IJMLNCE JOURNAL

International Journal of Machine Learning and Networked Collaborative Engineering

Journal Homepage: http://www.mlnce.net/home/index.html

DOI : https://doi.org/10.30991/IJMLNCE.2019v03i02.002

Machine Learning Prediction of Wikipedia Time Series Data using: R Programming

Yagyanath Rimal

Faculty of Science and Technology

Pokhara University, Nepal

Email: rimal.yagya@gmail.com

https://orcid.org/0000-0003-1045-7728

Abstract

This review article explains the prediction of automatic learning of Wikipedia time series data using r programming. Although many time series forecast researchers have been analyzed the time series could not cover the gap between chart interpretation and time series analysis of the Internet database directly. Its main objective is to explain the simplest way to time model series whose data structure was different using R programming, the result was sufficiently summarized with different forecast models. The simplest form of analysis with graphical interpretation to obtain conclusions from the time search is Cristiano_Ronaldo of Wikipedia, a best player in euro football team. Whose trend and prediction is analyzed for next 2020 from the past records trend. Therefore, this document presents the simplest way to predict time series data and its strengths for data analysis using R programming.

Keywords

Data Analytics Machine Learning, Autocorrelation Function, Particle Automatic Correlation Function

1. Introduction

A historical series is a set of observations at different measurement periods of an employee who collected at regular time intervals, such as monthly, weekly or yearly, such as the state budget, etc. According to (Gahirwal, 2018), the precondition is the one whose interval must be the same. Time series forecasts are widely used in econometrics, mathematical and financial forecasts of statesmen and different weather forecasts and earthquakes. Magazine without independent variable (VANNESCHI, 2017). With the time-based model, the researcher can interpolate the graphics of the model and then foresee a future project. The temporal variable usually uses yt = yt-1 + E as a univariate model. The data in cross section are such data collection procedures that are similar to the time series, but the time series data only have one variable depending on the interval and the association of a single value, but in the cross-data collection can collect many variables Elements in a fixed time interval. The data model may vary based on the time series model. It is not expected, although there are many types such as seasonal, trend, cyclic and random. Models are increasing or decreasing models. The period of occurrence has been corrected within a year or less. The cyclical model is a good example of the government budget. Some data are also random. Purely random, whose average is zero and the variance is constant (CHEN, 2013). So the dispersion diagram will not indicate the correct model. The automatic regressive model is a model that corresponds to the sequences yt-1, yt-2 and t-3. Therefore, the model becomes yt = b0 + b1yt-1 + b2yt-2 + b3y3-3 ... ..Et. The AR (0) means

B0, so the AR model is better when it works. The moving average model always speaks of the error terms in each regression and t = B0 + Et + Q1E1 + Q2Et-2 + QET-q. Therefore, ARMA is a precise model that uses AR and MA models in temporal data (Michael Jachan, 2007). regressive automatic movement media regressive integrated mobile media ARIMA is commonly known as the Box Jenkins methodology (1976), a method used to predict the basis of information from its own variables, based on an analysis of trends univariate tendencies. After analyzing the periodic properties of the variables, machine learning. Regulators, policy makers and companies to make serious and economic forecasts; however, the choice is based on the two hypotheses (SALAM, 2013). The AR model is applied in series as stationary if there is an invariable time variation. This decreases the major MA duration model applied to the autoregressive process convergent final order that uses the autocorrelation function (ACF) is the covariance of the following terms and functions automatic creation partial yt (PACF) and yt p. The series is not firm, can be fixed after differentiation.

Machine learning is the system that takes input and output as input parameters, so that the computer system automatically produces the above parameters based on some models. This can be applied to the analysis of neural networks, deep learning and artificial intelligence networks. traditional programming always requires two inputs and programs and possibly produces an output, while the process of machine learning always requires an input and real requirement, since the formal entry system produces the show and expected output depending on the needs of the system (Sunday 2018). A fixed series after an integrated differentiation in the order 1. Analysis Box - Jenkins concerns a systematic method of identification, regulation, monitoring and the use of integrated models of the integrated moving time series (ARIMA initial). The method is appropriate for medium to long time series. (Buncher 2018) The best correspondence between the ARIMA model and the time series data is the best correspondence of ARIMA (0,0,0) means that p = q = i = 0.

The data science process includes two processes; The first process begins with cleaning the raw data and data collection analysis using different algorithms to produce the display data in this process. At each stage, IT skills, mathematics, statistics and interpretation are required. However, the data available in today's world in any structured, unstructured and semi-structured format are raw data. The primary phase includes the integration of raw data and, therefore, the selection of the required data, at some point requires a processing required before the analysis. data cleaning requires 50% to 80% of the work of a set of scientific data (Ruiz, 2017). Time series analysis includes methods for analyzing time series data to extract significant statistics and other data characteristics. The prediction of time series is the use to predict future values based on the values observed above. Historical headline series and retail sales in this publication are widely used for non-stationary data such as the economic climate. We will demonstrate different approaches to predict the time series to detail. There are two types of machine learning: supervised learning and unsupervised learning. However, R programming, python and weka are the best tools for data analysis; The data scientist can use many other data analysis processes. R has extensive facilities for analyzing time series data. This section describes the creation of a historical series, seasonal decomposition, exponential models and modeling and prediction with the ARIMA package the prognosis (Change, 2018). Modeling time series, as the name suggests, means working with (time days, hours, minutes) time-based data for hidden information and making informed decisions. Time series models are very useful models when data is related in series. Most business houses that work with time series data to analyze the number of sales next year, website traffic, place of competition and much more. However, it is also one of the areas that many analysts do not understand. There are three basic criteria for a series to be classified as stationary series. The series average should not be a function of time, but must be a constant. The variance of the series should not be a function of time. This property is called homoscedasticity with a variable data distribution. The term covariance i-th term and (i + m) th should not be a function of time, therefore, covariance is not constant over time for uniform (Chohlan 2018). The reason I took this first section is that, unless the time series is stopped, it is not possible to create a time series model. In cases where the fixed criterion is violated, the first requirement becomes stationary time series and therefore to try out stochastic models to foresee this historical series. There are many ways to bring this stationarity. This is the most basic concept of time series. (Srivastavo, 2015).

2. Using R Programing

Here I use data set of the database Cristiano_Ronaldo of Wikipedia, is the top scorer in international football careers. He has won 26 trophies in his career, including five league titles, five UEFA Champions League titles and the UEFA European Championship. Ronaldo, a highest scorer, holds the record for the Master challenge.

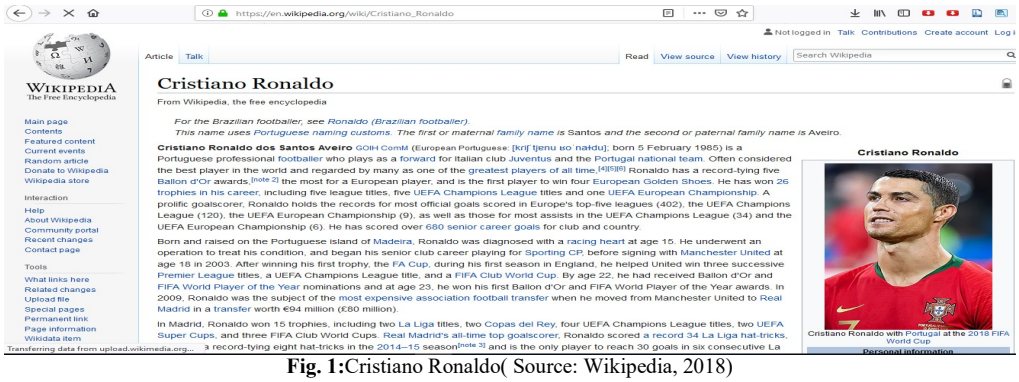

Fig. 1: Cristiano Ronaldo(Source: Wikipedia, 2018)

```
> install.packages("wikipediatrend")
Error in install.packages : Updating loaded packages
> library(wikipediatrend)
> devtools::install_github("petermeissner/wikipediatrend")
> library(wikipediatrend)
> wc=wp_trend( page="Cristiano_Ronaldo",from="2015-07-01",to="2018-10-31")
The package wp_trend() function that allows us to get page view of data time series data.
> wc
     granularity date       views
1124 daily       2018-0 ..36744          715  daily       2017-0 ..30132
612  daily       2017-0 ..19273          232  daily       2016-0 ..60506
927  daily       2018-0 ..23233          379  daily       2016-0 ..74485
1003 daily       2018-0 ..22562          454  daily       2016-0 ..18957
789  daily       2017-0 ..31208          994  daily       2018-0 ..38602   ... 1209 rows of data not shown
> library(ggplot2)              > View(wc)
> qplot(date,views,data=wc)
```

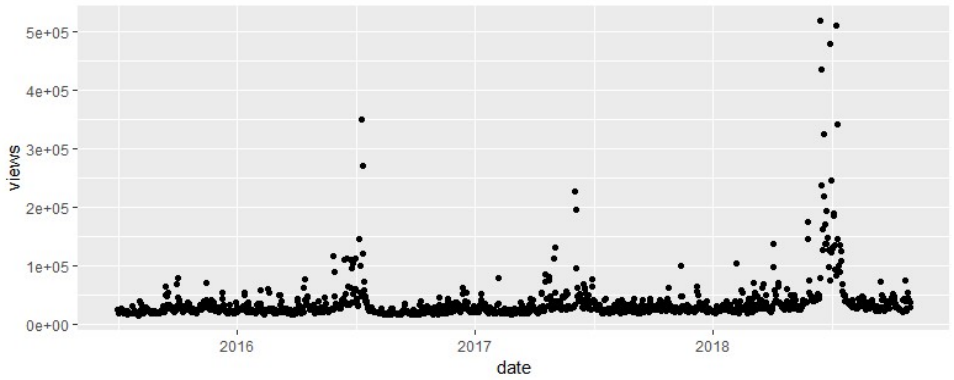

Fig. 2 :Daily Views Search for Cristiano Ronaldo keyword

From the above figure there were more seasonality in data sets but good in rising pattern in some duration.

```
> summary(wc)
project   language      article
Length:1219 Length:1219 Length:1219 Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character
access  agent       granularity
Length:1219 Length:1219 Length:1219
Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  : character         date                    views
Min.:2015-07-0100:00:Min.: 15478  1st Qu.:2016-04-30:01stQu.:23040
Median :2017-03-01: Median:27818
Mean:2017-03-01 00: Mean:37300
3rd Qu.:2017-12-30:3rd Qu.:37117
Max.:2018-10-31 00:Max.:519908
> wc$views[wc$views==0]=NA
> ds=wc$date
> y=log(wc$views)
> df=data.frame(ds,y)
> qplot(ds,y )
```

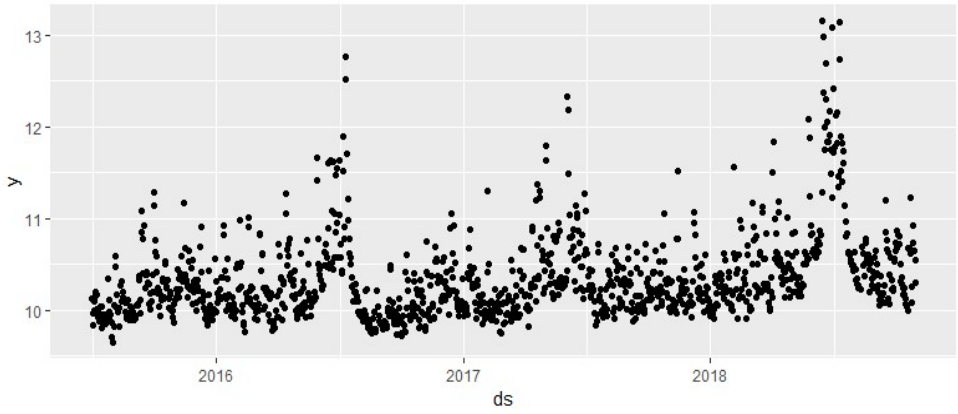

Fig. 3: Daily Log of goal Score search for the Keyword

This figure plots the log goal score and data variable in x axis shows pattern.

```
> installed.packages("prophet")
> library(prophet)
```

Here we are using forecast function columns ds (date type) and y, the time series. If growth is logistic, then df must also have a column cap that specifies the capacity at each ds. If not provided, then the model object will be instantiated but not fit; use fit.prophet(m, df) to fit the model.

```
> m=prophet(df)
> m
$`growth`
[1] "linear"
$changepoints
 [1] "2015-08-09 GMT" "2015-09-17 GMT" "2015-10-26 GMT"
 [4] "2015-12-04 GMT" "2016-01-12 GMT" "2016-02-20 GMT"
 ……
[22] "2017-11-04 GMT" "2017-12-13 GMT" "2018-01-21 GMT"
[25] "2018-03-01 GMT"
$n.changepoints            [1] 25
$changepoint.range         [1] 0.8
$yearly.seasonality        [1] "auto"
$weekly.seasonality        [1] "auto"
……
$uncertainty.samples     [1]1000
$specified.changepoints  [1] FALSE
$start       [1]"2015-07-01 GMT"
$y.scale     [1] 13.16141
$logistic.floor[1] FALSE
```

$t.scale [1] 105235200

$changepoints.t

[1] 0.03201970 0.06403941 0.09605911 0.12807882 0.160

……..

[6] 0.19211823 0.22413793 0.25615764 0.28817734 0.320

[11] 0.35221675 0.38423645 0.41543514 0.44745484 0.47

[16] 0.51149425 0.54351396 0.57553366 0.60755337 0.63

[21] 0.67159278 0.70361248 0.73563218 0.76765189 0.79

$seasonalities

$seasonalities$`yearly`

$seasonalities$`yearly`[1]365.25

$seasonalities$`$fourier.[1] 10

$seasonalities$`yea$r.sca[1] 10

$seasonalities$`$mode[1]"addite"

$seasonalities$weekly$mode [1]

$extra_regressors list()

$`stan.fit` NULL

$params

$params$`k` [1] 0.6589777

$params$m [1] 0.7410152

$params$delta

[,1] [,2] [,3] [,4]

[1,] -6.962949e-05 -0.4095019 -0.3294997 -5.841874e-08

[,5] [,6] [,7] [,8]

[1,] -4.191081e-08 -3.546061e-09 2.497178e-07 0.0244269

……

[,22] [,23] [,24] [,25]

[1,] -8.958495e-08 -3.950414e-08 -3.443782e-08 -0.0429

$params$sigma_obs [1] 0.02643767

$params$beta

[,1] [,2] [,3] [,4]

[1,] 0.005891249 -0.01776793 -0.008432092 0.01224181

[,5] [,6] [,7] [,8]

…….

[,25] [,26]

```
[1,] 0.0001815386 0.001226727
$history
ds      y floor      t  y_scaled
1  2015-07-01 10.129507   0 0.0000000000 0.7696371
2  2015-07-02  9.968854     0 0.0008210181 0.7574307
3  2015-07-03  9.834834     0 0.0016420361 0.7472479
……
198 2016-01-14 10.156656     0 01617405583 0.7716998
……
199  2016-01-15  9.999298     0
0.1625615764 0.7597438
200  2016-01-16 10.128190     0 0.1633825944 0.7695370
[ reached getOption("max.print") -- omitted 1019 rows ]
$history.dates
[1] "2015-07-01 GMT" "2015-07-02 GMT" "2015-07-03 GMT"      [4] "2015-07-04 GMT" "2015-07-05 GMT" "2015-07-06 GMT"
[7] "2015-07-07 GMT" "2015-07-08 GMT" "2015-07-09 GMT"……
…….
 [997] "2018-03-23 GMT" "2018-03-24 GMT" "2018-03-25 GMT"
[1000] "2018-03-26 GMT"
[ reached getOption("max.print") -- omitted 219 entries ]
[ reached getOption("max.print") -- omitted 219 entries ]
$train.component.cols
Additive weekly yearly multiplicati
1     1      0      1       0
2     1      0      1       0
……..
25    1      1      0       0
26    1      1      0       0
$component.modes
$component.modes$`additive`
[1] "yearly"           "weekly"
[3] "additive_terms"  "extra_regressors_additive"
[5] "holidays"
$component.modes$multiplicative
[1]"multiplicative_terms"
```

```
[2]"extra_regressorsmultiplicative"
attr(,"class")
[1] "prophet" "list"
> future=make_future_dataframe(m,periods=365) #For next year
> tail(future)
           ds
1579 2019-10-26              1580 2019-10-27
1581 2019-10-28              1582 2019-10-29
1583 2019-10-30              1584 2019-10-31
> forecast=predict(m,future)
> tail(forecast)
 ds trend additive additive_lowr
1579 2019-10-26 11.08094    -0.17044324 -0.17044324
1580 2019-10-27 11.08184    -0.08389419 -0.08389419
……..
additive_terms_weekly weeklylowr
1579-0.17044324  0.010662724  0.010662724
1580-0.08389419  0.093079248  0.093079248
……..
1584-0.18950107 -0.031694505 -0.031694505
weekly_upperyearly_lower upper
1579  0.010662724 -0.1811060   -0.1811060   -0.1811060
1580  0.093079248 -0.1769734   -0.1769734   -0.1769734
…….
1584 -0.031694505 -0.1578066   -0.1578066   -0.1578066
multiplicative_terms multiplicative
1579       0           0
……
1584       0           0
multiplicative_terms yhat yhat
1579 0  10.11238 11.64221
……..
1584 0  10.11592 11.63837
trend_lower trend_upper yhat
1579 10.50417  11.70324 10.91049
```

```
……..
1584 10.49862  11.72232 10.89598
> tail(forecast[c('ds','yhat','yhat_lower','yhat_upper')])
ds    yhat yhat_lower yhat_upper
1579 2019-26 10.91 10.11 11.642
……
1584 2019-31 10.89 10.11 11.638
> exp(10.89598)
[1] 53959.01
> plot(m,forecast)
```

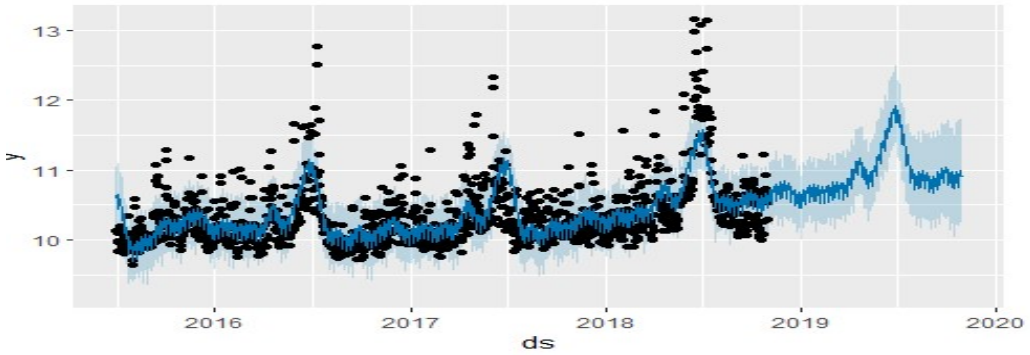

Fig. 4 : Daily Trend Forecasting for 'Cristino Ronaldo'

From the above plot Cristino Ronaldo had prosperous in upcoming years.

```
> prophet_plot_components(m,forecast)
```

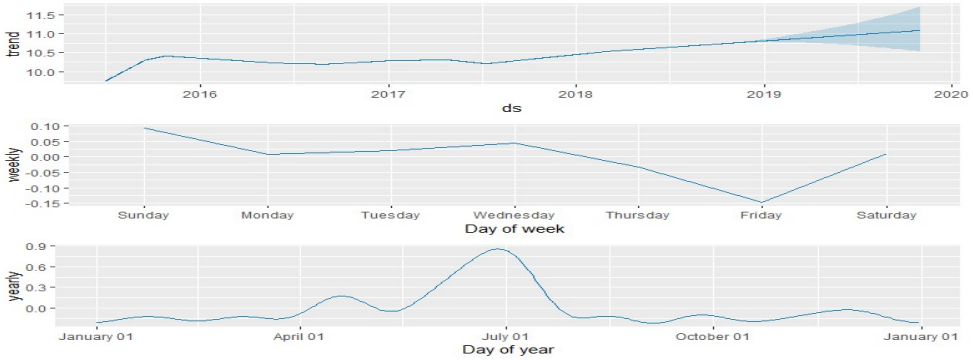

Fig. 5 : Daily Trend, Week Day Trend , and Date wise Trend for keyword 'Cristino Ronaldo'

From the above figure *Cristiano Ronaldo* had good future in real Madrid champions ship however Friday matches does not suite him and July is the best suitable moment.

Conclusion

A common goal of time series analysis is the extrapolation of past behavior in the future. Forecasting procedures include random walks, moving averages, trend models, simple, linear, quadratic, and seasonal

exponential time series models. Business forecasts can be based on historical data models used to predict future market behavior. The time series forecasting method is a data analysis tool that measures historical data points, using line charts to predict future conditions and events. It is essential to analyze trends before building any type of time series model. The details that interest us refer to any kind of trend, seasonality or random behavior in the series. Once we know that patterns, trends, cycles and seasonality, the function of the attacking forces and the defense of the competition could be analyzed, corrective measures will be taken, so Cristiano Ronaldo will have a good future in the next years after entering in the Real Champions League of Madrid has will good future coming day.

References

[1]. Buncher, D. (2018). "The Box-Jenkins methods". *NCSS Statistical Software*.
[2]. Changing, S. L. (2018). "The world, one article at a time". *Toronto Canada. Opinion=my own. ttps://www.linkedin.com/in/ susanli/*, Sr. Data Scientist,.
[3]. CHEN, M.-Y. (2013). " Time Series Analysis (I)". *Department of Finance*.
[4]. Chohlan, A. (2018). "A little book of R for time series" . *DataCamp's manipulating time series in R course by Jeffrey Ryan.* .
[5]. Domingos, P. (2018). "A Few Useful Things to Know about Machine Learning". *Department of Computer Science and Engineering*.
[6]. Gahirwal, M. (2018). "Inter Time Series Sales Forecasting". *Information Technology, Vivekanand Education Information Technology, Vivekanand Education*.
[7]. Michael Jachan. (2007). "Time-Frequency ARMA Models and Parameter Estimators for Underspread Nonstationary". *IEEE Transactions on Signal Processing,* Vol. 55, No. 9, September 2007.
[8]. Ruiz, A. (2017). "The Cognitive Coder InfoWorld".
[9]. SALAM, M. A. (2013). "Modeling and Forecasting Pakistan's Inflation by Using". *Statistical Officer, Statistics Department, State Bank of Pakistan, Karachi, Pakistan.*
[10]. Srivastavo, T. (2015). "A Complete Tutorial on Time Series Modeling in R".
[11]. Vanneschi, l. (2017). "Retail forecasting under the influence of promotional discounts". *Instituto Superior de Estatística e Gestão de Informação*.

Author's Biography

Yagyanath Rimal is lecturer of Computer Science Information Technology at Pokhara University, Nepal. He received his Master of Science in Information Technology from SMU in 2006, India with Information Technology and programming specialization, he had wide area of interest like teaching, writing books, doing research and publishing international research articles, web development and presentation at international conferences.

How to Cite

Rimal, Yagyanath, "Machine Learning Prediction of Wikipedia Time Series Data using: R Programming", *International Journal of Machine Learning and Networked Collaborative Engineering,* Vol. 03, No. 2, 2019, pp. 83-92.

doi : https://doi.org/10.30991/IJMLNCE.2019v03i02.002.