How can I calculate AUC (Area Under Curve) in Python?

Authored by

stats writer

May 7, 2024

Calculating the Area Under Curve (AUC) is a commonly used method for evaluating the performance of a predictive model. It is a measure of the overall performance of a model by quantifying the trade-off between the true positive rate and false positive rate. In Python, AUC can be calculated using the "roc_auc_score" function from the "sklearn.metrics" module. This function takes the true labels and predicted probabilities as inputs and returns the AUC value. Additionally, the "roc_curve" function can be used to plot the Receiver Operating Characteristic (ROC) curve, which is a graphical representation of the AUC calculation. By utilizing these functions, one can easily calculate and visualize the AUC of a model in Python.

Calculate AUC (Area Under Curve) in Python

Logistic Regression is a statistical method that we use to fit a regression model when the response variable is binary.

To assess how well a logistic regression model fits a dataset, we can look at the following two metrics:

Sensitivity: The probability that the model predicts a positive outcome for an observation when indeed the outcome is positive. This is also called the "true positive rate."Specificity: The probability that the model predicts a negative outcome for an observation when indeed the outcome is negative. This is also called the "true negative rate."

One way to visualize these two metrics is by creating a ROC curve, which stands for "receiver operating

characteristic" curve.

This is a plot that displays the sensitivity along the y-axis and (1 - specificity) along the x-axis.

One way to quantify how well the logistic regression model does at classifying data is to calculate AUC, which stands for "area under curve."

The closer the AUC is to 1, the better the model.

The following step-by-step example shows how to calculate AUC for a logistic regression model in Python.

Step 1: Import Packages

First, we'll import the packages necessary to perform logistic regression in Python:

```
import pandas as pd
import numpy as np
from sklearn.model_selectionimport train_test_split
from sklearn.linear_modelimport LogisticRegression
from sklearn import metrics
```

Step 2: Fit the Logistic Regression Model

Next, we'll import a dataset and fit a logistic regression model to it:

```
#import dataset from CSV file on Github
url = "https://raw.githubusercontent.com/Statology/Python-Guides/main/default.csv"
data = pd.read_csv(url)

#define the predictor variables and the response variable
X = data]
y = data

#split the dataset into training (70%) and testing (30%) sets
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=0)

#instantiate the model
log_regression = LogisticRegression()

#fit the model using the training data
log_regression.fit(X_train,y_train)
```

Step 3: Calculate the AUC

We can use the metrics.roc_auc_score() function to calculate the AUC of the model:

```
#use model to predict probability that given y value is 1
y_pred_proba = log_regression.predict_proba(X_test)

#calculate AUC of model
auc = metrics.roc_auc_score(y_test, y_pred_proba)

#print AUC score
print(auc)

0.5602104030579559
```

Recall that a model with an AUC score of 0.5 is no better than a model that performs random guessing.

Thus, in most cases a model with an AUC score of 0.5602 would be considered poor at classifying observations into the correct classes.

The following tutorials offer additional information about ROC curves and AUC scores: