# Python Sentiment Analysis

The sentiment analysis tool we have developed imports Yelp review data and trains a classifier to detect sentiment for Hotel reviews. This report will outline the methods used to import and clean the data as well as present our analysis.

## Part A – Importing/Cleaning the Data

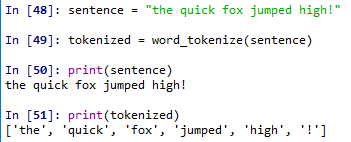
The data available from [www.yelp.com/dataset](http://www.yelp.com/dataset) is in JSON format. This presented difficulties when trying to load the data into our environment. However, CSV formats, which are far easier to import, were available online and accessed at <https://www.kaggle.com/yelp-dataset/yelp-dataset/data>.

Having obtained these CSV files, our code then establishes the path to both the Review and Business data. This is done without hardcoding any file path in order to avoid any errors that may occur when running the script on different systems. We then read in 100,000 reviews from the review data, extracting columns 2, 3 and 5. This will create a data frame containing reviews, business id and star rating. Similarly, we read in 100,000 businesses extracting the business id and category columns.

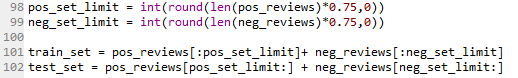
A merged data frame is created by joining the business ids with the required category (Hotels) to the corresponding reviews. As some businesses are present in multiple categories it was important to ensure that the categories column was checked to contain the selected industry, not to equal that industry.

## Part B – Sentiment Analysis

Once the subset has been generated appropriately, we can then begin to build the sentiment analysis tool. Two separate data frames are created, one for all the 5-star reviews and another for all 1 star reviews. This will allow us to develop predictive models to recognise features of a positive/negative review.

With this in mind we extracted the text column from each data frame and vectorised each review. This results in a series of lists each containing all the individual words from the review. For example:

It is important to reset the indices of these series in order to run the series through the “create\_word\_features” function. The loops which populate the review lists take considerable running time. This is a difficulty that may arise due to the size of the review data.

The data is then divided into a train set and a test set. This allows us to train out classifier and also test the performance of our analysis tool. In order to preserve the flexibility of our tool, we ensured

that the division of the sets was determined by length of the review list.

At this point our classifier can be determined by running the Naïve Bayes Classifier over the training set. This classifier works by analysing the frequency of the word content of the positive and negative reviews and builds two lexicons that contain the words associated with reviews of each type. We then test the accuracy of the classifier by running it against the predefined test set. This will return the following output:



An accuracy of approximately 75% was returned on each of the category types that we tested the tool for (Hotels, Food, Jewelry). This percentage is considered a good performance for a sentiment analysis tool.

To test this tool to a greater degree, text from two hotel reviews from TripAdvisor were selected and checked against our classifier.

Figure 2: Imperial Hotel Galway - Negative

Figure 1: Menlo Park Galway - Positive

In both cases the correct result was determined by our classifier tool. This demonstrates the classifiers ability to take reviews from any domain and correctly identify the sentiment.

## Part C – Limitations & Improvements

Our sentiment analysis tool determines whether a review is either positive or negative. It does not accommodate for neutral reviews. While there is an argument that reviews typically must be either positive or negative, the case where reviews select a 3-star review may indicate a review that is neutral overall. Our tool could be developed further to accommodate for this grey area.

The size of the data set made for a limited sample being taken. This is to save time when running the programme however this limit could be removed in order to train a more accurate classifier. Although this may take a considerable length of time, the accuracy of the classifier would be greatly improved.