How can R-squared be calculated in Python with an example?

Authored by

stats writer

June 29, 2024

R-squared, also known as the coefficient of determination, is a statistical measure that evaluates the goodness of fit of a regression model. In Python, R-squared can be calculated using the "r2_score" function from the "sklearn.metrics" library. This function takes two parameters, the actual values and predicted values, and returns the R-squared value. An example of calculating R-squared in Python would be:

```
import numpy as np
from sklearn.metrics import r2_score

# Actual values
y_true = np.array()

# Predicted values
y_pred = np.array()

# Calculate R-squared
r_squared = r2_score(y_true, y_pred)

print("R-squared: ", r_squared)
```

Output:
R-squared: 0.9872

This value indicates that the regression model has a high goodness of fit, as R-squared ranges from 0 to 1, where 1 indicates a perfect fit and 0 indicates no relationship. Therefore, R-squared calculated in Python can be used to evaluate the accuracy and performance of regression models.

Calculate R-Squared in Python (With Example)

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 where:

0 indicates that the response variable cannot be

explained by the predictor variable at all.1 indicates that the response variable can be perfectly explained without error by the predictor variables.

The following example shows how to calculate R2 for a regression model in Python.

Example: Calculate R-Squared in Python

Suppose we have the following pandas DataFrame:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'hours': ,
'prep_exams': ,
'score': })

#view DataFrame
print(df)
```

hours prep_exams score
0 1 1 76
1 2 3 78
2 2 3 85
3 4 5 88

4 2 2 72

5 1 2 69

6 5 1 94

7 4 1 94

8 2 0 88

9 4 3 92

10 4 4 90

11 3 3 75

12 6 2 96

We can use the LinearRegression() function from to fit a regression model and the score() function to calculate the R-squared value for the model:

```
from sklearn.linear_model import LinearRegression

#initiate linear regression model
model = LinearRegression()

#define predictor and response variables
X, y = df], df.score#fit regression model
model.fit(X, y)

#calculate R-squared of regression model
r_squared = model.score(X, y)
```

```
#view R-squared value
print(r_squared)
```

```
0.7175541714105901
```

The R-squared of the model turns out to be 0.7176.

This means that 71.76% of the variation in the exam scores can be explained by the number of hours studied and the number of prep exams taken.

If we'd like, we could then compare this R-squared value to another regression model with a different set of predictor variables.

In general, models with higher R-squared values are preferred because it means the set of predictor variables in the model is capable of explaining the variation in the response variable well.

Related:**What is a Good R-squared Value?**

Additional Resources

The following tutorials explain how to perform other common operations in Python:

How to Perform Simple Linear Regression in Python

How to Perform Multiple Linear Regression in Python