Text and Sentiment Analysis of Movie Reviews

Jinal Jain jjjain@wpi.edu
Manasee Godsay mrgodsay@wpi.edu
Mihir Sawant msawant@wpi.edu
Rushikesh Naidu ranadu@wpi.edu
Umesh Nair uunair@wpi.edu

INTRODUCTION

In the modern age of technology, the Internet puts a seemingly infinite collection of information at our fingertips. Facts and opinions are produced at the "speed of thought" and consumed by Internet users in massive quantities and at extraordinary rates. What is more, as individuals we may harness this stream of information in order to make informed decisions and make inferences.

Usually before most of us choose a movie we often seek out the opinions of others who have already done so. There was a time when this process was rather limited but today, the Internet allows us to read movie reviews from critics and non-critics and that too raw and without any filters. Websites such as RottenTomatoes and IMDB offer entire collections of reviews for current and prior films. And a lot more such reviews are available in the raw from on the Internet.

In fact, these informal "reviews" are the method by which many now shape their film decisions. When headed out to the theater, many will check what their Facebook friends or Instagram followers think of a new movie, as opposed to seeking out an official critic's review. As humans, understanding whether text in a passage is positive or negative is a skill we have developed over time. This process becomes more interesting on a large-scale, where the unstructured film reviews, authored on social media, are available to movie production companies and other industry players. Understanding the latent sentiment behind the text can help these corporations understand the popularity status of their film, as well as help shape their marketing strategies, and future directions. This will be the focus of this report: using machine learning techniques to understand the sentiment of unstructured film text. We will approach this problem through four objectives.

1. PRELIMINARY SENTIMENT ANALYSIS

This study utilizes the movie reviews of the v2.0 Polarity Dataset, available at "http://www.cs.cornell.edu/people/pabo/movie-review-data". This dataset consists of 2000 .txt files, each containing a movie review. A thousand of these reviews are classified as "positive" and 1,000 as "negative."

The following three problems are presented as Exercise 2 in the "Working with text Data" tutorial of the scikit-learn documentation "http://scikit learn.org/stable/tutorial/text_analytics/working_with_text_data.html":

1. Write a text classification pipeline to classify movie reviews as either positive or negative
2. Determine a good set of parameters using grid search
3. Evaluate the performance on a test set.

In order to solve these three problems, we first modified the solution provided in the scikit-learn documentation so that runs in the iPython/Jupyter notebook. First, we added a script which downloaded the data directly from the source website, and stored it in a local directory. From there, we proceeded by following the solution.

We first randomly split the 2,000 movie reviews into 75% for training our pipeline, and 25% for testing. Using the TfidVectorizer class, we built a vectorizer-classifier pipeline that filtered out tokens that were too rare or frequent, and fit a linear support vector classifier with relatively high penalty (c=1000). Using grid search, we determined a set of tokens to consider within our documents (reviews): words (1-grams) or words and pairs of words (1-grams and 2-grams). We combined this grid search with our classification pipeline on the training data to perform grid search, finding the following (mean test scores) scores:

ngram_range	Mean Test Score	Standard Test Score
(1 , 1)	0.84	0.01
(1 , 2)	0.84	0.00

Table 1 Grid Search CV scores

The grid scores indicate is that on the *training* data, the linear SVC pipeline performs more accurately when it considers both words and pairs of words contained in our document. Using these preferred parameters determined with grid search, we use our SVC pipeline to predict the class of each review in our held-out testing set. We obtained the following classification report:

Class	Precision	Recall	f1-score	Support
Negative	0.88	0.87	0.87	254
Positive	0.86	0.88	0.87	246

Table 2 Classification Report

The high precision values on both classes (in addition to a nearly equal number of samples of each class) indicates that our model performed relatively well on this test set. We can further evaluate performance using the confusion matrix, $\begin{bmatrix} 224 & 29 \\ 37 & 210 \end{bmatrix}$ which can be plotted as follows:

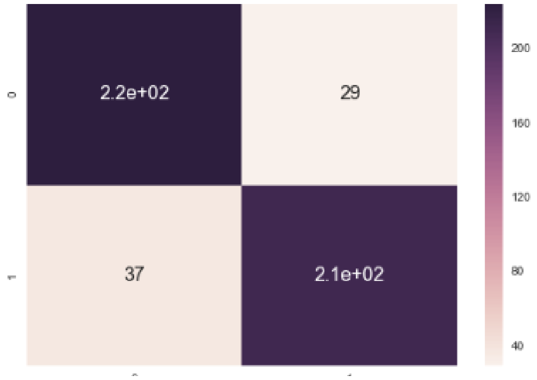

Figure 3 Confusion Matrix

The error rate (percentage of false negatives and false positives) is (37+29)/500 which is 13.2%. Combined with the classification report given above, this indicates that this model performed well on our test data set.

2. EXPLORE THE SCIKIT-LEARN TfidVectorizer CLASS

Term frequency–inverse document frequency, also known as TF-IDF, is a statistic that measures how important a word is to a document in a particular collection of documents. It is the product of two individual term statistics: term frequency and inverse document frequency.

The term frequency of a term t in a particular document d simply measures the frequency of t in d. While there are many ways to define this frequency, the simplest is the raw frequency: the number of occurrences of t in d divided by the total number of words in d.

The inverse document frequency tries to estimate the information content of a given word: common words (such as “stop words”) will appear in most documents and do not carry much information. More document-specific words are less likely to occur across a collection, and thus may be considered as carrying more "information" in that document. For a term t and collection of documents D, the standard definition of inverse document frequency is the (logarithmically scaled) number of documents in D divided by the number of documents containing term t:

$$\log \frac{|D|}{|\{ d \in D : t \in d \}|}$$

The TF-IDF is a measure of importance of a term *t* to a document *d*, among a collection of documents *D*. It is the product of the term frequency of *t* in *d* and the inverse document frequency of *t* in *D*. A word will have a high TF-IDF value if it's mentioned very frequently in that specific document, but not in a large number of documents in the collection. A word which occurs in a large number of documents will have a low IDF value, thus decreasing the TF-IDF statistic of that word in any document: this serves to filter-out common (or "stop") words.

Running the TfidVectorizer class on the training data.

The TfiDVectorizer class converts a collection of raw documents into a matrix of TF-IDF values for a set of features. The rows of this matrix are indexed by the documents in the collection, and the columns correspond to a vocabulary of terms (words or strings of words determined by the n_gram input to the class). The entries of the matrix are the TF-IDF stats for each term in each document.

Explore the min_df and max_df parameters of TfidVectorizer. What do they mean? How do they change the features you get?

When running the TfidVectorizer class on a collection of documents, we first build a vocabulary of terms contained in all documents, for which we will compute the TF-IDF stats. The parameters min_df and max_df are used in determining which terms to include in this vocabulary.

We ignore all terms that have a document frequency less than min_df: all terms in the vocabulary have a frequency of *at least* min_df in all documents in our collection. On the other hand, we also filter out all terms with a document frequency greater than max_df: all terms in the vocabulary have a frequency *at* most max_df in every document. The max_df parameter is often used to filter out stop words.

Using our collection of movie review documents, we ran the TfidVectorizer class for a range of min_df and max_df values and computed the number of features retained in the vocabulary. Plots of the number of features versus these parameter values are shown in Figures 4.1 and 4.2.

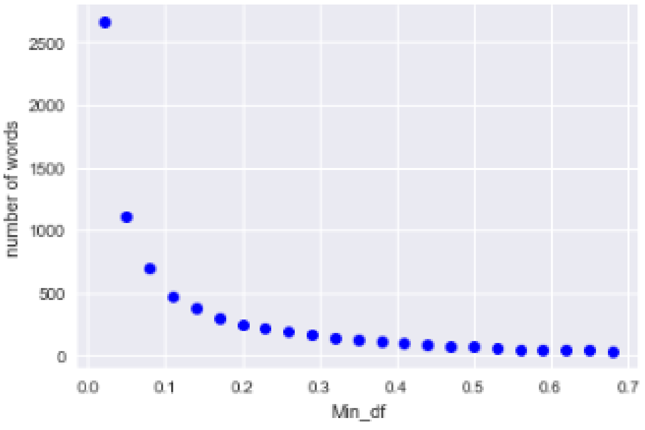

Figure 4.1 Min_df vs Number of words

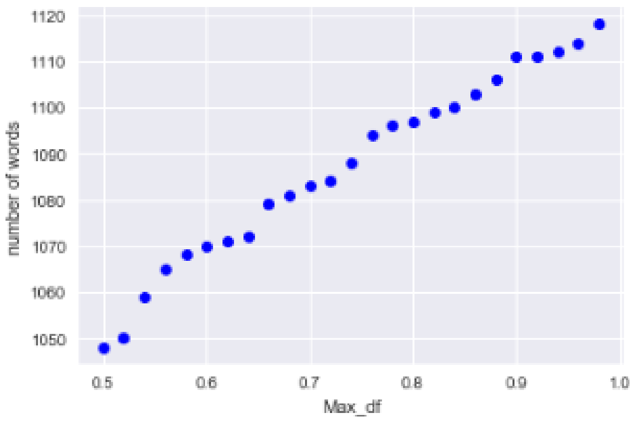

Figure 4.2 Max_df vs Number of words

These plots exhibit a clear relationship between the number of features in our vocabulary and the values of min_df and max_df. The first plot (fig 4.1) we try to change the values of min_df from 0.02 to 0.7 with increments of 0.03.We can see that the plot is exponential. When the minimum threshold for min_df is 0 all of the words are considered in the vocabulary but as we go an increasing the min_df we see a drop in the words being considered in the vocabulary. In the second plot (fig 4.2) we try to change the values of max_df from 0.5 to 0.99 with increments of 0.02. We can see that the plot is almost linear. When the maximum threshold for max_df is reasonable it does not consider the words that occur a lot of times across all the documents.

Exploring the different ngram_range parameter values of TfidVectorizer

An n-gram is a sequence of "n" consecutive words in a document. The 1-gram of a document is the collection of single word, the 2-gram are the collections of pairs of consecutive words, and so on where n-gram is a collection of n consecutive words across all the documents. The input ngram_range determines the sets of n-grams that will be included in the vocabulary of the TfidVectorizer. When we choose ngram_range = (1,1) it creates a vocabulary of all *words* in the document. Similarly when we choose ngram_range = (1 , 2), we now create a vocabulary of all consecutive pairs of words in the document. We consider ngram_range = (1,i) where the value for i changes from 1 to 6 with constant max_df = 0.95 and min_df = 0.05.

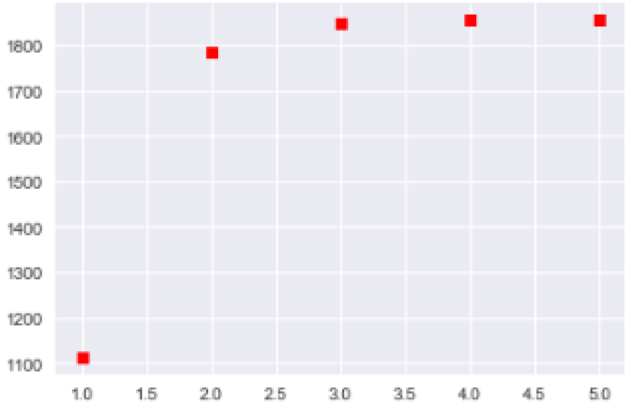

Figure 5 ngram_range vs Number of words

3. MACHINE LEARNING ALGORITHMS

We have used two machine learning algorithms, Linear Support Vector Classifier (SVC) and K-Nearest Neighbors (KNN), to predict the polarity of movie reviews. We use a 75% split for the training data set and 25% for the testing data set. First we have tried to fit the TfidfVectorizer using docs_train but also used the stop_words function to remove certain stop words. We then used the fit-transform to turn our Training and Testing documents into a pair of matrices. To ensure values of these matrices correspond to the same text tokens, we use the same TF-ID-weighted class, which is derived from the training documents (docs_train), to transform both the training and testing documents. We computed "Xtrain" (a TF-ID-weighted document-term matrix corresponding to the training documents) and "Xtest" (matrix corresponding to the testing documents).

We then examined two classifiers provided by scikit-learn.

1. Linear Support Vector Classifier. First we used our training matrix, Xtrain, to develop a set of linear support vector classifiers. We provided a default value for penalty, min_df = '6' and max_df = '0.5'. Then we set the parameters for the pipeline, where we only used different ngram_range values. Then applied grid search and printed the cross-validated scores. Then we calculated the best scores which is “0.848” corresponding to the (1,2) ngram_range value. The classification report is as follows:

Class	Precision	Recall	f1-score	Support
Negative	0.87	0.89	0.88	253
Positive	0.89	0.86	0.87	247

Table 6 Classification Report

Evaluating the performance of the model using a confusion matrix. The confusion matrix for this model, with values $\begin{bmatrix} 211 & 26 \\ 35 & 228 \end{bmatrix}$ is given by

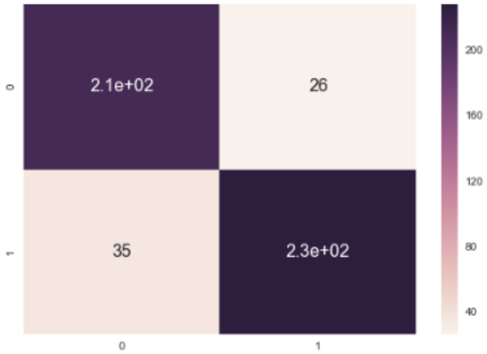

Figure 7 Confusion Matrix, Linear SVC with C = default

The error rate (percentage of false negatives and false positives) is (26+35)/500 which is 12.2%.

Then we considered the same model but with a penalty C = 0.1. The classification report is as follows:

Class	Precision	Recall	f1-score	Support
Negative	0.85	0.88	0.87	246
Positive	0.88	0.85	0.87	254

Table 8 Classification Report

We can evaluate the performance of this model using a confusion matrix. The confusion matrix for this model, with values $\begin{bmatrix} 211 & 26 \\ 53 & 210 \end{bmatrix}$ is given by

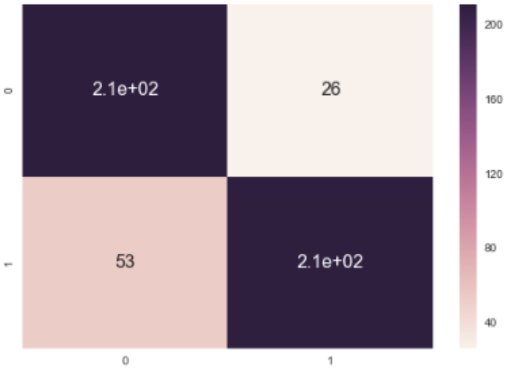

Figure 9 Confusion Matrix, Linear SVC with C = 0.1

The error rate (percentage of false negatives and false positives) is (26+53)/500 which is 15.8%.

Both the confusion matrix demonstrates a small number of test error rates in general. But the test error rate after introducing a small penalty is more than the one with the default penalty. But in general this indicates that a Linear Support Vector classifier does good job in predicting the polarity of movie reviews.

2. K-Nearest Neighbors. Next we used our TF-IDF training matrix, Xtrain, to develop a set of KNN classifiers. We used two different values for k = 5, 10 and a range of parameters:

PARAMETERS	VALUES
'vect__max_df'	(0.6, 0.7, 0.8)
'vect__min_df'	(5, 7, 9, 12)
'vect__ngram_range'	(1, 1), (1, 2)

Table 10 Parameter and Values

a. K = 5

After applying grid search we printed the cross-validated scores. Then we calculated the best score which is “0.6893” corresponding to the (1,2) ngram_range value, max_df = ‘0.8’ and min_df = ‘9’. The classification report is as follows:

Class	Precision	Recall	f1-score	Support
Negative	0.70	0.63	0.66	253
Positive	0.66	0.72	0.68	247

Table 11 Classification Report

We can evaluate the performance of this model using a confusion matrix. The confusion matrix for this model, with values $\begin{bmatrix} 160 & 93 \\ 70 & 177 \end{bmatrix}$ is given by

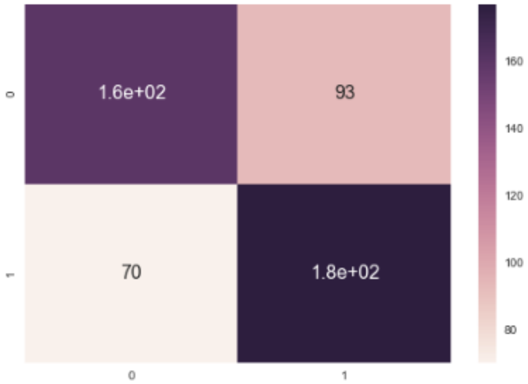

Figure 12 Confusion Matrix, KNN with k=5

The error rate (percentage of false negatives and false positives) is (70+93)/500 which is 32.6%.

b. K = 10

After applying grid search we printed the cross-validated scores. Then we calculated the best score which is “0.708” corresponding to the (1,2) ngram_range value, max_df = ‘0.7’ and min_df = ‘5’. The classification report is as follows:

Class	Precision	Recall	f1-score	Support
Negative	0.71	0.71	0.71	253
Positive	0.70	0.70	0.70	247

Table 13 Classification Report

We can evaluate the performance of this model using a confusion matrix. The confusion matrix for this model, with values $\begin{bmatrix} 180 & 73 \\ 74 & 173 \end{bmatrix}$ is given by

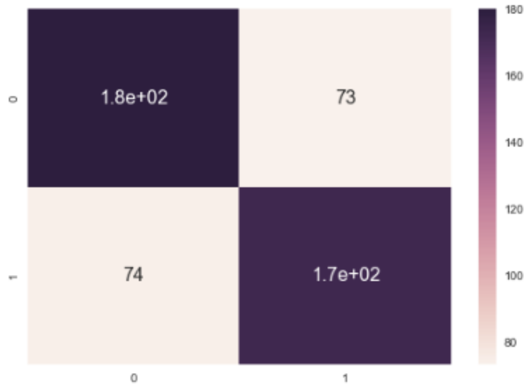

Figure 14 Confusion Matrix, KNN with k=10

The error rate (percentage of false negatives and false positives) is (74+73)/500 which is 29.4%.

Does one classifier, or one set of parameters work better?

In general the Linear SVC model works better than the KNN models. The SVC model provides an error rate of 12.2% when we consider the default penalty. This is the lowest testing error rate and performs the best on the test data set with 87.8% accuracy.

For a particular choice of parameters and classifier, look at 2 examples where the prediction was incorrect.

Review 1. (True Polarity: Positive; Predicted Polarity: Negative)

We conjecture that this review has been wrongly classified as positive because of the repetitive use of words like 'enhancers' which might be used in other reviews in a positive context. The entire text is available in the end under Appendix.

Review 2. (True Polarity: Positive. Predicted Polarity: Negative)

We conjecture that this review has been wrongly classified as negative because of the interesting aspect in this review that a lot of negative words are used to describe the story and the characters ("violent", "snobbish", "hot-blooded", "thuggish", "jarring", etc.). These might've been used by reviewers in the training set to describe a movie/character they didn't enjoy. Hence, the classifier may have classified this review as a negative one even though the review is perfectly positive and in favor of the movie. The entire text is available in the end under Appendix.

4. **FINDING THE RIGHT PLOT**

We are trying to make a two-dimensional plot here in which the positive and negative reviews are separated distinctively. Since we have to divide the reviews based on whether it is positive or negative simply using the features of the reviews, we approached the problem as an unsupervised one. Applied several clustering algorithms and then tried to reduce the data to two dimensions and then visualizing it. Similarly we also tried to reduce the data two dimensions and then applied several clustering algorithms after which we tried to visualize it.

For dimensionality reduction we used three different techniques, Principal Component Analysis, Multi-Dimensional Scaling (An MDS algorithm aims to place each object in N-dimensional space such that the between-object distances are preserved as well as possible) or Truncated Singular Value Decomposition (a factorization method that takes the top n eigenvectors of the data to find an approximate representation of the data in n dimensions and uses a randomized algorithm [the truncated part] to do so).

For clustering, I’ve used K-Means clustering, Density based Clustering (DBSCAN) and Spectral Clustering (it reduces the dimensionality of the data and then clusters it using K-Means).

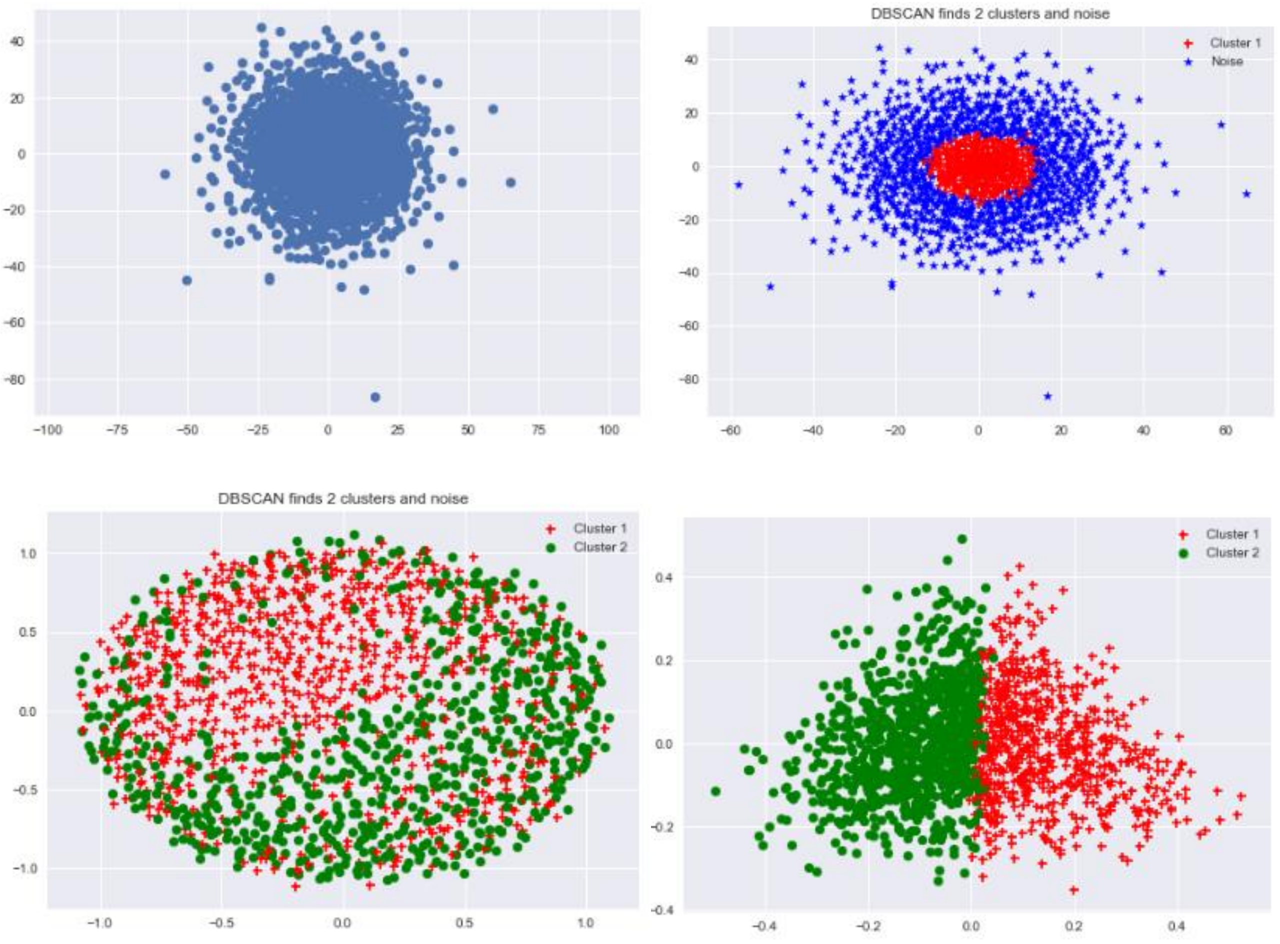

Figure 15 Stepwise plots using clustering techniques

None of the attempts gave satisfactory clustering of the data based on the sentiment of the reviews.

We then tried forming clusters using Hierarchical/Agglomerative clustering. For this, we first performed non-linear dimensionality reduction using Spectral Embedding in order to bring down the number of dimensions to 2. Spectral embedding creates an affinity (similarity) matrix using k-nearest neighbors or radial basis function (rbf) kernel, and then applies spectral decomposition to the corresponding graph Laplacian to result in a transformation whose values are given by the Eigen vectors for each data point.

Then we applied the AgglomerativeClustering() function from the ‘sklearn.cluster’ package on the above transformation, with number of clusters as 2 and linkage methods as ‘ward’, ‘average’ and ‘complete’, to specify which distance to use between sets of observations or clusters, and affinity as ‘euclidean’, ‘manhattan’, ‘l1’, ‘l2’ and ‘cosine’ which are metrics used to compute the distance. The best accuracy achieved was 59.55% using linkage method as ‘ward’ (affinity is ‘euclidean’ by default).

Finally, we plotted the clustering results with different colors and used labels from the training dataset to see where each data instance was clustered.

Note: ‘Ward’ linkage - finds the distance of the change in caused by merging the cluster. The information of a cluster is calculated as the error sum of squares(ESS) of the centroids of the cluster and its members.

*fBestDist = ESS * merged.size() - ESS1 * cluster1.size() - ESS2 * cluster2.size()*

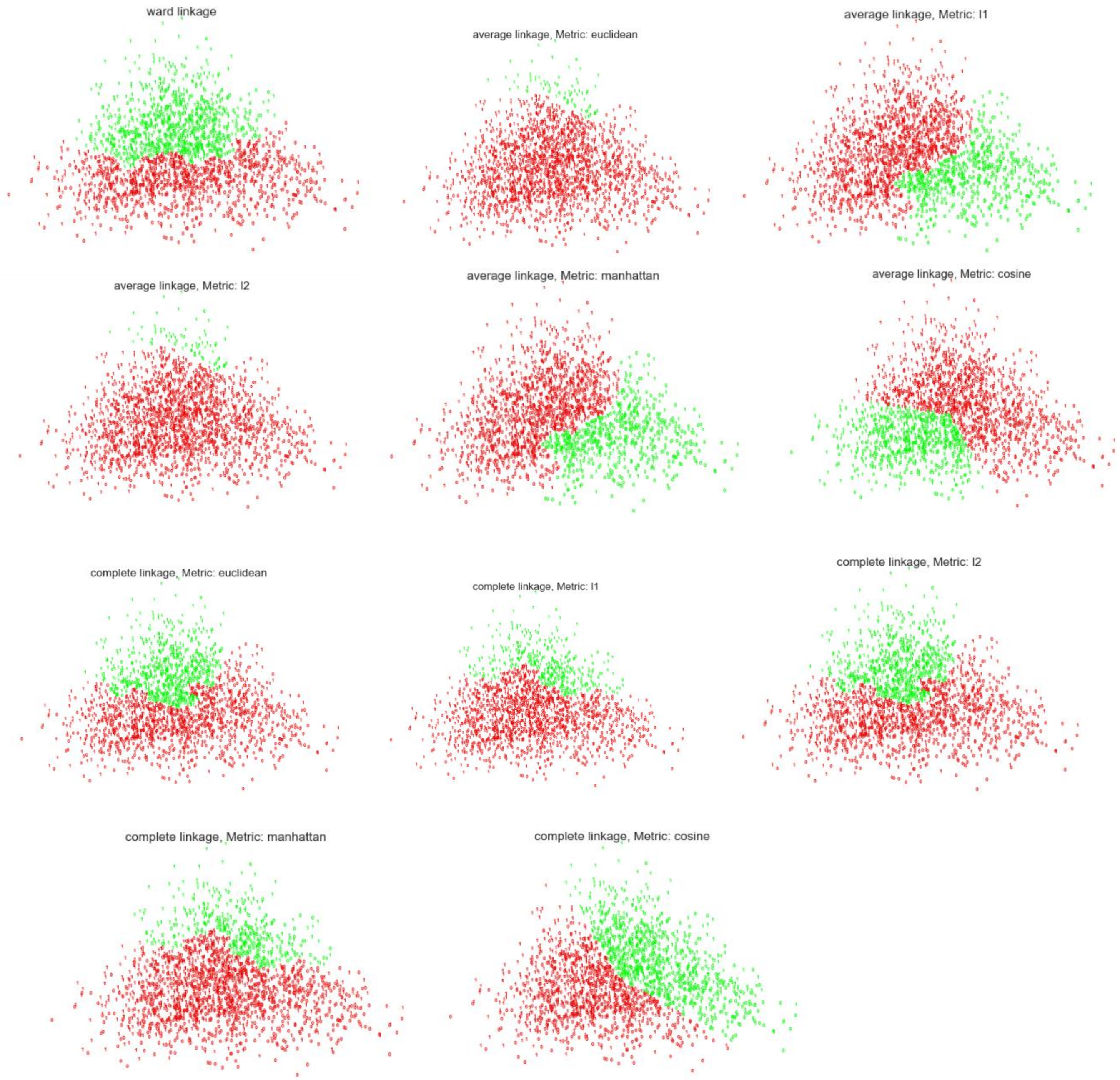

Figure 15 Stepwise plots using hierarchial clustering techniques

BUSINESS INTELLIGENCE & CONCLUSION

As mentioned in the introduction, countless informal movie reviews are authored on social media every day. Though presented as unstructured text, understanding the sentiment expressed can be extremely value to movie production companies especially companies like Netflix. There are official reviews by "critics" released for each film but the volume of movie opinions offered freely via social platforms like Facebook is massive. A machine learning method which can process and reliably extract the polarity of unstructured movie-related text could be beneficial. Specifically, it can indicate to production companies' specific areas, ages, genders, and demographics in which their film is performing well (and performing poorly). It can predict what genre of films is liked and is more popular by the target audiences and help in business decisions: What kind of movies should be made? Or maybe in building a recommender system.

In this report, we studied the polarity (positive/negative) of unstructured movie reviews, with a particular interest in prediction via machine learning algorithms. Using a dataset of 2,000 movie reviews, we first performed preliminary sentiment analysis via a text classification pipeline. Parameters were chosen via grid search, and performance was evaluated using a validation set. Next we explored the scikit-learn TfidVectorizer class. Understanding of the relevant statistics and parameters then allowed us to train two classification models: linear SVC and KNN. Parameters were chosen via cross-validation, and the models were evaluated with a validation set. The linear SVC performed quite will on the testing set, while K-NN performed poorly, particularly on the test set of negative reviews. We also took care in observing the run-times of the algorithms. In all cases, KNN required significantly longer computation time than did linear SVC and we had to run K=10 and K=5 separately as the computation was taking too much time. We compared all the different models and concluded that the Linear SVC model outperforms KNN and gives a better test accuracy.

Finally, we tried to create a 2-dimensional plot which separated the negative and positive reviews. We tried two different techniques and tried plotting it.

APPENDIX

1. POSITIVE REVIEW PREDICTED AS NEGATIVE

```
b'capsule : this super-light situation comedy from sweden tells the story of two close friends
with romantic problems . \nthe script involves formerly taboo subjects like erotic toys and
sexual enhancers but otherwise the writing is not a lot different from what is shown free on
television . \nthe characters are paper-thin and the interesting ideas purely non-existent .
\nthis is a decrement-life-by-90-minutes card . \n , 0 ( -4 to +4 ) \njalla ! jalla ! \nis
basically an exuberant tv situation comedy written instead for the wide screen . \nit tells
the story of two park custodians and the problems they are finding on the path to true love .
\nthe film is set in sweden where roro and mans ( fares fares and torkel petersson ) are
custodians at a public park . roro is from a tightly knit lebanese family who control him very
closely , mans is a swede from a much more liberal background . \nthey spend most of the day
in the bushes at their park , cleaning up after dogs . \nroro and mans each have girlfriends ,
but each has a problem . \nroro ( nicknamed " jalla " ) is having family problems . \nit seems
that his family wants to arrange a marriage between him and a nice lebanese woman , yasmin (
laleh pourkarim ) , but he is already in love with lisa ( tuva novotny ) . \nyasmin likes roro
, but does not want to get married either . \nmans on the other hand has been having a problem
of sexual impotence . \nthe two friends worry about their problems and discuss the problems
with each other . \nmans thinks the answer to his problem is to purchase sexual enhancers .
\nthe one catch is that he is too shy to go in and buy them . \nroro and yasmin decide to give
themselves some time by telling the families that they want to marry each other , but then
plan to break up before the wedding . \nnot too surprisingly neither finds that his idea works
out the way he quite expected . \nthe plot turns in several places are contrived . \none knows
fairly quickly that if things are going to work our happily for everybody certain plot
contrivances have to happen . \nlebanese-born josef fares who wrote and directed is perhaps a
better director than he is a writer . \nwhen things start to get slow , he just adds throws in
another story . \nfor example halfway into the film mans innocently antagonizes some local
toughs and a long chase is added to the film . \ncharacterization is a little better with roro
than it is with mans who does not seem to have a whole lot more personality beyond fear for
losing a biological function . \nwe do see some of roro\'s family life and his concerns .
\nthat may be because roro\'s background is a lot like that of the director . \nwhile the
story was entertaining , i did not feel that i got anything worthwhile from the film . \nit
was just a way to pass about an hour and a half in my life . \none does not have to go to the
movies to see entertainment like this . \ni rate it a 4 on the 0 to 10 scale and a 0 on the -4
to +4 scale . \n'
```

2. NEGATIVE REVIEW PREDICTED AS POSITIVE

```
b' " through a spyglass , i could see everything . " \nking louis xvi was beheaded on january
21 , 1793 , but instead of visualizing this act of regicide , legendary auteur eric rohmer\'s
the lady and the duke observes from afar . \nconsider it a view to a kill made abstract . \na
proper british ( yes , british ) gentlewoman , grace elliott ( lucy russell ) , and her loyal
maidservant gaze from a lofty terrace in meudon at the glistening city of paris , where
raucous crowds seem tinier than ants . \nthe maid narrates what little she sees of the
execution through her telescope ( often muttering , " i don\'t know , " ) as the sound of
cheering patriots and revolutionaries echoes through the air . \nwhat we don\'t see might not
be able to hurt us . \njust close your eyes and think of england . \nduring times of
revolution , the aristocracy may feel a false sense of calm in their parlor halls , discussing
tumultuous events over glasses of sherry until the walls cave in on them . \nadapted from
```

elliott\'s memoirs , journal of my life during the french revolution , rohmer\'s latest artistic tour-de-force may seem far removed from his domestic comedies (tales of the four seasons , etc .) , a period film set during the most violent changes in french history . \nresisting the temptation for grand-scale theatrics , much of the lady and the duke is about quiet , decisive moments between members of the cultural elite as they determine how to proceed as the world implodes . \ngrace elliott makes for an unlikely protagonist : a headstrong , snobbish blueblood , one unprepared for the machinations of history that sweep her along . \na foreigner who accepts the french king as her own , grace\'s life seems defined by fancy attire and lively political debate with her former lover , the king\'s hot-blooded cousin , prince philipe , duke of orleans (jean-claude dreyfus) . \nthe times are changing , though , and the gears inch ever closer toward violence . \nduring the september massacres of 1792 , she is encountered by a procession of rioters brandishing the head of the duke\'s sister-in-law on a stake . \nrohmer makes a harsh transition from tranquil , old fashioned , almost stagy parlor scenes to the swell of an angry mob . \nin doing so , he achieves what braveheart and the patriot could not : the face of death . \nwhen grace sees her friend\'s disembodied head on a pole , rohmer\'s attention drifts from the societal change to one woman\'s reaction shot , laden with hot tears . \ngrace finds herself taking in a fugitive from justice , sheltering him from the mob . \nthrough her relationship with the duke , she seeks a passport for this one activist\'s escape . \ngrace doesn\'t even understand her own actions (and the duke reacts in stunned disbelief at how she places herself in such danger) . \nshe endures persecution from robespierre and his gang of thuggish equalizers , ceaseless police monitoring , house searches , even a brief imprisonment for harmless international correspondence . \nmaintaining her stiff upper lip and pampered life (her imperious attitude to the servants never changes) , she becomes a heroine through circumstance . \nthe events themselves are intrusions upon her person , her home , and therefore her values . \naristocracy proves a glass house , one that can barely withstand the upheaval of stones . \nthe duke is called to vote on the king\'s punishment , and despite his hours of deliberation with friends and advisors , talk means nothing in the face of bloody action (or futile inaction) . \nthe episodic structure creates a wobbly , jarring detachment from the events of the french revolution , which serves as metaphor but also disconnects potential audience identification . \nlazy viewers (and critics) may also complain that knowledge of french history is required for enjoyment of the lady and the duke . \nthat\'s foolery , but brings up the valid criticism that rohmer\'s characters occasionally become didactic . \nrohmer\'s imperfect but assured push toward the future remains staunch and notable for casting a cautious eye upon the past while taking bold steps forward into an uncertain future . \nwhat may arouse interest in the lady and the duke outside of foreign film enthusiasts with literary and historical passions is rohmer\'s use of cutting edge digital technology as a means of exploring the theme of artifice as safety net or coping mechanism . \nthe actors were filmed against a bluescreen , then placed against painted backdrops recreating the vastness of 18th century paris . \nthis recreation calls attention to itself in every shot , a technicolor dream of fanciful buildings and wide-open streets . \nit looks as phony as titanic , but unlike james cameron\'s debacle , the lady and the duke plays with the notion of false security in those walls of stone . \nwhy ? \nthey aren\'t real . \nthe very foundation rohmer\'s characters stand upon is false , and in their groundlessness they must discover themselves , in all their insubstantial glory . \nscreened at the 2001 new york film festival (feature coming soon) . \n'