**PURPOSE-LED PUBLISHING™**

**PAPER • OPEN ACCESS**

# Text-based emotion prediction system using machine learning approach

To cite this article: Ahmad Fakhri Ab. Nasir *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **769** 012022

View the article online for updates and enhancements.

The Electrochemical Society
Advancing solid state & electrochemical science & technology

**247th ECS Meeting**
Montréal, Canada
May 18-22, 2025
*Palais des Congrès de Montréal*

**Abstracts due December 6th**

**Showcase your science!**

# Text-based emotion prediction system using machine learning approach

[1]**Ahmad Fakhri Ab. Nasir, [1]Eng Seok Nee, [1]Chun Sern Choong, [1]Ahmad Shahrizan Abdul Ghani, [1]Anwar P. P. Abdul Majeed, [1]Asrul Adam and [2]Mhd. Furqan**

[1]Faculty of Manufacturing and Mechatronic Engineering Technology, Universiti Malaysia Pahang, Pekan, Malaysia
[2]Department of Computer Science, Universitas Islam Negeri Sumatera Utara Medan

**Abstract.** Text-based input becomes a common channel for humans in sharing their opinions/emotions to the product or service through online social media, shopping platform etc. Humans are easy to make errors in interpreting emotions, especially the emotion that derived from text based. The main aim of this study is to develop text-based emotion recognition and prediction system. Several market challenges facing in the advancement of emotion analysis with accuracy being the main issue. Therefore, four supervised machine learning classification algorithms such as Multinomial Naïve Bayes, Support Vector Machine, Decision Trees, and *k*-Nearest Neighbors were investigated. The model was developed based on Ekman's six basic emotions which are anger, fear, disgust, joy, guilt and sadness. Data pre-processing techniques such as stemming, stop-words, digits and punctuation marks removal, spelling correction, and tokenization were implemented. A benchmark of ISEAR (International Survey on Emotion Antecedents and Reactions) dataset was used to test all models. Multinomial Naïve Bayes classifier resulted the best performance with an average accuracy of 64.08%. Finally, the best model was integrated to graphical user interface using Python Tkinter library to complete the whole system development. Besides, the detailed performance of the best model such as *tf-idf* and count vectorizer, confusion matrix, precision-recall rate, as well as ROC (Receiver Operating Characteristic) score were also discussed. Text-based emotion prediction system to interpret and understand human emotions was successfully developed.

## 1. Introduction

Human emotions can be expressed across two channels which is via verbal (expressed emotion in words, sounds or speech) and non-verbal (includes expression through facial, gesture or body movement). The way of human-computer interaction which allow computers to interpret and understand human's emotional and attentional expressions is necessary for the computer in several applications [1]. Emotion recognition (also known as Artificial emotional intelligence) is an affective computing whereby the development of systems utilised in the process of detecting, interpreting, and predicting human emotional state such as anger, happiness, sadness, etc. Affective computing is a device for computer interacts naturally with human where "affect" refers to emotion expression and the computing that measure, identifies with, emerge from and intentionally emotion affect [2]. This kind of system may have plenty helpful applications, for example it can use to measure how pleasant the residents of the country are. Since the late 20th century, most of the government and association in this globe concerned about the happiness economics which is the study of measurement on residents' happiness advance around health, life quality, sociology and economics field. This system is also useful in suicide prevention which initially interpreting user suicidal thoughts that shared in online platforms then some

further action is taken to save the user with heavy suicidal thought [3]. Besides, emotion recognition system also applied in analysing and understanding users or customer's satisfaction toward product or services based on customer review [4]. This indirectly advantages to a company in improving company's profit.

Nowadays, the World Wide Web website has emerged into Web2.0 which allow humans to communicate with each other through social media sites (such as Facebook, Twitter, and Instagram), video sharing sites (such as YouTube) or blog. Human widely interact with computer via texts while multimodal human-computer interaction denoted to be appealing. It cannot be denied that most of the people like to share their opinion or express their emotions via online social media or comment [17-18]. Thus, expressing emotion in texts or words become common, a huge amount of data especially textual data are generated and detecting the text-based emotion plays the biggest challenge for either human or machine. Publishing negative or positive complaints on highly visible websites increases the likelihood that the public will be conscious of the customer's complaint. For example, if a person with many "followers" or "friends" published his/her opinions in a simple manner of expression on social media (such as Facebook, Instagram and Twitter), it may go "viral". Yet, a negative comment might reflect a big influence to the company reputation. For this reason, answering the complaints made by customers become a quite challenging process for company. They need to train their staff to be as possible on how to handle customer complaints, especially dealing with social media channels. In such a context, it is important to analyse and predict early stage customer's emotion of a service conversation.

The rest of the paper is organized as follows: Section 2: Related works; presented the background knowledge of textual based emotion recognition. Section 3: Methodology; portrayed the process flow, the strategy, and the components of research, including materials and methods used. Section 4: Results and discussions; explained and discussed all of the results obtained from the experiment. Section 5: Conclusions; provided the research conclusions and the recommendations for future research works.

## 2. Related works

Textual based emotion recognition and detection is a recent field of study which is familiar to sentiment analysis. Previous studies summarised that analysis on emotion detection can be divided into two types: (i) sentiment analysis and (ii) emotion analysis (or affective computing). Usually, sentiment analysis and emotion analysis are applied interchangeably. Based on the comparison between sentiment and emotion analysis done by [5] was clearly distinguished the concept for both analysis. Sentiment analysis is used to calculate and to detect the emotional polarity (positive, negative or neutral) from user input in textual based. Conversely, emotion analysis used to recognize and interpret the state of emotion expression from user input texts depends on several emotion categories. According to psychological models, human's emotion can be categorised into six basic emotions (Ekman's model) such as anger, disgust, fear, happiness, sadness and surprise [6]. Based on the collection information of previous studies on emotion recognition, Ekman emotion categories has been utilised and popular in most of the affective computing research studies and in various systems that develop to detect emotional state from text-based data [7-11].

There are several approaches that can be utilised in emotion detection: (i) lexicon-based approaches and (ii) machine learning based approaches (including supervised and unsupervised learning). Lexicon based approach (is defined as a word-dictionary) is a technique that just utilised single or multiple lexical resources to detect human emotion. Lexicon-based approaches considered as an extension work of keyword-based approaches that referred certain predefined keyword to detect the emotion in text [12-14]. Conversely, machine learning based techniques is achieved with the construction and development of algorithms to make the computer able to "learn" from data and making data driven predictions without being explicitly compute. There are two types of machine learning: (i) supervised and (ii) unsupervised.

Supervised machine learning techniques is depending on a human-labelled training dataset. It is completed via interpreted a set of labelled training data then inferred a function that mapping a new user

input to output according to sample input-output pair or training dataset [15]. A vast and structured collection of words of a labelled corpus is required annotated with emotional tags. Handling with annotation task is time-consuming and unproductive and make the approach less efficient.  However, there is some of the related studies on textual based emotion recognition in Twitter with micro-blog post where the collected corpus is labelled and annotated automatically by hashtags and existed emoticon [16-20]. Besides, some studies introduced supervised machine learning approaches in emotion detection by using Naïve Bayes classifier [21-22]. Supervised machine learning approaches can be performed by developing an emotion classifier with collected corpus from micro-blogging platform based on multi-class SVM (Support Vector Machine) kernels and analyse the sentiments followed the Ekman basic emotions model [23-24]. Other studies utilised SVM technique in analysis emotion classification, such as [9, 18, 25-26]. Moreover, there is another related work on textual based emotion recognition using supervised machine learning presented in [17, 27, 29]. On the other hand, unsupervised learning is a machine learning technique which trying to make a decision by analysing the hidden pattern in unlabelled training data in order to develop emotion classification models [11, 28]. Unlike supervised machine learning approach, input and output data are unknown and unlabelled in unsupervised learning algorithms.

Research has demonstrated the performance of supervised learning using difference types of machine learning algorithm. However, a limited or no attempts have thus far devoted to measure the performance for several machine algorithms on this field. Therefore, the current investigation also aims to evaluate the performance of four different text classifiers such as Support Vector Machine (SVM), Naïve Bayes, *k*-Nearest Neighbour (*k*-NN) and Decision Tree.

## 3. Methodology
This section shows the methodology of this work and the overall research framework is illustrated in Figure 1.
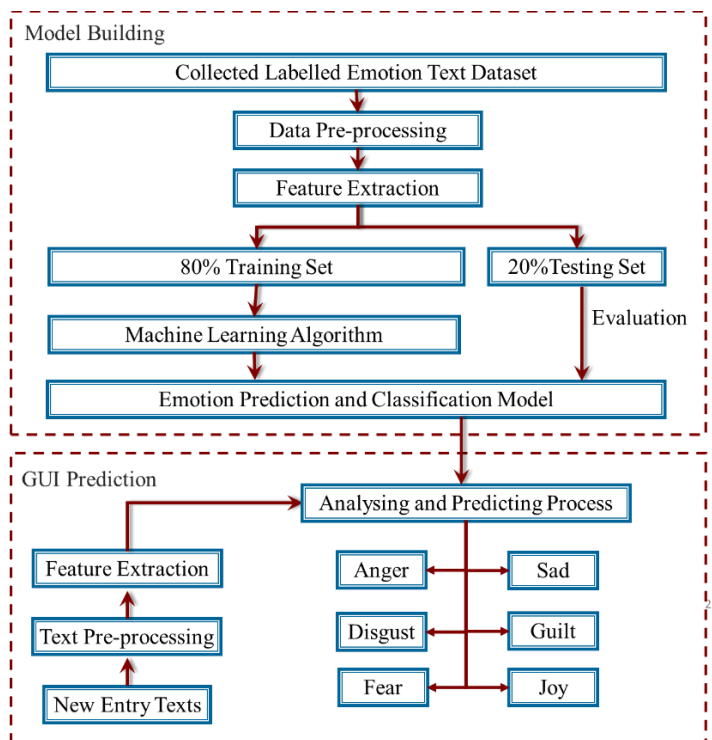


**Figure 1.** The overall framework of research methodology.

*3.1. Model of Emotions*

The emotion model created was tailored to fit with customer service applications. The emotions selected were categorized into two main sections, namely negative and positive. The negative emotions will consist of five basic emotions labelled as anger, sadness, fear, disgust and guilt and the positive emotions is happiness.

*3.2. Data Source*

Several resources were used in this project such as ISEAR (The International Survey on Emotion Antecedents and Reactions) dataset and NLTK (Natural Language Toolkits) corpus. ISEAR dataset is an emotion-labelled-datasets where the data has been cleaned and slightly normalized. It contains seven major emotions: joy, fear, anger, sadness, disgust, shame and guilt. Since ISEAR is a cleaned and established database, it was being tested for finding the best classification algorithms. Then, ISEAR also being further evaluated. Meanwhile, NLTK (Natural Language Toolkits) corpus is used as text processing libraries since in a real life, the input text is usually contains stop-words, wrong spelling, etc. Therefore, these unwanted text is needed to be removed. All data source utilised in this study is illustrated in table 1.

**Table 1.** Information of data source

| Emotion class | ISEAR |
|---------------|-------|
| Anger | 1096 |
| Disgust | 1096 |
| Sadness | 1096 |
| Fear | 1095 |
| Joy | 1094 |
| Guilt | 1093 |
| Total number | 6570 |

*3.3. Programming Language and Tools*

In this project, Python programming language is fully utilised in developing text-based emotion prediction system. It is a free, open source software which consists of modules for designing Graphical User Interface (GUI) and connecting to relational database. Moreover, it contains numerous tools and libraries which suitable to use in this project such as natural language toolkit that helped in text processing and data analysis it was frequently utilised in most of the artificial intelligent project such as building neural network or machine learning model with the help of built in libraries. The list of libraries and toolkits in Python that required in this project: (i) Pandas, (ii) Scikit-learn (sklearn), (iii) Natural Language Toolkit (NLTK), (iv) Numpy, (iv) Matplotlib, (v) TextBlob toolkits, and (vi) Tkinter (GUI toolkit)

*3.4. Model Development*

First and foremost, text dataset in csv file with the collected labelled emotion was imported and loaded using Pandas dependency. Before splitting the dataset into training and testing, the labelled emotion text undergoes data pre-processing and feature extracting processes.

*3.4.1. Data pre-processing.* (i) Stemming - every single or plural English word can be a noun, a verb, and an adjective. For example: the single word of "programmer" and plural word of "programmers" are as noun form, "program" as verb form and "programming" as adjective form which are return to stem word of "program" after stemming process. In short, stemming process is a process of removing the prefix or suffix and returned word into originate word which might not be an actual word in dictionary such as word "angry" will become "angri". Stemming is helping in minimizing database size and improving retrieval performance. (ii) Removing stop-words - the purpose of data cleaning is to remove

unwanted word that does not contain any significant emotion message in application. Stop-word is an unwanted frequent occurring word which hold needless space in database and expand the processing period. The sample of text is "you're", "you've", "you'll", "couldn't", "didn't", "doesn't", etc. (iii) Remove all the digits (0-9) and punctuation marks (such as [#"$!%&'^_` ()./:;<>=*+,-?@[]{}|\~]:). (iv) Spelling correction - the input text which contain misspelling is converted into correct word based. For example: the typing error of word "speling" undergoes correcting process and will result as "spelling". (v) Tokenization - it is a process used to split the input data or texts into pieces of single word named tokens. For example, the input text, "Hello world! This is Python programming" is changed to ["Hello", "world", "!", "This", "is", "Python", "programming", "."].

*3.4.2. Feature extracting.* Feature extraction is a process of transforming data into features which is capable used for machine learning model. Typically, machine learning algorithms programmed with numeric. Hence, the text or word is then mapped into numerical vectors feature (a series of numbers). This process is called text vectorization or "Bag of Words" representation. The features can be extract by using (i) count vectorizer - consider a straightforward method in feature extracting as it just counts the number of times a word/token appears in given document, or (ii) term frequency- inverse document frequency (tf-idf) term weighting technique in Scikit-learn machine learning library. It represents a statistical measurement method of evaluating the word significant in given document. Term frequency (*tf*) refer to how frequent a term appears in a document against total number of words in the document.

$$tf = \frac{number\ of\ occurence\ of\ a\ term\ in\ document}{total\ number\ of\ terms\ in\ the\ document} \tag{1}$$

Inverse document frequency *(idf)* is the measurement of the selected term's weight in the document. It is also explained as the log of the total number of given documents divided by the number of documents in which the selected term exists.

$$idf = \log\left(\frac{total\ number\ of\ given\ documents}{number\ of\ document\ with\ existing\ selected\ word}\right) \tag{2}$$

Both count vectorizer and *tf-idf* vectorizer consist adjustable *n*-gram size feature and *n*-gram is a consecutive word. Thus, 1-gram is just one word, also called unigram, while 2-gram is called bigram, 3-gram is called trigram and so on. While working with n-gram, all n-grams with a degree less than or equal to n are generated.

*3.4.3. Classification algorithms.*
(i) Multinomial Naïve Bayes classifier is a Naïve Bayes classifier for multinomial models which is suitable for classification with discrete features. It is a classification algorithm based on the Bayes theorem with independence assumption among predictors. This classifier learns through a document classification algorithm, and is based on a simple usage of the Bayes' rule:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \tag{3}$$

wherein,

  *c* is a class,
  *d* is a document,
  *P(c)* is a class probability,
  *P(d)* is the probability of *a* document,
  *P(d|c)* is conditional probability of the class for the given document *d*,
  *P(c|d)* is conditional probability that document *d* belongs to class *c*.

(ii) Support Vector Machine (SVM) represents a supervised machining learning algorithm in analysing data used for classification and regression analysis. SVM is defined as a discriminative classifier by

using separating hyperplane. This means that SVM execute an optimal hyperplane that maximises the margin between two classes and classifies new input data with a set of given labelled training dataset. (iii) Decision trees approach is a flow chart likely tree structure which applied the if-then-else statement technique and can be used in solving classification and regression problems. A decision tree describes as a tree where each node denoted as a feature, each branch denoted as a decision or rule and each leaf denoted as class labels or distribution. (iv) *k*-Nearest Neighbor represents a nonparametric supervised machine learning technique which able to use in classification and regression analysis. It is widely applied in the field of object or pattern recognition, text classification, data mining, and others [García-Laencina et al., 2009]. The algorithm categories an unknown input data by comparing the unknown input data with the training data to which it is high feature similarity. Then, the distance between the input features and all training data is calculated using Euclidean distance which is calculated by the square root of the sum of squared distance between two points. Euclidean distance (*d*) is calculated using equation 4, where the distance between two vectors, xA and xB, is calculated.

$$d = ((xA_0 - xB_0)^2 + (xA_1 - xB_1)^2)^{1/2} \tag{4}$$

*3.4.4. Performance measures.* The documents were tested for face detection and calculated on the basis of the confusion matrix shown in table 2 [29, 30] which contains the classifier's decisions in the rows, and the actual decision about classification in the class in the columns. The four fields of the table contain number of true positive (*TP*), true negative (*TN*), false positive (*FP*) and false negative (*FN*) classified documents.

**Table 2.** Confusion matrix

|  |  | actual | |
|---|---|---|---|
|  |  | YES | NO |
| predicted | YES | TP | FP |
|  | NO | FN | TN |

(i) Accuracy is expressed as the proportion of correctly classified cases over all cases and is calculated according to formula:

$$accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{5}$$

(ii) Precision is expressed as the proportion of positive cases that are correctly recognized as positive over all cases classified as positive and is calculated according to the formula:

$$precision = \frac{TP}{(TP+FP)} \tag{6}$$

(iii) Recall is expressed as the proportion of positive cases that are correctly recognized as positive over all actual positive cases and is calculated according to the formula:

$$\text{recall} = \frac{TP}{(TP+FN)} \tag{7}$$

(iv) A measure that combines precision and recall is called the $F_1$-measure and represents their weighted harmonic mean. It is calculated according to the formula:

$$F_1 = \frac{(2 \times precision \times recall)}{(precision+recall)} \tag{8}$$

(v) The last evaluation measure to be mentioned here is the ROC (Receiver Operating Characteristic) curve. It is a graphical representation of the binary classifier performance on which the curve represents a compromise between true positive and false positive cases.

## 4. Results and discussions

### 4.1. Four classification algorithms analysis

In first experiment, four classification algorithms were trained. Training score was collected by splitting dataset to 80% for training set and 20% for testing set. Meanwhile, cross-validation (CV) score was obtained using 5-CV. All setup for classification algorithms are based on Python sklearn library default parameters. The recognition accuracy results of four classification algorithm is as shown in Figure 2. Theoretically, the score in learning curve is used to describe how good the model is. If the training score is higher, the model is performed better. Based on Figure 2, the learning curve of SVM showed the lowest and unstable training score compare to others algorithms which is around 18.8% only with 500 training examples. By increasing the number of training examples, the accuracy of training score decreases. Same goes to the $k$-NN which showed a low training score (around 0.65 score). Thus, SVM and $k$-NN algorithms can be said not suitable in this textual emotion prediction project as both classifiers represented low performance and executed a bad learning curve with high bias, poor fit and poor generation in training a textual emotion prediction model.

The learning curve of Multinomial Naïve Bayes and Decision Tree classifier showed training score at maximum regardless of training examples. Besides, the CV score (marked with green line) increases with increasing of the training examples. The learning curve of Decision Tree showed bigger gap between CV score and training score compared to Multinomial Naïve Bayes, therefore it indicates that Multinomial Naïve Bayes classifier is proven to have a better performance. However, the training score is slightly dropped on increasing of training examples may cause due to mislabelled examples or duplicate example with different label. To conclude the analysis of the classification algorithms selection, an average mean score is calculated based on results from Figure 2. Multinomial Naïve Bayes classifier illustrated a better performance with highest accuracy mean score (64.08%) compared to SVM (15.37%), Decision Tree classifiers (52.36%) and k-NN (47.95%). Thus, Multinomial Naïve Bayes classifier is selected as the machine learning algorithm in this project.
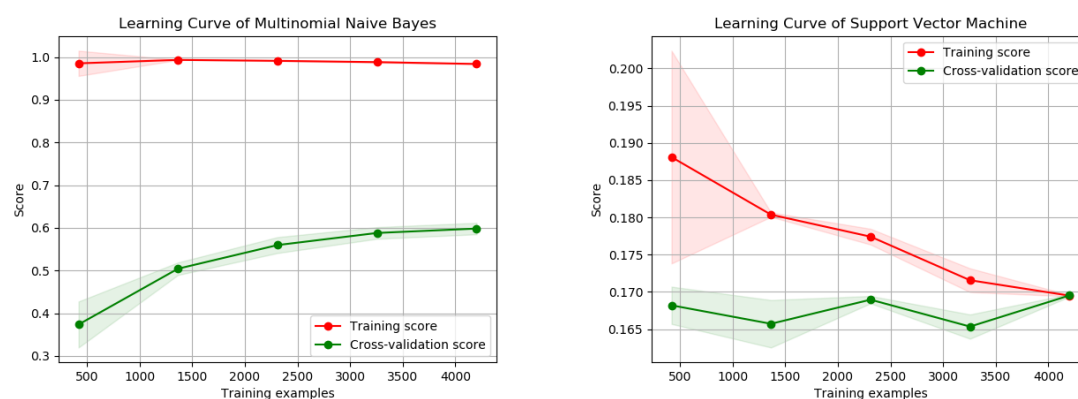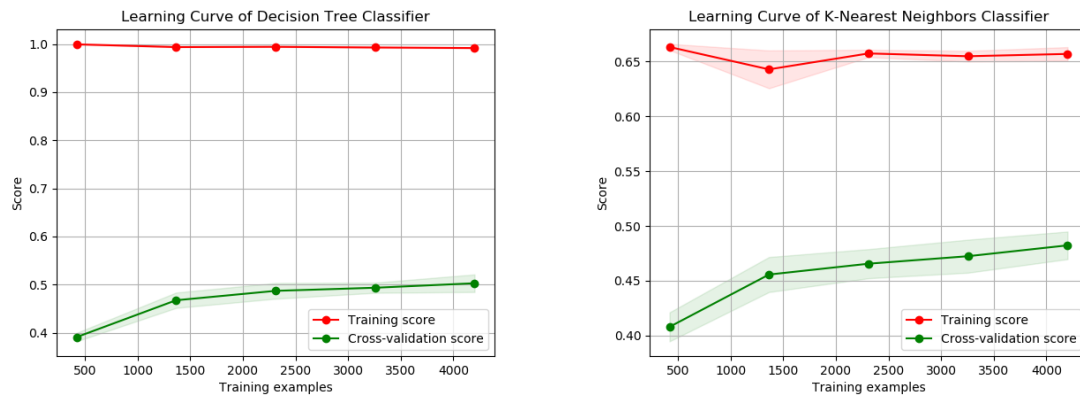
**Figure 2.** Learning curve for four classification algorithms.

*4.2. The effect of feature extraction method on recognition accuracy*

Figure 3 shows the comparison of recognition accuracy for two feature extraction methods on Multinomial Naïve Bayes algorithm. The accuracy score line of *tf-idf* vectorizer (blue line) is slightly above the accuracy score line of count vectorizer (red line). This indicates that the *tf-idf* vectorizer helps in improving the performance of the model by executing a higher recognition accuracy compared to count vectorizer. Besides that, the n-gram size feature can influence the result of recognition accuracy. The features extracted with trigram (*n*=3) from the document showed the highest recognition accuracy achieved is >64%. Hence, *tf-idf* vectorizer is selected as feature extraction method for this implementation.
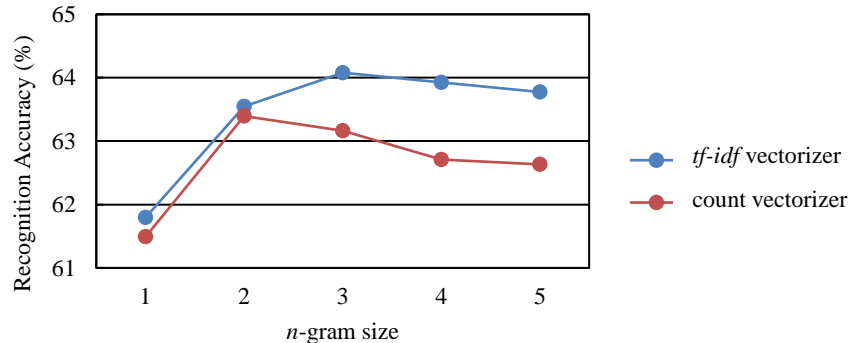


**Figure 3.** Recognition accuracy score comparison between *tf-idf* vectorizer (blue marked) and count vectorizer (red marked) with different size of n-gram.

*4.3. The best model detailed analysis*

Figure 4 shows the detailed performance of the developed model using confusion matrix, precision-recall rate, and ROC score. As previously mentioned, the dataset with 6570 sentences were split into 80% for training set and 20% for testing set. Thus, there is total 1314 testing data were picked-up randomly to measure the performance of the final model.

Precision is a measure of a result relevance while recall is a measure of how many relevant results are returned. The area under precision-recall curve (denoted as area in Figure 4) represent an average precision and it summarised the skill of a model across thresholds. Based on precision-recall results, that class 'fear', 'joy', and 'sadness' showed up more than 70% large of area under curve compared to 'anger', 'disgust', and 'guilt' classes. This indicates that 'anger', 'disgust' and 'guilt' have higher misclassification rate and overlapped each other. This is supported by confusion matrix table whereby the percentage of misclassification between 'anger', 'disgust' and 'guilt'.
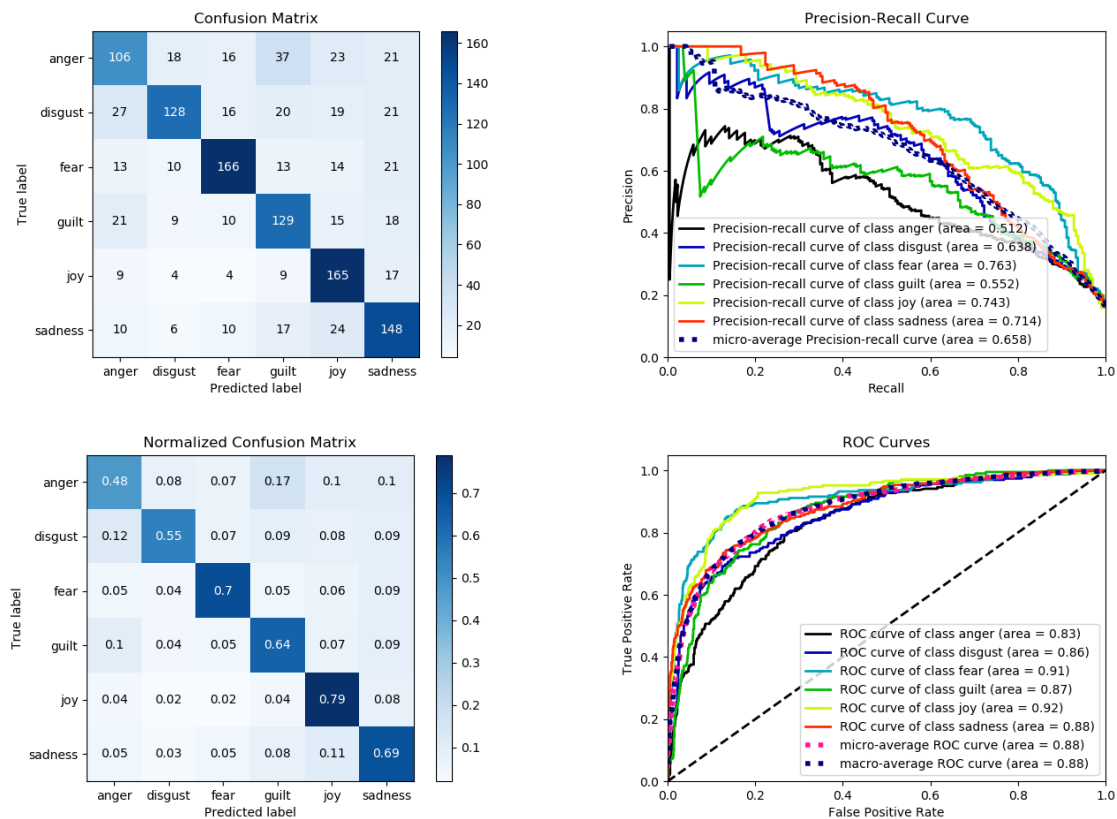
**Figure 4.** The detailed performance evaluations on the final model based on confusion matrix, precision-recall rate, and ROC score.

### 4.4. The Graphical User Interface

The design of the GUI is as shown in Figure 5. In GUI, user is allowed to insert English based message in the message entry box (beside "Customer Messages" text). In this example, the entry message is "Predict comes in good packaging. A new and solid plastic rack. Will buy again". Then, "Analyse" button is clicked to start the prediction process. The input texts undergo text pre-processing and feature extracting process, then it fitted into the final predictive model to predict the text's emotion classes by calculating the probability. Each probability of emotion classes is calculated and shown by using emoji stickers. The probabilities of emotion classes are converted into percentage for easy comparison. An emotion with a highest percentage is predicted as the estimate customer's emotion and the final emotion is displayed in the label box below (refer to Figure 5). After analysing process, the message is predicted as 'JOY' as the percentage of emotion joy is showed the highest (96.71%) compare to others (anger = 0.6%, sad = 2.62%, fear = 0.02%, disgust = 0.04% and guilt = 0.0%).
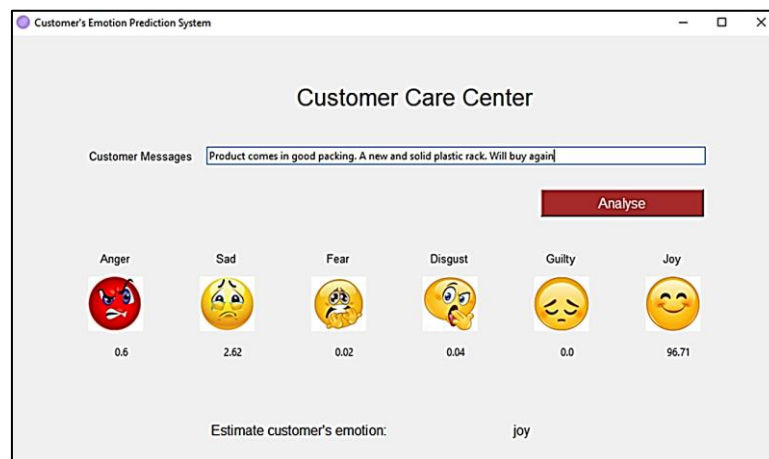
**Figure 5.** GUI layout for text-based emotion prediction system.

## 5. Conclusions

This paper presents the comparison of classifier's performance among four machine learning algorithms, Multinomial Naïve Bayes, Support Vector Machine, Decision Tree Classifier and *k*-Nearest Neighbors Classifier. Multinomial Naïve Bayes classifier results in a better performance and thus it was selected and proposed to develop a textual emotion prediction model. Besides, the investigation on feature extraction method revealed that *tf-idf* with 3-gram size is better for ISEAR dataset. An English text-based emotion prediction system with a simple graphical user interface for customer's emotion prediction from text system was successfully developed. The system is mainly focusing on analysing the Ekman's six basic emotions (anger, fear, disgust, joy, guilt and sadness). Although this is just only a base model, however the developed prediction system has a prospective future and its accuracy relies upon the type of classification algorithm applied, correct labelled emotional text database, data cleaning, and feature extraction method. Emotion texts might appear in complex sentence or called mix-emotion (examples: angry-sad, happy-angry) elements. To solve this problem, it is suggested to add some features or rules-based approaches in future work which might help in detecting emotions from text more correctly and perform better than the present result.

## 6. References

[1]    Sebe, N., Cohen, I., Gevers, T., & Huang, T. S. (2005). Multimodal approaches for emotion recognition: a survey. Paper presented at the Internet Imaging VI
[2]    Picard, R. W. (2003). Affective computing: challenges. Int. J. Hum.- Comput. Stud., 59(1-2), 55-64. doi:10.1016/s1071-5819(03)00052-1
[3]    Desmet, B., & Hoste, V. (2013). Emotion detection in suicide notes. Expert Systems with Applications, 40(16), 6351-6358. doi:10.1016/j.eswa.2013.05.050
[4]    Bafna, K., & Toshniwal, D. (2013). Feature based Summarization of Customers' Reviews of Online Products. Procedia Computer Science, 22, 142-151. doi:10.1016/j.procs.2013.09.090
[5]    Munezero, M. D., Montero, C. S., Sutinen, E., & Pajunen, J. (2014). Are They Different? Affect, Feeling, Emotion, Sentiment, and Opinion Detection in Text. IEEE Transactions on Affective Computing, 5(2), 101-111. doi:10.1109/TAFFC.2014.2317187
[6]    Ekman, P. (1992). An Argument For Basic Emotions (Vol. 6)
[7]    Agrawal, A., & An, A. (2012, 4-7 Dec. 2012). Unsupervised Emotion Detection from Text Using Semantic and Syntactic Relations. Paper presented at the 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology
[8]    Chen, Y.-L., Chang, C.-L., & Yeh, C.-S. (2017). Emotion classification of YouTube videos. Decision Support Systems, 101, 40-50. doi:10.1016/j.dss.2017.05.014
[9]    Li, W., & Xu, H. (2014). Text-based emotion classification using emotion cause extraction. Expert Systems with Applications, 41(4), 1742-1749. doi:10.1016/j.eswa.2013.08.073

[10]   Muljono, Winarsih, N. A. S., & Supriyanto, C. (2016, 5-6 Aug. 2016). Evaluation of classification methods for Indonesian text emotion detection. Paper presented at the 2016 International Seminar on Application for Technology of Information and Communication (ISemantic)

[11]   Yasmina, D., Hajar, M., & Hassan, A. M. (2016). Using YouTube Comments for Text-based Emotion Recognition. Procedia Computer Science, 83, 292-299. doi:10.1016/j.procs.2016.04.128

[12]   Chatzakou, D., Vakali, A., & Kafetsios, K. (2017). Detecting variation of emotions in online activities.     Expert     Systems     with     Applications,     89,     318-332. doi:https://doi.org/10.1016/j.eswa.2017.07.044

[13]   Rabeya, T., Ferdous, S., Ali, H. S., & Chakraborty, N. R. (2017, 22-24 Dec. 2017). A survey on emotion detection: A lexicon based backtracking approach for detecting emotion from Bengali text. Paper presented at the 2017 20th International Conference of Computer and Information Technology (ICCIT)

[14]   Shivhare, S. N., & Sri Khetwat, S. (2014). Emotion Detection From Text Documents (Vol. 4)

[15]   Russell, S. J., & Norvig, P. (2016). Artificial intelligence: a modern approach: Malaysia; Pearson Education Limited

[16]   Bandhakavi, A., Wiratunga, N., Deepak, P., & Massie, S. (2014). Generating a word-emotion lexicon from# emotional tweets. Paper presented at the Proceedings of the Third Joint Conference on Lexical and Computational Semantics (* SEM 2014)

[17]   Hasan, M., Rundensteiner, E., & Agu, E. (2014). EMOTEX: Detecting Emotions in Twitter Messages

[18]   Roberts, K., A Roach, M., Johnson, J., Guthrie, J., & M Harabagiu, S. (2012). EmpaTweet: Annotating and Detecting Emotions on Twitter

[19]   Suttles, J., & Ide, N. (2013). Distant Supervision for Emotion Classification with Discrete Binary Values (Vol. 7817)

[20]   Wang, W., Chen, L., Thirunarayan, K., & Sheth, A. P. (2012, 3-5 Sept. 2012). Harnessing Twitter "Big Data" for Automatic Emotion Identification. Paper presented at the 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing

[21]   Krishnan, H., Elayidom, M. S., & Santhanakrishnan, T. (2017). Emotion Detection of Tweets using Naïve Bayes Classifier. Emotion.

[22]   Strapparava, C., & Mihalcea, R. (2008). Learning to identify emotions in text.

[23]   Cbalabantaray, R., mohd, M., & Sharma, N. (2012). Multi-Class Twitter Emotion Classification: A New Approach (Vol. 4)

[24]   Li, J., Xu, Y., Xiong, H., & Wang, Y. (2010, 16-17 Aug. 2010). Chinese text emotion classification based on emotion dictionary. Paper presented at the 2010 IEEE 2nd Symposium on Web Society

[25]   Luyckx, K., Vaassen, F., Peersman, C., & Daelemans, W. (2012). Fine-grained emotion detection in suicide notes: a thresholding approach to multi-label classification. Biomedical informatics insights, 5(Suppl. 1), 61-69. doi:10.4137/BII.S8966

[26]   Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. Support Vector Regression Machines(7)

[27]   Perikos, I., & Hatzilygeroudis, I. (2016). Recognizing emotions in text using ensemble of classifiers.     Engineering     Applications     of     Artificial     Intelligence,     51,     191-201. doi:10.1016/j.engappai.2016.01.012

[28]   Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundations of Machine Learning: The MIT Press.

[29]   Karnalim, O. (2014). Language-Agnostic Source Code Retrieval Using Keyboard & identifier Lexical Pattern. International Journal of Software Engineering and Computer Systems, 4(1), 29-47