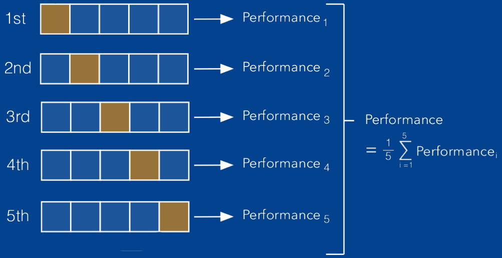
AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

LESSON 3: End-to-end ML

CARSTEN EIE FRIGAARD

AUTUMN 2020



"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." — Mitchell (1997).

Agenda

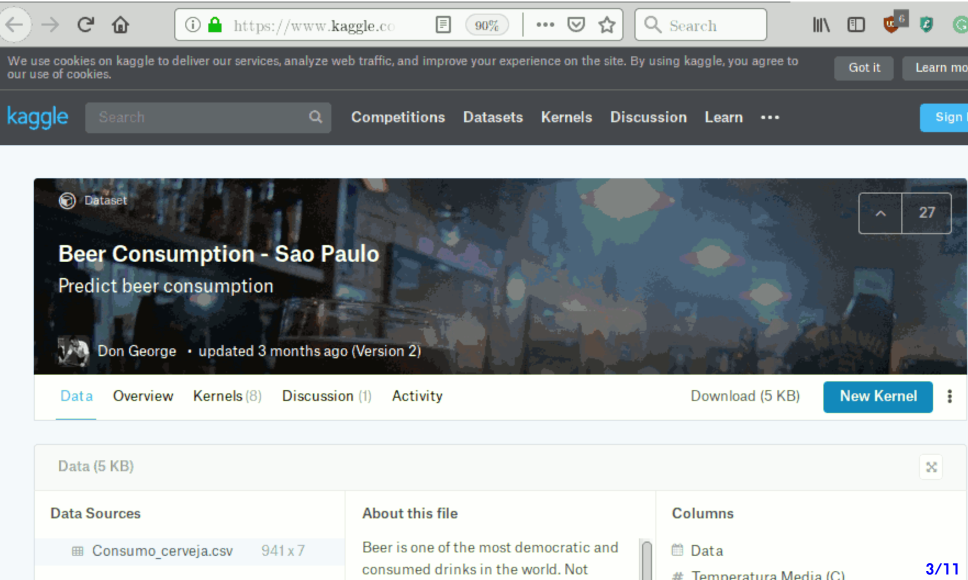
End-to-end Machine Learning

1. Admin
   - Zoom undervisnings, lektions videoer på BB,
   - Afleveringer, grupper, etc.
   - GITMAL og overskrivning af jeres filer!
2. Indledende undersøgelser og valg af data til slut-projekt (O4).
3. Generel repetition af § 2.
4. Algo. og Model selection, K-fold Cross validation.

Opg. L03 Beskrivelse af eget slutprojekt.pdf

Dit datasæt fra f.eks. `https://www.kaggle.com...`

(brug min login: user=cef@ase.au.dk, password=test123)



We use cookies on kaggle to deliver our services, analyze web traffic, and improve your experience on the site. By using kaggle, you agree to our use of cookies.

Beer Consumption - Sao Paulo

Predict beer consumption

Don George • updated 3 months ago (Version 2)

Data (5 KB)

Data Sources

Consumo_cerveja.csv	941 x 7

About this file

Beer is one of the most democratic and consumed drinks in the world. Not

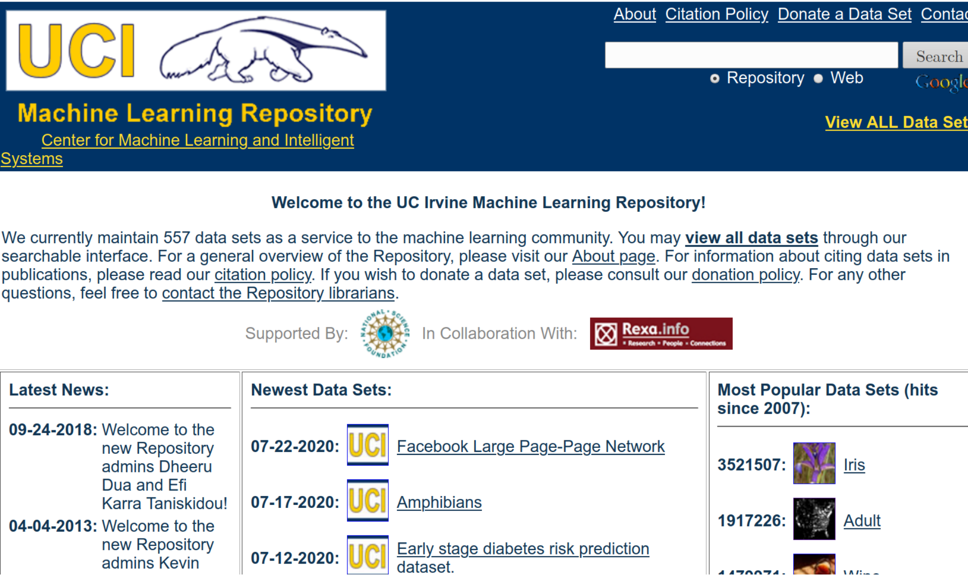
Columns

Data

Temperatura Media (C)

Opg. L03 Beskrivelse af eget slutprojekt.pdf

...eller UCI `https://archive.ics.uci.edu/ml/index.php`...



Center for Machine Learning and Intelligent Systems

Welcome to the UC Irvine Machine Learning Repository!

We currently maintain 557 data sets as a service to the machine learning community. You may **view all data sets** through our searchable interface. For a general overview of the Repository, please visit our About page. For information about citing data sets in publications, please read our citation policy. If you wish to donate a data set, please consult our donation policy. For any other questions, feel free to contact the Repository librarians.

Supported By: In Collaboration With: Rexa.info • Research • People • Connections

Latest News:

09-24-2018: Welcome to the new Repository admins Dheeru Dua and Efi Karra Taniskidou!

04-04-2013: Welcome to the new Repository admins Kevin

Newest Data Sets:

07-22-2020: Facebook Large Page-Page Network

07-17-2020: Amphibians

07-12-2020: Early stage diabetes risk prediction dataset.

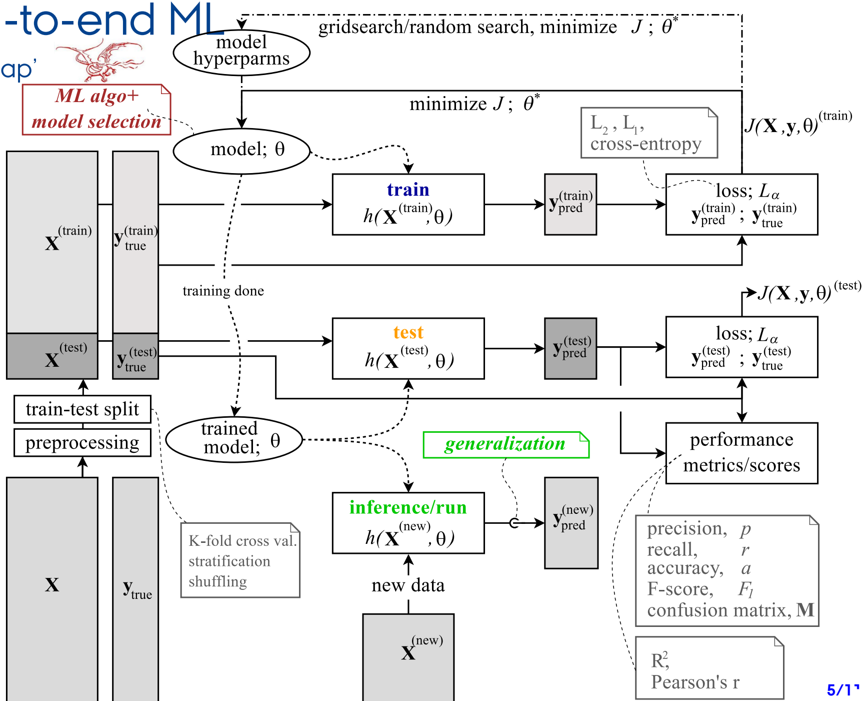
Most Popular Data Sets (hits since 2007):

3521507: Iris

1917226: Adult

End-to-end ML

'The Map'



gridsearch/random search, minimize J ; θ^*

model hyperparms

ML algo+ model selection

minimize J ; θ^*

$J(\mathbf{X},\mathbf{y},\theta)^{(\text{train})}$

model; θ

L_2, L_1, cross-entropy

train $h(\mathbf{X}^{(\text{train})},\theta)$

$\mathbf{y}^{(\text{train})}_{\text{pred}}$

loss; L_α $\mathbf{y}^{(\text{train})}_{\text{pred}}$; $\mathbf{y}^{(\text{train})}_{\text{true}}$

$\mathbf{X}^{(\text{train})}$

$\mathbf{y}^{(\text{train})}_{\text{true}}$

training done

$J(\mathbf{X},\mathbf{y},\theta)^{(\text{test})}$

$\mathbf{X}^{(\text{test})}$

$\mathbf{y}^{(\text{test})}_{\text{true}}$

test $h(\mathbf{X}^{(\text{test})},\theta)$

$\mathbf{y}^{(\text{test})}_{\text{pred}}$

loss; L_α $\mathbf{y}^{(\text{test})}_{\text{pred}}$; $\mathbf{y}^{(\text{test})}_{\text{true}}$

train-test split

preprocessing

trained model; θ

generalization

performance metrics/scores

inference/run $h(\mathbf{X}^{(\text{new})},\theta)$

$\mathbf{y}^{(\text{new})}_{\text{pred}}$

new data

$\mathbf{X}$

$\mathbf{y}_{\text{true}}$

K-fold cross val.
stratification
shuffling

$\mathbf{X}^{(\text{new})}$

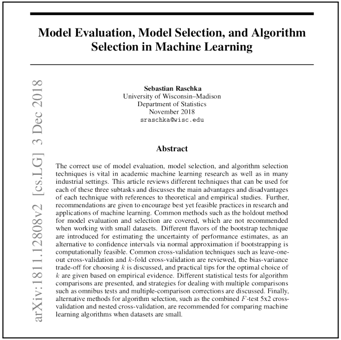
precision, p
recall, r
accuracy, a
F-score, F_1
confusion matrix, $\mathbf{M}$

R^2,
Pearson's r

ML Algorithm Selection and Model Selection

Manually Choosing an Algorithm and Tuning a Model..

- algorithm selection
  (choose a $h()$).
- model selection
  (set hyperparameters on $h()$),
- model evaluation,
- re-iteration and re-selection!



Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning

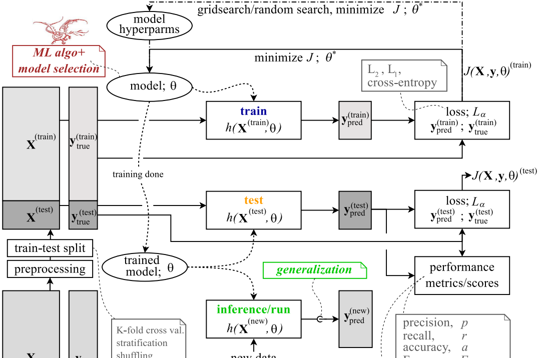
arXiv:1811.12808v2 [cs.LG] 3 Dec 2018

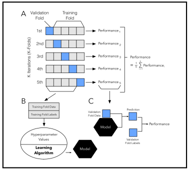
Sebastian Raschka
University of Wisconsin–Madison
Department of Statistics
November 2018
sraschka@wisc.edu

Abstract

The correct use of model evaluation, model selection, and algorithm selection techniques is vital in academic machine learning research as well as in many industrial settings. This article reviews different techniques that can be used for each of these three subtasks and discusses the main advantages and disadvantages of each technique with references to theoretical and empirical studies. Further, recommendations are given to encourage best yet feasible practices in research and applications of machine learning. Common methods such as the holdout method for model evaluation and selection are covered, which are not recommended when working with small datasets. Different flavors of the bootstrap technique are introduced for estimating the uncertainty of performance estimates, as an alternative to confidence intervals via normal approximation if bootstrapping is computationally feasible. Common cross-validation techniques such as leave-one-out cross-validation and k-fold cross-validation are reviewed, the bias-variance trade-off for choosing k is discussed, and practical tips for the optimal choice of k are given based on empirical evidence. Different statistical tests for algorithm comparisons are presented, and strategies for dealing with multiple comparisons such as omnibus tests and multiple-comparison corrections are discussed. Finally, alternative methods for algorithm selection, such as the combined F-test 5x2 cross-validation and nested cross-validation, are recommended for comparing machine learning algorithms when datasets are small.

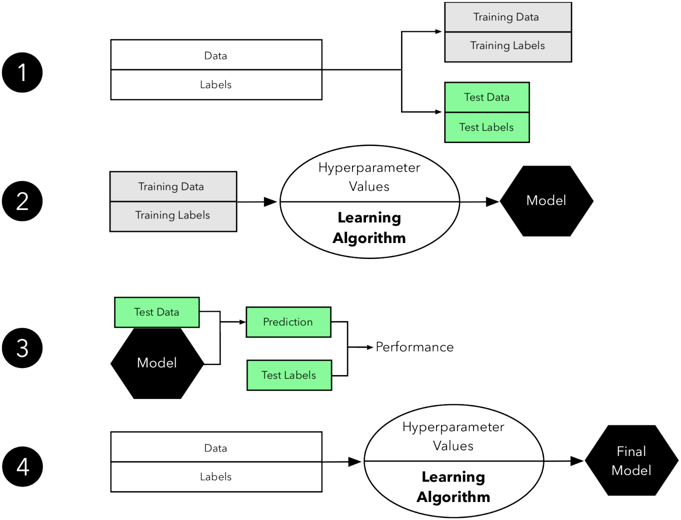
"Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning", Sebastian Raschka, 2018.


gridsearch/random search, minimize J ; θ^*
model hyperparms
ML algo+ model selection
minimize J ; θ^*
L_2 , L_1, cross-entropy
$J(\mathbf{X},\mathbf{y},\theta)^{(train)}$
model; θ
train
$h(\mathbf{X}^{(train)},\theta)$
$\mathbf{y}^{(train)}_{pred}$
loss; L_α
$\mathbf{y}^{(train)}_{pred}$; $\mathbf{y}^{(train)}_{true}$
$\mathbf{X}^{(train)}$
$\mathbf{y}^{(train)}_{true}$
training done
$J(\mathbf{X},\mathbf{y},\theta)^{(test)}$
test
$h(\mathbf{X}^{(test)},\theta)$
$\mathbf{y}^{(test)}_{pred}$
loss; L_α
$\mathbf{y}^{(test)}_{pred}$; $\mathbf{y}^{(test)}_{true}$
$\mathbf{X}^{(test)}$
$\mathbf{y}^{(test)}_{true}$
train-test split
preprocessing
trained model; θ
generalization
performance metrics/scores
inference/run
$h(\mathbf{X}^{(new)},\theta)$
$\mathbf{y}^{(new)}_{pred}$
new data
K-fold cross val.
stratification
shuffling
precision, p
recall, r
accuracy, a
X
y



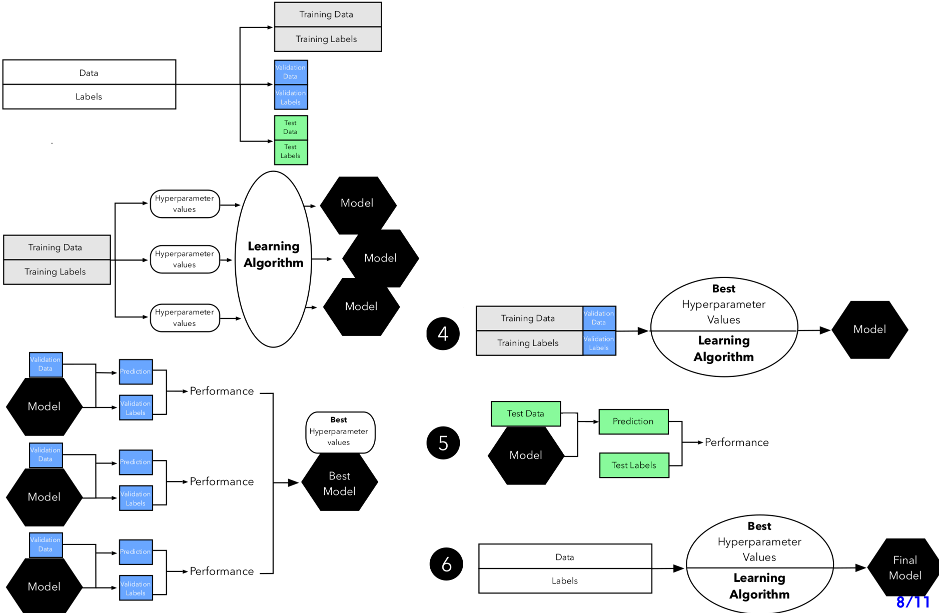
Model Evaluation

Simple Holdout Method (Train-Test Split)..



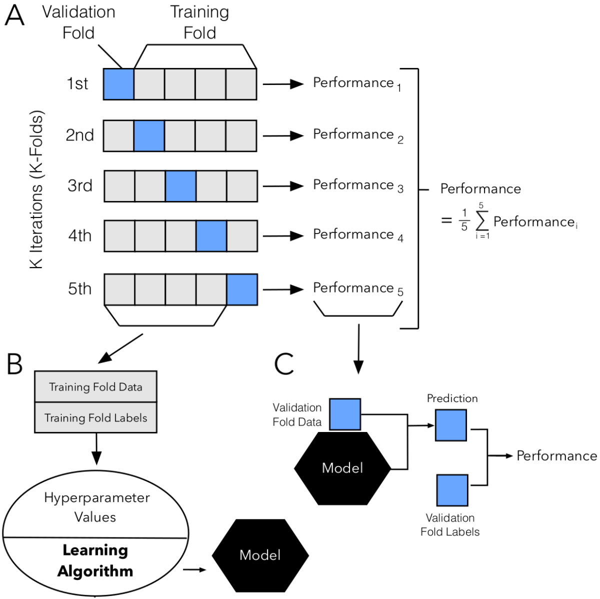
Model Evaluation and Selection

Three-way Holdout for Hyperparameter Tuning (Train-Validate-Test Split)...



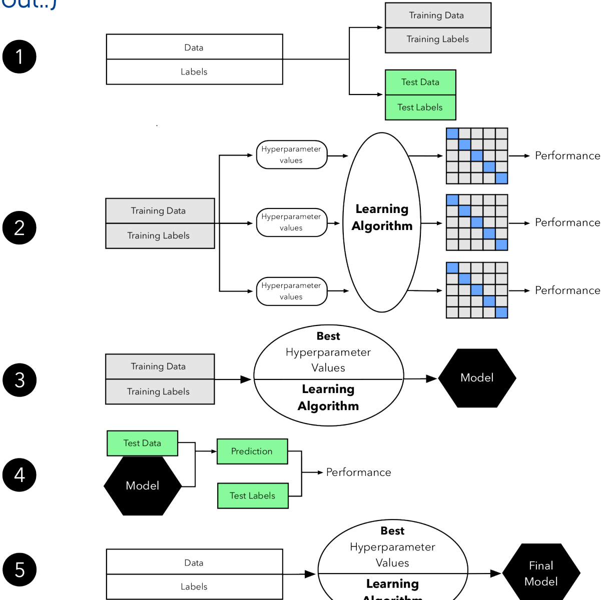
Model Evaluation

k-fold Cross-Validation Procedure, for *k*=5..

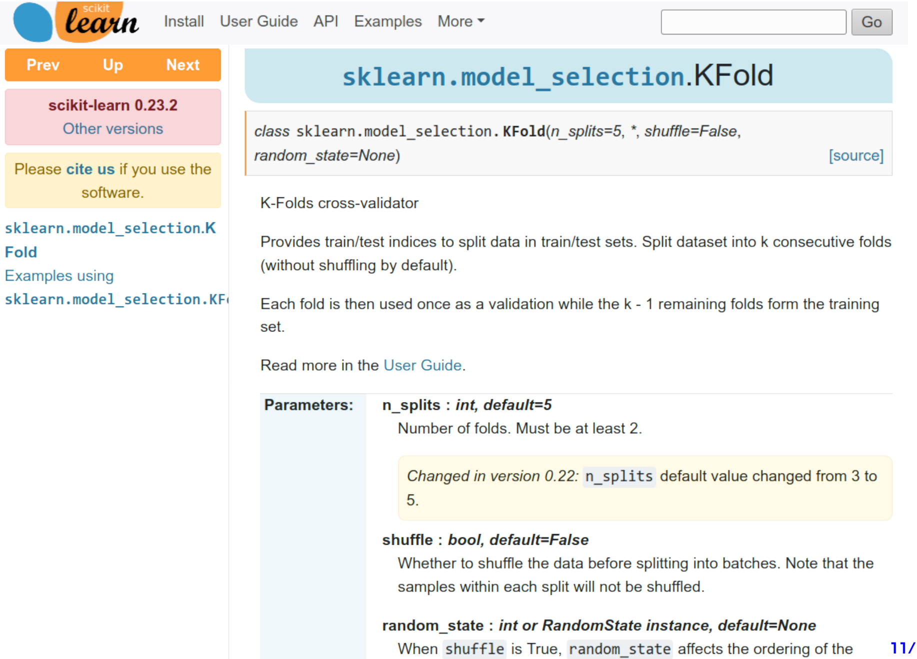


Model Evaluation and Selection

k-fold Cross-Validation for Hyperparameter Tuning (Somewhat Similar to Treeway Holdout..)



Scikit-learn K-fold Demo..



scikit-learn 0.23.2
Other versions

Please **cite us** if you use the software.

`sklearn.model_selection.KFold`
Examples using `sklearn.model_selection.KF`

`sklearn.model_selection`.KFold

class `sklearn.model_selection.`**`KFold`**(*n_splits=5*, ***, *shuffle=False*, *random_state=None*) [source]

K-Folds cross-validator

Provides train/test indices to split data in train/test sets. Split dataset into k consecutive folds (without shuffling by default).

Each fold is then used once as a validation while the k - 1 remaining folds form the training set.

Read more in the User Guide.

Parameters:

n_splits : ***int, default=5***

Number of folds. Must be at least 2.

> *Changed in version 0.22:* `n_splits` default value changed from 3 to 5.

shuffle : ***bool, default=False***

Whether to shuffle the data before splitting into batches. Note that the samples within each split will not be shuffled.

random_state : ***int or RandomState instance, default=None***

When `shuffle` is True, `random_state` affects the ordering of the