Machine Learning Engineer Nanodegree

Capstone Proposal

Akhilesh Jain
December 19th, 2018

Forecasting time series web traffic for wiki articles

Domain Background

Companies such as Google are interested in having models which can predict web traffic to certain websites potentially for better resource distribution and ad placements. This project is based on a competition floated by Google on Kaggle.com about a year ago. The objective is to explore the web traffic project and create a time-series model which can predict web traffic for hundreds of wiki articles.

Forecasting future events is a valuable many institutions such as stock markets, online sellers and advertisers. Forecasting web traffic is one of the most challenging problems the data has not just strong temporal effects but also dependencies on current events which are difficult to capture. Many techniques such as exponential smoothing [1], ARIMA [2] have been developed in the past to analyze and predict time series values and it remains an active area of research [3] with interest in forecasting in various field.

Problem Statement

The goal of the project is to predict the web traffic on the Wikipedia based on previous views for hundreds of wiki articles. Exploratory analysis will be performed on the dataset to find relevant information. A dataset will also be split and merged to find total daily views for each language. There will be two types of predictions made:

1. Prediction for daily views for every page; and,
2. Prediction for daily views by language.

For forecasting, the dataset will be divided into a training set and testing set and validation set. The final prediction will be made for about a month (50 days). The quality of the predictions made by the model will be evaluated using Root Mean Square Error (RMSE).

Datasets and Inputs

Based on the information provided in the Kaggle information, the training dataset consists of approximately 1,45,000 time series. Each of these time series represent a number of daily views of a different Wikipedia article, starting from July, 1st, 2015 up until December 31st, 2016.

For each time series, there is a name of the article and the type of traffic that the time series represents (all, mobile, desktop, spider). The data source for this dataset does not distinguish between traffic values of zero and missing values. A missing value may mean the traffic was zero or that the data is not available for that day.

Train_1.csv - contains traffic data. This a csv file where each row corresponds to a particular article and each column correspond to a particular date. Some entries are missing data.

The format for page name is '<name>_<project>_<access>_<agent>' (e.g. 'AKB48_zh.wikipedia.org_all-access_spider').

<*name*> is the name of the wikipedia page; <*project*> is the Wikipedia project (e.g. en.wikipedia.org). Note that the project may contain the language of the Wikipedia page; <*access*> is the type of access (e.g. desktop) and <*agent*> is the type of agent (e.g. all, mobile, desktop, spider). In other words, each article name has the following format: 'name_project_access_agent' (e.g. 'AKB48_zh.wikipedia.org_all-access_spider').

processed.csv – contains data from train_1.csv with extra features created from the column “Page”.

Solution Statement

The model will aim to provide the following predictions:

1. Predictions for total daily view counts of Wikipedia pages for 30+ days by language with an acceptable RMSE (< 0.25) i.e. there will be a time series prediction for each language – e.g. English, Russian, etc.

2. Predictions for daily views of 2-3 wiki pages with an acceptable RMSE (< 0.25). These pages will be selected from the thousands of pages available in the dataset but will require to have large number of views.

Benchmark Model

A simple model for prediction can be using the views in last n days as the prediction for next n days. For e.g. if a page has had 5, 10, 20 views on one each of the last 3 days, predict that it will get 5, 10, 20 views in the next 3 days. This simple model will be used for benchmarking and the R2 Score from this "simple" model will be compared with the prediction from this project. This may be a crude model on the lines of today will be just like yesterday, but it is very effective in determining what value our model is bringing. If our model is an improvement on the "simple" model by only a small margin, our model does not provide as much value.

Evaluation Metrics

Root Mean Square Error (RMSE) is used for evaluating the predictions made by the model. The mean squared error (MSE) is the average of the squared prediction errors. Squaring the error values gives a positive number and also puts more weight on large errors. RMSE is the root of MSE which has the same unit as the observations.

Project Design

The following workflow is planned for the project:

1. Preprocessing: The first step will be to upload the data and clean it to replace missing values. The missing values will replaced by zeros because according to the data source this is the strategy that has already been applied to the data. We will also remove outliers and normalize the data since anomalies can have a huge impact on the predictions.
2. Feature creation: The data in its raw form is devoid of too many features, as it only contains the wiki page name and views on each day. However, there is detailed information e.g. language and access embedded in the wiki page name which can be used to create more features.
   For e.g. each "Name" column can be broken down into "page name"
3. The dataset will also be split and merged to create a new dataset with total views for each day by language
4. Exploratory data analysis to understand the dataset and its features. We also perform seasonal decomposition of the time series to determine if there is any seasonality. Depending on the results, we may choose single, double or triple (Holt-Winters') exponential smoothing. It is important to note that single exponential smoothing is not good at capturing trend or seasonality and double exponential smoothing is not

good at capturing seasonality. Therefore, if the data ha seasonality, triple exponential smoothing or Holt-Winters' will be the mode to choose.

5. Split the dataset into training and testing set.
6. Time series analysis technique called exponential smoothing (single, double or triple depending on the seasonal decomposition) to make predictions for both daily views by language and individual web pages and compare with the "simple" model discussed earlier. The three exponential smoothing methods can be tried with holtwinters.ExponentialSmoothing in statsmodels library. The parameters for the models are trend, damped, seasonal, seasonal periods.

   For single exponential smoothing: "Trend" and "seasonal" is set to **none.**

   For double exponential smoothing: "seasonal" is set to **none.**

   For triple exponential smoothing: "trend", "seasonal" and "seasonal_periods" need to be specified.

   Unfortunately, the statsmodel function doesn't allow much control over some hyper-parameters of the mode. However, for refinement, the parameter "seasonal" will be set to "add" and "mul". The "damped" parameter will be set to True and False to further improve on the model.

   If exponential smoothing does not provide the expected performance, other techniques such ARIMA, XGBoost and LightGBM will be utilized.

7. The last step will be to use the evaluation metric to determine the quality of model.

Once the process is complete, the technique with the best performance from the results will be picked.

References

[1] "The Holt-Winters Forecasting Procedure", C. Chatfield (1978)
https://www.jstor.org/stable/2347162

[2] "Hypothesis Testing in Time Series Analysis", Almquist and Wicksell. Whittle, P. (1963). Prediction and Regulation.

[3] "A review of unsupervised feature learning and deep learning for time-series modeling", Martin Längkvist, Lars Karlsson, Amy Loutfi (2014)
https://www.sciencedirect.com/science/article/abs/pii/S0167865514000221