What is the process for testing for multicollinearity in Python?

Authored by

stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *What is the process for testing for multicollinearity in Python?*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=153869

ARABPSYCHOLOGY.COM

The process for testing for multicollinearity in Python involves several steps. First, the data is imported and prepared for analysis. Then, a correlation matrix is created to assess the relationship between the predictor variables. This is followed by calculating the Variance Inflation Factor (VIF) for each variable, which measures the degree of collinearity between the predictors. A VIF value of 10 or above indicates a high level of multicollinearity. Next, a scatterplot matrix can be created to visualize the relationship between the variables. Additionally, statistical tests such as the Eigenvalue test and the Condition Number test can be performed to further assess the presence of multicollinearity. Finally, if multicollinearity is detected, steps such as removing highly correlated variables or using regularization techniques can be taken to address the issue. Overall, the process for testing for multicollinearity in Python involves a combination of statistical tests and visualizations to identify and address any issues with collinearity in the data.

Test for Multicollinearity in Python

In regression analysis, occurs when two or more predictor variables are highly correlated with each other, such that they do not provide unique or independent information in the regression model.

If the degree of correlation is high enough between predictor variables, it can cause problems when fitting and interpreting the regression model.

The most straightforward way to detect multicollinearity in a regression model is by calculating a metric known as the variance inflation factor, often abbreviated VIF.

VIF measures the strength of correlation between predictor variables in a model. It takes on a value

ARABPSYCHOLOGY.COM

between 1 and positive infinity.

We use the following rules of thumb for interpreting VIF values:

VIF = 1: There is no correlation between a given predictor variable and any other predictor variables in the model.VIF between 1 and 5: There is moderate correlation between a given predictor variable and other predictor variables in the model.VIF > 5: There is severe correlation between a given predictor variable and other predictor variables in the model.

The following example shows how to detect multicollinearity in a regression model in Python by calculating VIF values for each predictor variable in the model.

Example: Testing for Multicollinearity in Python

Suppose we have the following pandas DataFrame that contains information about various basketball players:

```
import pandas as pd

#create DataFrame
```

ARABPSYCHOLOGY.COM

```
df = pd.DataFrame({'rating': ,
'points': ,
'assists': ,
'rebounds': })

#view DataFrame
print(df)
```

```
rating points assists rebounds
0 90 25 5 11
1 85 20 7 8
2 82 14 7 10
3 88 16 8 6
4 94 27 5 6
5 90 20 7 9
6 76 12 6 6
7 75 15 9 10
8 87 14 9 10
9 86 19 5 7
```

ARABPSYCHOLOGY.COM

Suppose we would like to fit a using rating as the response variable and points, assists, and rebounds as the predictor variables.

To calculate the VIF for each predictor variable in the

model, we can use the function from the statsmodels library:

```
from patsy import dmatrices
from statsmodels.stats.outliers_influenceimport variance_inflation_factor

#find design matrix for regression model using 'rating' as response variable
y, X = dmatrices('rating ~ points+assists+rebounds', data=df, return_type='dataframe')

#create DataFrame to hold VIF values
vif_df = pd.DataFrame()
vif_df = X.columns#calculate VIF for each predictor variable
vif_df = )]

#view VIF for each predictor variableprint(vif_df)

VIF variable
0 101.258171 Intercept
1 1.763977 points
2 1.959104 assists
3 1.175030 rebounds
```

ARABPSYCHOLOGY.COM

We can see the VIF values for each of the predictor variables:

points: 1.76assists: 1.96rebounds: 1.18

Note: Ignore the VIF for the "Intercept" in the model since this value is irrelevant.

Since each of the VIF values for the predictor variables in the model are close to 1, multicollinearity is not a problem in the model.

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

ARABPSYCHOLOGY.COM