16 MBTI Prediction Based On Social Text

The Use Of Classical Classification to Distinguish between 16 MBTI given a vectorized text using CBOW, BERT Models vs Classification using The LSTM model

Written By

Alhossien Waly

Ali Ibrahim

January 6 2024

Contents

1 Abstract

In this research endeavor, our primary objective was to develop a classification framework for the Meyers-Briggs Type Indicator (MBTI) based on social media posts. We adopted a dual-pronged approach to address this challenge. Initially, we employed a Long Short-Term Memory (LSTM) neural network model to categorize vectorized text into one of the 16 MBTI types. Subsequently, we took a dimensionality reduction approach, breaking down the 16 MBTI categories into their 4 fundamental dimensions which define each unique personality. We then applied traditional classification techniques to the vectorized text outputs from two Natural Language Processing (NLP) models, namely BERT and CBOW.

In the second phase of our approach, we leveraged a set of six distinct models to optimize our classification results. This comprehensive strategy allowed us to conduct a thorough and accurate comparative analysis of the outcomes produced by the BERT and CBOW models. Our findings contribute to a nuanced understanding of the effectiveness of different NLP models in the context of MBTI classification, paving the way for enhanced accuracy and insights into personality prediction based on social media content

2 Introduction

2.1 Background Knowledge

In this world, human beings' personalities are divided into 16 unique personalities according to the MBTI.
The Myers-Briggs Type Indicator (MBTI) is a widely used psychological tool designed to assess and categorize personality types based on four dichotomies, resulting in a total of 16 distinct personality types. Developed by Katharine Cook Briggs and her daughter Isabel Briggs Myers, the MBTI is grounded in Carl Jung's theory of psychological types. The four dichotomies that form the basis of the MBTI are:

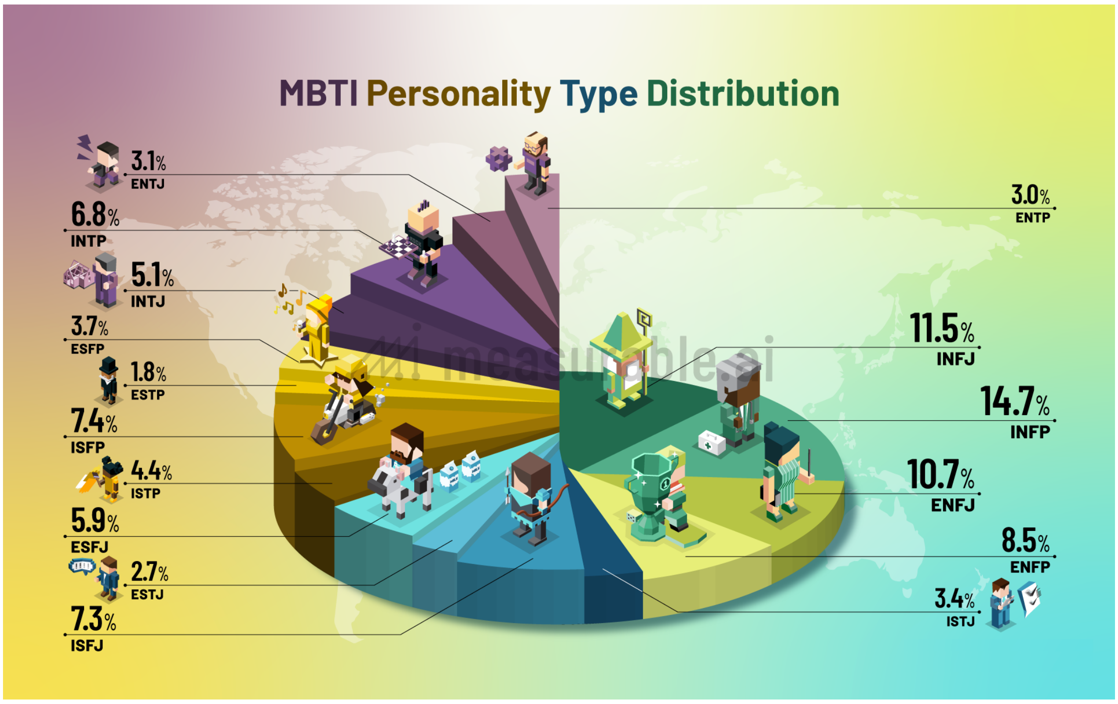


Figure 2.1: approximate Distribution of the 16 MBTI WorldWide according to statistics

Extraversion (E) vs. Introversion (I): Reflects the source and direction of energy. Extraverts are generally outgoing and derive energy from interacting with others, while introverts are more reserved and recharge through solitude.

Sensing (S) vs. Intuition (N): Describes the preferred method of gathering information. Sensing types rely on concrete, tangible details, whereas intuitive types are more inclined to focus on possibilities and potential.

Thinking (T) vs. Feeling (F): Examines the decision-making process. Thinking types prioritize logic and objective analysis, while feeling types consider personal values and the impact on individuals involved.

Judging (J) vs. Perceiving (P): Addresses an individual's approach to organizing and dealing with the external world. Judging types prefer structure and order, while perceiving types are more adaptable and spontaneous. These dichotomies result in 16 unique personality types, such as ISTJ (Introverted, Sensing, Thinking, Judging), ENFP (Extraverted, Intuitive, Feeling, Perceiving), and so on. The MBTI framework is often used in various settings, including personal development, career counseling, team building, and communication training.

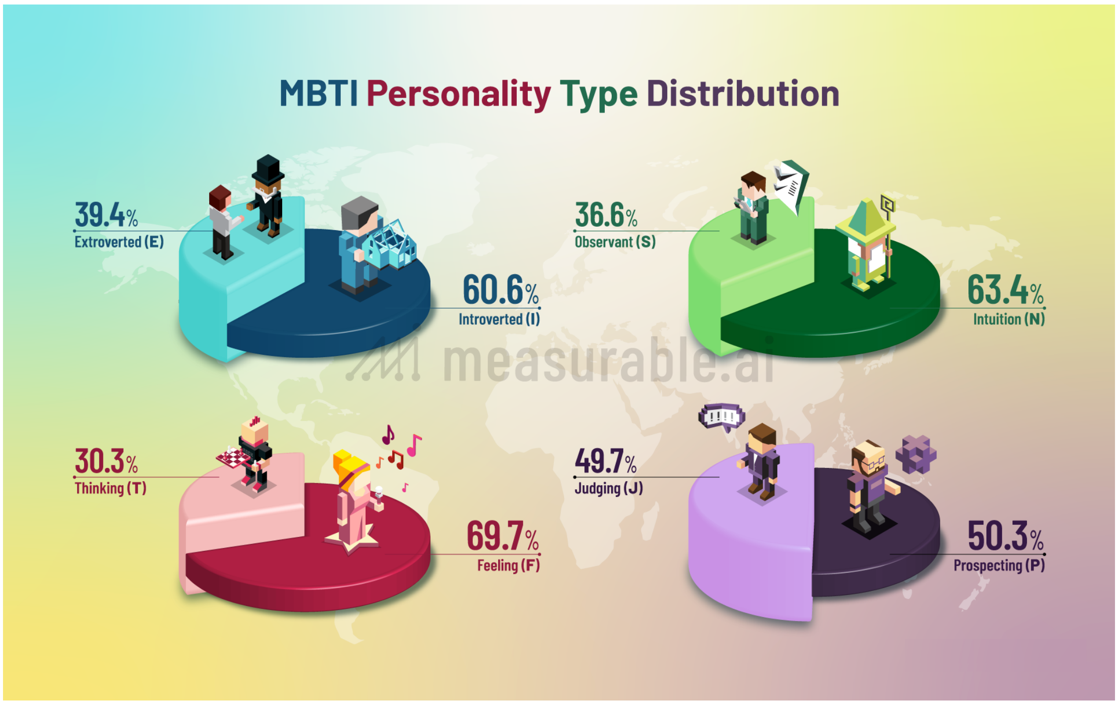


Figure 2.2: 4 dimensions of MBTI distribution WorldWide

While the MBTI has gained popularity, it is essential to approach it with a nuanced understanding. Critics point out limitations, such as the binary nature of the dichotomies and the lack of empirical evidence supporting some aspects of the theory. Nevertheless, the MBTI continues to be a valuable tool for self-reflection and understanding interpersonal dynamics. There is another way to classify the human personality called DISC. DISC is a psychological profiling tool that is used to assess and analyze human behavior. The DISC model is based on the work of

psychologist William Moulton Marston, who developed a theory that individuals exhibit predictable behavior patterns based on their personality traits. The DISC model categorizes these traits into four primary behavioral styles, each represented by a letter in the acronym DISC:
Dominance (D): Individuals with a dominant style are assertive, competitive, and goal-oriented. They tend to be direct and decisive, focusing on results and taking charge of situations.
Influence (I): People with an influential style are social, outgoing, and persuasive. They enjoy interacting with others, are enthusiastic, and often seek to influence and inspire.
Steadiness (S): Those with a steady style are cooperative, patient, and supportive. They value teamwork, seek harmony in relationships, and are often good listeners.
Conscientiousness (C): Individuals with a conscientious style are analytical, detail-oriented, and systematic. They value accuracy, follow rules, and prioritize quality in their work.
We Will be focusing on the MBTI in this experiment.

3 Methodology

3.1 Data Preparation

3.1.1 Dataset

Using Kaggle datasets we used 3 datasets combined to get a combined dataset of **122553** entities.
The dataset consists of 2 columns a post-column (Contains social post text), and a label-column (Contains the personality type of the person who wrote the post). The obtained data has a distribution that differs from real-world statistics as shown in **figure 3.1**.

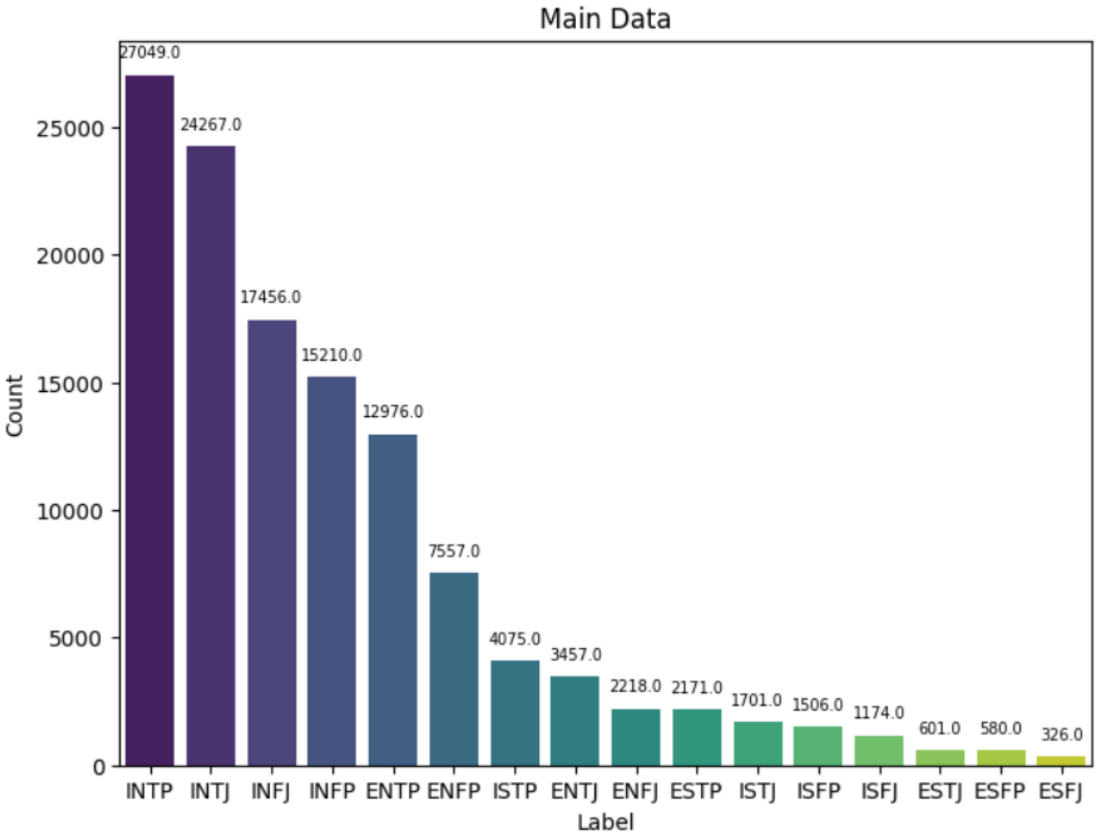


Figure 3.1: Distibution of the combined dataset

3.1.2 Data Preparation

To use the data that we have in both approaches that we have, we need to process it to get it in form and defects-free.

Data Cleaning

The datasets aren’t clean and the following defects is considered outliers
* **Removing Tags** all texts that starts with “@”sign
* **Removing links** In scrapping any video or photo is taken as link-text
* **Replacing |||| with new line** new line was read in scraping as ||||
* **HTML Encoding** removing all “& gt;” “& lt;3”A4 $and \n
* **Some Unknown Patterns** removing . . . and, “:WORD:”
* **Making all Labels uppercase** the labels varies in the combined dataset
* **Removing Duplicates** There are 229 duplicate data in our dataset

Preparing 4 dims dataframes

* **Dividing**
we divide our dataset into 4 datasets changing the labels into either [I,E], [N,S], [T,F],or [J,P]. Preparing it for the binary classification performed in the second approach.

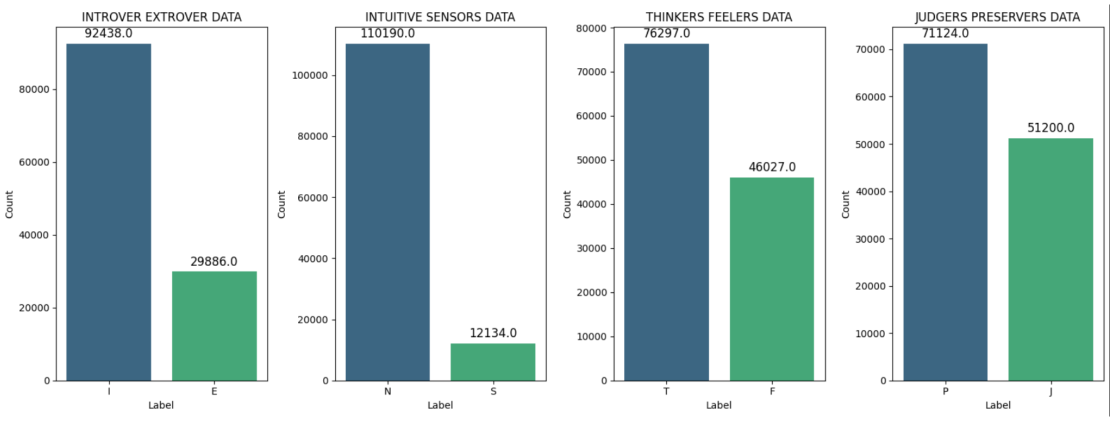


Figure 3.2: Distribution of Each dimension after changing the main data label scope

Note that the previous distribution is related only to the dataset and has nothing to do with real-world distribution
* **Equalizing**
To prepare the data for a fair chance of each class being classified in the output, we drop randomly values from each major class of the 4 dimensions.
* **Label Encoding**
The final step in this stage is to encode labels in each data frame of the 4 dims into [0,1] for binary classification and the main data frame into [0:15] for multi-class classification.

3.1.3 Data Preprocessing

CBOW:

Word embedding was executed using the Word2Vec method, a text representation technique that learns to convert words into numerical vectors with a length of n. By analyzing sentences, Word2Vec discerns patterns in word structures, providing a numerical representation for each word. This method, superior to the TF-IDF technique, excels in learning word relationships even for words not encountered during training.

Word2Vec comprises two models: Continuous Bag of Words (CBOW) and Skip-gram. In this context, CBOW was employed. CBOW predicts a word based on the context words within a phrase, effectively addressing the out-of-vocabulary problem in text corpora. This word-embedding approach enhances the model's understanding of semantic relationships between words, contributing to improved performance in subsequent tasks like sentiment analysis or information retrieval.

Architecture:

Input Layer:
Accepts word indices as input, representing the target word in a given context window.
Embedding Layer:
1- Transforms input word indices into dense vectors.
2- Learns and stores distributed representations (word embeddings) for each word in the vocabulary.
Hidden Layer (Average or Summation:)
1- Aggregates the embeddings of context words (surrounding the target word).
2- Utilizes averaging or summation to create a context vector.
Output Layer:
1- Predicts the target word based on the generated context vector.
2- Employs a softmax activation function for multiclass classification, assigning probabilities to each word in the vocabulary.

3.2 Modeling

3.2.1 Neural Network Approach

The model is a text classification architecture consisting of an Embedding layer to convert raw textual data into dense vectors, followed by an LSTM layer to capture sequential patterns. It takes raw sentences as input and predicts categories or labels associated with the text. The simplicity of the design prioritizes efficiency in capturing nuanced information for robust classification.

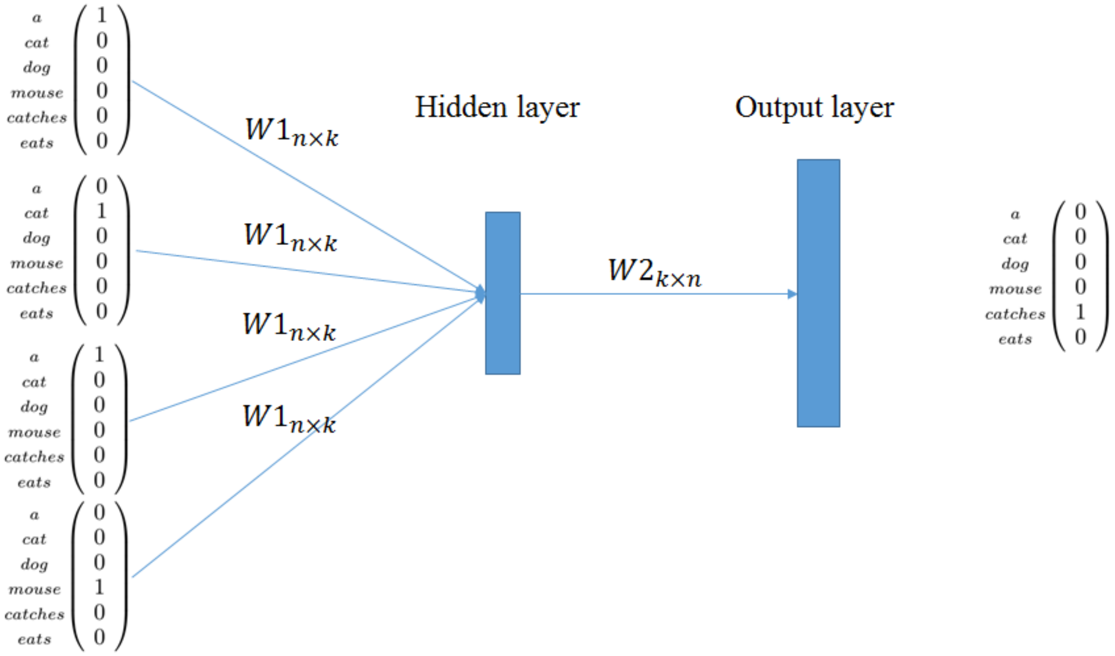


Figure 3.3: CBOW Model Architecture

Architecture:

1- Embedding Layer:

Description: The embedding layer, a cornerstone of natural language processing models, undertakes the task of translating word indices into dense vectors. This transformative process not only imparts a numerical representation to words but also captures semantic relationships within the vocabulary.
Parameters:
1- input_dim: Set at 2000, the input dimension signifies the size of the vocabulary, allowing the model to discern and learn from a diverse range of words.
2- output_dim: Set at 2000, the input dimension signifies the size of the vocabulary, allowing the model to discern and learn from a diverse range of words.
3- input_length: Dynamically adapting to the maximum sequence length in the training data through padding ensures that the model can effectively handle sequences of varying lengths.

2- LSTM Layer:

Description: The LSTM layer, an evolution of traditional recurrent neural networks, excels in capturing and remembering long-term dependencies within sequen-

tial data. In the context of text, this capability is invaluable for understanding contextual nuances and semantic relationships between words
Parameters: With a judicious selection of 64 units, we aim to endow the model with the capacity to discern intricate patterns and relationships in the textual data. This decision reflects a balance between model expressiveness and computational efficiency.

3- Dense Layer:

Description: The dense layer is a fully connected layer that produces the final output of the model. we use the softmax activation function, which is suitable for multi-class classification tasks.
Parameters: Set to 16, aligning precisely with the number of distinct MBTI classification categories. This alignment facilitates a seamless mapping of learned features to the specific personality dimensions.

Model Compilation:

1- Optimizer:

The Adam optimizer emerges as a strategic choice due to its adaptive learning rate mechanism. This adaptability, combined with a Root Mean Square Propagation (RMSprop) technique, empowers the model to navigate through intricate optimization landscapes. RMSprop is specifically designed to Reduce the effect of gradient issues by adjusting the learning rates individually for each parameter during training. This approach enhances both the efficiency and convergence speed of the optimization process.

2- Loss Function:

Sparse categorical cross-entropy assumes the role of the chosen loss function, aptly suited for our multi-class classification task. This choice is underpinned by its compatibility with integer-encoded class labels, simplifying the training process while ensuring robust model performance.

3- Metrics:

The accuracy metric takes center stage in our evaluation strategy, offering a clear and intuitive measure of the model's proficiency in making correct predictions across all personality classes. It serves as a reliable benchmark for gauging overall model performance.

3.2.2 preform Classical Predictions Approach

This approach involves training six different models: Logistic Regression, Support Vector Classification (SVC), Stochastic Gradient Descent (SGD), Random Forest classifier, Extreme Gradient Boosting (XGBoost), and CatBoost.

The training is conducted for two specific scenarios:

1- Training for 16 Types of MBTI Classes:
In this scenario, the models are trained to classify text data into one of the 16 Myers-Briggs Type Indicator (MBTI) personality types.

2- Training for Corresponding Dichotomies (IE, NS, TF, JP):
Here, the models are trained to classify text data based on the corresponding dichotomies within MBTI, namely Introversion-Extraversion (IE), Intuition-Sensing (NS), Thinking-Feeling (TF), and Judging-Perceiving (JP).

Architecture:

Data Preparation:
1- The input features (X) are derived from the embeddings generated using the Word2Vec model for the text data.
2- The target variable (y) is the MBTI personality type or the corresponding dichotomy labels.
Data Splitting:
The dataset is split into training and testing sets using the train_test_split function with a test size of 20% for evaluation.
Model Training:

1- Logistic Regression:
Description: Logistic Regression is a linear model used for binary classification tasks. It models the probability that a given instance belongs to a particular class.
Parameters: The model employs a logistic function to squash the output into a range [0, 1], making it suitable for binary classification.
2- SVC (Linear Support Vector Classification):
Description: SVC is a linear classification model that constructs a hyperplane to separate different classes in the feature space.
Parameters: It aims to find the hyperplane that maximizes the margin between classes, making it effective for linearly separable data.
3- SGD (Stochastic Gradient Descent):
Description: SGD is an optimization algorithm used for training linear classi-

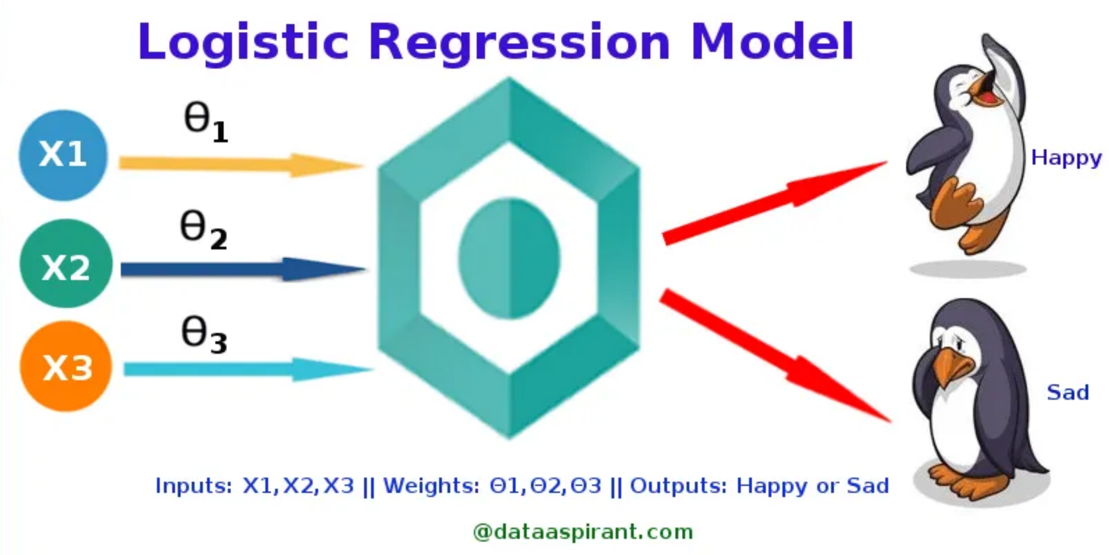

Logistic Regression Model
X1
X2
X3
θ1
θ2
θ3
Happy
Sad
Inputs: X1,X2,X3 || Weights: ϴ1,ϴ2,ϴ3 || Outputs: Happy or Sad
@dataaspirant.com

Figure 3.4: Logistic Regression model Architecture

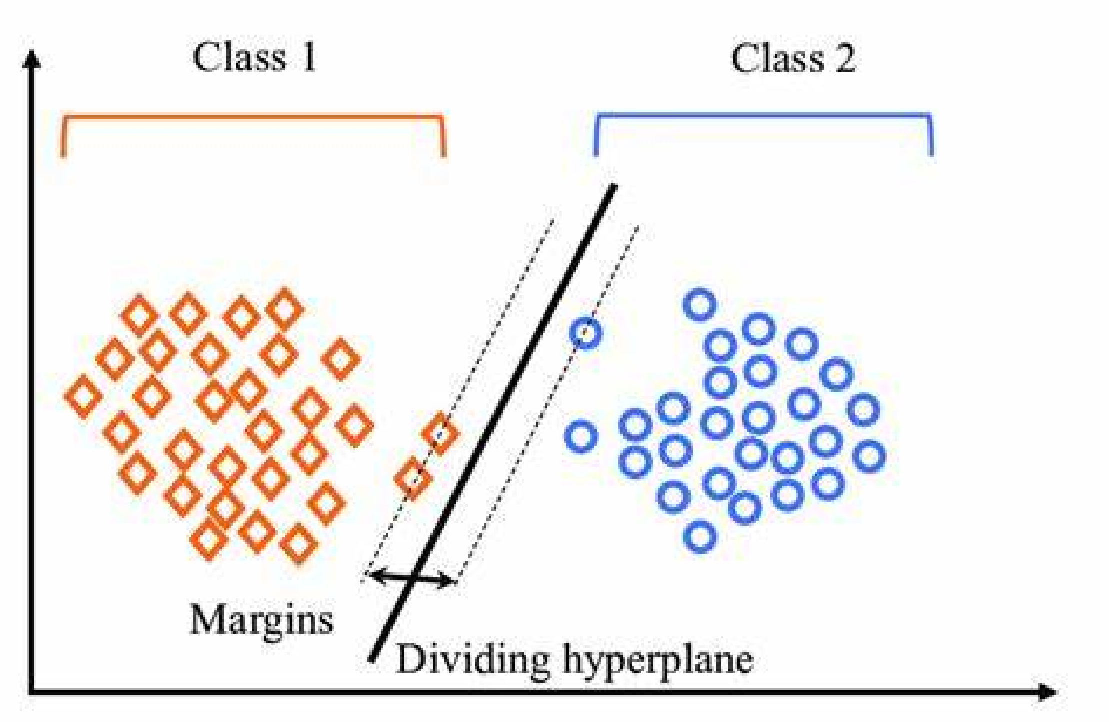


Figure 3.5: Linear Support Vector model

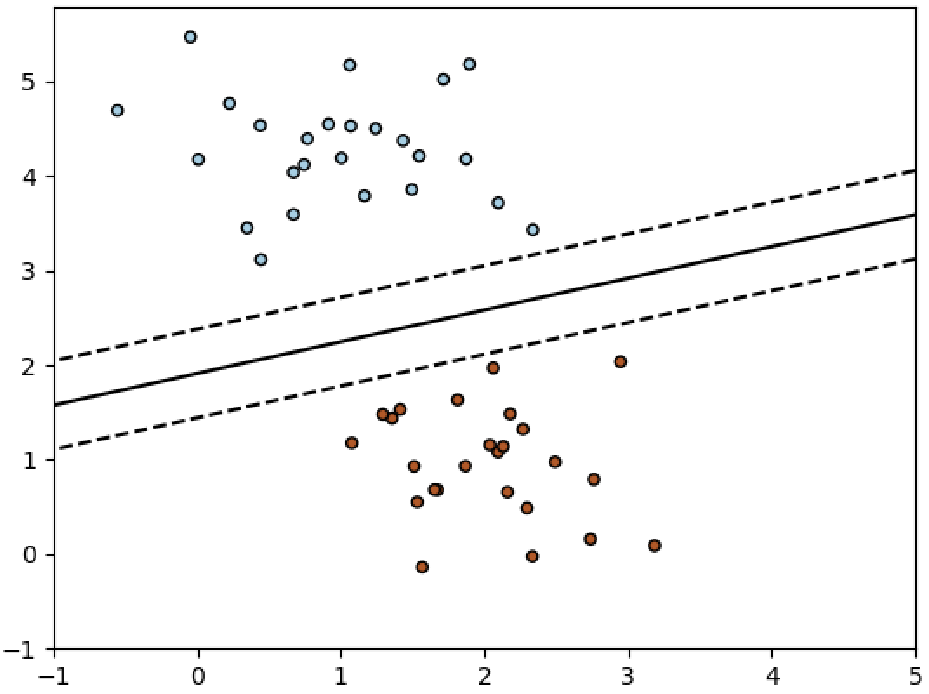


Figure 3.6: Stochastic Gradient Descent model Architecture

fiers, including Support Vector Machines (SVMs) and Logistic Regression.
Parameters: It updates the model parameters with each training example, making it suitable for large datasets and online learning scenarios.
4- Random Forest:
Description: Random Forest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the classes.
Parameters: It mitigates overfitting by aggregating the predictions of multiple decision trees, providing robustness and improved accuracy.
5- XGBoost (Extreme Gradient Boosting):
Description: XGBoost is a gradient-boosting algorithm known for its efficiency and performance. It builds a series of decision trees, each correcting the errors of the previous one.
Parameters: XGBoost employs a regularization term in its objective function, handling missing data, and utilizing parallel processing for faster training.

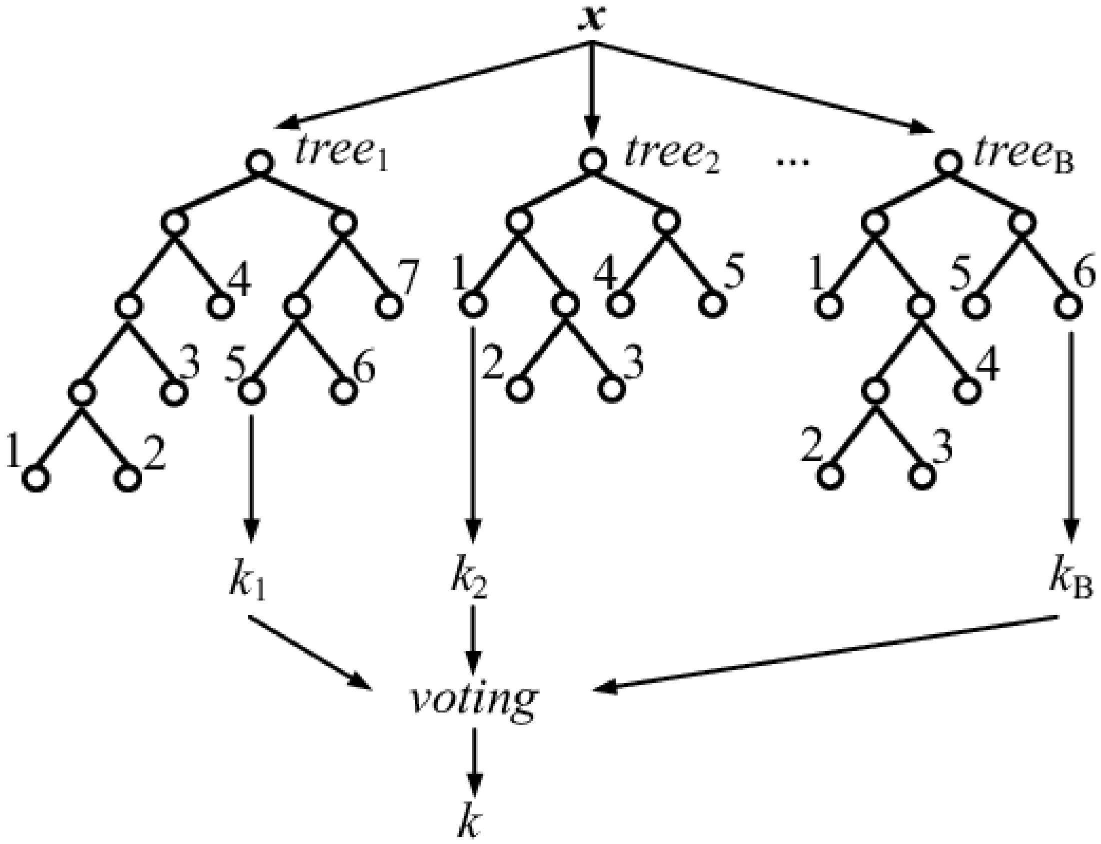


Figure 3.7: random forest model

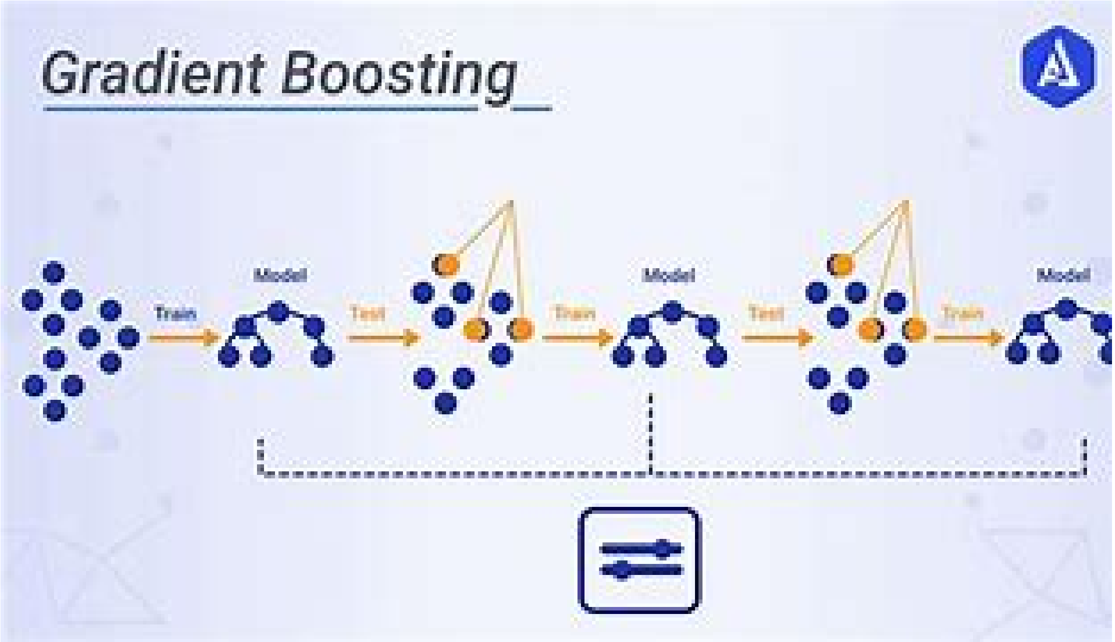


Figure 3.8: Extreme Gradient Boosting model Architecture

6- CatBoost:
Description: CatBoost is a gradient-boosting algorithm designed to handle categorical features without the need for preprocessing. It supports both numerical and categorical data.
Parameters: CatBoost employs a modified version of the learning rate scheduling technique, providing efficient training and accurate predictions for classification tasks.

Evaluation:

The accuracy of each model is evaluated using the accuracy_score metric, measuring the percentage of correctly predicted instances.

4 Results

4.1 Neural Network Approach

We trained the LSTM with 46 epochs and a batch size of 32 and managed to get an **Accuracy: 73.02%, loss: 1.1323** on evaluation testing. Values on training On the training can be seen in the below figure.

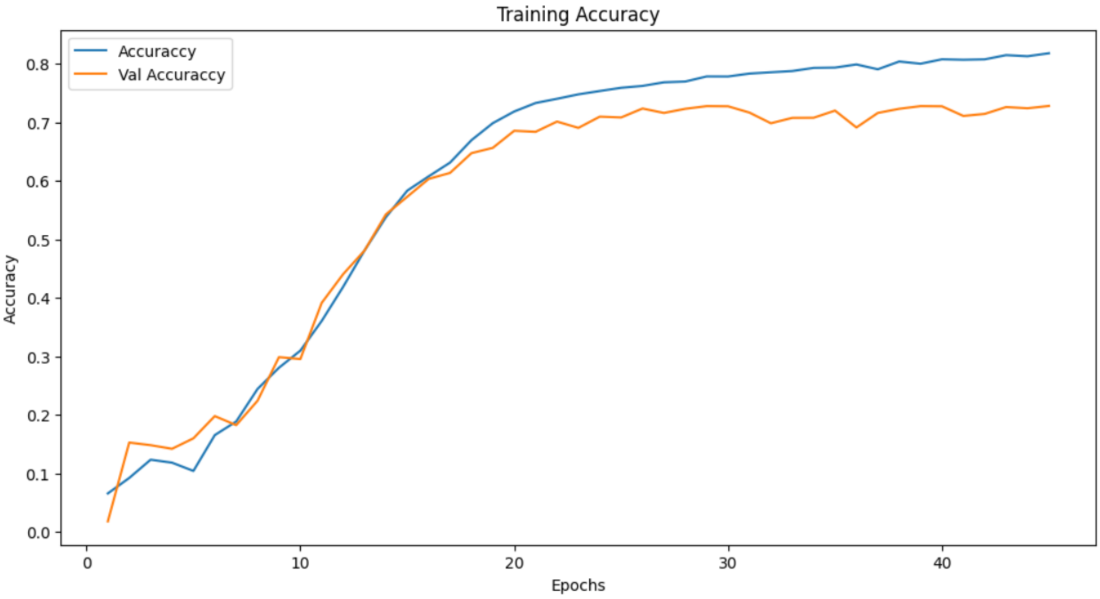


Figure 4.1: Accuracy curve during model training

The precision scores test is a bit inaccurate due to the imbalance of the test data counts.
From the previous, we can find that the perfect model before over-fitting is around 20 epochs.

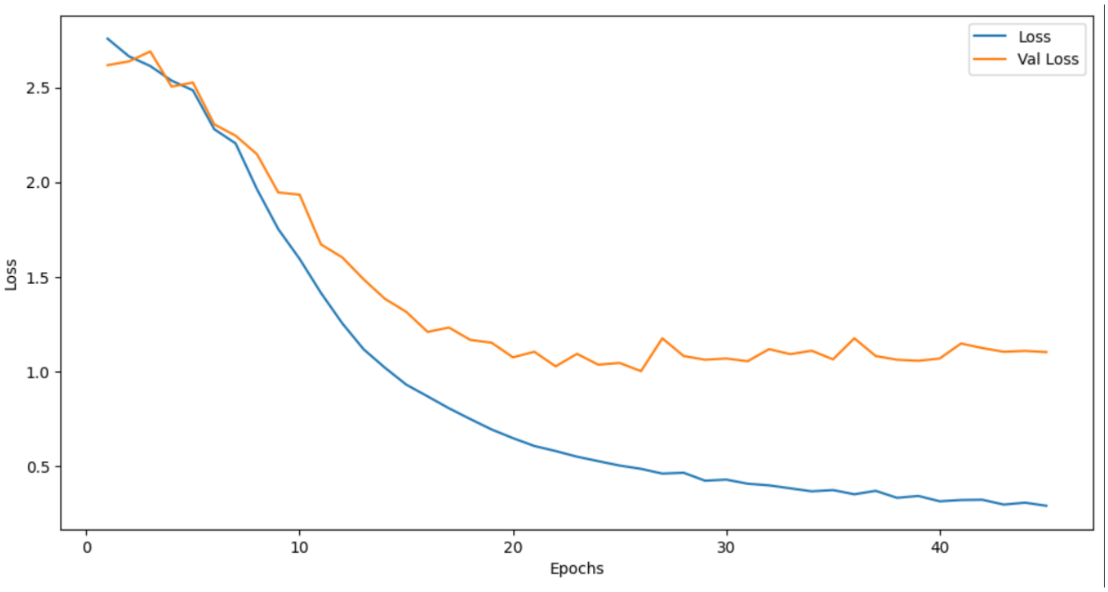


Figure 4.2: Loss curve during model training

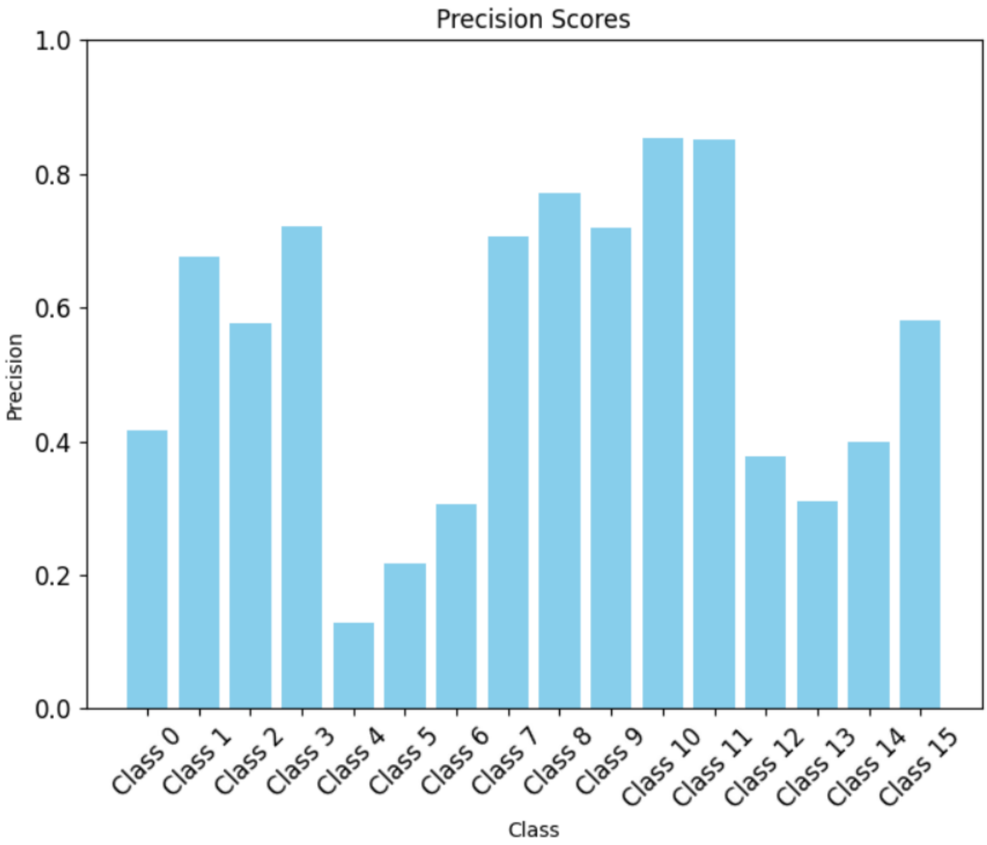
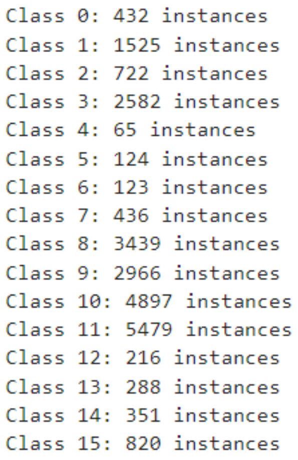


Figure 4.3: Precision of each class



```
Class 0: 432 instances
Class 1: 1525 instances
Class 2: 722 instances
Class 3: 2582 instances
Class 4: 65 instances
Class 5: 124 instances
Class 6: 123 instances
Class 7: 436 instances
Class 8: 3439 instances
Class 9: 2966 instances
Class 10: 4897 instances
Class 11: 5479 instances
Class 12: 216 instances
Class 13: 288 instances
Class 14: 351 instances
Class 15: 820 instances
```

Figure 4.4: Count of each class in the precision test

4.2 Classical classification on NLP model

4.2.1 CBOW Victorization

Model	I/E	N/S	T/F	J/P	AVG
Logistic Regression	0.72	0.757	0.8315	0.6724	0.7452
SVC Accuracy	0.7342	0.7834	0.8424	0.6858	0.7614
SGD Classifier	0.691	0.764	0.8199	0.6538	0.7321
Random Forest	0.6819	0.7347	0.8007	0.6512	0.7171
XGBoost	0.7104	0.7672	0.8192	0.6664	0.7408
CatBoost	0.7297	0.786	0.8370	0.6852	0.7594

The previous shows the resulting accuracy of all 6 models on the CBOW Victorized Text on each dimension of the 4

4.2.2 BERT Victorization

Model	I/E	N/S	T/F	J/P	AVG
Logistic Regression	0.8006	0.7896	0.7311	0.7311	0.7878
SVC Accuracy	0.7375	0.7682	0.8108	0.7032	0.7549
SGD Classifier	0.7801	0.7878	0.8252	0.7196	0.7781
Random Forest	0.7496	0.7124	0.769	0.6856	0.7291
XGBoost	0.7767	0.7434	0.7933	0.6995	0.7532
CatBoost	0.7838	0.7635	0.8064	0.7128	0.7666

The previous shows the resulting accuracy of all 6 models on the BERT Victorized Text on each dimension of the 4

Summary

In the Second Experiment, We found That the BERT Model generated a vector that we were able to classify a bit better than the CBOW Model. although the BERT is much slower than the CBOW.