Machine Learning Mastery
Making Developers Awesome at Machine Learning

How to Train a Final Machine Learning Model

by **Jason Brownlee** on March 17, 2017 in **Machine Learning Process**

The machine learning model that we use to make predictions on new data is called the final model.

There can be confusion in applied machine learning about how to train a final model.

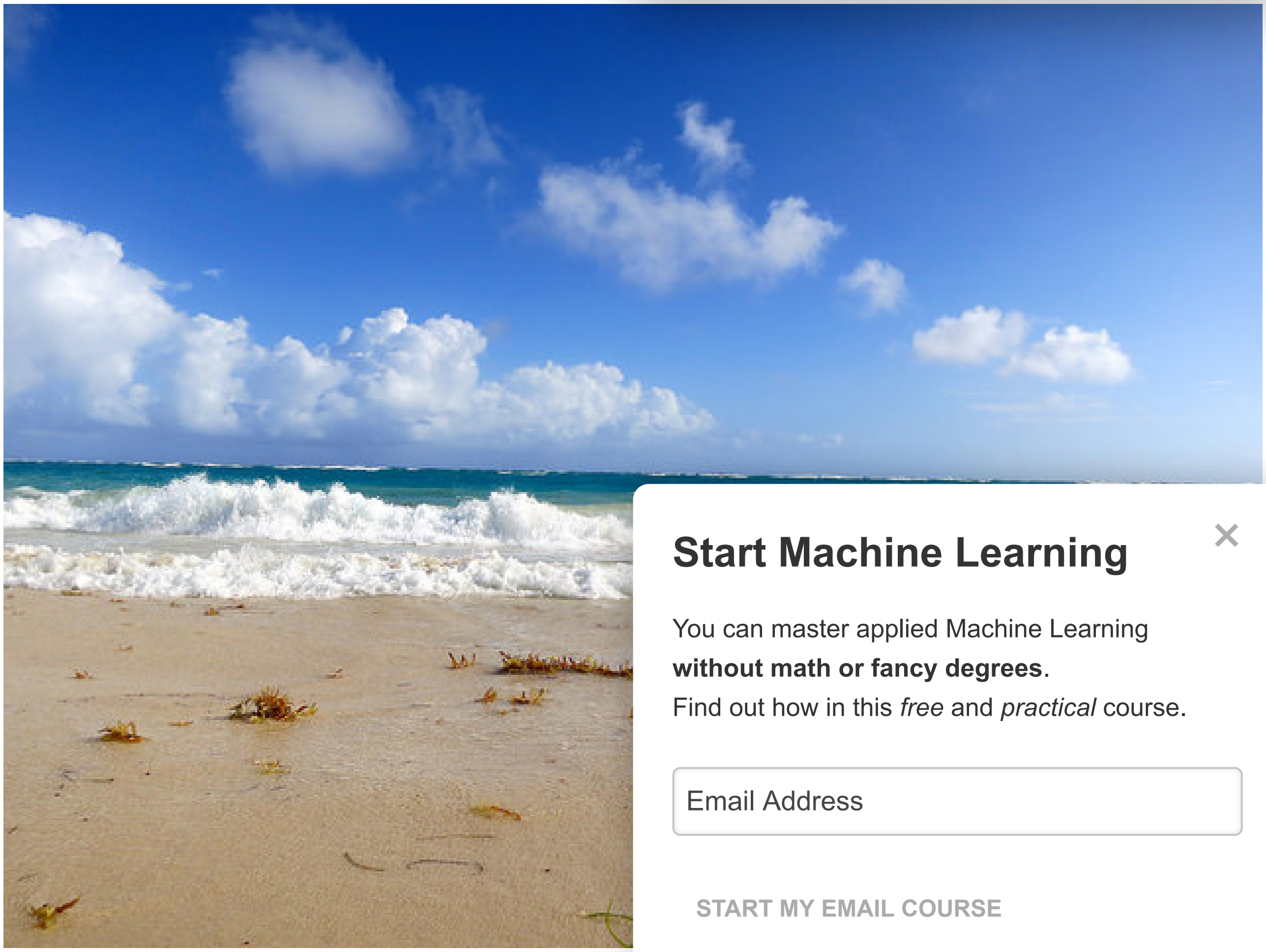
This error is seen with beginners to the field who ask questions such as:

- *How do I predict with cross validation?*
- *Which model do I choose from cross-validation?*
- *Do I use the model after preparing it on the training dataset?*

This post will clear up the confusion.

In this post, you will discover how to finalize your machine learning model in order to make predictions on new data.

Let’s get started.



How to Train a Final Machine Learning Model
Photo by Camera Eye Photography, some rights reserved.

What is a Final Model?

A final machine learning model is a model that you use to make predictions on new data.

That is, given new examples of input data, you want to use the model to predict the expected output. This may be a classification (assign a label) or a regression (a real value).

For example, whether the photo is a picture of a *dog* or a *cat*, or the estimated number of sales for tomorrow.

The goal of your machine learning project is to arrive at a final model that performs the best, where "best" is defined by:

- **Data**: the historical data that you have available.
- **Time**: the time you have to spend on the project.
- **Procedure**: the data preparation steps, algorithm or algorithms, and the chosen algorithm configurations.

In your project, you gather the data, spend the time you have, and discover the data preparation procedures, algorithm to use, and how to configure it.

The final model is the pinnacle of this process, the end you seek in order to start actually making predictions.

The Purpose of Train/Test Sets

Why do we use train and test sets?

Creating a train and test split of your dataset is one method to quickly evaluate the performance of an algorithm on your problem.

The training dataset is used to prepare a model, to train it.

We pretend the test dataset is new data where the ou
gather predictions from the trained model on the input
withheld output values of the test set.

Comparing the predictions and withheld outputs on the e
measure for the model on the test dataset. This is an
problem when making predictions on unseen data.

Let’s unpack this further

When we evaluate an algorithm, we are in fact evalua
training data was prepared (e.g. scaling), the choice of algorithm (e.g. kNN), and how the chosen algorithm was configured (e.g. k=3).

The performance measure calculated on the predictions is an estimate of the skill of the whole procedure.

We generalize the performance measure from:

- “*the skill of the procedure on the **test set***“

to

- “*the skill of the procedure on **unseen data***“.

This is quite a leap and requires that:

- The procedure is sufficiently robust that the estimate of skill is close to what we actually expect on unseen data.
- The choice of performance measure accurately captures what we are interested in measuring in predictions on unseen data.
- The choice of data preparation is well understood and repeatable on new data, and reversible if predictions need to be returned to their original scale or related to the original input values.
- The choice of algorithm makes sense for its intended use and operational environment (e.g. complexity or chosen programming language).

A lot rides on the estimated skill of the whole procedure on the test set.

In fact, using the train/test method of estimating the skill of the procedure on unseen data often has a high variance (unless we have a heck of a lot of data to split). This means that when it is repeated, it gives different results, often very different results.

The outcome is that we may be quite uncertain about how well the procedure actually performs on unseen data and how one procedure compares to another.

Often, time permitting, we prefer to use k-fold cross-validation instead.

The Purpose of k-fold Cross Validation

Why do we use k-fold cross validation?

Cross-validation is another method to estimate the ski est split.

Cross-validation systematically creates and evaluates t.

This, in turn, provides a population of performance me

- We can calculate the mean of these measures to n average.
- We can calculate the standard deviation of these measures to get an idea of how much the skill of the procedure is expected to vary in practice.

This is also helpful for providing a more nuanced comparison of one procedure to another when you are trying to choose which algorithm and data preparation procedures to use.

Also, this information is invaluable as you can use the mean and spread to give a confidence interval on the expected performance on a machine learning procedure in practice.

Both train-test splits and k-fold cross validation are examples of resampling methods.

Why do we use Resampling Methods?

The problem with applied machine learning is that we are trying to model the unknown.

On a given predictive modeling problem, the ideal model is one that performs the best when making predictions on new data.

We don’t have new data, so we have to pretend with statistical tricks.

The train-test split and k-fold cross validation are called resampling methods. Resampling methods are statistical procedures for sampling a dataset and estim

In the case of applied machine learning, we are interested in estimating the skill of a machine learning procedure on unseen data. More specifically, the skill of the predictions made by a machine learning procedure.

Once we have the estimated skill, we are finished with the resampling method.

- If you are using a train-test split, that means you can discard the split datasets and the trained model.
- If you are using k-fold cross-validation, that means you can throw away all of the trained models.

They have served their purpose and are no longer needed.

You are now ready to finalize your model.

How to Finalize a Model?

You finalize a model by applying the chosen machine

That's it.

With the finalized model, you can:

- Save the model for later or operational use.
- Make predictions on new data.

What about the cross-validation models or the train-test datasets?

They've been discarded. They are no longer needed. They have served their purpose to help you choose a procedure to finalize.

Common Questions

This section lists some common questions you might have.

Why not keep the model trained on the training dataset?

and

Why not keep the best model from the cross-validation?

You can if you like.

You may save time and effort by reusing one of the models trained during skill estimation.

This can be a big deal if it takes days, weeks, or months to train a model.

Your model will likely perform better when trained on all of the available data than just the subset used to estimate the performance of the model.

This is why we prefer to train the final model on all available data.

Won’t the performance of the model trained on all of the data be different?

I think this question drives most of the misunderstanding around model finalization.

Put another way:

- If you train a model on all of the available data, then how do you know how well the model will perform?

You have already answered this question using the resampling procedure.

If well designed, the performance measures you calcu bly describe how well the finalized model trained on all av

If you used k-fold cross validation, you will have an es the model will be on average, and the expected spread of

This is why the careful design of your test harness is s A more robust test harness will allow you to lean on the

Each time I train the model, I get a dif pick the model with the best score?

Machine learning algorithms are stochastic and this behavior of different performance on the same data is to be expected.

Resampling methods like repeated train/test or repeated k-fold cross-validation will help to get a handle on how much variance there is in the method.

If it is a real concern, you can create multiple final models and take the mean from an ensemble of predictions in order to reduce the variance.

I talk more about this in the post:

- Embrace Randomness in Machine Learning

Summary

In this post, you discovered how to train a final machine learning model for operational use.

You have overcome obstacles to finalizing your model, such as:

- Understanding the goal of resampling procedures such as train-test splits and k-fold cross validation.
- Model finalization as training a new model on all available data.
- Separating the concern of estimating performanc

Do you have another question or concern about finalizing your model that I have not addressed? Ask in the comments and I will do my best to help.

Tweet Share Share

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 Install a Python for Machine Learning on macOS on >

186 Responses to *How to Train a Final M*

Elie Kawerk March 17, 2017 at 7:06 am #

Hi Jason,

Thank you for this very informative post. I have a question regarding the train-test split for classification problems: Can we perform a rain/test split in a stratified way for classification or does this introduce what is called data snooping (a biased estimate of test error)?

Thanks
Elie

Jason Brownlee March 17, 2017 at 8:33 am # REPLY

The key is to ensure that fitting your model does not use any information about the test dataset, including min/max values if you are scaling.

kamran October 24, 2017 at 6:46 pm # REPLY

Hi Jason,
Do you have any blog on How to Deploy the Final ML model

Jason Brownlee October 25, 2017

This post will give you some ideas:
http://machinelearningmastery.com/deploy-machine-learning-model-to-production/

Indunil July 4, 2018 at 3:03 am # REPLY

How to save final model in Tenorflow and use it in Tenorflow.js

Dan March 18, 2017 at 5:59 am # REPLY

"Also, this information is invaluable as you ca
interval on the expected performance on a machine lea

I have to assume a normal distribution for that right? B
my data in a preprocessing step and then it would be c

Jason Brownlee March 18, 2017 at 7:53 am #

Hi Dan, great question!

Yes, we are assuming results are Gaussian to report results using mean and standard deviation.

Repeating experiments and gathering info on the min, max and central tendency (median, percentiles) regardless of the distribution of results is a valuable exercise in reporting on model performance.

Kleyn Guerreiro March 20, 2017 at 10:36 pm # REPLY

Great post….my little experience teached me that:
a) for classification you can use your final trained model with no risk
b) for regression, you have to rerun your model againt all data (using the parameters tuned during training)
b) specifically for time series regression, you can't use normal cross validation – it should respect the cronology of the data (from old to new always) and you have to rerun your model againt all data (using the parameters tuned during training) as well, as the latest data are the crucial ones for the model to learn.
Cheers!

Jason Brownlee March 21, 2017 at 8:40 am # REPLY

Thanks for the tips Kleyn.

jiggy August 5, 2019 at 9:17 pm # REPLY

Hi Jason
So, my question is how to predict the time series data not only for the test set but for further future forecast?
As I cannot used the saved model to predict the time series data.

Thank you

Jason Brownlee August 6, 2019 at 6:36 am # REPLY

Fit a final model on all available d

If new data becomes available, you must c
is.

Hank May 12, 2017 at 5:16 am #

Great post! I really learned a lot from your pos
there are few questions still in my mind. In our project,
with and without 10-fold cv, including logistics regressi
score of each model with 10-fold cross validation, but the problem is how can we get the final model with 10-fold? Does the cross-validation function as finding best parameter of the different model? (such determine k in kNN?) I am still a little bit confused about the purpose of cross-validation. Thanks

Jason Brownlee May 12, 2017 at 7:49 am # REPLY

Hi Hank, the above directly answers this question.

Cross-validation is a tool to help you estimate the skill of models. We calculate these estimates so we can compare models and configs.

After we have chosen a model and it’s config, we throw away all of the CV models. We’re done estimating.

We can now fit the “final model” on all available data and use it to make predictions.

Does that make sense?
Please ask more questions if this in not clear. This is really important to understand and I thought I answered all of this in the post.

Hank May 12, 2017 at 3:20 pm #

Hi Jason,

Thank you so much! Does that mean cross-validation is just a tool to help us compare different models based on cross-validation score?
After we are done with evaluation, we would apply original model to whole dataset and make predictions. Since I read a paper where the author compare auc, true positive rate, true negative rate, false positive rate and false negative rate between those models with and without cross-validation. It turns out that logistic regression with 10fold perform best. So I though we will apply logistics regression with 10-fold to test data. Is my understanding incorrectly? Thanks!

Jason Brownlee May 13, 2017 at 6 REPLY

Yes, CV is just a tool to compare

Petros May 12, 2017 at 10:55 am #

Hi Jason,

Great post.

It took me awhile to get this but when the penny dropp s validation to experimenting a process which you want t though.

When you cross validate you might say 10 folds of 3 repeats for each combination of parameters. Now say with whatever measure you are taking for accuracy you typically taken the mean from these 30. Is it sensible to bootstrap with replacement, particularly if it is not Gaussian, from this sample of 30 say 1000 times and from their calculate the median and 2.5/97.5 percentiles?

What does everyone else think!

PK

Jason Brownlee May 13, 2017 at 6:09 am # REPLY

Yes, I like to use the bootstrap + empirical confidence intervals to report final model skill.

I have a post showing how to do this scheduled for later in the month.

Warren van Niekerk May 12, 2017 at 2:56 pm # REPLY

Thanks for the very informative post. Just one question: When you train the final model, are you learning a completely new model or is some or all of th retained?

Jason Brownlee May 13, 2017 at 6:11 am # REPLY

Yes, generally, you are training an entirely new model. All the CV models are discarded.

Muralidhar SJ May 12, 2017 at 6:53 pm # REPLY

Thanks Jason. Very Useful info & insight , helping lot to take right approach.

Jason Brownlee May 13, 2017 at 6:12 am #

I'm glad it helped Muralidhar.

Imene May 14, 2017 at 6:57 am #

Thank you very much Jason. I found in this po

Jason Brownlee May 14, 2017 at 7:33 am # REPLY

I'm so glad to hear that Imene.

issam May 14, 2017 at 8:42 am # REPLY

Hi Jason
I want tank you for this informative post . I m working in project "emotion recognition on image" I want to know how can I create my model and train it.

thanks in advance

Jason Brownlee May 15, 2017 at 5:50 am # REPLY

I'm glad it helped issam.

EN MO May 14, 2017 at 5:37 pm # REPLY

Very informative, thanks alot, am also trying to see if this will be useful in a project I would like to do, and how it can be applied in biometrics and pattern recognition

Jason Brownlee May 15, 2017 at 5:52 am # REPLY

Thanks.

Ras May 17, 2017 at 10:38 pm #

Thanks for the article. What about the parame
set or via cross-validation. The optimum parameter set
Wouldn't it be left to chance for our optimized paramet
well?

Jason Brownlee May 18, 2017 at 8:37 am #

Hi Ras,

k-fold cross-validation is generally the best practice for using the training dataset to find a "good" configuration of a model.

Does that help? Is that clearer?

lalneirem May 23, 2017 at 7:28 pm # REPLY

thanks for this post
i know this may be useful but i don't know what we do in training phase using KNN
if u can write the details step that is done during training phase
i will be so grateful

Jason Brownlee May 24, 2017 at 4:53 am # REPLY

There is no training of knn, only predictions.

See this post:
http://machinelearningmastery.com/tutorial-to-imple

aquaq June 1, 2017 at 6:29 pm # REPLY

Thanks for this post, it has given a clear explanation for most of my questions. However, I still have one question: if I have used undersampling duting CV, how should apply it to my whole data. To be clearer
– I have a training set of around 1 million positive (+) and 130 thousand negative (-) examples. I also have an independent test data set with a hundred thousand (+) and 4000 (-) examples.
– I have estimated performance with 10-fold CV and applied undersampling (I have used R gmlnet package, logit regression with LASSO, training for AUC). It gave me super results for the CV.

And now I'm lost a bit. Training for all data would mean to randomly select 130 thousand (+) from the 1 million and only use this ~260 thousand examples? Should I evaluate my model after training on my test data set?

Thank you for your help!

Jason Brownlee June 2, 2017 at 12:56 pm #

If you can, I would suggest evaluating the

In fact, it is a good idea to understand the data set
diminishing returns.

Chayanika Mudiar June 19, 2017 at 10:19 pm # REPLY

I have a question. In the training process using gausion naive bayes, can you say what are the steps to be taken to train the model.

Jason Brownlee June 20, 2017 at 6:37 am # REPLY

Yes, see here:
http://machinelearningmastery.com/naive-bayes-classifier-scratch-python/

Tyrone July 7, 2017 at 5:40 pm # REPLY

Hi Jason. Thanks for a great article!

When you say that "You finalize a model by applying the chosen machine learning procedure on all of your data", does this mean that before deploying the model you should train a completely new model with the best hyperparameters from the validation phase, but now using training data + validation data + testing data, i.e. including the completely unseen testing data that y

This is how I interpret it, and it makes sense to me given that the whole the whole point of validation is to estimate the performance of a method of generating a model, rather than the performance of the model itself. Some people may argue, though, that because you're now training on previously unseen data, it is impossible to know how the new trained model is actually performing and whether or not the new, real-world results will be in line with those estimated during validation and testing.

If I am interpreting this correctly, is there a good technical description anywhere for why this works in theory, or a good explanation for convincing people that this is the correct approach?

Jason Brownlee July 9, 2017 at 10:37 am # REPLY

Yes. Correct.

Yes. The prior results are estimates of the perform

Tyrone July 10, 2017 at 11:19 pm #

Thanks Jason. It's great to have confi

or sources out there that spell this out explicitly

sound?

Jason Brownlee July 11, 2017 at 10:33 am # REPLY

Not off hand sorry.

aquaq July 25, 2017 at 10:56 pm # REPLY

Thanks Jason for this explanation. I would like to ask how to deal with test sets when I would like to compare the performance of my model to existing models. Do I have to hold out a test set, train my model on the remaining data and compare all models using my test set?
After that, can I merge this held out set to my original training set and use all data for training a final model?
What other solutions can be used?

Jason Brownlee July 26, 2017 at 7:54 am # REPLY

Yes. Choose models based on skill on the test set. Then re-fit the model on all available data (if this makes sense for your chosen model and data).

Does that make sense?

aquaq July 28, 2017 at 3:52 am # REPLY

Yes, it makes sense, thank you.

Jason Brownlee July 28, 2017 at 8:33 am # REPLY

Great!

Paul August 3, 2017 at 10:21 am #

Thank you for the great post Jason.
I have a question about forecasting unseen data in RN
I've built complete model using RNN with LSTM by usi
series-prediction-lstm-recurrent-neural-networks-pytho
How can we forecast unseen data(like ahead of curren
I mean we don't have any base data except time thoug

I already saw some comments that you replied "You ca
model.predict(X)" on that post. However, I couldn't und

Paul August 3, 2017 at 10:30 am # REPLY

I mean in real-time. ☺

Thanks in advance.

Best,
Paul

Jason Brownlee August 4, 2017 at 6:47 am # REPLY

In real time, the same applies, but you can decide whether you re-train a new model, update the model or do nothing and just make predictions.

Jason Brownlee August 4, 2017 at 6:46 am # REPLY

You can predict the next step beyond the available data by training the model on all current data, then calling predict with whatever input your model takes taken from the end of the training data.

Does that help?

Which part is confusing?

S H September 8, 2017 at 11:45 pm #

REPLY

Hi Jason.

Thanks a lot for this great and informative post. I have 2 questions I would be thankful if you can help me with them:

1- Is that possible to refresh (update) a model without r
built using 9 weeks of data (weekly snapshots). As the
update the model on a weekly basis, it takes a lot of tir
snapshot (say for week 10), without retraining the mod
snapshot)?

2- When I train my model and evaluate it using cross-v
which are consistently better than what I get when I sc
model. Why is that so, and how can I treat it? To elabo
first question, I use snapshot_date column as cross-va
algorithm uses 8 weeks of data for training, and test the model on the remaining unseen week. Therefore, I would end up with 9 different models and 9 different AUCs on the validation frame. All the AUCs are between 0.83 to 0.91. So I would expect that the real performance of the model built using whole data should be at minimum AUC 0.83. However, when I score the serving data, and the next week I assess the performance of the model, I see no better than AUC 0.78. I have experienced it for 3 weeks (3 times), so I don't think it's just random variation. Additionally, I am quite sure there is no data leakage and there is no future variable in my data. Also, I tune the model quite well and there is no overfitting.

Your help is highly appreciated.

Jason Brownlee September 9, 2017 at 11:57 am #

REPLY

You can update a model. The amount of updating depends on your data and domain. I have some posts on this and more in my LSTM book.

Model evaluation on test data is biased and optimistic generally. You may want to further refine your test harness to make the reported scores less biases for your specific dataset (e.g. fewer folds, more folds, more data, etc. depends on your dataset).

Kenny October 12, 2017 at 1:05 am #

Hello Jason, very interesting this post! what do you get

I finished my model (with a score of 91%), but how do I do or how to evaluate this model with the new dataset?

I have saved the model in model.pkl but in my new data (for example iris.csv) How to predict the field “species”? (in my datasets do I need to put this field blank?) how is this step?

Thks for your help because I’m confused

Jason Brownlee October 12, 2017 at 5:33 am # REPLY

Load the model and call model.predict(X)

Prasshanth VP January 16, 2018 at 10:05 am #

Hi Jason – Great post. This cleared things for

```
fitControl <- trainControl(
method = "repeatedcv",
number = 10,
savePredictions = 'final',
verboseIter = T,
summaryFunction = twoClassSummary,
classProbs = T)
```

```
glm_fit <- caret::train(dv ~. , data = dataset
,method = "glm", family=binomial, trControl = fitControl, metric = "ROC")
```

It says that the glm_fit now becomes the final model as it runs 10 fold based on trControl and finally trains model using entire data. Setting verboseIter = T, gives me a summary during this run a message at the end – "Fitting final model on full training set". So can I use this as a final model?

Jason Brownlee January 17, 2018 at 9:54 am # REPLY

Perhaps.

Martin Main January 18, 2018 at 2:38 am # REPLY

Hi there,
This article makes a lot of sense, but one thing I am surprised was not addressed was the problem of over-fitting. If there is no test/validation data used in the fina been seen to over-fit the data in testing, then we need

simple approach would be to guess from the 'correct' training times from the previous tests, but of course the final model with all data will naturally need longer training times. Is there a statistical approach we could use to determine the best time to stop training without using a validation set?

Jason Brownlee January 18, 2018 at 10:12 am # REPLY

Concerns of overfitting are addressed prior to finalizing the model as part of model selection.

Megat Haziq March 12, 2018 at 7:06 pm # REPLY

If I was using early stopping during k-
number of epoch and apply it to the finalized m
model for early stopping, so I thought of using
Please help me ☺

Jason Brownlee March 13, 2018 at

Yes, you could try that.

Tata March 7, 2018 at 5:13 am # REPLY

Hi Jason! Thank you so much for this informative post.

A little question though. If we don't have the luxury to acquire another dataset (because it's only for a little college project, for example), how do you apply k-fold cross validation (or test-training split) to evaluate models then?

My understanding is that once you apply, let's say, k-fold cross validation for choosing which model to use and then tuning the parameters to suit your need, you will run your model on another different dataset hoping the model you have built and tuned will give you your expected result.

Jason Brownlee March 7, 2018 at 6:17 am # REPLY

You can split your original dataset prior to using CV.

Rose March 21, 2018 at 8:46 am # REPLY

Hi Jason,

Glad to meet with your tutorial as these are one of the best in teaching deep wuth keras.
I have already read the notes which people asked you questions about using k-fold cv for training a final deep model but as I am a naive in working with deep learning models I could not understand some things.
I wanna train (or finalized) CNN,LSTM & RNN for text dataset (it is a sentiment analysis). In fact my teacher told me to apply k-fold cross validation for training=finalizing model to be able to predict the proability of belonging unseen data to each class (binary class).
my question is this:[is it wrong to apply k-fold cross validation to train a final deep model?]
as I wrote commands 15 epoches run in each fold. is there any thing wrong with it?
I am so sorry for my naive question as i am not a english native to understand perfect the above comments U all wrote about it.
my written code is like this:
[

```
from sklearn.model_selection import KFold
kf = KFold(10)
f1_lstm_kfld=[]
oos_y_lstm = []
oos_pred_lstm = []
fold = 0
for train, test in kf.split(x_train):
fold += 1
print("Fold #{}".format(fold))
print('train', train)
print('test', test)
x_train1 = x_train[train]
y_train1 = y_train[train]
x_test1 = x_train[test]
y_test1 = y_train[test]
print(x_train1.shape, y_train1.shape, x_test1.shape, y_test1.shape)

print('Build model…')
model_lstm = Sequential()
model_lstm.add(Embedding(vocab_dic_size, 128))
model_lstm.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))
model_lstm.add(Dense(1, activation='sigmoid'))

model_lstm.compile(loss='binary_crossentropy',
optimizer='adam',
metrics=['accuracy'])

print('Train…')
model_lstm.fit(x_train1, y_train1,
batch_size=32,
epochs=15,
validation_data=(x_test1, y_test1))
score_lstm, acc_lstm = model_lstm.evaluate(x_test1, y_test1,
batch_size=32)
```

```
sum_f1_lstm_kfld=0
for i in f1_lstm_kfld:
sum_f1_lstm_kfld =sum_f1_lstm_kfld+i
print ('sum_f1_lstm_kfld',sum_f1_lstm_kfld)
mean_f1_lstm_kfld=(sum_f1_lstm_kfld)/10
print ('mean_f1_lstm_kfld', mean_f1_lstm_kfld)
```

Please guide me as i get confused.
Thank U in advanced.
Rose

Jason Brownlee March 21, 2018 at 3:06 pm #

You cannot train a final model via CV.

I recommend re-reading the above post as to why.

Rose March 23, 2018 at 7:29 am #

Hi Jason,
I am afraid to ask about this issue again but wh
Resampling methods like repeated train/test o
handle on how much variance there is in the method. If it is a real concern, you can create multiple final models and take the mean from an ensemble of predictions in order to reduce the variance.
What do U mean by this sentence: "create multiple final models" do you mean applying k-fold cross-validation to achieve multiple models??
and also about this one: take the mean from an ensemble of predictions, you mean can we use "this take the mean" in finalizing model?
I wanna train a model=finalizing model.
You mentioned: "Why not keep the best model from the cross-validation?
You can if you like.You may save time and effort by reusing one of the models trained during skill estimation.Your model will likely perform better when trained on all of the available data than just the subset used to estimate the performance of the model.
what do u mean by saying above three sentences especially this one:"when trained on all of the available data than just the subset used to estimate the performance of the model?
you mean by "training on all of the available data" the procedure which we do not use k-fold cross validation?? and you mean applying k-fold cross validation from this sentence:"just the subset used to estimate the performance of the model"?
If I want to ask my question clearly I shold say in this way: [I wanna train a CNN ,LSTM and RNN deep model to define a deep model inorder to estimating the proability of unseen data, what should I do? applying splitting data set into train and test or any other procedures?

Any guidance will be appreciate.

Jason Brownlee March 23, 2018 at 8:30 am # REPLY

I mean, if there is a lot of variance in the model predictions (e.g. due to randomness in the model itself), you can train multiple final models on all training data and use them in an ensemble.

Sometimes training a model can take days or weeks. You may not want to retrain a model, hence, reuse a model from the time when you estimated model skill.

Does that help?

moses April 6, 2018 at 4:07 pm #

can u provide a sample code for prediction

Jason Brownlee April 7, 2018 at 6:08 am #

I have many examples on my blog for diff

Try a search and let me know if you don't find wha

Mariano April 26, 2018 at 12:44 pm # REPLY

Do you happen to have an example with code where you train with all your data and then predict unknown future data?

Jason Brownlee April 26, 2018 at 3:03 pm # REPLY

Yes, you can learn how to make predictions on new data here:
https://machinelearningmastery.com/faq/single-faq/how-do-i-make-predictions

Casey May 17, 2018 at 4:46 am # REPLY

Thanks for this post Jason, and two additional questions:

1) Is there a peer-reviewed article that can be cited to demonstrate the validity of this approach?

2) Do I understand correctly that if the uncertainty in the relationship derived for the training data is correctly propagated to the test data set, the “best” model can be selected based solely on cross-validation statistics? That is, goodness of fit measure for the training relatio

Thanks!

Jason Brownlee May 17, 2018 at 6:39 am # REPLY

Of finalizing a model? There may be, I don’t know sorry. It might be tacit knowledge.

Yes, skill estimated using a well configured k-fold cross-validation may be sufficient, but i the score is reviewed too often (e.g. to tune hyperparams), you can still overfit.

AKBAR HIDAYATULOH May 23, 2018 at 2:41 p

this post is very helpful for my final project to g

i have question, after done with train/test split, and nex
to use all of the dataset for data train no need to split a
configurations from train/test split or cross validation be

Thank you very much

Jason Brownlee May 23, 2018 at 2:43 pm #

I’m glad to hear that.

Correct, you would use all available data with hyperparameters chosen via testing on your train/test/validation sets.

Debasish Ghosh June 10, 2018 at 1:25 am # REPLY

Thanks Jason for the great post. I have one question though ..

During training I pre-process data e.g. scaling, feature reengineering etc. And then I train the model using train / validation /test set. Now I have the final model which I would like to use for prediction.

Now my prediction system is different (written using Java and TF) and there I import the trained model – incidentally all my training code are in Keras and Python. But in my prediction system I get the data points one at a time and I have to do prediction.

My question is how can I do the data pre-processing during prediction ? Pre-processing like scaling and feature extraction do not make sense on a single data point. With my use case prediction looks good if I accumulate all the data that I receive (unseen before), do similar pre-processing as in training, once I have quite a bit of them and then submit to the trained model for scoring. Otherwise I get very different and inaccurate results.

Would love to hear some suggestions on how to tackle

Jason Brownlee June 10, 2018 at 6:05 am # REPLY ↩

Excellent question!

The single data point must be prepared using the same methods used to prepare the training data.

Specifically, the coefficients for scaling (min/max or mean/stdev) are calculated on the training dataset, used to scale the training dataset, then used going forward to scale any points that you are predicting.

Does that help?

Maria June 25, 2018 at 11:26 pm #

Hi Jason, Thank you for the awesome tutorial
As I see, you emphasize on training a neural network c
the whole dataset as a Test data set in order to train ar
I have already trained cnn_model on the entire data se
set)) but I separate 20% of whole data set as the valida

'model_cnn.fit(x_datasetpad, y_datasetpad, validation_

I think I made a mistake about putting ((validation_split
Do I remove validation set to finalizing the cnn network??
Should I train a network on the entire data set { I mean Should I delete validation_split=0.2???}

Jason Brownlee June 26, 2018 at 6:38 am # REPLY ↩

Yes, remove the validation split for the final model.

Maria June 26, 2018 at 7:54 am # REPLY ↩

Hi Jason,
I am so grateful for the quick answer.

Jason Brownlee June 26, 2018 at 2:26 pm # REPLY ↩

No problem.

K.D.I. Madhuwantha July 4, 2018 at 10:02 pm # REPLY

How to save final model in Tenorflow and use it in Tenorflow.js

Jason Brownlee July 5, 2018 at 7:43 am # REPLY

Sorry, I don't have tensorflow or tensorflow.js examples.

Vaddi Ajay Kumar July 4, 2018 at 11:54 pm # REPLY

I Read Post, all Questions & answers.So final

Ex: I have Training data 100k values, test data: 50k va

1. We try various models like linear regr, decision tree,
150k values and see what model gives better performa
what algorithm/procedure works best on data .

Ex: Decision Tree.

2. Now let us run K fold validation with 150k values on d
check what value gives better performance measure.

3. we know what model and what hyperparameter "generally" works across the data.

4. Let us use all the data 150K values and train final Decision tree (FDT) with hyperparameter that we selected(which worked best) previously.

As model and hyperparameters are checked previously , the above post believes they will and should works best on the unseen data.

My thoughts: I might take a safer approach at the end by double checking, which means rather than train on all the data that i have i will keep 5% for testing (Unseen Data) and 95% for training.

Thank You for the Great Post . I thought this might help people who are concerned about hyper parameter tuning post model/ML procedure selection.

Jason Brownlee July 5, 2018 at 7:47 am # REPLY

All good except the final check is redundant and could be misleading. What if skill on the 5% is poor, what do you do and why?

Vaddi Ajay Kumar July 5, 2018 at 10:02 REPLY

I thought to keep 5% as a double check but after your question i began to ponder what if skill is poor – I have two things to say.

1. This 5% is a sample that is not representative of data . i.e.. Occurred by chance. So i should have other approach to test on representative of the data.

2. Model is not good enough or over-fitted – Even this time i cannot come to conclusion as 5% sample may not be representative of data.

Understood finally that Cross Fold validation is solution for above 2 points which we already did on the whole data prior and so " final check is redundant".

Thank You So much Jason Brownlee.

Jason Brownlee July 6, 2018 at 6:4

Nice reasoning!

Keep an open mind and adapt methods fo empirical discipline.

Vaddi Ajay Kumar July 6, 201

Thank You. Understood.

tranquil.coder October 8, 2019 at 7:14 pm #

REPLY

In Vaddi Ajay Kumar's step 1 and step 2 with cross validation, different algorithms(step 1) and different hyper-parameters (step 2) with different data. Questtons are:

1.Why not just use train/test split method?at least ensure use the same train data and test data. What is the advantage of cross validation against train/test method?

2.I know K shoud not be too small or too big, Some books recommend 10-fold (no mention how much optional values), then how to choose train/test in 10-fold cross validation (given each part named 0,1,2…9) if the optional hyper-parameter has only 5 values according to prior? If it is not right, how to choose K if the optional hyper-parameter has only 5 values according to prior? and 20 values?

Jason Brownlee October 9, 2019 at 8:09 am #

REPLY

K for k-CV is unrelated to the hyperparameters

CV or repeated CV give a less biased estimate of model skill than a single train/test split. K=10 has been found to be a good trade-off (less optimistic) when tested by many people on many differently sized dataset.

Luca October 10, 2019 at 10:11 am # REPLY

@Vaddi Ajay Kumar, why did you split your data in train and test set if you always use 150k that is the sum of the two to do your computation? For what do you use the test set? Thanks.

Ed O July 9, 2018 at 10:06 pm #

Thank you Jason. I am trying to get probabiliti
with the company. I have 1,500 records of individuals t
us. I need to get probabilities for all 500 associates tha

The issue is that the model is technically seeing all the
for the entire data set. I don't have "new" data I can ap
employees that are currently with us. How do I get prol
overfitting? Is it as simple as making the predictors mo

Jason Brownlee July 10, 2018 at 6:47 am # REPLY

You can fit models on some of your data and evaluate it on the rest.

Once you find a model that works, you can train it on all of your data and use it to make predictions on new data.

I assume you have historical records for people that have stayed or left, you train on that. Then you have people now that you want to know if they will stay or leave, this is new data for which you want to generate a prediction.

Thusitha Deepal July 15, 2018 at 2:59 am # REPLY

I have as ome problem.I am trying to predict currency exchange rte using historical data.I'm trying to predict tomorrow exchange rate using yesterday rate..I am littlebit confused.What artifical neural network should i used?And ilike to use K-fold cross validation to sampling..I like to know your ideas.

Jason Brownlee July 15, 2018 at 6:18 am # REPLY

Don’t waste your time:
https://machinelearningmastery.com/faq/single-faq/can-you-help-me-with-machine-learning-for-finance-or-the-stock-market

Luv Suneja July 26, 2018 at 1:12 am # REPLY

Hi Jason,

This is a fantastic article. Cleared a lot of things.

I used to think that number of folds for k-fold is another hyperparameter to find the best model. For example we test two models with k=[3,5,7,9]. But after reading y

validation does not choose a final model anyway.

So, do I just pick a single value of say, k=10 and run w

Thanks
Luv

Jason Brownlee July 26, 2018 at 7:44 am #

Yes.

More details here:
https://machinelearningmastery.com/k-fold-cross-validation/

sarah August 5, 2018 at 3:43 am # REPLY

Hi Jason,
I have become more knowledgeable via your tutorial.
I wanna write a paper so i have to mention a valid resource for this point you mentioned below:
“You finalize a model by applying the chosen machine learning procedure on all of your data.”
in which paper or resource you have seen that we should apply whole data set for training model in order to finalizing it??
please gimme the paper or book as a resource.
waiting for the reply.
Best
Sarah

Jason Brownlee August 5, 2018 at 5:36 am # REPLY

It is tacit knowledge, not written down in a

KALYAN August 11, 2018 at 5:20 am # REPLY

Hello Jason,

How to Re-Train a model with new data which is already trained with another data?

Thanks,
KALYAN.

Jason Brownlee August 11, 2018 at 6:14 am

I have an example here:
https://machinelearningmastery.com/update-lstm-r

Albert Tu August 13, 2018 at 6:25 pm #

Hi Jason,

Great post! Thanks!
A quick question here.
If I have a training dataset of n=500 instances.
Then I use 10-fold cross-validation and feature selection to identify an optimal machine learning algorithms based on this n=500 training dataset.

What would be a reasonable number of instances in the independent testing dataset that I can use to evaluate/test a performance of this machine learning algorithm on the un-seen dataset?

Thanks,
Albert Tu

Jason Brownlee August 14, 2018 at 6:15 am # REPLY

It really depends on the problem.

Perhaps try some different sized dataset and evaluate the stability of their assessment?

Joni August 15, 2018 at 3:15 am # REPLY

Hello Jason,

thank you very much for your deep and crystal clear ex

I have one question about two modelling Setups:

Setup I:

– Split the entire data into a Training-Set (80%) and Test-Set (20%)
– Make a 10-Fold Cross Validation on the Training-Set to find the optimal parameter configuration
– Train the model with the determined parameter configuration on the Trainings-Set
– Final Evaluation of the model on the Testset (“Holdout”, untouched data)

Setup II:

– Split the entire data into a Training-Set (80%) and Test-Set (20%)
– Make a 10-Fold Cross Validation on the entire data set to find the optimal parameter configuration
– Train the model with the determined parameter configuration on the Trainings-Set
– Final Evaluation of the model on the Testset (“Holdou

The only difference between setup I and II is that I mak I
I’m doing it on the entire dataset.

Which setup is do you think better or do you think both

Thanks in advance!
Kind regards from Germany
Jonathan

Jason Brownlee August 15, 2018 at 6:11 am #

There is no notion of better. Use an approach that gives you confidence in the findings of your experiment, enough that you can make decisions.

vamsi August 29, 2018 at 4:25 pm #

REPLY

i would like to know how to measure the performance of the skill of the model using cross validation. since we have k different models how do we measure the performance ? do we get any aggregate score of performance on all models ? please explain me.. to be specific i use h20 to train my model

Jason Brownlee August 30, 2018 at 6:27 am #

REPLY

Average the performance across each model.

More here:
https://machinelearningmastery.com/k-fold-cross-validation/

Steve Tmat September 4, 2018 at 5:40 am # REPLY

Thank you for this very informative post.

Jason Brownlee September 4, 2018 at 6:12 am # REPLY

You're welcome Steve. I'm happy that it helped.

keras_tf September 7, 2018 at 12:19 pm # REPLY

Why do you use the test data as the validatior suppose to have two different test and validation data

Jason Brownlee September 7, 2018 at 1:57 p

Validation is often a subset of train, more https://machinelearningmastery.com/difference-tes

I try to keep my examples simple and often reuse t

John Din September 26, 2018 at 11:53 am # REPLY

Would it help to strengthen the ability of classifier to be trained on various data sets (such as from Kaggle, like stock data, car accidents, crime data etc.)…. While we do so, we may tune underlying algorithms or math construct to deal with different issues such as over-fit, low accuracy, etc.

Jason Brownlee September 26, 2018 at 2:24 pm # REPLY

This approach could be used to learn generic features for a class of problem. E.g. like unsupervised pre-training or an autoencoder.

I think this approach is the future for applied ML. I have a post scheduled on this approach being used in time series forecasting with an LSTM autoencoder. Very exciting stuff.

Harshali October 5, 2018 at 9:56 pm # REPLY

Hey Jason,

Very Very nice article. Archived it for my next project. Thanks for sharing such an informative articles.

Jason Brownlee October 6, 2018 at 5:44 am # REPLY

I'm happy that it helped.

Xu Zhang October 10, 2018 at 9:53 am # REPLY

Thank you so much for your great article.

I understood that we should use all the data which we
stop training when I train my final model with dataset i
learning model. Let me explain it with an example:

I have a CNN model with 100,000 examples. I will do t
1. I split this dataset into training data 80,000, validatio
2. I used my validation dataset to guide my training an
to prevent overfitting.
3. Then I got my best performance and hyperparamete
trained my model 37 epochs, the losses were low and
4 I will finalize my model, train my final model with all n

Here is a problem. Without validation dataset, how can I know when I should stop training, that is how many epochs I should choose when I train my final models. Will I use the same epochs which are used before finalizing the model? or should I match the loss which I got before?

I think for the machine learning models without early stopping training, they are no problems. But like deep learning models, when to stop training is a critical issue. Any advice? Thanks.

Jason Brownlee October 10, 2018 at 3:00 pm # REPLY

Great question. It is really a design decision.

You can try to re-fit on all data, without early stopping by perhaps performing a sensitivity analysis on how many epochs are required on average.

You could sacrifice the a new validation set and refit a new final model on train-test.

There is no single answer, find an approach that makes the most sense for your project and what you know about your model performance and variance.

Xu Zhang October 12, 2018 at 10:00 am # REPLY

Thank you, Jason.
Did you have a chance to read this from Quora?
https://www.quora.com/Should-we-train-neural-networks-on-the-training-set-combined-with-the-validation-set-after-tuning
and also this one
https://stackoverflow.com/questions/39459203/combine-training-data-and-validation-data-how-to-select-hyper-parameters

What are your opinions? Thanks

Jason Brownlee October 12, 2018 at 11:29 am # REPLY

At the end of the day it's a design decisio
problem.

Xu Zhang October 13, 2018 at 3:50 am #

You are surely right! Thank you again

Tayyab October 16, 2018 at 5:21 am # REPLY

Hi Jason Brownlee,

I am reading your tutorials and writing the code the understand the knots and bolts of it. Two reasons I do it. To understand Applied Machine Learning and how scikit-learn works. I had an extensive class on machine learning 1.5 years ago and getting back to my notes I feel like I understand most of the algorithms (I need workout statistics and probability as bit). My Calculus and Linear Algebra is fine. I am working as a student data scientist. My pandas and numpy knowledge would be the same as a beginner and I believe I would do good when using them per need. What would you recommend in such a situation how should I proceed with Data Science? I have my rough route but I would love to hear your comments.

Jason Brownlee October 16, 2018 at 6:40 am # REPLY

Work through small projects and build up your skills.

Taylor November 28, 2018 at 11:58 am # REPLY

Hi Jason, thank you for the wealth of info you'
thing I'm having trouble with is selecting the most impo

I ran 5-fold cross-validation and was able to get feature importance values from each of the 5 models. It is acceptable to then take the average of the 5 importance values for each of the features and use that average to determine the top N features? If I then train a model on those top features and do another 5-fold cross-validation on that model, have I introduced leakage and risk overfitting? Is it better to just take the top N features using the feature importance of just one model from the initial cross-validation? Thanks for any input!

Jason Brownlee November 28, 2018 at 2:53 pm # REPLY ↰

Generally, the random forest will perform the feature selection for you as part of building the model, I would not expect a lift from using feature i

RF.

Njoud December 13, 2018 at 7:13 am #

Thank you, Dr. Jason, for the great website, I

Regarding this article, you said: "Why not keep the bes
You can if you like."

My questions are:

1- in python, how can I save one or the best model returned by cross-validation? as far as I know that the function cross_validation.cross_val_score (from scikit-learn package) don't return a trained model, it just returns the scores? is there another package or function in python that return 10 trained model and I could save one of them?

2- What about R, does cross-validation return models and I can save any one of them? if yes, then how?

I read your articles related to cross-validation, I read books and I searched too, but I did not find answers yet.

some of the links that I find about this issue:
https://stackoverflow.com/questions/32700797/saving-a-cross-validation-trained-model-in-scikit
https://stats.stackexchange.com/questions/52274/how-to-choose-a-predictive-model-after-k-fold-cross-validation

appreciate your help

Jason Brownlee December 13, 2018 at 8:00 am # REPLY ↰

You can step over the folds manually and save the models, for example:
https://machinelearningmastery.com/k-fold-cross-validation/

Paul January 4, 2019 at 4:00 am # REPLY ↰

Hi Jason,
Could you explain how nested CV comes into play here? And where in the process does hyperparameter tuning come into play as well?
Thanks!

Jason Brownlee January 4, 2019 at 6:34 am # REPLY ↰

All models created for evaluating using CV are discarded.

They occur prior to fitting a final model.

Marzi January 15, 2020 at 5:50 am #

Hi Jason,

This post was the best that I found in the Intern

However, I have the same problem. Why do sc

Thank you very much for your informative post

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

Thanks.

CV and nested CV are all used to find a model and set of configurations.

After that you can fit a final model on the chosen model and config.

Magnus January 23, 2019 at 2:14 am # REPLY ↰

Hi Jason,

Good post. So far I have been using the train/validation/test split, and used the validation set to select hyper parameters and avoid overfitting and finally evaluated the model on the test set. Based on this post I guess I can do the same here and train the final model on all the data, or at least on the training set and validation set together? Because I waste a lot of data, not being used for training.

In k-fold CV you mention that overfitting are addressed prior to finalizing the model. Can you elaborate on this? Because there is no validation set to stop the training. I assume you still use the validation set to stop training, for each fold.

Do you have a post where you use both k-fold CV and the train/validation/test split and compare the results and derive a final model? If not, it would be very interesting.

Jason Brownlee January 23, 2019 at 8:50 am # REPLY

Yes. Re-fit using same parameters on all data.

Not sure what you're referring to. CV is for estimating model performance only. If you use early stopping, a validation dataset must be used, even with the final model.

Great suggestion, thanks.

marta January 24, 2019 at 10:12 am #

Dear Jason,
first of all, thank you for helping!
I wonder if you can provide as some references regarc
It would be great!
Bye

Jason Brownlee January 24, 2019 at 1:24 pm # REPLY

It's too practical to be written up in a paper, if that is what you mean. Academics don't talk about a final model for use in operations.

Hussam April 8, 2019 at 7:07 pm # REPLY

Hello Jason,

Thanks for your great blog, it's very helpful.

I have a question about this part "If you train a model on all of the available data, then how do you know how well the model will perform? You have already answered this question using the resampling procedure.""

1.So is that a correct procedure to train the model on all the available dataset and just use resampling method like k-fold to see the accuracy of our mode?

2.If we still split the dataset into train/test sets, test model accuracy and then on the other hand, also use k-fold on the train set. Is there any possibility that we get a big difference between the mean accuracy in K-fold and the model accuracy?

Thanks!

Jason Brownlee April 9, 2019 at 6:22 am # REPLY ↩

Yes.

There could be, it depends on the model and the data. Perhaps test how sensitive your model model is to changes in the size of your dataset.

zzl April 14, 2019 at 11:21 am # REPLY ↩

hi, Jason,

I still wonder about k-fold cross validation. Support I ha into training dataset and test dataest first, and then spl dataset D into k-fold?

Another question is how to report final confuse matrix matrix.

Jason Brownlee April 15, 2019 at 7:50 am #

You can learn more about how k-fold cros
https://machinelearningmastery.com/k-fold-cross-validation/

Confusion matrix can only calculated for a single test set, not cross validation.

PC April 16, 2019 at 11:37 am # REPLY ↩

Hi Jason,
Thanks for a great post.

I need a clarification with the following code.

ensemble = VotingClassifier(estimators=[
(model1), (model2), (model3)], voting='hard')

ensemble = ensemble.fit(X_train, Y_train)

predictions=ensemble.predict(X_validation)

As you have said in this post do I have to discard the X_train and Y_train subsets created using 10-fold Cross validation for fitting the ensemble model for making predictions or is this code correct. Do I need to use the entire dataset for fitting the ensemble model for making prediction.

Kindly help me.

Jason Brownlee April 16, 2019 at 2:25 pm # REPLY

The code appear to define a voting ensemble using 3 models.

It is often a good idea to use different datasets to fit the ensemble vs the submodels.

One approach involves using the out of sample data during cross validation.

Ahmad May 15, 2019 at 1:25 am # REPLY

Great post. It answered all my questions I couldn't find in any other websites. Keep going!

Jason Brownlee May 15, 2019 at 8:18 am #

Thanks, I'm happy it helped!

S May 27, 2019 at 6:16 pm #

Hey Jason, nice article! Do you maybe know

written on this, specifically for time series data?

Jason Brownlee May 28, 2019 at 8:11 am # REPLY

No, it is an engineering consideration. E.g. how to use a model.

Arno Grigorian June 6, 2019 at 9:45 am # REPLY

Hi Jason,

Great post. I'm fairly new to the machine learning and python and still learning. I have built few ML models, but once i retrain the model how do i deploy it.

Like I'm not sure how to proceed in terms of how to save the models, retrain the ML model with entire training set and how to deploy it by code on new/future test data. Greatly appreciated

Jason Brownlee June 6, 2019 at 2:17 pm # REPLY

If you are using sklearn, perhaps this will help:
https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/

Arno Grigorian June 7, 2019 at 7:16 am # REPLY

Jason,
Yes that partially helped. Part where I'm really stuck is, once i train/test my model, cross validate it and decide which model to use.

How do i go back and essentially retrain my model on the entire data (training data set, instead of just on train/test cut of the data)

Jason Brownlee June 7, 2019 at 8:

Collect all of the data into a single

Perhaps I don't understand the difficulty yo

Sofia June 24, 2019 at 6:18 pm #

Thanks for great tutorial, I have a question, I have some pictures with labels, I make 3 copies by adding Gaussian white noise to pictures and 3 copies by making non align pictures, initial dataset was centered on our image, then I shuffle them and split to test and train, accuracy is around one, my question is that good accuracy could be because of overlap in train and test dataset? should I do any refinment? or is ok

Jason Brownlee June 25, 2019 at 6:15 am # REPLY

You cannot have copies of the same image in train and test, it would be an invalid evaluation of the model.

Data augmentation is only used on the training set.

Wu Xie June 27, 2019 at 9:54 am # REPLY

Hi Jason,
Recently I am always reading your excellent posts since I a new ML learner. These posts are clear enough to convey useful information. In recent days, I am confused and struggled to understand how cross-validation works.

I have 500 datasets (may be a little), below is my procedure how I employ ML algorithms. Plz correct me if somewhere is wrong. It is binary classification question.
1. Collect data to form 500 datasets
2. Split the 500 datasets into training (80 %, 400) and test (20 %, 100) datasets
3. Use 10-fold cross-validation to check 9 machine learning algorithms (Logistic Regression, Support Vector Machine, Decision Tree, Random Forest, and AdaBoost) on whole 500 datasets. I find AdaBoost has the highest predictive accuracy from Step 3. This means that AdaBoost has more generalized capacity when applied to my question.
4. Then I use GridSearchCV to find the best hyperparameter for AdaBoost on training (80 %, 400) datasets. Using test (20 %, 100) datasets to test the AdaBoost model with best hyperparameter. I can get the accuracy score, confusion matrix, AUC-ROC, TPR, FPR, ROC curve, PR curve.
My question:
(1) For step 3, it is whole 500 datasets or training (80 %
(2) For step 4, do I need to finalize the model using wh
industrial question.

Jason Brownlee June 27, 2019 at 2:16 pm #

Looks good.

Step 3 would just use the training set, step 4 would

More on train/test/validation datasets here:
https://machinelearningmastery.com/difference-test-validation-datasets/

Krishna July 8, 2019 at 4:13 am #

REPLY

Dear Jason

Request your opinion.

Assume we build an initial Model using,say 1000 features. The output of variables of importance lists only 800 of these 1000 features.

As I understand, while building final Model we would use the same parameters as earlier (e.g. optimal nrounds,etc..), but with complete data. I also use the same random seed as was used to build initial model.

But what about features, should we supply 1000 or 800 features (as obtained above) – while building final model?

Thanks

Jason Brownlee July 8, 2019 at 8:45 am #

REPLY

You would use the same framing of the problem (inputs and outputs) as was used during your experiment.

Smitha Rajagopal July 8, 2019 at 5:38 pm # REPLY

Hi. There is a publicly available dataset with pre-determined train(1,75,341) and test (82,332) splits. The performance of the model is always good with this experimental approach and I got 98% accuracy. However, in order to get a reliable assessment of the model, I combined train+test (2,57,673), then applied 80:20 split with cross validation and performed classification by stacking classifiers which resulted in an accuracy of 86%. Can I infer that although the accuracy is less, this method yields reliable predictions than the first method? Pls clarify.

Jason Brownlee July 9, 2019 at 8:06 am #

Generally, it is a good idea to estimate the
splits, ideally k-fold cross validation.

Salomon July 19, 2019 at 7:26 am #

Thanks Jason very useful information on how to get your final model.
I am working on a churn predictive model. I already got my final model done. Do only thing that is a little confusing is when I use my model to predict with new data, how do I know what time period does predictions are likely to happen? If I use a years accumulation of data to train the model, and 6 months worth of new data to make my predictions. Should I expect my predictions to happen in the next 6 months?
NOTE: all of my predictor variables are averaged per month.
Thanks in advance,

Salomon July 19, 2019 at 7:28 am # REPLY

*those instead of does

Jason Brownlee July 19, 2019 at 9:27 am # REPLY

Not sure I follow, sorry. Can you elaborate?

Salomon July 20, 2019 at 12:33 am #

For example, in my churn project if I am trying to predict clients that are likely to cancel. Lets say the model generated an output of 100 clients that are likely to cancel. Approximately when should I expect those clients to leave the company? Is it possible to know?
Thanks for your attention

Jason Brownlee July 20, 2019 at 10:54 am #

REPLY

It could be a time series classification task.

It could also be a survival analysis:
https://en.wikipedia.org/wiki/Survival_analy

Salomon July 22, 2019 at 12:12 p

Thanks Jason I really appreci

sandipan sarkar July 24, 2019 at 4:25 am #

Hello Jason,
I went through all the 150+ reviews concerning this particular article.Based on the reviews I can easily understand that cross validation is i very important topic.
As far as my understanding goes cross validation is to relate all the independent variables amongst each other to check their importance regards to model building hope I am correct.But this in turn also refers a little bit of multicollinearity. Isnt it???
Can you please clarify.
Thanks.
Best Regards
Sandipan Sarkar

Jason Brownlee July 24, 2019 at 8:09 am #

REPLY

What do you mean exactly? I don't follow sorry.

Bob July 27, 2019 at 9:44 am #

REPLY

Jason,

According to this exchange it claims that you can't report performance metrics of a model trained on a full data set using averages of the K-folds. Wouldn't it be more appropriate to train on the full data set and then have a test data set to report the model performance?
https://stats.stackexchange.com/questions/184095/should-final-production-ready-model-be-trained-on-complete-data-or-just-on-tra

Jason Brownlee July 28, 2019 at 6:37 am # REPLY

Your summary goes against many years of findings, and I'm not interested in debates.

My best advice is to prepare a test harness that give you and project stakeholders confidence in the estimate of model performance on new data.

James MA July 30, 2019 at 1:32 am #

Thank you for this great post, it's very informa
However, I can't imagine how the K trained models be

I have also followed this post: https://machinelearningr

It has mentioned that, "You finalize a model by applyin
your data."

But I'm sorry that, I still have no idea how to finalize the model. Just pick the best model on the K trained model? Combine the result to build up a better model? Or anything else?

Can you give some practical example to finalize a model based on the K-Flod Cross Validation? Thanks.

Jason Brownlee July 30, 2019 at 6:17 am # REPLY

The simple approach is once you have a algorithm and config that is reliable on your test harness/procedures, you use it and fit a model on all available data then use that model to start making predictions on new data.

If you have high variance in the final model (see seen on your test harness), you can reduce this variance by fitting K models on all training data, and using them together as an ensemble when making a prediction on new data.

Does that help?

James MA July 30, 2019 at 8:55 pm # REPLY

Thanks for the explanation.
I just start self-study on machine leanring with python for few days, it seems that I have mixed up something in k -fold Cross Validation.
Before I hit the topic k-fold Cross Validation, and I learn by splitting the known data into training and testing data, and we can estimate the performance by compare the predicated value with testing data.
So, I guess the k-fold Cross Validation is used to find the best model by using different training and testing data group. it seems wrong.
Can I said that, the k-fold cross validation is used to compare between different algorithm / config?
Then based on the score to choose the best algorithm, and build the model using this algorithm.

Thanks a lot.

Jason Brownlee July 31, 2019 at 6

k-fold cross validation is just anot
when making predictions on new data.

Just like a train/test split, but the result is le

You can learn more here:
https://machinelearningmastery.com/k-fold

skyrim4ever August 1, 2019 at 2:58 pm #

REPLY

I'm little confused about k-fold CV in general.

Initially I thought its purpose for evaluation to validate the created model. It was about using one particular ML model and repeat its evaluation multiple times to get multiple different scores in order to get average score. This is good since it avoids a "lucky" situation which happens if model training and evaluation was done once.

But then here k-fold CV is used for finding hyperparameters (such as best ML technique to train model) before doing actual training or evaluation.

Also, in some other sources the k-fold CV is used in whole dataset (training set + test set), in other sources k-fold CV is used only on training set.

I am quite confused… I appreciate if you are able to clarify these things about k-fold CV.

Jason Brownlee August 2, 2019 at 6:42 am #

REPLY

You are correct. In all cases, it is used to estimate the performance of a model on a dataset.
https://machinelearningmastery.com/k-fold-cross-v

It can operate at different scales, across configs, across models, for one model. At some higher orders, we may need to hold back some additional data to validate with in order that we don't overfit the model to the dataset.

You can learn about a validation dataset here:
https://machinelearningmastery.com/difference-test-validation-datasets/

Kenechukwu August 23, 2019 at 12:59 am #

REPLY

Hi,
Thanks for this great post.

I have followed through the whole process of building a
standardization, feature engineering and even saving t
my model to make predictions in the real world, i need
prepared the one for training. Knowing full well that my
training model, how do i standardize my new data acco

Does the standardization save rules for new data? Ho
into the model file because I would need to do some ro
in the training set.

How do I get all these? Does the model save them?

Jason Brownlee August 23, 2019 at 6:33 am #

REPLY

You must use the same coefficients used to prepare the training data or the same objects if that is easier. You may need to pickle the object/coefficients as well.

Coralie August 23, 2019 at 4:59 am #

REPLY

Hi Jason,

Thank you for your post.
I am confused with the step where we deploy the model. Are we keeping the 'old' coefficients that we get from trained data and applying it onto the new data? Or do we re-fit the model (that we conclude is 'best' through validation steps) on the new data?

Jason Brownlee August 23, 2019 at 6:35 am #

REPLY

The coefficients prepared on the training data is the model.

The model is used in a read-only manner to make

Does that help?

Coralie August 23, 2019 at 11:28 pm # REPLY

Thank you! Makes sense.

Jason Brownlee August 24, 2019 at 7:51 am # REPLY

You're welcome.

Lalit August 28, 2019 at 11:22 pm #

Hi Jason,

I have trained a model and save it, but the problem is t
test code not getting the training model classes name.

Jason Brownlee August 29, 2019 at 6:12 am # REPLY

What problem are you having exactly?

Mike September 12, 2019 at 3:02 am # REPLY

Hi Jason:
This series on cross-validation and training a final model have been very enlightening. I found both the information and your consistency in comment responses very helpful in my development of understanding this concept. I think my biggest hang-up was in hearing statements like "once you have a model that you like…" As I now understand CV from your explanation, we are not trying to optimize the result of model training during CV (model parameters, scaling factors, features, etc.), but instead we want to optimize the process that we apply to each fold that gives us the most consistent "score" across all folds. For example, we might try a method of feature selection based each feature's correlation to the predicted result. Applying that to each fold may in-fact result in different features per fold based on that fold's training data. But in the end, we are not interested in the best list of features, but the best method by which we programmatically choose those features based on how it affects the variance of results across all folds. Once we have assembled all of the methods we "like" we then apply those against the entire dataset to produce a trained and deployable model. Am I understanding this correctly?

Jason Brownlee September 12, 2019 at 5:24 am # REPLY ↩

Exactly!!!

We are searching for the process that reliably gives a good model for our data.

This will really help you now – tie it all together:
https://machinelearningmastery.com/applied-machine-learning-as-a-search-problem/

patrick boulay November 10, 2019 at 4:56 am # REPLY ↩

Jason — I emailed you a long question yester
posts. Then I suddenly found the topic elsewhere, esp
reply to my post of yesterday. I now understand that th
yields both a configuration and algorithm that prove eff
application to a new data set presents opportunities fo
the intelligence live," I can suggest this answer: "it lives
through experimentation." Thinks for providing the exc

Jason Brownlee November 10, 2019 at 8:27

Thanks Patrick.

Egor December 23, 2019 at 9:36 am # REPLY ↩

Hi Jason. Thanks for the article.

I recently had the following problem. I have two classifiers – Random Forest and GBoost . I use CV to find the optimal parameters. Initially I used as the final model the models trained only on the training set and I used the accuracy metrics from the test set to assess them (accuracy is around 95%).

Recently after reading your post, I decided to train the final models on the full data set. After doing that I estimate the accuracy of the final models on the full dataset, and while GBoost accuracy is around 95%, I get 100% accuracy for the Random Forest model. In other words, the final RF model classifies all the points in the full data set correctly.

How is this possible? Have you seen something like that?

Thanks,
Egor

Jason Brownlee December 24, 2019 at 6:35 am # REPLY

You cannot estimate the accuracy of the model on data used to train it.

You estimate accuracy using cross validation, then fit a final model. No need to evaluate it again as you already have the estimate.

John December 23, 2019 at 12:34 am # REPLY

Is there any reason not to just ensemble the models trained during cross-validation to produce a final model?

Surely, 10 models each trained with a different 90% of

trained on all of it?

The data I work with is noisy and non-stationary, so I p

my results.

Jason Brownlee December 23, 2019 at 6:53

Off hand: one model is simpler.

Ensemble is good only if there is ROI – e.g. a sufficient lift is skill. This is not always the case, can be neural or even negative.

Reduction in variance is an excellent reason also.

Try it and see!

MAK January 6, 2020 at 9:27 pm # REPLY

Hello Jason Brownlee,

I have developed my own model in robust regression, I also did some programming for my proposed model. But now I want to validate the proposed model by using cross validation, to find the optimal values from a defined grid for the tuning and hyper parameters. If you can provide me any tutorial or codes related to cross validation for our own models which we can do manually, rather than for available models.

Jason Brownlee January 7, 2020 at 7:23 am # REPLY

Perhaps start here:
https://machinelearningmastery.com/faq/single-faq

And here:
https://machinelearningmastery.com/k-fold-cross-validation/

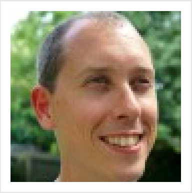
Leave a Reply

Name (required)

Email (will not be published) (requ

Website

SUBMIT COMMENT



Welcome!

My name is *Jason Brownlee* PhD, and I **help developers** get results with **machine learning**.
Read more

Never miss a tutorial:

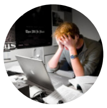
Picked for you:

Your First Deep Learning Project in Python with Keras Step-By-Step

Your First Machine Learning Project in Python Step-By-Step

How to Develop LSTM Models for Time Series Forecasting

Why Machine Learning Does Not Have to Be So Hard



Machine Learning for Programmers

Loving the

The EBook Cat
keep the ***Real***

SEE WHAT

Address: PO Box 206, Vermont Victoria 3133, Australia. | ACN: 626 223 336.
LinkedIn | Twitter | Facebook | Newsletter | RSS

Privacy | Disclaimer | Terms | Contact | Sitemap | Search