REPORT

Project 1: Naive Bayes and Logistic Regression for Text Classification

For this project I did the following steps that are mentioned in the project description.

- Downloaded the required datasets and uploaded them into colab.
- Unzipped the datasets and dealt with the anomaly of the path for dataset enron2
- Converted each of the text dataset into a matrix of features × examples for both Bag of Words model and Bernoulli model, where the features only consist of words using the alphabets. Here the Bag of Words representation has the frequency of words while the Bernoulli representation has either 0/1.
- Implemented the Multinomial NB algorithm in log scale along with add-one laplace smoothing. Here, I computed the priors and the conditional probabilities in log scale and using add-one laplace smoothing. And then predicted the values using the prior and conditional probabilities for the testing dataset. This was implemented for the Bag of Words model.
- Implemented the Discrete NB algorithm in log scale along with add-one laplace smoothing. Here, I computed the priors and the conditional probabilities in log scale and using add-one laplace smoothing. And then predicted the values using the prior and conditional probabilities for the testing dataset. This was implemented for the Bernoulli model.
- Implemented the MCAP LR algorithm along with L2 regularization. This was implemented for both the Bag of Words and Bernoulli models. Here, I computed the value of the sigmoid function, and then I computed the gradient of the conditional log likelihood. Using the gradient along with penalizing the weights using L2 regularization I trained the model. Before training the model on the full dataset, I trained it on 70% training data and simultaneously validated on the remaining 30% training data, for different lambdas, this was done in order to get the best lambda for training. For me, out of the values [0.1, 0.3, 0.5], the best lambda came out to be 0.1. I choose the learning rate to be 0.01. To find the best lambda, the number of iterations

used was 30. For the final training, the maximum number of iterations was set to 80 epochs, along with checking the conditional log likelihood or loss, so, if the log_cll<0.1 then we will break the loop and stop the training. Finally, after the completion of the training, I tested the implementation using the testing dataset along with the parameters realized from training.

- Implemented the SGDClassifier algorithm from sklearn and tuned the parameters using GridSearchCV from sklearn. This was implemented for both the Bag of Words and Bernoulli models. I tuned the following parameters: alpha, loss function, maximum iterations. I imputed the following: for alpha or learning rate, (0.001 and 0.01); for loss, (squared hinge, hinge and log): for maximum iterations, (30, 50, and 80). The tuning was done on the 30% training dataset, while the final training was done on the full training dataset. Finally, I tested the implementation using the trained classifier.

Following are the results for the above mentioned steps:

DATASET	**ALGORITHM**	**REPRESENTATION**	**ACCURACY**	**PRECISION**	**RECALL**	**F1 SCORE**
Enron1	MNB	BOW	72.8	76.08	78.94	72.55
	DNB	BERN	83.77	88.2	75.85	78.61
	MCAP LR	BOW	94.29	92.77	94.9	93.68
		BERN	100	100	100	100
	SGD	BOW	98.68	98.85	98.15	98.49
		BERN	100	100	100	100
Enron2	MNB	BOW	73.64	74.95	81.41	72.53
	DNB	BERN	87.23	89.81	77.74	81.32
	MCAP LR	BOW	97.07	95.41	97.5	96.38
		BERN	99.58	99.71	99.23	99.46

	SGD	BOW	99.58	99.24	99.71	99.47
		BERN	100	100	100	100
Enron4	MNB	BOW	88.76	93.25	79.93	83.83
	DNB	BERN	99.63	99.74	99.34	99.54
	MCAP LR	BOW	88.02	92.87	78.61	82.56
		BERN	99.44	99.61	99.01	99.31
	SGD	BOW	99.26	99.49	98.68	99.07
		BERN	99.26	99.49	98.68	99.07

Questions asked :

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

Stochastic Gradient Descent Classifier is yielding the best results, this might be attributed to the efficiency of the SGDClassifier. It is because the SGD is an efficient discriminative model.

2. Do 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.

From the table, it can be seen that in terms of accuracy, precision, recall, and F1 score, Multinomial Naive Bayes does not perform better than LR nor SGDClassifier for the Bag of words representation. This is because the Multinomial NB is a generative model while the SGDClasifier and LR are discriminative models.

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.

From the table, it can be inferred that on the basis of performance, Discrete Naive Bayes does not perform better than either SGDClassifier or the LR for the Bernoulli representation. Discrete Naive Bayes is a very simple algorithm, but it is out performed by SGDClassifier and LR, in which the parameters are tuned.

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.
From the observations, we can infer that the SGDClasifier slightly outperforms the LR implementation. This is because the SGDClassifier speeds up the optimization of the conditional log likelihood or the loss function.