Challenge Data

Real-time train crowding forecasting
by SNCF-Transilien

Introduction Public leaderboard Private leaderboard

Description

Competitive challenge | Environment | Industrial | Regression | Tabular

Less than 10MB | Basic level

Dates

Started on Jan. 6, 2023

Challenge context

SNCF-Transilien is the SNCF commuting network operator covering the Paris region. It runs more than 6,200 trains per day and moves 3.4 million of passengers. SNCF-Transilien gathers an increasing amount of data on the number of passengers on-board through infra-red door sensors. This increasing amount of data allows SNCF-Transilien to provide new services to passengers and improve its operational performance.

Challenge goals

The aim of this challenge is to **give SNCF-Transilien the tools to provide an accurate train occupancy rate forecasting**. Thus, deliver precise real time crowding information (RTCI) to its passengers through digital services.

Files given to participants

x_train
input data of the training set
y_train
output data of the training set
x_test
input data of the testing set
random_submission_examp
a random submission csv file exar

The challenge provider

Contact

For any questions regarding the c
provider at challengedata.transilie

Congratulation for the win

1 Franck Zibi
1 gsalle
1 Cyril Guerin

You can find the whole list of winr

When a traveler is waiting for a **k**-train at station **s**, the train may be several stations ahead, but we want to give the travelers the most consistent information possible about how busy the train will be when they board it. To do this, we need to **predict for each train the actual occupancy rates at the next stations**. Here, we propose to predict, more simply, the occupancy rate at the next station only.

Challenge data is supported by:

Data description

Dataset structure

k is used to specify the train, **s** for the station and **d** for the day. One stop is given by the triplet **k**, **s**, **d**.

- A Xtrain.csv dataset with 31 119 rows (i.e. 31 119 stops **k**, **s**, **d**) and 12 columns
- A Xtest.csv dataset with 13 752 rows (i.e. 13 752 stops **k**, **s**, **d**) and 12 columns
- A Ytrain.csv dataset with 31 119 rows and one column, giving the target variable to be predicted for the training set.
- A Ysample.csv dataset with 13 752 rows and one column with random values between [0, 1], which is a submission example that you can use to prepare your own submission

Data origin

The data comes from infra-red sensors located above each door of the rolling stocks (NAT, R2N) in Île-de-France, measuring the number of alighting and boarding passengers per door. This data is captured in real time and is accessible only at the train scale for this challenge. **The crowding factor or occupancy rate is the number of passenger on-board divided by the total capacity of the train**.

We want you to predict the occupancy rate in real time using the near past. We introduce p and q notations to describe the past: * p is used for the past values of the previous trains at the same station * q is used to describe previous stops for the same train.

The actual occupancy rate is: p0q0 $\in$ [0,1] which is crowding factor/occupancy rate, noted Y(k,s,d), for a stop which is a train **k** at a station **s** for a day **d**.

The columns, i.e., the features, are split into 6 contextual variables such as day, train id, etc. and 6 lag variables:

Contextual variables

- date **d**: YYYYMMDD is the date of the train id run
- train **k**: 1, 2, ..., 55 train id (unique by day **d**)
- station **s**: AA, AB, AC,... is the station id
- hour: HH:00:00 is the time slot
- way: 0, 1 is the train way with 0 (toward Paris) or 1 (toward suburb)
- composition: 1, 2 is the number of train units (one or two)

Lags variables

- p1q0 $\in$ [0,1] is the occupancy rate of the previous train **k-1** at the same station **s**
- p2q0 $\in$ [0,1] is the occupancy rate of the second previous train **k-2** at the same station **s**
- p3q0 $\in$ [0,1] is the occupancy rate of the third previous train **k-3** at the same station **s**
- p0q1 $\in$ [0,1] is the occupancy rate of the same train **k** at the previous station **s-1**
- p0q2 $\in$ [0,1] is the occupancy rate of the same train **k** at the second previous station **s-2**
- p0q3 $\in$ [0,1] is the occupancy rate of the same train **k** at the third previous

station **s-3**

The lag variables are NA if the lag does not exist for the stop because of timetable structure. All other missing values have been imputed by a business algorithm.

For more details on the data set or on the benchmark model, see our paper One-Station-Ahead Forecasting of Dwell Time, Arrival Delay and Passenger Flows on Trains Equipped with Automatic Passenger Counting (APC) Device.

Benchmark description

The benchmark used in this challenge is based on a multiple linear model. Each node is defined by the intersection of a train id and a station id. We use a simple linear model with at most one neighbor in p and q.

$$Y_{k,s} = \beta^{0}_{k,s} + \beta^{1,0}_{k,s} y_{k-1,s} + \beta^{0,1}_{k,s} y_{k,s-1} + \varepsilon_{k,s} \ ,$$

when a neighborhood exists. In R syntax, it is:

```
lm(p0q0 ~ p1q0 + p0q1, data = dataByNode)
```