Predicting container's dwell time

Maël Fabien

Stationnement - Conteneur - Deep Learning

1 Context

More than 80% of the goods consumed worldwide are transported in containers. The invention of the container is no less than the greatest revolution in our modern society since Internet. It is the basis of the modern supply chain and the globalization that has taken place over the last 30 years.

Every day, millions of containers are loaded and unloaded from cargo ships. Terminal operators unload containers from ships and store them while waiting for the trucking company to pick them up. We would like to predict the dwell time for each container.

2 Exploratory data analysis

2.1 Features

The features available in the given data set are the following :

- *idcontainer* : the reference of the container
- *feet* : length of the container
- *reefer* : is the container reefer or not
- *hazardous* : is the container dangerous or not
- *weight* : weight of the container
- *estimatedtimearrival* : estimated arrival time of the container
- *actualtimearrival* : actual arrival time of the container
- *idportcall* : reference of the ship's stopover
- *shippingline* : shipping company
- *freightforwarder*
- *truckingcompany*
- *finaldestination*
- *dwelltime*

2.2 Data Preparation

Several steps have done in order to clean the data set :

- replace NaN in *reefer* and *hazardous* columns by 0s
- replace missing weight values by the average weight
- replace missing estimated arrival time by the actual arrival time

- create a new category for missing $freightforwarder$ values

Then, in order to handle the diversity of the data types, all the variables are converted to categorical. The following changes have been made in order to allow this transformation :

- the weights have been embedded into a set of 20 categories
- the estimated arrival date has been removed and replaced by a time difference, in days, between the estimated and actual arrival date
- from the actual arrival date, we extracted the month, the day of the month and the day of the week
- the final destinations that are present less than 2 times in the data set have been grouped in a ”isRare” category

Some containers might come back several time during a given period. Therefore, among the train set, a label was created to state whether a container had already been seen during the year of observation or not. Indeed, the average dwell time of containers already seen was around 10% smaller.

Finally, some outliers were noticed essentially regarding the time difference between the estimated and actual arrival time. Some containers arrived 330 days earlier than expected given those data. Given the small volume of observations concerned, it was safe to remove them. Finally, the data set is passed through a One-Hot Encoder in order to deal with the categorical features created.

2.3 Distribution

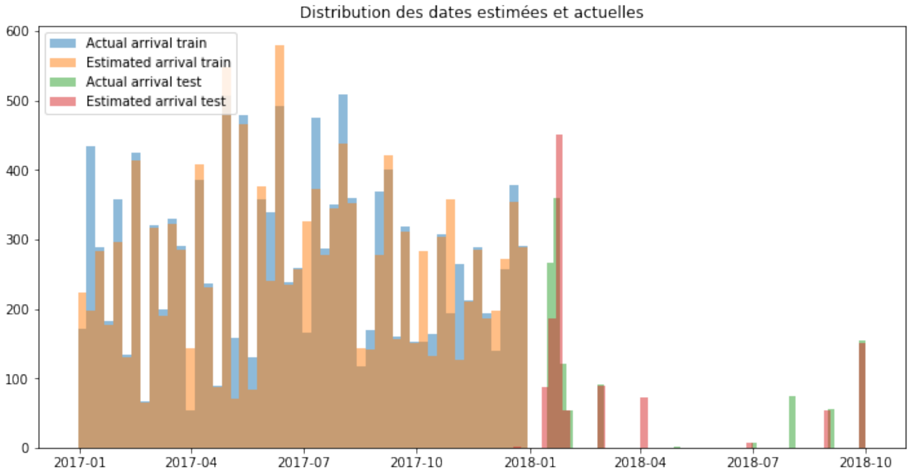


Fig. 1. Distribution of the estimated and actual times in the train and test set

The train set contains data collected in 2017. The test set contains data collected in 2018. The year of the estimated and actual dates can obviously not be used. We do expect the column of the months to be quite relevant since the test set is essentially collected on the first months of the year.

The weights of the containers follow a multimodal distribution. The split of 20 categories was randomly chosen. Other splits might have been applied and might produce better results.

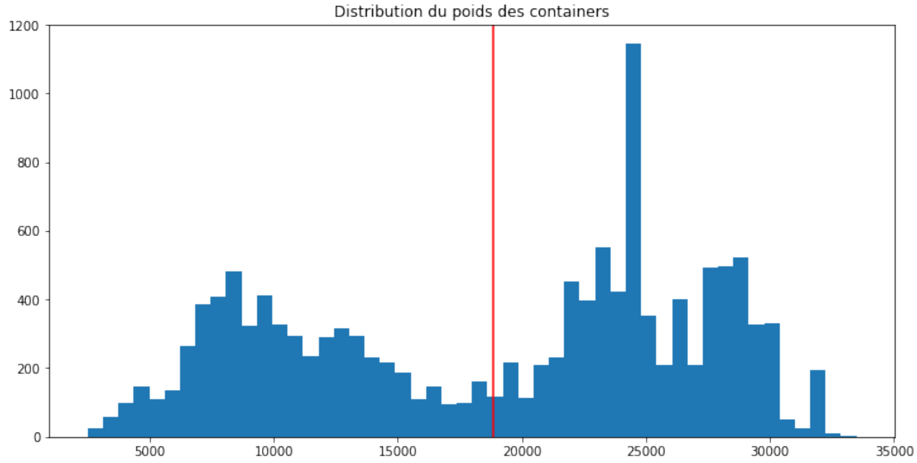


Fig. 2. Distribution of the weights of the containers

Most containers arrived between Tuesday and Thursday. The dwell time distribution has a lognormal shape. Some containers stay more than 3 months in the port. We would require an anomaly prediction algorithm in order to predict those subcases. We will focus on the most basic cases in which the dwell time is smaller than 50 days and delete the 16 outliers.

A feature importance can be established from the random forest regressor. The most importance features in the train set appear to be :

- the final destination
- the weight
- the reference of the ship's stopover
- the trucking company

The circle of correlation indicates that :

- the actual day and the dwell time are correlated and when projected on a 2D plan with a PCA, go along the same direction. The later we are in the month, the longer the dwell time

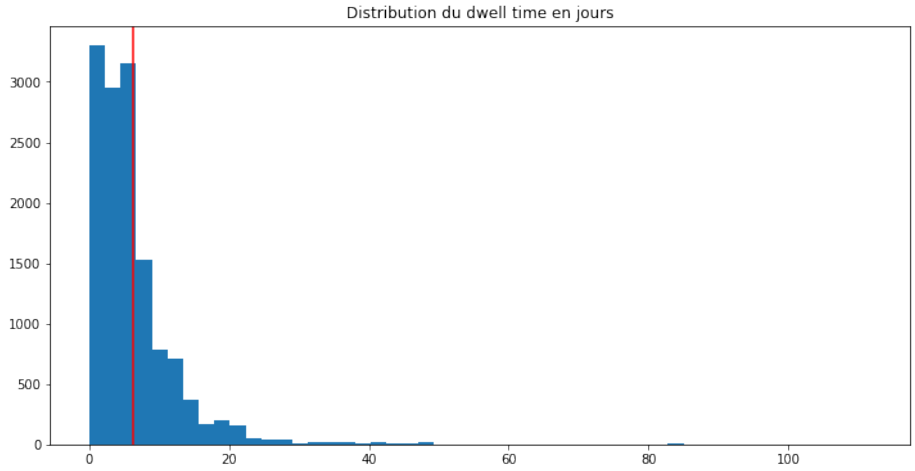


Fig. 3. Dwell time distribution in days

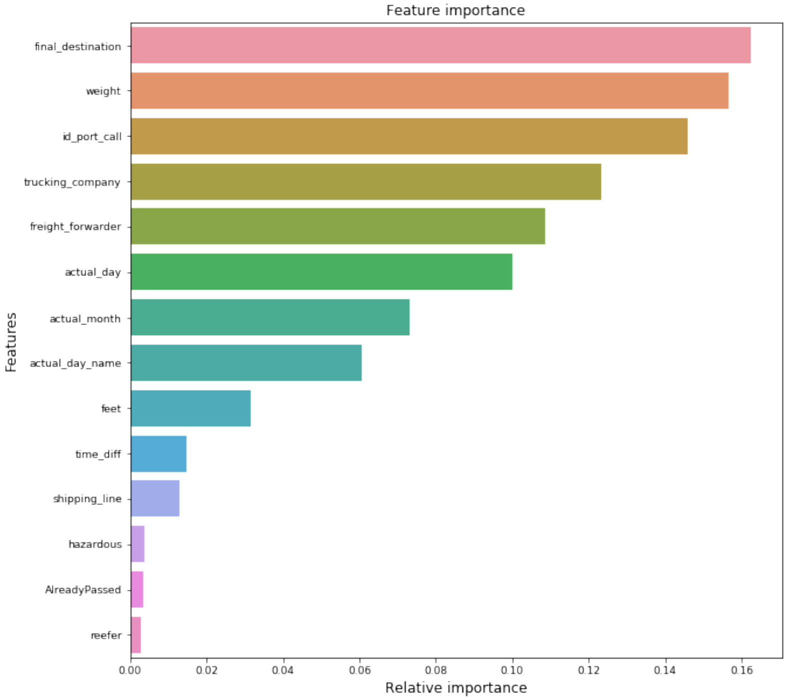


Fig. 4. Feature importance

- the actual day name is inversely proportional to the dwell time. The further in the week we are, the shorter the dwell time (since the containers should be gone before the week-end)
- the time difference between the estimation and the actual arrival time is also negatively correlated with the dwell time. A late container should be gone faster once it has arrived

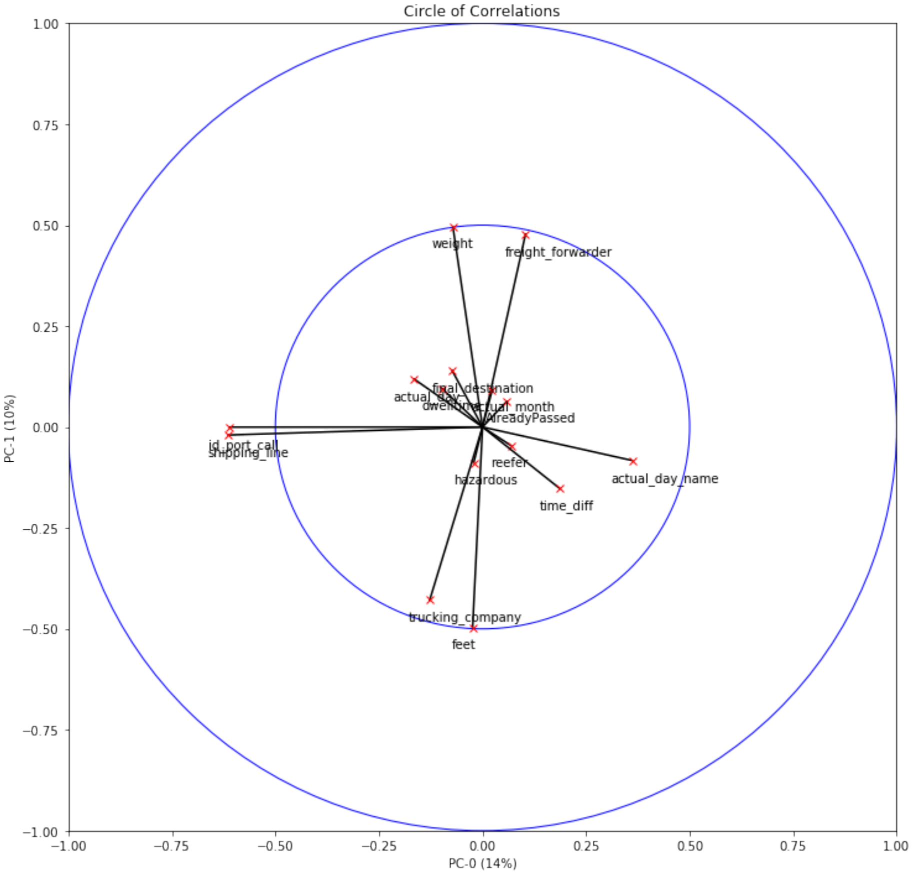


Fig. 5. Circle of correlation

2.4 Models and Results

The metric chosen is the Mean Absolute Pourcentage Error (MAPE).

$$MAPE = \frac{y_{pred} - y_{true}}{y_t rue} \tag{1}$$

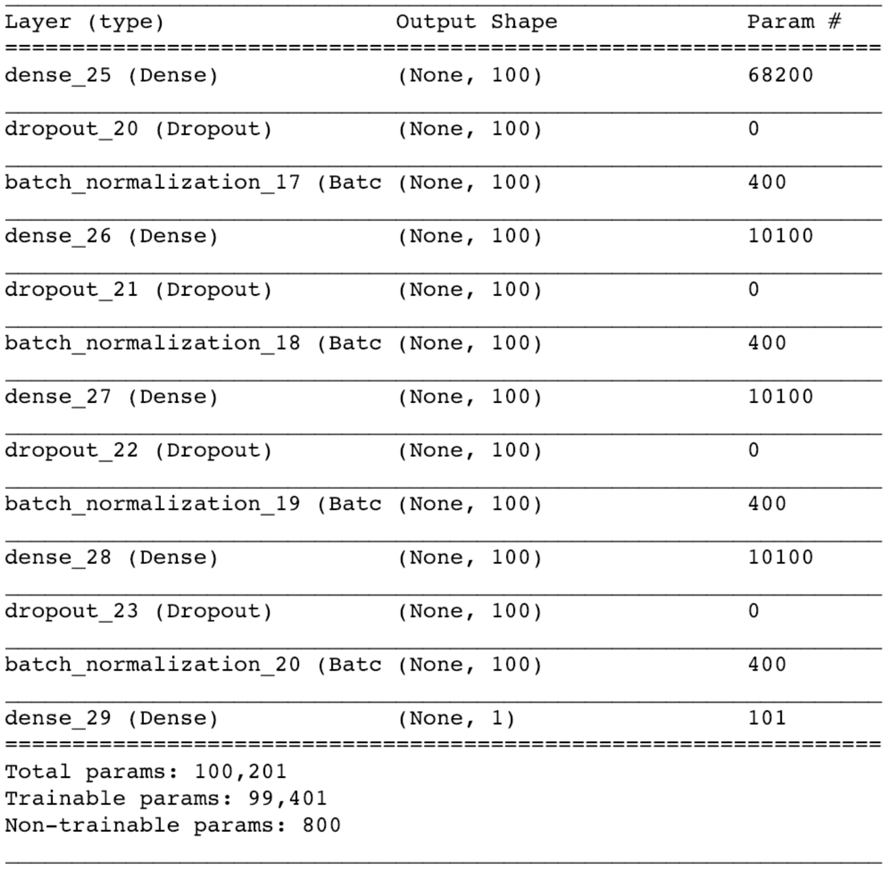
Different approaches have been tested on this problem :

- a K-Nearest Neighbors Regressor. The model performs poorly whether we perform first a dimension reduction with a MAPE of 140.
- a Random Forest Regressor which achieves a MAPE of 110
- a XGBoost Regressor which achieves a MAPE of 80
- an Artificial Neural Network (ANN) that acheives a MAPE of 69

Overall, the problem appears quite challenging due to the diversity of the data types and the limited amount of trainign examples. The best results have been achieved by the ANN due to the underlying complexity of the data. However, the train set only has 13'500 samples, which is a major restriction compared to the large amount of data needed by deep learning algorithms.

The model architecture was inspired by a recent paper by Ioanna Kourounioti : "Development of Models Predicting Dwell Time of Import Containers in Port Container Terminals – An Artificial Neural Networks Application". The model is made of 4 hidden layers. There is a large trend for overfitting in this data set. Therefore, several drop out and batch normalization layers are necessary.

There is room for a large amount of improvements on the structure proposed. There is a total of 99'401 parameters. There are some overfitting issues that could also be solved by a better deep learning model. Overall, the confidence interval at 95%, in hour, is [6.51 ; 53.54].



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_25 (Dense)             (None, 100)               68200     
_________________________________________________________________
dropout_20 (Dropout)         (None, 100)               0         
_________________________________________________________________
batch_normalization_17 (Batc (None, 100)               400       
_________________________________________________________________
dense_26 (Dense)             (None, 100)               10100     
_________________________________________________________________
dropout_21 (Dropout)         (None, 100)               0         
_________________________________________________________________
batch_normalization_18 (Batc (None, 100)               400       
_________________________________________________________________
dense_27 (Dense)             (None, 100)               10100     
_________________________________________________________________
dropout_22 (Dropout)         (None, 100)               0         
_________________________________________________________________
batch_normalization_19 (Batc (None, 100)               400       
_________________________________________________________________
dense_28 (Dense)             (None, 100)               10100     
_________________________________________________________________
dropout_23 (Dropout)         (None, 100)               0         
_________________________________________________________________
batch_normalization_20 (Batc (None, 100)               400       
_________________________________________________________________
dense_29 (Dense)             (None, 1)                 101       
=================================================================
Total params: 100,201
Trainable params: 99,401
Non-trainable params: 800
_________________________________________________________________
```

Fig. 6. Architecture of the ANN