Udacity Machine Learning Nanodegree

Capstone Proposal

Spam Email Detection

Gaurav Jain
February 2020

- ### Domain Background

  Nowadays, along with desired emails we get a lot of unsolicited emails like promotional, social, and other secondary types. Hence it becomes hard to distinguish between spam and genuine emails and create a lot of noise in our inbox.

- ### Problem Statement

  The problem is to create or use the existing machine learning model to predict whether an email is spam or not.

- ### Datasets and Inputs

  We'll take 30228_easy_ham.tar.bz2 and 30228_spam.tar.bz2 from Apache Spam Assassin's public datasets https://spamassassin.apache.org/old/publiccorpus/.
  There are approximately 2500 ham and 500 spam emails in the dataset. Compressed dataset's approx size is 3 MB so I'll attach with the final source code as well. We'll create a final dataset for training and testing using the above dataset, and that final set will have two columns that will be fed to the model.

- ### Solution Statement

  We'll parse the data, sanitize/tokenize it and then we'll apply LogisticRegression on it. We'll also make use of hyperparameters to tune the model and get the best result.

- ## Benchmark Model

  We have a testing set that we can use to verify our trained model and check the accuracy of the result.

- ## Evaluation Metrics

  Evaluation metrics would simply be the the final accuracy result. If the number is close to 1 then higher the chance of it being spam mail.

- ## Project Design

  We will simplify the data set by removing headers/metadata, urls and other unnecessary information. Once we have the sanitized data, we'll have two columns in the final dataset. Column1 would be the actual email message and column2 would be a label that represents the email as spam or not.
  We'll split the data in training and testing set 70%, 30% respectively.
  After doing this we'll have a clear data structure defined for data and label for both training and testing set. We'll use the LogisticRegression algorithm which seems to be best for this problem. We'll play with different hyperparameters to tune the model and submit the one that performs best.

- ## References

  https://en.wikipedia.org/wiki/Linear_regression

  https://en.wikipedia.org/wiki/Binary_classification

  https://en.wikipedia.org/wiki/Supervised_learning