Navigation

Start Here Blog Books About Contact

Search...

An Introduction to Feature Selection

by **Jason Brownlee** on October 6, 2014 in **Machine Learning Process**

Which features should you use to create a predictive model?

This is a difficult question that may require deep knowledge of the problem domain.

It is possible to automatically select those features in your data that are most useful or most relevant for the problem you are working on. This is a process called feature selection.

In this post you will discover feature selection, the types of methods that you can use and a handy checklist that you can follow the next time that you need to select features for a machine learning model.

An Introduction to Feature Selection
Photo by John Tann, some rights reserved

What is Feature Selection

Feature selection is also called variable selection or attribute selection.

It is the automatic selection of attributes in your data (such as columns in tabular data) that are most relevant to the predictive modeling problem you are working on.

> *feature selection… is the process of selecting a subset of relevant features for use in model construction*

— Feature Selection, Wikipedia entry.

Feature selection is different from dimensionality reduction. Both methods seek to reduce the number of attributes in the dataset, but a dimensionality reduction method do so by creating new combinations of attributes, where as feature selection methods include and exclude attributes present in the data without changing them.

Examples of dimensionality reduction methods include Principal Component Analysis, Singular Value Decomposition and Sammon's Mappin

> *Feature selection is itself useful, but it mostly acts as a filter, muting out features that aren't useful in addition to your existing features.*

— Robert Neuhaus, in answer to "How valuable do you think feature selection is in machine learning?"

The Problem The Feature Selection Solves

Feature selection methods aid you in your mission to create an accurate predictive model. They help you by choosing features that will give you as good or better accuracy whilst requiring less data.

Feature selection methods can be used to identify and remove unneeded, irrelevant and redundant attributes from data that do not contribute to the accuracy of a predictive model or may in fact decrease the accuracy of the model.

Fewer attributes is desirable because it reduces the complexity of the model, and a simpler model is simpler to understand and explain.

> *The objective of variable selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data.*

— Guyon and Elisseeff in "An Introduction to Variable and Feature Selection" (PDF)

Feature Selection Algorithms

There are three general classes of feature selection algorithms: filter methods, wrapper methods and embedded methods.

Filter Methods

Filter feature selection methods apply a statistical measure to assign a scoring to each feature. The features are ranked by the score and either selected to be kept or removed from the dataset. The methods are often univariate and consider the feature independently, or with regard to the dependent variable.

Some examples of some filter methods include the Chi squared test, information gain and correlation coefficient scores.

Wrapper Methods

Wrapper methods consider the selection of a set of features as a search problem, where different combinations are prepared, evaluated and compared to other combinations. A predictive model us used to evaluate a combination of features and assign a score based on model accuracy.

The search process may be methodical such as a best-first search, it may stochastic such as a random hill-climbing algorithm, or it may use heuristics, like forward and backward passes to add and remove features.

An example if a wrapper method is the recursive feature elimination algorithm.

Embedded Methods

Embedded methods learn which features best contribute to the accuracy of the model while the model is being created. The most common type of embedded feature selection methods are regularization methods.

Regularization methods are also called penalization methods that introduce additional constraints into the optimization of a predictive algorithm (such as a regression algorithm) that bias the model toward lower complexity (fewer coefficients).

Examples of regularization algorithms are the LASSO, Elastic Net and Ridge Regression.

Feature Selection Tutorials and Recipes

We have seen a number of examples of features selection before on this blog.

- **Weka**: For a tutorial showing how to perform feature selection using Weka see "Feature Selection to Improve Accuracy and Decrease Training Time".
- **Scikit-Learn**: For a recipe of Recursive Feature Elimination in Python using scikit-learn, see "Feature Selection in Python with Scikit-Learn".
- **R**: For a recipe of Recursive Feature Elimination using the Caret R package, see "Feature Selection with the Caret R Package"

A Trap When Selecting Features

Feature selection is another key part of the applied machine learning process, like model selection. You cannot fire and forget.

It is important to consider feature selection a , you may inadvertently introduce bias into your models which can result in overfitting.

> *… should do feature selection on a different dataset than you train [your predictive model] on … the effect of not doing this is you will overfit your training data.*

— Ben Allison in answer to “Is using the same data for feature selection and cross-validation biased or not?”

For example, you must include feature selection within the inner-loop when you are using accuracy estimation methods such as cross-validation. This means that feature selection is performed on the prepared fold right before the model is trained. A mistake would be to perform feature selection first to prepare your data, then perform model selection and training on the selected features.

> *If we adopt the proper procedure, and perform feature selection in each fold, there is no longer any information about the held out cases in the choice of features used in that fold.*

— Dikran Marsupial in answer to “Feature selection for final model when performing cross-validation in machine learning”

The reason is that the decisions made to select the features were made on the entire training set, that in turn are passed onto the model. This may cause a mode a model that is enhanced by the selected features over other models being tested to get seemingly better results, when in fact it is biased result.

> *If you perform feature selection on all of the data and then cross-validate, then the test data in each fold of the cross-validation procedure was also used to choose the features and this is what biases the performance analysis.*

— Dikran Marsupial in answer to “Feature selection and cross-validation”

Feature Selection Checklist

Isabelle Guyon and Andre Elisseeff the authors of “An Introduction to Variable and Feature Selection” (PDF) provide an excellent checklist that you can use the next time you need to select data features for you predictive modeling problem.

I have reproduced the salient parts of the checklist here:

1. **Do you have domain knowledge?** If yes, construct a better set of ad hoc”” features
2. **Are your features commensurate?** If no, consider normalizing them.
3. **Do you suspect interdependence of features?** If yes, expand your feature set by constructing conjunctive features or products of features, as much as your computer resources allow you.
4. **Do you need to prune the input variables (e.g. for cost, speed or data understanding reasons)?** If no, construct disjunctive features or weighted sums of feature
5. **Do you need to assess features individually (e.g. to understand their influence on the system or because their number is so large that you need to do a first filtering)?** If yes, use a variable ranking method; else, do it anyway to get baseline results.
6. **Do you need a predictor?** If no, stop
7. **Do you suspect your data is “dirty” (has a few meaningless input patterns and/or noisy outputs or wrong class labels)?** If yes, detect the outlier examples using the top ranking variables obtained in step 5 as representation; check and/or discard them.
8. **Do you know what to try first?** If no, use a linear predictor. Use a forward selection method with the “probe” method as a stopping criterion or use the 0-norm embedded method for comparison, following the ranking of step 5, construct a sequence of predictors of same nature using increasing subsets of features. Can you match or improve performance with a smaller subset? If yes, try a non-linear predictor with that subset.
9. **Do you have new ideas, time, computational resources, and enough examples?** If yes, compare several feature selection methods, including your new idea, correlation coefficients, backward selection and embedded methods. Use linear and non-linear predictors. Select the best approach with model selection
10. **Do you want a stable solution (to improve performance and/or understanding)?** If yes, subsample your data and redo your analysis for several “bootstrap”.

Further Reading

Do you need help with feature selection on a specific platform? Below are some tutorials that can get you started fast:

- How to perform feature selection in Weka (without code)
- How to perform feature selection in Python with scikit-learn
- How to perform feature selection in R with caret

To go deeper into the topic, you could pick up a dedicated book on the topic, such as any of the following:

- Feature Selection for Knowledge Discovery and Data Mining
- Computational Methods of Feature Selection
- Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches
- Subspace, Latent Structure and Feature Selection: Statistical and Optimization Perspectives Workshop
- Feature Extraction, Construction and Selection: A Data Mining Perspective

Feature Selection is a sub-topic of Feature Engineering. You might like to take a deeper look at feature engineering in the post: ”

You might like to take a deeper look at feature engineering in the post:

- Discover Feature Engineering, How to Engineer Features and How to Get Good at It

About Jason Brownlee

Dr. Jason Brownlee is a husband, proud father, academic researcher, author, professional developer and a machine learning practitioner. He is dedicated to helping developers get started and get good at applied machine learning. Learn more.

View all posts by Jason Brownlee →

< Discover the Methodology and Mindset of a Kaggle Master: An Interview with Diogo Ferreira

Building a Production Machine Learning Infrastructure >

78 Responses to *An Introduction to Feature Selection*

Zvi Boger October 2, 2015 at 12:05 am #

REPLY

People can use my automatic feature dimension reduction algorithm published in:

Z. Boger and H. Guterman, Knowledge extraction from artificial neural networks models. Proceedings of the IEEE International Conference on Systems Man and Cybernetics, SMC'97 Orlando, Florida, Oct. 1997, pp. 3030-3035.

or contact me at optimal@peeron.com to get a copy of the paper..

The algorithm analyzes the “activities” of the trained model’s hidden neurons outputs. If a feature dose not contribute to these activities, it either “flat” in the data, or the connection weights assigned to it are too small.

In both cases it can be safely discarded and the ANN retrained with the reduced dimensions.

Jason Brownlee December 29, 2015 at 4:13 pm # REPLY ↩

Thanks for sharing Zvi.

Joseph December 29, 2015 at 2:38 pm # REPLY ↩

Nice Post Jason, This is an eye opener for me and I have been looking for this for quite a while. But my challenge is quite different I think, my dataset is still in raw form and comprises different relational tables. How to select best features and how to form a new matrix for my predictive modelling are the major challenges I am facing.

Thanks

Jason Brownlee December 29, 2015 at 4:12 pm # REPLY ↩

Thanks Joseph.

I wonder if you might get more out of the post on feature engineering (linked above)?

doug February 9, 2016 at 4:22 pm # REPLY ↩

very nice synthesis of some of the ‘primary sources’ out there (Guyon et al) on f/s.

Jason Brownlee July 20, 2016 at 5:26 am # REPLY ↩

Thanks doug.

bura February 9, 2016 at 4:58 pm # REPLY ↩

hello
Can we use selection teqnique for the best features in the dataset that is value numbering?

Jason Brownlee July 20, 2016 at 5:27 am # REPLY ↩

Hi bura, if you mean integer values, then yes you can.

swati March 6, 2016 at 10:07 pm # REPLY ↩

how can chi statistics feature selection algorithm work in data reduction.

Jason Brownlee July 20, 2016 at 5:31 am # REPLY ↩

The calculated chi-squared statistic can be used within a filter selection method.

Poornima July 21, 2017 at 2:40 pm # REPLY ↩

Which is the best tool for chi square feature selection

Jason Brownlee July 22, 2017 at 8:30 am # REPLY ↩

It is supported on most platforms.

Poornima July 24, 2017 at 3:36 pm #

Actually i want to apply Chi square to find the independence between two attributes to find the redundancy between the two. The tools supporting CHI square feature selection onl

attribute and the class attrib e

Chi square to find the level of independence between two attributes….? PS: I cannot use an existing tool…thanks

Jason Brownlee July 25, 2017 at 9:32 am #

Sorry, I don't have the formula at hand. I'd recommend checking a good stats text or perhaps Wikipedia.
https://en.wikipedia.org/wiki/Chi-squared_test

Ralf May 2, 2016 at 5:56 pm #

REPLY ↩

which category does Random Forest's feature importance criterion belong as a feature selection technique?

Jason Brownlee July 20, 2016 at 5:29 am #

REPLY ↩

Great question Ralf.

Relative feature importance scores from RandomForest and Gradient Boosting can be used as within a filter method.

If the scores are normalized between 0-1, a cut-off can be specified for the importance scores when filtering.

swati June 23, 2016 at 10:58 pm #

REPLY ↩

CHI feature selection ALGORITHM IS is NP- HARD OR NP-COMPLETE

Jason Brownlee July 20, 2016 at 5:29 am #

REPLY ↩

I'm not sure swati, but does it matter?

Mohammed AbdelAal June 26, 2016 at 9:53 pm #

REPLY ↩

Hi all,
Thanks Jason Brownlee for this wonderful article.

I have a small question. While performing feature selection inside the inner loop of cross-validation, what if the feature selection method selects NO features?. Do I have to pass all features to the classifier or What??

Jason Brownlee June 27, 2016 at 5:42 am #

REPLY ↩

Good question. If this happens, you will need to have a strategy. Selecting all features sounds like a good one to me.

Dado July 19, 2016 at 10:20 pm # REPLY ↩

Hello Jason!

Great site and great article. I'm confused about how the feature selection methods are categorized though:

Do filter methods always perform ranking? Is it not possible for them to use some sort of subset search strategy such as 'sequential forward selection' or 'best first'?'

Is it not possible for wrapper or embedded methods to perform ranking? For example when I select 'Linear SVM' or "LASSO" as the estimator in sklearns 'SelectFromModel'-function, it seems to me that it examines each feature individually. The documentation doesn't mention anything about a search strategy.

Jason Brownlee July 20, 2016 at 5:34 am # REPLY ↩

Good question Dado.

Feature subsets can be created and evaluated using a technique in the wrapper method, this would not be a filter method.

You can use an embedded within a wrapper method, but I expect the results would be less insightful.

Youssef August 9, 2016 at 7:09 pm # REPLY ↩

Hi, thx all or your sharing
I had a quation about the limitation of these methods in terms of number of features. In my scope we work on small sample size (n=20 to 40) with a lot of features (up to 50)
some people suggested to do all combinations to get high performence in terms of prediction.
what do you think?

Jason Brownlee August 15, 2016 at 11:14 am # REPLY ↩

I think try lots of techniques and see what works best for your problem.

Jarking August 9, 2016 at 9:28 pm # REPLY ↩

hi,I'm now learning feature selection with hierarchical harmony search.but I don't know how to begin with it?could you give me some ideas?

Jason Brownlee August 15, 2016 at 11:15 am # REPLY ↩

Consider starting with some off the shelf techniques first.

L K September 3, 2016 at 3:06 am # REPLY ↩

hi,
i want to use feature extractor for detecting metals in food products through features such as amplitude and phase. Which algorithm or filter will be best suited?

Jason Brownlee September 3, 2016 at 6:59 am # REPLY ↩

Here is a tutorial for feature selection in Python that may give you some ideas:
http://machinelearningmastery.com/feature-selection-machine-learning-python/

laxmi k September 3, 2016 at 2:05 pm # REPLY ↩

I want it in matlab.

Jason Brownlee September 4, 2016 at 5:19 am # REPLY ↩

Sorry, I don't have examples in matlab.

Jaggi September 20, 2016 at 5:53 am # REPLY ↩

Hello Jason,

As per my understanding, when we speak of ‘dimensions’ we are actually referring to features or attributes. Curse of dimensionality is sort of sin where dimensions are too much, may be in tens of thousand and algorithms are not robust enough to handle such high dimensionality i.e. feature space.

To reduce the dimension or features, we use algorithm such as Principle Component Analysis. It creates a combination of existing features which try to explain maximum of variance.

Question: Since, these components are created using existing features and no feature is removed, then how complexity is reduced ? How it is beneficially?
Say, there are 10000 features, and each component i.e. PC1 will be created using these 10000 features. Features didn’t reduced rather a mathematical combination of these features is created.

Without PCA: GoodBye ~ 1*WorkDone + 1*Meeting + 1*MileStoneCompleted
With PCA: Goodbye ~ PC1
PC1=0.7*WorkDone + 0.2*Meeting +0.4*MileStoneCompleted

Jason Brownlee September 20, 2016 at 8:37 am #

REPLY ↰

Yes Jaggi, features are dimensions.

We are compressing the feature space, and some information (that we think we don’t need) is/may be lost.

You do have an interesting point from a linalg perspective, but the ML algorithms are naive in feature space, generally. Deep learning may be different on the other hand, with feature learning. The hidden layers may be doing a PCA-like thing before getting to work.

sai November 13, 2016 at 11:43 pm #

REPLY ↰

Is there any Scope for pursuing PhD in feature selection?

Jason Brownlee November 14, 2016 at 7:43 am #

REPLY ↰

There may be Sai, I would suggest talking to your advisor.

Poornima December 6, 2016 at 6:29 pm

REPLY ↰

What would be the best strategy for feature selection in case of text mining or sentiment analysis to be more specific. The size of feature vector is around 28,000!

Jason Brownlee December 7, 2016 at 8:55 am # REPLY ↩

Sorry Poornima, I don't know. I have not done my homework on feature selection in NLP.

Lekan December 22, 2016 at 6:31 am # REPLY ↩

How many variables or features can we use in feature selection. I am working on features selection using Cuckoo Search algorithm on predicting students academic performance. Kindly assist pls sir.

Jason Brownlee December 22, 2016 at 6:39 am # REPLY ↩

There are no limits beyond your hardware or those of your tools.

Arun January 11, 2017 at 2:21 am # REPLY ↩

can you give some java example code for feature selection using forest optimization algorithm

Jason Brownlee January 11, 2017 at 9:28 am # REPLY ↩

Sorry Arun, I don't have any Java examples.

Amina February 17, 2017 at 4:07 am #

REPLY ↩

Pls is comprehensive measure feature selection also part of the methods of feature selection?

Jason Brownlee February 17, 2017 at 10:01 am #

REPLY ↩

Hi Amina, I've not heard of "comprehensive measure feature selection" but it sounds like a feature selection method.

Birendra February 28, 2017 at 10:06 pm #

REPLY ↩

Hi Jason,
I am new to Machine learning. I applied in sklearn RFE to SVM non linear kernels.
It's giving me error. Is there any way to reduce features in datasets.

Jason Brownlee March 1, 2017 at 8:37 am #

REPLY ↩

Yes, this post describes many ways to reduce the number of features in a dataset.

What is your error exactly? What platform are you using?

Abdel April 6, 2017 at 6:37 am #

REPLY ↩

Hi Jason,

what is the best method between all this methods in prediction problem ??

is LASSO method great for this type of problem ?

Jason Brownlee April 9, 2017 at 2:37 pm # REPLY ↩

I would recommend you try a suite of methods and see what works best on your problem.

Al April 26, 2017 at 6:05 pm # REPLY ↩

Fantastic article Jason, really appreciate this in helping to learn more about feature selection.

If, for example, I have run the below code for feature selection:

test = SelectKBest(score_func=chi2, k=4)
fit = test.fit(X_train, y_train.ravel())

How do I then feed this into my KNN model? Is it simply a case of:

knn = KNeighborsClassifier()
knn.fit(fit) –is this where the feature selection comes in?
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
metric_params=None, n_jobs=1, n_neighbors=5, p=2,
weights='uniform')
predicted = knn.predict(X_test)

Jason Brownlee April 27, 2017 at 8:36 am # REPLY ↩

This post may help:
http://machinelearningmastery.com/feature-selection-machine-learning-python/

Nisha t m May 14, 2017 at 2:21 am # REPLY ↩

Sir,
I have multiple data set. I want to perform LASSO regression for feature selection for each subset. How I get [0,1] vector set as output?

Jason Brownlee May 14, 2017 a REPLY ↩

That really depends on your chosen library or platform.

Simone May 30, 2017 at 6:51 pm #

REPLY

Great post!

If I have well understood step n°8, it' s a good procedure *first* applying a linear predictor, and then use a non-linear predictor with the features found before. Is it correct?

Jason Brownlee June 2, 2017 at 12:34 pm #

REPLY

Try linear and nonlinear algorithms on raw a selected features and double down on what works best.

akram June 10, 2017 at 6:03 am #

REPLY

hello Jason Brownlee and thank you for this post,
i'am working on intrusion detection systems IDS, and i want you to advice me about the best features selection algorithm and why?
thanks in advance.

Jason Brownlee June 10, 2017 at 8:30 am #

REPLY

Sorry intrusion detection is not my area of expertise.

I would recommend going through the literature and compiling a list of common features used.

karthika July 28, 2017 at 6:43 pm #

REPLY

please tell me the evaluation metrics for feature selection algorithms

Jason Brownlee July 29, 2017 a

REPLY

Ultimately the skill of the model in making predictions.

That is the goal of our project after all!

Swati July 31, 2017 at 4:19 am # REPLY ↩

Hi!

I have a set of around 3 million features. I want to apply LASSO for feature selection/dimensionality reduction. How do I do that? I'm using MATLAB.

Swati July 31, 2017 at 4:23 am # REPLY ↩

When I use the LASSO function in MATLAB, I give X (mxn Feature matrix) and Y (nx1 corresponding responses) as inputs, I obtain an nxp matrix as output but I don't know how to exactly utilise this output.

Jason Brownlee July 31, 2017 at 8:21 am # REPLY ↩

Sorry, I cannot help you with the matlab implementations.

Jason Brownlee July 31, 2017 at 8:20 am # REPLY ↩

Perhaps use an off-the-shelf efficient implementation rather than coding it yourself in matlab?

Perhaps Vowpal Wabbit:
https://github.com/JohnLangford/vowpal_wabbit

Swati July 31, 2017 at 3:20 pm # REPLY ↩

Thanks

Elakkiya September 5, 2017 at 8:45 pm #

REPLY ↩

HI..
I need your suggestion on something. just assume i have 3 feature set and three models. for example f1, f2,f3 set. Each set produce different different output result in percentage. i need to assign weight to rank the feature set. any mathematical way to assign weight to the feature set based on three models output?

Jason Brownlee September 7, 2017 at 12:44 pm #

REPLY ↩

Yes, this is what linear machine learning algorithms do, like a regression algorithm.

Elakkiya September 8, 2017 at 3:39 pm #

REPLY ↩

Thank you. Still its difficult to find how regression algorithm will useful to assign weight . Can you suggest any material or link to read…

Jason Brownlee September 9, 2017 at 11:51 am #

REPLY ↩

Search linear regression on this blog.

Marie J October 4, 2017 at 5:18 am #

REPLY ↩

Hi Jason! Thank you for your articles – you've been teaching me a lot over the past few weeks. 🙂

Quick question – what is your experience with the best sample size to train the model? I have 290 features and about 500 rows in my dataset. Would this be considered adequate? Or is the rule of thumb to just try and see how well it performs?

Many thanks!

Jason Brownlee October 4, 201

REPLY ↩

Great question, see this post on the topic:
https://machinelearningmastery.com/much-training-data-required-machine-learning/

gene October 18, 2017 at 6:02 pm # REPLY ↩

Hello Jason,

I am still confused about your point regarding the feature selection integration with model selection. From the first link you suggested, the advice was to take out a portion of the training set to do feature selection on. Next start model selection on the remaining data in the training set

Jason Brownlee October 19, 2017 at 5:34 am # REPLY ↩

See this post on the difference between train/test/validation sets:
https://machinelearningmastery.com/difference-test-validation-datasets/

Does that help?

gene October 22, 2017 at 12:18 am # REPLY ↩

Yes, thanks for this post.
But in practice is there any way to integrate feature selection in model selecction while using GridSearchCV in scikit-learn ?

Jason Brownlee October 22, 2017 at 5:30 am # REPLY ↩

Yes, you could use a Pipeline:
https://machinelearningmastery.com/automate-machine-learning-workflows-pipelines-python-scikit-learn/

gene November 12, 2017 at 8:32 am # REPLY ↩

Hello again!
my feature space is over 8000 attributes. Wh

of feature? By constructing multiple classfiers (NB, SVM, DT) each of which returns different results.

Jason Brownlee November 12, 2017 at 9:11 am #

REPLY

There is no one best set or no one best model, just many options for you to compare. That is the job of applied ML.

Try building a model with each set and see which is more skillful on unseen data.

gene November 12, 2017 at 7:44 pm #

REPLY

I want to publish my results. Is it ok to report that for each model I used a different feature set with a different number of top features?

Jason Brownlee November 13, 2017 at 10:13 am #

REPLY

When reporting results, you should provide enough information so that someone else can reproduce your results.

gene November 13, 2017 at 6:44 pm #

REPLY

Yes I understand that, but I meant does that outcome look reasonable?

Hardik Sahi January 8, 2018 at 12:12 pm #

REPLY

Hi
I am getting a bit confused in the section of applying feature selection in cross validation step.
I understand that we should perform feature selection on a different dataset [let’s call it FS set] than the dataset we use to train the model [call it train set].

I understand the following steps:
1) perform Feature Selection on FS set.
2) Use above selected features on the traini

model.
3) Now, we want to evaluate the performance of the above fitted model on unseen data [out-of-sample data, hence perform CV]

For each fold in CV phase, we have trainSet and ValidSet. Now we have to again perform feature selection for each fold [& get the features which may/ may not be same as features selected in step 1]. For this, I again have to perform Feature selection on a dataset different from the trainSet and ValidSet.

This is performed for all the k folds and the accuracy is averaged out to get the out-of-sample accuracy for the model predicted in step 2.

I have doubts in regards to how is the out-of-sample accuracy (from CV) an indicator of generalization accuracy of model in step 2. Clearly the feature sets used in both steps will be different.

Also, once I have a model from Step 2 with m<p features selected. How will I test it on completely new data [TestData]? (TestData is having p features and the model is trained on data with m features. What happens to the remaining p-m features??)

Thanks

Jason Brownlee January 8, 2018 at 3:54 pm #

REPLY ↩

A simple approach is to use the training data for feature selection.

I would suggest splitting the training data into train and validation. Perform feature selection within each CV fold (automatically).

Once you pick a final model+procedure, fit on the training dataset use the validation dataset as a sanity check.

Leave a Reply

Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT

Welcome to Machine Learning Mastery

Hi, I'm Dr. Jason Brownlee.
My goal is to make practitioners like YOU awesome at applied machine learning.

Read More

You're a Professional (and you need results)!

The field moves quickly...
How Long Can You Afford To Wait?

Take Action Now!

GET THE TRAINING YOU NEED

POPULAR

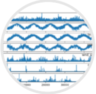
Get Your Start in Machine Learning

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

JUNE 10, 2016

Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras

JULY 21, 2016



Multivariate Time Series Forecasting with LSTMs in Keras

AUGUST 14, 2017

Develop Your First Neural Network in Python With Keras Step-By-Step

MAY 24, 2016

How to Setup a Python Environment for Machine Learning and Deep Learning with Anaconda

MARCH 13, 2017

Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras

JULY 26, 2016

Time Series Forecasting with the Long Short-Term Memory Network in Python

APRIL 7, 2017

Multi-Class Classification Tutorial with the Keras Deep Learning Library

JUNE 2, 2016

Regression Tutorial with the Keras Deep Learning Library in Python

JUNE 9, 2016

How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras

AUGUST 9, 2016

Privacy | Contact | About