SMAI

PROJECT

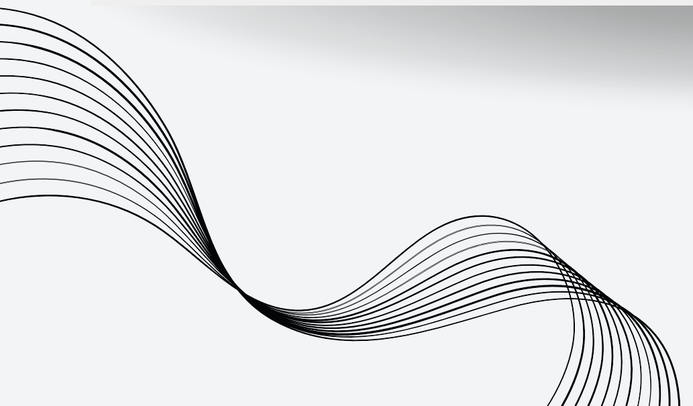
TEAM GALAXY

CONTENT

INTRODUCTION

Some algorithms require a large amount of labeled data to train the classifier. But, in some cases, it is difficult to get a large amount of labeled data. In such cases, the unlabeled data can be used to improve the performance of the classifier.

One such algorithm is the cotrain algorithm which is mentioned in the paper (https://www.cs.cmu.edu/~avrim/Papers/cotrain.pdf).



INTRODUCTION

The Cotrain algorithm is a semi-supervised learning algorithm that uses unlabeled data to improve the performance of the classifier. The algorithm uses two classifiers and two views of the data to train the classifiers.

Co-training assumes that

- Features can be split into two sets.
- Each sub-feature set is sufficient to train a good classifier.
- The two sets are conditionally independent given the class.

COTRAINING

Given:

- a set L of labeled training examples
- a set U of unlabeled examples

Create a pool U' of examples by choosing u examples at random from U

Loop for k iterations:

Use L to train a classifier h_1 that considers only the x_1 portion of x

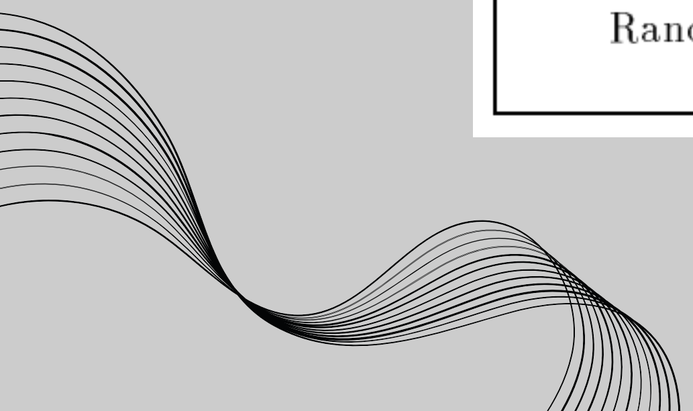
Use L to train a classifier h_2 that considers only the x_2 portion of x

Allow h_1 to label p positive and n negative examples from U'

Allow h_2 to label p positive and n negative examples from U'

Add these self-labeled examples to L

Randomly choose $2p + 2n$ examples from U to replenish U'



DELIVERABLES

01

DELIVERABLE 1

Implement the Cotraining paradigm described in the paper

02

DELIVERABLE 2

Generate, create, and find other relevant datasets which could be used to evaluate this method

03

DELIVERABLE 3

Do a comprehensive evaluation of the method

04

DELIVERABLE 4

Compare against current methods for similar training paradigm which uses both labelled and unlabelled datasets.

DATASETS USED

1

The WebKB Binary dataset comprises 1051 web pages from computer science departments categorized as Course (230) and Non-Course (821). The dataset is organized into Fulltext (web page content) and Inlinks (anchor text on hyperlinks) directories.

WEBKB (BINARY)

2

With 8,282 pages manually categorized into seven classes, including Student, Faculty, Staff, Department, Course, Project, and Other, the dataset features content from four specific universities and additional miscellaneous pages.

WEBKB (MULTICLASS)

DATASETS USED

3

News Category Dataset, encompassing 210k news headlines from HuffPost (2012-2022). With 42 categories, we focus on the top 15, such as POLITICS and WELLNESS. To address data imbalance, we randomly sampled 1000 articles per category for algorithmic analysis, utilizing two views: Headlines and Short Descriptions.

NEWS CATEGORY

4

Rt-polaritydata dataset, consisting of 10,662 snippets, divided equally into positive and negative sentiments. Each line represents a down-cased snippet, serving as the basis for creating an antonymous view. This dual-view dataset enhances our analysis by juxtaposing original and antonymous perspectives.

SENTENCE POLARITY

COMPARISION

Error rate in percent for classifying web pages as course home pages.

Results Shown in The Paper

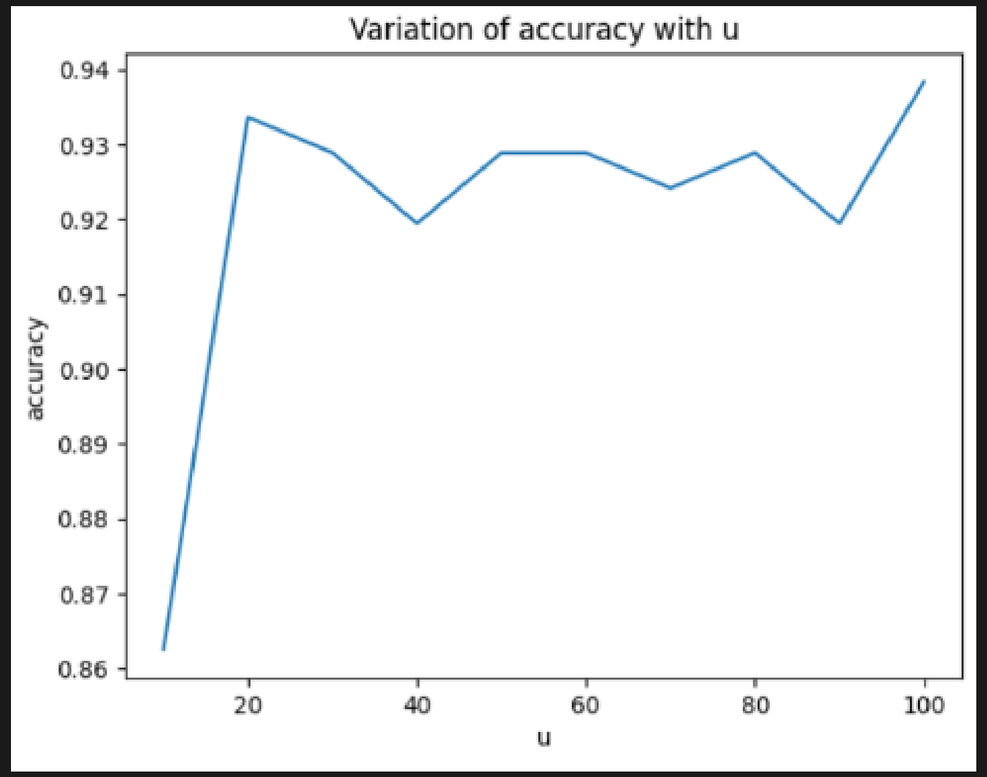
	Page-based classifier	Hyperlink-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

Results we obtained

	Algorithm	page-based	link-based	combined
0	co-training	8.53081	6.16114	7.109
1	supervised learning	14.6919	14.6919	14.6919

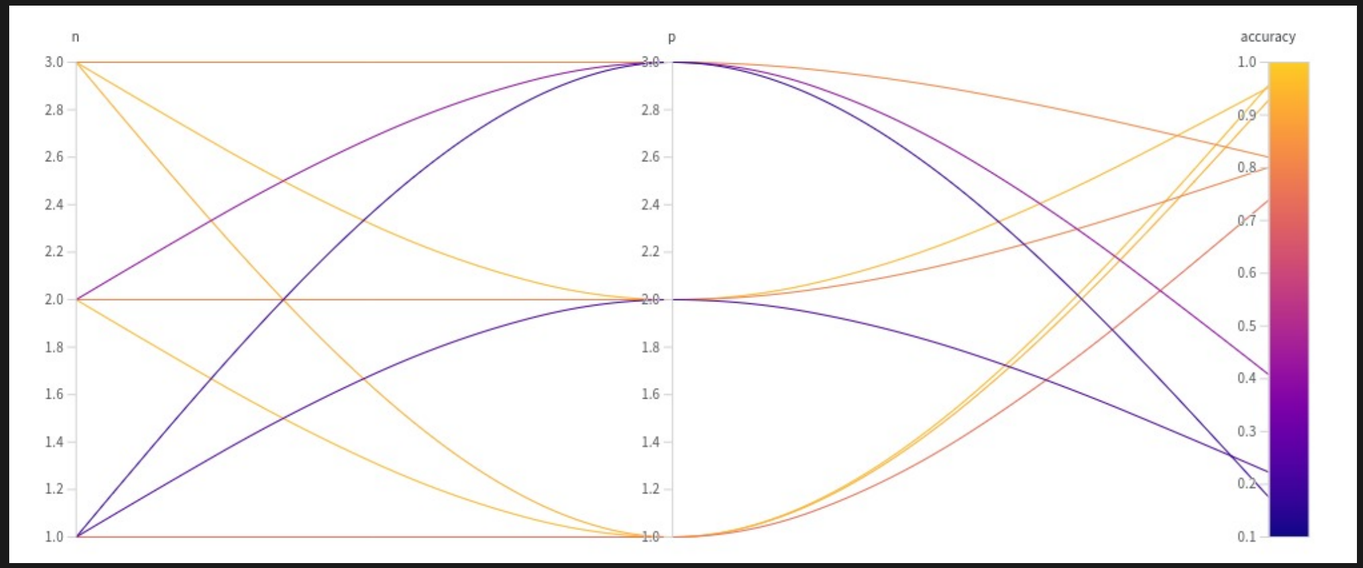
Our results with co-train are better than those of the paper.

Variation of Accuracy with U



- The graph shows the variation of accuracy with ‘u’, where u is the number of unlabelled examples.
- From the above graph, we can observe that the value of accuracy is increasing as u increases. However there are some ups and downs.

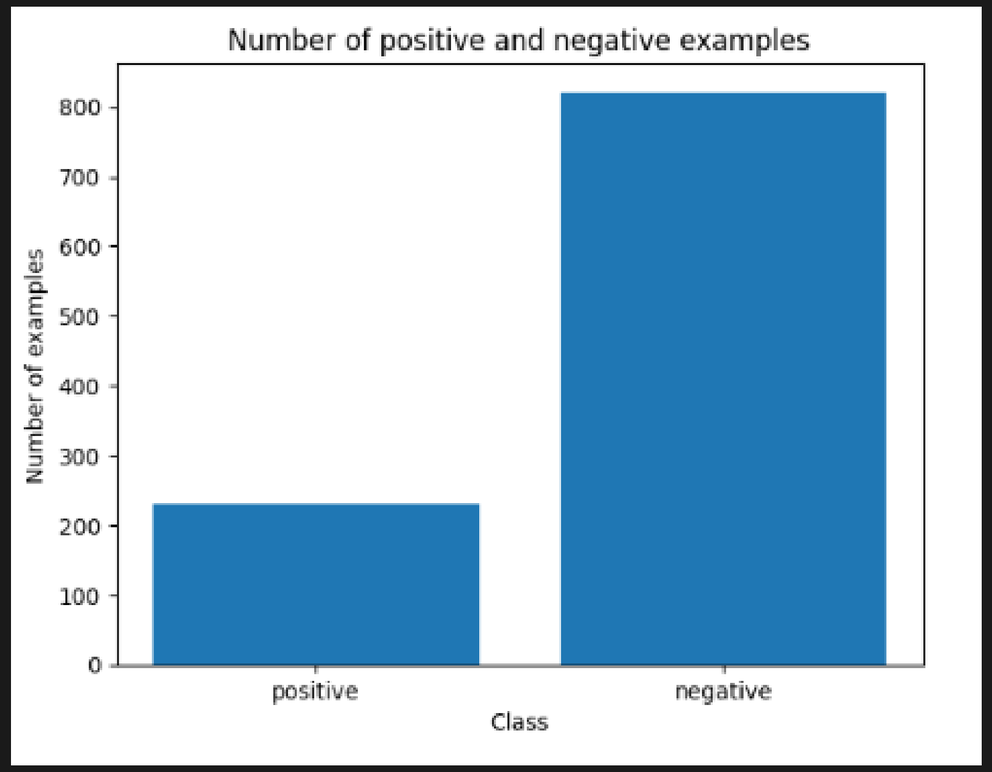
WandB sweep varying n and p



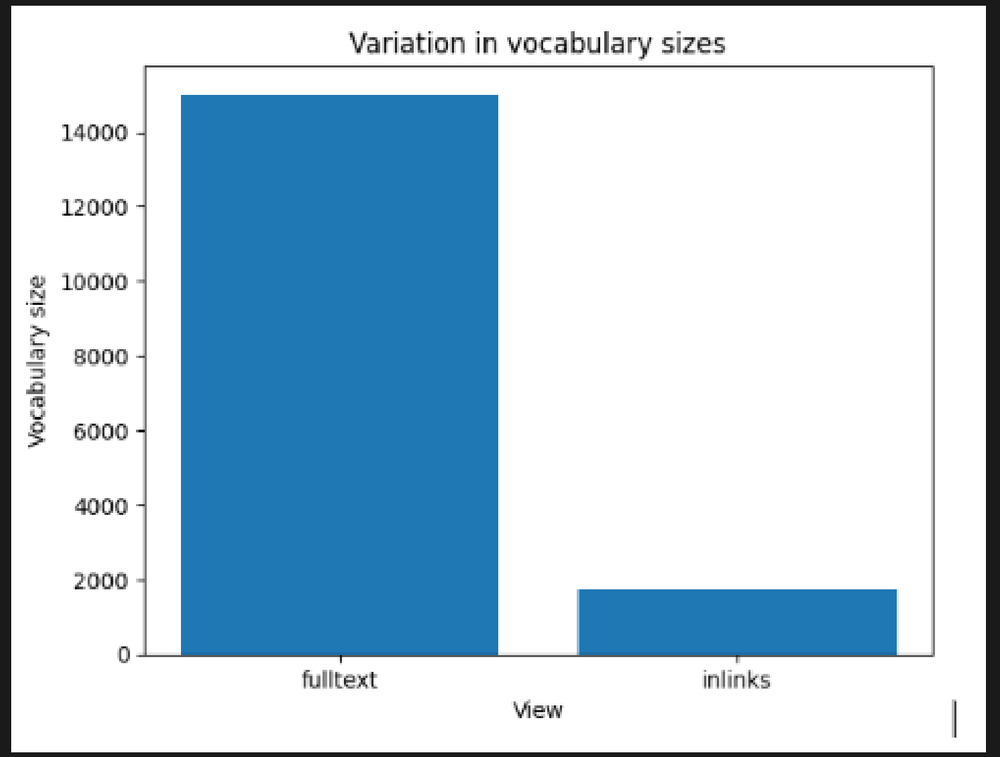
- The accuracy is maximum for the following (n,p) pair: (2,1)

OBSERVATIONS AND ANALYSIS

WEBKB BINARY

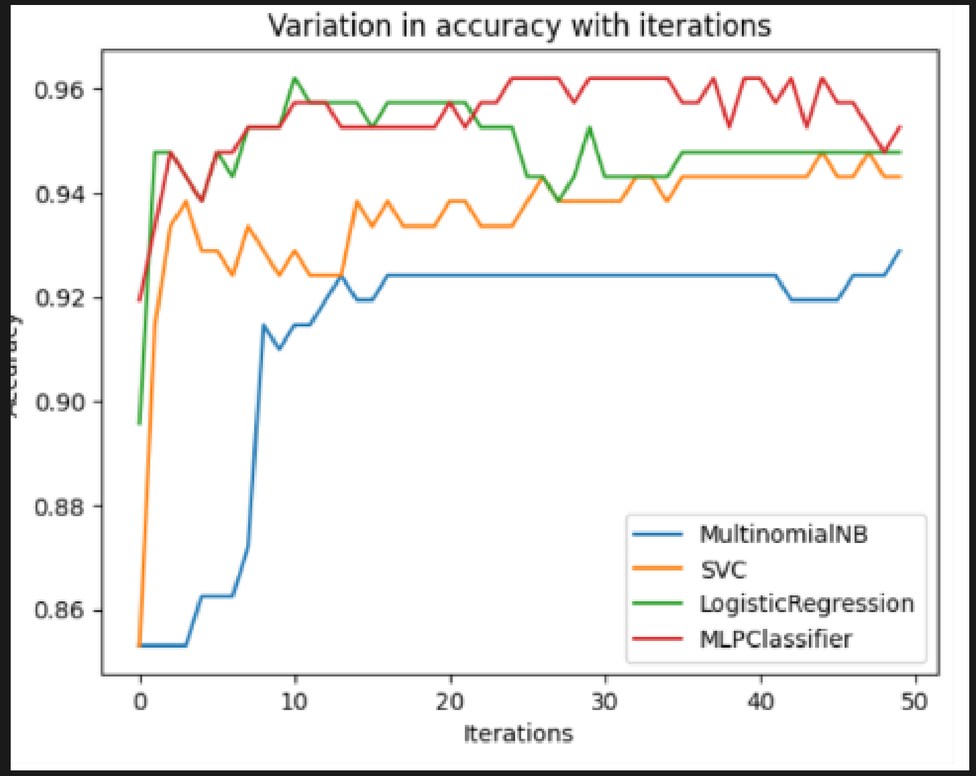


- The above figure shows the distribution of data samples between the two classes.
- From the graph, we can observe that the number of negative examples is 4 times that of the positive examples.

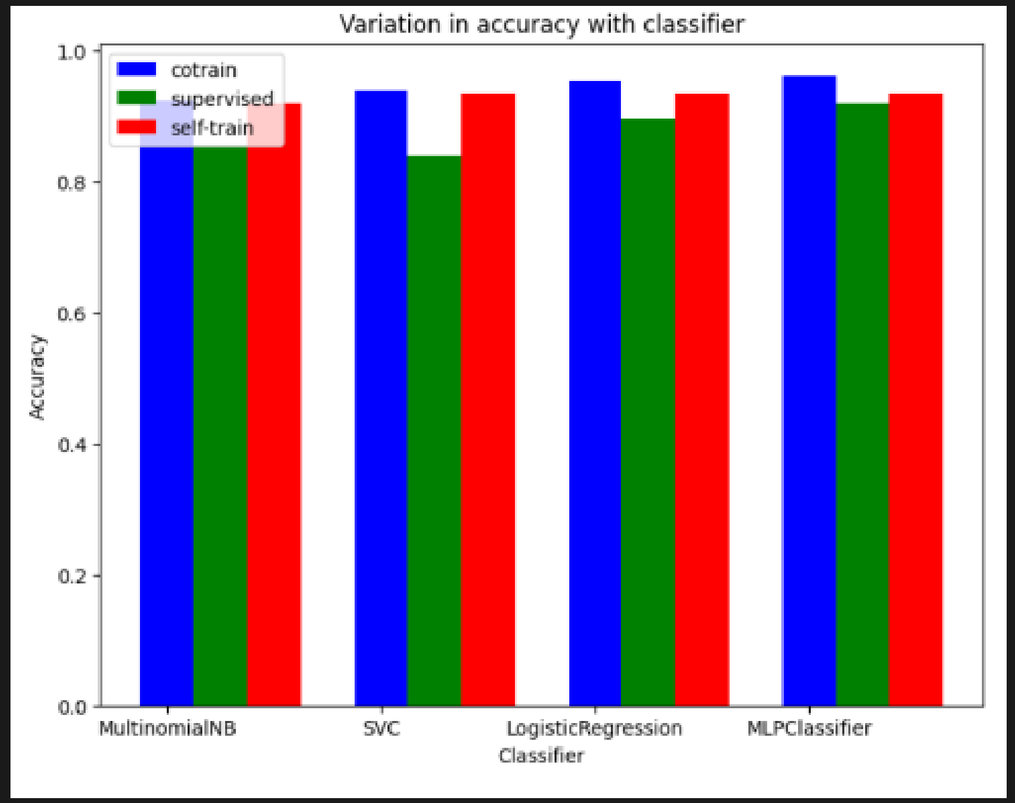


- The graph below shows the variation in vocabulary sizes of the two different views (FullText and Inlinks).
- The vocabulary size of the full-text view is much bigger compared to that of the in-links as the number of words in the webpage is more compared to that of the words used in hyperlinks pointing to that of the webpage.

WEBKB BINARY



From the above graph, as the number of iterations increases, the performance of the classifiers Naive Bayes, SVM and MLP increases. Whereas for the classifier Logistic Regression its performance first increases with increasing iterations and then starts decreasing.



- From the above graph, self-training and co-training are almost similar.
- From the previous graph, the performance of co-training might be increased by increasing the number of iterations of MLP, SVM, and Logistic Regression.

COMPARING THE THREE METHODS

COTRAINING

	Classifier	Accuracy	Precision	Recall	F1 Score
0	MultinomialNB	0.924171	0.959184	0.741935	0.80481
1	SVC	0.938389	0.898753	0.843728	0.868157
2	LogisticRegression	0.952607	0.954325	0.852061	0.89382
3	MLPClassifier	0.962085	0.946659	0.89767	0.920045

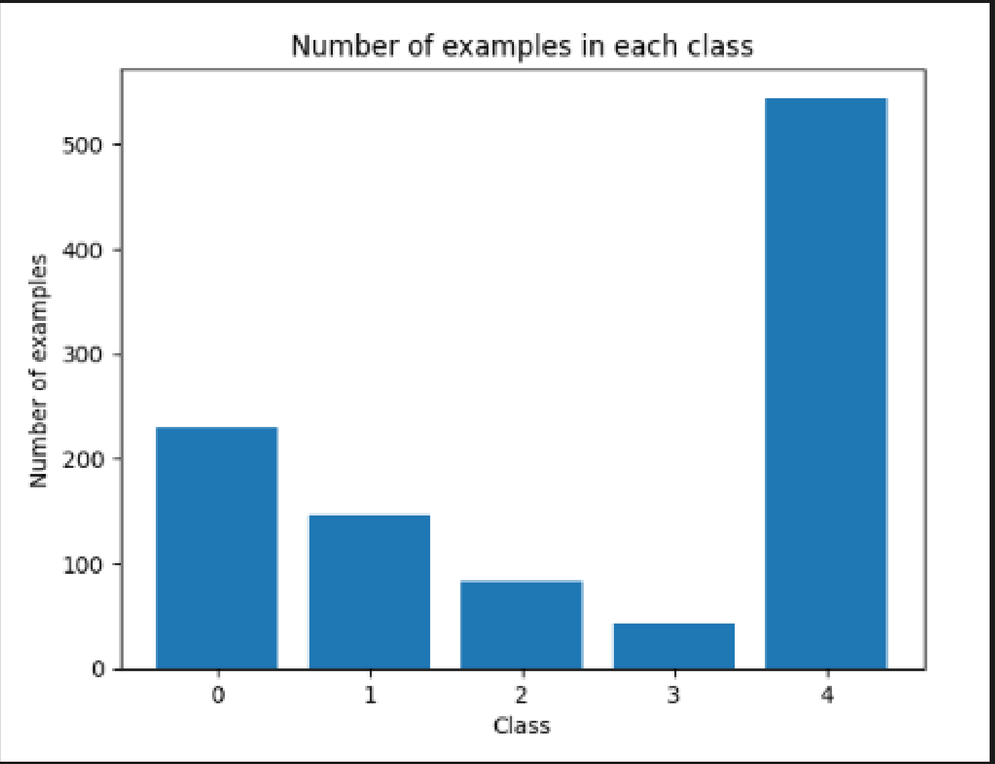
SELF-TRAINING

	Classifier	Accuracy	Precision	Recall	F1 Score
0	MultinomialNB	0.919431	0.956853	0.725806	0.788565
1	SVC	0.933649	0.963918	0.774194	0.83545
2	LogisticRegression	0.933649	0.93983	0.787545	0.841183
3	MLPClassifier	0.933649	0.861934	0.881004	0.871072

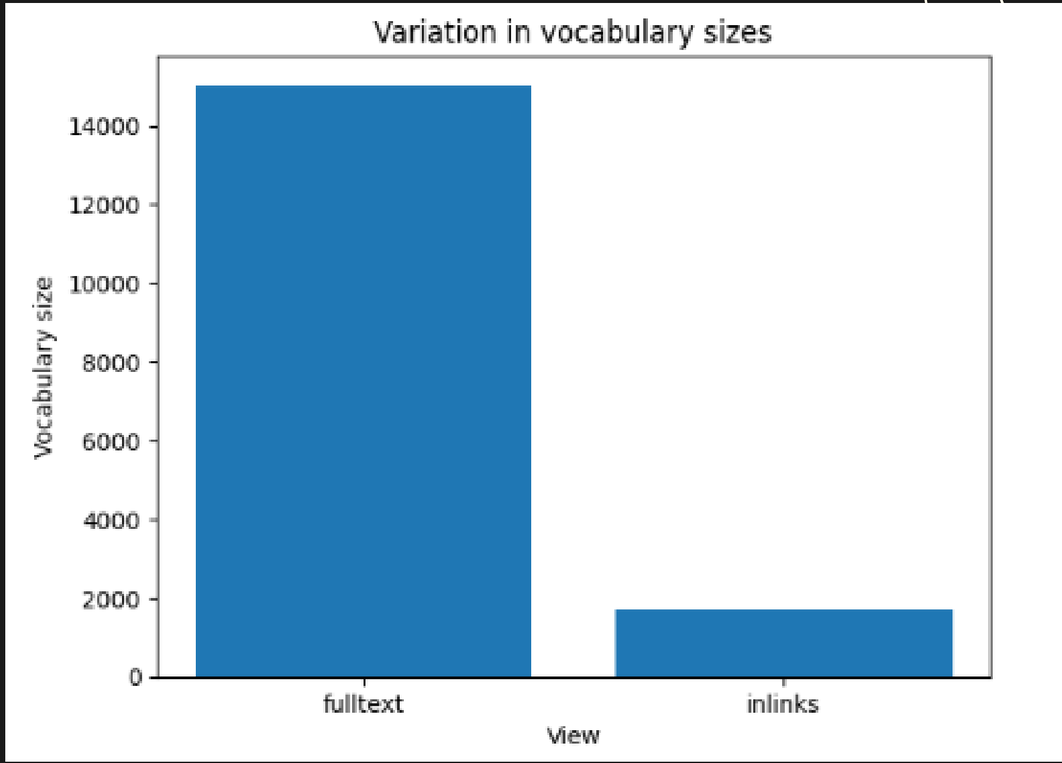
SUPERVISED LEARNING

	Classifier	Accuracy	Precision	Recall	F1 Score
0	MultinomialNB	0.853081	0.42654	0.5	0.460358
1	SVC	0.838863	0.425481	0.491667	0.456186
2	LogisticRegression	0.895735	0.787343	0.84543	0.811607
3	MLPClassifier	0.919431	0.847678	0.819265	0.832516

WEBKB MULTICLASS

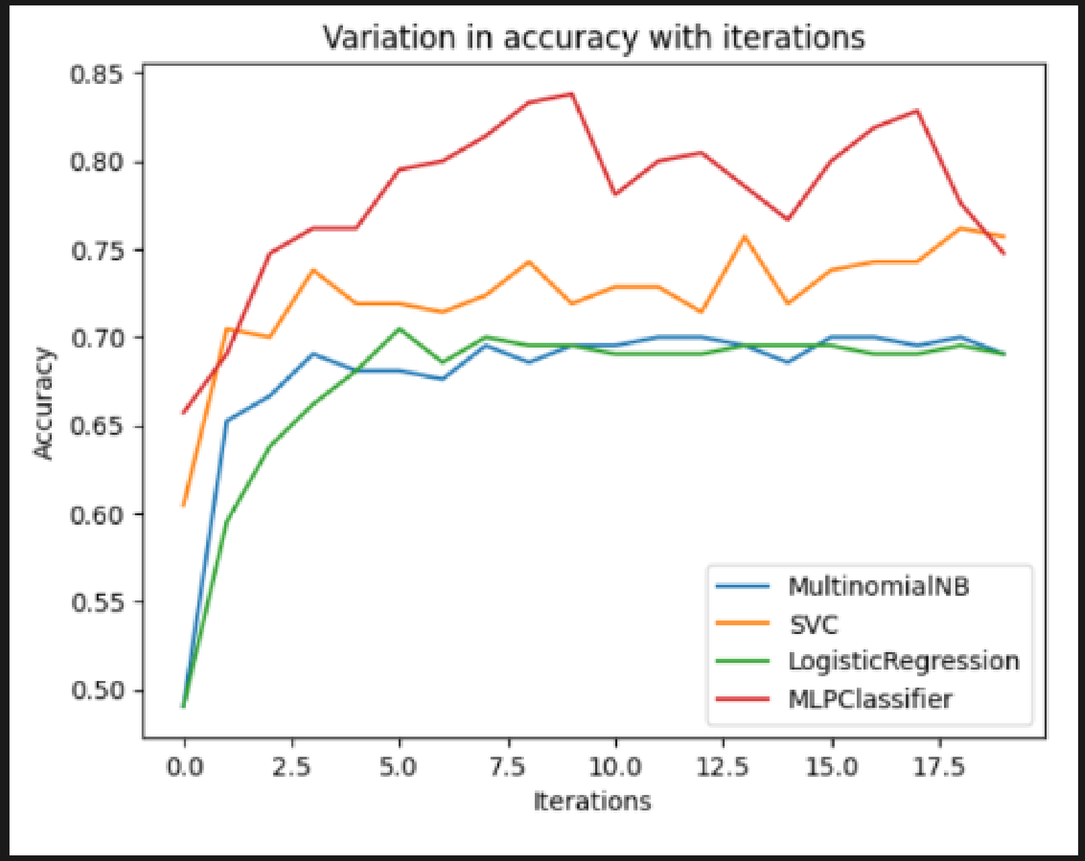


- The above figure shows the distribution of data samples between different classes.



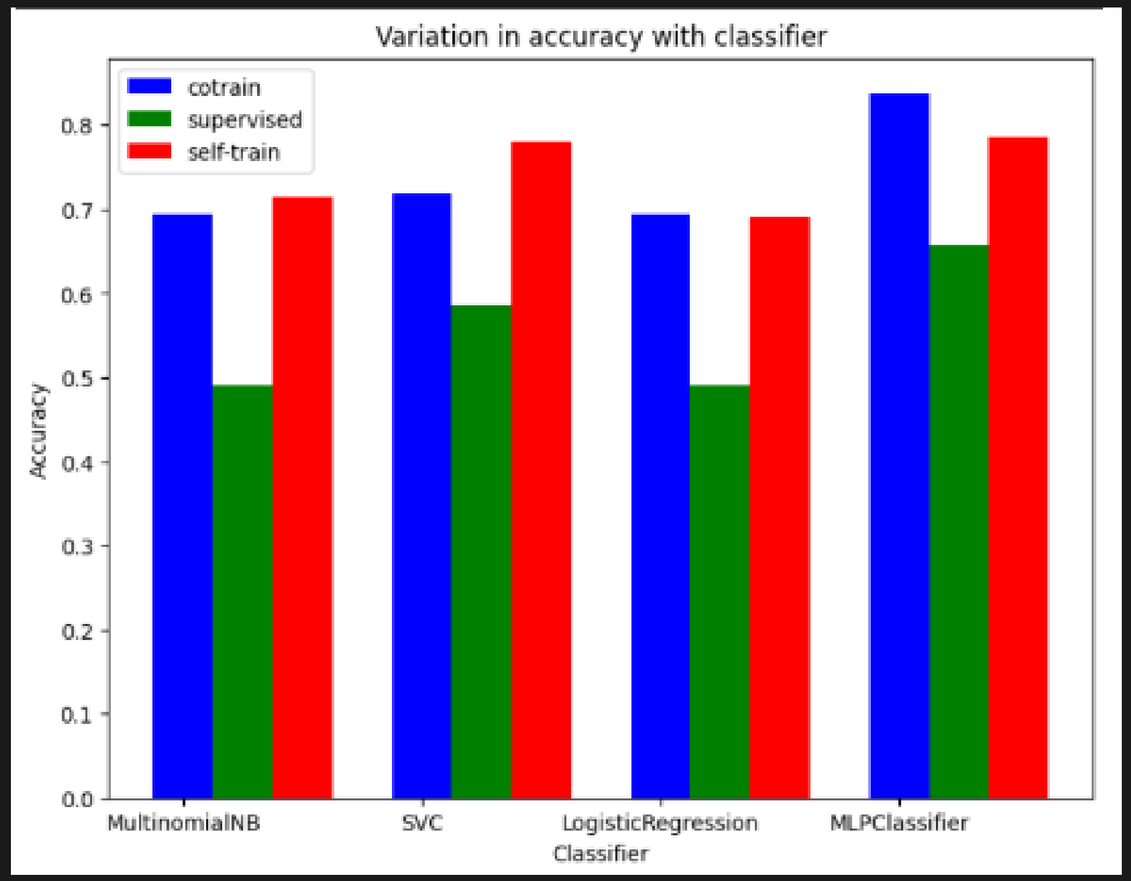
The vocabulary size of the full-text view is much bigger compared to that of the in-links as the number of words in the webpage is more compared to that of the words used in hyperlinks pointing to that of the webpage.

WEBKB MULTICLASS



With increasing the number of iterations, the performance of all the classifiers increases and remains constant after some iterations.

WEBKB MULTICLASS



From the graph, we can observe that the semi-supervised training methods co-training and self-training outperform the supervised training. In the case of Naive Bayes and Logistic Regression self-training and co-training are similar. But, in the case of SVM self-training outperforms the co-training and in the case of MLP co-training outperforms the self-training.

COMPARING THE THREE METHODS

COTRAINING

	Classifier	Accuracy	Precision	Recall	F1 Score
0	MultinomialNB	0.383333	0.470196	0.382084	0.362707
1	LogisticRegression	0.408	0.4021	0.405823	0.387177

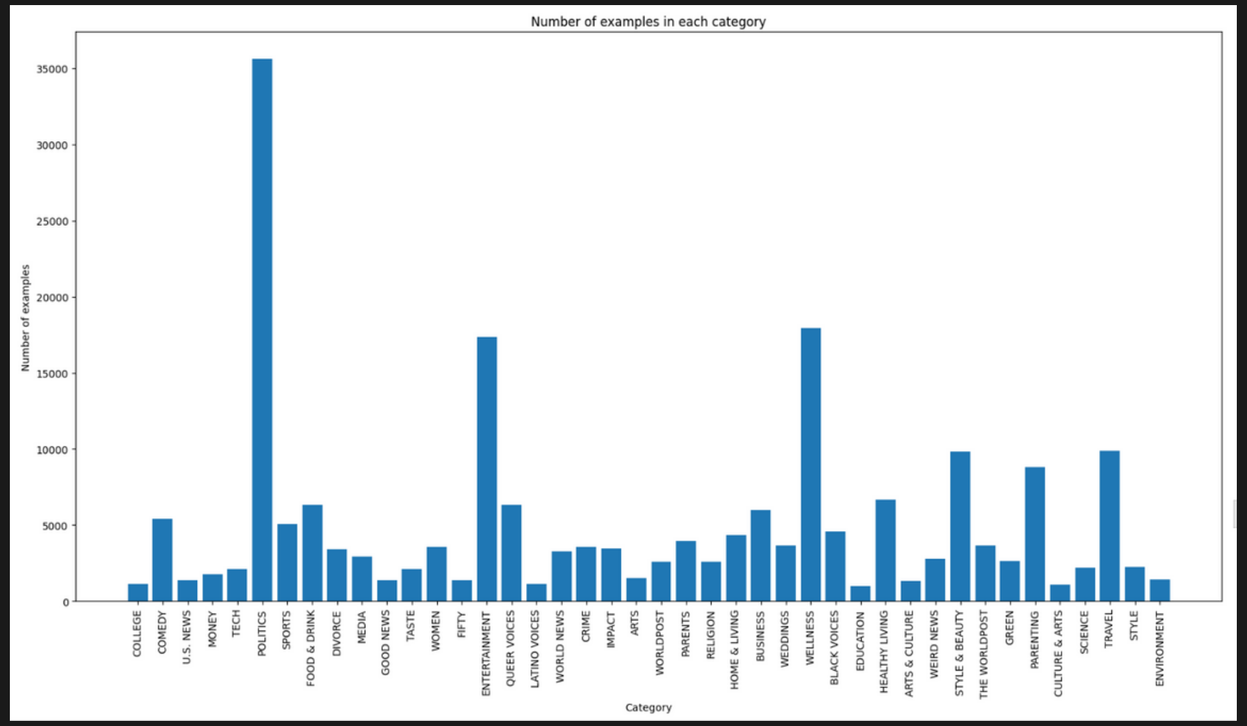
SELF-TRAINING

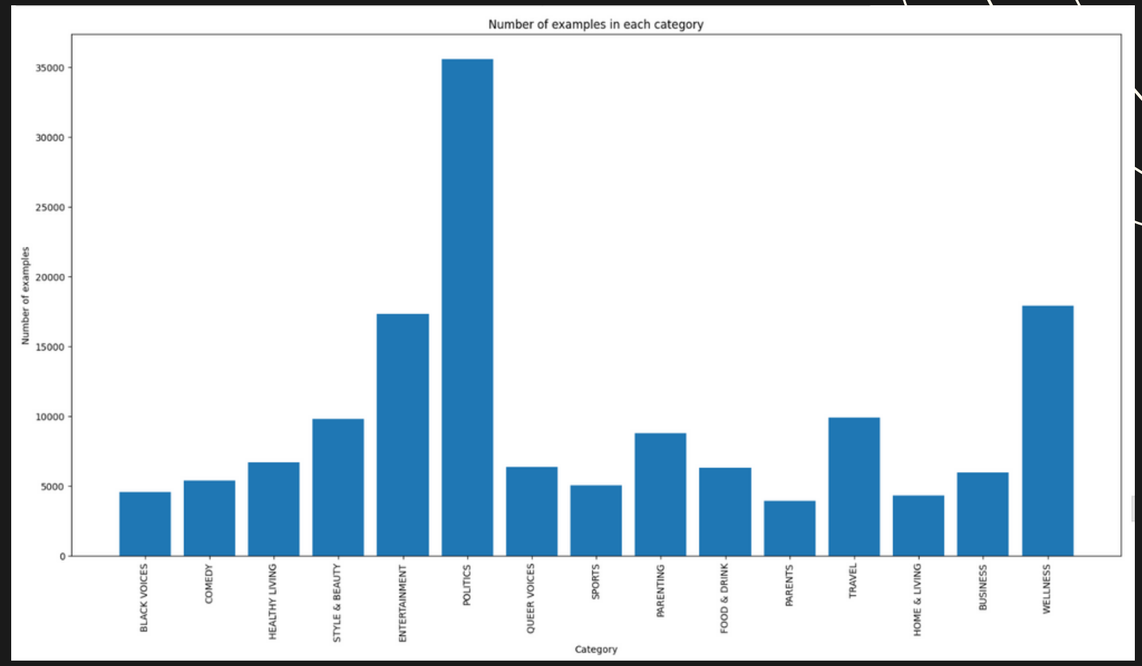
	Classifier	Accuracy	Precision	Recall	F1 Score
0	MultinomialNB	0.356333	0.529908	0.356813	0.341146
1	LogisticRegression	0.408667	0.414241	0.407984	0.391951

SUPERVISED LEARNING

	Classifier	Accuracy	Precision	Recall	F1 Score
0	MultinomialNB	0.280333	0.280497	0.27808	0.274581
1	LogisticRegression	0.284667	0.283885	0.282199	0.277191

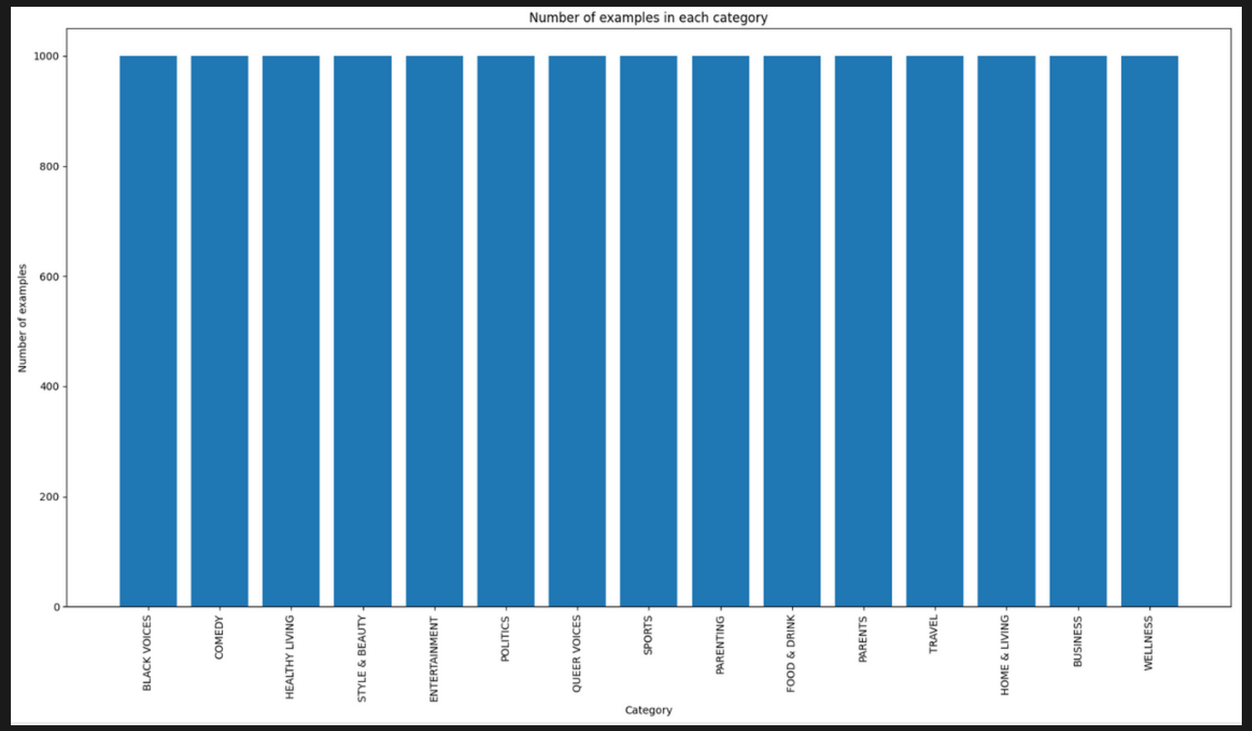
NEWS CATEGORY

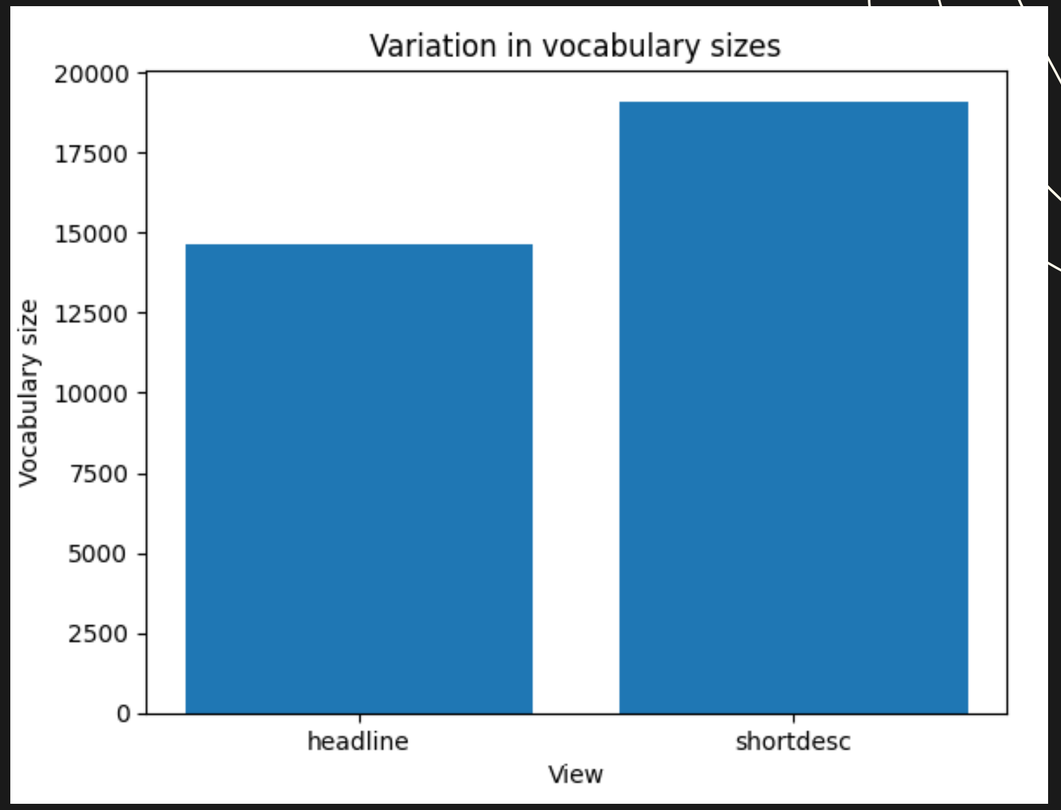




The above data contains 42 classes. We have taken top 15 classes from the data.

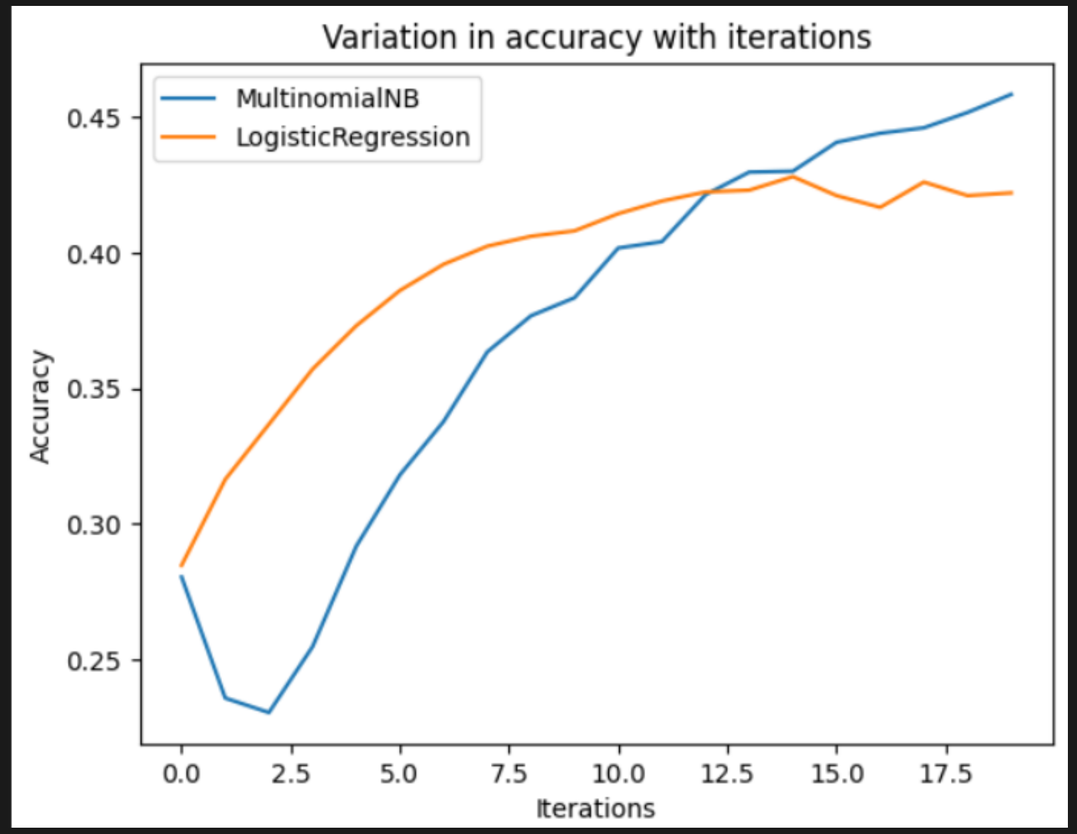
NEWS CATEGORY



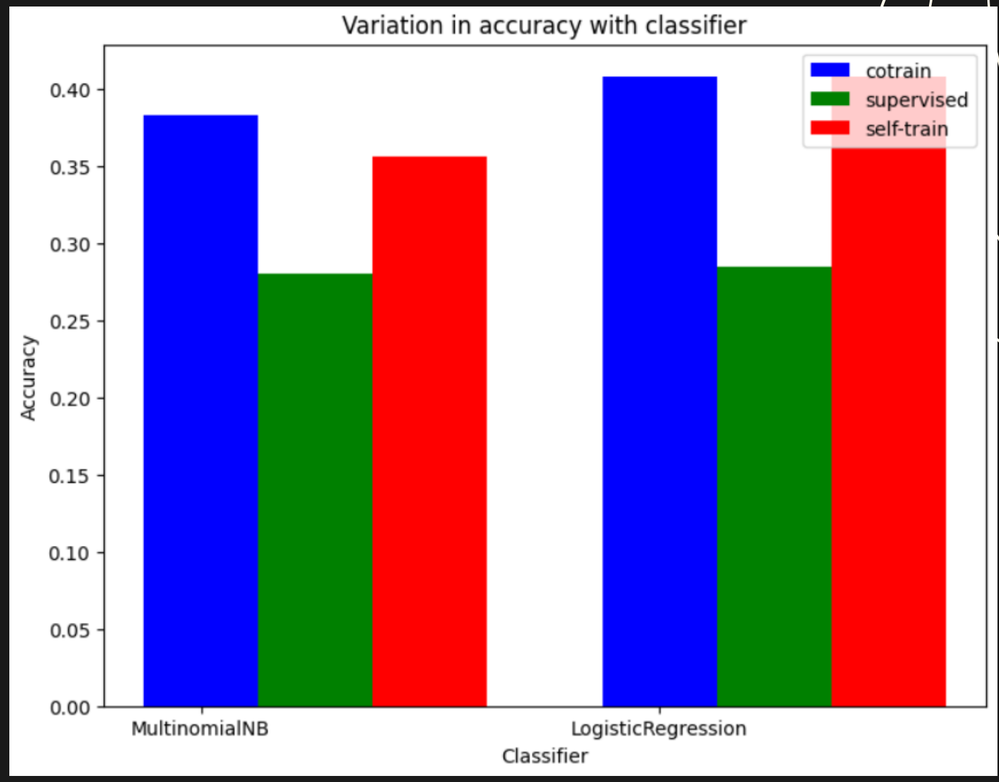


We have balanced the dataset by taking 1000 samples from each class.

NEWS CATEGORY



- The graph shows the variation in accuracy with the number of iterations for co-train.
- By increasing the number of iterations the accuracy of the classifiers increases.



- From the above graph, we can observe that the semi-supervised training methods co-training and self-training outperform the supervised training. In the case of Naive Bayes co-training outperforms self-training significantly and in the case of Logistic regression both are similar.
- Based on the previous graph, the performance of co-training can be increased for both classifiers by increasing the number of iterations.

COMPARING THE THREE METHODS

COTRAINING

	Classifier	Accuracy	Precision	Recall	F1 Score
0	MultinomialNB	0.326333	0.503227	0.326108	0.298407
1	LogisticRegression	0.422667	0.449802	0.42193	0.408237

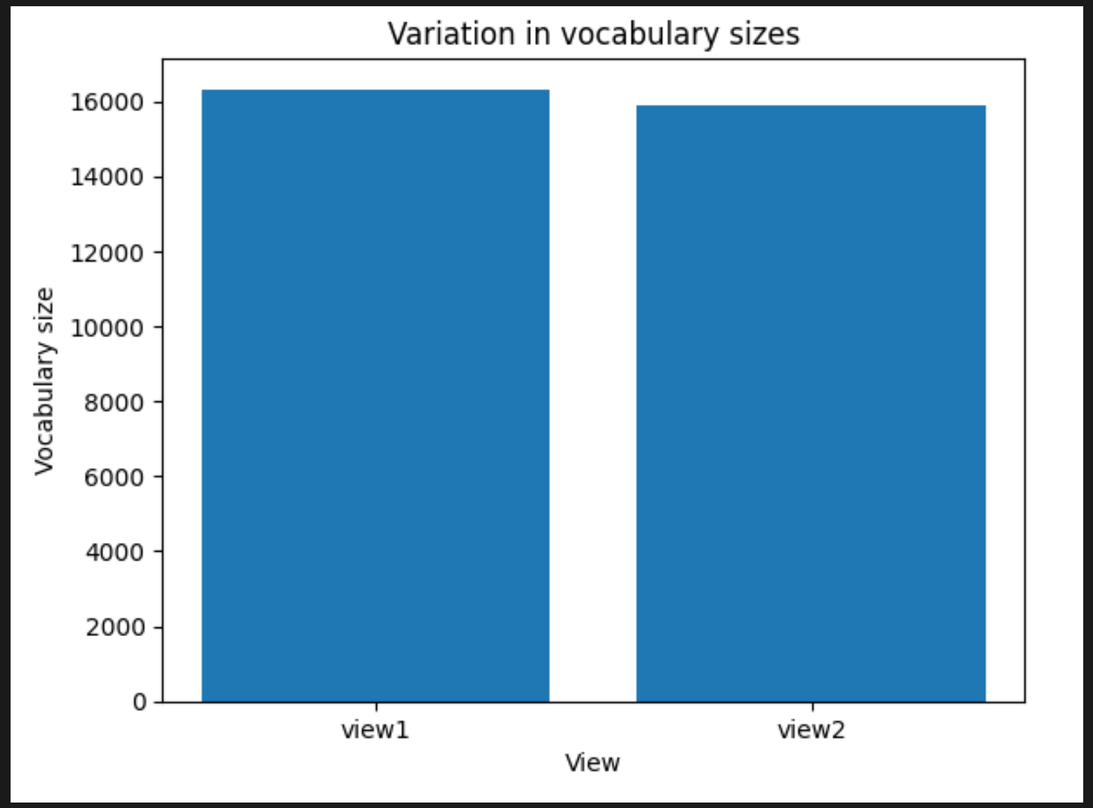
SELF-TRAINING

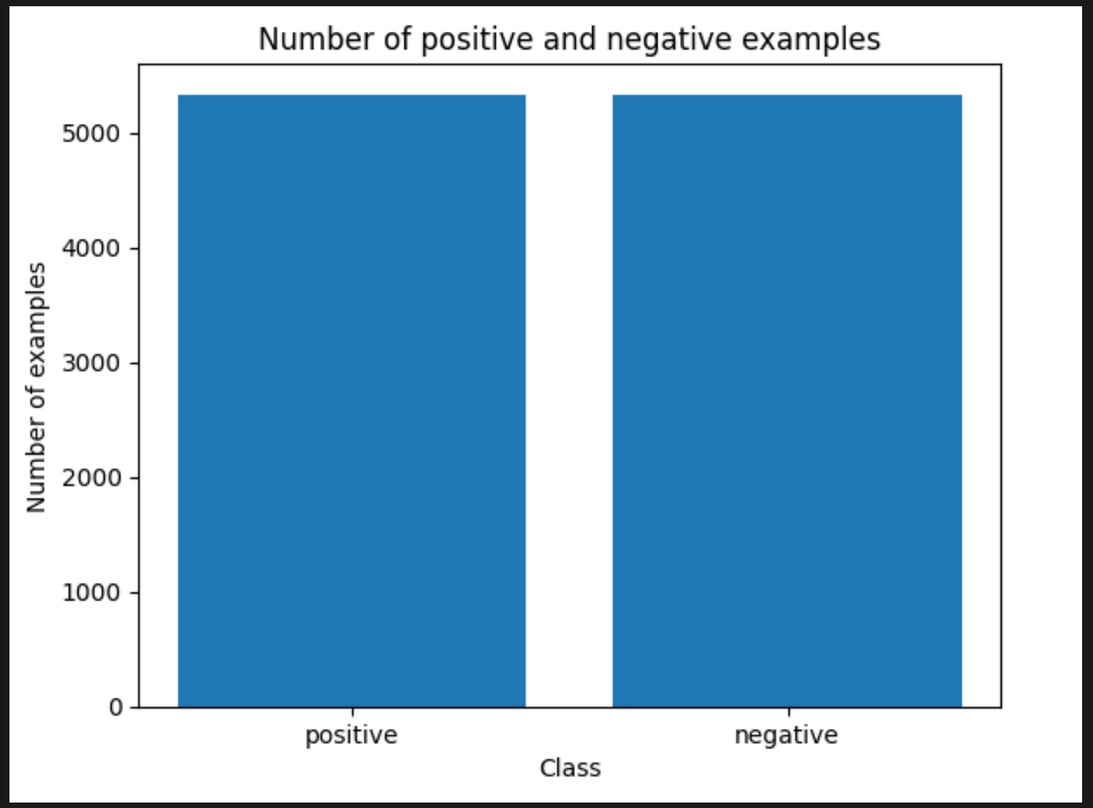
	Classifier	Accuracy	Precision	Recall	F1 Score
0	MultinomialNB	0.474667	0.525153	0.474262	0.462332
1	LogisticRegression	0.437667	0.475013	0.436982	0.424423

SUPERVISED LEARNING

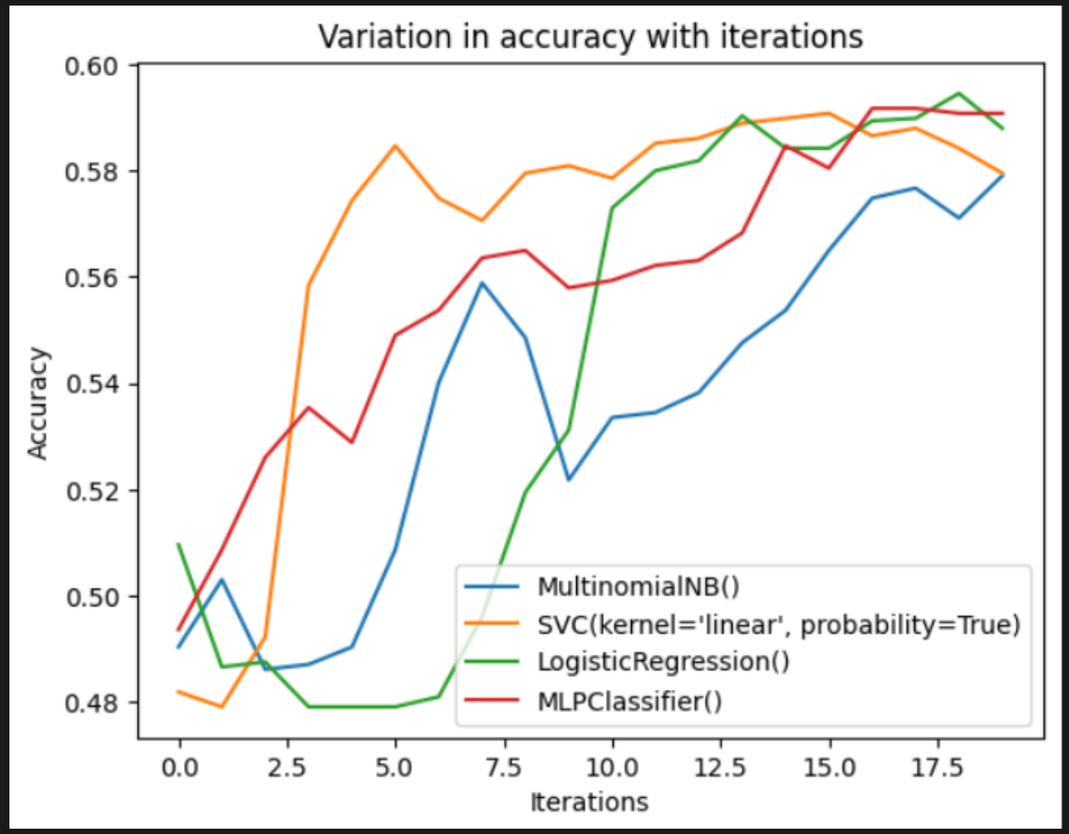
	Classifier	Accuracy	Precision	Recall	F1 Score
0	MultinomialNB	0.280333	0.280497	0.27808	0.274581
1	LogisticRegression	0.284667	0.283885	0.282199	0.277191

SENTENCE POLARITY



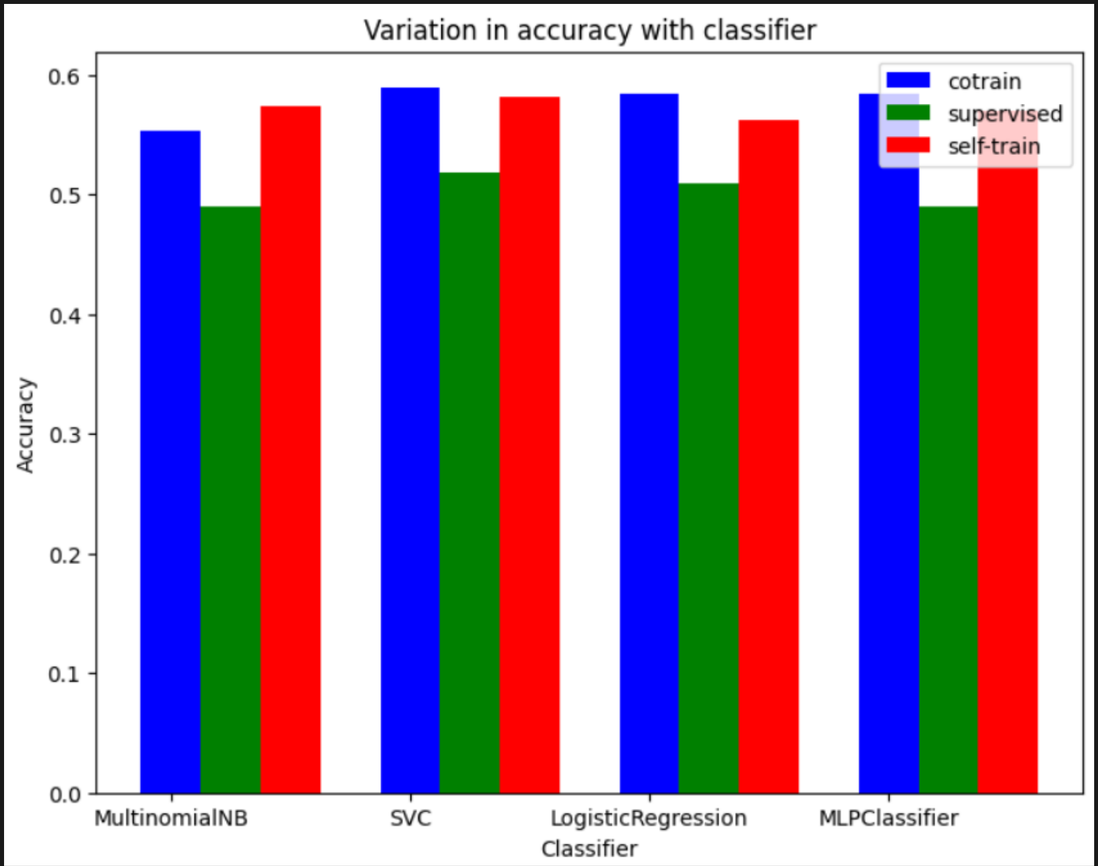


SENTENCE POLARITY



From the above graph, we can observe that with increasing the number of iterations, the accuracy of the classifiers is increasing.

SENTENCE POLARITY



- From the graph, we can observe that the semi-supervised training methods co-training and self-training outperform the supervised training. In MLP Classifier, Logistic Regression, and SVM, co-training outperforms self-training. In the case of Multinomial Naive Bayes self-training is performing better than co-training.
- In the previous graph, we observed that by increasing the number of iterations accuracy can be increased. So co-training method may outperform self-training by increasing the number of iterations in the case of Multinomial Naive Bayes.

COMPARING THE THREE METHODS

COTRAINING

	Classifier	Accuracy	Precision	Recall	F1 Score
0	MultinomialNB	0.55368	0.589762	0.564816	0.525593
1	SVC	0.58978	0.588879	0.586223	0.584782
2	LogisticRegression	0.584154	0.592074	0.588504	0.581495
3	MLPClassifier	0.584623	0.597468	0.590443	0.579069

SELF-TRAINING

	Classifier	Accuracy	Precision	Recall	F1 Score
0	MultinomialNB	0.574308	0.597188	0.58258	0.561028
1	SVC	0.58181	0.601183	0.589193	0.57187
2	LogisticRegression	0.562588	0.58756	0.571643	0.545636
3	MLPClassifier	0.56962	0.581404	0.575454	0.563771

SUPERVISED LEARNING

	Classifier	Accuracy	Precision	Recall	F1 Score
0	MultinomialNB	0.490389	0.492066	0.492107	0.490172
1	SVC	0.518519	0.511553	0.509311	0.48859
2	LogisticRegression	0.509611	0.499513	0.499624	0.473999
3	MLPClassifier	0.489451	0.498651	0.498927	0.466761

THANK YOU