ABSTRACT:

In the current phase of social media everyone are free to express their self. People's comment or reaction on a particular object justifies the opinion and the sentiments of the person. This feelings sometimes can be expressed in simple sentence order in a complex sentence.

Introducing a novel approach for classifying the the sentiments of the the comments what people make either on a object or to express themselves. This sentiments for the statement are classified as happy, sad or angry with respect to the query term.

INTRODUCTION:

In this Project, we'll develop a **Sentiment Analysis model** to categorize a statement as HAPPY, ANGRY or SAD

This is very useful because it allows feedback to be aggregated without manual intervention. Here we use results of machine learning and natural language processing algorithms for classifying the sentiment. Our training data consists of comments with their TAG as happy , angry and sad.

Sentiment Analysis is the process of 'computationally' determining whether a piece of writing is positive, negative or neutral. It's also known as **opinion mining**, deriving the opinion or attitude of a speaker.

PROPOSED SYSTEM:

The proposed system uses naive Bayes implementations: Bernoulli, multinational and Gaussian, and compared with their accuracy values. Later we impalement Neural Networks to find the emotion of the statement and accuracy for the emotion. The feature extraction procedure is implemented through browsing the data-set, cleaning and organizing, renaming the columns, deleting the unnecessary columns in the data-set. The trained data is obtained through splitting data-set into 2 splits train set and test set respectively.

The trained classifier models are used to classify the tweets in to positive and negative. The performance rate is measured through accuracy. Each of the algorithms is tested with their accuracy to determine the best model among all the other classifiers.

A. Import the libraries and data-set

The data set contains 635 statements stating Sad emotions, 708 statement stating happy emotions and 696 statements stating angry emotions.

B. Cleaning and visualizing the data-set

Text Preprocessing is traditionally an important step for Natural Language Processing (NLP) tasks. It transforms text into a more digestible form so that machine learning algorithms can perform better.

Lower Casing: Each text is converted to lowercase.

Replacing URLs: Links starting with "http" or "https" or "www" are replaced by "URL".

Replacing Emojis: Replace emojis by using a pre-defined dictionary containing emojis along with their meaning. (eg: ":)" to "EMOJIsmile")

Replacing Usernames: Replace @Usernames with word "USER". (eg: "@Kaggle" to "USER")

Removing Non-Alphabets: Replacing characters except Digits and Alphabets with a space.

Removing Stopwords: Stopwords are the English words which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. (eg: "the", "he", "have")

Lemmatizing: Lemmatization is the process of converting a word to its base form. (e.g: “Great” to “Good”)

C. Splitting the dataset into train and test

The Preprocessed Data is divided into 2 sets of data:

Training Data: The dataset upon which the model would be trained on. Contains 70 % data.Test Data: The dataset upon which the model would be tested against. Contains 30% data.After splitting the shape of each set can be viewed. Each model will be trained based on the split data to understand the degree of accuracy. **TF- IDF indicates what the importance of the word is in order to understand thedocument or dataset.**

PERFORMANCE EVALUATION:

METHODS	ACCURACY
Gaussian	47.26890756302521 %
Multinomial	67.43697478991597 %
Bernaulli	65.96638655462185 %

The neural networks helped to find the percentage of emotions containing in each statement. 3 layers of neurons were formed with rectified linear

activation function and Soft-max activation function

RESULTS

performed Different models and evaluated accuracy.

```
y_pred1=clf1.predict(X_test)
y_pred2=clf2.predict(X_test)
y_pred3=clf3.predict(X_test)
```

```
from sklearn.metrics import accuracy_score

print("Gaussian",accuracy_score(y_test,y_pred1))
print("Multinomial",accuracy_score(y_test,y_pred2))
print("Bernaulli",accuracy_score(y_test,y_pred3))
```

```
Gaussian 0.5210084033613446
Multinomial 0.707983193277311
Bernaulli 0.6218487394957983
```

```
test="i am upset "
test = np.array([test])
test = cv.transform(test)
```

```
clf2.predict(test)
```

```
array([2], dtype=int64)
```

Here Array[2] describes the SAD statement.

Working on the Neural Networks, we found some interesting results.

```
predict(model, "i love code.")
```

```
WARNING:tensorflow:Model was constructed with shape (None, 0, 96) for input KerasTensor(type_spec=TensorSpec(shape=(None, 0,
96), dtype=tf.float32, name='dense_input'), name='dense_input', description="created by layer 'dense_input'"), but it was ca
lled on an input with incompatible shape (None, 96).
Predictions:
sad: 3.991281e-17
happy: 1.0
angry: 5.916648e-13
```

```
predict(model, "its sad to be alone")
```

```
Predictions:
sad: 0.9998567
happy: 0.00014321767
angry: 9.4248595e-11
```

```
predict(model, "If you don't like me remember it's mind over matter, I don't mind and you don't matter.")
```

```
Predictions:
sad: 0.0055046375
happy: 0.006868111
angry: 0.98762727
```

And after stating the emotion of the statement we found the accuracy of the neural Network Model.

```
evaluate(model, "You are wonderful and gorgeous!", "happy")
```

```
1/1 [==============================] - 0s 25ms/step - loss: 0.0730 - accuracy: 1.0000
Loss: 0
Accuracy: 1
```

CONCLUSION AND FUTURE WORK:

Using emoticons as noisy labels for training data is an effective way to perform distant supervised learning. Algorithms **Multinomial Naive Bayes,** can achieve high accuracy for classifying sentiment when using this method.

The model is used to successfully predict the sentiment of the given Data.