LONG-TERM VISITS PREDICTION OF 140K WEB PAGES

- A Time Series Analysis

by Ying Shen
Contact: yingsh@umich.edu

Problem statement

Why Time Series

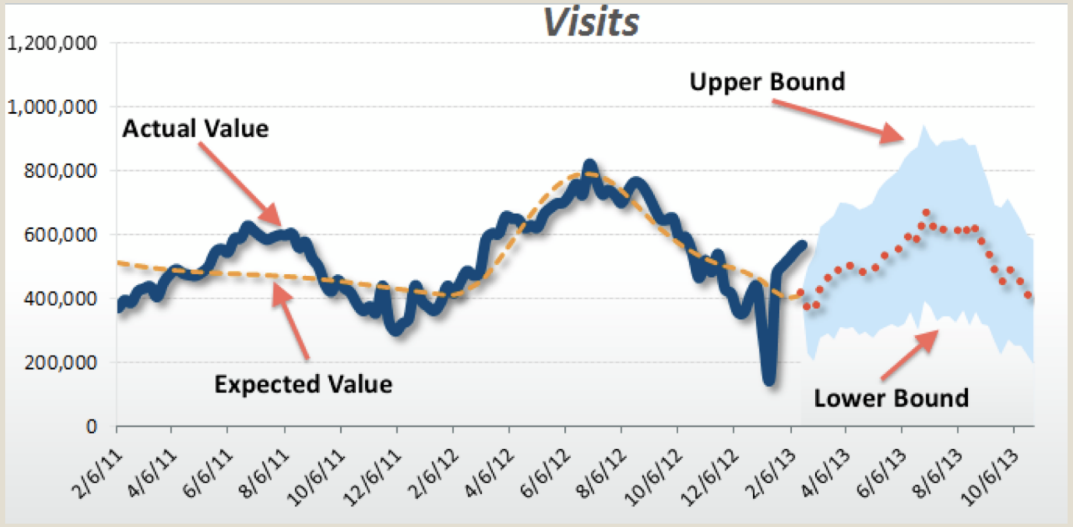
- Accurate time series forecasting is critical for business operations for optimal resource allocation, budget planning, anomaly detection and understanding stock market trends.

Why Web Traffic

- Hardware and network bandwidth need to be provisioned if a site is growing.
- Any revenue-generating site needs to predict its revenue.
- Sites that sell advertising space need to estimate how many page views will be available before they can commit to a contract from an advertising agency.

Who might care?

- The digital marketing teams of the web sites, the site operation staff, and search engine companies would all be interested in what the page visits going to be in the future, say, next week or month. The forecasting capability allows them to manage the resources beforehand and get alerts when there is something abnormal going on.



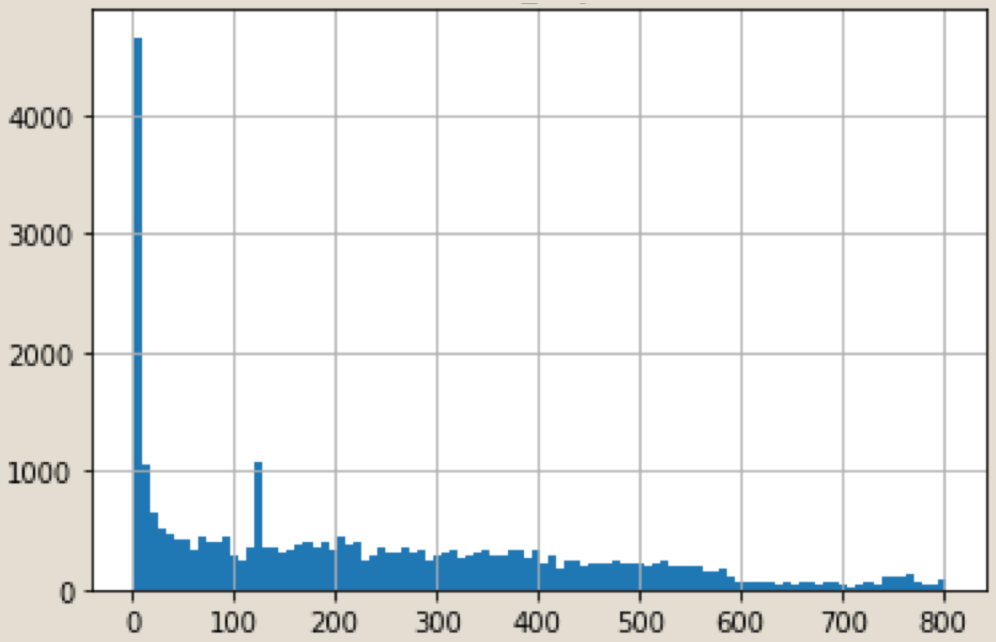
Data Source and Description

- The dataset originally comes from a Kaggle competition sponsored by Google in 2017.
- The goal of this project will be to forecast the future web traffic for approximately 145,000 Wikipedia articles during July 11, 2017 to September 10, 2017. And the training data would be generated from the daily page visits for the web article from July 01, 2015 to July 10, 2017.
- The original csv file is structured with 145,063 rows and 804 columns. Each row is the records of one Wikipedia page and each column represents the page visit for one day, which means that there are 803 days of data for these pages.

1. Missing Values

The first step is to check how many Wikipedia mages have missing values and what is the distribution of missing days.

Figure 1. Distribution of Null Days



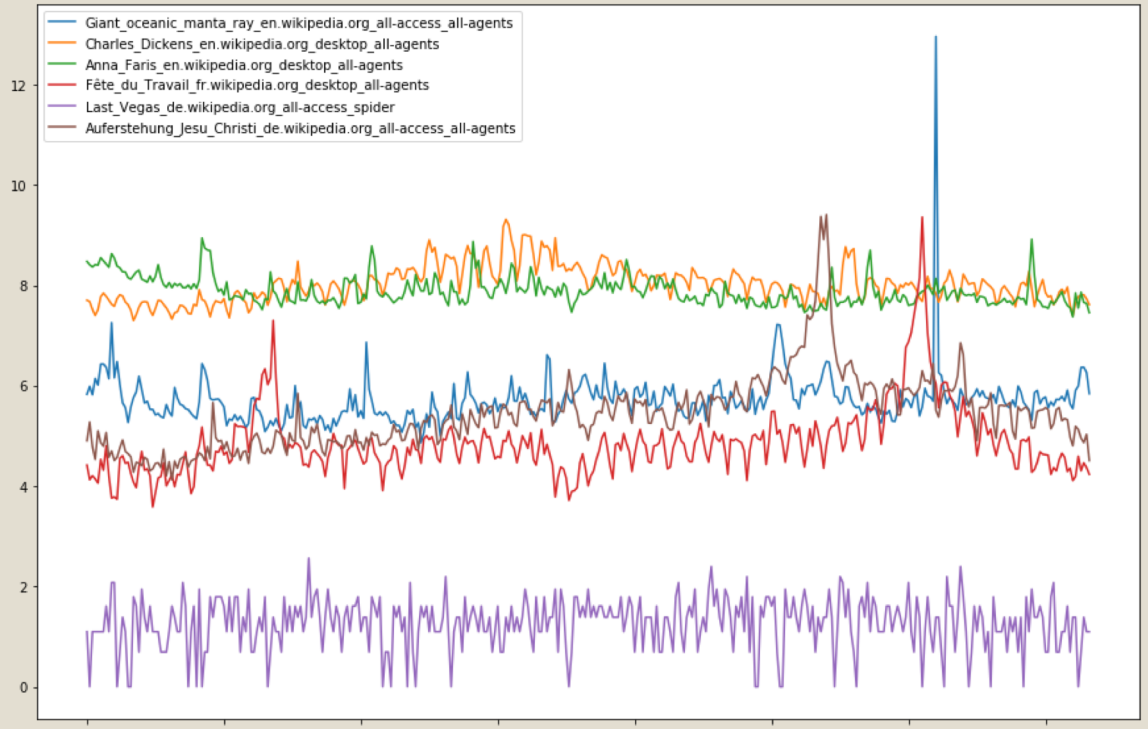
Exploratory Data Analysis

1. Missing Values

2. Outliers

There are large variations between each page. Some page (e.g., the blue one) has spikes and some page (e.g., the purple one) has low volume and quite constant trend.

Figure 2. Random Page Visits



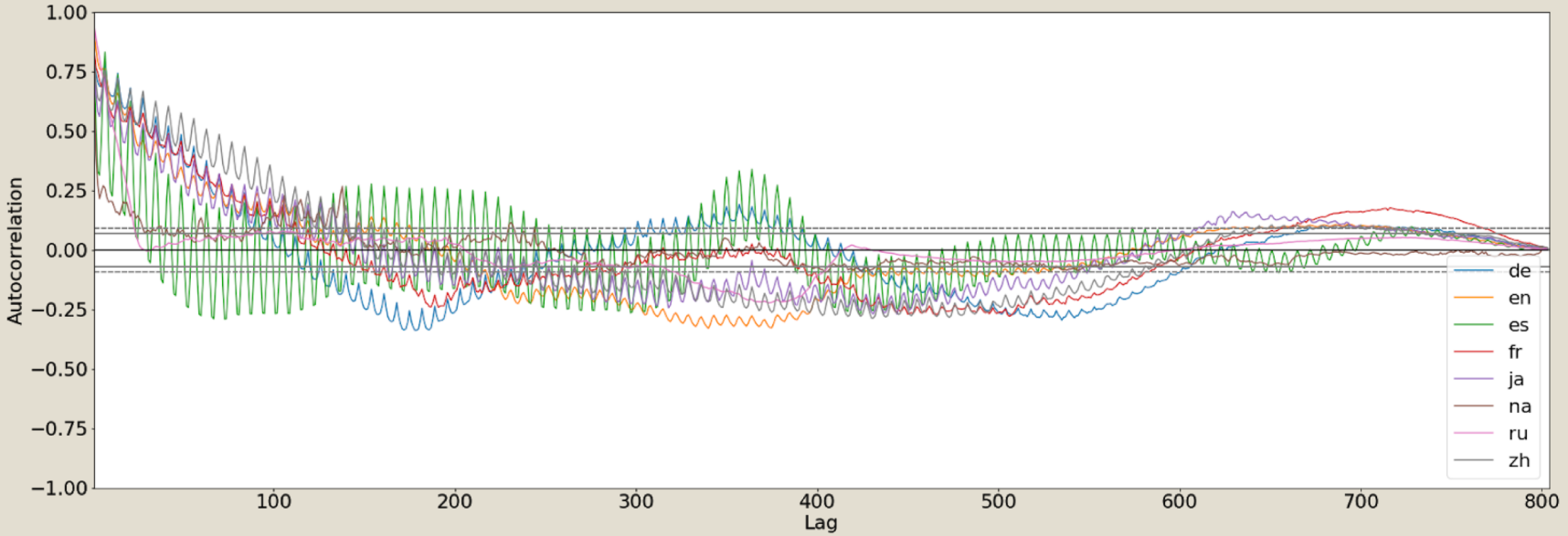
Exploratory Data Analysis

1. Missing Values
2. Outliers

3. Autocorrelations

It is interesting to know if there is a general trend associated with languages. Especially, for time series data, autocorrelation and seasonality is of great interest for the prediction task; therefore, I directly applied the autocorrelation plots for the mean daily visit stratified by language.

Figure 3. Autocorrelation Stratified by Language



Exploratory Data Analysis

1. Missing Values
2. Outliers
3. Autocorrelations

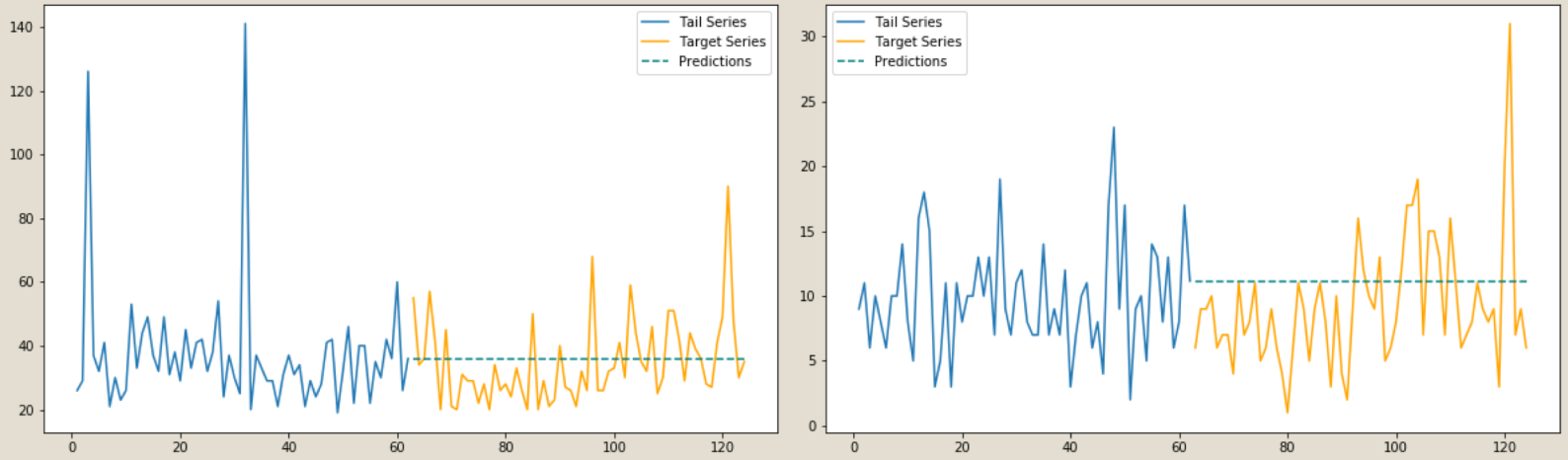
Challenge

- **Large amount of data** - 140K+ time series
- **Multiple outputs** - the goal is to predict the future 2 months visits for each page.
- Traditional statistical models like **ARIMA** ? – **not practical** for this kind of task since
  - each page would have difference in the parameters of ARIMA model (number of lags for autoregression model, number of differences, and number of lags for moving average model);
  - it would be hard to test if the (weak) stationarity assumption is met for the time series of each page.

1. Baseline Model

It is always nice to try a simple baseline model first and build/compare fancier models based on the baseline model. Before global features are engineered as features, there are 803 * 145,063 = 116,485,589 (**116 million**) cells already…

Figure 4. Predictions of Future 2 Months using Baseline Model
(Page ID: 227 [Left], 3374 [Right])



Machine Learning Models

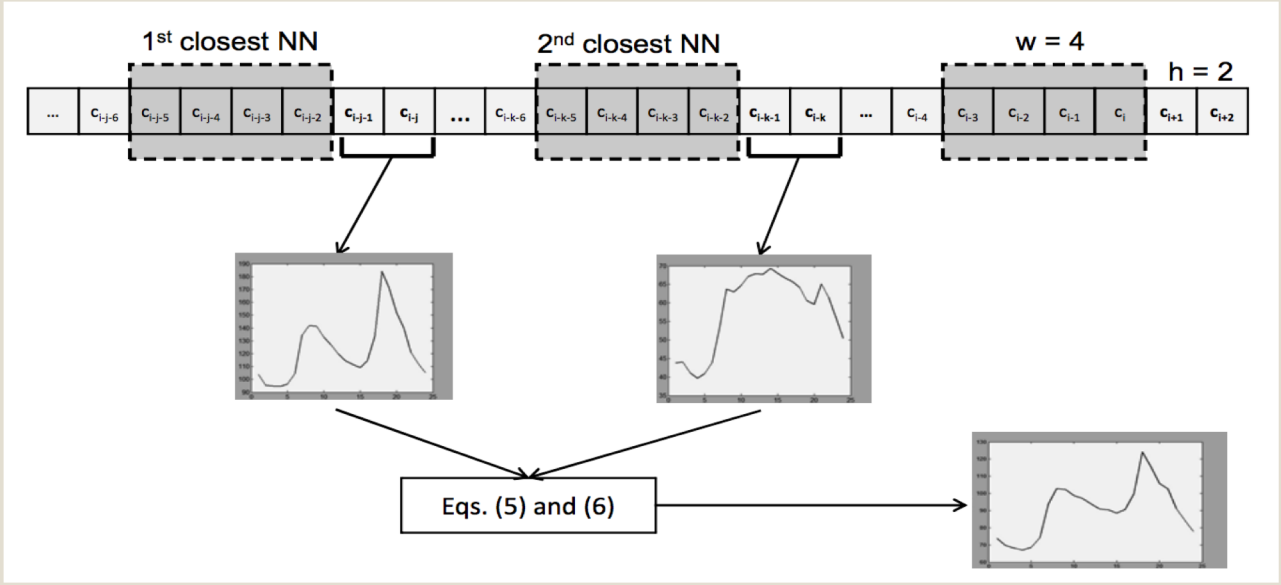
1. Baseline Model

2. K Nearest Neighbors Approach

Implement the approach suggested in the paper for time series data: http://eps.upo.es/martinez/papers/HAIS16.pdf

The basic idea is to find KNN that have similar "shape" of past days with those of the target, and use their medians as the prediction

Figure 5. Illustration of the KNN Approach



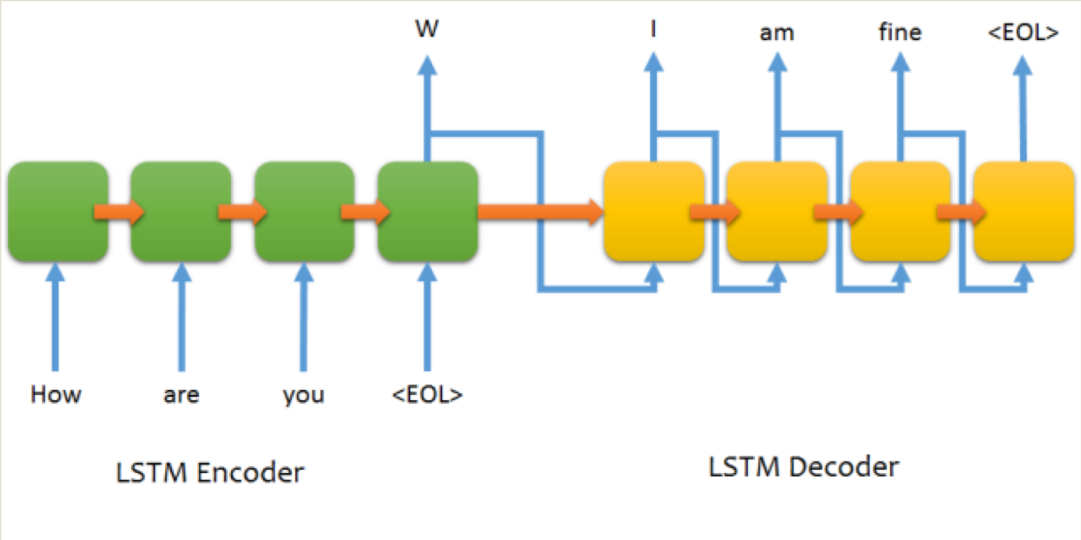
Machine Learning Models

1. Baseline Model
2. KNN

3. Sequence-to-sequence

Seq2seq has the capability to *map arbitrary-length sequences to other arbitrary-length sequences using fixed-size architectures,* which is very fit for this project.

Figure 6. Illustration of Seq2Seq Architecture



Machine Learning Models

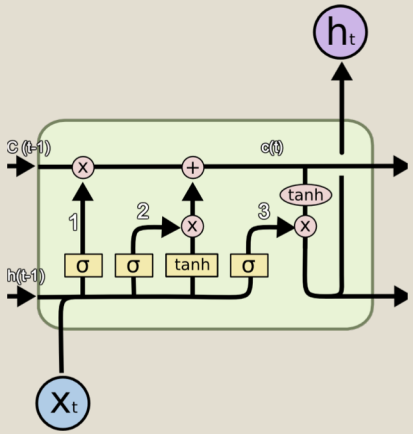
1. Baseline Model
2. KNN
3. Seq2seq

4. Seq2seq with Long Short-term Memory (LSTM)

Within seq2seq, two layers of convolutional (CNN) or recurrent neural network (RNN) - long short-term memory (LSTM) or gated recurrent unit (GRU) - are applied as the “encoder” and “decoder”.

The encoder layer deals with input data (past visits) and the decoder layer generates the final output (future visits).

Given the complexity of LSTM and the large amount of data to train, AWS EC2 instance was applied to obtain the computation power.



aws

amazon web services™ | EC2

Machine Learning Models

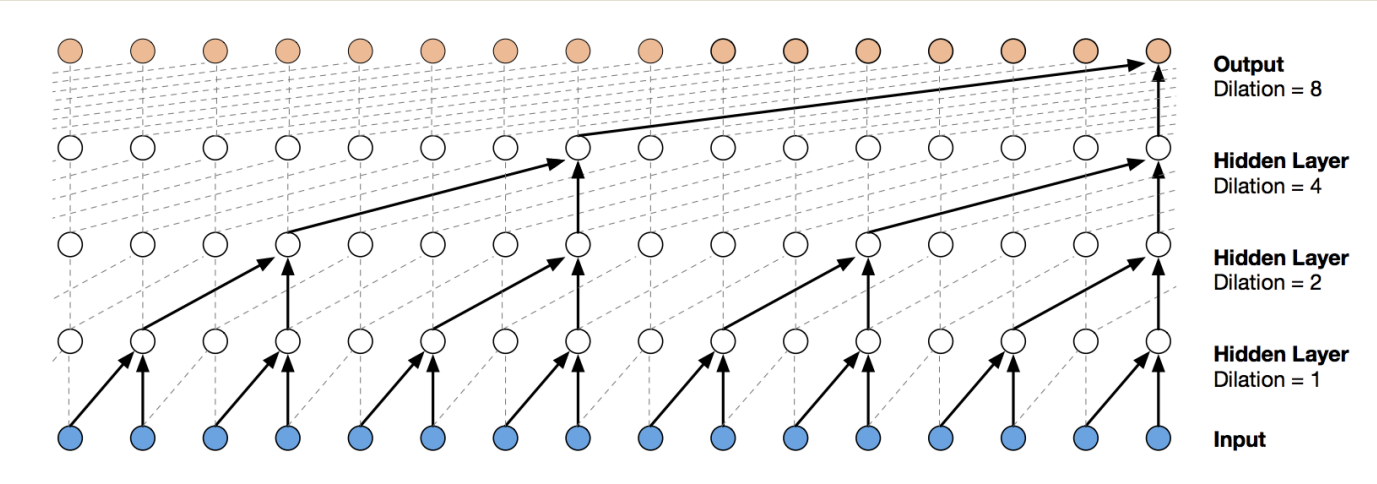
1. Baseline Model
2. KNN
3. Seq2seq
4. Seq2seq with LSTM

5. Seq2seq with Convolutional Neural Networks (CNN)

Thanks to this blog post, dilated (causal) convolutions was quickly implemented to this project.

It took less time to converge and used lower computation power.

Figure 7. Visualization of a Stack of Dilated Causal Convolutional Layers



Machine Learning Models

1. Baseline Model
2. KNN
3. Seq2seq
4. Seq2seq with LSTM
5. Seq2seq with CNN

Evaluation

- **Metrics selection**
  - Since the data is originally a Kaggle competition so I just used the evaluation score – SMAPE - that the competition used so that I can compare my result to the top scores.
  - Symmetric Mean Absolute Percent Error (SMAPE) is an alternative to Mean Absolute Percent Error (MAPE) when there are zero or near-zero demand for items.
  - SMAPE is the forecast minus actuals divided by the sum of forecasts and actuals as expressed in this formula:

$$\frac{2}{N}\sum_{k=1}^{N}\frac{|F_k - A_k|}{F_k + A_k}$$

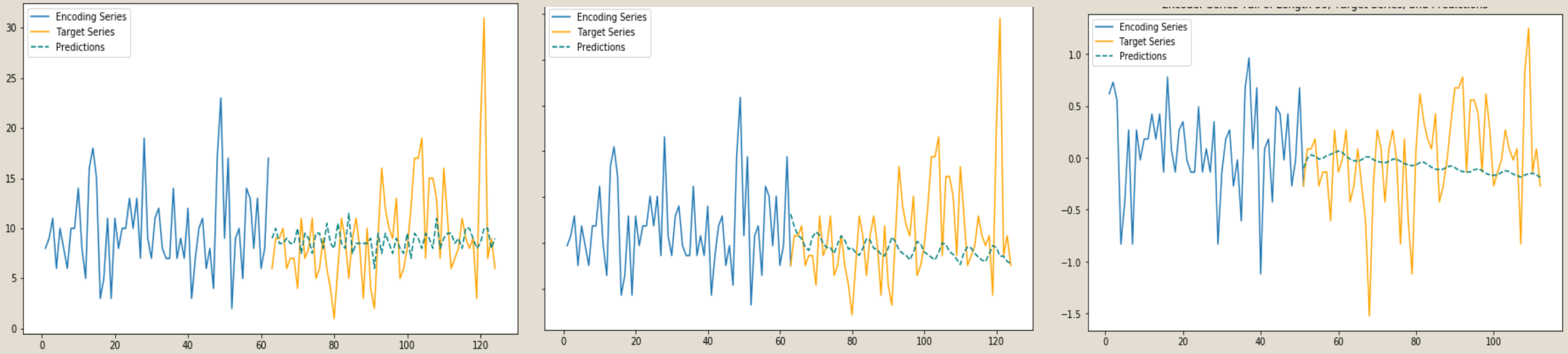
Evaluation

Table 1. SMAPE Scores for Different Models and Benchmarks

Models	SMAPE Scores
Baseline Model with Median	41.11
KNN	44.97
Seq2seq with LSTM	47.57
Seq2seq with CNN	45.24
Benchmark#1(Seq2seq CNN)	42.37
Benchmark#2 (Simple LSTM)	143.12

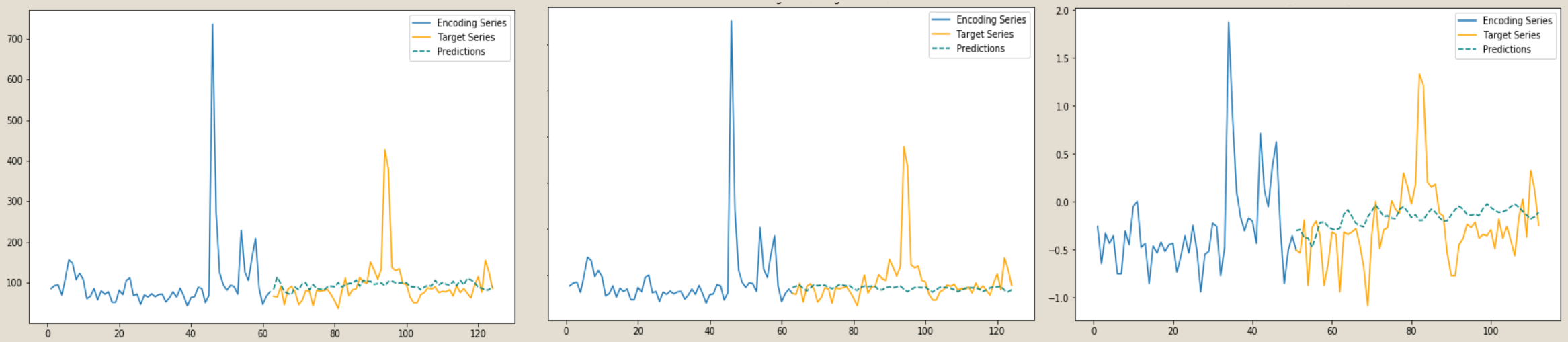
Evaluation - Visualization

Figure 8a. Predictions of Future 2 Months for Page ID 3374
(Left – KNN; Center – CNN; Right - LSTM)



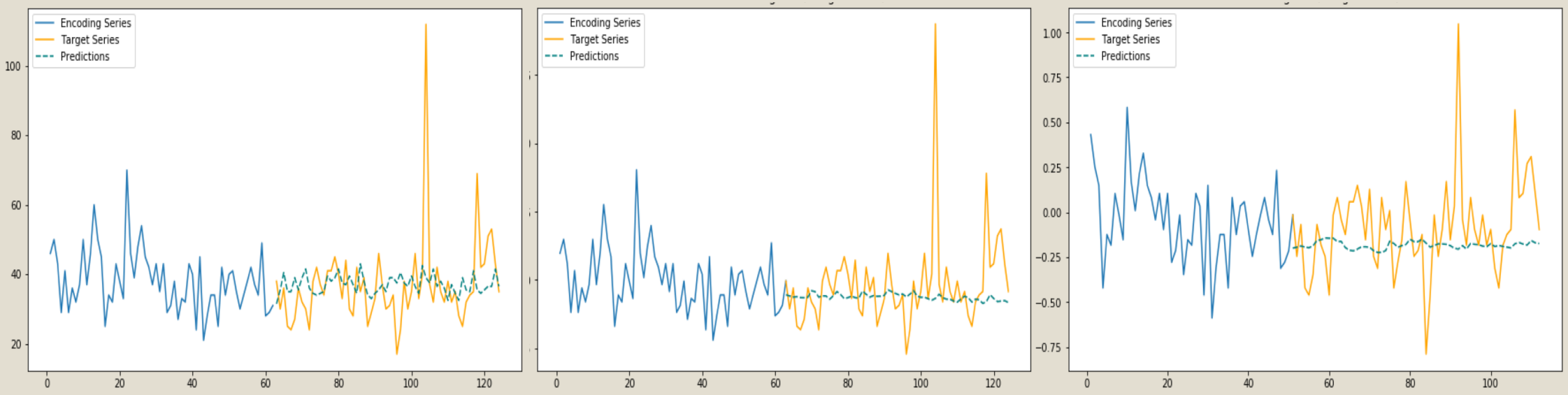
Evaluation - Visualization

Figure 8b. Predictions of Future 2 Months for Page ID 8135
(Left – KNN; Center – CNN; Right - LSTM)



Evaluation - Visualization

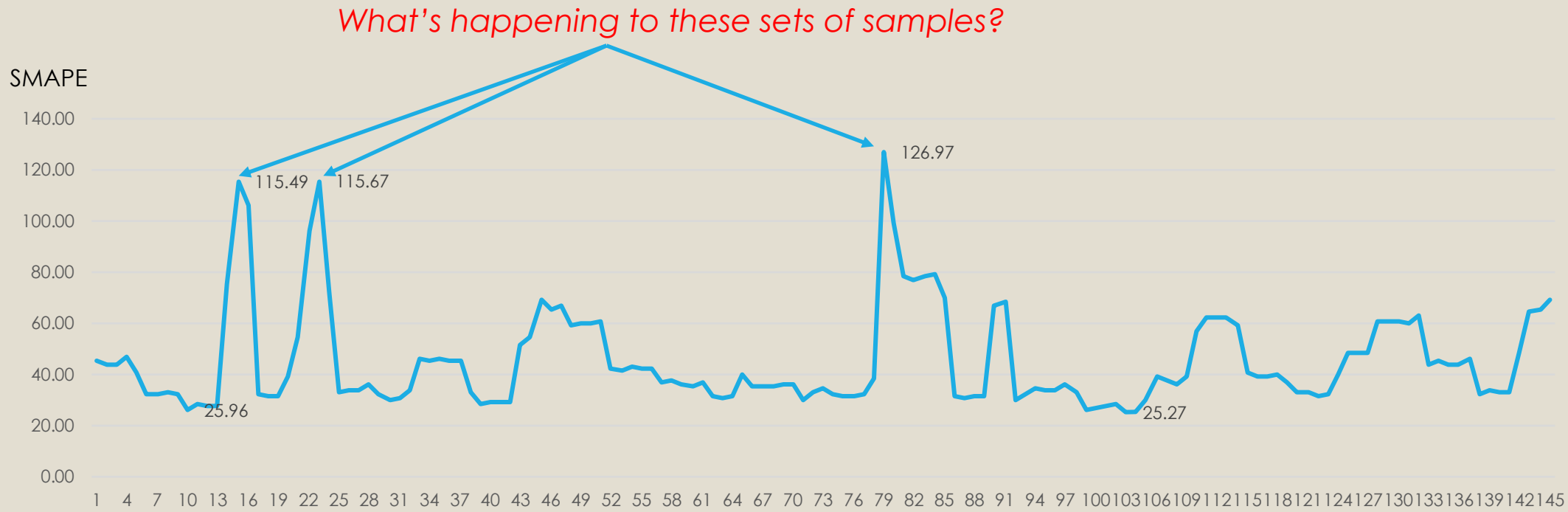
Figure 8c. Predictions of Future 2 Months for Page ID 109901
(Left – KNN; Center – CNN; Right - LSTM)



Conclusions

- **Model performance based on SMAPE** – It is very hard to beat the baseline model regarding SMAPE, but the median model was just repeating yesterday's value when the prediction went longer in time.
- **KNN model** was doing good with its flexibility to generate future predictions, but it is not scalable as the algorithm requires huge computation power as the data grows.
- Both of the **Seq2seq models** were generating future predictions with cycles, which is very interesting.
- **CNN model** cost less time and data (20,000 samples; 20 epochs) to train compared to LSTM model (100,000 samples; 100 epochs), with a better score. It seems to obtain a more predicting function from CNN.
- **Seq2seq with CNN can be utilized for long-term time series predictions with its flexibility to generate arbitrary length of predictions, computation-effectiveness, and promising precision.**

Future Work



- There are large variations regarding performance scores among different pages; further *parameter tuning* or *feature engineering* could be done to improve the Seq2seq with CNN model.