Social Media(Twitter) Emotion Analysis

Manish Chugani (151070014)
Prafull Parmar (151070025)
Vatsal Mankodi (151070035)

PROBLEM STATEMENT

Online social networks are currently a major way through which individuals interact on the Internet. The use of social network sites (SNSs) has grown dramatically across the world. Young people across many factors (gender, ethnicity, socioeconomic status etc.) appear to participate in online social networks.

Thus, social media platforms have become a very important way to judge the emotions and sentiment for the populace on a particular topic.

The attempt here is to use Twitter as the social media platform to gauge emotions, that is classify the among the following emotion classes:

1. Happiness
2. Sadness
3. Anger
4. Surprise
5. Neutrality

Input:

A trending topic/hashtag on Twitter as user input

Output:

- A sample classification of 2,000 tweets on the entered keyword(i.e. Hashtag)
- A pie chart summing up the sentiments on each of those 2,000 related to the topic using a set of 10,000 sample tweets as a training set for the Learner.

Use Case:

The model can be a tool for various marketing and sales analyzers to gauge customer or public reaction on policies, products and decisions…..

DATA SET DESCRIPTION

Train Dataset:

The dataset is imported and stored in two “ndarrays”(Lists in essence):

- tweet content

- emotion

Testing Data:

Realtime tweets imported during execution using the Twitter Python API - tweepy. The tweets are imported based on a user entered query and are analyzed by the program to sum up the emotion outpour on the particular topic.
-

Tweepy:

Tweepy supports accessing Twitter via Basic Authentication and the newer method, OAuth. Twitter has stopped accepting Basic Authentication so OAuth is now the only way to use the Twitter API.

OAuth authentication consists of consumer and access keys and is a bit more complicated, since it requires more effort, but the benefits it offers are very lucrative:

- Tweets can be customized to have a string which identifies the app which was used.
- It doesn't reveal user password, making it more secure.
- It's easier to manage the permissions, for example a set of tokens and keys can be generated that only allows reading from the timelines, so in case someone obtains those credentials, he/she won't be able to write or send direct messages, minimizing the risk.
- The application doesn't reply on a password, so even if the user changes it, the application will still work.

After logging in to the portal, and going to "Applications", a new application can be created which will provide the needed data for communicating with Twitter API.

DESCRIPTION OF TEXTUAL PRE-PROCESSING

Extracting Tweets based on a hashtag:

Code:

```
def get_tweets(hashtag):

    # Authorization to consumer key and consumer secret

    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)

    # Access to user's access key and access secret
```

```
    auth.set_access_token(access_key, access_secret)

    # Calling api

    api = tweepy.API(auth)

    # 2000 tweets to be extracted

    number_of_tweets = 2000

    #tweets = api.user_timeline(screen_name=username, count=number_of_tweets)

    tweets = []

    for tweet in tweepy.Cursor(api.search, q="#" + hashtag + "",lang="en",
since="1997-01-01").items(number_of_tweets):

        tweets.append(tweet)

    # Empty Array

    tmp=[]

     # create array of tweet information: username,

    tweets_for_csv = [tweet.text for tweet in tweets]

    for j in tweets_for_csv:

        tw = clean_tweet(j)

        tw = tw.lower()

        tw = tw.split(" ")

        tmp.append(tw)

    return tmp
```

Cleaning the extracted tweets:

For example,

1. original tweet: @peter I really love that shirt at #Macy. http://bet.ly//WjdiW4
    - processed tweet: I really love that shirt at Macy
2. original tweet: @shawn Titanic tragedy could have been prevented Economic Times: Telegraph.co.uk Titanic tragedy could have been preve... http://bet.ly/tuN2wx

    - processed tweet: Titanic tragedy could have been prevented Economic Times Telegraph co uk Titanic tragedy could have been preve
3. original tweet: I am at Starbucks http://4sh.com/samqUI (7419 3rd ave, at 75th, Brooklyn)
    - processed tweet: I am at Starbucks 7419 3rd ave at 75th Brooklyn

Code:

```
def clean_tweet(tweet):
    '''
    Utility function to clean tweet text by removing links, special characters
    using simple regex statements.
    '''
    return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", tweet).split())
```

Converting Text to a Learnable Dataset:

TFidFVectorizer

In information retrieval, **tf–idf** or **TFIDF**, short for **term frequency–inverse document frequency**, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf-idf value increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Nowadays, tf-idf is one of the most popular term-weighting schemes; 83% of text-based recommender systems in the domain of digital libraries use tf-idf.

Variations of the tf–idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. tf–idf can be successfully used for stop-words filtering in various subject fields, including text summarization and classification.

One of the simplest ranking functions is computed by summing the tf–idf for each query term; many more sophisticated ranking functions are variants of this simple model.

Motivations

Term frequency

Suppose we have a set of English text documents and wish to rank which document is most relevant to the query, "the brown cow". A simple way to start out is by eliminating documents that do not contain all three words "the", "brown", and "cow", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document; the number of times a term occurs in a document is called its *term frequency*. However, in the case where the length of documents varies greatly, adjustments are often made (see definition below). The first form of term weighting is due to Hans Peter Luhn (1957) which may be summarized as:

- The weight of a term that occurs in a document is simply proportional to the term frequency.

Inverse document frequency

Because the term "the" is so common, term frequency will tend to incorrectly emphasize documents which happen to use the word "the" more frequently, without giving enough weight to the more meaningful terms "brown" and "cow". The term "the" is not a good keyword to distinguish relevant and non-relevant documents and terms, unlike the less-common words "brown" and "cow". Hence an *inverse document frequency* factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

Karen Spärck Jones (1972) conceived a statistical interpretation of term specificity called Inverse Document Frequency (IDF), which became a cornerstone of term weighting:

- The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.

Definition

The tf–idf is the product of two statistics, term frequency and inverse document frequency. Various ways for determining the exact values of both statistics exist.

Term frequency

In the case of the **term frequency** tf(t,d), the simplest choice is to use the *raw count* of a term in a document, i.e. the number of times that term t occurs in document d. If we denote the raw count by $f_{t,d}$, then the simplest tf scheme is tf(t,d) = $f_{t,d}$. Other possibilities include[5]:128

- Boolean "frequencies": tf(*t*,*d*) = 1 if *t* occurs in *d* and 0 otherwise;
- term frequency adjusted for document length : $f_{t,d}$ ÷ (number of words in d)
- logarithmically scaled frequency: tf(*t*,*d*) = log (1 + $f_{t,d}$);[6]
- augmented frequency, to prevent a bias towards longer documents, e.g. raw frequency divided by the raw frequency of the most occurring term in the document:

$$\mathrm{tf}(t,d) = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}}$$

Inverse document frequency

The **inverse document frequency** is a measure of how much information the word provides, that is, whether the term is common or rare across all documents. It is the logarithmically scaled inverse fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Then tf–idf is calculated as

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D)$$

A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. Since the ratio inside the idf's log function is always greater than or equal to 1, the value of idf (and tf-idf) is greater than or equal to 0. As a term appears in more documents, the ratio inside the logarithm approaches 1, bringing the idf and tf-idf closer to 0.

Code:

```
from sklearn.feature_extraction.text import TfidfVectorizer

cv = TfidfVectorizer()

X = cv.fit_transform(corpus).toarray()
```

Here corpus is a list of list of words(a list of sentences, each split into its constituent words). The vectorizer returns float values(vectors, i.e. points on a 2-D plane) for each term based on TF-IDF.

Plotting a Pie chart

Matplotlib pie chart
The code below creates a pie chart:

```
import matplotlib.pyplot as plt

# Data to plot
labels = 'Happy', 'Sad', 'Angry', 'Surprise', 'Neutral'

sizes = [happy, sad, angry, surprise, neutral]

colors = ['gold', 'lightskyblue', 'red', 'orange', 'grey']

# Plot

plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True,
startangle=0)

plt.axis('equal')

plt.show()
```

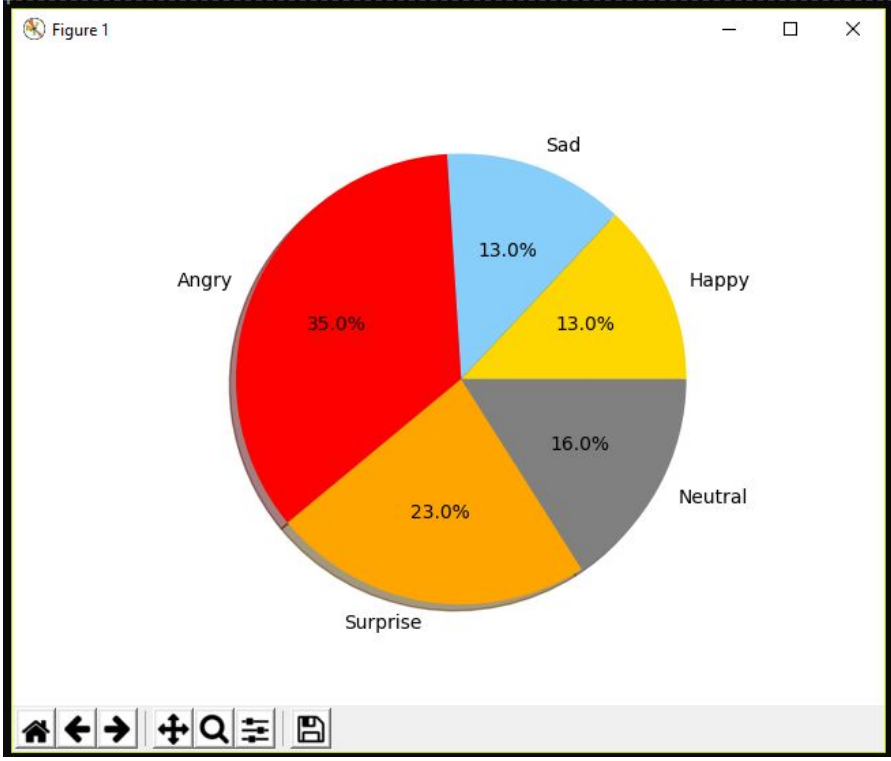
OUTPUT:

```
C:\WINDOWS\system32\cmd.exe - py  -3 nlp.py
Microsoft Windows [Version 10.0.16299.371]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\manis>cd C:\Users\manis\Desktop\Manish\VJTI Computer Engineering\Third Year\Sem 6\MLProject

C:\Users\manis\Desktop\Manish\VJTI Computer Engineering\Third Year\Sem 6\MLProject>py -3 nlp.py
Enter the hashtag of which you want to search Tweets
Asifa
```

Tweet	Emotion
listen to jkpcc president ga mir who too suspects kathua investigation is more revelations are t	surprise
this man rajith ram named his new born baby asifa s raj this is kerala for you this is d reason y it took 65 years for sang	sad
its not just asifa alll these women and their families need help too lets not highlight just one case lets tr	sad
pic of the day boy kicks the police vehicle during the protest in baramulla against asifa who was raped and killed in khutta	angry
pic of the day boy kicks the police vehicle during the protest in baramulla against asifa who was raped and killed in khutta	angry
who killed asifa did rohingya killed asifa if no why pdp gov t don t want cbi inquiry is	surprise
bjp leader lal singh who resigned from jammu amp kashmir state cabinet leads rally demanding cbi probe into kathu	angry
bjp leader lal singh who resigned from jammu amp kashmir state cabinet leads rally demanding cbi probe into kathu	angry
707070 who spoke up rashid all actors with the same message on their placard got paid by one single company the fac	angry
what is up to calls someone who demands cbi enquiry pro rapists this little girl asifa needs justice usin	neutral



ALGORITHM USED - Naive Bayes

Naive Bayes Algorithm

What is Naive Bayes algorithm?

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

Above,

P(c|x) is the posterior probability of class (c,target) given predictor (x,attributes).

P(c) is the prior probability of class.

P(x|c) is the likelihood which is the probability of predictor given class.

P(x) is the prior probability of predictor.

Gaussian Naive Bayes

A Gaussian Naive Bayes algorithm is a special type of NB algorithm. It's specifically used

when the features have continuous values. It's also assumed that all the features are following a gaussian distribution i.e, normal distribution.

Multinomial Naive Bayes

It is a specialized version of Naive Bayes that is designed more for text documents. Whereas simple naive Bayes would model a document as the presence and absence of particular words, multinomial naive bayes explicitly models the word counts and adjusts the underlying calculations to deal with in.

Pros and Cons of Naive Bayes

Pros:

It is easy and fast to predict class of test data set. It also perform well in multi class prediction

When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.

It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Cons:

If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as "Zero Frequency". To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called

Laplace estimation.

On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously.

Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Applications of Naive Bayes Algorithms

Real time Prediction:Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.

Multi class Prediction:This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.

Text classification/ Spam Filtering/ Sentiment Analysis:Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)

Recommendation System:Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

Multinomial Naïve Bayes algorithm is used as a classifier to classify the reviews as Deceptive/True.

Let's assume that we want to infer a classification. Denoting the classes as C_i and any relevant features F, the probability of a given class is given by Bayes Theorem,

$$p\left(C_i|\vec{F}\right) = \frac{p\left(\vec{F}|C_i\right)p(C_i)}{\sum_j p\left(\vec{F}|C_j\right)p(C_j)}$$

Now all we need to do is model the class likelihoods, $p\left(\vec{F}|C_i\right)$.

The Naive Bayes assumption is that the features are independent given a class, i.e.

$$p\left(\vec{F}|C_i\right) = \prod_j p(F_j|C_i)$$

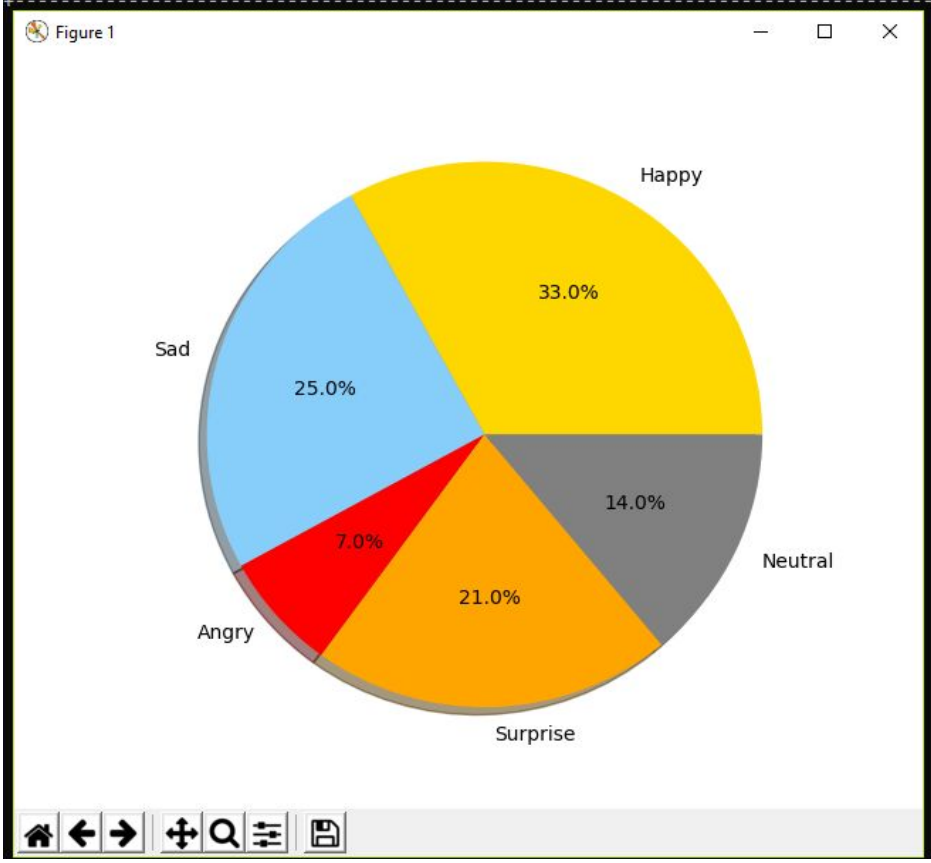
Popular choices of the $p(F_j|C_i)$ include the Bernoulli distribution (taking into account whether a binary feature occurs or not) and the Binomial distribution (taking into account not just the presence of a binary feature but also its multiplicity).

When a feature is discrete but not binary (or continuous but approximated by such discrete values) a common choice of likelihood is the multinomial distribution, hence Multinomial Naïve Bayes. This is a direct generalization of the Binomial Naive Bayes model.

RESULTS AND OBSERVATIONS:

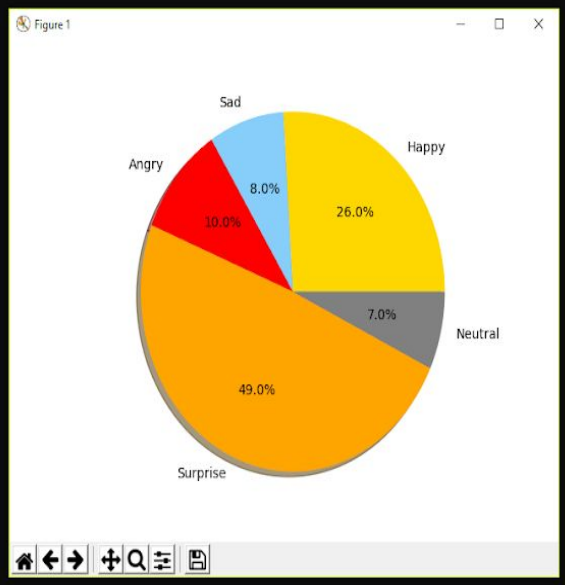
C:\WINDOWS\system32\cmd.exe - py -3 nlp.py

```
C:\Users\manis\Desktop\Manish\VJTI Computer Engineering\Third Year\Sem 6\MLProject>py -3 nlp.py
Enter the hashtag of which you want to search Tweets
MUFC
+-----------------------------------------------------------------------------------------------------------------------+---------+
| Tweet                                                                                                                 | Emotion |
+-----------------------------------------------------------------------------------------------------------------------+---------+
| brilliant mufc                                                                                                        | neutral |
| who do you think will be first to leave mufc rt mourinho pogba                                                        |  happy  |
| manchester united getting ready to cheer you on against afc bournemouth match info mufc boumnu                       |  happy  |
| ruscoe big debate who are the bigger club right now rt manchester united fav liverpool mufc liverpool                 |  happy  |
| programme officiel du tour2018 19 07 club america phoenix stadium arizona 22 07 san jose levi s stadium santa        |   sad   |
| earn 45 today bookiebashing now see free sign up mufc lfc bets london bets rt                                         |   sad   |
| na the best goal wey ronaldo don score for hin life rocker launcher ucl mufc                                          |   sad   |
| who do you think will be first to leave mufc rt mourinho pogba                                                        |  happy  |
| crouch on the issue of safe standing at football grounds you claimed that there is no desire amongst the t            |   sad   |
| mufc executive vice chairman ed woodward on this summer s american tour the facilities venues and fixtures will give th | neutral |
+-----------------------------------------------------------------------------------------------------------------------+---------+
```



```
C:\Users\manis\Desktop\Manish\VJTI Computer Engineering\Third Year\Sem 6\MLProject>py -3 nlp.py
Enter the hashtag of which you want to search Tweets
IPL
```

Tweet	Emotion
ipl contest can you spot the ball play it here ipl ipl2018	happy
kohli striking with 115 strike rate when they need 16 an over fixed ipl mivrcb	happy
kohli overtakes raina to become the highest run scorer in ipl history a stupendous achievement	surprise
kollywood strike ends releases may happen so time for seeman bharathiraja velmurugan amp gouthaman to say no film relea	happy
jaanu most runs in ipl history viratkohli 4559 sureshraina 4558 rohitsharma 4345 gautamgambhir 4210 davidwarner 40	surprise
kohli is looking for a reliable partner at other end unfortunately anoushka does not play cricket rcbvsmi rcb ipl	angry
virat kohli 4559 now leading run getter in ipl history bettered suresh raina 4558 mivrcb ipl2018	surprise
virat kohli 4559 now leading run getter in ipl history bettered suresh raina 4558 mivrcb ipl2018	surprise
make helmets compulsory for keepers batsmens and close fielders its a serious issue not funny kkkrvdd ipl	surprise
really feel bad for kohli here rest are falling like nine pins while he stands and watches from the other end mivsrcb ipl vivoipl2018	sad



It is observed that every time the program is run, different scores are obtained. This is because every time the program is being executed, the Multinomial Naïve Bayes algorithm creates a new model based on the input review and new scores are generated.

CONCLUSION & FUTURE SCOPE:

Sentiment analysis is a set of methods, typically (but not always) implemented in computer software, that detect, measure, report, and exploit attitudes, opinions, and emotions in online, social, and enterprise information sources. (As an aside, what makes it “analysis” is that you’re doing it systematically, with some goal in mind.)

Sentiment analysis much more than simplistically subtracting the number of “negative” words from the number of “positive” in a document or message in order to produce a score.It involves social conversations and also direct and indirect feedback (such surveys, contact-center notes, and warranty and insurance claims), online news, presentations, even scientific papers: Any information source that captures subjective information.

Sentiment analysis lets marketers (and market researchers, customer service and support staff, product managers, etc.) get at root causes, at explanations of behaviors that are captured in transaction and tracking records. Sentiment analysis means better targeted marketing, faster detection of opportunities and threats, brand-reputation protection, and the ultimate aim, profit.

Affective computing (sometimes called **artificial emotional intelligence**, or **emotion AI**) is the study and development of systems and devices that can recognize, interpret, process, and simulate human affects(emotions). It is an interdisciplinary field spanning computer science, psychology, and cognitive science. While the origins of the field may be traced as far back as to early philosophical inquiries into emotion, the more modern branch of computer science originated with Rosalind Picard's 1995 paper on affective computing. A motivation for the research is the ability to simulate empathy. The machine should interpret the emotional state of humans and adapt its behavior to them, giving an appropriate response to those emotions.

The difference between sentiment analysis and affective analysis is that the latter detects the different emotions instead of identifying only the polarity of the phrase.

In e-learning applications, affective computing can be used to adjust the presentation style of a computerized tutor when a learner is bored, interested, frustrated, or pleased.Psychological health services, i.e. counseling, benefit from affective computing applications when determining a client's emotional state.

Robotic systems capable of processing affective information exhibit higher flexibility while one works in uncertain or complex environments. Companion devices, such as digital pets, use affective computing abilities to enhance realism and provide a higher degree of autonomy.

Social robots, as well as a growing number of robots used in healthcare benefit from emotional awareness because they can better judge users' and patient's emotional states and alter their actions/programming appropriately. This is especially important in

those countries with growing aging populations and/or a lack of younger workers to address their needs.

Other potential applications are centered around social monitoring. For example, a car can monitor the emotion of all occupants and engage in additional safety measures, such as alerting other vehicles if it detects the driver to be angry. Affective computing has potential applications in human-computer interaction, such as affective mirrors allowing the user to see how he or she performs; emotion monitoring agents sending a warning before one sends an angry email; or even music players selecting tracks based on mood.

One idea put forth by the Romanian researcher Dr. Nicu Sebe in an interview is the analysis of a person's face while they are using a certain product (he mentioned ice cream as an example). Companies would then be able to use such analysis to infer whether their product will or will not be well received by the respective market.

One could also use affective state recognition in order to judge the impact of a TV advertisement through a real-time video recording of that person and through the subsequent study of his or her facial expression. Averaging the results obtained on a large group of subjects, one can tell whether that commercial (or movie) has the desired effect and what the elements which interest the watcher most are.

Affective computing is also being applied to the development of communicative technologies for use by people with autism.