Probabilistic Model Selection with AIC, BIC, and MDL

Brownlee on October 30, 2019 in **Probability**

Updated on August 28, 2020

Model selection is the problem of choosing one from among a set of candidate models.

It is common to choose a model that performs the best on a hold-out test dataset or to estimate model performance using a resampling technique, such as k-fold cross-validation.

An alternative approach to model selection involves using probabilistic statistical measures that attempt to quantify both the model performance on the training dataset and the complexity of the model. Examples include the Akaike and Bayesian Information Criterion and the Minimum Description Length.

The benefit of these information criterion statistics is that they do not require a hold-out test set, although a limitation is that they do not take the uncertainty of the models into account and may end-up selecting a simpler model.

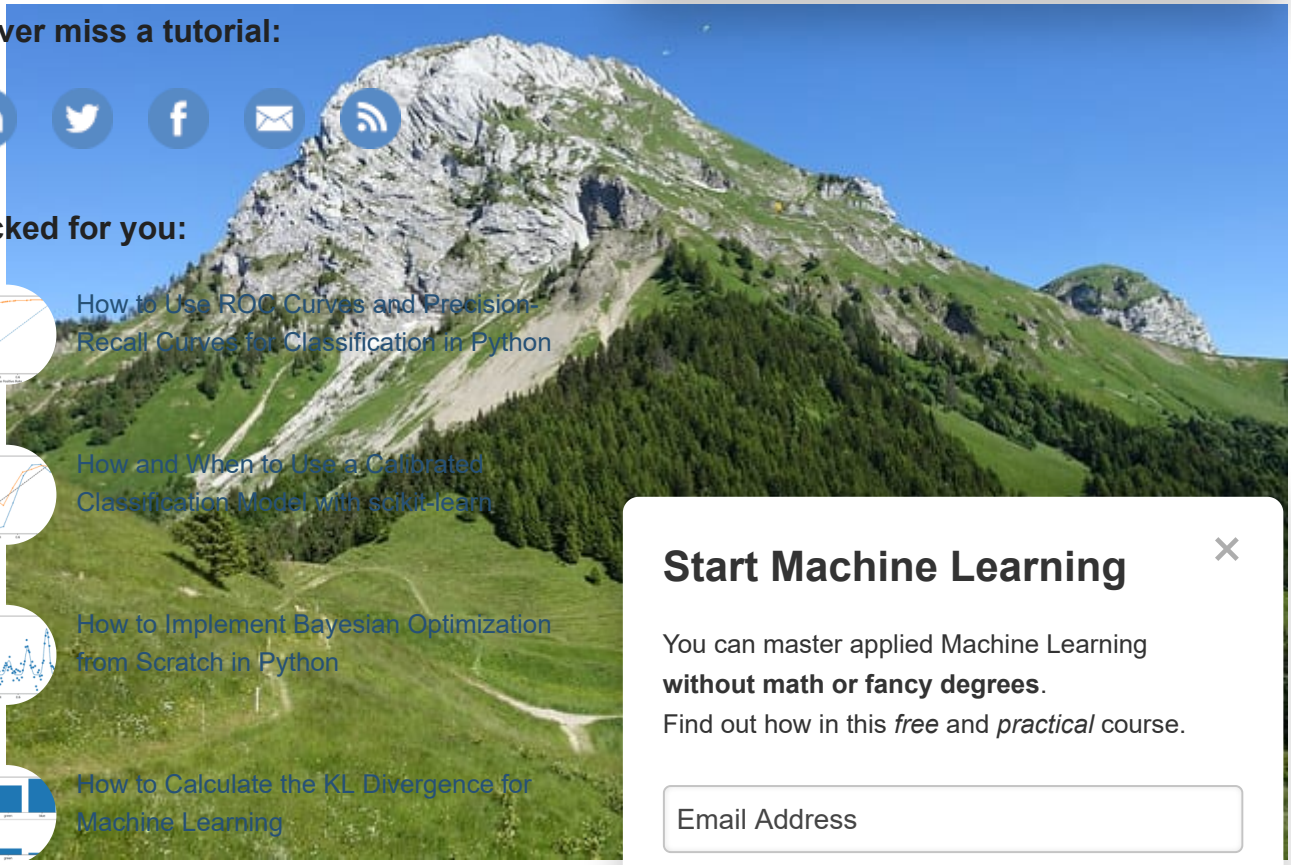
In this post, you will discover probabilistic statistics for machine learning model selection.

After reading this post, you will know:

- Model selection is the challenge of choosing one among a set of candidate models.
- Akaike and Bayesian Information Criterion are two ways of scoring a model based on its log-likelihood and complexity.
- Minimum Description Length provides another scoring method from information theory that can be shown to be equivalent to BIC.

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

Let's get started.



Probabilistic Model Selectio

Photo by Guilhem Ve

Overview

This tutorial is divided into five parts; they are:

1. The Challenge of Model Selection
2. Probabilistic Model Selection
3. Akaike Information Criterion
4. Baye
5. Minimum Description Length

The Challenge of Model Selection

Model selection is the process of fitting multiple models on a given dataset and choosing one over all others.

> *Model selection: estimating the performance of different models in order to choose the best one.*

— Page 222, The Elements of Statistical Learning, 2016.

This may apply in unsupervised learning, e.g. choosing a clustering model, or supervised learning, e.g. choosing a predictive model for a regression or classification task. It may also be a sub-task of modeling, such as feature selection for a given model.

There are many common approaches that may be used for model selection. For example, in the case of supervised learning, the three most common approaches are:

- Train, Validation, and Test Datasets.
- Resampling Methods.
- Probabilistic Statistics.

The simplest reliable method of model selection involves fitting candidate models on a training set, them on the validation dataset, and selecting a model that performs the best on the test dataset according to a chosen metric, such as accuracy or error. A problem with this approach is that it requires a lot of data.

pling techniques attempt to achieve the sam , although using a small dataset. An example is k-fo many train/test pairs and a model is fit and evaluat s selected with the best average score acro ach is that only model performance is assess

A third approach to model selection attempts to co ance of the model into a score, then select

We can refer to this approach as statistical or prob es a probabilistic framework.

Probabilistic Model Selection

Probabilistic model selection (or “information criteria”) provides an analytical technique for scoring and choosing among candidate models.

Models are scored both on their performance on the training dataset and based on the complexity of the model.

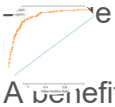
- **Model Performance**. How well a candidate model has performed on the training dataset.
- **Model Complexity**. How complicated the trained candidate model is after training.

Model performance may be evaluated using a prob the framework of maximum likelihood estimation. Mod

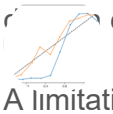
degrees of freedom or parameters in the model.

Historically various 'information criteria' have been proposed that attempt to correct for the bias of maximum likelihood by the addition of a penalty term to compensate for the over-fitting of more complex models.

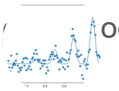
— Page 30, Pattern Recognition and Machine Learning, 2006.



A benefit of probabilistic model selection methods is that a test dataset is not required, meaning that all of the data can be used to fit the model, and the final model that will be used for prediction in the [illegible] can be scored directly.

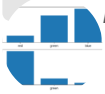


A limitation of probabilistic model selection metho[illegible] calculated across a range of different types of mod[illegible] for [illegible] odel.



It should be noted that the AIC statistic is d[illegible] models (as opposed to comparisons of ma[illegible] How to Use Kernel Density Estimation in Machine Learning

— Page 493, Applied Predictive Modeling, 2013.



Another limitation of these selection methods is th[illegible] to a[illegible]t.

Such criteria do not take account of the uncertainty in the model parameters, however, and in practice they tend to favour overly simple models.

— Page 33, Pattern Recognition and Machine Learning, 2006.

There are [illegible]es to estimating how well a given model fits a dataset and how complex the model is. And each can be shown to be equivalent or proportional to each other, although each was derived from a different framing or field of study.

They are:

- Akaike Information Criterion (AIC). Derived from frequentist probability.
- Bayesian Information Criterion (BIC). Derived from Bayesian probability.
- Minimum Description Length (MDL). Derived from information theory.

Each statistic can be calculated using the log-likelihood for a model and the data. Log-likelihood comes from Maximum Likelihood Estimation, a technique for finding or optimizing the parameters of a model in response to a training dataset.

In Maximum Likelihood Estimation, we wish to maximize the conditional probability of observing the data (*X*) given a specific probability distribution and its parameters (*theta*), stated formally as:

- P(X ; theta)

Where *X* is, in fact, the joint probability distribution of all observations from the problem domain from 1 to n.

- P(x1, x2, x3, …, xn ; theta)

The joint probability distribution can be restated as the multiplication of the conditional probability for observing each example given the distribution parameters. Multiplying many small probabilities together can be unstable; as such, it is common to restate this problem as the sum of the natural log conditional probability.

- sum i to n log(P(xi ; theta))

Given the frequent use of log in the likelihood function, it is commonly referred to as a log-likelihood function.

The log-likelihood function for common predictive … regression (e.g. linear regression) and log loss … logistic regression).

We will take a closer look at each of the three stati…

Akaike Information Criterion

The Akaike Information Criterion, or AIC for short, …

It is named for the developer of the method, Hirotugu Akaike, and may be shown to have a basis in information theory and frequentist-based inference.

> *This is derived from a frequentist framework, and cannot be interpreted as an approximation to the marginal likelihood.*

— Page 1… Probabilistic Perspective, 2012.

The AIC statistic is defined for logistic regression as follows (taken from "The Elements of Statistical Learning"):

- AIC = -2/N * LL + 2 * k/N

Where *N* is the number of examples in the training dataset, *LL* is the log-likelihood of the model on the training dataset, and *k* is the number of parameters in the model.

The score, as defined above, is minimized, e.g. the model with the lowest AIC is selected.

> *To use AIC for model selection, we simply choose the model giving smallest AIC over the set of models considered.*

— Page 231, The Elements of Statistical Learning, 2016.

Compared to the BIC method (below), the AIC statistic penalizes complex models less, meaning that it may put more emphasis on model performance on the training dataset, and, in turn, select more complex models.

> *We see that the penalty for AIC is less than for BIC. This causes AIC to pick more complex models.*

— Page 162, Machine Learning: A Probabilistic Perspective, 2012.

Bayesian Information Criterion

The Bayesian Information Criterion, or BIC for sho

It is named for the field of study from which it was

appropriate for models fit under the maximum

The BIC statistic is calculated for logistic regressio

Learning"):

BIC = -2 * LL + log(N) * k

Where *log()* has the base-e called the natural loga

of examples in the training dataset, and *k*

The score as defined above is minimized, e.g. the model with the lowest BIC is selected.

The quantity calculated is different from AIC, although can be shown to be proportional to the AIC. Unlike the AIC, the BIC penalizes the model more for its complexity, meaning that more complex models will have a worse (larger) score and will, in turn, be less likely to be selected.

> *N[…], this penalizes model complexity more heavily.*

— Page 217, Pattern Recognition and Machine Learning, 2006.

Importantly, the derivation of BIC under the Bayesian probability framework means that if a selection of candidate models includes a true model for the dataset, then the probability that BIC will select the true model increases with the size of the training dataset. This cannot be said for the AIC score.

> *… given a family of models, including the true model, the probability that BIC will select the correct model approaches one as the sample size N -> infinity.*

— Page 235, The Elements of Statistical Learning, 2016.

A downside of BIC is that for smaller, less representative training datasets, it is more likely to choose models that are too simple.

Minimum Description Length

The Minimum Description Length, or MDL for short, is a method for scoring and selecting a model.

It is named for the field of study from which it was derived, namely information theory.

Information theory is concerned with the representation and transmission of information on a noisy channel and as such, measures quantities like entropy, which is the average number of bits required to represent an event from a random variable or probability distribution.

In information theory perspective, we may want to transmit both the predictions (or more precisely, their probability distributions) and the model used to generate them. Both the predicted target variable and the model can be described in terms of the number of bits required to transmit them on a channel.

The Minimum Description Length is the minimum number of bits required to represent the data and t

> *The Minimum Description Length (MDL) pr*
> *minimizes the sum of these two description*

– Page 173, Machine Learning, 1997.

The MDL statistic is calculated as follows (taken fr

DL = L(h) + L(D | h)

Where *h* is the model, *D* is the predictions made by the model, *L(h)* is the number of bits required to represent the model, and *L(D | h)* is the number of bits required to represent the predictions from the model on the training dataset.

The score as defined above is minimized, e.g. the model with the lowest MDL is selected.

The numb ode ($D \mid h$) and the number of bits required to encode (h) can be calculated as the negative log-likelihood; for example (taken from "The Elements of Statistical Learning"):

- MDL = -log(P(theta)) – log(P(y | X, theta))

Or the negative log-likelihood of the model parameters (*theta*) and the negative log-likelihood of the target values (y) given the input values (X) and the model parameters (*theta*).

This desire to minimize the encoding of the model and its predictions is related to the notion of Occam's Razor that seeks the simplest (least complex) explanation: in this context, the least complex model that predicts the target variable.

> *The MDL principle takes the stance that the best theory for a body of data is one that minimizes the size of the theory plus the amount of information necessary to specify the exceptions relative to the theory …*

— Page 198, Data Mining: Practical Machine Learning Tools and Techniques, 4th edition, 2016.

The MDL calculation is very similar to BIC and can be shown to be equivalent in some situations.

> *Hence the BIC criterion, derived as approximation to log-posterior probability, can also be viewed as a device for (approximate) model choice by minimum description length.*

— Page 236, The Elements of Statistical Learning, 2016.

Worked Example for Linear Regression

I make the calculation of AIC and BIC concr

In this section, we will use a test problem and fit a
ne AIC and BIC metrics.

Importantly, the specific functional form of AIC and
been derived, making the example relatively straig
ms, it is important to either find an appropri
ion problem or look into deriving the calcula

In this example, we will use a test regression probl
. The problem will have two input variables
value.

```
...
# generate dataset
X, y = make_regression(n_samples=100, n_features=2, noise=0.1)
# define and fit the model on all data
```

We will fit a LinearRegression() model on the entire dataset directly.

```
...
# define and fit the model on all data
model = LinearRegression()
model.fit(X, y)
```

Once fit, we can report the number of parameters in the model, which, given the definition of the problem, we would expect to be three (two coefficients and one intercept).

```
...
# number of parameters
num_params = len(model.coef_) + 1
print('Number of parameters: %d' % (num_params))
```

The likelihood function for a linear regression model can be shown to be identical to the least squares function; therefore, we can estimate the maximum likelihood of the model via the mean squared error metric.

First, the model can be used to estimate an outcome for each example in the training dataset, then the mean_squared_error() scikit-learn function can be used to calculate the mean squared error for the model.

```
...
# predict the training set
yhat = model.predict(X)
# calculate the error
mse = mean_squared_error(y, yhat)
print('MSE: %.3f' % mse)
```

Tying this all together, the complete example of defining the dataset, fitting the model, and reporting the number of parameters and maximum likelihood estimate of the model is listed below.

```
# generate regression dataset and fit a linear regression model
from sklearn.datasets import make_regression
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
# generate dataset
X, y = make_regression(n_samples=100, n_features=2, noise=0.1)
# define and fit the model on all data
model = LinearRegression()
model.fit(X, y)
# number of parameters
num_params = len(model.coef_) + 1
print('Number of parameters: %d' % (num_pa
# predict the training set
yhat = model.predict(X)
# calculate the error
mse = mean_squared_error(y, yhat)
print('MSE: %.3f' % mse)
```


Note: Your results may vary given the stochastic n
differences in numerical precision. Consider runnir
age

Running the example first reports the number of parameters in the model as 3, as we expected, then reports the MSE as about 0.01.

```
Number of parameters: 3
MSE: 0.010
```

Next, we can adapt the example to calculate the AIC for the model.

Skipping the derivation, the AIC calculation for an ordinary least squares linear regression model can be calculated as follows (taken from "A New Look At The Statistical Identification Model", 1974.):

- AIC = n * LL + 2 * k

Where *n* is the number of examples in the training dataset, *LL* is the log-likelihood for the model using the natural logarithm (e.g. the log of the MSE), and *k* is the number of parameters in the model.

The *calculate_aic()* function below implements this, taking *n*, the raw mean squared error (*mse*), and *k* as arguments.

```
# calculate aic for regression
def calculate_aic(n, mse, num_params):
    aic = n * log(mse) + 2 * num_params
    return aic
```

The example can then be updated to make use of this new function and calculate the AIC for the model.

The complete example is listed below.

```
# calculate akaike information criterion for a linear regression model
from math import log
from sklearn.datasets import make_regression
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# calculate aic for regression
def calculate_aic(n, mse, num_params):
    aic = n * log(mse) + 2 * num_params
    return aic
# generate dataset
X, y = make_regression(n_samples=100, n_features=2, noise=0.1)
# define and fit the model on all data
model = LinearRegression()
model.fit(X, y)
# number of parameters
num_params = len(model.coef_) + 1
print('Number of parameters: %d' % (num_pa
# predict the training set
yhat = model.predict(X)
# calculate the error
mse = mean_squared_error(y, yhat)
print('MSE: %.3f' % mse)
# calculate the aic
aic = calculate_aic(len(y), mse, num_param
print('AIC: %.3f' % aic)
```


Running the example reports the number of param .

Note: Your results may vary given the stochastic n ce in numerical precision. Consider runnin age outcome.

In this case, the AIC is reported to be a value of about -451.616. This value can be minimized in order to choose better models.

```
Number of parameters: 3
MSE: 0.010
AIC: -451.616
```

We can a nple with the calculation of BIC instead of AIC.

Skipping the derivation, the BIC calculation for an ordinary least squares linear regression model can be calculated as follows (taken from here):

- BIC = n * LL + k * log(n)

Where n is the number of examples in the training dataset, *LL* is the log-likelihood for the model using the natural logarithm (e.g. log of the mean squared error), and *k* is the number of parameters in the model, and *log()* is the natural logarithm.

The *calculate_bic()* function below implements this, taking *n*, the raw mean squared error (*mse*), and *k* as arguments.

```
# calculate bic for regression
def calculate_bic(n, mse, num_params):
    bic = n * log(mse) + num_params * log(n)
    return bic
```

The example can then be updated to make use of del.

The complete example is listed below.


```
# calculate bayesian information criterion for a linear regression model
from math import log
from sklearn.datasets import make_regression
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# calculate bic for regression
def calculate_bic(n, mse, num_params):
    bic = n * log(mse) + num_params * log(n)
    return bic

# generate dataset
X, y = make_regression(n_samples=100, n_features=2, noise=0.1)
# define and fit the model on all data
model = LinearRegression()
model.fit(X, y)
# number of parameters
num_params = len(model.coef_) + 1
print('Number of parameters: %d' % (num_params))
# predict the training set
yhat = model.predict(X)
# calculate the error
mse = mean_squared_error(y, yhat)
print('MSE: %.3f' % mse)
# calculate the bic
bic = calculate_bic(len(y), mse, num_params)
print('BIC: %.3f' % bic)
```


Running the example reports the number of parameters, MSE and BIC.

Note: Your results may vary given the stochastic nature of the algorithm or evaluation procedure, or differences in numerical precision. Consider running the example a few times and compare the average outcome.

In this case, the BIC is reported to be a value of about -450.020, which is very close to the AIC value of -451.616. Again, this value can be minimized in order to choose better models.

```
Number of parameters: 3
MSE: 0.010
BIC: -450.020
```

Further Reading

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

Books

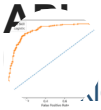
- Chapter 7 Model Assessment and Selection, The Elements of Statistical Learning, 2016.
- Section 1.3 Model Selection, Pattern Recognition and Machine Learning, 2006.
- Section 4.4.1 Model comparison and BIC, Pattern Recognition and Machine Learning, 2006.
- Section 6.6 Minimum Description Length Principle, Machine Learning, 1997.
- Section 5.3.2.4 BIC approximation to log marginal likelihood, Machine Learning: A Probabilistic Perspective, 2012.
- Applied Predictive Modeling, 2013.
- Section 28.3 Minimum description length (MDL), Information Theory, Inference and Learning Algorithms, 2003.

- Section 5.10 The MDL Principle, Data Mining: Practical Machine Learning Tools and Techniques, 4th edition, 2016.

Papers

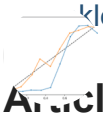
- A New Look At The Statistical Identification Model, 1974.

API



How to Use ROC Curves and Precision-Recall Curves for Classification in Python

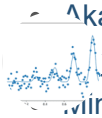
- klearn.datasets.make_regression API.
- sklearn.linear_model.LinearRegression API.
- klearn.metrics.mean_squared_error API.



How and When to Use a Calibrated Classification Model with scikit-learn

Articles

- Akaike information criterion, Wikipedia.
- yesian information criterion, Wikipedia.
- inimum description length, Wikipedia.



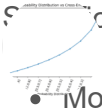
How to Implement Bayesian Optimization from Scratch in Python

How to Calculate the KL Divergence for Machine Learning

Summary

In this post, you discovered probabilistic statistics f

Specifically, you learned:



A Gentle Introduction to Cross-Entropy for Machine Learning

- Model selection is the challenge of choosing one among a set of candidate models.
- Akaike and Bayesian Information Criterion are two ways of scoring a model based on its log-likelihood and complexity.
- Minimum Description Length provides another scoring method from information theory that can be shown to be equivalent to BIC.

Do you have any questions?

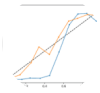
Ask your s below and I will do my best to answer.

On This Topic

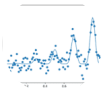
Never miss a tutorial:

Picked for you:

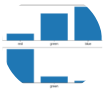
How to Use ROC Curves and Precision-Recall Curves for Classification in Python



How and When to Use a Calibrated Classification Model with scikit-learn



How to Implement Bayesian Optimization from Scratch in Python



How to Calculate the KL Divergence for Machine Learning

A Gentle Introduction to Cross-Entropy for Machine Learning

Loving the Tutorials?

The Probability for Machine Learning EBook is where you'll find the ***Really Good*** stuff.

>> SEE WHAT'S INSIDE

Model Selection Tips From Competitive Machine Learning

A Gentle Introduction to

Feature Selection to Improve Accuracy and Decrease…

Feature Importance and Feature Selection With…

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 →

‹ A Gentle Introduction to Logistic Regression With Maximur

A Gentle Introduction to Expectation-Maximization (EM Algorithm) ›

44 Responses to *Probabilistic Model Selection with AIC, BIC, and MDL*

Elie Kawerk November 1, 2019 at 7:00 am # REPLY

Hi Jason,

Thank you for this nice post!

I'm wondering how to deal with non-parametric mo
strongly on the data (tree-based methods).

Jason Brownlee November 1, 2019 at 1:

Good question.

Typically such measures are challenging to use

Pikachu November 15, 2019 at 12:54 am # REPLY

Hi Jason,

Instead of using mse, I have used log loss value in the equation of AIC and found good result on my datase the value with the AIC value found from statsmodel of python (logit). Here I alue performed better than mse. Can you please explain why has that happened?

Jason Brownlee November 15, 2019 at 7:55 am # REPLY

Log loss is for classification, e.g. logistic regression. The example in the tutorial is linear regression, e.g. predicting a numerical value and log loss is inappropriate.

Pikachu November 18, 2019 at 2:37 pm # REPLY

For classification algorithms (listed below) other than Logistic Regression, should we always use Log Loss for calculating,the AIC?

List of other classification algorithms:
– k Nearest Neighbor

– SVC
– Naive Bayes
– Linear Discriminant Analysis
– Decision Tree
– Random Forest

Jason Brownlee November 19, 2019 at 7:37 am #

REPLY

I don't think so.

Each algorithm will require its own AIC calculation to be derived, at least that is my understanding.

James Bowery January 7, 2020 at 4:22 pm #

Has anyone tried implementing MDL for k

Jason Brownlee January 8, 2020 at 8:20

No, sorry.

Jihoon Jang February 4, 2020 at 3:56 am #

REPLY

Hi, I have a question about parameters.

If I use the number of parameters to calculate AIC ?

Thank you

Jason Brownlee February 4, 2020 at 7:58 am #

REPLY

model.summary() can access the number of parameters.

Jihoon Jang February 4, 2020 at 8:54 pm #

REPLY

Hi Jason ☺

I have a one more question.

As you told me, I just run "model.summary()",

then it said "'MLPRegressor' object has no

How can I fix this problem?

Thank you !

Jason Brownlee February 5, 2020 at 8:07 am # REPLY

Perhaps your model has no layers?

Jihoon Jang February 4, 2020 at 3:59 am # REPLY

Or do I just insert number of hyper-parame

Thank you

Jason Brownlee February 4, 2020 at 7:5

I believe AIC requires a specialized ca

Jihoon Jang February 4, 2020 at 8:27 pm # REPLY

Hi Jason,

Thank you ☺

Is a specialized calculation the number of parameters using model.summary() ?

Jason Brownlee February 5, 2020 at 8:07 am # REPLY

No, I mean you will need to check the literature for how to calculate the metric for an MLP in an appropriate manner.

CMHennings February 28, 2020 at 12:08 am # REPLY

Jason, I'm finding your information and code examples most helpful as I work on my MS degree. Thank you for the time and effort it takes to compose these posts!!

To adapt the linear regression example for logistic regression, the calculation for AIC and BIC (line 9) requires adjustment, correct?

Earlier in this post you define the AIC and BIC calculations for Logistic Regression as:

AIC = -2/N * LL + 2 * k/N
BIC = -2 * LL + log(N) * k

My understanding is line 9 needs replacement with these equations and LL should be replaced with the logistic regression log-likelihood calculation described in your “Gentle Introduction to Logistic Regression” post:

log-likelihood = log(yhat) * y + log(1 – yhat) * (1 – y)

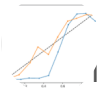
Am I on the right track?

How to Use ROC Curves and Precision-Recall Curves for Classification in Python

Jason Brownlee February 28, 2020 at 6:12 am #

REPLY

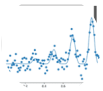
There are many formula for the AIC and BIC metrics.



How and When to Use a Calibrated Classification Model with scikit-learn

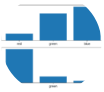
I tried to provide standard calculations and link

approaches that I have seen described.



How to Implement Bayesian Optimization from Scratch in Python

I cannot comment on the best or most correct f



How to Calculate the KL Divergence for Machine Learning

Christian March 4, 2020 at 8:23 am #

El BIC es el mejor criterio para selecciona

A Gentle Introduction to Cross-Entropy for Machine Learning

Jason Brownlee March 4, 2020 at 1:32 pm #

REPLY

It really depends on your project goals. There is never a best anything for all cases.

Jacob March 28, 2020 at 2:22 am #

REPLY

Thank you for the useful article. What I miss is how can MSE stand in place of L despite the fact that a model is better if it has smaller MSE and not larger, like when we deal with L?

Grzegorz Kępisty April 22, 2020 at 12:29 am #

REPLY

Hello again Jason, thank you for good lecture!

Question: Probabilistic model selection include complexity penalty along to error prediction minimization. Ine may ask, why don’t we just focus on test error score? I guess that the reasons for this are:
1) We prefer simple models (easier to interpret)
2) Simpler models normally require less memory, less train/test exectution time.

Is it correct and maybe you can add something to this list?
Regards!

Jason Brownlee April 22, 2020 at 5:58 am #

REPLY

bu re i rtant: simpler explanations are more likely correct and generalize than complex explanations.

Dr. James T. Walker April 22, 2020 at 7:20 pm #

REPLY

Hello: I am an Auxologist trying to develop a model for describing human height growth data from birth to maturity (0 to 21 years). My model assumes that human growth is due to the combination -nine logistic growth components. When I presented the paper at an international conference, I was that I could use the AIC and BIC method for se s needed for these data. Do you agree? Each data s AIC values vs n shows a u-shaped curve, showing n I fit the components to a data set containing) surements), the AIC values and plots change, Can I take this approach? Why does the AIC chang you like to collaborate with me ?

Jason Brownlee April 23, 2020 at 6:01 a

REPLY

Perhaps. Although you are preparing a descriptive rather than predictive model, e.g. statistics rather than machine learning.

The metrics change when the number of elements change because the number of elements impact the complexity of the model.

pm #

REPLY

Hi Jason,

Nicely articulated indeed!
I got some ambiguity here, btw. Computing statsmodels's aic(3026) on sklearn boston dataset is showing different result than this manual aic computation(1565).

Any help would be much appreciated!

Jason Brownlee May 31, 2020 at 6:21 am #

REPLY

Perhaps there is a difference in the implementation.

Nkue June 23, 2020 at 8:56 pm #

REPLY

Hi Jason

Thank you for this blog. Could you please provide R code for the calculation of the MDL for linear regression model.

regards

Nkue

Jason Brownlee June 24, 2020 at 6:31 am #

REPLY

Thanks for the suggestion, perhaps in the future.

Nkue June 25, 2020 at 3:40 pm #

Or Python code

Ghizlan September 22, 2020 at 12:22 am #

Thank you Jason for this post.

I have one question about using AIC to calculate the goodness of fit for neural network and random forest models ? If we can use it, which package to use in R

Best regards

Jason Brownlee September 22, 2020 at 6:49 am #

REPLY

I don't know of an R package off hand, perhaps try a google search.

gizlane September 22, 2020 at 7:34 am #

REPLY

Thank you Jason

Jason Brownlee September 22, 2020 at 7:45 am #

REPLY

You're welcome.

Mansi September 25, 2020 at 5:30 am #

REPLY

Hi Jason.

I've built two models.
Model 1- AIC= 8906
Model 2 AIC= -9501

Is it right to compare negative AIC with positive AIC? Also which AIC above, proves a better model?

Jason Brownlee September 25, 2020 at 6:41 am #

REPLY

Sorry, I don't interpret results.

Matt November 30, 2020 at 1:34 am #

REPLY

H Jason,

Thanks for this great post.

For purposes of calculating BIC for a linear regress
ber of terms (in the example above, 1 for the ir
ude the variance, also being estimated in the fit

I'm wondering because this https://en.wikipedia.org
ns to +1 for the variance.

Jason Brownlee November 30, 2020 at 6:37 am #

REPLY

Perhaps, I based my description on the textbooks listed at the end of the tutorial.

Jon March 14, 2021 at 12:02 pm #

REPLY

How do you calculate AIC and BIC for Logistic Regression Models in Python?

Jason Brownlee March 15, 2021 at 5:51 am #

REPLY

The above calculations will help directly.

Neetika May 26, 2021 at 1:23 am #

REPLY

It is really nicely articulated article. I had initially struggled to understand these concepts, but your article made it crystal clear. I wanted to implement new criteria for model selection via GLM based approach – stepwise forward regression using R or Python. Could you please suggest what parameters I

can consider for defining criteria. Also in case you have sample code for GLM or stepwise forward regression, it would be great help.

Jason Brownlee May 26, 2021 at 5:55 am # REPLY

You're welcome.

Perhaps you can implement the algorithm from a paper or textbook or start with an existing implementation.

Martin Zwanzig June 7, 2021 at 6:12 pm #

Do the methods listed above (make_regre

..not, the following principle is ignored:

"The theory of AIC requires that the log-likelihood h

models not fitted by maximum likelihood, their A

kage 'stats'))

In other words: When linear models are not fitted b

pared by measures such AIC and BIC.

h measures also cannot be used to compare models fitted to a different response (or when the response has been transformed in one case but not the other).

@ Dr. James T. Walker: Also not to the same response considering a different sample size!

Jason Brownlee June 8, 2021 at 7:14 am # REPLY

nes a new regression dataset, learn more here:

https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_regression.html

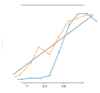
It's just the context for the tutorial.

Leave a Reply

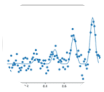
Never miss a tutorial:

Picked for you:

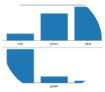
How to Use ROC Curves and Precision-Recall Curves for Classification in Python



How and When to Use a Calibrated Classification Model with scikit-learn



How to Implement Bayesian Optimization from Scratch in Python



How to Calculate the KL Divergence for Machine Learning

A Gentle Introduction to Cross-Entropy for Machine Learning

Start Machine Learning