

Naive Bayes:

With out stop words - 0.939

with stop words - 0.941

In naive bayes the prediction will increase when we are removing stop words. In logistic regression, the prediction accuracy depends on learning rate, lambda and number of iterations. By choosing proper learning rate, lambda and number of iterations will give the better results. But in my case Naive Bayes gives better accuracy than logistic regression and perceptron.

In Logisitic regression :

Format (learning\_rate, iterations, lambda )

0.06 50 1

for example:

LR No Stop Words

Accuracy 91.63179916317992

LR With Stop Words

Accuracy 84.72803347280335

This represent the model will predict the most common words that's why the accuracy without stopwords will high. After removing stop words the accuracy dropped slightly. Will these adjust based on choosing proper values

By removing the stop words the total number of available words count will decrease. So, calculating the prediction percentage against prediction without stop words will be high.

Format (learning rate, iterations, lambda )

0.06 50 0.5

LR No Stop Words

Accuracy 91.21338912133892

LR With Stop Words

Accuracy 91.84100418410041

Now the predicting accuracy is improved a lot. More number of iterations with good parameter chosen changed the results. By choosing minimum learning will train a model better.

In perceptron by choosing less learning and good number of iterations will give better results. Convergence require more number of iterations but the model trained with less number of iterations will result in inaccurate.

For example:

```
python3 logreg.py stop words train test 0.150 15 1.5
```

Perceptron No Stop Words

Accuracy 94.14225941422593

Perceptron With Stop Words

Accuracy 92.88702928870293

This Learning rate requires 15 iterations to get good accuracy. But in otherward

```
python3 logreg.py stop words train test 0.150 6 1.5
```

Perceptron No Stop Words.

Accuracy 76.15062761506276

Perceptron With Stop Words

Accuracy 81.58995815899581

The accuracy is reduced.

In perceptron, the learning is calculated based on weights in every iteration. It's faster to compute the model.