Report

1. Results

1.1. Using multinomial NB and BoW

Testing on enron1_testing
Accuracy: 0.8179824561403509 Precision: 1.0 Recall: 0.4429530201342282 F1: 0.6139534883720931

Testing on enron3_testing
Accuracy: 0.8284518828451883 Precision: 0.98 Recall: 0.3769230769230769 F1: 0.5444444444444444

Testing on enron4_testing
Accuracy: 0.9097605893186004 Precision: 0.8886363636363637 Recall: 1.0 F1: 0.941034897713598

1.2. Using NB and Benoulli

Testing on enron1_testing
Accuracy: 0.8201754385964912 Precision: 1.0 Recall: 0.44966442953020136 F1: 0.6203703703703703

Testing on enron3_testing
Accuracy: 0.8284518828451883 Precision: 0.98 Recall: 0.3769230769230769 F1: 0.5444444444444444

Testing on enron4_testing
Accuracy: 0.8895027624309392 Precision: 0.8669623059866962 Recall: 1.0 F1: 0.9287410926365796

1.3. Using Logistic Regression and Benoulli

Testing on enron1_testing
Accuracy: 0.6732456140350878 Precision: 0.0 Recall: 0.0 F1: 0.0

Testing on enron3_testing
Accuracy: 0.7280334728033473 Precision: 0.0 Recall: 0.0 F1: 0.0

Testing on enron4_testing
Accuracy: 0.27992633517495397 Precision: 0.0 Recall: 0.0 F1: 0.0

1.4. Using Logistic Regression and BoW

Testing on enron1_testing
Accuracy: 0.6732456140350878 Precision: 0.0 Recall: 0.0 F1: 0.0

Testing on enron3_testing
Accuracy: 0.7280334728033473 Precision: 0.0 Recall: 0.0 F1: 0.0

Testing on enron4_testing
Accuracy: 0.27992633517495397 Precision: 0.0 Recall: 0.0 F1: 0.0

1.5. Using SGD and BoW

Testing on enron1_testing
Accuracy: 0.31798245614035087 Precision: 0.25748502994011974 Recall: 0.5771812080536913 F1: 0.35610766045548653

Testing on enron3_testing
Accuracy: 0.5104602510460251 Precision: 0.30303030303030304 Recall: 0.6153846153846154 F1:

0.4060913705583757

Testing on enron4_testing
Accuracy: 0.5543278084714549 Precision: 0.7197640117994101 Recall: 0.6240409207161125 F1: 0.6684931506849315

1.6. Using SGD and Bernoulli

Testing on enron1_testing
Accuracy: 0.3706140350877193 Precision: 0.2076271186440678 Recall: 0.3288590604026846 F1: 0.2545454545454546

Testing on enron3_testing
Accuracy: 0.4665271966527197 Precision: 0.1921182266009852 Recall: 0.3 F1: 0.23423423423423423

Testing on enron4_testing
Accuracy: 0.6243093922651933 Precision: 0.775811209439528 Recall: 0.6726342710997443 F1: 0.7205479452054794

2. Questions

1. Which data representation and algorithm combination yields the best performance (measured in terms of the accuracy, precision, recall and F1 score) and why?

   Overall, Naive Bayes with Bag of Words performed the best for the text classification. I believe the reason it performed so well is because of the large number of features and the parameters being used more, however, the naive bayes trained on the Bernoulli dataset performed almost exactly as well.

2. Does Multinomial Naive Bayes perform better (again performance is measured in terms of the accuracy, precision, recall and F1 score) than LR and SGDClassifier on the Bag of words representation? Explain your yes/no answer.

   Yes, the Multinomial Naive Bayes outperformed every classifier. Logistic regression performed the worst in my case. For some reason, using logistic regression, I obtained the same results regardless of the dataset, which leads me to believe there was an error in programming, however the SgD also had similar results.

3. Does Discrete Naive Bayes perform better (again performance is measured in terms of the accuracy, precision, recall and F1 score) than LR and SGDClassifier on the Bernoulli representation? Explain your yes/no answer.

   Yes, the discrete naive bayes, performed better than both SGD and LR. It performed almost as well as the multinomial, give or take 1% in accuracy. Both were by far the best classifiers.

4. Does your LR implementation outperform the SGDClassifier (again performance is measured in terms of the accuracy, precision, recall and F1 score) or is the difference in performance minor? Explain your yes/no answer.

   Yes, my LR implementation outperformed the SGD classifier on both datasets, except in the case of the enron4 dataset. To understand how I trained LR, i found the best lambda value by making a list of candidate values: lambdas = [.01, .1, .001, 1, 10]. For each lambda I trained the data set and tested the accuracy on the validation set, picking the best lambda and retraining on the entire training set. The highest accuracy I was able to obtain was around 72%. I varied the maximum iterations between 1000 and 3000, but even in cases where the model converged before going through all iterations, my accuracy was always the same. The SGD model performed poorly with 50% for the bag of words and 60-70 for the Bernoulli dataset. All and all Naive Bayes performed the best.