How do I calculate SST, SSR, and SSE in Python?

Authored by

stats writer

July 1, 2024

Calculating SST, SSR, and SSE in Python involves using statistical functions and formulas to determine the total sum of squares (SST), regression sum of squares (SSR), and error sum of squares (SSE) for a given dataset. SST represents the total variation of the data from the mean, SSR represents the variation explained by the regression model, and SSE represents the unexplained variation or error of the model. These calculations are essential in assessing the effectiveness of a regression model and making predictions. By utilizing built-in functions and libraries in Python, such as numpy and scipy, users can easily compute SST, SSR, and SSE for their data sets and evaluate the accuracy of their regression models.

Calculate SST, SSR, and SSE in Python

We often use three different values to measure how well a fits a dataset:

1. Sum of Squares Total (SST) - The sum of squared differences between individual data points (yi) and the mean of the response variable (y).

SST = Σ(yi - y)2

2. Sum of Squares Regression (SSR) - The sum of squared differences between predicted data points (?i) and the mean of the response variable(y).

SSR = Σ(?i - y)2

3. Sum of Squares Error (SSE) - The sum of squared differences between predicted data points (?i) and observed data points (yi).

SSE = Σ(?i - yi)2

The following step-by-step example shows how to calculate each of these metrics for a given regression model in Python.

Step 1: Create the Data

First, let's create a dataset that contains the number of hours studied and exam score received for 20 different students at a certain university:

```
import pandas as pd

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

#view first five rows of DataFrame
df.head()
```

hours score
0 1 68
1 1 76
2 1 74
3 2 80

4 2 76

Step 2: Fit a Regression Model

Next, we'll use the OLS() function from the library to fit a simple linear regression model using score as the response variable and hours as the predictor variable:

```
import statsmodels.apias sm

#define response variable
y = df

#define predictor variable
x = df]

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

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

Step 3: Calculate SST, SSR, and SSE

Lastly, we can use the following formulas to calculate the SST, SSR, and SSE values of the model:

```
import numpy as np

#calculate sse
sse = np.sum((model.fittedvalues - df.score)**2)
print(sse)
```

331.07488479262696

```
#calculate ssr
ssr = np.sum((model.fittedvalues - df.score.mean())**2)
print(ssr)
```

917.4751152073725

```
#calculate sst
sst = ssr + sse
print(sst)
```

1248.5499999999995

Sum of Squares Total (SST): 1248.55Sum of Squares Regression (SSR): 917.4751Sum of Squares Error (SSE): 331.0749

We can verify that SST = SSR + SSE:

SST = SSR + SSE1248.55 = 917.4751 + 331.0749

Additional Resources

You can use the following calculators to automatically calculate SST, SSR, and SSE for any simple linear regression line:

The following tutorials explain how to calculate SST, SSR, and SSE in other statistical software: