

# Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background

Sadegh Bafandeh Imandoust And Mohammad Bolandraftar

Department of Economics, Payame Noor University, Tehran, Iran

## Abstract

In the present study  $k$ -Nearest Neighbor classification method, have been studied for economic forecasting. Due to the effects of companies' financial distress on stakeholders, financial distress prediction models have been one of the most attractive areas in financial research. In recent years, after the global financial crisis, the number of bankrupt companies has risen. Since companies' financial distress is the first stage of bankruptcy, using financial ratios for predicting financial distress have attracted too much attention of the academics as well as economic and financial institutions. Although in recent years studies on predicting companies' financial distress in Iran have been increased, most efforts have exploited traditional statistical methods; and just a few studies have used nonparametric methods. Recent studies demonstrate this method is more capable than other methods.

**Keywords:** Predicting financial distress- Machine learning-  $k$ -Nearest Neighbor.

## I. Introduction

### 1.1. Data mining In brief

Data mining is a process that uses a variety of data analysis tools to discover patterns and relationships in data that may be used to make valid predictions.

The first and simplest analytical step in data mining is to describe the data -summarize its statistical attributes (such as means and standard deviations), visually review it using charts and graphs, and look for potentially meaningful links among variables (such as values that often occur together).

Data mining takes advantage of advances in the fields of artificial intelligence (AI) and statistics. Both disciplines have been working on problems of pattern recognition and classification.

The increased power of computers and their lower cost, coupled with the need to analyze enormous data sets with millions of rows, have allowed the development of new techniques based on a brute-force exploration of possible solutions [1-5].

### 1.2. Different types of prediction using data mining techniques

(1) *Classification*: predicting into what category or class a case falls.

(2) *Regression*: predicting what number value a variable will have (if it is a variable that varies with time, it's called 'time series' prediction).

### 1.3. Classification

Classification problems aim to identify the characteristics that indicate the group to which each case belongs. This pattern can be used both to understand the existing data and to predict how new instances will behave. Data mining creates classification models by examining already classified data (cases) and inductively finding a predictive

pattern. These existing cases may come from a historical database, such as people who have already undergone a particular medical treatment or moved to a new long distance service. They may come from an experiment in which a sample of the entire database is tested in the real world and the results used to create a classifier. For example, a sample of a mailing list would be sent an offer, and the results of the mailing used to develop a classification model to be applied to the entire database. Sometimes an expert classifies a sample of the database, and this classification is then used to create the model which will be applied to the entire database [6-9].

### 1.4. Regression

Regression uses existing values to forecast what other values will be. In the simplest case, regression uses standard statistical techniques such as linear regression. Unfortunately, many real-world problems are not simply linear projections of previous values. For instance, sales volumes, stock prices, and product failure rates are all very difficult to predict because they may depend on complex interactions of multiple predictor variables. Therefore, more complex techniques may be necessary to forecast future values. The same model types can often be used for both regression and classification.

For example, the CART (Classification and Regression Trees) decision tree algorithm can be used to build both classification trees (to classify categorical response variables) and regression trees (to forecast continuous response variables).  $K$ -Nearest Neighbor method can create both classification and regression models as well.

There are varieties of data mining methods including Support Vector Machines (SVM), Artificial Neural

Networks (ANN), Naïve Bayesian Classifier, Genetic Algorithm, and K-Nearest Neighbor (KNN).

This paper aims to investigate KNN method in classification and regression, its historical background, and different applications of the method in several areas.

## II. Theoretical Background

### 2.1 KNN for classification

In pattern recognition, the KNN algorithm is a method for classifying objects based on closest training examples in the feature space. KNN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification [9-12].

The KNN is the fundamental and simplest classification technique when there is little or no prior knowledge about the distribution of the data [12-15]. This rule simply retains the entire training set during learning and assigns to each query a class represented by the majority label of its k-nearest neighbors in the training set. The Nearest Neighbor rule (NN) is the simplest form of KNN when  $K = 1$ .

In this method each sample should be classified similarly to its surrounding samples. Therefore, if the classification of a sample is unknown, then it could be predicted by considering the classification of its nearest neighbor samples. Given an unknown sample and a training set, all the distances between the unknown sample and all the samples in the training set can be computed. The distance with the smallest value corresponds to the sample in the training set closest to the unknown sample. Therefore, the unknown sample may be classified based on the classification of this nearest neighbor [15-20].

Figure 1 shows the KNN decision rule for  $K=1$  and  $K=4$  for a set of samples divided into 2 classes. In Figure 1(a), an unknown sample is classified by using only one known sample; in Figure 1(b) more than one known sample is used. In the last case, the parameter  $K$  is set to 4, so that the closest four samples are considered for classifying the unknown one. Three of them belong to the same class, whereas only one belongs to the other class. In both cases, the unknown sample is classified as belonging to the class on the left. Figure 2 provides a sketch of the KNN algorithm.

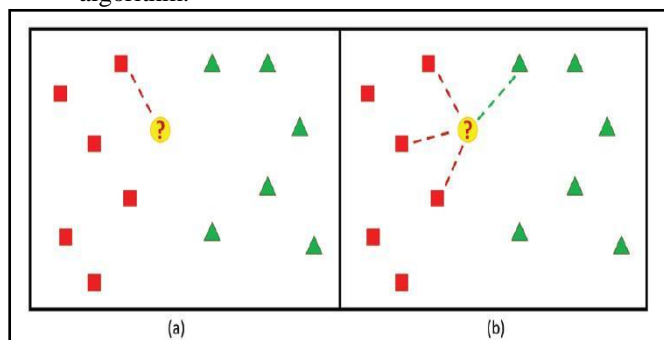


Figure 1. (a) The 1-NN decision rule: the point ? is assigned to the class on the left; (b) the KNN

decision rule, with  $K=4$ : the point ? is assigned to the class on the left as well

```

for all the unknown samples UnSample(i)
    for all the known samples Sample(j)
        compute the distance between
        UnSamples(i) and Sample(j)
    end for
    find the k smallest distances
    locate the corresponding samples
    Sample(j1),...,Sample(jk)
    assign UnSample(i) to the class which
    appears more frequently
end for
    
```

Figure 2. The KNN algorithm.

The performance of a KNN classifier is primarily determined by the choice of  $K$  as well as the distance metric applied [20-25]. The estimate is affected by the sensitivity of the selection of the neighborhood size  $K$ , because the radius of the local region is determined by the distance of the  $K$ th nearest neighbor to the query and different  $K$  yields different conditional class probabilities. If  $K$  is very small, the local estimate tends to be very poor owing to the data sparseness and the noisy, ambiguous or mislabeled points. In order to further smooth the estimate, we can increase  $K$  and take into account a large region around the query. Unfortunately, a large value of  $K$  easily makes the estimate over smoothing and the classification performance degrades with the introduction of the outliers from other classes. To deal with the problem, the related research works have been done to improve the classification performance of KNN.

How to select a suitable neighborhood size  $K$  is a key issue that largely affects the classification performance of KNN. As for KNN, the small training sample size can greatly affect the selection of the optimal neighborhood size  $K$  and the degradation of the classification performance of KNN is easily produced by the sensitivity of the selection of  $K$ . Generally speaking, the classification results are very sensitive to two aspects: the data sparseness and the noisy, ambiguous or mislabeled points if  $K$  is too small, and many outliers within the neighborhood from other classes if  $K$  is too large. From a theoretical point of view, the classification performance of KNN is determined by the estimate of the conditional class probabilities of the query in a local region of the data space, which is determined by the distance of the  $K$ th nearest neighbor to the query. So the classification performance is very sensitive to the selected value of  $K$ . Furthermore, the simplest majority voting of combining the class labels for KNN can be a problem if the nearest neighbors vary widely over their distances and the closer ones more reliably indicate the class of the query object. With the goal of addressing the sensitivity issue of different choices of the neighborhood size  $K$ , some

weighted voting methods have been developed for KNN.

It has been shown in when the points are not uniformly distributed; predetermining the value of K becomes difficult. Generally, larger values of K are more immune to the noise presented, and make boundaries smoother between classes. As a result, choosing the same (optimal) K becomes almost impossible for different applications.

## 2.2 KNN for Regression

### 2.2.1 Theory

The same method can be used for regression, by simply assigning the property value for the object to be the average of the values of its K nearest neighbors. It can be useful to weight the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.

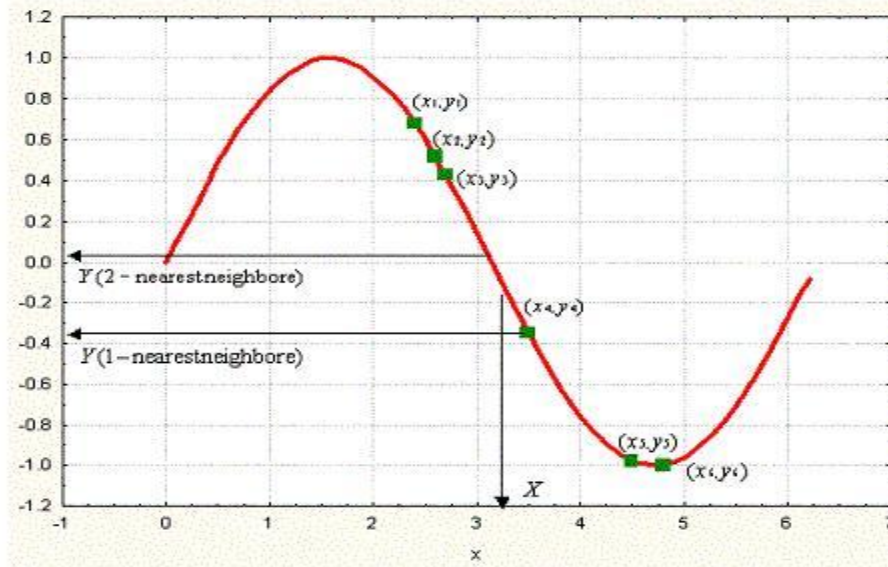


Figure 3. the KNN decision rule for regression

Regression problems are concerned with predicting the outcome of a dependent variable given a set of independent variables. To start with, we consider the schematic shown above in Figure 3, where a set of points (green squares) are drawn from the relationship between the independent variable  $x$  and the dependent variable  $y$  (red curve). Given the set of green objects (known as examples) we use the KNN method to predict the outcome of  $X$  (also known as query point) given the example set (green squares).

To begin with, let's consider the 1-nearest neighbor method as an example. In this case we search the example set (green squares) and locate the one closest to the query point  $X$ . For this particular case, this happens to be  $x_4$ . The outcome of  $x_4$  (i.e.,  $y_4$ ) is thus then taken to be the answer for the outcome of  $X$  (i.e.,  $Y$ ). Thus for 1-nearest neighbor we can write:

$$Y = y_4$$

For the next step, let's consider the 2-nearest neighbor method. In this case, we locate the first two closest points to  $X$ , which happen to be  $y_3$  and  $y_4$ . Taking the average of their outcome, the solution for  $Y$  is then given by:

$$Y = \frac{y_3 + y_4}{2}$$

The above discussion can be extended to an arbitrary number of nearest neighbors  $K$ . To summarize, in a KNN method, the outcome  $Y$  of the query point  $X$  is taken to be the average of the outcomes of its  $K$  nearest neighbors.

### 2.2.2 Distance Metric

As mentioned before KNN makes predictions based on the outcome of the  $K$  neighbors closest to that point. Therefore, to make predictions with KNN, we need to define a metric for measuring the distance between the query point and cases from the examples sample. One of the most popular choices to measure this distance is known as Euclidean. Other measures include Euclidean squared, City-block, and Chebychev.

$$D(x, p) = \begin{cases} \sqrt{(x - p)^2} & \text{Euclidian} \\ (x - p)^2 & \text{Euclidian squared} \\ |x - p| & \text{Cityblock} \\ \text{Max}(|x - p|) & \text{Chebychev} \end{cases}$$

where  $x$  and  $p$  are the query point and a case from the examples sample, respectively.

### 2.2.3 K-Nearest Neighbor Predictions

After selecting the value of  $K$ , you can make predictions based on the KNN examples. For regression, KNN prediction is the average of the  $K$  nearest neighbors outcome:

$$y = \frac{1}{K} \sum_{i=1}^k y_i$$

where  $y_i$  is the  $i$ th case of the examples sample and  $y$  is the prediction (outcome) of the query point. In contrast to regression, in classification problems,

KNN predictions are based on a voting scheme in which the winner is used to label the query. So far we have discussed KNN analysis without paying any attention to the relative distance of the K nearest examples to the query point. In other words, we let the K neighbors have equal influence on predictions irrespective of their relative distance from the query point. An alternative approach is to use arbitrarily large values of K (if not the entire prototype sample) with more importance given to cases closest to the query point. This is achieved using so-called 'distance weighting'.

#### 2.2.4 Distance Weighting

Since KNN predictions are based on the intuitive assumption that objects close in distance are potentially similar, it makes good sense to discriminate between the K nearest neighbors when making predictions, i.e., let the closest points among the K nearest neighbors have more say in affecting the outcome of the query point. This can be achieved by introducing a set of weights W, one for each nearest neighbor, defined by the relative closeness of each neighbor with respect to the query point. Thus:

$$W(x, p_i) = \frac{\exp(-D(x, p_i))}{\sum_{i=1}^k \exp(-D(x, p_i))}$$

where  $D(x, p_i)$  is the distance between the query point  $x$  and the  $i$ th case  $p_i$  of the example sample. It is clear that the weights defined in this manner above will satisfy:

$$\sum_{i=1}^k W(X_0, X_i) = 1$$

Thus, for regression problems, we have:

$$y = \sum_{i=1}^k W(X_0, X_i) y_i$$

For classification problems, the maximum of the above equation is taken for each class variables. It is clear from the above discussion that when  $K > 1$ , one can naturally define the standard deviation for predictions in regression tasks using,

$$err\ bar = \mp \sqrt{\frac{1}{K-1} \sum_{i=1}^k (y - y_i)^2}$$

### III. Advantages and Disadvantages

#### 3.1 Advantages

KNN has several main advantages: simplicity, effectiveness, intuitiveness and competitive classification performance in many domains. It is Robust to noisy training data and is effective if the training data is large.

#### 3.2 Disadvantages

Despite the advantages given above, KNN has a few limitations. KNN can have poor run-time performance when the training set is large. It is very sensitive to irrelevant or redundant features because

all features contribute to the similarity and thus to the classification. By careful feature selection or feature weighting, this can be avoided. Two other disadvantages of the method are:

- Distance based learning is not clear which type of distance to use and which attribute to use to produce the best results.
- Computation cost is quite high because we need to compute distance of each query instance to all training samples.

### IV. Historical Background

KNN classification was developed from the need to perform discriminant analysis when reliable parametric estimates of probability densities are unknown or difficult to determine. In an unpublished US Air Force School of Aviation Medicine report in 1951, Fix and Hodges introduced a non-parametric method for pattern classification that has since become known the k-nearest neighbor rule. They introduced a novel approach to nonparametric classification by relying on the 'distance' between points or distributions. The basic idea is to classify an individual to the population whose sample contains the majority of 'nearest neighbors. Later in 1967, some of the formal properties of the k-nearest-neighbor rule were worked out; for instance gave upper bounds for the limit of the risk of nearest neighbor classifiers. Once such formal properties of k-nearest-neighbor classification were established, a long line of investigation ensued including new rejection approaches, refinements with respect to Bayes error rate, distance weighted approaches, and soft computing methods.

Wagner and Fritz treated convergence of the conditional error rate when  $K = 1$ . Devroye and Wagner developed and discussed theoretical properties, particularly issues of mathematical consistency, for K-nearest-neighbor rules. Devroye found an asymptotic bound for the regret with respect to the Bayes classifier. Devroye et al. gave a particularly general description of strong consistency for nearest-neighbor methods. Psaltis, Snapp and Venkatesh generalized the results of Cover to general dimension, and Snapp and Venkatesh further extended the results to the case of multiple classes. Bax gave probabilistic bounds for the conditional error rate in the case where  $K = 1$ . Kulkarni and Posner addressed nearest-neighbor methods for quite general dependent data, and Holst and Irle provided formulae for the limit of the error rate in the case of dependent data. Related research includes that of Györfi and Györfi and Györfi, who investigated the rate of convergence to the Bayes risk when K tends to infinity as T increases.

Recent work on properties of classifiers focuses largely on deriving upper and lower bounds to regret in cases where the classification problem is relatively difficult, for example, where the classification boundary is comparatively unsmooth.



Research of Audibert and Tsybakov and Kohler and Krzyzak, for example, are in this category. The work of Mammen and Tsybakov, which permits the smoothness of a classification problem to be varied in the continuum, forms something of a bridge between the smooth case, which we treat, and the rough case.

## V. Applications

KNN as a data mining technique has a wide variety of applications in classification as well as regression. Some of the applications of this method are mentioned below:

### 5.1 Text mining

The KNN algorithm is one of the most popular algorithms for text categorization or text mining. Some of the most recent works on this topic are for instance. Different numbers of nearest neighbors are used for different classes in this approach, rather than a fixed number across all classes. In this way, the only parameter that needs to be chosen by the user when using KNN, the K value, becomes less sensible and hence it does not need to be carefully chosen as in the standard algorithm. Indeed, the probability that an unknown sample belongs to a class is computed by using only some top  $K_n$  nearest neighbors for that class. The  $K_n$  value is derived from K according to the size of the corresponding class in the training set. This modified KNN was efficient and less sensible to the K values when applied to text mining problems.

### 5.2 Agriculture

In general, KNN is applied less than other data mining techniques in agriculture related fields. It has been applied, for instance, for simulating daily precipitations and other weather variables. Another interesting application is the evaluation of forest inventories and for estimating forest variables. In these applications, satellite imagery is used, with the aim of mapping the land cover and land use with few discrete classes. The other applications of the k-NN method in agriculture include climate forecasting and estimating soil water parameters.

### 5.3 Finance

Data mining as a process of discovering useful patterns and correlations has its own niche in financial modeling. Similar to other computational methods almost every data mining method and technique has been used in financial modeling. An incomplete list includes a variety of linear and non-linear models multi-layer neural networks, k-means and hierarchical clustering, k-nearest neighbors, decision tree analysis, regression (logistic regression, general multiple regression), ARIMA, principal component analysis, and Bayesian learning.

Stock market forecasting is one of the most core financial tasks of KNN. Stock market forecasting includes uncovering market trends, planning investment strategies, identifying the best

time to purchase the stocks, and what stocks to purchase.

Some of other applications of KNN in finance are mentioned below:

- Forecasting stock market: Predict the price of a stock, on the basis of company performance measures and economic data.
- Currency exchange rate
- Bank bankruptcies
- Understanding and managing financial risk
- Trading futures
- Credit rating
- Loan management
- Bank customer profiling
- Money laundering analyses

### 5.4 Medicine

- Predict whether a patient, hospitalized due to a heart attack, will have a second heart attack. The prediction is to be based on demographic, diet and clinical measurements for that patient.
- Estimate the amount of glucose in the blood of a diabetic person, from the infrared absorption spectrum of that person's blood.
- Identify the risk factors for prostate cancer, based on clinical and demographic variables.

The KNN algorithm has been also applied for analyzing micro-array gene expression data, where the KNN algorithm has been coupled with genetic algorithms, which are used as a search tool. Other applications include the prediction of solvent accessibility in protein molecules, the detection of intrusions in computer systems, and the management of databases of moving objects such as computer with wireless connections.

## Acknowledgment

This research has been done by Payame Noor Financial grant.

## References

- [1] Audibert, J.Y. & Tsybakov, A.B. (2007) "Fast learning rates for plug-in classifiers under the margin condition", *Ann. Statist.*, 35: 608–633.
- [2] Bailey, T. & Jain, A. (1978) "A note on distance-weighted k-Nearest Neighbor rules", *IEEE Trans. Systems, Man, Cybernetics*, 8: 311-313.
- [3] Baoli, L., Shiwen, Y. & Qin, L. (2003) "An Improved k-Nearest Neighbor Algorithm for Text Categorization, ArXiv Computer Science e-prints.
- [4] Bauer, M.E., Burk, T.E., Ek, A.R., Coppin, P.R. Lime, S.D., Walsh, T.A., Walters, D.K., Befort, W. & Heinzen, D.F. (1994) "Satellite Inventory of Minnesota's Forest Resources", *Photogrammetric Engineering and Remote Sensing*, 60(3): 287–298.

- [5] Bax, E. (2000) "Validation of nearest neighbor classifiers", *IEEE Trans. Inform. Theory*, 46: 2746–2752.
- [6] Benetis, R., Jensen, C., Karciuskas, G. & Saltenis, S. (2006) "Nearest and Reverse Nearest Neighbor Queries for Moving Objects", *The International Journal on Very Large Data Bases*, 15(3): 229–250.
- [7] Bermejo, T. & Cabestany, J. (2000) "Adaptive soft k-Nearest Neighbor classifiers", *Pattern Recognition*, 33: 1999-2005.
- [8] Chitra, A. & Uma, S. (2010) "An Ensemble Model of Multiple Classifiers for Time Series Prediction", *International Journal of Computer Theory and Engineering*, 2(3): 1793-8201.
- [9] Cover, T.M. (1968) "Rates of convergence for nearest neighbor procedures", In *Proceedings of the Hawaii International Conference on System Sciences*, Univ. Hawaii Press, Honolulu, 413–415.
- [10] Cover, T.M. & Hart, P.E. (1967) "Nearest neighbor pattern classification", *IEEE Trans. Inf. Theory*, 13: 21–27.
- [11] Devroye, L. (1981) "On the asymptotic probability of error in nonparametric discrimination", *Ann. Statist.*, 9: 1320–1327.
- [12] Devroye, L. (1981) "On the equality of Cover and Hart in nearest neighbor discrimination", *IEEE Trans. Pattern Anal. Mach. Intell.* 3: 75–78.
- [13] Devroye, L., Györfi, L., Krzyżak, A. & Lugosi, G. (1994) "On the strong universal consistency of nearest neighbor regression function estimates", *Ann. Statist.*, 22: 1371–1385.
- [14] Devroye, L. & Wagner, T.J. (1977) "The strong uniform consistency of nearest neighbor density estimates", *Ann. Statist.*, 5: 536–540.
- [15] Devroye, L. & Wagner, T.J. (1982) "Nearest neighbor methods in discrimination, In *Classification, Pattern Recognition and Reduction of Dimensionality*", *Handbook of Statistics*, 2: 193–197. North-Holland, Amsterdam.
- [16] Domeniconi, C., Peng, J. & Gunopulos, D. (2002) "Locally adaptive metric nearest-neighbor classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 24(9): 1281–1285.
- [17] Dudani, S.A. (1976) "The distance-weighted k-nearest neighbor rule", *IEEE Transactions on System, Man, and Cybernetics*, 6: 325-327.
- [18] Eldestein, H.A. (1999) "Introduction to Data Mining and Knowledge Discovery", Two Crows Corporation, USA, ISBN: 1-892095-02-5.
- [19] Enas, G.G. & Choi, S.C. (1986) "Choice the smoothing parameter and efficiency of K-Nearest Neighbor classification", *Comp & Maths with Apps*, 12(2): 235-244.
- [20] Fix, E. & Hodges, J.L. (1951) "Nonparametric Discrimination: Consistency Properties", Randolph Field, Texas, Project 21-49-004, Report No. 4.
- [21] Fritz, J. (1975) "Distribution-free exponential error bound for nearest neighbor pattern classification", *IEEE Trans. Inform. Theory*, 21: 552–557.
- [22] Fukunaga, K. & Hostetler, L. (1975) "k-nearest-neighbor Bayes risk estimation", *IEEE Trans. Information Theory*, 21(3): 285-293.
- [23] Gil-Garcia, R. & Pons-Porrata, A. (2006) "A New Nearest Neighbor Rule for Text Categorization", *Lecture Notes in Computer Science 4225*, Springer, New York, 814–823.
- [24] Guo, G., Wang, H., Bell, D., Bi, Y. & Greer, K., (2006) "Using KNN Model for Automatic Text Categorization", *Soft Computing – A Fusion of Foundations, Methodologies and Applications 10(5)*: 423–430.
- [25] Gou, J., Du, L. Zhang, Y. & Xiong, T. (2012) "A New Distance-weighted k-nearest Neighbor Classifier", *Journal of Information & Computational Science*, 9(6): 1429-1436.
- [26] Györfi, L. (1978) "On the rate of convergence of nearest neighbor rules", *IEEE Trans. Inform. Theory*, 24: 509–512.
- [27] Györfi, L. (1981) "The rate of convergence of k-NN regression estimates and classification rules", *IEEE Trans. Information Theory*, 27: 362–364.
- [28] Györfi, L. & Györfi, Z. (1978) "An upper bound on the asymptotic error probability of the k-Nearest Neighbor rule for multiple classes", *IEEE Trans. Inform. Theory*, 24: 512–514.
- [29] Hall, P., Park, B.U. & Samworth, R.J. (2008) "Choice of neighbor order in nearest-neighbor classification", *the Annals of Statistics*, 36(5): 2135-2152.
- [30] Hastie, T., Tibshirani, R. & Friedman, J. (2009) "The Elements of Statistical Learning: Data Mining, Inference and Prediction", Springer, Stanford, CA, USA, ISBN: 978-0-387-84858-7.
- [31] Hellman, M.E., (1970) "The nearest neighbor classification rule with a reject option", *IEEE Trans. Systems, Man, Cybernetics*, 3: 179-185.
- [32] Hill, T. & Lewicki, P. (2007) "Statistics: Methods and Applications", Statsoft, Tulsa, OK. (Electronic version is available at: [www.statsoft.com/textbook/k-nearest-neighbors/](http://www.statsoft.com/textbook/k-nearest-neighbors/))