Classification Preservation Using Assorted Dimensionality Reduction Techniques

USMAN A. BABA[1], AUGUSTINE S. NSANG[2]
[1, 2] *Department of Computer Science, School of Information Technology and Computing, American University of Nigeria, Yola By-Pass, Yola, Nigeria*

Abstract- In this paper, we implement the perceptron classification algorithm and apply it to three two-class datasets which include the student, weather and ionosphere datasets. Then the k-Nearest Neighbors classification algorithm is also applied to the same two-class datasets. Each dataset is then reduced using fourteen different dimensionality reduction techniques. The perceptron and k-nearest neighbor classification algorithms are then applied to each reduced set and the performances of the dimensionality reduction techniques in preserving the classification of a dataset by the k-nearest neighbors and perceptron classification algorithm are compared. The extent to which the classification of a dataset is preserved by a given dimensionality reduction technique is evaluated using the rand index and confusion matrices.

Indexed Terms- Classification, Confusion Matrix, Dimensionality Reduction, Eager Learner, k-Nearest Neighbors, Lazy Learner, Perceptron, Rand Index.

I. INTRODUCTION

Data volumes and variety are increasing at an alarming rate making very tedious any attempt to glean useful information from these large data sets. Extracting or mining useful information and hidden patterns from the data is becoming more and more important but can be very challenging at the same time [1]. The biggest challenge is the number of variables (dimensions) associated with each observation. However, not all dimensions are required to understand the phenomenon under investigation in high-dimensional datasets. For this reason, reducing the dimension of the dataset can drastically improve the speed of the analysis while significantly maintaining the accuracy. This process is known as Dimensionality Reduction [2]. Dimensionality reduction provides a compact representation of an original high-dimensional data, which means the reduced data is free from any further processing and only the vital information is retained, so it can be used with many machine learning algorithms that perform poorly on high-dimensional data [3].

A lot of methods exist for reducing the dimensionality of data. There are two categories of these methods; in the first category, each attribute in the reduced dataset is a linear combination of the attributes of the original dataset. In the second category, the set of attributes in the reduced dataset is a subset of the set of attributes in the original dataset [4]. Techniques belonging to the first category include Random Projection (RP), Singular Value Decomposition (SVD), Principal Component Analysis (PCA), and so on; while techniques in the second category include but are not limited to the Combined Approach (CA), Direct Approach (DA), Variance Approach (Var), New Top-Down Approach (NTDn), New Bottom-Up Approach (NBUp), New Top-Down Approach (modified version) and New Bottom-Up Approach (modified version) [5].

Machine learning is a scientific field in which computer systems can automatically and intelligently learn their computation and improve on it through experience [6], [7]. Machine learning algorithms are of two main types: supervised learning algorithms and unsupervised learning algorithms. These algorithms have been used in solving a lot of complex real-world problems [8], [9]. In unsupervised learning, the set of observations are categorized into groups (clusters) basing the categorization on the similarity between them. This categorization is otherwise known as clustering [6]. Many clustering algorithms exist, among which *k-means* clustering is the most famous for a large number of observations [10].

Unlike clustering, classification is a supervised learning method in which the corresponding label for any valid input is predicted based on a number of training examples referred to as the "training set" [6], [10]. The learning algorithm is applied to the training set made up of past examples having the same set of attributes with the unseen example [6], [10]. However, before starting the training, the label of each example in the "training set" is known [12].

Classification algorithms can further be categorized into eager and lazy learners, and this investigation considers one from each category. Eager learning algorithms attempt to construct a general rule or create a generalization during the training phase which can further be used in classifying unseen instances [11]. Examples of eager learners include decision trees, support vector machines, and the perceptron.

To build a classifier model, an eager learner attempts to construct a general rule in the training phase which will subsequently be used in classifying unseen instances. On the other hand, a lazy learner delays the process until it is presented with an unseen instance [11]. The main disadvantage of eager learning is the long time which the learner takes in constructing the classification model but after the model is constructed, an eager learner is very fast in classifying unseen instances. For the lazy learner, the disadvantage is the amount of space it consumes in memory and the time it takes during the classification [13]. This makes dimensionality reduction a very crucial preprocessing step because it facilitates classification, and compression of high-dimensional data and thus conserves memory and provides a compact representation of an original high-dimensional data [5].

II. DIMENSIONALITY REDUCTION TECHNIQUES

This section gives a description of all the Dimensionality Reduction techniques implemented for the purpose of this investigation.

2.1 The New Random Approach

This is a technique suggested by [5]. With this technique, to reduce a data set D of dimensionality d to one of dimensionality k, a set S_k is formed consisting of k numbers selected at random from the set S shown in equation 2.1 below:

$S = \{x \epsilon \mathrm{N} \mid 1 \leq x \leq d\}$ (2.1)

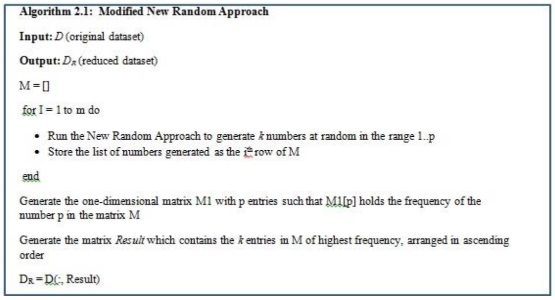
Then, our reduced set, D_R, is given by equation 3.2 below:

$D_R = D(:, S_k)$ (2.2)

That is, D_R is a data set having the same number of rows as D, and if A_i is the i^{th} attribute of D_R, then A_i will be the j^{th} attribute of D if j is the i^{th} element of S_k.

2.2 Modified New Random Approach

This technique is a modification of the new random approach proposed by [14]. To reduce a data set D of dimensionality p to one of dimensionality k, we shall use the algorithm below which is meant to generate a result less random than the results generated by the New Random Approach.



```
Algorithm 2.1: Modified New Random Approach
Input: $D$ (original dataset)
Output: $D_R$ (reduced dataset)
M = []
for I = 1 to m do
    • Run the New Random Approach to generate $k$ numbers at random in the range 1..p
    • Store the list of numbers generated as the $i^{th}$ row of M
end
Generate the one-dimensional matrix M1 with p entries such that M1[p] holds the frequency of the
number p in the matrix M
Generate the matrix Result which contains the $k$ entries in M of highest frequency, arranged in ascending
order
$D_R$ = D(:, Result)
```

2.3 Principal Component Analysis (PCA)

According to [15], to reduce a dataset, $D_{n\ x\ p}$, from p columns to q columns using *PCA*, we first find the *Singular Value Decomposition* of D. In other words we decompose D into three submatrices U, S and V as given in equation 2.3 below:

$$D = USV^T \qquad (2.3)$$

where:

U is an n x n orthogonal matrix whose columns are the left singular vectors of D,
V is a p x p orthogonal matrix whose columns are right singular vectors of D and
S is an n x p diagonal matrix whose diagonal elements are the singular values of D.

The transformed matrix is computed from equation 2.4 where V_q is a p x q matrix consisting of the first q columns of V.

$D_{PCA} = DV_q$ (2.4)

2.4 The Variance Approach

As explained by [16], with the *Variance* approach, to reduce a dataset *D* to a data set D_R, we start with an empty set, I, and then add dimensions of *D* to this set in decreasing order of their variances. That means that a set *I* of *r* dimensions will contain the dimensions of top r variances.

Thus, $I_r = \{i_1, . . . , i_r\} \subset \{1, . . . , n\}$, the collection of dimensions corresponding to the top *r* variances. That is i_1 denotes the dimension of largest variance, i_2 the dimension of second largest variance, etc. The reduced database, D_R, in equation 2.5 is obtained by extracting the data corresponding to the selected dimensions.

$$D_R = D(:, I_r) \qquad (2.5)$$

where D_R has the same number of rows as D and r columns: the i^th column of D_R is the column of the original database with the i^th largest variance".

2.5 The Combined Approach

According to [16], "like the previous approach, the *Combined* Approach is one approach which reduces a dataset *D* to a subset of the original attribute set. To reduce a dataset D_{nxp} to a dataset containing *k* columns, the *Combined Approach* selects the combination of *k* attributes which best preserves the interpoint distances, and reduces the dataset to a dataset containing only those *k* attributes. To do so, it first determines the extent to which each attribute preserves the interpoint distances. In other words, for each attribute, *x*, in *D*, it computes $g_x m$ and $g_x M$ given by equation 2.6 and equation 2.7 respectively.

$$g_x m = \min\{ \frac{\| f(u) - f(v) \|^2}{\| u - v \|^2} \} \qquad (2.6)$$

$$g_x M = \max\{ \frac{\| f(u) - f(v) \|^2}{\| u - v \|^2} \} \qquad (2.7)$$

where *u* and *v* are any two rows of *D*, and *f(u)* and *f(v)* are the corresponding rows in the dataset reduced to the single attribute *x.* The average distance preservation for the attribute *x* is then computed using equation 2.8.

$$g_x mid = (g_x m + g_x M)/2 \qquad (2.8)$$

To reduce the dataset *D* from *p* columns to *k* columns, this approach then finds the combination of *k* attributes whose average value of $g_x mid$ is maximum".

2.6 The Direct Approach

The authors of [16] came up with the direct approach which is similar to the *Combined Approach.* According to the authors, to reduce a dataset D_{nxp} to a dataset containing *k* columns, the *Direct Approach* selects the combination of *k* attributes which best preserve the interpoint distances, and reduces the original dataset to a dataset containing only those *k* attributes. To do so, it first generates all possible combinations of *k* attributes from the original *p* attributes. Then, for each combination, *C,* it computes $g_c m$ and $g_c M$ given in equation 2.9 and equation 2.10 respectively.

$$g_C m = \min\{ \frac{\| f(u) - f(v) \|^2}{\| u - v \|^2} \} \qquad (2.9)$$

$$g_C M = \max\{ \frac{\| f(u) - f(v) \|^2}{\| u - v \|^2} \} \qquad (2.10)$$

where *u* and *v* are any two rows of *D,* and *f(u)* and *f(v)* are the corresponding rows in the dataset reduced to the attributes in C. The average distance preservation for this combination of attributes is then computed using equation 2.11.

$$g_c mid = (g_c m + g_c M)/2 \qquad (2.11)$$

As we can see, the difference between the *Combined* and *Direct* Approaches is that for the *Combined Approach,* we first find the average distance preservation for each attribute, and then, for any combination of attributes, we compute its average distance preservation by finding the averages of the distance preservations of the individual attributes. With the *Direct Approach,* on the other hand, to find the average distance preservation for any combination of attributes, *C*, we reduce the original dataset directly to the dataset containing only the attributes in *C*, and then compute the average distance preservation for this combination using the formulas above.

2.7 Random Projection (RP)

With RP, a given dataset with *d* dimensions is projected onto a lower-dimensional subspace of *k*-dimensions using a random *d*k* matrix *R* whose columns have unit lengths [15].

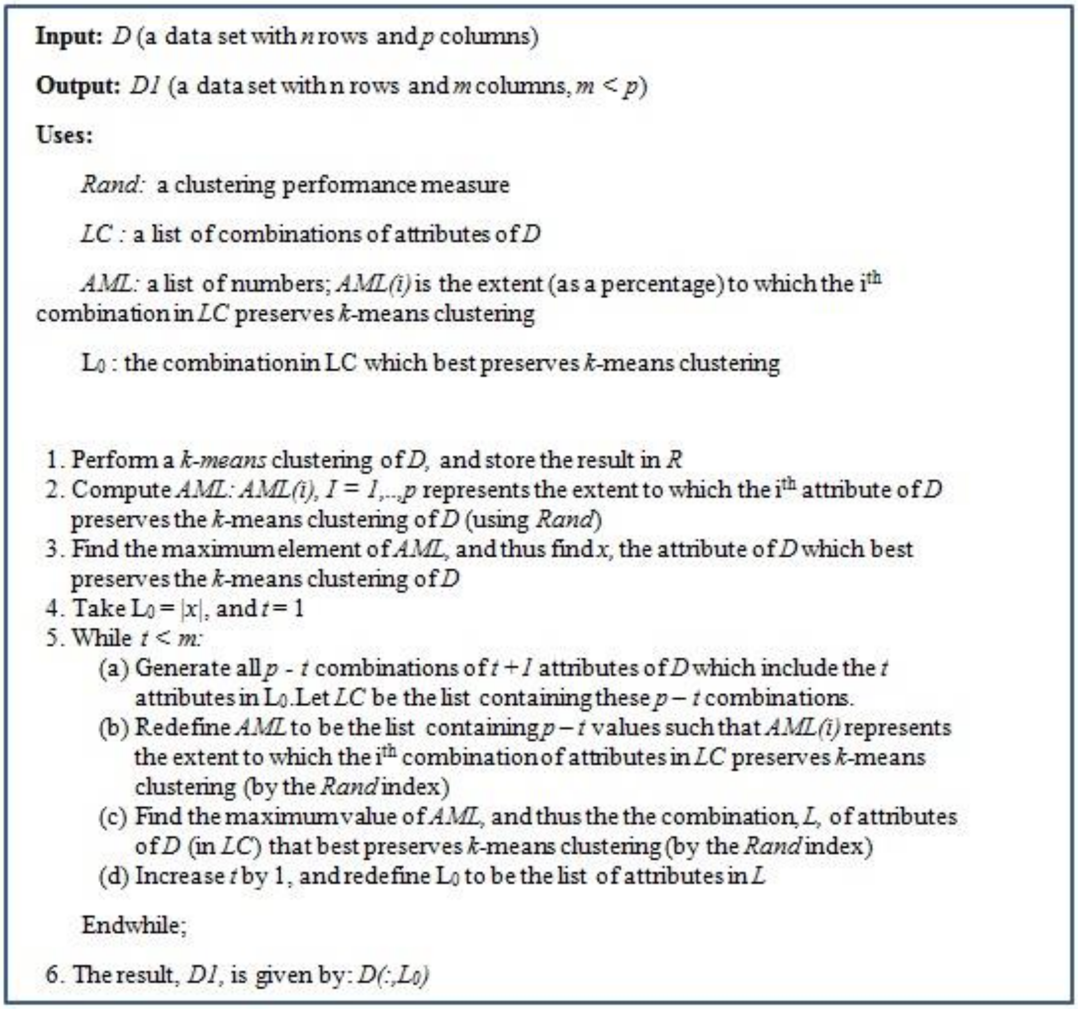
For instance, assuming D_{n*d} is the given dataset with d-dimensions, then the reduced *k*-dimensional dataset, *X,* is obtained as shown in equation 2.12.

$$X_{nxk} = D_{nxd} * R_{dxk} \qquad (2.12)$$

2.8 The New Bottom-Up Approach

This approach, proposed by [17], works by selecting subsets of attributes increased by one attribute at a time. With this technique, assuming we want to reduce a data set of p dimensions to another data set of m dimensions, the process is started with a subset S1, containing a single attribute, say y, from the original data set, which best preserves k-means clustering. It then increases to S2, which contains a total of two attributes including y that best preserves k-means clustering. S2 is then increased to another subset S3 that contains three attributes (the two attributes of S2 and another attribute from the original dataset apart from the two attributes of S2) which best preserves k-means clustering. This process continues until S_m (the subset that has the m attributes of the original dataset which best preserves k-means clustering) is obtained. The algorithm is shown in Algorithm 2.2.

Algorithm 2.2 The New Bottom-Up Approach

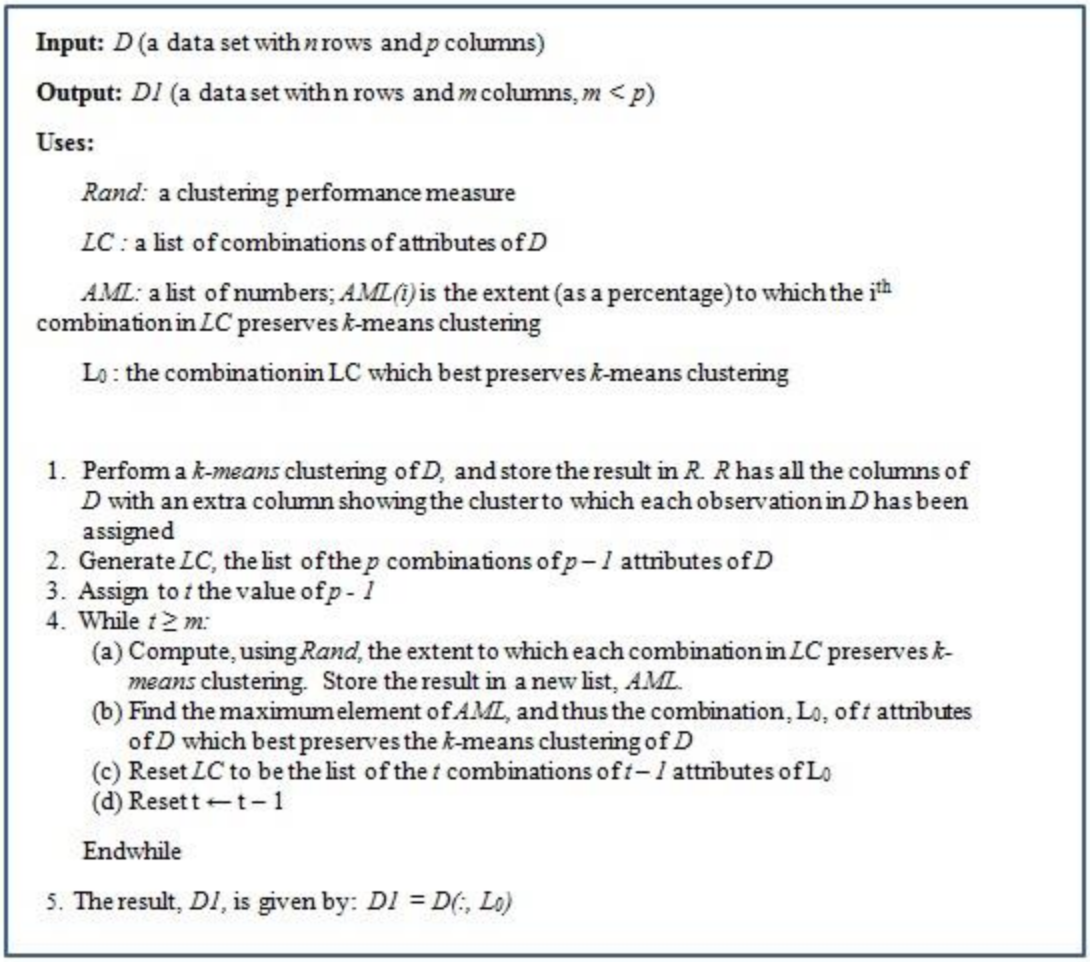


```
Input: $D$ (a data set with $n$ rows and $p$ columns)

Output: $D1$ (a data set with n rows and $m$ columns, $m < p$)

Uses:

    Rand: a clustering performance measure

    LC : a list of combinations of attributes of $D$

    AML: a list of numbers; $AML(i)$ is the extent (as a percentage) to which the $i^{th}$
combination in $LC$ preserves $k$-means clustering

    $L_0$ : the combinationin LC which best preserves $k$-means clustering


 1. Perform a $k$-means clustering of $D$, and store the result in $R$
 2. Compute $AML$: $AML(i)$, $I = 1,..,p$ represents the extent to which the $i^{th}$ attribute of $D$
    preserves the $k$-means clustering of $D$ (using Rand)
 3. Find the maximumelement of $AML$, and thus find $x$, the attribute of $D$ which best
    preserves the $k$-means clustering of $D$
 4. Take $L_0 = |x|$, and $t = 1$
 5. While $t < m$:
      (a) Generate all $p - t$ combinations of $t + 1$ attributes of $D$ which include the $t$
          attributes in $L_0$.Let $LC$ be the list containing these $p - t$ combinations.
      (b) Redefine $AML$ to be the list containing $p - t$ values such that $AML(i)$ represents
          the extent to which the $i^{th}$ combination of attributes in $LC$ preserves $k$-means
          clustering (by the Rand index)
      (c) Find the maximum value of $AML$, and thus the the combination, $L$, of attributes
          of $D$ (in $LC$) that best preserves $k$-means clustering (by the Rand index)
      (d) Increase $t$ by 1, and redefine $L_0$ to be the list of attributes in $L$

    Endwhile;

 6. The result, $D1$, is given by: $D(:,L_0)$
```

2.9 The New Top-Down Approach

This approach is also suggested by [17], and it operates in a very similar manner to the *New Bottom-Up* approach discussed above. However, instead of considering the subset of attributes increased by one attribute at a time, the Top-Down approach considers the subset of attributes decreased by one attribute at a time. Assuming we want to reduce a data set with p dimensions to one with m dimensions, the Top-Down approach starts by reducing the original dataset to the subset of $p-1$ attributes which best preserves k-means clustering, then to the subset of $p-2$ attributes which best preserve *k*-means clustering. The procedure continues until the subset of *m* attributes that best preserve the *k*-means clustering of the original data set is obtained. The algorithm is shown in Algorithm 2.3.

Algorithm 2.3 The New Top-Down Approach



```
Input: D (a data set with n rows and p columns)
Output: D1 (a data set with n rows and m columns, $m < p$)
Uses:
    Rand: a clustering performance measure
    LC : a list of combinations of attributes of D
    AML: a list of numbers; AML(i) is the extent (as a percentage) to which the i^th
combination in LC preserves k-means clustering
    $L_0$ : the combination in LC which best preserves k-means clustering

1. Perform a k-means clustering of D, and store the result in R. R has all the columns of
   D with an extra column showing the cluster to which each observation in D has been
   assigned
2. Generate LC, the list of the p combinations of $p – 1$ attributes of D
3. Assign to t the value of $p - 1$
4. While $t \geq m$:
    (a) Compute, using Rand, the extent to which each combination in LC preserves k-
        means clustering. Store the result in a new list, AML.
    (b) Find the maximum element of AML, and thus the combination, $L_0$, of t attributes
        of D which best preserves the k-means clustering of D
    (c) Reset LC to be the list of the t combinations of $t – 1$ attributes of $L_0$
    (d) Reset $t \leftarrow t – 1$
    Endwhile
5. The result, D1, is given by: $D1 = D(:, L_0)$
```

2.10 The New Top-Down Approach (Modified Version)

This technique is proposed in [18]. It is a modification of the New Top-Down approach. In this technique, assuming we want to reduce a data set with *p* dimensions to one with *m* dimensions, the process is started by a reduction to the subset of *p-1* attributes which best preserves the interpoint distances (instead of *k-means clustering*), then to the subset of *p-2* attributes which best preserve the interpoint distances, etc. The process continues till the subset of *m* attributes that best preserve the interpoint distances of the original data set is obtained.

2.11 The New Bottom- Approach (Modified Version)

This technique is also proposed in [18] as a modification of the New Bottom-Up approach. Suppose we want to reduce a data set of *p* dimensions to a data set containing *m* dimensions. This approach starts by looking for the subset *S1* containing the single attribute *x* from the original data set which best preserves the interpoint distances (instead of *k-means clustering* as in the New Bottom-Up Approach described above). It then finds the subset S2, which contains a total of two attributes including *x* that best preserves the interpoint distances. It then finds the subset *S3* that contains three attributes (the two attributes of S2 and another attribute from the original dataset) which best preserves the interpoint distances. This process continues until the subset S_m that has the m attributes of the original dataset which best preserves the interpoint distances is obtained.

2.12 First Novel Approach [30]

Suppose we want to reduce a data set of *p* dimensions to another data set of *m* dimensions. The First Novel Approach finds the *approximate* extent to which the interpoint distances is preserved by each attribute *x* in the original data set. To do so, it computes $g_x m$ and $g_x M$ as in equations 2.6 and 2.7 above.

It then selects *m* attributes from the original dataset with the largest $g_x mid$ values.

2.13 Second Novel Approach [30]

Unlike the First Novel Approach, the Second Novel Approach starts by computing adp_x for each attribute *x*, which is the actual extent to which it preserves the interpoint distances (see equation 2.16 below).

$$\text{adp}_x = \frac{\sum_{u=1}^{n}\sum_{v=u+1}^{n}\frac{\| f(u)-f(v)\|^2}{\| u-v\|^2}}{n_r} \qquad (2.16)$$

n_r is the number of pairs of rows in the original dataset (represented in equation 2.17):

$$n_r = {}^nC_r = \frac{n(n-1)}{2} \qquad (2.17)$$

Then, to reduce a dataset from its original *p* dimensions to *m* dimensions, the *m* attributes with the largest adp_x values are selected.

2.14 Third Novel Approach [30]

For this approach, to reduce a dataset from its original *d* dimensions to *k* dimensions, after computing the extent to which each attribute in the original dataset preserves k-means clustering, the *k* attributes which best preserve k-means clustering are selected.

III. THE K-NEAREST NEIGHBORS CLASSIFICATION ALGORITHM

The K-Nearest Neighbors Algorithm is a supervised learning algorithm and also one of the simplest machine learning algorithms [17], [19] . In this classification technique, the result/label of any given instance is predicted based on the label most common to its *k* nearest neighbors. K in this case is a user-defined positive integer, normally with a small value [20].

Figure 3.1 below illustrates how this classification algorithm works.

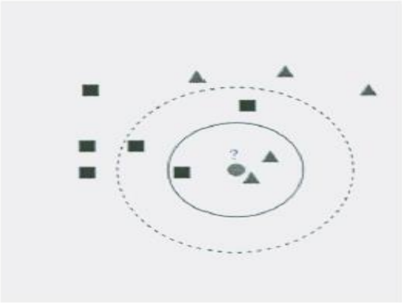
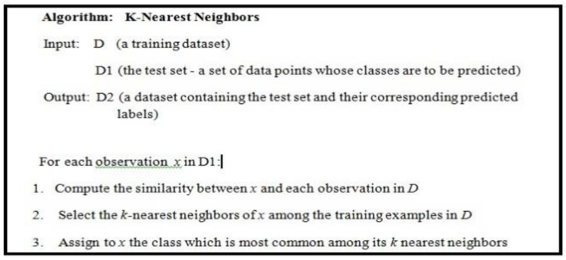


Figure 3.1: KNN example

In Figure 3.1, the each data point either belongs to the class of *squares* or the class of *triangles*. If k is 3, the test sample will be classified as a triangle since its 3 nearest neighbors include 2 triangles and 1 square. But if k is 5, the test sample will be classified as a square since its 5 nearest neighbors include 2 triangles and 3 squares.

The k-nearest neighbors algorithm is given below:



```
Algorithm:  K-Nearest Neighbors
Input:  D   (a training dataset)
        D1  (the test set - a set of data points whose classes are to be predicted)
Output: D2  (a dataset containing the test set and their corresponding predicted
             labels)

For each observation $x$ in D1:
1.  Compute the similarity between $x$ and each observation in $D$
2.  Select the $k$-nearest neighbors of $x$ among the training examples in $D$
3.  Assign to $x$ the class which is most common among its $k$ nearest neighbors
```

Algorithm 3.1: The K-Nearest Neighbors Algorithm

In this algorithm, the similarity between the test data and each observation in the training set is measured by computing the distance between them. For numerical data, which is the type of data used in this investigation, Euclidean distance is the most widely used distance metric. It performs relatively better than the cosine and Minkowsky distance [21]. For this reason, Euclidean distance is the distance metric that has been chosen for this investigation.

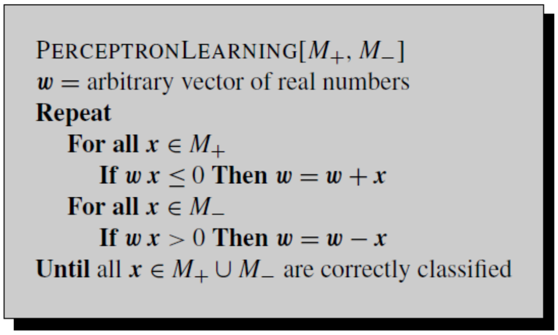
IV. THE PERCEPTRON

The perceptron [22] is a supervised learning algorithm used for classifying each point of a data set into either a positive or a negative label [20]. Basically, the perceptron takes a weighted sum of observations (real values) and if the sum is greater than some threshold value, it sends an output of one otherwise it sends zero (or -1) [23]. Unfortunately, in some cases, it takes a long time to train the perceptron because of the process of adjusting the weights until all observations are correctly classified. However, after training, the algorithm is very efficient in using the weights obtained for classification of unseen instances [24].

The perceptron is made up of a summation processor which takes the dot product of the inputs and the weights and then an activation function which uses a one-step function (shown in equation 4.1) to determine the output of the perceptron. Learning by the perceptron is completed when it happens that no error has occurred after an epoch (a complete pass through the training set) during the training phase [23]. When the training is complete, the perceptron will respond, for any input presented to it, with an output that is the same as the output of the observation used in the training phase.

$$f(x) = \begin{cases} -1 \; if \;\; w.x < 0 \\ \;\, 1 \quad if \;\; w.x \geq \; 0 \end{cases} \qquad (4.1)$$

The perceptron algorithm is depicted in Algorithm 4.1.



```
PerceptronLearning[$M_+$, $M_-$]
$w$ = arbitrary vector of real numbers
Repeat
    For all $x \in M_+$
        If $w\,x \leq 0$ Then $w = w + x$
    For all $x \in M_-$
        If $w\,x > 0$ Then $w = w - x$
Until all $x \in M_+ \cup M_-$ are correctly classified
```

Algorithm 4.1: The Perceptron

V. EXPERIMENTAL RESULTS

In this section, the dimensionality reduction techniques described in section 2 of this paper are compared by the extent to which they preserve the perceptron and K-Nearest Neighbors classification of the weather dataset, student dataset and ionosphere dataset obtained from UCI Machine Learning Repository [25]. The results obtained for the student data set are presented in Tables 5.1 and 5.2 below.

5.1 The Rand Index

One of the challenges faced during classification is in evaluating the performance of a classifier. Some common performance measures include Mean Squared Error (MSE), ROC Curves and so on. What these measures do is to compare the results of the supervised classification algorithm with the known labels, which means these measures do not consider the fact that the output labels could be switched even if the labels are perfectly identified [26].

The Rand Index is a measure for evaluating clustering based on the pairs of data that are in complete agreement i.e. those that have same labels or belong to the same cluster after each clustering and those that are in different clusters after each clustering [17]. We shall use the Rand Index as the evaluation measure in this section.

5.2 Preservation of K-Nearest Neighbors Classification using the Rand Index

To obtain the results in this section, the training set in each of the three data sets is used in training the K-Nearest Neighbors classifier. Each observation in the test set of the respective dataset is then assigned a label that is most common to its k nearest neighbors. The training and test sets of each of the three data sets are then reduced using each of the fourteen reduction techniques described in this paper, and each observation in the reduced test sets of the respective datasets is then assigned a label that is most common to its k nearest neighbors in the corresponding training set. The *rand* index is then used to measure the extent to which each of the fourteen techniques preserve the *k-nearest neighbors* classification of the original datasets.

Table 5.1: Comparing the reduction techniques for K-Nearest Neighbors classification preservation using the student data set

Reduction Techniques	12 Attributes	13 Attributes	14 Attributes

Variance	100%	100%	100%
Novel approach 1	100%	100%	100%
Novel approach 2	100%	100%	100%
Novel approach 3	100%	100%	100%
New Bottom-Up	100%	100%	100%
New Top-Down	100%	100%	100%
New Bottom-Up (Modified)	100%	100%	100%
New Top-Down (Modified)	100%	100%	100%
Principal Component Analysis	100%	100%	100%
Direct Approach	100%	100%	100%
Combined Approach	100%	100%	100%
New Random Approach	100%	100%	100%
New Random Approach (Mod)	100%	100%	100%
Random Projection	100%	100%	100%

Table 5.1 above shows the results obtained when we compared the reduction techniques for K-Nearest Neighbors classification preservation using the student data set. We obtained very similar results when we compared the reduction techniques for K-Nearest Neighbors classification preservation using the weather and ionosphere data sets.

5.3 Preservation of Perceptron Classification Using the Rand Index

To obtain the results in this section, the training set in each of the three datasets is used to train the perceptron. The weight vector obtained from the training phase then is then used in classifying the test set of each dataset. The training and test set of each dataset is reduced using the fourteen reduction techniques discussed in this paper, and the weight vector obtained using each reduced training set is used to classify the corresponding reduced test set. The rand is then used to evaluate the extent to which the perceptron classification of each dataset is preserved by each reduction technique. The results obtained when we compared the reduction techniques for perceptron classification preservation using the student data set are shown in Table 5.2 below.

Table 5.2: Comparing the reduction techniques for the perceptron classification preservation using the student data set.

Reduction Techniques	12 Attributes	13 Attributes	14 Attributes
Variance	83.3%	91.7%	91.7%
Novel approach 1	83.3%	83.3%	83.3%
Novel approach 2	83.3%	83.3%	91.7%
Novel approach 3	100%	100%	91.7%
New Bottom-Up (Modified)	83.3%	83.3%	66.7%
New Top-Down (Modified)	83.3%	83.3%	91.7%
New Bottom-Up	83.3%	83.3%	83.3%
New Top-Down	75%	66.7%	58.3%
Principal Component Analysis	58.3%	66.7%	83.3%
Direct Approach	58.3%	83.3%	83.3%
Combined Approach	83.3%	83.3%	83.3%
New Random Approach (Modified)	83.3%	83.3%	91.7%

New Random Approach	75%	83.3%	83.3%
Random Projection	91.7%	91.7%	75%

As mentioned above, we also compared the reduction techniques for perceptron classification preservation using the weather and ionosphere data sets. All the results we obtained showed that, on the average, for a reduction of a dataset with *p* attributes to a dataset with *q* attributes, where $q << p$, the *First, Second* and the *Third Novel Approaches* perform better than all the other techniques in preserving the perceptron classification of the original datasets.

VI. USE OF CONFUSION MATRICES

This section describes how the reduction techniques – compared against each other – preserve the perceptron and K-Nearest Neighbors classification of the original dataset using confusion matrices.

6.1 Confusion Matrix

A confusion matrix is a table containing the description of the actual and predicted classifications performed by a classifier. It is a widely used and effective metric for evaluating the performance of classifiers [27] [28]. In the context of this study, the four parameters that constitute a confusion matrix include:

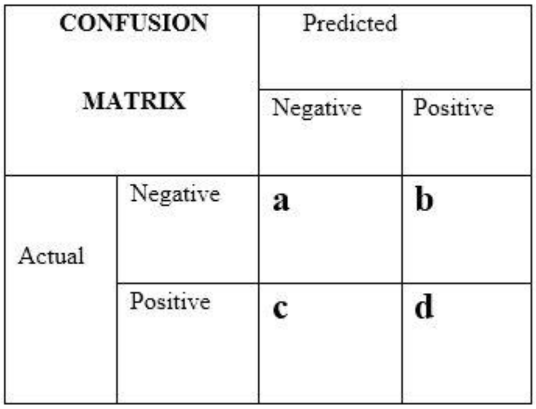
a is the number of negative observations correctly predicted as negative

b is the number of negative observations predicted as positive

c is the number of positive observations predicted as negative and

d is the number of positive observations correctly predicted as positive.

Table 6.1 is commonly used to show the confusion matrix of a classifier model having two classes.



CONFUSION MATRIX		Predicted	
		Negative	Positive
Actual	Negative	**a**	**b**
	Positive	**c**	**d**

Table 6.1: Confusion Matrix

According to [29], some of the evaluation measures that can be obtained from the confusion matrix include:

i) *Accuracy (AC):* This is the proportion of the total number of predictions that were correct. It is calculated as: $AC = \frac{a+d}{a+b+c+d}$

ii) *True Positive Rate (TP) or Recall:* This is the proportion of the number of positive instances correctly classified as positive and can be calculated as: $TP = \frac{d}{c+d}$

iii) *True Negative Rate (TN):* This is the proportion of the number of negative instances correctly classified as negative and can be calculated as: $TN = \frac{a}{a+b}$

iv) *False Positive Rate (FP*): This is the proportion of the number of negative instances incorrectly classified as positive and can be calculated as: $FP = \frac{b}{a+b}$

v) *False Negative rate (FN):* This is the proportion of the number of positive instances incorrectly classified as negative and can be calculated as: $FN = \frac{c}{c+d}$

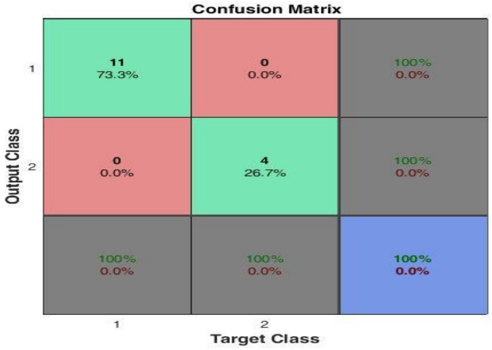
vi) *Precision (P):* This is the proportion of the number of correctly classified instances that are predicted as positive and can be calculated as: $P = \frac{d}{b+d}$

6.2 Preservation of Perceptron Classification Using Confusion Matrices

In this investigation, after partitioning a dataset *D* into training and test sets, a perceptron is built on the training set, and the weight vector obtained from the training phase is then used to classify the data points of the test set and the result of the classification is stored as Result1. The original dataset, D, is then reduced to fewer attributes using each of the fourteen dimensionality reduction techniques described above. A perceptron is also built for each of the reduced datasets (using the same size of training and test sets as in the original dataset, D) and then the weight vector obtained during the training phase of each of the reduced sets is used to classify the data points of the corresponding test set. The result of the classification is then saved as Result2.

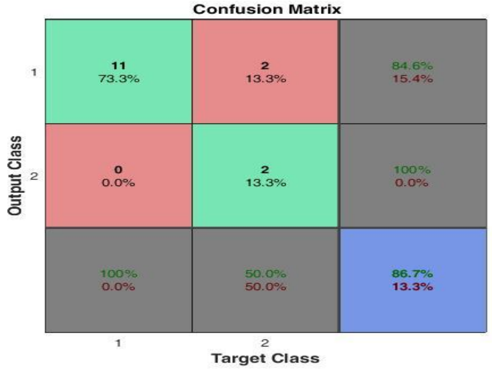
A confusion matrix is then used for the comparison of Result1 and Result2. This comparison gives us the extent to which the dimensionality reduction techniques preserve the classification of the original dataset using the perceptron.

Figures 6.1 and 6.2 below are *confusion matrices* which show the extent to which the perceptron classification are preserved by a reduction of the weather data set from 30 to 24 attributes using *PCA* and the *Variance* Approach.



	1	2	
1	11 73.3%	0 0.0%	100% 0.0%
2	0 0.0%	4 26.7%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%

Figure 6.1 *Confusion matrix* showing the extent to which the perceptron classification is preserved by a reduction of the Weather data set from 30 to 24 attributes using *PCA.*



	1	2	
1	11 73.3%	2 13.3%	84.6% 15.4%
2	0 0.0%	2 13.3%	100% 0.0%
	100% 0.0%	50.0% 50.0%	86.7% 13.3%

Figure 6.2 *Confusion matrix* showing the extent to which perceptron classification is preserved by a reduction of the Weather data set from 30 to 24 attributes using the *Variance* Approach.

In the confusion matrices presented in this paper, the first two diagonal cells of the 3 X 3 matrix represent the number and percentage of correctly classified instances for the first and second classes, and the third shows the overall percentage of correctly classified instances.

For example, in Figure 6.2 above, 11 instances are correctly predicted as positive which corresponds to 73.3% of all 15 instances of the test data. 2 of the negative instances are correctly classified as negative which represents 13.3% of the data. In total, 86.7% of the 15 instances are correctly predicted as positive or negative. This percentage is referred to as the accuracy of the prediction by the given confusion matrix. All of the positive instances are correctly classified, and this corresponds to 100% of the positive instances. Two of the four negative instances are correctly classified, and this corresponds to 50% of the negative instances. 84.6% of the 13 positive predictions are correct while 15.4% are wrong. Both of the 2 negative predictions are correct which corresponds to 100% of the negative predictions.

Apart from the two *confusion matrices* presented in this paper, we generated *confusion matrices* showing the extent to which the perceptron classification is preserved by a reduction of the weather data set from 30 to 24 attributes using all the other twelve reduction techniques. The *accuracy* results obtained from all fourteen techniques showed that:

- For a reduction from 30 to 24 attributes, PCA has the best performance in preserving the perceptron classification of the weather dataset, followed by the modified version of the New Top Down Approach. The Combined Approach, on the other hand, has the worst performance.
- All the other eleven reduction techniques are equally efficient in preserving the perceptron classification of the weather dataset.

6.3 Preservation of K-Nearest Neighbors Classification Using Confusion Matrices

For *k-nearest neighbors* classification, all the labels of the negative class (-1) of the *Weather* dataset are renamed to a class label "2", where 2 means "warm". As it was done with the perceptron above, after obtaining the result of the classification of the original test set, the *Weather* dataset is reduced using each of the dimensionality reduction techniques above and the results of classifying the reduced test sets are compared with the result of classifying the original test set to see the extent to which the techniques preserve classification using K-Nearest Neighbors.

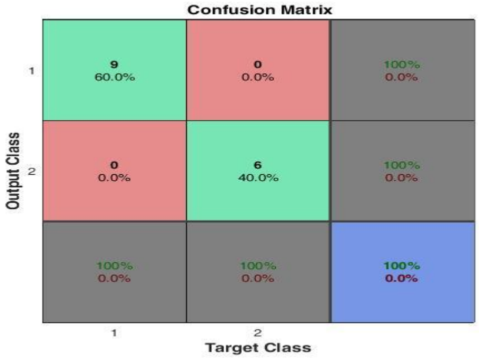


Figure 6.3 *Confusion matrix* showing the extent to which K-Nearest Neighbors classification is preserved by a reduction of the Weather data set from 30 to 24 attributes using any of the 14 approaches

The results in this section show that all the dimensionality reduction techniques perform much better at preserving the K-Nearest Neighbor classification of the *Weather* dataset than they do at preserving the classification of the *Weather* dataset using the perceptron.

In general, the dimensionality reduction techniques implemented in this paper proved to be very efficient in preserving the classification of different datasets using both the lazy and eager learners used for this investigation.

CONCLUSION

This paper started by pointing out the challenges faced in the extraction of useful information from available large pools of data which increases at an alarming rate. Dimensionality reduction was introduced as a method that provides a compact representation of an original high-dimensional data, thus making it a very powerful tool and also an invaluable preprocessing step in facilitating the implementation of many machine learning algorithms. We implemented many algorithms including fourteen dimensionality reduction techniques, two classification algorithms (the perceptron and K-Nearest Neighbors algorithms), the Rand Index and the confusion matrix. The results revealed the extent to which dimensionality reduction techniques preserve the perceptron and K-Nearest Neighbors classification of a given dataset.

The Rand Index and the confusion matrix were used to show the extent to which these fourteen dimensionality reduction techniques – compared against each other - preserve the perceptron and k-nearest neighbor classifications of the original datasets. This investigation revealed that the dimensionality reduction techniques implemented in this paper seem to perform much better at preserving K-Nearest Neighbors classification than they do at preserving the classification of the original datasets using the perceptron. In general, the dimensionality reduction techniques proved to be very efficient in preserving the classification of different datasets using both the lazy and eager learners used for this investigation.

It would be interesting and worth investigating the classification preservation of dimensionality reduction methods on more sophisticated classifiers like the support vector machine and decision trees.

REFERENCES

[1] N. Sharma and K. Saroha, “Study of dimension reduction methodologies in data mining,” in *International Conference on Computing, Communication and Automation*, 2015, pp. 133–137.

[2] I. K. Fodor, “A survey of dimension reduction techniques,” *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory*, no. 1, pp. 1–18, 2002.

[3] D. H. Deshmukh, T. Ghorpade, and P. Padiya, “Improving classification using preprocessing and machine learning algorithms on NSL-KDD dataset,” in *Proceedings - 2015 International Conference on Communication, Information and Computing Technology, ICCICT 2015*, 2015.

[4] A. S. Nsang, I. Diaz, and A. Ralescu, “Ensemble Clustering based on Heterogeneous Dimensionality Reduction Methods and Context-dependent Similarity Measures,” *Int. J. Adv. Sci. Technol.*, vol. 64, pp. 101–118, 2014.

[5] A. S. Nsang , F. Oguntoyinbo, H. Yusuf, and A. Maikori, “A New Random Approach To Dimensionality Reduction, in "Int’l Conf. on Advances in Big Data Analytics", pp. 69 – 74, 2015.

[6] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, and I. Chouvarda, “Machine Learning and Data Mining Methods in Diabetes Research,” *Comput. Struct. Biotechnol. J.*, vol. 15, pp. 104–116, 2017.

[7] T. M. Mitchell, *Machine Learning*, vol. 1, no. 3. 1997.

[8] S. B. Kotsiantis, “Supervised machine learning: A review of classification techniques,” *Informatica*, vol. 31, pp. 249–268, 2007.

[9] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, “Machine learning: A review of classification and combining techniques,” *Artif. Intell. Rev.*, vol. 26, no. 3, pp. 159–190, 2006.

[10] M. Capó, A. Pérez, and J. A. Lozano, “An efficient approximation to the K-means clustering for massive data,” *Knowledge-Based Systems*, 2016.

[11] Y. H. and W. Lam, “Lazy Learning for Classication Based on Query Projections,” in *Proceedings of the 2005 SIAM International Conference on Data Mining*, 2005, pp. 227–238.

[12] N. Singh, “Malware Analysis , Clustering and Classification : A Literature Review,” *IJCST Int. J. Comput. Sci. Technol.*, vol. 8491, pp. 68–72, 2015.

[13] I. M. Galván, J. M. Valls, M. García, and P. Isasi, “A lazy learning approach for building classification models,” *Int. J. Intell. Syst.*, vol. 26, no. 8, pp. 773–786, 2011.

[14] A. S. Nsang, A. M. Bello, and H. Shamsudeen, “Image Reduction Using Assorted Dimensionality Reduction Techniques,” in *Proceedings of the 26th Modern Artificial Intelligence and Cognitive Science Conference*, pp 139 - 146, 2015.

[15] E. Bingham and H. Mannila, “Random projection in dimensionality reduction: Applications To Image And Text Data,” *Int. Conf. Knowl. Discov. Data Min.*, pp. 245–250, 2001.

[16] Augustine S. Nsang and Anca Ralescu. Approaches to Dimensionality Reduction to a Subset of the Original Dimensions. In *Proceedings of the Twenty-First Midwest Artificial Intelligence and Cognitive Science Conference*, 70-77, 2010.

[17] Augustine Nsang. *Novel Approaches to Dimensionality Reduction and Applications: An Empirical Study*. Lambert Academic Publishing, Saarbrücken, Germany, 2011.

[18] A. S. Nsang, D. Edi, and C. Ahanonu, "Query-Based Dimensionality Reduction Applied To Images," in *Int'l Conf. on Advances in Big Data Analytics*, 2015, no. 2, pp. 81–86.

[19] L. E. Peterson, "K-nearest neighbors," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[20] W. Ertel, *Introduction to Artificial Intelligence*. 2011.

[21] L.-Y. Hu, M.-W. Huang, S.-W. Ke, and C.-F. Tsai, "The distance function effect on k-nearest neighbor classification for medical datasets," *Springerplus*, vol. 5, no. 1, p. 1304, 2016.

[22] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.

[23] S. Haykin, *Neural Networks and Learning Machines*, vol. 3. 2008.

[24] S. Haykin, "Rosenblatt's Perceptron," *Neural Networks Learn. Mach.*, no. 1943, pp. 47–67, 2009.

[25] K. Bache and M. Lichman, "UCI Machine Learning Repository," *University of California Irvine School of Information*, vol. 2008, no. 14/8. p. 0, 2013.

[26] J. M. Santos and M. Embrechts, "On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification.pdf," in *19th International Conference on Artificial Neural Networks*, 2009, pp. 1–10.

[27] S. Visa, B. Ramsay, A. Ralescu, and E. Van Der Knaap, "Confusion matrix-based feature selection," in *CEUR Workshop Proceedings*, 2011, vol. 710, pp. 120–127.

[28] S. Singh and R. Singla, "Comparative Performance of Fault-Prone Prediction Classes with K-means Clustering and MLP," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, 2016.

[29] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, pp. 427–437, 2009.

[30] U. A. Baba, A. S. Nsang and O. Adeseye. "Three Novel Approaches to Dimensionality Reduction." In *Proceedings of the International Conference of Artificial Intelligence,* Las Vegas, 2018.