CAPSTONE PROJECT PROPOSAL
Udacity – Machine Learning Engineer ND Program
Corentin Mouquet 2020-07-02

Walmart Store Sales Forecasting

- **Domain background**

  As a Supply Chain Manager in retail, I'm particularly interested in sales forecasting. Indeed, Supply Chain management is all about having the appropriate amount of products, in the right place, at the good time. Too little stock leads to high percentage of out of stock items, on the contrary too much stock leads to overstock: Two major Key Performance Indicators in Supply Chain. We easily understand the positive or negative impact for a company, in term of customer satisfaction, turnover and cashflow.

  Better sales forecasts allow a better supply chain management, enabling to order or ship the good quantity of products, at the right time, to the good store.

  In this project, I choose to work on sales forecasting for Walmart stores from a Kaggle dataset.

  Here is some information about Walmart from Wikipedia:
  "Walmart is an American multinational retail corporation that operates a chain of hypermarkets, discount department stores, and grocery stores (…) As of April 30, 2020, Walmart has 11,484 stores and clubs in 27 countries (…) Walmart is the world's largest company by revenue, with US$514.405 billion, according to the Fortune Global 500 list in 2019"

- **Problem statement**

  This is a time-series forecasting project. We get historical sales data for 45 Walmart stores and for each of theirs departments. We will have to project sales for each store and each department the more precisely as possible.

  Moreover we will have to deal with usual sales forecasting problems: missing data, no or short sales history for instance.

- **Datasets and inputs**

  The data we will use for this project is from Kaggle competition: "Walmart Recruiting - Store Sales Forecasting" (https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting)

  The dataset is composed of 3 csv files:

- A ***Train*** file containing weekly sales evolution for each store and department of the dataset from 2010-02-05 to 2012-11-01. With 45 stores and 81 departments the file contains 421570 lines

In this file we find the following information (this part directly from Kaggle)
- *Store: the store number*
- *Dept: the department number*
- *Date: the week*
- *Weekly_Sales: sales for the given department in the given store*
- *IsHoliday: whether the week is a special holiday week*

From this file, we will create training and test time series. Test time series will be 36 weeks long. This is the same length than original test file from Kaggle competition, but we will be able to calculate ourselves the evaluation metrics and display predicted versus real weekly sales.

- A ***Features*** files containing 12 features and their evolution through time for each store. The file contains 8190 lines

In this file we find the following information (this part directly from Kaggle)
- *Store: the store number*
- *Date: the week*
- *Temperature: average temperature in the region*
- *Fuel_Price: cost of fuel in the region*
- *MarkDown1-5: anonymized data related to promotional markdowns that Walmart is running. MarkDown data is only available after Nov 2011, and is not available for all stores all the time. Any missing value is marked with an NA.*
- *CPI: the consumer price index*
- *Unemployment: the unemployment rate*
- *IsHoliday: whether the week is a special holiday week*

During data exploration we will look at correlation between these features and weekly sales. If some features have really low impact on sales, we will not take them into account to train our model. Moreover, if between them, some features are highly correlated we could only take one of them into account to make our model more efficient.

Finally, to make a future prediction, the only certain and reliable features will be Store, Date, IsHoliday and Markdown if Walmart know them in advance. Other features are not precisely predictable (or it's another project) so it will be better to avoid taking them into account.

- A ***Stores*** file containing the size and type (A, B or C) of each store. The file contains 45 lines

In the same way as with the features files, we will take into account size and type of store in our model only if this has a clear impact on weekly sales evolution.

- **Solution statement**

  After analyzing the dataset, the solution will be a trained model that we will use to forecast sales and get better results than our benchmark model given evaluation metrics.

- **Benchmark model**

  To compare our results, we will use some simple predictions like a model based on last year's sales or last week's sales.

- **Evaluation metrics**

  We should first verify that our data sales evolutions are not following a random walk. Evaluation metrics in this case could seems pretty good but really misleading, as it will be by definition impossible to predict future outcomes. (https://towardsdatascience.com/how-not-to-use-machine-learning-for-time-series-forecasting-avoiding-the-pitfalls-19f9d7adf424)

  There are lots of evaluation metrics for time series forecasting, for instance:

  MSE - Mean Square Error
  RMSE - Root Mean Square Error
  MAPE - Mean Absolute Percentage Error
  SMAPE - Symmetric Mean Absolute Percentage Error
  MAE - Mean Absolute Error…

  We will use at least WMAE, weighted mean absolute error, which is the metric used to evaluate the competition by Kaggle.

$$\text{WMAE} = \frac{1}{\sum w_i} \sum_{i=1}^{n} w_i |y_i - \hat{y}_i|$$

  where

  - n is the number of rows
  - $\hat{y}_i$ is the predicted sales
  - y_i is the actual sales
  - w_i are weights. w = 5 if the week is a holiday week, 1 otherwise

- **Project design**

  1) Load and explore the data

  We will first load and save the files, pre-process the data, look at the correlation between features and global sales evolution.

2) Create training and test sets of time series

Then, we will create the appropriate inputs to train a chosen model:

- Benchmark models based on last year or last week sales
- AWS DeepAR model requires a JSON format to pass sales evolution, store and department number and chosen dynamical features
- Scikit-learn Random Forest model

3) Train the model
4) Deploy (for DeepAR) and get a prediction from the model
5) Evaluate the predictions with our metrics and select the final model