2017S2 KT Project 2 Reviews

Print Options:

☑ Include Questions & Answers ☑ Include Comments ☑ Include All Reviews ☑ Include File Info

Print

JIANGYUE YAN'S PEERMARK REVIEW OF ANONYMOUS'S PAPER
(0% COMPLETED)

ASSIGNED QUESTIONS

1. Please describe what the author has done, in a couple of sentences
2. Please indicate what you think the author has done well, and why
3. Please indicate what you think could have been improved, and why (we would prefer you to focus on the content, rather than the language, structure, or style)

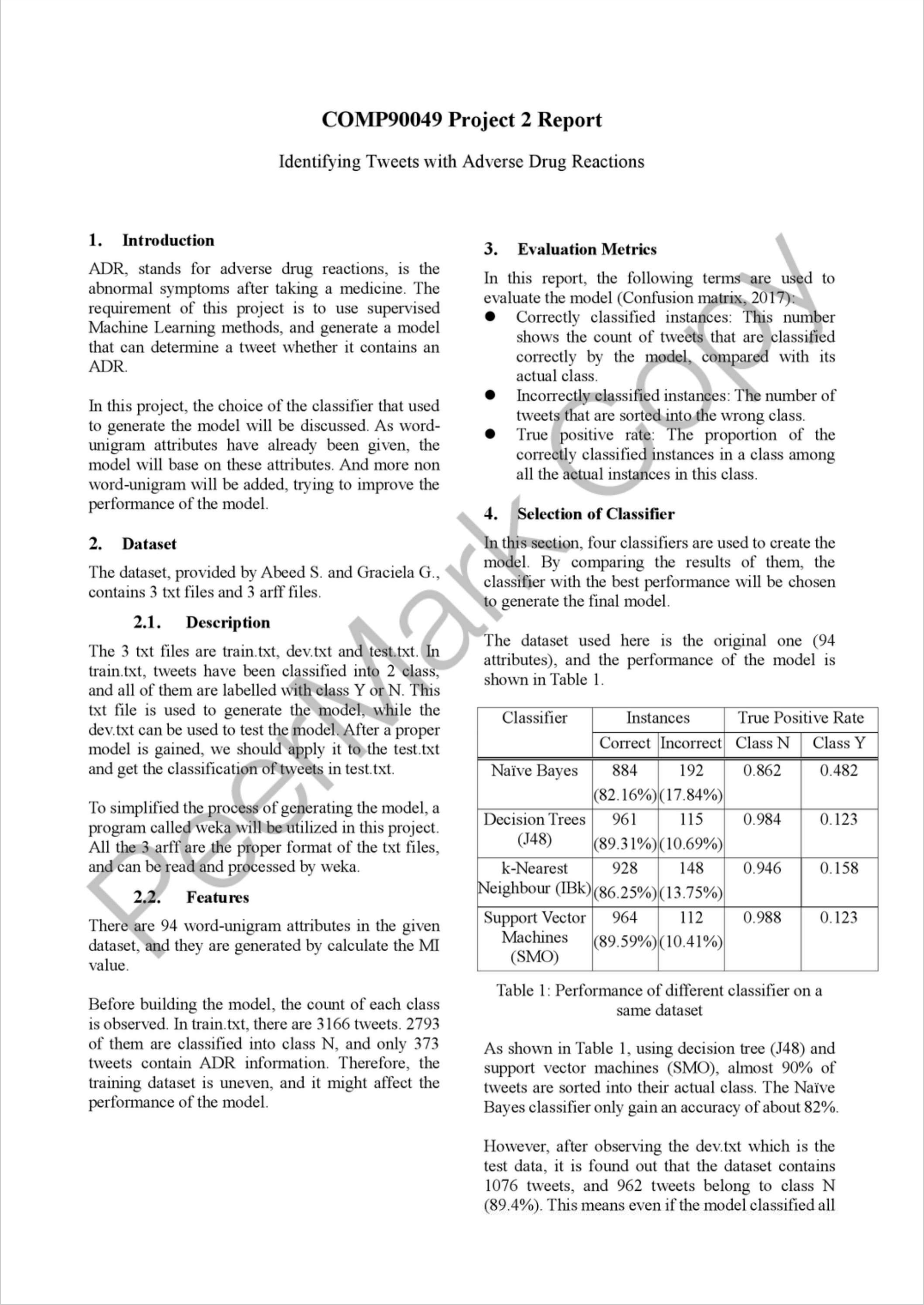
COMMENTS LIST

No comments added

SUBMITTED FILE INFO

file name	Xing_Wei_xwei4_-_Project_2_Report.pdf
file size	517.16K

"XING WEI (XWEI4) - PROJECT 2 REPORT" BY ANONYMOUS



COMP90049 Project 2 Report

Identifying Tweets with Adverse Drug Reactions

1. Introduction

ADR, stands for adverse drug reactions, is the abnormal symptoms after taking a medicine. The requirement of this project is to use supervised Machine Learning methods, and generate a model that can determine a tweet whether it contains an ADR.

In this project, the choice of the classifier that used to generate the model will be discussed. As word-unigram attributes have already been given, the model will base on these attributes. And more non word-unigram will be added, trying to improve the performance of the model.

2. Dataset

The dataset, provided by Abeed S. and Graciela G., contains 3 txt files and 3 arff files.

2.1. Description

The 3 txt files are train.txt, dev.txt and test.txt. In train.txt, tweets have been classified into 2 class, and all of them are labelled with class Y or N. This txt file is used to generate the model, while the dev.txt can be used to test the model. After a proper model is gained, we should apply it to the test.txt and get the classification of tweets in test.txt.

To simplified the process of generating the model, a program called weka will be utilized in this project. All the 3 arff are the proper format of the txt files, and can be read and processed by weka.

2.2. Features

There are 94 word-unigram attributes in the given dataset, and they are generated by calculate the MI value.

Before building the model, the count of each class is observed. In train.txt, there are 3166 tweets. 2793 of them are classified into class N, and only 373 tweets contain ADR information. Therefore, the training dataset is uneven, and it might affect the performance of the model.

3. Evaluation Metrics

In this report, the following terms are used to evaluate the model (Confusion matrix, 2017):

- Correctly classified instances: This number shows the count of tweets that are classified correctly by the model, compared with its actual class.
- Incorrectly classified instances: The number of tweets that are sorted into the wrong class.
- True positive rate: The proportion of the correctly classified instances in a class among all the actual instances in this class.

4. Selection of Classifier

In this section, four classifiers are used to create the model. By comparing the results of them, the classifier with the best performance will be chosen to generate the final model.

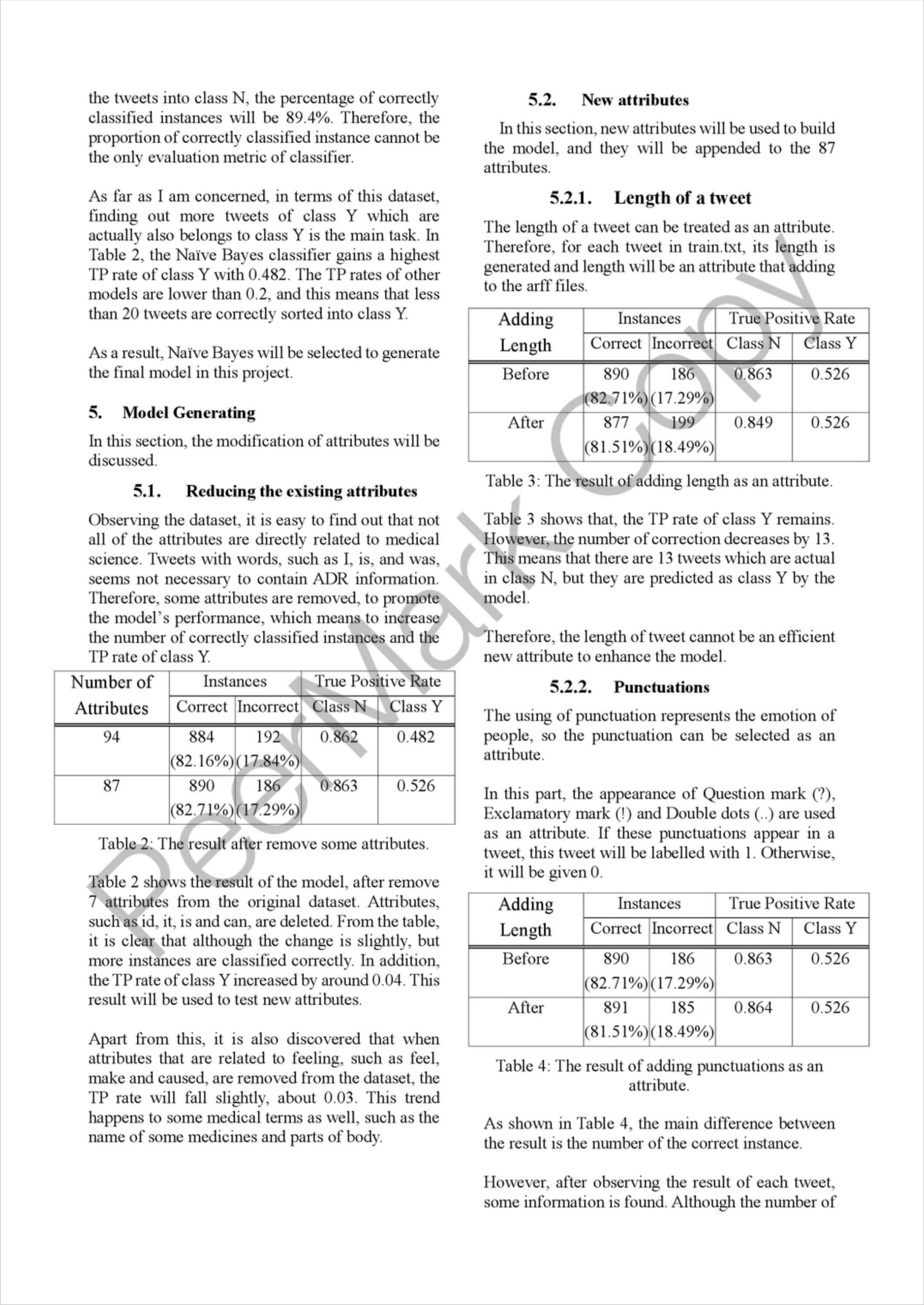
The dataset used here is the original one (94 attributes), and the performance of the model is shown in Table 1.

Classifier	Instances		True Positive Rate	
	Correct	Incorrect	Class N	Class Y
Naïve Bayes	884 (82.16%)	192 (17.84%)	0.862	0.482
Decision Trees (J48)	961 (89.31%)	115 (10.69%)	0.984	0.123
k-Nearest Neighbour (IBk)	928 (86.25%)	148 (13.75%)	0.946	0.158
Support Vector Machines (SMO)	964 (89.59%)	112 (10.41%)	0.988	0.123

Table 1: Performance of different classifier on a same dataset

As shown in Table 1, using decision tree (J48) and support vector machines (SMO), almost 90% of tweets are sorted into their actual class. The Naïve Bayes classifier only gain an accuracy of about 82%.

However, after observing the dev.txt which is the test data, it is found out that the dataset contains 1076 tweets, and 962 tweets belong to class N (89.4%). This means even if the model classified all



the tweets into class N, the percentage of correctly classified instances will be 89.4%. Therefore, the proportion of correctly classified instance cannot be the only evaluation metric of classifier.

As far as I am concerned, in terms of this dataset, finding out more tweets of class Y which are actually also belongs to class Y is the main task. In Table 2, the Naïve Bayes classifier gains a highest TP rate of class Y with 0.482. The TP rates of other models are lower than 0.2, and this means that less than 20 tweets are correctly sorted into class Y.

As a result, Naïve Bayes will be selected to generate the final model in this project.

5. Model Generating

In this section, the modification of attributes will be discussed.

5.1. Reducing the existing attributes

Observing the dataset, it is easy to find out that not all of the attributes are directly related to medical science. Tweets with words, such as I, is, and was, seems not necessary to contain ADR information. Therefore, some attributes are removed, to promote the model's performance, which means to increase the number of correctly classified instances and the TP rate of class Y.

Number of Attributes	Instances		True Positive Rate	
	Correct	Incorrect	Class N	Class Y
94	884 (82.16%)	192 (17.84%)	0.862	0.482
87	890 (82.71%)	186 (17.29%)	0.863	0.526

Table 2: The result after remove some attributes.

Table 2 shows the result of the model, after remove 7 attributes from the original dataset. Attributes, such as id, it, is and can, are deleted. From the table, it is clear that although the change is slightly, but more instances are classified correctly. In addition, the TP rate of class Y increased by around 0.04. This result will be used to test new attributes.

Apart from this, it is also discovered that when attributes that are related to feeling, such as feel, make and caused, are removed from the dataset, the TP rate will fall slightly, about 0.03. This trend happens to some medical terms as well, such as the name of some medicines and parts of body.

5.2. New attributes

In this section, new attributes will be used to build the model, and they will be appended to the 87 attributes.

5.2.1. Length of a tweet

The length of a tweet can be treated as an attribute. Therefore, for each tweet in train.txt, its length is generated and length will be an attribute that adding to the arff files.

Adding Length	Instances		True Positive Rate	
	Correct	Incorrect	Class N	Class Y
Before	890 (82.71%)	186 (17.29%)	0.863	0.526
After	877 (81.51%)	199 (18.49%)	0.849	0.526

Table 3: The result of adding length as an attribute.

Table 3 shows that, the TP rate of class Y remains. However, the number of correction decreases by 13. This means that there are 13 tweets which are actual in class N, but they are predicted as class Y by the model.

Therefore, the length of tweet cannot be an efficient new attribute to enhance the model.

5.2.2. Punctuations

The using of punctuation represents the emotion of people, so the punctuation can be selected as an attribute.

In this part, the appearance of Question mark (?), Exclamatory mark (!) and Double dots (..) are used as an attribute. If these punctuations appear in a tweet, this tweet will be labelled with 1. Otherwise, it will be given 0.

Adding Length	Instances		True Positive Rate	
	Correct	Incorrect	Class N	Class Y
Before	890 (82.71%)	186 (17.29%)	0.863	0.526
After	891 (81.51%)	185 (18.49%)	0.864	0.526

Table 4: The result of adding punctuations as an attribute.

As shown in Table 4, the main difference between the result is the number of the correct instance.

However, after observing the result of each tweet, some information is found. Although the number of tweets that classified into class Y are same, but for same tweet, the result might be different. For example, before adding new attribute, the tweet with number 789 is incorrectly sorted into class N. The new model fixes this mistake, but incorrectly classifies tweet 666, which is correct in the old model. This situation also happens to other tweets in the dataset.

Therefore, it is discovered that the classification of tweet may change, when adding new attributes into the model. However, the result will be slightly different, because the model is based on the combination of all attributes, instead of the new one.

6. Improvements

6.1. Increase tweets to training set

The training data set contains 3166 tweets, and the number of tweet labelled with class Y is 373. Increase the total number of training set may help to improve the model. On the other hand, the class of each tweet in training set should be checked. For example, "vyvanse makes me think too much" appears 4 times in train.txt, but 3of them are in class Y, and one of them is in class N. Therefore, one improvement is to get more training data.

6.2. Selection of attributes

In this project, the selection of attributes is less efficient, because the result of adding these new attributes is not significant. More non word-unigram attributes should be tested in the future, for example, whether the tweet contains parts of body. The manner of tweets may also help to improve the performance of the model.

7. Conclusions

In this project, a model that used to classified tweets with ADR information is generated. In terms of the given dataset, both the correctly classified instances and the TP rate are important evaluation metrics for models, because the dataset is unbalanced. By comparing the classifier, Naïve Bayes is selected. New attributes, the length of tweet and the punctuation appears in the tweet, are used to build the model. However, in this project these attributes are not very efficient. Therefore, other new attributes should be generated and improve the model in the future.

References

Abeed S. and Graciela G. (2015) *Portable automatic text classification for adverse drug reaction detection via multi-corpus training*. Journal of Biomedical Informatics, 53: 196-207.

Confusion matrix. (2017, Aug). Retrieved from https://en.wikipedia.org/wiki/Confusion_matrix