Early Momentum Forecasting

Team B09:
Aryan Jain, Ahrar Karim, Drishti Chulani,
Linh Le, Sai Leela Rahul Pujari

01 Executive Summary

Business Problem

- E-commerce wins are driven by early identification of high-potential products.
- Predicting top performers within the first 7–14 days enables smarter promotions, inventory decisions, and seller support before demand peaks.
- Our model uses early sales patterns to predict top 10% products using 60-day sales

Motivation

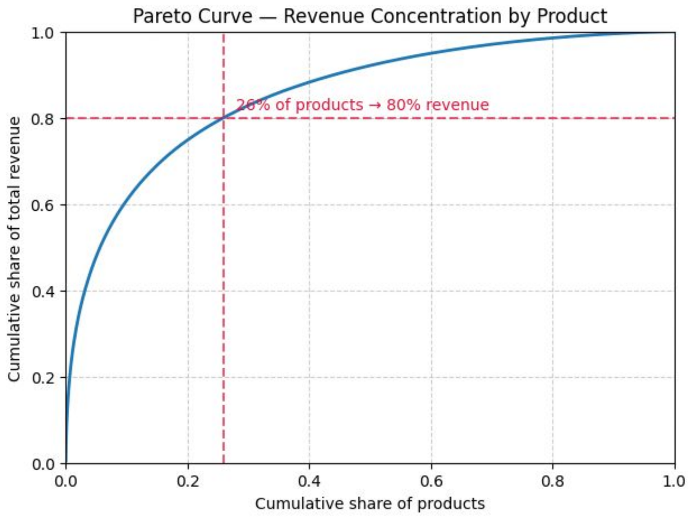
- Identification of early momentum patterns and segments products by trajectory (trending vs late-bloomer).
- Helps optimize marketing spend, inventory allocation, and product lifecycle strategy.
- Supports marketing, inventory, and marketplace teams with smarter promotions, fewer stockouts, and better seller guidance.

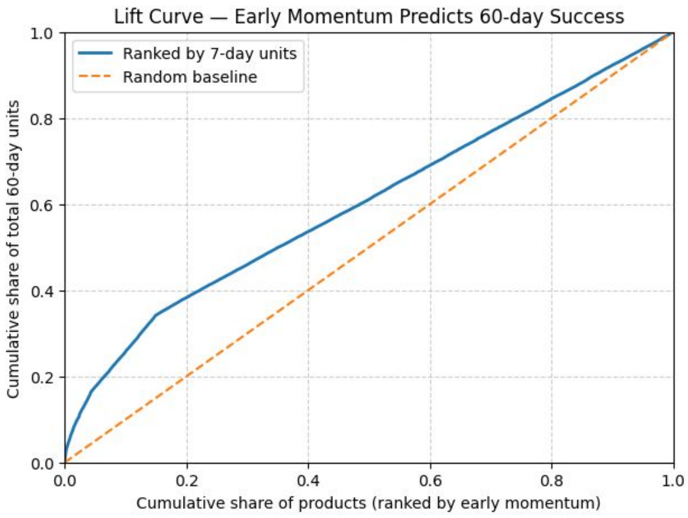
Datasource

- Real transaction data from Olist, Brazil's largest marketplace, spanning Oct 2016 – Sep 2018.
- Includes product listings, orders, payments, shipping details, and customer reviews.
- Represents activity from small businesses selling across multiple Brazilian marketplaces.
- Data Size ~ 113,000 records
  ~ 50+ columns

02

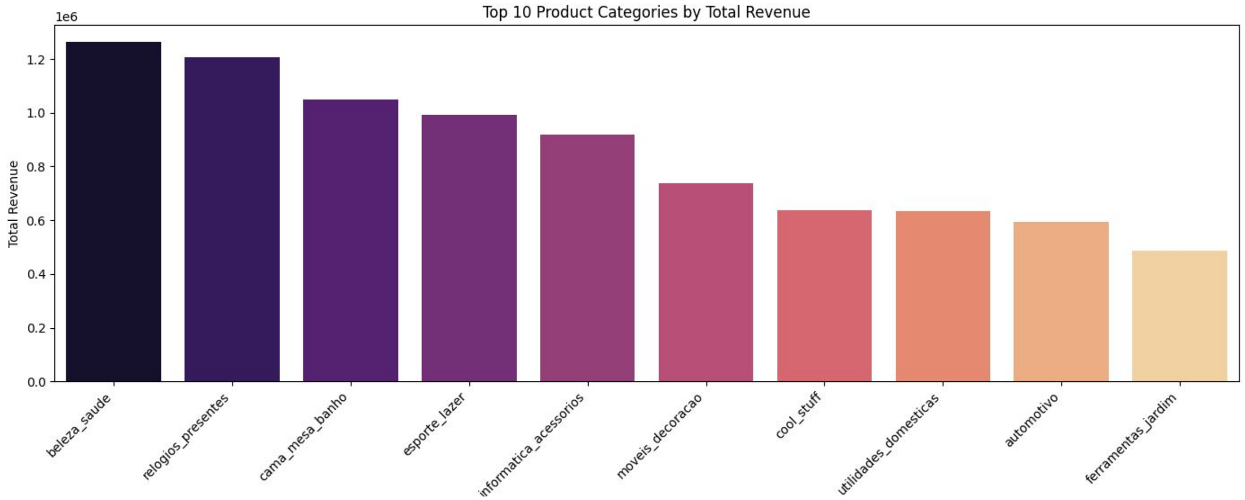
Exploratory Data Analysis

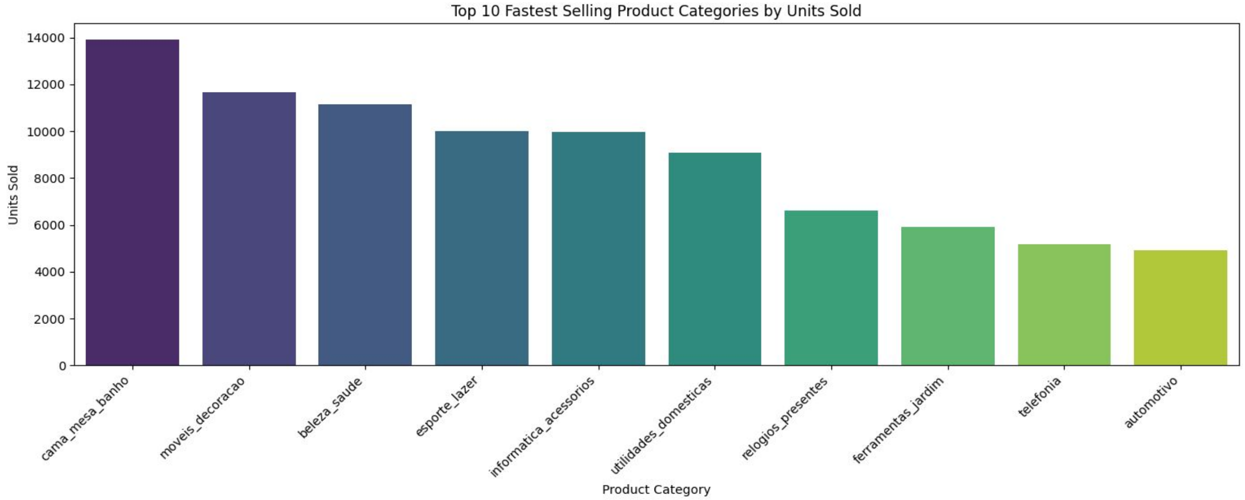




- E-commerce follows a Classic Pareto pattern (80/20).
- Only 20% of products generate ~80% of total revenue, meaning revenue is highly concentrated among a small set of winners.
- Most products contribute very little, while a few high performers drive the business.

- Demonstrates how well early sales (first 7 days) predict long-term success (60-day units).
- The blue line (ranked by early momentum) is well above the orange random baseline.
- This means products that sell well in the first week disproportionately make up later sales.





Top 10 Product Categories by Total Revenue

- Demonstrates which product categories drives the most revenue on the platform.
- Categories like beleza_saude and relogios_presentes dominate revenue, suggesting higher price points or premium demand.
- Lower-revenue categories still sell but may rely more on volume or lower margins.

Top 10 Fastest-Selling Categories by Units Sold

- Demonstrates which categories sell the largest number of units, regardless of price.
- Categories like cama_mesa_banho and moveis_decoracao lead in volume, meaning they have high demand and steady turnover.
- These may generate lower revenue per item but depend on mass purchasing behavior.

03

Modeling

Modeling

Data Preparation

- Merging of the 7 datasets
- Filtered the new products with full 60 day windows
- Made key features based on the existing variables

Target Definition

- Defined Top 10% products by 60-day units as "successful".
- Introduced momentum segmentation
    - Trending Front Loaded (early spike -> flatten)
    - Late Bloomers (slow start -> accelerates)

Modeling

- Trained multiple ML models:
    - Logistic Regression
    - Random Forest
    - Gradient Boosting
- Tuned hyperparameters for best precision on early winners.

Train/Test Strategy

- 80/20 split
- The positive class ~15%

Target Label Creation

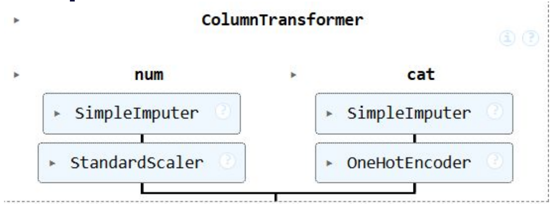
- Calculates Units Sold per Product & Time Window
  - units60: Total units sold in days [0, 60]
- Defines Bestseller Threshold
  - Compute the 90th percentile of units60 distribution across all products
- Creates Binary Label
  - Assign Best_Seller_60 = 1 if units60 ≥ threshold_60, else 0
- Computes Early Momentum Features
  - Calculate early_momentum_7 = units7 / units60 (ratio of early sales to total)
- Example: units7 (Days 1-7) = 292 units, units14 (Days 1-14) = 336 units, units60 (Days 1-60) = 351 units (Front Loader behaviour)
  The top 10% of products sell 56+ units in their first 60 days (threshold_60= 56)
  If units60 ≥ threshold then Best_Seller_60= 1 else 0
  351 ≥ 56, therefore Best_Seller_60= 1
  Early_momentum_7 ratio = 292/351 = 0.832 which is higher than 0.75, therefore classified as 'Trending'

Overview of Modeling

1. Features

- **Numeric** features used: ['units7', 'units14', 'price', 'freight_value', 'delivery_time_days', 'freight_ratio']
- **Categorical** features used: ['product_category_name', 'seller_id', 'order_status']
- 9 features and 1 target variable

2. Pre Processing Pipeline



3. Baseline Models

The 5 models:

- Logistic Regression
- Random Forest
- Decision Trees
- Gradient Boosting
- SVM

4. Param Grids/ CV

5 split StratifiedKFold

scoring="balanced_accuracy"

5. Test Set Validation

04
Results

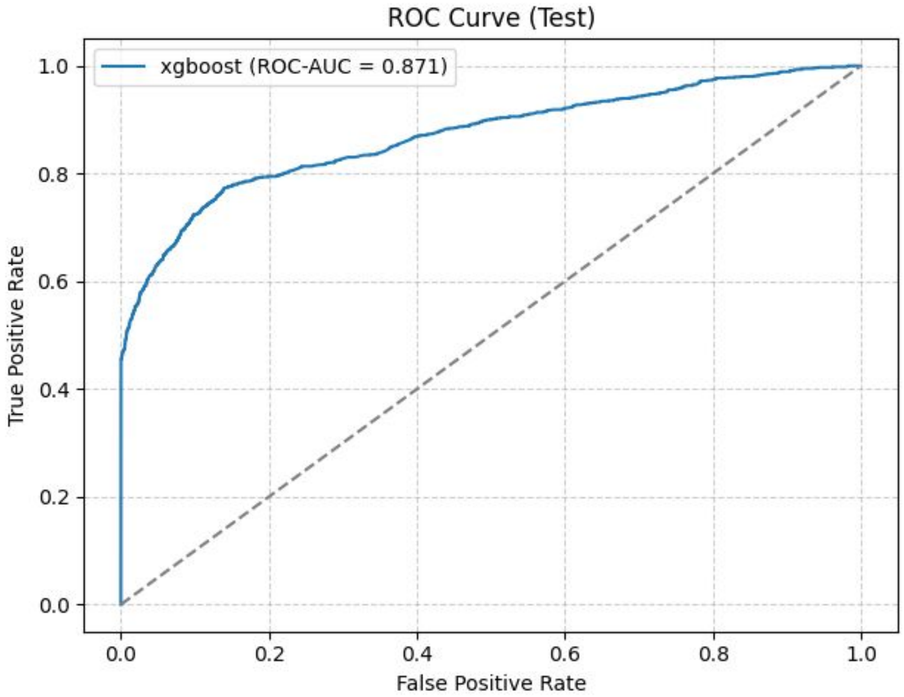
Train vs Test Performance

	model	best_params	cv_bal_acc_mean	cv_bal_acc_std	cv_roc_auc_mean	cv_roc_auc_std
0	log_reg	{'model__C': 0.1, 'model__solver': 'lbfgs'}	0.796179	0.007221	0.861310	0.008650
4	xgboost	{'model__colsample_bytree': 0.8, 'model__learn...	0.796133	0.006797	0.856315	0.007938
2	random_forest	{'model__max_depth': None, 'model__max_feature...	0.796118	0.007040	0.843504	0.009820
3	svm_rbf	{'model__C': 10.0, 'model__gamma': 'auto'}	0.795925	0.007462	0.852078	0.009188
1	decision_tree	{'model__max_depth': 5, 'model__min_samples_le...	0.795198	0.006547	0.833057	0.005883

	model	test_roc_auc	test_pr_auc	test_accuracy	test_balanced_accuracy	test_f1	test_precision	test_recall	tn	fp	fn	tp
4	xgboost	0.870566	0.750024	0.870885	0.810787	0.631761	0.560246	0.724206	5010	573	278	730
0	log_reg	0.873759	0.749943	0.867091	0.808954	0.625321	0.549624	0.725198	4984	599	277	731
3	svm_rbf	0.862131	0.739731	0.915946	0.725198	0.621067	1.000000	0.450397	5583	0	554	454
1	decision_tree	0.847028	0.685271	0.866181	0.803946	0.620155	0.547945	0.714286	4989	594	288	720
2	random_forest	0.855657	0.646987	0.863147	0.810690	0.621644	0.538517	0.735119	4948	635	267	741

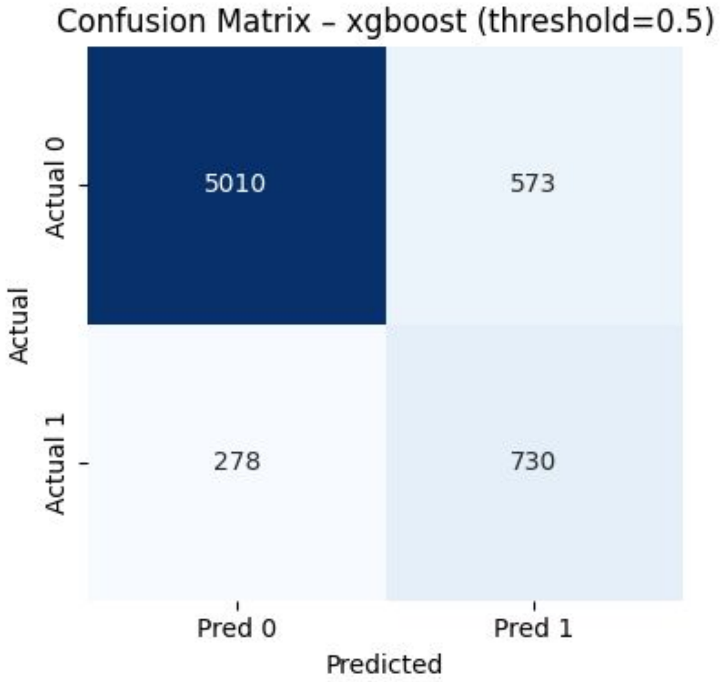
- Train and test Balanced Accuracy were similar → no overfitting
- Confirms the model generalizes well to unseen product launches
- Time based split also validates real world deployment reliability

ROC



- The ROC curve demonstrates strong separation ability, confirming that early sales indicators are predictive of 60-day success.
- **Balanced Accuracy** was used due to class imbalance, and it aligns with the strong AUC, confirming the model reliably identifies both winners and non-winners

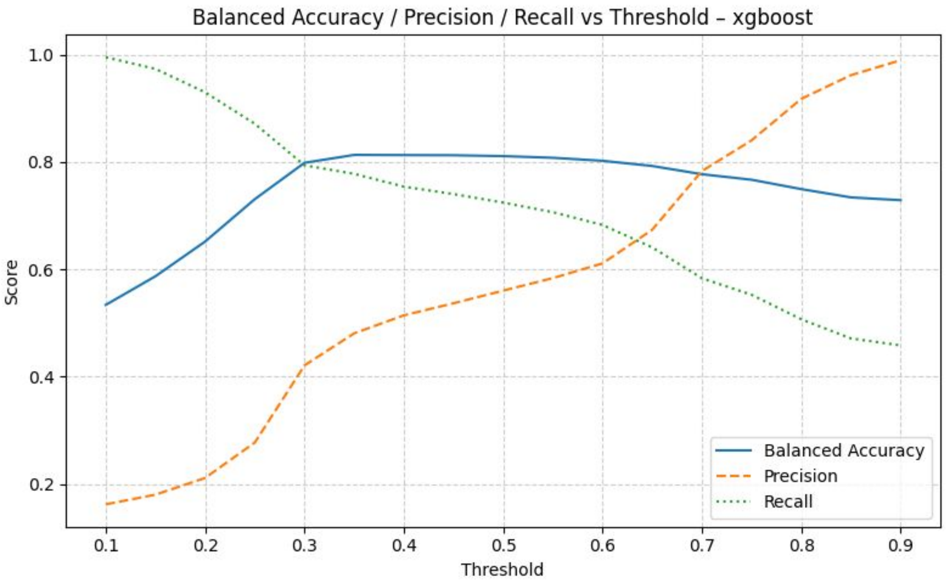
Confusion Matrix



- True Positives → correctly predicted future best-sellers
- True Negatives → avoids wasting marketing on weak products
- False Positives → often "front-loaded" items that spike early but plateau
- False Negatives → "late bloomers" caught by momentum segmentation

$730/(5010+573+278+730)=0.111$

Precision Recall Graph



	threshold	precision	recall	accuracy	balanced_accuracy	tp	fp	fn	tn
0	0.10	0.162245	0.995040	0.213473	0.533701	1003	5179	5	404
1	0.15	0.180231	0.973214	0.318920	0.587001	981	4462	27	1121
2	0.20	0.211369	0.929563	0.458807	0.651688	937	3496	71	2087
3	0.25	0.277322	0.871032	0.633136	0.730608	878	2288	130	3295
4	0.30	0.421053	0.793651	0.801548	0.798312	800	1100	208	4483
5	0.35	0.480687	0.777778	0.837506	0.813034	784	847	224	4736
6	0.40	0.514208	0.753968	0.853437	0.812682	760	718	248	4865
7	0.45	0.536691	0.740079	0.862540	0.812365	746	644	262	4939
8	0.50	0.560246	0.724206	0.870885	0.810787	730	573	278	5010
9	0.55	0.583607	0.706349	0.878015	0.807679	712	508	296	5075
10	0.60	0.611012	0.682540	0.884995	0.802044	688	438	320	5145
11	0.65	0.673618	0.640873	0.897588	0.792405	646	313	362	5270
12	0.70	0.782956	0.583333	0.911546	0.777069	588	163	420	5420
13	0.75	0.840121	0.552579	0.915491	0.766797	557	106	451	5477
14	0.80	0.917415	0.506944	0.917615	0.749353	511	46	497	5537
15	0.85	0.961538	0.471230	0.916249	0.733913	475	19	533	5564
16	0.90	0.989293	0.458333	0.916401	0.728719	462	5	546	5578

segmented_predictions.csv

product_id	pred_prob	actual	units7	units60	momentum_ratio	segment
86271c025e6ff0c1d327388c0b4c811b	0.152994812	0	1	1	1	TRENDING
253aede415ef331d1262ffc3a411224d	0.215381192	0	1	1	1	TRENDING
88a82488ded06b62a95df35c384cabfb	0.218177366	0	1	2	0.5	LATE_BLOOMER
0d954479e7991c06d35202c130844b57	0.231730186	0	1	1	1	TRENDING
8abc2d73b55855c07b6888d4b21b3da6	0.345042015	0	1	2	0.5	LATE_BLOOMER
b07fffe072c9adc235a35d8da7c0584d	0.130482312	0	1	1	1	TRENDING
4fb3a6cb6e0aa78466566ab0ec0666c6	0.253051197	0	1	1	1	TRENDING
68e3ddebebde61d68a6a35c3734bfb0f	0.435486324	0	1	1	1	TRENDING
3360da0bdc5e96e78beaade20beeefa4	0.209195098	1	1	6	0.166666667	LATE_BLOOMER
4d8ea5149cb1949048b389bc23797fef	0.274720538	0	1	1	1	TRENDING
d5280433d80f1eadab87e60292691602	0.176983836	0	1	1	1	TRENDING
0b9eab47f340cb0354b04f84b95940f9	0.162785599	0	1	1	1	TRENDING
265928225c1358e74bf8668ff65096f3	0.219452217	0	1	1	1	TRENDING
ce6450b4e1fbb3bc232eeb8b8e1c5757	0.363210397	0	1	1	1	TRENDING
e5cac955339b48ea3b9773f034623e29	0.152040925	0	1	1	1	TRENDING
216bb0e0cd43ffd832e0973d35e0377e	1	1	1	37	0.027027027	LATE_BLOOMER
4c68fa8fa43e3ffddf31ef176866f762	0.312721928	0	1	1	1	TRENDING
5e7b701349598f3728a3eb624c6570dc	0.223717772	0	1	1	1	TRENDING
98d472f20cae77b0c09f282210da5082	0.244705504	0	1	1	1	TRENDING
7340a3839a1de1e99d149b8cf052a2ec	0.649393281	1	2	5	0.4	LATE_BLOOMER

- Identifies false positives (early spikes that don't sustain) vs false negatives (slow starts that become winners)
- Trending(FRONT_LOADED) + high prob → "Flash deal" (Days 1-7, capture the spike)
- LATE_BLOOMER + high prob → "Watch & nurture" (Days 15-60, recheck at Day 28)

05
Conclusion

Recommendations

- Adopt category specific seasonal segmentation thresholds
- Flag non best sellers for better images, price, description, etc.
- Homepage slots, and paid ads to the top slice

Challenges

- Does not capture external influence such as brand image and quality
- Parameter tuning revealed that simpler models often beat complex ones
- Defining the appropriate label

Next Steps

- Further analysis on segmented predictions.csv
- Expand feature engineering to include external factors.
- Implement A/B testing to validate business impact

Google Colab Link:

https://colab.research.google.com/drive/1FQHHjC4g98R9T_1YLuvtLpsVWRc4Wm7X#scrollTo=ZbEXID5m5MiN

Thank You!

Any questions?