Luke Kvamme, Phillip Garcia, Marc D’Avanzo

CSC-272

12-2-2022

Sentiment Analysis of News Sites’ Home Pages

Introduction

In today's climate, many people feel that the news is overwhelmingly negative. Every day there is a new crisis affecting the world, like a new conflict happening between Ukraine and Russia or a local tragedy such as a fatal car crash. In this vein of thinking, this project’s central question was to examine if there was a correlation between the day of the week, news source, or the political leaning of the source on the sentiment of the news given, either positive or negative. To accomplish this, we downloaded the HTML of various home pages from different news sources from January 2021 to November 2022 and three sentiment Lexica to help us analyze the HTML files.

We found inconclusive results from the analysis of our data. It was determined that the day of the week contained no correlation with its polarity, as there is a fairly equal distribution of positive and negative data instances per day. Each day maintained the same ratio of positive to negative instances. However, there were notable differences in the polarity of positive and negative stories between “Left” and “Right” leaning political sources, as well as a large variance in the polarity ratio of individual sources.

Road Map

Our report is in 7 sections, laid out in this manner:

- Section 1 discusses the data set in detail – Page 2
- Section 2 discusses our data preparation in detail – Page 7
- Section 3 discusses our data analysis in detail – Page 10
- Section 4 discusses our results – Page 13
- Section 5 discusses our overall results – Page 33
- Section 6 is our conclusion – Page 34
- Section 7 is our appendix – Page 36

1. Dataset Description

We have 5 different data sets for this project. One is a testing set and the other 4 are training data sets. There is also a 6th data set used to help with visualization. All data sets have the same basic format: each instance is a specific HTML file of a news source's home page on their website. For that instance we would track the year it was from, the day of the week, the source, its political leaning, and its sentiment. The below table Further explains the data sets.

Attribute	Possible Values	Description	Data Type
Year	2021 2022	The year in which the HTML file was published	Nominal
Day of Week	Monday Tuesday Wednesday Thursday Friday Saturday Sunday	The day of the week the file was published	Nominal
Company Name	ABC Breitbart Buzzfeed CBS Daily Kos Daily wire Fox HuffPo MSNBC	The names of the sources we captured HTML files from	Nominal

	National Review Slate The blaze Vox Wall Street Journal Washington post		
PolLeaning	Left Right	The general leaning of the news source. The political leaning of a news source can vary by subject and author, so this designation is generalized. The following is how each news source is classified: ABC – > Left Breitbart – > Right Buzzfeed – > Left CBS – > Left Daily Kos – > Left Daily wire – > Right Fox – > Right HuffPo – > Left	Nominal

		MSNBC – > Right National Review – > Right Slate – > Left The blaze – > Right Vox – > Left Wall Street Journal – > Left Washington post – > Left	
Sentiment	Negative Positive	The class attribute of the data set is determined by counting the number of positive and negative words in the file. The category with a higher count denotes its classification.	Nominal

The reason there are four different training data sets is that each represents one of three respective sentiment Lexica we found, and the fourth is a combined Lexicon of all three. We gave each Lexicon a simple numerical indicator of 1, 2, or 3 and for the combined we called it Lexicon “C” or “Combined”. We wanted to assess the different Lexica to see if we could determine if one was more accurate than the others.

Lexicon 1 is called the MPQA Subjectivity Lexicon from the University of Pittsburgh Computer Science Department (http://mpqa.cs.pitt.edu/#subj_Lexicon). It contains 8,222 different words. Lexicon 2 has no specific name, but was created by Dr. Bing Liu and Dr. Minqing Hu of the University of Illinois Chicago (https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#datasets). Their Lexicon contains 6,800 words. Lexicon 3 was found from a website called SenticNet, which describes itself as helping machines learn, leverage, and love (http://sentic.net/downloads/). Their Lexicon contains 150,000 total entries making it magnitudes larger than the other two entries. SenticNet's Lexicon contained not only single words like the other Lexica, but also emojis and up to 4-word n-grams. The emojis and ngrams needed to be removed to bring Lexicon 3 to equal footing with the other two Lexica. After removing the phrases and emojis, Lexicon 3 contains 39,500 words. When all three Lexica were combined together for the fourth Lexicon, all of their unique words total 43,732.

The test data set contains 120 files taken out of the training data set. Instead of having these HTML files be classified using the four Lexica, we manually divided the files between the three of us and parsed them by hand. We kept track of the number of positive and negative words to determine the sentiment of the file once we finished reading it. While classifying the HTML files we recreate the automated algorithm to the best of our abilities. For each of the test files, we went word by word and ignored the context of what the sentence was conveying. For example, if a headline stated "kind and innocent found murdered", there would be 2 positive words, "kind" and "innocent", and one negative word, "murdered". This tedious process took a very long time because some of the sources have up to 10,000 words in their HTML file.

The final data set we created was a conglomeration of the results of Lexica 1 through 3 and C. This file contained all of the counts of positive and negative words for all of the Lexica as well as the sentiment differences (count of positive - count of negative) used to determine the class attribute. This file was used to compare differences between how the Lexica classified the HTML files. As a class attribute for the file we included an average sentiment, where whatever the majority sentiment from Lexica 1 through 3 was, it became the class attribute/sentiment.

Lexicon	Number of Words	Positive Classifications	Negative Classifications	Political Leaning Ratio (L:R)
Lexicon 1	8222	6653	2246	5607:3292
Lexicon 2	6800	1756	7143	5607:3292
Lexicon 3	39500	8670	229	5607:3292
Lexicon Combined	43732	8618	281	5607:3292
Test Data	Human Intuition	10	110	76:44

2. Data Preparation

In order to answer if specific news sources, their political leanings, or different days of the week affected the polarity of news, we needed files containing each news sources’ website headlines and a way to parse each file and discern the polarity of each word.

One thing we were concerned about was getting enough data. Because the project was started part way through the semester, we realized in order to obtain a sufficiently large data set we could not just track news sources from the project commencement to the project conclusion. This led us to want to gather data from the past. In order to gather news from the past, we created a web scraper to systematically crawl through the internet archives of Archive.org and download all of the articles on the home page of each news source for every day from January 1, 2021 to November 4, 2022. To create the web scraper, we had to use Eclipse to create a Maven project so that we could download all of the dependencies necessary for making a web connection and then parsing through all of the HTML, CSS, and JavaScript to grab the data we wanted so that we could then save the website to an output file. For the web connection we used the “JSOUP” package and for the disabling of CSS and JavaScript so we could parse just the words on the website we used ”htmlunit”. Throughout the web scraping process we encountered many problems that culminated in us shifting away from grabbing just the articles off the home page to instead downloading the entire home page..

Archive.org has a strict policy on web scraping, which took us a long time to get around. It uses JavaScript to detect if the user is actually a human or a bot, in which case it will refuse connection to the request. This resulted in a long process of debugging our code in order to avoid getting our connection refused. At a high level, we had to create functionality to change the user agent every once in a while so as to not flag its bot detection system. We also had to use an insecure SSL connection, otherwise disabling CSS and JavaScript would not function properly. Finally, we made the program sleep for 5 seconds after completing our request.

Archive.org gave us further issues due to having a processing period of around 10-30 seconds per request in order to run its anti-bot check and fetch the past web page from a massive database. This 10-30 second span of time coupled with the 5 seconds of sleeping per request and the multitude of errors thrown throughout reduced the amount of data we originally desired from the past 5 years to the past 2 years. On one of the mental health days we were given, we even went into the Mac lab at 8am and tried using 15 computers to each run a different source at the same time. This did not work however, and instead resulted in us being blocked by Archive.org for a couple of days.When we were able to access a home page, there was a non significant chance that the file would be unreadable by our web scraper due it having a horrible HTML setup, giving us useless, bad files. The combination of all these issues reduced our time frame of data from January 1, 2021 to November 4th 2022, giving us 8,899 total HTML files of around 100 MB in size.

After we had collected our HTML files we needed to create another program that would scan through the HTML files to calculate the sentiment of the file. This was done with a Python program, which was used to create the four training data sets. The Python file would read through every collected HTML file and then output the data into a new file. The Python program added the file's name, total count of positive and negative words, and the difference between the two counts in addition to the needed attributes. These unneeded attributes were removed manually afterwards to remove any highly correlated values. The program was also used to reformat the different Lexica into CSV formatted files with two attributes: Word and Sentiment.

The program was essentially one large nested for-loop. The program would search through every HTML. When reading an HTML file, the program would go line by line reading every word. If the program saw a word for the first time it would add it to a dictionary, its key being the new word and the value being 1, if the word was already present then the key with the same word would have its value incremented by 1.

After every line in the file was read, the dictionary would be iterated through searching for matching words in one of the four Lexica. After the frequency list dictionary was iterated through, the results would be added to a Pandas data frame. The Python program went through several iterations to make it as efficient as possible. In its first version, the program made no attempt to improve efficiency. This was problematic as the amount of iteration through every file and every line caused significant slow downs. In the final version of the program, two dictionaries were used for O(1) searching and accessing. There was one dictionary for the frequency of words found in HTML files and one dictionary for the words in a Lexicon. For this dictionary the key would be a word and its value would be its polarity. After these optimizations were applied to the algorithm, the program went from classifying 500 files in 50 minutes to classifying all 8,899 files in less than 2 minutes.

3. Data Analysis

The focus of this project was on the classification of HTML files into “Positive” or “Negative”. This restricted us to only using classification algorithms.

3.1 ZeroR

This is the simplest classification algorithm. Given a supplied training data set, ZeroR will calculate the majority value of the class attribute. It will then predict this majority class attribute for any new test cases. ZeroR ignores all other attributes but the class attribute, and because of this is often used as a baseline for a data set.

For our project, ZeroR will be used to determine which polarity either positive or negative is predicted more in Lexica 1 through 3, C, and the test file.

3.2 OneR

Like ZeroR, 1R is a simple classification algorithm often used as a baseline. Unlike ZeroR, 1R will analyze other attributes of a data set to determine which is the single most predictive of the class attribute. It will then output a set of rules for the most predictive attribute and the class attribute. There will be a rule for each value of the most predictive attribute that best predicts the class attribute. If there are more class attribute values than values the most productive attribute has, then some of the class attribute’s values will not have a rule associated with them.

For our project 1R will be used as a baseline and to see if either year, day of week, company name, or political leaning are very predictive of sentiment.

3.3 J48 Decision Tree

For our project we also ran the J48 algorithm in Weka to analyze our data and what attributes it chose. The J48 algorithm results in a tree in which each node is an attribute and the branches will be values of the attribute it is connected to. The reason for this is that when an attribute is chosen by the algorithm, a split occurs where the attribute’s criteria determines the path to be taken in the tree. The final level of the tree is the resulting class attribute for an instance. Sometimes not all attributes will be present in a J48 tree and this is due to certain attributes being unpredictive. The algorithm works by choosing attributes in which the best possible split occurs and tries to handle as many instances in a split. The algorithm uses accuracy and information gain to choose the best possible splits. The result is a tree that effectively classifies new instances in the data set. The J48 algorithm was run with and without a cost sensitive classifier for our purposes. The reason we performed this algorithm was to see which attributes the algorithm would select and the certain splits that occurred at each node.

3.4 Support Vector Machine

The support vector machine is an algorithm used to minimize the error when classifying test data. The algorithm works by finding the best line on a graph that splits instances by their class value. The hope of this algorithm is to minimize error and incorrect classifications with the test data set. We used this algorithm to maximize our accuracy and to give insight onto misclassified instances and analyze those instances further. The algorithm was run with and without a cost sensitive classifier. In order to use this algorithm, we had to use the package manager to install the LibSVM package.

3.5 Random Forest

Random Forest is an algorithm that relies on decision trees and ensemble learning to create a model. In simple terms, the random forest works by creating a decision tree of decision trees that are not correlated. The “random” portion of “random” forest comes from the fact that every decision tree is created with a random subset of the data and a random selection of features. The random subset of data is called the bootstrap sample. A specific ensemble method called bagging is used to randomly choose features which ensure low correlation between individual decision trees within the forest. This algorithm was run with and without a cost sensitive classifier.

4. Results

4.1 ZeroR

Lexicon	Accuracy	Rules
Lexicon 1	74.76%	Predicts Positive
Lexicon 2	80.27%	Predicts Negative
Lexicon 3	97.42%	Predicts Positive
Lexicon Combined	96.84%	Predicts Positive

ZeroR showed us that the majority of our Lexica predict a positive outcome. Lexicon 1 and 3 both overwhelmingly classify HTML files as positive, while surprisingly Lexicon 2 classified files as overwhelmingly negative.

4.2 OneR

Lexicon	**Rules**	**Accuracy**
Lexicon 1	ABC – > Positive Breitbart – > Negative Buzzfeed – > Positive CBS – > Positive Daily Kos – > Positive Daily Wire – >Negative Fox – >Positive Huffington Post – > Positive MSNBC – >Positive National review – > Positive Slate – >Positive The Blaze – >Negative Vox – >Positive Wall Street Journal – > Positive Washington Post –>Positive	27.5%

Lexicon 2	ABC – > Negative Breitbart – > Negative Buzzfeed – > Positive CBS – > Negative Daily Kos – > Negative Daily Wire – > Negative Fox – > Negative Huffington Post – > Negative MSNBC – > Negative National review – > Negative Slate – > Negative The Blaze – > Negative Vox – > Positive Wall Street Journal – > Positive Washington Post – > Negative	83.33%
Lexicon 3	2021 – > Positive 2022 – > Positive	8.33%
Lexicon Combined	2021 – > Positive	8.33%

	2022 – > Positive	

Depending on the Lexicon, 1R predicted one of two variables, either company name or year. For Lexicon 1 1R output ruled that company name was most predictive, with most sources predicting a positive classification. The only sources that were predicted to be negative were Breitbart, Daily Wire, and the Blaze. For Lexicon 2, 1R predicted again that source of the HTML file is the most predictive attribute. It was opposite of Lexicon 1, as most companies predicted a negative sentiment, except for Buzzfeed, Vox, Wall Street Journal. These two rules show that Lexicon 1 and Lexicon 2 are more balanced in how they classified the HTML files as there was a mix of both sentiments.

On the other hand for Lexicon 3 and Lexicon combined, 1R predicted that that the year of the article was most predictive. The low accuracy is caused by the large imbalance of positive files that the Lexica classified with the large amount of negative HTML files that we manually classified. Lexicon 3 seems to dominate over the other Lexica for words found due to its increased word count. This leads to the results of Lexicon 3 being over represented in the combined Lexicon.

4.3 J48

4.3.1 Lexicon 1

Base Accuracy: 22.69%

a	**b**	**← Classified As**
9	0	a = Positive
92	18	b = negative

Cost Sensitive Evaluator Accuracy: 78.99%

a	**b**	**← Classified As**
7	2	a = positive
23	87	b = negative

Lexicon 1’s base accuracy of 22.69% is alright on its own, but after performing a cost sensitive analysis of J48 with weights of 20 on guessing false positives and 1 on false negatives we were able to get the accuracy to increase to 78.99%.

4.3.2 Lexicon 2

Base Accuracy: 84.87%

a	**b**	**← Classified As**
7	2	a = Positive
16	94	b = negative

Cost Sensitive Evaluator Accuracy: 98.32%

a	b	← Classified As
7	2	a = positive
0	110	b = negative

Lexicon 2 had an amazing base accuracy of 84.87% with J48, and increased even further when it was run again with weights of 20 on false positives and 1 on false negatives using the cost sensitive filter on J48.

4.3.3 Lexicon 3

Base Accuracy: 7.56%

a	b	← Classified As
9	0	a = Positive
110	0	b = negative

Cost Sensitive Evaluator Accuracy: 11.76%

a	b	← Classified As
9	0	a = positive
105	5	b = negative

The base accuracy for Lexicon 3 was only 7.56% so we also applied weights of 20 on the false positives and 1 on the false negatives on it with a cost sensitive J48 and were able to increase the accuracy to 11.76%. Because Lexicon 3 leans so heavily towards the positive side, regardless of what weights we applied against false positives the maximum accuracy we could achieve was 11.76%.

4.3.4 Lexicon C

Base Accuracy: 7.56%

a	**b**	**← Classified As**
9	0	a = Positive
110	0	b = negative

Cost Sensitive Evaluator Accuracy: 11.76%

a	**b**	**← Classified As**
9	0	a = positive
105	5	b = negative

Because Lexicon 1 and 3 are so similar in their functionality, the combined Lexicon results in the same base accuracy of 7.56%. The highest accuracy we could attain with the cost sensitive filter was 11.76% by using a weight of 20 on the false positives.

4.4 Support Vector Machine

4.4.1 Lexicon 1

Base Accuracy: 27.73%

a	b	← Classified As
9	0	a = Positive
86	24	b = negative

Cost Sensitive Evaluator Accuracy: 78.99%

a	b	← Classified As
7	2	a = positive
23	87	b = negative

The base accuracy of SVM was 27.73%. After applying weights of 10 for the false positives and 1 for the false negatives by using the cost sensitive filter, we were able to increase the accuracy to 78.99%.

4.4.2 Lexicon 2

Base Accuracy: 84.87%

a	**b**	**← Classified As**
7	2	a = Positive
16	94	b = negative

Cost Sensitive Evaluator Accuracy: 98.34%

a	**b**	**← Classified As**
7	2	a = positive
0	110	b = negative

The base accuracy of Lexicon 2 was 84.87% and, as opposed to Lexicon 1 and 3 which overwhelmingly predict positive, we were able to increase the accuracy to 98.34% by using weights of 20 for false positives and 1 for false negatives. This combination allowed us to bring over every instance of negatives that were being predicted positive, but did not improve the false negatives.

4.4.3 Lexicon 3

Base Accuracy: 7.56%

a	**b**	**← Classified As**
9	0	a = Positive
110	0	b = negative

Cost Sensitive Evaluator Accuracy: 11.76%

a	**b**	**← Classified As**
9	0	a = positive
105	5	b = negative

Cost Sensitive Evaluator Accuracy: 92.44%

a	**b**	**← Classified As**
0	9	a = positive
0	110	b = negative

Lexicon 3 did not perform as well as Lexicon 1 or 2, with a base accuracy of 7.56%. By using a weight of 10 on the false positives and 0 on the false negatives, we were able to increase the accuracy to 11.76%. We flew too close to the sun however, and added more weight to the false positives and counterweights to the false negatives in an effort to increase the predictability of the negative values. This seemed like an amazing increase at first glance, 11.76% to 92.44%, however it now predicts all 9 positive values incorrectly. The only reason why the new model’s accuracy is 92.44% is because of the overwhelming amount of negatives in proportion to the positives. Therefore, the 11.76% model is arguably more accurate and preferable because it predicts all of the positives correctly and is able to predict some of the negatives properly as well.

4.4.4 Lexicon C

Base Accuracy: 7.56%

a	**b**	**← Classified As**
9	0	a = Positive
110	0	b = negative

Cost Sensitive Evaluator Accuracy: 11.76%

a	**b**	**← Classified As**
9	0	a = positive
105	5	b = negative

The combined Lexicon gets overly influenced by Lexicon 3 and mirrors its exact behavior, with a 7.56% baseline accuracy and an 11.76% accuracy when a weight of 10 is added to the false positives.

4.5 Random Forest

4.5.1 Lexicon 1

Base Accuracy: 23.33%

a	b	← Classified As
10	0	a = Positive
92	18	b= Negative

Cost Sensitive Evaluator Accuracy: 91.67%

a	b	← Classified As
0	10	a = Positive
0	110	b= Negative

The base accuracy of the random forest on Lexicon 1 is only 23.33% but is arguably better then the 91.67% of the cost sensitive evaluator random forest. While the base accuracy was extremely low it was able to correctly predict all of the manually evaluated positive HTML files while at the same time predict a few of the negative files. Using the cost sensitive evaluator allowed the model to predict all of the negative stories correctly at the cost of all the correct positive stories. No amount of different weight combinations with the evaluator was able to improve on the base random forest. All attempts at changing the costs caused the model to either predict everything as positive or negative.

4.5.2 Lexicon 2

Base Accuracy: 84.17%

a	**b**	**← Classified As**
7	3	a = Positive
16	94	b = negative

Cost Sensitive Evaluator Accuracy: 91.67%

a	**b**	**← Classified As**
0	10	a = positive
0	110	b = negative

Cost Sensitive Evaluator Accuracy: 41.67%

a	**b**	**← Classified As**

8	2	a = positive
68	42	b = negative

Due to the high amount of negative classifications in Lexicon 2 the random forest performed exceptionally well in its base configuration. With an accuracy of 84%, the model predicted a good ratio of both positive classifications and negative classifications correctly. With the cost sensitive evaluator we could not generate an improvement. We were able to achieve an accuracy of 91.67% by putting a cost of 15 on false positives and a cost of 1 on false negatives. This however caused the model to only predict all instances as negative.

Our second attempt that was somewhat close to the base accuracy was by applying a weight of 5 to false positives and 150 to false negatives. This got an accuracy of 41.67%, it was slightly better at predicting positive instances, at 8 instead of 7. However, this came at the cost of 40% accuracy.

4.5.3 Lexicon 3

Base Accuracy: 9.80%

a	**b**	**← Classified As**
10	0	A = positive
110	0	B = negative

Cost Sensitive Evaluator Accuracy: 8.33%

a	b	← Classified As
10	0	A = positive
110	0	B = negative

4.5.4 Lexicon C

Base Accuracy: 9.17%

a	b	← Classified As
10	0	A = positive
109	1	B = negative

Cost Sensitive Evaluator Accuracy: 8.33%

a	b	← Classified As
10	0	A = positive
110	0	B = negative

Both Lexicon 3 and Lexicon Combined have the same issue as Lexicon 1. When using the evaluator on these Lexica, we were unable to find a combination that did not cause the model to predict only one type of classification. We believe that the Lexicon 2 was the only real successful model because its training data contained a significant number of negative classifications compared to the other three Lexica.

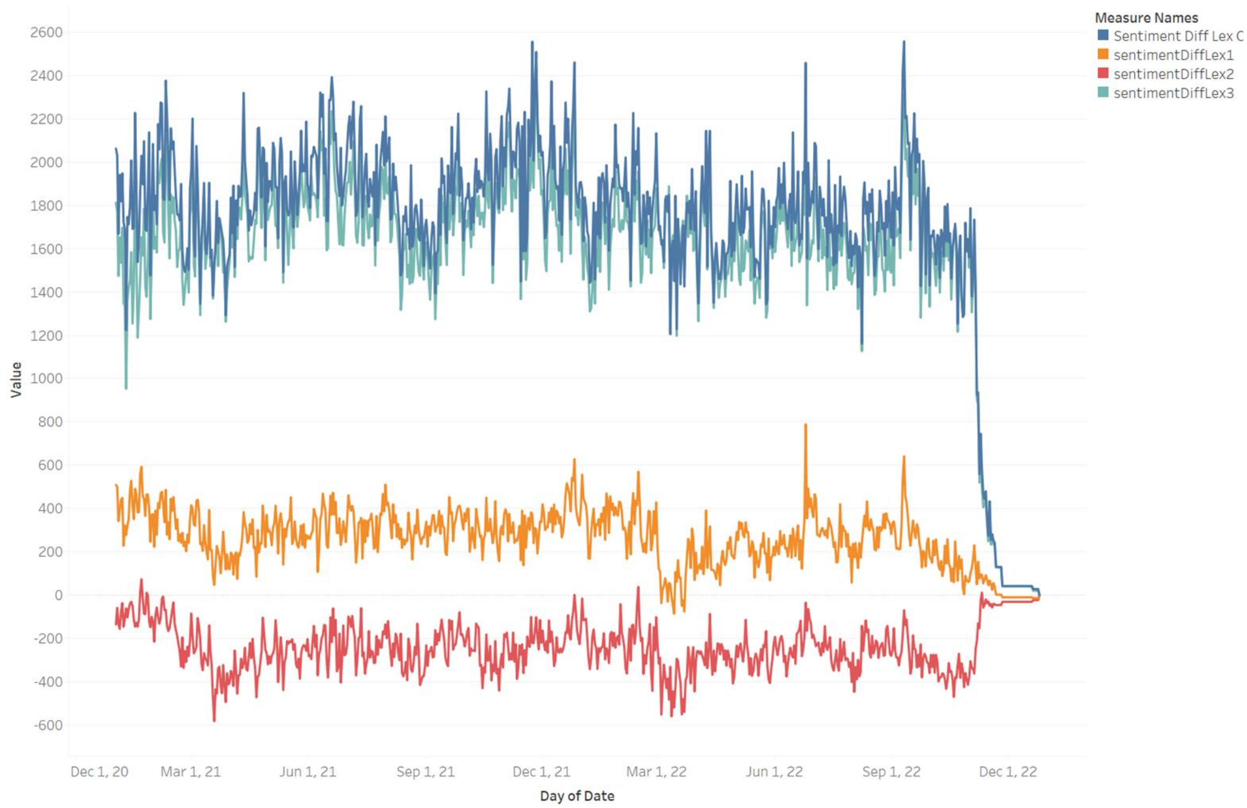


Figure 1: The total sentiment difference per day for Lexicon 1,2,3, and C

Figure 1 shows the total sentiment difference per day for each Lexicon; it is important to note that due to Archive.org's implementation, some HTML files were encoded with the wrong “Publish” date. As mentioned earlier, sentiment difference is calculated by the total count of positive words for a file subtracted from the total number of negative words for a file. This chart allows for an easy way to compare how positive or negative each Lexicon thought a day was. It is important to note that even if a Lexicon has positive or negative sentiment difference sum, there could still be files classified as the opposite value. It just indicates what the majority of files were classified as. As expected, Lexicon 2 is the only Lexicon that has a consistent negative total sentiment, and this corresponds with its overwhelming negative classification. The chart also confirms our suspicion that Lexicon C is dominated by the results of Lexicon 3 as both Lexica have almost identical total sentiment differences per day.

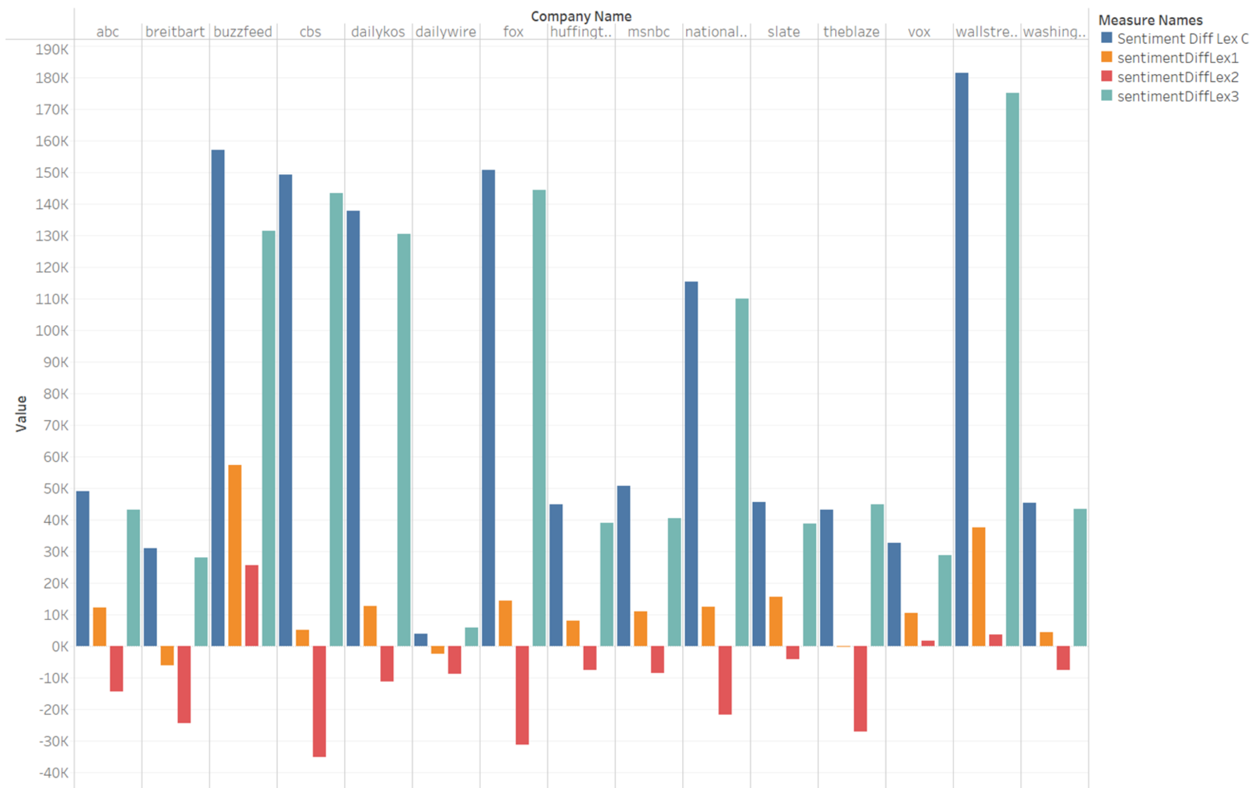


Figure 2: The total sentiment difference by company for Lexicon 1,2,3, and C

Figure 2 shows the total sentiment difference for each of the new sources for each Lexicon. The chart confirms what was shown in Figure 1, that being Lexicon 3 and Combined have the largest sentiment difference values, Lexicon 1 has a modestly positive sentiment difference sum, and Lexicon 2 has a negative sentiment difference. However, we also see that each news source can make a significant difference in the sentiment difference sum. The Wall

Street Journal, Vox, and Buzzfeed, seem to contain a significant number of positive words for each Lexicon as they are the only three sources with all the Lexica having a positive sentiment difference sum.

Another insight this visualization brings forward is the lack of correlation between the Lexica. Buzzfeed, Fox and the Wall Street Journal according to Lexicon 3 and Lexicon Combined are extremely positive. They have some of the largest positive sentiment difference sums; however, Lexicon 2 had dramatically different sentiment sums. For Lexicon 2, Buzzfeed compared to all other sources was extremely positive and Fox was one of the most negative, even though Lexicon 3 and Lexicon Combined had similar sentiment differences between Fox and Buzzfeed. Compared to Buzzfeed and Fox, the Wall Street Journal has a larger sentiment difference sum for Lexicon 3 and Lexicon Combined, yet Lexicon 2 only has a marginally positive sentiment difference sum when compared to the Buzzfeed total.

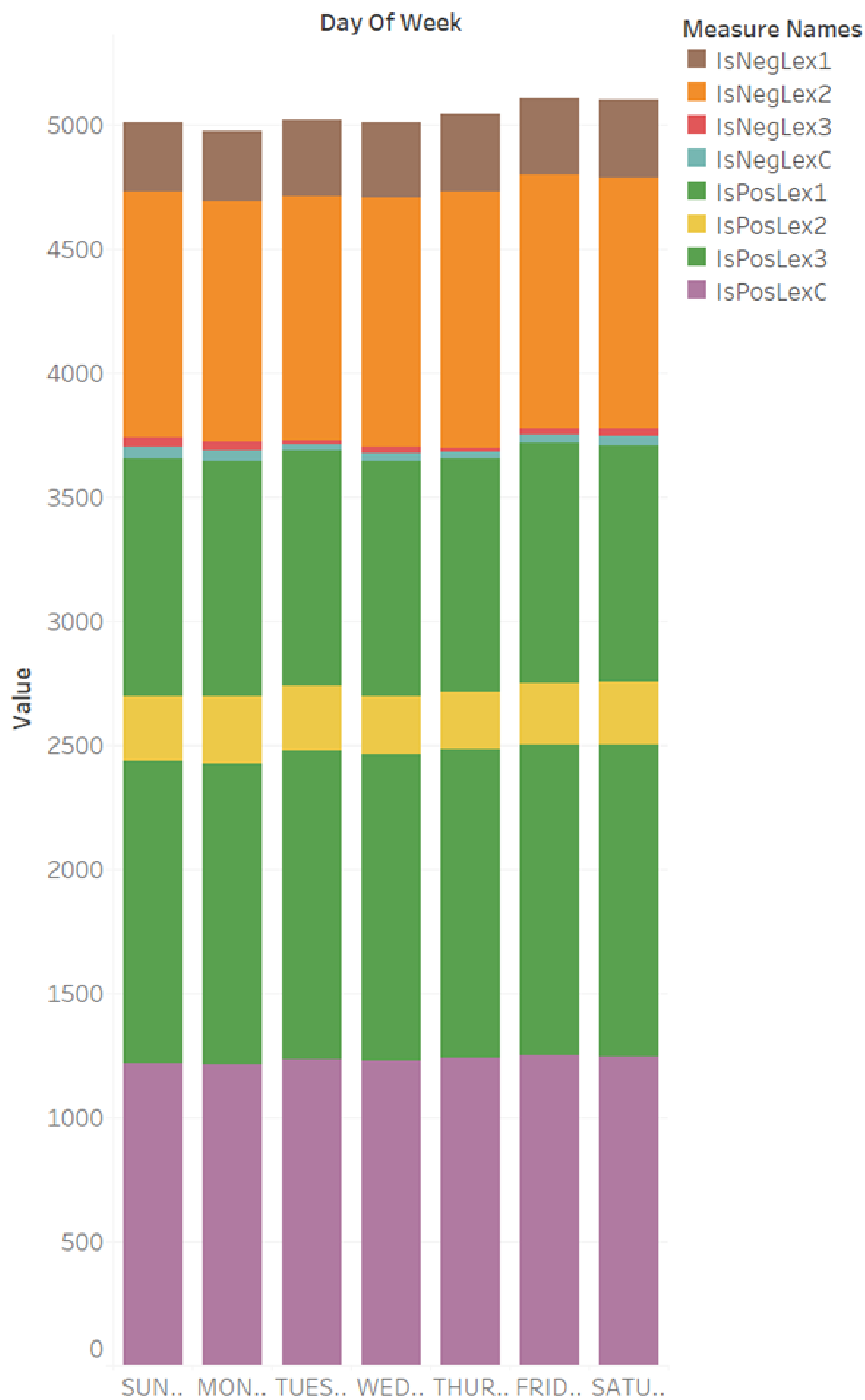


Figure 3: The total count of positive and negative HTML files per day of the week for each Lexicon

Figure 3 shows the count of positive and negative HTML files for every Lexicon for each day of the week. As has been shown in the previous figures, Lexicon 2 is the only Lexicon with a significant number of negative classifications. The visualization also demonstrates that the day of the week makes no impact with the count of positive or negative HTML files, as each day has no significant classification differences.

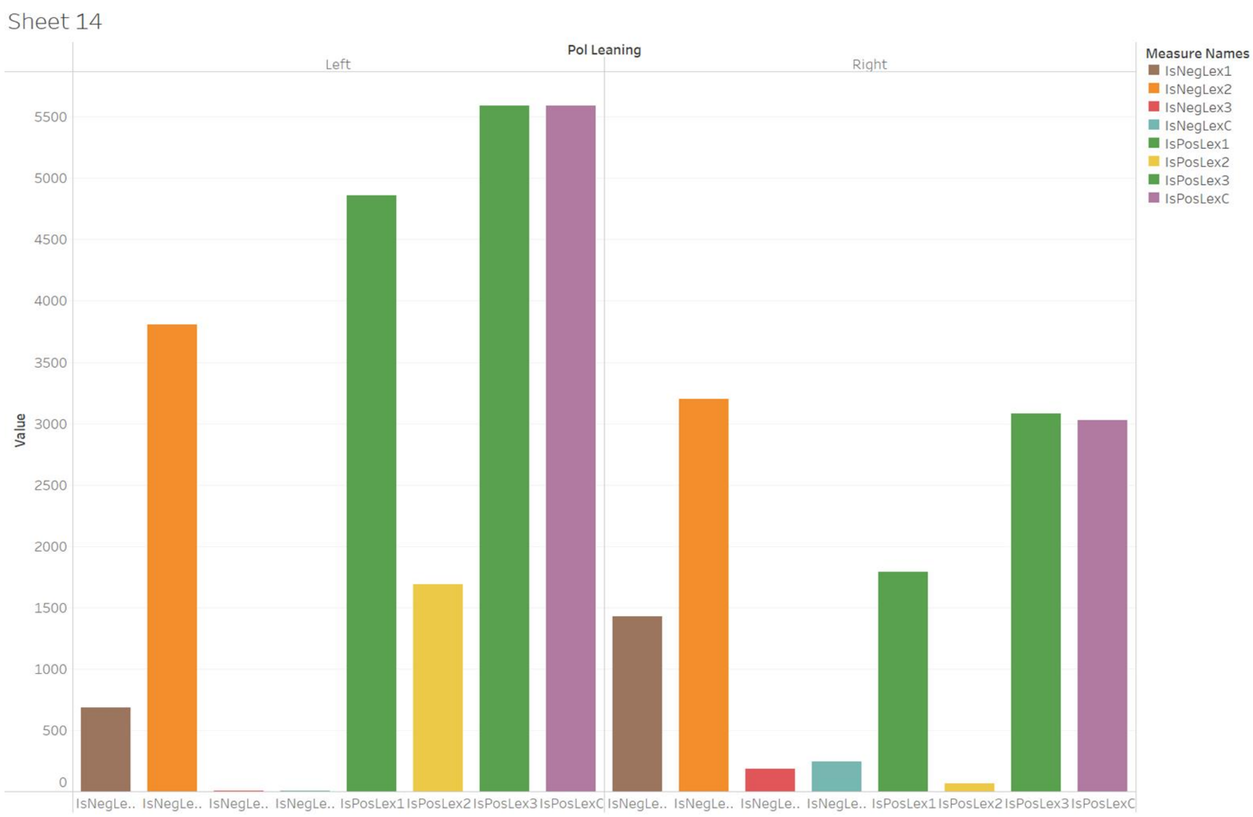


Figure 4: The count of negative and positive classified HTML files by their political leaning, Left or Right

Figure 4 shows that the political leaning of a news source/HTML file makes an impact on the classification of either positive or negative. While the Lexica classify fewer right leaning HTML files, Lexica 1, 3, and Combined have much larger counts of negative files in right leaning files. Due to the underrepresentation of right leaning files in the training data sets, their higher counts may indicate right leaning sources trend more negatively. On the other hand, Lexicon 2 has a larger count of negative files in left leaning files, which would indicate no real difference between the left or right sources because in the training data set there are more left leaning files present.

5. Overall Results

The results of the various learning algorithms and the visualizations demonstrate that the files published year and day of the week make no impact on the classification of the news site’s polarity; however, our results gave credence that company name and political leaning may have some predictive power in home page classification.

While OneR with some Lexica used year to predict the classification, those rules produced very low accuracy ratings when used on the testing data set. Furthermore, Figure 3 showed that there were no significant differences between the day of the week and the count of negative and positive files for all four Lexica.

The more advanced machine learning algorithms of J48, SVM, and Random Forest were able to generate models with high amounts of accuracy, normally close to 80% or more. However, these high accuracies were at the cost of classifying all of the test HTML files as negative, which was the dominant classification. This made these high accuracies less desirable. We attempted to create models with lower accuracy, but were able to predict a mix of both positive and negative files. This was done using a cost sensitive evaluator. However, even with the cost sensitive evaluator Lexica 1, 3, and Combined found limited success due to their skewed inclination toward positive classification. Lexicon 2 we found to be the most successful, as even though it would not have the highest accuracy, it was able to predict the best mix of positive and negative files.

6. Conclusion

After running our tests, we feel that we cannot draw any conclusions about our results and apply it to news sources. We believe this is due to a multitude of reasons. Firstly our results were extremely skewed by the massive difference between the human classified testing data set and how the 4 Lexica classified the HTML files. Secondly, the limitations of Achrive.org forces us to look at a new source’s home page instead of individual stories.

Most of the human evaluated sources were classified as negative. When the rest of the training data was run through the sentiment program, most turned out to be positive. This means there must be some discrepancy or issue in how the “hand-done” analyses were done. If any work on this project were to continue a more suitable Lexicon would be needed. Our results indicated that Lexicon 2 produces the results most similar to how we classified HTML files. However, it was made of the smallest list of words, indicating that there may be positive words we missed along with Lexicon 2 when classifying the files. Alternatively the extra words present in the other Lexica could have caused the skewed positive classification, which could be inaccurate at least in the context of news headlines.

Archive.org forced us into an abstraction of our initial idea. By looking at only the home page of the news source, we lost the ability to look at an article's author, the genre of the article, and made the experiment more susceptible to click bait titles. By not being able to capture the author or genre, we could not see if specific news categories or authors disproportionately affect the positive or negative classification. Furthermore, by not looking at the text of the article, clickbait titles may be skewing our data. Click bait titles are often used by authors to try and quickly grab a reader's attention by using an exaggerated title. These titles could heavily skew our data and in the experiment's current iteration we had no way to account for these titles.

Ultimately, while this experiment was unable to produce any definitive results and conclusions, for future projects continuing in this vein, the lessons learned here will be useful in producing better results.

7. Appendix

Java code to: create connection, gather home page text and output to file, and each source’s url

```java
public class WebScraper {

    public WebScraper() {
    }

    /**
     * gets the html from the input url, sleeps for 5 seconds upon making connection
     * to avoid IP banning
     *
     * @param url
     * @return
     */
    public HtmlPage getDocument(String url) {

        HtmlPage page = null;
        try {
        try (final WebClient webClient = new WebClient(BrowserVersion.FIREFOX)) {
            webClient.getOptions().setUseInsecureSSL(true);
            webClient.getOptions().setCssEnabled(false);
            webClient.getOptions().setJavaScriptEnabled(false);

            try {
                Thread.sleep(5000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            page = webClient.getPage(url);

        } catch (IOException e) {
            e.printStackTrace();
        }
        return page;
        } catch(com.gargoylesoftware.htmlunit.FailingHttpStatusCodeException e) {
//          e.printStackTrace();
        }
        return null;
    }
```

```java
public void fetchHomePages() throws IOException {
    DayOfWeek day = LocalDate.parse(this.startDate.toString()).getDayOfWeek();
    String[] dateArr;
    String htmlText;
    String incrementedURL;
    File file = new File("null file instantiated");

    for (int i = 0; i < 365; i++) {
        dateArr = getDate(i, day);
        file = createNextFile(dateArr, file); // used with file.delete()

        incrementedURL = incrementUrl(dateArr);

        try {
        htmlText = WS.getDocument(incrementedURL).asNormalizedText();
        } catch(NullPointerException n) {
            file.delete();
            continue;
        }
        if (noTimeStamp(htmlText, dateArr[0])) {
            createNewBadFile(dateArr);
            file.delete();
            continue;
        }
        clipHtml(htmlText, dateArr[0]);
        // testing purposes : bw.write(htmlText);

        System.out.println((i + 1) + " files have been created, most recent file: " + dateArr[0]);

    }
    badFiles.close();
}
```

```java
public class Abc extends Source {
    // https://web.archive.org/web/20221110033948/https://abcnews.go.com/

    public Abc() {
        super("https://web.archive.org/web/", "033948/https://abcnews.go.com/");
    }

    public Abc(String urlFirstHalf, String urlSecondHalf) {
        super(urlFirstHalf, urlSecondHalf);
    }

    public void fetchHomePages() throws IOException {
        super.fetchHomePages();
    }
}
```

Python Code to classify HTML files as Positive/Negative

```python
import pandas as pd
import numpy as np
import os
import csv
import re
import sys
import io
from soupsieve import match
```

[32] Python

```python
directory = 'C:/Users/Marc/Desktop/CSC-272/SentimentProject/Data'
DAYOFWEEK = {'MONDAY','TUESDAY', 'WEDNESDAY', 'THURSDAY', 'FRIDAY', 'SATURDAY', 'SUNDAY'}
sentimentdfOne = pd.read_csv("C:/Users/Marc/Desktop/CSC-272/SentimentProject/Lexicon/LexiconCombined.csv", encoding='utf-8')
SentimentDictOne = {}
for index in range(0,len(sentimentdfOne)):
    SentimentDictOne[sentimentdfOne['Word'][index]] = sentimentdfOne['Polarity'][index]

```

[33] Python

```python
df = pd.DataFrame({'FileName' : pd.Series(dtype = 'str'),
                    'Year' : pd.Series(dtype = 'str'),
                    'DayOfWeek' : pd.Series(dtype = 'str'),
                    'companyName' : pd.Series(dtype= 'str'),
                    'TotalScoreGood' : pd.Series(dtype = 'int'),
                    'TotalScoreBad' : pd.Series(dtype = 'int'),
                    'TotalScoreNeutral' : pd.Series(dtype = 'int'),
                    'sentimentDiff' : pd.Series(dtype='int'),
                    'Sentiment' : pd.Series(dtype = 'str')})
```

[34] Python

The following 4 blocks are used to create the updaded lexcion dataset. The new data set only keeps the word, the type, and the polarity of the orginal data set

Markdown

```python
lexicon = []
lexicondf = pd.DataFrame({'Word' : pd.Series(dtype= 'str'), 'Type' : pd.Series(dtype = 'str'), 'Polarity' : pd.Series(dtype = 'str')})
```

[ ] Python

```python

with open('C:/Users/Marc/Desktop/CSC-272/SentimentProject/Lexicon/subjclueslen1-HLTEMNLP05.tff') as f:
    for line in f:
        lexicon.append(line)

print(len(lexicon))
```

[128] Python

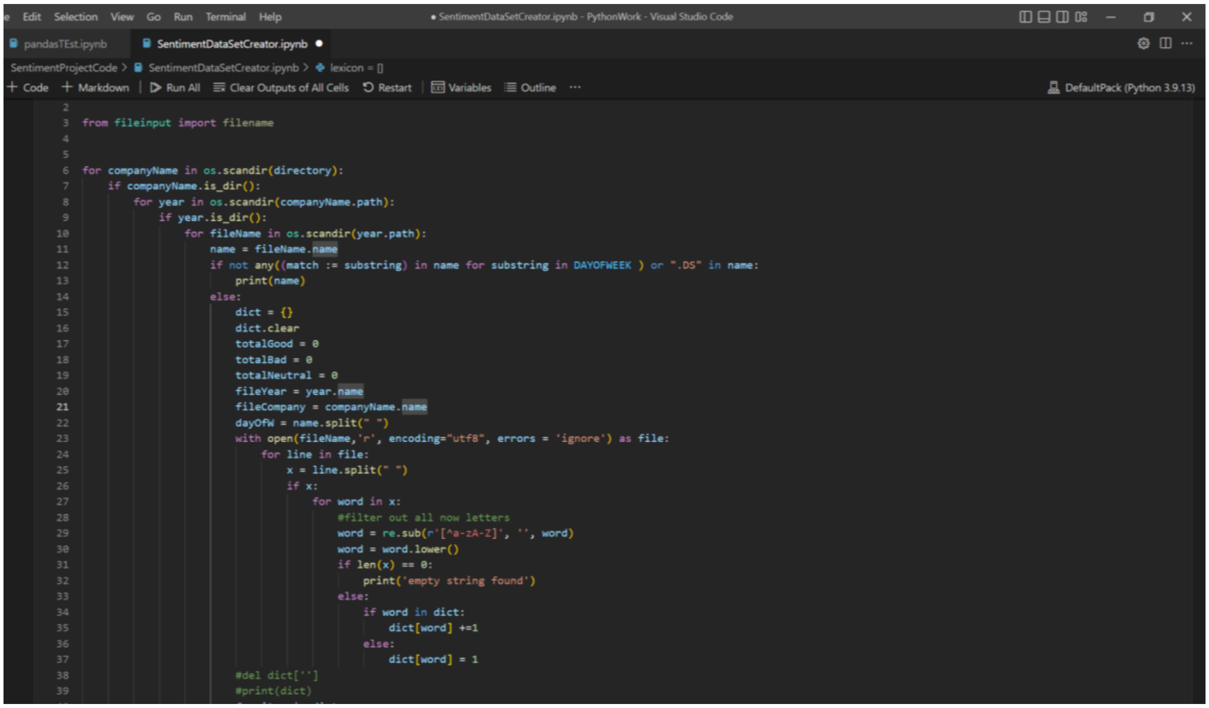
```
8222
```

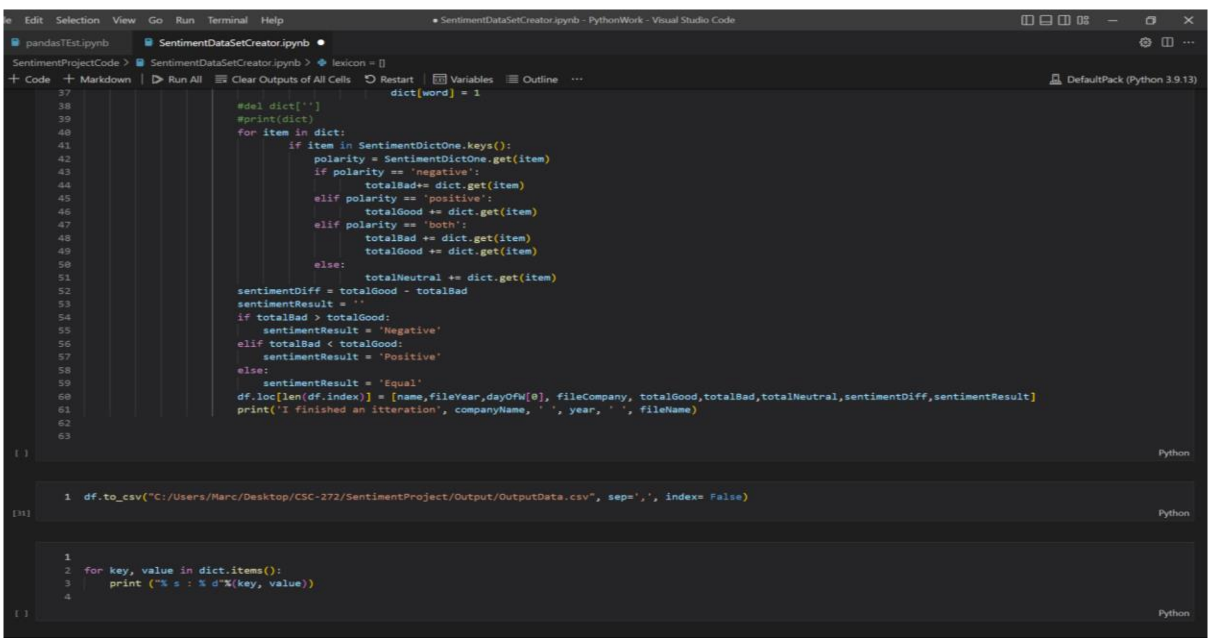
```python
for line in lexicon:
    x = line.split(" ")
    for e in range(0,len(x)):
        x[e] = x[e].replace('word1=', '')
        x[e] = x[e].replace('type=', '')
        x[e] = x[e].replace('priorpolarity=', '')
        x[e] = x[e].replace('\n', '')
    lexicondf.loc[len(lexicondf.index)] = [x[2],x[0],x[5]]

print(lexicondf)
```

[ ] Python

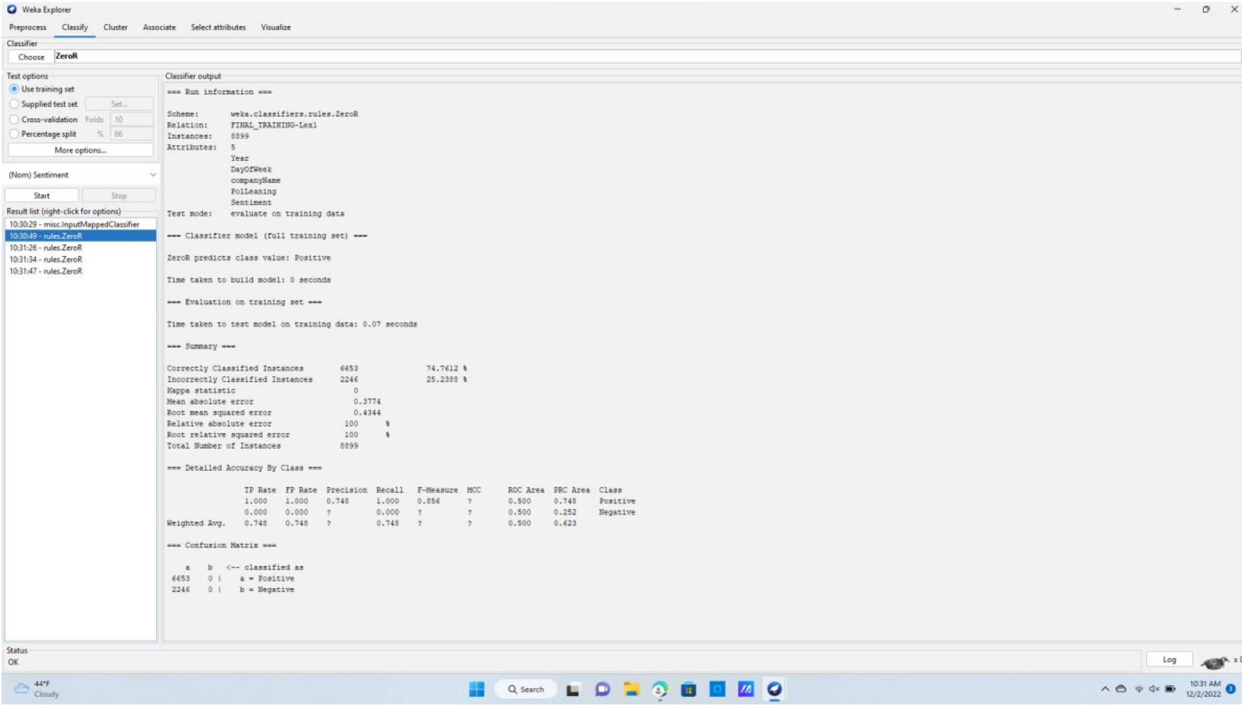
```python
lexicondf.to_csv("C:/Users/Marc/Desktop/CSC-272/SentimentProject/Lexicon/Lexicon.csv", sep=',', index= False)
```

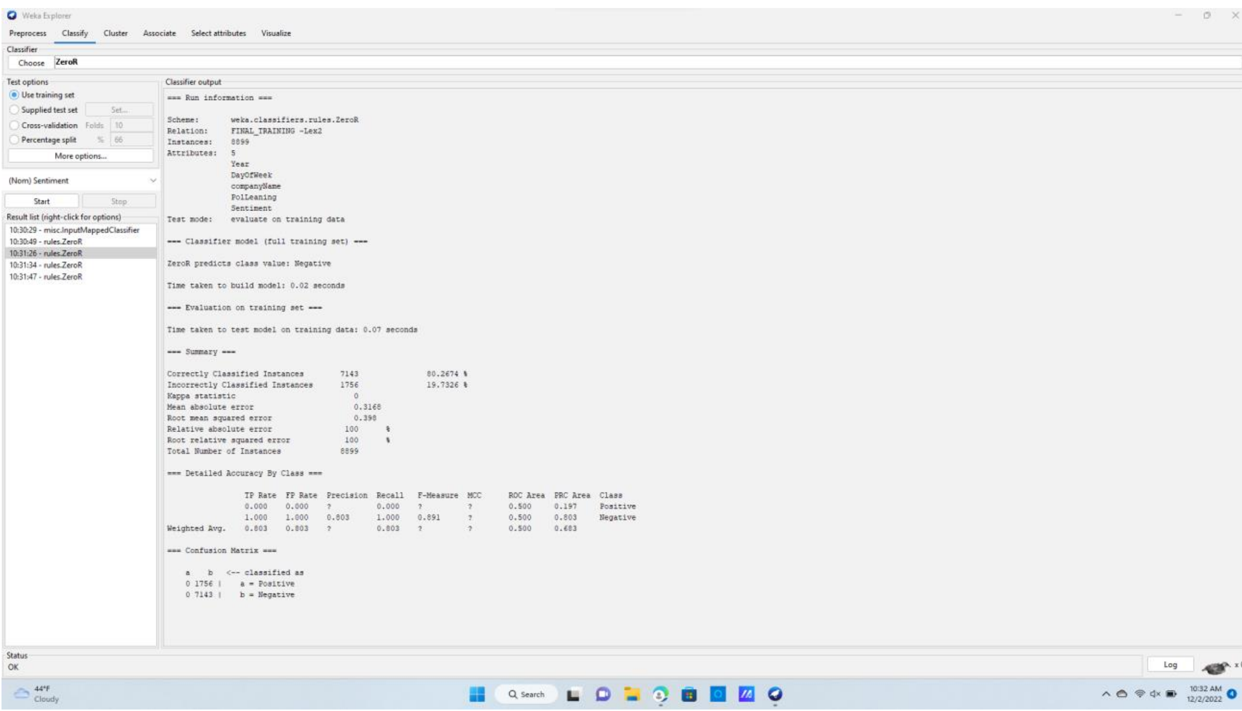
[131] Python

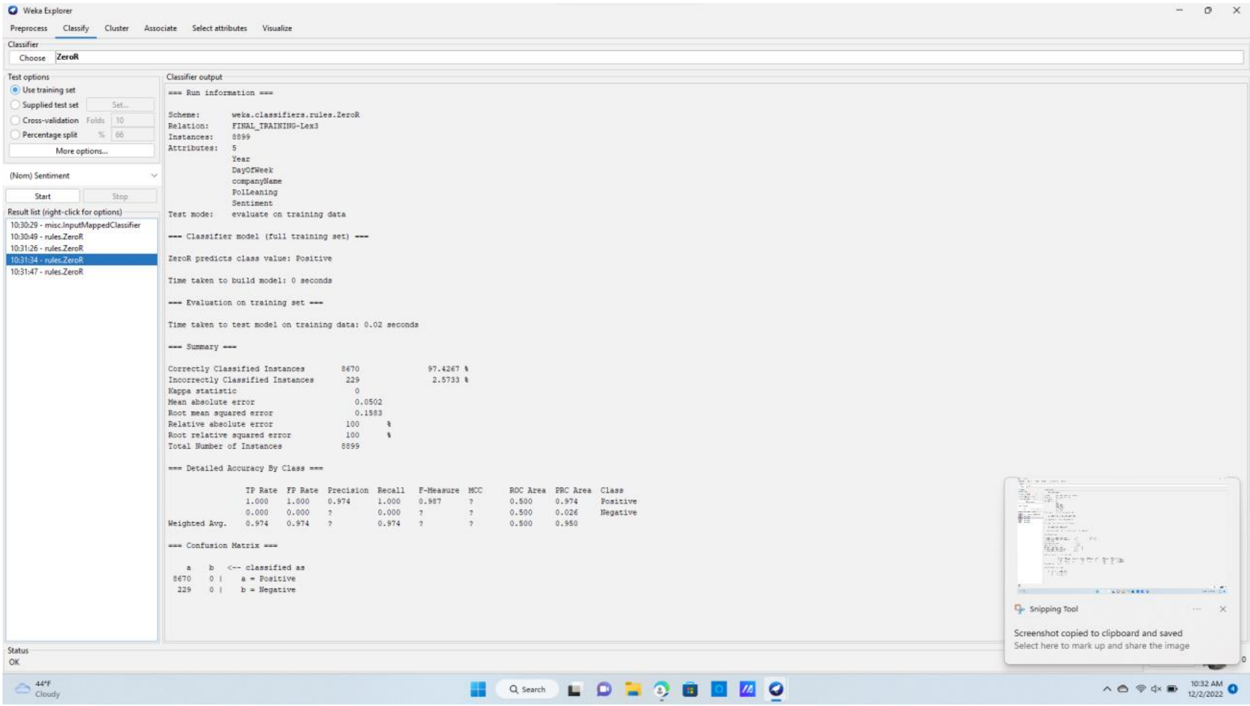


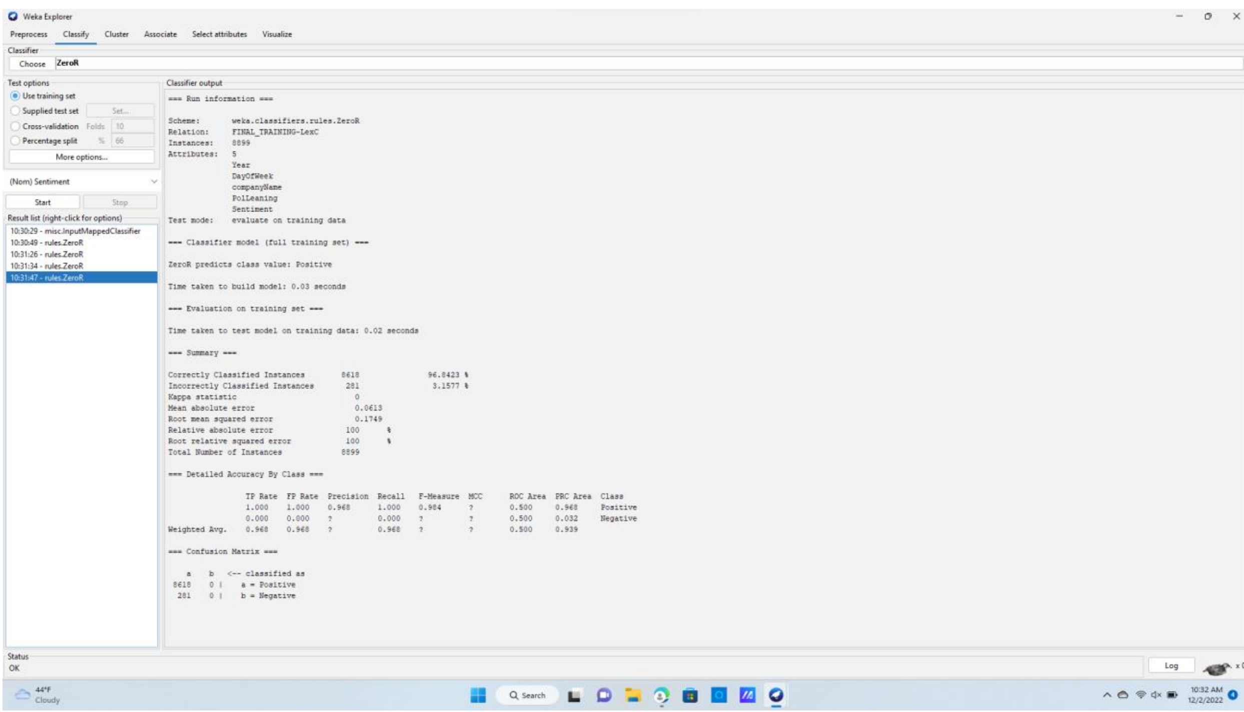


ZeroR - Lexicon 1 - Lexicon C

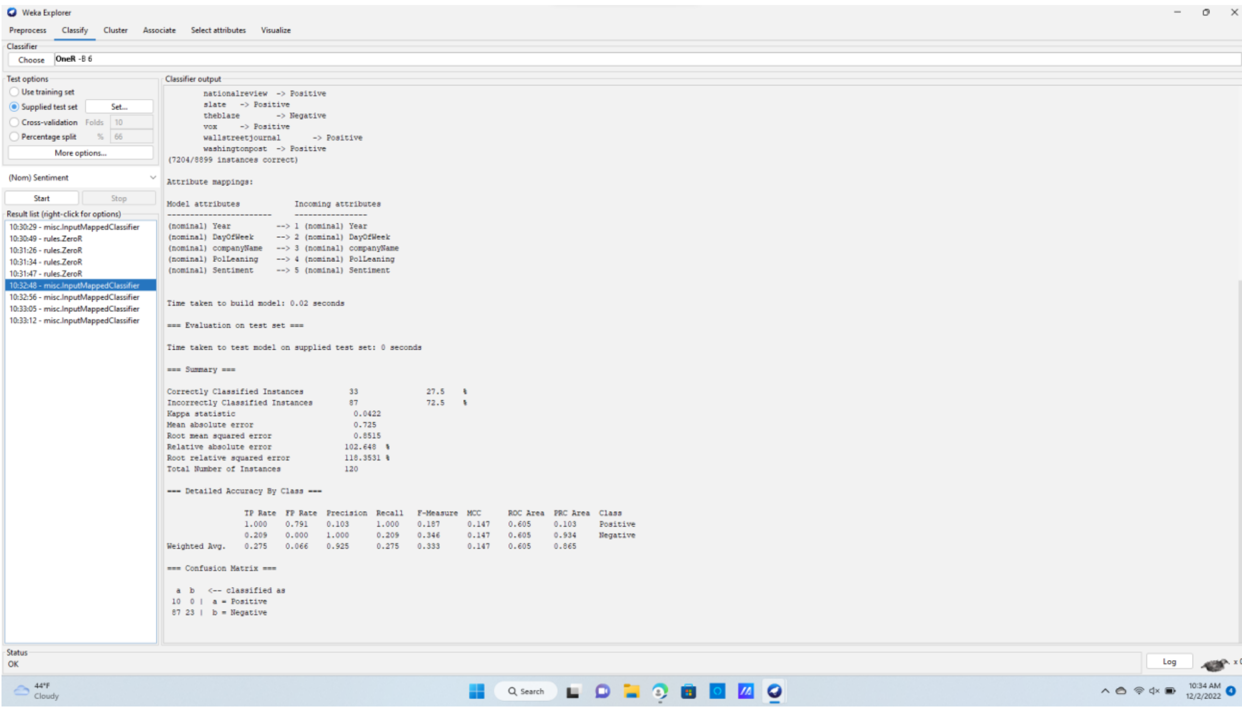

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ZeroR
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) Sentiment
Start
Stop
Result list (right-click for options)
10:30:29 - misc.InputMappedClassifier
10:30:49 - rules.ZeroR
10:31:26 - rules.ZeroR
10:31:34 - rules.ZeroR
10:31:47 - rules.ZeroR
Classifier output
=== Run information ===
Scheme: weka.classifiers.rules.ZeroR
Relation: FINAL_TRAINING-Lex1
Instances: 8899
Attributes: 5
Year
DayOfWeek
companyName
PolLeaning
Sentiment
Test mode: evaluate on training data
=== Classifier model (full training set) ===
ZeroR predicts class value: Positive
Time taken to build model: 0 seconds
=== Evaluation on training set ===
Time taken to test model on training data: 0.07 seconds
=== Summary ===
Correctly Classified Instances 6653 74.7612 %
Incorrectly Classified Instances 2246 25.2388 %
Kappa statistic 0
Mean absolute error 0.3774
Root mean squared error 0.4344
Relative absolute error 100 %
Root relative squared error 100 %
Total Number of Instances 8899
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
1.000 1.000 0.748 1.000 0.856 ? 0.500 0.748 Positive
0.000 0.000 ? 0.000 ? ? 0.500 0.252 Negative
Weighted Avg. 0.748 0.748 ? 0.748 ? ? 0.500 0.623
=== Confusion Matrix ===
a b <-- classified as
6653 0 | a = Positive
2246 0 | b = Negative
Status
OK
Log
44°F
Cloudy
Search
10:31 AM
12/2/2022

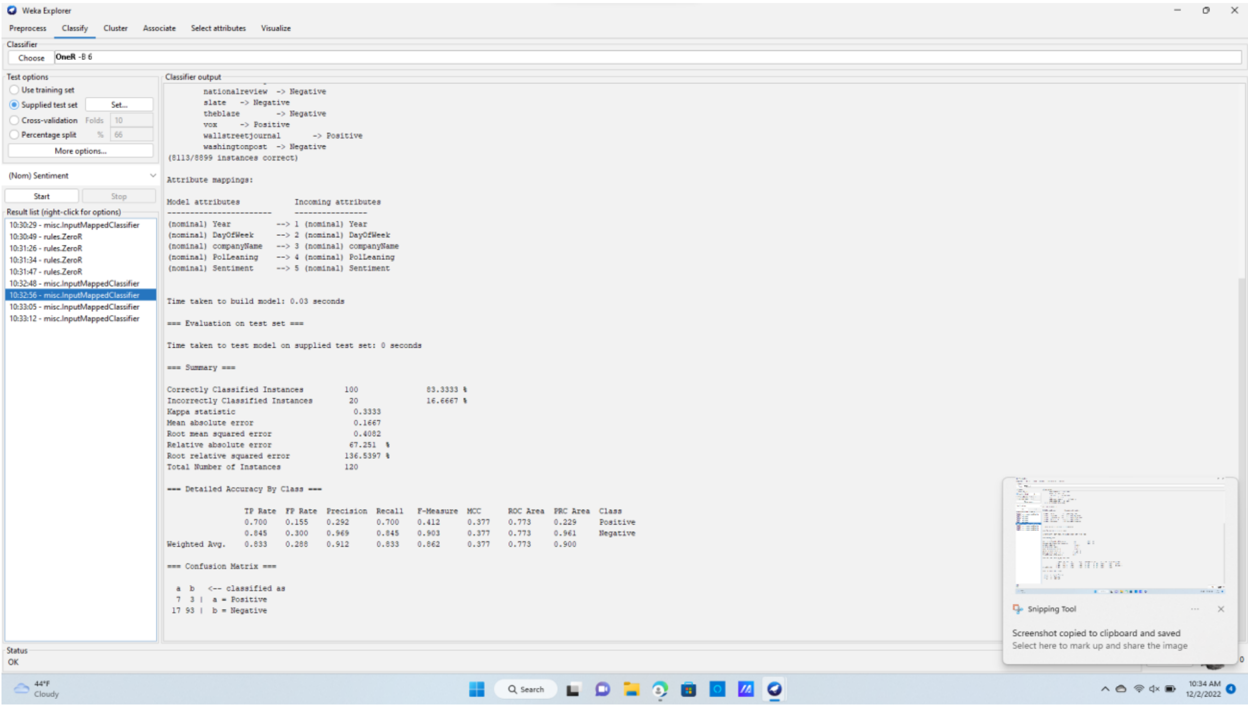

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ZeroR
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) Sentiment
Start
Stop
Result list (right-click for options)
10:30:29 - misc.InputMappedClassifier
10:30:49 - rules.ZeroR
10:31:26 - rules.ZeroR
10:31:34 - rules.ZeroR
10:31:47 - rules.ZeroR
Classifier output
=== Run information ===
Scheme: weka.classifiers.rules.ZeroR
Relation: FINAL_TRAINING -Lex2
Instances: 8899
Attributes: 5
Year
DayOfWeek
companyName
PolLeaning
Sentiment
Test mode: evaluate on training data
=== Classifier model (full training set) ===
ZeroR predicts class value: Negative
Time taken to build model: 0.02 seconds
=== Evaluation on training set ===
Time taken to test model on training data: 0.07 seconds
=== Summary ===
Correctly Classified Instances 7143 80.2674 %
Incorrectly Classified Instances 1756 19.7326 %
Kappa statistic 0
Mean absolute error 0.3168
Root mean squared error 0.398
Relative absolute error 100 %
Root relative squared error 100 %
Total Number of Instances 8899
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.000 0.000 ? 0.000 ? ? 0.500 0.197 Positive
1.000 1.000 0.803 1.000 0.891 ? 0.500 0.803 Negative
Weighted Avg. 0.803 0.803 ? 0.803 ? ? 0.500 0.683
=== Confusion Matrix ===
a b <-- classified as
0 1756 | a = Positive
0 7143 | b = Negative
Status
OK
Log
44°F
Cloudy
Search
10:32 AM
12/2/2022

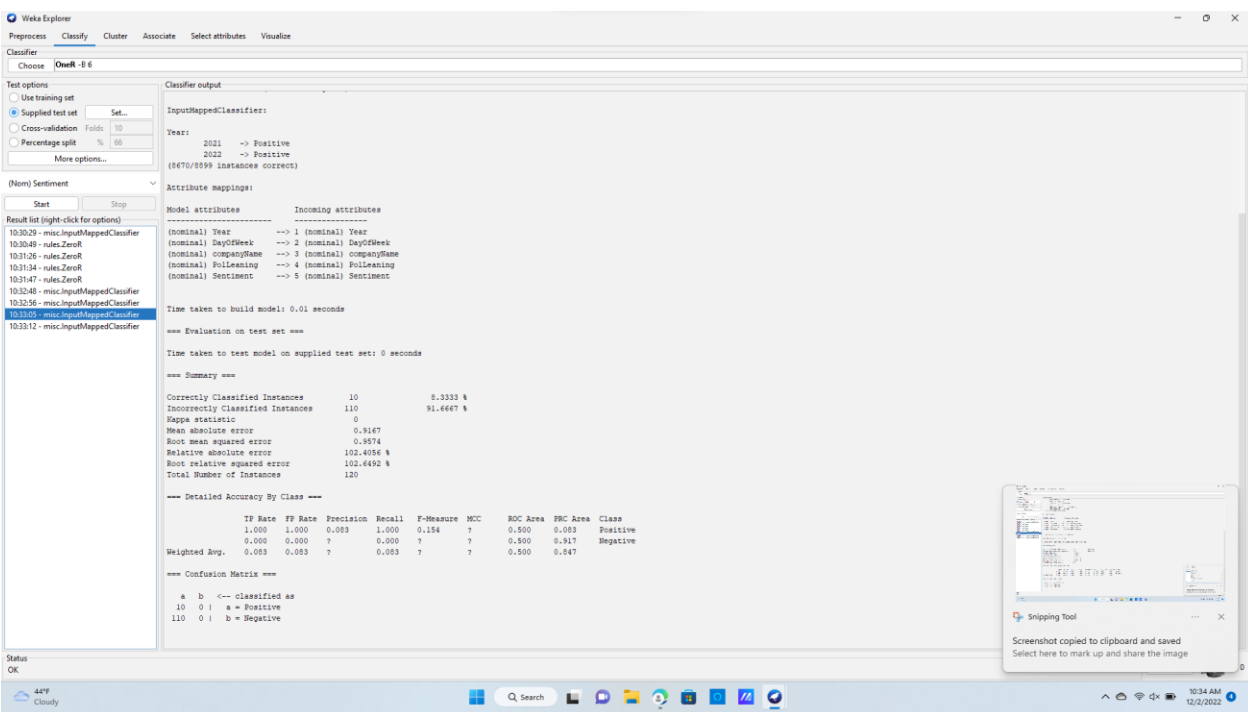

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ZeroR
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) Sentiment
Start
Stop
Result list (right-click for options)
10:30:29 - misc.InputMappedClassifier
10:30:49 - rules.ZeroR
10:31:26 - rules.ZeroR
10:31:34 - rules.ZeroR
10:31:47 - rules.ZeroR
Classifier output
=== Run information ===
Scheme: weka.classifiers.rules.ZeroR
Relation: FINAL_TRAINING-Lex3
Instances: 8899
Attributes: 5
Year
DayOfWeek
companyName
PolLeaning
Sentiment
Test mode: evaluate on training data
=== Classifier model (full training set) ===
ZeroR predicts class value: Positive
Time taken to build model: 0 seconds
=== Evaluation on training set ===
Time taken to test model on training data: 0.02 seconds
=== Summary ===
Correctly Classified Instances 8670 97.4267 %
Incorrectly Classified Instances 229 2.5733 %
Kappa statistic 0
Mean absolute error 0.0502
Root mean squared error 0.1583
Relative absolute error 100 %
Root relative squared error 100 %
Total Number of Instances 8899
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
1.000 1.000 0.974 1.000 0.987 ? 0.500 0.974 Positive
0.000 0.000 ? 0.000 ? ? 0.500 0.026 Negative
Weighted Avg. 0.974 0.974 ? 0.974 ? ? 0.500 0.950
=== Confusion Matrix ===
a b <-- classified as
8670 0 | a = Positive
229 0 | b = Negative
Snipping Tool
Screenshot copied to clipboard and saved
Select here to mark up and share the image
Status
OK
44°F
Cloudy
Search
10:32 AM
12/2/2022

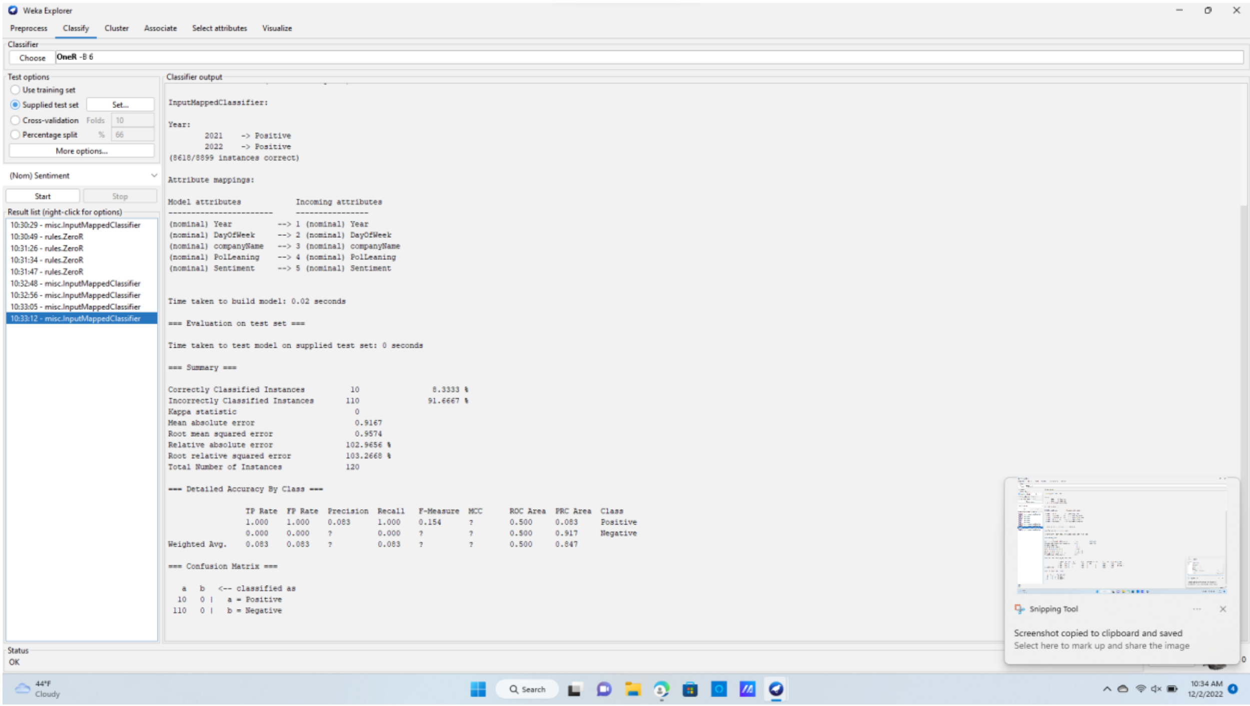

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ZeroR
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) Sentiment
Start
Stop
Result list (right-click for options)
10:30:29 - misc.InputMappedClassifier
10:30:49 - rules.ZeroR
10:31:26 - rules.ZeroR
10:31:34 - rules.ZeroR
10:31:47 - rules.ZeroR
Classifier output
=== Run information ===
Scheme: weka.classifiers.rules.ZeroR
Relation: FINAL_TRAINING-LexC
Instances: 8899
Attributes: 5
Year
DayOfWeek
companyName
PolLeaning
Sentiment
Test mode: evaluate on training data
=== Classifier model (full training set) ===
ZeroR predicts class value: Positive
Time taken to build model: 0.03 seconds
=== Evaluation on training set ===
Time taken to test model on training data: 0.02 seconds
=== Summary ===
Correctly Classified Instances 8618 96.8423 %
Incorrectly Classified Instances 281 3.1577 %
Kappa statistic 0
Mean absolute error 0.0613
Root mean squared error 0.1749
Relative absolute error 100 %
Root relative squared error 100 %
Total Number of Instances 8899
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
1.000 1.000 0.968 1.000 0.984 ? 0.500 0.968 Positive
0.000 0.000 ? 0.000 ? ? 0.500 0.032 Negative
Weighted Avg. 0.968 0.968 ? 0.968 ? ? 0.500 0.939
=== Confusion Matrix ===
a b <-- classified as
8618 0 | a = Positive
281 0 | b = Negative
Status
OK
Log
44°F
Cloudy
Search
10:32 AM
12/2/2022

OneR - Lexicon 1 - LexiconC




Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
OneR -B 6
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) Sentiment
Start
Stop
Result list (right-click for options)
10:30:29 - misc.InputMappedClassifier
10:30:49 - rules.ZeroR
10:31:26 - rules.ZeroR
10:31:34 - rules.ZeroR
10:31:47 - rules.ZeroR
10:32:48 - misc.InputMappedClassifier
10:32:56 - misc.InputMappedClassifier
10:33:05 - misc.InputMappedClassifier
10:33:12 - misc.InputMappedClassifier
Classifier output
nationalreview -> Negative
slate -> Negative
theblaze -> Negative
vox -> Positive
wallstreetjournal -> Positive
washingtonpost -> Negative
(8113/8899 instances correct)
Attribute mappings:
Model attributes
Incoming attributes
(nominal) Year --> 1 (nominal) Year
(nominal) DayOfWeek --> 2 (nominal) DayOfWeek
(nominal) companyName --> 3 (nominal) companyName
(nominal) PolLeaning --> 4 (nominal) PolLeaning
(nominal) Sentiment --> 5 (nominal) Sentiment
Time taken to build model: 0.03 seconds
=== Evaluation on test set ===
Time taken to test model on supplied test set: 0 seconds
=== Summary ===
Correctly Classified Instances 100 83.3333 %
Incorrectly Classified Instances 20 16.6667 %
Kappa statistic 0.3333
Mean absolute error 0.1667
Root mean squared error 0.4082
Relative absolute error 67.251 %
Root relative squared error 136.5397 %
Total Number of Instances 120
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.700 0.155 0.292 0.700 0.412 0.377 0.773 0.229 Positive
0.845 0.300 0.969 0.845 0.903 0.377 0.773 0.961 Negative
Weighted Avg. 0.833 0.288 0.912 0.833 0.862 0.377 0.773 0.900
=== Confusion Matrix ===
a b <-- classified as
7 3 | a = Positive
17 93 | b = Negative
Snipping Tool
Screenshot copied to clipboard and saved
Select here to mark up and share the image
Status
OK
44°F
Cloudy
Search
10:34 AM
12/2/2022


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
OneR -B 6
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) Sentiment
Start
Stop
Result list (right-click for options)
10:30:29 - misc.InputMappedClassifier
10:30:49 - rules.ZeroR
10:31:26 - rules.ZeroR
10:31:34 - rules.ZeroR
10:31:47 - rules.ZeroR
10:32:48 - misc.InputMappedClassifier
10:32:56 - misc.InputMappedClassifier
10:33:05 - misc.InputMappedClassifier
10:33:12 - misc.InputMappedClassifier
Classifier output
InputMappedClassifier:
Year:
2021 -> Positive
2022 -> Positive
(8670/8899 instances correct)
Attribute mappings:
Model attributes
Incoming attributes
(nominal) Year --> 1 (nominal) Year
(nominal) DayOfWeek --> 2 (nominal) DayOfWeek
(nominal) companyName --> 3 (nominal) companyName
(nominal) PolLeaning --> 4 (nominal) PolLeaning
(nominal) Sentiment --> 5 (nominal) Sentiment
Time taken to build model: 0.01 seconds
=== Evaluation on test set ===
Time taken to test model on supplied test set: 0 seconds
=== Summary ===
Correctly Classified Instances 10 8.3333 %
Incorrectly Classified Instances 110 91.6667 %
Kappa statistic 0
Mean absolute error 0.9167
Root mean squared error 0.9574
Relative absolute error 102.4056 %
Root relative squared error 102.6492 %
Total Number of Instances 120
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
1.000 1.000 0.083 1.000 0.154 ? 0.500 0.083 Positive
0.000 0.000 ? 0.000 ? ? 0.500 0.917 Negative
Weighted Avg. 0.083 0.083 ? 0.083 ? ? 0.500 0.847
=== Confusion Matrix ===
a b <-- classified as
10 0 | a = Positive
110 0 | b = Negative
Snipping Tool
Screenshot copied to clipboard and saved
Select here to mark up and share the image
Status
OK
44°F
Cloudy
Search
10:34 AM
12/2/2022



J48 Lexicon 1 - Base

```
=== Summary ===

Correctly Classified Instances          27               22.6891 %
Incorrectly Classified Instances        92               77.3109 %
Kappa statistic                          0.0287
Mean absolute error                      0.7019
Root mean squared error                  0.7492
Relative absolute error                 98.8446 %
Root relative squared error            103.7575 %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.836    0.089      1.000    0.164      0.121    0.698     0.118     Positive
                 0.164    0.000    1.000      0.164    0.281      0.121    0.700     0.950     Negative
Weighted Avg.    0.227    0.063    0.931      0.227    0.272      0.121    0.700     0.887

=== Confusion Matrix ===

  a  b   <-- classified as
  9  0 |  a = Positive
 92 18 |  b = Negative
```

J48 Lexicon 1 - Cost Sensitive Evaluator

```
=== Summary ===

Correctly Classified Instances          94               78.9916 %
Incorrectly Classified Instances        25               21.0084 %
Kappa statistic                          0.2746
Mean absolute error                      0.2932
Root mean squared error                  0.4409
Relative absolute error                 41.2955 %
Root relative squared error             61.061  %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.778    0.209    0.233      0.778    0.359      0.346    0.842     0.450     Positive
                 0.791    0.222    0.978      0.791    0.874      0.346    0.840     0.979     Negative
Weighted Avg.    0.790    0.221    0.921      0.790    0.835      0.346    0.841     0.939

=== Confusion Matrix ===

  a  b   <-- classified as
  7  2 |  a = Positive
 23 87 |  b = Negative
```

J48 Lexicon 2 - Base

```
=== Summary ===

Correctly Classified Instances         101               84.8739 %
Incorrectly Classified Instances        18               15.1261 %
Kappa statistic                          0.3689
Mean absolute error                      0.1634
Root mean squared error                  0.2922
Relative absolute error                 67.2053 %
Root relative squared error            100.3947 %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.778    0.145    0.304      0.778    0.438      0.423    0.873     0.803     Positive
                 0.855    0.222    0.979      0.855    0.913      0.423    0.869     0.983     Negative
Weighted Avg.    0.849    0.216    0.928      0.849    0.877      0.423    0.869     0.969

=== Confusion Matrix ===

  a  b   <-- classified as
  7  2 |  a = Positive
 16 94 |  b = Negative
```

J48 Lexicon 2 - Cost Sensitive Evaluator

```
=== Summary ===

Correctly Classified Instances         117               98.3193 %
Incorrectly Classified Instances         2                1.6807 %
Kappa statistic                          0.8661
Mean absolute error                      0.0409
Root mean squared error                  0.1383
Relative absolute error                 16.8066 %
Root relative squared error             47.499  %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.778    0.000    1.000      0.778    0.875      0.874    0.873     0.803     Positive
                 1.000    0.222    0.982      1.000    0.991      0.874    0.869     0.983     Negative
Weighted Avg.    0.983    0.205    0.983      0.983    0.982      0.874    0.869     0.969

=== Confusion Matrix ===

   a   b   <-- classified as
   7   2 |   a = Positive
   0 110 |   b = Negative
```

J48 Lexicon 3 - Base

```
=== Summary ===

Correctly Classified Instances           9                7.563  %
Incorrectly Classified Instances       110               92.437  %
Kappa statistic                          0
Mean absolute error                      0.9025
Root mean squared error                  0.9367
Relative absolute error                100.01   %
Root relative squared error            100.0109 %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.076      1.000    0.141      ?        0.500     0.075     Positive
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.917     Negative
Weighted Avg.    0.076    0.076    ?          0.076    ?          ?        0.500     0.853

=== Confusion Matrix ===

   a   b   <-- classified as
   9   0 |   a = Positive
 110   0 |   b = Negative
```

J48 Lexicon 3 - Cost Sensitive Evaluator

```
=== Summary ===

Correctly Classified Instances          14               11.7647 %
Incorrectly Classified Instances       105               88.2353 %
Kappa statistic                          0.0072
Mean absolute error                      0.8205
Root mean squared error                  0.8675
Relative absolute error                 90.9206 %
Root relative squared error             92.624  %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.955    0.079      1.000    0.146      0.060    0.518     0.094     Positive
                 0.045    0.000    1.000      0.045    0.087      0.060    0.518     0.924     Negative
Weighted Avg.    0.118    0.072    0.930      0.118    0.091      0.060    0.518     0.861

=== Confusion Matrix ===

   a   b   <-- classified as
   9   0 |   a = Positive
 105   5 |   b = Negative
```

J48 Lexicon Combined - Base

```
=== Summary ===

Correctly Classified Instances           9                7.563  %
Incorrectly Classified Instances       110               92.437  %
Kappa statistic                          0
Mean absolute error                      0.8976
Root mean squared error                  0.9311
Relative absolute error                100.01   %
Root relative squared error            100.0108 %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.076      1.000    0.141      ?        0.500     0.075     Positive
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.917     Negative
Weighted Avg.    0.076    0.076    ?          0.076    ?          ?        0.500     0.853

=== Confusion Matrix ===

   a   b   <-- classified as
   9   0 |   a = Positive
 110   0 |   b = Negative
```

J48 Lexicon Combined - Cost Sensitive Evaluator

```
=== Summary ===

Correctly Classified Instances          14               11.7647 %
Incorrectly Classified Instances       105               88.2353 %
Kappa statistic                          0.0072
Mean absolute error                      0.8234
Root mean squared error                  0.871
Relative absolute error                 91.7471 %
Root relative squared error             93.548  %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.955    0.079      1.000    0.146      0.060    0.518     0.094     Positive
                 0.045    0.000    1.000      0.045    0.087      0.060    0.518     0.924     Negative
Weighted Avg.    0.118    0.072    0.930      0.118    0.091      0.060    0.518     0.861

=== Confusion Matrix ===

   a   b   <-- classified as
   9   0 |   a = Positive
 105   5 |   b = Negative
```

SVM Lexicon 1 - Base

```
=== Summary ===

Correctly Classified Instances          33               27.7311 %
Incorrectly Classified Instances        86               72.2689 %
Kappa statistic                          0.0405
Mean absolute error                      0.7227
Root mean squared error                  0.8501
Relative absolute error                101.7713 %
Root relative squared error            117.7314 %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.782    0.095      1.000    0.173      0.144    0.608     0.094     Positive
                 0.218    0.000    1.000      0.218    0.358      0.144    0.609     0.935     Negative
Weighted Avg.    0.277    0.059    0.932      0.277    0.344      0.144    0.609     0.871

=== Confusion Matrix ===

  a  b   <-- classified as
  9  0 |  a = Positive
 86 24 |  b = Negative
```

SVM Lexicon 1 - Cost Sensitive Evaluator

```
=== Summary ===

Correctly Classified Instances          94               78.9916 %
Incorrectly Classified Instances        25               21.0084 %
Kappa statistic                          0.2746
Mean absolute error                      0.2101
Root mean squared error                  0.4583
Relative absolute error                 29.5847 %
Root relative squared error             63.4765 %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.778    0.209    0.233      0.778    0.359      0.346    0.781     0.192     Positive
                 0.791    0.222    0.978      0.791    0.874      0.346    0.795     0.965     Negative
Weighted Avg.    0.790    0.221    0.921      0.790    0.835      0.346    0.794     0.906

=== Confusion Matrix ===

  a  b   <-- classified as
  7  2 |  a = Positive
 23 87 |  b = Negative
```

SVM Lexicon 2 - Base

```
=== Summary ===

Correctly Classified Instances         101               84.8739 %
Incorrectly Classified Instances        18               15.1261 %
Kappa statistic                          0.3689
Mean absolute error                      0.1513
Root mean squared error                  0.3889
Relative absolute error                 62.2046 %
Root relative squared error            133.6064 %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.855    0.222    0.979      0.855    0.913      0.423    0.777     0.961     Negative
                 0.778    0.145    0.304      0.778    0.438      0.423    0.817     0.253     Positive
Weighted Avg.    0.849    0.216    0.928      0.849    0.877      0.423    0.780     0.908

=== Confusion Matrix ===

  a  b   <-- classified as
 94 16 |  a = Negative
  2  7 |  b = Positive
```

SVM Lexicon 2 - Cost Sensitive Evaluator

```
=== Summary ===

Correctly Classified Instances         117               98.3193 %
Incorrectly Classified Instances         2                1.6807 %
Kappa statistic                          0.8661
Mean absolute error                      0.0168
Root mean squared error                  0.1296
Relative absolute error                  6.9116 %
Root relative squared error             44.5355 %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.778    0.000    1.000      0.778    0.875      0.874    0.889     0.794     Positive
                 1.000    0.222    0.982      1.000    0.991      0.874    0.850     0.973     Negative
Weighted Avg.    0.983    0.205    0.983      0.983    0.982      0.874    0.853     0.960

=== Confusion Matrix ===

   a   b   <-- classified as
   7   2 |   a = Positive
   0 110 |   b = Negative
```

SVM Lexicon 3 - Base

```
=== Summary ===

Correctly Classified Instances           9                7.563  %
Incorrectly Classified Instances       110               92.437  %
Kappa statistic                          0
Mean absolute error                      0.9244
Root mean squared error                  0.9614
Relative absolute error                102.4302 %
Root relative squared error            102.6496 %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.076      1.000    0.141      ?        0.500     0.075     Positive
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.917     Negative
Weighted Avg.    0.076    0.076    ?          0.076    ?          ?        0.500     0.853

=== Confusion Matrix ===

   a   b   <-- classified as
   9   0 |   a = Positive
 110   0 |   b = Negative
```

SVM Lexicon 3 - Cost Sensitive Evaluator No False Negative Weights

```
=== Summary ===

Correctly Classified Instances         110               92.437  %
Incorrectly Classified Instances         9                7.563  %
Kappa statistic                          0
Mean absolute error                      0.0756
Root mean squared error                  0.275
Relative absolute error                  8.3807 %
Root relative squared error             29.3618 %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.075     Positive
                 1.000    1.000    0.924      1.000    0.961      ?        0.500     0.917     Negative
Weighted Avg.    0.924    0.924    ?          0.924    ?          ?        0.500     0.853

=== Confusion Matrix ===

   a   b   <-- classified as
   0   9 |   a = Positive
   0 110 |   b = Negative
```

SVM Lexicon 3 - Cost Sensitive Evaluator

```
=== Summary ===

Correctly Classified Instances          14               11.7647 %
Incorrectly Classified Instances       105               88.2353 %
Kappa statistic                          0.0072
Mean absolute error                      0.8824
Root mean squared error                  0.9393
Relative absolute error                 97.7743 %
Root relative squared error            100.2895 %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.955    0.079      1.000    0.146      0.060    0.523     0.078     Positive
                 0.045    0.000    1.000      0.045    0.087      0.060    0.523     0.920     Negative
Weighted Avg.    0.118    0.072    0.930      0.118    0.091      0.060    0.523     0.857

=== Confusion Matrix ===

   a   b   <-- classified as
   9   0 |   a = Positive
 105   5 |   b = Negative
```

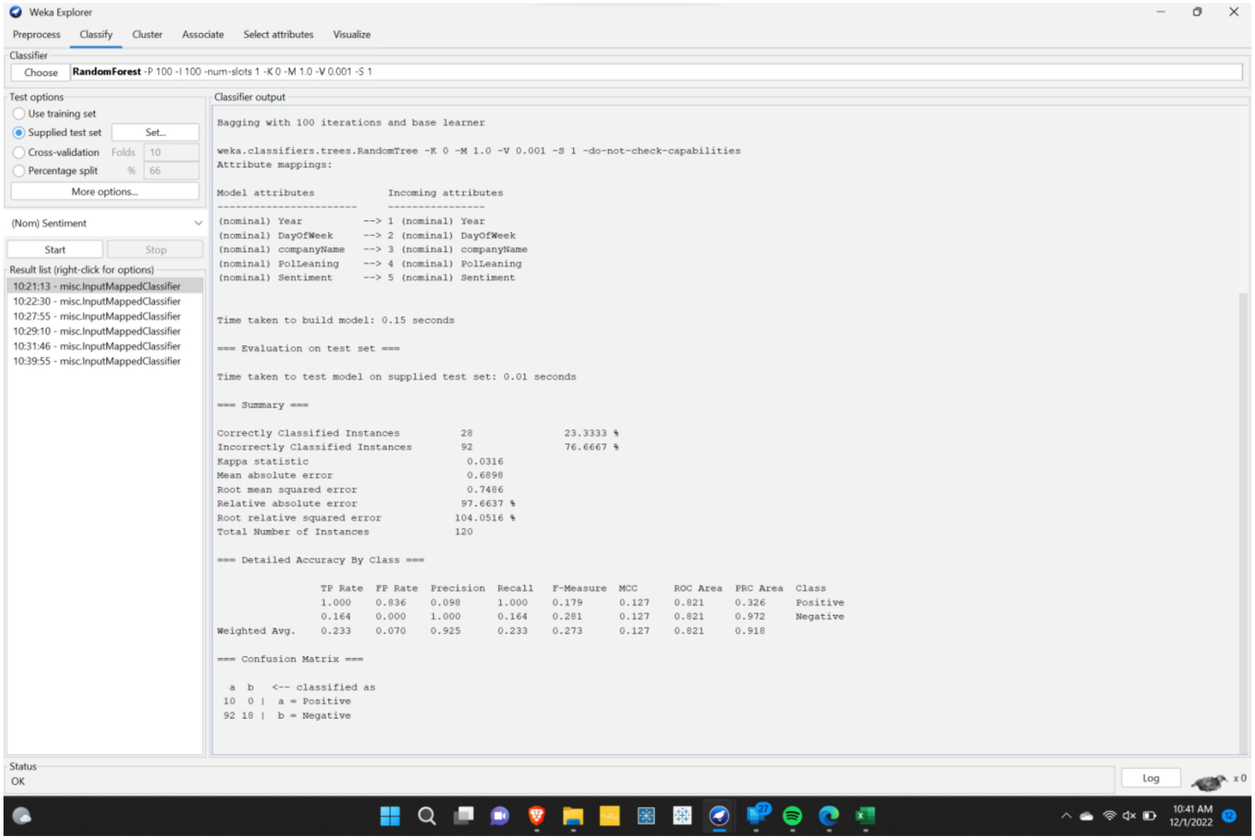
SVM Lexicon Combined - Base

```
=== Summary ===

Correctly Classified Instances           9                7.563  %
Incorrectly Classified Instances       110               92.437  %
Kappa statistic                          0
Mean absolute error                      0.9244
Root mean squared error                  0.9614
Relative absolute error                102.9961 %
Root relative squared error            103.2673 %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.076      1.000    0.141      ?        0.500     0.075     Positive
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.917     Negative
Weighted Avg.    0.076    0.076    ?          0.076    ?          ?        0.500     0.853

=== Confusion Matrix ===

   a   b   <-- classified as
   9   0 |   a = Positive
 110   0 |   b = Negative
```

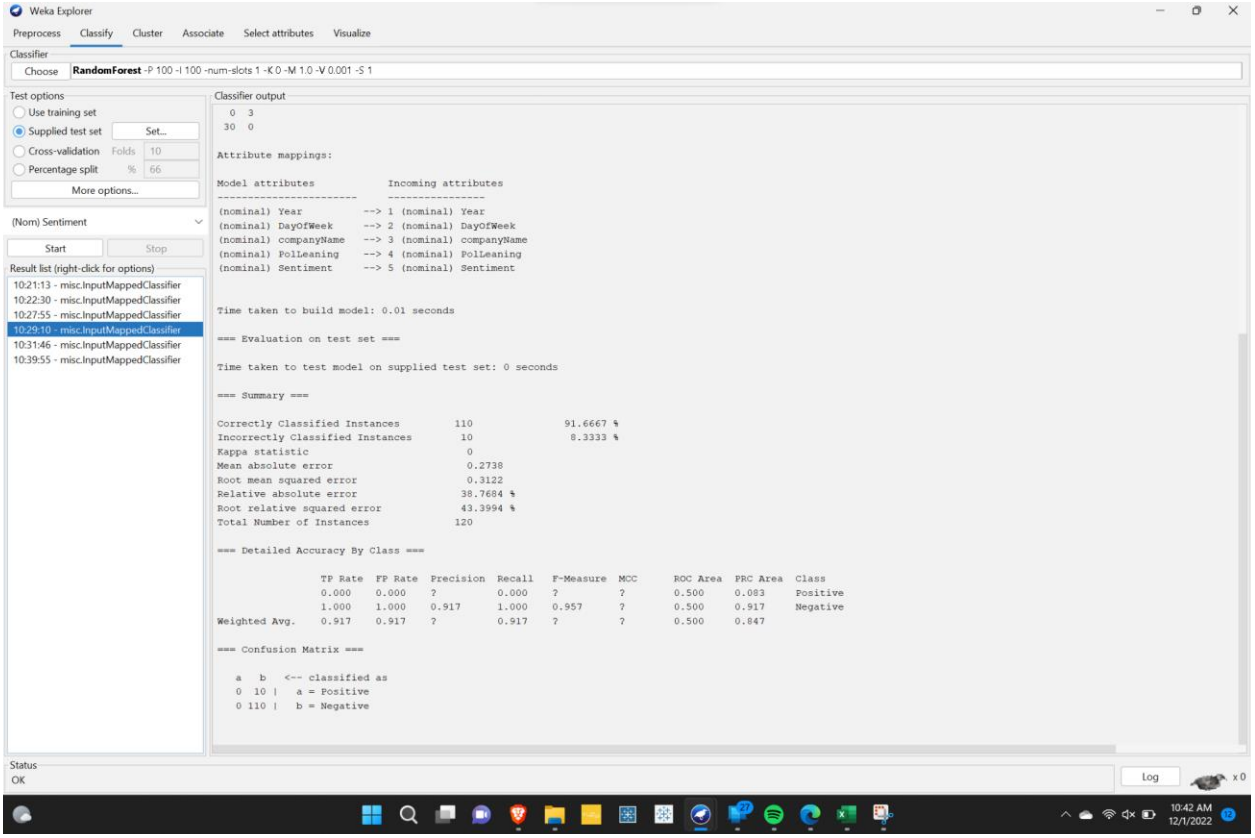
SVM Lexicon Combined - Cost Sensitive Evaluator

```
=== Summary ===

Correctly Classified Instances          14               11.7647 %
Incorrectly Classified Instances       105               88.2353 %
Kappa statistic                          0.0072
Mean absolute error                      0.8824
Root mean squared error                  0.9393
Relative absolute error                 98.3145 %
Root relative squared error            100.893  %
Total Number of Instances              119
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.955    0.079      1.000    0.146      0.060    0.523     0.078     Positive
                 0.045    0.000    1.000      0.045    0.087      0.060    0.523     0.920     Negative
Weighted Avg.    0.118    0.072    0.930      0.118    0.091      0.060    0.523     0.857

=== Confusion Matrix ===

   a   b   <-- classified as
   9   0 |   a = Positive
 105   5 |   b = Negative
```

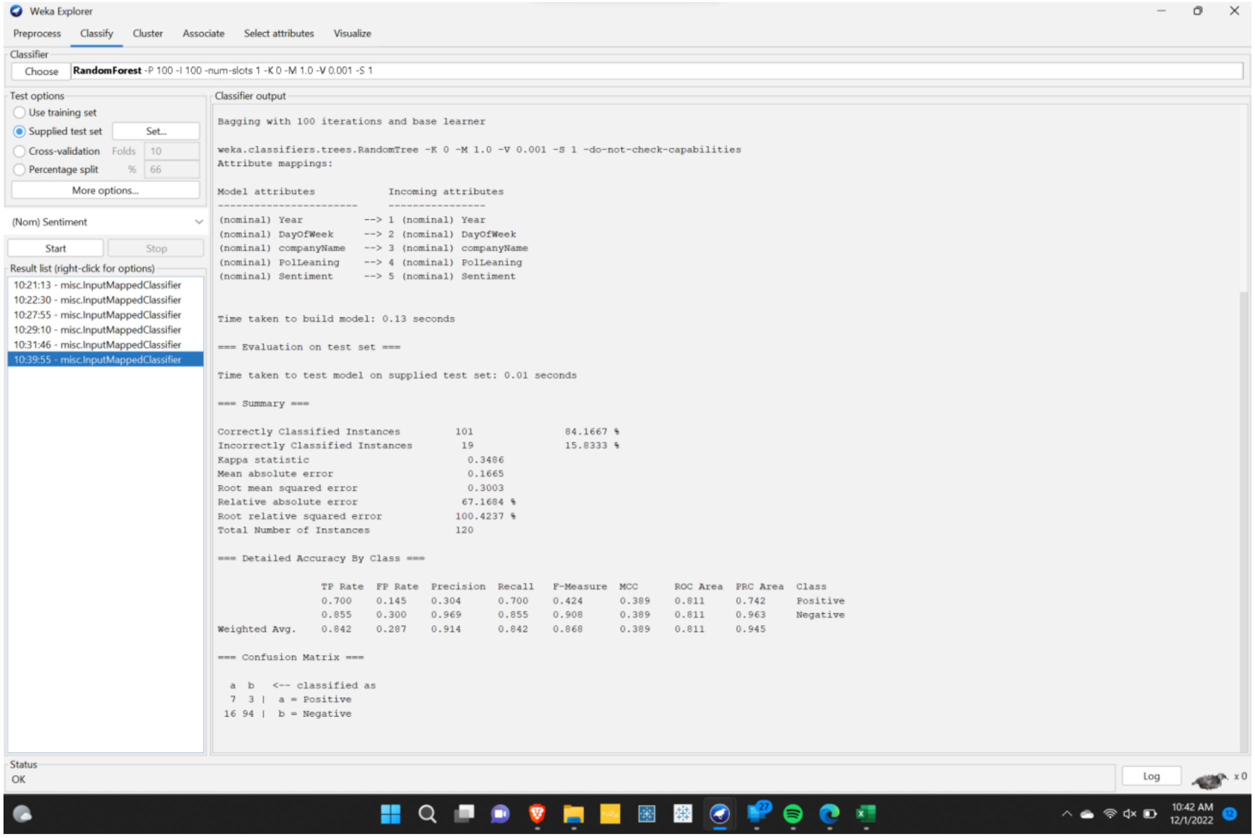
Random Forest Lexicon 1 - Base



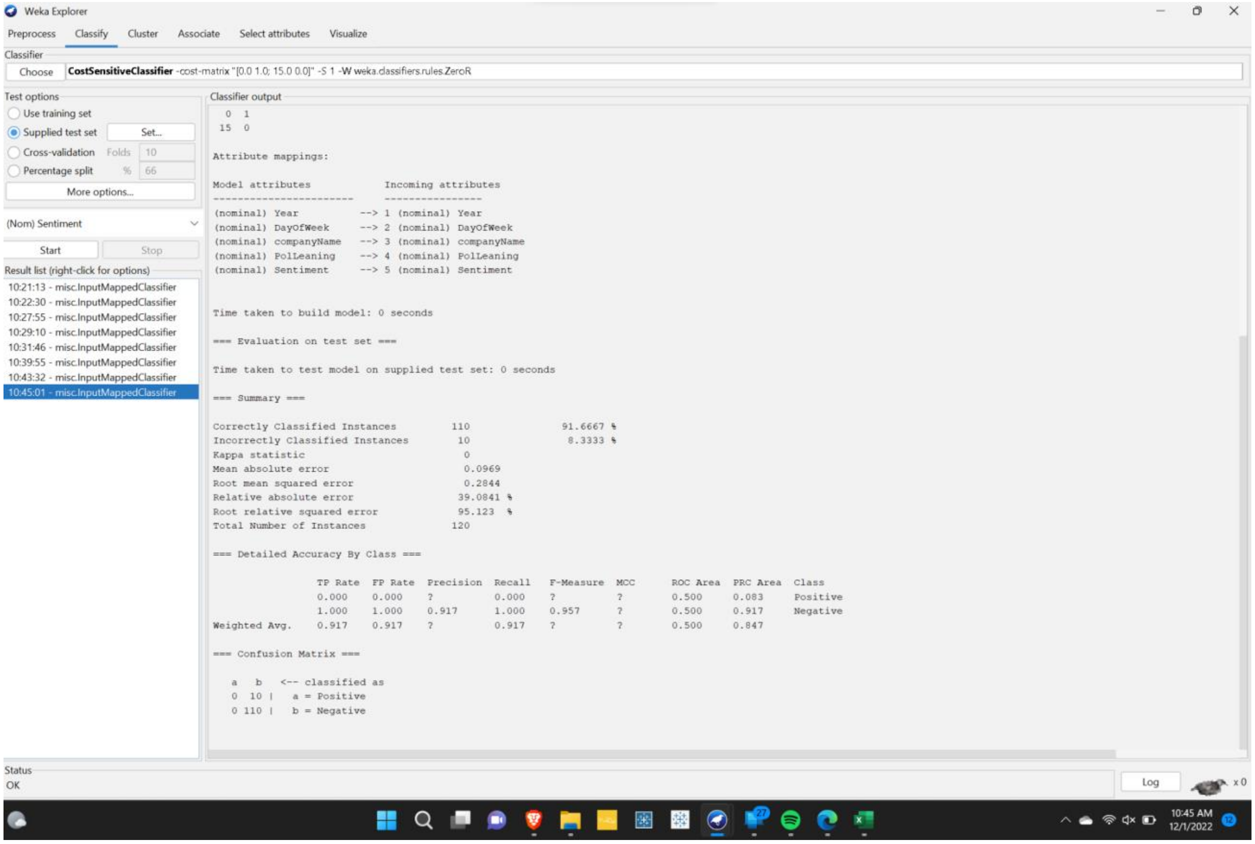
Random Forest Lexicon 1 - Cost Sensitive Evaluator

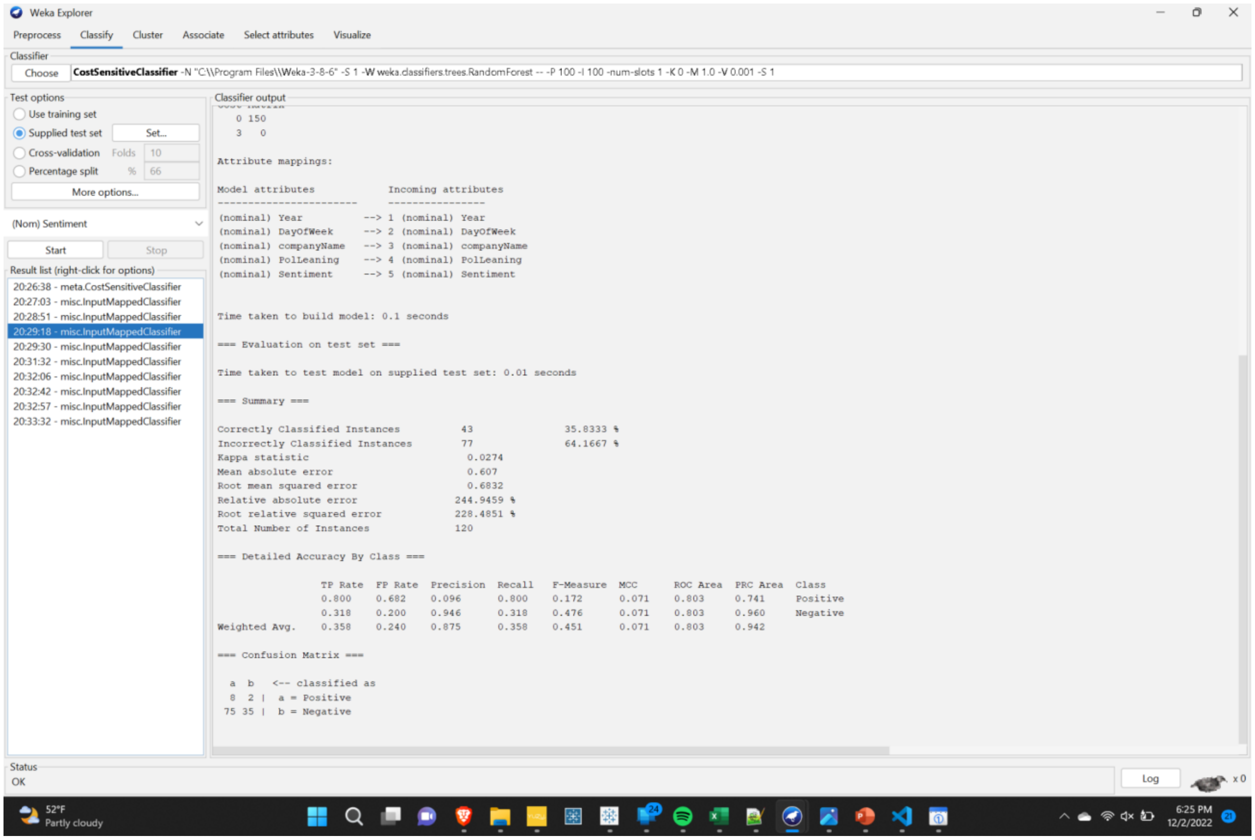


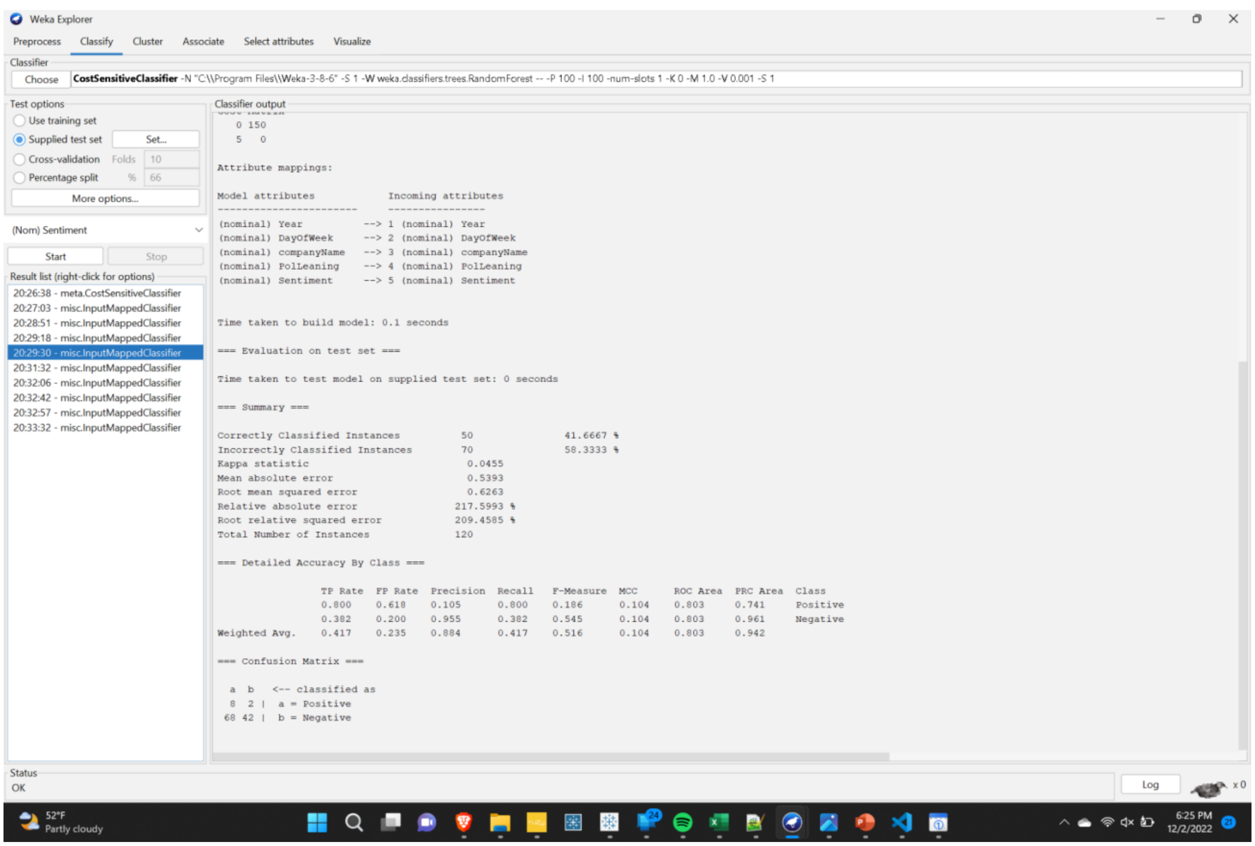
Random Forest Lexicon 2 - Base



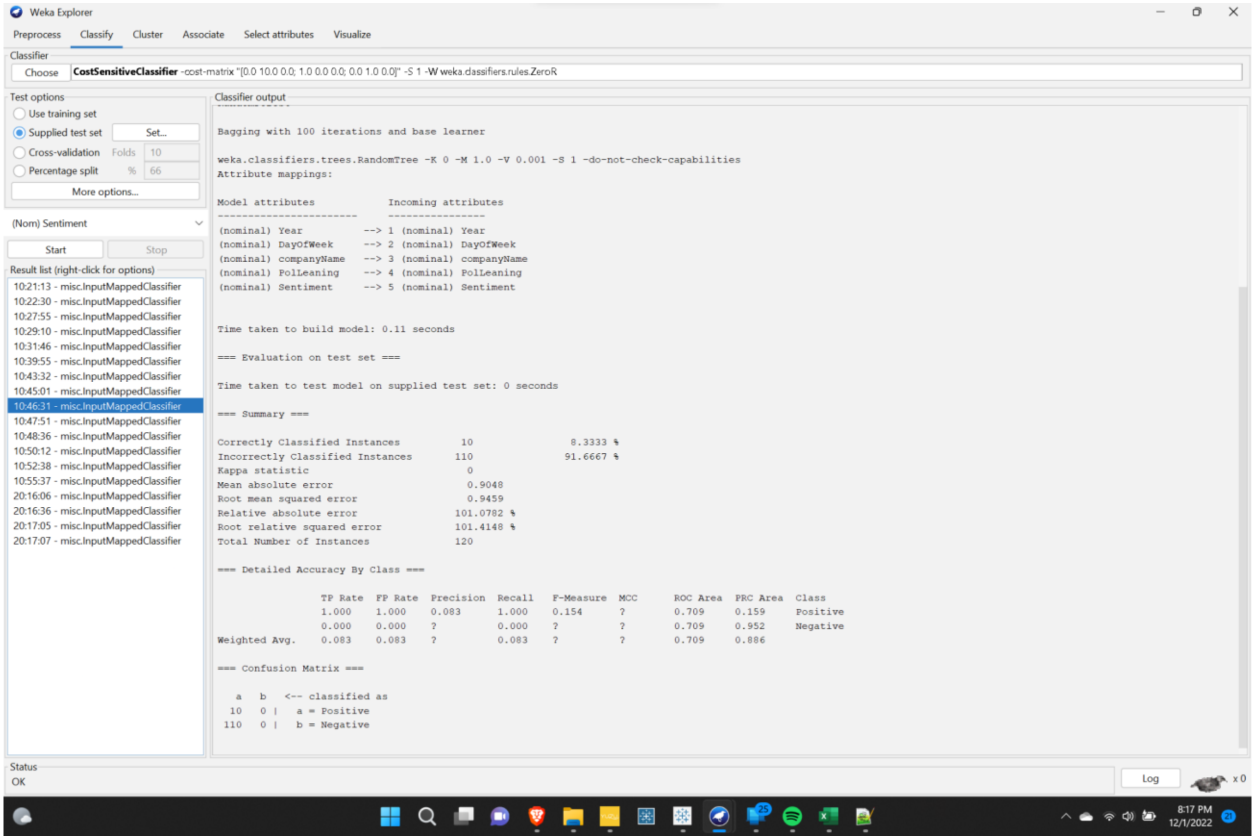
Random Forest Lexicon 2 - Cost Sensitive Evaluator



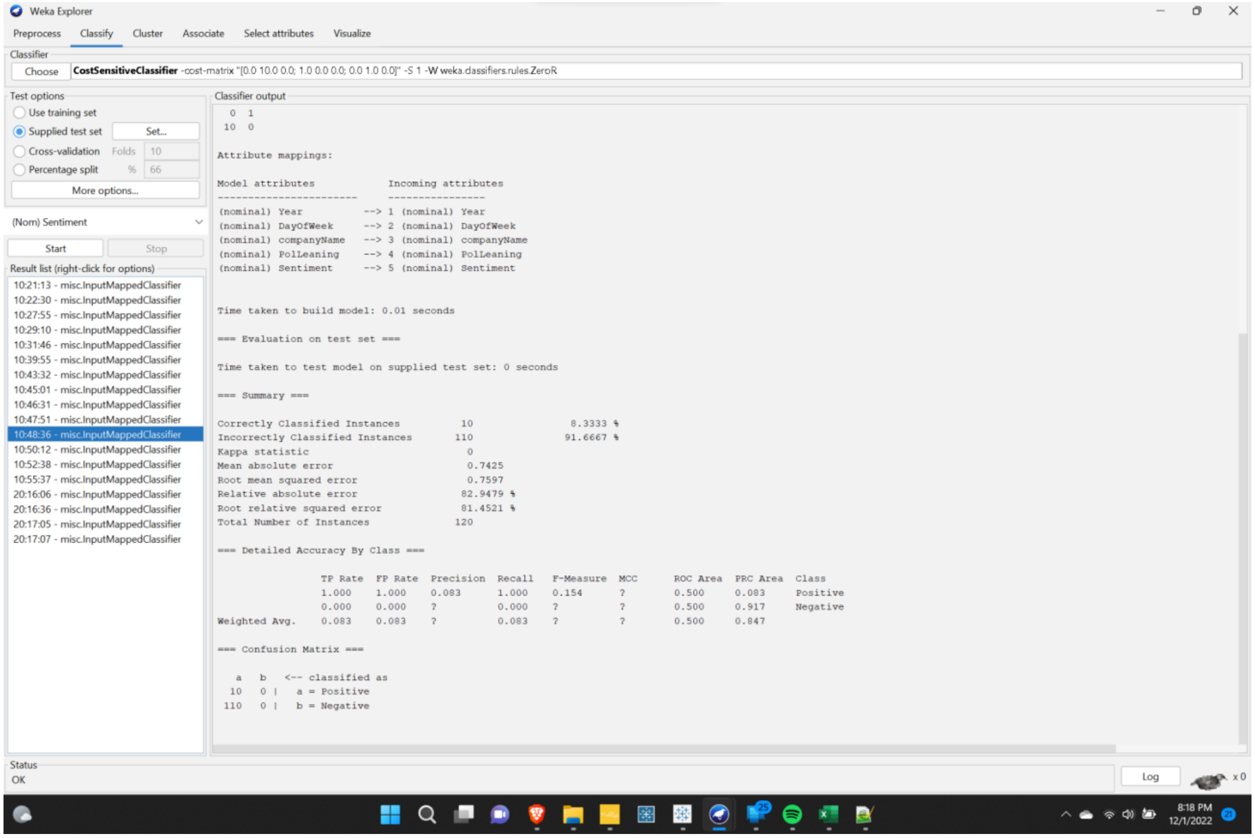

Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier
Choose CostSensitiveClassifier -N "C:\\Program Files\\Weka-3-8-6" -S 1 -W weka.classifiers.trees.RandomForest -- -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Test options
Use training set
Supplied test set Set...
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) Sentiment
Start Stop
Result list (right-click for options)
20:26:38 - meta.CostSensitiveClassifier
20:27:03 - misc.InputMappedClassifier
20:28:51 - misc.InputMappedClassifier
20:29:18 - misc.InputMappedClassifier
20:29:30 - misc.InputMappedClassifier
20:31:32 - misc.InputMappedClassifier
20:32:06 - misc.InputMappedClassifier
20:32:42 - misc.InputMappedClassifier
20:32:57 - misc.InputMappedClassifier
20:33:32 - misc.InputMappedClassifier
Classifier output
0 150
3 0
Attribute mappings:
Model attributes Incoming attributes
(nominal) Year --> 1 (nominal) Year
(nominal) DayOfWeek --> 2 (nominal) DayOfWeek
(nominal) companyName --> 3 (nominal) companyName
(nominal) PolLeaning --> 4 (nominal) PolLeaning
(nominal) Sentiment --> 5 (nominal) Sentiment
Time taken to build model: 0.1 seconds
=== Evaluation on test set ===
Time taken to test model on supplied test set: 0.01 seconds
=== Summary ===
Correctly Classified Instances 43 35.8333 %
Incorrectly Classified Instances 77 64.1667 %
Kappa statistic 0.0274
Mean absolute error 0.607
Root mean squared error 0.6832
Relative absolute error 244.9459 %
Root relative squared error 228.4851 %
Total Number of Instances 120
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.800 0.682 0.096 0.800 0.172 0.071 0.803 0.741 Positive
0.318 0.200 0.946 0.318 0.476 0.071 0.803 0.960 Negative
Weighted Avg. 0.358 0.240 0.875 0.358 0.451 0.071 0.803 0.942
=== Confusion Matrix ===
a b <-- classified as
8 2 | a = Positive
75 35 | b = Negative
Status
OK
Log
x 0
52°F
Partly cloudy
6:25 PM
12/2/2022


Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier
Choose CostSensitiveClassifier -N "C:\\Program Files\\Weka-3-8-6" -S 1 -W weka.classifiers.trees.RandomForest -- -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Test options
Use training set
Supplied test set Set...
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) Sentiment
Start Stop
Result list (right-click for options)
20:26:38 - meta.CostSensitiveClassifier
20:27:03 - misc.InputMappedClassifier
20:28:51 - misc.InputMappedClassifier
20:29:18 - misc.InputMappedClassifier
20:29:30 - misc.InputMappedClassifier
20:31:32 - misc.InputMappedClassifier
20:32:06 - misc.InputMappedClassifier
20:32:42 - misc.InputMappedClassifier
20:32:57 - misc.InputMappedClassifier
20:33:32 - misc.InputMappedClassifier
Classifier output
0 150
5 0
Attribute mappings:
Model attributes Incoming attributes
(nominal) Year --> 1 (nominal) Year
(nominal) DayOfWeek --> 2 (nominal) DayOfWeek
(nominal) companyName --> 3 (nominal) companyName
(nominal) PolLeaning --> 4 (nominal) PolLeaning
(nominal) Sentiment --> 5 (nominal) Sentiment
Time taken to build model: 0.1 seconds
=== Evaluation on test set ===
Time taken to test model on supplied test set: 0 seconds
=== Summary ===
Correctly Classified Instances 50 41.6667 %
Incorrectly Classified Instances 70 58.3333 %
Kappa statistic 0.0455
Mean absolute error 0.5393
Root mean squared error 0.6263
Relative absolute error 217.5993 %
Root relative squared error 209.4585 %
Total Number of Instances 120
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.800 0.618 0.105 0.800 0.186 0.104 0.803 0.741 Positive
0.382 0.200 0.955 0.382 0.545 0.104 0.803 0.961 Negative
Weighted Avg. 0.417 0.235 0.884 0.417 0.516 0.104 0.803 0.942
=== Confusion Matrix ===
a b <-- classified as
8 2 | a = Positive
68 42 | b = Negative
Status
OK
Log
x 0
52°F
Partly cloudy
6:25 PM
12/2/2022

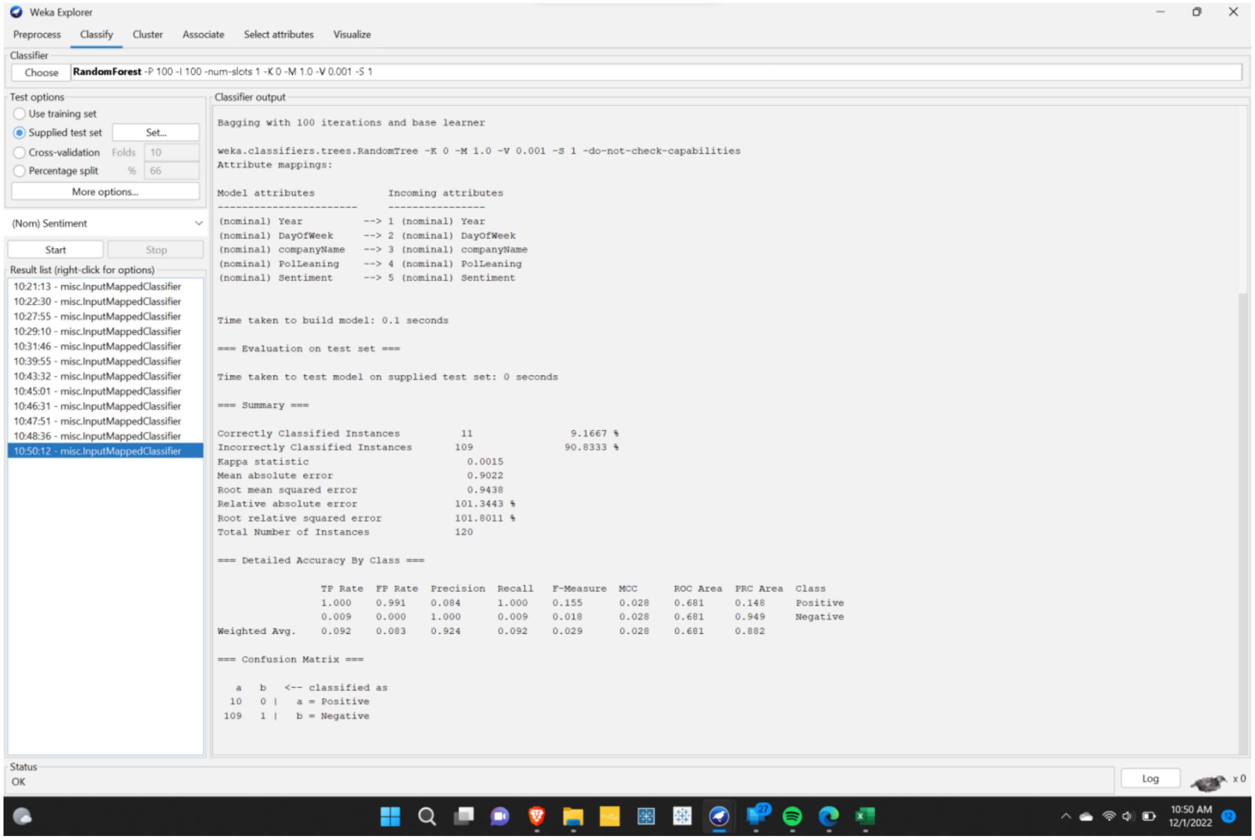
Random Forest Lexicon 3 - Base



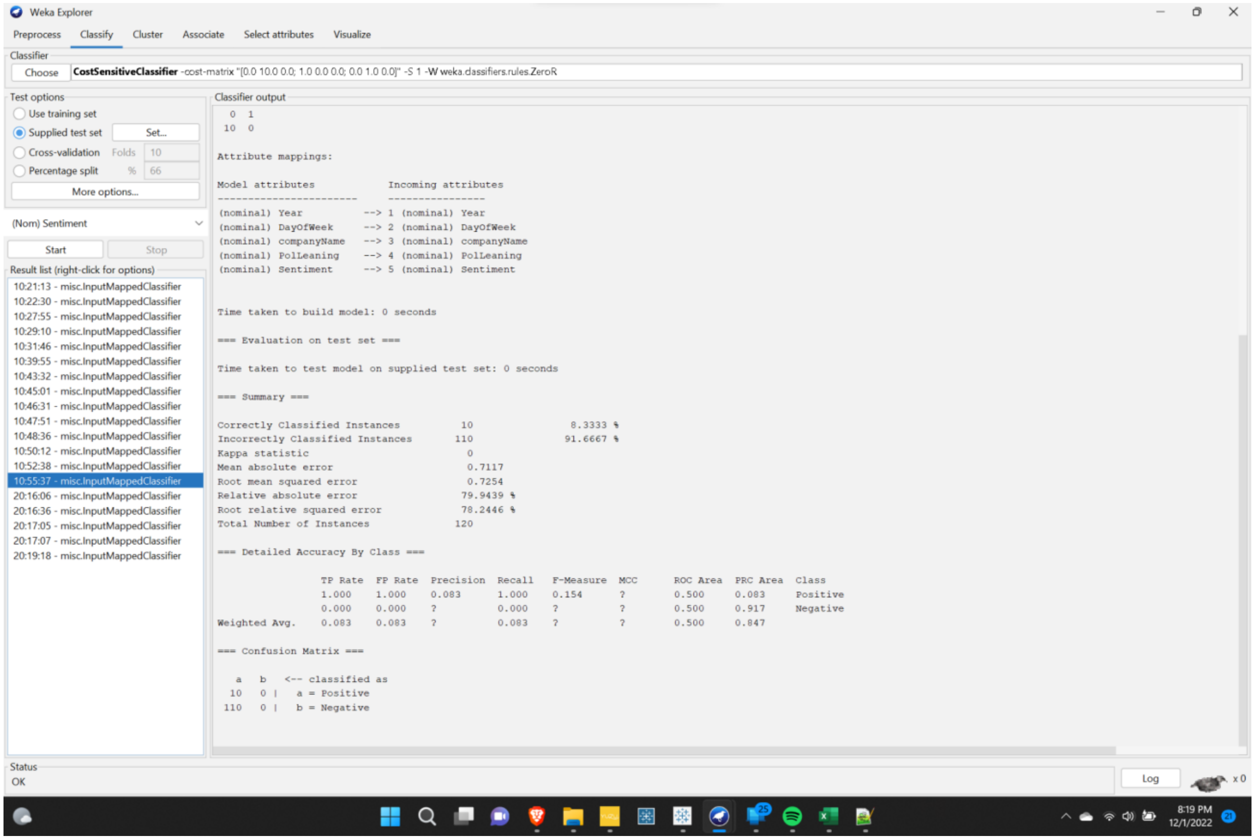
Random Forest Lexicon 3 - Cost Sensitive Evaluator



Random Forest Lexicon Combined - Base



Random Forest Lexicon Combined - Cost Sensitive Evaluator



References

[What is Random Forest? | IBM]

[Support Vector Regression (SVR) — One of the Most Flexible Yet Robust Prediction Algorithms| by Saul Dobilas | Towards Data Science]

[Random forest - Wikipedia]

[Bootstrap aggregating - Wikipedia]

[MPQA University of Pittsburgh Lexicon]

[Bing Liu and Minqing Hu Sentiment Lexicon]

[SenticNet Lexicon]

More visualizations can be found in the attached Viz.pptx file