Wiki Website Traffic Prediction and Exploration

Group# 1

Divya Lingwal, 810975286, *dlingwal@kent.edu*
Karishma Patil, 810976028, *kpatil1@kent.edu*
Monalisa Singh, 810976309, *msingh11@kent.edu*
Shivani Mathur, 810975274, *smathur@kent.edu*

1. Introduction

Use of Internet is universal now, with more and more people using it in their daily life. Each website has large number of users, it is essential to deal with the problem of overload, so that we can provide quality of service to the potential users. In order to deal with this problem, we forecast future traffic volume. Forecasting is the process of estimating the future behaviour, with respect to the current and past data. We use the Wikipedia websites data from year 2015 to 2017 to implement the process of data analysis. To predict the future behaviour we found efficient and effective methods that are able to estimate the traffic volume

- **Motivation examples of this project**

Wikipedia is one of the important source used for professional and educational purpose. Wikipedia websites capacity planning is necessary to prevent the server from overload so, it is important to take this problem in account. One challenge of controlling the traffic is the prediction of the traffic.

- **Real applications**

In real world it is necessary to:

1. Focus on websites capacity.
2. For making important decisions towards the websites business.
3. Segmentation of articles on website depending upon the quality, popularity and importance of articles.
4. Effective and applicable to bursty traffic, also useful for Web server overload prevention.

2. Project Description

- **Brief descriptions of your project**

Our aim is to get better prediction for future behaviour of the views for www.wikipedia.com , to achieve this we are doing the process of forecasting by using the time series approach.

Short term prediction of wikipedia volume can be used to improve Quality of Service (QoS) as well as for congestion control.

Variety of factors affect the time series data so we are decomposing the views by using the time series components like tread and seasonality.

We first understood the variables provided in the dataset.

Data cleaning: Then we found out the missing values (na values) and the percentage of missing values in the dataset.

Data transformation: We observed that the links of the article consists of different url names like 'wikimedia', 'mediawiki' and 'wikipedia', so we separated these url names in different columns. Using ggplot we plotted the time series data and smoothing technique to reduce irregularities in wikipedia time series data, this gives us a clear and effective view of the behaviour of the series. We are doing prediction for 90 days.
Model used for forecasting: ARIMA (Autoregressive Integrated Moving Average)

- **Challenges and technical contributions** (new problems or new solutions?) in your project
    1. Determining the exact required library for the required functions was challenging.
    2. Due to large dataset, processing of the code took time.
    3. It was computationally expensive, so we took only a sample of data to do forecasting.
    4. For better prediction we had to choose small time frame as it was not visually clear to understand the prediction in large time frames.

- **The workload distribution for each member in your team**
  First we distributed the workload, but later on while integrating the initial part we faced some problems so, we all worked together and everyone contributed equally towards the completion of project successfully.

3. Background

- **Related papers (or surveys for graduate teams)**

    1. Yasseri, T. & Bright, J. EPJ Data Sci. (2016). Wikipedia traffic data and electoral prediction: towards theoretically informed models. https://doi.org/10.1140/epjds/s13688-016-0083-3

    2. P Ij Van Hinsbergen, C & Lint, J.W.C. & M Sanders, F. (2007). Short Term Traffic Prediction Models. 14th World Congress on Intelligent Transport Systems, ITS 2007.

    3. Li, Jia & W. Moore, Andrew. (2008). Forecasting Web Page Views: Methods and Observations. Journal of Machine Learning Research. 9. 2217-2250.

    4. Shu, Yantai & Yu, Minfang & YANG, Oliver & Liu, Jiakun & Feng, Huifang. (2003). Wireless Traffic Modeling and Prediction Using Seasonal ARIMA Models. IEICE Transactions on Communications. E88B. 10.1093/ietcom/e88-b.10.3992.

- **Software tools**:
  IDE- Rstudio
  DBMS- Not required
  Libraries used- ggplot2, ggthemes, scales, grid, gridExtra, corrplot, dplyr, tidyr, data.table, broom, lazyeval, forcats and forecast.

- **Required hardware**:
    1. Any configuration with more than 4GB.

2. For faster processing of data, GPU is prefered because dataset size is large.

- **Related programming skills**:
    1. Data analysis using R Programming Language
    2. Basic understanding of Rstudio for using this code.
    3. Should be aware of time series forecasting model i.e. ARIMA for understanding of this project.

4. Problem Definition

- This project generates a forecast of Views on wiki Pages in next three months and then help us see the performance of our model using various metrics like AIC, BIC, MAE, etc.

- **Challenges of tackling the problems:** Few challenges which we faced while solving our problem were:

1. Data cleaning and transformation was a big challenge because we had to figure out a particular format which would be helpful in being able to generate time series object and visualize it .
2. Understanding different patterns and what they signify was also something which was a challenge in beginning of project but eventually helped us to learn.
3. Rstudio sometimes run the code successfully while shows the error the other time even with the same program.
4. The big size of data was a challenge and we had to do sampling in order to be able to run code efficiently.

- **A brief summary of general solutions in your project**: We have used Auto Arima Forecast method. Initially we identified samples having extreme parameters like the ones having high Standard Deviation. Also, we found the articles which are very popular. The forecasting method provided by us will perform robustly for a wide range of different time series variabilities and shapes. We have used hold-out sampling method which is very robust. Our group has decided to take 90 days as prediction length. We took a shorter prediction duration, otherwise we will most likely get a lower performance. We have used ARIMA model for forecasting. We turned view counts into a time series object (using TS function). Also, our function took care of cleaning and outlier rejection using the tsclean tool.
  We have shown the articles which were very popular by looking at their average views. Slope attribute helped us to determine which articles are gaining and which ones are losing popularity.

5. The Proposed Techniques

1. **Framework**: The dataset in our project has around 145k time series. All of these time series data are numbers that basically describe a number of daily views of various Wikipedia articles that start from July, 1st, 2015 until December 31st, 2016.
   We have taken the last three months of this dataset as the test set and the rest for training. We have done this so that we can compare the predicted outcomes vs the actual outcomes to see how efficiently our model is performing.

In our training data, each time series has been provided with the name of the article as well as the type of traffic that this time series represent (all, mobile, desktop, spider. The dataset does not distinguish between traffic values of zero and missing values. This results in the conclusion that a missing value may mean the traffic was zero or that the data is not available for that day.

2. **Data Cleaning & Transformation**: The first step towards building a model was to first clean and transform the data. As part of the data cleaning, we saw that we had 0.07 missing values in our data as described above. We dealt with these missing values by imputing these with median. We did so because median imputation does not affect the data as a whole.

```
library(caret)
preprocessed <-preProcess(TrainData[,-1],method = "medianImpute")
Newdata<-predict(preprocessed,TrainData)

```

```
[1] 0.07747971
```

As a part of the transformation, we picked on separating training data into the following two parts: the article information (from the Page columns) alongside the time series data (Onlydates) from the date columns. We have done this splitting to make our initiative simpler.

```
#Extracting article information from page data and separating it into different columns to
make processing easier
trainprocess <- Newdata %>% select(Page) %>% rownames_to_column()
mediawiki <- trainprocess  %>% filter(str_detect(Page, "mediawiki"))
wikimedia <- trainprocess %>% filter(str_detect(Page, "wikimedia"))
wikipedia <- trainprocess %>% filter(str_detect(Page, "wikipedia")) %>%
  filter(!str_detect(Page, "wikimedia")) %>%
  filter(!str_detect(Page, "mediawiki"))
```

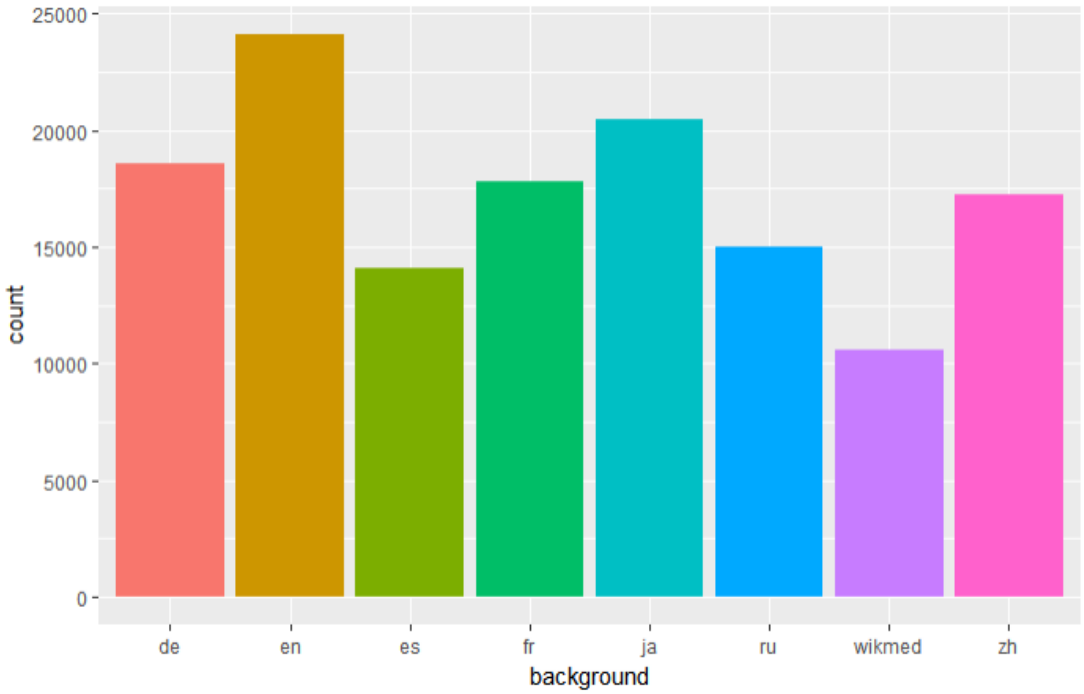
In the next step the article information was divided as per the data source whether it was from wikipedia, wikimedia, or mediawiki. We did this because all these pages were in different formats and it was necessary to bring them back together into a uniform format. After the formatting was done, we connected back all these pages using the 'join' command in R into a separate data frame called Onlypages. In the example below we are showing the top 9 rows of our Onlypages dataframe.

```
#Combining all above created datasets into one
Onlypages <- wikipedia %>%
  full_join(wikimedia, by = c("rowname", "article", "background", "access", "agents"))

head(Onlypages, n = 9)
```

	rowname <chr>	article <chr>	background <chr>	access <chr>	agents <chr>
1	1	2NE1	zh	all-access	spider
2	2	2PM	zh	all-access	spider
3	3	3C	zh	all-access	spider
4	4	4minute	zh	all-access	spider
5	5	52_Hz_I_Love_You	zh	all-access	spider
6	6	5566	zh	all-access	spider
7	7	91Days	zh	all-access	spider
8	8	A'N'D	zh	all-access	spider
9	9	AKB48	zh	all-access	spider

9 rows

Doing so gave us the ability to search for any particular page and view their information.



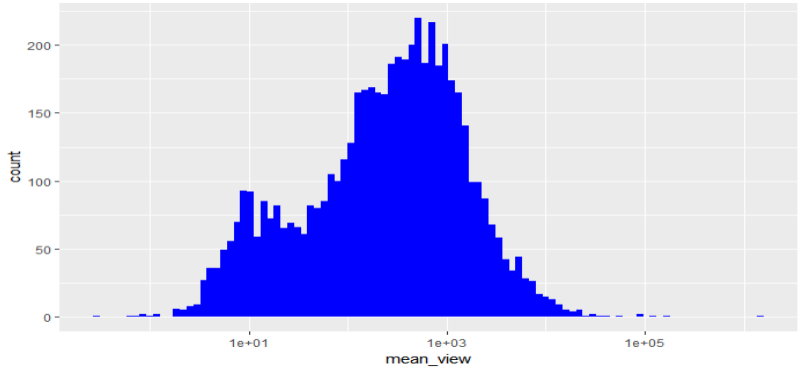
3. **Extracting Time Series**: Once the data was rearranged into the format that we needed, we wanted to start extracting some of the time series and plot them together. The function in R that we used in order to be able to do so helped us see the view of counts as integers in the data and included some other functions like changing the format of the data to year month & date, gathering the view counts along the dates, spreading the page names along the counts and so on.

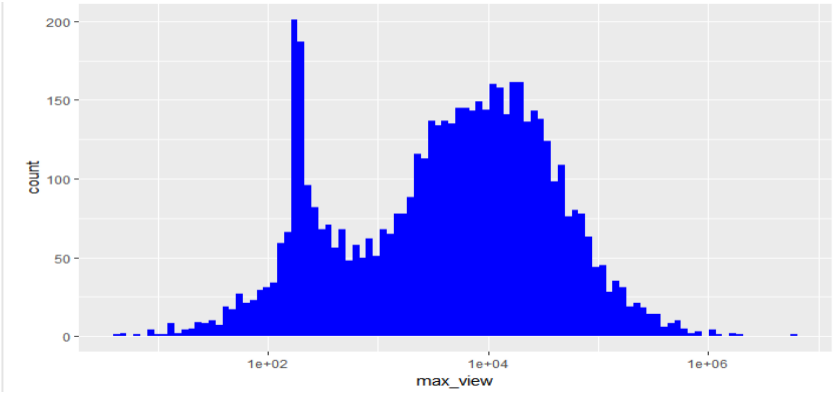
```r
TS1 <- function(rownr){
  Onlydates %>%  filter_((interp(~x == row_number(), .values = list(x = rownr)))) %>%
  rownames_to_column %>% gather(dates, value, -rowname) %>% spread(rowname, value) %>%
   mutate(dates = ymd(dates),views = as.integer(`1`)) %>% select(-`1`)
}

TS_norm <- function(rownr){
  Onlydates %>% filter_((interp(~x == row_number(), .values = list(x = rownr)))) %>%
    rownames_to_column %>% gather(dates, value, -rowname) %>% spread(rowname, value) %>%
    mutate(dates = ymd(dates),views = as.integer(`1`)) %>% select(-`1`) %>%
    mutate(views = views/mean(views))
}
```

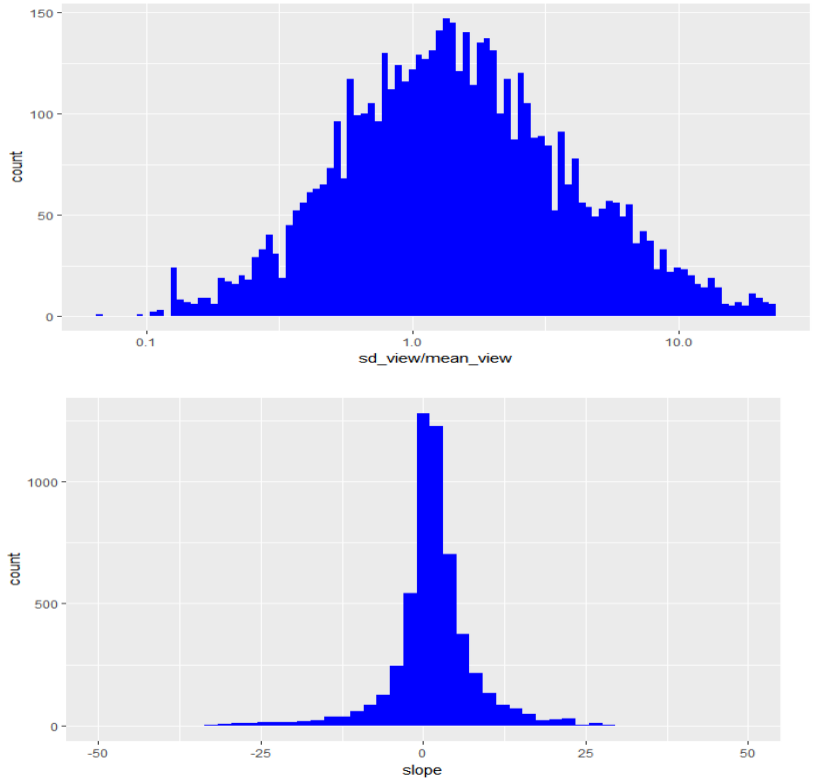
Using this function above helped us in extracting time series for all the individual time series data and this will be a part of our processes later on.

4. **Defining Basic Parameters**: We next wanted to create parameters like mean, standard deviation, amplitude and slope for the data. This needed to be done because we wanted to see some visualizations like the change in distribution for the average views in the data, distributions for the maximum views, distribution of standard deviations and for the slope. These visualizations help us in getting an idea about where are the peaks in these time series and where all are the distributions skewed.

   Skews in the distributions can help us in identifying the trends in the time series and the variability in those trends.







From the plots we can see the following:

1)From the first two plots we can see that both of them appear to have 2 maxima points in the distributions. It's interesting to figure out how & why this may have happened.

2)We can see that the distribution is somewhat skewed for the standard deviations when divided by the mean and this shows that there is some sort of trend in the data that is varying. Our goal would be to forecast these predictions as accurately as we can.

3)As far as the slope is concerned we can see that the distribution for the slope is somewhat symmetric.

5. **Modeling Techniques:** After a couple of visualizations we switched over to modeling part of the project. Our first choice was the ARIMA model. ARIMA stands for Autoregressive Integrated Moving Average models. The single vector ARIMA works best when the data has a consistent pattern over time with a minimum amount of outliers, it means that there should not be any trend or seasonality in the time series.
   The first parameter that needs to be checked before applying ARIMA methodology is whether the data is stationary or not. If a trend exists, as in most economic or business applications, then your data is NOT stationary. Differencing is a substantial way of transforming a non-stationary series to a stationary one. Differencing is implemented by subtracting the observation in the current period from the previous one.

   Autocorrelations are numerical values that quantify how a data series is related to itself over a period of time. It uses Lags which is the number of periods. For example, an autocorrelation at

lag 1 measures how values 1 period apart are correlated to one another throughout the series. Moving average parameters relate what happens in period t only to the random errors that occurred in past time periods.

ARIMA methodology allows the usage of both the autoregressive and moving averages models in order to incorporate both autoregressive and moving average parameters together. The ARIMA architecture that generates optimal forecasts.

For a stationary time series y_t *t = 1, 2, 3,* ...an autoregressive model of order *p*, denoted as AR(p), is expressed as

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots + \beta_p Y_{t-p} + \epsilon_t$$

For a time series y_t, centered around 0, a moving average model of order *q*, denoted MA(q) is

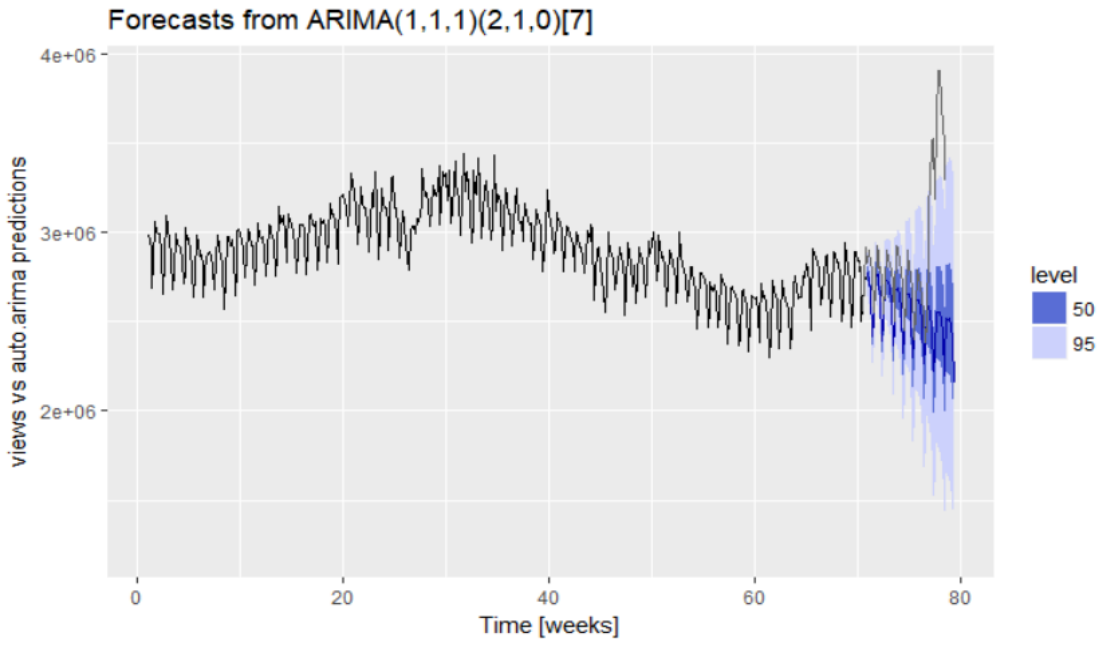
$$Y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q}$$

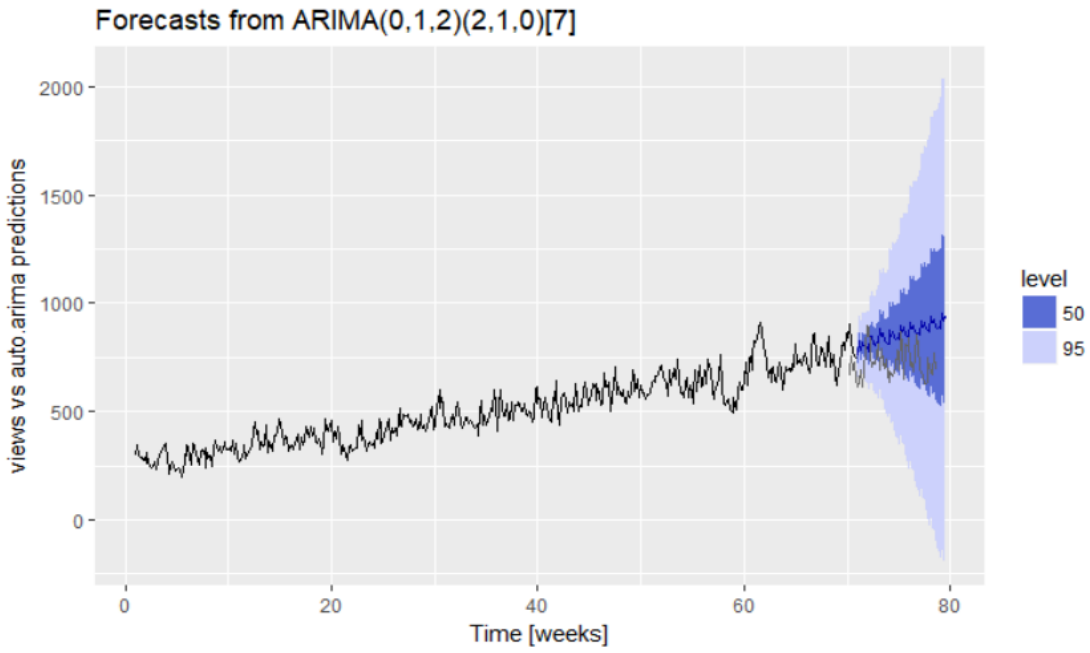
Differencing can also be applied multiple times to get a stationary time series. Once we have a stationary time series, we can then apply an ARMA (p, q) model.

$$d_t = y_t - y_{t-1} \quad \text{for t} = 2, 3, \ldots, \text{n}$$

The Autoregressive Integrated Moving Average model is similar to the structure of the ARMA model, with the ARMA (p, q) model applied to the time series y_t after applying differencing *d* times.

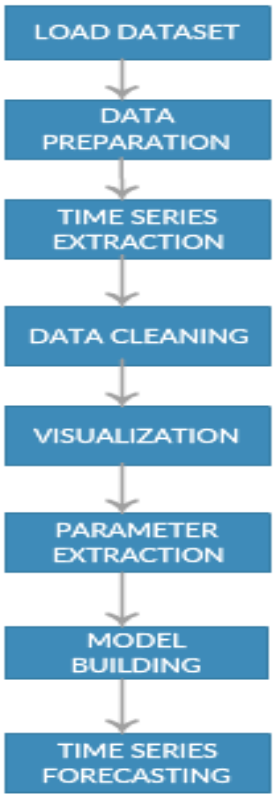
Some of the predicted forecasts are below. The graphs below show the actual vs predicted outcomes of our model.The light blue line is the predicted forecast with 95% confidence interval while the darke blue line is the predicted value with 50% confidence interval. The grey line shows the actual graph. This week has been done using weekly duration to be able to see better insights. The metrics that we used for evaluating the efficiency of our model will be explained in the later sections of this paper.


Forecasts from ARIMA(1,1,1)(2,1,0)[7]
4e+06
3e+06
2e+06
views vs auto.arima predictions
0
20
40
60
80
Time [weeks]
level
50
95


Forecasts from ARIMA(0,1,2)(2,1,0)[7]
2000
1500
1000
500
0
views vs auto.arima predictions
0
20
40
60
80
Time [weeks]
level
50
95

6. Visual Applications

Process of Data analysis and Forecasting flowchart



7. Experimental Evaluation

- **Dataset Description**:
  The training dataset had approximately 145k time series. Each of these time series represent the number of views different wikipedia articles starting from July, 1st 2015 until December 31st 2016.
  We did prediction on the data from January until March 2017. All the time series provided to us has the article name as well as the the type of device used to access those articles like desktop,mobile or spider. The data has many zero rows which could either be zero values or missing data. The missing data could be because of two reasons : either the data is missing or there are no views for those articles.
  The datasets used are:
  1. train.csv- This csv file has each row corresponding to a particular article and each column correspond to a particular date.

The page names contain the Wikipedia project (e.g. en.wikipedia.org), type of access (e.g. desktop) and type of agent (e.g. spider).

	Page	2015-07-01	2015-07-02	2015-07-03	2015-07-04	2015-07-05	2015-07-06	2015-07-07	2015-07-08	2015-07-09	2015-07-10	2015-07-11	2015-07-12	2015-07-13	2015-07-14
1	2NE1_zh.wikipedia.org_all-access_spider	18	11	5	13	14	9	9	22	26	24	19	10	14	15
2	2PM_zh.wikipedia.org_all-access_spider	11	14	15	18	11	13	22	11	10	4	41	65	57	38
3	3C_zh.wikipedia.org_all-access_spider	1	0	1	1	0	4	0	3	4	4	1	1	1	6
4	4minute_zh.wikipedia.org_all-access_spider	35	13	10	94	4	26	14	9	11	16	16	11	23	145
5	52_Hz_I_Love_You_zh.wikipedia.org_all-access_spider	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
6	5566_zh.wikipedia.org_all-access_spider	12	7	4	5	20	8	5	17	24	7	12	11	7	9
7	91Days_zh.wikipedia.org_all-access_spider	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
8	A'N'D_zh.wikipedia.org_all-access_spider	118	26	30	24	29	127	53	37	20	32	17	23	47	33
9	AKB48_zh.wikipedia.org_all-access_spider	5	23	14	12	9	9	35	15	14	22	8	16	18	12
10	ASCII_zh.wikipedia.org_all-access_spider	6	3	5	12	6	5	4	13	9	15	18	7	8	12

- **Parameter settings**

  A popular approach in time series forecasting is to use an autoregressive integrated moving average model

  1. autoregressive / p: We are using past data to compute a regression model for future data. The parameter *p* indicates the range of lags; e.g. ARIMA(3,0,0) includes *t-1*, *t-2*, and *t-3* values in the regression to compute the value at *t*.
  2. integrated / d: This is a *differencing* parameter, which gives us the number of times we are subtracting the current and the previous values of a time series. Differencing removes the change in a time series in that it stabilises the mean and removes (seasonal) trends. This is necessary since computing the lags (e.g. difference between time *t* and time *t-1*) is most meaningful if large-scale trends are removed. A time series where the variance (or amount of variability) (and the autocovariance) are time-invariant (i.e. don't change from day to day) is called *stationary*.
  3. Moving average /q: This parameter gives us the number of previous error terms to include in the regression error of the model.

- **Evaluation measures**-

  It is important to evaluate forecast accuracy using genuine forecasts. That is, it is invalid to look at how well a model fits the historical data; the accuracy of forecasts can only be determined by considering how well a model performs on new data that were not used when fitting the model. When choosing models, it is common to use a portion of the available data for fitting, and use the rest of the data for testing the model.. Then the testing data can be used to measure how well the model is likely to forecast on new data.

  The size of the test set is typically about 20% of the total sample, although this value depends on how long the sample is and how far ahead we want to forecast. The size of the test set should ideally be at least as large as the maximum forecast horizon required. The following points should be noted:

  1. A model which fits the data well does not necessarily forecast well.
  2. A perfect fit can always be obtained by using a model with enough parameters.

3. Overfitting a model to data is as bad as failing to identify the systematic pattern in the data.

The function accuracy gives us multiple measures of accuracy of the model fit: mean error (ME), root mean squared error (RMSE), mean absolute error (MAE), mean percentage error (MPE), mean absolute percentage error (MAPE), mean absolute scaled error (MASE) and the first-order autocorrelation coefficient(ACF1).
The two most commonly used scale-dependent measures are based on the absolute errors or squared errors:

```
Training set error measures:
                     ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
Training set -0.1479399 5.811764 3.916493 -9.252873 26.12201 0.8017369 0.0870278
Series: tsclean(ts(pre_views$views, frequency = 7))
ARIMA(0,1,1)(2,1,0)[7]

Coefficients:
          ma1     sar1     sar2
      -0.8345  -0.7090  -0.4086
s.e.   0.0359   0.0418   0.0411

sigma^2 estimated as 15.82:  log likelihood=-1350.99
AIC=2709.97   AICc=2710.05   BIC=2726.68
```

One of the performance metric which we considered for calculating the model accuracy is MAPE (Mean absolute percentage error). It considers actual values fed into model and fitted values from the model and calculates absolute difference between the two as a percentage of actual value and finally calculates mean of that. As we can see from the above screenshot that MAPE is 26.12 that means the model is 74% accurate which is quite good prediction.
As the seasonality is not considered while building this model the results are good enough at this stage.

8. Future Work

One of the next steps that we are considering in this project is to be able to capture interactions/correlations between pages. One way we can materialize this is by implementing the neural network algorithm. This could give us better insights about the important past behaviors and their significance in future predictions. The fact that we have a lot of time series data in this project might give us good results when neural networks would be implemented over it. Especially the Recurrent neural networks that have proven to be useful when we need to deal with the time series data. We can use it for multivariate time series forecasting. We have also decided on using the Seasonal Arima for this project to demonstrate long-term seasonal variations in the data is crucial for a successful forecasting of our time series data. We can account for these seasonal patterns by using a seasonal autoregressive integrated moving

average model, denoted by ARIMA (p, d, q) X (P, D, Q), where

- p, d, q are as before
- s denotes the seasonal period, 52 for weekly data; 12 for monthly data; 7 for daily data
- P is the number of terms in the AR model across s periods
- D is the number of differences applied across s periods
- Q is the number of terms in the MA model across s periods

We will be applying the Holt-Winters forecasting procedure which is a simple widely used projection method that helps in coping with trend and seasonal variation in the time series data. This approach uses the multiplicative seasonality for each of the time series data.

9. References

1. Yisheng, Lv & Duan, Yanjie & Kang, Wenwen & Li, Zhengxi & Wang, Fei-Yue. (2014). Traffic Flow Prediction With Big Data: A Deep Learning Approach. IEEE Transactions on Intelligent Transportation Systems. 16. 865-873. 10.1109/TITS.2014.2345663.

2. Srinivasa Prasad, Kalli & Ramakrishna, Seelam. (2014). An Efficient Traffic Forecasting System Based on Spatial Data and Decision Trees. International Arab Journal of Information Technology. 11. 186-194.

3. Yasseri, T. & Bright, J. EPJ Data Sci. (2016). Wikipedia traffic data and electoral prediction: towards theoretically informed models. https://doi.org/10.1140/epjds/s13688-016-0083-3

4. P Ij Van Hinsbergen, C & Lint, J.W.C. & M Sanders, F. (2007). Short Term Traffic Prediction Models. 14th World Congress on Intelligent Transport Systems, ITS 2007.

5. Li, Jia & W. Moore, Andrew. (2008). Forecasting Web Page Views: Methods and Observations. Journal of Machine Learning Research. 9. 2217-2250.

6. Shu, Yantai & Yu, Minfang & YANG, Oliver & Liu, Jiakun & Feng, Huifang. (2003). Wireless Traffic Modeling and Prediction Using Seasonal ARIMA Models. IEICE Transactions on Communications. E88B. 10.1093/ietcom/e88-b.10.3992.

We also referred following links:

https://media.readthedocs.org/pdf/a-little-book-of-r-for-time-series/latest/a-little-book-of-r-for-time-series.pdf

https://medium.com/@aneesha/timeseries-forecasting-with-the-forecast-r-package-and-shiny-6fa04c64196