Python Lab - SMS Spam Ham Classifier : Multinomial

We had mentioned earlier that Naive Bayes finds applications in spam detection. Let’s now see a full-fledged implementation of the Naive Bayes classifier in python on the ‘SMS dataset’.

We’ll build an SMS spam classifier and compare the results of Multinomial NB and Bernoulli NB. For detailed information about the dataset used you can refer to the link: **SMS dataset**

Please find the Spam Ham dataset here and the code file here

Live Help

In the last lecture, we pre-processed our real SMS: Spam/Ham classifier data. Now in the next lecture, we will be building and evaluating the model based on Specificity, Sensitivity and ROC curve.

Let’s understand them in detail.

Sensitivity implies that out of all the actual Spam's, how many of them were correctly predicted by the model as Spam.

Specificity implies that out of all the genuine SMS's, how many of them were correctly predicted as legitimate by the model.

If we look back to the problem statement, We are fine if some of the Spams are classified as Hams, but it will not be good if an important Ham is classified as Spam by our model. So our motive is to maximise Specificity.

ROC Curves are used to see how well your classifier can separate positive and negative examples and to identify the best threshold for separating them.

The area under the ROC Curve shows how far the curve from the base line. For the baseline, it's 0.5, and for the perfect classifier, it's 1. In our case the AUC obtained was 99% which is very good.

You could read about ROC curve from the link.

Report an error

PREVIOUS
Python Lab - Education Or Cinema ?

NEXT
Python Lab - SMS Spam Ham Classifier : Bernoulli