

# DUCS at SemEval-2022 Task 6: Exploring Emojis and Sentiments for Sarcasm Detection

**Vandita Grover**

Department of Computer Science  
University of Delhi  
vgrover@cs.du.ac.in

**Hema Banati**

Dyal Singh College  
University of Delhi  
hemabanati@dsc.du.ac.in

## Abstract

This paper describes the participation of team DUCS at SemEval 2022 Task 6: iSarcasmEval - Intended Sarcasm Detection in English and Arabic. Team DUCS participated in SubTask A of iSarcasmEval which was to determine if the given English text was sarcastic or not. In this work, emojis were utilized to capture how they contributed to the sarcastic nature of a text. It is observed that emojis can augment or reverse the polarity of a given statement. Thus sentiment polarities and intensities of emojis, as well as those of text, were computed to determine sarcasm. Use of capitalization, word repetition, and use of punctuation marks like '!' were factored in as sentiment intensifiers. An NLP augments were used to tackle the imbalanced nature of the sarcasm dataset. Several architectures comprising of various ML and DL classifiers, and transformer models like BERT and Multimodal BERT were experimented with. It was observed that Multimodal BERT outperformed other architectures tested and achieved an F1-score of 30.71%. The key takeaway of this study was that sarcastic texts are usually positive sentences. In general emojis with positive polarity are used more than those with negative polarities in sarcastic texts.

## 1 Introduction

According to the Collins Dictionary <sup>1</sup>

"Sarcasm is speech or writing which actually means the opposite of what it seems to say".

Fox Tree et al. (2020) report that sarcasm is challenging to identify, even for humans. People often use gestures like rolling of eyes or heavy tonal stress to express sarcasm (Pandey et al., 2019), in speech or in-person communication. Intonation

and stress in speech, too, are strong indicators of sarcasm (Castro et al., 2019). Also, the context in the form of shared knowledge between the speaker and the audience can be leveraged to detect sarcasm (Amir et al., 2016). But most of the time the tone or the context is missing in text data, especially tweet data. Some users attempt to use capitalization like "OH YEAH" or repetitions like "woowww" to indicate tonal intensity. But, most of the time these cues are not enough, especially in absence of context to gauge sarcasm. This makes sarcasm detection a challenging task.

In this work, team DUCS participated in *SemEval 2022 Task 6 : iSarcasmEval* Subtask A, (Abu Farha et al., 2022), to determine if the given text is sarcastic or not. The English text training dataset contains 3468 sentences. Each text is labeled as sarcastic or non-sarcastic by the text-author. Of the 3468 sentences 867 are sarcastic and 2601 are non-sarcastic. In the dataset corresponding to a sarcastic text, a rephrased non-sarcastic version has also been provided. These sarcastic texts are labeled as one of sarcasm, irony, satire, understatement, overstatement, and rhetorical question. These ironic speech categories are determined by a linguist expert. The target task is to determine whether a text is sarcastic or not on the Test dataset which contains 1400 sentences.

For sarcasm detection in the above-discussed dataset, an attempt was made to derive cues from the text itself. It is observed that in real scenarios, the absence of labels by users like #not or #sarcasm or any other contextual information makes sarcasm detection challenging (Chaudhari and Chandkhede, 2017). Thus, in this work, the authors attempted to use the information available in the text itself. Emoji occurrences in the text are captured to study their impact in discerning sarcasm. Sentiment intensity and polarity of both the emojis and the text were computed to train the classifiers

<sup>1</sup><https://www.collinsdictionary.com/dictionary/english/sarcasm>

for sarcasm classification.

This paper is organized as follows. Section 2 discusses related work in sarcasm detection. In Section 3, System Overview, the proposed approach to incorporate polarity and intensity of emojis along with text sentiment polarity and intensity is discussed. Section 4, describes the Experimental Setup and the data augmentation approach used for the given dataset. The results of the experiments conducted are reported in Section 5. In the last section Conclusion, the key takeaways and learnings are discussed.

## 2 Related work

The sarcastic nature of the text can deceive classifiers as well as humans, given it conveys the opposite of what it means. Many researchers have attempted to use text sentiment and emojis for sarcasm detection.

[Subramanian et al. \(2019\)](#) use word and emoji embeddings simultaneously to train deep learning models with attention layer for sarcasm detection. [Pamungkas and Patti \(2018\)](#) utilize structural features and affective features of tweets for irony detection. Structural features include the presence of hashtags, links, emojis, quotes, etc. The counts of mentions, exclamations, upper-case, intensifiers, links, along with the counts of verbs, nouns, and adjectives are also used along with the use various affective resources to capture sentiment polarity and emotions. [Lemmens et al. \(2020\)](#) used an ensemble approach with LSTM for representing emojis and hashtags, CNN-LSTM for representation of cases, stop-words, punctuation, and sentiments. MLP was trained with Facebook's InferSent embeddings ([Conneau et al., 2017](#)) while SVM was trained with emotion and stylometric features. A decision tree with Adaboost served as the base estimator for the ensemble. This ensemble was used to predict sarcasm on the conversational text where context was available. [Sundararajan and Palanisamy \(2020\)](#) propose a rule-based classifier with an ensemble of 20 different features. They observed that the features based on sentiment can predict sarcasm better in combination with contradictory features.

In most work so far researchers have tried to capture sentiment and linguistic features to identify sarcasm. A few researchers have attempted to include emojis in sarcasm classification. But, how the usage of emojis with text impacts or conveys sarcasm is yet to be explored in depth. It needs to

be studied how frequent is sentiment incongruence of emojis in the text they are used with and can the polarity of emojis and text help deduce sarcasm. Hence in this work, the authors focus on these features - emoji sentiment and text sentiment as well as their polarities to understand how this combination impacts sarcasm. This approach is discussed in the next section.

## 3 System overview

In the given dataset, situational context is not available. Although linguistic markers are provided, they may not be available in real scenarios, eg when the users post on social media. In the absence of context, it is important to take into account other factors that may point to sarcasm. For instance, the surface sentiment of a text is often used in many sarcasm classifiers ([Joshi et al., 2016](#)). [Sulis et al. \(2016\)](#), report that most often sarcasm is used with an apparently positive statement to produce a negative impact. It is noted that the users may use intensity in the form of capitalization, repeated letters, etc. to express sarcasm ([Chaudhari and Chandakhede, 2017](#)). In this work, to identify sarcasm, the sentiment polarity of the text and intensity in the form of sentiment score was incorporated for the sarcasm classification task. Intensity as a feature is used as it may help capture intonation or stress a user may want to express.

It has been reported that features like emojis can augment as well as alter the sentiment polarity ([Grover, 2021](#)).

[Pamungkas and Patti \(2018\)](#) observed that emoji incongruity with the text may help improve the irony classification. Thus, this work investigates whether in sarcastic texts users use in-congruent emojis with the text to express sarcasm. And whether the use of emojis in the text helps in detecting sarcasm?

**Data preparation** The text dataset was cleaned to remove URLs as they do not contribute to the text sentiment. Most text preprocessing approaches in sentiment classification convert characters to lower case, reduce repeating letters, and perform lemmatization. To compute the sentiment score and polarities, special care was taken to not remove repetition of letters, capitalization or punctuation as they may serve as intensifiers. These special usages can later be removed during the experimentation while training the classifiers.

**Emoji extraction** In the next step, the emojis are separated from text. The training dataset for this task contained 672 text sentences with emojis of which 195 were sarcastic. The *demoji 1.1.0* Python package <sup>2</sup>, is used for emoji extraction.

**Text sentiment polarity and intensity** To evaluate intensity or the sentiment score of the text and text polarity the VADER sentiment analyzer (Hutto and Gilbert, 2014), available in Python’s natural language toolkit is employed. VADER is observed to work better in the language used by the users on social media (Illia et al., 2021), (Bonta and Janardhan, 2019). Users tend to communicate informally, using slang, abbreviations, capitalization and punctuation marks, etc. on social media. VADER can capture these components while computing sentiment polarity and sentiment scores. The compound scores provided by VADER are used as the sentiment score. If the compound score is  $\leq -0.5$  the sentiment polarity is considered negative, between  $(-0.5, 0.5)$ , the polarity is considered as neutral, while the score  $\geq 0.5$  is considered to be positive.

**Emoji sentiment polarity** As discussed above, since emojis are known to contribute to text sentiment and in many cases, they may even reverse the sentiment polarity, the emojis extracted for each text are stored separately in the list. A total of 672 texts have emojis in the provided dataset. For extracting emoji sentiment, Emoji Sentiment Ranking (Novak et al., 2015) is employed. Emoji Sentiment Ranking gives the sentiment score of the emojis.

A text may contain numerous emojis which is an indicator of both the context as well as associated emotion with the text. For example, a single emoji used several times with a text may be considered as a sentiment intensifier. Thus, all the emojis in the text were considered while computing emoji polarity and emoji sentiment score.

Emoji sentiment scores were captured for all emojis. If an emoji did not appear in the Emoji Sentiment Ranking, it was demojized using the *emojiz 1.6.3* Python package<sup>3</sup>. Demojizing here means replacing the emojis with their Unicode Consortium description in English, (Unicode, 2022). These emoji descriptions are passed to the VADER sentiment analyzer to compute the sentiment score of

the emojis. Again the compound score provided by VADER is used to compute the final score. Scores of all the emojis are then added to compute the total emoji score for all the emoji scores associated with the text. These emojis are then normalized as done in VADER and corresponding emoji polarities are computed.

$$normalizedScore = \frac{score}{\sqrt{score^2 + \alpha}} \quad (1)$$

where default value of  $\alpha$  is 15.

Each text with emojis is labeled with the computed emoji score and emoji polarity. For the text with no emojis, the emoji sentiment score is assigned to zero, and polarity is assigned as neutral.

For example if the text is "*@AsdaServiceTeam imagine your delivery being 2 hours late, and imagine calling up your service team only for them to hang up at 10pm, coincidentally the same time the office closes. But it's okay, my £3 delivery fee is being refunded though 🙄*". The VADER text score is computed as 0.2263 with "+" text polarity. The VADER emoji sentiment score is computed as 0.0150371002692231 with "+" emoji polarity. The prepared dataset with the derived features of text sentiment scores, text polarity, emoji sentiment score, and emoji polarity is now used to train the different classifiers discussed in the next section.

## 4 Experimental setup

For sarcasm detection, this work employs various machine and deep learning classifiers. The proposed set up is also tested with transformer models like BERT and Multimodal BERT.

First, the text is cleaned to remove stop words, newlines, and spaces. The cleaned text is then converted to lower case and then tokenized to prepare for loading the word embeddings. Pre-trained GloVe embeddings, (Pennington et al., 2014) of 100 dimensions are used to represent the tokenized text.

80% of the dataset was used for training while the remaining 20% was used for validation.

**Data augmentation** This dataset was highly imbalanced with 867 sarcastic and 2601 non-sarcastic texts, i.e. (3:1 ratio for non-sarcastic: sarcastic texts). So the dataset was augmented to improve this ratio of sarcastic and non-sarcastic texts. The sarcastic texts were oversampled with *nlpaug* python library <sup>4</sup> for text augmentation.

<sup>2</sup><https://pypi.org/project/demoji/>

<sup>3</sup><https://pypi.org/project/emojiz/>

<sup>4</sup><https://nlpaug.readthedocs.io/en/latest/index.html>

The contextual BERT embeddings were used to synthesize new texts from existing texts in the dataset. Two different augmented datasets were created. In the first augmented dataset (Aug1 Training Set), each sarcastic text was synthesized once to create one more text. In this set, the ratio of non-sarcastic: sarcastic texts was 2:1. In the next augmented dataset (Aug2 Training Set), each sarcastic text was synthesized to create two more texts, thus resulting in the ratio of 1:1 for non-sarcastic to sarcastic texts.

For example, if the tweet was, "See Brexit is going well". Then in the first augmented text (Aug1 Training Set) the original tweet (mentioned above) was retained and a synthesized tweet in the form "can see brexit negotiations is going extremely well" was added to the dataset. This was done for every sarcastic tweet, making the ratio of non-sarcastic:sarcastic texts as 2:1.

In the Aug2 Training Set) the number of synthesized texts was increased to two. So two synthesized tweets apart from the original tweet were appended to the dataset making the ratio of non-sarcastic:sarcastic as 1:1.

The computed sentiment polarity, text sentiment score, emoji sentiment polarity, and emoji sentiment score were replicated from the original texts in the synthesized texts.

**Model architecture** The different architectures experimented with are as follows:

- Gaussian Naive Bayes (Perez et al., 2006)
- Support Vector Machine with Linear Kernel (Wang, 2005)
- Logistic Regression (Wright, 1995)
- Sequential model with 1 dense fully connected layer (Sutskever et al., 2014)
- LSTM with Adadelta optimiser, 128 batch size, run over 10 epochs (Sundermeyer et al., 2012)
- Bi-LSTM (Graves and Schmidhuber, 2005) with Adadelta optimiser, 128 batch size, run over 10 epochs
- BERT (Devlin et al., 2018) with the learning rate of 1e-5, batch size 32 run over 4 epochs with the base uncased classifier

- Multimodal BERT with the batch size 32 run over 6 epochs with the base uncased classifier

## 5 Results

This section reports the results of various classifiers on which the proposed approach was tested with. The original training set (Original Training Set) provided for the competition is an imbalanced dataset contains 2601 non-sarcastic and 867 sarcastic texts. The training set augmented with 1X sarcastic tweets (Aug1 Training Set) contains 1734 sarcastic and 2601 non-sarcastic texts. 867 sarcastic texts were synthesized to create 867 more sarcastic texts, resulting in a total of 1734 sarcastic texts. The training set augmented with 2X sarcastic tweets (Aug2 Training Set) contains 2601 sarcastic and 2601 non-sarcastic texts. In this set, each of the 867 sarcastic texts were synthesized twice. Hence the total number of sarcastic texts were 867(original) and 1734(each original tweet synthesized twice).

Validation-F1 reports F1 score (%) that were obtained during the validation. Results obtained on the test dataset, (Abu Farha et al., 2022), provided during the competition are reported in Test-F1.

The results of models trained without sentiment scores and polarities on the original training set in Table 1, training set augmented with 1X sarcastic tweets in Table 2, and training set augmented with 2X sarcastic tweets in Table 3 are reported.

Classifier	Validation-F1	Test-F1
Gaussian Naive Bayes	25.87	14.52
Logistic Regression	7.03	5.45
SVM	7.11	4.65
Sequential	44.09	24.53
LSTM	44	25
Bi-LSTM	44	25
BERT	77.56	22.46

Table 1: Original Training Set without Sentiment Scores and Polarities

The results of models trained with text and emoji sentiment scores and their respective polarities on Original, Aug1, and Aug2 are reported in 4, 5, 6 respectively.

The Multimodal BERT trained with 1736 sarcastic and 2601 non-sarcastic sentences was eventually used to determine sarcastic text in the test data. With this work the authors achieved an overall F1-Score of 30.71% on the test dataset and was ranked 24 out of the 43 participating teams.



Classifier	Validation-F1	Test-F1
Gaussian Naive Bayes	50.73	19.39
Logistic Regression	19.13	16.18
SVM	19.5	16.42
Sequential	56.48	24.53
LSTM	56.48	25
Bi-LSTM	56.48	24.53
BERT	64	23.94

Table 2: Aug1 Training Set without Sentiment Scores and Polarities

Classifier	Validation-F1	Test-F1
Gaussian Naive Bayes	59.04	16.58
Logistic Regression	61.4	22.5
SVM	61.57	22.58
Sequential	66.71	25
LSTM	66.71	25
Bi-LSTM	66.71	25
BERT	64	23.94

Table 3: Aug2 Training Set without Sentiment Scores and Polarities

It was observed that in general, models trained with sentiment scores of text and emojis alongwith their polarities performed better than those trained only with text

Figure 1 shows the distribution of emojis across sarcastic text in the train and test dataset. From Figure 1 of iSarcasmEval dataset

- It is observed that sarcastic texts mostly have emojis with positive polarity.
- Sarcastic texts generally demonstrate positive sentiment.

Some other observations noted during the experiments are reported as follows.

- Sarcastic text when augmented twice results in overfitting and results are not satisfactory on the test set. The best F1-score on the test dataset is achieved when each sarcastic text is augmented once.

Classifier	Validation-F1	Test-F1
Gaussian Naive Bayes	21.3	15.72
Logistic Regression	6.41	2.84
SVM	6.52	2.87
Sequential	37.57	24.53
LSTM	47.03	28.47
Bi-LSTM	47.03	28.48
BERT	35.29	23.28
MultiModal BERT	50.00	25.12

Table 4: Original Training Set with Text and Emoji Sentiment Scores and Polarities

Classifier	Validation-F1	Test-F1
Gaussian Naive Bayes	32.0	18.5
Logistic Regression	17.56	11.11
SVM	17.53	10.89
Sequential	56.75	24.52
LSTM	40.73	28.49
Bi-LSTM	40.51	28.49
BERT	73.184	29.27
MultiModal BERT	75.22	<b>30.71</b>

Table 5: Aug1 Training Set with Text and Emoji Sentiment Scores and Polarities

Classifier	Validation-F1	Test-F1
Gaussian Naive Bayes	57.51	20.93
Logistic Regression	54.3	20.62
SVM	54.77	20.75
Sequential	64.26	24.52
LSTM	49.82	28.50
Bi-LSTM	49.82	28.50
BERT	78.36	24.31
MultiModal BERT	81.22	24.69

Table 6: Aug2 Training Set with Text and Emoji Sentiment Scores and Polarities

- Overall the Multi-modal BERT, which can handle tabular data, performs the best.
- Models trained with sentiment scores and polarities of text and emojis performed slightly better than those trained with only text.
- In the test dataset only 18 sarcastic sentences had emojis.
- F1-score on test dataset for text with emojis was found to be 21.05% while that of text without emojis was 31.02%. This performance may be improved if more sentences with emojis are available.

## 6 Conclusion

In this work, the authors took their first steps in understanding how the sentiment of emojis alongwith the text sentiment helps in detecting sarcasm. Since sarcasm is difficult to detect without context, authors attempted to uncover information implicit in the text. For this the intensity and polarity of both the available emojis and the text itself were used to identify sarcasm.

It was observed that sarcastic texts generally have positive polarity. Sarcastic texts employed emojis with positive polarities more than the negative emojis. This information can be used further to improve results of sarcasm classification. The given dataset

Dataset		Sarcastic With Emojis	
<b>Train</b> Total (3468)		<b>Text Polarity</b> Total: 195	
Sarcastic	867	+ve	112:
	With Emojis		195
	Without Emojis		672
Non-Sarcastic	2601	-ve	56:
	With Emojis		477
	Without Emojis		2124
<b>Test</b> Total (1400)		<b>Text Polarity</b> Total: 18	
Sarcastic	200	+ve	17:
	With Emojis		18
	Without Emojis		182
Non-Sarcastic	1200	-ve	0:
	With Emojis		285
	Without Emojis		915

Figure 1: Distribution of Emojis in Sarcastic Text in Train and Test Datasets of iSarcasmEval

had relatively fewer texts with emojis, thus, more work is required to fully capitalize on the proposed approach. This motivates the authors to explore more sarcasm datasets with emojis to exhaustively study the impact of emojis in identifying sarcasm.

## 7 Acknowledgement

The authors thank the organizers of SemEval-22 organizers and iSarcasmEval for the conduct of this competition. We thank fellow competitors and peer reviewers for contributing to our learning.

## References

Ibrahim Abu Farha, Silviu Oprea, Steven Wilson, and Walid Magdy. 2022. SemEval-2022 Task 6: iSarcasmEval, Intended Sarcasm Detection in English and Arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mário J Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976*.

Venkateswarlu Bonta and Nandhini Kumares2and N Janardhan. 2019. A comprehensive study on lexicon based approaches for sentiment analysis. *Asian Journal of Computer Science and Technology*, 8(S2):1–6.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards multimodal sarcasm detection (an obviously perfect paper). *arXiv preprint arXiv:1906.01815*.

Pranali Chaudhari and Chaitali Chandankhede. 2017. Literature survey of sarcasm detection. In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 2041–2046.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jean E Fox Tree, J Trevor D’Arcey, Alicia A Hammond, and Alina S Larson. 2020. The sarcasm: Sarcasm production and identification in spontaneous conversation. *Discourse Processes*, 57(5-6):507–533.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610.

Vandita Grover. 2021. Exploiting emojis in sentiment analysis: A survey. *Journal of The Institution of Engineers (India): Series B*, pages 1–14.

Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225.

Fathonah Illia, Migunani Puspita Eugenia, and Sita Aliya Rutba. 2021. Sentiment analysis on pedulilindungi application using textblob and vader library. In *Proceedings of The International Conference on Data Science and Official Statistics*, volume 2021, pages 278–288.

Aditya Joshi, Pushpak Bhattacharyya, and Mark James Carman. 2016. Automatic sarcasm detection: A survey. *CoRR*, abs/1602.03426.

Jens Lemmens, Ben Burtenshaw, Ehsan Lotfi, Iliia Markov, and Walter Daelemans. 2020. Sarcasm detection using an ensemble approach. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 264–269.

P Kralj Novak, J Smailovic, B Sluban, and I Mozetic. 2015. Emoji sentiment ranking. *PLoS ONE*, 10(12):e0144296.

- Endang Wahyu Pamungkas and Viviana Patti. 2018. #nondicevosulserio at semeval-2018 task 3: Exploiting emojis and affective content for irony detection in english tweets. In *International Workshop on Semantic Evaluation*, pages 649–654. Association for Computational Linguistics.
- Avinash Chandra Pandey, Saksham Raj Seth, and Mahima Varshney. 2019. Sarcasm detection of amazon alexa sample set. In *Advances in Signal Processing and Communication*, pages 559–564. Springer.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Aritz Perez, Pedro Larranaga, and Inaki Inza. 2006. Supervised classification with conditional gaussian networks: Increasing the structure complexity from naive bayes. *International Journal of Approximate Reasoning*, 43(1):1–25.
- Jayashree Subramanian, Varun Sridharan, Kai Shu, and Huan Liu. 2019. Exploiting emojis for sarcasm detection. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 70–80. Springer.
- Emilio Sulis, Delia Irazú Hernández Farías, Paolo Rosso, Viviana Patti, and Giancarlo Ruffo. 2016. Figurative messages and affect in twitter: Differences between # irony, # sarcasm and # not. *Knowledge-Based Systems*, 108:132–143.
- Karthik Sundararajan and Anandhakumar Palanisamy. 2020. Multi-rule based ensemble feature selection model for sarcasm type detection in twitter. *Computational intelligence and neuroscience*, 2020.
- Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.
- The Unicode. 2022. The Unicode Consortium. <http://www.unicode.org/emoji/charts/full-emoji-list.html>. [Online; accessed 15-February-2022].
- Lipo Wang. 2005. *Support vector machines: theory and applications*, volume 177. Springer Science & Business Media.
- Raymond E Wright. 1995. Logistic regression.