ABSTRACT

The online community may openly express their thoughts without revealing their identities thanks to the social media industry's tremendous expansion. False opinions may be easily posted by individuals with ulterior motives in an effort to disparage particular goods, services, politicians, or organisations. Because of the expansion of varied sites with a vast amount of ideas that are expressed in several languages, monitoring opinions and distilling their intimacys with these enormous data remains a challenging issue. As a result, the purpose of this report is to present a systematic literature review on multilingual intimacy analysis. It does so by outlining the common languages supported in multilingual intimacy analysis, pre-processing methods, current intimacy analysis approaches, and evaluation models that have been used for multilingual intimacy analysis. The thorough analysis of the literature's findings showed that most models supported two languages. The combination of languages for English, Chinese, French, Spanish, Portuguese, and Italian has not been accommodated in any of the evaluated literature. Tokenization, normalisation, capitalization, and machine translation are the often used pre-processing methods for the multilingual domain. Hybrid intimacy analysis, which incorporates localised unsupervised topic clustering and multilingual intimacy analysis, are the classification algorithms for intimacy analysis for multilingual relationships. Most research employed precision, recall, and accuracy as the standard for measuring findings while evaluating them.

ACKNOWLEDGEMENTS

The sense of contentment and elation that accompanies the successful completion of our project and its report would be incomplete without mentioning the names of the people who helped us in accomplishing this.

We are indebted to our guide Dr. Shankar Gangisetty who was nothing but a constant source of enthusiasm and whose profound guidance, valuable suggestions, and beneficent direction was mainly responsible for us to complete this project.

We take this opportunity to express our deep sense of gratitude and sincere thanks to our Head Dr. Meena S. M. for her tremendous source of inspiration and help in challenging our effort in the right direction.

Last but not least we like to thank all the course faculty, teaching and non-teaching staff for helping us during the project.

Priyansh Raj
Aayushi Gupta
Darshan Dixit
Shreyas Jakati

Contents

List of Figures

1 INTRODUCTION

1.1 All about CodaLab Challenge

Language's basic social function is intimacy. Predicting the intimacy of tweets in more than 10 languages is the topic of this SemEval shared challenge. The University of Michigan and Snap Inc. jointly coordinated this project. This work aims to forecast the intimacy of tweets in 10 different languages. To train your model, you are provided a collection of tweets in six different languages (English, Spanish, Italian, Portuguese, French, and Chinese) that have been annotated with intimacy ratings that range from 1 to 5.

You are encouraged (but not required) to also use the question intimacy dataset (Pei and Jurgens, 2020) which contains 2247 English questions from Reddit as well as another 150 questions from Books, Movies, and Twitter. Please note that the intimacy scores in this dataset range from -1 to 1 so you might need to consider data augmentation methods or other methods mapping the intimacy scores to the 1-5 range in the current task. Please check out the paper for more details about this question intimacy dataset. The model performance will be evaluated on the test set in the given 6 languages as well as an external test set with 4 languages not in the training data (Hindi, Arabic, Dutch and Korean).

1.2 Motivation

For a number of purposes, such as audience engagement, influencer identification, and sentiment analysis, tweet intimacy analysis can be a helpful tool. It could be able to determine how involved an audience is with a particular Twitter account or with a certain issue by looking at the intimacy level in tweets. Businesses or organisations trying to increase their social media presence and forge closer ties with their audience may find this to be helpful. To find people who are highly influential within a group or on a given issue, tweet intimacy analysis may be employed. Researchers trying to analyse online communities or firms looking to target their marketing campaigns may find this valuable. A tweet's level of intimacy is frequently a reliable predictor of its emotion. It could be feasible to determine the general sentiment of a specific Twitter user or group by looking at the intimacy of tweets, which might be helpful for sentiment analysis applications.

1.3 Objectives

• To predict the intimacy of tweets in 6 languages.
• To train the model with the MINT dataset to classify the tweets based on intimacy level.

1.4 Literature Survey

1) **A Review on Multi-Lingual intimacy Analysis by Machine Learning Methods**

The key languages that have been addressed or for which a separate corpus has been established have been identified, and we have analysed the existing research in multi-lingual intimacy analysis to determine the approaches being employed, their contributions, and their accuracy rates. We have observed many approaches being used to address the intimacy analysis issue. In this area, methods like machine translation are frequently used. The creation of comprehensive corpora in several languages has advanced significantly. A field that hasn't been studied as much, multi-lingual intimacy analysis provides a lot of room for future research. Even though we have seen a lot of work done in a few languages, there are still a number of understudied languages that may be considered for more research.With the investigation of more effective strategies and processes, accuracy rates might be raised from the already used methods' above average levels. It's important to take into account factors like accuracy, speed, and handling homonyms and homographs in both the source and destination languages. As a result, there is still much room for study in this area, which has applications in business, science, and user awareness.

2) **Multilingual intimacy Analysis: A systematic litreature review**

The characteristics of common languages supported in multilingual intimacy analysis, pre-processing procedures, intimacy analysis methodologies, and assessment model have all been covered in this paper's systematic review of works published between 2010 and 2019.

They investigated lexicon- and machine-learning-based methods for analysing multilingual intimacy, and they found that by integrating a number of cutting-edge technologies, translation software may enhance the study of multilingual intimacy. We use a variety of sample techniques to survey the characteristics of a certain language that will be taken into account while doing a multilingual intimacy analysis and the active learning.

3) **Twitter Sentiment Analysis Based on Ordinal Regression**

This article uses a variety of machine learning approaches to explain sentiment analysis of twitter data in relation to ordinal regression. In the framework of this study, we provide a method for extracting Twitter sentiment analysis through the development of a balancing and scoring model, followed by the use of machine learning classifiers to divide tweets into a number of ordinal groups. In this work, classifiers such Decision Trees, Support Vector Regression, Multinomial Logistic Regression, and Random Forest are employed. NLTK corpora resources' publicly accessible Twitter data collection is used to optimise this method.

According to experimental findings, Support Vector Regression and Random Forest exhibit accuracy that is nearly comparable to, and hence superior to, that of the Multinomial Logistic Regression classifier. The Decision Tree, however, has the best accuracy (91.81). According on experimental findings, the suggested model can accurately identify ordinal regression in Twitter using machine learning techniques. Accuracy, Mean Absolute Error, and Mean Squared Error are used to gauge the performance of the model. We want to enhance our strategy in the future by attempting to incorporate bigrams and trigrams. Additionally, we want to research several deep learning and machine learning methods, including Deep Neural Networks, Convolutional Neural Networks, and Recurrent Neural Networks.

4) **Sentiment Analysis of Twitter Data on Demonetization Using Machine Learning Techniques**

The recent problem of demonetization is the subject of sentiment analysis on the Twitter dataset. Though there are some unfavourable opinions, the study overwhelmingly reveals support for the change. To compare the findings, several machine learning classification techniques are utilised. SVM displays the highest level of accuracy for our dataset. Our method has a significant drawback since so many tweets are written in regional tongues like Hindi. The English-written Hindi terms were not included in the positive and negative corpus since they are written in English. All tweets in Hindi that are written in English are therefore categorised as neutral. In the future, those terms might be added and analysis accuracy could be improved. In the future, similar methods can be employed to lower the neutral count.

5) **Multilingual Sentiment Analysis for Web Text Based on Word to Word Translation**

We were able to evaluate the effectiveness of the multilingual sentiment analysis technique we had described by contrasting it to the earlier classifiers "VADER" and "GCP." It was shown that our classifier has the advantage of low translation costs since it just employs word-to-word translation to estimate sentiment values for each phrase rather than translating the complete text. In other words, our classifier has the potential to be used with many additional undiscovered languages since it can be used even when the input texts' syntax is unknown. With the use of their native tongues, people of all nationalities may evaluate sentiment data for a range of unfamiliar languages using the advantages of our method.The classifiers' evaluation experiment was conducted in English, German, French, and Spanish. Due to linguistic variances between languages, the investigation's findings cannot demonstrate any appreciable changes in values for any evaluation criterion. The results show that the classifier is appropriate for usage with data that is multilingual. Nevertheless, the classifier's overall accuracy is insufficient for practical use. Linguistic plasticity in syntactic analysis is essential for performance enhancement.

1.5 Problem definition

The challenge is to design and implement a Machine learning model to predict the intimacy of tweets in 6 languages. Given a set of tweets in six languages (English, Spanish, Italian, Portuguese, French, and Chinese) annotated with intimacy scores ranging from 1-5.

2 PROPOSED SYSTEM

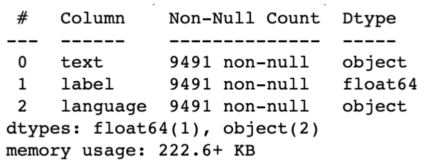
2.1 Understanding the Data

What is intimacy?

Intimacy refers to the sensation of being in a close, emotional bond and belonging together. It is a familiar and highly intimate emotive connection with someone created by knowing and experience of the other person.

2.2 Data set description

• There are 9491 tweets in the given dataset.
• There are three attributes: Text, label and language.
• Our dataset obtained via Twitter API consists of data related to tweets stored in a .csv format with the attributes.
• We have divided our dataset as 70-30. 70 percent train data and 30 percent test data.



```
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   text      9491 non-null   object 
 1   label     9491 non-null   float64
 2   language  9491 non-null   object 
dtypes: float64(1), object(2)
memory usage: 222.6+ KB
```

Figure 1: Overall Dataset.

Inference: From the shapes of the csv files, we can understand that the dataset has

2.3 Visualisation of Data

2.3.1 Analysis of Data distribution

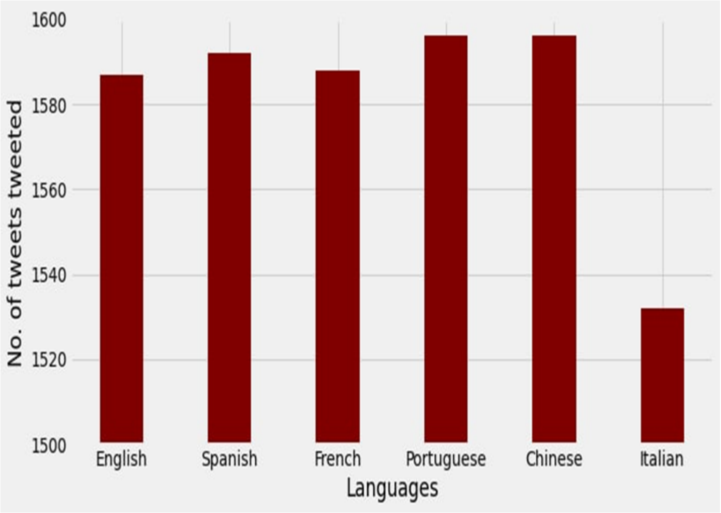


Figure 2: Number of tweets in each language.

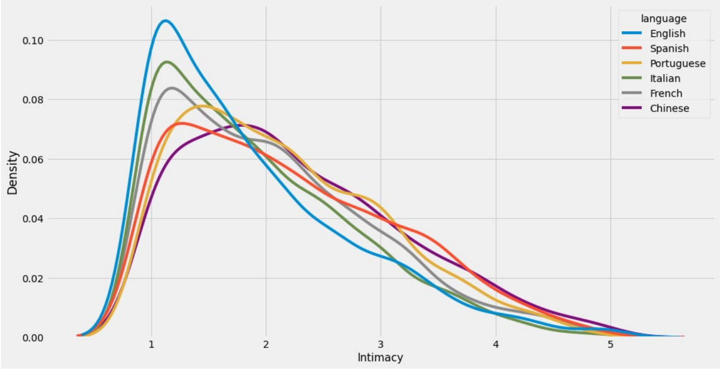


Figure 3: the intimacy level on x-axis and density of intimacy on y-axis.

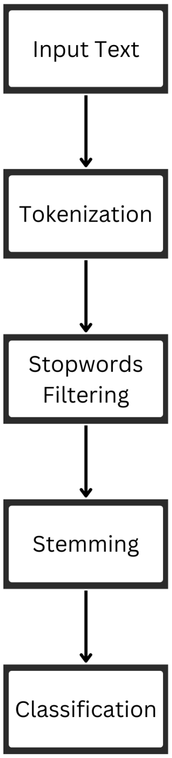
2.4 Data Pre-processing

From the Inferences through Our EDA, we have performed the necessary text preprocessing steps on the Tweet text.The conclusions on the dataset were as follows:

1. Removal of links to websites, such as news sites, YouTube, etc.
2. Removal of punctuation, special characters and emojis.
3. Removal of stop words.
4. Perform stemming on words in the given tweets.
5. Tokenizing the tweets.

Pre-processing methods taken into action:

1. Tokenization: Tokenization is the process of breaking down text into individual words, phrases, or other crucial parts. Whitespace, punctuation, and line breaks are used to separate tokens; punctuation is frequently removed during the tokenization process. Comparatively speaking to other preprocessing methods, tokenization is believed to be fairly straightforward.
2. Stopword Removal: A method of removing words like ”a,” ”of,” and ”is” that do not have any significant significance.
3. Lemmatization: The action of changing words back to what they were originally. Lemmatization, as opposed to stemming, adapts the word to the text’s context and gives it the right structure. Lemmatization maps the word ”caring” to the word ”care,” whereas stemming changes the word ”caring” to the word ”car.”
4. Stemming: The procedure of figuring out a word’s root. The word ”great” is the stem that links words like ”greatly,” ”greatest,” and ”greater.”
5.Noise reduction: Excluding distracting components like HTML, keywords, scripts, or ads.



2.5 Initial Approach

Tokenization, or breaking a tweet up into separate words or phrases, is one method for processing tweets. A toolkit or library for natural language processing can be used for this. The tokens can then undergo some straightforward text cleaning to get rid of punctuation, URLs, and other superfluous information. After the tweet has been cleaned, you can run some simple analyses on it to pull out the important details. Using a named entity recognition tool, we can locate named entities like persons, businesses, and locations. Additionally, you can tell whether a tweet is intimate, whether it is favourable, negative, or neutral. You can then choose how to continue processing the tweet based on the findings of the analysis. We might classify the tweet into predetermined categories or conduct additional analysis using the listed things that have been detected. A customer care team might be contacted if the tweet is unfavourable, for example. You could also utilise the intimacy of the tweet to decide how to address it. In general, the precise strategy for twitter processing will rely on the particular objectives and conditions of the project. In order to make the process successful and efficient, it is crucial to thoroughly plan and design it.

2.6 Proposed methodology

2.6.1 Proposed Classfication Techniques: Classfication Techniques

KNN -

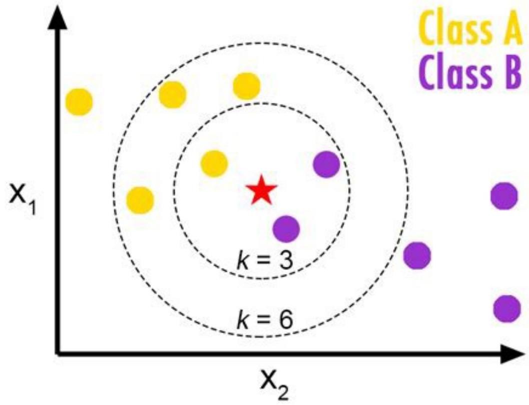


Figure 4: KNN Model

For the study of tweets, the k-nearest neighbours (KNN) algorithm is a categorization approach that is frequently used in machine learning. In order to classify the input data point based on the class labels of the k nearest points, it first determines the k number of ”nearest” data points to a particular input data point using some distance measure. In a twitter analysis scenario, the algorithm might take a tweet as input and, using a similarity metric such the cosine similarity of the word vectors, select the k tweets that are most similar to it. The input tweet would then be categorised using the k nearest tweets' majority class.

Gausian Naive Bayes

A classification algorithm that is frequently used in machine learning for tweet analysis is the Gaussian naive Bayes algorithm. Given the class label, it is a probabilistic model that assumes the data's properties are unrelated to one another. The algorithm might be applied to twitter analysis to categorise tweets according to the emotions they convey (positive, negative, or neutral). The features of the tweet would be the words, and the algorithm would determine the likelihood that the tweet reflects a particular sentiment by calculating the probabilities of each word occurring in each sentiment class. Following that, the tweet would be categorised according to the feeling that had the highest probability.

Stochastic Gradient descent

A common optimization approach in machine learning, notably when analysing tweets, is stochastic gradient descent (SGD). It is a form of gradient descent algorithm, which means that it updates the model's parameters to minimise loss using the gradient of a loss function. SGD uses a small, randomly chosen portion of the data, referred to as a batch or minibatch, to compute the gradient at each step as opposed to batch gradient descent, which computes the gradient of the loss function with respect to the model's parameters using the whole dataset at each step. Especially for big datasets, this makes SGD significantly quicker and more scalable than batch gradient descent.

Linear SVC

(Linear SVCs) Linear Support Vector Classifiers are effective for sentiment categorization and spam filtering in twitter analysis. To anticipate the sentiment of a specific tweet is the objective of sentiment categorization (e.g. positive, negative, or neutral). A dataset of labelled tweets, where each tweet is connected to a recognised sentiment, can be used to train a linear SVC. The computer can then figure out how a tweet's words relate to the emotion it conveys and use that knowledge to forecast how new, unread tweets will feel.

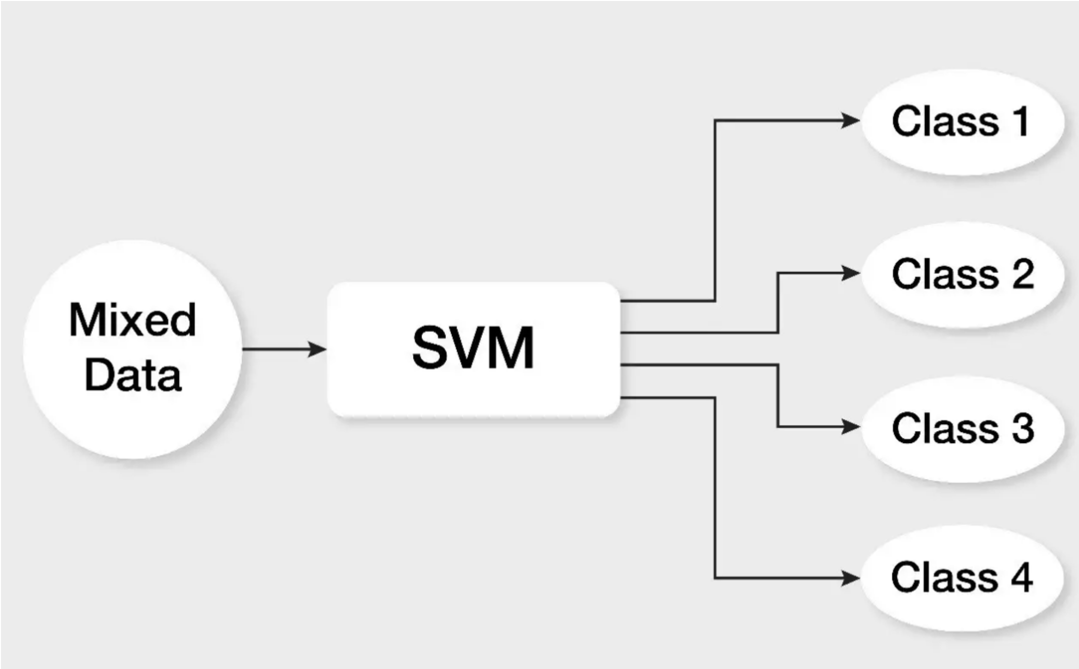


Figure 5: Linear SVC

Descision Tree

In order to classify data, decision tree classifiers build a tree-like model of the data, with each leaf node standing in for a class or value and each inside node representing a judgement based on the value of a particular characteristic. By dividing the data on the characteristic that optimises the reduction in entropy, the algorithm creates the tree (i.e. the amount of uncertainty or randomness in the data).

A decision tree classifier might be utilised in tweet analysis for tasks like sentiment analysis and spam filtering. The system might be trained using a dataset of tweets that have been categorised and each one has a known sentiment (e.g. positive, negative, or neutral). The computer might then figure out the connection between a tweet's words and the mood it conveys, using this knowledge to forecast the sentiment of future, unread tweets.

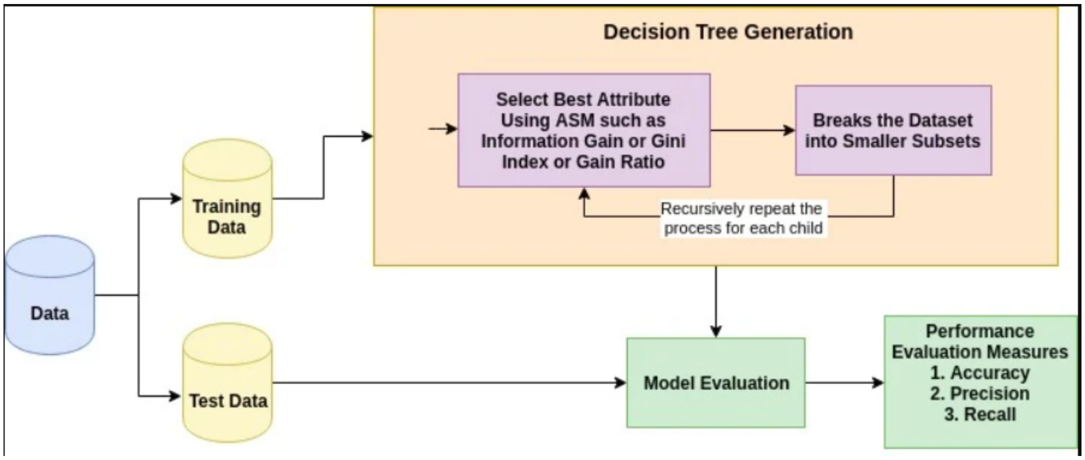

Decision Tree Generation
Select Best Attribute Using ASM such as Information Gain or Gini Index or Gain Ratio
Breaks the Dataset into Smaller Subsets
Recursively repeat the process for each child
Training Data
Data
Test Data
Model Evaluation
Performance Evaluation Measures
1. Accuracy
2. Precision
3. Recall

Figure 6: Decision Tree

2.6.2 Proposed Deep Learning Method

LSTM

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) that are well-suited for modeling sequential data, such as natural language text. LSTM networks are designed to remember information for long periods of time, which allows them to effectively capture long-term dependencies in the data.

In tweet analysis, LSTM networks could be used for tasks such as sentiment classification and language translation. For example, an LSTM network could be trained on a dataset of labeled tweets, where each tweet is associated with a known sentiment (e.g. positive, negative, or neutral). The network could then learn the relationship between the words in a tweet and the sentiment it expresses, and could use this information to predict the sentiment of new, unseen tweets.

Similarly, an LSTM network could be used for language translation by training it on a dataset of sentences in two languages. The network could then learn the relationship between the words in the two languages and could use this information to translate new sentences from one language to the other.

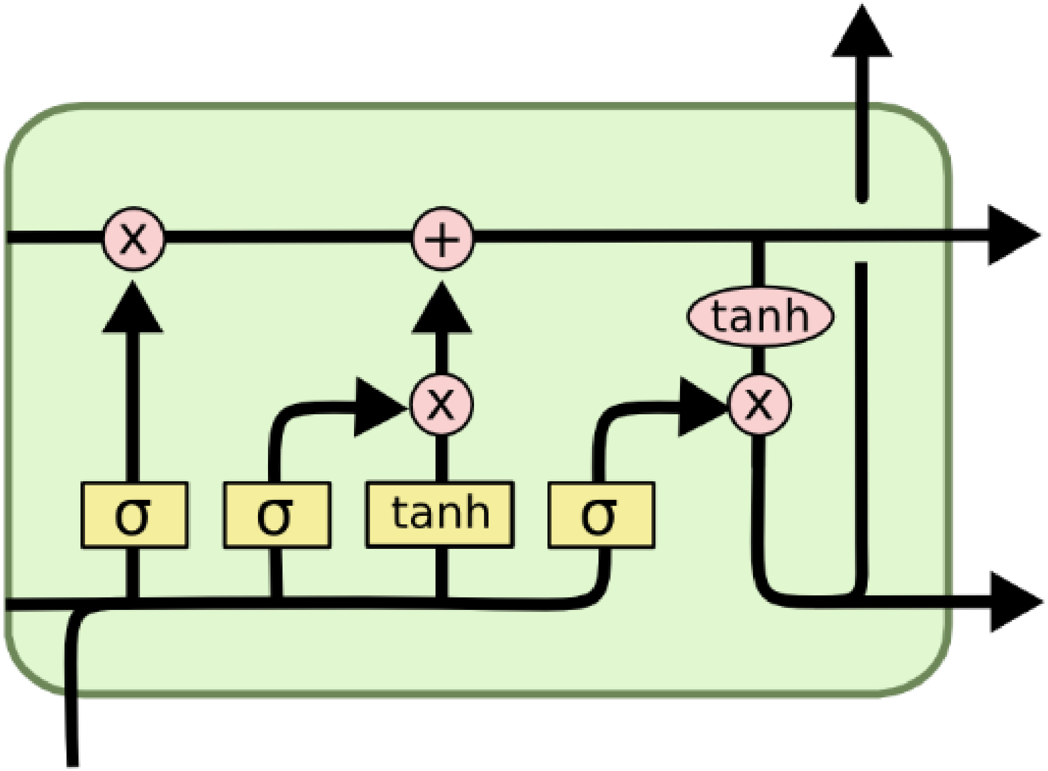


Figure 7: LSTM

3 IMPLEMENTATION

3.1 Logistic Regression -

Here is how Logistic Regression Works - First, a dataset of labeled tweets is collected, with each tweet being labeled as either positive, negative, or neutral.

The dataset is split into a training set and a test set. The training set is used to build the logistic regression model, while the test set is used to evaluate the model's performance.

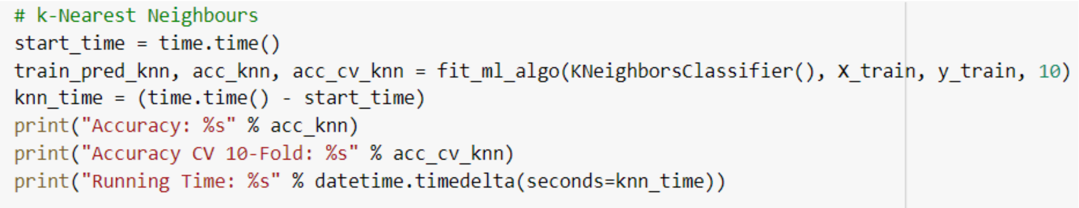
The logistic regression model is trained by using the words and phrases in the training set tweets as features, and the label (positive, negative, or neutral) as the target variable.

The model is then used to predict the sentiment of the tweets in the test set. The predictions are compared to the actual labels to evaluate the model's performance.

If the model's predictions are accurate, it can then be used to classify the sentiment of new, unseen tweets.

This classifier gave:
Accuracy: 87.9
Accuracy 10-fold: 40.49
Running time: 0:08:49.943452

3.2 K Nearest Neighbours -



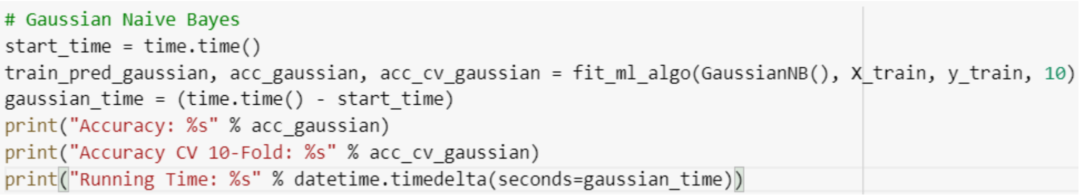
```
# k-Nearest Neighbours
start_time = time.time()
train_pred_knn, acc_knn, acc_cv_knn = fit_ml_algo(KNeighborsClassifier(), X_train, y_train, 10)
knn_time = (time.time() - start_time)
print("Accuracy: %s" % acc_knn)
print("Accuracy CV 10-Fold: %s" % acc_cv_knn)
print("Running Time: %s" % datetime.timedelta(seconds=knn_time))
```

Figure 8: K Nearest Neighbours

Here, We compute the distance between d and every sample in D Choose the k samples from D that are nearest to d Assign d the label y of the majority class in Sd.
This classifier gave:

Accuracy: 53.5
Accuracy CV 10-Fold: 36.42
Running Time: 0:00:59.402435

3.3 Gaussian Naive Bayes -

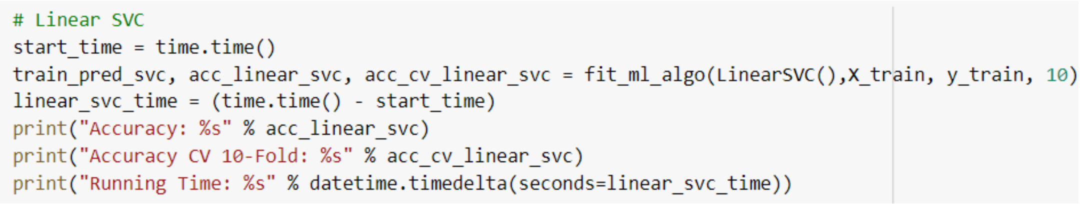


```
# Gaussian Naive Bayes
start_time = time.time()
train_pred_gaussian, acc_gaussian, acc_cv_gaussian = fit_ml_algo(GaussianNB(), X_train, y_train, 10)
gaussian_time = (time.time() - start_time)
print("Accuracy: %s" % acc_gaussian)
print("Accuracy CV 10-Fold: %s" % acc_cv_gaussian)
print("Running Time: %s" % datetime.timedelta(seconds=gaussian_time))
```

Figure 9: Gaussian Naive Bayes

To estimate the sentiment of a tweet, we will take the product of the probability ratio of each word occurred in the tweet.
Note, the words which are not present in our vocabulary will not contribute and will be taken as neutral.
This classifier gave the
Accuracy: 78.78
Accuracy CV 10-Fold: 28.19
Running Time: 0:00:21.751811

3.4 Linear SVC -



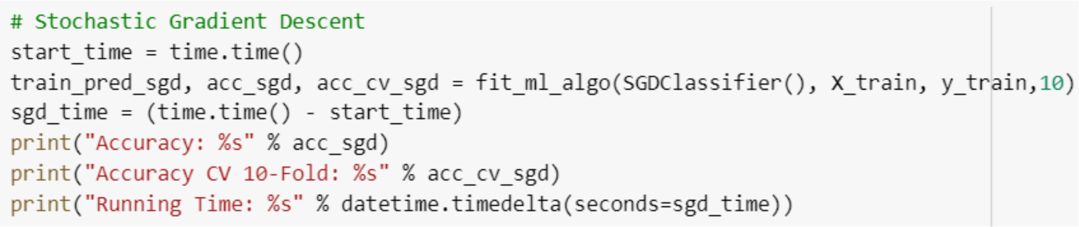
```
# Linear SVC
start_time = time.time()
train_pred_svc, acc_linear_svc, acc_cv_linear_svc = fit_ml_algo(LinearSVC(),X_train, y_train, 10)
linear_svc_time = (time.time() - start_time)
print("Accuracy: %s" % acc_linear_svc)
print("Accuracy CV 10-Fold: %s" % acc_cv_linear_svc)
print("Running Time: %s" % datetime.timedelta(seconds=linear_svc_time))
```

Figure 10: Linear SVC

We will build a simple, linear Support-Vector-Machine (SVM) classifier. The classifier will take into account each unique word present in the sentence, as well as all consecutive words. To make this representation useful for our SVM classifier we transform each sentence into a vector. The vector is of the same length as our vocabulary, i.e. the list of all words observed in our training data, with each word representing an entry in the vector. If a particular word is present, that entry in the vector is 1, otherwise 0.
This classifier gave the

Accuracy of 95.79
Accuracy CV 10-Fold: 37.29
Running Time: 0:00:13.849591

3.5 Stochastic Gradient Descent -

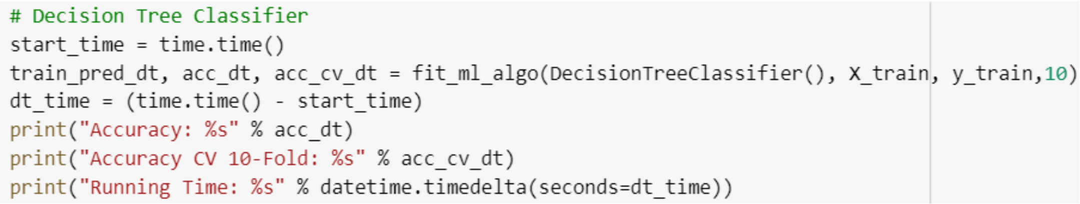


```
# Stochastic Gradient Descent
start_time = time.time()
train_pred_sgd, acc_sgd, acc_cv_sgd = fit_ml_algo(SGDClassifier(), X_train, y_train,10)
sgd_time = (time.time() - start_time)
print("Accuracy: %s" % acc_sgd)
print("Accuracy CV 10-Fold: %s" % acc_cv_sgd)
print("Running Time: %s" % datetime.timedelta(seconds=sgd_time))
```

Figure 11: Stochastic Gradient Descent

The model would be trained on a large dataset of labeled tweets, where the labels indicate the type of content in the tweet (e.g. positive, negative, neutral). The model would use SGD to iteratively update its parameters based on the training data, in order to minimize the classification error on the training set.
Once the model is trained, it can be used to classify new tweets that it has not seen before. For example, it could be used to automatically identify and filter out negative or inappropriate tweets, or to automatically categorize tweets into different topics or sentiment classes.
This classifier gave the
Accuracy of 96.92
Accuracy CV 10-Fold: 36.81
Running Time: 0:23:19.448569

3.6 Decision Tree Classifier -



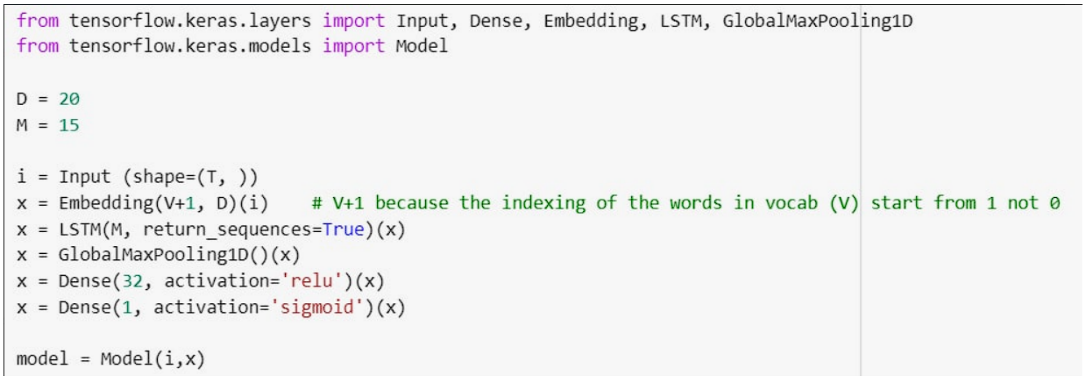
```
# Decision Tree Classifier
start_time = time.time()
train_pred_dt, acc_dt, acc_cv_dt = fit_ml_algo(DecisionTreeClassifier(), X_train, y_train,10)
dt_time = (time.time() - start_time)
print("Accuracy: %s" % acc_dt)
print("Accuracy CV 10-Fold: %s" % acc_cv_dt)
print("Running Time: %s" % datetime.timedelta(seconds=dt_time))
```

Figure 12: Decision Tree Classifier

Here it is working by constructing a tree-like model of decisions, where each internal node in the tree represents a decision about the input data, and each leaf node represents

a classification or prediction.
In tweet analysis, a decision tree classifier could be used to classify tweets based on their content. The classifier would be trained on a large dataset of labeled tweets, where the labels indicate the type of content in the tweet (e.g. positive, negative, neutral). The classifier would use the training data to learn the characteristics of each class, and to build a decision tree that can be used to make predictions on new tweets.
This classifier gave the
Accuracy of 99.54
Accuracy CV 10-Fold: 36.11
Running Time: 0:23:16.939001

3.7 Long Short Term Memory -



```
from tensorflow.keras.layers import Input, Dense, Embedding, LSTM, GlobalMaxPooling1D
from tensorflow.keras.models import Model

D = 20
M = 15

i = Input (shape=(T, ))
x = Embedding(V+1, D)(i)    # V+1 because the indexing of the words in vocab (V) start from 1 not 0
x = LSTM(M, return_sequences=True)(x)
x = GlobalMaxPooling1D()(x)
x = Dense(32, activation='relu')(x)
x = Dense(1, activation='sigmoid')(x)

model = Model(i,x)
```

Figure 13: Long Short Term Memory

LSTM works by introducing a series of gates that control the flow of information through the network. The gates allow the LSTM to decide which information to remember and which to forget, allowing it to learn and make predictions based on long-term dependencies in the data. The LSTM would learn to identify patterns in the words and phrases used in the tweets, and use this information to make predictions about the sentiment of new, unseen tweets.
Embedding Layer: that converts our word tokens (integers) into embedding of specific size
LSTM Layer: defined by hidden state dims and number of layers
Accuracy we found in LSTM (After 4 epochs): 60.01
Validation Accuracy we found : 55.10

4 RESULTS AND DISCUSSION

4.1 Cross Validation Scores

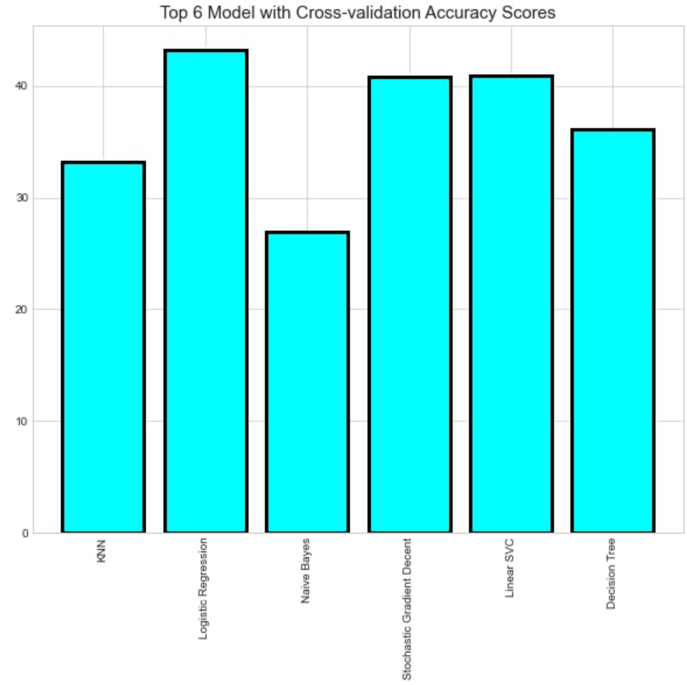


Figure 14: This plot shows the cross-validation scores of the top 6 classification models where the X-axis defines the classification models and the Y-axis defines the cross-validation scores.

Inference: The model shows a higher cross-validation score and has a comparatively high accuracy score even after shuffling the data.

4.2 Accuracy Scores

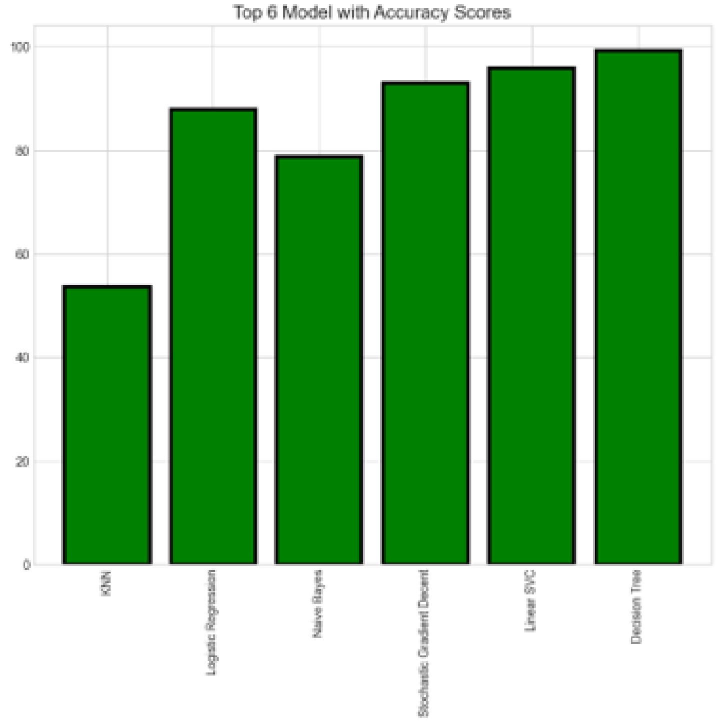


Figure 15: This figure represents the accuracy scores of the top 6 classification models that we have used in our implementation process. The X-axis represents the names of classification models and the Y-axis represents the accuracy scores.

Inference: If the accuracy score is high for a model we can say that it classifies the data into classes more efficiently as compared to other classification models.

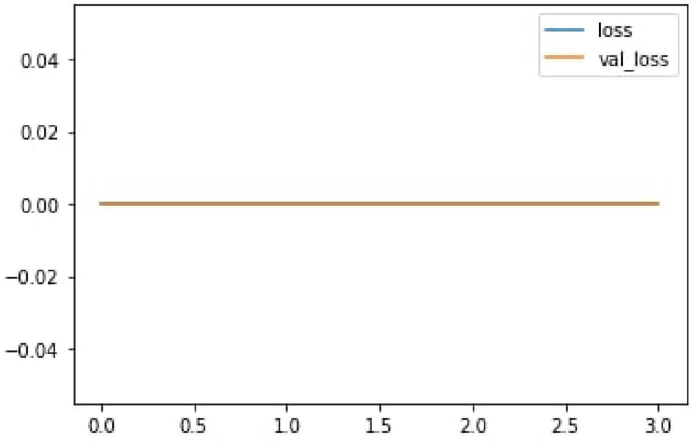


Figure 16: This plot represents the variation of loss and validation loss while using LSTM technique. X and Y axis represents units of loss.

Gradually the loss and validation loss became equal, and we didn't observe any increment in the learning process of the model.

5 CONCLUSION

This report uses a variety of machine-learning approaches to explain the Intimacy analysis of Twitter data in relation to ordinal regression. In this work, we offer a method that divides tweets into a number of ordinal classes using machine learning classifiers, then builds a balancing and scoring model to extract Twitter Intimacy analysis. The suggested model can detect ordinal regression in Twitter using machine learning techniques, according to experimental data, with a good accuracy result. However, the LSTM gives the accuracy (60) and validation accuracy (55).

6 REFERENCES

1.Nur Atiqah Sia Abdullah,Universiti Teknologi MARA and Nur Ida Aniza Rusli. *Multilingual Sentiment Analysis: A Systematic Literature Review.*

2.Alexandra Balahur, European Commission Joint Research Centre and Marco Turchi Fondazione Bruno Kessler-IRST - *Improving Sentiment Analysis in Twitter Using Multilingual Machine Translated Data.*

3.Mingxing Tan 1 Quoc V. Le 1. - *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.*

4.Shihab Elbagir and Jing Yang. - *Twitter Sentiment Analysis Based on Ordinal Regression*

5. Keita Fujihira and Noriko Horibe - *Multilingual Sentiment Analysis for Web Text Based on Word to Word Translation*

6. Sangmi Kim and Yuting Guo - *Natural language model for automatic identification of Intimate Partner Violence reports from Twitter*

7. Shamsuddeen Hassan Muhammad, David Ifeoluwa Adelani - *A Nigerian Twitter Sentiment Corpus for Multilingual Sentiment Analysis*

8. FARKHUND IQBAL, RABIA BATOOL - *Toward Tweet-Mining Framework for Extracting Terrorist Attack-Related Information and Reporting*

9. Natasha Suri, Prof. Toran Verma - *Multilingual Sentiment Analysis on Twitter dataset using Naive Bayes Algorithm*

10. Abhijit Bera, Dibyendu Kumar Pal - *Sentiment Analysis of Multilingual Tweets Based on Natural Language Processing (NLP)*