ITISE 2022

Feature Study on Forecasting Hourly Retail Time Series

Bruno Mendes, IEETA
Ana Tomé, IEETA
José Moreira, DETI

Retail Time Series

Time Series: A set of data collected at successive points in time or over successive periods of time.

Retail Time Series, specially when studied hourly, have two special features:

- they have multiple seasonal components (daily, weekly and annual);
- they are inherently irregular.

Objective

Provide useful information on forecasting customer flow by **performing a feature study** on this type of time series.

- Use Support Vector Regressors with an Autoregressive Prediction strategy to **study the feature importance**.
- Build a Regression Pipeline to **validate** the results obtained.

The Datasets

The present work comprises a total of seven time series, all of them consisting of customer flow data from real retail stores.

- The data was recorded between January 2015 and October 2020.
- Stores 1-5 are used for the study itself. Stores 6-7 for testing.
- Three months will be chosen as test horizons:
  - November 2018
  - June 2019
  - February 2020

The Datasets
Detailed Description

	store 1	store 2	store 3	store 4	store 5	store 6	store 7
Mean	113,07	91,16	85,92	70,57	58,94	76,69	90,91
Std	54,43	50,64	48,82	37,58	27,82	35,76	35,69
Min	1	1	1	1	1	1	1
25%	69	50	49	44	39	48	62
50%	106	78	79	67	57	71	85
75%	150	129	117	96	79	100	120
Max	316	262	245	228	164	224	224

Normalizing Schedules
Why?

A machine learning model doesn’t read actual dates and times, but ordered samples.

This procedure will help to achieve two goals:

- Preserve the **daily seasonality**: by asserting that every day has the same number of hours (i.e. the same amount of data samples).
- Preserve the **weekly seasonality**: by asserting that every week has the same number of days.

Normalizing Schedules
Workflow

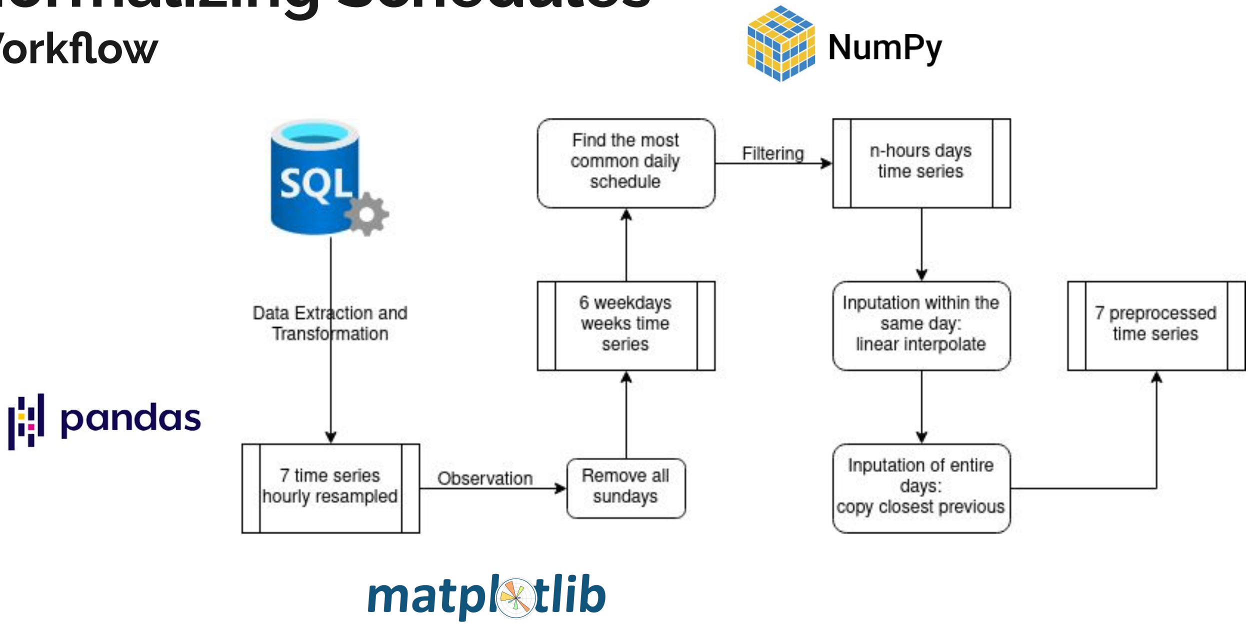

Fig. 1. Time Schedule Normalization Workflow.

Preparing the Data

Two more final preparations before the forecast:

- Transform the time series into a supervised machine learning problem
  - for **Autoregressive strategy**.
  - 5 weeks of lags were used (5 * 6 * 13 = 390 lags).
- Standardize the time series (with z-score normalization)
  - for **Support Vector Machine**.

Feature Importance Study
Support Vector Regression

The linear SVR model can be formulated with the equation below.

$$x[n] = \sum_{k=1}^{P} w_k x[n-k] + w_0, \quad n = P+1 \ldots$$

Each predicted value is the result of a weighted sum of *P* past samples and a bias w_0.

For this study, the weight values will be used for feature ranking.

Feature Importance Study

Autoregressive Predictions strategy

The training set is formed applying a sliding window to read $P + 1$ samples, in which P represents the length of the feature vector as well as the number of samples in the past that will be used to forecast a single value.

The next input is obtained by moving the window forward one step, so the last predicted value becomes a feature in the next feature vector. The oldest value is dropped.

- In the present case, 390 sequential values (5 weeks of information) will be used to predict each time point.

Feature Importance Study
Forecasting

Model used: sklearn.svm.LinearSVR

Configuration:

- dual = False (solve the primal optimization problem)
- loss = ‘squared_epsilon_insensitive’
- everything else was left with the default values

Number of features: 5 weeks * 6 days * 13 hours = 390 features

Total of tests to run: 5 stores * 3 horizons = 15 tests

Feature Importance Study
Forecasting Results

	store 1	store 2	store 3	store 4	store 5
R2	0.92	0.94	0.93	0.89	0.84
MAPE	0.11	0.13	0.13	0.13	0.13
MAE	10.64	9.37	9.33	8.77	7.94
RMSE	13.78	12.11	11.65	11.36	10.14

Feature Importance Study
Feature Importance Decay

After the 10th feature, the value of the coefficients decreased around one order of magnitude.

- No more than the 20 top features from each model were studied.

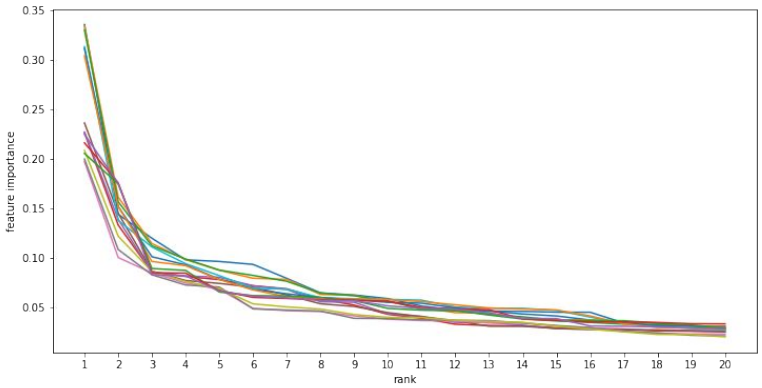

Fig. 2. Feature importance decay on the top 20 features.

Feature Importance Study

Features with the most occurences

Across all 15 cases of study, there were 11 features that were always ranked in the top 20:

- The one lag referring to the **previous hour**;
- The five lags referring to the **target hour from the previous five weeks**;
- The five lags referring to the **target hour from the previous five days**.

Features referring to **the hours previous to the target hour** were also frequent but not absolute as the features mentioned above. Their presence went as follows:

- 2 hours ago: 13 occurrences;
- 3 hours ago: 12 occurrences;
- 6 hours ago: 10 occurrences;
- 5 hours ago: 7 occurrences;
- 4 hours ago: 6 occurrences.

Regression Pipeline

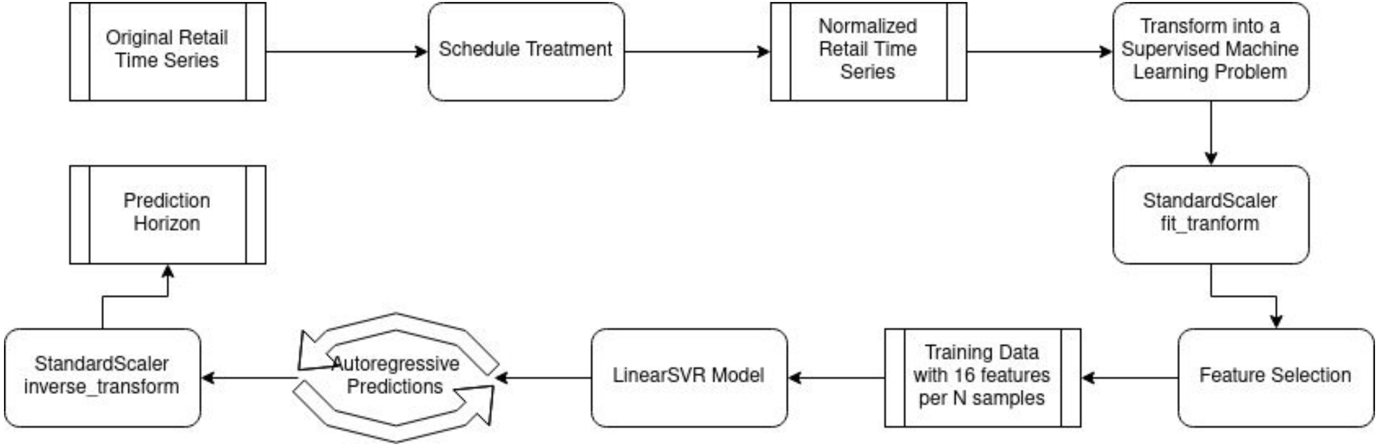

Fig. 3. Regression pipeline developed.

Results

Pipeline Forecasting Results

	store 1	store 2	store 3	store 4	store 5	store 6	store 7
R2	0.90	0.94	0.92	0.82	0.78	0.85	0.84
MAPE	0.12	0.13	0.14	0.18	0.17	0.15	0.13
MAE	11.75	9.65	9.97	11.67	9.58	7.98	11.01
RMSE	15.17	12.48	12.55	14.58	11.90	10.42	13.82

Results

Pipeline Forecasting Plots

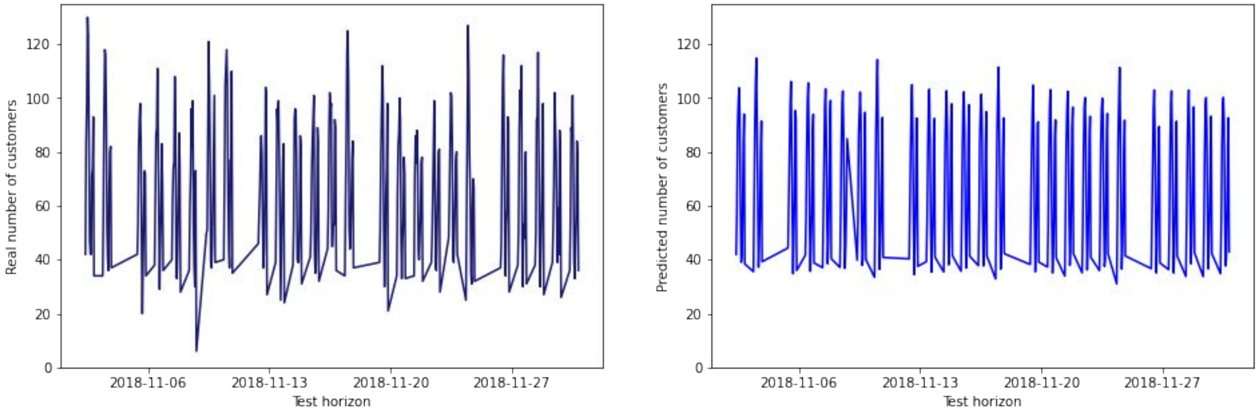

Fig. 4. Graphic representation of the predictions results for store 6 in the first horizon test. *Left*: real customer flow; *Right*: predicted customer flow.

Results

Autoregressive Test Error

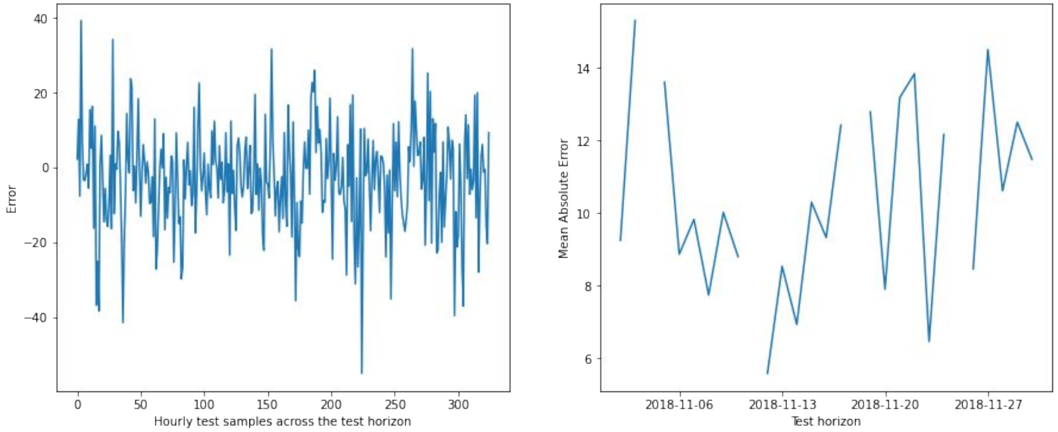

Fig. 5. Prediction error for a store. *Left*: time series error; *Right*: mean absolute error per day.

Discussion

- The application of the regression pipeline was proved to be a **resilient solution**.
  - The number of features was reduced from 390 to only 16.
  - Stores 1 to 3 barely suffered any performance penalty.
- In figure 5, it's possible to see that the pipeline was **able to capture both weekly and daily seasonalities**.
  - In fact, 10 out of the 16 selected features represent seasonal values. This selection is only possible because the schedules are normalized.
- **No trend** was present in any of the studied time series.
  - An artificial trend was added for testing and the results were just as good.

Thank you!

universidade de aveiro