Greg Witt

Machine Learning Winter 2024

Program #2

The process of building a Naïve Bayes classifier was done within the Main method of the program attached, The “buildTrainingTestData()” method takes the original dataset of spam and ham emails. and then splits it into trainingSpambaseData and testingSpamBaseData both of which have a close split of 60% Non-Spam (Ham) and 40% Spam email features and labels. The Classifier class “emailClassifier” is then built. The classifier has a couple constructor methods which will pull the mean and standard deviation for each of the training data features for Hams (Non-Spam) and Spams, are which they’re established on the class properties. The next process is to call the “prepareTestData()” method which takes the target labels and feature vectors and splits them into separate entities for the classification process of the model. The *emailClassifier* object’s “classifyEmails()” is then called to classify the emails and pull attributes of true positive, true negative, false positive and false negative as well as the values to build the confusion matrix after the classification process has been completed all while looping through the supplied data.

The classifier was very accurate when implementing the Gaussian Naïve bayes Algorithm. The resulting accuracies, precision, and recall can be seen below.

A screenshot of a computer screen

Description automatically generated

With this small number of instances (2273) in total for the test set, the classifier performed very well. Assuming the independence of feature correlations the Naïve bayes classifier was able to perform well into the 90 and high 80 percentages for Accuracy. Recall and precision were also within the same metric range as well. Below is the confusion matrix for this same execution of the email Classifier against the test data.

A graph showing a number of spam

Description automatically generated

It can be said that for the most part the assumption of independence between these features of words with specific values is highly unlikely in the real world. Words such as “Free”, “Confirmation” and others would most likely be correlated to a spam emails. Being the naïve approach that this classifier takes the data was classified very well. The assumption of independence has a strength within this model. Upon further inspection of the features with the pandas “corr()” function on the test set further below in the main method of the program. There appears to be no correlation of these features amongst visually inspecting the off-diagonal matrix returned from the call. Each of the values in this returned matrix correspond with the correlation of the given feature within the data. Each value is below 0 or at -0.0X ranges indicating a negative correlation or no correlation. This can be seen in the image below.

A black background with numbers

Description automatically generated

But overall, the assumption of independence was perfectly adequate to classify these emails and prove the power of this model. The model might struggle if there is a significantly larger portion of data in which the results might suffer. But overall, the performance is excellent. And despite its potential to perform poorly with larger data that might not be classified. I would count this as a solid classifier.