How can BIC be calculated in Python?

Authored by
stats writer

May 6, 2024

RECOMMENDED CITATION

stats writer (2024). *How can BIC be calculated in Python?*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=143421

BIC (Bayesian Information Criterion) is a statistical measure used for model selection in data analysis. It is calculated by taking into account the likelihood of the data, the complexity of the model, and the sample size. In Python, BIC can be calculated using the formula or by using built-in functions in libraries such as statsmodels. The steps for calculating BIC in Python involve obtaining the log-likelihood of the data, determining the number of parameters in the model, and then using these values to compute the BIC score. This process can be automated by creating a function that takes in the necessary inputs and outputs the BIC score. By implementing BIC calculation in Python, researchers and data analysts can efficiently compare and select the most appropriate model for their data.

Calculate BIC in Python

The Bayesian Information Criterion, often abbreviated BIC, is a metric that is used to compare the goodness of fit of different regression models.

In practice, we fit several regression models to the same dataset and choose the model with the lowest BIC value as the model that best fits the data.

We use the following formula to calculate BIC:

BIC: (RSS+log(n)dσ?2) / n

where:

d: The number of predictorsn: Total observationsσ?: Estimate of the variance of the error associate with each response measurement in a regression

modelRSS: Residual sum of squares of the regression modelTSS: Total sum of squares of the regression model

To calculate the BIC of several regression models in Python, we can use the statsmodels.regression.linear_model.OLS() function, which has a property called bic that tells us the BIC value for a given model.

The following example shows how to use this function to calculate and interpret the BIC for various regression models in Python.

Example: Calculate BIC of Regression Models in Python

Suppose we would like to fit two different using variables from the mtcars dataset.

First, we'll load this dataset:

from sklearn.linear_modelimport LinearRegression
import statsmodels.apias sm
import pandas as pd

#define URL where dataset is located

```
url = "https://raw.githubusercontent.com/Statology/Python-Guides/main/mtcars.csv"

#read in data
data = pd.read_csv(url)

#view head of data
data.head()

model mpg cyl disp hp drat wt qsec vs am gear carb
0 Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4
1 Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4
2 Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1
3 Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1
4 Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2
```


Next, we'll fit the following two regression models:

Model 1: mpg = β0 + β1(disp)+ β2(qsec)Model 2: mpg = β0 + β1(disp)+ β2(wt)

The following code shows how to fit the first model and calculate the BIC:

```
#define response variable
y = data

#define predictor variables
x = data]

#add constant to predictor variables
x = sm.add_constant(x)

#fit regression model
model = sm.OLS(y, x).fit()

#view BIC of model
print(model.bic)

174.23905634994506
```

The BIC of this model turns out to be 174.239.

```
#define response variable
y = data

#define predictor variables
x = data]
```

```
#add constant to predictor variables
x = sm.add_constant(x)

#fit regression model
model = sm.OLS(y, x).fit()

#view BIC of model
print(model.bic)

166.56499196301334
```

The BIC of this model turns out to be 166.565.

Since the second model has a lower BIC value, it is the better fitting model.

Once we've identified this model as the best, we can proceed to fit the model and analyze the results including the R-squared value and the beta coefficients to determine the exact relationship between the set of predictor variables and the .

Two other metrics that are commonly used to compare the fit of regression models are AIC and adjusted R-squared.

The following tutorials explain how to calculate each of these metrics for regression models in Python: