Don't explain the basics and anything you've seen in the lecture

Describe what you did that is distinctive (e.g. vocabulary pruning)

Report running time and memory footprint

Plots of performance changes when varying hyperparameters

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TRY:Kernels for SVM

LOOK INTO: Class imbalance for Logistic

As preprocessing for all three approaches, we used a stop word filter (provided) and a Porter stemmer.

For Naive Bayes approach, Laplace smoothing is used.

The lowest accepted threshold of accepted (W|C) is set to 0.3

On a 1000 test set, it takes 36.909 s to run and GB of memory.

After training, the Naive Bayes approach attained a F1 score of

This error is likely to be due to

For Logistic Regression approach, we chose one-vs-all instead of a multi-class approach, hence are checking against each potential classifier as a linear classification problem.

Here, the threshold is set as 0.5

The iteration is set to 20, and the learning rate at 0.01

We used the logistic regression function provided by the tinyIR library

On a 1000 test set, it takes 478.536 s to run and GB of memory.

After training, the Logistic Regression approach attained a F1 score of

This error is likely to be due to

//question, on line 27, var lR = new LogisticRegression(config, theta, 0.5, 10)

//the 10 has to be equal test.size instead right?

For Support Vector Machine approach,

Lambda is chosen to be 1.

The standard linear SVM is used without any kernels functions.

On a 50,000 test set, it takes 29.121 s to run and GB of memory.

After training, the Support Vector Machine approach attained a F1 score of

This error is likely to be due to