K-nearest-neighbors is a simple classifier that can be considered as an important building block of many machine learning algorithms. This algorithm classifies an unknown example with the most common class among *k* closest examples. This means that, using an increasing size of training set data, the algorithm will assign a class to a value depending on its k nearest training data points in Euclidean space, where k is some number chosen by the user. In the figure 1.1, it shows an illustration as to how this algorithm is performed. The chosen number by the user for “*k*”, in this case, is three. It looks for the closest neighbors to determine the class of ‘c’. In this case, the closest neighbors are two elements of ‘o’ and one element of ‘a.’ Since the number of ‘o’ elements is greater than the number of ‘a’ elements, the class of ‘c’ is identified to be ‘o.’ The algorithm is quite simple and powerful, and no training is actually involved as new training examples can be easily be added.

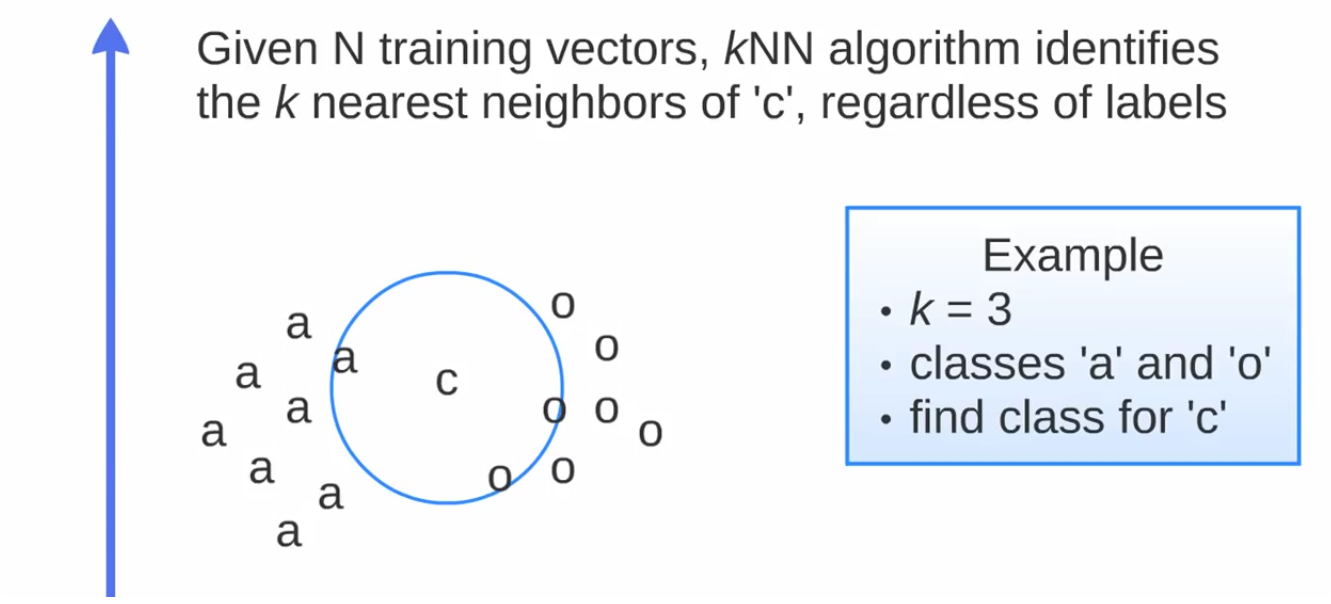


Figure 1.1

Support Vector Machine (SVM) is another supervised machine learning algorithm used for classification and regression practices by a separated hyperplane. This means that it collects a training set and the program categorizes new examples from that training set by an optimal hyperplane; which the program outputs. In the figure 2.1, it shows an illustration as to how this algorithm is performed. The training set consists of a set of circle and square classes. The green and red lines are known to be the hyper-plane and it segregates the two classes through distance. The distance from the hyper-plane and the element is known to be the margin. The goal of using SVM is it to find an optimal boundary between the possible classes provided**.** A hyperplane that passes too close to the classes is not is not recommended because it will be noise sensitive and will not generalize correctly. Therefore, the hyperplane of the line is best if it is passed between the classes as far as possible.

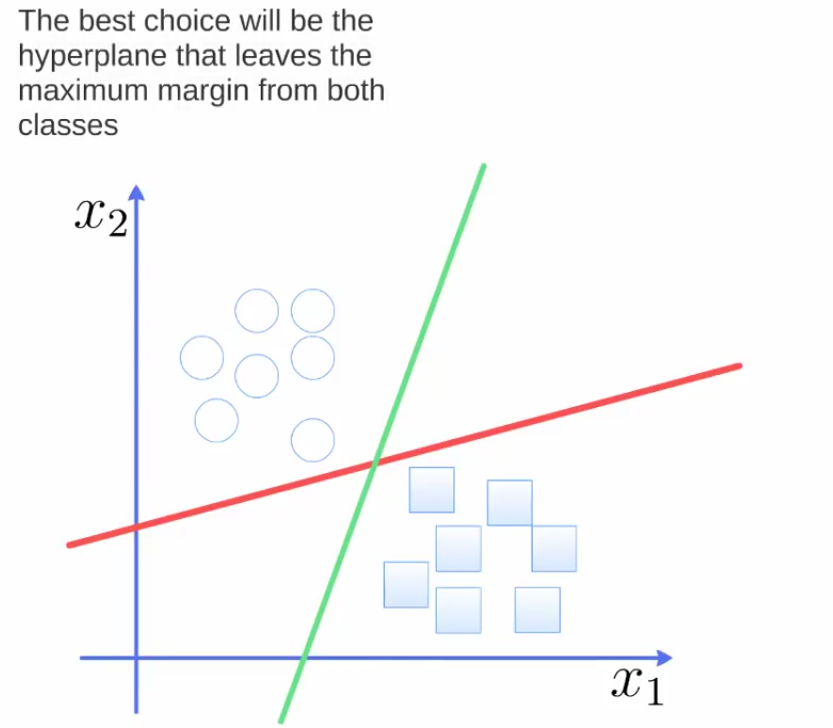


Figure 2.1

Naive Bayes is another supervised machine learning algorithm that makes use of the Bayes Theorem. The algorithm is performed by predicting the membership probabilities for each class. This involves the probability that given record or data point belongs to, in a particular class. In figure 3.1, it presents the formula used in this algorithm. The formula functions on how the features of each class are “generated.” The algorithm uses the training sets, and to implement the training set into the formula, it consists of the target value (hypothesis) and each attribute value at of each datum instance.

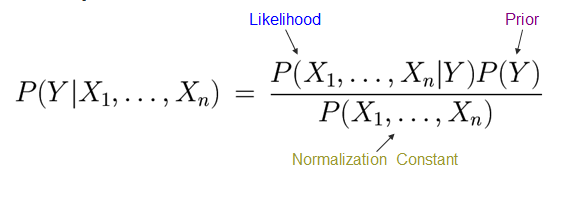


Figure 3.1

Maximum A Posteriori (MAP) is finding the class with the highest probability. This is considered to be the best class. All features in the naive bayes algorithm is considered independent; which is known to be the Naïve Bayes assumption. In figure 3.2, the classifier is often represented as the graph presented. In the graph, the direction of the arrows clarify that each class causes certain features, with a certain probability.

Figure 3.2

Parvaneh used a tool she found online where she can upload the privacy policy document. In it, it would create an output of each word in a sentence. Her goal is to determine whether each sentence is considered sensitive or non-sensitive.

She wanted to classify each sentence based on the presence or absence of each word. Again, her goal is to determine whether each sentence is considered sensitive or non-sensitive. For example, there was a sentence in one of the documents which included the terms “access” “store” “information.” The program will output a file in excel where each word is arranged in a vector with either a number of a zero or a one. In this case, the terms, “access” “store” “information,” will have a one at the bottom of the cell as it clarifies that it is a sensitive word. Other terms, for example “process,” was not considered a sensitive word and thus will have a zero at the bottom of the cell.

Using the three algorithms perspectives, each vector of 1 and 0’s will then calculate the distance between each vector. It then checks if they are similar; either sensitive or non-sensitive. For example, one sentence had ten neighbors (words). Seven of them are sensitive while three of them are non-sensitive. Thus, the result is that the sentence is sensitive. Using the algorithm, it calculated that it is 0.7 sensitive and 0.3 non - sensitive; making it a sensitive sentence.

The next step is to break each privacy policy documents, based on its topic. For example, this sentence is about collection while the other is about sharing….