Wikipedia Web Traffic Forecasting
Machine Learning on Big Data - HES-SO Master

Maxime Alexandre Lovino Marco Rodrigues Lopes

June 24, 2019

1 Introduction and project description

This project is inspired by a competition posted by Google on Kaggle[1] and is based around the idea of forecasting time series values. For this project, we drew inspirations by submissions from other competitors on Kaggle as well as the work of *JEddy92* on GitHub[2] which inspired our use of the WaveNet model for this project.

Time series are used in multiple domains, the most well known being the stock market values, so we could apply the same type of models to predict stock market changes and become rich....But in practice it would actually be more efficient to use external factors such as trending topics or news articles to find correlations with the changes in the time series.

In this project, we didn't have only one time series to predict but actually a lot of them. 145000 different time series were provided spanning 2.5 years with a granularity of 1 day. The goal of the project is to predict the values for each of them for 60 days in the future. For this project, we didn't use external data to help us with the predictions and relied solely on the time series themselves to predict future values.

In order to respect the maximum number of pages specified for this report, we will only discuss models, techniques and results from our best model(s) and will discuss discarded ones shortly during the oral presentation.

[1] `https://www.kaggle.com/c/web-traffic-time-series-forecasting/`
[2] `https://github.com/JEddy92/TimeSeries_Seq2Seq`

2 Data description

The dataset that can be downloaded on Kaggle consists of two training files in the CSV format and other files which are not relevant to our problem as they're only used for actually submitting an answer for the competition. The two training files consists of the same set of 145000 pages and the same starting date but the second one:`train_2.csv` is longer, spanning almost 803 days. We decided to use the longer for our experiment has it contains at least two complete years of data. The CSV file weighs 400 MB uncompressed.

The structure of the file consists of a first column containing the page information concatenated in a string, with the title of the page, the lang, the access type and the agent and then a column for each day with the number of views that the specific page got on that day. Some page have `NaN` values for view on certain dates and this corresponds to the page not having been created yet on that date or just missing values.

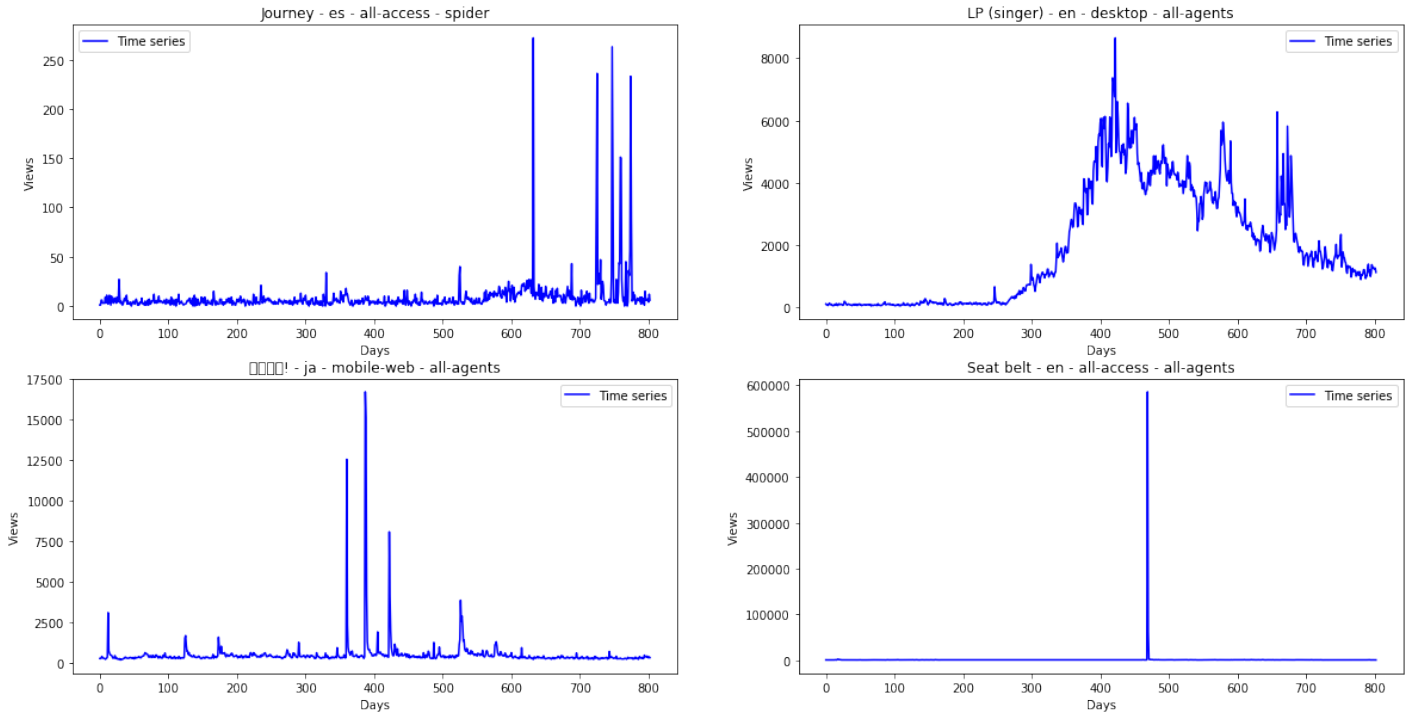

Figure 1: Some of the pages present in the dataset

3 Data cleaning and pre-processing

The preprocessing and cleaning of the CSV file is handled in the `preprocess_csv_to_pickle` static method of the `WikiSeries` classes we have created. This will take a CSV file as an input, extract the information contained in the first column to split in 4 columns (title, lang, access, agent) and remove any page which doesn't belong to a specific country Wikipedia subdomain[3]. We will also remove all pages containing `NaN` values in their series as we can't really replace them with 0 because they didn't actually get 0 views as they didn't exist so we removed them to only work with clean data. After all these steps, we save the resulting Dataframe as a Pickle which will then be used by our other programs. With this cleaning, we reduced our number of pages from 145'063 pages down to 106'328 pages.

[3]Wikimedia for example

4 Machine Learning techniques used

4.1 Walk forward validation

Before training our models, we had to decide how we wanted to split our data. Splitting by pages doesn't make a lot of sense for time series so instead we decided to use Walk Forward validation so that we evaluate the same pages but for a different 60-days windows when evaluating our models. When specifically fitting the model though, we took 40'000 pages and dedicated 20% for Keras validation so we used a side by side split to fit the model.

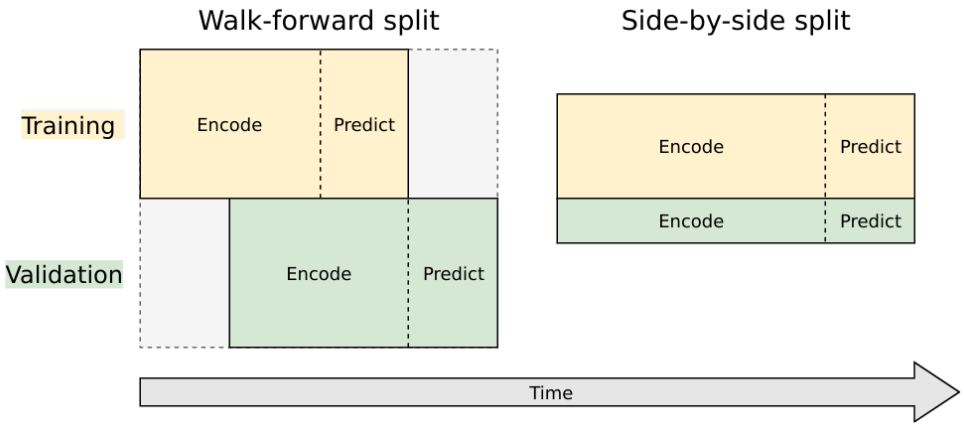

Figure 2: Walk forward validation

4.2 Seq2Seq models

4.3 WaveNet

5 Experiments and results

5.1 Training in the cloud

We had to use cloud infrastructures, such as Google Cloud and Amazon AWS in order to train our models on big GPUs in a reasonable time. We mainly used Tesla K80 cards on Google Cloud as well as some Tesla V100 cards from Nvidia. Our more complex model, the full WaveNet with the most layers, skips and residuals took 2 minutes per epoch to train.

Evaluating prediction for all 100'000 time series took a long time as well, we did it in batches but it still took more or less 20 minutes on those same GPUs for the most complex model.

5.2 Evaluating results with SMAPE

The metric we used to evaluate our results was the SMAPE[4] which was used in the Kaggle competition. The SMAPE provides a relative error so that errors in pages with a high number of views won't influence our results more than errors on small pages. The formula used to compute the SMAPE is the following[5]:

$$\frac{100\%}{days} * \sum_{t=0}^{t=days} \frac{2 * |\hat{y}_t - y_t|}{|y_t| + |\hat{y}_t|}$$

5.3 Results for single day predictions

We evaluated at first for single day prediction, that means that for each day t, we inserted the target y_{t-1} as last step for the prediction. With this we obtained SMAPE values of 24.885% for the simple WaveNet and 23.708% for the full one.

5.4 Results for 60 days prediction

Then we evaluated for the full 60 days, that means that for each day t, we inserted the prediction $\hat{y}_{t-1}$ as last step for the prediction. With this we obtained SMAPE values of 37.2910% for the simple WaveNet and 34.6499% for the full one. This last result is impressive as it is better than the result from the winner of the competition. However, the results were not computed on the same period as the period used for computing results has never been published and also our result has only been computed on pages containing non `NaN` values.

[4] Symmetric Mean Absolute Percentage Error

[5] This gives directly the percentage value, so a result of 35 means a 35% SMAPE value

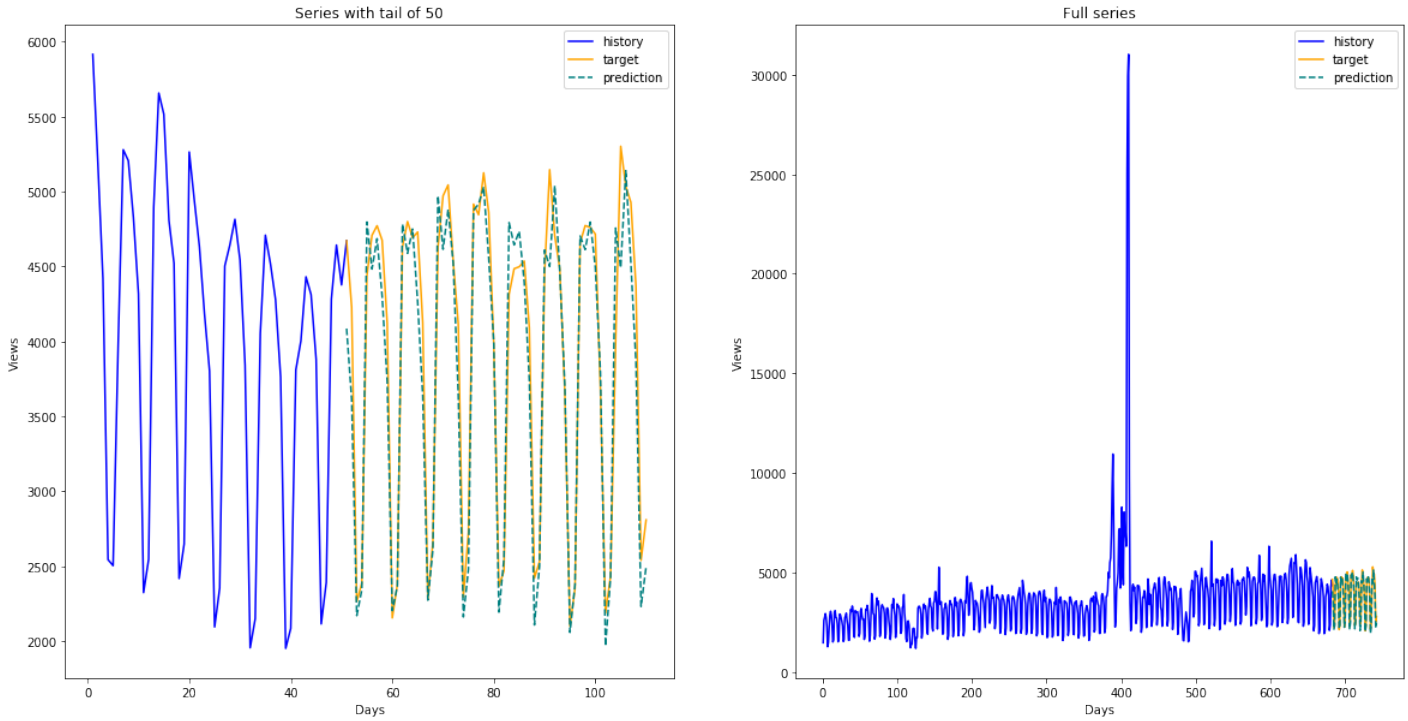

Figure 3: Single day prediction for Wikipedia EN Main page

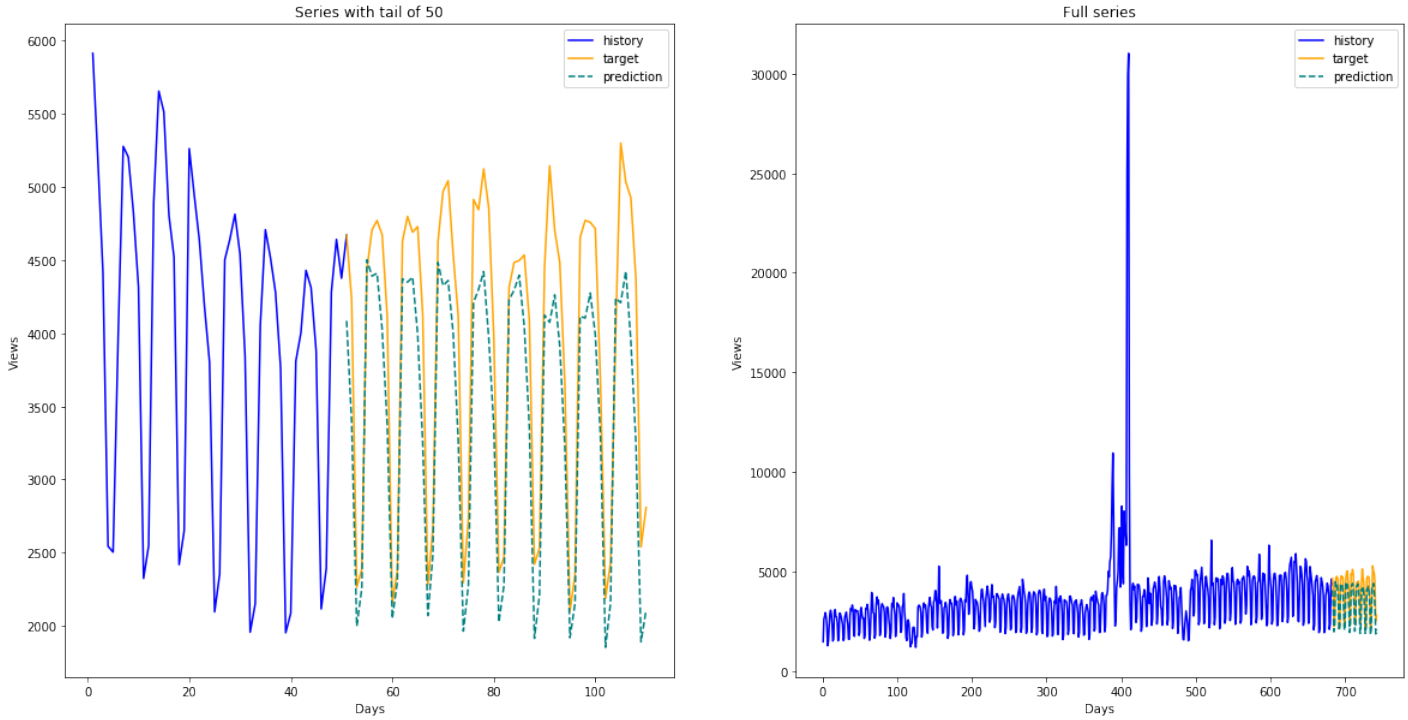

Figure 4: 60 days prediction for Wikipedia EN Main page

6 Analysis and conclusions