SMAI PROJECT REPORT

TEAM - GALAXY

Team Members

Kukkapalli Shravya - 2021101051

Penumalla Aditya Pavani - 2021101133

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).

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.

For example, the description of a web page can be partitioned into the words occurring on that page, and the words occurring in hyperlinks that point to that page. Assuming that either view of the example would be sufficient for learning if there is enough labeled data, using both views together to allow inexpensive unlabeled data to augment a much smaller set of labeled examples could be helpful. Specifically, the presence of two distinct views of each example suggests strategies in which two learning algorithms are trained separately on each view, and then each algorithm's predictions on new unlabeled examples are used to enlarge the training set of the other.

The algorithm is implemented for text classification using the bag of words model. The algorithm is tested on the WebKB dataset and the News Category dataset. The algorithm is also tested on the Sentence Polarity dataset to check the performance of the algorithm on a binary classification problem.

Co-Train Algorithm:

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.

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'

The algorithm stated in the paper is as follows:

- For each example, webpage **x**, **x_1** was considered to be the bag of words appearing on the web page, and **x_2** was considered to be the bag of words underlined in all links pointing to the web page from other pages in the database. Classifiers were trained separately for **x_1** and for **x_2**, using the naive Bayes algorithm.
- Consider a set $**L**$ of labeled examples and a set $**U**$ of unlabeled examples.
- The algorithm first creates a smaller pool **U'** containing **u** unlabeled examples. It then iterates the following procedure.

- First, use **L** to train two distinct classifiers: **h_1** and **h_2**. **h_1** is a naive Bayes classifier based only on the **x_1** portion of the instance and **h_2** is a naive Bayes classifier based only on the **x_2** portion.
- Second, allow each of these two classifiers to examine the unlabeled set **U'** and select the **p** examples it most confidently labels as positive and the **n** examples it most confidently labels negative.
- Each example selected in this way is added to **L** along with the label assigned by the classifier that selected it.
- Finally, the pool **U** is replenished by drawing **2p+2n** examples from **U** at random.

Different Datasets Used:

1. **WebKB Binary (https://www.cs.cmu.edu/~webkb/)**

   This data set contains a subset of the WWW pages collected from computer science departments of various universities.

- The 1051 pages were manually classified into the following categories.
  - Course (230)
  - Non-Course (821)
- The files are organized into a directory structure with two directories at the top level
  - Fulltext - This directory contains the text on the web pages
  - Inlinks - This directory contains the anchor text on the hyperlinks pointing to the page.
- Under each of the two directories, there is one directory for each class (course, non-course). These directories in turn

contain the Web pages. The file name of each page corresponds to its URL, where '/' was replaced with '^'.

2. **WebKB MultiClass (https://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/)**

   Following the completion of the binary classification part, we extended our project to work with datasets where we have more than two categories to classify (multi-class classification).

   This data set contains WWW pages collected from computer science departments of various universities.

- The 8,282 pages were manually classified into the following categories:
  - Student (1641)
  - Faculty (1124)
  - Staff (137)
  - Department (182)
  - Course (930)
  - Project (504)
  - Other (3764)
- For each class, the data set contains pages from the four universities
  - Cornell (867)
  - Texas (827)
  - Washington (1205)
  - Wisconsin (1263)

  and 4,120 miscellaneous pages collected from other universities.

- The files are organized into a directory structure, one directory for each class. Each of these seven directories contains 5 subdirectories, one for each of the 4 universities and one for the miscellaneous pages. These directories in turn contain the Web pages. The file name of each page corresponds to its URL, where '/' was replaced with '^'.
- The data used for task 1 has 2 views and each is classified into either course or non-course.
- The data above has only one view.. i.e full text.
- So, the above data is used to separate the non-course data from task 1 into respective classes (Student, Faculty, Staff, Project).

3. **News Category Dataset (https://www.kaggle.com/datasets/rmisra/news-category-dataset)**

   This dataset contains around 210k news headlines from 2012 to 2022 from HuffPost. This is one of the biggest news datasets and can serve as a benchmark for a variety of computational linguistic tasks.

   Each record in the dataset consists of the following attributes:

- Category: category in which the article was published.
- Headline: the headline of the news article.
- Authors: list of authors who contributed to the article.
- Link: link to the original news article.
- Short_description: Abstract of the news article.
- Date: publication date of the article.

There are a total of 42 news categories in the dataset. The top 15 categories are taken in the project and corresponding article counts are as follows:

POLITICS: 35602

WELLNESS: 17945

ENTERTAINMENT: 17362

TRAVEL: 9900

STYLE & BEAUTY: 9814

PARENTING: 8791

HEALTHY LIVING: 6694

QUEER VOICES: 6347

FOOD & DRINK: 6340

BUSINESS: 5992

COMEDY: 5400

SPORTS: 5077

BLACK VOICES: 4583

To handle the imbalance in the data, we have taken 1000 random samples from each class and performed the algorithm.

The two views in the dataset are Headlines and Short Descriptions.

4. Sentence Polarity Dataset

rt-polaritydata.tar.gz: contains two data files

Specifically:

* rt-polarity.pos contains 5331 positive snippets

* rt-polarity.neg contains 5331 negative snippets

Each line in these two files corresponds to a single snippet (usually containing roughly one single sentence); all snippets are down-cased.

This dataset is used to create the antonymous view of each of the data samples. This way another view was created, one view being the original dataset without any changes.

Comparison with the results mentioned in the Paper

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

The results obtained:

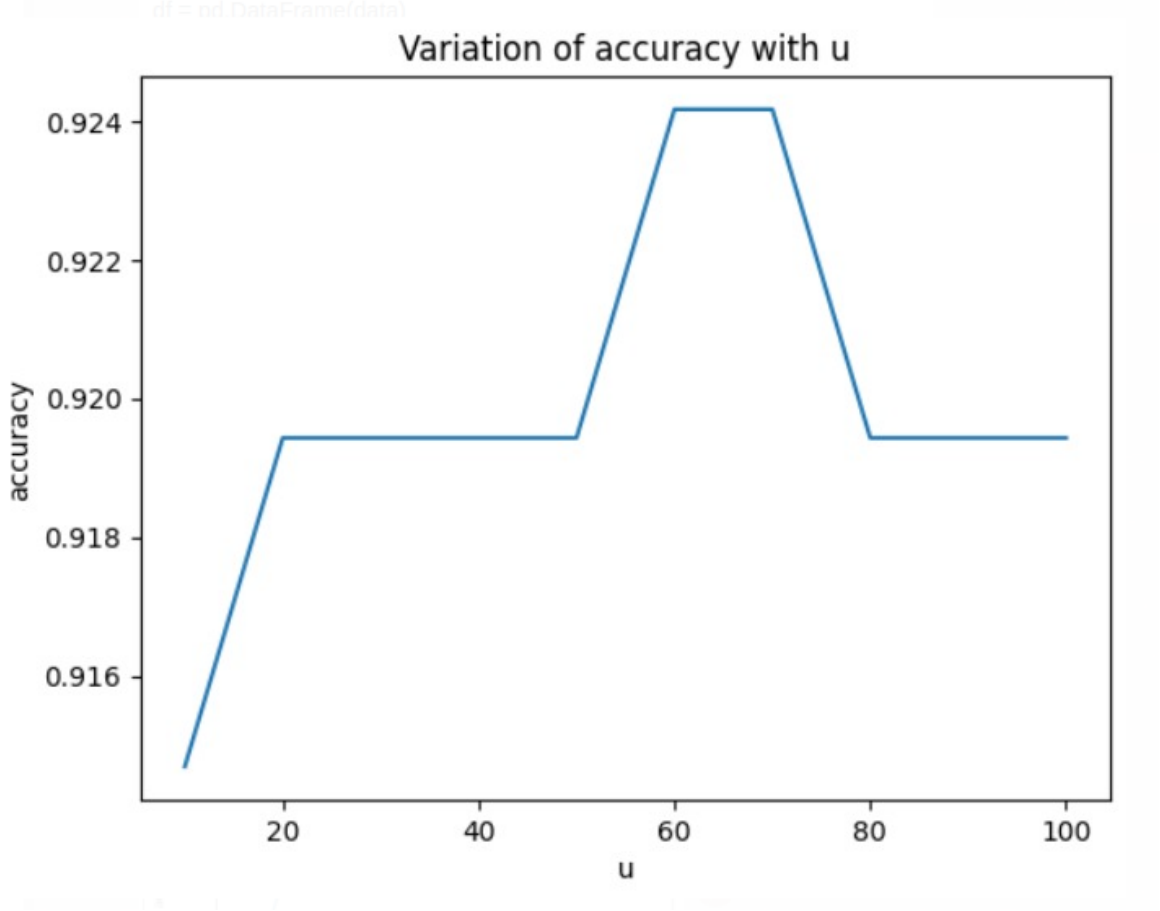
	Algorithm	page-based	link-based	combined
0	co-training	7.58294	40.2844	8.53081
1	supervised learning	14.6919	14.6919	14.6919

The 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

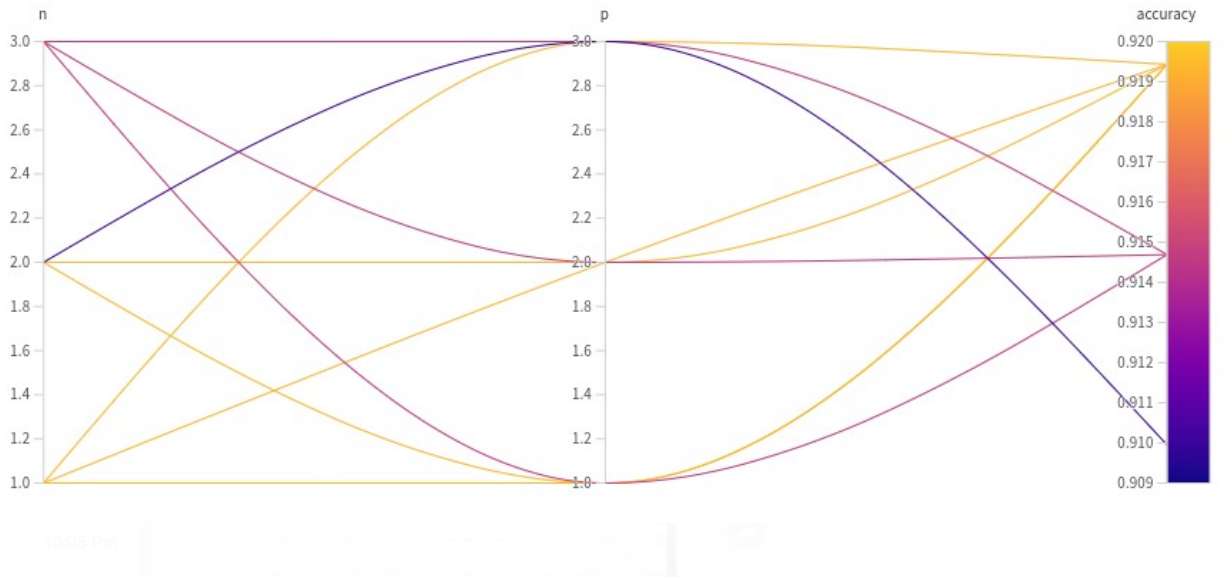
The difference in the error rates might be due to different methods of preprocessing data (tokenization, lemmatization, etc).

The below graph shows the variation of accuracy with ‘u’, where u is the number of unlabelled examples.



From the above graph, we can observe that as the value of u increases at first accuracy increases reaches a maximum and then decreases.

WandB Sweep for different values of n and p

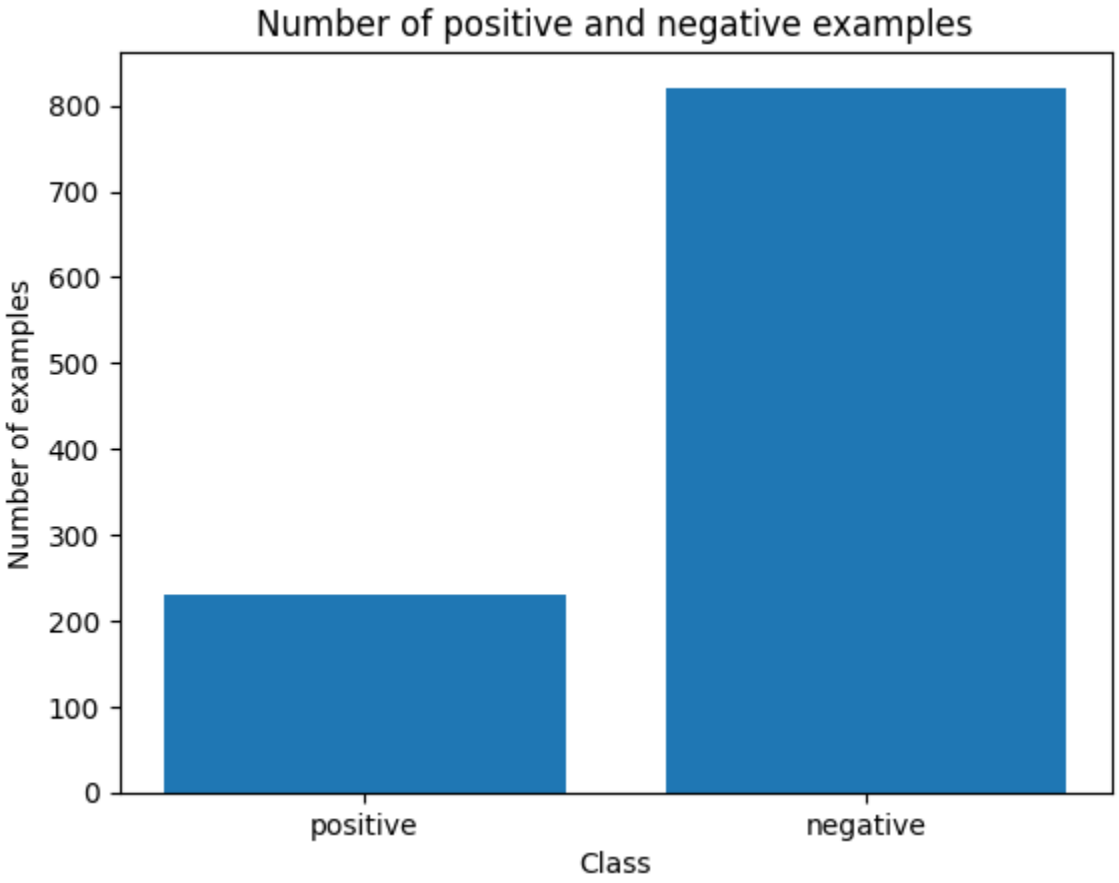


The accuracy is maximum for the following (n,p) pairs:
(1,3); (1,2); (2,2); (1,1)

Analysis and Results

Dataset 1

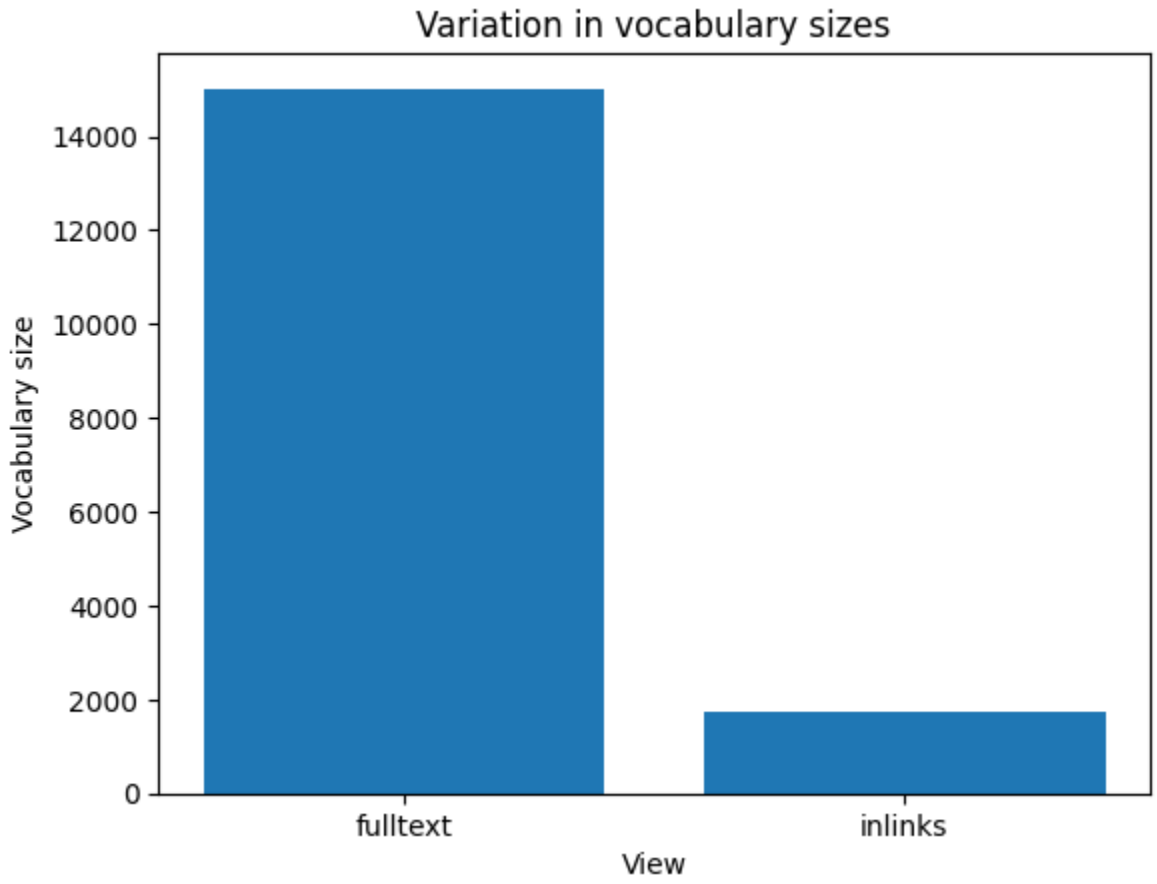
The below figure shows the distribution of data samples between the two classes.



Observation:

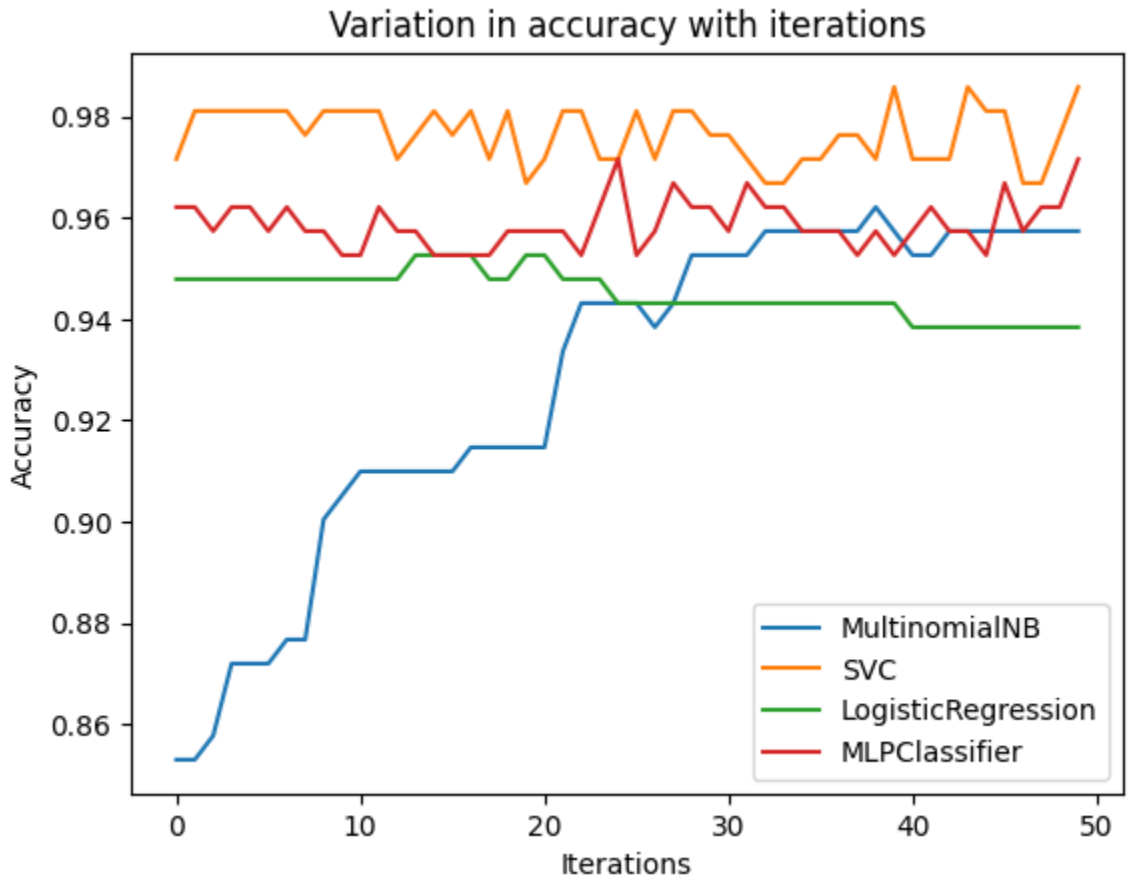
From the graph, we can observe that the number of negative examples is 4 times that of the positive examples.

The graph below shows the variation in vocabulary sizes of the two different views (FullText and Inlinks).



Observation:
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.

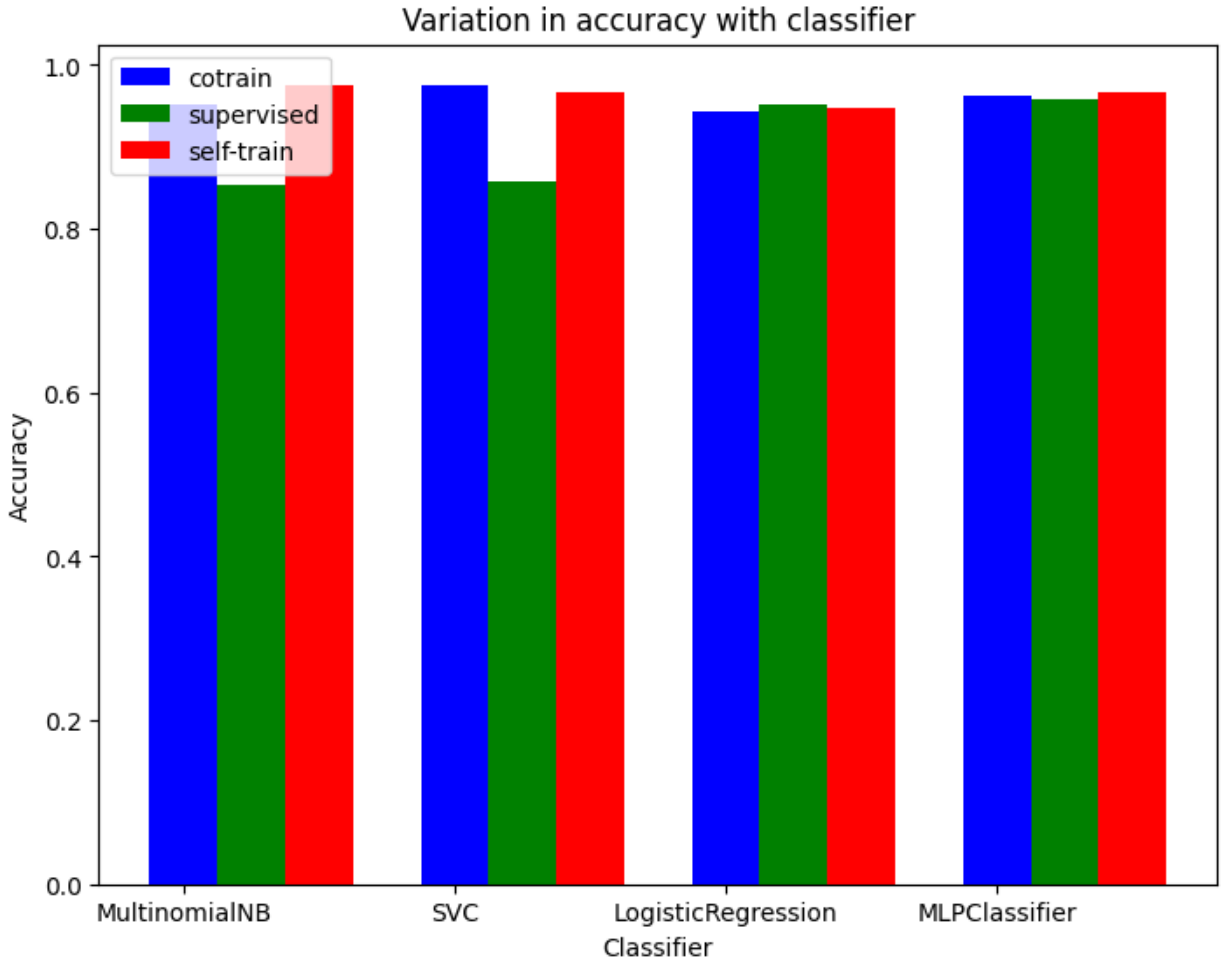
The below graph shows the variation in accuracy with the number of iterations for co-train.



Observation:

From the above graph, as the number of iterations increases the performance of Multinomial Naive Bayes increases whereas, for Multinomial Logistic Regression at first, it increases and then starts decreasing. For the other two classifiers, there is no significant change in performance by increasing or decreasing the number of iterations.

The below graph shows the variation in accuracy with the Classifier.



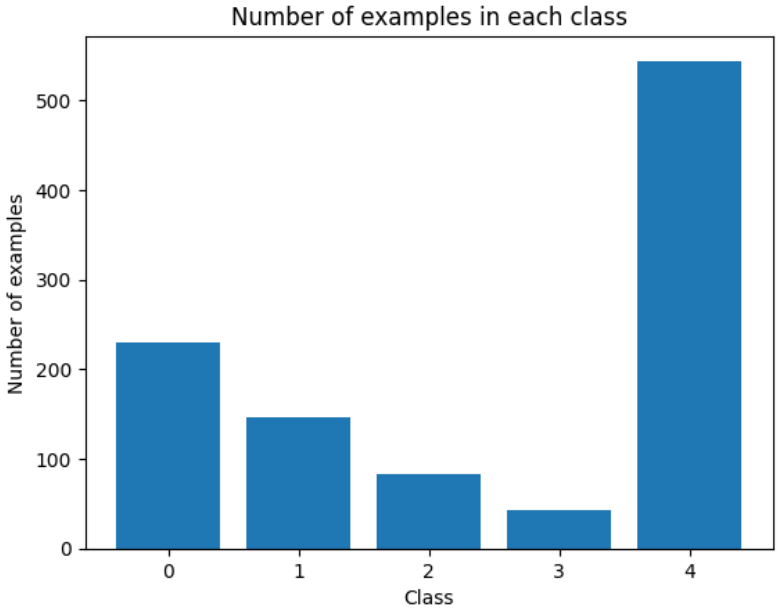
Observation:

Using the Naive Bayes classifier and SVM, co-training and self-training outperform supervised training. In the case of MLP and logistic regression, the three methods are almost the same.

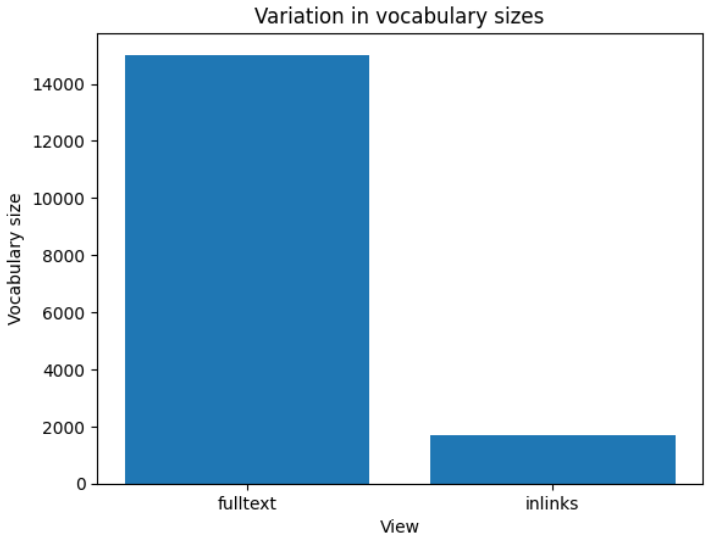
By increasing the number of iterations co-training may outperform self-training in the case of Naive Bayes.

Dataset 2

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



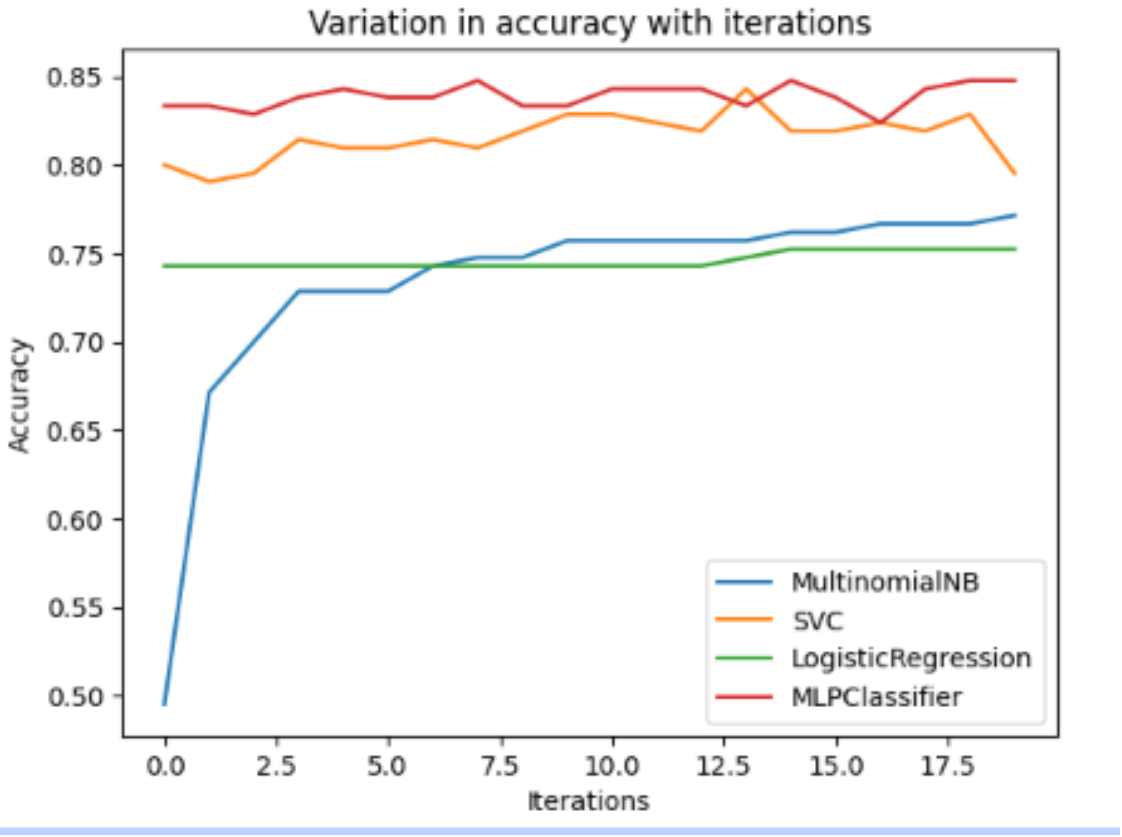
The graph below shows the variation in vocabulary sizes of the two different views (FullText and Inlinks).



Observation

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.

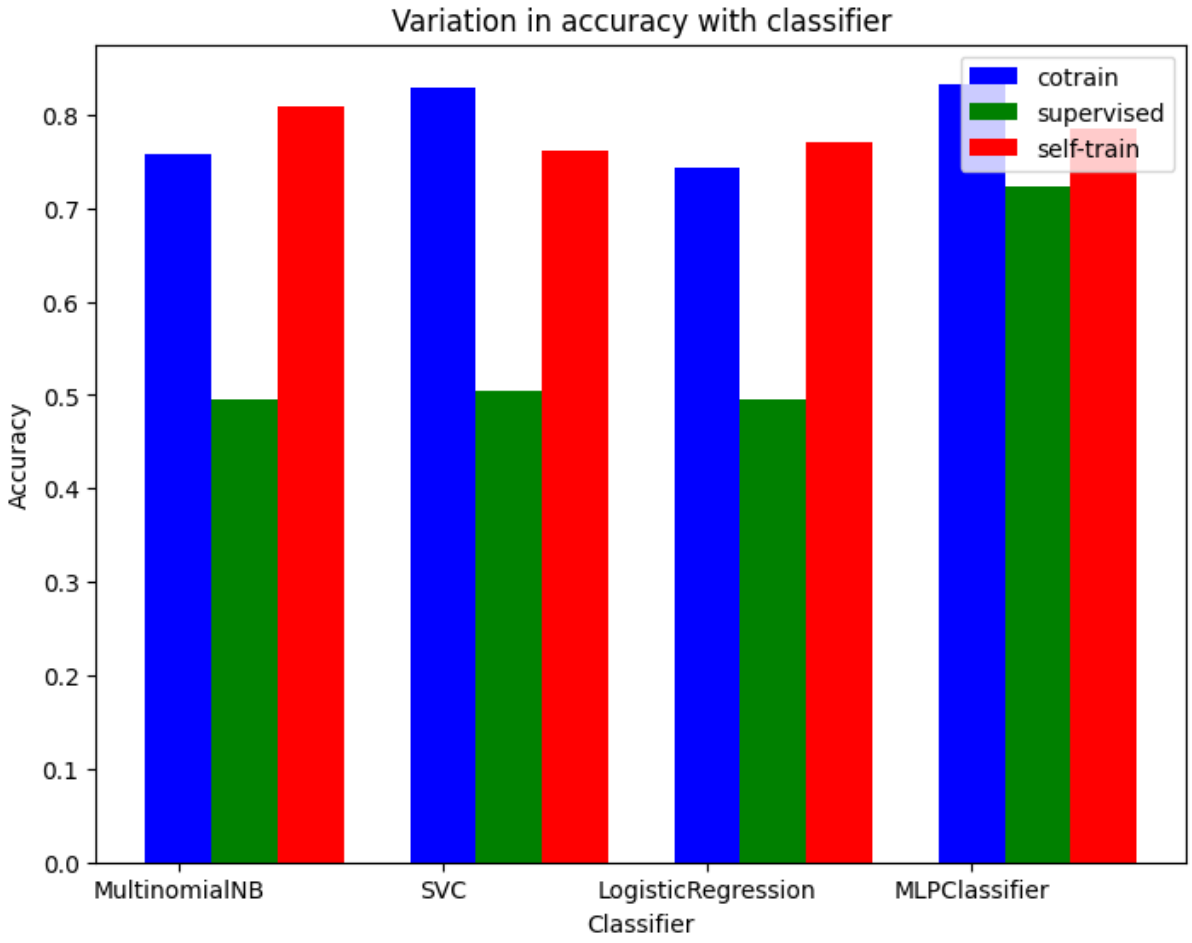
The below graph shows the variation in accuracy with the number of iterations for co-train.



Observation

With increasing the number of iterations, the performance of the Naive Bayes classifier increases. But, the performance of the other three classifiers is almost constant.

The below graph shows the variation in accuracy with the Classifier.

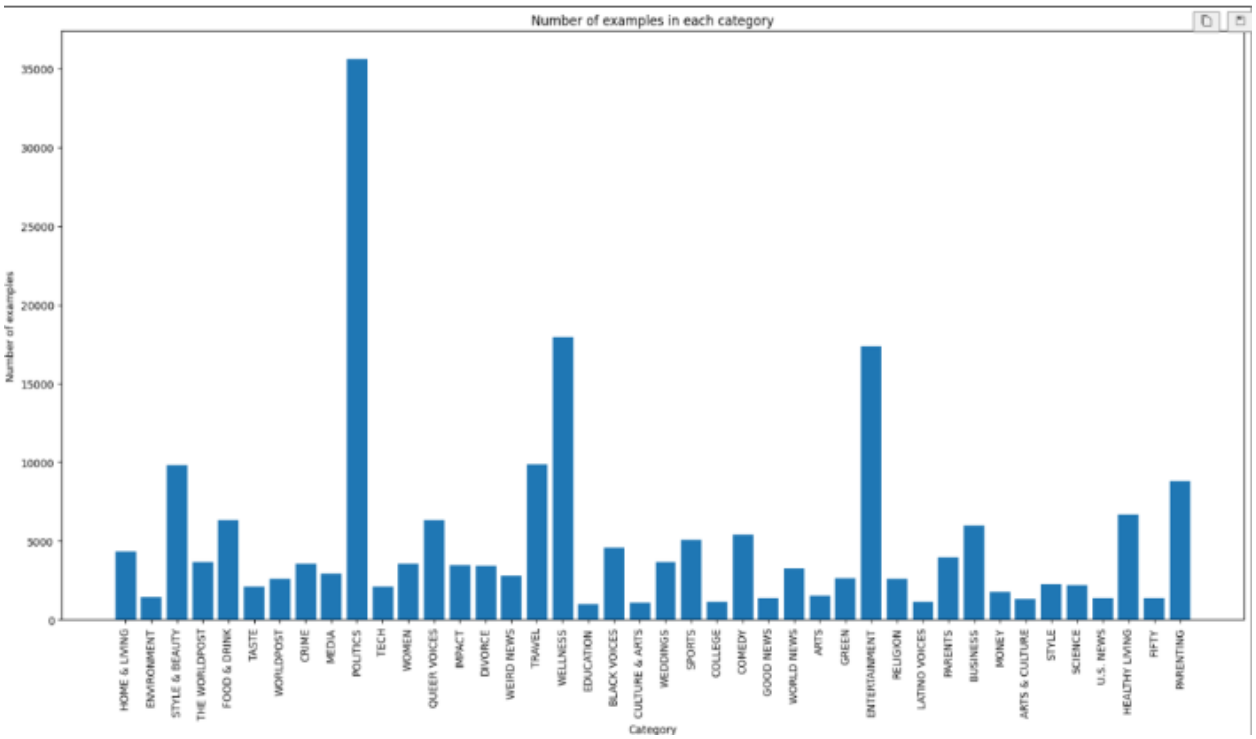


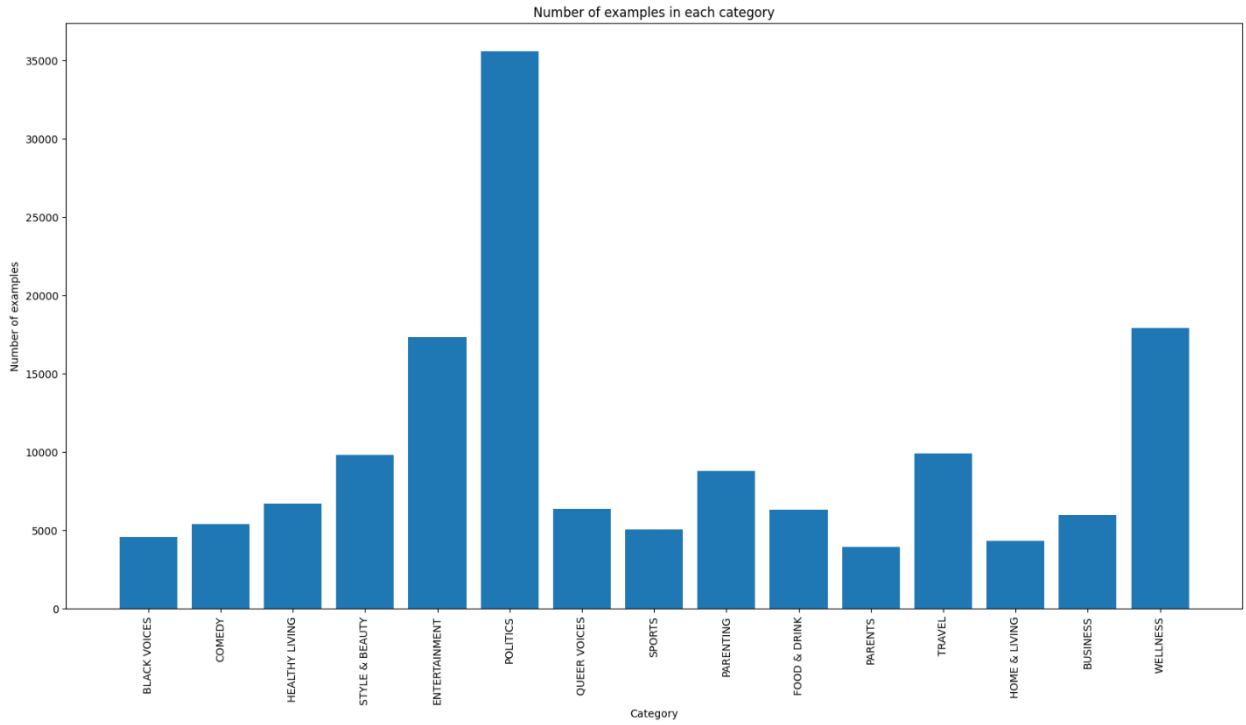
Observation

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

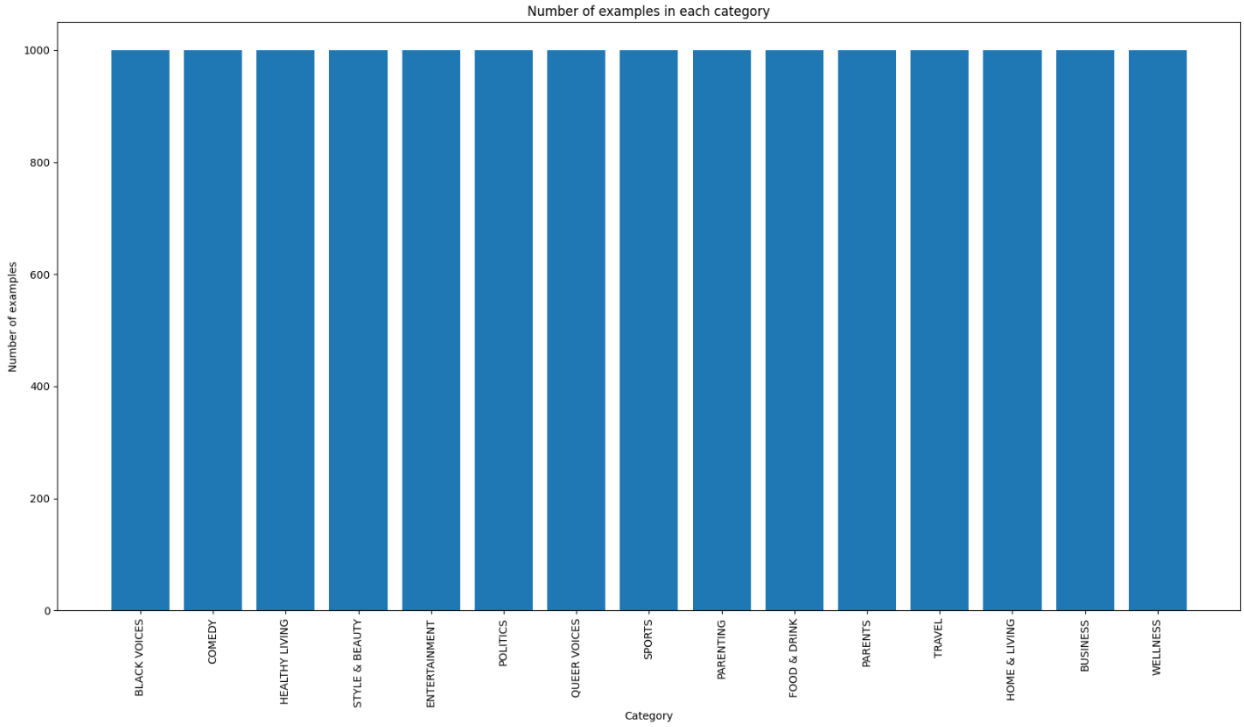
In the previous graph, we observed that with increasing the number of iterations performance of Multinomial Naive Bayes using the co-training method can be improved.
So in the case of Naive Bayes classifier, by increasing the number of iterations co-training method may outperform self-training.

Dataset 3

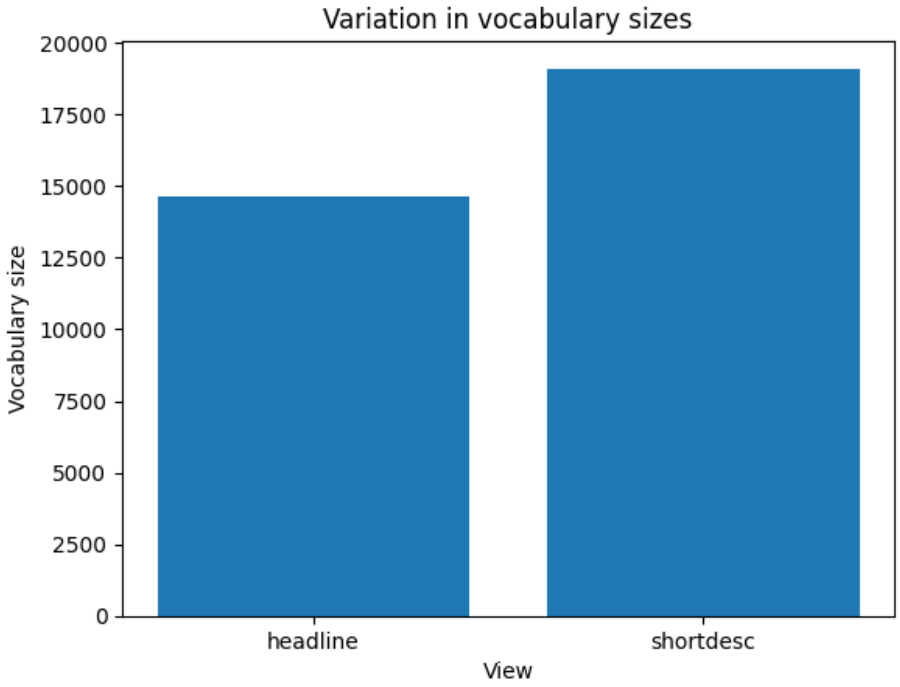

Number of examples in each category
Number of examples
Category
0
5000
10000
15000
20000
25000
30000
35000
HOME & LIVING
ENVIRONMENT
STYLE & BEAUTY
THE WORLDPOST
FOOD & DRINK
TASTE
WORLDPOST
CRIME
MEDIA
POLITICS
TECH
WOMEN
QUEER VOICES
IMPACT
DIVORCE
WEIRD NEWS
TRAVEL
WELLNESS
EDUCATION
BLACK VOICES
CULTURE & ARTS
WEDDINGS
SPORTS
COLLEGE
COMEDY
GOOD NEWS
WORLD NEWS
ARTS
GREEN
ENTERTAINMENT
RELIGION
LATINO VOICES
PARENTS
BUSINESS
MONEY
ARTS & CULTURE
STYLE
SCIENCE
U.S. NEWS
HEALTHY LIVING
FIFTY
PARENTING


Number of examples in each category
Number of examples
Category
0
5000
10000
15000
20000
25000
30000
35000
BLACK VOICES
COMEDY
HEALTHY LIVING
STYLE & BEAUTY
ENTERTAINMENT
POLITICS
QUEER VOICES
SPORTS
PARENTING
FOOD & DRINK
PARENTS
TRAVEL
HOME & LIVING
BUSINESS
WELLNESS

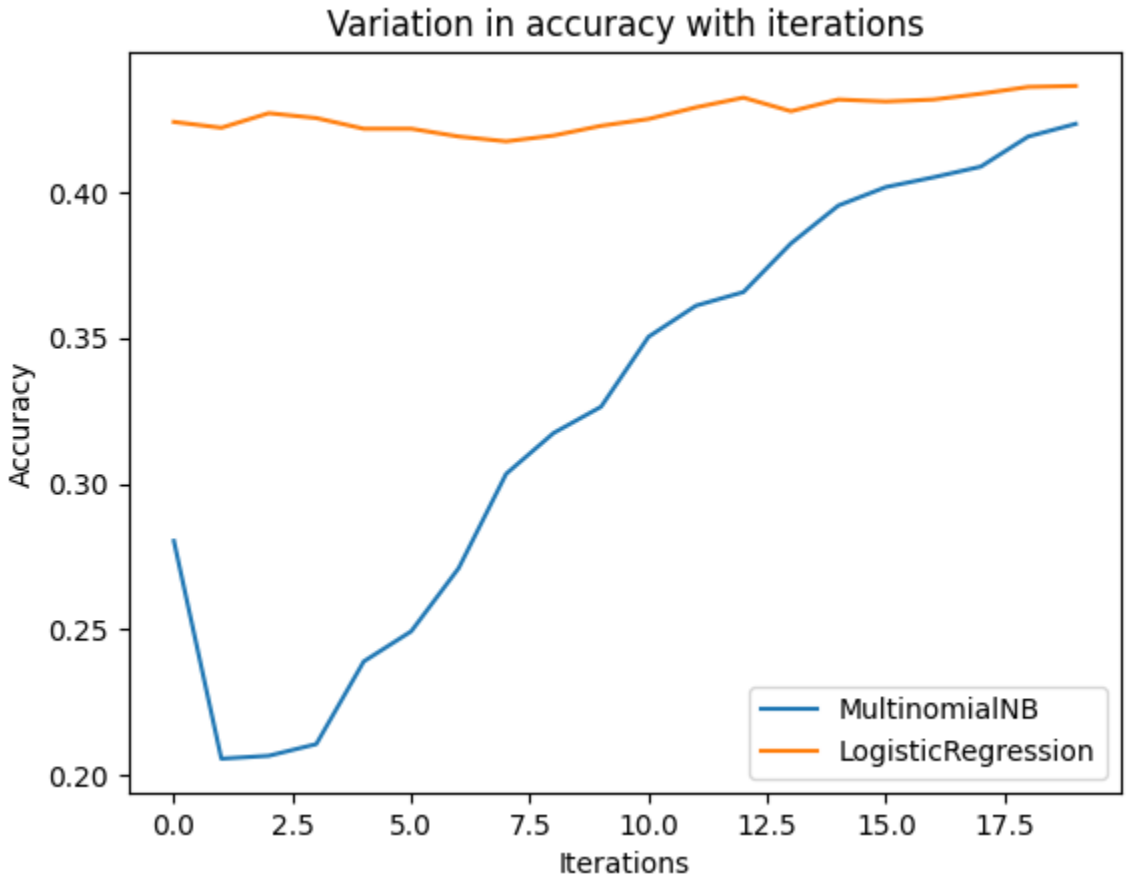
The above data contains 42 classes. We have taken top 15 classes from the data.



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

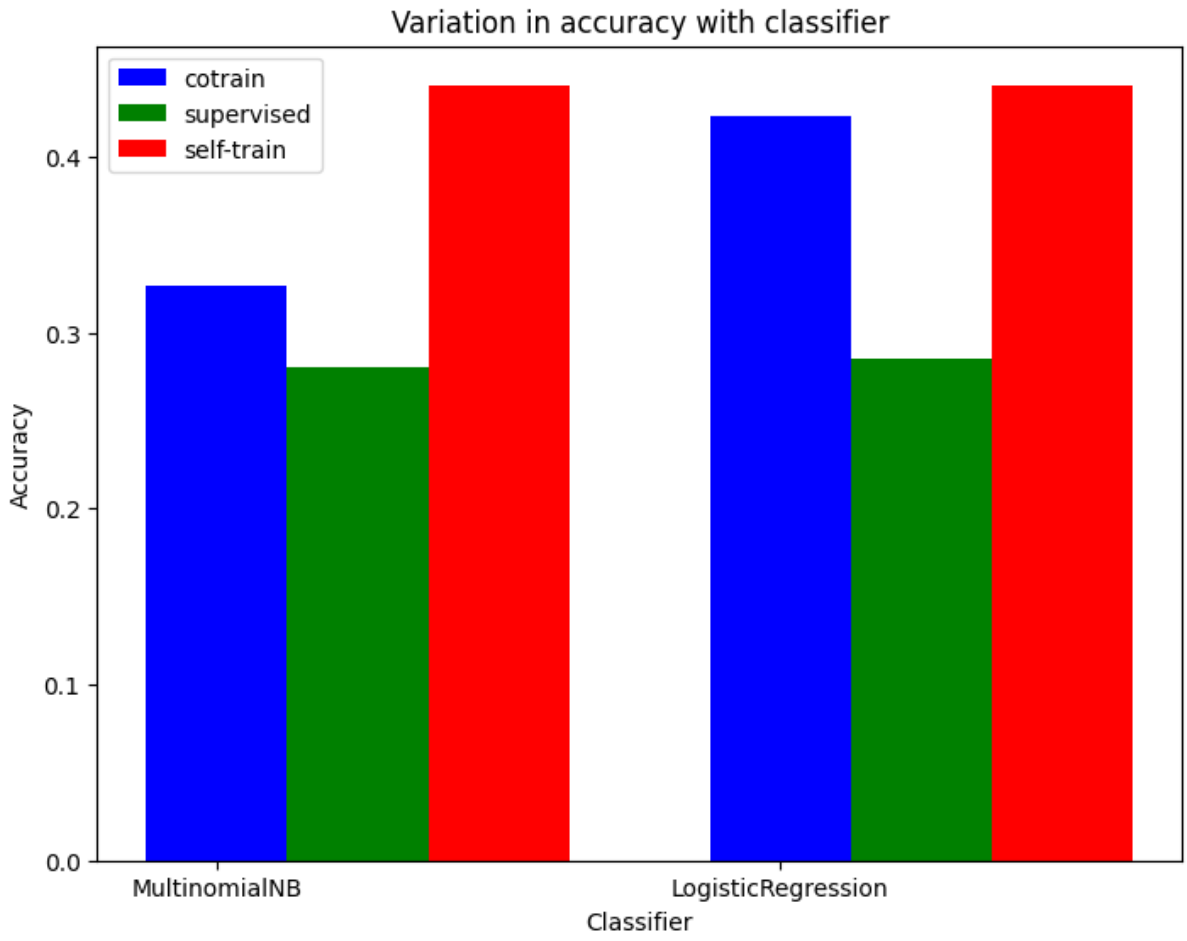


The below graph shows the variation in accuracy with the number of iterations for co-train.



By increasing the number of iterations the performance of Naive Bayes is increasing significantly while the performance of Logistic regression is slightly increasing.

The below graph shows the variation in accuracy with the Classifier.

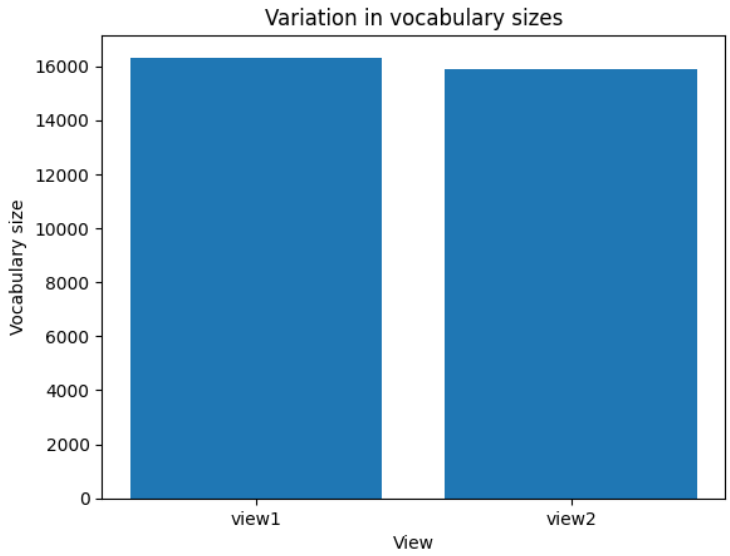


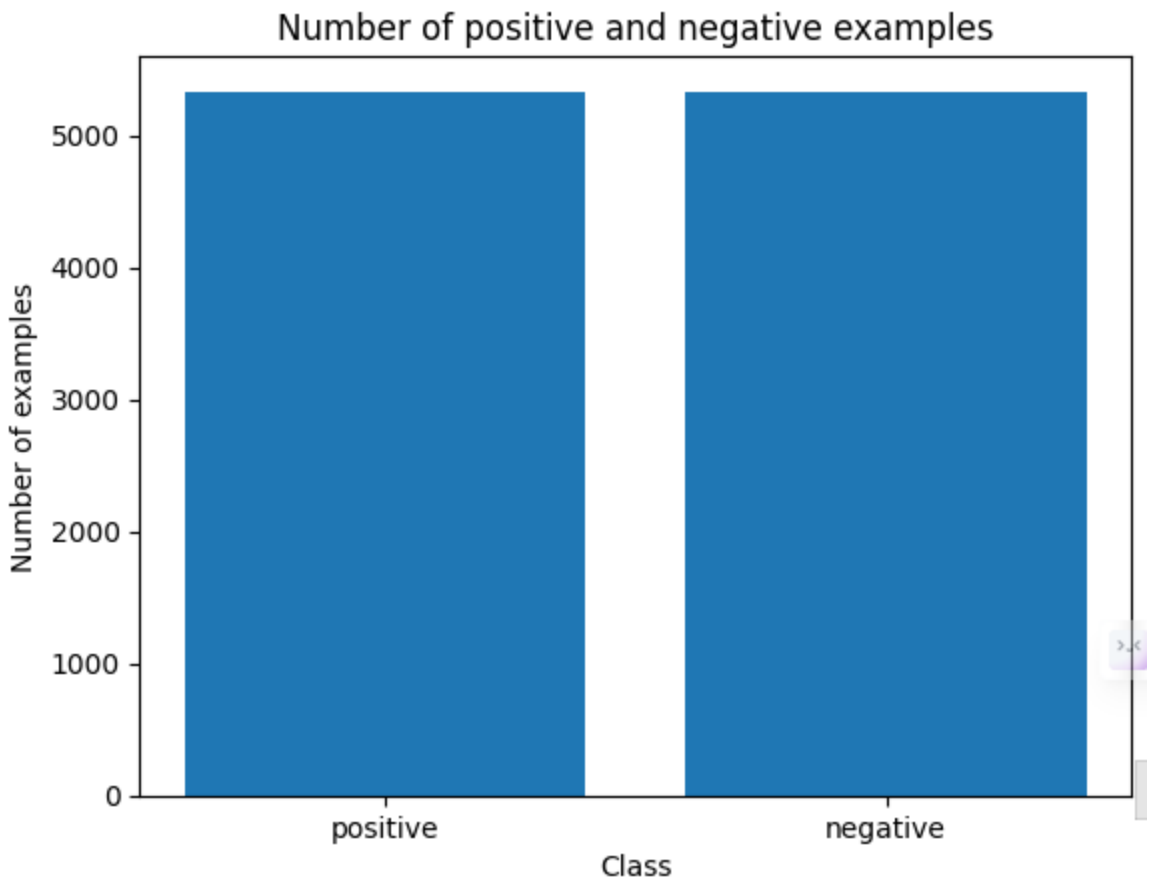
Observations:

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 self-training outperforms co-training significantly and in the case of Logistic regression both are similar.

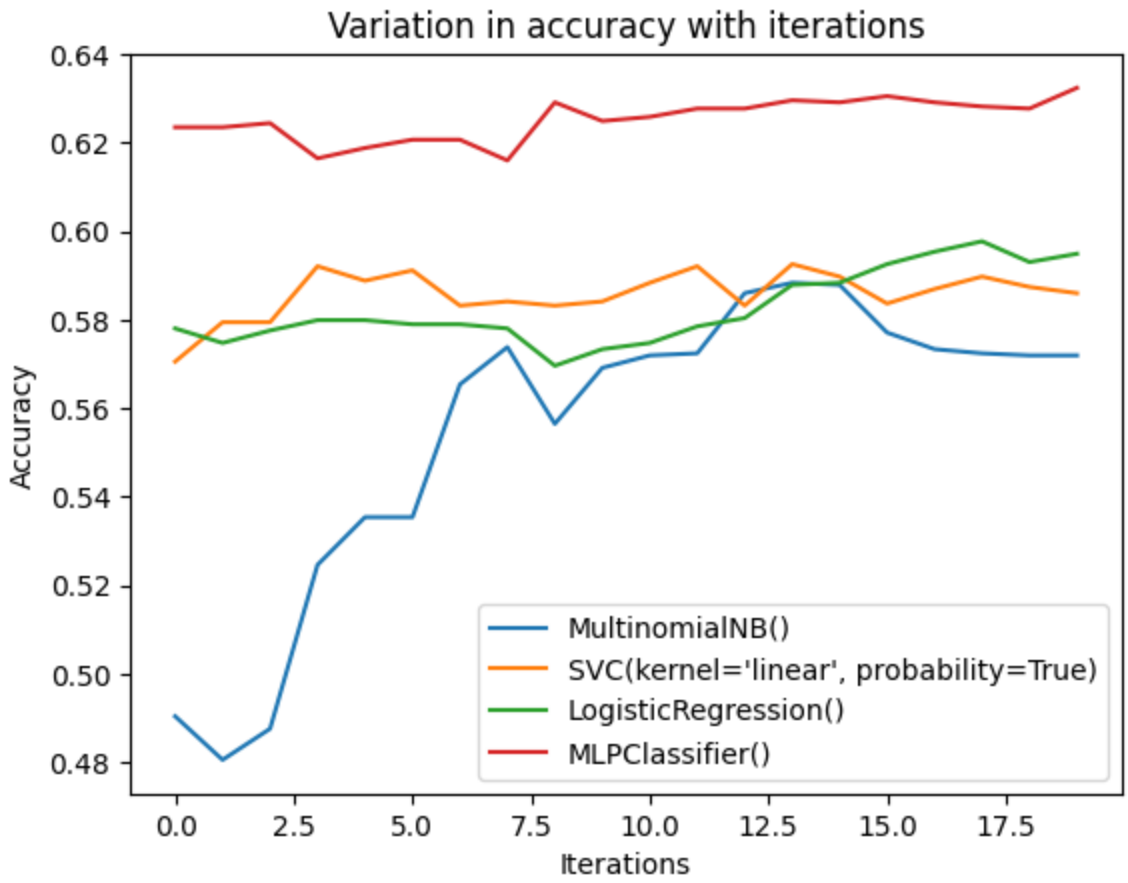
Based on the previous graph, the performance of co-training can be increased for both classifiers by increasing the number of iterations.

Dataset 4


Variation in vocabulary sizes
Vocabulary size
16000
14000
12000
10000
8000
6000
4000
2000
0
view1
view2
View



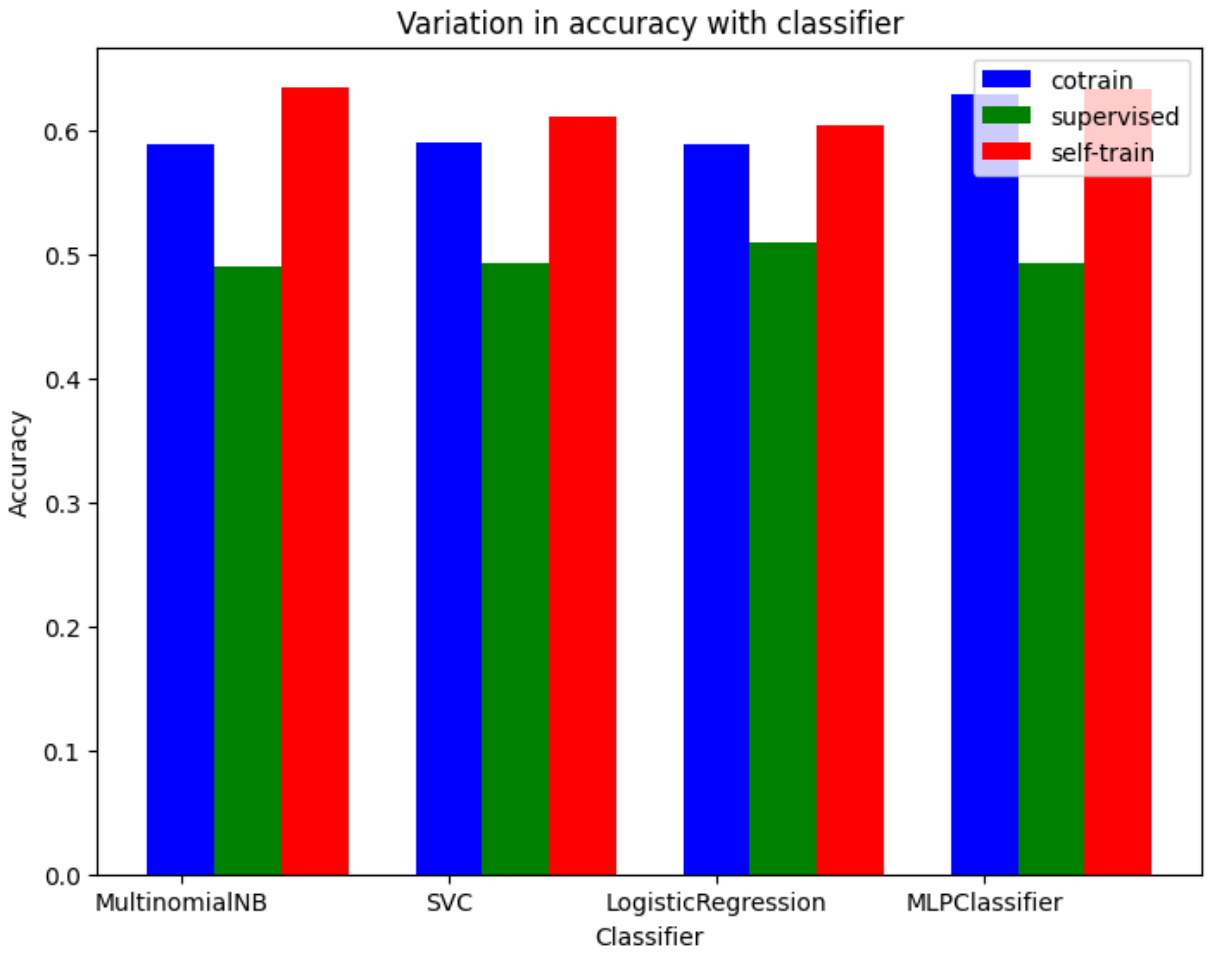
The below graph shows the variation in accuracy with the number of iterations for co-train.



Observations

From the above graph, we can observe that with increasing the number of iterations, the performance of MLP and Logistic regression slightly increases. And the performance of Naive Bayes increases with iterations and then slightly decreases. The performance of SVM doesn't vary much.

The below graph shows the variation in accuracy with the Classifier.



Observation

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

In the previous graph, we observed that by increasing the number of iterations performance of Multinomial Naive Bayes, MLP and logistic regression can be increased using the co-training method. So co-training method may outperform self-training by increasing the number of iterations in the case of MLP, Logistic Regression, and Naive Bayes.