

Machine Learning Approach to Identifying the Dataset Threshold for the Performance Estimators in Supervised Learning

Zanifa Omary, Fredrick Mtenzi
Dublin Institute of Technology
Ireland

Abstract

Currently for small-scale machine learning projects, there is no limit which has been set by its researchers to categorise datasets for inexperienced users such as students while assessing and comparing performance of machine learning algorithms. Based on the lack of such a threshold, this paper presents a step by step guide for identifying the dataset threshold for the performance estimators in supervised machine learning experiments. The identification of the dataset threshold involves performing experiments using four different datasets having different sample sizes from the University of California Irvine (UCI) machine learning repository. The sample sizes are categorised in relation to the number of attributes and number of instances available in the dataset. The identified dataset threshold will help unfamiliar machine learning experimenters to categorise datasets correctly and hence selecting the appropriate performance estimation method.

Keywords: machine learning, machine learning algorithms, dataset threshold, performance measures, supervised machine learning.

1. Introduction

In recent years, the goal of many researchers' in different fields has been to build systems that can learn from experiences and adapt to their environments. This evolution has resulted into an establishment of various algorithms such as decision trees, K-Nearest Neighbours (KNN), Support Vector Machines (SVM) and Random Forests (RF) that are transforming problems rising from industrial and scientific fields. Based on the nature of the dataset, either balanced or unbalanced, different performance measures and estimation methods tend to perform differently when applied to different machine learning algorithms. The available performance measures, such as accuracy, error rate, precision, recall, f1-score and ROC analysis, are used while assessing and comparing one machine learning algorithm from the other. In addition to machine learning performance measures, there are various statistical tests, such as McNemar's test and a test of the difference of two proportions, also used to assess and compare classification algorithms. Authors of this paper describe three machine learning

performance estimation methods these are, hold-out method, k-fold cross validation and leave-one out cross validation. The performance of these estimators depends on the number of instances available in the dataset. From research literature, the holdout method has been identified to work well on very large datasets, but nothing has been identified for the remaining two estimators. Therefore, for this paper we will identify the dataset threshold for the remaining two estimators.

In this paper we present the results of the experiments performed using four different datasets from UCI machine learning repository together with two performance estimators. The accuracy of the dataset with all instances will be regarded as the threshold, that is, the minimum value for the two performance estimators. Only one performance measure, f1- score, and one machine learning algorithm, decision tree, together with two performance estimators will be used in the experiment for identifying the dataset threshold.

The rest of this paper is organised as follows. Section 2 provides the background of machine learning where its definition, categories and the review of the machine learning classification techniques will be provided. Section 3 provides the discussion on classification evaluations where performance measures in machine learning and statistical tests together with performance estimation methods will be covered. Experiments for identifying the dataset threshold will be covered in section 4 followed by the results of the dataset threshold in section 5. Conclusion of the paper will be provided in section 6.

2. Background

In this section, the background of machine learning will be provided. The section is divided into three subsections; machine learning definition will be provided in the first section followed by the discussion on its categories in second subsection. The review of classification techniques will be provided on the third and last subsection.

2.1. What is Machine learning?

Prior to delving into formal definitions of machine learning it is worthwhile to define, in Information and Communication Technology (ICT) context, the two terms that make up machine

learning; that is, machine or computer and learning. Defining these terms will be a guideline on the selection of appropriate machine learning definition for this paper.

According to Oxford English Dictionary, computer is a machine for performing or facilitating calculation; it accepts data, manipulates them and produces output information based on a sequence of instructions on how the data has to be processed. Additionally, learning can be defined as a process of acquiring modifications in existing skills, knowledge and habits through experience, exercise and practice. From the identified learning definition, Witten and Frank [1] argue that “things learn when they change their behaviour in a way that makes them perform better in the future”. From Witten and Frank’s definition, learning can be tested by observing the current behaviour and comparing it with past behaviour. Therefore, a complete definition of machine learning for this paper has to incorporate two important elements that are; computer based knowledge acquisition process and has to state where skills or knowledge can be obtained.

Mitchell [2] describes machine learning as a study of computer algorithms that improve automatically through experience. This means computer programs use their experience from past tasks to improve their performance. As we identified previously there are two important elements that any machine learning definition has incorporate in order to be regarded as appropriate for this paper, however this definition does not reflect anything related to knowledge acquisition process for the stated computer programs, therefore it is considered insufficient for this paper.

Additionally, Alpaydin [3] defines machine learning as “the capability of the computer program to acquire or develop new knowledge or skills from existing or non existing examples for the sake of optimising performance criterion”. Contrary to the Mitchell’s definition which lacks knowledge acquisition process, this definition is of more preference to this paper as it includes two elements identified previously that is; knowledge acquisition process and it indicates where skills or knowledge can be obtained.

Over the past 50 years, machine learning as any field of study has grown tremendously [4]. The growing interest in machine learning is driven by two factors as outlined by Alpaydin [3], removing tedious human work and reducing cost. As the result of automation of processes, huge amounts of data are produced in our day-to-day activities. Doing manual analysis on all of this data is slow, costly and people who are able to do such analysis manually are rare to be found [5]. Machine learning techniques, when applied to different fields such as in medical diagnosis, bio-surveillance, speech and handwriting recognition, computer vision and detecting credit

card fraud in financial institutions, have proved to work with huge amounts of data and provide results in a matter of seconds [3, 4]. In the next section a review of the two machine learning categories is provided.

2.2. Machine learning categories

Machine learning can be categorised into two main groups that is, supervised and unsupervised machine learning. These two learning categories are associated with different machine learning algorithms that represent how the learning method works.

- **Supervised learning:** Supervised learning comprises of algorithms that reason from externally supplied instances to produce general hypothesis which then make predictions about future instances. Generally, with supervised learning there is a presence of the outcome variable to guide the learning process. There are several supervised machine learning algorithms such as decision trees, K-Nearest Neighbour (KNN), Support Vector Machines (SVM) and Random Forests [6]. These algorithms will be briefly described in the next section.

- **Unsupervised learning:** Contrary to supervised learning where there is a presence of the outcome variable to guide the learning process, unsupervised learning builds models from data without predefined classes or examples [7]. This means, no “supervisor” is available and learning must rely on guidance obtained heuristically by the system examining different sample data or the environment [2, 8]. The output states are defined implicitly by the specific learning algorithm used and built in constraints [7].

2.3. Machine Learning Algorithms

Although, there are various machine learning algorithms depending on the application domain; only four techniques, that is decision tree, k-nearest neighbour, support vector machines and random forest, will be discussed. These four are enough to give readers’ an understanding of the variations in approaches present in various supervised machine learning algorithms taken to classification.

- **Decision tree:** Decision tree is defined as “as a hierarchical model based on nonparametric theory where local regions are identified in a sequence of recursive splits in a smaller number of steps that implements divide-and-conquer strategy used in classification and regression tasks”[3]. As indicated in figure 1, the hierarchical structure of the decision tree is divided into three parts that is; root node,

internal nodes and leaf nodes. From the presented decision tree of the golf concept; *outlook* is the root node, *wind* and *humidity* are internal nodes while *yes/no* are leaf nodes. The process starts at the root node, and is repeated recursively until the leaf node is encountered. The leaf node provides the output of the problem.

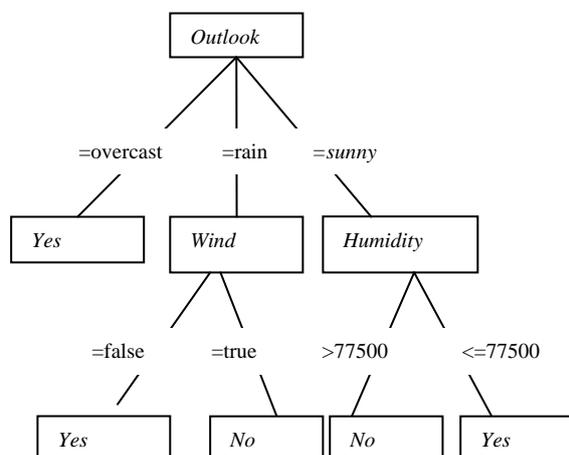


Figure 1: Decision tree for the golf concept

- **K-Nearest Neighbour (KNN):** K-Nearest Neighbour abbreviated as KNN is one among the methods referred to as *instance-based* learning which falls under the supervised learning category [2]. KNN works by simply storing the presented training data; when a new query or instance is fired, a set of similar related instances or neighbours is retrieved from memory and used to classify the new instance [2, 8]. While classifying, it is often useful to take more than one neighbour into account and hence referred to as k-nearest neighbour [9]. The nearest neighbours to an instance are measured in terms of the Euclidean distance, which measures the dissimilarities between examples represented as vector inputs, and some other related measures. However, the basis for classifying a new query using Euclidean distance is that, instances in the same group are expected to have a small separating distance compared to instances that fall under different groups.

- **Support Vector Machine (SVM):** is a relatively new machine learning technique proposed by Vladimir Vapnik and colleagues at AT&T Bell laboratories in 1992 and it represents the state of the art in machine learning techniques. The general idea of the SVM is to find separating hyperplanes between training instances that maximize the margin and minimize the classification errors [10]. Margin or sometimes referred to as geometric margin is referred to as *the “distance between the hyperplanes separating two classes and the closest data points to*

the hyperplanes” [11]. The SVM algorithm is capable of working with both linearly and nonlinearly separable problems in classification and regression tasks.

- **Random Forests:** Breiman [12] defines a random forest as a *classifier consisting of a collection of tree-structured classifiers* $\{h(x), Q_k, k=1\dots\}$ where the $\{Q_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for most popular class at input x . This technique involves the generation of an ensemble of trees that vote for the most popular class [12]. Although there are several supervised machine learning techniques, random forest has two distinguishing characteristics; firstly, the generalisation error converges as the number of trees in the forest increases and the technique does not suffer from overfitting [12]. Accuracy of the individual single trees that make up a forest enforces the convergence of the generalisation errors and hence improvement in classification accuracy.

As the main aim of this paper is to identify the dataset threshold for performance estimators in supervised machine learning experiments then the next section provides a review of the classification evaluation methods. Some of these methods will be referred in experiments section.

3. Classification evaluations

While assessing and comparing performance of one learning algorithm over the other, accuracy and error rate are among the common methods that are widely used. Other evaluation factors include speed, interpretability, ease of programmability and risk when errors are generalised [3, 13]. This section describes such evaluation methods that are used by machine learning researchers while comparing and assessing the performance of the classification techniques. The authors will also integrate machine learning and statistics by introducing statistical tests for the purpose of evaluating the performance of the classifier and for the comparison of the classification algorithms. The first part of this section provides the review of machine learning performance measures which include accuracy, error rate, precision, recall, and f1-score and ROC analysis. The second section will cover statistical tests.

3.1. Machine Learning Performance Measures

In machine learning and data mining, the preferred performance measures for the learning algorithms differ according to the experimenter’s viewpoint [14]. This is much associated with the background of the experimenter as either in machine

learning, statistics or any other field as well as an application domain where the experiment is carried out. In some application domains, experimenters' are interested in using accuracy and error rate while to others precision, recall and f1-score are of preference. This section provides the discussion of the performance measures used in machine learning and data mining projects.

- **Accuracy:** Kostiantis [15] defines accuracy as “the fraction of the number of correct predictions over the total number of predictions”. The number of predictions in classification techniques is based upon the counts of the test records correctly or incorrectly predicted by the model. As indicated in table 1, these counts are tabulated into a *confusion matrix* (also referred as contingency) table where true classes are presented in rows while predicted classes are presented in columns. The confusion matrix shows how the classifier is behaving for individual classes.

Table 1: Confusion matrix table

TRUE CLASS	PREDICTED CLASSES	
	YES	NO
YES	TP	FN
NO	FP	TN

TP= True Positives FP= False Positives
 TN= True Negatives FN= False Negatives

TP Indicates to the number of positive examples correctly predicted as positive by the model.

TN Indicates the number of negative examples correctly predicted as negative by the model

FP Indicates the number of negative examples wrongly predicted as positive by the model.

FN Indicates the number of positive examples wrongly predicted as negative examples by the model.

Equation 1

$$\text{Accuracy} = \frac{\text{number of correct predictions}}{\text{Total number of predictions}}$$

As indicated in equation 1, accuracy only measures the number of correct predictions of the classifier and ignores the number of incorrect predictions. With this limitation, error rate was introduced to measure the number of incorrect

predictions relating to the performance of the classifier.

- **Error rate:** As described previously, error rate measures the number of incorrect predictions against the number of total predictions. As for some applications it is of interest to know how the system responds to wrong answers. This has been the motive behind the introduction of error rate [16]. Computationally, in relation to accuracy, error rate is just 1- Accuracy on the training and test examples [8]. It is an appropriate performance measure(s) for the comparison of the classification techniques given balanced datasets. Precision, recall and f1-score are appropriate performance measures for unbalanced datasets. Equation 2 presents the formula of calculating error rate.

Equation 2

$$\text{Error rate} = \frac{\text{Number of incorrect predictions}}{\text{Total number of predictions}}$$

As most of the datasets used in our daily lives are unbalanced, that is, there is an imbalanced distribution of classes; there is a need of having different classification evaluation factors for different types of datasets. Precision, recall, f1-score and ROC analysis are the metrics which work well with unbalanced datasets [17].

- **Precision:** In the area of information retrieval (IR) where datasets are much unbalanced, precision and recall are the two most popular metrics for evaluating classifiers [17, 18]. However, precision is used in many application domains where the detection of one class seems to be much more important than the other such as in medical diagnosis, pattern recognition, credit risks and statistics. As indicated in equation 3, it represents the proportion of selected items that the system got right [17] as the positive examples to the total number of true positive and false positives examples.

Equation 3

$$\text{Precision, } p = \frac{TP}{TP + FP}$$

- **Recall:** It represents the proportion of the number of items that the system selected as the positive examples to the total number of true positives and false negatives [17]. Contrary to equation 3, where false positive is used, recall, as indicated in equation 4, uses false negatives. Recall is supposed to be high in order to reduce the number of positive examples wrongly predicted as negative examples.

Equation 4

$$\text{Recall, } r = \frac{TP}{TP + FN}$$

Manning and Schutze [17] argue on the advantage of using precision and recall over accuracy and error rate. Accuracy refers to things got right by the system while error rate refers to things got wrong by the system. As indicated in equation 1 and 2 respectively, accuracy and error rate are not sensitive to any of the TP, FP and FN values while precision and recall are. However, for this behaviour, there is a possibility of getting high accuracy while selecting nothing. Therefore, as we are surrounded by unbalanced dataset and the biasness of the accuracy and error rate over TP, FP and TN values; accuracy and error rate are usually replaced by the use of precision and recall unless the dataset is really balanced.

Additionally, in some applications, there is a trade-off between precision and recall. Where as in selecting a document in information retrieval for example, one can get low precision but very high recall of up to 100% [17]. Indeed, it is difficult to evaluate algorithm with high precision and low recall or otherwise. Therefore, f1-score, which combines precision and recall, was introduced.

- **F1-Score:** It combines precision and recall with equal importance into a single parameter for optimization and is defined as

Equation 5

$$\text{F1 - Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Receiver Operating Characteristic (ROC) graph**

Fawcett [18] defines ROC graph as “a technique for visualising, organising and selecting classifiers based on their performance in a 2D space”. Despite having several definitions, Fawcett’s definition has been adopted for this book as it shows directly where the technique is used and in which space. Originally conceived during World War II to assess the capabilities of radar systems, ROC graphs which uses area under the ROC curves abbreviated as AUC-ROC have been successful applied in different areas such as in signal detection theory to depict hit rate and false alarm rates, medical decision making, medical diagnosis, experimental psychology and psychophysics and in pattern recognition [18].

The difference with the previous performance measures is that, ROC graphs are much more useful for domains with skewed class distribution and unequal classification error costs [18]. With this ability, ROC graphs are much more preferred than accuracy and error rate. ROC graphs are plotted using two parameters; TP rate (fraction of true positives) or *sensitivity* which is plotted on the Y axis

and FP rate (fraction of false positives) or *1-specificity* plotted in X axis as presented in figure 2. When several instances are plotted on a graph then a curve known as ROC curve is drawn. The points on the top left of the ROC curve have high TP rate and low FP rate and so represent good classifiers [19].

Equation 6

True Positive Rate (TPR) or sensitivity

$$\text{TPR} = \frac{TP}{TP + FN}$$

Equation 7

True Negative Rate (TNR) or specificity

$$\text{TNR} = \frac{TN}{TN + FP}$$

To compare classifiers we may want to reduce the ROC performance to a single scalar value representing expected performance. The common method for reducing the ROC performance is to measure the area under the ROC curve. After drawing the ROC curves of different classifiers, the best classifier is supposed to be nearby top left of the ROC curve. Figure 2 is an example of ROC graph for the comparison of three classifiers; SLN which is a traditional neural network, SVM and C4.5 rules

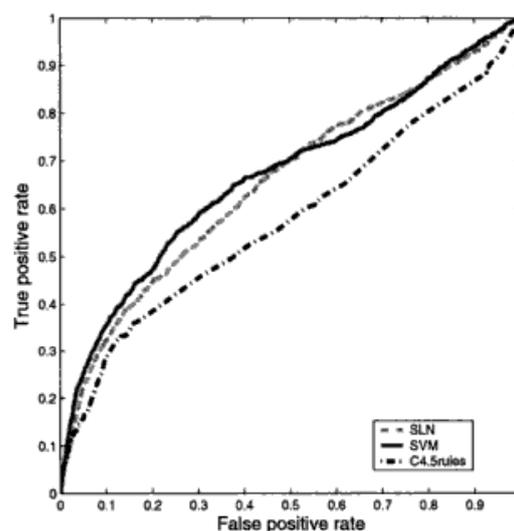


Figure 2: ROC curve for the comparison of three classifiers [19]

3.2. Statistical tests

The classifiers induced by machine learning algorithms depend on the training set for the measurement of its performance. Statistical tests come into play when assessing the expected error

rate of the classification algorithm or comparing the expected error rates of two classification algorithms. Though there are many statistical tests, only five approximate statistical tests for determining whether one learning algorithm outperforms another will be considered. These are McNemar’s test, a test of the difference of two proportions, resampled paired t test, k-fold cross validated paired t test and the 5 x 2 cross validated paired t test.

• **McNemar’s Test:**

Is a statistical test named after Quinn McNemar (1947) for comparing the difference between proportions in two matched samples and analysing experimental studies. It involves testing paired dichotomous measurements; “measurements that can be divided into two sharply distinguished parts or classifications” such as yes/no, presence/absence, before/after. The paired responses are fabricated in a 2 x 2 contingency table and the responses are tallied in appropriate cells.

This test has been widely applied in a variety of applications to name a few; in marketing while observing brand switching and brand loyalty patterns for the customers, measuring the effectiveness of advertising copy or advertising a campaign strategy, studying the intent to purchase versus actual purchase patterns in consumer research, public relations, operational management and organisational behaviour studies and in health services.

Consider the application of McNemar’s test in health institutions for example, where specific number of patients is selected at random based on their visits to a local clinic and assessed for a specific behaviour that is classified as risk factor for lung cancer. The classification of the risk factor is either present or absent. During their visits to the clinic they are educated about the incidence and associated risks for lung cancer. Six months later the patients are evaluated with respect to the absence or presence of the same risk factor. The risk factor before and after instructions can be tallied as tabulated in table 3 and evaluated using McNemar’s test.

	Risk factor before Instructions
Risk factor after Instruction	

Table 2: Matched paired data for the risk factors before and after instructions

Where

e_{00} : The number of patients’ that shows the presence of the risk factor for Response 1 and Response 2.

e_{01} : The number of patients’ shows the *absence* of the risk factor for Response 1 and the *presence* of the risk factor for Response 2.

e_{10} : The number of patients’ shows the *presence* of the risk factor for Response 1 and the *absence* for Response 2.

e_{11} : The number of patients’ responded for the absence of the risk factor for Response 1 and Response 2.

$e_{00} + e_{01} + e_{10} + e_{11}$ Represent the total number of examples in the test set.

Under the null hypothesis the change in risk factors; from presence to absence and vice versa should have the same error rates, which means

$$e_{01} = e_{10} \text{ [20]}$$

For McNemar’s, the statistic is as follows

Equation 8

$$x^2_{McNemar} = \frac{(e_{01} - e_{10})^2}{e_{01} + e_{10}}$$

In a 2 x 2 contingency table with 1 degree of freedom (1-column x 1-row), that is having one column and one row, the statistic for the McNemar’s test changes to

Equation 9

$$x^2_{McNemar} = \frac{(|e_{01} - e_{10}| - 1)^2}{e_{01} + e_{10}}$$

		Response 1		
	Response 2	Present	Absent	Total
The null hypothesis would identify that	Present	e_{00}	e_{01}	$e_{00} + e_{01}$
	Absent	e_{10}	e_{11}	$e_{10} + e_{11}$
	Total	$e_{00} + e_{10}$	$e_{01} + e_{11}$	$e_{00} + e_{10} + e_{01} + e_{11}$

there is no significant change in characteristics between the two times (as in table 2 for example, before and after instructions). Thus we will compare our calculated statistic with a critical χ^2, α with 1 degree of freedom or 3.84. If the $\chi^2_{McNemar} > 3.84$, the null hypothesis is rejected and assumes a significant change in the two measurements.

Everitt [21] comments on how to apply McNemar test for the comparison of the classifiers. Having available sample of data S divided into training set and testing set, both algorithms A and B are trained on the training set which results in two classifiers P_1 and P_2 . These two classifiers are then tested using the test set. The *contingency table*, provided in table 1, is used to record how each example has been classified.

If the null hypothesis is correct then, the probability that the value for the χ^2 with 1-degree of freedom is greater than 3.84 is less than 0.05 and the null hypothesis may be rejected in favour of the hypothesis that the two algorithms have different performance measurements when trained in a particular training set.

Dietterich [20] comments on the advantage of using this test compared to other statistical test as such Mc Nemar's test has been yielded to provide low type 1 error. Type 1 error means *ability to incorrectly detect differences while there is no difference that exists* [20]. Despite having aforementioned advantages, this test is associated with several problems. Firstly, a single training set is used for the comparison of the algorithms and hence the test does not measure the variations due to the choice of the training data. Secondly, Mc Nemar's test is a simple holdout test, where by having available sample data; test can be applied after the partition of the data into training set and testing set. For the comparison of the algorithms, the performance is measured using the training data rather than the whole sample of data provided.

Mc Nemar's test as a performance measure for the comparison of the algorithms from different application domains is associated with several shortcomings. These shortcomings have resulted into the growth of other statistical tests such as a test for the difference of two proportions, the resampled t test, k-fold cross validated t-test and 5 x 2 CV paired t test.

- **A Test for the Difference of Two Proportions**

A test for the difference of two proportions measures the difference between the error rate of algorithm A and the error rate of algorithm B [20]. Consider for example, P_A be the proportion of the test examples

incorrectly classified by algorithm A and P_B be the proportion of the test examples incorrectly classified by algorithm B,

Equation 10

$$P_A = \frac{e_{00} + e_{01}}{e}, \quad P_B = \frac{e_{00} + e_{10}}{e}$$

The assumption underlying this statistical test is that when algorithm A classifies an example n from test set the probability of misclassification is P_A . Hence, the number of misclassification for n test examples is a binomial distribution with mean is nP_A .

This statistical test is associated with several problems, firstly as P_A and P_B are measured on the same test set, they are not independent. Secondly, the test does not measure the variations due to the choice of the training set or the internal variation of the algorithm. Lastly, this test suffers with the same problem as McNemar test; does not measure the performance of the algorithm in the whole dataset (with all sample size) provided; rather it measures the performance on the smaller training data after partition.

- **The Resampled Paired t Test**

With this statistical test, usually a series of 30 trials is conducted, in each trial, the available sample data is randomly divided into training set of specified size and testing set [20]. Learning algorithms are trained on the training set and the resulting classifiers are tested on the test set.

Consider, P_A and P_B be the proportion of test examples misclassified by algorithm A and algorithm B respectively. For the 30 trials we will result into having 30 differences

Equation 11

$$P^i = P_A^{(i)} - P_B^{(i)} \quad [20]$$

Among the potential drawbacks of this approach is, the value of the differences (P^i) are not independent because the training and testing sets in the trials overlap.

- **The k-fold cross validated Paired t test**

The k-fold cross validated paired t test was introduced to overcome the problem underlined by the resampled paired t test that is, overlapping of the trials. This test works by dividing the sample size into k disjoint sets of equal size $T_1 \cdots T_k$ and then k

trials are conducted. In each trial, the test set is T_i and the training set is the union of all the other sets.

This approach is advantageous as each test set is independent of the others. However this test suffers from the problem that the training data overlap [20]. Consider for example, when $k=10$, in a 10-fold cross validation, each pair of the training set shares 80% of the examples [3]. This overlapping behaviour may prevent this statistical test from obtaining a good estimate of the variation that would be observed if each training set were completely independent of the previous training sets.

• **The 5 x 2 cross validated Paired t Test**

With this test, 5 replications of the twofold cross validation are performed [3]. In each replication, the available data are partitioned into two equal sized sets, let's say S_1 and S_2 . Each learning algorithm is trained on one set and tested on the other set and this results into four error estimates as shown in figure 3.

The choice of the number of replications is not the responsibility of the experimenter; this is how the test requires. The test allows the applications of only five replications in a twofold cross validation as exploratory studies shows that, the use of more or less of five replications increases the risk of type I error which is supposed to be low for the betterment of the test [20].

This test has one disadvantage, in each fold the training set equals the testing set and hence results into learning algorithms to be trained in training sets half the size of the whole training sets [20]. For better performance of the learning algorithm, training set is supposed to be larger than the training set.

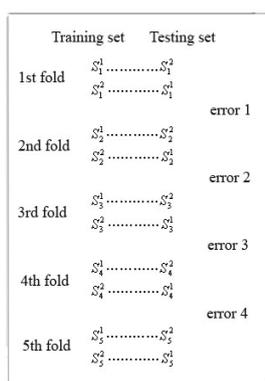


Figure 3: 5 x 2 cross validation (Adapted from [3])

3.3. Performance estimation methods

In this subsection we review three performance estimation methods namely, hold-out method, k-fold cross validation and leave one out method. These methods are used to estimate the performance of the machine learning algorithms.

- **Hold out method:** The holdout or sometimes called test set estimation [13] works by randomly dividing data into two mutually exclusive subsets; training and testing or holdout set [22, 23]. As shown in figure 4, two-third (2/3) of all data is commonly designated for the training and the remaining one-third, 1/3, for testing the classifier. The holdout method is repeated k times and the accuracy is estimated by averaging the accuracies obtained from each holdout [22]. However, the more instances left out for test set, the higher the bias of the estimate [22]. Additionally, the method makes an inefficient use of data which inhibits its application to small sample sizes [14].

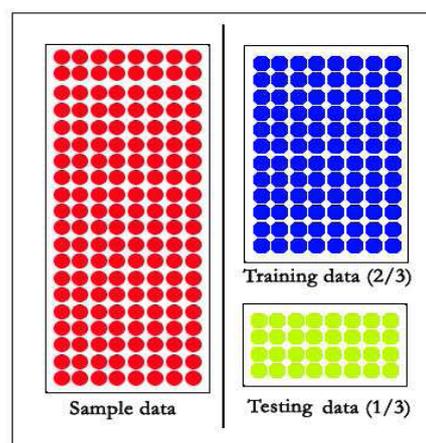


Figure 4: Process of dividing data into training set and testing set using the holdout method (Source: Authors)

- **K-Fold Cross Validation:** With K-fold cross validation, the available data is partitioned into k separate sets of approximately equal size [13]. The cross validation procedure involves k iterations in which the learning method is given $k-1$ as the training data and the rest used as the testing data. Iteration leaves out a different subset so that each is used as the test set once [13]. The cross-validation is considered as a computer intensive technique, as it uses all available examples in the dataset as training and test sets [14]. It mimics the use of training and test sets by repeatedly training the algorithm k times with a function $1/k$ of training examples left out for testing purposes. It is regarded as the kind of holdout test estimate.

With this strategy of k-fold cross validation, it is possible to exploit much larger dataset compared to leave-one out method. However, since the training and testing is repeated k times with different parts of the original dataset, it is possible to average all test errors (or any performance measure used) in order to obtain a reliable estimate of the model performance on the newly test data [24].

- **Leave one out cross validation:** It is also referred to as n fold cross validation where n is the number of instances [25]. For instance, given the dataset with n cases, one observation is left out for testing and the rest $n-1$ cases for training [26]. Each instance is left out once and the learning algorithm is trained on all the training instances. The judgement on the correctness of the learning algorithm is based on the remaining instances. The results of all n assessments, one for each instance, are averaged and the obtained average represents the final error estimate of the classification algorithm.

The method is attractive as there is a greatest possible amount of data which is used for training in each case, this increases the possibility of having accurate classifier [25]. Additionally, the method tends to simplify repetition which is performed in k-fold cross validation (repeated 10times for 10-fold cross validation, for example) as the same results are obtained every time.

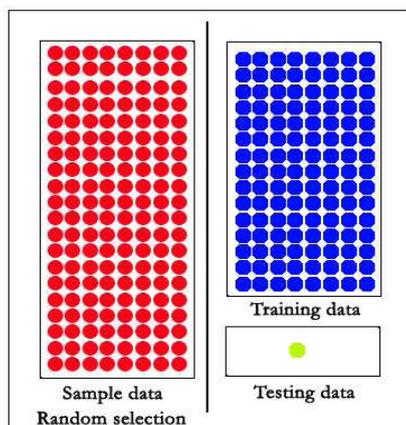


Figure 5: Process of randomly selecting a data sample for use in the test set with the remaining data going towards training

4. Experiments

In this section experiments for identifying the dataset threshold for performance estimators will be performed and results will be presented in the next section.

- **Experimental setting and methodology**

From the research literature, hold out method has been identified to work well on very large datasets, but nothing has been identified for the remaining two performance estimators. As previously discussed, the main aim of this paper is to determine the dataset threshold for supervised machine learning experiments. The established dataset threshold will help unfamiliar machine learning experimenters to

decide appropriate performance estimation method for the dataset based on the number of instances. To achieve this, experiments will be performed using one supervised machine learning algorithm that is, decision tree, four datasets with different sample sizes (range from 4177 to 1000 instances) from UCI machine learning repository together with two performance estimation methods (10 fold cross validation and leave one out). Performance of estimation methods will be measured using f1-score. The experiments will be carried out using an open source machine learning software called RapidMiner.

Datasets will be randomly divided to create small datasets with different sample sizes. Performance of estimation methods will be carried out for each randomly created dataset. The accuracy of the dataset will be observed and will be considered as the threshold or the minimum value for performance estimation methods. Differences in performance for the two estimation methods will then be analysed and plotted in order to identify which performance estimation method works better than the other.

4.1 Abalone dataset

The first experiment involves the use of the Abalone dataset. The accuracy threshold between the two values has been calculated and 0.5979 is obtained. The result of this experiment is shown in Table 3. In figure 6, the line crossing the two performance estimators, with value 0.5979, indicates the accuracy threshold. Analysis from figure 6 however, indicates 10 fold CV outweighs leave one out method when dataset has 4177 instances.

Sample Size	F1-Score		Difference (f1 score)
	10 fold CV	Leave one out	
4177	0.7448	0.7452	-0.0004
2006	0.7502	0.7502	0
1000	0.7307	0.7334	-0.0027
750	0.6442	0.6473	-0.0031
500	0.6603	0.6599	0.0004
250	0.6514	0.608	0.0434
100	0.6216	0.5941	0.0275
50	0.5455	0.4878	0.0577

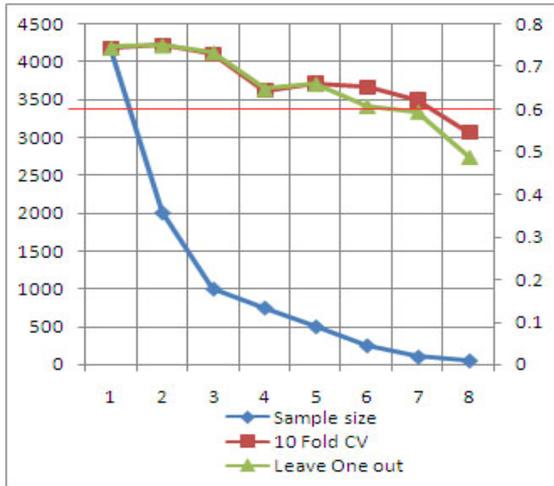


Figure 6: Line graph for the Abalone dataset

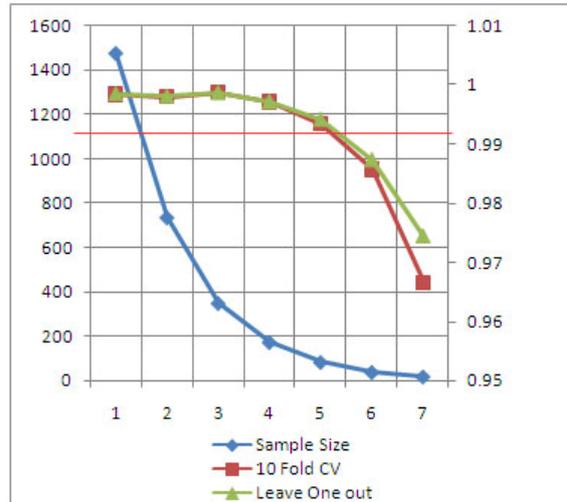


Figure 7: Line graph for the contraceptive method dataset

4.2. Contraceptive method choice

The second experiment for the establishment of the dataset threshold involves the dataset with 1473 instances and 10 attributes. The accuracy threshold for the performance estimation methods is 0.9966. Results have been presented in table 4. From figure 7, it can be concluded that, for the dataset with 1473 instances leave one out method is appropriate performance estimation method compared to 10 fold CV.

Table 4: Results for the 10 fold CV and leave one out estimation for the Contraceptive method choice

Sample Size	F1 Score		Difference (f1 score)
	10 Fold CV	Leave One Out	
1473	0.9983	0.9984	-0.0001
735	0.9979	0.9980	-0.0001
350	0.9986	0.9986	0
175	0.9970	0.9971	-0.0001
85	0.9933	0.9941	-0.0008
40	0.9857	0.9873	-0.0016
20	0.9667	0.9744	-0.0077

4.3. Ozone Level Detection Dataset

The third dataset comprise of 2536 instances and 73 attributes. The accuracy threshold obtained for this dataset is 0.7856. Results have been presented in table 4. From figure 6, it can be concluded that, for dataset with 2536 instances, leave one out method performs better compares to 10 fold cross validation.

Table 5: Results for the 10 fold CV and leave-one-out estimation for the Ozone layer dataset

Sample Size	F1 Score		Difference (f1 score)
	10 Fold CV	Leave One Out	
2536	0.8799	0.8800	-0.0001
1268	0.8804	0.8805	-0.0001
634	0.8858	0.8858	0
317	0.8759	0.8759	0
158	0.8911	0.8912	-0.0001
79	0.8101	0.8120	-0.0019
40	0.8190	0.8235	-0.0045
20	0.8000	0.8235	-0.0235

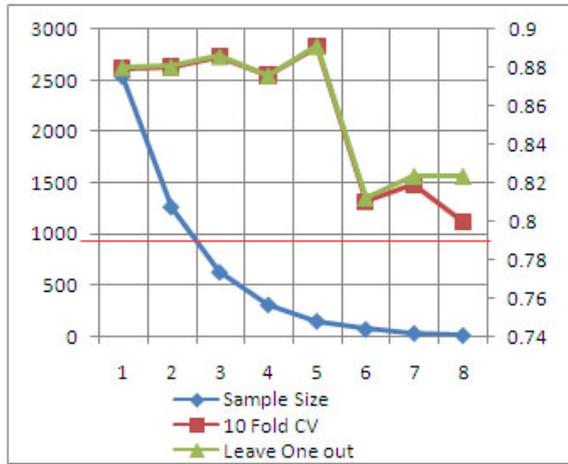


Figure 8: Line graph for the Ozone level detection dataset

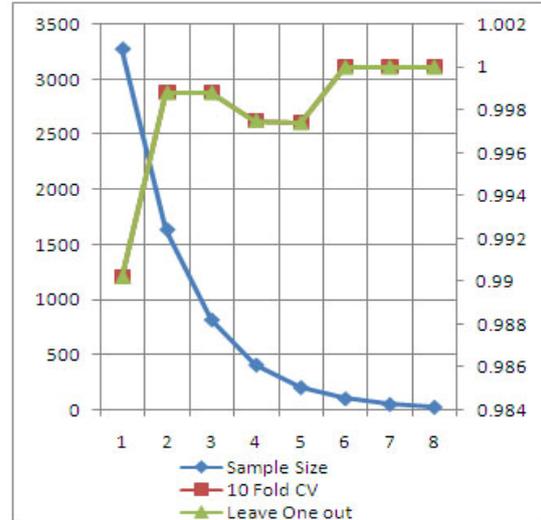


Figure 9: Line graph for the Internet advertisement dataset

4.4. Internet advertisement

This is the last experiment which will determine the dataset threshold for the two performance estimators. From the previous subsections, experiments have been performed for the dataset with 4177, 2536 and 1473 instances and the performance estimators obtained are k-fold cross validation for the first dataset while the other two, leave-one-out cv has been identified as the appropriate performance estimation method. This experiment involves the use of the dataset with instances that lie between the obtained results. This dataset contains 3279 instances and 1558 attributes.

5. Dataset threshold result

As previously discussed, the principal aim of performing these experiments was to establish number of instances which can result into the classification of the dataset as small, medium or large. However, from the previous literature, hold out method has been identified to work well with large datasets but nothing has been done for k-fold cross validation and leave one out method. Summary of the results from the experiments performed is presented in table 7.

Sample Size	F1 Score		Difference (f1 score)
	10 Fold CV	Leave One Out	
3279	0.9902	0.9902	0
1639	0.9988	0.9988	0
819	0.9988	0.9988	0
409	0.9975	0.9975	0
204	0.9974	0.9974	0
102	1.000	1.000	0
51	1.000	1.000	0
25	1.000	1.000	0

Sample Size (number of instances)	Method
4177	k-fold CV
3279	Neutral
2536	Leave one out
1473	Leave one out

From results indicated in table 6 and figure 9 respectively, there is no any difference between the two performance estimation methods.

From table 7, with 4177 instances k-fold cross validation outweighs leave one out method and this means for this number of instances, k-fold is the appropriate method. With 2536 and 1473 instances, both are supported by leave one out method. The threshold is obtained when the number of instances is 3279. Therefore, for the unfamiliar machine learning experimenters, the dataset threshold between leave one out and k-fold cross validation is 3279. Figure 10 represents dataset threshold with appropriate performance estimation method.

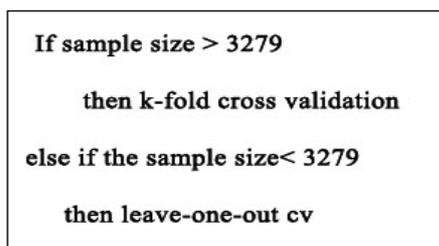


Figure 10: Dataset threshold results

6. Conclusions

In this paper we have presented results from the experiments performed in order to establish the dataset threshold for the performance estimation methods. From the experiments performed, the threshold has been identified when the dataset has 3179 instances where by the difference between the two methods is 0. The establishment of the dataset threshold will help unfamiliar supervised machine learning experimenters such as students studying in the field to categorise datasets based on the number of instances and attributes and then choose appropriate performance estimation method.

7. References

- [1] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*: Morgan Kauffman, 2005.
- [2] T. Mitchell, *Machine Learning*: MIT Press, 1997.
- [3] Alpaydin Ethem, *Introduction to Machine Learning*. Cambridge, Massachusetts, London, England: MIT Press. 2004.
- [4] T. Mitchell, "The Discipline of Machine Learning," Carnegie Mellon University, Pittsburgh, PA, USA, 2006.
- [5] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Communications of the ACM*, vol. 39, pp. 27-34, 1996.
- [6] L. Rokach and O. Maimon. Part C, "Top-down induction of decision trees classifiers - a survey.," *Applications and Reviews, IEEE Transactions on Systems, Man, and Cybernetics*, vol. 35, pp. 476-487., 2005.
- [7] T. Caelli and W. F. Bischof, *Machine Learning and Image Interpretation*. York, NY, USA: Plenum Press, 1997.
- [8] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*: Kauffman Press., 2002.
- [9] P. Cunningham and S. J. Delany, "K-Nearest Neighbour Classifiers," vol. 2008, 2007.
- [10] C. Campbell, *An Introduction to Kernel Methods*, 2000.
- [11] M. Berthold and D. J. Hand, "Intelligent Data Analysis," 2003.
- [12] L. Breiman, "Random Forests," 2001.
- [13] M. W. Craven, "Extracting Comprehensible Models from Trained Neural Networks." 1996.
- [14] Y. Bengio and Y. Grandvalet, "No Unbiased Estimator of the Variance of KFold Cross-Validation," *Journal of Machine Learning Research*, vol. 5, pp. 1089-1105, 2004.
- [15] S. Kostiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Informatica*, vol. 31, pp. 249-268, 2007.
- [16] J. Mena, "Data Mining Your Website," 1999.
- [17] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*: MIT Press, 1999.
- [18] T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Data Mining Researchers," 2004.
- [19] J. Winkler, M. Niranjan, and N. Lawrence, *Deterministic and Statistical Methods in Machine Learning*: Birkhauser, 2005.
- [20] T. G. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithm," pp. 1895-1923, 1998.
- [21] B. Everitt, *The Analysis of Contingency Tables*: Chapman & Hall/CRC, 1992.
- [22] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," pp. 1137-1143, 1995.
- [23] E. Micheli-Tzanakou, *Supervised and Unsupervised Pattern Recognition*: CRC Press, 1999.
- [24] O. Nelles, "Nonlinear System Identification," 2001.
- [25] I. H. Witten and E. Frank, *Data Mining*: Morgan Kauffman, 2000.
- [26] Y. e. a. Tang, "Granular Support Vector Machines for Medical Binary Classification Problems," pp. 73-78., 2004.