Store sales prediction

Ralf Grossfeldt, Andreas Närep

Business goals

Background

Favorita is a grocery store chain located in Ecuador which needs a model for predicting sales more accurately. The project focuses on forecasting daily sales for a retail chain across multiple stores and product categories over several years. Accurate, data-driven sales predictions will enable better decision-making in inventory control, marketing campaigns, and staffing.

Primary goal: Develop an accurate sales forecasting model at the store and product-category level.

Specific goals:

1. Predict daily sales per store and product category to improve inventory and supply chain planning.
2. Assess the impact of promotions, holidays, and external factors (such as oil prices) on sales.
3. Identify time-related patterns including day-of-week and seasonal trends for optimizing store inventory management.

Business Success Criteria

- Achieve a Root Mean Squared Logarithmic Error (RMSLE) below 0.5 on unseen data, representing substantial improvement over current forecast methods.
- Predictive insights should allow managers to optimize inventory and promotions, reducing overstock or stockout events.

Situation assessment

Inventory of Resources

- Historical sales data (2013–2017) including store number, product category, date, sales, and promotions.
- External datasets: holidays, oil prices, transactions, and store characteristics.
- Tools and technology: Python, Pandas, Scikit-learn, LightGBM, XGBoost, and computational resources for training machine learning models.

Requirements, Assumptions, and Constraints

- Requirement: Daily time-series sales predictions with a horizon of at least 1–7 days.
- Assumptions: Sales patterns are influenced by promotions, holidays, and historical trends. Lag features and rolling averages are relevant for predictive modeling.
- Constraints: Time-series nature prevents standard bootstrapping; missing data must be addressed. Model training/tuning must be feasible on available hardware.

Risks and Contingencies

- Risk: Unusual events (e.g., natural disasters) might distort historical patterns.

Terminology

- **Lag features:** Previous values of a time series used as predictors.
- **Rolling averages:** Moving averages over a window to capture short-term trends.
- **RMSLE:** A measure of prediction accuracy that penalizes underestimation proportionally.

Costs and Benefits

- **Costs:** Computational resources for model training, data cleaning, and feature engineering.
- **Benefits:** Reduced inventory costs, optimized promotions, improved customer satisfaction, and data-driven decision support.

Data-Mining Goals

- Build machine learning models (Random Forest, LightGBM, XGBoost) to predict daily sales at store-category granularity.
- Incorporate time-series features including lags (1, 7, 14, 28 days), rolling averages, promotions, and holiday indicators.
- Evaluate models using a timewise consistent training/test split to prevent leakage.(i.e. no random samples etc)

Data-Mining Success Criteria

- RMSLE below 0.6 on a validation set.
- Models should meaningfully capture the effects of promotions, weekends, and holidays.
- Generated features (e.g., days until next holiday, weekend indicator) should improve model performance and provide actionable insights.

Gathering data

Data requirements

The project requires detailed, time-series data that captures daily sales trends at the granularity of store and product family. To accurately model retail dynamics, the dataset must include:

- **Historical sales** by date, store, and product family.
- **Store metadata** (location, city, state, type).
- **Product category information**
- **Promotion indicators**
- **Holiday and special-event data** with indicators of national, regional, and local events.
- **External variables** such as oil prices, which may reflect macroeconomic influences.
- **Transaction counts** as a proxy for store foot traffic.

The data must support creation of time-series features such as lag values, rolling statistics, and temporal fields (day of week, month, year).

Verifying data availability

All of the aforementioned data requirements are met with the datasets provided to us from kaggle.

Define Selection Criteria

The project will use data for all stores and product families with complete time-series coverage. Only relevant fields will be selected:

- From **train.csv**: date, store_nbr, family, sales, onpromotion.
- From **stores.csv**: city, state, store type, cluster.
- From **holidays_events.csv**: date, holiday type, transferred flag.
- From **oil.csv**: date and oil price column.
- From **transactions.csv**: date and transaction count.

Fields unrelated to predictive performance (e.g., descriptive holiday names, id-s) will be excluded.

Describing Data

The combined dataset contains millions of time-series records across 54 stores and 33 product families. Key fields:

- **date:** daily timestamps.
- **store_nbr:** store identifier.
- **family:** product category (string).
- **sales:** target variable, numeric with strong right-skew, with lots of 0 values.
- **onpromotion:** binary indicator of whether the item was on promotion on that date.
- **city/state/type:** categorical descriptors of each store.
- **holiday_type/month/day/year/is_weekend:** derived temporal features.
- **oil price:** external continuous variable with occasional missing values.
- **transactions:** daily store-level customer counts.

Sales distribution is highly skewed, with many low or no-sale days and occasional spikes during holidays or promotions. Missing values appear mainly in oil prices and in some holiday categories, which need to be addressed especially in time-series modelling.

3. Exploring Data

Initial exploratory analysis reveals:

- **Strong weekly seasonality:** sales peak Friday–Sunday; weekends show increased demand.
- **Holiday effects:** major holidays such as Christmas, New Year's, and the days leading up to them produce noticeable spikes.
- **Promotion effects:** days with active promotions correlate with higher sales.
- **Store variation:** some stores show consistently high transaction volumes and higher mean sales, reflecting demographic or regional differences.
- **Oil price trends:** A sharp drop in oil prices can be seen in the data. Overlaying it with sales we see that the drop correlates with a drop in sales. Afterwards however, sales seem to recover whilst oil prices remain low. This means that oil prices may not be a good indicator of sales over the long term.

Exploration of missing values confirms:

- **oil price** has missing entries due to non-trading days or data gaps; interpolation is appropriate.
- **holidays "transferred"** require consolidation to avoid double-counting.

Verifying Data Quality

Data quality checks identify:

- **Missing days:** Train.csv has missing days, which need to be addressed for modelling to work on the data.
- **Consistency:** store and family values match between datasets.
- **Duplications:** none detected for the primary key (date, store_nbr, family).

- **Type consistency:** some fields are objects requiring conversion to categorical types for modeling efficiency.
- **Missing oil prices:** must be filled using forward/backward fill.
- **Holiday inconsistencies:** “transferred” holidays must be reconciled so that only one holiday indicator exists per date.

The dataset is overall of high quality for time-series modeling, requiring only standard cleaning and feature engineering steps.

Project plan

1. Fixing the missing entries in train.csv and oil.csv. Dealing with the other aforementioned data-quality problems. Andreas will work on these ~ 2h
2. Doing EDA with plotting to get a better feel for the dataset and ideas for feature engineering. Ralf will work on this ~ 4h
3. Feature engineering - Creating lags merging the multiple datasets etc. Andreas and Ralf will both work on this for ~1.5h.
4. Creating the first working models (XGBoost, RandomForest, LightGBM) to get the first baseline results ~ Andreas 2h
5. Optimizing datatypes to make model fitting more efficient. Ralf ~ 1h
6. Doing additional feature engineering and optimization for these models to improve RMSLE ~ Ralf 3h
7. Hyperparameter tuning. Andreas ~ 3h + however long the fitting takes on the computer.
8. Analysis of the results. We don’t know yet, depends on the results.
9. Troubleshooting!!! We will inevitably run into all sorts of errors and issues whether it is GIT or Python errors. We don’t know how long this part will take but it would be fair to add another +2h for the both of us.

NB! All of these tasks will probably take longer than anticipated, it’s Hofstadter’s Law:).

Repository link: https://github.com/RalfGrossfeldt/Store-Sales-Prediction