Social media has led to an explosion of varied kinds of information. This is due to the nature of microblogs on which people are able to post real time messages about their opinions and feelings on a variety of topics, discuss current issues, complain, and express positive sentiments on their daily life. It is no wonder then that many companies have started polling these microblogs as part of their research to get a general sentiment for a product they may be developing or marketing. As for most operations in the modern age, it is the task of many scientists and engineers to replace manual labor with machine labor. This is done to improve speed and efficiency. One challenge then, in the field of polling microblogs, is to develop technology that can detect and summarize an overall sentiment. This process is called *sentiment analysis*.

In this paper, I will look at one such popular called Twitter and build a model for classifying “tweets” into positive, negative, and neutral sentiment. I will build a model that can classify sentiment into positive and negative classes and then create a task for classifying a tweet into positive, negative, or neutral sentiment. The model will consist of a naïve Bayes classifier as well as a few other probability theorems that will be discussed further later. Additionally, I will use the Natural Language Toolkit 3.0 (NLTK), a library for Python (the language of choice) to aid in the classification process.

Before I begin to explain technical details that went into this project, let’s discuss why sentiment analysis is important. As discussed before, many companies use social media as a medium to communicate with consumers. These entities include corporate businesses, celebrities (musicians, actors, writers, etc.), and politicians among others. Politics is one of the biggest arenas that can benefit from sentiment analysis. Such analysis is especially useful when new bills are passed or presidential debates are held in order to see what the popular opinion is. With this valuable data, theoretically, politicians could review their platform and adapt in a way that could get them a better popular opinion. Sentiment analysis is also useful for movie producers. It lets them plan for the future; for example, if a movie is successful and many people are talking about it, executives can increase sales by more marketing and investing in close fields (books, video games, toys, etc.), or take opposite action if the movie reaction is unfavorable.

Finally, I will explain why Twitter was specifically used in contrast to other social media websites like Facebook, Instagram, Pinterest, etc. Twitter is unique in that the data available is open. Although there is a social graph and friend/follower connections, much of the data is open to sift through by anyone, which is why Twitter is often used for polling. It’s easy to query tweets by certain keywords. In contrast, Facebook is a closed-network where most of the data can only be valuable and extractable among a group of friends. Ultimately, since we are doing sentiment analysis, we need to parse text. This is why Instagram and Pinterest is not preferred because data from those sites are in the form of pictures which with very little words—making it difficult to analyze.

The main algorithm used to detect sentiment is known as the naïve Bayes classification model. This is a simple probabilistic classifier based on applying Bayes’ theorem with strong (naïve) independence assumptions. In simple terms, a naïve Bayes classifier assumes that the value of a particular feature is unrelated to the presence (or absence) or any other feature. For example, a cat may be considered a feline if it has a tail, claws, fur, and omits a certain cry. A naïve Bayes classifier considers each of these features to contribute independently to the probability that a cat is a feline, regardless of the presence or absence of any other features.

Abstractly, the probability model for a classifier is the conditional model

(C \vert F_1,\dots,F_n)\,

Using Bayes’ theorem, this can be written as

(C \vert F_1,\dots,F_n) = \frac{p(C) \ p(F_1,\dots,F_n\vert C)}{p(F_1,\dots,F_n)}. \,

Using Bayesian Probability terminology, the above is the same as

mbox{posterior} = \frac{\mbox{prior} \times \mbox{likelihood}}{\mbox{evidence}}. \,