


Lab 02 - Time Series Forecasting


Leaderboard Score



Store Sales - Time Series Forecasting

Submit Prediction

[Overview](#)
[Data](#)
[Code](#)
[Models](#)
[Discussion](#)
[Leaderboard](#)
[Rules](#)
[Team](#)
[Submissions](#)

27	UOM_210706H		0.38040	2	10s
----	-------------	---	---------	---	-----

Task

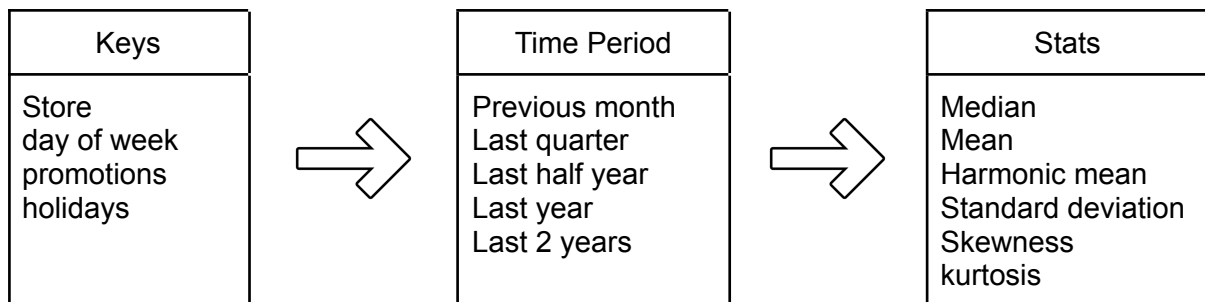
The task is to build a model that accurately predicts the unit sales for thousands of items sold at different Favorita stores. Five different tables were provided in order to forecast store sales.

Solution

The solution employed Extreme Gradient Boosting (XGBoost), a robust ML algorithm based on decision trees. The approach was designed around comprehensive feature engineering and strategic model ensembling, guided by three primary principles:

1. *Incorporating Recent Data:*

Leveraging store-specific sales history to capture recent trends and patterns.



2. Integrating Temporal Information:

Embedding temporal to account for seasonal effects, promotions, holidays, and other cyclical events.

1. Day counters (how each record relates to events or cycles)
 - The number of days before, after within the event
 - Events
 - ❖ Promotion cycle
 - ❖ Summer holidays
2. Day of week, day of month, day/ week/ month of year
3. Number of holidays during the current week, last week and next week

3. *Capturing Current Trends:*

Modeling ongoing sales trends to anticipate future changes.

- Last quarter and last year
- Store specific linear model (Linear Regression) on
 - ❖ The day number
 - ❖ Day of week
 - ❖ Promotions

The feature selection and ensembling process involved:

1. **Random Feature Selection:** Trained numerous models on randomly selected feature subsets to reduce bias from manual selection.
2. **Systematic Ensemble Building:** Evaluated validation errors for each pair of model ensembles. The top-performing model pairs were combined into a larger ensemble consisting of more than ten distinct models.
3. **Final Model Combination:** Aggregated features from all selected models into a single comprehensive model resulting in a robust and high-performing ensemble.

The modeling phase exclusively utilized **XGBoost**, emphasizing feature extraction and selection.

I applied a logarithmic transformation to the dependent variable (sales) to stabilize variance and normalize the distribution. Zero sales records were excluded from training.

Also utilized the harmonic mean to aggregate ensemble predictions effectively.

Data cleaning and preprocessing steps

Handling Missing Dates

- **Identified Issue:** The training dataset was missing four dates (December 25th from 2013 to 2016), likely due to store closures on Christmas.
- **Solution:** Added the missing dates to the training set and imputed corresponding `id`.

Missing Value Imputation

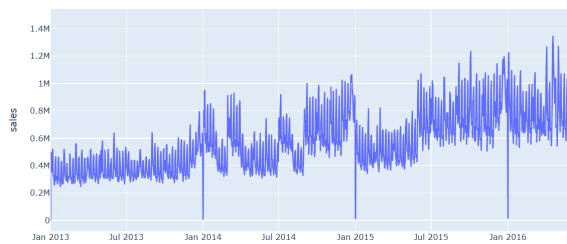
- **Numerical Columns** (`store_nbr`, `sales`, `onpromotion`): Filled missing values with `0`.
- **Categorical Column** (`family`): Filled missing values with `'none'`.
- **Oil Data** (`dcoilwtico`): Filled missing values using backward fill (`bfill`).

Data Merging and Cleaning - Removed columns such as `state`, `type_x`, `city`, `locale`, `locale_name`, `description`, and `transferred`.

Feature Engineering

- **One-Hot Encoding:** Applied one-hot encoding to the `family` categorical variable to prepare it for modeling.
- **Scaling:** Standardized numerical features (`store_nbr`, `sales`, `onpromotion`, `store_cluster`, `oil_price`, `Year`, `Month`, `Day`, `Quarter`, `Week of Year`, `Day of Week`, `is_weekend`) using `StandardScaler`.

Observed an upward trend on year (from EDA visualization)



Additional evaluation metrics

Beyond the commonly used Root Mean Squared Logarithmic Error (RMSLE), additional evaluation metrics were employed:

- **Mean Absolute Error (MAE):** Provides an average of the absolute errors, offering insight into the average magnitude of errors.
- **Mean Absolute Percentage Error (MAPE):** Expresses the error as a percentage.
- **Coefficient of Determination (R2 square)**

Alternative solution - Hybrid Model with LSTM and XGBoost

This hybrid model can capture both temporal and structural patterns, LSTM handles the time dependency, while XGBoost captures the impact of categorical and numerical features.

- By feeding the historical sales and temporal features (e.g., day, month, seasonality) into an LSTM model, the network learns underlying patterns and predicts sales trends for each store.
- After generating predictions using LSTM, feed these predictions as a new feature into XGBoost along with other manually engineered features (like promotions, holidays etc.). XGBoost can then model complex relationships between various static features and the LSTM's forecasted sales trends.

The hybrid approach will result in better accuracy compared to using either model alone.

Issues with your solution and how to improve it

Issues:	Improvements:
<ol style="list-style-type: none">1. Overfitting Risk- The extensive feature set and model ensembling, while beneficial, increase the complexity of the model.2. While the solution incorporated temporal features and trend estimation, higher-order seasonal patterns might not have been adequately addressed.	<ol style="list-style-type: none">1. Utilize automated feature engineering tools like AutoML to discover and create optimal feature sets.2. Utilize STL or SARIMA models to better capture and model higher-order seasonal patterns.3. Utilize time series cross-validation techniques, such as rolling window or expanding window approaches