# Study of Deep Learning Models on Educational Channel Video from YouTube for Classification of Hinglish Text


Akshaya Sankar

10517653

A dissertation submitted in partial fulfillment of the requirements of

Dublin Business School for the degree of

MSc in Data Analytics

April 2019

Supervisor Name: Ms. Shubham Sharma

# Declaration

I certify that this dissertation which I now submit for examination for the award of MSc in Data Analytics, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Business School and has not been submitted in whole or part for an award in any other Institute or University.

The work reported in this dissertation conforms to the principles and requirements of the University's/Institute's guidelines for ethics in research.

**Signed:** Akshaya Sankar

**Dated:** 18th May 2020

# Abstract

The applications of data mining process are widespread. Within the framework of CRISP-DM a data mining project, this research exposes key areas like data preparation, feature engineering, model training and evaluation techniques.

Sentiment analysis / opinion mining is the technique for categorizing and defining computationally the opinions or feelings expressed in a piece of text to decide whether the attitude of individuals towards any topic, interest or product is a polarity of positive, negative or neutral. In numerous natural language processing assignments, deep learning is one of the most widely recognized methodologies. Deep learning models on normal language processing undertakings likewise beat regular AI models.

It is a common practice these days for public to use social media to share their opinions and ideas about most of the things and topics related to education is one of the most common searches among many social media posts or videos. Everyday lot of educational videos are uploaded to YouTube platform and most people share their opinions about the channel or video in the comment section. The research focuses on how well deep learning techniques work in extracting student/viewer tendency of YouTubers from the Hinglish dataset.

This research shall serve as a basis to provide useful information to teachers and online tutors by understanding and acting on the emotions of the students during the process of learning from their uploaded video and help them improve more on their methods of teaching.

**Keywords:** Sentiment analysis, Deep learning, Social media, Natural language processing

# Acknowledgements

I would like to express my sincerest thanks to my Research Supervisor Ms. Shubham Sharma, for her guidance, inputs and direction throughout the duration of this project.

I would also like to thank Marta Silva, M.Sc. Dissertation Coordinator, for her guidelines during the formulation and design of research proposal.

I would also like to express my gratitude to the lecturers and friends at Dublin Business School for their support over the course of my time here.

Finally, I would like to share my gratitude for everything I have been blessed with - family, friends and continued support from everyone.

# Contents

# List of Figures

# List of Equations



# List of Tables

# List of Graphs



# List of Abbreviations

| | |
|---|---|
| SA | Sentiment Analysis |
| AI | Artificial Intelligence |
| NLP | Natural Language Processing |
| IT | Information Technology |
| TV | Television |
| RQ | Research Question |
| DL | Deep Learning |

| | |
|---|---|
| NN | Neural Network |
| DNN | Deep Neural Network |
| CNN | Convolutional Neural Network |
| RNN | Recursive Neural Network |
| LSTM | Long Short-Term Memory |
| MLP | Multi-Layer Perceptron |
| CRISP-DM | Cross Industry process for Data Mining |
| NLTK | Natural Language Toolkit |
| TF-IDF | Term Frequency-Inverse Document Frequency |
| NER | Named Entity Recognition |
| GPU | Graphical Processing Unit |
| TP | True Positive |
| FP | False Positive |
| TN | True Negative |
| FN | False Negative |
| TPR | True Positive Rate |
| FPR | False Positive Rate |
| NYT | New-York Times |
| HPNN | Helpfulness Prediction Neural Network |
| DCNN | Deep Convolutional Neural Network |
| MNIST | Mixed National Institute of Standard Technology |
| RCNN | Region-Based Convolutional Neural Network |
| API | Application Program Interface |
| ACC | Accuracy |
| RELU | Rectified Linear Unit |

# Chapter 1: Introduction

## 1.1 Background

YouTube, an American, multilingual online-video sharing platform is most widely used today. It has a lot of channels drawing people of all age groups from different regions and culture according to their interest. A few examples would be Cartoon channels for kids to enjoy language, animation and graphics, Education channels for students to clarify their queries or learn some new topic that they are not clear with, Cookery channels to draw the attention of ladies and to increase their interest and skills in cooking and News channels for men to keep them updated with the latest happenings around them even if they are at their work place or anywhere away from home.

In this study we are trying to analyze one of the landmark topics of today that draws Indian students' interest towards watching Hinglish (Hindi+English) education-based videos that helps them understand concepts better when explained this way and improve their acumen.

**Trend of Video Learning taking over Classroom setup**

Youth of today are firmly seated with their smart phones in hand and watching the current hottest YouTube influencer. It's not amazing that they devour media in this style now. YouTube gives endlessly a larger number of potential outcomes than TV ever could. Social cooperation, incomparable decision over what to

watch, convenience and a simple method to talk with the genuine individual behind the screen has created interest in students today (Anonymous, 2017).

YouTube likewise offers a fortune trove of edutainment open entryways for the two, students and teachers. Various students find learning on YouTube better than sitting in a hall. The world's greatest video website is empowering another time of both online students and online teachers who are building colossal groups and indicating an immense number of people through the Internet (Pearce K., 2017).

The talk model has gotten old and does not work very well on YouTube. The new YouTube educators are fun, innovative and skillful with the gadgets and practices of video describing. They get your attention promptly and they hold it through a rollercoaster ride of visual describing, silliness and interest. These are the factors that mainly developed an interest in me to take up this topic for study (Pearce K., 2017).

## Factors that make Video Learning effective

With the end goal for video to fill in as a profitable piece of a learning experience than the normal talk model, it is important for the teacher to consider three components for video making and usage (Brame, C.J., 2015):

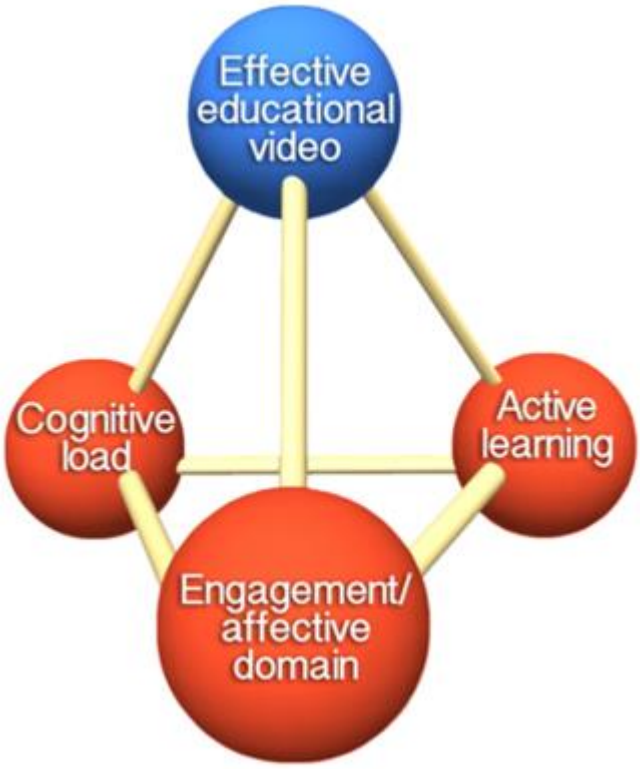



**Fig 1:** Steps to make a video useful as an educative and effective tool

Brame, C.J., 2015

One of the essential contemplations while building instructive materials, including video, is **cognitive load**. All the information from the sensory memory is processed in working memory which is very limited in storage. Now it works on what matter to retain and what to leave out which is a pre-requisite for encoding into long-term memory (Brame, C.J., 2015).

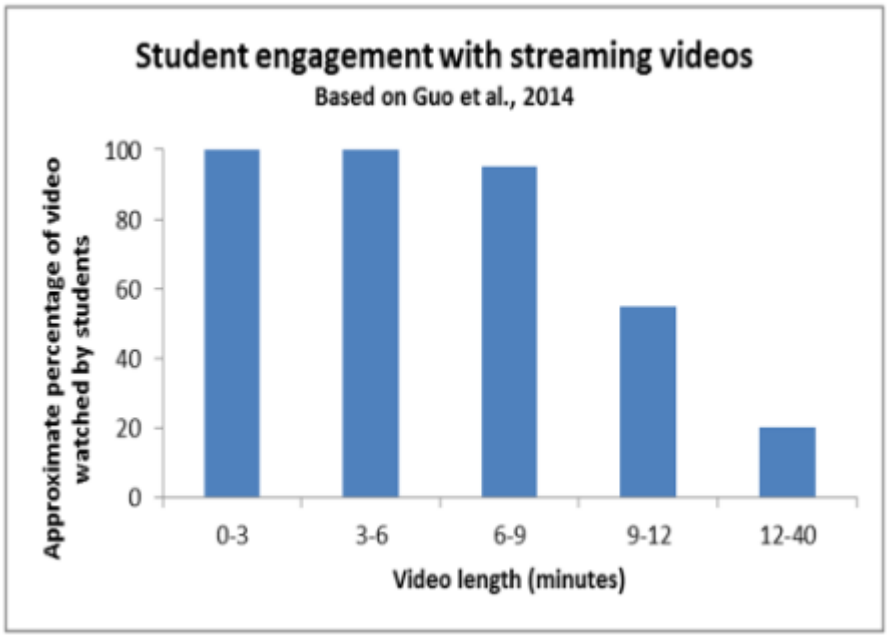
| Process | Effect on cognitive load | Examples |
|---|---|---|
| Signaling: Highlighting important information | Can reduce extraneous load<br>Can enhance germane load | Key words on screen highlighting important elements<br>Changes in color or contrast to emphasize organization of information<br>Changes in color or contrast to emphasize relationships within information<br>Brief out-of-video text explaining purpose and context for video (e.g., learning objective for video) |
| Segmenting: Chunking the information | Manages intrinsic load<br>Can enhance germane load | Short videos (6 minutes or less)<br>Chapters or click-forward questions within videos |
| Weeding: Eliminating extraneous information | Reduces extraneous load | Eliminating music<br>Eliminating complex backgrounds |
| Matching modality: Using the auditory and visual channels to convey complementary information | Can enhance germane load | Khan-style tutorial videos that illustrate and explain phenomena<br>Narrated animations |

**Fig 2:** Various steps involved in Cognitive Load

Brame, C.J., 2015

Following Cognitive Load, the next most significant part of making instructive recordings is to incorporate components that help advance **student commitment**. In the event that students don't watch the recordings, they can't gain from them. So, in order to achieve this, it is important to keep our video short with the following points given more concentration:

1. Using a perfect conversation style
2. Speak with enthusiasm
3. Make sure the material is apt for this class of students (Brame, C.J., 2015).




**Graph 1:** Graph that shows Student Engagement with Videos

Guo et al., 2014

After student commitment, next is to provide information on tools that they can use to test their understanding ability **(active learning)** so that they will know what exactly they have gained from their study. We can include all of these in a video by:

1. Using guiding questions related to the subject
2. Using interactive features that give students the control to answer their queries
3. Integrate working questions in the videos
4. Give them a larger portion of homework assignment (Brame, C.J., 2015).

At long last, it is demonstrated that use of video-learning to help chat between students, improve their social capabilities and teach them to outline conflicts

dependent on verification. After some time, these aptitudes will transform into a sound duty to their learning over the study plan and will set them up for persistent learning and preparing in later life (Anonymous, 2017).

## 1.2 Research Problem

We observed huge rush of comments on educational channel videos as it was drawing students more when it came to Hinglish Videos. Indian viewers were extremely excited and leaving their reviews on YouTube videos. This motivated us to conduct sentiment analysis in the education domain.

In India, the most spoken language is Hindi, and these comments are written in Hinglish language. That is a combination of Hindi language plus English language. Additional to this Hinglish language is not much explored in the field of natural language processing. In India, this is the most used for expressing their opinions on social media platform as most of the people prefer using Hindi, but they are so used to using most common English words in daily life like Bus, Work, Mobile, Time, Water, Brother, Company etc. end up using both the language.

There are certain challenges we face in Sentiment Analysis. Features such as subject recognition and conclusion identification are some of them. When it comes to multi-lingual dialects it becomes all the more complex. Another extraordinary related trouble is the mapping of sentiment resources from English

to any morphological language. Hinglish is morphologically rich language when contrasted with English (Kaur et al., 2019). When it comes to more complicated languages like Chinese and Urdu, as there are no specific stop words and other features explained for calculation it becomes unpredictable to manage datasets.

We took up this challenge and worked on the Hinglish language. Hinglish means Hindi language (one of the languages in India) written in English script with many words from English vocabulary (Kaur et al., 2019). For instance, “Meri sister IT company me work karti he”. Here, “sister”, “company” and “work” words are written in English script meaning “My sister is working in an IT company”. Most of the comments in the datasets are of this type.

### 1.2.1 Research Questions

The following questions of research are key drivers of the research discussed in this paper:

**RQ 1.** Can Hinglish text be classified effectively using deep learning classifier?

**RQ 2.** Over Hinglish dataset can deep learning techniques work accurately?

During our examination, we researched the previously mentioned RQs and had the option to investigate the knowledge through the study.

## 1.3 Research Aim

The aim of this research is to examine the usefulness of sentiment analysis to understand how the students' mood affects the learning process through videos. This research findings can be used to predict the videos based on the students' sentiment and to identifying the emotions, sentiments of students before and after going through the video and help instructors to address those issues during their next videos.

## 1.4 Hypothesis

LSTM, a deep learning model is trained to classify Hinglish text on educational channel video dataset. Does this significantly give better accuracy than CNN-LSTM, RNN, etc., other artificial neural network algorithms trained on the same dataset?

## 1.5 Research Objectives

The main goal is to evaluate the effectiveness of deep learning approaches and to decide whether deep learning models applied for YouTube Educational video dataset in Hinglish language provides useful insights which may help students in their studies.

The specific goal of this examination is to find a recognized classifier that can effectively classify the sentiments of the Hinglish remarks made on YouTube. It

can uncover some helpful measurable data for improving the techniques for study and instruction.

## 1.6 Research Methodologies

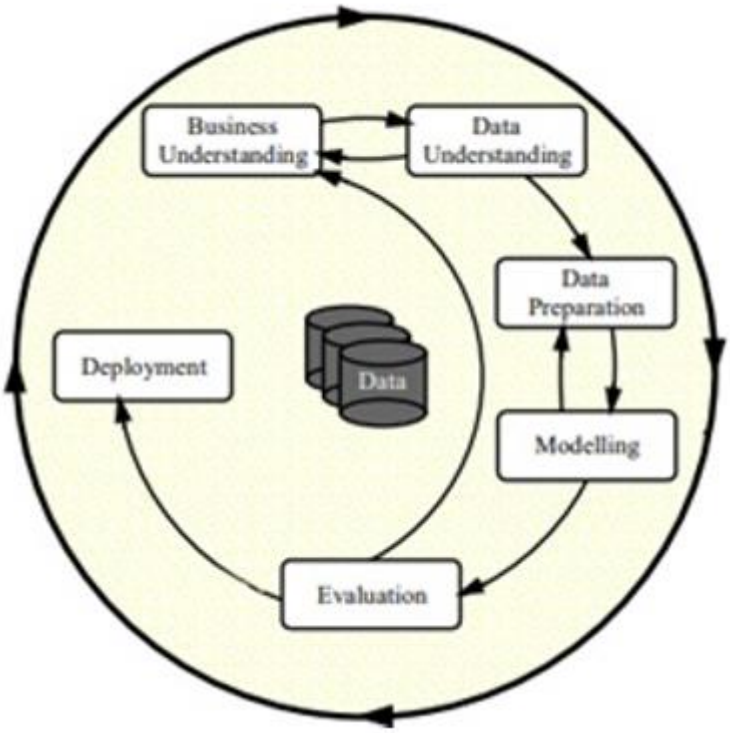



**Fig 3:** Cross-Industry process for Data Mining (CRISP-DM)

Wirth et al., 2000

The research carried out in this project is secondary because it is focused on the collected dataset from comments on an educational video published on the YouTube (Kaushik et al., 2019). The research conducted is an empirical study in which experiments, direct observations and empirical evidence can gain knowledge or feasible solution. The reasoning approach for this study is

deductive (top-down approach) as it is made by a hypothesis that will be accepted or rejected by following other measures.

The research methodology follows the well-known Cross-Industry Process for Data Mining (CRISP-DM) as shown in Figure above. CRISP-DM is a leading data mining process model. This work includes the following steps- Business understanding, data understanding, preparation of data, modeling and evaluation.

## 1.7 Scope and Limitations

Sentiment analysis often makes use of data in the form of user generated data like posts on social media such as Twitter, Facebook, etc. and customer reviews on any of the products, movies, etc. because sentiment analysis is most commonly used in social media and customer support to discover the sentiment or emotion of the people and how people feel about a topic. But it is limited to this that it can be used on data like audio, video and images to extract the sentiment using multi model sentimental analysis.

YouTube itself is a huge area where people upload thousands of videos daily and express their opinions on them. For this study, the dataset is from YouTube comments. The scope of this research is limited to the use of dataset which has limited number of comments extracted from video uploaded during the online class session.

For this study, open source resources such as Python libraries like NLTK, KERAS and SCIKIT learning were used. The accuracy of the results obtained in this research is largely dependent on the accuracy of pre-labeling data and the tools / libraries used to accomplish the task.

## 1.8 Document Outline

1. **Chapter 2 – Literature Review:** This chapter provides an outline of machine learning process and its types. Brings about the concept of feature selection, their representation, algorithms and evaluation techniques that had worked for various researchers. A critical review on various papers on data analysis and prediction techniques will be provided.

2. **Chapter 3 -Experiment Design and Methodology:** This chapter outlines the design approach of the project and explains the steps that have been taken along with the implementation. Here is addressed every step of the CRISP-DM methodology. This chapter specifically provides data understanding, data description, data exploration and pre-processing steps used to conduct the experiment. Following the document representation, this chapter addresses model preparation, model assessment and performance indicators used for the study.

3. **Chapter 4 – Results, Evaluation and Discussion:** The chapter explains the implementation of the model and its results. The models used for this research,

their implementation and evaluation are discussed. Finally discussing the results of all models to see the best performed model for this research.

4. **Chapter 5 – Conclusion and Future work:** This chapter will summarize the research carried out throughout this project. It provides the results achieved and insights gained, discusses the contribution of this research towards the research question. Lastly, this chapter discusses what future work after this can be accomplished.

# CHAPTER 2: Literature Review

## 2.1 Machine Learning

A PC program is said to learn from fact E as for some class of undertakings T and execution measure P, if its performances in T, as estimated by P, improves with experience E.

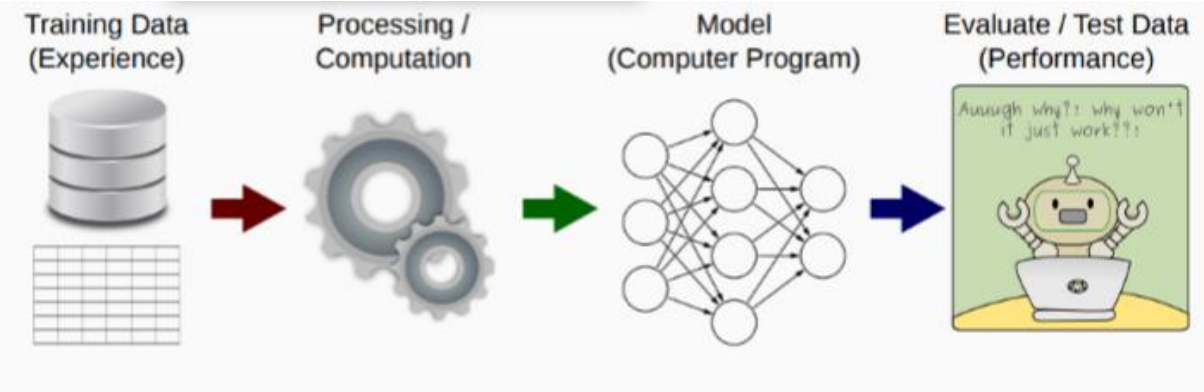



**Fig 4:** Machine Learning Process

Mitchell, 1999

## 2.2 Feature Extraction

Feature extraction/engineering is a process of understanding the data with respect to the domain and extract features from the dataset to make machine learning algorithms work. Features are very important in predictive modelling and have direct impact on the results.

Following are some of the important methods for feature representation:

### 2.2.1 Count Vectors as features

This is the most basic way to represent text data numerically. Here we will be creating vectors whose dimensions are equal to our vocabulary size. If the text data features the word in the vocab, we count one in that dimension. The count will be increasing, for every time we encounter that word again. We will be leaving 0's wherever we don't find the word even once (Fitzgerald et al., 2012).

We are going to get very large vectors as a result of this method. We will get accurate counts of the word in the real text data however it won't provide any relational or semantic information.

### 2.2.2 TF-IDF Vectors as features

In a text or corpus, TF-IDF score is used to describe the relative importance of a term/word. It consists of two terms: TF-normalized and IDF-Inverse Text Frequency, computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term/word appears (Fitzgerald et al., 2012).

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

IDF(t) = log e (Total number of documents / Number of documents with term t in it)

TF-IDF Vectors can be generated at various input tokens levels:

**Word Level TF-IDF:** This refers to the matrix in different documents describing TF-IDF scores of each word.

**N-gram Level TF-IDF:** N-grams are the combination of N terms together.

**Character Level TF-IDF:** This applies to matrix representing TF-IDF scores of character level n-grams in the corpus. In this research, TF-IDF is used as one of the methods of feature representation.

### 2.2.3 Word Embeddings

Word Embedding allows word with similar meaning to have similar representation. The position of the word is learned from the text within the vector space and this is based on the words that surrounds the word while it is used (Ghosal et al., 2017). This can be trained with input corpus or can be generated using the pre-trained word embeddings like Fast Text, Word2Vec and Glove.

## 2.3 Sentiment Analysis and Natural Language Processing

Sentiment analysis alludes to analyzing the opinions or remarks on an area. Due to high volume of sentiments it is also called as opinion mining (Naithani et al., 2015). Great experience, awful or neutral kind of inputs are being given by the users/customers.

Sentiment analysis is a subset of text mining which is the process of analyzing textual information for discovering structure and implicit meanings 'hidden' within texts as explained by Van Looy in 2016. It is a relatively recent technological development of using machine learning methods, data mining and analysis of natural languages.

NLP is the subject of software engineering to improve the capacity of machines to effectively process human language. 10 years back, PCs did not have the ability to perform such undertakings. Be that as it may, with the progression in innovation PCs have gotten able to perform NLP tasks. For Eg: Classification of documents, Named Entity Recognition (NER), Part-of-Speech Tagging and Analysis of sentiments (Collobert et al., 2008). Sentiment analysis aims to understand opinions of people and take advantage of it in different decision making (Robertson, 2004).

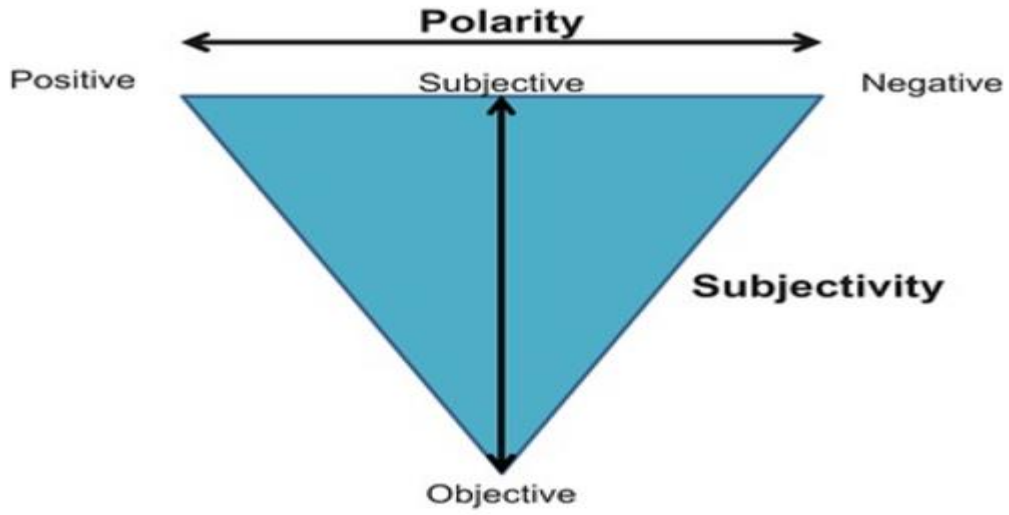



**Fig 5:** Polarity of opinion in Sentiment Analysis

Van Looy, 2016

### 2.3.1 Classification Levels in Sentiment Analysis

There are three different levels of the sentiment analysis (SA).

**Document Level:** The first level of Sentiment analysis is document level. We take almost the whole document for classification. And based on the final sentiment the whole document is classified. For example, for a given tweet, the tweet is classified whether it is positive/negative about the subject (Kharde et al., 2016).

**Sentence Level:** Classifies each sentence into a positive, negative or neutral opinion.

**Feature Level:** Here, it considers every feature of the target and then tries to classify to positive, negative or neutral. For example, in every sentence we consider the area of product that is considered.

In our study, we have done analysis of opinion at the sentence level. Here, sentence refers to each comment posted by the users which is classified into favor of either the video, the subject or the professor.

**Word Level:** Most recent works have used the prior polarity of words and phrases for sentiment classification at sentence and document levels. Word sentiment classification use mostly adjectives as featured but are finally used as adverbs (Kharde et al., 2016).

The two methods of automatically interpreting sentiment at the word level are:

1. Dictionary-Based Approaches

2. Corpus-Based Approaches

### 2.3.2 Resources of Sentiment Analysis

**Blogs and Forums:** Site and blog pages are broadly used to do explores in sentiment analysis. Web journals with respect to different areas like voyaging, products, instruction are posted on the page. Critics who read web journals post inputs like valuable, happy, satisfied and so on beneath the online journals. Analysis in this field has given significance to web journals as an asset pool for voicing sincere beliefs.

**Reviews:** Opinions of individuals about something specific is known as survey (Sabri et al., 2016). The comments in it could be specifically long, short, formal or informal. In this field, numerous investigations concentrated on reviews, their accessibility and extravagance with the opinion. Film and item reviews were among the most examined (Chen et al., 2008). These explain the achievement of something in the important areas. For instance, item surveys. When we consider the benefit of both association and user, item surveys play major role. In such cases organization web pages help alot. For example, Amazon website.

**News Articles:** Formal writing can be well studied in news articles. As an information source, they cannot be overlooked. Particularly, articles that involve raw data like numbers and digits follow the formal writing style (Sabri et al., 2016). In such data, duplication or removal of data becomes a challenge. For example, images.

**Social Networks:** Twitter, Facebook, Instagram all are the most well-known informal community locales. Twitter is a microblogging administration. These permits clients to post and read presents alluded on as "tweets" in twitter or remarks on Facebook or YouTube. On the web, internet-based life and blog posts, a great many reviews are distributed each day. By applying NLP and sentiment analysis, opinions can be classified in positive, negative or unbiased. Also, our source of the information is a YouTube video channel. Hinglish Comments on the YouTube instructive video are accumulated and orchestrated into a legitimate format with labels.

## 2.4 Exploration of Sentiment Analysis Methodologies

### 2.4.1 Supervised learning and Unsupervised learning

Algorithms for machine learning are categorized into Supervised and Unsupervised learning.

**Supervised learning:**

Supervised learning is a model building process which operates under supervision by providing the actual outcomes for all the instances in the training examples. Then model can predict the outcome of the new examples (Conneau et al., 2017). In other words, you can say machine already knows the output of the algorithm even before it starts working on it.

An important distinction between supervised learning tasks based on the nature of outcome/target variable to be predicted is where outcome variable can be a number to be predicted or a category to which outcome variable belongs.

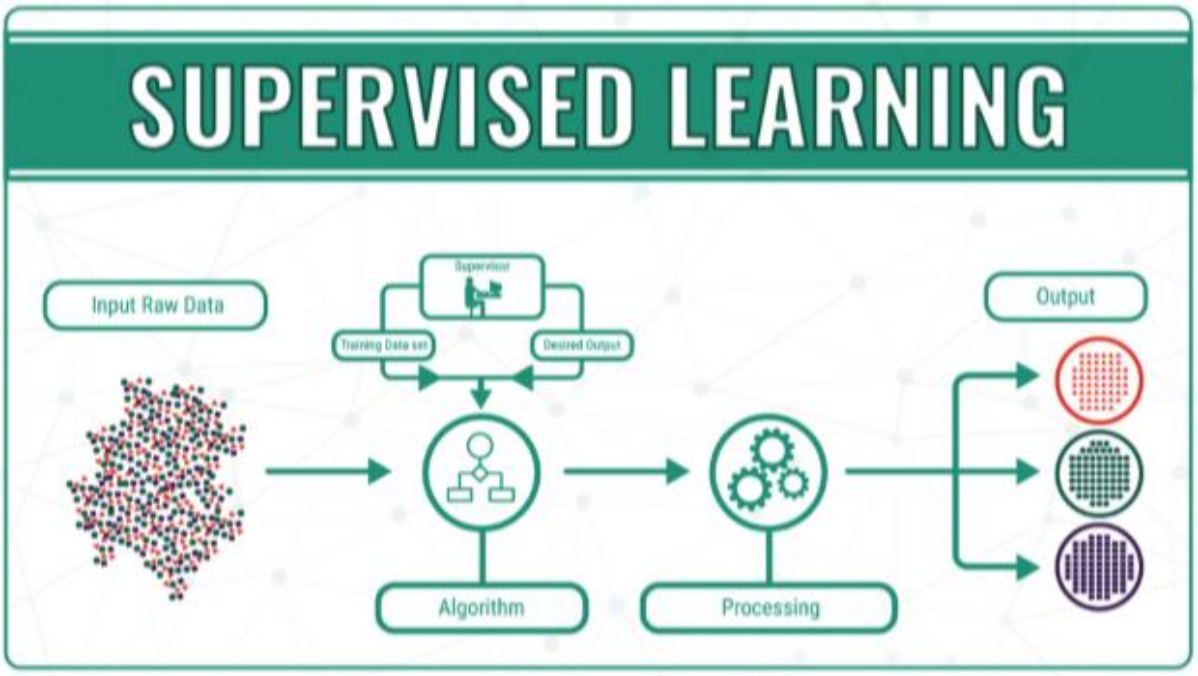



**Fig 6:** Supervised Learning Process

CHI Software, 2019

• **Classification** – Typically this is achieved when we want to map output labelled data. One of several predefined groups belongs to the outcome variable. This research is a classic problem of classification.

Eg. Includes demographic data such as marital status, age group and sex.

• **Regression** – This is typically done when we want to map input to regression. The outcome variable to be predicted is continuous or a real number and this problem follows a linear format.

Eg. Predicting a stock market index or projected sales figures.

This research is a classic Text Classification problem, supervised machine learning task with a labelled dataset containing text documents and their labels is used for training a classifier.

**Unsupervised Learning:**

With unsupervised learning, there is no specified outcome variable (unlabeled data) and the task is to discover any structure in the data that is "interesting" (Conneau et al., 2017). In this machine does not have concrete datasets as for most of problems the outcome is unknown.

Once machine has the input data, it learns all it can from information available in hand. Here, machine works on its own to identify the problem of classification like difference in shapes and colors. With this information in hand, this system

will then recognize all similar colors, shapes, objects and group them together. Machine itself will assign labels to those objects.

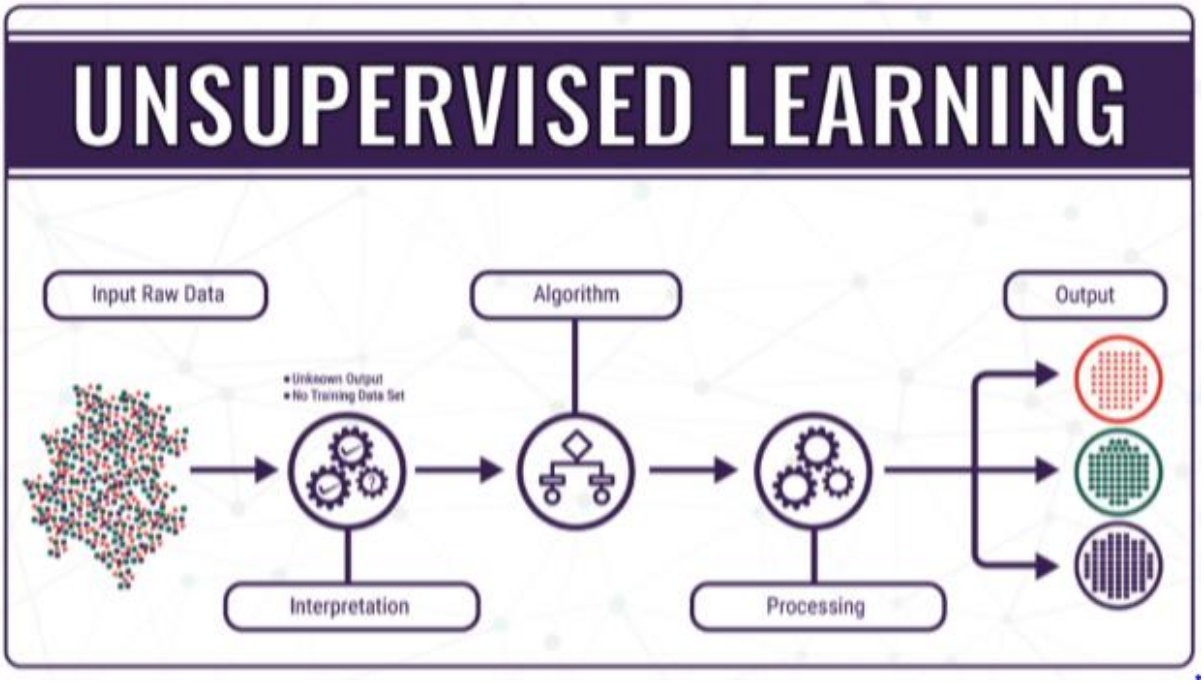



**Fig 7:** Unsupervised Learning Process

CHI Software, 2019

Unsupervised learning is most useful in exploratory analysis as it identifies trends and structure of data automatically.

In situations where it is impossible for a human to identify trends in the data, then unsupervised clustering can provide the valuable insights/trends in the data.

**Dimensionality reduction** is an important feature where we can represent the data with less columns or features.

As mentioned above, our research problem is classic classification problem under Supervised learning.

**Classification** can be split into two separate issues: binary classification and classification of multiple classes.

- **Binary Classification:** Classification task where only two classes are involved.
- **Multiclass Classification:** Classification task where it involves assigning an instance to one of the several classes. Our research belongs to this category as we have four classes to classify from the given dataset.

### 2.4.2 Neural Network

Neural network is analogous to the brain of humans. In other words, NN is a collection of algorithms that attempt to recognize the hidden patterns / relationships in a dataset through a method similar to human brains. Neural network can adapt to changing input, as a result it does not require any redesign to get the best possible result.

Stacked neural network can be called as Deep learning. Each network contains several layers, each layer is made of nodes and node is just a place in which computation happens. A node layer is a neuron-like row of switches that turns on and off as the data input is fed through the network. The following layers will be the output of each layer; however, input layer receives data as the input.

A model is known as a Deep neural network when it consists of more number of hidden layers (DNN) (Hinton et al., 2012). With growth in equipment configurations and the rise of deep learning approach, NLP has emerged to improve every feature of life. Its front-line applications incorporate handwriting examination, PC vision, and natural language processing. Deep learning strategies include more than 2-layer neural systems with different structures.

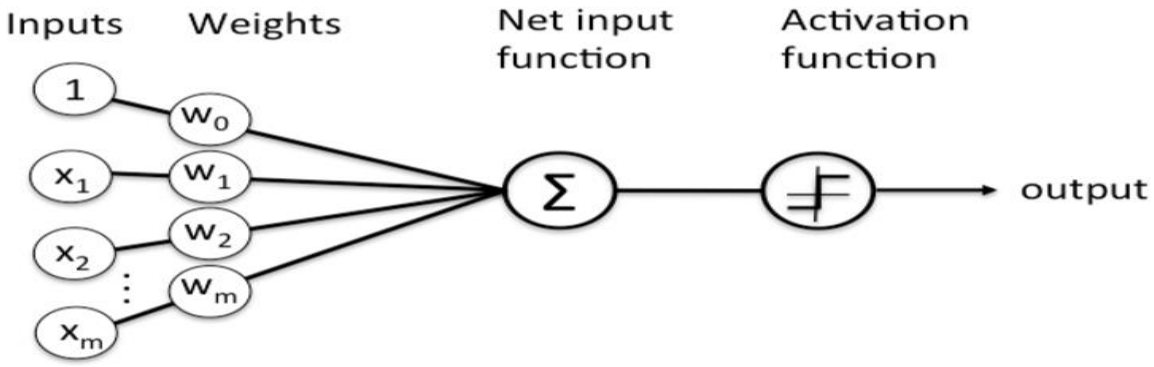



**Fig 8:** Node Structure

Nicholson, C.

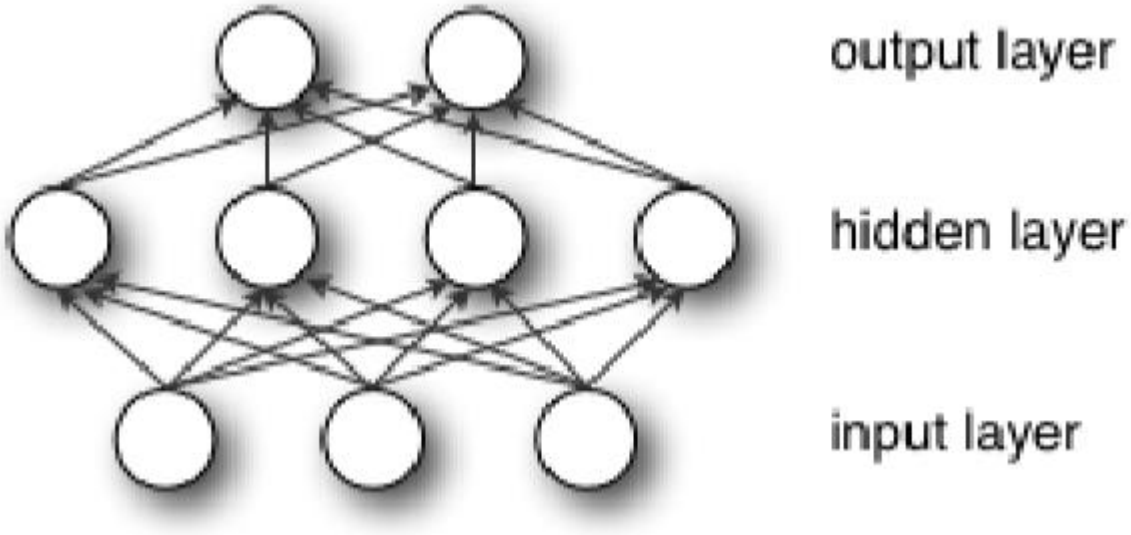



**Fig 9:** Neural Network

Nicholson, C.

Krizhevsky et al., 2012 has mentioned that deep learning models have achieved impeccable result in field of computer vision, natural language processing (NLP) and speech recognition in recent years.

When measurements are huge we obtain better predictions using Deep learning models. For instance, when we consider image classification, CNN works a lot better than other models. Another favorable position of neural system is that they are acceptable at dealing with nonlinear dataset when contrasted with AI calculations. In addition, they can be utilized in both supervised and unsupervised issues.

Three important reasons for the improvement of deep learning models: improved processing power chips (GPU), low-cost hardware and significant enhancements in algorithms of machine learning (Guo et al., 2016). Sepp Hochreiter showed that because of the gradient vanishing problem the learning rate increases through experiment and also enumerated methods to overcome the gradient issue (Hochreiter, 1998).

Variations of such networks are Multilayer Perceptron, Convolutional Neural Networks with Long Short-Term memory, etc. which we have implemented in the experiment.

## 2.5 Evaluation Measures

After Studying Machine learning techniques to train, let's see some of the methods to evaluate those results. Machine learning to create a value to any organization, should be able to give accurate predictions (Resnik et al., 2010). Even though training a machine learning model is a key aspect in any setup, however we cannot undermine how the machine learning model generalizes on unseen data. So, let's see some of the techniques to evaluate how well a machine learning model generalizes to unseen or new data.

Methods for evaluating any model performance are divided into two categories.

**Holdout**

The main aim of holdout evaluation is to test a machine learning model on unseen/different data than it was trained on because if we use the same data that we used to build the model, model will simply remember the whole training set and will always predict the correct label. This is called overfitting (Daelemans et al., 2002). Holdout provides an unbiased estimate of learning performance.

In this method, the dataset is randomly divided into three subsets:

**1. Training set:** It is a subset of the dataset used to build predictive models.

**2. Validation set:** It is a subset of the dataset used to assess the performance of the model built during training phase. However, not all modeling algorithms need a validation set.

**3. Test set/unseen data:** It is a subset of the dataset used to evaluate performance of a model.

In this research we have used holdout method for splitting the dataset. This method is useful because of its speed, flexibility and simplicity.

**Cross-Validation**

It is a technique that involves dividing/partitioning the original dataset into a training set, used to train the model and an independent set used to evaluate the analysis (Daelemans et al., 2002).

The most common cross-validation technique is k-fold cross-validation, in which the original dataset is divided into k equal size subsamples, called folds. The k will be the user-specified number, usually with 5 or 10 as its preferred value. This will be repeated k times, such that each time, one of the k subsets is used as the test set/validation set and the remaining k-1 subsets are put together to form a training set. The error estimation will be averaged over all k trials to get the total effectiveness of our model.

### 2.5.1 Model Evaluation Metrics

Model evaluation metrics are used to quantify model performance. The choice of evaluation metrics depends on a given machine learning task such as classification, clustering, ranking, regression, topic modeling and many others.

And there are some metrics, such as precision-recall, which are useful for multiple tasks. Supervised learning tasks such as classification and regression constitute most machine learning applications.

**Accuracy**

This is the most commonly used metric to judge a model and is not a clear indicator of the performance. But it won't be the great tool when classes are imbalanced. It is one of the most widely used classification performance measures and is defined as a ratio of the correctly classified samples to the total sample number (Tharwat, 2018).

$$Acc = \frac{TP+TN}{TP+TN+FP+FN}$$

**Eqn 1:** Accuracy Measurement

Tharwat, 2018

**Precision**

Percentage of positive instances out of the total predicted positive instances (Tharwat, 2018).

$$\frac{TP}{TP + FP}$$

**Eqn 2:** Precision

Tharwat, 2018

**Recall/Sensitivity/True Positive Rate**

Percentage of positive instances out of the total actual positive instances. The denominator (TP + FN) here is the actual number of positive instances present in the dataset (Tharwat, 2018).

$$\frac{TP}{TP + FN}$$

**Eqn 3:** Recall/Sensitivity

Tharwat, 2018

**Specificity**

Percentage of negative instances out of the total actual negative instances. The denominator (TN + FP) here is the actual number of negative instances present in the dataset (Tharwat, 2018).

$$\frac{TN}{TN + FP}$$

**Eqn 4:** Specificity

Tharwat, 2018

**F1 score**

It is the harmonic measure of precision and recall that takes the commitment of both, the higher the F1 score, better the classifier (Tharwat, 2018).

$$\frac{2}{\frac{1}{precision}+\frac{1}{recall}} = \frac{2 * precision * recall}{precision + recall}$$

**Eqn 5:** F1 Score

Tharwat, 2018

### 2.5.2 Related Work

Classifying text sentences dependent on extremity has been a convention of opinion mining. Aside from this, sentiment analysis is thriving and one of the prime models is that it has likewise been utilized to advocate national corrections and recommendations by dissecting parliamentary discussions. Besides, studies to separate notable issues in the public eye from general reviews via social media is turning into a fascinating subject these days.

YouTube is one of the platforms where most of the education related videos like teaching videos, edutainment channels videos etc. are uploaded and viewed by students. People may like the videos or may disagree with that situation, there will be mixed opinions on every video uploaded. Most of the viewers express their opinions by commenting on the videos. These comments can be collected

and used to determine the viewers sentiment on the education style and system at that time.

Commenting systems on the Social media have been growing in popularity in the past few years, from blogs and social media sites like Facebook, YouTube and Twitter to major news sites like NYTimes.com (Hsu et al., 2009).

Akshat Bakliwal alongside others completed 3-class sentiment classification probes corpus of 2,624 tweets framed before the Irish General Elections in February 2011. They accomplished 61.6% utilizing supervised learning model (Bakliwal et al., 2013).

Sangjae Lee and Yeon proposed HPNN - a helpfulness prediction neural network. This was experimented to decide if surveys on the items are useful to plan or improve business. A variation of multilayer perceptron neural system model with back prorogation is HPNN. It performed superior to linear regression with nearly lower mean squared error (Lee et al., 2014).

Work of Ayesha Rashid demonstrated restrictions on different opinion level with techniques utilized in sentiment analysis. Utilizing three different models Naive Bayesian, Decision Tree and KNN demonstrated that Naive Bayes beats the other two calculations and accomplishes 97% exactness, accuracy and recall in the work on content classification of sentiments written in Roman-Urdu and English (Bilal et al., 2017).

Lei Huang used Naive Bayes, Maximum Entropy, and SVM models for twitter message classification of opinion with AI calculations and accomplished exactness above 80% when prepared with the sentiment dataset (Y. Wang et al., 2016).

Work on NLP to build up a model to recognize Named Entities in Hinglish tweets which are blend of Hindi in addition to English language tweets contrasted the created technique and existing off-the-self NER instruments. Their model was 33.18% more precise relatively (Kumaraguru et al., 2018).

Sentiment analysis on Bangla and Romanized Bangla content is a remembered work for deep learning. They developed Deep recurrent model and LSTM which accomplished 70% precision (Hassan et al., 2016).

There was a basic model already built and with slight hyper parameter modification to the design improved execution in sentiment examination and question classification using CNN model (Kim, 2014).

A convolutional architecture of the Dynamic convolutional neural network (DCNN) was designed to accurately predict binary and multi-class sentiment analysis, six-way question classification and Twitter sentiment prediction (Grefenstette et al., 2014).

By grouping 1.2 million high-resolution pictures in the ImageNet ILSVRC-2012 competition the result of the test accomplished brilliant execution in the wake of

building up a deep convolutional neural system. They demonstrated record breaking results using an assorted picture dataset using deep convolutional neural system. This placed them among the top 5-winners. Using supervised learning accomplished the best blunder pace of 15.3% (Krizhevsky et al., 2012).

With countless manually written data, a gathering actualized CNN of 4 convolutional layers on MNIST dataset. This showed how deep learning can accomplish elite in digit or character acknowledgment with an exactness of close to 98% (Garg et al., 2019).

Some others introduced a programmed framework for tag recognition with a deep learning approach. Depending on the resolution of the picture it defines if the picture is high or of low resolution and finds out if the picture was taken during the part of day or night. This improved precision to a greater extent (Alimi et al., 2017).

In contrast to the past work on difficult objects, Fast Region-based Convolutional Network technique (Fast R-CNN) accomplished high precision of 66% to 68% (Girshick, 2015b).

# Chapter 3: Experiment Design and Methodology

## 3.1 Introduction

This section explains this research's general structure. The CRISP-DM process will be used, as shown in figure 3. Text mining process has several phases: market understanding, data understanding, data planning, modeling, evaluation and implementation. We will use a similar approach in this research, as mentioned by Botzenhardt et al., 2011.

## 3.2 Business Understanding

The business problem is to assess from a dataset of YouTube comments, which cohorts the views of the students on the topic of Percentage in general mathematics. The goal is to use deep learning methods to analyze the sentiment of the students, whether they are favoring or against the video/topic and to find how well these sentiments determine the quality of the video.

The overall business objective of this project is to find the best predictive deep learning model to classify the students' sentiments.

## 3.3 Data Understanding

The data interpretation phase begins with data collection and gaining first insights of the data to get acquainted with it. In this research, data is available from

YouTube Channel named Wi-fi Study, gathered from a video on Percentage topic published on the YouTube.

The data was collected with the help of YouTube video API. The dataset is divided into 4 classifications:

Label 0 – Answer comments

Label 1 – Neutral comments

Label 2 – Praising comments

Label 3 – Query comments

We have converted these tags to numbers ranging from 0-3 using simple code where we convert classification to values. There is a section called Comments in the dataset. It is depending on this Comments section that the labelling is done. Dataset is available on URL : https://www.youtube.com/watch?v=YvLIPPgkJN8.

The dataset has 2382 comments of every category which involves 4 different labels = 4*2382 = 9528 total comments.

## 3.4 Data Preparation (Pre-Processing)

Pre-processing incorporates removal of stop words, invalid values, numbers, unique characters and punctuation, changing over the whole content into lower case, tokenization and stemming.

In the wake of removing such ambiguities the remarks can be effectively reasonable by the model. A short time later, different word vectorization procedures: TF-IDF and Keras tokenizer is utilized to get ready information into deep learning required arrangement. After pre-handling, different procedures of deep learning and their output examination is finished with different configurations of layers and hubs.

**Removing Stop words:** Stop words are the words which do not add any significance to the sentences. For instance, "is", "am", "we" and so on. It additionally incorporates punctuation. Such stop words ought to be expelled from content corpora before preparing model. However, it is fitting to first train models without expelling stop words to comprehend the credibility of content information. On the off chance that the model neglects to give better exactness just, at that point should we attempt to remove stop words and check execution.

One explanation for this is that we cannot comprehend at the first glimpse what words are significant and which words are of less significance for building a model. In the investigation, we have removed the English stop words as Hinglish

does not have particular set of stop words. Consequently, not all stop words, but rather just punctuations were expelled from the crude text data.

**Stemming:** Stem is additionally referred to a root type of a word. In this way, stemming sentence changes over each expression of sentence to their unique structure in the wake of expelling suffix and prefix. For instance: changing to change, played to play. In the investigation stemming decreased model execution due to the Hinglish idea of information. Nltk or Keras does not represent the capacities to work with Hinglish language information. Henceforth, stemmed words change the importance of Hinglish words that lessens the output of the model.

**Lemmatization:** Then again, morphological investigation of the word is a thought while changing over the word to its root structure. The stem probably would not be a real word while, the lemma is a real language word. To do as such, point by point word references are required for lemmatization calculations to glance through and interface the word back to its lemma.

**Tokenizing:** Includes breaking each sentence into the word or character level tokens and evacuating words that do not add significance to the sentences (Naithani et al., 2015). Vectorizing procedures like TF-IDF or Keras-text-to-sequence changes over this raw data into vectors representing each word as a number. By restricting words, the vectorized information prepares deep neural systems and are classified with deep learning procedures (Hassan et al., 2016).

## 3.5 Data Splitting

After the data is prepared, the data is divided into training and test data with 80% and 20% respectively using Hold-Out technique to evaluate the models on unseen or different data than it was trained on, so if we use the same data that we used to create the model, the model will simply remember the entire training set and will always predict the correct label resulting in overfitting.

## 3.6 Document Representation

In this research we have used the following different representations in these experiments:

• **TF-IDF:**

The aim behind TF-IDF is that not all words occurring most frequently in a document are important or related to documents. Instead of just counting the occurrence of words, this algorithm allocates weights or importance to each word and creates a tabular form of words and documents (Robertson, 2004). Inverse Document Frequency defines a measure of the importance of a word.

The terms like “is”, “of”, “that” which occur in many documents should be less important than one like “Awesome” which occurs in few studies related documents (Robertson, 2004). TF (Term Frequency) measures the frequency of the term in the document itself. Importance of the word is now multiplication

of these two values IDF and TF. It is called a TF-IDF weight.

TF-IDF weight = TF(t) * IDF(t)

**• Keras text to sequence:**

Transforms each text into a sequence of integers. Only the most frequent words known by the tokenizer will be considered.

## 3.7 Modeling

Following the understanding of business and results, pre-processing, splitting and feature extracted, classifier are fitted on the training data. To compare with each other, five deep learning models are developed.

### 3.7.1 Simple Deep Learning Model

Artificial Neural Network or Deep Learning models is one of the most astonishing contributions to the world of AI. They are properly made to solve the sentiment-based issues with may be binary or multi-class problems for every single to grasp and explore when it is being displayed differently in PC's. For instance, PC can figure multiplication of 345656585957 and 9003051827 in fraction of second while trying to differentiate a cat from a cub present in a picture becomes a little complex. With the huge growth in the field of deep learning there has been huge success in this sort of opinion-based issues (Sharma, 2019).

Keras is an API that is utilized to run different deep learning libraries for example Tensorflow, Theano, and so on. It is to be noticed that Keras does not have an execution and it is an elevated level API that suddenly spikes in demand on top of other deep learning libraries. The difficulty we are endeavoring to comprehend is an opinion based one. The program builds a deep learning model, trains utilizing the training set and afterward tests it utilizing the test set. At last, the prepared model is utilized to anticipate the qualities (Sharma,2019).

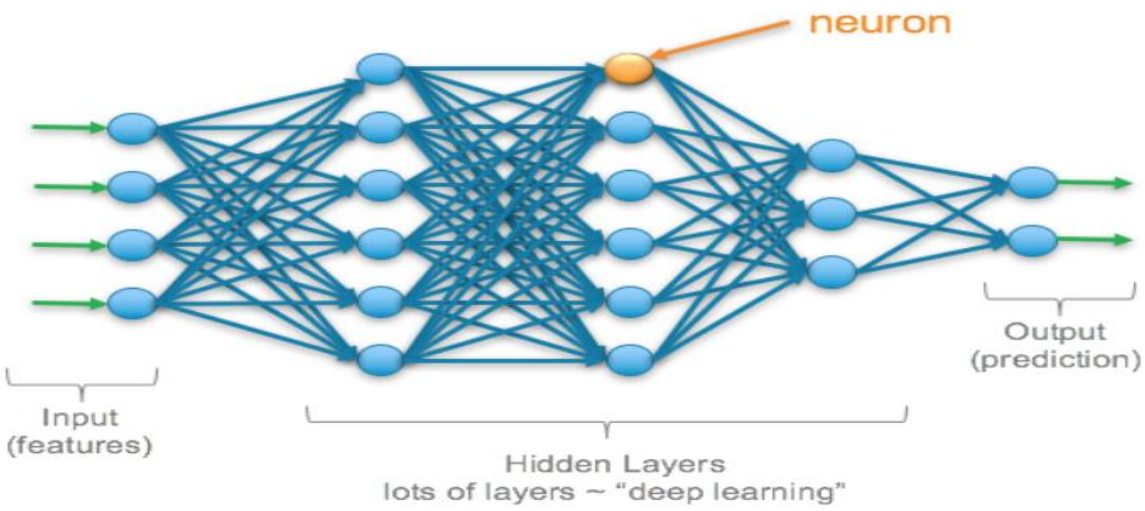



**Fig 10:** Architectural Diagram of simple Deep Learning Model

Ronaghan, S. 2018

In our code, we instated a model as pile of layers (Keras.Sequential) and afterward straightened the information cluster to a vector (Flatten()). The flattening part additionally happens to be the primary layer of the neural system. The second layer of the system comprises of 32 nodes and the activation functions we are utilizing is RELU (Rectified Linear unit) and softmax (Sharma, 2019).

The design of the model depends, to a huge degree, on the issue that we are attempting to understand. The model we have made above would not work very well for scientific calculation issue. The optimizer we are utilizing is 'adam' which is a force-based analyzer and keeps the model from getting stuck in nearby minima (Ba et al., 2014). The loss function we are using is ‘categorical cross-entropy’ as we are dealing with multi-class problem.

The training set will be fed to the network 5 times (epochs) for the training purpose as the dataset is very huge. The epoch needs to be carefully selected as a lesser number of epochs may lead to an under-trained network while too many epochs may lead to overfitting, wherein the network works well on the training data but not on the test data set.

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 250)               0
_________________________________________________________________
embedding_1 (Embedding)      (None, 250, 128)          552576
_________________________________________________________________
flatten_1 (Flatten)          (None, 32000)             0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                1024032
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 132
=================================================================
Total params: 1,576,740
Trainable params: 1,576,740
Non-trainable params: 0
```

**Fig 11:** Model Summary of Simple Deep Learning Model

After trying various configurations, we achieved the highest accuracy of 76% with epoch set to 5 and batch size 64.

### 3.7.2 Recurrent Neural Network (RNN)

Basic RNNs are a network of neuron-like nodes organized into successive layers. Each node in a given layer is connected with a directed (one-way) connection to every other node in the next successive layer. Each node (neuron) has a time-varying real-valued activation. Each connection (synapse) has a modifiable real-valued weight. Nodes are either input nodes (receiving data from outside the network), output nodes (yielding results), or hidden nodes (that modify the data en route from input to output).

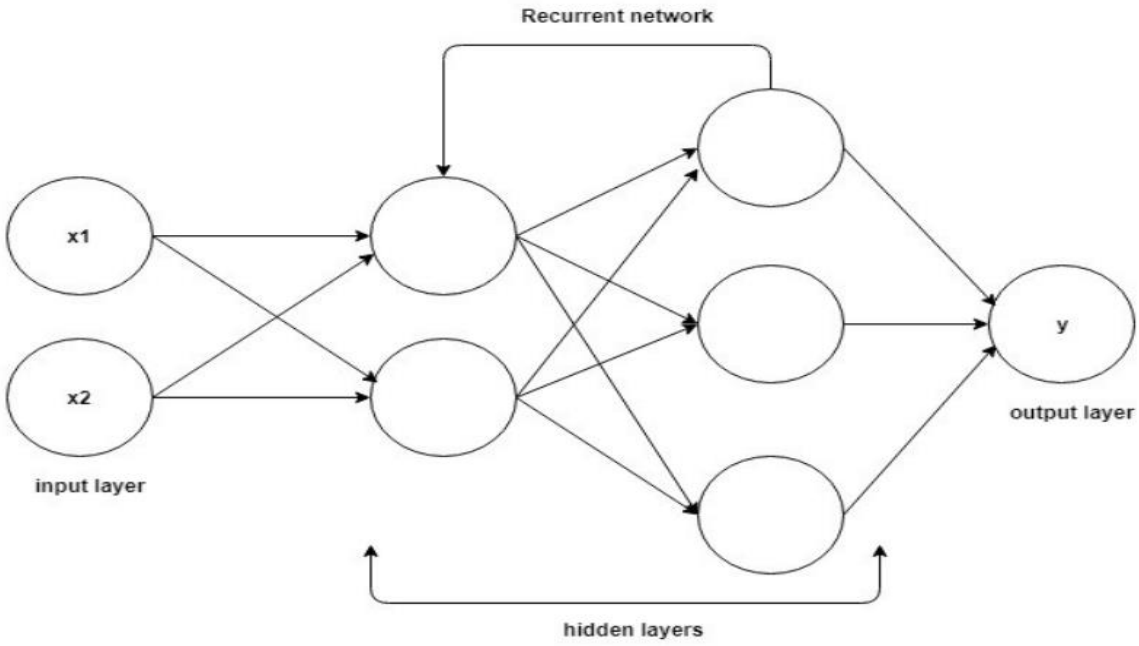



**Fig 12:** Architectural Diagram of Recurrent Neural Network

Debarko De, 2018

In our code the first layer is LSTM embedding layer and the second is a layer with 32 nodes and the same activation functions and optimizer as before are used here. The training set is said to have 5 epochs again and the summary of the model is as mentioned below.

```
Model: "model_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 250)               0
_________________________________________________________________
embedding_2 (Embedding)      (None, 250, 128)          552576
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                49408
_________________________________________________________________
dense_3 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_4 (Dense)              (None, 4)                 132
=================================================================
Total params: 604,196
Trainable params: 604,196
Non-trainable params: 0
```

**Fig 13**: Model Summary of RNN

The highest accuracy that we achieved here is 85%.

### 3.7.3 Long Short-Term Memory (LSTM)

LSTM systems have some interior relevant state cells that go about as long haul or momentary memory cells. The yield of the LSTM organize is regulated by the condition of these cells. This is a significant property when we need the forecast of the neural system to rely upon the recorded setting of sources of info, as opposed to just on the absolute last information (Moawad, 2018).

LSTM systems figure out on how to keep relevant data of contributions by coordinating a circle that permits data to flow from one stage to the next. These circles cause RNN to appear mystical. LSTM predictions are always conditioned by the past experience of the network's inputs (Moawad, 2018).

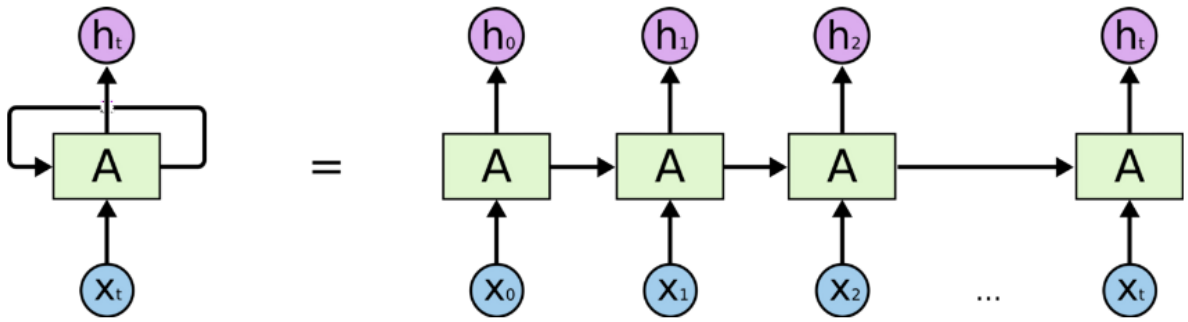



**Fig 14:** Architectural Diagram of LSTM

Moawad, 2018

In our code, Spatial dropout is used so that we can get rid of the noise in the data while we add 100 layers as the first layer. We use the same optimizers and loss function as before. The training set is again sent over 5 epochs and the summary is as below:

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 250)               0
_________________________________________________________________
embedding_1 (Embedding)      (None, 250, 128)          552576
_________________________________________________________________
flatten_1 (Flatten)          (None, 32000)             0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                1024032
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 132
=================================================================
Total params: 1,576,740
Trainable params: 1,576,740
Non-trainable params: 0
```

**Fig 15:** Model Summary of LSTM

The accuracy that we received here is 88%. Out of all the models it is LSTM that has given the highest accuracy.

### 3.7.4 Convolutional Neural Network (CNN-LSTM)

Convolution Neural Network is very important network in Deep learning (Georgakopoulos et al., 2018). This is used to perform very complex tasks with texts, images, audio, video etc.

CNN has a hidden layer called convolutional layer. The hidden layers will contain series of layers that convolve with multiplication or another dot product. RELU layer is the activation function and is followed by pooling layers and normalization layers, referred to as hidden layers as their inputs and outputs are masked by the activation function and final convolution.

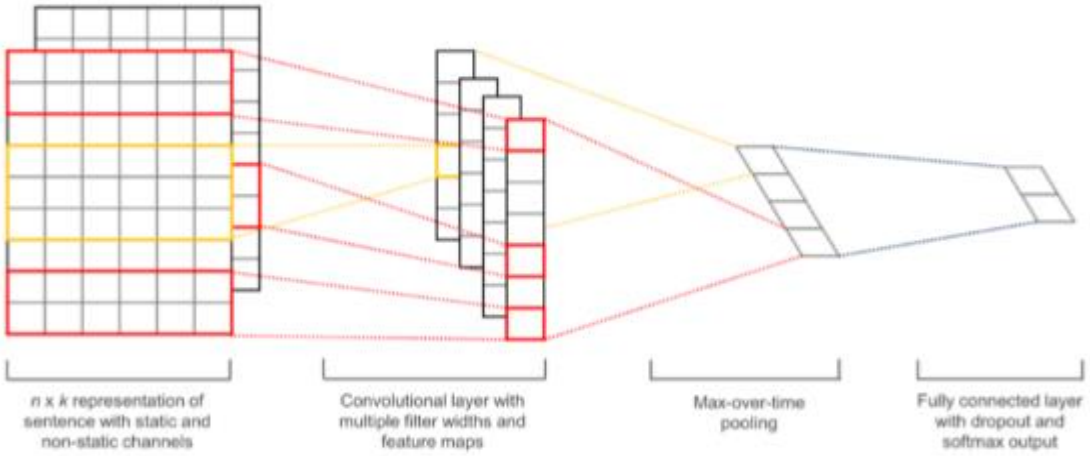



**Fig 16:** Architectural diagram of CNN-LSTM

Compsci, 2018

The design is allowed to have numerous non-convolutional layers too. But the basic of CNN is a convolutional layer. The function of it is to receive the input and transfer it to the next layer in the network. This basic transfer of neurons in this process is the convolutional part. And to accomplish this we need to specify the number of filters.

These filters detects the patterns. The precise meaning of patterns could be edged in images, words in sentences, objects like eyes, nose from faces, etc. The deeper the network the more sophisticated it is (Girshick, 2015a).

Pooling is included after a convolutional layer. If the past layer result is huge, the system will bring about moderate results. The pooling layer replaces certain areas of input portrayal with values processed from close by methods. This activity is prepared on each cut of the info portrayal or convoluted matrix. At last, decreased

or changed representation lessens the necessary calculation.

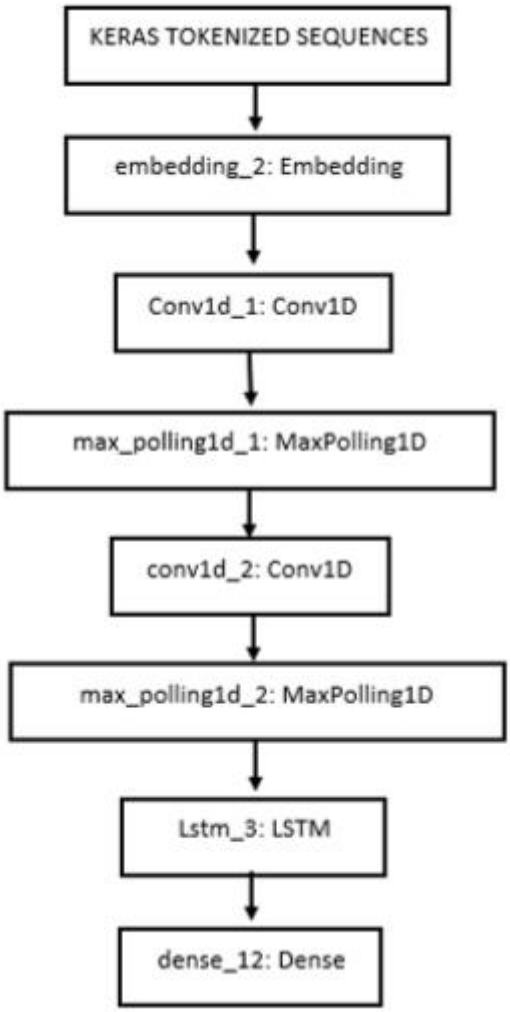



**Fig 17:** Block Diagram of CNN-LSTM

In our code, hidden layers include two convolution layers, two max pooling layers, and one LSTM layer. The initial Conv 1D layer consists of 64 neurons with 5 filters. The first and second max pooling layer behind is with a size of 4. The second layer being the Conv 1D layer consists of 64 neurons and 2 filters. Each of this layer has neurons with Relu (Rectified Linear unit) activation function. This is followed by the LSTM layer of 64 cells which is then connected

to the output layer comprising the 'softmax' activation function with 4 neurons. Here, we use the most efficient optimizer known as the 'adam' (Ba et al., 2014).

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 250)               0
_________________________________________________________________
embedding_1 (Embedding)      (None, 250, 128)          552576
_________________________________________________________________
flatten_1 (Flatten)          (None, 32000)             0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                1024032
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 132
=================================================================
Total params: 1,576,740
Trainable params: 1,576,740
Non-trainable params: 0
```

**Fig 19:** Model Summary of CNN-LSTM

After a huge number of trials this was the deep learning model that has given an accuracy of 72%.

### 3.7.5 Multi-Layer Perceptron with TF-IDF Tokenizer

Multilayer Perceptron is an Artificial Neural Network (ANN) type of Feedforward. It is the deep neural system engineering that is the most well-known sort. This comprises of three layers: an input layer, hidden layer and an output layer (Teufl et al., 2010).

Every hub in the system utilizes a nonlinear activation function except for input layer. For preparing, Multilayer Perceptron (MLP) utilizes supervised learning procedure called backpropagation (Bakliwal et al., 2011). MLP is used to apply in computer vision but now it has been succeeded by Convolution neural network. It has the characteristic of a fully connected layer in which each perceptron is connected with every other perceptron.

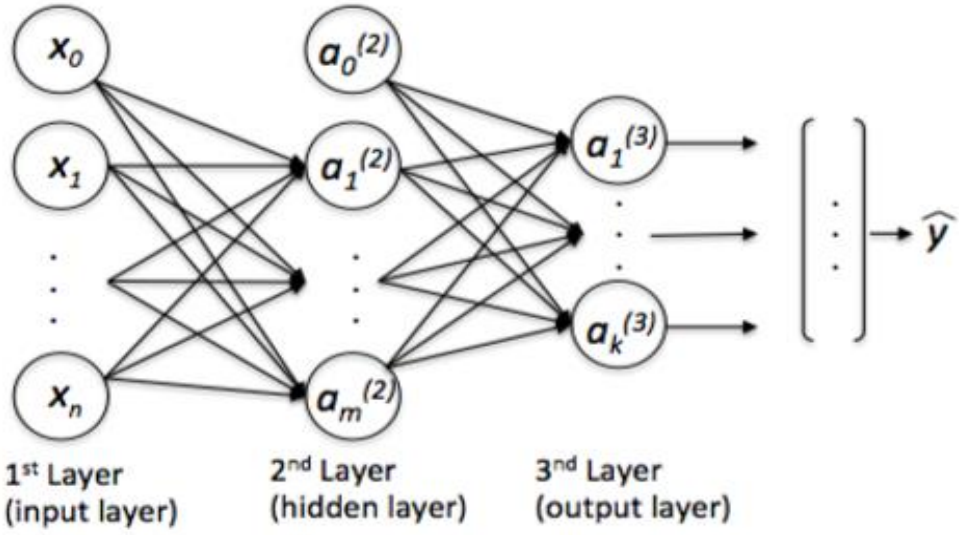



**Fig 19:** Basic Architecture of MLP

Rasbt

Every neuron in the input layer, output layer and hidden layers are connected to each neuron in the next and previous layer in MLP architectures. This is a kind of nonlinear classifier model that is used to fit high order dimensions. Though they are said to be a high performance model, increase in the hidden layers makes the model more complex. And even if we try to decrease the neurons by any

chance without an idea then it would result in a model that is underfitting with a less accuracy. At the same time more hidden layers also causes the problem of overfitting. With proper parameters chosen we can receive a good result. In this research, we examine how the different number of layers and the different number of neurons affect the accuracy of the multi classifier network.

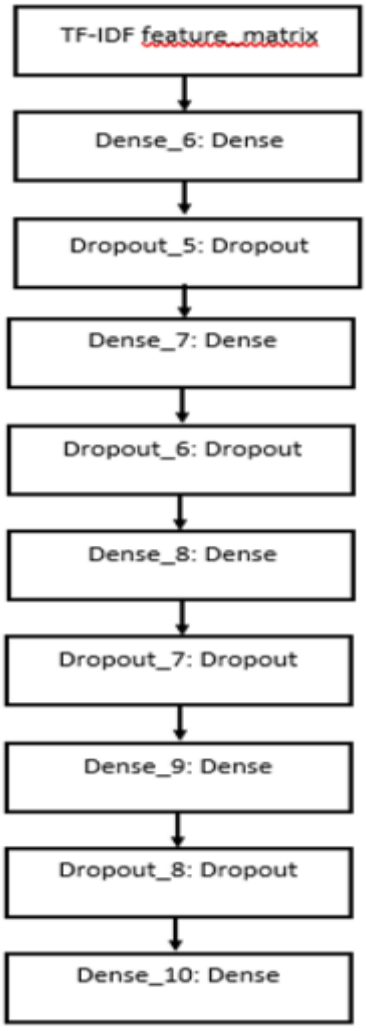



**Fig 20:** Block Diagram of MLP

I/P layer is of 500 neurons. Each hidden layer holds neurons with Relu (Rectified Linear Unit) function with maximum features to 5300. The first hidden layer

contained 550 neurons. The second hidden layer occupied 100 neurons and the third hidden layer contained 10 hidden neurons. The last layer is output layer with sigmoid activation function and four neurons. For compilation we have used the ‘adamax’ activation function. It is to look at the present value of the individual and past ingredients (Bushaev. V, 2018).

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_15 (Dense)             (None, 500)               2650500
_________________________________________________________________
dropout_5 (Dropout)          (None, 500)               0
_________________________________________________________________
dense_16 (Dense)             (None, 550)               275550
_________________________________________________________________
dropout_6 (Dropout)          (None, 550)               0
_________________________________________________________________
dense_17 (Dense)             (None, 100)               55100
_________________________________________________________________
dropout_7 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_18 (Dense)             (None, 10)                1010
_________________________________________________________________
dropout_8 (Dropout)          (None, 10)                0
_________________________________________________________________
dense_19 (Dense)             (None, 4)                 44
=================================================================
Total params: 2,982,204
Trainable params: 2,982,204
Non-trainable params: 0
```

**Fig 22:** Model Summary of MLP

But the highest accuracy that we could achieve using this model was 86% with an epoch of 5 and batch size of 150 along with a learning rate of 0.001.

## 3.8 Conclusion

In any case, it is straightforward and builds up an instinctive comprehension of neural systems and how a feeling-based issue, can be taken care of in the Deep Learning world. Deep learning is a black box-based model of critical thinking, so the outcomes change with various parameters. The client needs to get acquainted with the various parameters and how to mess with them to build up an instinctive comprehension of what parameters work for an issue close by (Sharma, 2019).

The primary goal of our research is to compare five deep neural networks on multi classification of educational video comments data for Hinglish text. The results of all the features used to design the models are being discussed in the next section. The results will be based on the following parameters:

| **Model-1** Simple Deep Learning | **Model-2** Recurrent Neural Network (RNN) | **Model-3** Long-Short Term Memory (LSTM) | **Model-4** Convolutional Neural Network (CNN-LSTM) | **Model-5** Multi-Layer Perceptron (MLP) |
|---|---|---|---|---|
| Accuracy | | | | |
| Precision | | | | |
| Recall | | | | |
| F1-Score | | | | |

**Tab 1:** Result Parameters for various models used in this research

| **Model-1** Simple Deep Learning | **Model-2** (RNN | **Model-3** LSTM | **Model-4** CNN-LSTM | **Model-5** MLP |
|---|---|---|---|---|
| Considering Learning Rate, Decay, Momentum each separately and then finally all 3 together to find out their Evaluation score | | | | |

**Tab 2:** Result Parameters after applying K-Fold

# Chapter 4: Results, Evaluation and Discussion

## 4.1 Introduction

This chapter describes the execution of this study's experiments. As we covered the data description, pre-processing stages already in the previous chapter, this chapter will deal with results and evaluation of the models.

The objective is to present a better model among the models selected to classify the YouTube comments correctly, whether they are in favor or against an educational channel/video.

The deep learning classification algorithms used in this section are using the Python Scikit learn library. Their performance against each other will also be compared.

The aim of this research is to provide the instructors with a system to classify their students according to their sentiments, so that they can address the issues during the online teaching accordingly. This may not help an instructor to change the method of teaching but can try to address the requests.

Our results should test the hypothesis of the research that sentiment analysis has the potential to classify the comments from the social media, whether students sentiments are with them or against them, in turn, can inform the instructor/channel to address the situation.

## 4.2 Results

This section will cover the output parameters received in each model and will help us decide on which model gives a better performance on our Multi-Class Hinglish dataset.

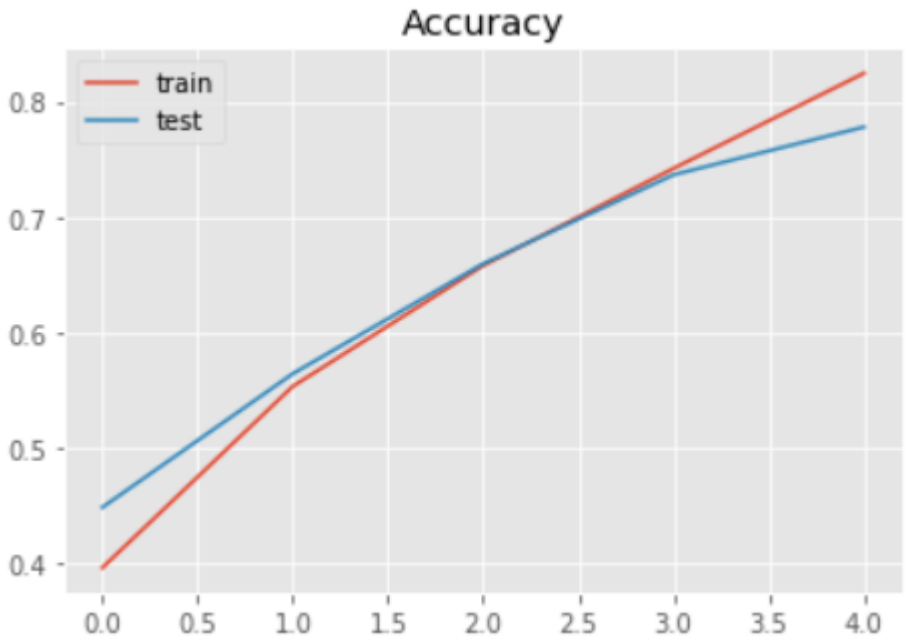
In this research we have applied five models and we are discussing the performance of them one by one below.

### 4.2.1 Simple Deep Learning Model Performance

Here, we have used very simple deep learning. The flattened layer acts as the input layer and the second layer with 32 nodes and RELU activation function is said to be the output layer here. Then, we try to compile the model using 'Categorical Cross-Entropy' as it is a multi-class problem.

With trial and error method it was observed that increase in batch size affected the accuracy badly and resulted in an underfitting model. However, the model is improved with a batch size of 64 and epoch size of 5 constant throughout the model.

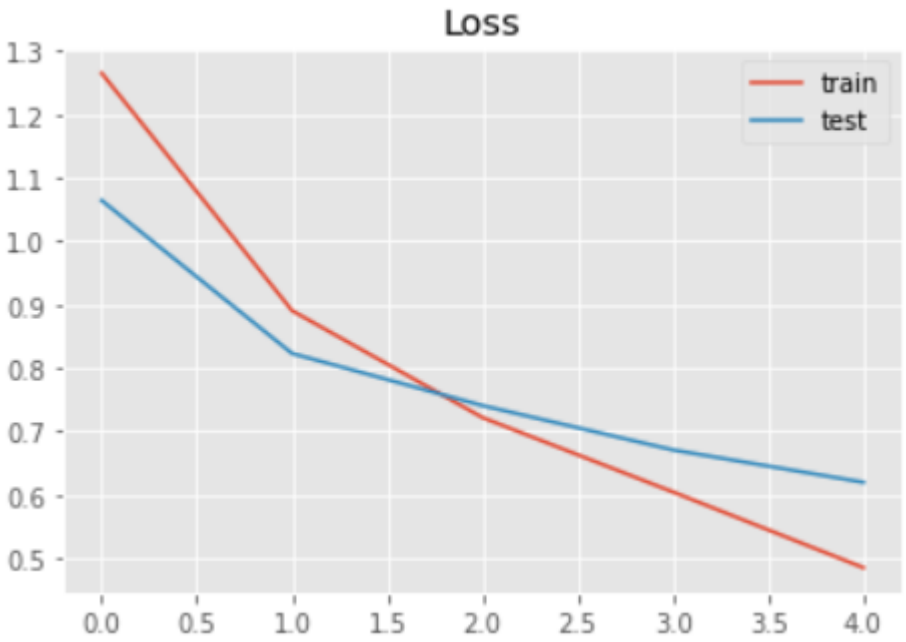
Accuracy is a metric that can be applied to classification tasks only. It describes what percentage of your test data are classified correctly. For example, in our research, we have multiple classification. If out of 100 test samples 85 is classified correctly then our accuracy would be 85%.




**Graph 2:** Train and Test set Accuracy for Simple Deep Learning Model

In this image here, we clearly see that RED line represents the accuracy of Train set and the BLUE line shows the accuracy of the test set. Always the train set has more data than the test set. So, in most of the cases the accuracy of the train set will be more than the test set.

Loss depends on how we predict classes for your classification problem. For example, in our research, our model uses probabilities to predict multiple class between 1 and 0. So if probability of Class-1 is 0.3, Class-2 is 0.2, Class-3 is 0.4 and Class-4 is obviously 0.1 to sum up to 1. Loss will be sum of the difference between predicted probability of the real class of the test set and 1.




**Graph 3**: Train and Test set Loss for Simple Deep Learning Model

Here, the loss for the train set is shown in RED and that of the test set is in BLUE. The train set holding the maximum part of the data will always have a loss lower than that of the test set as it is only a small part of the entire dataset.

So, a low accuracy and huge loss means we made huge errors on a lot of data and a great accuracy with low loss means we made low errors on a few data (best case).

With the accuracy recorded as 76%, the other results of the parameters achieved in this model are displayed below.

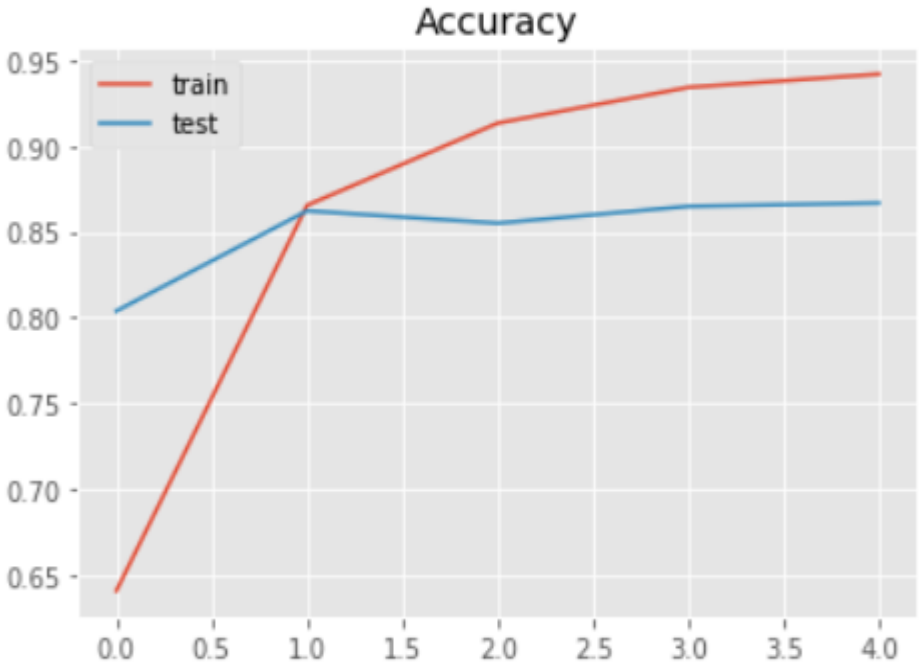
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.97 | 0.94 | 0.95 | 496 |
| 1 | 0.68 | 0.61 | 0.64 | 469 |
| 2 | 0.57 | 0.70 | 0.63 | 453 |
| 3 | 0.86 | 0.78 | 0.82 | 488 |
| accuracy | | | 0.76 | 1906 |
| macro avg | 0.77 | 0.76 | 0.76 | 1906 |
| weighted avg | 0.77 | 0.76 | 0.76 | 1906 |

**Fig 22:** Precision, Recall and F1-Score of Simple Deep Learning Model

### 4.2.2 Recurrent Neural Network Performance

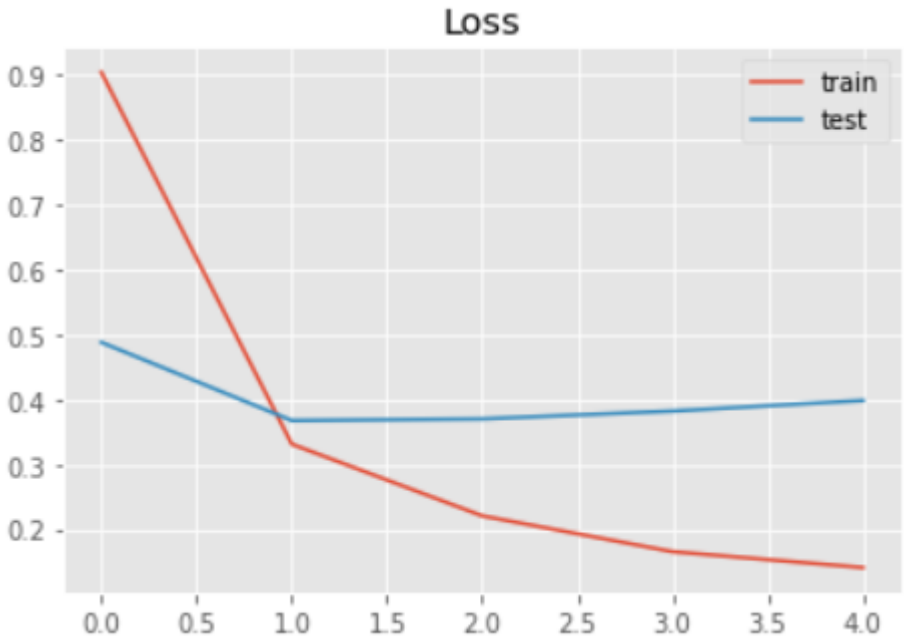
In this model we have used an LSTM layer with 64 nodes as input and output layer is the same as 32 nodes with RELU activation function used. We compile the model and fit it with a batch size of 64 and epoch size of 5 as the dataset is too huge and takes too much time for compilation.

The accuracy of the train and test set increases with increase in epochs. The maximum accuracy is achieved only by the train set whose graph is as shown below.




**Graph 4:** Accuracy of train and test set for RNN

Loss of the train and test set always decreases with increase in epochs. The minimum loss is achieved by the train set and the graph is displayed below.




**Graph 5:** Loss of train and test set for RNN

Accuracy being 85% we have the other parameters as recorded below.

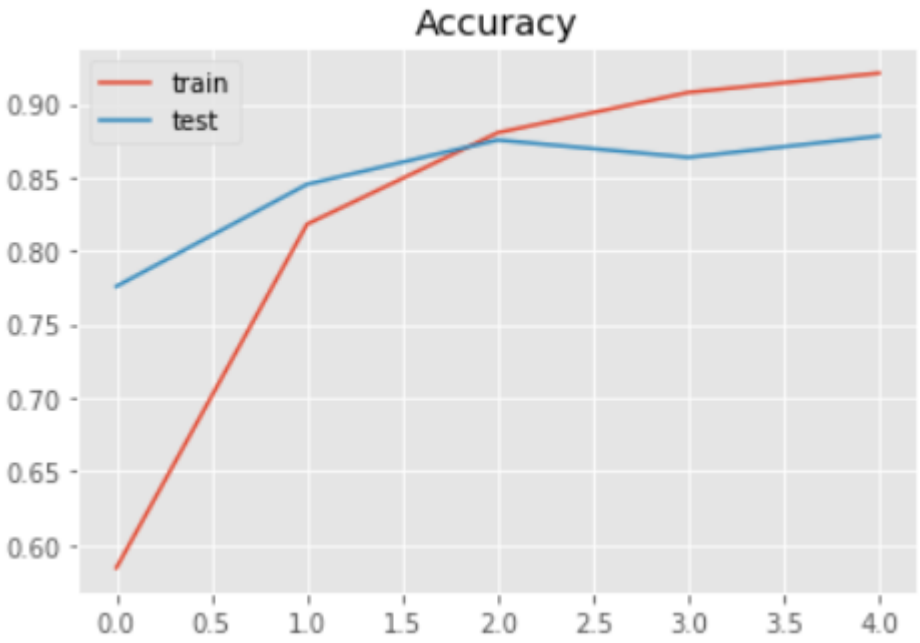
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 0.94 | 0.96 | 496 |
| 1 | 0.68 | 0.79 | 0.73 | 469 |
| 2 | 0.84 | 0.72 | 0.77 | 453 |
| 3 | 0.91 | 0.93 | 0.92 | 488 |
| accuracy | | | 0.85 | 1906 |
| macro avg | 0.85 | 0.85 | 0.85 | 1906 |
| weighted avg | 0.86 | 0.85 | 0.85 | 1906 |

**Fig 23:** Precision, Recall and F1-Score of RNN

### 4.2.3 Long Short-Term Memory (LSTM) Performance

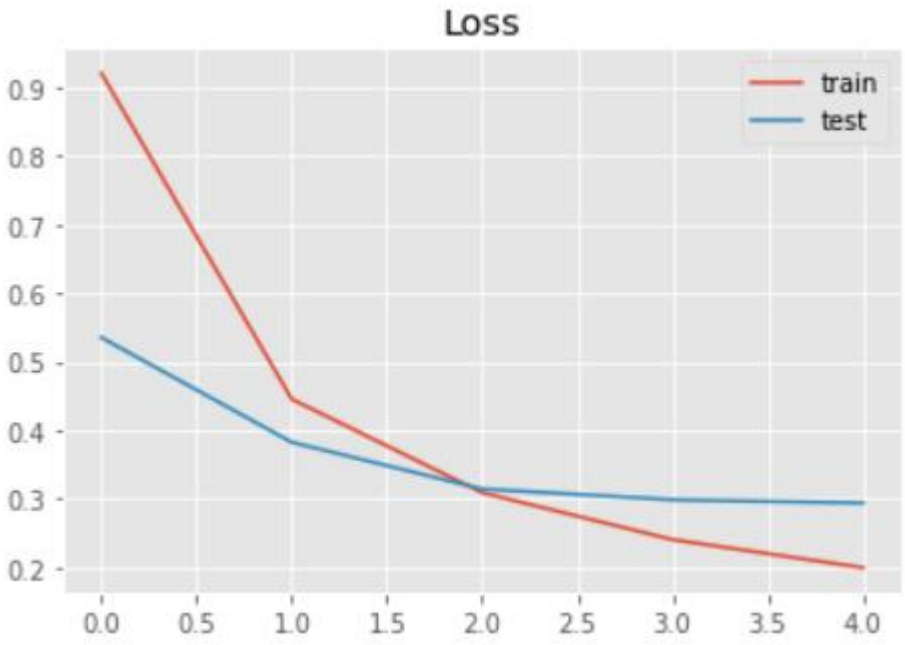
Here, we have used keras-to-sequence tokenizer for pre-processing data. This transforms each text to integer. Early Stopping is used here where we are allowed to give huge epochs to training set and can suddenly stop the training once the model performance stops to improve accuracy on a hold-out validation dataset. We have maximum number of words used to 5000 and maximum words in a record is set as 250. Then the tokenizer is fit on comments of the dataset. After splitting the data to X test, Y test, X train and Y train we start to build the model. Since LSTM is a collection of layers we have 3 input layers here. The first one has embedding layer, second one has a Spatial Dropout which gets rid of the noise in the data and the third one being an LSTM layer with 100 nodes and fourth layer is the output layer with 4 nodes representing the four classifications of our dataset. We finally fit the model with an epoch and batch size of 5 and 64 respectively.

This graph shows the accuracy of the train and test set where the train set obviously has the highest accuracy recorded.




**Graph 6:** Train and Test set Accuracy of LSTM

We also have the loss of the same being recorded where train set has the lowest loss recorded as epoch increases.




**Graph 7:** Train and Test set Loss of LSTM

We have the highest accuracy of 88% recorded here and this is the model which portrays that it is most suitable for our dataset. Other parameters are as mentioned below.

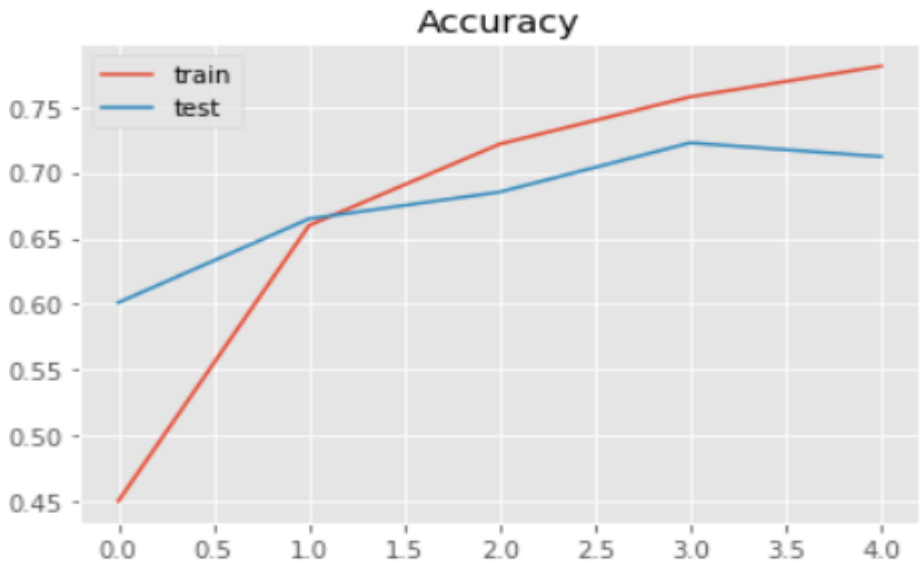
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.954 | 0.979 | 0.966 | 471 |
| 1 | 0.804 | 0.725 | 0.762 | 480 |
| 2 | 0.843 | 0.826 | 0.834 | 494 |
| 3 | 0.920 | 0.944 | 0.931 | 461 |
| micro avg | 0.882 | 0.867 | 0.874 | 1906 |
| macro avg | 0.880 | 0.868 | 0.874 | 1906 |
| weighted avg | 0.879 | 0.867 | 0.872 | 1906 |
| samples avg | 0.867 | 0.867 | 0.867 | 1906 |

**Fig 24:** Precision, Recall and F1-Score of LSTM

### 4.2.4 Convolutional Neural Network with LSTM (CNN-LSTM) Performance

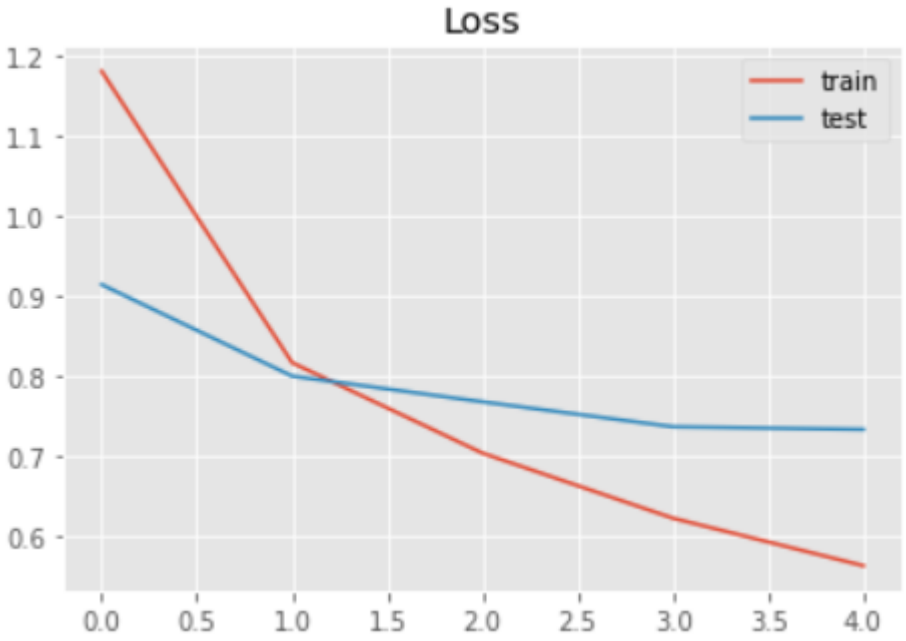
We have common pre-processing method as LSTM for CNN-LSTM as well. When we build the model, we have so many input layers here with the first layer acting as the Embedding layer. Rest of the 4 layers are alternative with Convolutional and Max Pooling Layers. Convolutional layer allows you to have large filter size and it creates a kernel that passes over a single dimension to produce tensor of outputs. And Max pooling enables dimensionality reduction and enables assumptions to be made out of features. The last layer is the output layer with 4 nodes. We then try to fit the model with a batch size of 64 and epoch size of 5.

The accuracy between the train and test set is as shown below




**Graph 8:** Train and Test set Accuracy of CNN-LSTM

The loss of the same is also displayed below.




**Graph 9:** Train and Test set Loss of CNN-LSTM

The accuracy recorded here is close to 72% and other parameters are recorded below.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.612 | 0.987 | 0.755 | 471 |
| 1 | 0.571 | 0.419 | 0.483 | 480 |
| 2 | 0.887 | 0.555 | 0.682 | 494 |
| 3 | 0.905 | 0.926 | 0.915 | 461 |
| micro avg | 0.722 | 0.717 | 0.720 | 1906 |
| macro avg | 0.744 | 0.722 | 0.709 | 1906 |
| weighted avg | 0.744 | 0.717 | 0.707 | 1906 |
| samples avg | 0.717 | 0.717 | 0.717 | 1906 |

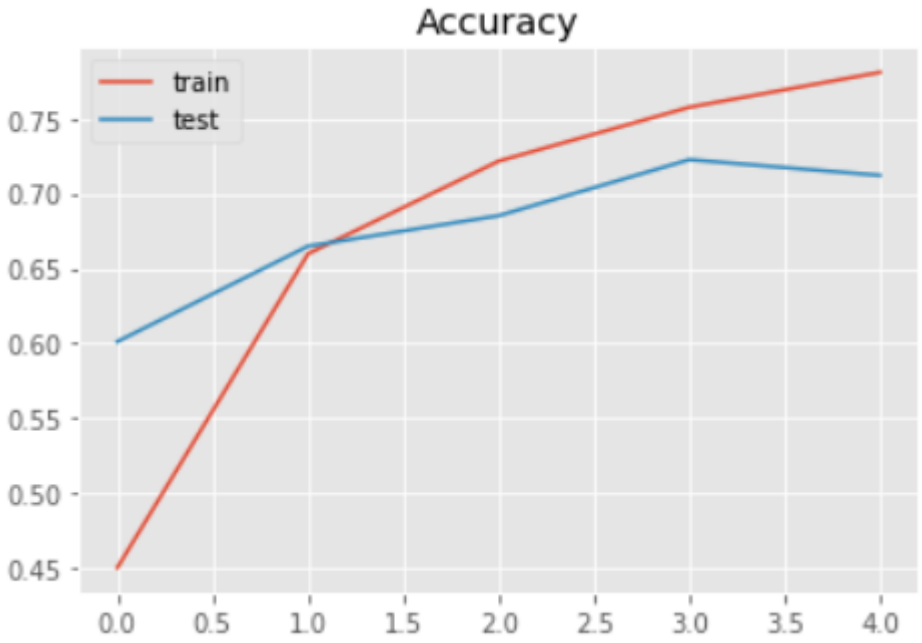
**Fig 25:** Precision, Recall and F1-Score of CNN-LSTM

### 4.2.5 Multi-Layer Perceptron with TF-IDF Vectorizer Performance

We have different pre-processing techniques used here. We have trained MLP network using Vectorization = TF-IDF which involves Stemming, Lemmatization and stop words =PUNCTUATIONS. The MLP model is using 3 hidden layers with 0.1 dropouts. Input layer comprises of 500 neurons. The first hidden layer has 550 neurons, the second has 150 and the third hidden layer accommodates 10 neurons. The output layer has four neurons. Followed by this we have vectorizer fit and train test data split.

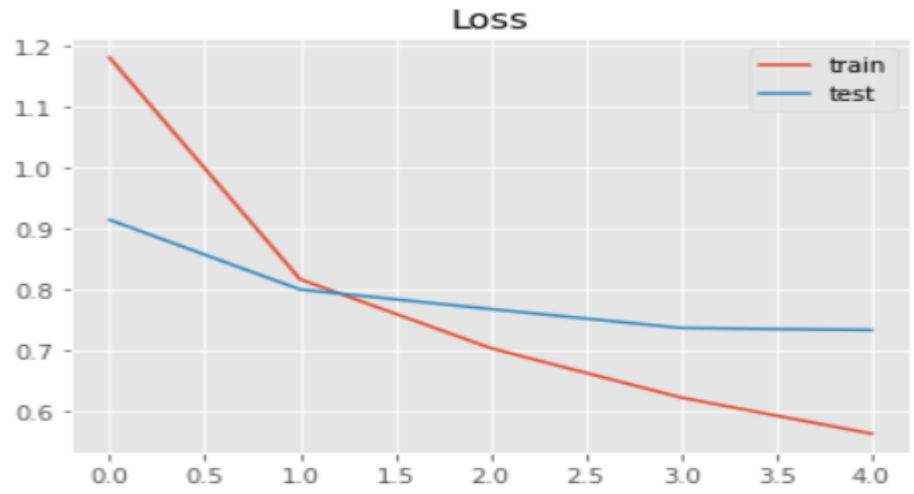
With trial and error, we can say decrease in learning rate and increase in batch size reduces the overall accuracy of the model which is underfitting. However, the model is further improved to achieve the highest accuracy by increasing the learning rate from 0.001 to 0.01, 5 epoch and 150 batch size, considering these features as constant throughout the model. Then we compile and fit the model with ‘adamax’ optimizer and ‘sigmoid’ activation function.

The accuracy graph between the train and test set is displayed below.




**Graph 10:** Train and Test set Accuracy of MLP

The loss graphs may all look the same like the ones before as they all portray the same information that loss decreases as the epochs increase and that train set records the lowest loss.




**Graph 11:** Train and test set Loss of MLP

We have an accuracy of 86% recorded here and other parameters are as shown below.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.949 | 0.955 | 0.952 | 471 |
| 1 | 0.609 | 0.704 | 0.653 | 480 |
| 2 | 0.762 | 0.870 | 0.813 | 494 |
| 3 | 0.985 | 0.829 | 0.900 | 461 |
| micro avg | 0.808 | 0.839 | 0.823 | 1906 |
| macro avg | 0.826 | 0.840 | 0.830 | 1906 |
| weighted avg | 0.824 | 0.839 | 0.828 | 1906 |
| samples avg | 0.768 | 0.839 | 0.792 | 1906 |

**Fig 26:** Precision, Recall and F1-Score of MLP

## 4.3 Hyper parameter tuning with K-Fold Cross validation

Hyperparameter tuning is a term that goes hand in hand with K-Fold Cross validation. The parameters used to define a model and then the process of searching for a perfect model by changing the model parameters is known as hyperparameter tuning (Jordan. J., 2017).

Cross validation is using the limited sample of data and testing the model performance rather than on the training set. The general procedure of K-fold validation is:

Shuffle the dataset, split to k-groups (generally 10), then split data to train and test set, fit the model on training set and test it on the test set and finally get the

evaluation score (Brownlee, J., 2018). The parameters that I have considered here are Learning Rate, Decay, Momentum (each of them separately) and then considering all of them together to see each model performance with K-Folds applied to them.

**Learning Rate:** The number of times the weights are updated during training a model

**Decay:** Learning Rate / Epochs

**Momentum:** Tricks to speed up optimization methods like gradient descent.

```
from keras.models import Model
epochs=5
learning_rate = 0.1
decay_rate = learning_rate / epochs
momentum = 0.8
sgd = SGD(lr=learning_rate, momentum=momentum, decay=decay_rate, nesterov=False)
```

```
def create_baseline():
  model_LSTM=Sequential()
  model_LSTM.add(Embedding(MAX_NB_WORDS, EMBEDDING_DIM, input_length=X.shape[1]))
  model_LSTM.add(SpatialDropout1D(0.2))
  model_LSTM.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
  model_LSTM.add(Dense(4, activation='softmax'))
  model_LSTM.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
  return model_LSTM
```

```
# evaluate model with standardized dataset
from sklearn.model_selection import StratifiedKFold


estimator = KerasClassifier(build_fn=create_baseline, epochs=5, batch_size=64, verbose=0)
kfold = StratifiedKFold(n_splits=10, shuffle=True)
results = cross_val_score(estimator, X, y, cv=kfold)
print("Baseline: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))
```

```
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/indexed_slices.py:434: UserW
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
Baseline: 87.71% (1.25%)
```

**Fig 27:** Example of K-Fold Cross validation

## 4.4 Discussion

Interesting results were observed for all the models. In this research, having all the parameters being measured, it is concluded that all the models are assessing semantic meaning rather than syntactic meaning.

The models are compared based on their Accuracy, Precision, Recall and F1-Score. Recall expresses the ability to find the relevant instances in the data set whereas precision express the proportion of the data instances our model says was relevant which were truly relevant. We can find the optimal blend of precision and recall, by combining both the metrics using F1-score. It is a harmonic mean of precision and recall.

Let us tabulate all the results of all the models and discuss.

| **Model-1** Simple Deep Learning | **Model-2** Recurrent Neural Network (RNN) | **Model-3** Long-Short Term Memory (LSTM) | **Model-4** Convolutional Neural Network (CNN-LSTM) | **Model-5** Multi-Layer Perceptron (MLP) |
|---|---|---|---|---|
| Accuracy=76 | Accuracy=85 | Accuracy=88 | Accuracy=72 | Accuracy=86 |
| Precision=0.77 | Precision=0.86 | Precision=0.88 | Precision=0.74 | Precision=0.82 |
| Recall=0.76 | Recall=0.85 | Recall=0.87 | Recall=0.72 | Recall=0.84 |
| F1Score=0.76 | F1Score=0.85 | F1Score=0.87 | F1Score=0.71 | F1Score=0.83 |

**Tab 3:** Results showing all the parameters of all models applied

By looking at the results that are tabulated, we see very close accuracies recorded in 3 models RNN, LSTM and MLP among which LSTM has recorded the highest

and proves that this is the best suited deep learning model for our multi-class Hinglish dataset.

And now, depending on the accuracies, the values of precision, recall and F1 score of every model is evaluated and is observed that they are hand in hand with their accuracies.

By taking all parameters into consideration, LSTM stands first in perfect model fit that suits our dataset while MLP is the second as it has the second highest accuracy recorded along with parameters and finally comes RNN.

After applying K-Fold to the models with different parameters, these were the results observed.

| **Model-1** Simple Deep Learning | **Model-2** Recurrent Neural Network (RNN) | **Model-3** Long-Short Term Memory (LSTM) | **Model-4** Convolutional Neural Network (CNN-LSTM) | **Model-5** Multi-Layer Perceptron (MLP) |
|---|---|---|---|---|
| LR=48.36 | LR=51.41 | LR=88.06 | LR=87.32 | LR=68.47 |
| Decay=49.77 | Decay=50.70 | Decay=87.86 | Decay=75.51 | Decay=69.73 |
| Mom=45.38 | Mom=87.82 | Mom=88.02 | Mom=75.36 | Mom=65.32 |
| All 3=47.39 | All 3=87.35 | All 3=87.71 | All 3=75.59 | All 3=67.58 |

**Tab 4:** Result showing all parameters after applying K-Fold

Even in K-Fold cross validation LSTM seems to be the model that best suits our dataset. When considering Learning rate, decay and momentum separately, they

still show high values and finally when considering all 3 parameters together they still have highest evaluation score.

Knowing the true positives here is very important as we are dealing with the sentiment of students and the model with the highest and best precision helps in it.

From the findings and the evaluations discussed above, we can say LSTM is better performing model for classification of Hinglish texts than other models rejecting the null hypothesis.

# Chapter 5: Conclusion and Future Work

## 5.1 Introduction

This chapter contains the findings of this inquiry. It summarizes the research overview, problem definition, design / experiment design, evaluation and results as discussed in this research's previous chapters. And it addresses the contributions and effects of research and future work as well as proposals for further studies that can be carried out in this area.

## 5.2 Research Overview

To know the sentiments of the people, there are lot of ways to get opinions. It can be from survey, social media etc. This study has developed a system utilizing deep learning models to computationally identify and categorize the sentiments expressed on educational videos, in order to predict if a students' sentiments is in favor or against them.

## 5.3 Problem Definition

The research problem was 'Can deep learning classifier effectively classify the Hinglish text on educational video comments dataset?' The primary purpose of the research was to validate the following hypothesis:

**Null Hypothesis:** LSTM, a deep learning model is trained to classify Hinglish text on educational channel video dataset. Does this give significantly give better

accuracy than CNN-LSTM, RNN, etc., other artificial neural network algorithms trained on the same dataset.

**Alternative Hypothesis:** LSTM, a deep learning model is trained to classify Hinglish text on educational channel video dataset. Does this not give significantly give better accuracy than CNN-LSTM, RNN, etc., other artificial neural network algorithms trained on the same dataset.

## 5.4 Design/Experimentation, Evaluation and Results

In this research we have used CRISPM-DM approach for model building. The dataset collected is understood, prepared as per the requirement, pre-processed, trained the models and evaluated their results. Different techniques were used to label the data as per the research requirement, then the data was pre-processed using various techniques and then finally feature selection before modeling. We have used Holdout method for data splitting.

Interesting results were observed for all the models. In this research we are evaluating the models with respect to different parameters. After all the evaluation it is seen that the models are assessing semantic meaning rather than syntactic meaning.

All the models are compared based on the parameters recorded and we see that

accuracy, precision, recall and F1-Score of LSTM seems to be the highest among the models that were built. And even when K-Fold hyperparameter tuning was

done, considering the parameters in it, again there LSTM proved to be the best in evaluation score.

Hence, from the above results it is observed that LSTM trained on the educational comments dataset to classify the Hinglish text has better accuracy than CNN-LSTM, RNN, etc. hybrid network trained on same dataset, which rejects the null hypothesis.

## 5.5 Contributions and Impact

In the current research, a thorough and detailed analysis of processing of the data collected from YouTube comments was performed. This work will add to the body of knowledge of sentiment analysis as it incorporates sentiment analysis, concentrating on the education sphere based on comments from social media. It may be of extra value as it is focused on a narrow domain area based on the sentiment of Indian students towards video/topic. The results may prove useful for the channel to strategize and improve their methods of teaching.

Research on Hinglish text is not the most common stream in text analysis, but this work mainly concentrates only on it.

## 5.6 Future work and Recommendations

Future work could look into exploring different languages as this research is limited to use of Hindi and English languages. The data is gathered from comments on an educational video published on the YouTube. Further we can

consider extracting the data from the YouTube video using some new bot, so that can increase the sample size.

This research is limited to multi-class classification as we are considering 4 labels which was a little difficult when it comes to pre-processing and manual labeling. To make it simple and easier, further the same can be considered for binary classification problem to make the problem simple.

We can also use Fast R-CNN concept that was applied for object detection in our theory of sentiment detection in order to achieve high accuracy over models.

Also, the extended LRP version to a bi-directional LSTM model for the sentiment prediction of sentences might give better results  demonstrating that the word relevance's trustworthy reveal words supporting the classifier's decision for or against a specific class, and perform better than those obtained by a gradient-based decomposition.

# Bibliography

Anonymous. (2017, April 6).6 Unusual but Educational YouTube Channels to Inspire Your Students To Look Beyond Academia. Retrieved from https://teach.com/blog/6-unusual-but-educational-youtube-channels/

Bakliwal, A., Arora, P., Patil, A., & Varma, V. (2011). Towards enhanced opinion classification using NLP techniques. In Proceedings of the workshop on sentiment analysis where ai meets psychology (SAAIP 2011) (pp. 101–107).

Bakliwal, A., Foster, J., van der Puil, J., O'Brien, R., Tounsi, L., & Hughes, M. (2013). Sentiment analysis of political tweets: Towards an accurate classifier.

Botzenhardt, A., Witt, A., & Maedche, A. (2011). A text mining application for exploring the voice of the customer. In Amcis.

Brame, C.J. (2015). Effective educational videos. Retrieved from http://cft.vanderbilt.edu/guides-sub-pages/effective-educational-videos/

Brownlee, J., (2018, May 23). A Gentle introduction to K-Fold Cross Validation. Retrieved from: https://machinelearningmastery.com/k-fold-cross-validation/

Bushaev, V. (2018, Oct 22). Adam — latest trends in deep learning optimization. Retrieved from https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c

Chen, Y., & Xie, J. (2008). Online consumer review: Word-of-mouth as a new element of marketing communication mix. Management science, 54(3), 477–491.

CHI Software. (2019, May 20). Supervised vs. Unsupervised Machine Learning. Retrieved from https://medium.com/@chisoftware/supervised-vs-unsupervised-machine-learning-7f26118d5ee6

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multi-task learning. In Proceedings of the 25th international conference on machine learning (pp. 160–167).

Compsci. (2018, Feb 20). Twitter Sentiment Analysis using combined LSTM-CNN models. Retrieved from konukoii.com/blog/2018/02/19/twitter-sentiment-analysis-using-combined-lstm-cnn-models/

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364

Daelemans, W., & Hoste, V. (2002). Evaluation of machine learning methods for natural language processing tasks. In 3rd international conference on language resources and evaluation (lrec 2002).

De, D. (2018, March 1). RNN or Recurrent Neural Network for Noobs. Retrieved from https://hackernoon.com/rnn-or-recurrent-neural-network-for-noobs-a9afbb00e860

Fitzgerald, S., Mathews, G., Morris, C., & Zhulyn, O. (2012). Using NLP techniques for file fragment classification. Digital Investigation, 9, S44–S49.

Garg, A., Gupta, D., Saxena, S., & Sahadev, P. P. (2019). Validation of random dataset using an efficient CNN model trained on MNIST handwritten dataset. In 2019 6th international conference on signal processing and integrated networks (spin) (pp. 602–606).

Georgakopoulos, S. V., Tasoulis, S. K., Vrahatis, A. G., & Plagianakos, V. P. (2018). Convolutional neural networks for toxic comment classification. In Proceedings of the 10th Hellenic conference on artificial intelligence (p. 35).

Ghosal, D., Bhatnagar, S., Akhtar, M. S., Ekbal, A., & Bhattacharyya, P. (2017). IITP at semeval-2017 task 5: an ensemble of deep learning and feature based models for financial sentiment analysis. In Proceedings of the 11th international workshop on semantic evaluation (semeval-2017) (pp. 899–903).

Girshick, R. (2015a). Fast R-CNN. In Proceedings of the IEEE international conference on computer vision (pp. 1440–1448).

Girshick, R. (2015b). Proceedings of the IEEE international conference on computer vision. IEEE Computer Society Washington, DC, USA.

Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 v1 [cs. CL], 21–22.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. Neurocomputing, 187, 27–48.

Hassan, A., Amin, M. R., Mohammed, N., & Azad, A. (2016). Sentiment analysis on Bangla and Romanized Bangla text (brbt) using deep recurrent models. arXiv preprint arXiv:1610.00369.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., ... others (2012). Deep neural networks for acoustic modeling in speech recognition. IEEE Signal processing magazine, 29.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(02), 107–116.

Hsu, C.-F., Khabiri, E., & Caverlee, J. (2009). Ranking comments on the social web. In 2009 international conference on computational science and engineering (Vol. 4, pp. 90–97).

Jordan, J., (2017, Nov 2). Hyperparameter tuning for Machine Learning Models.

Kaur, G., Kaushik, A., & Sharma, S. (2019). Cooking is creating emotion: A study on Hinglish sentiments of YouTube cookery channels using semi-supervised approach. Big Data and Cognitive Computing, 3(3), 37.

Kaushik, A., & Chaudhari, S. (2019, May). Comments on YouTube videos of top two Indian political parties. Zenodo. Retrieved from https://doi.org/10.5281/zenodo.3055456

Kaushik, A., & Naithani, S. (2015). A study on sentiment analysis: Methods and tools. International journal of Science and Research, 4, 287–291.

Kharde, V., Sonawane, P., et al. (2016). Sentiment analysis of twitter data: a survey of techniques. arXiv preprint arXiv:1601.06971.

Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Image Net classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097–1105).

Lee, S., & Choeh, J. Y. (2014). Predicting the helpfulness of online reviews using multilayer perceptron neural networks. Expert Systems with Applications, 41(6), 3041–3046.

Mitchell, T. M. (1999). Machine learning and data mining. Communications of the ACM, 42(11).

Moawad, A. (2018, Feb 2). The magic of LSTM neural networks. Retrieved from https://medium.com/datathings/the-magic-of-lstm-neural-networks-6775e8b540cd

Nicholson, C. A Beginner's Guide to Neural Networks and Deep Learning. Retrieved from https://pathmind.com/wiki/neural-network

Pearce, K. (2017, June 14). Edutainment! The best YouTube Education Channels for Online Learning. Retrieved from https://www.diygenius.com/best-educational-youtube-channels/

Rasbt. MultiLayer Perceptron. Retrieved from 4http://rasbt.github.io/mlxtend/userguide/classifier/MultiLayerPerceptron

Resnik, P., & Lin, J. (2010). 11. evaluation of NLP systems. The handbook of computational linguistics and natural language processing, 57.

Retrieved from: https://www.jeremyjordan.me/hyperparameter-tuning/

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. Journal of documentation, 60(5), 503–520.

Ronaghan, S. (2018, Aug 14). Deep Learning: Common Architectures. Retrieved from https://medium.com/@srnghn/deep-learning-common-architectures-6071d47cb383

Sabri, B., & Saad, S. (2016). Arabic sentiment analysis with optimal combination of features selection and machine learning approaches. Research Journal of Applied Sciences, Engineering and Technology, 13(5), 386–393.

Selmi, Z., Halima, M. B., & Alimi, A. M. (2017). Deep learning system for automatic license plate detection and recognition. In 2017 14th IAPR international conference on document analysis and recognition (ICDAR) (Vol. 1, pp. 1132–1138).

Sharma, A. Confusion Matrix in Machine Learning. Retrieved from https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

Singh, K., Sen, I., & Kumaraguru, P. (2018). Language identification and named entity recognition in Hinglish code mixed tweets. In Proceedings of ACL 2018, student research workshop (pp. 52–58).

Teufl, P., Payer, U., & Lackner, G. (2010). From nlp (natural language processing) to MLP (machine language processing). In International conference on mathematical methods, models, and architectures for computer network security (pp. 256–269).

Tharwat, A. (2018). Classification assessment methods. Applied Computing and Informatics.

Van Looy, A. (2016). Sentiment analysis and opinion mining (business intelligence 1). In Social media management (pp. 133–147).

Wang, Y., Huang, M., Zhao, L., et al. (2016). Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the 2016 conference on empirical methods in natural language processing (pp. 606–615).

Wirth, R., & Hipp, J. (2000). Crisp-dm: Towards a standard process model for data mining. In Proceedings of the fourth international conference on the practical application of knowledge discovery and data mining (pp. 29–39).