Performance comparison of classifiers on twitter sentimental analysis

Shruti Wadhwa[1], Karuna Babber[2,]

[1]*Chief Operating Officer, Nidus Technologies Pvt. Ltd., Chandigarh, India*

[2]*Assistant Professor, Panjab University, Chandigarh, India*

Abstract

Twitter sentimental analysis is the way to examine polarity in tweeted opinions. The computational process involves implementing machine learning classifiers to categorize the tweets into positive, negative and neutral sentiments. To identify a suitable classifier for the task is a prime issue. In this paper we have presented the performance comparison of base classification techniques like Decision Tree, Random Forest, Naive Bayes, K-Nearest Neighbour and Logistic Regression on analysis of tweets. The results thus obtained show Logistic Regression analyze tweets with highest accuracy rate of 86.51% and the least performer comes out to be K-Nearest Neighbour with an average accuracy rate of 50.40%.

Keywords: Twitter, sentimental analysis, machine learning, classifiers and algorithms

1. Introduction

Twitter with more than 321 million users across the globe contributes to a daily average of 500 million unstructured social media data [1]. The textual data is one form of unstructured data. The people can post, read, update the short text messages called ‘tweets’ on this platform. Through tweets users can express their views, share opinions about a particular topic. The Sentimental Analysis (SA) is the way to categorize the polarity of a text message “tweet” in this case. This technique is being used in varied fields like politics, e-commerce, entertainment or public sectors. Many e-commerce companies are using SA to monitor customer/consumer opinions and to further recommend customers the best product based on this analysis. The prime task of twitter SA is to check the mood of users’ opinions that is the user tweet is a positive opinion or a negative one [2]. This task surely has its own challenges like acronyms and abbreviations used in tweets make it difficult to understand its mood, secondly many tweets contain informal language and show limited indication about the various and differing sentiments.

The base classifiers like Naive Bayes, Logistic Regression, K-Nearest Neighbour, Decision Tree and Random Forest can be used for Twitter SA. Since all the classifiers are based on different techniques the result of analysis of tweets is likely to vary. This paper presents the performance comparison of the basic classifiers on Twitter SA. The related work is presented in section 2 and in section 3 the data description with visualizations of data analysis are provided. The comparison and result analysis is carried out in section 4. The section 5 finally concludes the paper with the directions of future work.

2. Related work

The sentimental analysis is bracketed under Natural Language Processing task. Initially a document level classification [3] was done, the work was further extended to sentence level [4] and more recently SA is performed at phrase level [5, 6]. The others [7, 8] use positive emoticons like "☺" and negative emoticons like "☹" to segregate tweets. For feature extraction unigram, bigram and n-gram along with Parts-of-speech (PoS) are used by some researchers [9, 10, 11]. The carefully chosen linguistic features can contribute to classifier accuracy [12, 13]. A survey on SA algorithms and applications [14] provides an insight. The researchers [15] determined sentiments with hashtags and emoticons. The PoS and lexicons have been used as linguistic resources [16, 17]. In [18] an efficient ensemble classifier is used for SA. The hashing and Bag-of-words (BoW) are used for feature representation in SA [19]. The ensemble classifier based on 'Majority vote' is provided in [20]. The hashing feature is used with logistic regression base learner technique in SA [21]. The authors [22] used n-grams, sentiwordnet, PoS and lexicons as feature set for SA. The work in [23] clarified that the sentiment of a specific phrase may differ from the sentiment of whole tweet. The authors [24] developed an ensemble technique with bootstrap aggregation, specific feature set and datasets for twitter SA. The more accurate classifiers for SA are discussed in [25]. The literature work indicates careful extraction of features along with appropriate selection of classifiers for an accurate SA.

3. Data Description

Twitter provides microblogging services that allow users to post short real time messages (restricted up to 140 characters in length) known as 'tweets'. As a result users here use emoticons, acronyms (like gr8t - great, lol – loads of laughter, bff – best friend forever), miss-spell words or use special characters to express special meanings. A brief description of tweets is given below:

i) Emoticons: These represent facial expressions pictorially represented by punctuation letters or otherwise to express the positive or negative mood of user like: "☺" and "☹".
ii) Hashtags: To increase the visibility and highlight the topic of their comment generally users use hashtags.
iii) Special Character: The users type "@" to refer their tweet to other users. The other special character like "#" is used to express special meaning.

3.1 **Data pre-processing:** We acquire 18,000 tweets from the site by streaming process. No location, language or other restriction was imposed to fetch these tweets. The data pre-processing is done to decrease the size of the feature set and to make it suitable for classification purposes. The following steps are followed for pre-processing the tweets.

i) Emoticons are replaced with meaningful sentimental text.
ii) Punctuation symbols are removed from the tweets.
iii) Stop words are removed from the tweets.

iv) "Stemming" is performed to de-value the word to its root word.
v) "Slangs" are converted to equivalent meaningful words.

This leaves us with 18,000 tweets of 32107 words. In total 15 to 25 percent data is used for testing of SA. Figure 1 shows the pre-processed twitter data. We use base classifiers like Decision Tree, Random Forest, K-Nearest Neighbour (KNN), Logistic Regression, and Naive Bayes to check their performance and accuracy rate on twitter SA.

Figure 1: Twitter pre-processed data

```
In [3]: print('Dataset size:',tweet_df.shape)
        print('Columns are:',tweet_df.columns)

        Dataset size: (18000, 6)
        Columns are: Index(['TweetContent', 'Date', 'User', 'Source', 'T
        weetID', 'Tweet URL'], dtype='object')

In [4]: tweet_df.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 18000 entries, 0 to 17999
        Data columns (total 6 columns):
        TweetContent    18000 non-null object
        Date            18000 non-null object
        User            18000 non-null object
        Source          18000 non-null object
        TweetID         18000 non-null float64
        Tweet URL       18000 non-null object
        dtypes: float64(1), object(5)
        memory usage: 843.8+ KB
```

a. Sentiment Classification using base classifiers

Base classifiers are widely used on sentimental analysis. The detail of these classifiers is provided below:

i. Naive Bayes (NB)

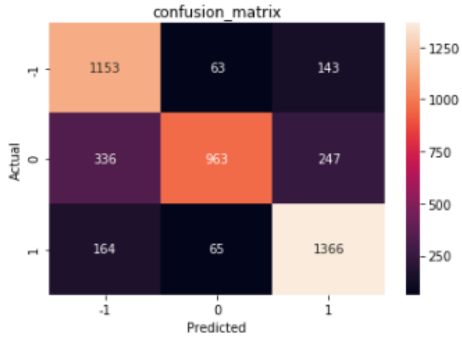
This is a probabilistic classifier and applies 'Bayes' theorem with strong independence assumptions between features [26]. NB computes posterior probability using the below given formula:

$$\text{Posterior probability} = \frac{\text{likelihood X prior probability}}{\text{Evidence}}$$

The confusion matrix of NB is drawn (figure 2) wherein the 25% of twitter data is taken into consideration for testing. The NB performed fairly well with the correctly identified positive tweets 1366 (represented by '1') out of the total 1595 positive tweets. The same pattern followed with correctly identified negative tweets 1153 (represented by '-1') out of the total 1359 negative tweets but the accuracy rate drops little on neutral tweets with the correctly identified neutral tweets 963 (represented by '0') out of the total 1546 neutral tweets.

Figure 2: Confusion matrix of Naive Bayes algorithm on twitter SA

```
[[1153   63  143]
 [ 336  963  247]
 [ 164   65 1366]]
```



The figure 3 presents the weighted average report using Precision, Recall and F1 score rate of NB classifier. Although the F1 score of neutral tweets dipped little in comparison to positive and negative tweets but the overall weighted F1 average score of 77% indicates the above average performance of NB on twitter SA.

Figure 3: Naive Bayes weighted average report

```
In [42]: print(classification_report(y_test, y_pred))
              precision    recall  f1-score   support

          -1       0.70      0.85      0.77      1359
           0       0.88      0.62      0.73      1546
           1       0.78      0.86      0.82      1595

   micro avg       0.77      0.77      0.77      4500
   macro avg       0.79      0.78      0.77      4500
weighted avg       0.79      0.77      0.77      4500
```

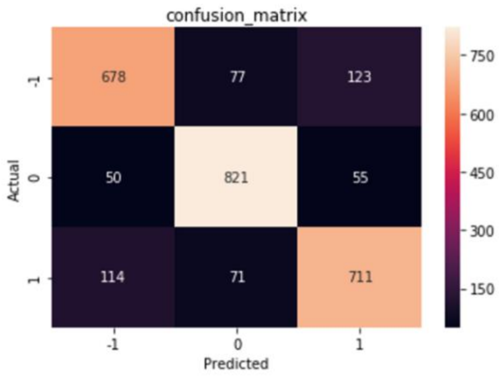
ii. Decision Tree (DT)

This algorithm can be used both for classification and regression. Based on if-then-else construction such tree based algorithms provide high accuracy and stability especially in supervised learning methods [27].

The DT classifier is used to classify the 15% of 18,000 tweets for SA (figure 4). This classifier shows greater accuracy of almost 88% with 821 correctly identified neutral tweets (represented by ‘0’) out of total 926 neutral tweets. Its’ accuracy decreases a little with correctly identified positive and negative tweets comes out to be 711 and 678 out of the total 896 positive and 878 negative tweets respectively.

Figure 4: Confusion matrix of decision tree algorithm on twitter SA

```
[[678  77 123]
 [ 50 821  55]
 [114  71 711]]
```



The figure 5 shows the weighted average report rate of DT classifier implemented on twitter test data. The DT F1 score of neutral tweets has a maximum percentage followed by slight decrease in the F1 score of positive and negative tweets. This report signifies that DT algorithm is best on picking Bag of Words (BoW) and Parts of Speech (PoS) of text but performs little less on recognizing punctuation or emoticons in textual data.

Figure 5: Decision Tree weighted average report

```
In [8]: print(classification_report(y_test, y_pred))
              precision    recall  f1-score   support

          -1       0.81      0.77      0.79       878
           0       0.85      0.89      0.87       926
           1       0.80      0.79      0.80       896

   micro avg       0.82      0.82      0.82      2700
   macro avg       0.82      0.82      0.82      2700
weighted avg       0.82      0.82      0.82      2700
```

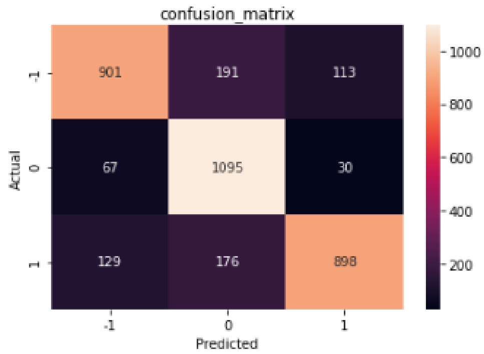
iii. Random Forest (RF)

It is an ‘Ensemble’ of decision trees i.e. it builds multiple decision trees and then merges them together to get higher accuracy and stable prediction [28]. Like DT it can be used for both classification and regression problems.

The RF classifier is implemented on 20% of the total 18,000 tweets. The confusion matrix shows (see figure 6) just like its parent classifier DT, RF performed exceptionally well with 91% accuracy on neutral tweets (1095 correctly identified out of total 1192 neutral tweets). Its accuracy decreases to 74% to identify positive and negative tweets (898 and 901 correctly identified out of total 1203 positive and 1205 negative tweets).

Figure 6: Confusion matrix of Random Forest algorithm on twitter SA

```
[[ 901  191  113]
 [  67 1095   30]
 [ 129  176  898]]
```



The Precision, Recall, F1 score and weighted average rate of RF classifier implemented on twitter test data is provided in figure 7. As expected with the performance of its parent DT classifier, the F1 score of RF classifier is maximum for neutral tweets (83%) which slides further to 80 and 78 percent for positive and negative tweets. The overall weighted average report of RF depicts it's profess in extracting unigram, bigram or n-gram features but little less proficiency in extracting punctuation marks.

Figure 7: Random Forest classifier weighted average report

```
In [9]: print(classification_report(y_test, y_pred))

              precision    recall  f1-score   support

          -1       0.82      0.75      0.78      1205
           0       0.75      0.92      0.83      1192
           1       0.86      0.75      0.80      1203

   micro avg       0.80      0.80      0.80      3600
   macro avg       0.81      0.80      0.80      3600
weighted avg       0.81      0.80      0.80      3600
```

iv. K-Nearest Neighbour

It is a non-parametric and instance-based learning algorithm as it doesn't assume anything about the underlying data [29]. In KNN a feature is classified by the plurality vote of its neighbours.

The performance of KNN classifier on 15% of 18,000 tweets for twitter SA shows a grim picture (see figure 8) with only 22 and 36 percent correctly identified positive and negative tweets (205 positive and 319 negative tweets out of total 896 positive and 878 negative tweets). The average report of KNN also shows below average performance (see figure 9). The weighted average F1 score is just 47% of all the tweets. The confusion matrix and average report clearly indicates its overall below average performance.

Figure 8: Confusion matrix of KNN algorithm on twitter SA

```
[[319 527  32]
 [ 78 837  11]
 [118 573 205]]
```

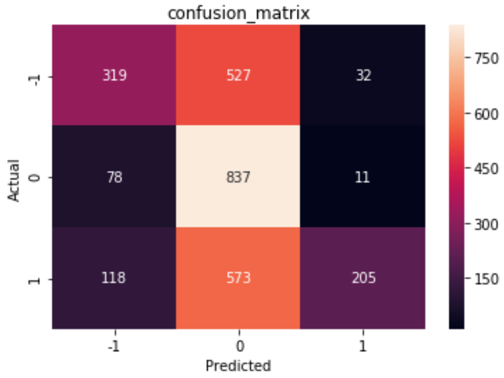


Figure 9: K-Nearest Neighbour classifier weighted average report

```
In [16]: print(classification_report(v1_test, v1_pred))
              precision    recall  f1-score   support

          -1       0.62      0.36      0.46       878
           0       0.43      0.90      0.58       926
           1       0.83      0.23      0.36       896

   micro avg       0.50      0.50      0.50      2700
   macro avg       0.63      0.50      0.47      2700
weighted avg       0.62      0.50      0.47      2700
```

v. Logistic Regression (LR)

This classifier is purely based on the concept of probability and to calculate the probability it uses ‘Sigmoid Function’ also called as ‘Logistic Function’ [30]. Here the dependent variable is binary in nature.

The 15% of test data is taken out of total 18,000 tweets to classify twitter data using Logistic Regression algorithm. The confusion matrix (figure 10) thus obtained indicates good performance with 94% correctly identified neutral tweets (879/926) followed by a little slid in performance with 83% positive and 81% negative tweets (744/896 positive and 713/878 negative tweets). The weighted average report (figure 11) generated also shows 89% F1 score of neutral tweets followed by 86% F1 score of positive and 84% F1 score of negative tweets. The overall weighted average performance of LR classifier is 86% on twitter SA which is quite creditable.

Figure 10: Confusion matrix of Logistic Regression algorithm on twitter SA

```
[[713 103  62]
 [ 23 879  24]
 [ 76  76 744]]
```

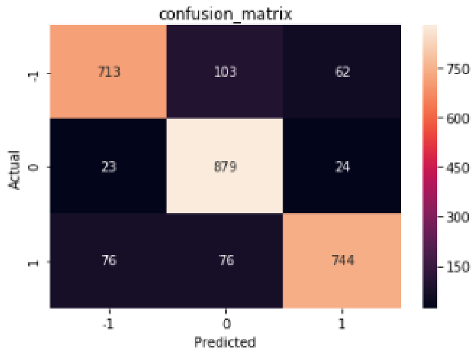


Figure 11: Logistic Regression classifier weighted average report

```
In [8]: print(classification_report(v1_test, v1_pred))
              precision    recall  f1-score   support

          -1       0.88      0.81      0.84       878
           0       0.83      0.95      0.89       926
           1       0.90      0.83      0.86       896

   micro avg       0.87      0.87      0.87      2700
   macro avg       0.87      0.86      0.86      2700
weighted avg       0.87      0.87      0.86      2700
```

4. Comparison and Result analysis

The base classifiers are implemented on twitter data to analyze the hidden sentiments of the tweets. The cross comparison results (see table 1) thus obtained indicates the best overall performance of LR classifier on all types of tweets followed by DT and RF classifiers wherein both are quite good to identify neutral tweets in comparison to positive and negative tweets. The NB classified all types of tweets with a fairly good accuracy but KNN classifier comes out to be below average performer with an overall accuracy rate of just 47%. The evaluation metrices [31] used for the purpose are illustrated below:

$$\text{Precision} = \frac{True_{Positive_{Statement}}}{True_{Positive_{Statement}} + False_{Positive_{Statement}}}$$

$$\text{Recall} = \frac{True_{Positive_{Statement}}}{True_{Positive_{Statement}} + False_{Negative\ Statement}}$$

$$\text{F1 score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$\text{Accuracy} = \frac{True_{Positive_{Statement}} + True_{NegativeStatement}}{True_{Positive_{Statement}} + False_{NegativeStatement} + True_{Negative\ Statement} + False_{PositiveStatement}}$$

Table 1: Cross comparison of the results obtained from base classifiers. Pre, Rec and F1 refers to the Precision, Recall and F-measure

Techniques	Accuracy (%)	Positive Class			Neutral			Negative Class			Average
		Pre (%)	Rec (%)	F1 (%)	Pre (%)	Rec (%)	F1 (%)	Pre (%)	Rec (%)	F1 (%)	F1 (%)
Naive Bayes (NB)	77.37	78	86	82	88	62	73	70	85	77	77
Decision Tree (DT)	81.85	80	79	80	85	89	87	81	77	79	82
Random Forest (RF)	80.38	86	75	80	75	92	83	82	75	78	80
K-Nearest Neighbour (KNN)	50.40	83	23	36	43	90	58	62	36	46	47
Logistic Regression (LR)	86.51	90	83	86	83	95	89	88	81	84	86

Conclusion

Through this paper a comparison of base classifiers is performed on twitter SA. The Logistic Regression shows the highest accuracy of 86.51% with an average F1 score of 86% for all the three types of tweets. The Decision Tree and Random Forest classifiers displays the near similar pattern, as both the classifiers analyze neutral tweets with much accuracy than the positive and negative tweets. The observation can be attributed by the fact that both the classifiers are little less efficient to extract punctuation signs and emoticons. The Naive Bayes classifier also performed fairly well with the accuracy rate of 77.37%. The least performer among all the base classifiers is K-Nearest Neighbour with the accuracy rate of just 50.40% and an average F1 score rate further slips to 47%. The results thus obtained can be helpful for the companies to analyze their product related customer opinions and also to customers to choose the best product based on public reviews. For future work we will work on ensemble classification techniques to classify public opinions.

References

[1] Statista. 2019. https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/.

[2] Pak A. and Paroubek P. (2010). *Twitter as a Corpus for Sentiment Analysis and Opinion Mining*. In the proceedings of the seventh conference on International Language Resources and Evaluation, pp. 1320-1326.

[3] Pang B. and Lee L. (2004). *A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts*. ACL Anthology, https://www.aclweb.org/anthology/P04-1035/.

[4] Hu M. and Liu B. (2004). *Mining and summarizing customer reviews*. KDD'04, Washington, USA. https://www.cs.uic.edu/~liub/publications/kdd04-revSummary.pdf.

[5] Wilson T., Wiebe J. and Hoffman P. (2005). *Recognizing contextual polarity in phrase level sentiment analysis*. ACM Digital Library. https://dl.acm.org/doi/10.3115/1220575.1220619.

[6] Agarwal Apoorv, Biadsy Fadi, and Mckeown Kathleen. (2009). *Contextual phrase-level polarity analysis using lexical affect scoring and syntactic n-grams*. In the proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), pp. 24–32.

[7] Hassan A., Abbasi A. And Zeng D. (2013). *Twitter sentiment analysis: A bootstrap ensemble framework*. Social Computing (SocialCom) IEEE international conference, pp. 357–364.

[8] Go Alec, Bhayani Richa, and Huang Lei. (2009). *Twitter sentiment classification using distant supervision*. Technical Report, Stanford.

[9] Barbosa Luciano and Feng Junlan. (2010). *Robust sentiment detection on twitter from biased and noisy data*. In the proceedings of the 23rd International Conference on Computational Linguistics, pp. 36–44.

[10] Bermingham Adam and Smeaton Alan. (2010). *Classifying sentiment in microblogs: is brevity an advantage is brevity an advantage?* ACM Digital Library, pp. 1833–1836.

[11] Speriosu M., Sudan N., Upadhyay S. and Baldridge J. (2011). *Twitter polarity classification with label propagation over lexical links and the follower graph*. In the proceedings of the first workshop on Unsupervised Learning in NLP, Association for Computational Linguistics. pp. 53–63.

[12] Zhang L., Ghosh R., Dekhil M., Hsu M. and Liu B. (2011). *Combining lexicon-based and learning-based methods for twitter sentiment analysis*. HP Laboratories. http://www.hpl.hp.com/techreports/2011/HPL-2011-89.html.

[13] Onan A., Korukolu S. And Bulut H. (2016). *A multi objective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification*. Expert Systems with Applications. Vol. 62, pp. 1–16. doi.org/10.1016/j.eswa.2016.06.005.

[14] Medhat W., Hassan A. And Korashy H. (2014). *Sentiment analysis algorithms and applications: A survey*. Ain Shams Engineering Journal. Vol. 5, pp. 1093–1113. doi.org/10.1016/j.asej.2014.04.011.

[15] Davidov D., Tsur O. And Rappoport A. (2010). *Enhanced sentiment learning using twitter hashtags and smileys*. In the proceedings of the 23rd international conference on computational linguistics: posters, Association for Computational Linguistics.pp.241–249.

[16] Da Silva N.F. and Hruschka E.R. (2014). *Tweet sentiment analysis with classifier ensembles*. Decision Support Systems, vol. 66, pp. 170-179 doi.org/10.1016/j.dss.2014.07.003.

[17] Wan Y. and Gao Q. (2015). *An ensemble sentiment classification system of twitter data for airline services analysis*. In the proceeding of IEEE international conference on Data Mining Workshop (ICDMW), pp. 1318–1325.

[18] Prusa J., Khoshgoftaar T.M. and Dittman D.J. (2015). *Using ensemble learners to improve classifier performance on tweet sentiment data*. In the proceedings of IEEE international conference on Information Reuse and Integration (IRI), pp.252–257.

[19] Feldman R. (2013). *Techniques and applications for sentiment analysis*. Communications of the ACM, vol. 56(4), pp. 82-89.

[20] Kim H., Moon H. And Ahn H. (2011). *A weight-adjusted voting algorithm for ensembles of classifiers*. Journal of the Korean Statistical Society, vol. 40, pp. 437–449. doi.org/10.1016/j.jkss.2011.03.002.

[21] Rezapour R., Wang L., Abdar O. and Diesner J. (2017). *Identifying the overlap between election result and candidates' ranking based on hashtag-enhanced, lexicon-based sentiment analysis*. In the proceedings of 11th IEEE international conference on Semantic Computing (ICSC), pp. 93–96.

[22] Esuli A. and Sebastiani F. (2006). *Sentiwordnet: A High-Coverage Lexical Resource for Opinion Mining*. Institute of Information Science and Technologies (ISTI) of the Italian National Research Council (CNR): Pisa, Italy.

[23] Levy M. (2016). *Playing with Twitter Data. R-bloggers*, https://www.r-bloggers.com/playing-with-twitter-data/.

[24] Kanavos A., Nodarakis N., Sioutas S., Tsakalidis A., Tsolis D. and Tzimas G. (2017). *Large scale implementations for twitter sentiment classification*. Algorithms. Vol. 10, pp. 33-37.

[25] Ortigosa A., Martín J.M. and Carro R.M. (2014). *Sentiment analysis in Facebook and its application to e-learning*. Computers in Human Behavior, vol. 31, pp. 527-541.

[26] Naive Bayes Classifier: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c.

[27] Decision Tree Algorithm: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm.

[28] Random Forest Algorithm: https://builtin.com/data-science/random-forest-algorithm.

[29] KNN: https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f26

[30] Logistic Regression: https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148.

[31] Evaluation metrices: https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226.