NATURE AND BIOLOGY INSPIRED APPROACH OF CLASSIFICATION TOWARDS REDUCTION OF BIAS IN MACHINE LEARNING

HAN LIU, ALEXANDER GEGOV, MIHAELA COCEA

School of Computing, University of Portsmouth, Portsmouth, PO1 3HE, United Kingdom
E-MAIL: han.liu@port.ac.uk, alexander.gegov@port.ac.uk, mihaela.cocea@port.ac.uk

Abstract:

Machine learning has become a powerful tool in real applications such as decision making, sentiment prediction and ontology engineering. In the form of learning strategies, machine learning can be specialized into two types: supervised learning and unsupervised learning. Classification is a special type of supervised learning task, which can also be referred to as categorical prediction. In other words, classification tasks involve predictions of the values of discrete attributes. Some popular classification algorithms include Naïve Bayes and K Nearest Neighbour. The above type of classification algorithms generally involves voting towards classifying unseen instances. In traditional ways, the voting is made on the basis of any employed statistical heuristics such as probability. In Naïve Bayes, the voting is made through selecting the class with the highest posterior probability on the basis of the values of all independent attributes. In K Nearest Neighbour, majority voting is usually used towards classifying test instances. This kind of voting is considered to be biased, which may lead to overfitting. In order to avoid such overfitting, this paper proposes to employ a nature and biology inspired approach of voting referred to as probabilistic voting towards reduction of bias. An extended experimental study is reported to show how the probabilistic voting can manage to effectively reduce the bias towards improvement of classification accuracy.

Keywords:

Data mining; Machine learning; Naïve Bayes; K Nearest Neighbour; Probabilistic classification;

1. Introduction

Machine learning approaches have become increasingly popular in real world applications such as multi-criteria decision making [1], sentiment analysis [2] and ontology engineering [3]. In the form of learning strategies, machine learning approaches can be divided into two special types: supervised learning and unsupervised learning. Supervised learning means learning with a teacher, i.e. data used in the training stage is labelled. In practice, supervised learning can be involved in classification and regression tasks. Classification is also known as categorical prediction due to the fact that the aim of this type of tasks is to predict the values of discrete attributes. Similarly, regression is also known as numerical prediction due to the fact that this type of tasks aims to predict the values of continuous attributes. Unsupervised learning generally means learning without a teacher, i.e. data used in the training stage is unlabelled. In practice, unsupervised learning can be involved in association and clustering tasks. Association is aimed at identifying any correlations between different attributes and clustering is aimed at grouping of objects on the basis of their similarity.

This paper focuses on classification tasks. Some popular classification algorithms include Naïve Bayes [4] and K Nearest Neighbour [5]. Naïve Bayes belongs to Bayesian learning because this algorithm is designed through use of Bayes theorem [6]. K Nearest Neighbor belongs to lazy learning due to the fact that learning is not started until there are any unseen instances loaded into computer memory. In general, learning algorithms, like Naïve Bayes and K Nearest Neighbor, involve voting towards classifying any unseen instances. For example, Naïve Bayes manages to make the weighted voting through selecting the class with the highest posterior probability given on the basis of the values of all independent attributes from the training data. Similary, K Nearest Neighbor typically employes majority voting towards classifying unseen instances. The above types of voting are considered to be biased, which may lead to overfitting of training data [7]. In order to avoid such overfitting, this paper proposes to employ a probabilistic voting, which is inspired by nature and biology, towards reduction of bias.

The rest of this paper is organized as follows: Section 2 presents some theoretical preliminaries which include illustration of the essence of Naïve Bayes and K Nearest Neighbour algorithms. Section 3 presents the nature and biology inspired approach of voting referred to as probabilistic voting and justifies how this voting is effective towards reduction of bias. Section 4 reports an experimental study for validation of the above proposed approach and analyses the results obtained. Section 5 summarizes the contribution of this paper and suggests further directions.

2. Theoretical Preliminaries

Section 1 pointed out the limitations of the voting strategies that are involved in classification algorithms such as Naïve Bayes and K Nearest Neighbour. This section presents theoretical preliminaries relating to Bayes Theorem, distance measures as well as the essence of Naïve Bayes and K Nearest Neighbour algorithms.

2.1. Bayes Theorem

Bayes theorem, which is essentially used in Bayesian learning [7], is stated mathematically as equation (1):

$$P(Y \mid X) = \frac{P(Y) \cdot P(X \mid Y)}{P(X)} \quad (1)$$

where X and Y are two events:

- P(X) is read as the probability that event X occurs to be used as evidence supporting event Y.
- P(Y) is read as the prior probability that event Y occurs on its own.
- P(Y|X) is read as the posterior probability that event Y occurs given that event X truly occurs.
- P(X|Y) is read as conditional probability that event X occurs subject to that event Y must occur.

2.2. Distance Measures

In K Nearest Neighbour algorithm, distance measures are mainly used to estimate the similarity between a training instance and a test instance. Some popular metric distances, according to [8], include Euclidean Distance, Manhattan Distance, Maximum Distance, Minkowski Distance and Mahalanobis Distance. These measures are illustrated in the following:

Euclidean Distance is used to measure the distance between two points in a Euclidean space. The distance can be calculated in the way following equation (2):

$$D(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (2)$$

Where *n* is the dimensionality (the number of attributes) and *i* is the index of the attribute. For example, if the data is in two dimensions, then the equation (2) can be rewritten as equation (3) as follows:

$$D(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad (3)$$

Manhattan Distance is defined in [9] as "the distance between two points measured along axes at right angles" in two dimensional data and can be calculated as shown in equation (4):

$$D(x,y) = |x_1 - y_1| + |x_2 - y_2| \quad (4)$$

In terms of high dimensional data, equation (4) can be generalized to equation (5):

$$D(x,y) = \sum_{i=1}^{n} |x_i - y_i| \quad (5)$$

Maximum Distance is defined to be the maximum value of the distances between the coordinates of two points as illustrated in equation (6) below:

$$D(x,y) = \max_{0<i\le n} |x_i - y_i| \quad (6)$$

Minkowski Distance, which is also referred to as L_p-norm, can be calculated in the way following equation (7):

$$D(x,y) = \sqrt[p]{\sum_{i=1}^{n} (x_i - y_i)^p} \quad (7)$$

where p is read as the order of Minkowski Distance. In fact, the Minkowski Distance is essentially viewed as a generalization of Euclidean Distance, Manhattan Distance and Maximum Distance [8]. For example, while *p*=2, equation (7) applies to Euclidean Distance.

Mahalanobis Distance is defined as a measure of similarity between two points of the same distribution with a covariance matrix [10] and the distance can be calculated following equation (8):

$$D(x,y) = \sqrt{(x-y)S^{-1}(x-y)^T} \quad (8)$$

where x and y are two points and S is read as the covariance matrix regarding the distribution of x and y [11].

2.3. Naïve Bayes

As mentioned in Section 1, Naïve Bayes is a particular method of Bayesian learning. The learning outcome of Naïve Bayes is to find the class that has the highest posterior probability given all input attributes with their values as the joint condition. The learning strategy of this algorithm is in the following procedure based on Bayes theorem:

Step 1: Calculating the posterior probability of each class given each attribute with its value $P(y = c_k \mid x_i = v_{ij})$ on the basis of training instances, where y is the class attribute, c_k is the kth class label of y, x is the input attribute, i is the index of the x and v_{ij} is the jth value of x_i.

Step 2: Calculating the posterior probability of a class: $\prod_{i=0}^{n} P(y = c \mid x_i = v)$, where i is the index of the attribute and n is the number of attributes.

Step 3: Assigning the test instance the class that has the highest posterior probability on the basis of all the attribute values.

Table 1 shows an example for illustration of the above procedure of Naïve Bayes.

TABLE 1. ILLUSTRATIVE EXAMPLE FOR NAIVE BAYES

x_1	x_2	x_3	y
0	0	1	0
0	1	0	1
1	0	1	1
1	1	0	0
0	1	1	1

Following Step 1 of the procedure illustrated above, the posterior probability of each class given each attribute with its value is listed below:

$P(y=0|x_1 =0) = 0.33$, $P(y=1|x_1 =0) = 0.67$,
$P(y=0|x_1 =1) = 0.5$, $P(y=1|x_1 =1) = 0.5$,
$P(y=0|x_2 =0) = 0.5$, $P(y=1|x_2 =0) = 0.5$,
$P(y=0|x_2 =1) = 0.33$, $P(y=1|x_2 =1) = 0.67$,
$P(y=0|x_3 =0) = 0.5$, $P(y=1|x_3 =0) = 0.5$,
$P(y=0|x_3 =1) = 0.33$, $P(y=1|x_3 =1) = 0.67$,

Following Step 2 of the above procedure, the value of y, when $x_1 = 1$, $x_2 = 1$ and $x_3 = 1$, is essentially calculated in the following way:

$P(y=0| x_1 = 1, x_2 = 1, x_3 = 1) = P(y=0|x_1 =1) \times P(y=0|x_2 =1) \times P(y=0|x_3 =1) = 0.5\times0.33\times0.33$;
$P(y=1| x_1 = 1, x_2 = 1, x_3 = 1) = P(y=1|x_1 =1) \times P(y=1|x_2 =1) \times P(y=1|x_3 =1) = 0.5\times0.67\times0.67$;

Following Step 3 of the above procedure, the following implication is made:

$P(y=1| x_1 = 1, x_2 = 1, x_3 = 1)> P(y=0| x_1 = 1, x_2 = 1, x_3 = 1)$ => y= 1;

Therefore, the test instance is assigned the value of 1 for y.

2.4. K Nearest Neighbour

As mentioned in Section 1, K Nearest Neighbor is a method of lazy learning. The learning outcome of this algorithm is to assign the test instance the class which is the most commonly occurring from the k instances chosen from the training set. The learning strategy of this algorithm is in the following procedure:

Step 1: Choosing a method of measuring the distance between two points, e.g. Euclidean Distance.

Step 2: Determining the value of K, i.e. the number of training instances being selected.

Step 3: Finding the k instances (data points) that are closest to the given test instance.

Step 4: Identifying the majority class that is most commonly occurring from the k instances chosen at step 2.

Step 5: Assigning the test instance the majority class identified at step 3.

Fig.1 shows an example for illustration of the above procedure of K Nearest Neighbour.

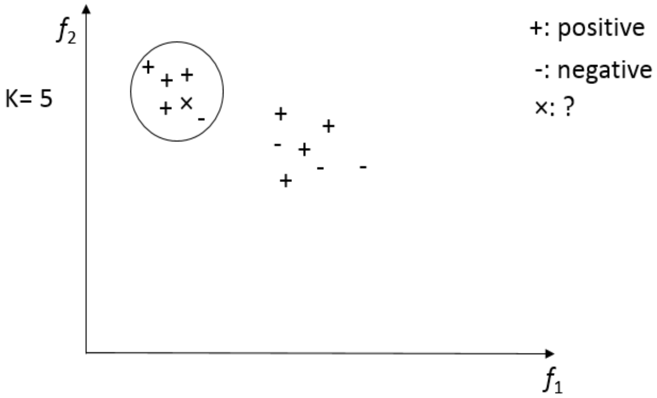

Figure 1. K Nearest Neighbour for two class classification

In the above example, the value of K is determined to be 5 and there are two possible classes (positive and negative) towards classifying the test instance. As shown in Fig.1, the five instances, which are closest to the test instance in terms of Euclidean Distance, are surrounded within a circle. In particular, there are four positive instances and one negative instance among the chosen ones. Therefore, the positive class is finally selected through majority voting towards classifying the above test instance.

3. Nature and Biology Inspired Voting

Section 1 pointed out the issue that voting involved in learning algorithms like Naïve Bayes and K Nearest Neighbour could increase the bias resulting in overfitting of training data. This section proposes a nature and biology inspired voting, which is referred to as probabilistic voting. The advances leading to reduction of bias are also justified in this section.

3.1 Key Features

The probabilistic voting is considered to be inspired by nature and biology since the voting is made on the basis of the hypothesis that the class with the highest probability only has the best chance of being selected towards classifying an unseen instance. In other words, it is not guaranteed that the class with the highest probability will definitely be selected for being assigned to the unseen instance. The essence of the probabilistic voting is illustrated as follows:

Step 1: Calculating the probability P_i for each single class *i*.

Step 2: Calculating the total probability P for all classes.

Step 3: Calculating the probability weight W_i for each single class *i*, i.e. $W_i = P / P_i$.

Step 4: Randomly selecting a single class *i* with the probability W_i towards classifying an unseen instance.

Based on the example shown in Table 1, the probability P_i for each single class *i* is listed below:

$P(y=0| x_1 = 1, x_2 = 1, x_3 = 1) = P(y=0|x_1 =1) \times P(y=0|x_2 =1) \times P(y=0|x_3 =1) = 0.5\times0.33\times0.33=0.05405$;
$P(y=1| x_1 = 1, x_2 = 1, x_3 = 1) = P(y=1|x_1 =1) \times P(y=1|x_2 =1) \times P(y=1|x_3 =1) = 0.5\times0.67\times0.67=0.04489$;

Following Step 2 in the above procedure, the total probability *P* is 0.09894= 0.05405+0.04489;

Following Step 3 in the above procedure, the probability weight W_i for each single class *i* is listed below:

For class 0, $W_0 = 0.05405/0.09894 = 0.5463$
For class 1, $W_1 = 0.04489/0.09894 = 0.4537$

Following Step 4 in the above procedure, an unseen instance is classified by probabilistic selection of one of the classes.

3.2 Justification

The probabilistic voting illustrated in Section 3.1 is very similar to the natural selection as one step of the procedure of Genetic Algorithms [12]. In particular, the selection of a class involved in Step 4 of the procedure of the probabilistic voting is inspired by the Roulette Wheel Selection [13].

The motivation of proposing the probabilistic voting is to make the voting based classification more natural. In this way, bias orginated from learning algorithms would be reduced effectively leading to reduction of overfitting of training data. In fact, it is fairly difficult to guarantee that the training data collected for a classification task is complete. For voting based classification algorithms such as Naïve Bayes and K Nearest Neighbor, the incompleteness of training data is very likely to result in bias towards selecting the class with the highest probability or frequency for classifying test instances. In other words, the probability or frequency for each class is estimated on the basis of any collected data and thus needs to be considered to be empirical rather than truly precise. This is very similar to the human reasoning approach that people generally make decisions and judgements based on their previous experience without the guarantee that the decisions and judgements are absolutely right. From this point of view, it is thus difficult to guarantee that the most frequently occuring class on the basis of the collected training data is the most accurate one being selected towards further classifying any test instances.

In particular, when the training data is imbalanced or even contains any noise, it is very likely to occur that a test instance is assigned a wrong class due to selecting the most frequently occuring class on the basis of the training data. For example, as shown in Fig.1, there are five training instances, four of which are positive, inside that circle. On the basis of the given training data, the test instance is finally assigned the positive class through majority voting. In reality, it is fairly possible that there are more negative instances supposed to be located inside the circle but unfortunately these negative instances have not been found or collected. Also, it is fairly possible that there are positive instances supposed to be outside the circle due to presence of noise, i.e. the coordinates of the points may be incorrectly recorded.

However, as inspired by nature and biology, the most frequently occuring class mentioned above can fairly be considered to have the best chance to contribute towards accurately classifying test instances, especially when the above conjecture concerning imblanced and noisy training data cannot be proved in a reasonable way.

A more detailed experimental study of the probabilsitic voting is reported in Section 4 and the obtained results are analysed critically and comparatively.

4. Experimental Setup and Results

The probabilistic voting approach illustrated in Section 3 is validated in an experimental study, and is compared with the majority voting and weighted voting, in terms of their impact on accuracy, while voting based algorithms are selected for classification tasks. In particular, Naïve Bayes and K Nearest Neighbour algorithms are chosen as the representatives of the voting based classification methods due to their popularity in practical applications.

The experiments are conducted by using 15 data sets retrieved from the UCI repository [14]. The characteristics of these data sets are illustrated in Table 2. The results are discussed in both quantative and qualitative terms.

TABLE 2 DATA SETS

Name	Attribute Types	#Attributes	#Instances	#Classes
anneal	mixed	38	798	6
audiology	discrete	69	226	24
autos	mixed	26	205	6
breast-cancer	discrete	9	286	2
breast-w	continuous	10	699	2
colic	mixed	23	368	2
credit-a	mixed	15	690	2
credit-g	mixed	20	1000	2
ecoli	continuous	8	336	8
glass	continuous	10	214	6
heart-c	mixed	76	920	4
heart-h	mixed	76	920	4
heart-statlog	continuous	13	270	2
hepatitis	mixed	20	155	2
ionosphere	continuous	34	351	2

NB: Mixed means containing both discrete and continuous attributes

The above data sets are chosen by considering the computational constraints due to the disadvantage of K Nearest Neighbour algorithm that it may perform poorly in terms of efficiency if the training data is large [15]. In fact, all the training instances must remain loaded into the computer memory during the testing stage so that any upcoming test instances can be classified eventually by going through the entire training set of instances. On the basis of the above consideration, all of the above chosen data sets have the dimensionalities lower than 100 and less than or equal to 1000 instances. In addition, the chosen data sets contain both discrete and continuous attributes. This is in order to study the extent to which the probabilistic voting can impact on classification accuracy for data with both types of attributes.

The experimental study is conducted by splitting each data set into a training set and a test set in the ratio of 70:30. For each data set, the experiment is repeated 10 times and the average of these accuracies is taken for comparative validation. As mentioned earlier, the K Nearest Neighbour algorithm could usually have a poor performance on large training data in terms of computational efficiency. Therefore, cross validation is not adopted in this study. In addition, the value of K for the K Nearest Neighbour algorithm is chosen to be 5 as this is one of the popular settings for relatively small data sets. Euclidean Distance is used to measure the similarity. The results are presented in Table 3.

TABLE 3 CLASSIFICATION ACCURACY

Data set	NB I	NB II	KNN I	KNN II
anneal	88%	**92%**	95%	**97%**
audiology	72%	**74%**	59%	**65%**
autos	48%	**65%**	54%	**67%**
breast-cancer	67%	**73%**	70%	**75%**
breast-w	95%	**98%**	95%	**97%**
colic	71%	**80%**	76%	**80%**
credit-a	75%	**81%**	**85%**	83%
credit-g	**75%**	**75%**	**73%**	**73%**
ecoli	84%	**90%**	85%	**86%**
glass	42%	**60%**	63%	**70%**
heart-c	86%	**88%**	**87%**	**87%**
heart-h	86%	**87%**	78%	**85%**
heart-statlog	86%	**91%**	84%	**85%**
hepatitis	83%	**92%**	89%	**95%**
ionosphere	81%	**91%**	77%	**90%**

NB: NB II and KNN II mean that Naïve Bayes and K Nearest Neighbour employ probabilistic voting for final classification. Otherwise, majority voting or weighted voting is employed.

For K Nearest Neighbour algorithm, the experiment study also includes the investigation of how the change of the K value would impact on classification accuracy while using any one of the two voting methods namely majority voting and probabilistic voting. The results concerning these two voting methods are compared as shown in Fig.2.

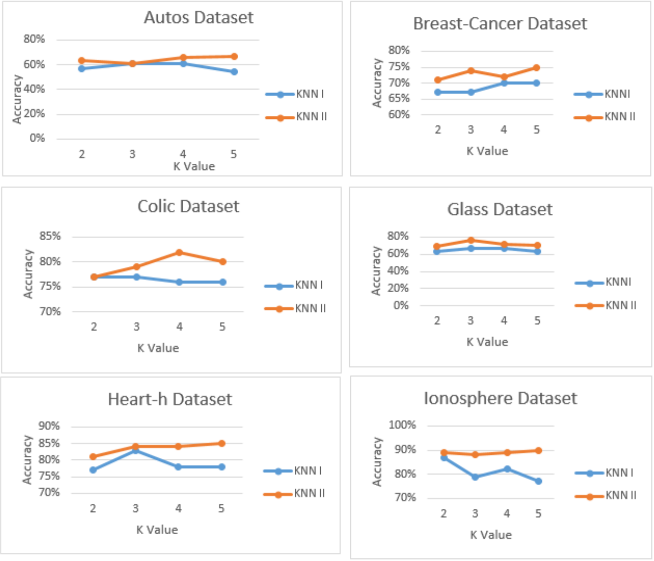

Figure 2 Performance of KNN with different K values

Table 3 shows that for both Naive Bayes and K Nearest Neighbour the probabilistic voting manages to increase the classification accuracy in comparison with the other two voting strategies in most cases. In particular, for Naïve Bayes, the probabilistic voting manages effectively to achieve higher accuracy than the weighted voting does in 14 out of 15 cases, except for the case on *credit-g* data set that the accuracy is the same while the two voting methods are compared. For K Nearest Neighbour, there are 12 out of 15 cases that the probabilistic voting manages effectively to achieve higher accuracy than the majority voting and 2 cases that the two voting methods perform the same. In addition, Fig.2 shows that for K Nearest Neighbour the probabilistic voting usually manages to achieve higher accuracy than the majority voting does while the K value is changed incrementally from 2 to 5. It can also be seen in Fig.2 that the probabilistic voting generally manages to keep a similar level of variance or even a lower level of that in comparison with the majority voting.

5. Conclusions

This paper has proposed a nature and biology inspired voting referred to as probabilistic voting. The experimental results show that the proposed voting can effectively manage to increase the classification accuracy in comparison with the weighted voting and majority voting, while Naïve Bayes and K Nearest Neighbour are used as the classification algorithms. The results also indicate that the probabilistic voting can generally lead to reduction of bias originated from algorithms and thus overfitting of training data can be avoided to some extent, due to the advantages of natural selection that is involved in the probabilistic voting. However, voting is also popularly involved in ensemble learning approaches such as Bagging [16], which indicates that similar issues concerning the voting strategies may also arise with such ensemble learning approaches. Therefore, it is worth to investigate the probabilistic voting further towards advancing the ensemble learning approaches through reduction of bias on final voting.

Acknowledgements

This paper is supported by the Computational Intelligence Research Group at the University of Portsmouth.

References

[1] V. Maliene, "Specialised property valuation: Multiple criteria decision analysis," *Journal of Retail & Leisure Property,* vol. 9, no. 5, p. 443–50, 2011.

[2] B. Liu, Sentiment Analysis and Ontology Mining, California: Morgan &Claypool Publishers, 2012.

[3] C. Roussey, F. Pinet, M. A. Kang and O. Corcho, "An Introduction to Ontologies and Ontology Engineering," in *Ontologies in Urban Development Projects*, vol. 1, G. Falquet, C. Métral, J. Teller and C. Tweed, Eds., London, Springer, 2011, pp. 9-38.

[4] I. Rish, "An empirical study of the naive Bayes classifier," *IJCAI 2001 workshop on empirical methods in artificial intelligence,* vol. 3, no. 22, pp. 41-46, 2001.

[5] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory,* vol. 13, no. 1, pp. 21-27 , 1967.

[6] P. M. Lee, Bayesian Statistics: An Introduction, 4 ed., Chichester, West Sussex: Wiley, 2012.

[7] D. Barber, Bayesian Reasoning and Machine Learning, Cambridge: Cambridge University Press, 2012.

[8] C. Cassisi, P. Montalto, M. Aliotta, A. Cannata and A. Pulvirenti, "Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining," in *Advances in Data Mining Knowledge Discovery and Applications*, Rijeka, InTech, 2012, pp. 71-96.

[9] P. E. Black, "Manhattan distance," National Institute of Standards and Technology, 31 May 2006. [Online]. Available: http://www.nist.gov/dads/HTML/manhattanDistance.html. [Accessed 25 March 2016].

[10] P. C. Mahalanobis, "On the generalised distance in statistics," *Proceedings of the National Institute of Sciences of India,* vol. 2, no. 1, p. 49–55, 1936.

[11] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, 2 ed., New York: John Wiley & Sons, 2012.

[12] K. F. Man, K. S. Tang and S. Kwong, "Genetic Algorithms: Concepts and Applications," *IEEE Transactions on Industry Electronics,* vol. 43, no. 5, pp. 519-534, 1996.

[13] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Physica A: Statistical Mechanics and its Applications,* vol. 391, no. 6, p. 2193–2196, 2012.

[14] M. Lichman, "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, 2013. [Online]. Available: http://archive.ics.uci.edu/ml. [Accessed 25 March 2016].

[15] P. Cunningham and S. J. Delany, "k-Nearest Neighbour Classifiers," Dublin, 2007.

[16] L. Breiman, "Bagging predictors," *Machine Learning,* vol. 24, no. 2, p. 123–140, 1996.