How can the AIC of regression models be calculated using Python?

Authored by

stats writer

April 30, 2024

The AIC (Akaike Information Criterion) is a statistical measure used for model selection and evaluation in regression analysis. In order to calculate the AIC of regression models using Python, there are a few steps that need to be followed. First, the model must be fitted using the appropriate Python library, such as StatsModels or Scikit-learn. Then, the log-likelihood of the model must be calculated using the appropriate function. Finally, the AIC can be computed by adding the double of the number of parameters to the negative two times the log-likelihood. This process can be repeated for multiple models, and the model with the lowest AIC value is considered the best fit for the data. Using Python to calculate the AIC of regression models allows for efficient and accurate model selection and can aid in making informed decisions in data analysis.

Calculate AIC of Regression Models in Python

The Akaike information criterion (AIC) is a metric that is used to compare the fit of different regression models.

It is calculated as:

AIC = 2K - 2*ln*(L)

where:

K: The number of model parameters. The default value of K is 2, so a model with just one predictor variable will have a K value of 2+1 = 3.*ln*(L): The log-likelihood of the model. This tells us how likely the model is, given the data.

The AIC is designed to find the model that explains the most variation in the data, while penalizing for models

that use an excessive number of parameters.

Once you've fit several regression models, you can compare the AIC value of each model. The model with the lowest AIC offers the best fit.

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

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

Example: Calculate & Interpret AIC 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
```

Here are the predictor variables we'll use in each model:

Predictor variables in Model 1: disp, hp, wt, qsecPredictor variables in Model 2: disp, qsec

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

```
#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 AIC of model
print(model.aic)
```

157.06960941462438

Next, we'll fit the second model and calculate the AIC:

```
#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 AIC of model
print(model.aic)

169.84184864154588
```

The AIC of this model turns out to be 169.84.

Since the first model has a lower AIC 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 .