A Comparative Analysis of the Neural Network and Naive Bayes Classifiers

Ivan Belik
Norwegian School of Economics, Norway

Abstract
The main purpose of the given research is to make a comparative analysis of two classifiers with different nature. Specifically, we build the Naïve Bayes Classifier, based on the Bayesian Statistics concept, and the Artificial Neural Network classifier, based on the functional aspects of the biological neural networks. These two competing paradigms are running over ten-real world datasets to show the most possible classification accuracy results.

Keywords
Classification problem; Bayesian Statistics; Neural Networks

1. Introduction

The problem of the classification has a wide area of application [1]. The statistical analysis is purposed for understanding the nature of different social, economic, technical events and processes. Its primary application is prediction and classification [2, 3]. Due to the requirement of the high-level accuracy of the classification in many real-world practical problems there exists many mechanisms of the classification, which have a very different nature of their implementation [4, 5].

Statistical classifiers and machine learning classifiers have different ways of implementing the classification procedure [6]. In the given research, we compare machine learning based and statistics based approaches. We consider two classifiers with different nature and compare them in terms of their basic goal – the correctness and accuracy of the classification procedure.

Specifically, we are interested to consider Bayesian Statistics concept (specifically, Naïve Bayes classifier) vs. Artificial Neural Networks concept (specifically, Multilayer Perceptron classifier).

2. Methods

2.1 Classifiers training and testing mechanisms

Assume that we have a dataset with *m* instances. In the given research we adapt three testing mechanisms for the Naïve Bayes Classifier and the Multilayer Perceptron:

- Full training set

  Initially, the classifier is built based on the overall *m* instances. Then, the resulted classifier is applied to the given *m* instances for the classification purpose. The given method shows high results of the correctness, but it does not correspond to the real-world testing conditions [7].

- N-fold cross-validation

  The mechanism is based on run over *n* sets, where each of them has a size of *m/n* with no data overlaps [8]. The cross-validation mechanism is repeated *n*-times: *n*-1 sets are used for training and one set is used for testing. Each of *n* sets is used only once as a testing set over *n*-iterations.

Corresponding author:
Ivan Belik, Norwegian School of Economics, Helleveien 30, 5045 Bergen, Norway
Email: ivan.belik@nhh.no

- Percentage split (n %)

  This mechanism is based on the idea of the random split of dataset into two parts: *n* % of the dataset is applied for the classifier training and (*n*-1) % is applied for the classification testing [9].

We employ the given testing mechanisms and apply them to ten datasets retrieved from [10].

2.2 The Naïve Bayes Classifier (NBC)

2.2.1 Bayes' Theorem

A naive Bayes classifier is a classification method, which is based on the Bayesian Statistics [11, 12]. Bayes' Theorem is at the core of NBC. It allows working with cause and effect events, i.e. based on the known factor (cause event) we can calculate the probability of the effect event. The cause events are called hypothesizes, because they are "presumed" events, which are causing some effect event. The unconditional probability of the hypothesis is called as a prior probability. The conditional probability of the event (i.e. taking into consideration that caused event occurred) is called as a posterior probability.

The formalization of the Bayes' Theorem is represented in (1).

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}, \tag{1}$$

where

$P(A)$ is a prior probability of the event *A*;

$P(A|B)$ is a posterior probability of the event *A*, taking into consideration that event *B* occurred;

$P(B|A)$ is a probability of the event *B*, taking into consideration that event *A* occurred;

$P(B)$ is a probability of the event *B*.

Considering *A* and *B* as the random variables (i.e. $\{A=a\}$ and $\{B=b\}$, respectively) we have the following cases:

(1) *A* and *B* are continuous variables:

$$f_A(a|B=b) = \frac{f_B(b|A=a)f_A(a)}{f_B(b)} \tag{2}$$

(2) *A* is a continuous variable and *B* is a discrete variable:

$$f_A(a|B=b) = \frac{P(B=b|A=a)f_A(a)}{P(B=b)} \tag{3}$$

(3) *A* is a discrete variable and *B* is a continuous variable:

$$P(A=a|B=b) = \frac{f_B(b|A=a)P(A=a)}{f_B(b)} \tag{4}$$

In (2) – (4) f_A and f_B are the probability densities.

2.2.2 Classification mechanism based on the Bayes' Theorem

Bayes' Theorem is widely used for the classification purposes. More precisely, if we assume that we have an object *O* to be classified, and we have a set of classes *C={c1,…,ck}*, then we have to find the class *c*, which will have the maximum probability for the given *O*. We can formalize it in the following way:

$$c = \underset{C}{argmax}\ p(C|O) \tag{5}$$

Object O is characterized by the set of features $\{F_1, \dots, F_n\}$, and then we can apply the Bayes' Theorem to compute $p(C|O)$:

$$p(C|F_1, \dots, F_n) = \frac{p(C)p(F_1, \dots, F_n|C)}{p(F_1, \dots, F_n)} \tag{6}$$

Since we are looking for the maximum of function then we are not interested in the denominator. In other words, (2) can be rewritten in the follow way:

$$posterior = \frac{prior * likelihood}{evidence} \tag{7}$$

The important feature of the Naïve Bayes Classifier is the "naive" assumption that the features $\{F_1, \dots, F_n\}$ depend only on *C*, and do not depend on each other: $p\left(F_i \middle| C, F_j\right) = p\ (F_i|C)$.

As the result, we have:

$$c = \underset{c \in \mathrm{C}}{argmax}\ p(c|F_1, \dots, F_n) = = \underset{c \in \mathrm{C}}{argmax}\ p(\mathrm{c}) \prod p(\,F_1, \dots, F_n|c) \tag{8}$$

The training procedure of the Naïve Bayes Classifier is based on the calculation of *p(C)* and $p(F_1, \dots, F_n|C)$ following the given training dataset.

Despite the naïve form and the simplified conditions, Naïve Bayes Classifier can show high results in terms of the efficiency in many real-life classification problems [13]. An advantage of the Naïve Bayes Classifier is that it requires small datasets for training comparing to other classification methods.

2.2.3 Illustrative explanation of the NBC classification

Initially, we consider a small dataset "weather.nominal" [10] to show how we implement the classification mechanisms.

According to the given dataset, it is required to decide, whether there exist appropriate weather conditions to play football outside. Weather conditions and the decision are characterized by five attributes (see Table 1):

Table 1. Attributes and the corresponding features

#	Attributes	Features
1	outlook	sunny overcast rainy
2	temperature	hot mild cool
3	humidity	high normal
4	windy	true false
5	play	yes no

Also, there are 14 days (i.e. instances) that represent the training dataset (see Table 2).

Table 2. Training dataset

#	outlook	temperature	humidity	windy	Play
1	sunny	hot	high	false	no
2	sunny	hot	high	true	no
3	overcast	hot	high	false	yes
4	rainy	mild	high	false	yes
5	rainy	cool	normal	false	yes
6	rainy	cool	normal	true	no
7	overcast	cool	normal	true	yes
8	sunny	mild	high	false	no
9	sunny	cool	normal	false	yes
10	rainy	mild	normal	false	yes
11	sunny	mild	normal	true	yes
12	overcast	mild	high	true	yes
13	overcast	hot	normal	false	yes
14	rainy	mild	high	true	no

According to Table 2, the decision (play or do not play) is made based on the weather conditions.

Now, we assume that we have a new instance with the unknown decision (play or do not play), which we have to classify (see Table 3).

Table 3. New instance

#	outlook	temperature	humidity	windy	Play
15	sunny	hot	normal	false	n/a

Applying our NBC theoretical approach for the current example, we have the following interpretation. *O* is an object "weather conditions to play football outside", and it is characterized by the following attributes:

F_1 – "outlook"
F_2 – "temperature"
F_3 – "humidity"
F_4 – "windy"

The given object *O* (i.e. instance #15) has to be classified based on the set of classes *C={c1, c2}={"play=yes", "play=no"}* corresponding to the decision to play or do not play football outside. Therefore, it is required to calculate the posterior $p\left(c_1 \middle| F_1, F_2, F_{3,} F_4\right)$ and $p\left(c_2 \middle| F_1, F_2, F_{3,} F_4\right)$ following (6) – (7). On the final step, we have to choose the highest between these two probabilities to make a decision whether to play or do not play football.

For the posterior probabilities, we have to calculate four components:

(1) The *prior* probability of "play football": $p(c_1)$.

It is calculated based on the training dataset represented in Table 2. There are 9 out of 14 instances that classified as "yes" (i.e., play football):

$$p(c_1) = \frac{9}{14} \approx 0.6428$$

The prior probability of "do not play football": $p(c_2)$.

$$p(c_2) = \frac{5}{14} \approx 0.3571$$

(2) *Likelihood* of "play football": $p(F_1, F_2, F_{3,}F_4|c_1)$.

Following instance #15, which should be classified, the attribute F_1 – "outlook" has a "sunny"-feature. According to Table 2, we have only two instances that simultaneously have "outlook = sunny" and classified as "play=yes" (i.e. instances #9 and #11). It means that if we consider the "outlook"-attribute only then the likelihood of playing football is equal to 2/9 (i.e. $p(F_1|c_1)$=2/9).

To calculate $p(F_1, F_2, F_{3,}F_4|c_1)$ we have to take into consideration all features $\{F_1, F_{2,}F_3, F_4\}$:

$$p(F_1, F_2, F_{3,}F_4|c_1) = \frac{2}{9} * \frac{2}{9} * \frac{6}{9} * \frac{6}{9} \approx 0.0219$$

(3) *Likelihood* of "do not play football": $p(F_1, F_2, F_{3,}F_4|c_1)$. Based on Table 2 we have the following result:

$$p(F_1, F_2, F_{3,}F_4|c_2) = \frac{3}{5} * \frac{2}{5} * \frac{1}{5} * \frac{2}{5} \approx 0.0192$$

(4) *Evidence*: $p(F_1, \dots, F_n)$.

$$p(F_1, F_2, F_{3,}F_4) = p(c_1) \ * p(F_1, F_2, F_{3,}F_4|c_1) + p(c_2) * p(F_1, F_2, F_{3,}F_4|c_2)$$

$$\mathrm{p}(\mathrm{F}_1, \mathrm{F}_2, \mathrm{F}_{3,}\mathrm{F}_4) \ \approx 0.6428 * 0.0219 + 0.3571 * 0.0192 \approx 0.0209$$

Based on (6), the posterior probability of "play football" is twice larger than the probability of "do not play football":

$$p(c_1|F_1, F_2, F_{3,}F_4) \approx \frac{0.6428 * 0.0219}{0.0209} \approx 0.6735$$

$$p(c_2|F_1, F_2, F_{3,}F_4) \approx \frac{0.3571 * 0.0192}{0.0209} \approx 0.328$$

Thus, following the given results the new instance (i.e. instance #15) should be classified as "play football".

The described classification mechanism is applicable for all instances, which are out of the training set, using three training and testing mechanisms described in section 2.1.

2.2.4 NBC computational issues and practical usage

The main computational issue of NBC, applied for the real-world problems, is the non-static nature of the new instances that has to be classified. It is a common situation, when NBC has to deal with new instances that have the unknown attributes and features. In other words, the new instance, that has to be classified, can have attributes which are not represented in the training dataset. It leads to the "false alarms" during the classification. For example, in our "weather" example, the instance with new attribute "football field condition" characterized by the feature "bad" can appear. This attribute does not exist in the training dataset (Table 2), and new instance can be incorrectly classified as "play football" instead of "do not play football".

In this case, to keep the accuracy of NBC on the desired level, it is required to update the training dataset. Sequentially, it leads to its significant growth. As the result, we can lose one of the basic NBC advantages: comparatively small datasets required for training NBC.

Another important problem is called an NBC "overtraining". This is the situation when the equilibrium between the number of instances correctly classified (and included to the training dataset) and the number of instances incorrectly classified (and also included to the training dataset) is broken. This leads to the avalanche growth of the “false alarms” that affects the accuracy of the classification process. NBC is often used in anti-spam filtering, and “false alarms” is a regular problem for many anti-spam systems that are based on NBC [14].

2.3 The Artificial Neural Networks Classifier

2.3.1 Bayes’ Theorem Rosenblatt Perceptron

The Perceptron concept is at the core of any artificial neural network model. The trivial perceptron consists of three basic types of elements [16]:

- S-elements. It is a layer of sensors (i.e. receptors). The real-world example of sensors is photoresistors in photo cameras. Each sensor can have one of two states: active and non-active. If sensor is in active state, it transmits the single signal “+1” or “-1” to the next layer, which is called A-elements.
- A-elements (“associative” layer). A trivial A-element is a decision logic element. It calculates the sum of all input signals from S-elements. If this sum is greater than a priori specified threshold value Θ, then it gives a positive output signal “+1” (i.e. S-element becomes active). Otherwise, the output signal is equal to zero.
- Single R-element. Signals from the active A-elements are transmitting to the R-adder. Each input signal from i-th A-element has a coefficient w_i. This coefficient is called the A-R weight. R-element computes the total sum of the input signals multiplied by the corresponding A-R weights (linear form). Finally, R-element generates the output “1” (i.e. the perceptron output) if the linear form is greater than some threshold value. Otherwise, the output is equal to “-1”.

The core idea of the perceptron training is based on the update of w_i coefficients in A-R links until the correct perceptron reaction is approached following the training dataset.

The trivial classification mechanism is the following. Perceptron processes the unknown instance, which has to be classified, by its transmission from the activated S-elements to the A-layer. Next, A-elements transmit the signal to the R-element. This signal is a sum of the corresponding weights. If the sum is positive, then the unknown instance is classified as some class #1. Otherwise, the unknown instance is classified as class #2.

The generalized Rosenblatt Perceptron structure is represented in Figure 1.

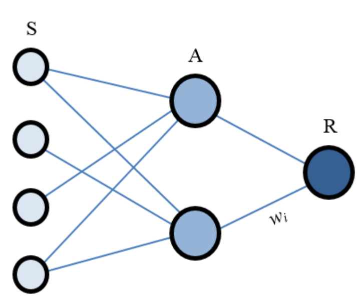


Figure 1. Rosenblatt Perceptron

2.3.2 Multilayer perceptron

In the given research the artificial neural network model, called Multilayer Perceptron (MLP), is used as the classifier that is alternative to the Naïve Bayes classifier [17]. MLP is a special case of Rosenblatt Perceptron [18], where the backpropagation algorithm [19] trains all layers of nodes (including S-A layers).

MLP consists of the multiple layers (i.e. more than one “associative” layers) encapsulated in the directed graph. Layers consist of the perceptrons (excepting the input nodes), which have the nonlinear activation functions used for the threshold values computation. Each non-linear activation function has to be differentiable. This makes MLP different from the Rosenblatt’s perceptron [15], which is characterized by the linear transformation of the input data. The most commonly used non-linear activation function is Sigmoid that has the following format [15]:

$$out = \frac{1}{1 + exp(-\alpha Y)}, \tag{9}$$

where the change of the angle of slope corresponds to the change of α – parameter.

Each layer is fully connected to the next layer in the graph, and each synaptic node-to-node (i.e. perceptron-to-perceptron) connection is characterized by the corresponding weight.

MLP consists of three basic elements:

(1) Input nodes (input S-layer)
(2) Hidden layer (more than one A-layers)
(3) Output layer (R-layer)

MLP contains one or more inner hidden layers, which are not a part of the network’s input or output. Hidden layers are responsible for the network’s training based on the backpropagation method. It is a supervised learning method, which represents the modified gradient descent algorithm [20]. The basic idea here is to forward the error signals, calculated in the output layers of the perceptrons, back to the inputs of the perceptrons, layer by layer, following the idea of the gradient descent.

The quantity of input and output elements (in the input and output layers, respectively) depends on the specific classification problem. An important feature of MLP is high connectivity of neurons (i.e. perceptrons). Any change in connectivity level requires the change in the synaptic connections and weights.

2.3.3 Multilayer perceptron

Using WEKA software [10], we get the training and classification results for dataset “weather.nominal” represented in Table 4.

Table 4. Classification results

Test options	Correctly classified (%)	Incorrectly classified (%)
Full training set	100	0
10-fold-cross validation	71.4286	28.5714
5-fold-cross validation	64.2857	35.7143
Percentage split (65 %)	60	40

The general structure of the MLP is represented in Figure 2.

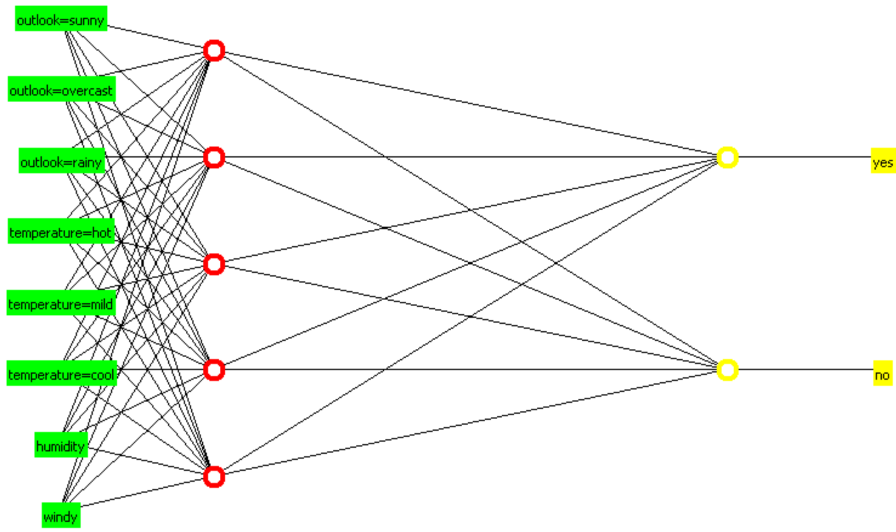


Figure 2. General structure of the illustrative dataset's MLP

The detailed structure of the generated MLP is represented in Appendix A.

2.3.4 MLP computational issues and practical usage

MLP is used in many real-world problems, such as economics, medicine, avionics and etc. It shows high level results solving many complicated classification and prediction problems [16]. However, MLP is not a universal and ideal apparatus, and it has its own limitations. The most significant issue of MLP was analyzed in [21]. It shows that problems, that have to be solved by perceptron, can take significantly large computational time and significantly large memory capacity. For example, there exist classification problems where the coefficients of the "associative" layer (i.e. A-elements) has to be so large that for their storage (in the computer memory) it will be required more memory than for the trivial memorization of all objects' classes. This issue also affects the efficiency of the back propagation method. Significantly large output values, which have to be sent back to the inputs of the perceptrons, can lead to the effect of the "paralysis" of the network freezing the classification process.

3. Results

For approaching the most realistic classification results, ten datasets from different real-world areas are chosen for classification [10]. The percentile of the correctly and incorrectly classified instances are presented in Appendices B and C.

The highest classification accuracy was achieved applying the "Full training set"-mechanism. The average performance for the MLP is equal to 92.86% and for NBC is equal to 75.12%. This is an expected result: the exploitation of the full training set both for training and testing gives a very high accuracy, but this type of testing has an "artificial" nature. It is applicable for the analysis of classifiers' mechanisms, but it does not show realistically how they work in the real-world applications dealing with data, which is characterized by highly incomplete information [22]. The overall classification results are represented in Figure 3.

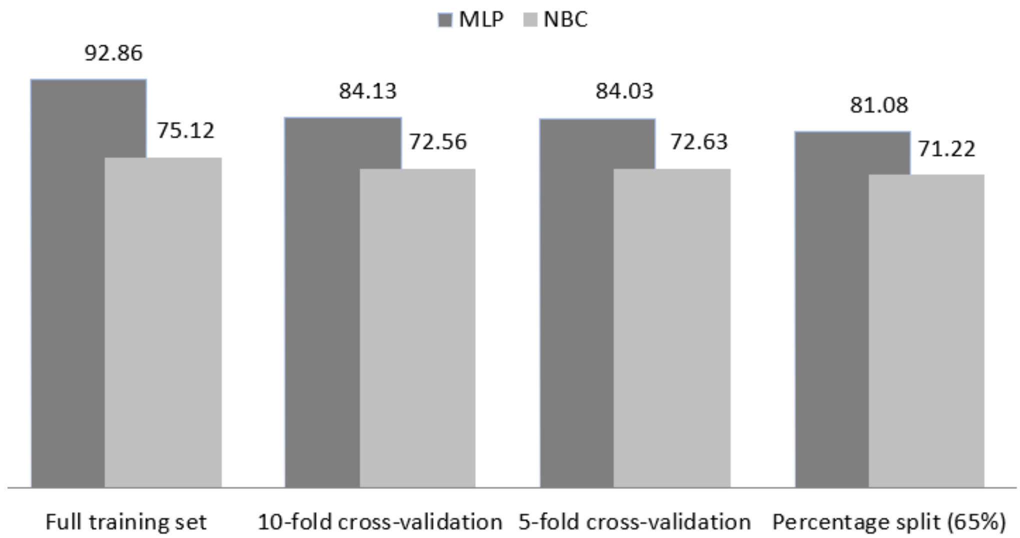
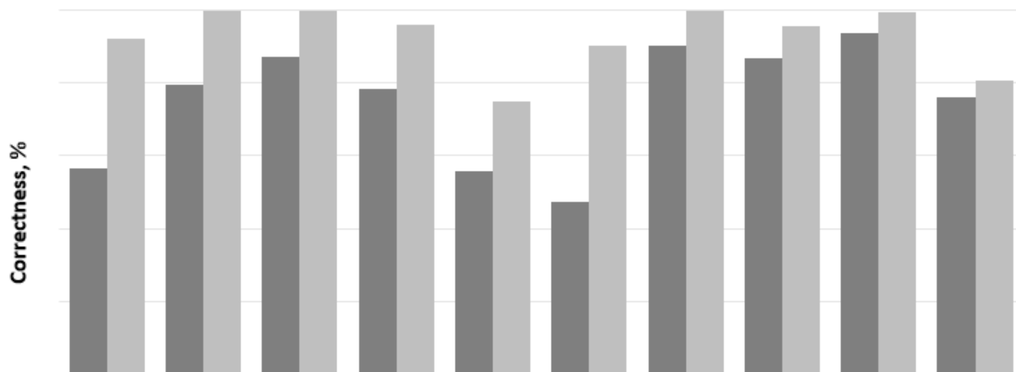


Figure 3. The average classification performance over ten datasets

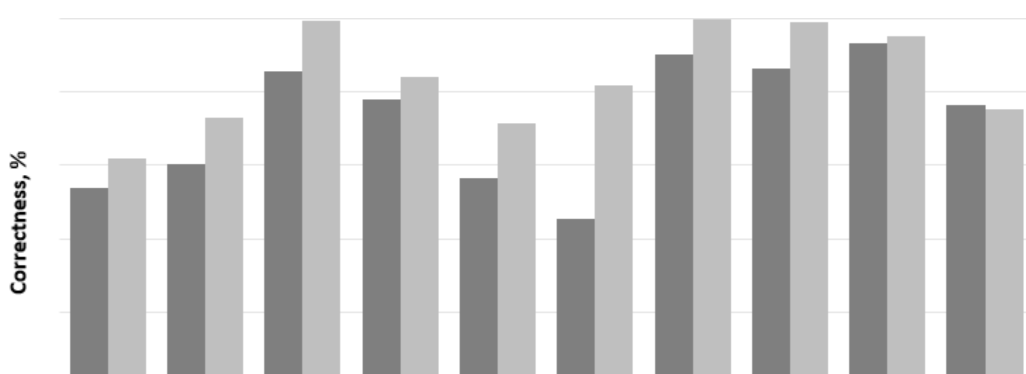
Based on Figure 3, 10-fold cross-validation and 5-fold cross-validation show almost equal accuracies for both test mechanisms: approximately, 84% and 72% for MLP and NBC, respectively. This is explained by the application of cross-validation mechanism for both tests with five and ten folds. The lowest classification accuracy is achieved by the "Percentage split (65%)"-mechanism: 81.08% for MLP, and 71.22% for NBC. These results can be explained by the nature of test, which is (comparatively) the closest to the real-world conditions. Both classifiers take 65% of the dataset for training and only 35% - for testing. The nature of the testing instances of the chosen 35% can be much different from the 65% training instances. This fact makes a big challenge for the classifiers to perform well, and the approached results are good even compared to the "Full training set"-mechanism results.

The detailed performance of the classifiers over ten datasets is represented in Figures 4-7.



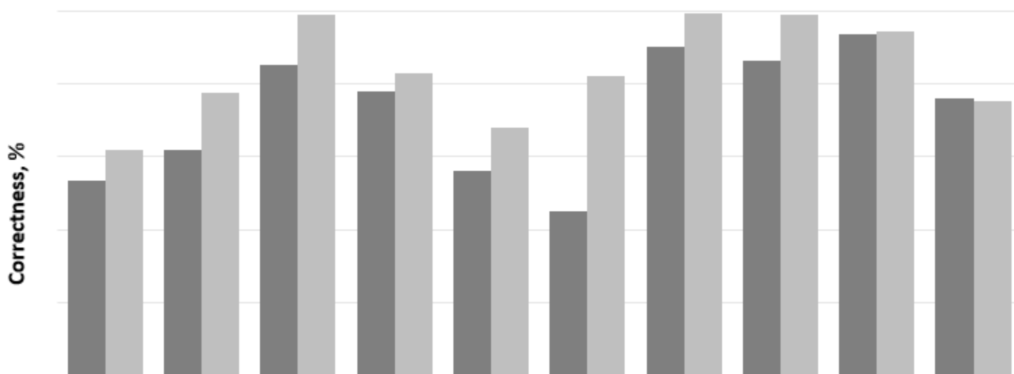
	eucalyptus	white-clover	car	credit-a	liver-disorders	vehicle	nursery	anneal	solar-flare_1	diabetes
NBC (%)	56.3859	79.3651	87.0949	78.2609	55.942	47.2813	90.3009	86.637	93.808	76.1719
MLP (%)	91.9837	100	100	96.087	74.7826	90.3073	99.9846	95.4343	99.3808	80.599

Figure 4. The classification results applying “Full training set”-mechanism



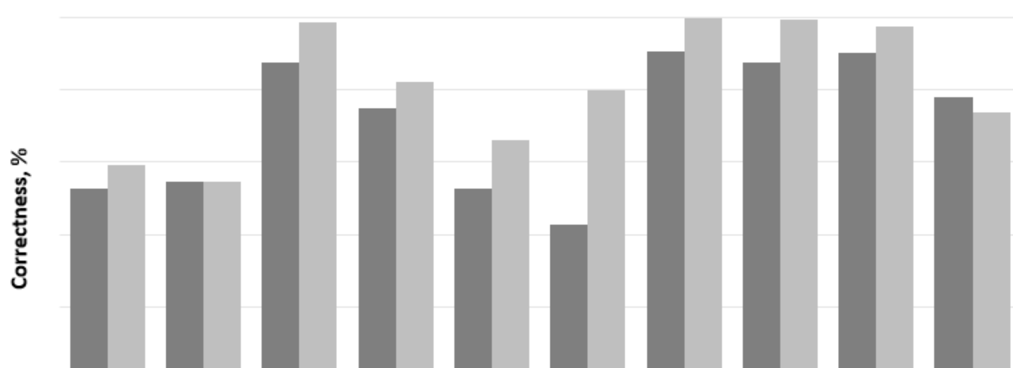
	eucalyptus	white-clover	car	credit-a	liver-disorders	vehicle	nursery	anneal	solar-flare_1	diabetes
NBC (%)	54.0136	60.3175	85.5324	77.8261	56.5217	45.2719	90.3241	86.3029	93.1889	76.3021
MLP (%)	61.9565	73.0159	99.537	84.2029	71.5942	81.6785	99.7299	98.8864	95.356	75.3906

Figure 5. The classification results applying “10-fold cross-validation”-mechanism



	eucalyptus	white-clover	car	credit-a	liver-disorders	vehicle	nursery	anneal	solar-flare_1	diabetes
NBC (%)	53.4694	61.9048	85.1852	77.8261	56.2319	45.1537	90.2469	86.5256	93.808	75.9115
MLP (%)	61.8207	77.7778	99.0162	83.0435	68.1159	82.3877	99.4753	99.1091	94.4272	75.1302

Figure 6. The classification results applying “5-fold cross-validation”-mechanism



	eucalyptus	white-clover	car	credit-a	liver-disorders	vehicle	nursery	anneal	solar-flare_1	diabetes
NBC (%)	52.7132	54.5455	87.6033	75.1037	52.8926	42.9054	90.4982	87.5796	90.2655	78.0669
MLP (%)	59.3023	54.5455	98.6777	82.1577	66.1157	79.7297	99.9559	99.3631	97.3451	73.6059

Figure 7. The classification results applying “Percentage split (65%)”-mechanism

Following the results represented in Figures 4-7, we conclude that the general performance on MLP is better than on NBC. However, we should consider the implementation complexity of both classifiers. MLP has a more complicated structure and implementation mechanism, which is also more time-consuming comparing to NBC. Based on the simple probabilistic laws and statistics nature, NBC shows the accuracy results that are appropriate in many real-life problems. In other words, it is convenient to exploit NBC for the purposes, where the high-level accuracy is not expected, but the simple and non-time consuming implementation is required.

An important aspect is time required for the classification. It becomes even more critical, when the datasets are extremely large, and the threshold for the classification accuracy is small enough to be satisfied with NBC application. Training and testing processes are running much faster on NBC than on MLP, and NBC is comparatively better in terms of the classification time consumption.

Another important aspect of the classification is a size of the dataset. The biggest dataset “nursery” contains 12960 instances. Consequently, it gives the highest classification accuracy results: 90.25-90.45% and 99.47-99.98% for NBC and MLP, respectively. The second largest dataset “car” contains 1728 instances, but the classification results are still comparatively high: 85.18-87.6% and 98.67-100% for NBC and MLP, respectively.

The approached testing results show that the simplicity and fast classification running time of NBC in combination with the large enough dataset exploited gives the classification mechanism, which is effective and appropriate in many real-world classification problems. MLP shows higher results than NBC, but we should consider the more complicated construction of MLP and slower classification running time, which might be critical for many real-world problems.

4. Conclusion

In the given research, we constructed, trained and exploited two classifiers with different nature. First, we described the theoretical bases for both NBC and MLP. It includes the general explanation of the algorithmic and mathematical concepts for both classifiers.

Next step, we realized the rigorous search for the datasets that correspond to the real-world problems. Ten datasets with the varying sizes and from different application areas were retrieved for the classification procedure.

In the experimental part of the research we constructed and explained the classification mechanisms based on the relatively small dataset, called “weather.nominal”. Specifically, we showed the classification procedure of the new unknown instance applying Naïve Bayes Classifier. Also, we constructed the Multilayer perceptron with the detailed description of the sigmoid functions for each node of the resulted artificial neural network.

Next, we applied the NBC and MLP classifiers for ten datasets using four training and testing mechanisms:

- Full training set
- 10-fold cross-validation
- 5-fold cross-validation
- Percentage split (65%)

The classification results have been analyzed in the different aspects. Even though MLP showed the higher classification accuracy, it was found that NBC takes less time for the classification procedure. Furthermore, NBC shows high classification accuracy dealing with the large datasets (i.e. more than ten thousand instances, but size might vary depending on the specific problem). The main advantage of the NBC is its simplicity of constructing. It gives an efficient, fast and appropriate classifier for many real-world problems.

References

[1] Huang GB, Zhu QY, Siew CK. Extreme learning machine: Theory and applications. Neurocomputing. 2006; 70 (1-3): 489–501.

[2] Fisher NI. Statistical analysis of circular data. Cambridge University Press, 1995.

[3] Little RJA, Rubin DB. Statistical analysis with missing data. New York: Wiley; 1987.

[4] Pereira F, Mitchell T, Botvinick M. Machine learning classifiers and fMRI: A tutorial overview. NeuroImage. 2009; 45(1).

[5] Soysal M, Schmidt EG. Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison. Performance Evaluation. 2010; 67(6): 451–67.

[6] Carnahan B, Meyer G, Kuntz LA. Comparing Statistical and Machine Learning Classifiers: Alternatives for Predictive Modeling in Human Factors Research. Human Factors: The Journal of the Human Factors and Ergonomics Society. 2003; 45(3): 408–23.

[7] Lewis DD. A sequential algorithm for training text classifiers. ACM SIGIR Forum. 1995 Jan; 29(2): 13–9.

[8] Lenth RV, Hjorth JSU. Computer Intensive Statistical Methods: Validation, Model Selection, and Bootstrap. Technometrics. 1995; 37(4): 458.

[9] Holte RC. Very simple classification rules perform well on most commonly used datasets. Machine learning. 1993; 11(1), 63–90.

[10] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software. ACM SIGKDD Explorations Newsletter. 2009;11(1):10.

[11] Rish I. An empirical study of the naive Bayes classifier. IJCAI 2001 workshop on empirical methods in artificial intelligence. IBM New York, 2001; 3(22): 41-46.

[12] Larsen K. Generalized Naive Bayes Classifiers. ACM SIGKDD Explorations Newsletter. 2005 Jan; 7(1): 76–81.

[13] Kazmierska J, Malicki J. Application of the Naïve Bayesian Classifier to optimize treatment decisions. Radiotherapy and Oncology. 2008; 86(2): 211–6.

[14] Zhang Z. A Bayesian Topic Model for Spam Filtering. Journal of Information and Computational Science. 2013 Oct; 10(12): 3719–3727.

[15] Rosenblatt F. The perceptron, a perceiving and recognizing automaton. Buffalo, NY: Cornell Aeronautical Laboratory; 1957.

[16] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review. 1958; 65(6): 386–408.

[17] Haykin SS. Neural networks: a comprehensive foundation. Upper Saddle River, NJ: Prentice Hall; 1999.

[18] White BW, Rosenblatt F. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. The American Journal of Psychology. 1963; 76(4): 705.

[19] Yam Y, Chow T. Extended backpropagation algorithm. Electronics Letters. 1993; 29(19): 1701.

[20] Snyman JA. Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms. New York: Springer; 2005.

[21] Mycielski J. Book Review: Perceptrons, An Introduction to Computational Geometry. Bulletin of the American Mathematical Society. 1972 Jan; 78(1): 12–6.

[22] Libkin L. Data exchange and incomplete information. InProceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems 2006 Jun 26 (pp. 60-69). ACM.

Appendix A. MLP structure

Sigmoid Node 0

Inputs	*Weights*
Threshold	-4.597967080790813
Node 2	2.433270074007239
Node 3	2.0546443732203774
Node 4	1.364159803860347
Node 5	2.6974766889493536
Node 6	3.908322709064356

Sigmoid Node 1

Inputs	*Weights*
Threshold	4.601251960011152
Node 2	-2.4045226373071156
Node 3	-2.0532744956144127
Node 4	-1.379986429753948
Node 5	-2.756274547604192
Node 6	-3.877948258791871

Sigmoid Node 2

Inputs	*Weights*
Threshold	-0.1550798021501342
Attrib outlook=sunny	-1.323464477913686
Attrib outlook=overcast	1.6602675280399888
Attrib outlook=rainy	-0.3207802552865604
Attrib temperature=hot	-0.2873122456981835
Attrib temperature=mild	1.181190360097958
Attrib temperature=cool	-0.7853150475848826
Attrib humidity	2.808930687905
Attrib windy	1.9190213581350706

Sigmoid Node 3

Inputs	*Weights*
Threshold	-0.18031675012278034
Attrib outlook=sunny	-1.1524514010228344
Attrib outlook=overcast	1.5760227701429683
Attrib outlook=rainy	-0.32578400279223824
Attrib temperature=hot	-0.2760307631136823
Attrib temperature=mild	1.0450876279343007
Attrib temperature=cool	-0.6318819517738498
Attrib humidity	2.4504774603875408
Attrib windy	1.678251292646871

Sigmoid Node 4

Inputs	*Weights*
Threshold	-0.3554146745674961
Attrib outlook=sunny	-0.46574052680925143
Attrib outlook=overcast	1.4382073898080827
Attrib outlook=rainy	-0.6194183985830608
Attrib temperature=hot	-0.0670794406887232
Attrib temperature=mild	0.6337484752708613
Attrib temperature=cool	-0.20814280117719502
Attrib humidity	1.982466584793048
Attrib windy	0.9946423645131915

Sigmoid Node 5

Inputs	*Weights*
Threshold	-0.06888405078498452
Attrib outlook=sunny	-1.3982064219096493
Attrib outlook=overcast	1.8084944112736516
Attrib outlook=rainy	-0.31997269602762973
Attrib temperature=hot	0.3035821635771427
Attrib temperature=mild	1.2908528760310662
Attrib temperature=cool	-0.8921466424329777
Attrib humidity	3.1090049574873424
Attrib windy	2.0747113212966872

Sigmoid Node 6

Inputs	*Weights*
Threshold	0.04399369934901554
Attrib outlook=sunny	-1.80182134279014
Attrib outlook=overcast	2.2544547024444554
Attrib outlook=rainy	-0.40095717506501327
Attrib temperature=hot	-0.41558677311306397
Attrib temperature=mild	1.589170285947685
Attrib temperature=cool	-1.2545441906677217
Attrib humidity	4.119310666164331
Attrib windy	2.740851006387263

Class yes

Input	Node 0

Class no

Input	Node 1

Appendix B. Naïve Bayes classifier results

Tested dataset	Number of instances	TESTING							
		Full training set		10-fold cross-validation		5-fold cross-validation		Percentage split (65%)	
		correct	incorrect	correct	incorrect	correct	incorrect	correct	incorrect
eucalyptus	736	56,3859	43,6141	54,0136	45,9864	53,4694	46,5306	52,7132	47,2868
white-clover	63	79,3651	20,6349	60,3175	39,6825	61,9048	38,0952	54,5455	45,4545
car	1728	87,0949	12,9051	85,5324	14,4676	85,1852	14,8148	87,6033	12,3967
credit-a	690	78,2609	21,7391	77,8261	22,1739	77,8261	22,1739	75,1037	24,8963
liver-disorders	345	55,942	44,058	56,5217	43,4783	56,2319	43,7681	52,8926	47,1074
vehicle	846	47,2813	52,7187	45,2719	54,7281	45,1537	54,8463	42,9054	57,0946
nursery	12960	90,3009	9,6991	90,3241	9,6759	90,2469	9,7531	90,4982	9,5018
anneal	898	86,637	13,363	86,3029	13,6971	86,5256	13,4744	87,5796	12,4204
solar-flare_1	323	93,808	6,192	93,1889	6,8111	93,808	6,192	90,2655	9,7345
diabetes	768	76,1719	23,8281	76,3021	23,6979	75,9115	24,0885	78,0669	21,9331

Appendix C. Multilayer perceptron classifier results

Tested dataset	Number of instances	TESTING							
		Full training set		10-fold cross-validation		5-fold cross-validation		Percentage split (65%)	
		correct	incorrect	correct	incorrect	correct	incorrect	correct	incorrect
eucalyptus	736	91,9837	8,0163	61,9565	38,0435	61,8207	38,1793	59,3023	40,6977
white-clover	63	100,00	0,00	73,0159	26,9841	77,7778	22,2222	54,5455	45,4545
car	1728	100,00	0,00	99,537	0,463	99,0162	0,9838	98,6777	1,3223
credit-a	690	96,087	3,913	84,2029	15,7971	83,0435	16,9565	82,1577	17,8423
liver-disorders	345	74,7826	25,2174	71,5942	28,4058	68,1159	31,8841	66,1157	33,8843
vehicle	846	90,3073	9,6927	81,6785	18,3215	82,3877	17,6123	79,7297	20,2703
nursery	12960	99,9846	0,0154	99,7299	0,2701	99,4753	0,5247	99,9559	0,0441
anneal	898	95,4343	4,5657	98,8864	1,1136	99,1091	0,8909	99,3631	0,6369
solar-flare_1	323	99,3808	0,6192	95,356	4,644	94,4272	5,5728	97,3451	2,6549
diabetes	768	80,599	19,401	75,3906	24,6094	75,1302	24,8698	73,6059	26,3941