AN IMPLEMENTATION OF ALGORITHM SELECTION, MODEL EVALUATION AND MODEL SELECTION IN MACHINE LEARNING

Omprakash Dewangan
Assistant Professor, Faculty of Information Technology
Kalinga University, Naya Raipur, Chhattisgarh

Abstract: Proper model assessment, model selection, and algorithm selection procedures are critical for both academic and industrial machine learning researchers. These three subtasks are examined in this article, and the key benefits and drawbacks of each approach are discussed with reference to theoretical and empirical investigations. Additional suggestions are made to support the best practises in machine learning research and implementation. For model assessment and selection, common approaches like the holdout method are presented, which are not suggested for small datasets. Many bootstrap samples methods may be used to assess performance uncertainty instead of utilising confidence intervals based on normal approximation. In this article, the bias/variance trade-offs for picking k and practical ideas for finding the best k are explored, as are cross-validation approaches such as leave one out and k-fold cross-validation. Statistical tests and methods for coping with numerous comparisons, such as omnibus tests and multiple-comparison adjustments, are covered in great length in this section. The most efficient way to compare machine learning algorithms is to utilise the F-test 5x2 cross validation and the nested cross validation approaches.

Keyword- Algorithm , Model Evaluation, Machine Learning

INTRODUCTION

Term and Technique Glossary for Model Evaluation Machine learning has become a vital part of our life as consumers, customers, and, ideally, researchers and practitioners. There's one thing I believe we all share when it comes to using predictive modelling techniques, whether for academic and professional purposes: we all want to make ‘‘excellent’‘ forecasts. How can we tell if our model generalises successfully to new data sets that we haven't seen before? Is there any way to verify that it doesn't merely store the data and fail to make accurate predictions on future samples, which it has never seen before?? In the first place, how do we choose a good model? The task at hand may necessitate a different type of learning method. Following model evaluation, we continue our machine learning workflow. First and foremost, we need to prepare ahead and employ acceptable methods for the task at hand, so we can get started. Several of these approaches, as well as where they fit into a typical machine learning pipeline, will be discussed in length in this article.

1.1 Performance Estimation: Generalization Performance vs. Model Selection

‘‘How can we measure the performance of a machine learning model??’‘ What's the most common reaction? ‘‘First, we provide training data to our algorithm so that it can build a model of the world around us. We then forecast the test set's labels. Third, we use the test dataset to measure the accuracy of the model's predictions by counting the number of wrong predictions.’‘ However, assessing a model's performance is not that simple, depending on our goals. ‘‘Why do we care about performance estimations in the first place?’‘ Making predictions about future data is a major challenge in machine learning applications and algorithm development. A model's projected performance should be based on how well it performs with hypothetical data. While machine learning often involves a great deal of testing, one example is the fine-tuning of a learning algorithm's hyperparameters, the so-called internal. Running a learning algorithm with a variety of hyperparameter settings on a training dataset can produce distinct models. We must devise a system for comparing each of these models in order to find the one that performs the best. Aside from just tweaking an algorithm, we're often testing with more than one algorithm at a time to see which one is the most effective in a specific situation. Predictive and computational performance is the most common measure by which we evaluate different algorithms. Following is a brief summary of the primary reasons for modeling's predictive ability:

1. We are interested in assessing our model's generalisation performance, or its ability to accurately predict new data.
2. We aimed to improve prediction performance by changing the learning algorithm and selecting the best performing model from a given hypothesis space.
3. The purpose of this study is to examine a range of machine learning algorithms in order to identify the one that performs best and the best model from the algorithm's hypothesis space.

1.2 Assumptions and Terminology Model evaluation is certainly a complex topic.

Let's assume a few things and go through a few of the terms we'll use in this post in order to keep this essay focused. i.i.d. Expected to be independent and identically distributed, which implies that all training examples are taken from the same probability distribution and statistically independent from each other. The best example of this would be working with temporal data or time-series data in a circumstance where the training examples are not independent.

Supervised learning and classification

A subset of machine learning in which the goal values of a dataset are known is discussed in this article. Classification, the labelling of training and test examples with category goals, will be the subject of our discussion.

0-1 loss and prediction accuracy.

When it comes to the accuracy of forecasts, the number of right predictions divided by the total amount of data points is what we'll be focusing at next. For our prediction accuracy, we divide the number of correct answers by the number of examples n. The accuracy of the ACC's forecasts is referred to as the ACC's accuracy in more formal terms.

ACC = 1 − ERR,…….. (1)

Bias. Bias (statistical prejudice) is referred to throughout this article as ‘‘bias’‘ (in contrast to the bias in a machine learning system). The discrepancy between the estimator's expected value E[βˆ] and the true parameter's βˆ underlying value that is being estimated is called the estimator's mistake..:

Bias = E[βˆ] − β…….. (2)

Variance. An estimator's variance βˆ is simply its statistical variance, for example, by calculating the squared difference between its expected value E[ˆβ] and its estimator. :

Variance = E h……..(3)

Target function

Modeling a certain process is a common goal in predictive modelling. We're attempting to find or approximate a previously unidentified function.We wish to simulate the actual function f() by using the target function f(x) = y.

Hypothesis

You might think of hypotheses as functions that we believe (or hope) are close enough to our desired target function f(). In the context of spam categorization, this would be a rule that we devised to help us differentiate between spam and legitimate communications.

Model In the field of machine learning, the terms ‘‘hypothesis’‘ and ‘‘model’‘ are frequently used interchangeably.Other fields of study may have distinct interpretations for these terms: Hypothesis is the scientist's ‘‘educated guess’‘; model is the scientist's ‘‘educated guess’‘ put into practice.

Learning algorithm

Again, our goal is to locate or approximate the target function, and the learning algorithm is a sequence of instructions that try to model the target function based on a training dataset. Before constructing a final hypothesis for an unknown target function, a learning algorithm could study a hypothesis space.

Hyperparameters

An L2 penalty in logistic regression's loss function or a decision tree classifier's maximum depth are examples of hyperparameters, which are machine learning algorithms' tuning parameters. Learning algorithms can utilise a model parameter instead to specify how the model relates to data it receives. What are model parameters? Weight coefficients (slope) and bias term (y-axis intercept) of linear regression lines are model parameters.

1.3 Resubstitution Validation and the Holdout Method

A model's holdout strategy is undoubtedly the quickest and easiest way to evaluate it. It is initially essential to partition the data set into two parts: one for training and one for testing. A model is then built using the training data and predictions are made for each label. How many forecasts were correct may be determined by comparing the predicted labels with test sets' ground-truth labels and finding how many predictions were accurate. Resubstitution validation (also known as resubstitution assessment) is the practise of training and testing a model on the same training dataset, which can lead to overfitting and a negative bias. What we don't know is whether the model simply remembers its training data or if it can generalise to fresh, unlabeled input. ' However, we can also measure this so-called ‘‘optimism bias’‘ by comparing training and test results.

1.4 Stratification

We must please remember that the data in any dataset is a random sample drawn from the distribution of the total population. Without replacement, further subsampling alters the statistical properties of the sample (mean, percentage and variance). Subsampling without replacement has a higher impact on the sample statistic as the sample size increases. An example utilising the Iris dataset 2 may be seen here, with 2/3 of the training data and 1/3 of the testing dataset randomly split. (The source code for this image may be found on GitHub3.))

1.5 Holdout Validation Figure 1 presents a visual summary of the holdout validation approach before we discuss its benefits and drawbacks.

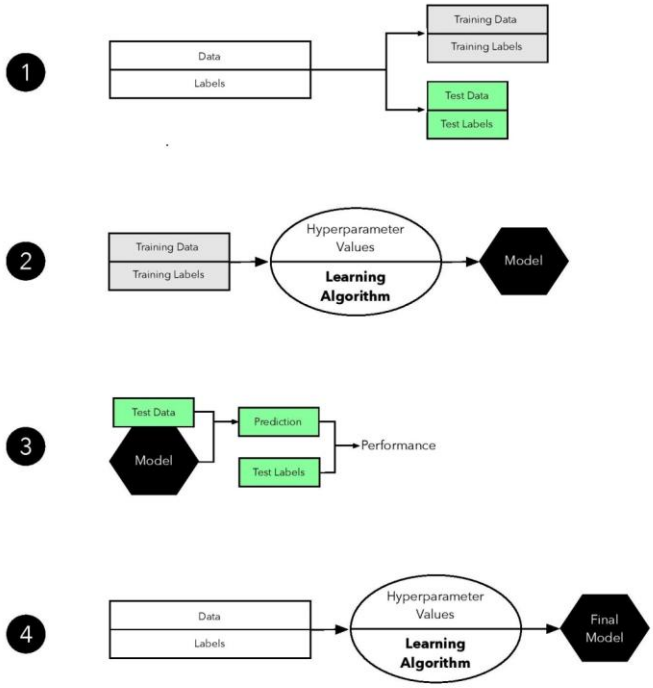

Figure 1: Visual summary of the holdout validation method

2 Bootstrapping and Uncertainties

Model assessment in supervised machine learning was first described in (Terms and Techniques for Effective Model Evaluation). For model validation, we discussed the holdout strategy, which helps us deal with real-world constraints such as limited access to new, labelled data sets The holdout strategy was used to divide our dataset into a training and a test set. The training data is fed into an algorithm known as a supervised learning algorithm. The training set's labelled observations are used by the learning algorithm to build a model. To evaluate the model's capacity to anticipate future outcomes, we gather new data that has never been seen before. An assessment of our performance's uncertainty based on a single test set can be represented using the normal approximation by making certain assumptions that allow us to generate confidence intervals. Advanced model evaluation methods are discussed in this section. After that, we'll talk about methods for estimating the model's variance and stability. The next post in this series will focus on cross-validation approaches for model selection, now that we've covered the fundamentals. There are three distinct but related tasks or purposes for which we need to evaluate systems:

1. A model's ability to predict future data (i.e., its generalisation accuracy) is what we're interested in monitoring
2. A better learning algorithm and the selection of the best-performing model within a given hypothesis space can improve predictive performance.
3. Finding the most appropriate machine learning algorithm is essential. Because of this, we wish to evaluate many algorithms and choose the best-performing model from the algorithm's hypotheses space.

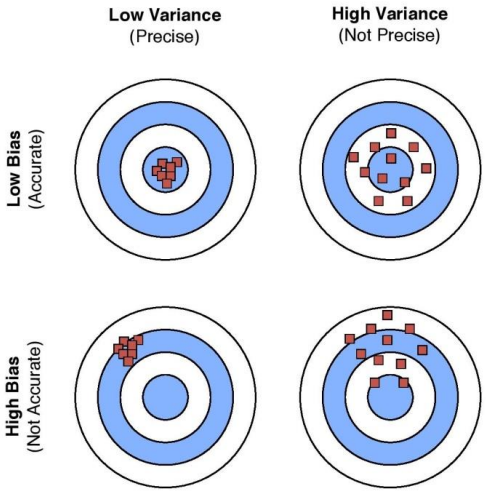

Figure 2: Illustration of bias and variance.

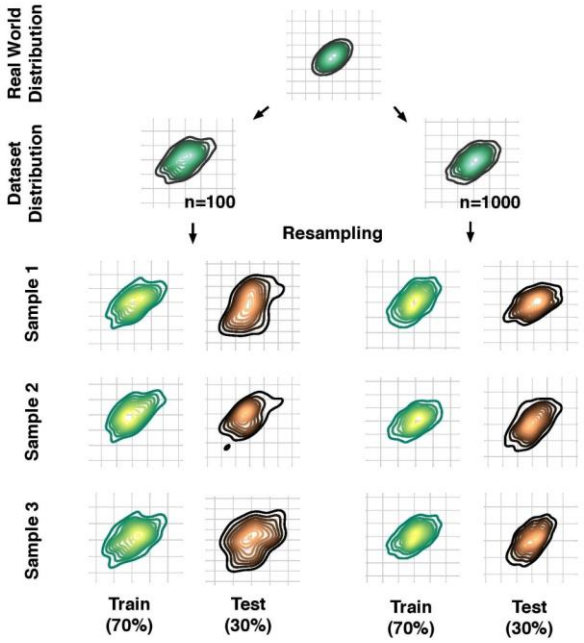

Figure 3: Repeated subsampling from a two-dimensional Gaussian distribution.

2.3 Repeated Holdout Validation

This method's average performance can be calculated by running it k times with different random seed values. This will give us a more precise estimate of performance.

2.4 The Bootstrap Method and Empirical Confidence Intervals

Monte Carlo Cross-Validation may have convinced us that repeated holdout validation may produce a more accurate assessment of a model's performance on random test sets than an evaluation based on a single train/test split through holdout validation (Section 15 1.5). The recurrence of holdouts may also give us information into the robustness of our model's design. This section examines a bootstrap approach for evaluating models and calculating uncertainty.

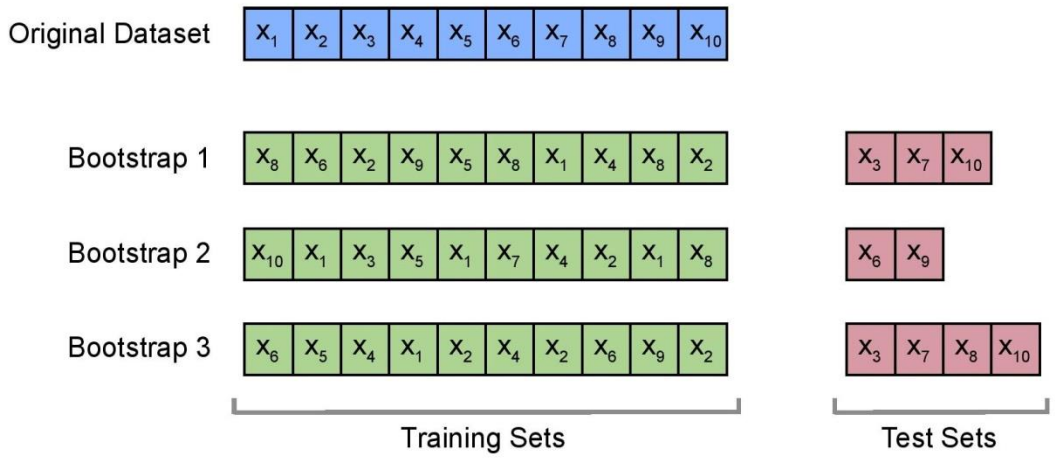

Figure 4: Illustration of training and test data splits in the Leave-One-Out Bootstrap (LOOB).

3 Cross-validation and Hyperparameter Optimization

A significant number of settings must be specified by machine learning researchers and practitioners for almost every algorithm. Machine learning algorithms may be tuned to get the best performance by changing the so-called hyperparameters, which are tuning knobs that help us discover the correct balance between bias and variance. As a result, while optimising performance using hyperparameters, no set of rules can be relied upon to provide optimal performance on a single dataset. The generalisation performance of a model may be estimated using holdout and bootstrap approaches, which have already been discussed. Bias and variance were investigated as a trade-off for measuring generalisation performance and strategies for calculating uncertainty in performance assessments. This section focuses on cross-validation methods for model evaluation and model selection. Cross-

validation methods are used to rank models from a variety of hyperparameter configurations and evaluate how well they translate to other datasets.

Hyperparameters and Model Selection

In the past, the holdout technique and other flavours of the bootstrap were utilised to evaluate our prediction models' generalisation ability. For the sake of training and testing, it was split into two separate datasets. We evaluated the model's performance on the training set using a separate test set that we didn't give the machine learning algorithm access to during model fitting.

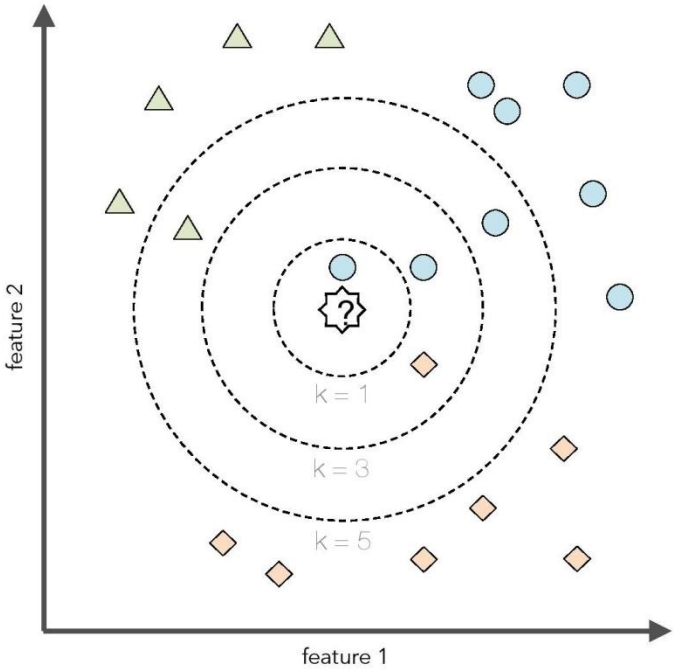

Figure 5 illustrates the k-nearest neighbours method with various k values.

Hyperparameter settings, such as the amount of k in the k-nearest neighbours technique that we were discussing, were fixed in our learning algorithms while we were addressing difficulties like bias-variance tradeoff. A priori, before we fit a model, hyperparameters are defined as the learning algorithm's parameters. In contrast, we referred to our model's parameters as model parameters.

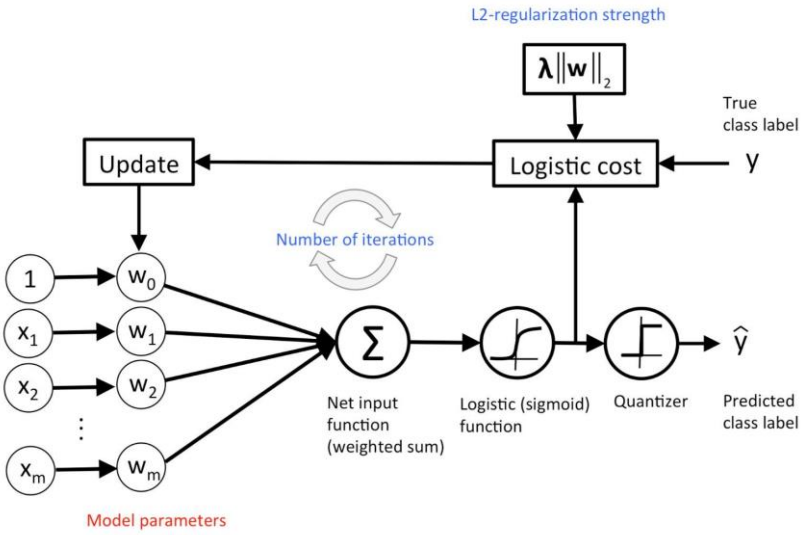

Figure 6: Conceptual overview of logistic regression.

Three-Way Holdout Method for Hyperparameter Tuning Using the resubstitution validation method to estimate the generalisation performance is a bad idea, as explained in Section 1. The holdout approach was used to divide the dataset into two parts, a training set and an independent test set, in order to assess how well our model generalises to new data. Can we use the holdout method for

hyperparameter tuning? Yes, that's correct. However, we must slightly modify our first strategy, the ‘‘two-way’‘ split, and divide the dataset into three parts: a training, a validation and a test set.

1. We want to understand how well a model is in making generalisations to new data, and how effective it is at predicting the future data.
2. By modifying the algorithm and selecting the best-performing model from a set of hypotheses, we hope to improve the accuracy of our predictions.
3. The goal of this research is to examine several machine learning algorithms and choose the best-performing one and the best-performing model from the hypothesis space of those algorithms.

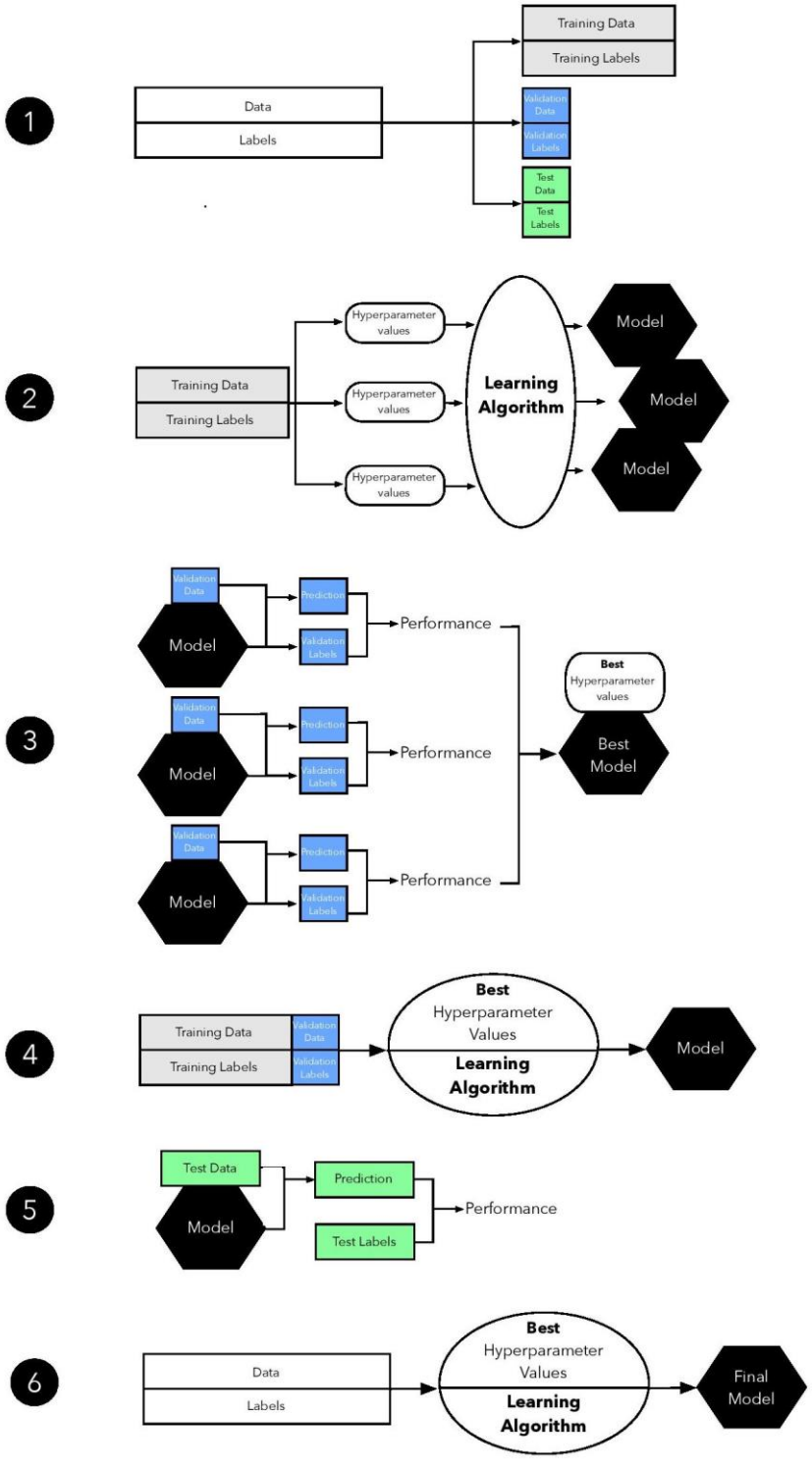

Figure 7: Hyperparameter tweaking using the three-way holdout approach is illustrated in the figure below.

Introduction to k-fold Cross-Validation k-fold cross-validation is the most commonly used method for evaluating and selecting machine learning models. In the literature, practitioners and academics may refer to the holdout approach as a cross-validation methodology.Cross validation, on the other hand, may be better seen as a series of repeated cycles of training and testing that cross over into each other. Cross-validation is based on the premise that every sample in our dataset gets tested. In k-fold cross-validation, we repeat the same process k times over a dataset. According to Figure 13, a fivefold cross-validation dataset is divided into k parts, one of which is used for validation and the rest is put together in order to generate a training subset for model evaluation.

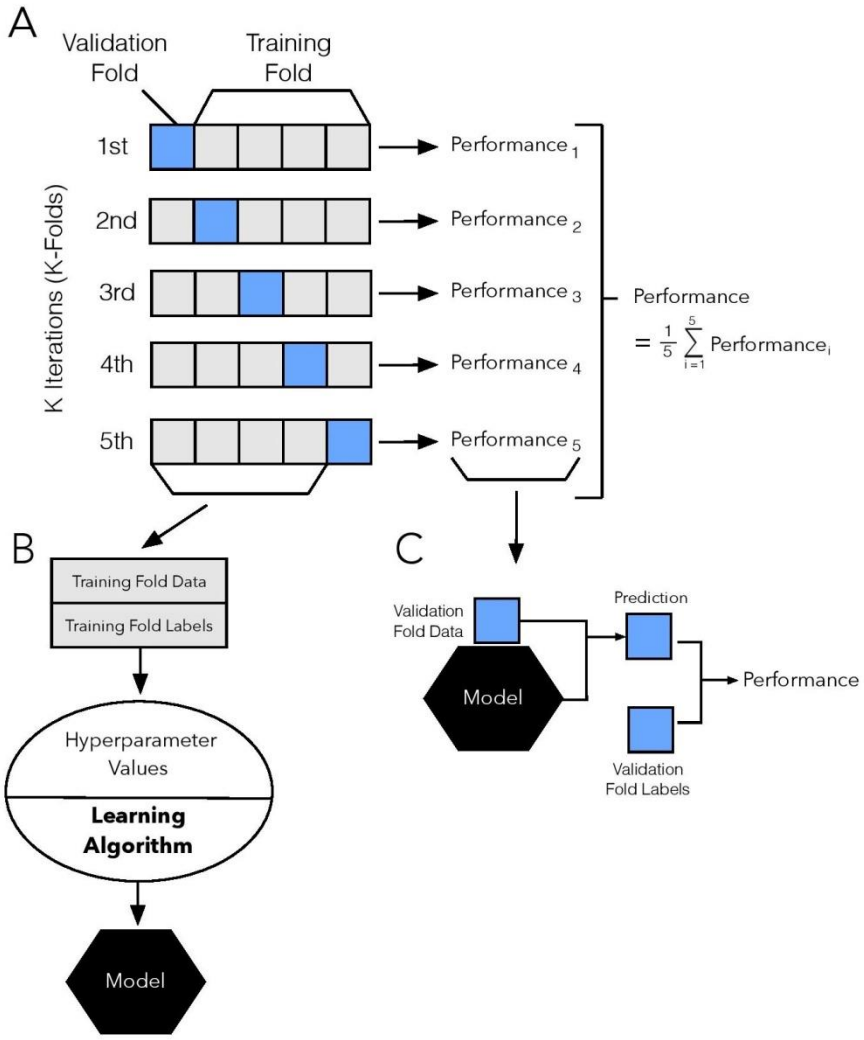

Figure 8: Illustration of the k-fold cross-validation procedure.

Special Cases: 2-Fold and Leave-One-Out Cross-Validation

k = 5 was used to show k-fold cross-validation in the prior section, and you may be wondering why this was done. For one thing, the k-fold cross-validation may be shown in a more compact form. Since k = 5 is less computationally intensive than k = 10, it is also usual to employ it. Keeping in mind that the model is more sensitive to how the data were divided, the performance estimate may have a pessimistic bias if k is too small and the estimate's variance may grow as a result. K-fold cross validation contains two important special situations: k = 2 and n. Both of these examples are noteworthy exceptions to the rule. Two-fold cross-validation is commonly used interchangeably with the holdout technique. This statement would only be accurate if the holdout method was used with a two-round rotation of the training and validation sets (for instance, using exactly 50 percent data for training and 50 percent of the examples for validation in each round, swapping these sets, repeating the training and evaluation procedure, and eventually computing the performance estimate as the arithmetic mean of the two performance estimates on the validation sets). As a result of the way the holdout approach is most commonly applied, it is discussed in this article as two separate procedures in Figure 14.

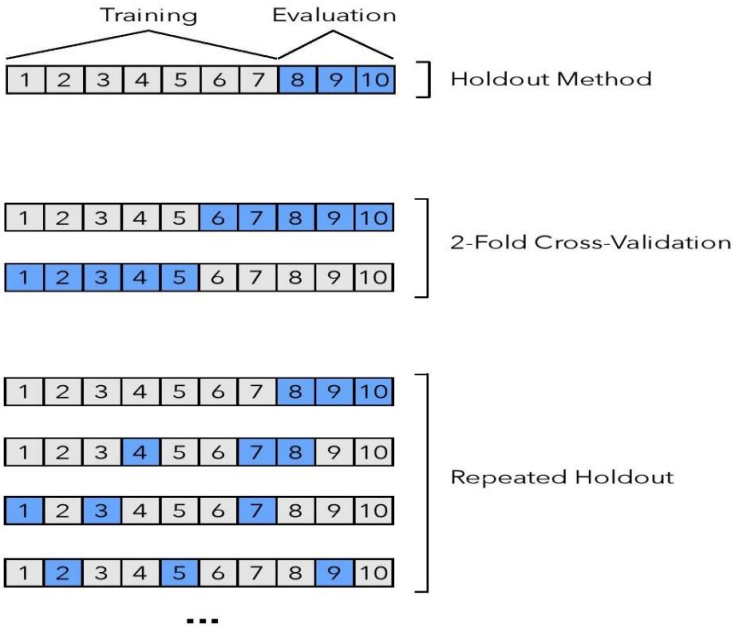

Figure 9: The comparison of the holdout method, the 2-fold cross-validation method, and the repeated holdout method.

Figure 10: Illustration of leave-one-out cross-validation

3.6 k-fold Cross-Validation and the Bias-Variance Trade-off

In Section 3.5, Hawkins and colleagues [Hawkins et al., 2003] show that for small and moderately big datasets, we may favour LOOCV over single train/test splits using the holdout approach. There is less of an optimistic bias in LOOCV (k = n) than in k < n-fold cross-validation because practically all (for example, n 1) training data are accessible for model fitting. Because of this, we

can consider the LOOCV estimate to be about as fair as one might hope. The following points summarise the discussion of the bias-variance trade-off by outlining the general trends while increasing the number of folds or k in the next section before moving on to model selection.:

- The performance estimator's bias is reducing (more accurate)
- Increased variability in performance estimates (more variability)
- The cost of computation rises (more iterations, larger training sets during fitting)
- The variance in small datasets can also be increased by reducing the k-fold cross-validation value to a modest number (for example, 2 or 3).

3.7 Model Selection via k-fold Cross-Validation

In the preceding sections, k-fold cross-validation for model assessment was introduced. This section explains the k-fold cross-validation method for model selection in greater depth than the previous one. The key to preventing test data leaking during training is to have a separate test dataset. This dataset isn't used during training or model selection. Figure 16 shows the entire procedure.

3.8 A Note About Model Selection and Large Datasets

The 3-way holdout strategy is the most often used method for model assessment in both the prior (non-deep learning) literature and the current (deep learning). In terms of computing costs, three-way holdout may be preferred over k-fold cross-validation. We only utilise deep learning methods in situations where we can overlook the high variance – the sensitivity of our estimates to the way in which we split the dataset for training, validation, and testing – and still obtain decent long-term outcomes despite the great computational efficiency. A big enough dataset means that k-fold cross-validation model selection and training, validation, or testing splits may be used in the holdout strategy to select models.

3.9 A Note About Feature Selection During Model Selection

Instead of running these operations on the entire dataset before splitting the data into folds, we frequently do so inside the k-fold cross-validation loop. Through the use of cross-validation, it is possible to eliminate any potential for bias.

4. Algorithm Comparison

For machine learning model and algorithm comparisons, there are a variety of statistical hypothesis testing methodologies available. This includes statistical tests based on goal forecasts for several independent test sets (the difficulties of utilising a single test set were discussed above), as well as cross-validation procedures for algorithm comparisons. Lastly, nested cross-validation will be presented in this last part. This approach is commonly used to compare methods on datasets of small to moderate size and is highly recommended. It's time to close this series of papers with a list of my personal recommendations for model evaluation, selection, and method selection.

Testing the Difference of Proportions

Statistical hypothesis testing frameworks such as the difference of two proportions (here proportions reflect the expected generalisation accuracy from a test set) with a 95% confidence level are often used to compare the performance of classification models for comparison purposes. One of the most straightforward ways to compare models is by doing the z-score test on two population proportions. However, this is by no means the best method. Briefly stated, we may rule out equality in classification results between two classifiers at a level of certainty = 0.05 if their respective 95% confidence interval accuracy accuracies do not overlap (or 5 percent probability). On the other hand, as Thomas Dietterich discovered in a simulated study [Dietterich, 1998], this test has a significant false positive rate (i.e., incorrectly detecting a difference when none exists), which is one of the reasons it is not recommended in reality. This procedure (which applies to many of the hypothesis tests covered below) is provided for completeness' sake and since it is a commonly used method: If you are conducting a two-tailed test, the hypothesis to be tested is that the proportions are the same, and the alternative hypothesis is that they are different. Before we can reject the null hypothesis, we must define a significant threshold (for example, the likelihood of finding an even more dramatic difference than the one witnessed is over 5%). the data is analysed in order to generate the test statistic (here: the z-score) and compare its p-value (probability) to the previously specified significance threshold; Based on the p-value and significance level, you accept or reject the null hypothesis, and then interpret the data appropriately.

Comparing Two Models with the McNemar Test

According to Dietterich [Dietterich, 1998], it is better to use the McNemar test rather than the ‘‘difference of proportions’‘ test. Quinn McNemar created the McNemar test in 1947 [McNemar, 1947], a non-parametric statistical test for paired comparisons that can be used to evaluate the performance of two machine learning classifiers.

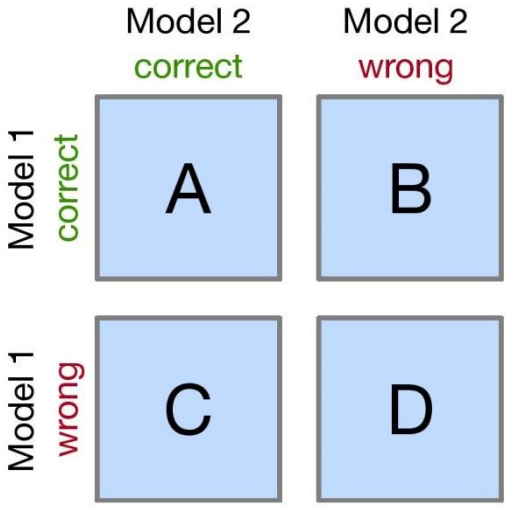

Figure 11: Confusion matrix layout in context of McNemar's test.

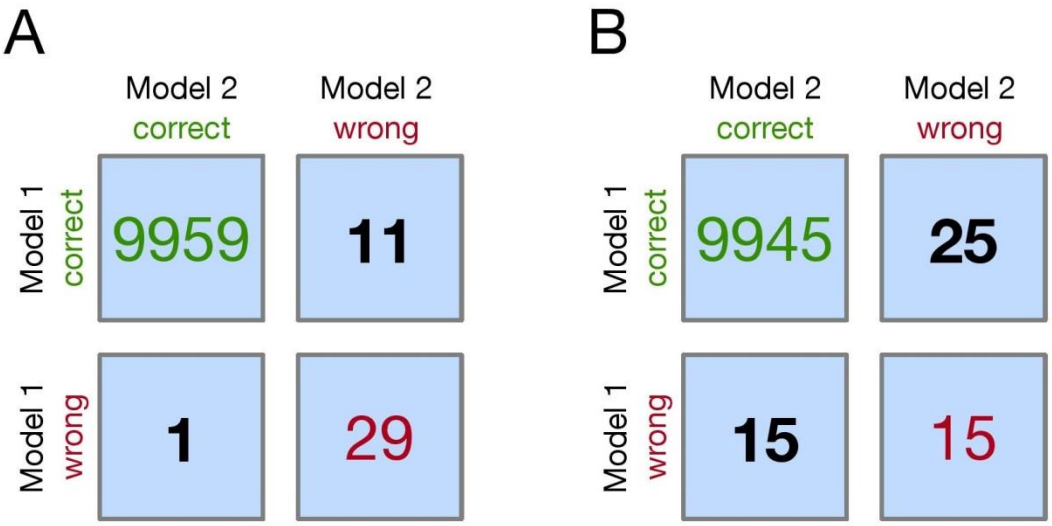

Figure 12: Model 1 and Model 2 classification outcomes confusion structures.

Exact p-Values via the Binomial Test

Even if the p-values in cells B and C are better represented by McNemar's test if they are more than 50 (according to the 2x2 confusion matrix shown previously), a more computationally costly binomial test is used if B and C are relatively small – since the McNemar test chi-squared value may not be well-approximate by the chi-squared.

Multiple Hypotheses Testing

Using McNemar's test, we were able to compare two machine learning classifiers earlier. However, in reality, we often have more than two models to evaluate based on their projected generalisation performance, for example, the predictions on an independent test set. Multiple hypothesis testing is a common problem that arises when the same technique is repeated over and over again, as is the case here. The following is a popular strategy for dealing with problems like these::

1. Under the assumption that there is no change in the classification accuracy, do an omnibus test to determine the statistical quality of the results.
2. A pairwise post hoc analysis may be used to find out where the variations in model performance arose after the omnibus test denied the null hypothesis. For example, we may utilise McNemar's test.)

4.6 Cochran's Q Test for Comparing the Performance of Multiple Classifiers

Cochran's Q test may be seen as an expanded version of McNemar's test, which can be used to compare up to three classifiers. The Cochran's Q test and an ANOVA for nominally matched data are quite comparable. Although it doesn't tell us which groups (or

models) are different, it does tell us that the models are distinct. Due to the fact that McNemar's test is based on two models, it has a chi-squared distribution with one degree of freedom because there are two models being tested. This test is used to find out if the categorization accuracy differs from one group to another if the null hypothesis (H0) holds [Fleiss et al., 2013]:
H0 : ACC1 = ACC2 = : : : = ACCM:
The F-test for Comparing Multiple Classifiers
Instead of a tangle of 1s and 0s, as Cochran predicted in his Q test study [Cochran, 1950], the F-test would practically automatically be carried out when data were measured variables that seemed to be regularly distributed. The F-test may be used as an approximation even if the table is made up of only one and zeros, as I have advised to researchers on a handful of occasions.
When comparing two classifiers, we may use the F-test [Looney, 1988] approach to compare the two classifiers, although it should be noted that Looney suggests an improved version of the F-test called the F+ test. The null hypothesis in the context of the F-test is once again that there is no difference in classification accuracy :
pi : H0 = p1 = p2 = = pL:

4.8 Comparing Algorithms
As previously indicated, statistical tests do not account for training sets with small sizes and algorithms that are sensitive to perturbations in the training sets; this might be a problem.
Resampled Paired t-Test
There are several advantages to comparing the performance of two models (classifiers or regressions), however Dietterich [Dietterich, 1998] warns that this technique has significant flaws and should not be utilised in practise.

:**k-fold Cross-validated Paired t-Test**
It is prevalent in elderly literature to use the cross-validated paired T-test, like the re-sampled paired t-test, for statistical testing. Although some of its issues have been resolved, this technique is not recommended for usage in practise since the training sets overlap. [Dietterich, 1998] resampled paired t-test

4.11 Dietterich's 5x2-Fold Cross-Validated Paired t-Test
Dietterich (Dietterich, 1998) created the 5x2cv paired t-test to address the shortcomings of earlier approaches, such as the resampled paired t-test and the k-fold cross-validated paired t-test, which were discussed in the first two sections of this article. Paired t-tests are identical to the preceding t-test versions, except the splitting (50 percent training and 50 percent test data) is repeated five times. Each of the five iterations, we fit two classifiers C1 and C2 to the training split and test their performance on the test split for each one. Rotating training and test sets (the training set becomes the test and vice versa) creates two performance difference measures :
ACCA = ACCA;C1 □ ACCA;C2

4.12 Alpaydin's Combined 5x2cv F-test
Alpaydin [Alpadin, 1999] suggested the 5x2cv combined F-test as a more robust alternative to Dietterich's 5x2cv paired t-test approach in the previous section.
Consider classifiers 1 and 2, and use the notation in the previous section to demonstrate how this method works. F is computed as follows as a result:

4.13 Effect size
Moreover, we may wish to evaluate effect sizes, which is something that is sadly rarely done in practise due to the fact that big samples raise p-values and can cause things to appear statistically significant. Theoretical relevance does not always entail ''practical significance,'' as the phrase goes. When it comes to determining an impact's magnitude, it is impossible to provide a thorough answer because the problem, task, or subject at hand determines the size.

4.14 Nested Cross-Validation
Because a large (or, ideally, infinitely large) test set is rarely accessible in real applications, an unbiased assessment of a model's true generalisation error could be obtained. When dealing with datasets of limited size, we are always looking for "better'' ways to deal with them.

5 Conclusions
The following conclusion are obtained
Performance estimation based on large dataset

- 2-way (train/test) holdout method
- Normal approximation of the confidence interval

Performance estimation based on small dataset

- No independent test set (repeated k-fold cross-validation)
- Cross-validation without the need of a separate testing dataset
- 0.632+ is inside the bootstrap sample confidence interval.

Based on large datasets, model selection (hyperparameter tuning) and performance appraisal.

- 3-way holdout method (train/validation/test split)

A small dataset is used to estimate model performance (hyperparameter tuning).

- k-fold cross-validation with independent test set (repeated)
- Cross-validation using an independent test set that use the leave-one-out technique

An algorithm and model comparison using a big set of data.

- Various training and testing events that are independent of each other (algorithm comparison, AC)
- McNemar assessment (model comparison, MC)
- McNemar test + Cochran's Q (MC)

Model & algorithm comparison based on small dataset

- Combined 5x2cv F test (AC)
- Nested cross-validation (AC

References

1. Alpaydin, E. (1999). Combined 5x2cv F test for comparing supervised classification learning algorithms. Neural Computation, 11(8):1885–1892.
2. Bengio, Y. and Grandvalet, Y. (2004). No unbiased estimator of the variance of k-fold cross-validation. Journal of Machine Learning Research, 5(Sep):1089–1105.
3. Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze, 8:3–62.
4. Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). Classification and regression trees. CRC press.
5. Cochran, W. G. (1950). The comparison of percentages in matched samples. Biometrika, 37(3/4):256–266.
6. Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. Neural computation, 10(7):1895–1923. [Dunn, 1961] Dunn, O. J. (1961). Multiple comparisons among means. Journal of the American statistical association, 56(293):52–64.
7. Edwards, A. L. (1948). Note on the “correction for continuity” in testing the significance of the difference between correlated proportions. Psychometrika, 13(3):185–187.
8. Efron, B. (1981). Nonparametric standard errors and confidence intervals. Canadian Journal of Statistics, 9(2):139–158.
9. Efron, B. (1983). Estimating the error rate of a prediction rule: improvement on cross-validation. Journal of the American Statistical Association, 78(382):316–331.
10. Efron, B. (1992). Bootstrap methods: another look at the Jackknife. In Breakthroughs in Statistics, pages 569–593. Springer.
11. Efron, B. and Tibshirani, R. (1997). Improvements on cross-validation: the .632+ bootstrap method. Journal of the American Statistical Association, 92(438):548–560.
12. Efron, B. and Tibshirani, R. J. (1994). An Introduction to the Bootstrap. CRC press.
13. Fleiss, J. L., Levin, B., and Paik, M. C. (2013). Statistical Methods for Rates and Proportions. John Wiley & Sons.
14. Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). In The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York.
15. Hawkins, D. M., Basak, S. C., and Mills, D. (2003). Assessing model fit by cross-validation. Journal of Chemical Information and Computer Sciences, 43(2):579–586.
16. Iizuka, N., Oka, M., Yamada-Okabe, H., Nishida, M., Maeda, Y., Mori, N., Takao, T., Tamesa, T., Tangoku, A., Tabuchi, H., et al. (2003). Oligonucleotide microarray for prediction of early intrahepatic recurrence of hepatocellular carcinoma after curative resection. The lancet, 361(9361):923–929.
17. James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). In An Introduction to Statistical Learning: With Applications in R. Springer, New York.
18. Kim, J.-H. (2009). Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. Computational Statistics & Data Analysis, 53(11):3735–3745.
19. Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. International Joint Conference on Artificial Intelligence, 14(12):1137–1143.
20. Kuncheva, L. I. (2004). Combining Pattern Classifiers: Methods and Algorithms. John Wiley & Sons.
21. Looney, S. W. (1988). A statistical technique for comparing the accuracies of several classifiers. Pattern Recognition Letters, 8(1):5–9.
22. McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika, 12(2):153–157.
23. Molinaro, A. M., Simon, R., and Pfeiffer, R. M. (2005). Prediction error estimation: a comparison of resampling methods. Bioinformatics, 21(15):3301–3307.
24. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. Journal of Machine Learning Research, 12(Oct):2825–2830.
25. Perneger, T. V. (1998). What’s wrong with bonferroni adjustments. Bmj, 316(7139):1236–1238.
26. Raschka, S. (2018). Mlxtend: Providing machine learning and data science utilities and extensions to python’s scientific computing stack. The Journal of Open Source Software, 3(24).

27. Refaeilzadeh, P., Tang, L., and Liu, H. (2007). On comparison of feature selection algorithms. In Proceedings of AAAI Workshop on Evaluation Methods for Machine Learning II, pages 34–39.
28. Rothman, K. J. (1990). No adjustments are needed for multiple comparisons. Epidemiology, pages 43–46.
29. Tan, P.-N., Steinbach, M., and Kumar, V. (2005). In Introduction to Data Mining. Pearson Addison Wesley, Boston.
30. Varma, S. and Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. BMC bioinformatics, 7(1):91.
31. Varoquaux, G. (2017). Cross-validation failure: small sample sizes lead to large error bars. Neuroimage.
32. Westfall, P. H., Troendle, J. F., and Pennello, G. (2010). Multiple McNemar tests. Biometrics, 66(4):1185–1191