Task 5: Cross-Validation and Evaluation Sets

Step 1: Data Split

Split your training and validation dataset into n equal folds in two different ways.

1. Random: Assign the datapoints to folds randomly.
2. Stratified random: Assign the datapoints to folds randomly but make sure that there is the same number of positive and negative reviews in each fold.

We will experiment with 10-fold cross validation (i.e. $n = 10$), but it is important that your function works for other values of n.

Step 2: Cross-Validation

You can now perform cross-validation using the folds. You should evaluate the accuracy of the Naive Bayes classifier (smoothed version) using 9 folds for training and the 10th one for testing. Repeat this 10 times so that each fold is tested only once. The cross-validation score is the average accuracy across all the runs. How do the two methods of splitting the data compare?

Apart from mean accuracy it is useful to calculate the variance across the cross-validation runs:

$$var = \frac{1}{n}\sum_{i}^{n}(x_i - \mu)^2$$

where x_i is the score of the i^{th} run over a total of n folds and μ is the average of the scores.

To get an unbiased estimate of the true / population variance when calculating sample variance, we would normally divide by $n - 1$; for now, however, we will just use n.

Step 3: Additional Evaluation Sets

If this section we will look at two different evaluation settings, which can both be downloaded from Moodle. One is a set of reviews from 2016 (the reviews you have been using so far were collected before 2004). The other is the test set of 2004 reviews which we held out. The `main` method of `tick5.py` trains your Naive Bayes classifier using the training set from before, and evaluates it using both of these sets of data.

The term "Wayne Rooney effect" is not standard but it's been used by people in the UK sentiment business to refer to the situation when a word or phrase that was a good cue for sentiment in the original data turned out to indicate the opposite sentiment in later data.

How does your system (trained on your original training set) perform on the new review set, in comparison to the held-out data from the original set?

Implement the confusion matrix code to get a better sense of the model's output.

Step 4: Simple Sentiment Classifier Performance

Adjust the code in the `main` method to compute the performance of the simple classifer (without your improvements) from Tick 1 on both of these datasets. How does its performance on the held-out data compare to the 2016 data? Is the difference between the naive bayes classifier and the simple classifier on the 2016 data *significant*? (When we use the word 'significant', we mean it in a technical sense, i.e. do they pass the significance test from `tick4`?)

Describe your findings in your lab book before signing up to get the tick.

Starred Tick (Optional)

Are there any words which are showing a "Wayne Rooney effect" in the 2016 data? What other reasons are possible for changes in performance? To what extent do these explain your results?

Note that we are expecting you to look at the 2016 data in order to do this **error analysis**. However, in a real experiment, after looking at test data, you have to obtain additional data to evaluate any further experiments.