

A MACHINE LEARNING HANDBOOK

PUBLISHER OF THIS BOOK



Copyright © 2018 ICS5110 APPLIED MACHINE LEARNING class of 2018/9, University of Malta.

JEAN-PAUL EBEJER, DYLAN SEYCHELL, LARA MARIE DEMAJO, DANIEL FARRUGIA, **ADD YOUR NAME TO THIS LIST**

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

First printing, December 2018

Contents

<i>Introduction</i>	5
<i>On the Use of the tufte-book Document Class</i>	7
<i>Cross-Validation</i>	15
<i>Confusion Matrix</i>	19
<i>Bibliography</i>	23

Introduction

This book explains popular Machine Learning terms. We focus to explain each term comprehensively, through the use of examples and diagrams. The description of each term is written by a student sitting in for ICS5110 APPLIED MACHINE LEARNING¹ at the University of Malta (class 2018/2019). This study-unit is part of the MSc. in AI offered by the Department of Artificial Intelligence, Faculty of ICT.

¹ <https://www.um.edu.mt/courses/studyunit/ICS5110>

On the Use of the tufte-book Document Class

The Tufte- \LaTeX document classes define a style similar to the style Edward Tufte uses in his books and handouts. Tufte's style is known for its extensive use of sidenotes, tight integration of graphics with text, and well-set typography. This document aims to be at once a demonstration of the features of the Tufte- \LaTeX document classes and a style guide to their use.

Page Layout

Headings

This style provides A- and B-heads (that is, `\section` and `\subsection`), demonstrated above.

If you need more than two levels of section headings, you'll have to define them yourself at the moment; there are no pre-defined styles for anything below a `\subsection`. As Bringhurst points out in *The Elements of Typographic Style*,² you should "use as many levels of headings as you need: no more, and no fewer."

The Tufte- \LaTeX classes will emit an error if you try to use `\subsubsection` and smaller headings.

² Robert Bringhurst. *The Elements of Typography*. Hartley & Marks, 3.1 edition, 2005. ISBN 0-88179-205-5

IN HIS LATER BOOKS,³ Tufte starts each section with a bit of vertical space, a non-indented paragraph, and sets the first few words of the sentence in SMALL CAPS. To accomplish this using this style, use the `\newthought` command:

³ Edward R. Tufte. *Beautiful Evidence*. Graphics Press, LLC, first edition, May 2006. ISBN 0-9613921-7-7

```
\newthought{In his later books}, Tufte starts...
```

Feature	VDQI	EI	VE	BE
Author				
Typeface	serif	serif	serif	sans serif
Style	italics	italics	italics	upright, caps
Size	24 pt	20 pt	20 pt	20 pt
Title				
Typeface	serif	serif	serif	sans serif
Style	upright	italics	upright	upright, caps
Size	36 pt	48 pt	48 pt	36 pt
Subtitle				
Typeface	–	–	serif	–
Style	–	–	upright	–
Size	–	–	20 pt	–
Edition				
Typeface	sans serif	–	–	–
Style	upright, caps	–	–	–
Size	14 pt	–	–	–
Publisher				
Typeface	serif	serif	serif	sans serif
Style	italics	italics	italics	upright, caps
Size	14 pt	14 pt	14 pt	14 pt

Sidenotes

One of the most prominent and distinctive features of this style is the extensive use of sidenotes. There is a wide margin to provide ample room for sidenotes and small figures. Any `\footnotes` will automatically be converted to sidenotes.⁴ If you'd like to place ancillary information in the margin without the sidenote mark (the superscript number), you can use the `\marginnote` command.

The specification of the `\sidenote` command is:

```
\sidenote[⟨number⟩][⟨offset⟩]{Sidenote text.}
```

Both the `⟨number⟩` and `⟨offset⟩` arguments are optional. If you provide a `⟨number⟩` argument, then that number will be used as the sidenote number. It will change of the number of the current sidenote only and will not affect the numbering sequence of subsequent sidenotes.

Sometimes a sidenote may run over the top of other text or graphics in the margin space. If this happens, you can adjust the vertical position of the sidenote by providing a dimension in the `⟨offset⟩` argument. Some examples of valid dimensions are:

```
1.0in    2.54cm    254mm    6\baselineskip
```

If the dimension is positive it will push the sidenote down the page; if the dimension is negative, it will move the sidenote up the page.

While both the `⟨number⟩` and `⟨offset⟩` arguments are optional, they must be provided in order. To adjust the vertical position of the sidenote while leaving the sidenote number alone, use the following syntax:

```
\sidenote[][⟨offset⟩]{Sidenote text.}
```

The empty brackets tell the `\sidenote` command to use the default sidenote number.

⁴ This is a sidenote that was entered using the `\footnote` command.

This is a margin note. Notice that there isn't a number preceding the note, and there is no number in the main text where this note was written.

If you *only* want to change the sidenote number, however, you may completely omit the `<offset>` argument:

```
\sidenote[<number>]{Sidenote text.}
```

The `\marginnote` command has a similar *offset* argument:

```
\marginnote[<offset>]{Margin note text.}
```

References

References are placed alongside their citations as sidenotes, as well. This can be accomplished using the normal `\cite` command.⁵

The complete list of references may also be printed automatically by using the `\bibliography` command. (See the end of this document for an example.) If you do not want to print a bibliography at the end of your document, use the `\nobibliography` command in its place.

To enter multiple citations at one location,⁶ you can provide a list of keys separated by commas and the same optional vertical offset argument: `\cite[<offset>]{bibkey1,bibkey2,...}`.

```
\cite[<offset>]{bibkey1,bibkey2,...}
```

Figures and Tables

Images and graphics play an integral role in Tufte's work. In addition to the standard `figure` and `tabular` environments, this style provides special figure and table environments for full-width floats.

Full page-width figures and tables may be placed in `figure*` or `table*` environments. To place figures or tables in the margin, use the `marginfigure` or `marginfigure` environments as follows (see figure 1):

```
\begin{marginfigure}
\includegraphics{helix}
\caption{This is a margin figure.}
\label{fig:marginfig}
\end{marginfigure}
```

The `marginfigure` and `marginfigure` environments accept an optional parameter `<offset>` that adjusts the vertical position of the figure or table. See the “Sidenotes” section above for examples. The specifications are:

```
\begin{marginfigure}[<offset>]
...
\end{marginfigure}

\begin{marginfigure}[<offset>]
...
\end{marginfigure}
```

Figure 2 is an example of the `figure*` environment and figure 3 is an example of the normal `figure` environment.

⁵ The first paragraph of this document includes a citation.

⁶ Edward R. Tufte. *Beautiful Evidence*. Graphics Press, LLC, first edition, May 2006. ISBN 0-9613921-7-7; and Edward R. Tufte. *Envisioning Information*. Graphics Press, Cheshire, Connecticut, 1990. ISBN 0-9613921-1-8

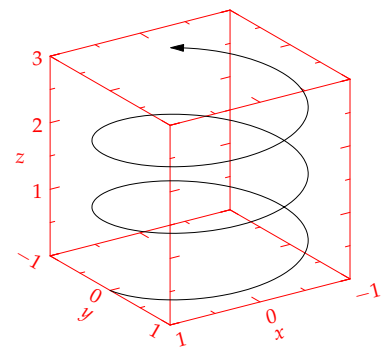


Figure 1: This is a margin figure. The helix is defined by $x = \cos(2\pi z)$, $y = \sin(2\pi z)$, and $z = [0, 2.7]$. The figure was drawn using Asymptote (<http://asymptote.sf.net/>).

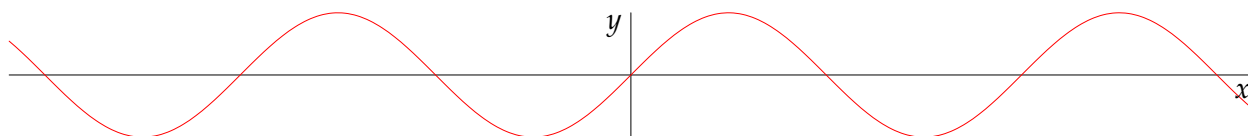


Figure 2: This graph shows $y = \sin x$ from about $x = [-10, 10]$. Notice that this figure takes up the full page width.

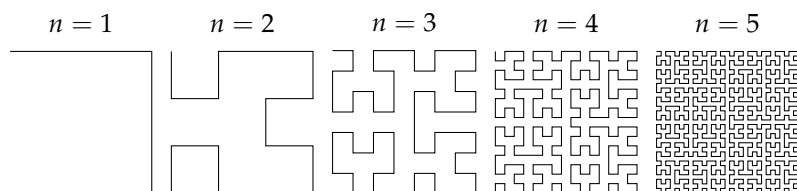


Figure 3: Hilbert curves of various degrees n . Notice that this figure only takes up the main textblock width.

As with sidenotes and marginnotes, a caption may sometimes require vertical adjustment. The `\caption` command now takes a second optional argument that enables you to do this by providing a dimension $\langle offset \rangle$. You may specify the caption in any one of the following forms:

```
\caption{long caption}
\caption[short caption]{long caption}
\caption[][ $\langle offset \rangle$ ]{long caption}
\caption[short caption][ $\langle offset \rangle$ ]{long caption}
```

A positive $\langle offset \rangle$ will push the caption down the page. The short caption, if provided, is what appears in the list of figures/tables, otherwise the “long” caption appears there. Note that although the arguments $\langle short\ caption \rangle$ and $\langle offset \rangle$ are both optional, they must be provided in order. Thus, to specify an $\langle offset \rangle$ without specifying a $\langle short\ caption \rangle$, you must include the first set of empty brackets `[]`, which tell `\caption` to use the default “long” caption. As an example, the caption to figure 3 above was given in the form

```
\caption[Hilbert curves...][6pt]{Hilbert curves...}
```

Table 1 shows table created with the `booktabs` package. Notice the lack of vertical rules—they serve only to clutter the table’s data.

Margin	Length
Paper width	8 ¹ / ₂ inches
Paper height	11 inches
Textblock width	6 ¹ / ₂ inches
Textblock/sidenote gutter	³ / ₈ inches
Sidenote width	2 inches

Table 1: Here are the dimensions of the various margins used in the Tufte-handout class.

OCCASIONALLY \LaTeX will generate an error message:

```
Error: Too many unprocessed floats
```

\LaTeX tries to place floats in the best position on the page. Until it’s finished composing the page, however, it won’t know where those positions are. If you have a lot of floats on a page (including sidenotes,

margin notes, figures, tables, etc.), L^AT_EX may run out of “slots” to keep track of them and will generate the above error.

L^AT_EX initially allocates 18 slots for storing floats. To work around this limitation, the Tufte-L^AT_EX document classes provide a `\morefloats` command that will reserve more slots.

The first time `\morefloats` is called, it allocates an additional 34 slots. The second time `\morefloats` is called, it allocates another 26 slots.

The `\morefloats` command may only be used two times. Calling it a third time will generate an error message. (This is because we can’t safely allocate many more floats or L^AT_EX will run out of memory.)

If, after using the `\morefloats` command twice, you continue to get the Too many unprocessed floats error, there are a couple things you can do.

The `\FloatBarrier` command will immediately process all the floats before typesetting more material. Since `\FloatBarrier` will start a new paragraph, you should place this command at the beginning or end of a paragraph.

The `\clearpage` command will also process the floats before continuing, but instead of starting a new paragraph, it will start a new page.

You can also try moving your floats around a bit: move a figure or table to the next page or reduce the number of sidenotes. (Each sidenote actually uses *two* slots.)

After the floats have placed, L^AT_EX will mark those slots as unused so they are available for the next page to be composed.

Captions

You may notice that the captions are sometimes misaligned. Due to the way L^AT_EX’s float mechanism works, we can’t know for sure where it decided to put a float. Therefore, the Tufte-L^AT_EX document classes provide commands to override the caption position.

Vertical alignment To override the vertical alignment, use the `\setfloatalignment` command inside the float environment. For example:

```
\begin{figure}[btp]
\includegraphics{sinewave}
\caption{This is an example of a sine wave.}
\label{fig:sinewave}
\setfloatalignment{b}% forces caption to be bottom-aligned
\end{figure}
```

The syntax of the `\setfloatalignment` command is:

```
\setfloatalignment{⟨pos⟩}
```

where `⟨pos⟩` can be either `b` for bottom-aligned captions, or `t` for top-aligned captions.

Horizontal alignment To override the horizontal alignment, use either the `\forceversofloat` or the `\forcerectofloat` command inside of the float environment. For example:

```
\begin{figure}[btp]
\includegraphics{sinewave}
\caption{This is an example of a sine wave.}
\label{fig:sinewave}
\forceversofloat% forces caption to be set to the left of the float
\end{figure}
```

The `\forceversofloat` command causes the algorithm to assume the float has been placed on a verso page—that is, a page on the left side of a two-page spread. Conversely, the `\forcerectofloat` command causes the algorithm to assume the float has been placed on a recto page—that is, a page on the right side of a two-page spread.

Full-width text blocks

In addition to the new float types, there is a `fullwidth` environment that stretches across the main text block and the sidenotes area.

```
\begin{fullwidth}
Lorem ipsum dolor sit amet...
\end{fullwidth}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Typography

Typefaces

If the Palatino, Helvetica, and Bera Mono typefaces are installed, this style will use them automatically. Otherwise, we'll fall back on the Computer Modern typefaces.

Letterspacing

This document class includes two new commands and some improvements on existing commands for letterspacing.

When setting strings of ALL CAPS or SMALL CAPS, the letterspacing—that is, the spacing between the letters—should be increased slightly.⁷ The `\allcaps` command has proper letterspacing for strings of FULL CAPITAL LETTERS, and the `\smallcaps` command has letterspacing for SMALL CAPITAL LETTERS. These commands will also automatically convert the case of the text to upper- or lowercase, respectively.

⁷ Robert Bringhurst. *The Elements of Typography*. Hartley & Marks, 3.1 edition, 2005. ISBN 0-88179-205-5

The `\textsc` command has also been redefined to include letterspacing. The case of the `\textsc` argument is left as is, however. This allows one to use both uppercase and lowercase letters: THE INITIAL LETTERS OF THE WORDS IN THIS SENTENCE ARE CAPITALIZED.

Document Class Options

The `tufte-book` class is based on the \LaTeX book document class. Therefore, you can pass any of the typical book options. There are a few options that are specific to the `tufte-book` document class, however.

The `a4paper` option will set the paper size to A4 instead of the default US letter size.

The `sfsidenotes` option will set the sidenotes and title block in a sans serif typeface instead of the default roman.

The `twoside` option will modify the running heads so that the page number is printed on the outside edge (as opposed to always printing the page number on the right-side edge in `oneside` mode).

The `symmetric` option typesets the sidenotes on the outside edge of the page. This is how books are traditionally printed, but is contrary to Tufte's book design which sets the sidenotes on the right side of the page. This option implicitly sets the `twoside` option.

The `justified` option sets all the text fully justified (flush left and right). The default is to set the text ragged right. The body text of Tufte's books are set ragged right. This prevents needless hyphenation and makes it easier to read the text in the slightly narrower column.

The `bidi` option loads the `bidi` package which is used with \XeLaTeX to typeset bi-directional text. Since the `bidi` package needs to be loaded before the sidenotes and cite commands are defined, it can't be loaded in the document preamble.

The `debug` option causes the Tufte- \LaTeX classes to output debug information to the log file which is useful in troubleshooting bugs. It will also cause the graphics to be replaced by outlines.

The `nofonts` option prevents the Tufte- \LaTeX classes from automatically loading the Palatino and Helvetica typefaces. You should use this option if you wish to load your own fonts. If you're using \XeLaTeX , this option is implied (*i.e.*, the Palatino and Helvetica fonts aren't loaded if you use \XeLaTeX).

The `nols` option inhibits the letterspacing code. The Tufte- \LaTeX classes try to load the appropriate letterspacing package (either pdfTeX 's `letterspace` package or the `soul` package). If you're using \XeLaTeX with `fontenc`, however, you should configure your own letterspacing.

The `notitlepage` option causes `\maketitle` to generate a title block instead of a title page. The book class defaults to a title page and the handout class defaults to the title block. There is an analogous `titlepage` option that forces `\maketitle` to generate a full title page instead of the title block.

The `notoc` option suppresses Tufte- \LaTeX 's custom table of contents (toc) design. The current toc design only shows unnumbered chapter titles; it doesn't show sections or subsections. The `notoc` option will revert to \LaTeX 's toc design.

The `nohyper` option prevents the `hyperref` package from being loaded. The default is to load the `hyperref` package and use the `\title` and `\author` contents as metadata for the generated PDF.

Cross-Validation

Cross-validation (CV) is an estimation method used on supervised learning algorithms to assess their ability to predict the output of unseen data ⁸. Supervised learning algorithms are computational tasks like classification or regression, that learn an input-output function based on a set of samples. Such samples are also known as the labeled training data where each example consists of an input vector and its correct output value. After the training phase, a supervised learning algorithm should be able to use the inferred function in order to map new input unseen instances, known as testing data, to their correct output values ⁹. When the algorithm incorporates supervised feature selection, cross-validation should always be done external to the selection (feature-selection performed within every CV iteration) so as to ensure the test data remains unseen, reducing bias ¹⁰. Therefore, cross-validation, also known as out-of-sample testing, tests the function's ability to generalize to unseen situations ¹¹.

Cross-validation has two types of approaches, being i) the exhaustive cross validation approach which divides all the original samples in every possible way, forming training and test sets to train and test the model, and ii) the non-exhaustive cross validation approach which does not consider all the possible ways of splitting the original samples ¹². Each of these approaches are further divided into different cross-validation methods, which are explained below.

Exhaustive cross-validation

- Leave- p -out (LpO)

This method takes p samples from the data set as the test set and keeps the remaining as the training set, as shown in Fig. 4a. This is repeated for every combination of test and training set formed from the original data set and the average error is obtained. Therefore, this method trains and tests the algorithm $\binom{n}{p}$ times when the number of samples in the original data set is n , becoming inapplicable when $p > 1$ ¹³.

- Leave-one-out (LOO)

This method is a specific case of the LpO method having $p = 1$. It requires less computation efforts than LpO since the process is only repeated $n_{choose1} = n$ times, however might still be inapplicable for large values of n ¹⁴.

⁸ Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):91, 2006; and Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995

⁹ Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised and unsupervised feature selection. In *Proceedings of the national academy of sciences*, 99(10):6562–6566, 2002; and Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *Statistical learning with sparsity: the art of model selection*. Springer, 2010; and Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995

¹² Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010

¹³ Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010

¹⁴ Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010

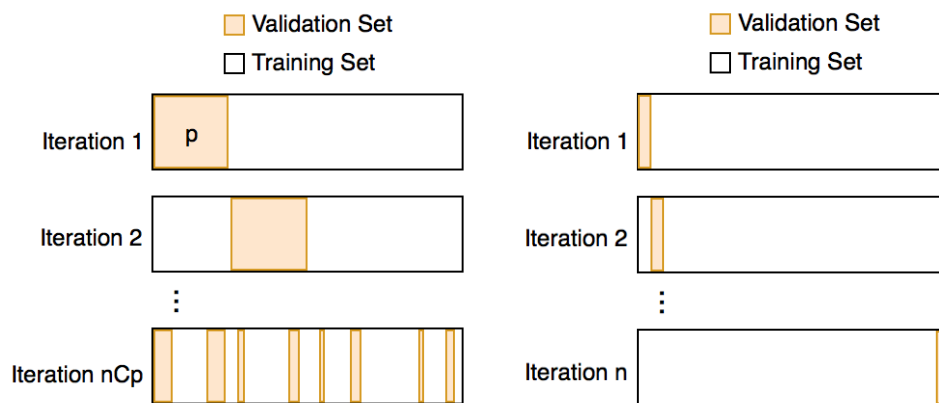


Figure 4: Exhaustive cross-validation methods: Leave-p-Out (left) & Leave-One-Out (right)

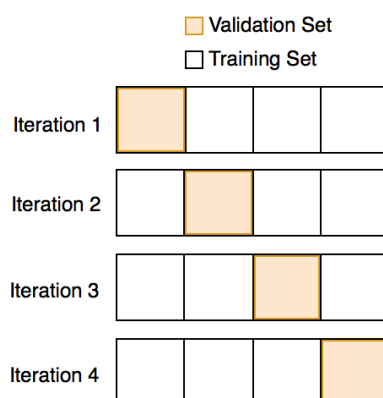
Non-exhaustive cross-validation

- Holdout method

This method randomly splits the original data set into two sets being the training set and the test set. Usually, the test set is smaller than the training set so that the algorithm has more data to train on. This method involves a single run and so must be used carefully to avoid misleading results. It is therefore sometimes not considered a CV method ¹⁵.

- k -fold

This method randomly splits the original data set into k equally sized subsets, as shown in Fig. 5. The function is then trained and validated k times, each time taking a different subset as the test data and the remaining $(k - 1)$ subsets as the training data, using each of the k subsets as the test set once. The k results are averaged to produce a single estimation. Stratified k -fold cross validation is a refinement of the k -fold method, which splits the original samples into equally sized and distributed subsets, having the same proportions of the different target labels ¹⁶.



- Repeated random sub-sampling

This method is also known as the Monte Carlo CV. It splits the data set randomly with replacement into training and test subsets

¹⁵ Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995

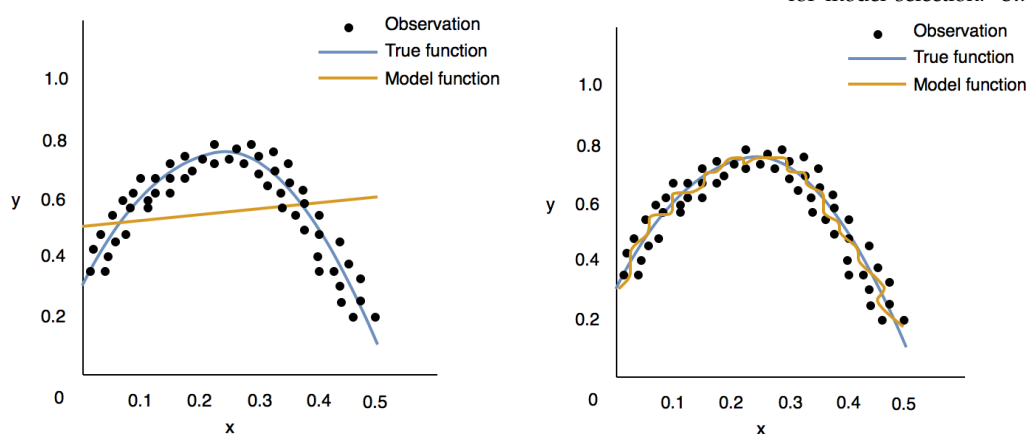
¹⁶ Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, Figure 5: k -Fold Cross Validation where $k=4$, volume 14, pages 1137–1145. Montreal, Canada, 1995

using some predefined split percentage, for every run. Therefore, this generates new training and test data for each run but the test data of the different runs might contain repeated samples, unlike that of k -fold ¹⁷.

All of the above cross-validation methods are used to check whether the model has been overfitted or underfitted and hence estimating the model's ability of fitting to independent data. Such ability is measured using quantitative metrics appropriate for the model and data ¹⁸. In the case of classification problems, the misclassification error rate is usually used whilst for regression problems, the mean squared error (MSE) is usually used. MSE is represented by Eq. 1, where n is the total number of test samples, Y_i is the true value of the i^{th} instance and \hat{Y}_i is the predicted value of the i^{th} instance.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (1)$$

Underfitting is when the model has a low degree (e.g. $y = x$, where the degree is 1) and so is not flexible enough to fit the data making the model have a low variance and high bias ¹⁹, as seen in Fig. 6a. Variance is the model's dependence on the training data and bias is model's assumption about the shape of the data ²⁰. On the other hand, as seen in Fig. 6b, overfitting is when the model has a too high degree (e.g. $y = x^{30}$, where the degree is 30) causing it to exactly fit the data as well as the noise and so lacks the ability to generalize ²¹, making the model have a high variance. Cross-validation helps reduce this bias and variance since it uses most of the data for both fitting and testing and so helps the model learn the actual relationship within the data. This makes cross-validation a good technique for models to acquire a good bias-variance tradeoff ²².



As stated in ²³, the LOO method gives a 0% accuracy on the test set when the number of target labels are equal to the number of instances in the dataset. It is shown that the k -fold CV method gives much better results, due to its lower variance, especially when $k = 10, 20$. Furthermore, R. Kohavi et al. state that the best accuracy is achieved

¹⁷ Qing-Song Xu and Yi-Zeng Liang. Monte carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56 (1):1–11, 2001.

¹⁸ Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995; and Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

¹⁹ Knut Baumann. Cross-validation as the objective function for variable-selection techniques. *TrAC Trends in Analytical Chemistry*, 22(6):395–406, 2003.

²⁰ Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

²¹ Knut Baumann. Cross-validation as the objective function for variable-selection techniques. *TrAC Trends in Analytical Chemistry*, 22(6):395–406, 2003.

²² Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*,

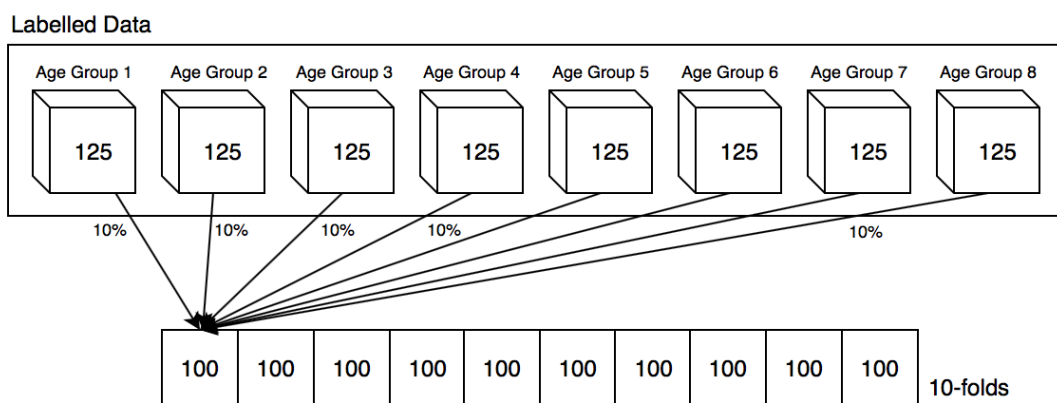
Figure 6: Underfitting (left) & Overfitting (right)

²³ Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

when using the stratified cross-validation method, since this has the least bias.

Therefore, let's take an example using the stratified k -fold cross-validation method with $k = 10$. Let's say that we are trying to solve age group classification, using eight non-overlapping age groups being 0-5, 6-10, 11-20, 21-30, 31-40, 41-50, 51-60, and 61+. We are using the FG-NET labelled data set, which contains around 1000 images of individuals aged between 0 and 69. Before we can start training our model (e.g. CNN), we must divide our data set into training and test subsets and this is where cross validation comes in. Therefore, we start by taking the 1000 images of our data set and splitting them according to their target class. Let us assume we have an equal amount of 125 (1000/8) images per class²⁴. As depicted in Fig. 7, we can now start forming our 10 folds by taking 10% of each age-group bucket, randomly without replacement. Hence, we will end up with 10 subsets of 100 images that are equally distributed along all age-groups. With these subsets, we can estimate our model's accuracy with a lower bias-variance tradeoff. Since we are using 10-fold CV, we will train and test our model 10 times. For the first iteration, we shall use subset 1 as the validation set and subsets 2 to 10 as the training set, for the second iteration we use subset 2 as the test set and subsets 1 plus 3 to 10 as our training set, and so on (as shown in Fig. 5). For each iteration we use the misclassification error rate to obtain an accuracy value and we finally average the 10 accuracy rates to obtain the global accuracy of our model when solving age group classification, given the FG-NET data set. Hence, we have now estimated the prediction error of the model and have an idea of how well our model performs in solving such a problem. It is important to note that cross-validation is *just* an estimation method and when using our model in real-life applications we do not apply CV but rather train our model with all the data we have.

²⁴ Down-sampling or up-sampling are common techniques used when there is an unequal amount of samples for the different classes.



As concluded by ²⁵, cross-validation is well implemented when everything is taken place within every CV iteration (including preprocessing, feature-selection, learning new algorithm parameter values, etc.), and the least bias can be achieved when using nested CV methods.

Figure 7: Stratified 10-fold cross-validation on 1000 labelled images of 8 different classes.
²⁵ Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):91, 2006

Confusion Matrix

A *Confusion Matrix*²⁶ (CM) is a cross-tabulation illustrating how well a model classifies categorical data²⁷. By convention, the CM of an N-class model is an N×N matrix indexed by the true class in the row dimension and the prediction in the column dimension (Table 2).

	Predicted Class	
	spam	¬spam
True Class	spam	10 1
	¬spam	2 100

Even though CMs are commonly used to evaluate binary classifiers, they aren't restricted to 2-class models²⁸. A CM of a multi-class model would show the number of times the classes were predicted correctly and which classes were confused with each other (Table 3).

	M&M's	Skittles	Smarties
M&M's	34	3	8
Skittles	1	28	5
Smarties	2	4	22

The CM of the model $h : X \mapsto C$ over the concept $c : X \mapsto C$ using dataset $S \subset X$ is formally defined²⁹ as a matrix Ξ such that $\Xi_{c,S}(h)[d_1, d_2] = |S_{h=d_1, c=d_2}|$. Algorithmically, the CM is constructed by incrementing the element corresponding to the true class *vis-a-vis* the predicted class for each object in the dataset (Algorithm 1).

$\Xi \leftarrow 0$
for $x \in S$ do
$d_1 \leftarrow c(x)$
$d_2 \leftarrow h(x)$
$\Xi_{d_1, d_2} \leftarrow \Xi_{d_1, d_2} + 1$

In binary classification, the CM consists of 2 specially designated classes called the *positive* class and the *negative* class³⁰. As indicated in Table 4, positive true class outcomes which are classified correctly are called *True Positives* (TP), whilst misclassifications are called *False Negatives* (FN). On the other hand, negative true class outcomes which are classified correctly are called *True Negatives* (TN), and misclassifications are called *False Positives* (FP). In natural sciences, FP are called *Type I Errors* and FN are known as *Type II Errors*³¹.

²⁶ Also called a *Contingency Table*.

²⁷ Claude Sammut and Geoffrey I Webb. *Encyclopedia of machine learning and data mining*. Springer, 2017

Table 2: CM of a hypothetical binary classifier which predicts whether out-of-sample text objects are spam or not. In this example, 10 spam and 100 non-spam objects are classified correctly, whilst 1 spam and 2 non-spam objects are misclassified.

²⁸ James H Martin and Daniel Jurafsky. *Speech and language processing*. Pearson, 2018

Table 3: CM of a hypothetical sweets classifier. The main diagonal of the CM shows the number of correct predictions, whilst the remaining elements indicate how many sweets were misclassified.

²⁹ Pawel Cichosz. *Data mining algorithms: explained using R*. Wiley, 2014

Algorithm 1: The CM is initialised to the zero matrix, and populated by iterating over all the objects x with corresponding true class d_1 and predicted class d_2 and incrementing the element (d_1, d_2) by 1 for each matching outcome.

³⁰ Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLoS one*, 10(3):e0118432, 2015

³¹ Alan H Fielding and John F Bell. A review of methods for the assessment of prediction errors in conservation presence/absence models. *Environmental conservation*, 24(1):38–49, 1997

	+ve	-ve
+ve	TP	FN
-ve	FP	TN

The information presented in the CM can be used to evaluate the performance of different binary classifiers³². A number of statistics (Eq. 2-8) derived from the CM have been proposed in the literature³³ to gain a better understanding of what are the strengths and weaknesses of different classifiers. Caution should be exercised when interpreting metrics³⁴, since the CM could be misleading if the data is imbalanced and an important subrange of the domain (e.g. a rare diseases) is underrepresented³⁵.

These metrics are important in situations in which a particular type of misclassification, i.e. FP or FN, could have worse consequences than the other³⁶. For example, FP are more tolerable than FN in classifiers which predict whether a patient has a disease or not. Even though both misclassifications are undesirable, in medical applications it's better to err on the side of caution since FN could be fatal.

Accuracy (ACC) is the proportion of correct predictions (Eq. 2). It is a class-insensitive metric because it can give a high rating to a model which classifies majority class objects correctly even though it misclassifies interesting minority class objects³⁷. The other metrics should be preferred since they're more class-sensitive and give better indicators when the dataset is imbalanced.

$$ACC = \frac{|TP \cup TN|}{|TP \cup FP \cup TN \cup FN|} \quad (2)$$

Negative Predictive Value (NPV) is the ratio of the correct negative predictions from the total negative predictions (Eq. 3).

$$NPV = \frac{|TN|}{|TN \cup FN|} \quad (3)$$

True Negative Rate (TNR), or *Specificity*, is the ratio of the correct negative predictions from the total true negatives (Eq. 4).

$$TNR = \frac{|TN|}{|TN \cup FP|} \quad (4)$$

True Positive Rate (TPR), *Sensitivity* or *Recall*, is the ratio of the correct positive predictions from the total true positives (Eq. 5).

$$TPR = \frac{|TP|}{|TP \cup FN|} \quad (5)$$

Sensitivity and Specificity can be combined into a single metric (Eq. 6). These metrics are often used in domains in which minority classes are important³⁸. For example, a medical classifier³⁹ could be evaluated by measuring Sensitivity to find the number of patients with a condition who tested positive, and Specificity to measure the number of patients who don't have the condition and tested negative.

Table 4: CMs of binary classifiers have positive (+ve) and negative (-ve) classes, and elements called *True Positives* (TP), *False Positives* (FP), *True Negatives* (TN) and *False Negatives* (FN).

³² Zhiyong Lu, Duane Szafron, Russell Greiner, Paul Lu, David S Wishart, Brett Poulin, John Anvik, Cam Macdonell, and Roman Eisner. Predicting sub-cellular localization of proteins using machine-learned classifiers. *Bioinformatics*, 20(4):547–556, 2004

³³ Xinyang Deng, Qi Liu, Yong Deng, and Sankaran Mahadevan. An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences*, 340:250–261, 2016

³⁴ László A Jeni, Jeffrey F Cohn, and Fernando De La Torre. Facing imbalanced data—recommendations for the use of performance metrics. In *Affective Computing and Intelligent Interaction (ACII)*, 2013 *Humaine Association Conference on*, pages 245–251. IEEE, 2013

³⁵ Troy Raeder, George Forman, and Nitesh V Chawla. Learning from imbalanced data: evaluation matters. In *Data mining: Foundations and intelligent paradigms*, pages 315–331. Springer, 2012

³⁶ Aboul Ella Hassanien and Diego Alberto Oliva. *Advances in Soft Computing and Machine Learning in Image Processing*, volume 730. Springer, 2017

³⁷ Paula Branco, Luís Torgo, and Rita P Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):31, 2016

³⁸ Max Kuhn and Kjell Johnson. *Applied predictive modeling*, volume 26. Springer, 2013

³⁹ El-Sayed Ahmed El-Dahshan, Tamer Hosny, and Abdel-Badeeh M Salem. Hybrid intelligent techniques for mri brain images classification. *Digital Signal Processing*, 20(2):433–441, 2010

$$\text{Sensitivity} \times \text{Specificity} = \frac{|TP| \times |TN|}{|TP \cup FN| \times |TN \cup FP|} \quad (6)$$

Positive Predictive Value (PPV), or *Precision*, is the ratio of the correct positive predictions from the total positive predictions (Eq. 7).

$$PPV = \frac{|TP|}{|TP \cup FP|} \quad (7)$$

Precision and Recall are borrowed from the discipline of *Information Extraction*⁴⁰. A composite metric called *F-score*, *F1-score*, or *F-measure* (Eq. 8) can be derived by finding their harmonic mean⁴¹.

$$F\text{-score} = 2 \times \frac{PPV \times TPR}{PPV + TPR} \quad (8)$$

The complements of ACC, NPV, TNR, TPR and PPV are called, respectively, *Error Rate*, *False Omission Rate*, *False Positive Rate*, *False Negative Rate* and *False Discovery Rate*.

Using all metrics could be counterproductive⁴² due to information redundancy, but none of the metrics is enough on its own. For instance, Recall is class-sensitive but it would give a perfect score to an inept model which simply returns the positive class. Thus, the best approach is to evaluate with complementary pairs⁴³ such as Sensitivity *vs.* Specificity, or Precision *vs.* Recall; or a combined measure such as the F-score.

The metrics can be adapted for evaluating multi-class models by decomposing an N-class CM into 2-class CMs and evaluating them individually⁴⁴. The literature describes two alternative methods for decomposing this kind of CM. In the *1-vs-1* approach, 2-class CMs are constructed for each pair combination as shown in Table 5.

+ve	-ve
M&M's	{Skittles, Smarties}
Skittles	{M&M's, Smarties}
Smarties	{M&M's, Skittles}

In the *1-vs-rest* approach, 2-class CMs are constructed for each class and the remaining classes combined together as shown in Table 6.

+ve	-ve
M&M's	Skittles \cup Smarties
Skittles	M&M's \cup Smarties
Smarties	Skittles \cup M&M's

Taking into account the above, CMs are suitable for visualising, evaluating, and comparing the performance of binary or multi-class classifiers. They should be used in conjunction with metrics such as the F-measure to avoid bias, especially if the dataset is unbalanced. For further details on the theoretical aspects of CMs and for practical examples in R refer to Cichosz⁴⁵; for examples in Python refer to Müller & Guido⁴⁶.

⁴⁰ Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009

⁴¹ John D Kelleher, Brian Mac Namee, and Aoife D'Arcy. *Fundamentals of machine learning for predictive data analytics*. MIT Press, 2015

⁴² Yan Ma and Bojan Cukic. Adequate and precise evaluation of quality models in software engineering studies. In *Proceedings of the third International workshop on predictor models in software engineering*, page 1. IEEE Computer Society, 2007

⁴³ Qiong Gu, Li Zhu, and Zhihua Cai. Evaluation measures of the classification performance of imbalanced data sets. In *International Symposium on Intelligence Computation and Applications*, pages 461–471. Springer, 2009

⁴⁴ Mathias Stager, Paul Lukowicz, and Gerhard Troster. Dealing with class skew in context recognition. In *26th International Conference on Distributed Computing Systems*, pages 58–58. IEEE, 2006

Table 5: 2-class CMs derived from the classes in Table 3. The +ve classes are paired separately with each -ve class.

Table 6: 2-class CMs derived through decomposition of the 3-class CM from Table 3 using the 1-vs-rest approach.

⁴⁵ Pawel Cichosz. *Data mining algorithms: explained using R*. Wiley, 2014

⁴⁶ Andreas C Müller, Sarah Guido, et al. *Introduction to machine learning with Python: a guide for data scientists*. O'Reilly, 2016

The following example is motivated by the samples in the *Scikit-Learn* documentation and the work of Géron⁴⁷. The models in Fig. 8 were trained on 70% of the *wines* dataset included with Scikit-Learn, whilst the remaining 30% was kept as a test set to evaluate the model. The dataset has 3 classes and 13 features, but only the results for class 1 and 2; and the alcohol and malic acid features; will be presented. The remaining class and features can be taken into consideration by adopting the 1-vs-1 multi-class evaluation approach. Furthermore, the dataset was scaled by standardising the features using Eq. 9.

$$\frac{x_i - \mu}{\sigma} \quad (9)$$

For all $x_i \in S$ using mean μ and standard deviation σ .

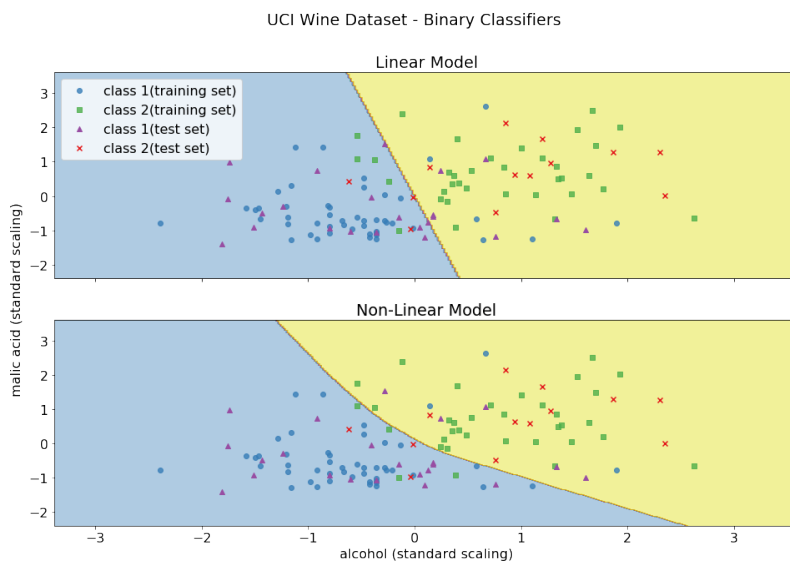


Figure 8: Decision boundary learned by a linear and non-linear binary classifier.

	Linear	Non-Linear
Accuracy	0.72	0.78
Specificity	0.77	0.77
Sensitivity	0.70	0.78
Precision	0.84	0.86
F-score	0.76	0.82

Table 7: Metrics derived from the CMs in Fig. 9.

As it can be intuitively deduced from Fig. 8, the decision boundary of the non-linear model is a better fit than the linear model. The CMs in Fig. 9 also show that non-linear model performs better with a higher TP, and consequently lower TN. The biggest advantage of the non-linear model is the higher Sensitivity resulting in a better F-score.

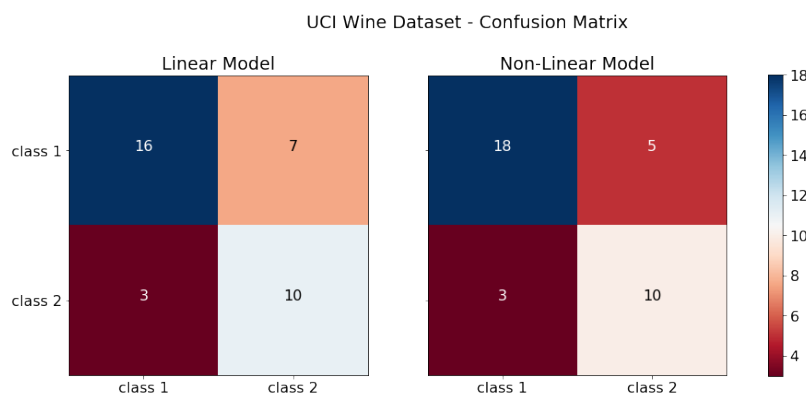


Figure 9: The linear classifier has 16 TP, 10 TN, 7 FN and 3 FP, whilst the non-linear classifier has 18 TP, 10 TN, 5 FN and 3 FP.

Bibliography

- Christophe Ambroise and Geoffrey J McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the national academy of sciences*, 99(10):6562–6566, 2002.
- Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.
- Knut Baumann. Cross-validation as the objective function for variable-selection techniques. *TrAC Trends in Analytical Chemistry*, 22(6):395–406, 2003.
- Paula Branco, Luís Torgo, and Rita P Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):31, 2016.
- Robert Bringhurst. *The Elements of Typography*. Hartley & Marks, 3.1 edition, 2005. ISBN 0-88179-205-5.
- Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.
- Pawel Cichosz. *Data mining algorithms: explained using R*. Wiley, 2014.
- Xinyang Deng, Qi Liu, Yong Deng, and Sankaran Mahadevan. An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences*, 340:250–261, 2016.
- El-Sayed Ahmed El-Dahshan, Tamer Hosny, and Abdel-Badeeh M Salem. Hybrid intelligent techniques for mri brain images classification. *Digital Signal Processing*, 20(2):433–441, 2010.
- Alan H Fielding and John F Bell. A review of methods for the assessment of prediction errors in conservation presence/absence models. *Environmental conservation*, 24(1):38–49, 1997.
- Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.
- Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly, 2017.

- Qiong Gu, Li Zhu, and Zhihua Cai. Evaluation measures of the classification performance of imbalanced data sets. In *International Symposium on Intelligence Computation and Applications*, pages 461–471. Springer, 2009.
- Aboul Ella Hassanien and Diego Alberto Oliva. *Advances in Soft Computing and Machine Learning in Image Processing*, volume 730. Springer, 2017.
- László A Jeni, Jeffrey F Cohn, and Fernando De La Torre. Facing imbalanced data—recommendations for the use of performance metrics. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 245–251. IEEE, 2013.
- John D Kelleher, Brian Mac Namee, and Aoife D’Arcy. *Fundamentals of machine learning for predictive data analytics*. MIT Press, 2015.
- Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.
- Max Kuhn and Kjell Johnson. *Applied predictive modeling*, volume 26. Springer, 2013.
- Zhiyong Lu, Duane Szafron, Russell Greiner, Paul Lu, David S Wishart, Brett Poulin, John Anvik, Cam Macdonell, and Roman Eisner. Predicting subcellular localization of proteins using machine-learned classifiers. *Bioinformatics*, 20(4):547–556, 2004.
- Yan Ma and Bojan Cukic. Adequate and precise evaluation of quality models in software engineering studies. In *Proceedings of the third International workshop on predictor models in software engineering*, page 1. IEEE Computer Society, 2007.
- James H Martin and Daniel Jurafsky. *Speech and language processing*. Pearson, 2018.
- Andreas C Müller, Sarah Guido, et al. *Introduction to machine learning with Python: a guide for data scientists*. O’Reilly, 2016.
- Troy Raeder, George Forman, and Nitesh V Chawla. Learning from imbalanced data: evaluation matters. In *Data mining: Foundations and intelligent paradigms*, pages 315–331. Springer, 2012.
- Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.
- Claude Sammut and Geoffrey I Webb. *Encyclopedia of machine learning and data mining*. Springer, 2017.
- Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

- Mathias Stager, Paul Lukowicz, and Gerhard Troster. Dealing with class skew in context recognition. In *26th International Conference on Distributed Computing Systems*, pages 58–58. IEEE, 2006.
- Edward R. Tufte. *Envisioning Information*. Graphics Press, Cheshire, Connecticut, 1990. ISBN 0-9613921-1-8.
- Edward R. Tufte. *Beautiful Evidence*. Graphics Press, LLC, first edition, May 2006. ISBN 0-9613921-7-7.
- Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):91, 2006.
- Qing-Song Xu and Yi-Zeng Liang. Monte carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56(1):1–11, 2001.