Machine Learning Mastery
Making Developers Awesome at Machine Learning

A Gentle Introduction to k-fold Cross-Validation

by **Jason Brownlee** on October 4, 2023 in **Statistics** 299

Cross-validation is a statistical method used to estimate the skill of machine learning models.

It is commonly used in applied machine learning to compare and select a model for a given predictive modeling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.

In this tutorial, you will discover a gentle introduction to the k-fold cross-validation procedure for estimating the skill of machine learning models.

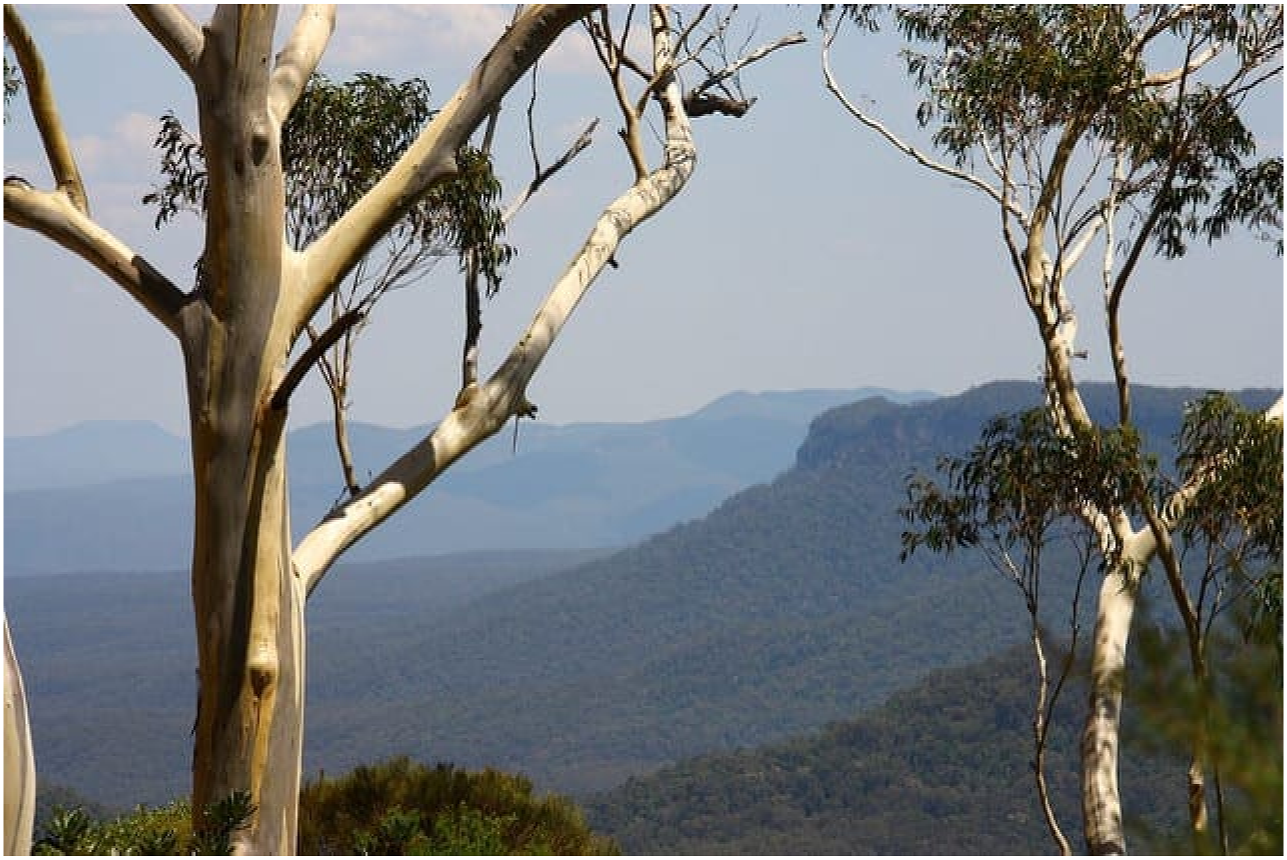
After completing this tutorial, you will know:

- That k-fold cross validation is a procedure used to estimate the skill of the model on new data.
- There are common tactics that you can use to select the value of k for your dataset.
- There are commonly used variations on cross-validation such as stratified and repeated that are available in scikit-learn.

Kick-start your project with my new book Statistics for Machine Learning, including *step-by-step tutorials* and the *Python source code* files for all examples.

Let’s get started.

- **Updated Jul/2020**: Added links to related types of cross-validation.



A Gentle Introduction to k-fold Cross-Validation
Photo by Jon Baldock, some rights reserved.

Tutorial Overview

This tutorial is divided into 5 parts; they are:

1. k-Fold Cross-Validation
2. Configuration of k
3. Worked Example
4. Cross-Validation API
5. Variations on Cross-Validation

k-Fold Cross-Validation

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

If you have a machine learning model and some data, you want to tell if your model can fit. You can split your data into training and test set. Train your model with the training set and evaluate the result with test set. But you evaluated the model only once and you are not sure your good result is by luck or not. You want to evaluate the model multiple times so you can be more confident about the model design.

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

Note that k-fold cross-validation is to evaluate the model design, not a particular training. Because you re-trained the model of the same design with different training sets.

3. For each unique group:
    1. Take the group as a hold out or test data set
    2. Take the remaining groups as a training data set
    3. Fit a model on the training set and evaluate it on the test set
    4. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores

Importantly, each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set 1 time and used to train the model k-1 times.

> *This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining k − 1 folds.*

— Page 181, An Introduction to Statistical Learning, 2013.

It is also important that any preparation of the data prior to fitting the model occur on the CV-assigned training dataset within the loop rather than on the broader data set. This also applies to any tuning of hyperparameters. A failure to perform these operations within the loop may result in data leakage and an optimistic estimate of the model skill.

> *Despite the best efforts of statistical methodologists, users frequently invalidate their results by inadvertently peeking at the test data.*

— Page 708, Artificial Intelligence: A Modern Approach (3rd Edition), 2009.

The results of a k-fold cross-validation run are often summarized with the mean of the model skill scores. It is also good practice to include a measure of the variance of the skill scores, such as the standard deviation or standard error.

Configuration of k

The k value must be chosen carefully for your data sample.

A poorly chosen value for k may result in a mis-representative idea of the skill of the model, such as a score with a high variance (that may change a lot based on the data used to fit the model), or a high bias, (such as an overestimate of the skill of the model).

Three common tactics for choosing a value for k are as follows:

- **Representative**: The value for k is chosen such that each train/test group of data samples is large enough to be statistically representative of the broader dataset.
- **k=10**: The value for k is fixed to 10, a value that has been found through experimentation to generally result in a model skill estimate with low bias a modest variance.
- **k=n**: The value for k is fixed to n, where n is the size of the dataset to give each test sample an opportunity to be used in the hold out dataset. This approach is called leave-one-out cross-validation.

> *The choice of k is usually 5 or 10, but there is no formal rule. As k gets larger, the difference in size between the training set and the resampling subsets gets smaller. As this difference decreases, the bias of the technique becomes smaller*

— Page 70, Applied Predictive Modeling, 2013.

A value of k=10 is very common in the field of applied machine learning, and is recommend if you are struggling to choose a value for your dataset.

> *or k = 10, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance.*

— Page 184, An Introduction to Statistical Learning, 2013.

If a value for k is chosen that does not evenly split the data sample, then one group will contain a remainder of the examples. It is preferable to split the data sample into k groups with the same number of samples, such that the sample of model skill scores are all equivalent.

For more on how to configure k-fold cross-validation, see the tutorial:

- How to Configure k-Fold Cross-Validation

Worked Example

To make the cross-validation procedure concrete, let’s look at a worked example.

Imagine we have a data sample with 6 observations:

```
[0.1, 0.2, 0.3, 0.4, 0.5, 0.6]
```

The first step is to pick a value for k in order to determine the number of folds used to split the data. Here, we will use a value of k=3. That means we will shuffle the data and then split the data into 3 groups. Because we have 6 observations, each group will have an equal number of 2 observations.

For example:

We can then make use of the sample, such as to evaluate the skill of a machine learning algorithm.

Three models are trained and evaluated with each fold given a chance to be the held out test set.

For example:

- **Model1**: Trained on Fold1 + Fold2, Tested on Fold3
- **Model2**: Trained on Fold2 + Fold3, Tested on Fold1
- **Model3**: Trained on Fold1 + Fold3, Tested on Fold2

The models are then discarded after they are evaluated as they have served their purpose.

The skill scores are collected for each model and summarized for use.

Cross-Validation API

We do not have to implement k-fold cross-validation manually. The scikit-learn library provides an implementation that will split a given data sample up.

The *KFold()* scikit-learn class can be used. It takes as arguments the number of splits, whether or not to shuffle the sample, and the seed for the pseudorandom number generator used prior to the shuffle.

For example, we can create an instance that splits a dataset into 3 folds, shuffles prior to the split, and uses a value of 1 for the pseudorandom number generator.

```
kfold = KFold(3, True, 1)
```

containing the indexes into the original data sample of observations to use for train and test sets on each iteration.

For example, we can enumerate the splits of the indices for a data sample using the created *KFold* instance as follows:

```
# enumerate splits
for train, test in kfold.split(data):
 print('train: %s, test: %s' % (train, test))
```

We can tie all of this together with our small dataset used in the worked example of the prior section.

```
# scikit-learn k-fold cross-validation
from numpy import array
from sklearn.model_selection import KFold
# data sample
data = array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6])
# prepare cross validation
kfold = KFold(3, True, 1)
# enumerate splits
for train, test in kfold.split(data):
 print('train: %s, test: %s' % (data[train], data[test]))
```

Running the example prints the specific observations chosen for each train and test set. The indices are used directly on the original data array to retrieve the observation values.

```
train: [0.1 0.4 0.5 0.6], test: [0.2 0.3]
train: [0.2 0.3 0.4 0.6], test: [0.1 0.5]
train: [0.1 0.2 0.3 0.5], test: [0.4 0.6]
```

Usefully, the k-fold cross validation implementation in scikit-learn is provided as a component operation within broader methods, such as grid-searching model hyperparameters and scoring a model on a dataset.

Nevertheless, the *KFold* class can be used directly in order to split up a dataset prior to modeling such that all models will use the same data splits. This is especially helpful if you are working with very large data samples. The use of the same splits across algorithms can have benefits for statistical tests that you may wish to perform on the data later.

Variations on Cross-Validation

There are a number of variations on the k-fold cross validation procedure.

- **LOOCV**: Taken to another extreme, k may be set to the total number of observations in the dataset such that each observation is given a chance to be the held out of the dataset. This is called leave-one-out cross-validation, or LOOCV for short.
- **Stratified**: The splitting of data into folds may be governed by criteria such as ensuring that each fold has the same proportion of observations with a given categorical value, such as the class outcome value. This is called stratified cross-validation.
- **Repeated**: This is where the k-fold cross-validation procedure is repeated n times, where importantly, the data sample is shuffled prior to each repetition, which results in a different split of the sample.
- **Nested**: This is where k-fold cross-validation is performed within each fold of cross-validation, often to perform hyperparameter tuning during model evaluation. This is called nested cross-validation or double cross-validation.

Extensions

This section lists some ideas for extending the tutorial that you may wish to explore.

- Find 3 machine learning research papers that use a value of 10 for k-fold cross-validation.
- Write your own function to split a data sample using k-fold cross-validation.
- Develop examples to demonstrate each of the main types of cross-validation supported by scikit-learn.

If you explore any of these extensions, I’d love to know.

Further Reading

This section provides more resources on the topic if you are looking to go deeper.

Related Tutorials

- How to Configure k-Fold Cross-Validation
- LOOCV for Evaluating Machine Learning Algorithms
- Nested Cross-Validation for Machine Learning with Python

Books

- Applied Predictive Modeling, 2013.
- An Introduction to Statistical Learning, 2013.
- Artificial Intelligence: A Modern Approach (3rd Edition), 2009.

API

- sklearn.model_selection.KFold() API
  sklearn.model_selection: Model Selection API

Articles

- Resampling (statistics) on Wikipedia
- Cross-validation (statistics) on Wikipedia

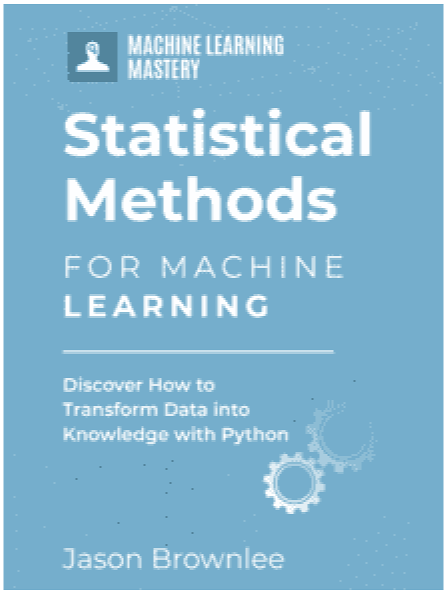
Summary

In this tutorial, you discovered a gentle introduction to the k-fold cross-validation procedure for estimating the skill of machine learning models.

Specifically, you learned:

- That k-fold cross validation is a procedure used to estimate the skill of the model on new data.
- There are common tactics that you can use to select the value of k for your dataset.
- There are commonly used variations on cross-validation, such as stratified and repeated, that are available in scikit-learn.

Do you have any questions?
Ask your questions in the comments below and I will do my best to answer.



MACHINE LEARNING MASTERY

Statistical Methods

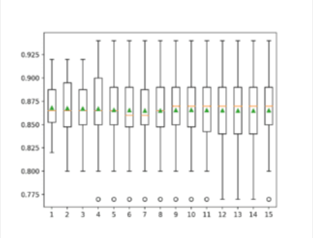
FOR MACHINE LEARNING

Discover How to Transform Data into Knowledge with Python

Jason Brownlee

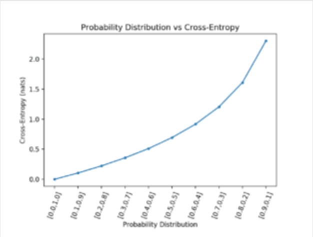
Share Tweet Share

More On This Topic



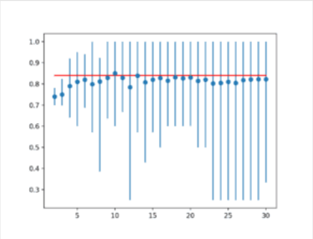
Repeated k-Fold Cross-Validation for Model…

Nested Cross-Validation for Machine Learning with Python

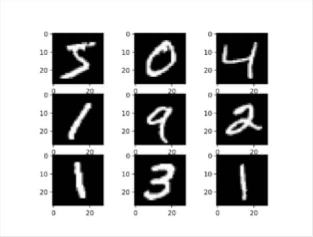


A Gentle Introduction to Cross-Entropy for Machine Learning

How to Use Out-of-Fold Predictions in Machine Learning



How to Configure k-Fold Cross-Validation



How to Develop a CNN for MNIST Handwritten Digit…

About Jason Brownlee

Jason Brownlee, PhD is a machine learning specialist who teaches developers how to get results with modern machine learning methods via hands-on tutorials.

View all posts by Jason Brownlee →

< How to Transform Data to Better Fit The Normal Distribution

A Gentle Introduction to the Bootstrap Method >

299 Responses to *A Gentle Introduction to k-fold Cross-Validation*

Kristian Lunow Nielsen May 25, 2018 at 4:30 pm #

REPLY ↩

Hi Jason

Nice gentle tutorial you have made there!
I have a more technical question; Can you comment on why the error estimate obtained through k-fold-cross-validation is almost unbiased? with an emphasis on why.

I have had a hard time finding literature describing why.
It is my understanding that everyone comments on the bias/variance trade-off when asked about the almost unbiased feature of k-fold-cross-validation.

✕

Thanks.

Good question.

We repeat the model evaluation process multiple times (instead of one time) and calculate the mean skill. The mean estimate of any parameter is less biased than a one-shot estimate. There is still some bias though.

The cost is we get variance on this estimate, so it's good to report both mean and variance or mean and stdev of the score.

walid July 15, 2020 at 5:31 am # REPLY

when i validate my model with cross validation i can see every time i get new result from my model. My sample size is 325. can you explain why it is happened and what is the solution?

Jason Brownlee July 15, 2020 at 8:30 am # REPLY

Yes, this is to be expected. We report the average performance of the model on the dataset.

This may also help:
https://machinelearningmastery.com/faq/single-faq/why-do-i-get-different-results-each-time-i-run-the-code

aya January 22, 2021 at 9:37 am # REPLY

thanks, i tried to make reasult one at the same time of run , using stratified k fold, is this true and what is the best value of random state and how select it?

Jason Brownlee January 22, 2021 at 1:21 pm #

This is a common question that I answer here:
https://machinelearningmastery.com/faq/single-faq/what-value-should-i-set-for-the-random-number-seed

learn/issues/4757 but this is exactly what I need in my work. I do it like this:

```
from sklearn.model_selection import StratifiedKFold
import numpy as np

n_splits = 3

X = np.ones(10)
y = np.arange(1,11,dtype=float)

# binning to make StratifiedKFold work
yc = np.outer(y[::n_splits],np.ones(n_splits)).flatten()[:len(y)]
yc[-n_splits:]=yc[-n_splits]*np.ones(n_splits)

skf = StratifiedKFold(n_splits=n_splits)
for train, test in skf.split(X, yc):
    print("train: %s test: %s" % (train, test))
```

Vladislav Gladkikh May 25, 2018 at 7:26 pm #

REPLY

How to make code formatting here?

Jason Brownlee May 26, 2018 at 5:53 am #

REPLY

You can use PRE HTML tags. I formatted your code for you.

Zeinab January 20, 2020 at 5:31 am #

REPLY

Do you have a tutorial about python for machine learning; that include all python basics needed for machine learning?

Jason Brownlee January 20, 2020 at 8:45 am #

Yes many, perhaps start here:
https://machinelearningmastery.com/start-here/#python

Jason Brownlee May 26, 2018 at 5:53 am #

REPLY

Thanks for sharing!

Should be used k cross-validation in deep learning?

Jason Brownlee May 30, 2018 at 6:44 am #

REPLY ↰

It can be for small networks/datasets.

Often it is too slow.

Chan June 8, 2018 at 9:45 pm #

REPLY ↰

Dear Jason,

Thanks for this insight ,especially the worked example section. It’s very helpful to understand the fundamentals. However, I have a basic question which I didn’t understand completely.
If we throw away all the models that we learn from every group (3 models in your example shown), what would be the final model to predict unseen /test data?

Is it something like:

We are using cross-validation only to choose the right hyper-parameter for a model? say K for KNN.
1. We fix a value of K;train and cross-validate to get three different models with different parameters (/coefficients like Y=3x+2; Y=2x+3; Y=2.5X+3 = just some random values)
2. Every model has its own error rate. Average them out to get a mean error rate for that hyper-parameter setup / values
3. Try with other values of Hyper-parameters (step 1 and 2 repetitively for all set of hyper-parameter values)

4. Choose the hyper-parameter set with the least average error
5. Train the whole training data set (without any validation split this time) with new value of hyper-parameter and get the new model [Y=2.75X+2.5 for eg.,]
6. Use this as a model to predict the new / unseen / test data. Loss value would be the final error from this model

Is this the way? or May be I understood it completely wrong.

Sorry for this naive question as I’m quite new or just a started. Thanks for your understanding 🙂

Jason Brownlee June 9, 2018 at 6:52 am #

REPLY ↰

I explain how to develop a final model here:
https://machinelearningmastery.com/train-final-machine-learning-model/

✕

Hi, I am working on a project and I have 200,000 observations and am a little confused between test set and crossvalidation.
1. I split this dataset into training, which has 70% of the observations and testing which has the remaining 30% of the observations.
2. I am running Rweka to create a decision tree model on the training dataset and then utilize this model to make predictions on the test data set.
3. My confusion matrix will give me the actual test class vs predicted class to evaluate the model. Is this correct?
4. Do I need to evaluate the weka classifer on the training data set and when I do this should I use cross-validation? Or is this not necessary because I have a test set and I already plan to see the confusion matrix here to assess performance?I am a little confused here. Anything you can do to help will be appreciated.
Thanks

Jason Brownlee June 5, 2020 at 8:20 am # REPLY

Generally, you must choose an appropriate model evaluation strategy for your dataset.

One approach is to use a train/test set.
Another is to use k-fold cross-validation on all the dataset.

If you are not sure, then perhaps use k-fold cross-validation.

Also, this may help:
https://machinelearningmastery.com/estimate-performance-machine-learning-algorithms-weka/

rakesh November 7, 2020 at 4:16 pm #

Sir, Is it possible to split the entire dataset into train and test sample and then apply k-fold-cross-validation on the train dataset and evaluate the performance on test dataset.

I have 2500 data sample. First I split it into 2000 for training and 500 for testing.
Then I applied 10-fold on training dataset and I evaluate the performance avg.
Then I fit into test sample. I just want to know wheather it is a right way or not.

Jason Brownlee November 8, 2020 at 6:37 am #

Jac February 17, 2022 at 10:34 am #

So eg you take tome separeted amples for test set – that configuration mogth happen to be useful irl

James Carmichael February 17, 2022 at 1:18 pm #

Hi Jac…Thank you for the feedback! Let me know if you have any specific questions I may help with.

Hoc November 24, 2020 at 7:58 pm # REPLY ↰

Dear Chan,

Your question is so interesting, and I have a concern like you, but I did not found the best answer. I think you have the answer to your question, would you mind if you help me to explain it.

Thank you so much
Regards

Tina August 25, 2021 at 4:33 am # REPLY ↰

Hi Hoc, I have the same question here. Have you figure it out clearly?

teja_chebrole June 21, 2018 at 9:40 pm # REPLY ↰

awesome article..very useful…

Jason Brownlee June 22, 2018 at 6:06 am # REPLY ↰

I'm glad it helped.

REPLY ↰

✕

Jason Brownlee January 20, 2020 at 8:45 am # REPLY

Repeated cross-validation repeats the cross-validation procedure with different splits of data (folds) each repeat.

M.sarat chandra July 7, 2018 at 5:32 pm # REPLY

if loocv is done it increase the size of k as datasets increase size .what would u say abt this. when to use loocv on data. what is use of pseudo random number generator.

Jason Brownlee July 8, 2018 at 6:17 am # REPLY

In turn it increases the number of models to fit and the time it will take to evaluate.

The choice of random numbers does not matter as long as you are consistent in your experiment.

marison July 10, 2018 at 4:20 pm # REPLY

hi,

1. can u plz provide me a code for implementing the k-fold cross validation in R ?

2. do we have to do cross validation on complete data set or only on the training dataset after splitting into training and testing dataset?

Jason Brownlee July 11, 2018 at 5:52 am # REPLY

Here is an example in R:
https://machinelearningmastery.com/evaluate-machine-learning-algorithms-with-r/

It is your choice how to estimate the skill of a model on unseen data.

Zhian July 16, 2018 at 7:36 pm # REPLY

Hello,

or choose a part of every experiment for that purpose? every experiment contain different features which control the state of the system. When I want to validate I would like to to take the initial state of the system and with the vector of features to propagate the state in time. This is exactly what I need in practice.

Could you please provide me your comments on that. I hope I am clear abot my issue.
Thanks.

Jason Brownlee July 17, 2018 at 6:16 am # REPLY

You could use walk-forward validation:
https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/

Tamara August 8, 2018 at 5:29 am # REPLY

Hi Jason,
Firstly, your tutorials are excellent and very helpful. Thank you so much!
I have a question related to the use of k-fold cross-validation (k-fold CV) in testing the validity of a neural network model (how well it performs for new data). I'm afraid there is some confusion in this field as k-fold CV appears to be required for justifying any results.
So far I understand we can use k-fold CV to find optimal parameters while defining the network (as accuracy for train and test data will tell when it is over or under fitting) and we can make the choices that ensure good performance. Once we made these choices we can run the algorithm for the entire training data and we generate a model. This model has to be then tested for new data (validation set and training set). My question is: on how many new data sets has this model to be tested din order to be considered useful?
Since we have a model, using again k-fold CV does not help (we do not look for a new model). I my understanding the k-fold CV testing is mainly for the algorithm/method optimization while the final model should be only tested on new data. Is this correct? if so, should I split the test data into smaller sets, and use these as multiple tests, or using just the one test data set is enough?

Many thanks,
Tamara

Jason Brownlee August 8, 2018 at 6:25 am # REPLY

Often we split the training dataset into train and validation and use the validation to tune hyperparameters.

Perhaps this post will help:
https://machinelearningmastery.com/difference-test-validation-datasets/

ashish August 14, 2018 at 7:21 pm # REPLY ↩

Hi jason , thanks for a nice blog

my dataset size is 6000 (image data). how do we know which type of cross validation should use (simply train test split or k- fold cross validation) .

Jason Brownlee August 15, 2018 at 5:58 am # REPLY ↩

Start with 10-folds.

Carlos August 16, 2018 at 2:46 am # REPLY ↩

Good morning!

I am an Economics Student at University of São Paulo and I am researching about Backtesting, Stress Test and Validation Models to Credit Risk. Thus, would you help me answering some questions? I researching how to create a good procedure to validate prediction models that tries to forecast default behavior of the agents. Thereby, suppose a log-odds logit model of Default Probability that uses some explanatory variables as GDP, Official Interest Rates, etc. In order to evaluate it, I calculate the stability and the backtesting, using part of my data not used in the estimation with this purpose. In the backtesting case, I use a forecast, based on the regression of relevant variables to perceive if my model is corresponding to the forecast that has interval of confidence to evaluate if they are in or out. Furthermore, I evaluate the signal of the parameters to verify if it is beavering according to the economic sense.
After reading some papers, including your publication here and a Basel one ("Sound Practices for Backtesting Counterparty Credit Risk Models"), I have some doubts.

1) Do a pattern backtesting procedure lead completely about the overfitting issue? If not, which the recommendations to solve it?
2) What are the issues not covered by a pattern backtesting procedure and we should pay attention using another metrics to lead with them?
3) Could you indicate some paper or document that explains about Back-pricing, conception introduced by "Sound Practices for Backtesting Counterparty Credit Risk Models"? I have not found another document and I had not understood their explanation.
"A bank can carry out additional validation work to support the quality of its models by carrying out back-pricing. Back-pricing, which is similar to backtesting, is a quantitative comparison of model predictions with realizations, but based on re-running current models on historical market data. In order to make meaningful statements about the performance of the model, the historical data need to be divided into distinct calibration and verification data sets for each initialization date, with the model calibrated using the calibration data set before the initialization date and the forecasts after initialization tested on the

Thus, I appreciate your attention and help.

The best regards.

Jason Brownlee August 16, 2018 at 6:12 am # REPLY

Too much for one comment, sorry. One small question at a time please.

You can get started with back-testing procedures for time series forecasting here:
https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/

Scott Miller September 6, 2018 at 11:48 pm # REPLY

Hi Jason, I'm using k-fold with regularized linear regression (Ridge) with the objective to determine the optimial regularization parameter.

For each regularization parameter, I do k-fold CV to compute the CV error.

I then select the regularization parmeter that achieves the lowest CV error.

However, in k-fold when I use 'shuffle=True' AND no 'random_state' in k-fold, the optimal regularization parameter changes each time I run the program.

kf=KFold(n_splits=n_kfolds, shuffle=True)

If I use a random state or 'shuffle = False', the results are always the same.

Question: Do you feel this is normal behavior and any recommendations.

note: Predictions are really good, just looking for general discussion.

Thanks.

Jason Brownlee September 7, 2018 at 8:06 am # REPLY

Yes, it might be a good idea to repeat each experiment to counter the variance of the model.

Going even one step further, you might even want to use statistical tests to help determine whether "better" is real or noise. I have tutorials on this under the topic of statistics I believe.

Pascal Schmidt October 4, 2018 at 1:35 pm # REPLY

When I do feature selection before cross validation then my error will be biased because I chose the features based on training and testing set (data leakage). Therefore, I believe I have to do feature selection inside the cross validation loop with only the training data and then test my model on the test data.

So my question is when I end up with different predictors for the different folds, should I choose the predictors that occured the majority of the time? And after that, should I do cross validation for this model with the same predictors? So, do k-fold cv with my final model where every predictor is the same for the different folds? And then use this estimate to be my cv error?

It would be really great if you could help me out. Thanks again for the article and keep up the great work.

Jason Brownlee October 4, 2018 at 3:30 pm # REPLY

Thanks.

Correct. Yes, you will get different features, and perhaps you can take the average across the findings from each fold.

Alternately, you can use one hold out dataset to choose features, and a separate set for estimating model performance/tuning.

It comes down to how much data you have to “spend” and how much leakage/bias you can handle. We almost never have enough data to be pure.

Pascal Schmidt October 6, 2018 at 3:32 am # REPLY

Thanks, Jason. I guess statistics is not as black and white as a discipline like mathematics. A lot of different ways to deal with problems and no one best solution exists. This makes it so challenging I feel. A lot of experience is required to deal with all these unique data sets.

Jason Brownlee October 6, 2018 at 5:50 am # REPLY

Yes, the best way to get good is to practice, like programming, driving, and everything else we want to do in life.

Bilal October 16, 2018 at 6:16 pm # REPLY

for which purpose we calculate the standard deviation from any data set.

In what context?

Leontine Ham October 16, 2018 at 9:21 pm #

REPLY

Thank you for explaining the fundamentals of CV.
I am working with repeated (50x) 5-fold cross validation, but I am trying to figure out which statistical test I can use in order to compare two datasets. Can you help me? Or is that out of the scope of this blog?

Jason Brownlee October 17, 2018 at 6:50 am #

REPLY

Yes, see this post:
https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/

kingshuk October 22, 2018 at 1:27 am #

REPLY

Hi Jason ,

What is the difference between Kfold and Stratified K fold?

Jason Brownlee October 22, 2018 at 6:21 am #

REPLY

Kfold uses random split of the into k folds.
Stratified tries to maintain the same distribution of the target variable when randomly selecting examples for each fold.

Rana Muhammad Kashif December 5, 2018 at 3:30 pm #

REPLY

Thanks for this post!

Can we split the data by ourselves and then train some data and test the remaining?
For example, my data is on cricket and i want to train the data based on two splits i.e. 0-6 overs and 7-15 overs, and test the 16-20 overs data in a 20 overs match. Is it rational? If yes how can we do this within R?

You can get started with caret in R here:
https://machinelearningmastery.com/start-here/#r

Ruslan December 5, 2018 at 10:19 pm # REPLY

Hi Jason! Good article!

What should we do when not all parts are equal? Say we have 5 5 5 5 6 or 7 7 7 8 or 9 9 9 9 8

Should we skip the biggest/least one? Should we apply weighting somehow? Do the same as if it had the same size?

Thank you.

Jason Brownlee December 6, 2018 at 5:55 am # REPLY

Try to make each fold equal, but if they are mostly equal, that is okay.

Jason Quadras January 17, 2019 at 1:08 am # REPLY

Very Good article. Simple and easy to understand!

Jason Brownlee January 17, 2019 at 5:28 am # REPLY

Thanks, I'm glad it helped.

Rose January 17, 2019 at 3:44 pm # REPLY

Hi Jason
Thanks for this post !
How to evaluate the overall accuracy of learning classifiers in K folds cross validation ?
I think that
Accuracy = (sum of accuracy in each folds)/K;
This is true or false ?

Yes, the average of the accuracy scores of the model as calculated across the test folds.

Oscar January 22, 2019 at 3:05 am #

REPLY ↩

Hello Jason,

One of the best tutorials on CV that I have found. But there is still something I don't get. What is the point of doing all this if in the end you just discard the models? I've been having a lot of problems with this, because I find different information in different places:

* In some tutorials, it is said that you use always the same model for training and validation iteratively, keeping a test set independent for when you finish training with CV, so you can check if your model is good.
* In other tutorials, it is said that you create one independent model on each iteration, and then you keep the one that gave you the best test results. But if this is the case, then why would I want to calculate the average of the accuracy scores, if I only care about the best one.

Hope you can help me, I am really having some trouble with all of this.

Jason Brownlee January 22, 2019 at 6:27 am #

REPLY ↩

We discard the models because CV is only used to estimate the performance of the model.

Once we have the estimate and we want to use the model, we fit a final model and start using it:
https://machinelearningmastery.com/train-final-machine-learning-model/

Iman February 28, 2019 at 12:17 pm #

REPLY ↩

I have question on selecting data when it comes to multiple linear regression in the form, y = B0 + B1X1 +B2X2
Say,
Y (response) = dataset 0 (i.e 3,4,5,6,7,8)
X1 (predictor)= dataset 1 (i.e 1,5,7,9,4,5)
X2(predictor) = datset 2 (i.e 7,4,6,-2,1,3)

Do you take all the data into account and divide into k groups,
Ie [3,4],[5,6],[7,8],[1,5],[7,9],[4,5],[7,4],[6,-2],[1,3]

Or just one dataset at time, such as,
Y and corresponding values x1
I.e [3,4] to [1,5] …..
Y and corresponding values x2

✕

Or is it some other way you select the data?

Thanks

Jason Brownlee February 28, 2019 at 2:33 pm # REPLY

Good question, you cannot use k-fold cross validation for time series.

Instead, you can use walk-forward validation, more here:
https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/

Anthony The Koala November 26, 2019 at 5:10 am # REPLY

Dear Dr Jason,

In a similar vein, can you use the 'simpler' train test and split for time series.

```
#For example sunspot data
from sklearn.model_selection import train_test_split
import pandas
df = pandas.read_csv('monthly-sunspots.csv')
X = [i for i in range(len(np.array(df.Month)))] # Convert month/year to integer
y = np.array(df.Sunspots) # total number of sunspots for a given year.
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_st
```

Thank you,

Anthony of Sydney

Jason Brownlee November 26, 2019 at 6:15 am # REPLY

That would be invalid as the train_test_split() would shuffle the examples.

See this:
https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/

Anthony The Koala November 26, 2019 at 6:50 pm #

Dear Dr Jason,

Thank you for that. It is appreciated.

Anthony of Sydney

No problem.

Vandana March 6, 2019 at 9:18 pm # REPLY

Your articles are the best. Every time I have a doubt machinelearningmastery solves it for me. Thanks a lot 🙂

Jason Brownlee March 7, 2019 at 6:48 am # REPLY

Thanks!

heldie March 7, 2019 at 7:51 pm # REPLY

Good explanation sir, ty 🙂 I have some clarity missing regarding the application of K-Fold CV for finidng – how many knots, where to place knots in case of piecewise polynomials / Regression Splines. Can u pls explain.

Jason Brownlee March 8, 2019 at 7:47 am # REPLY

Sorry, I don't have a tutorial on "regression splines".

heldie March 8, 2019 at 9:48 pm # REPLY

thx 4 d reply sir, in order to choose a best-fit degree of the polynomial, how K-Fold CV can be applied, pls explain Sir, thanks in adv 🙂

Jason Brownlee March 9, 2019 at 6:27 am # REPLY

I recommend a grid search over different model configurations, this is unrelated to k-fold cross validation, although CV could be used for each configuration tested.

Hi Jason, many thanks for the tutorial. It clarified many things for me, however, I am newbei in this fied. My question is how many times we can do a CV for a model?
For example is it reseanable to repeat 100 times 10-fold CV for our model?
I really appreciate any hint that can help me out.
Thanks!

Jason Brownlee March 22, 2019 at 8:44 am # REPLY

We repeat the CV process to account for the variance of the model itself, e.g. due to a stochastic learning algorithm like SGD.

Often a few repeats is sufficient, e.g 10, no more than 30.

Rahil March 22, 2019 at 7:23 pm # REPLY

Many Thanks for the reply Jason.
I am still confused.
when we are using 10-fold CV. It means that we partitioned our data randomely in 10 equal subsamples and then we keep one subsample for test and use others (9 subsamples) for train. So in this case only for 10 times we can get different results because there are just 10 different options to be kept for test and others to be used for train.
I mean after 10 times the way of arranging the data for train and test will be the same as one the previous states, right?! So, what is the advantage of repeating the process more than 10 times? Please help me out of this confusion. Thanks!

Jason Brownlee March 23, 2019 at 9:19 am # REPLY

Some algorithms will produce different results on the same dataset due to the stochastic nature of the learning algorithm. Stochastic gradient descent is an example.

This will introduce additional variance in the estimate of model performance that can be countered by repeating the evaluation more times.

Rahil March 23, 2019 at 6:17 pm #

Many thanks Jason!!

Hi Jason,
A quick question, if you decide to gather performance metrics from instances not used to train the model recurring to an evaluation scheme based on training-testing splits. Which fold-based evaluation scheme is more adequate? Why?

Jason Brownlee March 27, 2019 at 9:00 am # REPLY

If you are unsure what to use, the default should be 10 fold cross validation.

Federico March 28, 2019 at 2:52 am # REPLY

Why is that?

Jason Brownlee March 28, 2019 at 8:20 am # REPLY

It has proven effective as a default in terms of a balance between bias and variance of the estimated model performance.

This was established decades ago too, and has stood the test of time well.

itisha March 28, 2019 at 6:36 pm # REPLY

Hello sir,
i want to get the result of 10 fold cross validation on my training data in terms of accuracy score.
I performed grid search to find the hyperparameters of classifier and used cv value =10 in grid search function.i got the optimised parameters value and also the best score in terms of accuracy through grid search results.
a) is that accuracy (obtained by grid search) can be considered as the result of 10 fold cross validation?
b) if not, then should i use cross_val_score( ) to get the mean accuracy of 10 fold?
c) Also, while passing classifier in cross_val_score ( ) should i use optimised parameters of classifiers?

Jason Brownlee March 29, 2019 at 8:28 am # REPLY

You can report the score from CV if you want.

Itisha March 29, 2019 at 10:25 am #

REPLY

Ok thanks sir

Itisha March 29, 2019 at 10:35 am #

REPLY

I have. A query whuch is not relates to I told

Lets say classifier 1 is final classifier with optimized hyperparameters that m going to test on dataset A. Classifier 1 is trained on feature vectors of size 20.

Now I want to test on A again but this time with reduced features just to check impact of different features.

In this way I want to present the results on test set A with classifier trained on full feature set 20 nd same classifier trained on reduced feature set.

So should I use the same optimized hyperparameters with the classifier to be trained on reduced feature set?

Jason Brownlee March 29, 2019 at 2:02 pm #

REPLY

Good question.

I recommend varying one thing in a comparison, e.g. just the features and use the same data and model.

Alternately, you can vary one thing, the features, then use the same “process” of tuning each model for each subset of features.

Both are reasonable.

Itisha March 29, 2019 at 5:28 pm #

REPLY

Ok so if I go with first option…that means test data should be same nd classifier used for testing with original nd reduced features should be same with same optimized hyperparameters. ?

I have only one confusion:
Let’s say classifier is svm with c=10 (obtained by grid search on train data).
Now I ttrain svm with c=10 on entire taining set with feature vectors of size 20 andthen evalute it on test set T

Jason Brownlee March 30, 2019 at 6:24 am # REPLY

It is your choice, as long as you are consistent in methodology between the two things being compared.

Maria March 31, 2019 at 8:21 am # REPLY

For an imbalanced dataset with 0.7 positive class and 0.3 negative class. How do you do a cross-validation while preserving 50% positive and 50% negative samples in the train and test sets?

Jason Brownlee March 31, 2019 at 9:32 am # REPLY

Perhaps use stratified cross validation?

Keyvan February 12, 2021 at 4:33 am # REPLY

Hi Jason, I have the same problem.

I want to do model selection before testing the model, and my data is imbalanced. I first use stratified k-fold cross validation to make sure I have minority class in the test folds. Then, I perform model selection and choose a model with minimum cross validation error. The problem is that the test folds have already been used in model selection, so how can I test the model on new data as there is not test set?

Keyvan February 12, 2021 at 5:36 am # REPLY

Can I used nested cross validation for my problem as follows:

1. Use 3-fold CV
2. Perform hyperparameter tuning
3. Select the hyperparameters based on the minimum error on the validation folds
4. Tune the machine learning algorithms with the selected hyperparameters
5. Use stratified 10-fold CV
6. The out-of-fold predictions are treated as the test\unseen data

1. Use stratified train/test split
2. Use stratified 10-fold CV on train set
3. Tune hyperparameter
4. Train again on all train data with the selected hyperparameters
5. Evaluate the train models on the test set.

Thanks

Jason Brownlee February 12, 2021 at 5:54 am #

No need for step 5 as you already have evaluated the model on the hyperparameters.

Jason Brownlee February 12, 2021 at 5:52 am #

Step 2 and 4 are the same.

Perhaps you want to use nested cv: https://machinelearningmastery.com/nested-cross-validation-for-machine-learning-with-python/

Jason Brownlee February 12, 2021 at 5:51 am # REPLY ↩

The mean result from the stratified k-fold cv can be used to compare and select a model.

Perhaps I don't understand the problem?

AVIJIT PRASAD DAS April 4, 2019 at 8:05 am # REPLY ↩

really, its quite worthy

Jason Brownlee April 4, 2019 at 8:14 am # REPLY ↩

Thanks.

✕

Nice Tutorial!!! Enjoyed It !!
can you provide me the Matlab code for K-Fold Cross validation
Thank You

Jason Brownlee April 17, 2019 at 6:54 am #

REPLY ↩

I do not have any matlab code, sorry.

rolf May 27, 2019 at 11:30 pm #

REPLY ↩

I don’t really understand what you mean by

> Train/Test Split: Taken to one extreme, k may be set to 1 such that a single train/test split is created to evaluate the model.

… if k=1, then you are not dividing your data into parts: There is only one part.

Could you explain what you mean? Note also, that sklearn.model_selection.kfold does not accept k=1 as an input

Jason Brownlee May 28, 2019 at 8:15 am #

REPLY ↩

You are right, k=2 is the smallest we can do.

I have updated the post, thanks!

Sara June 11, 2019 at 6:50 am #

REPLY ↩

Does ‘scikit-learn train_test_split’ consider values of features and targets when shuffling and spliting the dataset?

Thank you

Jason Brownlee June 11, 2019 at 8:04 am #

REPLY ↩

Yes.

✕

Thank you Jason 🙂 I’m BIG fan of yours. Best!

Jason Brownlee July 4, 2019 at 2:50 pm # REPLY ↩

Thanks.

RAVI July 6, 2019 at 12:59 am # REPLY ↩

Jason sir, this K-fold CV tutorial is very helpful to me. Thank you so much !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Jason Brownlee July 6, 2019 at 8:40 am # REPLY ↩

You’re welcome, I’m glad it helped.

Quentin July 11, 2019 at 7:13 pm # REPLY ↩

Hi, thanks for this introduction,

I’m working on very small dataset (31 data) with 107 features. I have to apply features selection. For that I use XGBOOST and RFECV and other techniques.

I have one question :

Do I have to first split my dataset into 80% train and 20% test and apply an k-fold cross validation onto the train part and verify with the 20% remaining ? Or, do k-fold cross-validation without any split before ?

Jason Brownlee July 12, 2019 at 8:33 am # REPLY ↩

It might be a good idea to perform feature selection within each fold of the k-fold cross validation – e.g. test the procedure for selecting features rather than a specific set of features.

Quentin July 16, 2019 at 4:49 pm # REPLY ↩

Thanks, but if I want to show that a specific set of features remains the best, How can I do

I compare all the arrays of features selected in the n loop with the score (accuracy or F1)
And so on for the other techniques ?

Jason Brownlee July 17, 2019 at 8:16 am #

REPLY

Sounds like a reasonable approach.

Remember, we cannot know what is best, only gather evidence for what is good relative to other methods we test.

Shivani July 18, 2019 at 6:52 pm #

REPLY

I have been working on 10fold cross validation.In the predicted labels(Logistic Regression classifier),I am getting like this:
0.32460216486734716
-1.6753312636704334
1.811621906115853
0.19109397406265038
-2.11867198332618
-1.4679812760800461
0.02600304205260273
-2.0000670438930332
I dnt know how to tackle with negative and non binary values.Please help.

Jason Brownlee July 19, 2019 at 9:16 am #

REPLY

What is the problem exactly?

R.Aser August 5, 2019 at 6:29 pm #

REPLY

Hello,
I Have two questions:
1. I have a dataset, I used k=5 and 10 but some times I found there was a large difference in the R2, MAE and RMSE (i.e. for K=10, R=0.8 – MAE=3.5 – RMSE=6.5 , for K=5, R=0.62 – MAE=4.8 – RMSE=9.4) what is the reason of that difference? In other words, how to select the correct K which provide me reliable results? I know that there might a difference in using K=5 and 10 but m=not large one.

Thanks in advance,
R.Aser

Ramy August 6, 2019 at 9:02 am # REPLY

Hello Jason,
Do you need me to describe more to understand my point

Jason Brownlee August 6, 2019 at 2:04 pm # REPLY

Good questions.

Choosing a good K is hard. If in doubt, use 10. If you have the time, perhaps evaluate descriptive statistics of the data with different size K and find a point at which statistical significance tests report a difference in distribution – it is crude but might be a useful start.

Perhaps you can use stratified cross validation that focuses not only on the target, but on input variables as well?

I hope that helps.

Ponraj August 6, 2019 at 5:48 am # REPLY

Hello Jason,

I split this post as BACK GROUND & QUESTION Section.

BACK GROUND :
I am performing Binary Classification task using LSTM's. (either 0 or 1)
Data_size (205, 100, 4) [Out of 205 samples 110 belongs to class 0 & 95 belongs to Class1]

train_test_split : (train : 85 % & test : 15 % , random_seed = 7)
Fixed train data shape = (174,100,6)
Fixed test Data = (31,100,6)

Step 1: – MODEL TRAINING
I train the model (No random_seed weight intialization (like no numpy seed or tf seed))
1.1) Model Structure picture link : https://imgur.com/2IljyvE
1.2) Plot the Acc & Loss graph (both train & Validate)
– Picture Link : https://imgur.com/IduKcUp
– No Overfitting
1.3) Prediction result : using trained model : 3 out of 31 testing data were wrong

and trained the model 5 times to see behavior of the model based on your post (https://machinelearningmastery.com/diagnose-overfitting-underfitting-lstm-models/)

2.1) for i in range(5) : # run 5 times with same model structure
– Plot the Acc & Loss graph (Picture Link :https://imgur.com/WNH6m9F)
– RESULT : It follows a pattern (found behavior of the model)

Step 3: – K FOLD CROSS VALIDATION (CV)
Performed K fold CV (Fold – 7) (random seed = 7) (merged train + test data = original data (205,100,6))
3.1) Picture link : https://imgur.com/cZfR1wJ
3.2) Some folds results in Over fitting
3.3) Every fold the acc value calculated and mean acc value is 79.46 % (+/- 5.60 %)
(I followed your post : https://machinelearningmastery.com/evaluate-performance-machine-learning-algorithms-python-using-resampling/)

QUESTIONS ONLY ABOUT CROSS VALIDATION :
1. On cross validation results, more number of Over fitted model/graphs found,

a) What can I understood from CV results ? improper hyper parameters ?
b) Std. deviation of +/- 6% is huge or it is normal ?
c)How can I relate my trained model result (Step:1) with CV results (Step: 3) ? I understand how it works but can I use initial trained model as a final model since my prediction is 90 % correct ?
d) I reduced LSTM units size and performed K fold CV again.
Picture link : https://imgur.com/UsU3zso (Less Overfit models)
Mean Acc & Std : 79% +/- 3.91
Based on Std dev, whether i should fix with this hyper parameter in model ?
e) My friend suggested me to go for LOOCV, but will that make any difference ?

Jason Brownlee August 6, 2019 at 6:45 am #

REPLY

Way too much going on there, sorry, I cannot follow or invest the time to figure it out.

Are you able to boil your problem down to one brief question?

Ponraj August 6, 2019 at 7:36 pm #

REPLY

I trained my LSTM binary classification model and gets prediction accuracy of 90 %. No over fitting occurs. (https://imgur.com/IduKcUp)

But when I do K fold CV (K = 7), I can found over fitting models in those 7 folds.
What can i understood from over fitting in CV models ? (https://imgur.com/cZfR1wJ)

On CV results, i get the mean accuracy of 79.5 % & Std. deviation of +/- 6%.

(https://imgur.com/UsU3zso – Less Overfit models)
Since my Std dev is low compared to previous model, whether i should fix with this hyper parameter in model ?

My friend suggested me to go for LOOCV, but will that make any difference instead K fold CV ?

Jason Brownlee August 7, 2019 at 7:49 am # REPLY ↰

In practice, k-fold cross validation is a bad idea for sequence data/LSTMs, instead, you must use walk-forward validation:
https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/

Perhaps the datasets used in k-fold cross validation are smaller and less representative and in turn result in overfitting?

Model performance is always relative:
https://machinelearningmastery.com/faq/single-faq/how-to-know-if-a-model-has-good-performance

LOOCV sounds like a good idea if you have the resources.

Ponraj August 8, 2019 at 9:05 pm #

thanks for your reply.
I understood your post related to walk-forward validation.But I am confused, whether it can be applied for my Data set. (Since I am performing classification)

Overview about my Dataset : X.shape= (205,100,4) and Y.shape = (205,)

In X, each sample/sequence is of shape (100, 4), whereas each row in 100 rows corresponds to 100 milli sec.(10 sec for 1 sample)
Out of 210 samples, 110 samples belongs to class 0 & 95 Samples belongs to class 1.

Model Structure : https://imgur.com/2IljyvE
Model : https://imgur.com/tdfxf3l
Note : Used TimeDistributed Wrapper around Dense layer so that my model gets trained for each 100 ms corresponds to respective class for every sample/sequence.

My aim is to predict early the Class, If i input, test data of shape (10,60,4) –
(10 samples, 60 (6 seconds), 4 features) whether it belongs to class 0 or 1.

In that case, how can I approach Walk forward validation

✕

Yes, this would be a time series classification task which can be evaluated with walk forward validation.

I give examples of time series classification here that you can use as a starting point:
https://machinelearningmastery.com/start-here/#deep_learning_time_series

Marshal August 9, 2019 at 3:39 am # REPLY ↩

Good day Jason,

Thank you for all of your tutorials, they are very clear and helpful.

Which method for calculating R2 for the evaluation of the test set is appropriate?

I ask because it seems that the caret package in R defaults to R2 = cor(obs, pred)^2, but I thought 1 – sum((obs – pred)^2) / sum((obs – mean)^2) was most appropriate. Both methods give the same result on the full data set, but I am getting different results when I use them on the test sets (higher R2 for cor()^2).

I'm using the caret package to cross validate a predictive linear model that I have built. I'm using train function with trainControl method = repeatedcv and the summary default of RMSE and Rsquared. I get high R2 when I cross validate using caret, but a lower value when I manually create folds and test them.

Any insight or direction would be greatly appreciate.

Thank you

Jason Brownlee August 9, 2019 at 8:18 am # REPLY ↩

Perhaps this will help:
https://en.wikipedia.org/wiki/Coefficient_of_determination

SHAIKH MOHD FARAZ August 11, 2019 at 5:01 pm # REPLY ↩

Hii Jason

Very nice and clear tutorial on K-fold validation.

I have one doubt. Let's say we are implementing a K-fold cv on K'-NN algorithm.
Since we will be using the cv dataset to determine the best value of K' and then use test dataset to determine the accuracy of the model, How do you think we should split our dataset? Can you please explain with an example.

✕

A good default is 10 folds.

Abishek Balaji September 7, 2019 at 8:05 pm # REPLY

Hey Jason, It's a great tutorial, but I have just one question what do you exactly mean by the following statement in this article.

"It is also important that any preparation of the data prior to fitting the model occur on the CV-assigned training dataset within the loop rather than on the broader data set. This also applies to any tuning of hyperparameters."

Jason Brownlee September 8, 2019 at 5:17 am # REPLY

It means that you must be careful not to use information from the whole dataset to prepare the data or tune the hyperparameters.

It suggests only using the training dataset from each fit/fold to figure out how to prepare the train/test sets and tune the model. This is to avoid data leakage:
https://machinelearningmastery.com/data-leakage-machine-learning/

Ralph September 11, 2019 at 11:24 pm # REPLY

Hi, and thanks for this clear post for a practical implementation of k-fold validation. Btw thanks for your answers on other posts 😉

I have a general question regarding this topic: it seems that all existing method take continuous portions of the training and test set, instead of mixing both.

To be more clear on an example, assume we have 1000 samples, and we split in 0:799 for the training set, and 800:999 for the test set.

Wouldn't it be better to mix the indexes? For instance [0,5,10,..,995] for the test set and all other indexes for the training set. In the case for instance of chronological data, it makes more sense as no sample is biased towards a particular time.

Jason Brownlee September 12, 2019 at 5:17 am # REPLY

Great question!

Meriem September 19, 2019 at 11:42 pm # REPLY

Hi,
How can I get the Accuracy of each model (1,2,3) after CV?

Jason Brownlee September 20, 2019 at 5:45 am # REPLY

You can iterate each fold and for each fold fit a model no the train set and make predictions on the test set and then calculate a score for the predictions – then print that score.

krishna November 27, 2019 at 5:02 am # REPLY

Sir could you plz explain a working example on SVM classifier?
thanks

Jason Brownlee November 27, 2019 at 6:13 am # REPLY

Sure, see this:
https://machinelearningmastery.com/support-vector-machines-for-machine-learning/

A November 28, 2019 at 11:17 pm # REPLY

Hi Jason,

I have growth, climate data sets of crop and i want to do ML prediction model to predict yield. I want to use Regression because i want to know the value and not classification.
Here I ask you how can I make label for Yield? any link, tips?

After doing labelling which step do I need to follow to do the Regression model? any link, tips would really help.

Regards
Amora

Jason Brownlee REPLY

I recommend using this tutorial as a template:
https://machinelearningmastery.com/spot-check-regression-machine-learning-algorithms-python-scikit-learn/

Anthony The Koala December 2, 2019 at 3:08 am #

REPLY

Dear Dr Jason,
I had a go with a larger data set of size 100, with 10 folds.
Adapting from your above example:

```
from numpy import array
from sklearn.model_selection import KFold
data = array([[(i+1) for i in range(100)]]) # my data set of 100 items.
kfold = KFold(10,True,1); #10 folds.
for train, test, in kfold.split(data):
 print('train %s, test %s, len(train) %s, len(test) %s' % (data[train],data[test],len(tra

#Some of the output, the first split of 10.
train [  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  19
  20  21  22  23  24  25  26  27  28  29  30  31  32  33  35  36  38  39
  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57
  58  59  60  61  62  63  64  65  67  68  69  71  72  73  74  75  76  77
  78  79  80  84  86  87  88  89  90  91  92  93  95  96  97  98  99 100],
test [18 34 37 66 70 81 82 83 85 94],
len(train) 90, len(test) 10
....
#there are nine more outputs (not printed).
....
```

In sum, the original data size was 100.
There are 10 folds with the 10 elements in each test array.

My question:
* How can I vary the length of the train and test. For example I would like 10 test folds, but the train length is 0.66666, and test length = 0.3333

Thank you,
Anthony of Sydney

Jason Brownlee December 2, 2019 at 6:06 am #

REPLY

It does not work that way.

k-fold CV requires you divide your dataset into k equal or mostly equally sized parts.

sized parts…..” means:

* applying the primary school maths that you find the number of folds must be a factor of the number of datapoints. So if you had 63 datapoints, the number of folds must be 3, 7, 9, 21. Similarly if you had 100 datapoints, the number of folds must be 2, 4, 5, 10, 20, 25, 50.

*What about prime number of datapoints of which to divide into folds? Eg 71 data points.

* accordingly, the number of test points is then = no. of datapoints/no. of folds.

* it follows that the number of training points = total number of datapoints – no of test points.

Thank you
Anthony of Sydney

Anthony The Koala December 2, 2019 at 8:45 am #

REPLY

Dear Dr Jason,
I did an experiment with prime and non prime numbers and it appears that if a number does not factor into the number of datapoints, then the number of test points are.

```
no of test points = floor(no of datapoints/no of folds)
```

The code to replicate is adapted from the above demo code:

```
def donkey(thedata,thefold):
 from numpy import round
 from numpy import floor
 for train, test in thefold.split(thedata):
 itfactor = len(thedata)/thefold.n_splits
 print("train = %s, test = %s, len(train) = %s, len(test) %s, len(data)/no. spli
```

Conclusion
I did a few experiments with the number of datapoints being prime and non-prime where the number of test data points is:

```
no of test data points = np.floor(len(data)/no.of folds)
```

If follows that the number of train data points is:

```
no of train data = len(data) - no of test data points
```

Thank you,
Anthony of Sydney

Jason Brownlee December 2, 2019 at 1:54 pm #

REPLY

Yes, it does not have to be perfectly even, just as even as possible, i.e. one fold might

Kelvin December 11, 2019 at 12:33 am #

REPLY ↩

Hi Jason,

If no model selection nor hyperparameter tuning that needs to be done. Does that mean it is not necessary to apply cross-validation?

Thanks in advance.

Regards,
Kelvin

Jason Brownlee December 11, 2019 at 7:00 am #

REPLY ↩

Yes, to estimate how performance changes with the data, e.g. the model variance.

Mike Kelly December 14, 2019 at 2:39 am #

REPLY ↩

For binary classifier models when we want the class to be balanced during training, should we maintain separate KFold() objects for each label in the class to ensure that each fold is balanced or is it enough to balance the dataset as a whole and let the folds be randomly sampled?

Jason Brownlee December 14, 2019 at 6:22 am #

REPLY ↩

No, use a stratified version of k-fold cross validation.

For example:
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

Yan January 8, 2020 at 6:25 am #

REPLY ↩

Hi, Jason,

Nice introduction! I've been using the k-fold for a long time, even in scientific publications, but I still don't feel I have a good understanding about testing its statistical significance.

First, how many times should we shuffle and do the whole things, i.e., how many "repetitions" are enough? Let's say we have 100 samples. Would 50 or 200 repetitions be enough for a 10-fold CV?

am never sure if I used the correct n here, which I set as the number of samples (i.e., 100), not the number of repetitions.

I see a lot of "comparing two k-fold models" online, but not the test of a single model alone.

Thank you so much!

Jason Brownlee January 8, 2020 at 8:35 am #

REPLY ↩

There are no good answers. Enough to capture the variance/improve the estimate of the population mean.

I like 3 to 10 repetitions, 30 to 100 if I have the time (I rarely do).

Re statistical tests for cross-validation and comparing algorithms, see this:
https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/

Priyash February 1, 2020 at 10:54 am #

REPLY ↩

Hi Jason,

" It is also important that any preparation of the data prior to fitting the model occur on the CV-assigned training dataset within the loop rather than on the broader data set. This also applies to any tuning of hyperparameters. A failure to perform these operations within the loop may result in data leakage and an optimistic estimate of the model skill. "

This particular line says that any data preparation, let's say data cleansing, feature engineering and other tasks should not be done before the cross-validation and instead be done inside the cross-validation.

Can you take some time to explain that.

Jason Brownlee February 2, 2020 at 6:20 am #

REPLY ↩

Yes.

Consider scaling. If you scale using all data, it uses knowledge of the min/max or mean/stdev of data in the test set not seen during training.

You can discover more on the topic here:
https://machinelearningmastery.com/data-leakage-machine-learning/

✕

what is the utility of using KFold and StratifiedKFold?

Jason Brownlee February 10, 2020 at 1:20 pm # REPLY

Faster, simpler, appropriate for regression instead of regression.

Marc February 12, 2020 at 3:44 am # REPLY

Hi Jason,

Thank you for all of your tutorials, they are very clear and helpful.

Is that more meanfull to split first all data in training and test set, for after processing a CV on only training data?
What is the best approch ? CV on all data or just training data ?

Jason Brownlee February 12, 2020 at 5:51 am # REPLY

You’re welcome.

There is no best approach, you need to find an approach that makes sense for your project.

I think this will help:
https://machinelearningmastery.com/difference-test-validation-datasets/

Marc February 13, 2020 at 9:34 pm # REPLY

Thanks from France 😉

Jason Brownlee February 14, 2020 at 6:34 am # REPLY

You’re welcome from Australia!

Kollol February 24, 2020 at 2:45 am # REPLY

I have a query.How can we do cross validation in case of multi label classification?

Thanks

Jason Brownlee February 24, 2020 at 7:43 am # REPLY ↩

Sure.

Yong March 1, 2020 at 1:15 pm # REPLY ↩

if the dataset is unbalanced , what is the procedure during the use of 10 fold cross validation?

Jason Brownlee March 2, 2020 at 6:14 am # REPLY ↩

Use stratified cross-validation.

If you are using data sampling on the training set, use it within each fold of the CV via a pipeline.

lopamudra das March 31, 2020 at 1:45 pm # REPLY ↩

Hi, can 10 fold cross-validation be applicable to DNA sequence data for cancer analysis?

Jason Brownlee April 1, 2020 at 5:44 am # REPLY ↩

Probably.

Joyce April 20, 2020 at 6:32 am # REPLY ↩

Can I ask why is the standard deviation is an important factor when it comes to evaluate k-fold cross validation?

Jason Brownlee April 20, 2020 at 7:36 am # REPLY ↩

✕

Tarik April 21, 2020 at 12:33 am # REPLY

Please, I have a question regarding Cross-validation and GridSearchCV. (This question is already asked by itisha March 28, 2019 at 6:36 pm, but i did not anderstand your answer)
I have a small dataset, and i can not devide it on test/validation/traing sets. I decided to make a coross-validation to estimate the performance of model based on SVM classifier. My question is what are the hyper-parameters to use during this Corss-validation ? Can I execute GridSearchCV and report the results of the best corss-validation perormance (CV with best hyper-parameters) as the final Cross-validation results.

Jason Brownlee April 21, 2020 at 5:58 am # REPLY

Yes, you can use grid search within the cross-validation, this is called nested cross validation and allows you to evaluate a tuned version of your model.

Tarik April 21, 2020 at 7:10 am # REPLY

Thank you very much for your answer, I understand now.

Jason Brownlee April 21, 2020 at 7:44 am # REPLY

You’re welcome.

Anand May 9, 2020 at 7:25 pm # REPLY

Thank you for this article! Its amazing, yet 1 question stil remains in mind and want to clear my confusion.
If i use 10- fold CV on training dataset, then that training dataset is divided into 10 sets , so now i have 10 iterations for training model on 9-fold of data and test on 1fold data in every iteration right? Apart from this we have test data which we splitted before training the model to test on right!

If i am right in above querry then , if we apply k-fold on entire dataset would that benefit us more or less, just a question!

Thank You!

You're welcome!

No, typically we would use cross-validation or a train-test split. Not both. Yes, cross-validation is used on the entire dataset, if the dataset is modest/small in size.

If we have a ton of data, we might first split into train/test, then use CV on the train set, and either tune the chosen model or perform a final validation on the test set.

K S May 23, 2020 at 7:24 pm # REPLY ↰

Just one clarification – In cross validation, as given one data set (train or test) is divided into 10 folds (as example). Then 9 folds are used to train and 1 fold to test which is part of data set given earlier. And, this process repeats where each of these 10 folds become part of test once. So, with this above understanding, only one data set is used which is given as input to K Fold and not both. Please clarify as in above answer it specifies that both sets (train and test) are used.

Jason Brownlee May 24, 2020 at 6:06 am # REPLY ↰

Correct.

Yulia December 21, 2021 at 6:20 pm # REPLY ↰

Dear Jason, thaks a lot for your tutorials!
Could you please clarify the confusion.

https://machinelearningmastery.com/training-validation-test-split-and-cross-validation-done-right/
This arcticle tells the approach: first split into train/test, then CV on train set and choose model, afterwards train model on whole train set, finally evaluate model on test set.

From comments here it looks like this approach is only for "ton of data" case? And if "ton of data" is not the case, then: CV on whole dataset, choose model and this is it ?

James Carmichael December 24, 2021 at 5:55 am # REPLY ↰

Hi Yulia…The following resource will hopefully help clarify best practices with training, testing and validation.

https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-

✕

Vidhan May 19, 2020 at 1:58 pm # REPLY

Hii Jason,

It was very good article. I have a doubt about k-fold cross validation. Please help me out. I am confused over usage of k-fold cross validation. Is it used to compare two different algorithmic models like SVM and Random forest or is it used for comparison between same algorithm with different hyperparameters ?

Jason Brownlee May 20, 2020 at 6:18 am # REPLY

Thanks!

Both.

K S May 23, 2020 at 7:13 pm # REPLY

It is extremely useful article and one of best article I have read on cross validation. I have doubt on how cross validation actually works and need your help to clarify. In 10-fold cross-validation where we need to select between 3 different values of one parameter (please note parameter is one but possible values are 3 from which we need to select) then how does this 10 fold cross validation works in this case …how many models are trained and evaluated? Will it be 10 models or 3 models or 30 models?

And, second part of doubt is that will these above models be trained and as well as evaluated on portion of training set only? Right?

Requesting you to help clarity both parts.

Jason Brownlee May 24, 2020 at 6:05 am # REPLY

Thanks.

In this case, we us CV on each config and compare the mean of each run. Yes, 3 * 10-fold cv is 30 models. All of which are discarded at the end.

Each model is trained on the training folds and tested on the test folds as the folds are enumerated.

K S May 24, 2020 at 3:31 pm # REPLY

not 10. I had understanding that in each iteration of 10 fold cross validation, we will build model using 9 folds of data and then validate this model on 10th fold (the fold of data which is not used in training for that iteration) for all 3 possible cp values. So, model created in each iteration will be one but tested 3 times for each possible cp value (0.1, 0.2 and 0.3). Or will it be 3 models each iteration and hence resulting 30 models in total for 10 fold cross validation. Requesting you to help clarify.

So to "best describes how 10-fold cross-validation works when selecting between 3 different values (i.e. 0.1, 0.2 or 0.3) of cp parameter?" using below statement

"X" models are created on subset of training set and evaluated on "Y"

What will be values of X and Y?

What will be X – 10 or 30
What will be Y – "test set" or "portion of training set"

Assumption here is that before cross validation, data is split into train and test data and cross validation is done on training set. So, considering this assumption please help that what will be value of Y in above statement.

Jason Brownlee May 25, 2020 at 5:44 am # REPLY

Each configuration is evaluated (3) and the evaluation of each configuration uses cross-validation (10). The evaluation of each configuration is a separate process.

K S May 25, 2020 at 5:59 am #

Thanks but can you please help in clarifying ..the question which is asked in my course is that given this scenario (as explained above), how many models it will be generated? And, where this model be evaluated – will these be evaluated on portion of training set or testing set as part of cross validation? The problem statement also confirms that testing set is carved out separately before initiating cross validation and Cross validation is run on training set. So, considering this please help in clarifying. .

Jason Brownlee May 25, 2020 at 1:22 pm #

Perhaps talk to your teacher directly about your homework. You've paid them after all…

"30 models are created on subset of training set and evaluated on portion of training set" Is this correct understanding or will it be "30 models are created on subset of training set and evaluated on testing set" where testing set is separate set carved out before cross validation starts and cross validation is done on training set,

Cuong May 28, 2020 at 12:23 am # REPLY

Hi Jason Brownlee,
I split my data into 80% for training and 20% for testing (unseen data). And, I use trainning data to train, and compare machine learning models and use K-fold CV through training model. Finally, I use selected model to check the accuracy on the testing data (unseen data, 20% of data).
Could you please explain if I have done right or wrong?

Thank you.

X. C Nguyen

Jason Brownlee May 28, 2020 at 6:16 am # REPLY

It is not about right and wrong, instead, you have chosen a different approach.

If it works for you, go for it.

Rouzbeh Talebi June 5, 2020 at 7:52 am # REPLY

Hello Jason,
I am a little confused.
I split my data into training and testing datasets. Is it possible to train a model by cross-validation and then apply the model for testing data?
All I saw on the internet was for the whole dataset.
example:
cross_val_score (model, X, y, cv=4, scoring="neg_mean_squared_error")
or
cross_val_predict (model, X, y, cv=4, scoring="neg_mean_squared_error")

But I want to make a model from the training dataset and then apply for the test dataset. I do not know how to code in python:
All I want is:
CV = ?(model, X_train, y_train, cv=4, scoring="neg_mean_squared_error")
then

Is it possible for this? If yes, what should I write instead of "?"? Or is there any way to reach my goal?
It would be much appreciated if you help me out.

Jason Brownlee June 5, 2020 at 8:30 am # REPLY

Both approaches evaluate the model when making predictions on unseen data.

Once we have the estimates, we discard the models, fit the model on all available data and start using it to make predictions.

Rouzbeh June 5, 2020 at 8:37 am # REPLY

I did not get exactly.
I want to test the model on a particular dataset.
Is there any example on the website that guides me

Jason Brownlee June 5, 2020 at 1:40 pm # REPLY

Yes many, perhaps start here:
https://machinelearningmastery.com/evaluate-performance-machine-learning-algorithms-python-using-resampling/

Nilarun Mukherjee June 20, 2020 at 6:18 pm # REPLY

I would like to know two thing:
1. From fold to fold are weights are preserved (updates in previous fold) or weights are initialized randomly in each fold?
2. If I want to save the best model of certain fold what to do?

Jason Brownlee June 21, 2020 at 6:20 am # REPLY

Each fold we train an entirely new model and at the end of the fold we discard the model.

No need to save the best model as we are only estimating the performance of the modeling pipeline. Once we know how well it performs, we can compare it to other models/pipelines, choose one, then fit

RAKESH KUMAR July 6, 2020 at 5:11 am # REPLY ↩

As There are 7 empirical performance measurement models, can k-fold CV be applied for selection of optimal performance measurement model. If yes, then how?

Jason Brownlee July 6, 2020 at 6:40 am # REPLY ↩

We cannot know the optimal model or how to select it for a given predictive modeling problem.

The best we can do is to use robust methods and try to discover the best performing model in a reliable way given the time we have.

RAKESH KUMAR July 13, 2020 at 7:07 am # REPLY ↩

Respected Sir, I like to know that if we have three performance measurement models like- Balance Scorecard, Key Performance Indicators (KPI) model and Capability Maturity Model (CMM) , so can k-fold CV be used for selection among these models? If yes, then How? Plz guide me in this regard.

Jason Brownlee July 13, 2020 at 1:34 pm # REPLY ↩

I recommend selecting one metric and using that to select a model.

RAKESH KUMAR July 31, 2020 at 5:17 am # REPLY ↩

respected sir,

plz, brief about it, that how i proceed

Adnan Bin Amanat Ali July 14, 2020 at 4:12 pm # REPLY ↩

Can stratified k fold cross validation be helpful in dealing imbalance data?

REPLY ↩

Yes, it is required:
https://machinelearningmastery.com/cross-validation-for-imbalanced-classification/

Adnan Bin Amanat Ali July 15, 2020 at 12:07 pm # REPLY

Thanks a lot.

Jason Brownlee July 15, 2020 at 1:59 pm # REPLY

You're welcome.

gio July 21, 2020 at 1:57 am # REPLY

Hey very interesting article.

I ‘d like to ask if you think that k-fold cross validation can be used for AB testing.

Lets say I have an 80/20 AB test, could I split the 80 on 4 random 20s and then form 5th dataset as the average of those 4 datasets and compare my variant with it?

Is there something wrong with this approach?

Thank you.

Jason Brownlee July 21, 2020 at 6:07 am # REPLY

No, they are different methods for different problems.

cv estimates model skill when making predictions on data not seen during training.

a/b tests estimate a binomial or multinomial probability distributions via sampling.

Gamze July 27, 2020 at 6:30 am # REPLY

Dear Jason,

I made a manual 5 fold cross-validation because my methodology is different. Thus, I have individual R square values for each fold. I just wanted to ask can I take the average of R squared values from each fold

Jason Brownlee July 27, 2020 at 1:02 pm # REPLY

Yes.

Gamze July 27, 2020 at 1:13 pm # REPLY

Thank you so much for your reply.

But, I could not explain this to myself. I have searched this and there is a lot of confusion. It is said that the overall R2 or RMSE is not equal to the average of the folds results.

Jason Brownlee July 28, 2020 at 6:37 am # REPLY

Not sure why that would be the case.

sajad July 28, 2020 at 7:20 am # REPLY

Hi,

Thanks for your article.

I have a Lidar bathymetry data set in the shallow water. I would like to use the Cross-validation for my model.

Please guide me in that step by step.

Thanks.

Jason Brownlee July 28, 2020 at 8:35 am # REPLY

Perhaps you can use the code in tutorial as a starting point and adapt it for your data.

Josseline Perdomo July 28, 2020 at 5:20 pm # REPLY

Hello Jason! First of all, thanks for this explanation, it was very helpful, especially for the new people in the subject, like me :).

and validation and regular hold out to get the test split. Could you please give me any advice about best practices when in a paper use this KFolds CV approach?

Thanks!

Jason Brownlee July 29, 2020 at 5:48 am #

REPLY ↩

It is a good practice to use 10 splits and report the mean and standard deviation of your performance metric calculated on each test set.

Balaji Sundararaman July 31, 2020 at 10:32 pm #

REPLY ↩

Hi Jason,
Thanks for the tutorial. When I try out the code in your tutorial, I used the below code :

```
data = [0.1,0.2,0.3,0.4,0.5,0.6]
kfold = KFold(n_splits=3, shuffle= True, random_state= 1)
```

```
for trn_idx, tst_idx in kfold.split(data):
print('Training Index : {}, Test Index : {}'.format(trn_idx,tst_idx))
```

Now how do I use trn_idx and tst.idx to split the original data?

When I try :

```
train_data = data[trn_idx]
test_data = data[tst_idx]
```

I get the below error:

```
——————————————————————————
TypeError Traceback (most recent call last)
in
—-> 1 train_data = data[trn_idx]
2 test_data = data[tst_idx]
```

```
TypeError: only integer scalar arrays can be converted to a scalar index
```

Jason Brownlee August 1, 2020 at 6:10 am #

REPLY ↩

Sorry to hear that, this may help:
https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me

Jason – You've posted a range of well written, easily digestible articles in the ML arena which I have found quite useful. Thank you for excellent work…

– Eric

Jason Brownlee August 8, 2020 at 6:02 am # REPLY

Thanks Eric!

Terrell August 9, 2020 at 12:59 pm # REPLY

What's up, after reading this awesome paragraph i am also glad to share my know-how here with friends.

Jason Brownlee August 10, 2020 at 5:45 am # REPLY

Thanks.

Pedro August 10, 2020 at 1:38 pm # REPLY

Thank you for your content.

Does k-fold cross validation in conjunction with GridSearchCV replace the traditional model.fit() when training a model? And how a proper GridSearchCV should be performed? I mean, if I want to perform a GridSearch of the batch_size+neurons+learning_rate+dropout_rates, should I mix all those together at same time?

Jason Brownlee August 11, 2020 at 6:26 am # REPLY

Cross-validation is only used to estimate the performance of the model.

You can use a "grid search" as the model, in which case it will find the best config for you automatically.

Yes, you can tune multiple hyperparameters at once, but it can be very slow.

Suppose I'm evaluating my results based on accuracy(need of the client). After comparing my CV accuracy and training set accuracy I find that my model is overfitting. I performed Randomsearch CV and obtained the best hyperparameters. Using these best hyperparameters the training accuracy decreases but the new CV accuracy improves very little(2%). My question is have I been able to solve the problem of overfitting? Another question is that the best hyperparameters that I'm choosing are choosen using the process of CV(Randomsearch CV). Are these effective when I'm using them on the trainnig data ?

Jason Brownlee August 14, 2020 at 6:10 am # REPLY

You can overcome overfitting in this case by using a robust test harness and choosing the best model based on average out of sample predictive skill.

Don't choose a model or model hyperparameters based on skill on the training dataset, it is not the goal of the project.

MS August 14, 2020 at 7:00 pm #

Then on what basis should I choose a model or model hyperparametrs? The learner for which the hyperparameter is tunned what should be it's evaluation criteria?

Jason Brownlee August 15, 2020 at 6:20 am #

k-fold cross-validation allows you to estimate model/config performance when used to make a prediction on new data.

Choose a model based on mean skill from k-fold cross-validation, ideally repeated+stratified k-fold cross-validation for classification, repeated k-fold cross-validation for regression, nested/double k-fold cross-validation for hyperparameter tuning.

MS August 14, 2020 at 7:05 pm #

I cannot use the test set as I'm still unsure whether my learner has combat the problem of overfitting.

Combatting overfitting is only a practical issue for algorithms that learn incrementally, like neural networks and boosting ensembles.

MS August 16, 2020 at 12:34 am #

thank you Jason

Jason Brownlee August 16, 2020 at 5:53 am #

No problem.

Sakorpio August 15, 2020 at 5:56 am #

REPLY

Let say if i have 1000 images in my dataset and my train test split is 80/10 and i choose k=10 how it will perform 10 folds ?
will it repeat its data in folds ?

Jason Brownlee August 15, 2020 at 6:37 am #

REPLY

You use train/test OR cross-validation, not both.

Data is not repeated in folds.

toufik August 25, 2020 at 7:22 pm #

REPLY

thank you Jason, for this article, it's possible with a dataset to iterate 100 iteration with k-fold= 5

Jason Brownlee August 26, 2020 at 6:49 am #

REPLY

What do you mean 100 iterations?

I used the accuracy scores from some sample results from KFold.
For example, I use n_split = 5, then use each sample to find out the predicted value and calculate its accuracy.

From this accuracy value I get one less good sample. What should I do with this sample data?

Jason Brownlee October 30, 2020 at 6:52 am # REPLY ↩

Sorry, I don't understand. Perhaps you can rephrase your question?

Sakura October 30, 2020 at 4:42 pm # REPLY ↩

Hi, I'm not sure if this is the best page to ask, but if I have an n-example set and a k-fold, how many classifiers are we training?

Jason Brownlee October 31, 2020 at 6:46 am # REPLY ↩

It will train k classifiers.

After we estimate the performance of the model from the mean of the results, all classifiers are discarded.

Matthew October 31, 2020 at 5:43 am # REPLY ↩

Thanks for the nice tutorial, Jason. I have one question.

When performing cross validation, is it important to separate a portion of your dataset as your test dataset to evaluate the performance of your model later, or is it sufficient to have just the results as the mean and variance of the model skill scores obtained during the cross-validation process?

Once again, thanks for the article.

Jason Brownlee October 31, 2020 at 6:51 am # REPLY ↩

Using CV alone is often sufficient.

✕

Sir, I have 1000 data, I split it into 80% (training dataset) 20% (test dataset). Then I will use the training dataset to perform 10-fold validations where internally it will further split the training dataset into 10% (8% from 1000) validation data and 90% (72% from 1000) training data and rotate each fold and based on generated accuracy I will select my model (model selection). Once model is selected, I will test it with the held-out 20% test data (20% from 1000).

Is the approach correct…

Jason Brownlee November 8, 2020 at 6:39 am # REPLY ↰

You can evaluate models anyway you like, as long as you trust the results.

Jay November 18, 2020 at 8:17 am # REPLY ↰

Hey, I came across many websites where they mention k=n has high variance when compared with k=10 or for any other value of k, could you give an explanation for that?

Jason Brownlee November 18, 2020 at 1:07 pm # REPLY ↰

I would expect low variance, see this:
https://machinelearningmastery.com/loocv-for-evaluating-machine-learning-algorithms/

Jay November 18, 2020 at 10:22 pm # REPLY ↰

https://machinelearningmastery.com/how-to-configure-k-fold-cross-validation/
From the error plot in this tutorial the variance seems to be increasing as we increase the number of folds. We know that lower k values would give a high bias, so high k values would give a lower bias. If the higher k values give both low bias and variance, then what is stopping us from using LOOCV other than the fact that it is computationally expensive.

Jason Brownlee November 19, 2020 at 7:44 am # REPLY ↰

Maybe, maybe not. In that post we are plotting the min/max not the IQR or stdev.

The relationship might not be so linear.

✕

Hi, thanks for this tutorial. I know I'm late to the party, but I'm struggling to understand the scores that cross-validation gives me. For a certain model run on my dataset, I get the following scores: [0.93625769, 0.89561599, 1.07315671, 0.69597903, 0.62485697, 1.67434609, 1.3102791, 1.42337181, 0.80694321, 1.15642967]

Mean score with depth = 2: 1.0597236282939932
Mean absolute error: 0.4903895091309324

I just want to know, how do I know if this is a good score or not?

Thanks again.

Jason Brownlee November 24, 2020 at 7:47 am # REPLY ↩

You're welcome.

Great question! A good score is relative to the score achieved by a naive model, more here:
https://machinelearningmastery.com/faq/single-faq/how-to-know-if-a-model-has-good-performance

Ahmet SOLAK January 28, 2021 at 7:06 pm # REPLY ↩

Hi Jason,
First of all thank you for this post. I have a question.
When model finish training, is it test images with last model (e.g. 1000 epochs training and model for 1000. epoch) or is it test with best model (e.g. train with 1000 epochs but best model at 978. epoch)?

Jason Brownlee January 29, 2021 at 6:01 am # REPLY ↩

You're welcome.

No the model is fit (all epochs) on the train folds and test on the hold out folds, and repeated allowing each fold to be used as the held out fold.

Ahmet SOLAK January 30, 2021 at 11:51 pm # REPLY ↩

I think you misunderstood me.
Let's assume I train model with 5-fold cross validation and model trained 1000 epochs for each fold. For each fold save best model (for example; according to minimum loss) and best model at

✕

Jason Brownlee January 31, 2021 at 5:35 am # REPLY

if you are using early stopping with cv, then yes, performance on each fold will be calculated whenever early stopping stopped training or saved the best model.

Ahmet SOLAK January 30, 2021 at 11:55 pm # REPLY

I have an extra question. When I made 10-k cross validation , it gets interrupted at the 6th or 7th fold. When I change epochs or batch size, it continues. Is this due to lack of resources (GPU, CPU, etc.) or something else?

Jason Brownlee January 31, 2021 at 5:35 am # REPLY

This sounds specific to your machine. Perhaps you can debug the cause.

Shahbaz January 31, 2021 at 12:20 am # REPLY

what is mean of this … plz explain
Importantly, each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set 1 time and used to train the model k-1 times.

Jason Brownlee January 31, 2021 at 5:37 am # REPLY

It means that each row of data belongs to one fold. That the algorithm operates on folds of data rather than rows of data.

Kimia February 17, 2021 at 3:59 am # REPLY

Hi Jason,

Thanks for this! I have one question for you. Is that ok to do the k-fold cross validation on the same dataset that we used to train the model?

Jason Brownlee February 17, 2021 at 5:31 am #

REPLY ↩

No, in k-fold cross-validation, the model is fit on the training folds and evaluated on the hold out fold, repeated k times for different folds.

nd February 25, 2021 at 3:36 am #

REPLY ↩

After finding loss on every model in k-folds on test dataset. how to find loss and sd ans accuracy of model

Jason Brownlee February 25, 2021 at 5:36 am #

REPLY ↩

Sorry, I don’t understand your question, can you please rephrase or elaborate?

nd February 25, 2021 at 7:53 am #

REPLY ↩

i got different losses for models in k fold . i asking how to find final loss and standard deviation value from that and also accuracy?

Jason Brownlee February 25, 2021 at 8:54 am #

REPLY ↩

You can evaluate your model on the hold out test set and calculate loss and accuracy for each fold, then calculate mean and stdev of the population of scores.

This will help you get started:
https://machinelearningmastery.com/evaluate-skill-deep-learning-models/

This has an example:
https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-from-scratch-for-mnist-handwritten-digit-classification/

nd February 26, 2021 at 6:41 am #

REPLY ↩

thanks

✕

After 3 days of studying k-fold-cross validation for a multi layer perceptron, I wonder, why I cant find any answer on the fellowing problem:

I have k different learning sets. Let j be the number of the learning set, where the j-th fold is the validation set.

Do I calculate , only focussing on learning set 1, all final weights going through lots of iterations and epochs and then check the validation-accuracy on fold 1 of the network, and then doing the same for all learning sets, receiving k accuracy values?
If so, why do I average all these accuracy values, having DIFFERENT weights??? For every learning set I get a different multi layer perceptron at the end of all calculated epochs.
And, which weights do I finally take????? The ones established by learning set 1 or k or the average???

Jason Brownlee March 1, 2021 at 5:38 am #

REPLY

Great questions.

Yes, each of the k folds gets a turn to be used as a test set while all other folds are used as train. The train can be further split into train and val if you like for tuning or whatever.

Model skill is reported as the mean performance on all the test sets.

All models are then discarded once you have an estimate. You fit a final model on all data and start making predictions on new data.

Danny March 5, 2021 at 10:23 am #

REPLY

I do not understand this quote: “The choice of k is usually 5 or 10, but there is no formal rule. As k gets larger, the difference in size between the training set and the resampling subsets gets smaller. As this difference decreases, the bias of the technique becomes smaller.” Can you please define “resampling subsets”? Are they the folds?

Jason Brownlee March 5, 2021 at 1:35 pm #

REPLY

It means the data after it is split up. E.g. the actual rows used to train the model and test the model for a given split.

Danny March 5, 2021 at 10:27 am #

REPLY

Jason Brownlee March 5, 2021 at 1:36 pm # REPLY

No, it is referring to all rows you have vs the size of the rows to train the model.

if k=n-1 (all data except one row) then then difference between a training set and the entire dataset is 1.

Mariana March 8, 2021 at 4:03 am # REPLY

I have no questions! I would just like to thank you for summarizing so many imporant topics!!!

Jason Brownlee March 8, 2021 at 4:56 am # REPLY

You're very welcome!

Alessandra April 6, 2021 at 9:13 pm # REPLY

Thank you for your information… How would I have to deal with ROC analysis in case of a K-fold cross val? Should I compute the curve each time and then average the k outcomes at the end?
Thank you!

Jason Brownlee April 7, 2021 at 5:09 am # REPLY

Generally you don't, you would use a train/test split to estimate a roc curve.

Shahbaz Khan May 4, 2021 at 4:34 pm # REPLY

Hi, what is k-fold accuracy? Is it the same?

Jason Brownlee May 5, 2021 at 6:08 am # REPLY

Perhaps it is the mean accuracy calculated from k-fold cross-validation.

✕

Hi,
Thank you very much for such a nice article. I am working on a SED project and I am using DCASE 2017 Task 3 dataset for polyphonic SED. The dataset comes in two phases i.e., Development and evaluation. In the development Dataset there a ready-made data for 4-Folds training and evaluation. In the case of training folds, there are around 15 clips in each fold, and in the case of evaluation, there are 5 clips per fold. But, I am not sure if these evaluation clips should be used as validation data or should I take validation data from the training data. Kindly guide me on how to use the evaluation data folds as test data folds if I take validation data (soy 10%) from the training data folds.

Jason Brownlee May 21, 2021 at 6:02 am # REPLY ↩

I'm not familiar with that dataset, perhaps discuss the data with the stakeholders that provided it to you.

Maha July 30, 2021 at 8:17 am # REPLY ↩

Hi
If there are good citation references to cite cross-validation?

Jason Brownlee July 31, 2021 at 5:33 am # REPLY ↩

Not sure off hand, perhaps check for the first paper on scholar.google.com

Sumayya August 29, 2021 at 5:03 am # REPLY ↩

Hi, thanks for the good explanation! I got a question:

When using kfold which splits our data into folds of train and test data, so if we probably use that with gridSearch which takes in a model as well, we do gridSearch.fit(X_train,y_train). Are the test portions of our kfold get used to fit the model or they're used when we do .predict()?
I guess all the kfold data (train and test) used to fit the data, right?

Adrian Tam August 29, 2021 at 12:30 pm # REPLY ↩

k-fold always hold one out. They will not use it for predict() unless you write your code to do

⊗

ali September 25, 2021 at 2:39 am #

REPLY

you have to classify the Iris flowers based on the provided sepal and petal features in the Iris dataset using KNN. There are 50 samples for each of the three classes. Split the data into 80% – 20% training
– testing samples for each class to do a 5-fold cross validation.
Try different values of K (e.g. 1, 3, 5, 7 —–) and different distances (L1 and L2).
Report what value of K and distance metric (L1/L2) gives the best results in terms of “Accuracy”

References:
https://machinelearningmastery.com/k-fold-cross-validation/
https://scikit-learn.org/stable/modules/cross_validation.html

how to solve?

Alexandre November 29, 2021 at 6:50 pm #

REPLY

Hi Jason,

You mentioned here that “Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample”

I wonder how big/small this data sample must be in order to reject or accept an approach like k-fold cross validation ?

thanks

Adrian Tam December 2, 2021 at 12:30 am #

REPLY

Believe it or not – it can take many pages answering this question. In practice, I would like to have the data sufficient in each training set to train the model and sufficient to tell the test result is reliable. A decision tree, for example, probably a few hundred data point is good enough. But for a rigorous answer, you will need to consider the confidence interval, t-test, etc.

Barney December 4, 2021 at 8:58 am #

REPLY

Thought you may find this interesting is you haven’t already come across it:

https://onlinelibrary.wiley.com/doi/10.1111/ecog.02881

Thanks for sharing!

AGGELOS PAPOUTSIS January 6, 2022 at 4:55 am #

REPLY

Hi,

here https://scikit-learn.org/stable/modules/cross_validation.html

states that "A test set should still be held out for final evaluation"

How does is this possible as far as we do not use a train/test split?

Thank you

James Carmichael January 6, 2022 at 10:46 am #

REPLY

Hello Aggelos…The following may be of interest to you:

https://machinelearningmastery.com/training-validation-test-split-and-cross-validation-done-right/

Mahmud Kiru January 7, 2022 at 6:31 am #

REPLY

I have run a cv method on 6 different models with a range of k = 1 to 20. I realised that as the number of k increases, the performance of the model becomes more better. this trend cut across all the models in my experiment. what could have been the reason please?

James Carmichael January 7, 2022 at 8:04 am #

REPLY

Hi Mahmud…Would it be possible to reframe your question to a specific code listing in the original blog post or other materials we offer?

Regards,

Jullian February 16, 2022 at 7:11 pm #

REPLY

If suppose the remaining data K is split into equally sized blocks. How large should the blocks be chosen?

Hi Jullian…Most often either k=5 or k=10 is chosen however you may certainly try other values as suggested in the following discussion.

https://stats.stackexchange.com/questions/27730/choice-of-k-in-k-fold-cross-validation

Sophia Yue March 13, 2022 at 5:15 pm # REPLY

Hi Jason,
Can k_fold, cross-validation, gridsearchcv be used for unsupervised learning?

Adi March 31, 2022 at 1:19 pm # REPLY

Please someone help me to understand this thing very well.

When we use K-fold cross validation can we input the complete dataset for cross validation or just training part of data set for cross validation?
Like most of the studies describe train, validate , test concept about it.

Let suppose, my dataset has 1000 records.
With train_test_split I divide it into 2:1 ration (666 for testing and 334 for testing)

Now shall I only provide these 666 records for K-fold cross validation or whole 1000 records. If only 666 records.

Then if we train model on K-1 part and validate on 1 part on each iteration and finally got mean score of the model through cross_val_score. that is 96.09

Is it my models final accuracy?
Then what is the use of these 334 records left for testing >

James Carmichael April 1, 2022 at 9:15 am # REPLY

Hi Adi…You may find the following of interest:

https://vitalflux.com/k-fold-cross-validation-python-example/

Tayfun Han November 30, 2022 at 8:31 am # REPLY

Thank you very much for this excellent article.

to latent variables.

However, after this classification, I would like to implement k-fold cross-validation, just to validate my model. Unfortunately, tutorials on the internet generally explain with raw data (X) and target (y). Therefore, I am not sure which data I should use for the validation after the implementation of my model.

The point here is to find a logical way of validating my model with k-fold cross-validation.

Thanks for any advice.
Best.

James Carmichael November 30, 2022 at 8:54 am # REPLY ↩

Hi Tayfun…The following resource may be of interest:

https://machinelearningmastery.com/training-validation-test-split-and-cross-validation-done-right/

Zilah Maria Cheuiche July 12, 2023 at 6:46 am # REPLY ↩

Hello Jason, thank you for the excellent article! I performed a 3-fold validation to measure the accuracy of allele imputation, but I got the same result in all three validations. My advisor said there's something wrong, but I couldn't find the error. What do you think?

James Carmichael July 12, 2023 at 11:44 am # REPLY ↩

Hi Zilah…More detail on this method can be found here:

https://machinelearningmastery.com/repeated-k-fold-cross-validation-with-python/

Murilo August 7, 2023 at 6:22 am # REPLY ↩

Hello, i have one question.

After performing the cross-validation, we discard all the models because we know, in average, the performance of our model in the unseen data. After that, i have read that we should train a new model using the whole dataset, is that correct? Or is there any other approach?

Could you share a reference to a book or a paper so i could read more into this (i.e., the procedure that should be done AFTER the cross-validation)?

✕

hi Jason,

I guess if the number of fold is 10, the data will be trained 10 times. Does it mean 10 models are generated? Does the Kfold train the model with all the data one more time to get the final model? So that means the first 10 times is just to evaluate the average performance of the model. The last one with all the data is for generating the final model. Not sure my understanding is correct or not.

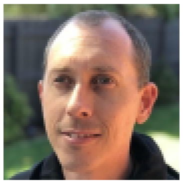
Thanks,
Sam

Leave a Reply

Name (required)

Email (will not be published) (required)

SUBMIT COMMENT



Welcome!
I'm *Jason Brownlee* PhD
and I **help developers** get results with **machine learning**.
Read more

Never miss a tutorial:

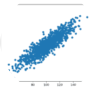
Picked for you:

Statistics for Machine Learning (7-Day Mini-Course)

A Gentle Introduction to k-fold Cross-Validation

Statistical Significance Tests for Comparing Machine Learning Algorithms

How to Calculate Bootstrap Confidence Intervals For Machine Learning Results in Python


How to Calculate Correlation Between Variables in Python

LinkedIn | Twitter | Facebook | Newsletter | RSS