Feature Selection Process

The dataset saved in the json file contained a total of 85 columns after most primary keys with nested json objects were expanded. From the 85 features extracted, their data types consisted of: 64 object features (strings, lists, etc.), 4 boolean features, 11 float features, and 6 integer features. The feature selection process followed a systematic discarding procedure that began with analyzing null values.

Null Value Analysis:

From a simple null count, 8 features were immediately eliminated because no non-null values were present (100% null). These features included: "shipping_methods", "coverage_areas", "listing_source", "differential_pricing", "pictures_quality", "subtitle", "shipping_free_methods_rule_value".

Next, features with null values between 60% and 99% were analyzed. First, any column related to an ID was dropped as these contain random alphanumeric identifiers that have no relationship to product condition (11 features dropped). For the remaining features in this interval, the following criteria were applied:

- ➢ Features with >95% null values were dropped regardless of distribution, as insufficient data existed for reliable patterns
- ➢ Features with 85-95% null values were only retained if they showed > 10 percentage point deviation from the baseline class distribution (53% new, 47% used) when non-null
- ➢ Features with 60-85% null values were evaluated based on both missingness correlation with target class and distribution of non-null values

Features meeting retention criteria were converted to boolean indicators (has_feature) to capture the presence/absence signal. Features showing no meaningful skew (distributions matching the 50/50 split baseline) were dropped.

High Correlation Filter

A high correlation filter was applied to numeric fields to remove redundant features. Two features that are highly correlated provide nearly identical information to the model, and retaining both can lead to issues without improving predictive power.

The Pearson Correlation coefficient was calculated between all pairs of numeric fields. Features with correlations above 0.8 were removed, keeping the feature most directly related to the business problem.

In this analysis, two fields were dropped:

- ➢ "price" was removed in favor of "base_price"
- ➢ "available_quantity" was removed in favor of "initial_quantity"

Numeric Univariate Analysis

For the remaining numeric fields, a univariate analysis was conducted to examine each feature's distribution and measures of central tendency. During this process, two timestamp fields ("start_time" and "stop_time") were converted to four new features: "listing_duration_days", "days_since_listed", "listing_month", and "listing_day_of_week". These derived features provide more interpretable

information than raw timestamps for predictive modeling. Both “listing_month” and “listing_day_of_week” were categorical features created to give a sense of the time of the year and the week the product was listed in.

Upon further analysis, "listing_duration_days" was converted to a boolean field ("is_60day_listing") as 95% of listings had a duration of 60 days, making this the dominant category worth capturing as a binary indicator.

The "base_price" feature contained extreme outliers on both ends of the distribution. The 99th percentile value was 130,000 while the maximum reached 2,222,222,222, likely showing a data quality issue. A threshold of 130,000 was applied to remove rows with outlier prices, affecting less than 1% of the dataset. For the lower bound, the 1st percentile was 16.99 while the minimum value was 0.87. Products with base_price values below 16.99 were removed as unrealistically low prices that likely represented data entry errors too. Distribution plots helped in identifying the outliers for these types of features.

Categorical Univariate Analysis

Categorical features underwent a similar examination of their distributions and value counts. Features with zero variance, where 100% of values were identical, were removed as they provide no discriminative power for classification. This included "seller_address_country_name" and "international_delivery_mode", both of which contained only a single unique value across the entire dataset.

No additional encoding was applied to categorical features, as CatBoost was selected as the modeling algorithm due to its native ability to handle categorical variables without manual encoding.

Final Selection

After completing the feature selection process, 29 features were retained for model training and evaluation. The final feature set comprised 21 native features from the original dataset and 8 engineered features created during the preprocessing phase. The breakdown is as follows:

Feature name	Native/Generated	Datatype
seller_address_state_name	Native	object
seller_address_city_name	Native	object
warranty	Native	object
base_price	Native	float
shipping_local_pick_up	Native	object
shipping_free_shipping	Native	object
shipping_mode	Native	object
buying_mode	Native	object
tags	Native	object
last_updated	Native	object
accepts_mercadopago	Native	object
title	Native	object
automatic_relist	Native	object
date_created	Native	object
status	Native	object
initial_quantity	Native	float

sold_quantity	Native	float
shipping_dimensions_indicator	Native	object
variations_seller_custom_field_indicator	Native	object
original_price_indicator	Native	object
has_non_mercado_pago_methods	Generated	object
has_pictures	Generated	object
has_warranty	Generated	object
days_since_listed	Generated	float
listing_month	Generated	object
listing_day_of_week	Generated	object
has_variations_combinations	Generated	object
is_60day_listing	Generated	object
has_sold_quantity	Generated	object

This represents a reduction from 85 original columns to 29 features, improving model interpretability and computational efficiency while retaining the most predictive signals for distinguishing between new and used items. The "has_sold_quantity" indicator captures whether any units have been sold (sold_quantity > 0), providing a binary signal of sales activity.

Model Training and Evaluation

The final feature set was used to train a CatBoost classifier, leveraging its ability to handle categorical features natively and its strong performance on tabular data. The model achieved the following performance metrics:

Metric	Validation	Test
Accuracy	0.86	0.86
Precision	0.81	0.81
Recall	0.90	0.89
AUC	0.86	0.86

Performance Analysis and Secondary Metric Justification

While the model achieved an accuracy of 0.86, meeting the minimum threshold requirement, accuracy alone does not provide a complete picture of model performance. In binary classification, it is essential to examine both Precision and Recall to understand the types of errors the model makes.

Precision measures the proportion of predicted used items that are actually used. A precision of 0.81 means that when the model predicts an item is used, it is correct 81% of the time. This metric penalizes false positives or instances where the model incorrectly labels a new item as used.

Recall, selected as the secondary evaluation metric, measures the proportion of actual used items that the model correctly identifies. A recall of 0.89-0.90 indicates the model successfully detects approximately 89% of all used items in the dataset. This metric penalizes false negatives or instances where the model incorrectly labels a used item as new.

Importance of High Recall in This Context

For this e-commerce classification problem, Recall is particularly critical for several reasons:

1. Misclassifying a used item as new (false negative) has more severe consequences than the reverse. Customers who receive a used item when expecting new may file complaints, request refunds, or lose trust in Mercadolibre's marketplace.
2. False negatives can lead to poor customer experiences and potential regulatory issues if used items are systematically mislabeled as new, damaging the platform's credibility.
3. With a relatively balanced dataset (53% new, 47% used), recall provides meaningful insight into the model's ability to identify the used class without being skewed by extreme class imbalance.

The model's high recall of 0.89-0.90 demonstrates strong performance in identifying used items, effectively minimizing the risk of mislabeling them as new. The consistency between validation (0.90) and test (0.89) recall scores, indicates the model generalizes well to unseen data without overfitting. Combined with the AUC score of 0.86, the model shows robust discriminative ability across different decision thresholds.