Forecasting Store Sales Using Machine Learning and External Factors

Erdun E, Hongyu Lai

Northeastern University

Introduction

Accurate sales forecasting is essential for inventory planning and retail decision-making. Using data from Corporación Favorita in Ecuador, this project builds machine-learning models that incorporate external influences — including oil prices, holidays, and store-level transactions—to predict product-level sales.

Good forecasting allows the company to determine how much inventory to keep in stock, which directly supports smoother and more efficient supply-chain operations.

Project Goal

The goal of this project is forecasting daily sales for different product families across multiple Ecuadorian stores by leveraging regression-based machine learning models and feature engineering that incorporates exogenous variables such as oil prices, holidays, and store-level transactions.

Method

Dataset

The dataset contains daily sales records from Corporación Favorita, a large retail chain in Ecuador, spanning January 1, 2013 – August 15, 2017. It integrates transactional data with several external sources such as oil prices, holidays, and store-level transactions, allowing regression modeling with exogenous variables.

File	Shape (rows * columns)	Description
`train.csv`	(3,000,888 × 6)	Main dataset containing labeled sales data
`test.csv`	(28,512 × 5)	Same structure as train but without `sales` (used for Kaggle submission)
`stores.csv`	(54 × 5)	Store metadata (city, state, store type, and cluster)
`oil.csv`	(1,218 × 2)	Daily oil prices (`dcoilwtico`), representing an economic factor
`holidays_events.csv`	(350 × 6)	National and local holidays with type, locale, and transfer flag
`transactions.csv`	(83,488 × 3)	Daily transaction counts per store, representing customer traffic

Model

1. Ridge Regression provides a simple and interpretable linear baseline.
2. Random Forest captures non-linear interactions and handles categorical splits effectively.
3. XGBoost extends this idea using gradient boosting, offering stronger generalization on tabular data.

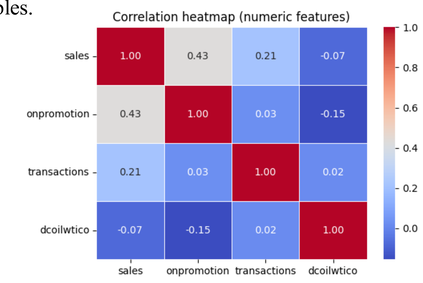
Summary of Model

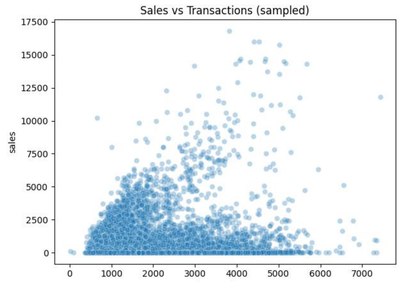
This combination moves from a simple linear approach to more complex ensemble learners, balancing interpretability and predictive power.

More EDA

Correlation Heat Map

No pair of features shows very strong correlation ($|r| > 0.8$), indicating no major multicollinearity issues among the numeric variables.





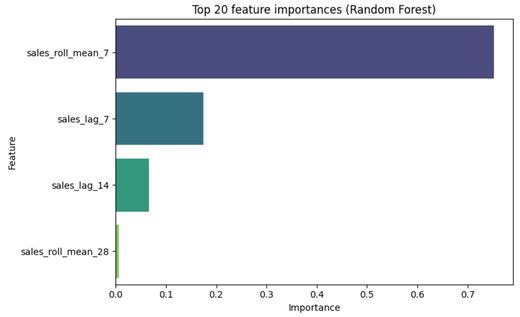
The scatter plot between `transactions` and `sales` supports this observation: a loose upward trend with wide dispersion, typical of aggregated retail data.

Overall, the numeric features are sufficiently independent to be included together in the regression models without strong redundancy

Feature Engineering

Lag and rolling features were added to capture short-term patterns. `sales_lag_7` and `sales_lag_14` store the values from one and two weeks earlier, while the 7-day and 28-day rolling means smooth daily noise. These features keep about 99% of the records and help the model learn weekly and monthly trends more effectively.

About 2.8 million records remain after removing rows without enough historical data. The new features show gradual variation and preserve the original ordering, confirming they were generated correctly within each time series. They are expected to improve the model's ability to capture trend continuity and seasonality.



A small random forest model was trained to estimate feature importance. `sales_roll_mean_7` and `sales_lag_7` dominated, confirming that short-term weekly patterns drive most predictive power. Longer lags such as `sales_lag_14` and `sales_roll_mean_28` also contribute moderately, capturing slower sales trends.

Data Pre-processing & EDA

Data Pre-processing

1. Helper tables (stores, transactions, oil, holidays and event information) are merged into train dataset to integrate training information and create this one large table for machine learning model training.
2. Below is the table sample with three rows (note: this is one table split into half)

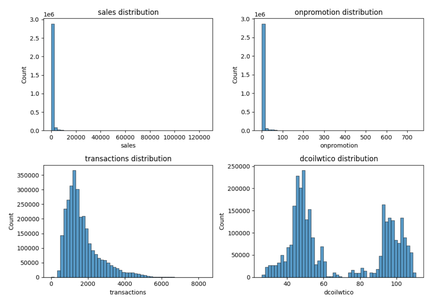
	id	date	store_nbr	family	sales	onpromotion	city	state	type	cluster	transactions	dcoilwtico	locale
0	0	2013-01-01	1	AUTOMOTIVE	0.0	0	Quito	Pichincha	D	13	NaN	NaN	National
1	1	2013-01-01	1	BABY CARE	0.0	0	Quito	Pichincha	D	13	NaN	NaN	National
2	2	2013-01-01	1	BEAUTY	0.0	0	Quito	Pichincha	D	13	NaN	NaN	National

transferred	holiday_Additional	holiday_Bridge	holiday_Event	holiday_Holiday	holiday_Transfer	holiday_Work Day
False	False	False	False	True	False	False
False	False	False	False	True	False	False
False	False	False	False	True	False	False

3. One-Hot-encoder is used to code categorical columns including family, city, state, type and cluster.
4. Numerical features are standardized: onpromotion, transactions and dcoilwtico so that values share a similar scale.

EDA

The summary statistics show large differences in scale among numeric variables. Daily sales range from 0 to about 125,000 with a mean of roughly 358, while `transactions` average around 1,687 per store and vary widely. The `onpromotion` column is zeros, reflecting that most items are not on promotion on a gimostlyven day. Oil prices `dcoilwtico` fluctuate between 26 and 110, showing multiple peaks that correspond to different market periods.



Experiments & Results

Hyperparameters Used

1. Random Forest:
   1. Parameter List: `rf_param_dist = {"n_estimators": randint(100, 400), "max_depth": randint(5, 20), "min_samples_split": randint(2, 20), "min_samples_leaf": randint(1, 10)}`
   2. Used RandomizedSearchCV to reduce searching time burden
   3. Best Parameters: `{"n_estimators": 164, "max_depth": 12, "min_samples_split": 8, "min_samples_leaf": 4}`
2. Ridge Regression:
   1. Parametes List: `alpha_grid = np.logspace(-3, 3, 7)`
   2. Used GridSearchCV
   3. Best Parameter: `alpha = (0.001)`

Results

Random Forest (tuned)

• Train RMSLE: 0.5783 Valid RMSLE: 0.4610

• Strong validation performance and the best among the three models.

XGBoost (default)

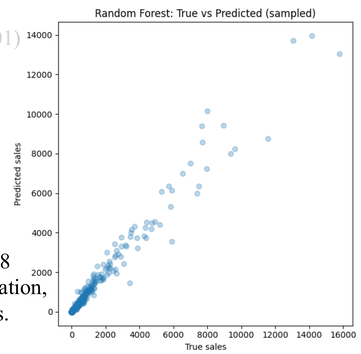
•Train RMSLE: 0.6578 Valid RMSLE: 0.4718

• Slightly worse than Random Forest on validation, but more stable between train and valid scores.

Ridge Regression (tuned)

• Train RMSLE: 1.0860 Valid RMSLE: 0.7883

•It captures overall trends but cannot model non-linear patterns.



Conclusions

The main takeaway is that tree-based ensemble models work much better than linear models for this sales forecasting task. Random Forest and XGBoost both captured the non-linear relationships in lag features and rolling statistics, while Ridge Regression underfit the data. The best overall performance came from the tuned Random Forest model. It handled non-linear relationships, interactions between lag features, and outliers better than Ridge Regression. Compared to XGBoost (default), Random Forest required less tuning to perform well and showed more stable validation metrics. XGBoost likely needs deeper tuning to outperform RF. Hyperparameter tuning also made a clear difference—especially for Random Forest, which became more stable and lowered its RMSLE.

References

Kaggle, "*Store Sales – Time Series Forecasting*," Kaggle, Accessed: Dec. 1, 2025. [Online]. Available: https://www.kaggle.com/competitions/store-sales-time-series-forecasting