Store Sales - Time Series Forecasting

Nicolas Desan - Damien Chanet

1) Introduction

The purpose of this dataset is to use time series forecasting to the forecast store sales of the Ecuadorian grocery retailer : Favorita. The interest of such study for the grocery store is to potentially decrease food waste related to overstocking and improve customer satisfaction. Moreover, we already possess a training and a test dataset with also Store metadata including city, state, type and cluster (grouping of similar stores), an oil dataset containing daily oil price and a holiday and events metadata.

2) The variables

In the training dataset we possess 5 features and 1 target variable (sales) :

- store_nbr : identifies the store at which the products are sold.
- family : identifies the types of products sold. There are 33 different types of products.
- sales : gives the total sales for a product family at a particular store at a given date.
- onpromotion : gives the total number of items in a product family that were being promoted at a store at a given date.

Moreover we have the date of all those sales daily so we have many sales per day. It might be considered later to group those sales by day and to study daily sales overall to get some information. Also, the train dataset has the shape (3 000 888,6) and has 1684 days recorded.
The test dataset has the same features and we will predict the target sales. The test dataset has the values of 15 days after the training data. The target variable is absent from this dataset. Therefore the test dataset will no longer be considered after and we will only consider the train dataset as our main dataset.

Finally, we can already get some statistical information about the train dataset :

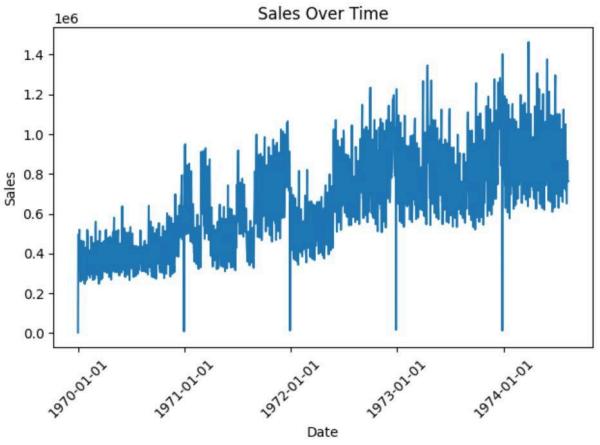
	store_nbr	sales	onpromotion
count	3.000888e+06	3.000888e+06	3.000888e+06
mean	2.750000e+01	3.577757e+02	2.602770e+00
std	1.558579e+01	1.101998e+03	1.221888e+01
min	1.000000e+00	0.000000e+00	0.000000e+00
25%	1.400000e+01	0.000000e+00	0.000000e+00
50%	2.750000e+01	1.100000e+01	0.000000e+00
75%	4.100000e+01	1.958473e+02	0.000000e+00
max	5.400000e+01	1.247170e+05	7.410000e+02

3) Overall view of the sales

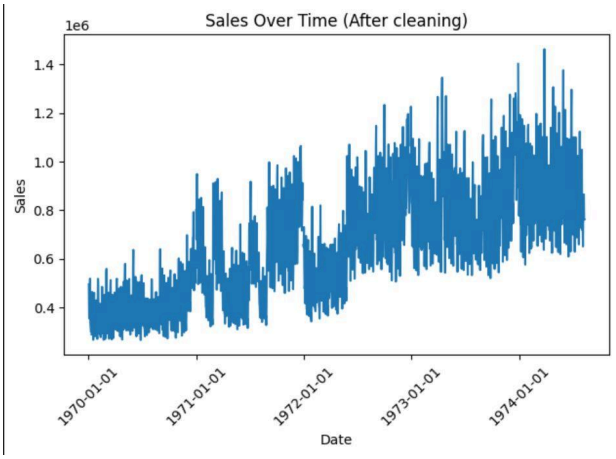
It is interesting to see the general tendency of all the sales by summing their daily values in order to get a more global point of view of the general tendencies.
Considering the overall sales per day (without specifying the store or the promotions) we get those information about the sales :

- count : 1684
- mean : 637 556,4
- std : 234 410,2
- min : 251,16
- 25% : 442 711
- 50% : 632 188,9
- 75% : 785 945,5
- max : 1 463 084

Additionally, with a graphical visualization of the data, we notice a discontinuity at the beginning of each year in the set of sales, which occurs periodically. This could be due to an annual maintenance error that should be corrected by replacing the null values with the average sales (specifically, 637,556):



Here we have the 99,5% interval for the sales is [265334.02404564247, 1271123.038601575]. So we can consider that all the values under 265335 can be replaced with the mean value and keep a good approximation and get rid of the discontinuity.



4) The other datasets

Upon gaining an overview of the dataset **train.csv**, our next step involves analyzing additional datasets:

1. stores.csv : The "Stores" dataset provides information about various stores, including details such as city, state, type, and cluster. This dataset is crucial for understanding the characteristics of different stores.

2. transactions.csv : The "Transaction" data exhibits a high correlation with the sales column in the training dataset ("train.csv"). Analyzing this dataset enables us to discern sales patterns across different stores. The Correlation between Total Sales and Transactions: 0.8175.

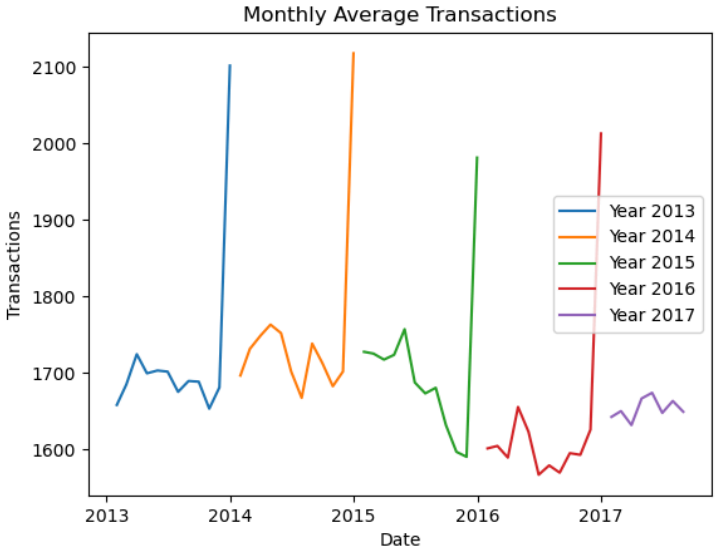
3. hollidays_events : The data related to holidays and events serves as metadata. This dataset helps for comprehending historical sales and identifying trends and seasonality components.

4. oil.csv : The dataset containing daily oil prices is another valuable resource.

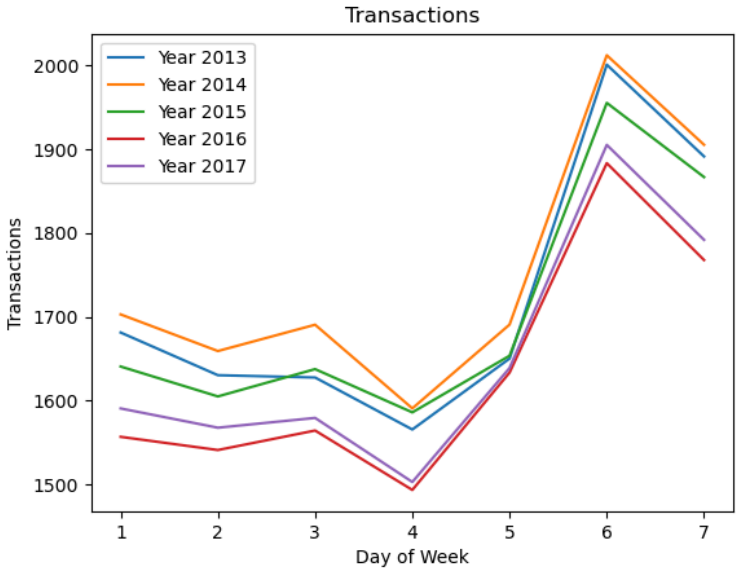
The transactions :

The transactions and sales are highly correlated. Studying transactions can help us to understand the sales, which are the targets of our project and our model.

There is a stable pattern in Transaction. Yet, every year, the transactions skyrocket in december. Taking those pseudo-periodic patterns can help our deep learning model to make better predictions for transactions.

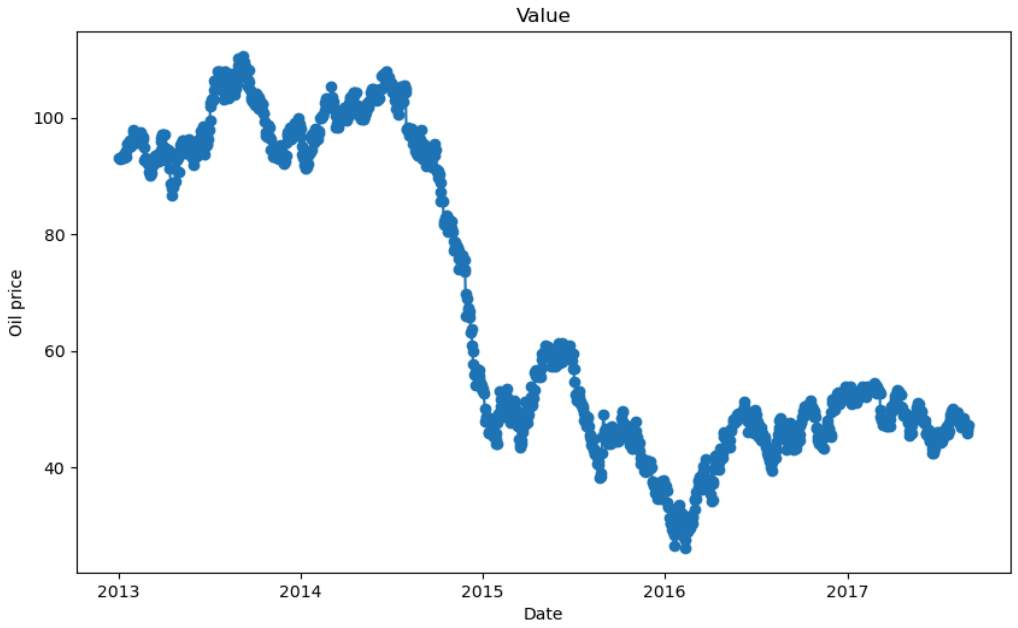


The significance of weekdays in the context of shopping cannot be overstated, as they reveal distinct patterns. Across the years 2013 to 2017, stores consistently observe heightened transactional activity during weekends. Notably, Saturday emerges as the pivotal day for shopping, consistently reflecting the most substantial impact on transaction volumes. Taking the weekdays into account in our model will make better predictions.



The oil price :

Ecuador faces a unique situation as an oil-dependent country. Fluctuations in oil prices directly impact the economic landscape, introducing a variable that can significantly affect the sales and our model. The changing oil prices in Ecuador are anticipated to introduce variance and necessitate consideration in our analytical approach. We can see the higher the oil price is, the less sales we have. The oil price has a negative correlation with the sales.



Choice of the input parameters :
The next step is to create new parameters and normalize our dataset “train” taking into account all the useful information for our model. We will use the month of the year, the day of week, the day of the month, the oil price, the daily sales and the daily transactions of each store with their store number.

5) The Final Dataset :

With all the datasets at our disposal, we decided to gather all the data that significantly influences the daily sales of each store. Therefore, we selected the following parameters:

- store_nbr : the store id
- daily_sales : the daily sales of the store of all type of products
- daily_transactions : the daily transactions of the store
- daily_onpromotion : the daily promotions of the store of all type of products
- dcoilwtico_interpolated : the oil price including interpolated values because of missing ones
- day_of_week : the day of the week
- day_of_month, month, year
- is_national_holidays : if there is a national event on this day of not
- is_regional_holidays : if there is a regional event on this day or not
- is_local_holidays : if there is a local event on this day or not .

This dataset will be separated in two parts, the train and the test dataset.

Then, with the both train and test datasets, we can create X_train X_test and Y_train Y_test which respectively represent the train and test parameters and the train and test target. The variable Y_test will be useful for us because we can compare it to the Y_predict, the output of our model including the predicted daily sales of each store, in order to evaluate our model.

In order to train the models correctly and avoid a disproportionate influence of some variables on the model. Ensuring that all features contribute equally to more accurate and meaningful data interpretations and predictions. Also, this can improve the overall training efficiency.

6) The choice of the model :

The choice of LSTM

For the model, we have decided to develop an LSTM model. Indeed, LSTMs are excellent for time series predictions such as sales forecasts because they can learn and retain long-term dependencies in the data, which is crucial for understanding trends and seasonal cycles. Moreover, LSTMs are particularly suited for time series as they are designed to work with data sequences. Finally, LSTMs can handle time series with irregular time intervals and variable data patterns.
Finally, in order to choose our model, we will compare the different RMSE.

The choice of RNN

In some works of the kaggle, there is some work with the RNN model. Indeed, the RNN is particularly adapted to study time series because it has a memory that allows it to remember the previous input information. Also, the RNN are conceived to recognize and modelize the temporal dependencies which is vital to see the seasonal tendencies.

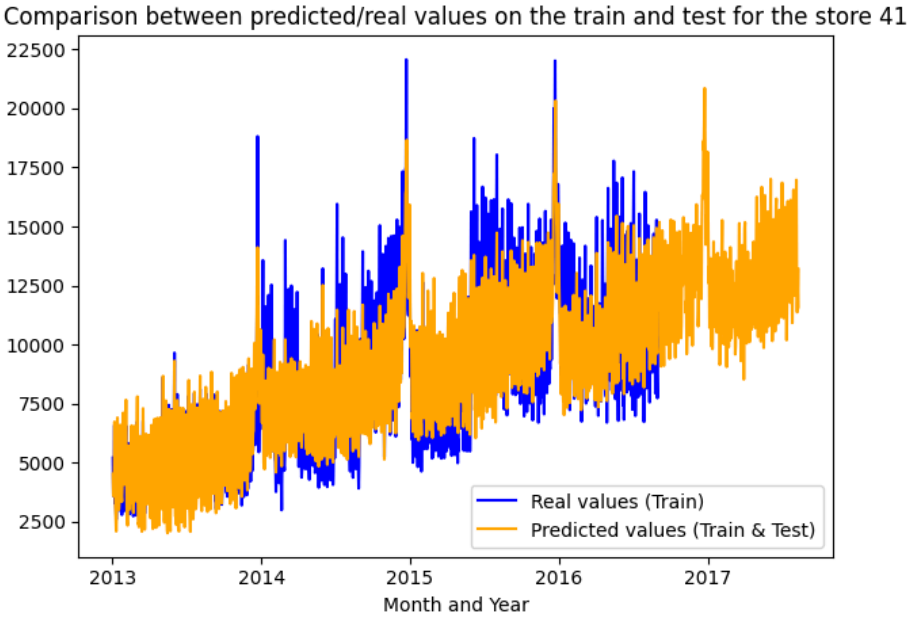
The input and the output of the model

Our dataset comprises all the daily data from the 54 stores essential for the daily sales predictions that our model will provide. Two choices were available to us: 1) take the data from all 54 stores as input and return the sales forecasts for each store as output, 2) take the data from a selected store as input and return the daily sales predictions for that store as output. For ease in developing our model and neural network, we opted for option 2. Additionally, displaying the results for one store at a time enhances comfort and readability of the sales predictions. When we seek the sales forecasts for store “store_nbr”, we extract only the data pertaining to store “store_nbr” from the dataset. Therefore, we have X_train and X_test as input, Y_train_predict and Y_test_predict as output, and Y_train and Y_test as the actual daily sales values.

The model

To create the model, we developed the function 'make_store_prediction.' This function defines a regression model based on Long Short-Term Memory (LSTM) recurrent neural networks to predict the daily sales of a specific store (store_nb). The LSTM model is specified with a layer containing a defined number of neurons (lmst_nb). The default activation function is used, and a Dense layer with a single output is added for the prediction of daily sales. The model is compiled using the Adam optimizer and the loss function defined

as the mean squared error.Subsequently, the model is trained on the training data for 20 epochs with a batch size of 32. After training, the model is evaluated on the test data, and performance metrics such as RMSE (Root Mean Squared Error) and R2 are calculated and displayed.
Predictions are then denormalized to obtain the actual daily sales values, and the results are visualized using graphs (Comparison between predicted/real values on the train, Comparison between predicted/real values on the test). Below are the graphs generated when executing the code 'make_store_prediction(50, 20)."



7) The results :

Here below are the results for different stores with different number of LSTM neurons :

Input\Output	RMSE Train	RMSE Test	R² Train	R² Test
Store n°3 / LSTM(20)	0.20 ± 0.03	0.26 ± 0.03	0.77 ± 0.12	0.53 ± 0.12
Store n°3 / LSTM(50)	0.58 ± 0.03	0.64 ± 0.03	0.72 ± 0.12	0.48 ± 0.12
Store n°27 / LSTM(20)	0.20 ± 0.02	0.24 ± 0.02	0.80 ± 0.10	0.60 ± 0.10
Store n°31 / LSTM(20)	0.17 ± 0.04	0.24 ± 0.04	0.84 ± 0.12	0.60 ± 0.12
Store n°31 / LSTM(40)	0.16 ± 0.06	0.27 ± 0.04	0.86 ± 0.18	0.51 ± 0.18
Store n°46 / LSTM(20)	0.43 ± 0.16	0.74 ± 0.16	0.83 ± 0.10	0.63 ± 0.10

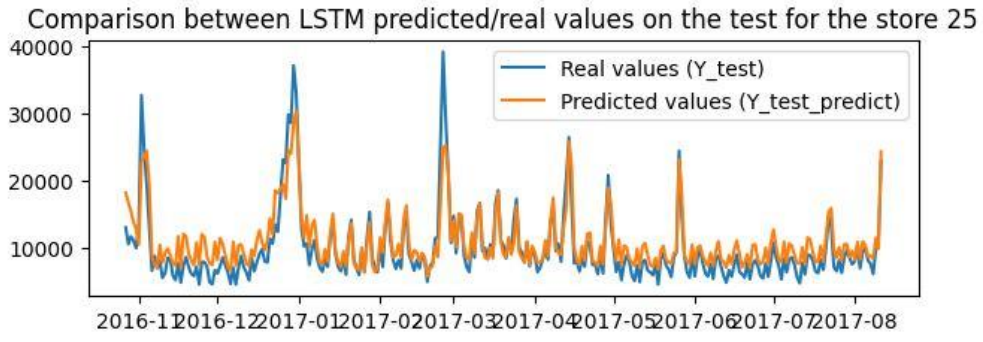
We tested various structures of the LSTM neural network, and the structure that yielded the best scores is a single layer with 20 LSTM neurons as input. Adding an additional layer of LSTM neurons resulted in a decrease in the model's performance and scores (RMSE and R^2).The graphs illustrating the comparison between actual and predicted values show that the model has effectively captured the pseudo-periodic patterns in daily sales for weeks and years, integrating them into the sales forecasts. However, it is noticeable that the model struggles to predict extreme values. On average, the model predicts less variation than the actual values.

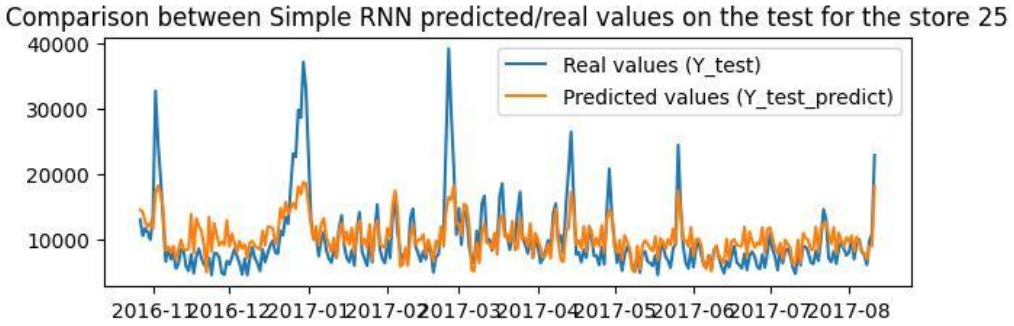
Comparison between the results of 3 different models :

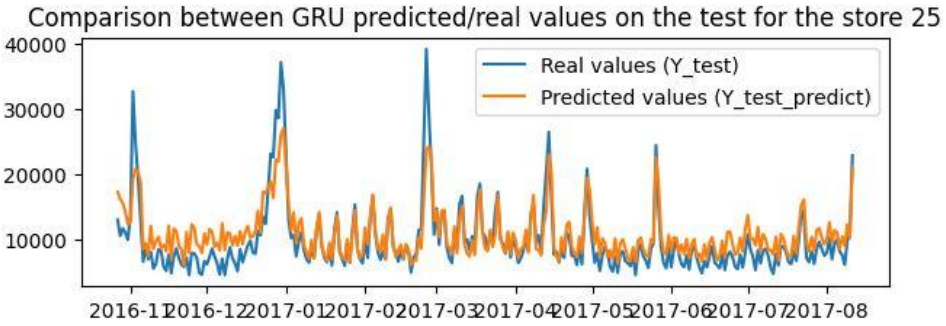
In order to be sure of having chosen the best type of neurons for our model, we tried to compare the performance results of our first model with LSTM neurons with two other models comprising two other types of neurons. Here are the comparison of the performances of the 3 models with different types of recurrent neurons (Simple RNN, LSTM, GRU) for three different stores (Store #4, Store #14, Store #38) with different neuron configurations (20, 15 , 40 neurons) :

	Simple RNN	LSTM	GRU
Store n°4 20 neurons	RMSE Train: 0.22 RMSE Test: 0.26 R^2 Train: 0.70 R^2 Test: 0.50	RMSE Train: 0.20 RMSE Test: 0.25 R^2 Train: 0.76 R^2 Test: 0.56	RMSE Train: 0.20 RMSE Test: 0.22 R^2 Train: 0.76 R^2 Test: 0.64
Store n°14 15 neurons	RMSE Train: 0.14 RMSE Test: 0.23 R^2 Train: 0.44 R^2 Test: -0.24	RMSE Train: 0.08 RMSE Test: 0.13 R^2 Train: 0.82 R^2 Test: 0.60	RMSE Train: 0.08 RMSE Test: 0.13 R^2 Train: 0.83 R^2 Test: 0.62
Store n°38 40 neurons	RMSE Train: 0.23 RMSE Test: 0.29 R^2 Train: 0.68 R^2 Test: 0.48	RMSE Train: 0.20 RMSE Test: 0.24 R^2 Train: 0.77 R^2 Test: 0.64	RMSE Train: 0.20 RMSE Test: 0.26 R^2 Train: 0.77 R^2 Test: 0.58

The LSTM and GRU models appear to be solid choices for these daily sales predictions, with a slight preference for LSTM in some specific cases. This confirms the general trend in the field where LSTM and GRU are preferred for processing complex sequences with long-term dependencies.







The analysis of the results

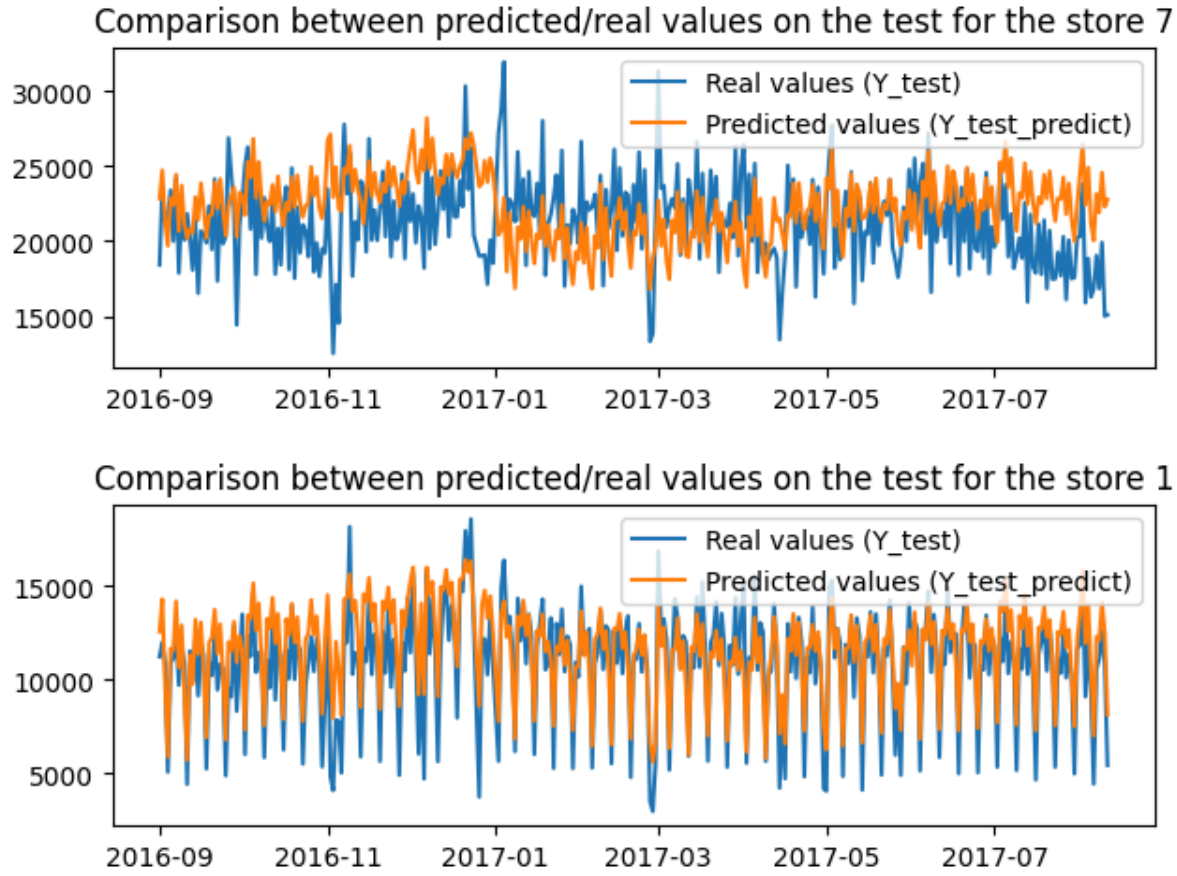
Overall, LSTM models show robust ability to model temporal sequences, with high R^2 values on both training and test sets (R^2 train between 70 and 85 - R^2 test between 50 and 70). Performance varies slightly from one store to another, which may be attributed to differences in data complexity and specific patterns in each store. The relatively high test scores suggest that LSTM models generalize well to new data, which is crucial for forecasting applications.

8) Conclusion and perspectives :

LSTM models appear to be a solid choice for these daily sales predictions, demonstrating a strong ability to capture temporal dependencies in time series data sequences and generalize to new scenarios. The daily sales predictions for each store are generally well-performed; however, new perspectives emerge :

- Our model performs well for around 90% of the 54 stores, but has difficulty predicting the daily sales of some stores like store n°7. The perspectives available to us and to discover the reasons for its 10% bad predictions. There could be several reasons why your model for predicting daily sales across the 54 stores is not performing well for 10% of the cases. There are maybe missing values, input errors, or inconsistencies in the data that can negatively impact model performance, or insufficient or poorly selected features. We can experiment with different values for these hyperparameters because neural networks are sensitive to hyperparameters. Moreover, some stores may have more variable sales behaviors than others. Optimizing the performance of a sales prediction model is an iterative process that involves thorough data exploration, careful feature selection, hyperparameter tuning, and error analysis for continuous improvement.

We can see the difference between the bad predictions for store n°7 and the good ones for store n°1.



- The original dataset includes daily sales for the 33 product types across the 54 stores of the Ecuadorian Favorita brand. Due to the substantial data volume when importing the 'train.csv' file and for the simplification of our model, we decided to aggregate all product types into daily sales for each store to make predictions for the daily sales of any chosen store among the 54. The new perspective before us is to create a new, more general model that predicts the daily sales of the 33 product types for a chosen store. Therefore, the model will take as input a matrix of size 33 by 11 representing the parameters of the 33 product types and will output a vector of size 33 representing the daily sales prediction for each of the 33 product types.