Sentiment Analysis of Movie Review Using Machine Learning

Nida Fatima

July 12, 2022

Sentiment Analysis of Movie Review using Machine Learning

Nida Fatima
Dep. Of Computer Science
University of Sialkot
Sialkot, Pakistan
21101009-027@uskt.edu.pk

Abstract

A significant amount of data is present on social media and other online platforms in the form of tweets, blogs, status updates, postings, etc. Sentiment analysis of the data is useful to express the view of the group or of any individual. Sentiment analysis is the examination of feelings and viewpoints in any kind of literature. Additionally, perform classifications on weka, movie reviews about the same movies were compared. The major objective was to learn more about market research analysis of traditional and social media sources. The resulting study aids moviegoers and the film business in understanding how reviews are perceived.

Keywords: Sentiment Analysis, Machine Learning, Entertainment, Movie Reviews

1. Introduction

Due to two factors, sentiment analysis is one of the most rigorously expanding study fields. First of all, there are many situations in real life where having an opinion is crucial to carrying out any task. Considering other people's perspectives when making decisions is advantageous and essential to one's behaviour. Second, there are many difficult obstacles in research that have not been encountered before the year 2000. The lack of prior studies has been the primary factor in the limited usage of digital text for opinion mining. Therefore, it should come as no surprise that the sector's emergence and the growth of social media coincide with the Internet's explosive growth, where sentiment analysis finds a place in the majority of real time applications [1].

It is impossible to exaggerate the value of entertainment in our lives. It brings people together and is a great way for the whole family to be involved in the tie-in. If we don't have entertainment in our lives, we have nothing to eat or drink. The majority of people in our country enjoy watching movies and TV shows as soon as they are released, while a small percentage wait to see if the film in question has received positive reviews before watching it. One may argue that the success of a certain movie depends on a detailed review of it [1].

We analyse the sentiment expression to categorise the polarity of the movie review on a scale from negatively as disliked to positively as liked. We then perform feature extraction and ranking, using these features to train our classifier, and use the results to categorise the movie review into its appropriate label.

Then With the help of provided training data, a number of multi-class classification algorithms have been compared to determine which is the most effective. The same training set cannot be used to access algorithm accuracy since the model can over fit the training data.

2. Methodology

2.1 Data Collection

In this paper, we used Turkish movie reviews dataset which is publically available from the website Kaggle.com. The dataset contains 105 movie reviews in Turkish with three classes. Classes Turkish to English.

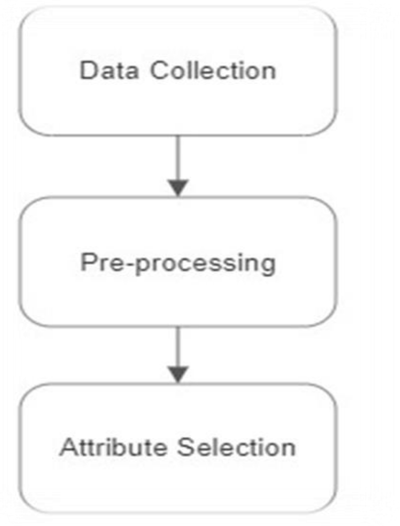
1) Negatif -> Negative
2) Pozitif -> Positive
3) Tarafsiz -> Neutral

2.2 Pre-processing

We perform pre-processing steps on dataset:

1. In 1st step we use Stratified Remove Fold which outputs a specific fold for cross validation from a dataset.
2. In 2nd step we use Remove Percentage which removes the percentage from the dataset.

3. Knowledge Flow



4. Results and Discussion

We used **Weka** tool version **3.8.6** to perform the experiment on Turkish movie reviews dataset.

4.1 Without Text Pre-processing

The dataset in arff format are loaded in weka. Select the predicted class as class attribute. Then run the selected classifiers (Bagging, Classification via Regression, Random Tree, OneR and KStar) that are mentioned above. Bagging classifier 34% correctly classified the instances, CVR 47%, RT 34%, OneR 34%, and KS 100% correctly classified the instances. The below figures presented the experimental results on this dataset.

```
Classifier output
Size of the tree : 119

Time taken to build model: 0.04 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          36               34.2857 %
Incorrectly Classified Instances        69               65.7143 %
Kappa statistic                          0.0143
Mean absolute error                      0.4381
Root mean squared error                  0.6619
Relative absolute error                 98.4752 %
Root relative squared error            140.2632 %
Total Number of Instances              105
```

Figure 3: Experimental Result for Random Tree

```
Bagging with 10 iterations and base learner

weka.classifiers.trees.REPTree -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0

Time taken to build model: 1.49 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances          18               34.6154 %
Incorrectly Classified Instances        34               65.3846 %
Kappa statistic                          0.0397
Mean absolute error                      0.4351
Root mean squared error                  0.48
Relative absolute error                 97.2535 %
Root relative squared error            100.8585 %
Total Number of Instances               52
```

Figure 1: Experimental Result for Bagging

```
Time taken to build model: 0.13 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          36               34.2857 %
Incorrectly Classified Instances        69               65.7143 %
Kappa statistic                          0.0143
Mean absolute error                      0.4381
Root mean squared error                  0.6619
Relative absolute error                 98.4349 %
```

Figure 4: Experimental Result for OneR

```
Time taken to build model: 1.84 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          50               47.619  %
Incorrectly Classified Instances        55               52.381  %
Kappa statistic                          0.2143
Mean absolute error                      0.4028
Root mean squared error                  0.4967
Relative absolute error                 90.5403 %
Root relative squared error            105.2675 %
Total Number of Instances              105
```

Figure 2: Experimental Result for Classification via Regression

```
Time taken to test model on training data: 11.9 seconds

=== Summary ===

Correctly Classified Instances         105              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
```

Figure 5: Experimental Result for KStar

3.2 With Text Pre-processing

We performed the supervised Stratified Remove Fold (outputs specified fold for cross validation) and ReSample, unsupervised Remove percentage filter (remove percentage from a given dataset), ReservoirSample filter. The below figures presented the results with preprocessing. Baggong predict 27% accurate result, Random Tree 61%, Classification via Regression 9%, OneR 37% and KStar predicts 90% accurate result. It is notable that the KStar classifier perform best with 90% accurate results among all the classifiers.

```
Correctly Classified Instances          3               27.2727 %
Incorrectly Classified Instances        8               72.7273 %
Kappa statistic                        -0.1139
Mean absolute error                     0.491
Root mean squared error                 0.5333
Relative absolute error               103.9369 %
Root relative squared error           106.1707 %
Total Number of Instances              11
```

Figure 1: Experimental Result for Bagging

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         32               61.5385 %
Incorrectly Classified Instances       20               38.4615 %
Kappa statistic                         0.0988
Mean absolute error                     0.2564
Root mean squared error                 0.5064
Relative absolute error                84.4973 %
Root relative squared error           131.6328 %
Total Number of Instances              52
```

Figure 2: Experimental Result for Random Tree

```
Time taken to build model: 0.78 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          1                9.0909 %
Incorrectly Classified Instances       10               90.9091 %
Kappa statistic                        -0.4286
Mean absolute error                     0.5558
Root mean squared error                 0.7002
Relative absolute error               117.6579 %
Root relative squared error           139.378  %
Total Number of Instances              11
```

Figure 3: Experimental Result for Classification via Regression

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         37               37      %
Incorrectly Classified Instances       63               63      %
Kappa statistic                         0.0555
Mean absolute error                     0.42
Root mean squared error                 0.6481
Relative absolute error                94.4086 %
```

Figure 4: Experimental Result for OneR

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         10               90.9091 %
Incorrectly Classified Instances        1                9.0909 %
Kappa statistic                         0.8608
Mean absolute error                     0.1212
Root mean squared error                 0.2462
Relative absolute error                25.6579 %
Root relative squared error            49.0072 %
Total Number of Instances              11
```

Figure 5: Experimental Result for KStar

3.3. With Default Selection

The default selected classifiers outputs are: Bagging classifier predict 35% correct results and 64% uncorrect results. Next, RT classifier predict 34% correct results and 65% uncorrect results.Next one, CvR classifier predict 47% correct results and 52% uncorrect results. OneR classifier predict 35% correct and 64% uncorrect results, and KStar classified 33% correct and 66% uncorrect results.

3.4.With Attribute selection

We performed experiments with attribute selection the results of classifiers are: Bagging 35%, RT 34%, CvR 9%, OneR 35% and KStar predict result of 33%. It is noticeable that OneR and Bagging performs best result with attribute selection.

3.5. With Execution Time

The total execution time to build trained model using the above mentioned five ML classifiers are: Bagging (1.49, 0.38, 1.11, 1.05) seconds, Random Tree (0.04, 0, 0.02, 0.01) seconds, CvR (1.84, 0.78, 1.31, 0.91) seconds, OneR (0.13, 0.11, 0.05, 0.03) seconds, and KStar (11.9, 0, 0, 0,) seconds respectively. OneR classifier substantially take less time (as compared to other four classifiers) to train the model.

4. Conclusion

This paper has presented analysis on numeric data of movie reviews. Five classifiers; bagging, Random Tree, Classification via Regression, OneR and KStar are applied on dataset collected from social media reviews. First, reviews are collected. Second, we have proposed a model consisting of various tasks such as pre-processing, and classification. Classification is carried out by five different classifiers and it is concluded that Random Tree is a promising classifier than others comparatively. Furthermore, we have also computed the feature selection that improve the result of OneR classifier. Moreover, we calculate the execution time for the five classifiers. We are looking forward to using deep learning models to by changes dataset from numeric to textual data in the future, and we intend to increase the dataset size with a balanced distribution.

References

[1] S. S. A. V. N. T. S. R. N. H. SWATHI, “Sentiment Analysis of Movie Review using data Analytics Techniques,” *IRE,* vol. 2, no. 9, p. 5, 2019.

[2] P. T. S. K. V. K. Abdul Aleem, “Entertainment Advisor Using Sentiment Analysis,” *Scientific Research,* vol. 66, no. 2, p. 7, 2022.