AVERAGE MONTHLY RAINFALL FORECAST IN ROMANIA BY USING 
k-NEAREST NEIGHBORS REGRESSION

MARINOIU CRISTIAN
ASSOC. PROF. PH.D., PETROLEUM-GAS UNIVERSITY OF PLOIEȘTI

e-mail: marinoiu_c@yahoo.com

Abstract

The discovery of the best strategies for achieving future values forecast of a time series represents a permanent concern in time series analysis, highly motivated from a theoretical point of view, but especially from a practical point of view. In the context of the explosive growth of machine learning techniques, their use in time series forecast is a natural step to find modern alternatives to overcome existing limitations of traditional techniques. Although it is a relatively a simple method of learning, \( k\text{-}nn \) (k-nearest neighbor) regression seems to be a good competitor to traditional methods. The purpose of this paper is to describe how to use this method for forecasting time series and for achieving Monthly Average Rainfall (AMR) forecast in Romania.

Key words: time series, forecast, \( k\text{-}nn \) regression, average monthly rainfall

JEL Classification: C22, C53

1. Introduction

Nowadays "forecasting is an important aid to effective and efficient planning" [7] in various fields of human activity: financial mathematics, marketing, meteorology, geology, hydrology etc. For this reason, development of methods for forecasting time series is of real interest not only to the researchers in science data, but also to the beneficiaries of these methods. Using machine learning methods in the field of time series forecast proved, in many instances, to be a viable alternative to using traditional methods. An overview on how to apply the methods of machine learning in the prediction of time series is carried out in [4], while the works [9]-[10] make specific reference to the methodology for the application of one of the simplest methods of this type, \( k\text{-}nn \) regression method.

Climate change is a very important and active research topic, as reflected in a rich specific literature [11]. Given that the precipitations and in particular rainfalls together with temperature and wind parameters are essential characteristics of climate [18] forecasting rainfall amounts rise great practical interest, as reflected in numerous works. For example, in [8] monthly rainfall forecast is performed on a time horizon of 12 months, based on data collected from thirty rainfall stations in Bangladesh, in [2] for monthly rainfall forecast, data collected are used for the period 1980-2006 and in [12] monthly rainfall forecast in Kavala city, Greece, North-Eastern Mediterranean basin, is made based on data collected from weather stations in the area, along 10 years of monthly observations. All these relatively recent works use the well known ARIMA method, developed in [5], for making forecasts. Given the complexity of the physical phenomenon of precipitation and the practical necessity to obtain accurate forecasting, many researchers use today methods from the field of machine learning, applied either individually or in combination, this latter approach being known as ensemble modeling. In this way, in [14], the methods Artificial Neural Network (ANN), Multivariate Adaptive Regression Spline (MARS), \( k\text{-}nn \) regression, Radial Basis Support Vector Regression (RBSVR) are used both individually and together, combined in a hybrid model, in order to obtain monthly rainfall forecast in Fukuoka city from Japan. While in the works mentioned above the authors have as a basis for forecasting only records from the past of rainfall amounts, authors like [17] and [1] include in their forecasting model other related attributes of the weather which influence the rainfall volume: Temperature, Atmospheric Pressure, Relative humidity etc.
In this paper we chose to apply the \( knn \) method for forecasting Average Monthly Rainfall in Romania (AMR), having as forecasting basis only the past values can be found in [19]. Because we have not noticed a lack of observations over the period considered and the values of the series have comparable orders of magnitude, it was not necessary to perform some preliminary operations on the data, such as filling or excluding some records, data normalization etc.

This paper is organized as follows. In chapter 2, a short presentation of the strategies for time series prediction is made and, in chapter 3, \( knn \) regression method and ways of organizing data to be used in forecasting time series. The illustration of the actual use of this method is covered in chapter 4, where a case study is provided: the AMR time series forecast. The degree of accuracy for the predictions obtained using \( knn \) is compared to that obtained using two other methods (ARIMA and Neural Network), in chapter 5. The conclusions and future directions are presented in chapter 6 and the references in chapter 7.

2. Time series forecast strategies

Let \( y = (y_1, y_2, \ldots, y_N) \) be a time series of length \( N \). A classification of forecast strategies of the time series is performed in [10] depending on the nature of the outputs produced by the used prediction model:

- Single-output strategy, where the prediction model output is a scalar value;
- Multiple-output strategy, where the prediction model output is a vector.

If the forecast horizon of the series is \( h = 1 \), meaning that only the estimation of the future value of the time series \( y_{N+1} \) is desired, then we use only once the first strategy (one-step-ahead). If the forecast horizon of the time series is \( h > 1 \), then multi-step-ahead prediction can be obtained through the following two approaches:

1. By using repeatedly single-output strategy;
2. By using multiple-output strategy.

In the first category are the recurrent or iterative method and the direct method. In the case of the recurrent method a number of \( h \) one-step-ahead forecasts is achieved, by indicating that, at every step, the one-step ahead value forecast is added to the set of values on which the following one-step-ahead forecast is made. The weakness of this strategy is that making a one-step-ahead forecast based on previous similar forecasts favors the propagation of errors which affect the accuracy of the obtained results. On the other hand, in the case of the direct method, each forecast \( y_{N+i} \), \( i = 1, 2 \ldots h \) is carried out independently of the other, which requires a more complex functional modeling in order to counter the inability of this approach to maintain natural stochastic dependence between the predicted values.

The second strategy (time series forecast with output vector) was introduced in [3]. In this case, the \( h \) values of the forecast horizon are obtained by means of a predictive model in which the output is a vector of size \( h \). Among the methods using this strategy we mention the MIMO method (Multi-Input Multi-Output) from [3] and the MISMO method (Multi-Input Multi-Output Several), introduced in [15].

3. The \( knn \) method

3.1 General presentation of the \( knn \) method

The \( knn \) \( (k \) nearest neighbors) method [6] is a technique from the supervised learning field of machine learning that can be used both for classification and for regression. Supervised learning is characterized by the fact that to the available input data, output values (labels) are associated, that guide the learning process. The standard form of data presentation (called training data), as a basis for developing the model for supervised learning algorithms, is outlined in Figure 1, in the form of a table, where:

- rows, called records or instances, represent available objects for which the label is known;
• columns represent the attributes of the objects, except for the last column where the labels of the instances are stored.

<table>
<thead>
<tr>
<th></th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$\vdots$</th>
<th>$A_p$</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance$_1$</td>
<td>:</td>
<td></td>
<td></td>
<td></td>
<td>$L_1$</td>
</tr>
<tr>
<td>Instance$_2$</td>
<td>:</td>
<td></td>
<td></td>
<td></td>
<td>$L_2$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>Instance$_n$</td>
<td>:</td>
<td></td>
<td></td>
<td></td>
<td>$L_n$</td>
</tr>
</tbody>
</table>

Figure 1. Organizing data for using supervised learning methods

In the case of $knn$ method, the developed algorithms are based on exploiting the information obtained from the nearest $k$ neighbors of the instance for which the forecast is made. The strategy of $knn$ method is very simple: in the case of the classification, the new instance is classified in the class with the highest frequency (majority class) and, for regression, the forecasted value is obtained as the arithmetic mean of the values related to the $k$ nearest neighbors. For the successful use of the method, the value of the $k$ parameter must be specified or selected in relation to a particular performance criterion and a distance metric on the set of instances available must be properly chosen or defined.

3.2 The use of $knn$ regression in forecasting one step ahead time series

In order to use the $knn$ regression for time series forecast, the values of the time series must be, first of all, organized in a convenient form, suitable for the application of supervised learning algorithms, meaning under the form presented in Figure 1. In this way, by considering the time series $y = (y_1, y_2, \ldots, y_N)$ of length $N$, the column vectors of the attributes $A_1, A_2, \ldots, A_p$ are generated on the basis of the lagged values of the $y$ time series. The main idea is that for an output $L_t = y_t$, $t > p$, the entries are $A_i = y_{t-p+i}$, $i = 0,1,\ldots,p-1$. For example, if the length of the $y$ time series is $N = 6$, and the length of the lagged is $p = 3$, then the training data will be organized according to the table in the Figure 2.

<table>
<thead>
<tr>
<th></th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance$_1$</td>
<td>$y_1$</td>
<td>$y_2$</td>
<td>$y_3$</td>
<td>$y_4$</td>
</tr>
<tr>
<td>Instance$_2$</td>
<td>$y_2$</td>
<td>$y_3$</td>
<td>$y_4$</td>
<td>$y_5$</td>
</tr>
<tr>
<td>Instance$_3$</td>
<td>$y_3$</td>
<td>$y_4$</td>
<td>$y_5$</td>
<td>$y_6$</td>
</tr>
</tbody>
</table>

Figure 2. The organization of the training data used in $knn$ regression based on the values of $y = (y_1, y_2, \ldots, y_6)$ time series
In this way, for the entries \( y_1, y_2, y_3 \) the output (the label) is \( y_4 \), for the entries \( y_2, y_3, y_4 \) the output is \( y_5 \), and for the entries \( y_3, y_4, y_5 \) the output is \( y_6 \). By following this logic we may conclude that \( Instance_4 \) will have the form shown in Figure 3, where the values \( y_4, y_5, y_6 \) are known, and the value \( y_7 \) is unknown.

<table>
<thead>
<tr>
<th>( Instance_4 )</th>
<th>( y_4 )</th>
<th>( y_5 )</th>
<th>( y_6 )</th>
<th>? ( y_7 )</th>
</tr>
</thead>
</table>

Figure 3. The instance on which the forecast of the future value \( y_7 \) of the \( y = (y_1, y_2, \ldots y_6) \) time series is accomplished.

The unknown value \( y_7 \) of the \( y \) time series can be forecasted by using the \( knn \) regression model developed based on the training data from Figure 2, the entry data of the model being \( A_1 = y_4, A_2 = y_5, A_3 = y_6 \).

Finally, the steps of the algorithm for achieving one-step-ahead prediction of a time series, by using \( knn \) regression, are the following:

- Read the time series values \( y = (y_1, y_2, \ldots y_N) \), the lagged \( p \) value and the number \( k \) of neighbors;
- Organize the values of the \( y \) time series using the model shown in Figure 2;
- Choose the appropriate distance metric for calculating the distance between any two instances;
- Establish which are the closest \( k \) neighbors of the instance which has the attributes \( A_i = y_{N+1-p+i}, i = 0, 1, \ldots, p - 1 \) and estimates \( y_{N+1} \) as the average of the label values of these neighbors;
- Stop.

### 4. The AMR time series forecast by using \( knn \) regression

The case study presented in this chapter is based on AMR time series from [19], which contain observations on monthly average rainfall recorded in Romania during 1991-2015, in millimeters. The purpose of this study is to achieve, with \( knn \) method, the forecast of the values of AMR time series on a time horizon \( h = 48 \) months. The graph of the time series is presented in Figure 1 and shows, as expected, a very obvious oscillatory character.

![Rainfall 1991-2015](image)

Figure 1. The graph of average monthly rainfall in Romania for the period 1991-2015. Source: made by the author using functions from R [13]
From the previous chapter it follows that, in order to achieve AMR timeseries forecast, we must state the following:

- the metric distance used for computing the *k* nearest neighbors;
- the size of *h* forecast time horizon;
- the number of attributes *p* (the number of lagged values) of the training data table;
- the number of *k* neighbors used in *knn* regression;
- the multi-step-ahaed strategy prediction used.

The metric distance used in this case is the most commonly used distance, meaning the Euclidean distance, and the forecast horizon is *h* = 48. Because AMR time series is a monthly time series, an adequate value for the *p* parameter is *p* = 12 [9]. As for the *k* parameter, the following variants are recommended [10]:

- the heuristic variant: *k* is chosen as being the radical of the number of instances in the training data;
- the optimal variant: *k* is chosen in order to minimize the forecast error;
- the combinatorial variant: a number of *knn* regression models are generated for different preset *k* values and the final prediction will be the arithmetic mean or the median of the predictions achieved for each *k* value.

The heuristic variant has the advantage of speed but, obviously, does not have a rigorous foundation. On the other hand, the advantage of the forecast accuracy of the optimal variant is diminished by the long time required for achieving the optimum. For this reason we chose the combinatorial variant based on the use of the arithmetic mean of the predictions obtained for the values *k* = 2,3,5, as a middle way between the two extreme options. Since we are using a multi-step-ahaed method, we preferred the choice of MIMO method because it is the method which preserves “between forecasted values, the stochastics dependency characterizing the time series” [15] and ensures a good accuracy of the forecast. The results were obtained by using the functions of the software package *tsfknn* from R [13].

5. Results and discussions

In Figure 2 is represented the graph of the original time series and of the obtained forecasts
For a good estimation of the prediction capacity of the method presented in this case, it is useful to compare it with other methods. Thus, we chose for comparison the classical, very well known method ARIMA and other black-box method, based on Neural Networks. The values of the forecasts for 2019, obtained for the three methods are shown in Table 1.

**Table 1.** Forecasted values obtained for each of the used methods

<table>
<thead>
<tr>
<th>Month Year</th>
<th>knn Point Forecast</th>
<th>ARIMA Point Forecast</th>
<th>Neural Network Point Forecast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 2019</td>
<td>47.53450</td>
<td>53.14955</td>
<td>44.37076</td>
</tr>
<tr>
<td>Feb 2019</td>
<td>34.67377</td>
<td>52.42465</td>
<td>44.45416</td>
</tr>
<tr>
<td>Mar 2019</td>
<td>50.59274</td>
<td>55.64657</td>
<td>44.49235</td>
</tr>
<tr>
<td>Apr 2019</td>
<td>41.37621</td>
<td>54.91415</td>
<td>44.54306</td>
</tr>
<tr>
<td>May 2019</td>
<td>50.53533</td>
<td>56.66596</td>
<td>44.50243</td>
</tr>
<tr>
<td>Jun 2019</td>
<td>78.64142</td>
<td>58.05291</td>
<td>44.52037</td>
</tr>
<tr>
<td>Jul 2019</td>
<td>69.72496</td>
<td>55.49576</td>
<td>44.50670</td>
</tr>
<tr>
<td>Aug 2019</td>
<td>85.53458</td>
<td>55.65713</td>
<td>44.52357</td>
</tr>
<tr>
<td>Sep 2019</td>
<td>94.63957</td>
<td>56.74393</td>
<td>44.54284</td>
</tr>
<tr>
<td>Oct 2019</td>
<td>45.55664</td>
<td>57.48820</td>
<td>44.53600</td>
</tr>
<tr>
<td>Nov 2019</td>
<td>54.46762</td>
<td>56.18533</td>
<td>44.52579</td>
</tr>
<tr>
<td>Dec 2019</td>
<td>55.79163</td>
<td>51.7930</td>
<td>44.49329</td>
</tr>
</tbody>
</table>

Source: made by the author with results obtained in R

Forecast errors in the three cases studied were estimated using three indicators: RMSE (Root Mean Squared Error), MAE (Mean Absolute Error) and MAPE (Mean Absolute Percentage Error). To calculate these indicators values we used as training data the first 80% of the time series values, and the last 20% of the values as data test. The results are summarized in Table 2:

**Table 2.** Estimated values of the forecast errors

<table>
<thead>
<tr>
<th>Forecast error indicator</th>
<th>RMSE</th>
<th>MAE</th>
<th>MAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>knn</td>
<td>29.60</td>
<td>22.42</td>
<td>64.83</td>
</tr>
<tr>
<td>ARIMA</td>
<td>27.41</td>
<td>21.11</td>
<td>69.67</td>
</tr>
<tr>
<td>Neural Network</td>
<td>31.06</td>
<td>25.74</td>
<td>87.52</td>
</tr>
</tbody>
</table>

Source: made by the author with results obtained in R

Analyzing the data in Table 2 we see that, using the knn method we obtain smaller values than using Neural Network under all indicators. Comparing the results obtained when using knn and ARIMA methods, we note the following:
MAPE indicator value is smallest when using the $knn$ method;
RMSE and MAE indicators values are somewhat smallest when using the method ARIMA.

6. Conclusions and future work

In this paper we described the procedure through which $knn$ regression method can be used in order to obtain time series predictions and we applied this procedure to Average Monthly Rainfall time series forecast for the period 2016-2020. The accuracy of the forecasts obtained was compared with that obtained by using other two methods, the ARIMA method - representative of the traditional approach and the Neural Networks method - representative for the use of machine learning methods in predicting time series. From this analysis it can be concluded that, in this case study, the performance of $knn$ and ARIMA methods can be regarded as similar (with a slight advantage for the $knn$ method), while $knn$ method clearly outperforms the Neural Networks method. Given that, as previously noted, the ARIMA method is still commonly used to create rainfall forecast, we believe that the result obtained by using the $knn$ method is encouraging. This is important in order to find simpler data-driven alternatives from the field of machine learning. In this way, the utilization of other methods of machine learning and a comparative study of the performance obtained represent definitely one of the future directions of research. At the same time, the inclusion in the forecast model of other forecasting variables can substantially improve the quality of the predictions obtained.

7. Bibliography


Time Series Forecasting with KNN in R: the tsfknn Package, available at
https://cran.r-project.org/web/packages/tsfknn/vignettes/tsfknn.html, [accessed on 3.07.2018]
https://doi.org/10.1007/s10462-017-9593-z [accessed on 4.07.2018]
available at https://www.carbonbrief.org/analysis-the-most-cited-climate-change-papers, [accessed on
4.07.2018]
Analysis, Modeling and Forecasting in Kavala City, Greece, North-Eastern Mediterranean Basin,
International Conference on Efficient & Sustainable Water Systems Management toward worth
learning models and its application to Fukuoka city case, International Journal of Applied
Series by combining Direct and MIMO strategy, Proceedings of the 2009 IEEE International Joint
Conference on Neural Networks, Atlanta U.S.A., pp. 3054-3061, available at
1.1.&ie=utf-8&oe=utf-8&client=firefox-b. [accessed on 4.07.2018]
[16] Taieb, S., B., Sorjamaa, A., Bontempi, G., Multiple-output modeling for multistep-ahead time
https://www.researchgate.net/publication/265868920_Multiple-Output_Modelling_for_Multi-Step-
Ahead_Forecasting, [accessed on 4.07.2018]
156-161, July-September 2016, available at
https://www.researchgate.net/publication/307849752_Sectional_rainfall_forecast_for_cropping_patt
ern_planning_using_a_modified_k-nearest_neighbor_model, [accessed on 30.08.2018]
Assessment Report: Climate Change 2007
isCCode=ROU