that Cluster 1 corresponds to *high workload intensity* (busy partners nearly equal to on-shift and high outstanding orders), whereas Cluster 0 reflects *moderate capacity pressure*. This interpretation is consistent with the engineered workload feature `busy_ratio`. However, given the low silhouette score, the clusters are best viewed as *soft segments* rather than sharply separated order types. Even so, such segmentation can be operationally useful: orders resembling Cluster 1 may be candidates for proactive delay messaging or priority dispatch under near-saturated courier capacity.

3. Regression

3.1. Main Model

A Random Forest Regressor was selected as the primary model because delivery time is driven by non-linear effects and interactions (e.g., time-of-day × workload, basket size × store category). Tree ensembles can capture such structure without requiring manual specification of interaction terms.

Following Breiman's original definition, random forests are "a combination of tree predictors," where each tree depends on a random vector and the ensemble generalises by aggregating many diverse trees. In the regression setting, for T trees, the forest prediction is given by the average

$$\hat{y}(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} h_t(\mathbf{x}), \tag{6}$$

where $h_t(\cdot)$ denotes the prediction of the t-th decision tree.

Each tree h_t is trained on a bootstrap sample of the training data and, at each split, considers a random subset of features. These two sources of randomness decorrelate the trees, reducing variance and mitigating overfitting relative to a single decision tree.

Why alternatives were not used as the main model, write down something

Ridge regression. Ridge is linear in the (encoded) feature space. While it provides a strong, stable baseline, it can underfit when the true relationship is piecewise, non-linear, and driven by interaction effects (e.g., workload × time-of-day). For this reason, Ridge was retained as a comparator rather than the primary model.

Dummy mean predictor. A mean predictor is necessary as a floor baseline to quantify the value added by modelling. However, it ignores all covariate information and therefore cannot leverage operational or basket features to improve delivery-time predictions.

3.2. Experiments

3.2.1. Preprocessing

Feature inclusion/exclusion (creation-time constraint). To preserve a realistic prediction setting, only variables available at order creation time were retained. The following variables were excluded to prevent target leakage:

1. `actual_delivery_time`, because it directly defines the delivery-time outcome;
2. `delivery_mins`, because it is the regression target itself;
3. complaint-related fields (`complaint`, `complaint_id`, `complaint_flag`), because they are labels and/or recorded after the outcome is realised.

Including any of these variables would violate the creation-time-only requirement and inflate reported performance by introducing *target leakage*.

Target construction. The delivery-time target `delivery_mins` was computed as the time difference (in minutes) between `created_at` and `actual_delivery_time`. To handle orders that cross midnight, a correction was applied: if `actual_delivery_time` occurs earlier than `created_at` on the clock (i.e., `actual` $<$ `created`), then 24 hours were added to the time delta. This prevents negative durations and ensures `delivery_mins` reflects the true operational interval.

Encoding and scaling. Categoricals use one-hot encoding; numerics are median-imputed and standardised. While scaling is not required for tree models, using a single consistent preprocessing pipeline ensures fair comparison with Ridge and avoids accidental preprocessing differences.

3.2.2. Feature engineering (creation-time features only)

All engineered predictors were derived using only information available at order creation time.

Temporal encodings from `created_at`. Let t denote the number of seconds since midnight for the order creation timestamp. Time-of-day was encoded cyclically as

$$\texttt{tod\_sin} = \sin\left(\frac{2\pi t}{86400}\right), \qquad \texttt{tod\_cos} = \cos\left(\frac{2\pi t}{86400}\right), \tag{7}$$

with `created_hour`, `created_minute`, and `created_sec` retained as additional granular signals. A cyclic encoding is preferable to a raw hour variable because it preserves the wrap-around structure of time (e.g., 23:59 is close to 00:00).

Similarly, letting $d \in \{0, \ldots, 6\}$ denote the day of week, day-of-week was encoded as

$$\texttt{dow\_sin} = \sin\left(\frac{2\pi d}{7}\right), \qquad \texttt{dow\_cos} = \cos\left(\frac{2\pi d}{7}\right). \tag{8}$$

Basket complexity and value. Basket-level features were constructed to capture order complexity:

$$\texttt{avg\_item\_price} = \frac{\texttt{subtotal}}{\texttt{total\_items}}, \qquad \texttt{price\_range} = \texttt{max\_item\_price} - \texttt{min\_item\_price}. \tag{9}$$

Supply–demand and congestion proxies. Operational workload and capacity pressure were summarised by ratio and difference features:

$$\texttt{busy\_ratio} = \frac{\texttt{total\_busy\_partners}}{\texttt{total\_onshift\_partners}},$$

$$\texttt{free\_partners=total\_onshift\_partners-total\_busy\_partners}, \tag{10}$$

$$\texttt{outstanding\_per\_partner} = \frac{\texttt{total\_outstanding\_orders}}{\texttt{total\_onshift\_partners}}. \tag{11}$$

These engineered features are operationally motivated: they are direct proxies for capacity, congestion, and basket complexity, which plausibly drive both delivery duration and downstream complaint risk.

3.2.3. Experimental Settings

Train/test split. The dataset was split into training and test sets using an 80/20 partition with a fixed random seed (42). Hyperparameter search and cross-validation were performed *only* on the training data to avoid optimistic bias from information leakage into the test set.

Evaluation metrics. Model performance was assessed using complementary regression metrics:

- **MAE (minutes):** the most operationally interpretable measure, representing the average absolute error in minutes.
- **RMSE (minutes):** penalises large errors more strongly, which is important when extreme late deliveries occur.
- $\boldsymbol{R^2}$: the proportion of variance explained; this is useful for comparing explanatory power, but can appear low even when MAE is operationally acceptable.

Baselines. Two baselines were included to contextualise performance:

- **DummyRegressor (mean):** predicts the global mean delivery time and provides a minimum performance floor.
- **Ridge regression:** a regularised linear benchmark, appropriate when the dominant signal is additive and approximately linear. Ridge regression provides a regularised linear baseline under multicollinearity:

$$\min_{w,b} \sum_{i=1}^{n} \left(y_i - (w^\top x_i + b)\right)^2 \; + \; \alpha \|w\|_2^2. \tag{12}$$

Hyperparameter tuning. Random Forest hyperparameters were tuned using randomised search (20 trials), exploring the following dimensions:

$$\texttt{n\_estimators}, \quad \texttt{max\_depth}, \quad \texttt{min\_samples\_leaf}, \quad \texttt{max\_features}.$$

Randomised search was preferred to an exhaustive grid because it samples diverse configurations efficiently, reducing reliance on any single arbitrary parameter choice while remaining computationally practical.

3.2.4. RESULTS

Table 5. Cross-validation performance for delivery-time regression. Lower is better for MAE/RMSE; higher is better for R^2.

Model	CV MAE (↓)	CV RMSE (↓)	CV R^2 (↑)
Random Forest	**11.40**	**15.18**	**0.147**
Ridge	11.75	18.54	-0.375
Dummy Mean	12.52	16.47	-0.004

Best hyperparameters. RandomisedSearchCV selected the following Random Forest configuration:

$$\texttt{n\_estimators} = 400, \quad \texttt{max\_depth} = 10, \quad \texttt{min\_samples\_leaf} = 10, \quad \texttt{max\_features} = 0.5.$$

Holdout test performance (final model). On the 20% holdout test set, the tuned Random Forest achieved:

$$\text{MAE} = 12.57 \text{ minutes}, \qquad \text{RMSE} = 17.05 \text{ minutes}, \qquad R^2 = 0.208.$$

Interpretation. The Random Forest improves upon the dummy mean baseline in cross-validation (MAE $12.52 \rightarrow 11.40$), indicating the presence of genuine predictive signal in the creation-time features. Ridge regression performs substantially worse in R^2 and RMSE, which is consistent with a misspecified linear functional form and limited ability to represent non-linear interactions (e.g., workload saturation effects). The test MAE is slightly higher than the cross-validated MAE, which is expected when moving from training-based estimates to evaluation on an unseen holdout split.

3.3. Critical Review

Best-performing model. The tuned Random Forest was selected as the primary regressor because it achieved the lowest MAE while maintaining an acceptable RMSE and a positive R^2, indicating meaningful predictive signal beyond the mean baseline.

Why performance is bounded. The achievable accuracy is limited by information that is not present in the dataset. Delivery time depends on major unobserved drivers—such as travel distance, traffic conditions, restaurant preparation-time variability, courier routing/batching decisions, and dynamic courier availability at fine spatial resolution—none of which are directly measured here. Consequently, a more complex model cannot recover missing information, and a modest explanatory power (e.g., $R^2 \approx 0.21$) is plausible. In addition, the strong right tail of delivery times (up to 133 minutes) inflates RMSE and makes rare extreme delays difficult to predict from creation-time features alone.

Operational significance. An MAE of approximately 12.6 minutes implies that an ETA issued at order creation is typically off by roughly one quarter of the mean delivery time. Even with this uncertainty, the model can still support operational decisions, including:

- **Ranking orders by expected lateness** to guide dispatch prioritisation;
- **Proactive customer messaging** for "high-risk delay" orders;
- **Staffing and capacity planning** by anticipating high workload conditions from supply–demand features.

Figure 12 shows why ETA predictions should be treated as uncertainty ranges: most errors fall near MAE, but rare large delays inflate RMSE and can still cause customer dissatisfaction. The MAE ($\approx$ 12.6 minutes) represents typical ETA error, while the RMSE ($\approx$ 17.0 minutes) indicates occasional larger mistakes due to long-tailed delays. Operationally, predictions should be used as risk signals and communicated as ranges rather than exact times.

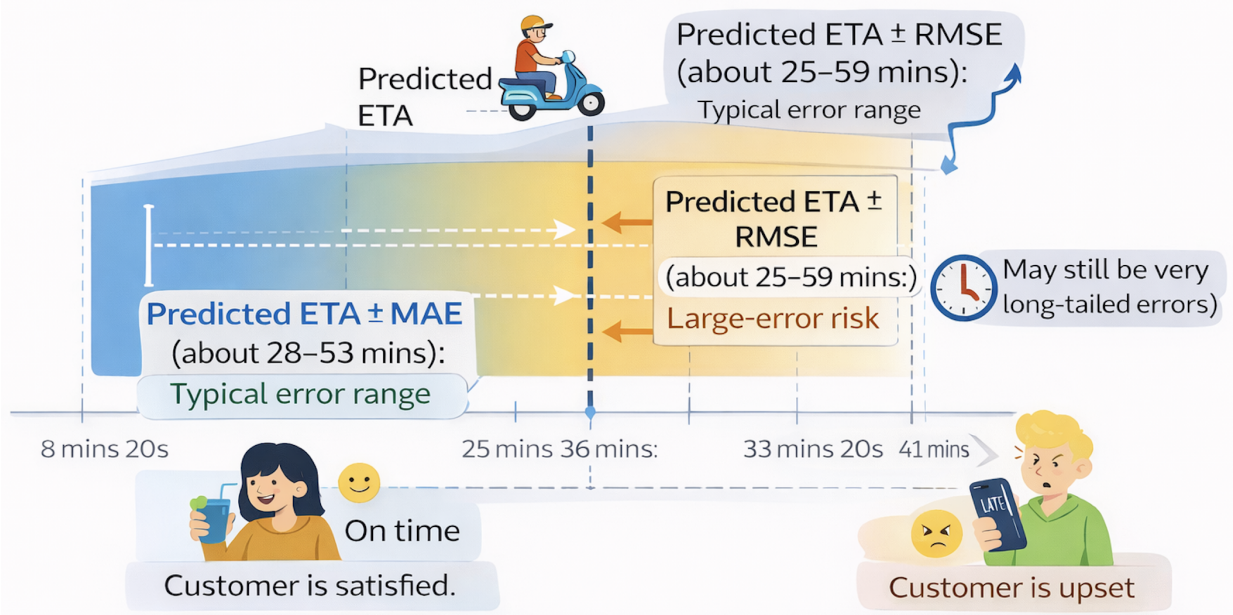

Figure 12. Real-world interpretation of delivery-time regression errors.

Future improvements. Further gains would likely require richer covariates and uncertainty-aware outputs:

- **Additional features:** distance estimates, GPS-based travel-time proxies, store-level historical mean preparation time, weather, traffic indicators, and dynamic courier density.
- **Alternative models:** gradient-boosted decision trees often outperform Random Forests on tabular datasets and are a natural next step if permitted.
- **Prediction intervals:** quantile regression or calibrated intervals (e.g., 50th/90th percentile ETA) may be more actionable than a single point estimate in operational settings.

4. Classification

4.1. Main Model

Complaint prediction is imbalanced, with a positive-class prevalence of 24.4%. Under such imbalance, standard classifiers can over-emphasise the majority class and achieve superficially high accuracy by defaulting to *no-complaint* predictions. To address this, the main model uses a `BalancedRandomForestClassifier`, which modifies Random Forest training through internal resampling.

According to the `imbalanced-learn` documentation, a balanced random forest differs from a standard random forest in that it "draws a bootstrap sample from the minority class and … the same number … from the majority class." This repeated balancing encourages the ensemble to learn minority-class structure rather than being dominated by majority-class frequency, making it preferable to fitting a plain Random Forest "as-is".

The binary target is,

$$y = \texttt{complaint\_flag} = \begin{cases} 1, & \text{complaint recorded,} \\ 0, & \text{otherwise.} \end{cases}$$

Why not logistic regression as the main model? Logistic regression provides a strong and interpretable baseline, but it assumes a linear decision boundary in the encoded feature space. Complaint outcomes are plausibly driven by non-linear feature combinations (e.g., late-night orders under high `busy_ratio` for certain store categories), for which a tree ensemble offers greater flexibility. Logistic regression was therefore retained as a benchmark rather than the primary model.

4.2. Experiments

4.2.1. Preprocessing

The feature set for complaint prediction matches the regression task: only creation-time variables were included and all leakage-prone columns were removed. The same engineered predictors were retained—including `busy_ratio`, `outstanding_per_partner`, cyclical time encodings, and basket features—because they plausibly influence both delivery lateness and downstream dissatisfaction.

Imbalance handling. Class imbalance was addressed in a model-specific manner:

- **Logistic regression:** `class_weight="balanced"`.
- **RandomForestClassifier:** `class_weight="balanced"`.
- **Balanced Random Forest:** internal balanced bootstrap sampling, drawing equal-sized samples from the minority and majority classes for each tree.

Precision, recall and F_1 score are:

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN}, \tag{13}$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{14}$$

Because complaints are the minority class, PR-AUC (average precision) and ROC-AUC are reported as threshold-independent ranking metrics.

4.2.2. Experimental Settings

4.2.3. Split, cross-validation, and evaluation protocol (complaints)

Data splitting and cross-validation. A stratified 80/20 train/test split was used to preserve the complaint prevalence in both partitions. Model selection was performed using stratified 5-fold cross-validation on the training set only; the test set was reserved for a single final evaluation to avoid optimistic bias.

Metrics for imbalanced classification. Because complaints are the minority class, evaluation focused on imbalance-appropriate, ranking-based metrics:

ROC-AUC Threshold-free measure of ranking quality across all possible classification thresholds.

PR-AUC (Average Precision) Summarises the precision–recall curve and is often more informative when positives are rare, since it directly reflects performance on the minority class.

Prior work emphasises that precision–recall curves can be more revealing than ROC curves under strong class imbalance davis$_g oadrich_2$006, $saito_r ehmsmeier_2$015.

Baselines. To contextualise results, three baselines were included:

- **DummyMostFrequent:** predicts all non-complaints (sanity-check lower bound).
- **LogisticRegression (balanced):** linear benchmark in the encoded feature space.
- **RandomForestClassifier (`class_weight="balanced"`):** non-linear baseline without internal resampling.

4.2.4. RESULTS

Table 6. Classification cross-validation results (training set only). Higher is better for all metrics.

Model	CV ROC-AUC (↑)	CV PR-AUC (↑)	CV F1 (↑)
RandomForestBalanced	0.614	0.387	0.193
BalancedRandomForest	0.617	0.383	0.384
LogRegBalanced	0.622	0.361	0.406
DummyMostFreq	0.500	0.244	0.000

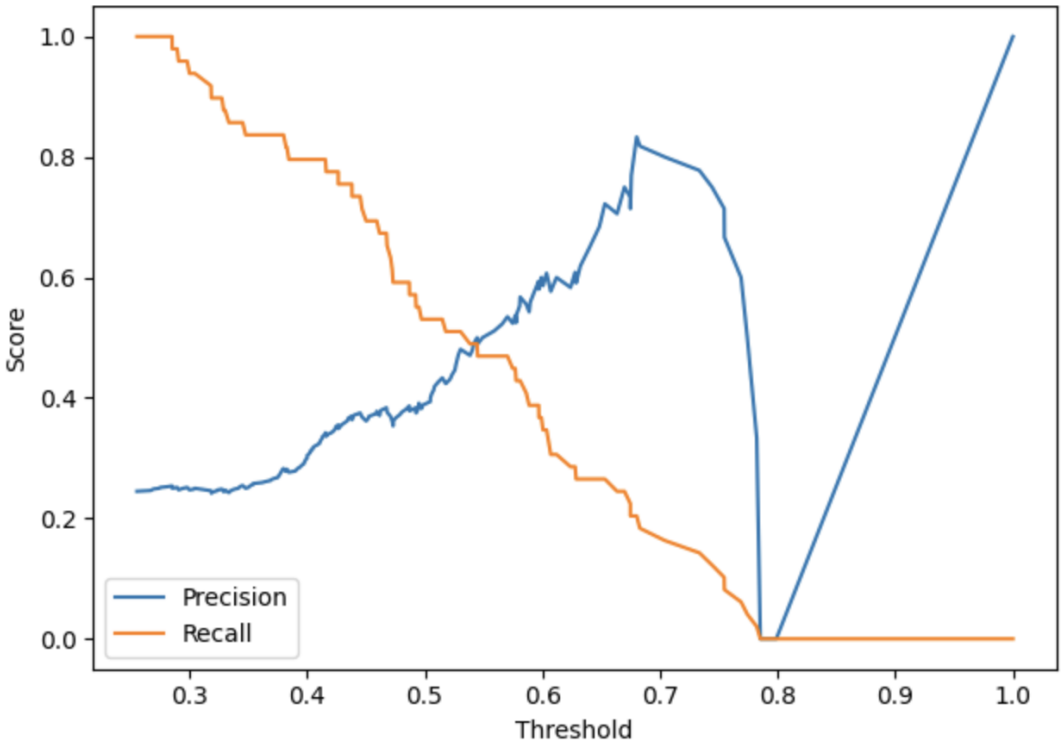

Figure 13. Precision and recall as functions of the classification threshold for complaint prediction.

Figure 13 supports threshold selection based on operational costs. If proactive interventions are inexpensive, a lower threshold may be preferred to increase recall; if interventions are costly, a higher threshold may be chosen to prioritise precision and reduce false positives.

Table 7. Best hyperparameters for the Balanced Random Forest (RandomisedSearchCV).

Hyperparameter	Selected value
`n_estimators`	400
`max_depth`	10
`min_samples_leaf`	10
`max_features`	`sqrt`

Table 8. Holdout test performance for the tuned Balanced Random Forest (threshold = 0.5).

Metric	Value
ROC-AUC	0.692
PR-AUC	0.482
F1-score	0.452
Precision	0.394
Recall	0.531

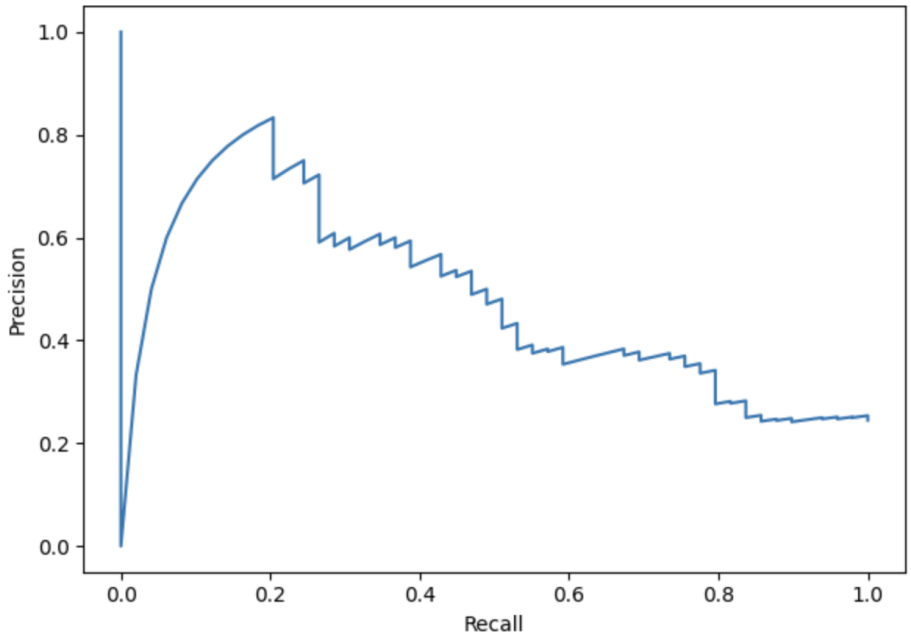

Figure 14. Precision–Recall (PR) curve for the complaint prediction model on the hold-out test set.

Figure 14 summarises the trade-off between precision and recall across thresholds. Since complaints are the minority class, PR performance is more informative than accuracy: a model can achieve high accuracy by predicting the majority class while still failing to detect complaints.

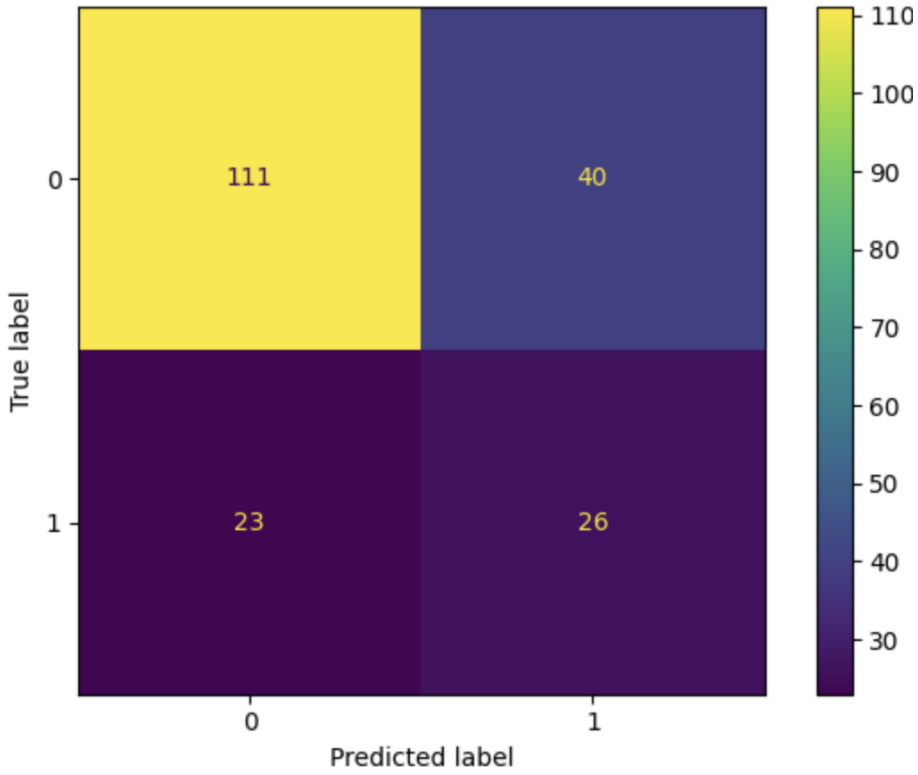

Figure 15. Confusion matrix for complaint prediction on the hold-out test set using a decision threshold of 0.5.

Using the default threshold of 0.5 (Figure 15), the model achieves TN=111, FP=40, FN=23, and TP=26. This corresponds to a precision of approximately 0.394 and recall of approximately 0.531, indicating that the model identifies around half of true complaints while producing some false alarms.

Table 9. Confusion matrix for the tuned Balanced Random Forest on the holdout test set (threshold = 0.5).

	Predicted 0	**Predicted 1**
Actual 0	TN = 111	FP = 40
Actual 1	FN = 23	TP = 26

This corresponds to a precision of,

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{26}{26 + 40} \approx 0.394$$

and a recall of,

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{26}{26 + 23} \approx 0.531,$$

indicating that just over half of true complaints are detected at this threshold, with a moderate number of false positives.

4.3. Critical Review

Figure 16 illustrates the practical impact of the confusion matrix results: higher recall allows more complaints to be proactively addressed, but low precision increases the workload due to false alarms.

Preferred model. The tuned Balanced Random Forest (BRF) was selected as the operational classifier because it achieves a strong PR-AUC (0.482) and a recall of 0.531, indicating meaningful minority-class detection without collapsing into majority-class predictions.

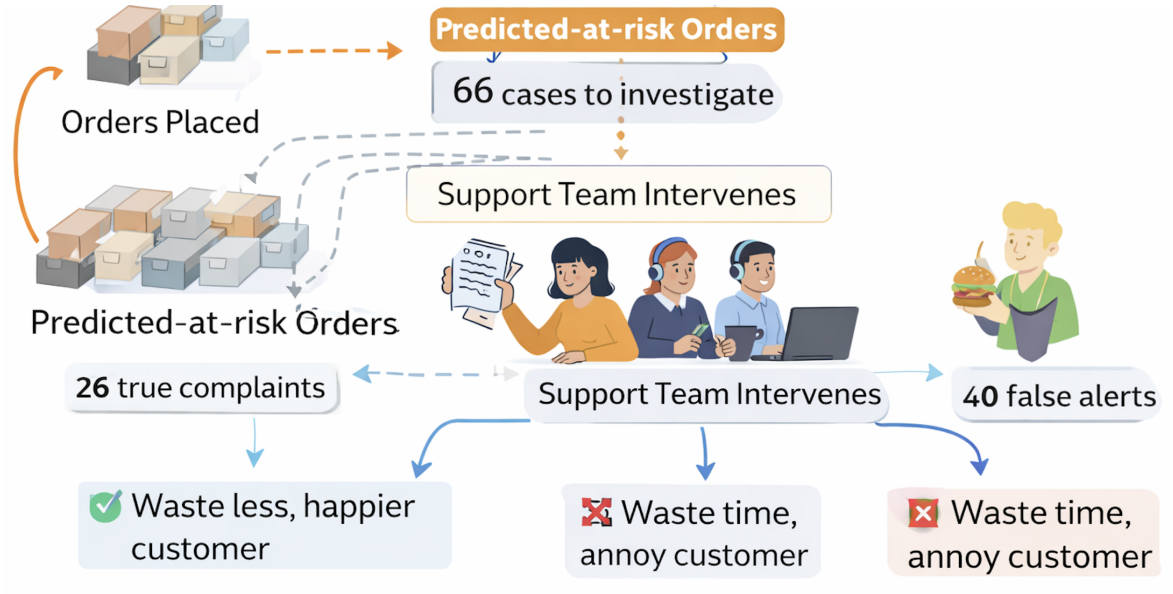

Figure 16. Operational interpretation of low precision and higher recall for complaint prediction.

Why cross-validation and holdout results may differ. Table 10 shows that CV PR-AUC values ($\approx$ 0.38–0.41) are lower than the holdout PR-AUC (0.482). With a moderate sample size (n = 1,000), this gap can plausibly arise from sampling variation: a particular holdout split may be "easier" if complaint patterns align more clearly. The appropriate interpretation is therefore not that the model is perfect, but that it exhibits consistent above-baseline signal with non-trivial uncertainty.

Table 10. PR-AUC comparison between cross-validation (training only) and holdout test performance.

Evaluation setting	**PR-AUC**
Cross-validation (typical range)	0.38–0.41
Holdout test (final BRF)	0.482

Cost-sensitive reading of the confusion matrix. The holdout confusion matrix implies a trade-off between recall and precision. **Cost-sensitive interpretation (threshold = 0.5).** The confusion matrix implies a clear precision–recall trade-off. With **recall** = 0.531, the model identifies just over half of true complaints, which is desirable when missed complaints (false negatives) are costly. However, **precision** = 0.394 indicates that many flagged orders are false alarms (40 FP versus 26 TP), so interventions may be applied unnecessarily.

Operational decision rule. If the intervention is *cheap* (e.g., proactive messaging or a small voucher), it is reasonable to *lower* the classification threshold to prioritise recall. If the intervention is *expensive* (e.g., a premium delivery upgrade), it is preferable to *raise* the threshold to prioritise precision. This is why threshold curves (precision/recall versus threshold) are operationally important: selecting the "best" threshold is a business decision rather than a purely technical one.

This motivates examining threshold curves (precision/recall vs. threshold): the "best" operating point is a business decision, not purely a technical one.

Future improvements.

- **Probability calibration:** apply Platt scaling or isotonic regression to improve the reliability of predicted probabilities for decision-making.
- **Delay-related signals:** include a *predicted* `delivery_mins` feature (generated using creation-time information only) as an input to complaint prediction, since lateness is a common complaint driver.

- **Prior-risk features:** add store-level historical complaint rates and time-of-day complaint patterns (if permitted) to capture systematic baseline risk differences.

5. Conclusion

This coursework analysed a food delivery order dataset and addressed the three objectives required by the specification: (i) exploratory data analysis (EDA) and clustering, (ii) delivery-time prediction, and (iii) complaint prediction, using only creation-time information throughout.

EDA indicated that delivery times are moderately variable with a heavy right tail, reflecting rare but severe delays. Complaint prevalence was 24.4%, implying an imbalanced classification setting. K-means clustering selected $k = 2$ under the silhouette criterion; however, the low silhouette score suggests weak cluster separation, so the clusters are best interpreted as broad segments rather than strict order "types".

For delivery-time regression, a tuned Random Forest achieved a holdout test MAE of 12.57 minutes with $R^2 = 0.208$, outperforming linear and dummy baselines but remaining constrained by unobserved operational drivers (e.g., distance, traffic, and preparation-time variability). For complaint prediction, a tuned Balanced Random Forest achieved ROC-AUC $= 0.692$ and PR-AUC $= 0.482$, substantially above baseline and suitable for risk ranking. Nevertheless, the precision–recall trade-off implies that the operating threshold should be selected in a cost-sensitive manner based on the business consequences of false positives versus false negatives.

Overall, the analysis demonstrates a leakage-safe modelling pipeline with feature engineering grounded in operational mechanisms and evaluation metrics aligned with decision-making needs. The most impactful future improvements would likely come from incorporating missing operational covariates (distance estimates, traffic indicators, and preparation-time proxies) and adopting cost-sensitive threshold selection (and/or calibrated probabilities) for complaint interventions.