Assignment 2.

Clustering, Classification and Topic Modeling

MSDS 453: Natural Language Processing

Husein Adenwala

Northwestern University

1. Introduction & Problem Statement

The purpose of this assignment is to perform unsupervised clustering analysis, sentiment analyses and Topic modeling on the 249 movie reviews. The unsupervised clustering analysis is done to decipher structure and patterns in the data and to see if the assigned groups in the data reflect the clustering pattern. I will also evaluate the ability of various classification algorithms to classify positive and negative sentiment/reviews in the data. I will also perform Topic Modeling using LDA and LSA techniques to understand the word patterns in the documents and to see if they correlate within the movie labels.

In this research, I will experiment with K means DBSCAN and agglomerative hierarchical clustering techniques on the TF-IDF vectors and Doc2Vec embedding vectors. I will also experiment with SVM, Naïve Bayes, Random Forest and Logistic Regression classification algorithms for sentiment analysis and finally, I will evaluate LSA and LDA topic modeling methods to understand and compare patterns of topics in the data.

2. Data preparation, exploration, visualization

The data consists of 249 movie reviews of 25 movies that was stored and uploaded in a csv file. The tables and plots below show the movies' names, the number of movie reviews and movie genres.

All 25 movies have 10 reviews except the movie Martian and Red Notice, which have 9 and 20 reviews respectively. The number of positive and negative reviews are equal for all movies except Martian. Also, it is important to note that the count for each movie genre is unbalanced; the Drama genre has only 9 reviews.

Movie Name	Number of Reviews
Arrival	10
CasinoRoyale	10
Cruella	10
Encanto	10
Frozen2	10
GroundhogDay	10
GuardiansOfTheGalaxy	10
Hereditary	10
Interstellar	10
KingOfStatenIsland	10
Lamb	10
MissionImpossibleFallout	10
PACIFICRIM	10
PiratesOfTheCaribbean:TheCurseOfTheBlackPearl	10
PiratesOfTheCarribean	10
Poltergeist	10
RedNotice	20
SpeedRacer	10
SpiderMan3	10
TheConjuring3	10
TheMartian	9
TheMatrixResurrecton	10
TheRing	10
Us	10

Genre of Movie	Number of Reviews
Action	70
Comedy	60
Drama	9
Horror	60
Sci_Fi	50

```
Total word count:  198977
```

Movie Title	Review Type (pos or neg)	
Arrival	Negative	5
	Positive	5
Casino_Royale	Negative	5
	Positive	5
Cruella	Negative	5
	Positive	5
Encanto	Negative	5
	Positive	5
Frozen_2	Negative	5
	Positive	5
Groundhog_Day	Negative	5
	Positive	5
Guardians_of_the_Galaxy	Negative	5
	Positive	5
Hereditary	Negative	5
	Positive	5
Interstellar	Negative	5
	Positive	5
KingOfStatenIsland	Negative	5
	Positive	5
Lamb	Negative	5
	Positive	5
Mission_Impossible_Fallout	Negative	5
	Positive	5
Pacific_Rim	Negative	5
	Positive	5
Pirates_of_the_Caribbean_Curse_of_the_Black_Pearl	Negative	5
	Positive	5
Poltergeist	Negative	5
	Positive	5
Red_Notice	Negative	10
	Positive	10
Speed_Racer	Negative	5
	Positive	5
Spider_Man_3	Negative	5
	Positive	5
Sudden_Impact	Negative	5
	Positive	5
The Martian	Negative	5
	Positive	4
The Ring	Negative	5
	Positive	5
The_Conjuring_The_Devil_Made_Me_Do_It	Negative	5
	Positive	5
The_Matrix_Resurrecton	Negative	5
	Positive	5
Us_2019	Negative	5
	Positive	5

Data Processing

All movie reviews were combined to create the Corpus. This corpus was tokenized and normalized by removing punctuation, stop word removal, stemming, lemmatization, lower cases and removing non-alphabet words.

The Corpus was tokenized using TF-IDF and Doc2Vec and Word2Vec embeddings.

3. Research Design and Modeling Method(s)

Step 1 Clustering

I performed K means, DBSCAN, and Agglomerative Clustering using the TFIDF and Doc2Vec word embedding and evaluated the clustering methods using the silhouette score, R^2, homogeneity score and by visualizing the elbow plot/method.

Step 2 Sentiment Analysis

I performed sentiment analysis by implementing SVM, Naïve Bayes, Random Forest and Logistic Regression models on TFIDF and Doc2Vec embeddings. The data for this was split between test and train in a 3:2 ratio. I also experimented with feature reduction/selection techniques using scikit learn select k best function which improved the accuracy. I used accuracy, F1 score, confusion matrix and an ROC curve to evaluate the models.

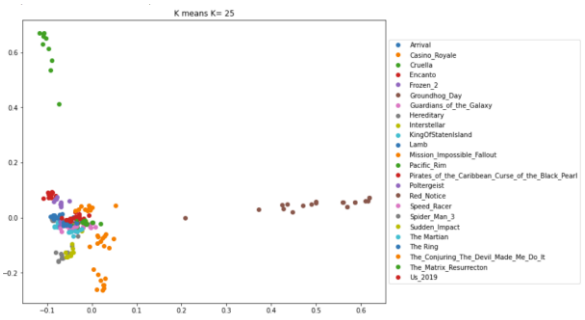
Step 3 Topic Modeling

I used LSA and LDA topic modeling methods on TFIDF and Word2Vec embeddings. I experimented with the number of topics and numbers of word parameters to get optimal coherence and perplexity score for the model.

4. Results

Clustering :

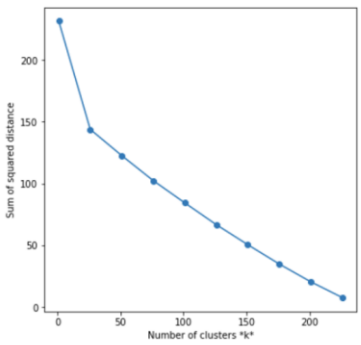
Using TFIDF, I found the optimal silhouette score of 0.14 did correspond to approximately 25 clusters. The plot below shows K-means clustering for the 25 groups which approximately correspond to the movie titles.

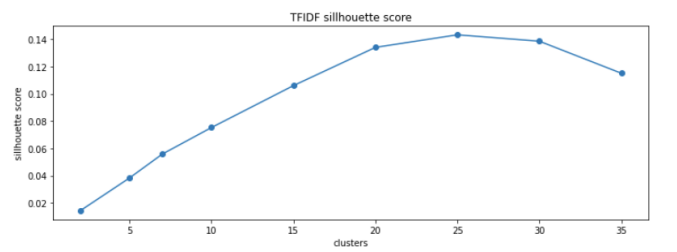


	cluster	Movie Title
0	0	Guardians_of_the_Galaxy
1	1	The_Conjuring_The_Devil_Made_Me_Do_It
2	2	Red_Notice
3	3	Poltergeist
4	4	Casino_Royale
5	5	Speed_Racer
6	6	KingOfStatenIsland
7	7	Frozen_2
8	8	The Ring
9	9	Groundhog_Day
10	10	Cruella
11	11	Pirates_of_the_Caribbean_Curse_of_the_Black_Pearl
12	11	The_Matrix_Resurrecton
13	12	Hereditary
14	13	Lamb
15	14	Mission_Impossible_Fallout
16	15	The_Matrix_Resurrecton
17	16	Spider_Man_3
18	17	Us_2019
19	18	Encanto
20	19	Sudden_Impact
21	20	Interstellar
22	20	The Martian
23	21	Encanto
24	22	Arrival
25	23	Pacific_Rim
26	24	Arrival

TF-IDF Matrix

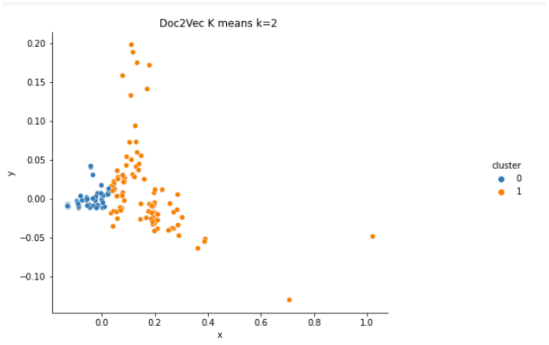
Elbow plot

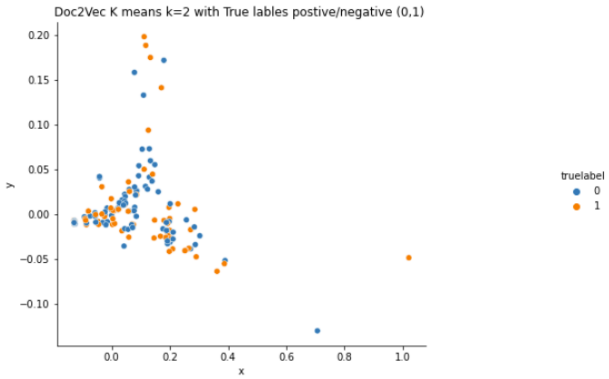




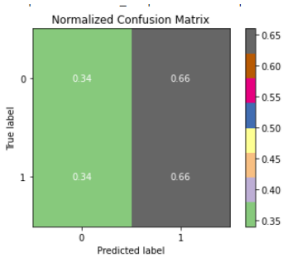
Using Doc2vec, I found the optimal silhouette score of 0.62 did correspond to 2 clusters. The plot below shows K-means clustering for the 2 groups. However, those groups don't correspond to the positive or negative reviews; it appears to be random as it clusters the positive and negative reviews accurately approximately 50% of the time.

Clustering, Classification and Topic Modeling

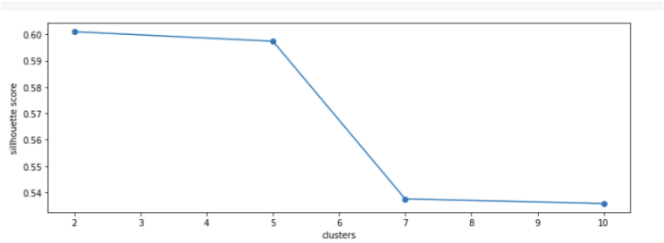




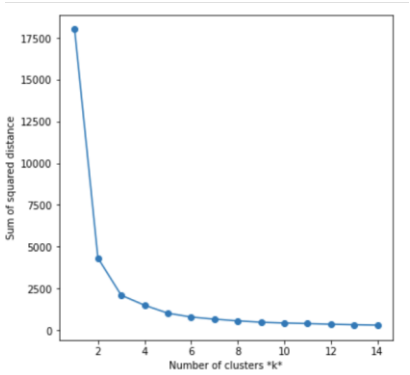
cluster	Review Type (pos or neg)	Count
0	Negative	42
0	Positive	42
1	Negative	83
1	Positive	82



Silhouette Score plot



Elbow plot



Sentiment Analysis:

I conducted 7 experiments for sentiment analysis with different classification algorithms and word embedding and dimension reduction techniques. The results are as follows:

Clustering, Classification and Topic Modeling

Method 1 Dimension Reduction using LDA. Num of topics = 47 based on opitmal cohernce . TFIDF

	Train Accuracy	Test Accuracy	F1	AUC	PR-AUC
SVM	72	47	38	46	46
Naïve Bayes	71	45	41	48	57
Random Forest	71	45	54	54	65
Logistic Regression	71	45	51	58	57

Method 2 TFIDF Tokens after cleaning data = 15479

	Train Accuracy	Test Accuracy	F1	AUC	PR-AUC
SVM	100	37	42	41	56
Naïve Bayes	100	49	48	49	61
Random Forest	100	63	65	64	73
Logistic Regression	100	51	52	53	63

Method 3 select best features using logistic regression classifier. Number of Token = 5141 TFIDF

	Train Accuracy	Test Accuracy	F1	AUC	PR-AUC
SVM	100	59	55	70	68
Naïve Bayes	100	81	80	82	87
Random Forest	100	59	63	68	72
Logistic Regression	100	87	86	72	90

Method 4 Doc2Vec embedding Tokens after cleaning data = 15479

	Train Accuracy	Test Accuracy	F1	AUC	PR-AUC
SVM	100	37	42	41	56
Naïve Bayes	100	49	48	48	61
Random Forest	100	55	60	62	69
Logistic Regression	100	51	51	53	63

Method 5 select best features using Random Forest Classifier. Number of Tokens = 3261 Bag of words Binary embedding

	Train Accuracy	Test Accuracy	F1	AUC	PR-AUC
SVM	100	66	38	76	77
Naïve Bayes	100	78	41	83	84
Random Forest	100	63	54	69	73
Logistic Regression	100	75	51	81	82

Method 6 Bag of words Binary embedding (word vectorization) Tokens after cleaning data = 15479

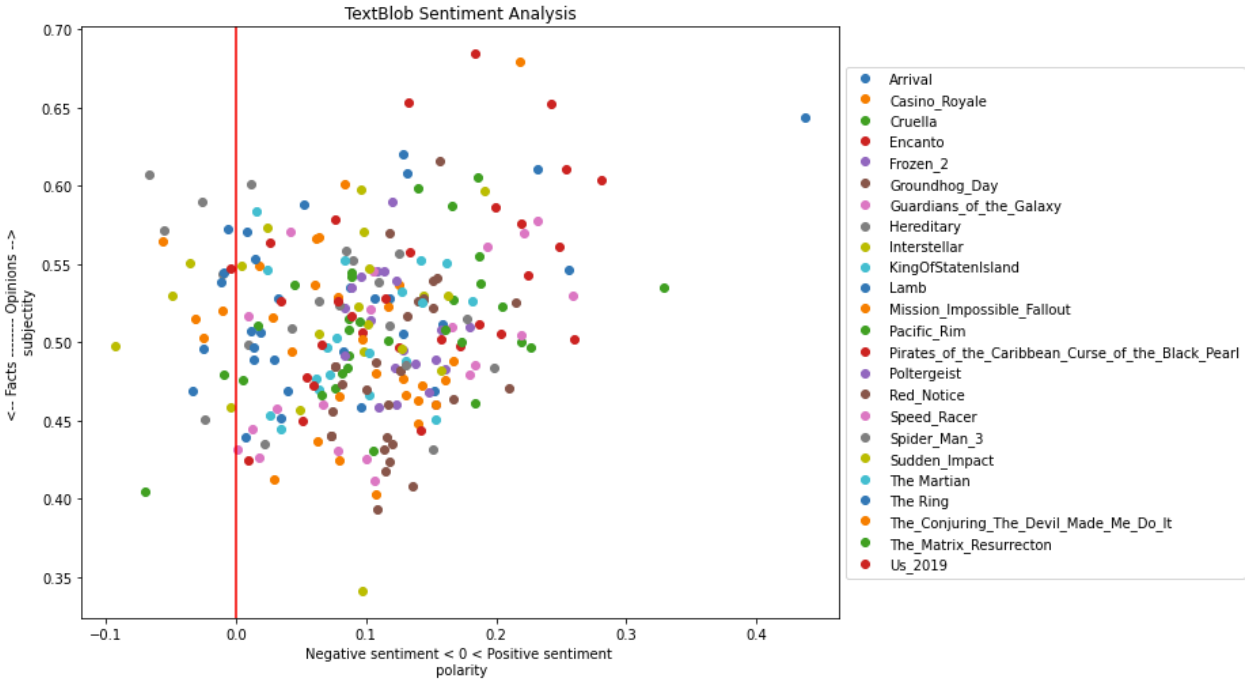
	Train Accuracy	Test Accuracy	F1	AUC	PR-AUC
SVM	99	51	64	58	72
Naïve Bayes	100	53	55	55	66
Random Forest	100	60	62	57	71
Logistic Regression	100	65	65	64	74

Method 7 Dimension Reduction with PCA on Bag of words Binary embedding (word vectorization) Tokens after cleaning data = 212

	Train Accuracy	Test Accuracy	F1	AUC	PR-AUC
SVM	91	60	65	61	73
Random Forest	100	61	60	68	71
Logistic Regression	100	59	57	62	68

In addition to the experiments above, I used the NLTK TextBlob pretrained sentiment model, which provides the polarity and subjective score of text. Polarity is between -1 and +1 where greater than 0 is positive sentiment and less than 0 is negative. According to this model, 90% of the reviews were positive, which obviously does not correlate with the human coding, in which the split between negative and positive reviews was 50/50.

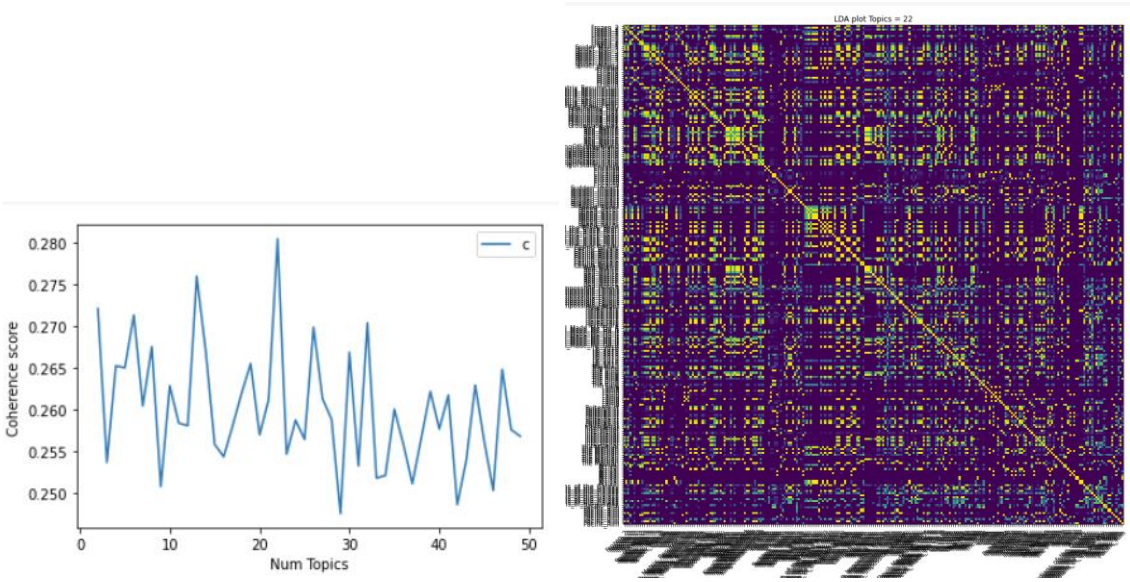
228 reviews have polarity greater than 0.



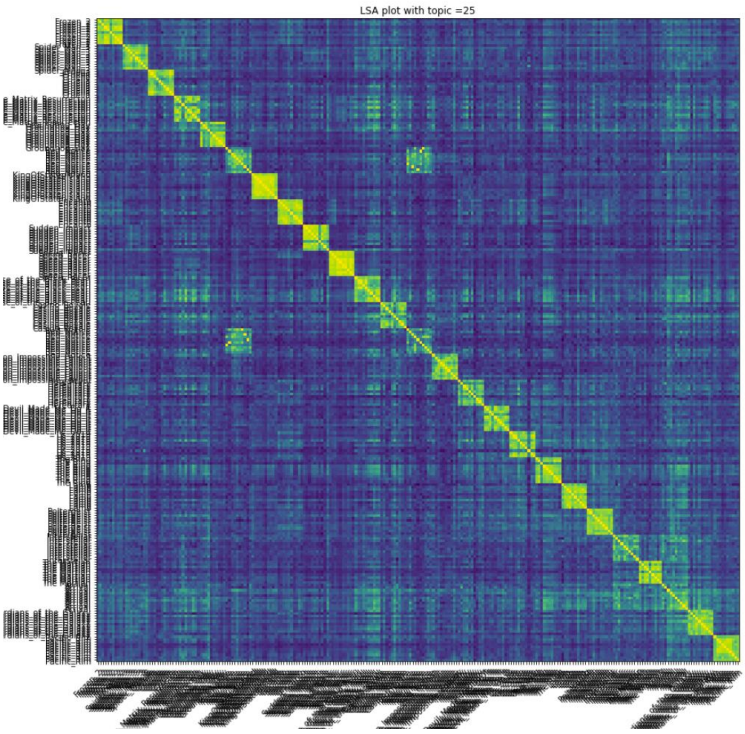
Topic Modeling:

For LDA 22, topics have the highest coherence of 0.28; however, based on the similarity matrix, as shown below, the topics for the most part don't correlate with the movie titles.

LDA



The LSA model has a coherence score that is approximately the same for all the topics (0.26) that range from 6 to 50. The similarity matrix shows that the topics correlate with the movie title.



5. Analysis and Interpretation

Key Findings:

Clustering.

As seen in the results above, clustering on TFIDF provides 25 clusters, which represent the clusters by movie names. However, there is some overlap of movie titles between a few clusters. I found K means clustering to provide the best clustering. Agglomerative hierarchical clustering also provided good results, but DBSCAN was unable to provide distinct clusters (using KNN to find epsilon). See appendix A for Agglomerative and DBSCAN plots.

Sentiment Analysis.

The important thing to note is that the movie reviews are usually based on rating scales and a low review does not necessarily mean there is negative sentiment in the review. The pretrained NLTK TextBlog model identified 90% of reviews as positive.

Models in method 3 and method 5 as shown above provided the highest test accuracy. These models used Random Forest or logistic regression classifiers for feature selection which reduced the noise in the data. However, that is a supervised learning method which would not generalize well on text data. Also, Doc2Vec embedding performed poorly in the classification model.

I also experimented by removing nouns using NTLK pos tagging package but it adversely affected the model's performance.

I believe that Method 6, 7 and 2 would generalize better. In Method 6, I used binary bag of word embedding (term document matric has 0 and 1 only) and in Method 7, I used PCA to reduce dimension on the TFIDF embedding. In Method 2, I used TFIDF metric on the clean data.

All models overfit the training data and had high variance; this is partially because we have a small dataset.

The best model without feature selection was the **logistics regression classifier** in **Method 6** that used binary bag of word vectorization because of its simplicity and highest accuracy among the models that did not perform feature selection using a supervised classifier.

The data is balanced with an equal number of negative and positive reviews and the test train split was stratified to maintain the balance; therefore, test accuracy was a good metric and was the same to the f1 score in Method 6.

The AUC score of 64% suggests that the model can predict better than a random guess (50%).

train:

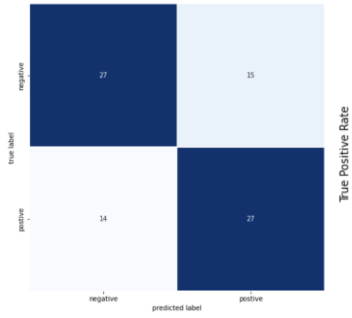
	precision	recall	f1-score	support
0	1.00	1.00	1.00	83
1	1.00	1.00	1.00	83
accuracy			1.00	166
macro avg	1.00	1.00	1.00	166
weighted avg	1.00	1.00	1.00	166

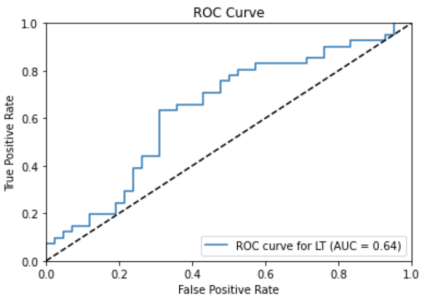
test:

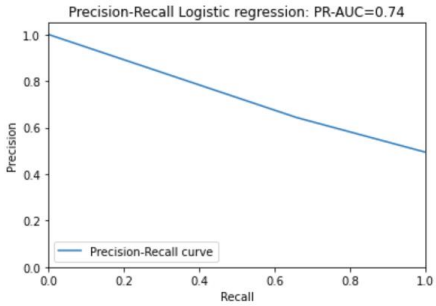
	precision	recall	f1-score	support
0	0.66	0.64	0.65	42
1	0.64	0.66	0.65	41
accuracy			0.65	83
macro avg	0.65	0.65	0.65	83
weighted avg	0.65	0.65	0.65	83

Method 6 Bag of words Binary embedding (word vectorization) Tokens after cleaning data = 15479

	Train Accuracy	Test Accuracy	F1	AUC	PR-AUC
Logistic Regression	100	65	65	64	74







Topic Modeling

LSA is analogous to principal component analysis applied to text data. LDA is probabilistic and uses Dirichlet prior.

LSA gives a correlated movie similarity matrix and LDA does not. This does not mean the LDA model is not selecting the correct topics. The reviews written for the movies often refer to other movies and topics to give analogies and there can be a lot of variation between reviews for each movie.

LSA and LDA topic modeling techniques had a low coherence score of 26-28%. This suggests that the words selected for each topic as a group are not very meaningful for human interpretation.

6. Conclusion

The clustering result showed that the word embedding can discriminate between movie titles, but not between the positive and negative reviews. This is because a movie with a low rating does not necessarily mean the reviews contain negative sentiment. Most of the movies in the corpus were successful based on Box office revenue and the low rating for these movies do not always translate to negative reviews and this can be seen the NTLK TextBlob polarity score.

The classification algorithm did not perform well unless the features were reduced by selecting only the important features by using a supervised classification model. However, this is not suitable dimension reduction method for text data and the model will not generalize on unseen data. The key improvement for the model comes from data cleaning and from the vectorizing/word embedding technique. The best model i.e., the logistic regression classifier performance improvement came from using a simple binary bag of word model (0 and 1 only).

Removing noise from the data by removing punctuation, stop words, non-alphabet words, lemmatization and converting all the text to lower case also improved the model performance but based on the results of unsupervised clustering and topic modeling, vector space cannot distinguish between the two labels of positive and negative reviews, which suggests an inherent limitation in the dataset.

Finally, the LSA and LDA models don't provide much insight into the data. The topics selected by LSA are correlated within each movie title and not among movies, where the topics selected by LDA show very little correlation within or among the movies. This shows the LDA topic selection has more variance and does not group the topics for the movie review to its labeled group of movie title or genre or positive/negative review type.

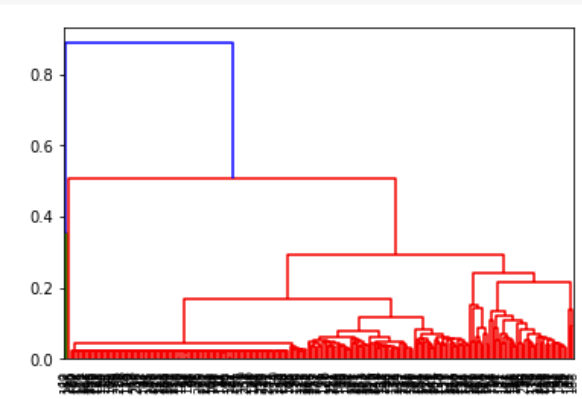
Appendix:

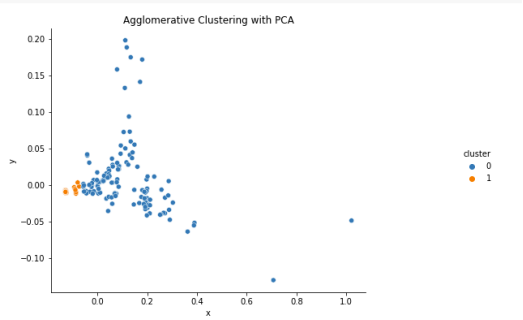
Appendix A

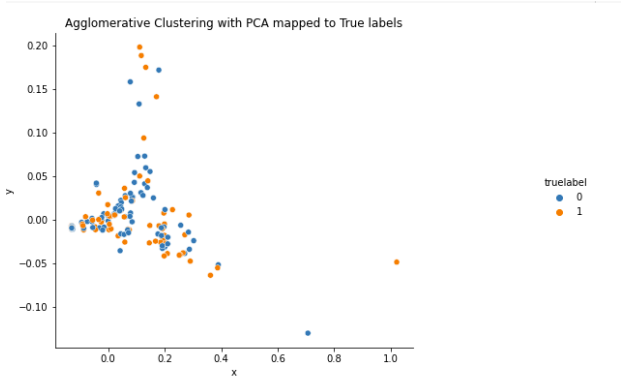
Clustering

Clustering, Classification and Topic Modeling

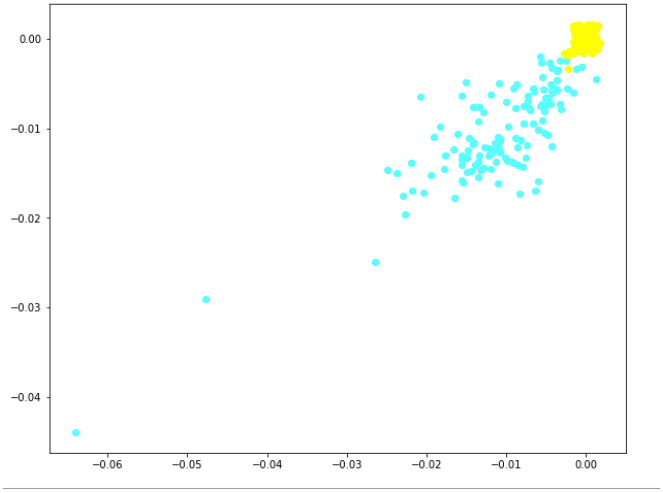
Agglomerative





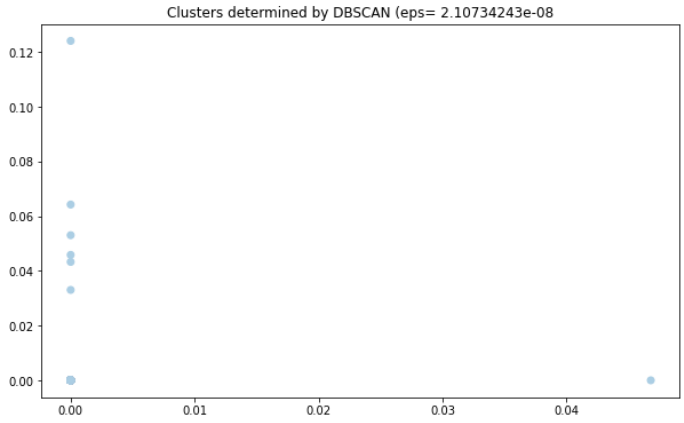



```
<matplotlib.collections.PathCollection at 0x7f03b9c30350>
```



DBSCAN

Clustering, Classification and Topic Modeling



Appendix B

Model with highest accuracy and f1 score

Logistic Regression

```
lr_model = model = LogisticRegression(solver='liblinear')
lr_model = lr_model.fit(X_train,y_train)
```

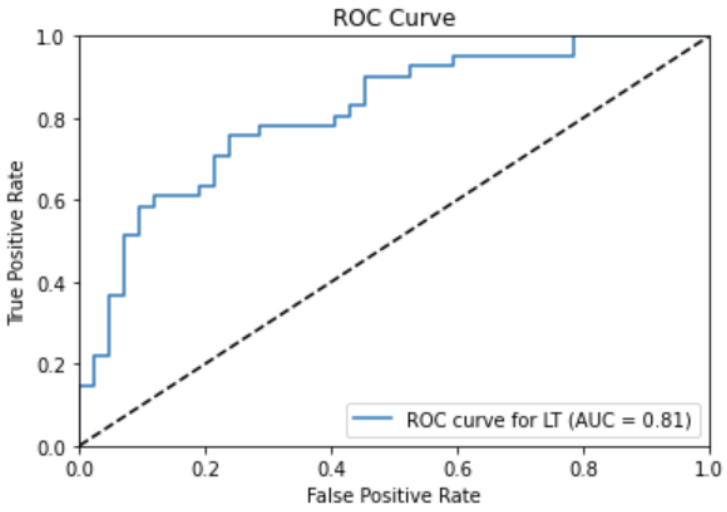
```
lr_test_acc, lr_train_acc ,f1= get_acc(lr_model)

print("train acc: ", lr_train_acc,"\n", "test acc: ",lr_test_acc ,"\n", f1)
```

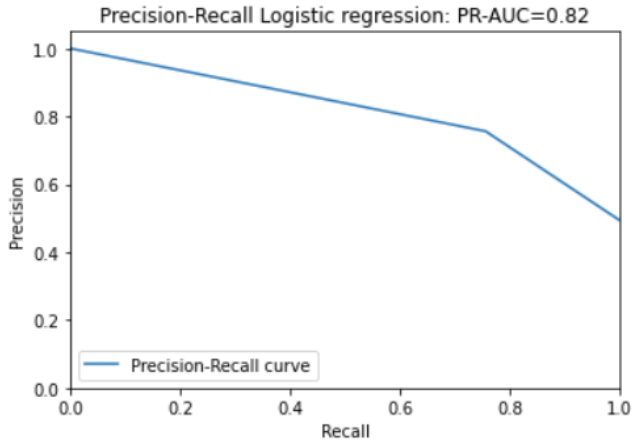
```
train acc:  1.0
 test acc:  0.7590361445783133
 0.7560975609756099
```

metrics

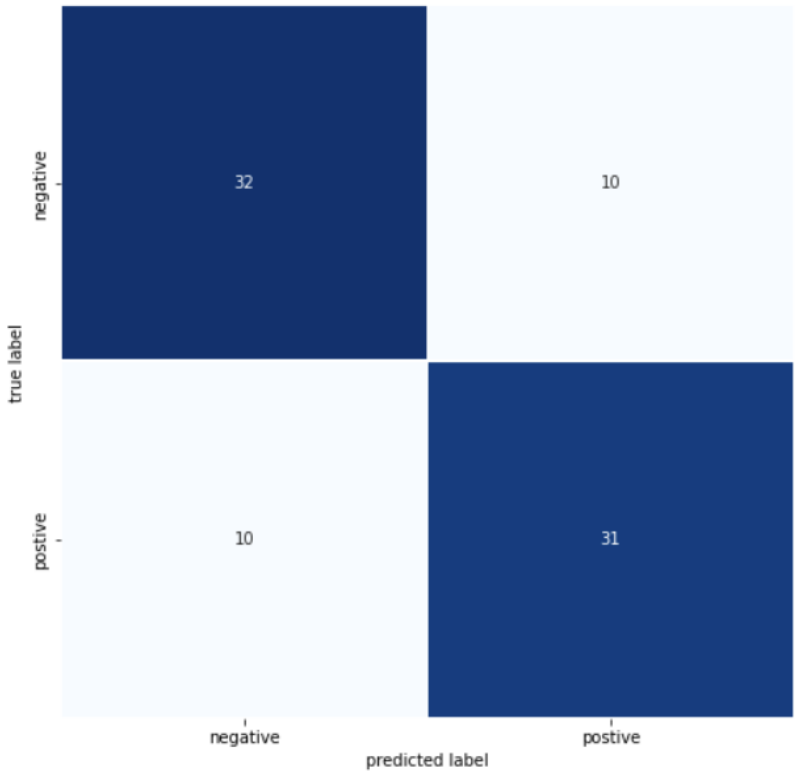
```
plt_roc_curve("LT",model = lr_model, has_proba=True)
```



```
Area Under Curve: 0.82
```




```
1 conf_mat(model=lr_model)
```



Clustering, Classification and Topic Modeling

T