

How can I extract the regression coefficients from a Scikit-Learn model?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I extract the regression coefficients from a Scikit-Learn model?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153515>

To extract the regression coefficients from a Scikit-Learn model, one can use the "coef_" attribute. This attribute returns an array of coefficients corresponding to each feature in the model. These coefficients can then be used to understand the impact of each feature on the target variable and make predictions. Additionally, the "intercept_" attribute can be used to retrieve the intercept value of the model. By accessing and analyzing these attributes, one can gain valuable insights into the relationship between the features and the target variable in a Scikit-Learn regression model.

Extract Regression Coefficients from Scikit-Learn Model

You can use the following basic syntax to extract the regression coefficients from a regression model built with scikit-learn in Python:

```
pd.DataFrame(zip(X.columns, model.coef_))
```

The following example shows how to use this syntax in practice.

Example: Extract Regression Coefficients from Scikit-Learn Model

Suppose we have the following pandas DataFrame that contains information about hours studied, number of prep exams taken, and final exam score received by 11 students in some class:

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'hours': ,  
'exams': ,  
'score': })
```

```
#view DataFrame
```

```
print(df)
```

```
hours 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
```

We can use the following code to fit a using hours and exams as the predictor variables and score as the response variable:

```
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)
```

We can then use the following syntax to extract the regression coefficients for hours and exams:

```
#print regression coefficients
```

```
pd.DataFrame(zip(X.columns, model.coef_))
```

```
0 1
```

```
0 hours 5.794521
```

```
1 exams -1.157647
```

From the output we can see the regression coefficients for both predictor variables in the model:

Coefficient for hours: 5.794521 Coefficient for exams:
-1.157647

If we'd like, we can also use the following syntax to extract the intercept value for the regression model:

```
#print intercept value  
print(model.intercept_)
```

70.48282057040197

Using each of these values, we can write the fitted regression model equation:

$\text{Score} = 70.483 + 5.795(\text{hours}) - 1.158(\text{exams})$

We can then use this equation to predict the final exam score of a student based on their number of hours spent studying and number of prep exams taken.

$\text{Score} = 70.483 + 5.795(\text{hours}) - 1.158(\text{exams})$
 $\text{Score} = 70.483 + 5.795(3) - 1.158(2)$
 $\text{Score} = 85.55$

Related:

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

ARABPSYCHOLOGY.COM