How do you calculate the adjusted R-squared in Python?

Authored by
stats writer

April 22, 2024

RECOMMENDED CITATION

stats writer (2024). *How do you calculate the adjusted R-squared in Python?*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=137990

ARABPSYCHOLOGY.COM

The adjusted R-squared is a statistical measure used to evaluate the goodness of fit of a regression model in Python. It takes into account the number of independent variables included in the model and adjusts the R-squared value accordingly. This helps to prevent overfitting and provides a more accurate representation of the model's performance. To calculate the adjusted R-squared in Python, one can use the built-in function "statsmodels.api.OLS" and then apply the "rsquared_adj" method to the results. This will provide the adjusted R-squared value, which can be interpreted as the percentage of the variation in the dependent variable that is explained by the independent variables in the model.

Calculate Adjusted R-Squared in Python

R-squared, often written R2, is the proportion of the variance in the response variable that can be explained by the predictor variables in a linear regression model.

The value for R-squared can range from 0 to 1. A value of 0 indicates that the response variable cannot be explained by the predictor variable at all while a value of 1 indicates that the response variable can be perfectly explained without error by the predictor variables.

The adjusted R-squared is a modified version of R-squared that adjusts for the number of predictors in a regression model. It is calculated as:

Adjusted R2 = 1 -

where:

ARABPSYCHOLOGY.COM

R2: The R2 of the modeln: The number of observationsk: The number of predictor variables

Since R2 always increases as you add more predictors to a model, adjusted R2 can serve as a metric that tells you how useful a model is, *adjusted for the number of predictors in a model*.

This tutorial shows two examples of how to calculate adjusted R2 for a regression model in Python.

What is a Good R-squared Value?

Example 1: Calculate Adjusted R-Squared with sklearn

The following code shows how to fit a multiple linear regression model and calculate the adjusted R-squared of the model using sklearn:

```
from sklearn.linear_modelimport LinearRegression
import pandas as pd

#define URL where dataset is located
url = "https://raw.githubusercontent.com/Statology/Python-Guides/main/mtcars.csv"
```

ARABPSYCHOLOGY.COM

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

#fit regression model
model = LinearRegression()
X, y = data], data.hp
model.fit(X, y)

#display adjusted R-squared
1 - (1-model.score(X, y))*(len(y)-1)/(len(y)-X.shape-1)

0.7787005290062521
```

The adjusted R-squared of the model turns out to be 0.7787.

Example 2: Calculate Adjusted R-Squared with statsmodels

The following code shows how to fit a multiple linear regression model and calculate the adjusted R-squared of the model using statsmodels:

```
import statsmodels.apias sm
import pandas as pd

#define URL where dataset is located
```

ARABPSYCHOLOGY.COM

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

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

#fit regression model
X, y = data], data.hp
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()

#display adjusted R-squared
print(model.rsquared_adj)

0.7787005290062521
```

The adjusted R-squared of the model turns out to be 0.7787, which matches the result from the previous example.

How to Perform Simple Linear Regression in Python

How to Perform Multiple Linear Regression in Python