Spam Recognition using Machine Learning

Lucas Hyatt - llh@uoregon.edu Olivia Pannell - olp@uoregon.edu

ABSTRACT

While some spam can be harmless, there is an overwhelming security risk associated with malicious attackers. These attackers can cause unwanted behavior through hidden links and misleading content. Identifying malicious messages is an ongoing battle within cyber security. Our objective is to create a functional and accurate classification system in which the user can detect if a message is spam or ham. This system will rely on multiple machine learning models for classification and prediction.

FEATURE EXTRACTION

{ ‘free’ : 5, ‘money’ : 2, **‘or’** : 24, … , ‘buy’ : 1 }

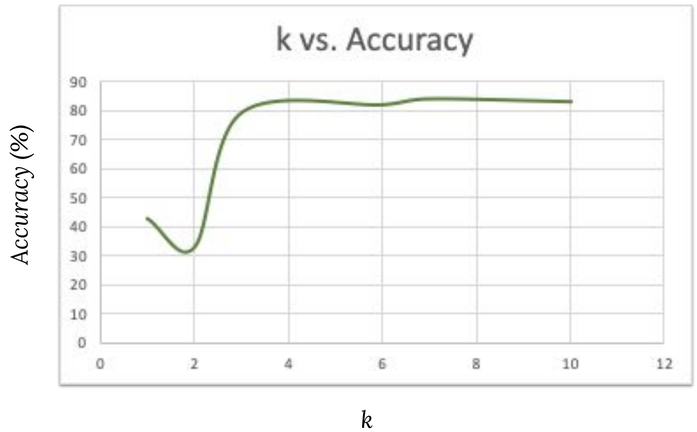
{ ‘free’ : 5, ‘money’ : 2, … , ‘buy’ : 1 }

< 1, 1, 1, 0, 0, 0, 1, … 0, 1 >

The goal of *feature extraction* is to have tangible data which we can use to train a model. Extracting features from spam messages begins by taking an email (spam or ham), and counting the occurence of each word and placing the result in a dictionary. We then remove useless training words also known as “stop words” such as ‘and’, ‘or’, and ‘the’. Finally, we convert the dictionary to a vector, where the occurrences or words represent the weights.

K-NN MODEL

The *K-Nearest Neighbor* model is an algorithm which observes available cases in order to classify new cases based on similarity. We used this model to train a dataset of roughly 80% ham emails and 20% spam, extracting features and creating a feature matrix including the feature vectors for each email. The model relies on this feature matrix in order to compare the feature vectors for test emails to the *k nearest* feature vectors in the training matrix using euclidean distance. This model identifies spam with an ~84% accuracy.



PERCEPTRON MODEL

The perceptron algorithm is an algorithm for supervised learning of binary classifiers. In this sense, our binary classifier is a function predicting whether an email is spam (1) or ham (0). The perceptron model maps input feature vectors to binary outputs, using a set of weights (w) and a bias (b). Together, w and b form a decision boundary. This boundary can be used to distinguish linearly separable data, such as spam and ham emails within a dataset. The algorithm can be adjusted via a hyperparameter called *max_iter* which represents the maximum number of iterations allowed for convergence of the perceptron algorithm to occur. The following function is used to map a feature vector x to a single binary output:

$$f(x) = 1 \text{ if } w * x + b > 0, \text{ or } 0 \text{ otherwise.}$$