Personality Detection using Deep Learning Techniques

By

Sarah Ali

00000319217

Supervisor

Assoc Prof Dr. Hammad Afzal

A thesis submitted in the Department of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan for the partial fulfillment of the requirement for degree of MS in Software Engineering.

September 2022

Personality Detection using Deep Learning Techniques

By

Sarah Ali

00000319217

Supervisor

Assoc Prof Dr. Hammad Afzal

A thesis submitted in conformity with the requirements for the degree of *Master of Science* in Software Engineering

Department of Computer Software Engineering

Military College of Signals (MCS)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

September 2022

Declaration

I, *Sarah Ali* declare that this thesis titled ”Personality Detection using Deep Learning Techniques” and the work presented in it are my own and has been generated by me as a result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a Master of Science degree at NUST
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at NUST or any other institution, this has been clearly stated
3. Where I have consulted the published work of others, this is always clearly attributed
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work
5. I have acknowledged all main sources of help
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself

Sarah Ali,
00000319217

This thesis is dedicated to *myself*

Acknowledgment

First, I thank Almighty Allah for bestowing me the strength and the passion to complete this research work. Secondly, I would like to acknowledge the tireless and prompt help of my supervisor, Dr. Hammad Afzal. He supported me throughout the process from defining and exploring idea until its execution. His valuable help in form of critical reviews and suggestions, throughout the experimental process and thesis work, have been the major contributing factor to the success of this study. I would also like to thank my dear colleague, Ms Rida Ayesha, for her assistance and guidance in completion of this project. Finally, my parents', my husband and my siblings for their constant support that made me think positively.

Abstract

Personality refers to the distinguishing set of qualities of an individual that impacts their attitude, habits, behaviors and pattern of thoughts. Personality traits have been shown to have governing effect on major outlook of life such as success in the political temperament, general and workplace emotional stability. Textual data accessible on Social Networking sites yields an opportunity to automatically identify personality traits of an individual. Since technology has progressed expeditiously, personality detection has become a popular research field that bestows personalization to users. Presently, researchers have employed data on social media for automatic prediction of personality. However, the extraction of the social media data is a complex process as it is noisy, available in different formats and lengths. This research proposes a machine learning model and a deep learning model to predict the personality of an individual based on Myers–Briggs Type Indicator (MBTI) personality model. The proposed machine learning models (SVM, LR, MLP and XGBoost) were trained on MBTI and MBTI500 datasets with imbalanced and balanced instances (using SMOTE). The proposed deep learning model was trained using CNN with GloVe word embeddings. SVM model achieved the highest accuracy of 96.81% for machine learning model on MBTI500 dataset with SMOTE. However, CNN exhibited the highest accuracy of 99.54% on MBTI dataset which supersedes the existing models.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviations

MBTI	Myers-Briggs Type Indicator Personality Model
ML	Machine Learning
NLP	Natural Language Processing
SVM	Support vector machine
LR	Logistic Regression
MLP	Multi-layer Perceptron
XGBoost	Extreme Gradient Boosting
TF-IDF	Term Frequency - Inverse Document Frequency
DL	Deep Learning
CNN	Convolutional Neural Network
SMOTE	Synthetic Minority Oversampling TEchnique
GloVe	Global Vectors for word representation

Chapter 1

Introduction

1.1 Overview

Data Science and Artificial Intelligence are reforming the world through specialized changes. We can notice many AI applications in everyday lives, except one of the best utilization of AI is to arrange people in view of their character attributes. Every individual on this planet is remarkable and conveys a novel character. The accessibility of a high-layered and enormous measure of information has prepared for expanding marketing efforts' viability by focusing on specific individuals. All through the personality research history, endeavors for personality detection went from conventional psychological strategies (for example questionnaires), by means of psycho-phonetic methodologies (for example counting explicit word types in texts), to the new absolutely natural language processing (NLP) moves toward that effort to identify personality characteristics from a lot of virtual social networking data. Particularly with latest technological progress and big data, the exploration region has reached out with the reason that advanced impressions could catch roundabout and normal attributes as well as psychological insights on more profound levels. Such character based correspondences are profoundly successful in expanding the fame and appeal of items and administrations. It expanded use, consumer loyalty, and more extensive acknowledgment among clients. Character qualities are firmly connected with a person's way of behaving and inclinations. Subsequently the combination of a character based approach has essentially expanded the Recommender System's engaging quality and personalized visualization. Countries like Japan have also made AI based dating apps, to control the increasing divorce rate, which detects

the personality based on user's behavior across all social media and then suggest the perfect match.
Automatic prediction of personality requires a comprehensive perception of the construct, due to its complex and miscellaneous structure, which is a challenging task even taking into account the technological advancements of this era. This research mainly focuses on developing a ML and a DL model to detect the personality type of an individual automatically using textual data obtained from social networking sites.

1.2 Problem Statement

Personality detection is a tool used to automatically assess personality of human beings. Personality detection and testing through text allude to techniques intended to gauge the trademark examples of attributes that individuals display across different circumstances using their textual data. Algorithms that detect personality usually extract features from the textual data and use these features to train the model. In recent years, personality based approach are being used by companies in advertisements, music and movie recommendation systems. Companies are also using personality detection in hiring procedures to designate the best suited job for each individual.
However, with all the potential benefits of personality detection, researchers are still looking for methods and algorithms to increase the accuracy of detection of all traits of personality. Therefore, this study proposes a new machine learning model using synthetic minority oversampling technique (SMOTE) with different machine learning algorithms and a deep learning model using global vectors for word representation (GLOVE) embedding and a Convolution Neural Network for automatic personality detection.

1.3 Aims and Objectives

The research objective includes

- To explore better textual features which will improve the model for detection of personality.

- To explore different deep learning and machine learning methods for detection of personality.
- To improve the accuracy of personality detection algorithm as compared to state-of-the-art techniques.

1.4 Research Contribution

The main contributions of this research are:

- It automates the process of personality prediction using different ML and DL techniques.
- It enhances the performance of ML models by applying the re-sampling technique (SMOTE) to the imbalanced datasets of personality.
- It produces the ML and DL models with highest results in terms of evaluation metrics such as, accuracy and F1-score as compared to the baseline models [1].

1.5 Organization

This dissertation presents a developed model for automated personality detection using textual data. The organization of thesis is as follows:

- **Chapter 1 - Introduction:**
  This chapter provides the overview of the research topic along with it's main aims and objectives. The main research contributions are also discussed in this chapter.
- **Chapter 2 - Background**
  This chapter introduces the background of the problem, explains Myers-Briggs Type Indicator personality model developed by psychologists to determine the personality of an individual.
- **Chapter 3 - Literature Review:**
  This chapter provides the literature review and the related work of recent years. It also

covers a comprehensive review of different ML and DL techniques used for personality detection.

- **Chapter 4 - Proposed System:**
  This chapter discusses the experimental setup for four ML models which includes Logistic Regression (LR), SVM, MLP and XGBoost, and one DL model which Convolutional Neural Network (CNN) with GloVe embeddings.

- **Chapter 5 - Evaluating Trained Model for Automatic Personality Detection:**
  In this chapter, the evaluation of the model by using the metrics is focused. The results of ML and DL models are compared in terms of their accuracy and F1-score. It also provides the analysis on the basis of the computed results.

- **Chapter 6 - Conclusion:**
  This chapter concludes the entire research by discussing the outcome of the research along with few limitations, challenges and future research directions.

Chapter 2

Background

2.1 Introduction

Machine learning is the field of artificial intelligence (AI). ML is a branch of computer science, and it is distinctive from conventional computer strategies. For this reason, machine learning makes a difference computers build models from tests of information to create choices based on information input. For this reason, there is another thing to keep in mind. Working with machine learning methods or examining the effect on the machine learning process in contemporary years, an extensive amplify in developments has been stated with uses of machine learning for clinical purposes in all three general medical professions diagnostic, therapeutic and treatment. Personality detection lies under the umbrella of therapeutic medical science. Psychologists have designed various personality tests for detection of personality type. Conducting these tests and then manually determining the personality test requires a lot of time and personnel. On the other hand, machine learning capabilities can easily procure parts of parameters from textual data of an individual like tweets, essays or status updates, that can identify and utilize the relationships between these many attributes, making personality detection an easier and cheaper task.

2.2 Personality Models

Over the years psychologists have developed a number of models to determine the personality of an individual, but one of the most widely used models is MBTI Personality Model.

This section explains MBTI model in detail.

2.2.1 MBTI Model

Myers-Briggs Type Indicator is a reflective, self assessment that recognizes an individual's character type and mental inclinations. The reason for this assessment is to allot people into one of four classes in view of how they see the world and decide, empowering respondents to additionally investigate and grasp their own characters. No personality type is better than the other. The four classes are as follows:

i. **Sensing (S) – Intuition (N):** This scale refers to how people process information. The 'sensing' ones are the people who enjoys the present and like to make decisions based on facts and figures. Mostly they are motivated to do work because of the paycheck and then spending it on things they enjoy.
   However, people who are intuitive need to have a deeper meaning for their work so they can be motivated enough to do the work even if they don't really enjoy doing it. They become less productive if they are working only for the paycheck. They like to think of future possibilities and are highly imaginative people.

ii. **Extraversion (E) – Introversion (I):** An individual can either be an extrovert or introvert.Extroverts usually gain energy from social interactions. Extroverts can easily start conversations and are happy being the center of attention. They can easily turn their acquaintances into friends and therefore have an extensive social circle. They are outgoing people who seek excitement almost in all situations and become sad or depressed if they are alone.
   Whereas introverts tend to become exhausted in situations requiring interaction with people. They enjoy being alone and quiet time. They become uncomfortable if they become center of attraction. They are also unable to start conversations and make small talk with new acquaintances. They are only comfortable in social interactions with their close circle. After a specific time in crowded or social situations, like parties, they need some alone time to recharge their social battery.

iii. **Judging (J) – Perceiving (P):** People with judging personalities approach life in an organized and coordinated way, making short-and long plans which can assist them with accomplishing their objectives. Plans and schedules are important to them. They lean toward understanding what they are getting into and feel baffled in circumstances of vagueness and change.

On the contrary, perceptive individuals approach life in a freewheeling and unconstrained way, liking to keep their choices open instead of making a set schedule for any activity. They see structure as restricting and look for adaptability in their lives. They like adjusting to new circumstances and feel baffled by the monotonous routine of schedules.

iv. **Thinking (T) – Feeling (F):** This scale describes how people come to specific decisions in their life. Thinkers mostly use factual data and logical theories to come to a definite judgement. They like to make strategical decisions moving toward a better future. Thinkers tend to have a black and white logical criteria for every judgment or decision they make.

Feelers, on the other hand, tend to take into account feelings, emotions and people around them when coming to an important decision. Feelers are artistic people who also serve as 'social glue' to keep the people more attuned to each other's feelings and emotions.

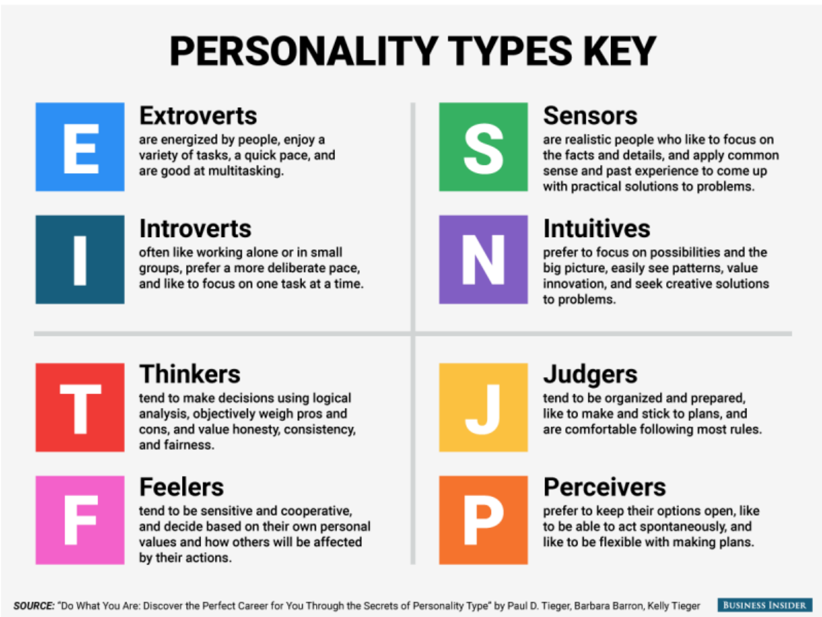


Figure 2.1: MBTI Personality Model

Every individual is said to have one favored quality from every class, creating sixteen one of a kind personality types. Each type is listed by its four-letter code:

i. **INFJ - The Advocate:** Ingenious and scientific, they are viewed as one of the most uncommon Myers-Briggs types.

ii. **INTJ - The Architect:** Highly rational, they are both exceptionally imaginative and analytical.

iii. **INFP - The Mediator:** Visionary with high qualities, they endeavor to make the world a greater place.

iv. **INTP - The Thinker:** Silent and thoughtful, they are known for having a rich inward world.

v. **ISTP - The Crafter:** Highly autonomous, they appreciate new encounters that give direct learning.

vi. **ISTJ - The Inspector:** Reserved and viable, they will quite often be steadfast, efficient, and customary.

vii. **ISFP - The Artist:** Easy-going and and adaptable, they will more often than not, be private and imaginative.

viii. **ISFJ - The Protector:** Warm-hearted and devoted, they are generally prepared to safeguard individuals they care about.

ix. **ESTJ - The Director:** Self-confident and rule-situated, they have high standards and a propensity to assume responsibility.

x. **ESTP - The Persuader:** Unreserved and emotional, they appreciate investing energy with others and zeroing in on the present time and place.

xi. **ESFJ - The Caregiver:** Tender-hearted and cordial, they will more often than not, give others the benefit of the doubt.

xii. **ESFP - The Performer:** Uninhibited and unconstrained, they appreciate becoming the overwhelming focus.

xiii. **ENFJ - The Giver:** Faithful and compassionate, they are known for being understanding and liberal.

xiv. **ENFP - The Champion:** Charming and vigorous, they appreciate circumstances where they can give their imagination something to do.

xv. **ENTJ - The Commander:** Forthright and certain, they are perfect at making arrangements and putting together ventures.

xvi. **ENTP - The Debater:** Highly innovative, they love being encircled by thoughts and will generally begin many activities (however may battle to complete them).

2.3 Summary

The MBTI personality model makes a good interpretation of behavioral conduct. For instance, Extroverts favor offline communication modes, because of socialization and actual closeness, while Introverts favor online communication, because of the obscurity of online mode [2]. With regards to professional bearing, the two MBTI classes, Sensing/Intuition and Thinking/Feeling, are the most powerful character perspectives as the two of them are exceptionally connected with inclinations for information handling. Work fulfillment and employee turnover can also be affected by MBTI character types. Different MBTI types also have a difference in taste of music, movies, books and other daily life needs. Hence automatic personality detection is an important area of research in today's era. In this research, a machine learning and a deep learning model has been developed for automatic detection of personality through text. The next chapter includes the literature review for the related work.

Chapter 3

Literature Review

3.1 Introduction

Few of the relevant papers were read and summarized in the literature review. Summary details of the papers is discussed the section 3.2

3.2 Related Work

The related work is summarized and discussed in detail below:
In [3], they applied traditional and deep learning models like SVM, Naive Bayes and Recurrent neural networks (RNN) to MBTI dataset. They preprocessed the data using selective word and character removal using python's NLTK. After that lemmatization was performed on the dataset. As for tokenization they used two different techniques. For Naive Bayes and SVM, they used NLTK tokenizer and applied TF-IDF and Bag of Words to convert words to vectors. On the other hand, they used Keras word tokenizer for RNN. They divided the dataset into 75% training and 25% testing data using Scikit-Learn Python Package. Naive Bayes and SVM were then fitted on the dataset to gain the average accuracy of 80.16% and 80.66% respectively, across all four classes. RNN was also trained on the dataset using CONV1D and Bi-LSTM was added to store information in both directions. RNN gave the best accuracy of 83.5%.
In [1], they prepared the MBTI dataset by removing all URLs, punctuation marks, symbols, numbers, emoticons, target names (@someone), non-English character and hashtags. Then

the dataset was preprocessed in three steps. Firstly, a word tokenizer was applied to the dataset. After which the stop words were removed and Stemming was applied. Then, the distinct features were extracted from the dataset using CountVectorizer. The dataset was divided into two parts: 75% training and 25% testing data and was fitted into three different classifiers separately. They used Naive Bayes using default parameters and got an average accuracy of 78.5%. The parameters of SVM classifier were optimized and gave an average accuracy of 79.5%. The parameters of XGBoost classifier were also optimized and gave the highest average accuracy of 85%.

In [4], they used data level re-sampling on MBTI dataset to overcome the class imbalance problem. In their proposed system, data was divided into training, validation and testing data. They also used k-fold cross-validation technique. For text preprocessing, the tokenization of data was performed, stop words were removed and word stemming was applied a a text normalization technique. For feature extraction, CountVectorizer was used to make word vectors and TF-IDF was used to calculate the importance of a word. The ML model is then trained using both labelled and textual data. They trained XGBoost classifier using optimized parameters and got the highest average accuracy of 97.34% across all four classes. They also compared different ML algorithms (like KNN, decision trees, random forest, MLP, SVM etc.) using MBTI dataset before and after re-sampling and proved that class balancing techniques improve accuracy by a large margin.

In [5], they also removed stop words using NLTK package and then filtered out all URLs from the MBTI dataset. Then, they lemmatized all word to their root forms and then extract features using CountVectorizer which extracts the TF-IDF matrix. They also applied Glove embedding on the cleaned text and made a portrayal of every tweet by consecutively adding the word embeddings from Glove weighted using its associate TF-IDF value for every word in the tweet. They used SMOTE for re-sampling of dataset to address the issue of class imbalance. After this, they applied SVM and XGBoost classifiers separately on the dataset and the result of these two classifiers are combined centered on the prediction probability output.

In [6], they also used MBTI dataset. They used word2Vec technique for vectorization of data. The dataset was divided into training (90%) and testing (10%) data and random forest, Logistic Regression, KNN Neighbor and Support Vector Machine(SVM) models were ap-

plied to the data using Numpy and Sci-kit learn. They also applied these machine learning algorithms to the four classes of MBTI model separately to get better results.
In [7], two datasets were used: The essays dataset [8] and MBTI dataset. They used three separate components and created models from their different combinations. A 'contour encoder' that is used to get converts a a hidden representation vector from a sequence of psycho linguistic features, is the first component. Second is BERT, a pre-trained transformer-based language model, that gets hidden representation vector from a sequence of tokens. Third is a classifier that uses the hidden representation of the sample and returns the probability of a personality feature. They created three models using these components. First included contour encoder and the classifier. They used BiLSTM with and without attention models in contour encoder. They also experimented with feature based frozen model weights and fine tuning the unfrozen layers in the hybrid models.They created three types of models for detection of personality type.

i. Contour encoder with classifier

ii. Hybrid models that join the contour encoder and language model based on transformers, further with a classifier

iii. A stacking model that joins ten reiterations of the model with best results

On both datasets, the ensemble model achieved the highest accuracy, in which ten repetitions of a hybrid model combining a fine-tuned BERT model with an attention-based BiLSTM model also used text contours for training. The average accuracy on Big five dataset and MBTI dataset was calculated to be 63.5% and 86.51% respectively.
In [9], they also used two datasets: The essays dataset [8] and MBTI dataset. They extracted two types of features from the textual data.

i. Psycho-linguistic features which were Mairesse features, SenticNet features, NRC Emotion Lexicon, VAD Lexicon and Readability.

ii. Language Model Features using BERT [10] , Alberta [11] and Roberta [12] which were giving similar results.

They used a multi-layer perceptron (MLP) and 'relu' non-linearity, SVM and Logistic Regression for fine-tuning. 10 fold cross-validation was used. They also used Adam optimizer [13] and for loss function they used binary cross entropy.The highest average accuracy achieved for essay dataset and MBTI dataset was 60.6% using BERT-base + MLP and 77.1% using BERT-large + MLP respectively.

Mohammad Hossein et al. [14], used MBTI dataset from Kaggle. They first preprocessed the dataset to remove URLs and stop words from text data. The textual data was then vectorized using TF–IDF with count vectorizer. The dataset was divided into training (70%) and testing (30%). They created a Gradient Boosting algorithm using sklearn, XGBoost and Numpy. They got the best average accuracy of 74.43%.

Z. Ren et al. [15] proposed a model which consisted of following three parts on MBTI dataset and Big Five dataset.

i. Sentence Embedding using BERT model.

ii. Sentiment analysis using SenticNet5 dictionary.

iii. Neural network classification using Recurrent Neural Network (RNN) and CNN.

They removed URLs, stop words and some special characters. The dataset was also balanced using under-sampling. They then created models with variations of above mentioned components for both datasets. The best average accuracy for Big Five dataset was achieved using Bert with CNN from single-label technique. The best average accuracy for MBTI dataset was achieved using Bert with SenticNet5 and CNN from multi-label technique.

Hussain Ahmad et al. [16], proposed a hybrid deep learning model for personality detection. For preprocessing step they applied lower casing of textual data of MBTI dataset, then they deleted stop words and further tokenized the data using Keras tokenizer. Their embedding layer represented the words in numeric form, then they used CNN to extract features from the numeric data. LSTM model was used to learn information that is long-term and finally Softmax layer was applied to classify the data into different personality traits. Their proposed technique gave the highest average accuracy of 87.5%.

In [17], M. Kuchhal et al. applied various Machine learning algorithms on MBTI dataset. They vectorized the textual data using TF-IDF and then used truncated SVD for dimentionality reduction as the feature vector created from vectorization was of very large dimensions. 70% data was used for training and 30% data was used for validation. They used SMOTE for re-sampling the dataset to solve the class imbalance problem. Random Forest Classifier, one vs rest classifier, k - nearest neighbours, extreme gradient boosting and multi-layered perceptron model were trained on the over-sampled dataset. Extreme gradient boosting was the algorithm which gave the highest accuracy score of 90.88%.

Hans Christian et al. in [18] developed a model using pretrained transformers. They used myPersonality dataset of Facebook statuses and manually collected a twitter dataset. Each dataset was divided into training (70%), validation (15%) and testing data (15%). They first removed all URLs and symbols from the dataset and then expanded the contractions in all sentences. Then lower casing was performed on the dataset, stop words were removed and stemming was incorporated. They extracted two types of features from the cleaned text data.

i. Pre-trained model features using BERT, RoBERTa, and XLNet.

ii. Statistical features using TF-IGM (term frequency and inverse gravity moment).

These features were then fed into a self-attention mechanism which creates association of words with each other. The features from each pre-trained embedding are used as inputs in three feed-forward neural networks and the Big five personality is predicted using unweighted model averaging.

M.M. Tadesse et al. [19], used myPersonality dataset to predict the Big Five Personality traits of an individual. Firstly, they applied tokenization to the text data. Then they removed removed URLs, spaces, names, lower cases and symbols. They didn't applied stemming as they used SPLICE (Structured Programming for Linguistic Cue Extraction), LIWC (Linguistic Inquiry and Word Count) and SNA (Social Network Analysis) for feature extraction. PCA (Pearson correlation analysis) was used to measure the importance of features for personality classification and their relationship between each other. They used XGBoost as their primary classification method. Support Vector Machine (SVM), Logistic Regression and Gradient Boosting were used as baseline method for comparison of results. XGBoost performed better than the baseline models with all feature sets for all other traits of Big Five personality traits other than extraversion in which XGBoost performed better with only SNA feature set.

F.M. Deilami et al. [20] used the James and Pennebaker Essays dataset [8] for the prediction of Big Five personality traits. In their proposed model, a coalescence of CNN (Convolutional Neural Network) and Ada-boost algorithm [21] was used. To analyze the text data and take out the important low-level features from it, various filters of different sizes were incorporated, such that, each CNN is created separately, having it's own layers (convolutional, pooling, and classification). Therefore, each CNN executes separate classification task. The results from this classification are then used as inputs to an Ada-Boost aggregation algorithm which is used to detect the personality of an individual formulated on the more accurate classification using the different weights of these various classifiers. They experimented with different variations of this setup. The result was obtained from CNN with Ada-Boost and 2 channels which were trained and tested using 5 fold cross-validation.

N. Taghvaei et al. [22], developed a hybrid model using three FNN (Fuzzy Neural Networks) and DNN (Deep Neural Networks). They normalized the textual data by removing any special characters and stop words. Then the feature extraction was used to take out a distinct set of features for each of these classifiers (FNN and DNN) on which they are separately trained. The features that were extracted for fuzzy neural networks were structural features using Social Networks Analysis (SNA). For deep neural networks to be trained, feature of status description were extracted using Linguistic Analysis (LA). Decision fusion strategy is used two times in their given model. The results obtained from three fuzzy neural networks are combined using the fusion algorithm which incorporates majority based decision. The result of the deep neural networks are then combined with this majority based decision in the second decision fusion to finally detect the personality on the basis of Big Five Personality model. They argue that as each classifier views the essays dataset from a different perspective as each of them is getting a different set of features extracted from the text. They experimented with MLP, CNN, LSTM and CNN + LSTM in their suggested framework, separately. MLP gave the best average accuracy of 78.62%.
K. El-Demerdash et al. in [23] proposed a model which had fusion techniques on both classifier level and data level. For data fusion, they used low level fusion in which they combined two datasets using Big Five Personality Model namely Essays dataset and myPersonality dataset. After data fusion, they finetuned three Learning models, Embeddings from Language Model (ELMo) [24], Universal Language Model Fine-Tuning Method (ULMFiT) [25] and Bidirectional Encoder Representations Transformer (BERT) [10] for this combined dataset, separately. Thus, each classifier predicts the personality of an individual separately and then their results are fused together to increase the average accuracy. Their proposed

model gave the higher results for both when trained on fused dataset and tested on essays dataset (61.85%) and when tested on myPersonality dataset (73.91%), than the classifiers when trained separately.

A. Kazameini et al. [26], divided each of the essays in the Essays dataset into sub-documents in order to take out as much information from the textual data as possible. Each sub-document had 200 tokens. They split each sentence at any full stop and question mark and only kept ASCII characters, exclamation marks, quotations and digits. They then expanded the short forms of the words (I've to I have) which increased the maximum size of the sub-document almost up to 250. These tokens serve as input for the pre-trained BERT base model. The contextual representation of tokens of each layer of BERT is averaged and then Mairesse features of essays dataset is joined with the last four layers of BERT. This document vector is the input to 10 SVM models which run in parallel similar to the bagging model which gives 10 results of classification of personality. The final personality type is selected through majority voting. Their proposed model gave the average accuracy of 59.03% on Essays dataset.

Veronica E. Lynn et al. [27], proved that message level attention is better than word level attention using filtered myPersonality dataset and Facebook statuses. Each word in the document was fed to a Gated Recurrent Unit (GRU) in their model, which gave a word level representation and then these word level representations serves as input for another set of GRU which gives a message level representation. These message level representations are then fed to another hidden unit which also incorporates GRU which creates user level representation where personality related sentences are used. After another hidden unit the final prediction of personality is given. Their model gave better results than only using word level attention mechanism but in three out of five cases of Big Five personality model, pre-trained BERT gave better results.

Bruno Fernandes et al. in [28], collected a new dataset in which the personality of an individual is determined by the adjectives selected by that individual to describe themselves. The personality was determined using Big Five personality model. In preprocessing step, they first handled the zero rating problem by converting to the nearest value. As the attributes selected by a person were in text format, they used Multi-Label Binarizer for one-hot encoding. Association rule learning algorithm and Apriori algorithm were used to determine the relation between the selected adjectives by a person to describe themselves. They created two architectures of machine learning. The first one takes preprocessed data as input and incorporates five regressors for each trait to predict personality. The second one takes preprocessed data as input and incorporates five classifiers for each trait to predict personality. The regressor architecture used Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to evaluate the results while F1-score was used to evaluate the classifier architecture. Their both architectures improved the results.

M. Jayaratne et al. [29], used the dataset of PredictiveHire FirstInterview (TM) online product. They used HEXACO personality model for personality detection, which is an extension to the Big Five Personality Model. They constructed a regression model that had the option to surmise a rating for every one of the six character qualities in the HEXACO model utilizing literary responses given to open-ended interview questions. Given the significance of mathematical portrayal of language in building an AI model, they looked at the exhibition of five unique message portrayal techniques specifically, terms (TF-IDF), topics (Latent Dirichlet Allocation, LDA), Doc2Vec, Word2Vec and LIWC. For classification, they used Random Forest Classifier. The best accuracy for their model came when TF-IDF was used with LDA. Word2Vec also gave good accuracy.

3.3 Summary

All the above literature review is related to my work and objective because these papers focus on detection of personality of an individual using their textual data from various resources like Facebook, Twitter etc. Some research papers are related to deep learning, some related to a neural network, text preprocessing, some are related to machine learning and some are related to pretrained transformer models like BERT. This literature review helped me in getting a direction for my research and different variations of several deep learning, machine learning and pretrained transformer models. This helped me in selecting an appropriate methodology for solving the problem of personality detection using text.

In this chapter, a literature review for the related work was to be discussed, the next chapter includes the proposed methodology and its implementation details.

Chapter 4

Proposed System

4.1 Introduction

After going through many related works and performing the literature review we finally, came up with a solution and named it automated-approach. In this approach, we acquired two MBTI datasets from Kaggle. Later, data preprocessing was applied. Afterward, this dataset was trained and tested on two types of models.

i. **Machine learning:** Support Vector Machines (SVM), Linear Regression (LR), Multi-Layer Perceptron (MLP) and XGBoost were used in experimentation of ML model.

ii. **Deep Learning:** Convolutional Neural Network was used in Deep learning in this research.

4.1.1 Machine Learning

Machine learning (ML) to consequently learn from data, progress execution from encounters, and predict things without being expressly modified. Employments data to identify different designs in a given dataset. It can learn from past information and move forward naturally. It may be an information-driven innovation that's nearly comparative to data mining because it deals with a huge amount of data.
[30] Figure 4.1 Block diagram describes the functionality of the machine learning algorithm:

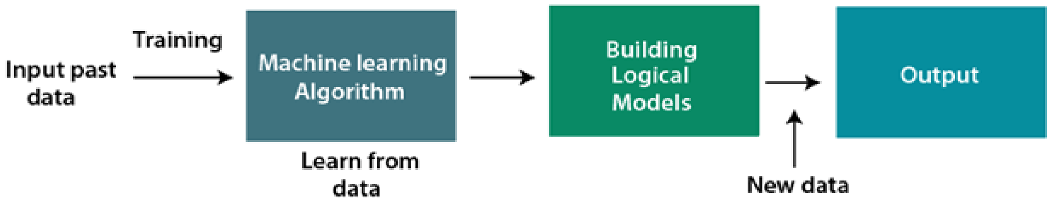


Figure 4.1: Block Building the Machine Learning Algorithm

4.1.2 Deep Learning

Machine Learning is an umbrella under which deep learning falls. It is a discipline that depends on learning and enhancing on its own by inspecting computer algorithms. While less difficult ideas are utilized in machine learning, deep learning works with artificial neural networks, which are intended to mimic how people think and learn. As of not long ago, neural networks were restricted by computing power and subsequently were restricted in intricacy. Be that as it may, evolution in Big Data analytics have allowed bigger, refined neural networks, permitting computers to notice, learn, and respond to complex circumstances quicker than people. Deep learning has supported tedious tasks like classification of images, language interpretation and recognition of speech. It very well may be utilized to tackle any problem that involves recognition of patterns and without needing any human intercession.

4.2 Classification of Machine Learning Algorithm

Machine learning algorithm is classify in to three types. [31] Figure 4.2 shows classification of machine learning.

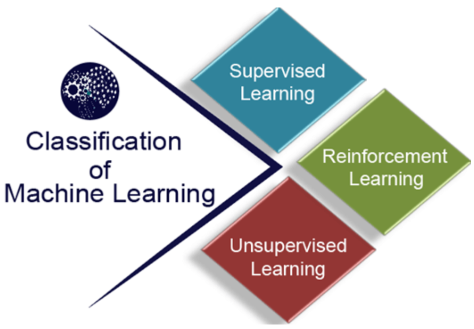


Figure 4.2: Classification of Machine Learning Algorithm

4.2.1 Supervised Learning

Supervised learning could be a type of learning strategy in which we offer sample labeled information to the framework to prepare it, and on that premise, it predicts the yield. Supervised learning can be grouped further in two categories of algorithms.

4.2.2 Classification

In machine learning,
Predictive modeling problem labels to predict the classes are known as classification. Types of classification

- Binary Classification
- Multi-Class Classification
- Multi-Label Classification
- Imbalanced classification

Binary classification refers to that classification that have 2 class labels.
Multi-class classification is term as a classification that has more than two class names.
Multi-label classification is a term as to that classification that has two or more class labels, where the prediction might include one or more class labels.
Multi-label classification is defined, classification may be predicted that has two or more class labels.
Imbalanced classification alludes to classification where the number of illustrations in each class is unequally conveyed.
Regression models are being utilized to forecast a continuous value.

4.2.3 Unsupervised Learning

Unsupervised learning could be a sort of learning strategy in which a machine learns or work without any supervision.

4.2.4 Reinforcement Learning

Reinforcement learning Could be a feedback-based learning strategy, in which a learning specialist gets a remunerate for each correct activity and gets a penalty for each off-base activity.

Supervised (ML) based model built upon CPD items, in this study the multi-class classification method is used.

4.3 SUPERVISED LEARNING CLASSIFICATION ALGORITHM

1. Support Vector Machines
2. Logistic Regression
3. Multi-layer Perceptron
4. Gradient Boosting Classifiers

4.3.1 Support Vector Machines (SVM)

SVMs are a set of supervised learning methods utilized for class, regression. An SVM demonstration could be a representation of different preparing in a hyperplane in a multi-dimensional space.The hyperplane might be generated iteratively through the SVM technique can be minimized errors. The purpose of SVM is to divide into training to discover a maximum marginal hyperplane (MMH).

4.3.2 Logistic Regression

Logistic regression could be a classification algorithm utilized to decide the chances of occasion success and occasion failure. Utilized when the subordinate variable is double (0/1, Genuine / Wrong, Yes / No) by default. Underpins classifying data into discrete categories by considering relationships from a given set of labeled information. It can be simple to change over into numerous categories(multi-country retreats) and the natural concept of guessing in the classroom. It makes non-judgmental assumptions about class allocation in the feature space.

4.3.3 Multi-Layer Perceptron (MLP)

Multilayer perceptron (MLP) may be an area of the artificial neural network (ANN). The MLP-Classifier is predicated on the thought of a Neural network to do the work of agreement. MLP incorporates at slightest three layers of nodes: input layer, hidden layer, and output body. In addition to input notes, each node may be a neuron that employments a nonlinear activation work..

4.3.4 Gradient Boosting Classifiers

Gradient boosting classifiers are a bunch of algorithms that combine powerless learning models to make effective prescient models. It usually provides unpredictable ac-curacy predictions. Multiple frequencies can magnify a variety of loss functions and provide many parameter correction options that make the task less visible.

4.4 DEEP LEARNING ALGORITHM (CNN)

Convolutional Neural Networks (CNN) were first created for image processing for recognition of objects from images. They attained record breaking results in the domain of image processing. However, CNN can also be used for Natural Language Processing (NLP) with some changes. On account of NLP undertakings, i.e., when applied to textual data rather than images, we have a 1 layered cluster addressing the text. Therefore, here the engineering of the CNN is changed to 1D convolutional-and-pooling layers.

4.4.1 Convolution Layer

The extraction of high level features from the input data is the main objective of convolution layers. Ordinarily, the primary convolution layer is liable for catching the Low-Level features but the model adjusts to the High-Level features too, with added layers, producing an organization having the healthy comprehension of text in the dataset, similar to how a human would.

Suppose we have a sequence of words, represented as,$w_{1:n} = w_1, w_2, ..., w_n$, where every word is related to an embedding vector of dimension d. When a k-sized sliding window is moved over the text, a 1D convolution having a width k is created. The same kernel or

convolution layer is then applied to every window in the text by getting a dot product of a weight vector u and the concatenation of embedding vectors for a given window. After that, a non-linear activation function g is applied.
Taking into account a window of words, $w_i, w_{i+1}, ..., w_{i+k}$ the linked vector of the ith window at that point is:

$$\boldsymbol{x_i} = [\boldsymbol{w_i}, \boldsymbol{w_{i+1}}, \ldots, \boldsymbol{w_{i+k}}] \in \boldsymbol{R}^{\,\boldsymbol{k} \times \boldsymbol{d}}$$

We get scalar values r_i, for ith window, by applying a convolution filter to each window, given by:

$$\boldsymbol{r_i} = \boldsymbol{g}(\boldsymbol{x_i} \cdot \boldsymbol{u}) \in \boldsymbol{R}$$

Typically, more filters are applied, given by, $u_1, ..., u_l$, where l is the dimensional output. This can then be represented as a vector multiplied by a matrix U and with an addition of a bias term b:

$$\mathbf{r}_i = g(x_i \cdot U + b)$$

where

$$\mathrm{r}_i \in R^l, \quad x_i \in R^{\,k \times d}, \quad U \in R^{\,k \cdot d \times l} \ \text{ and } \ b \in R^l$$

Channels are used when different characteristics or views of input data are required, where each view will be stored in different matrix. It's normal to apply an alternate arrangement of filters to each channel, and afterward get a single vector by joining the three resulting vectors. Likewise, the numerous channels worldview can also be applied in text handling. For instance, for a given expression or window of text, one channel could be the succession of words, one more channel the grouping of relating Parts-of-speech (POS) tags, and a third one the shape of the words, as shown in figure 4.3:

Words	prefer	staying	at	home	and	reading
POS tags	VERB	VERB	PROP	NOUN	CONJ	VERB
Shape	Xxxx	Xxxxx	xx	xxxx	xxx	Xxxxx

Figure 4.3: Example for channels in text processing

When convolution is applied over words, POS tags and shape, they will result in m vectors, each. These three different channels can then be combined either by summation:

$$p_i = words_{1:m} + pos_{1:m} + shapes_{1:m}$$

or using concatenation:

$$p_i = [words_{1:m} : pos_{1:m} : shapes_{1:m}]$$

It should be kept in mind that each channel can in any case have various convolutions that read the source text utilizing different kernel sizes, for example, applying different setting windows over POS-tags, words, or shapes.

4.4.2 Pooling Layer

The main focus of the pooling layer is to apply dimensionality reduction by combining the resulting vectors obtained from various convolution windows to get a single l-dimension vector. In the most ideal scenario, pooling will get the most relevant features of the textual data.

There are two kinds of Pooling: Average Pooling and Max Pooling. Max Pooling yields the greatest worth from part of the input data covered by Kernel. In opposition, Average Pooling yields normal of relative multitude of values from the piece of input data covered by the Kernel.

Noise is suppressed by using Max Pooling. It disposes of uproarious initiations through and through and furthermore performs de-noising alongside reducing dimensionality. Contrastingly, Average Pooling just performs dimensionality reduction. Consequently, it can be said that Max Pooling plays out significantly superior compared to Average Pooling.

4.4.3 Dropout Layer

Dropouts are the regularization procedure that is utilized for prevention of over-fitting in the model. Dropouts are added to randomly switch off some level of neurons of the network. When the neurons are dropped, the connections to those neurons are also dropped, whether incoming or outgoing. The purpose of this technique is to improve the learning of the model.

4.4.4 Fully Connected Layer

A fully connected layer is added at the end for classification. Since the input textual data is now changed into a reasonable structure for Multi-Level Perceptron, the text data will be flattened into a segment vector. A feed-forward neural network is given this flattened vector as input. Further, every iteration is applied with back-propagation while the model is trained. Over a progression of epochs, the model can recognize ruling and determined low-level elements in text data and group them utilizing the Softmax Classification procedure.

4.5 LIBRARIES AND LANGUAGES

4.5.1 PYTHON

Python is object-oriented, and it translates high-quality language with dynamic semantics. The syntax of python is simple and easy-to-read emphasizes readability, hence reducing framework support costs. Python supports modules and bundles, which advance framework layout and reuse the code.

4.5.2 Python's Import Module

Python's import module works the same as the header file '#include' used in C/C++. Code can be accessed between different modules of python by importing the function or file utilized to import. The most used method of invoking the import machinery is the import statement.

4.5.3 Pandas

Pandas are provided with fast, visual, and descriptive data structures designed to work with simple "relationship" or "labeled" data. It aims to be a state-of-the-art building block for performing effective, real-world analysis of world data in Python.

4.5.4 SciPy

SciPy (articulated "Sigh Pie") is an open-source programming language for math, science,and once again search. It incorporates modules for insights, improvement, joining, straight polynomial math, Fourier transforms, sign and picture preparing, ODE solvers.

4.6 Tools

4.6.1 Introduction to Google Colab

It could be a free note pad environment that runs completely within the cloud. It has highlights that assist you to alter reports the same way you work with Google Docs. Co-lab supports numerous prevalent and high-level libraries which can be effectively stacked in your notebook.

4.6.2 Google Colab Python

Google collab is used , Python Installation, Colab Note pad bookmark border. The Earth Engine Python API can be included, within the Google Colaboratory notepad. Colab scratchpads are Jupyter cloud notebooks that work within the cloud and are exceptional coordinates with Google drive, making it smooth installation access and share.Google Colab or Collaboratory is a popular research and educational tool. Colab comes with a number of pre-installed Python libraries to help data scientists perform better.

4.7 Dataset Description

Two datasets from Kaggle were acquired namely, MBTI dataset [32] and MBTI 500 dataset [33]. Table 4.1 shows the number of records and number of classes in each of the dataset used in this research.

Table 4.1: Table of Dataset Description

SNo.	Dataset Name	No. of records	No. of classes
1.	MBTI Dataset	8675	16
2.	MBTI 500 Dataset	106067	16

4.7.1 MBTI dataset

This data was gathered through the PersonalityCafe discussion forum, as it gives an enormous choice of individuals and their MBTI character type, as well as what they have posted. It has 8675 rows and two textual columns. Each row represents one individual. One column is of last 50 posts from an individual where each post is separated using ||| and the other column tells their MBTI personality type. Figure 4.3 depicts a sample of this MBTI dataset.

	type	posts
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...
1	ENTP	'I'm finding the lack of me in these posts ver...
2	INTP	'Good one _______ https://www.youtube.com/wat...
3	INTJ	'Dear INTP, I enjoyed our conversation the o...
4	ENTJ	'You're fired.|||That's another silly misconce...

Figure 4.4: Sample of MBTI dataset

4.7.2 MBTI 500 dataset

This dataset was created by combining the above mentioned MBTI dataset [32] with another MBTI dataset [34] which used Google big query to collect this data from Reddit. This combined MBTI 500 dataset consists of 106067 rows and two columns of text data. Similar to previous dataset, every row in this dataset also represents one individual. One column has the posts from the individual and other represents the category of MBTI personality to which the individual belongs. In this dataset, each post has equal number of words, i.e. 500 words per post. Preprocessing has already been performed on this dataset. Figure 4.4 represents a sample of this MBTI 500 dataset.

	posts	type
0	know intj tool use interaction people excuse a...	INTJ
1	rap music ehh opp yeah know valid well know fa...	INTJ
2	preferably p hd low except wew lad video p min...	INTJ
3	drink like wish could drink red wine give head...	INTJ
4	space program ah bad deal meing freelance max ...	INTJ

Figure 4.5: Sample of MBTI 500 dataset

4.8 Process Flow for Machine Learning Model

Following process was performed to train a personality detection for machine learning:

i. Reading and Writing CSV Files using Pandas

ii. Preprocessing Dataset

iii. Splitting the dataset in Train and Test

iv. Training the model

v. Evaluating the model

Figure 4.6 represents the process flow of the proposed ML model.

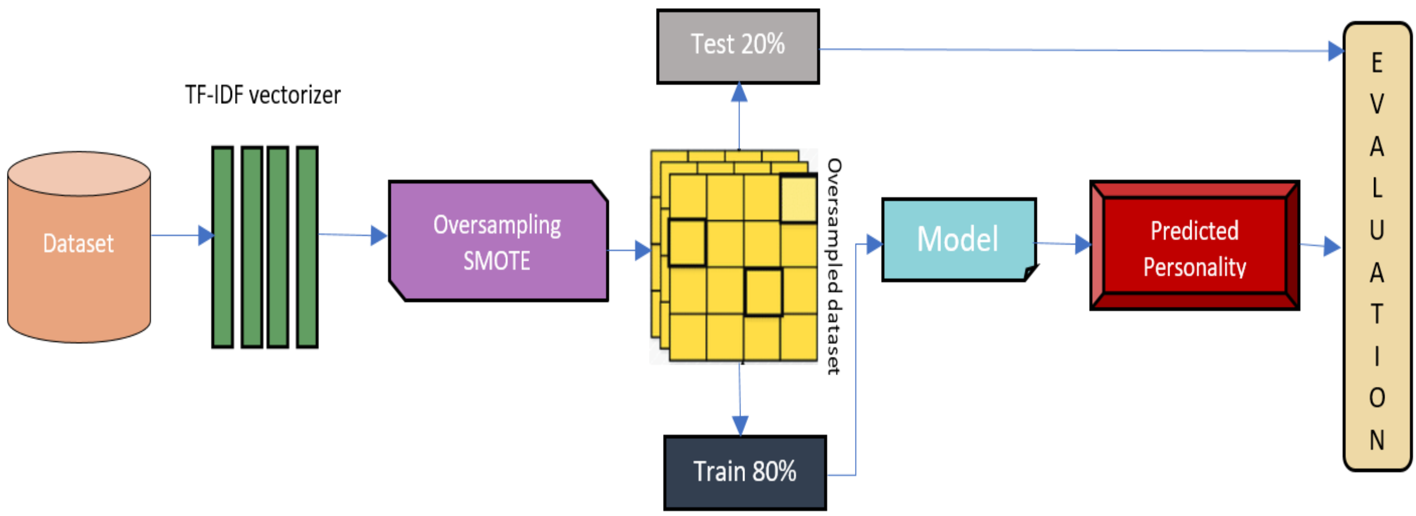


Figure 4.6: The Process Flow of the Proposed ML Model

4.8.1 Reading and Writing CSV Files using Pandas

Pandas is a completely effective and famous framework for data evaluation and manipulation. One of the maximum striking features of Pandas is its potential to look at and write numerous kinds of files along with CSV and Excel. you may efficiently and easily manipulate CSV documents in Pandas.

4.8.2 Preprocessing the Dataset

Data Preprocessing is the method of making data appropriate for utilization while training a model.

4.8.2.1 Cleaning the Dataset

Data cleaning was done only on MBTI dataset as MBTI 500 dataset has cleaned text. The text data in MBTI dataset was lower cased, stop words and punctuations were removed. The words were converted to their root words with lemmatization and stemming using 'WordNetLemmatizer' and 'SnowballStemmer'. As this dataset was collected from a forum of discussion regarding personality types, there were a lot of keywords like INTJ, ESFP etc, which could affect the accuracy of the machine learning model. Hence these keywords were removed from the data.

4.8.2.2 Vectorizing the Text Data

When any algorithm has to be applied to text, textual data should be changed to a numeric structure. Subsequently, there emerges a requirement for some preprocessing strategies that can change the textual data to numbers. In this research, TF-IDF was used for this purpose on both datasets. TF-IDF is a preprocessing strategy that can create a numeric structure from the given text. TF-IDF works by relatively expanding the times any word pops up in the record however is counterbalance the quantity of records in which it is available. Thus, words like 'that', 'is' and so on, that are frequently present in every one of the records are not given an exceptionally high position. In any case, a word that is available too often in a couple of the records will be given a higher position as it very well might indicate of the context of the record.

4.8.2.3 Oversampling

As both of these datasets have the class imbalance problem, as shown in Figure 4.7 (for MBTI dataset) and Figure 4.9 (for MBTI 500 dataset). A class imbalance is when occurrence of some classes is far more than others in a dataset. The problem arises because mostly machine learning model will give a high accuracy if the class with majority of occurrences in the dataset is predicted correctly, but the model will not be able to predict the class with far less occurrences in the dataset, concluding that the model is only accurate for majority classes. To solve this problem, Synthetic Minority Oversampling Technique (SMOTE) was applied on both dataset. SMOTE is an oversampling strategy where the engineered records are created for the minority class. This algorithm assists with conquering the over-fitting issue presented by oversampling the data randomly. It centers around the feature space to create new records with the assistance of addition between the positive occurrences that lie together. Figure 4.7 and Figure 4.8 depicts the occurrences of each class in the MBTI dataset before and after the application of SMOTE oversampling, respectively.

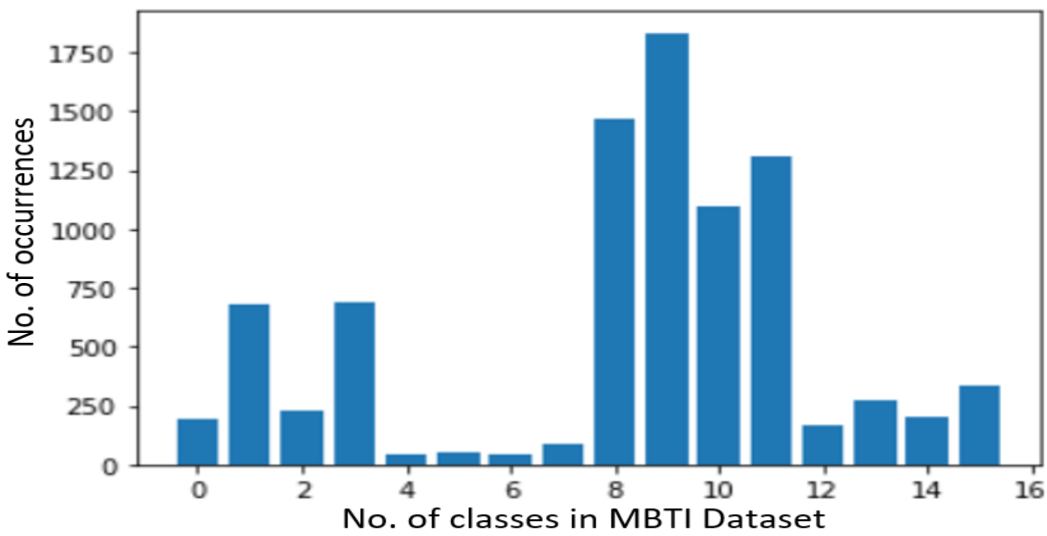


Figure 4.7: The no. of occurrences of each class in the MBTI dataset before SMOTE

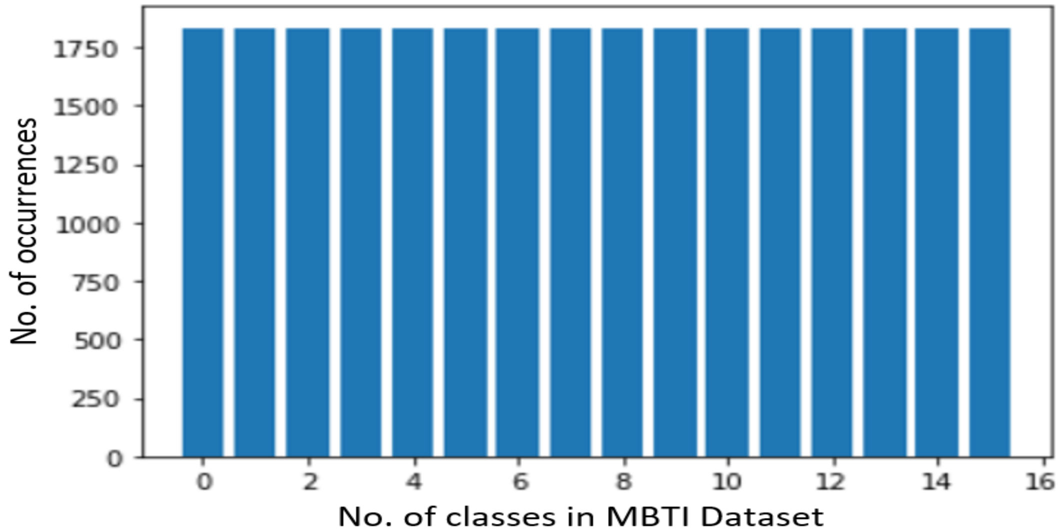


Figure 4.8: The no. of occurrences of each class in the MBTI dataset after SMOTE

Figure 4.9 and Figure 4.10 depicts the occurrences of each class in the MBTI 500 dataset before and after the application of SMOTE oversampling, respectively.

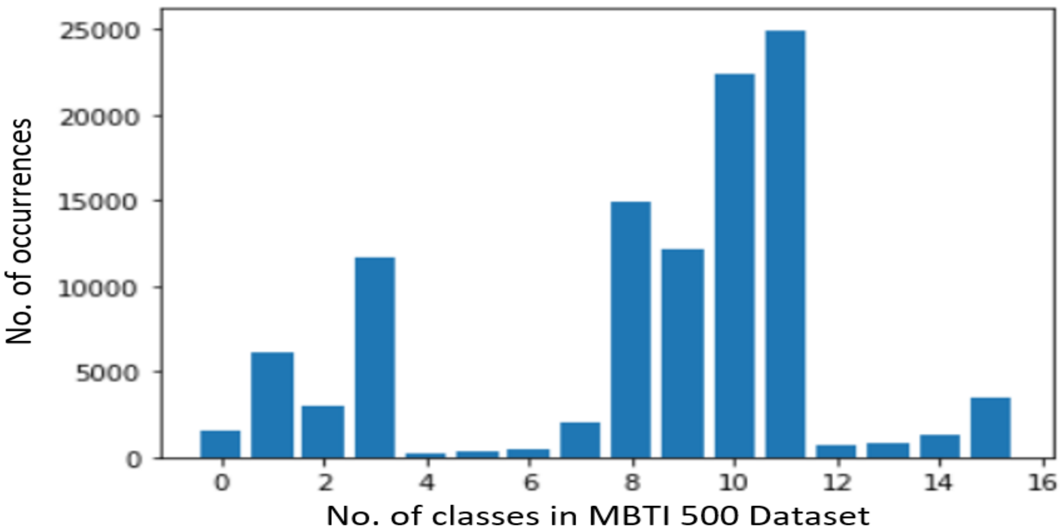


Figure 4.9: The no. of occurrences of each class in the MBTI 500 dataset before SMOTE

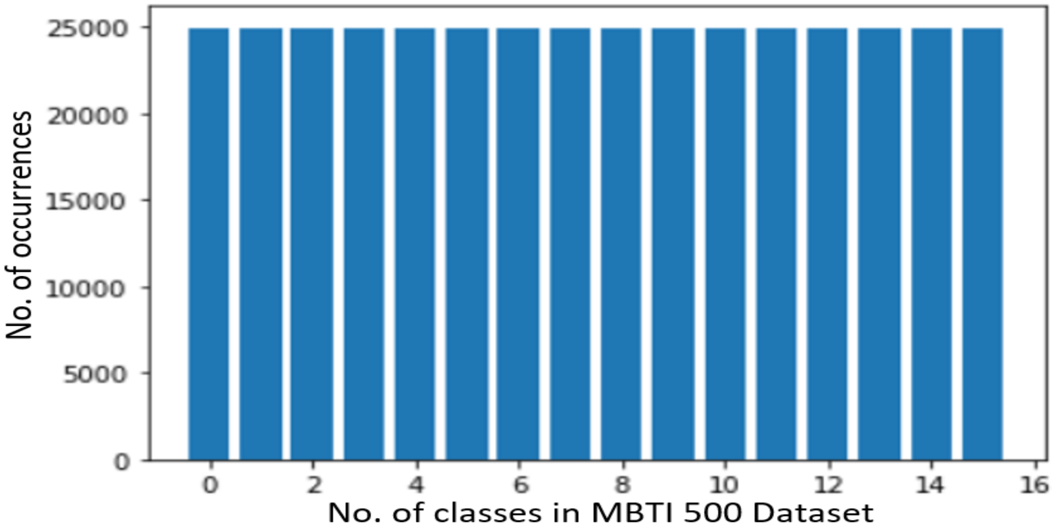


Figure 4.10: The no. of occurrences of each class in the MBTI 500 dataset after SMOTE

4.8.3 Splitting the Dataset in Train and Test

The datasets were then divided into 2 parts. 80% for train and 20% for test. The split data function of sklearn created the matrices $X_{t}rainandX_{t}estwhichcontainedthefeaturevectorofthetextdatafortrain$

4.8.4 Training

The set of data used to fit the model is called the training dataset.
Command used for training the model on multi class classifier is:

$$classifier.fit(X\_train, y\_train)$$

4.8.5 Evaluation

The model was evaluated on the 20% testing data from the datasets obtained using split function. Command used for predicting the model on multi class classifier is:

$$y\_pred = classifier.predict(X\_test)$$

Accuracy, F1-score, precision and weighted recall were used as evaluating metrics for the model. This is explained in section xyz in detail.

4.9 Process Flow for Deep Learning Model

Following process was performed to train a personality detection for machine learning:

i. Preprocessing the Dataset

ii. Vectorizing the Dataset

iii. Splitting the Dataset in Train, Test and Validation

iv. Applying GLOVE embedding

v. Convolutional Neural Network Model

vi. Evaluating the Model

Figure 4.11 represents the process flow of the proposed deep learning model.

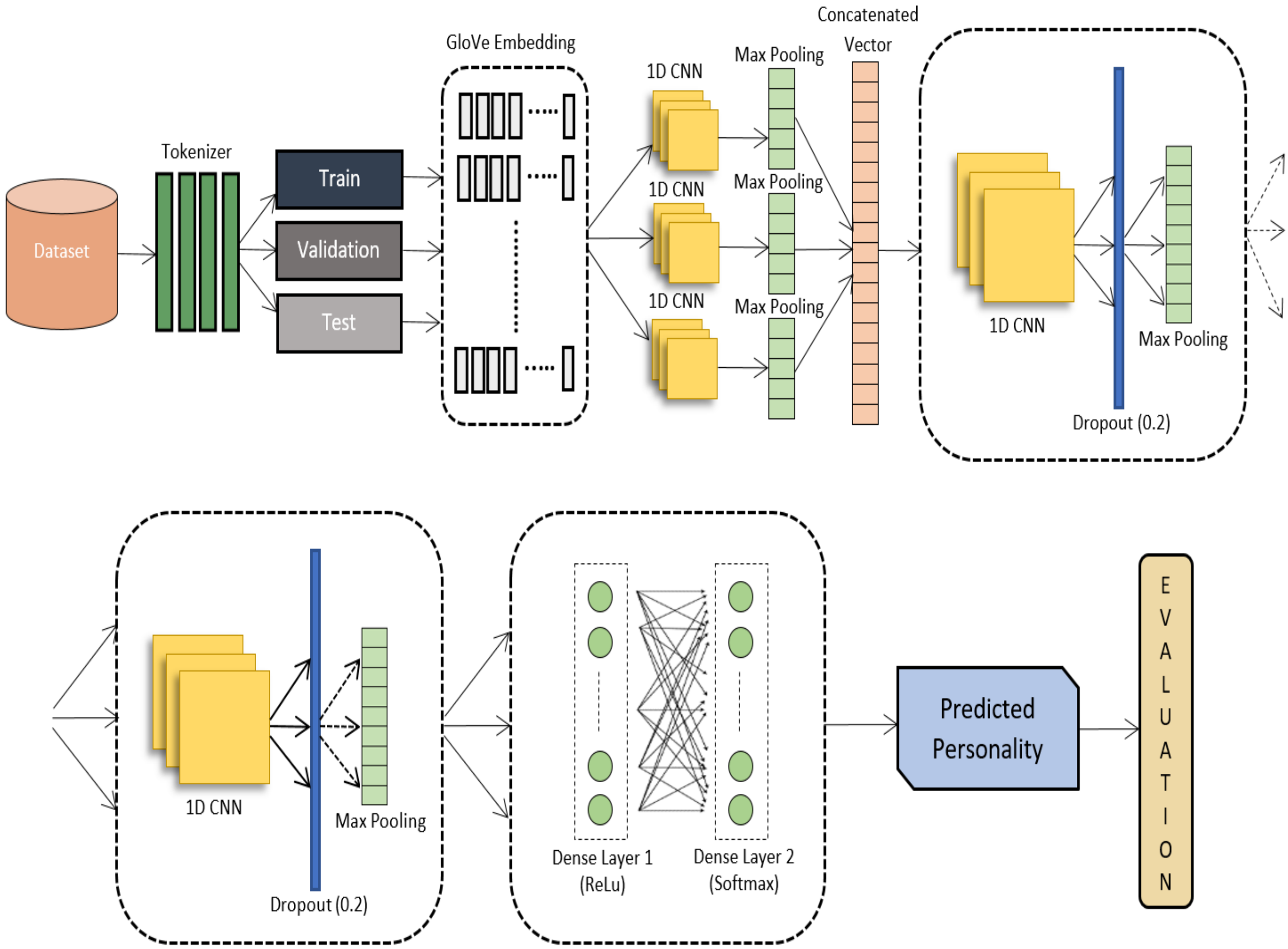


Figure 4.11: The Process Flow of the Proposed Deep Learning Model

4.9.1 Preprocessing the Dataset

As mentioned in segment 4.8.2.1, MBTI 500 dataset is a preprocessed data so this data cleaning was performed only for MBTI dataset. For MBTI dataset, text was first converted to lower case and then removal of stop words and punctuations was done. Each word was then changed to it's root word using stemming.

4.9.2 Vectorizing the Dataset

Keras preprocessing Tokenizer was used for vectorizing the datasets. The word index was generated by the tokenizer on the dataset content. The sequences were then padded using the maximum sequence length of the dataset.

4.9.3 Splitting the Dataset in Train, Test and Validation

The datasets were randomly shuffled first and then they were divided into three parts. 80% of the dataset was used for training the model, 10% was used for validation and 10% was used for testing the model.

4.9.4 Applying GLOVE embedding

GloVe (Global Vectors) is an unsupervised learning algorithm for acquiring vector portrayals for words. To have pre-trained word vectors with contrasts in tokens, size, and vocab size the models were trained on Twitter, Wiki and normal crept information. For this research, we will utilize the glove.6b.100d.txt pretrained glove word vector.

4.9.5 Convolutional Neural Network Model

In the CNN architecture, the embedding layer was used as input to three different filter windows of 3, 4 and 5 in three 1D convolution and max pooling operations. After obtaining the resulting vectors from these three convolution layers,they were then combined using concatenation. The vector acquired from this concatenation was then fed to convolution layers followed by a dropout layer with a dropout rate of 0.2. The 'Max' pooling operation was then applied to the regularized vector with a pool size of 5 in one layer and 20 in the next layer.

The flattening layer was used on the resultant vector to get a segment vector which is further fed to a feed forward neural network with one hidden layer using 'Dense' layer with 'relu' activation function. After training the neural network on 95 epochs using a batch size of 32 using softmax activation function, the model classified the text into 16 categories of MBTI personality. The 'categorical-cross entropy' as the loss function and 'rmsprop' as the optimizer were used for model compilation.

4.9.6 Evaluating the Model

The model was validated on the 10% validation data from the datasets and was evaluated on the 10% testing data from the datasets. Accuracy, F1-score, precision and recall were used as evaluating metrics for the model. This is explained in section xyz in detail.

4.10 Summary

In this chapter, the algorithms used for ML and DL are explained. The libraries, language and tools which were used in this research were also discussed in this chapter. Further, the proposed technique outline and its flow are characterized for Personality Detection. Step-by-step process involved in training the detection models for both ML and DL are established. After following the complete process, the trained models are achieved, which are used for testing purposes to evaluate the model.

Chapter 5

Evaluating Trained Model for Automatic Personality Detection

5.1 Introduction

Analyzing and simplifying the data is a necessary portion of the assessment, and there are numerous assessment strategies accessible. This can be to organize and make the results that are understood, therefore, that can utilize the result and progress them. The machine learning models were evaluated on the 20% test data and the deep learning model was evaluated on 10% test data. The test data in both was from the MBTI and MBTI 500 dataset.

5.2 Experimentation Setup and Metrics

The evaluation metrics results are explained and discussed below. In order to understand the metrics we have to understand the following terms.

5.2.1 True Positive (TP)

The real esteem is true/positive and the model predicted that's true, i.e. the model predicted class A as A.

5.2.2 True Negative (TN)

The actual esteem was negative, and the model predicted it was negative, i.e. the model predicted any other class as not A.

5.2.3 False Positive (FP)

The actual value is incorrect/wrong (Type I error), but the model prediction is that true, i.e. the model predicted any other class as A.

5.2.4 False Negative (FN)

The actual value is valid (Type II error), but the model prediction is negative, i.e. the model predicted class A as not being A.

5.3 Formulas

5.3.1 Accuracy

Accuracy is a metric that is usually describes the performance of model across all classes. It is calculated by dividing the total number of correct predictions by the total number of predictions, given by the following formula:

$$Accuracy = \frac{TruePositives + TrueNegatives}{Total predictions}$$

5.3.2 Precision

It is defined, as calculating the value of a positive prediction made by a classifier.The formula explains that it divides true positive values from the sum of the all true positive values and negative values. The precision reflects how reliable the model is in classifying samples as Positive.
Precision is calculated by:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

5.3.3 Recall

It is defined as when the classifier calculates the actual positive value divided by the positive and false negative values, the formula is explained. High recall value means that the correct classifier and has a low negative number is false negatives. The recall measures the model's capacity to detect Positive samples. The more positive samples detected, the higher will be the recall,
Recall is calculated by:

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

5.3.4 F1 score

F1-score is defined , ratio between Precision and recall. In which the F1-score comes to its best esteem of 1 and the worst rating to 0. The restricted commitment of precision and memory is rise to the F1-score, and the consonant definition helps to discover the most excellent exchange between the two numbers. F1-Score is calculated by

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

In addition of "Precision" and "Recall" can refer to both binary and multi-stage divisions, in the case of the binary we view only the Positive category, so True Negative items do not have 5 Metrics temporarily in the case of multiple categories looking at all classes individually.

5.3.5 Weighted Average

The weighted average is defined as the weighted average of precisionrecall, and f1-score.

5.4 Evaluating Trained Models for Personality Prediction

Both datasets, MBTI and MBTI 500 were randomly divided into 80% train data and 20% test data. The classic machine learning models for multi-class classification were trained using train data. These trained models were then evaluated using the test data. The metrics mentioned in segment 5.3 were used for evaluation of different classifier models on the test data.

5.4.1 Evaluation of Classical Machine Learning Classifiers without SMOTE

Firstly, classical machine learning models were trained on both datasets without applying SMOTE on datasets. The problem of imbalanced classes was not addressed for this experiment. The comparison of test results of proposed machine learning models, without re-sampling of datasets, on MBTI dataset and MBTI 500 dataset, with the baseline models [1] is represented in Table 5.1 and Table 5.2, respectively.

Table 5.1: Test Results of different classical ML classifiers without SMOTE on MBTI dataset for Multi-class classification

Algorithms	Accuracy	Precision	Recall	F1 Score	Accuracy [1]	Precision [1]	Recall [1]	F1 Score [1]
MLP	40.23%	36.12%	40.23%	34.55%	-	-	-	-
LR	41.38%	35.88%	41.38%	35.53%	-	-	-	-
SVM	40.63%	**96.73%**	39.59%	37.49%	79.5%	80.5%	80.5%	80.5%
XGB	30.95%	32.13%	30.95%	27.23%	85%	84.25%	84.25%	84%
NB	-	-	-	-	78.5%	77%	79.75%	77.25%

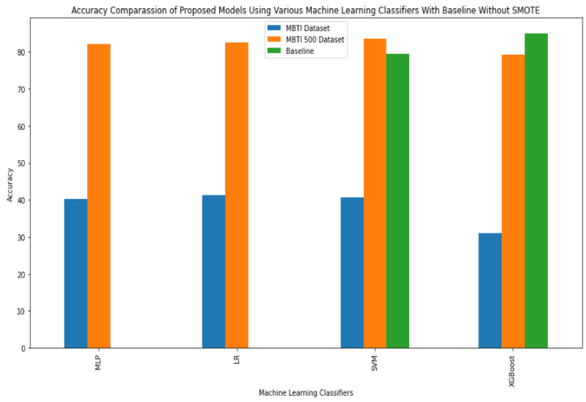
As presented in Table 5.1, all machine learning classifiers in [1] model performed better than the proposed methodology in this research, on MBTI dataset, in terms of all evaluation metrics i.e. accuracy, recall, precision and f1-score. However, only the precision of SVM classifier in our proposed model is 96.73%, which is higher than the baseline paper [1].
The Machine Learning models in paper [1] were only trained on MBTI dataset but as mentioned in segment 4.7, the MBTI500 dataset is an extension of MBTI dataset, which contains additional records other than that of MBTI dataset. Therefore, a comparison of the proposed methodology with the baseline paper [1] on MBTI500 dataset has been represented in Table 5.2.

Table 5.2: Test Results of different classical ML classifiers without SMOTE on MBTI 500 dataset for Multi-class classification

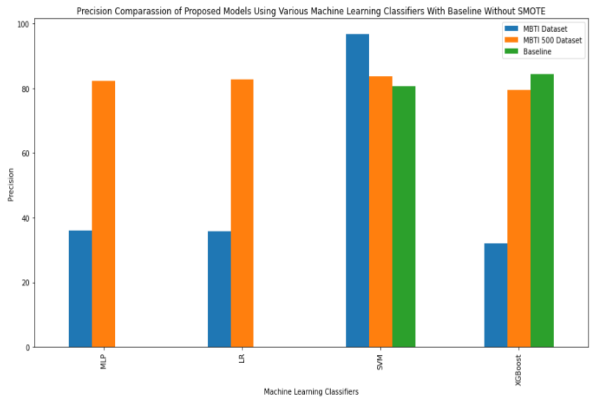
Algorithms	Accuracy	Precision	Recall	F1 Score	Accuracy (MBTI dataset) [1]	Precision (MBTI dataset) [1]	Recall (MBTI dataset) [1]	F1-Score (MBTI dataset) [1]
MLP	82.17%	82.26%	82.17%	82.11%	-	-	-	-
LR	82.48%	82.72%	82.48%	82.21%	-	-	-	-
SVM	**83.54%**	**83.62%**	**83.54%**	**83.44%**	79.5%	80.5%	80.5%	80.5%
XGB	79.34%	79.48%	79.34%	79.07%	85%	84.25%	84.25%	84%
NB	-	-	-	-	78.5%	77%	79.75%	77.25%

As shown in Table 5.2, the proposed model performs better with SVM classifier on MBTI500 dataset as compared to MLP, LR and XGBoost classifiers giving the highest accuracy, recall, precision and f1-score of *83.54%*, *83.62%*, *83.54%* and *83.44%*, respectively. The proposed model with SVM classifiers also performs better than the baseline model [1], as the data in MBTI500 dataset is extensive than MBTI dataset and hence, the model received more data for training. However, the model in [1] performs better with XGBoost Classifier, in comparison with all classifiers, even with fewer instances.

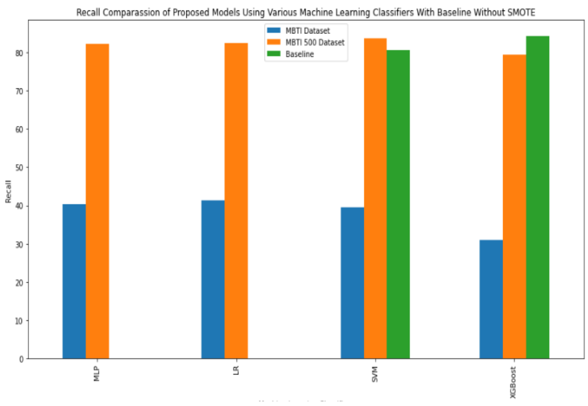
Figure 5.1 depicts the graphical comparison of evaluation metrics for machine learning classifiers on MBTI and MBTI500 datasets, without re-sampling. The comparison with baseline models [1] is also depicted in this figure. The blue bar represents the results of models trained on MBTI dataset, the orange bar represents the results of models trained on MBTI500 dataset and the green bar represents the results of models in [1]. As Table 5.1 and Table 5.2 shows that model in [1] were not trained with MLP and LR classifiers, therefore the graphs in Figure 5.1 does not include its representation (green bar) for MLP and LR classifiers.



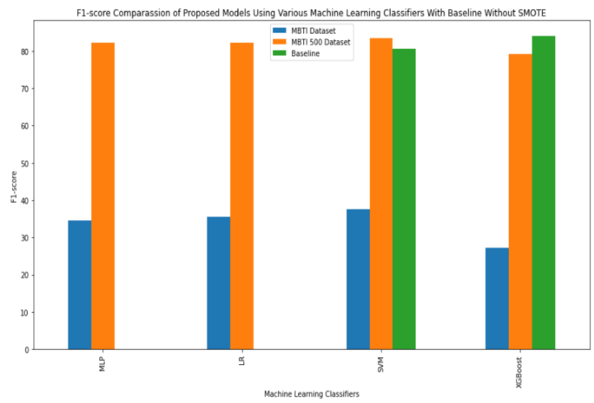
(a) Comparison of Accuracy



(b) Comparison of Precision



(c) Comparison of Recall



(d) Comparison of F1-score

Figure 5.1: Comparison of Test Results of Proposed Models Using Various Machine Learning Classifiers With Baseline Without SMOTE

5.4.2 Evaluation of Classical Machine Learning Classifiers with SMOTE

After training the classical machine learning models on both imbalanced datasets and analyzing the results, re-sampling technique, SMOTE was used to overcome the problem of imbalanced classes. The over-sampled dataset was then used to train the classical machine learning classifiers. The comparison of test results of proposed machine learning models, with re-sampling of datasets, on MBTI dataset, with the baseline models [1] is represented in Table 5.3.

Table 5.3: Test Results of different classical ML classifiers with SMOTE on MBTI dataset for Multi-class classification

Algorithms	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Accuracy (without SMOTE) [1]**	**Precision (without SMOTE) [1]**	**Recall (without SMOTE) [1]**	**F1-Score (without SMOTE) [1]**
MLP	90.14%	**90.03%**	90.14%	**90%**	-	-	-	-
LR	86.81%	86.48%	86.81%	86.56%	-	-	-	-
SVM	**90.27%**	89.78%	**90.27%**	**90%**	79.5%	80.5%	80.5%	80.5%
XGB	85%	86.83%	84.58%	85.44%	85%	84.25%	84.25%	84%
NB	-	-	-	-	78.5%	77%	79.75%	77.25%

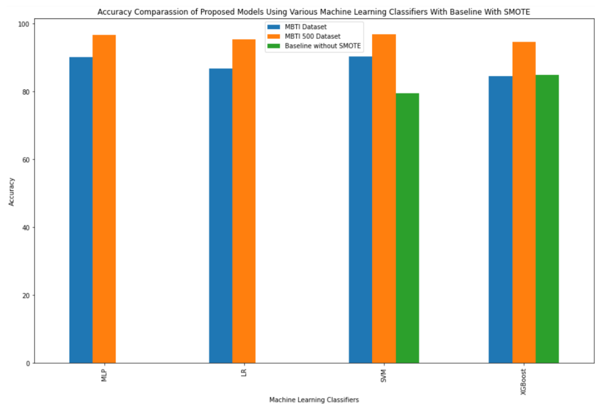
As shown in Table 5.3, after applying SMOTE on MBTI dataset and overcoming the class imbalance problem, the proposed model performed better than the baseline model [1] for all classifiers. SVM classifier gave the highest accuracy of *90.27%* with a minor difference with MLP classifier which gave accuracy of *90.14%*. MLP and SVM classifiers also gave highest f1-score of *90%*.
The comparison of test results of proposed machine learning models, with re-sampling of datasets, on MBTI500 dataset, with the baseline models [1] is represented in Table 5.4.

Table 5.4: Test Results of different classical ML classifiers with SMOTE on MBTI 500 dataset for Multi-class classification

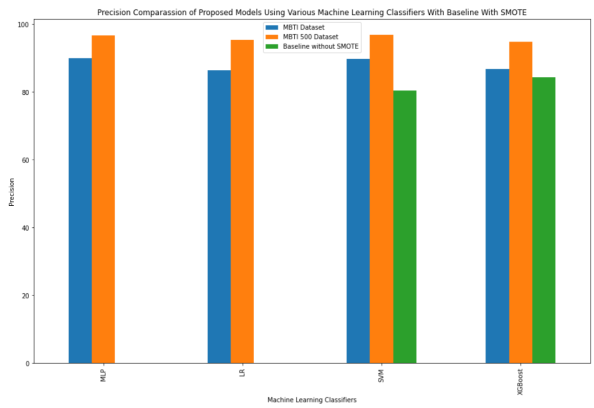
Algorithms	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Accuracy (MBTI dataset without SMOTE) [1]**	**Precision (MBTI dataset without SMOTE) [1]**	**Recall (MBTI dataset without SMOTE) [1]**	**F1-Score (MBTI dataset without SMOTE) [1]**
MLP	96.72%	96.68%	96.72%	96.70%	-	-	-	-
LR	95.35%	95.33%	95.35%	95.34%	-	-	-	-
SVM	**96.81%**	**96.77%**	**96.81%**	**96.8%**	79.5%	80.5%	80.5%	80.5%
XGB	94.68%	94.73%	94.68%	94.70%	85%	84.25%	84.25%	84%
NB	-	-	-	-	78.5%	77%	79.75%	77.25%

As Table 5.4 represents, the results on MBTI500 dataset are way better than MBTI dataset as the more data is available for training. The proposed model gives better results for personality detection with SMOTE on MBTI500 dataset than the baseline model [1]. SVM classifier gives better results than MLP, LR and XGBoost classifiers. The accuracy and f1-score of SVM and MLP classifier shows a very small difference of approximately *0.1%*.

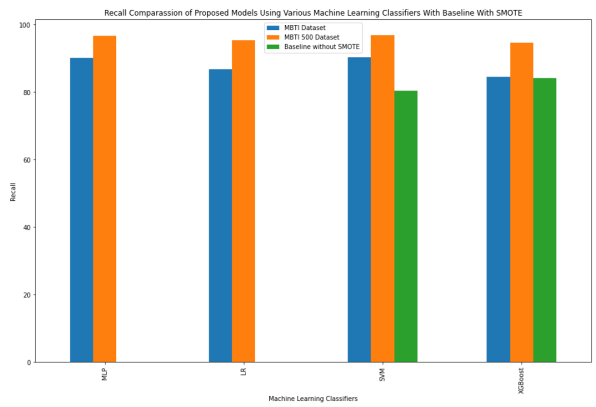
Figure 5.2 depicts the graphical comparison of evaluation metrics for machine learning classifiers on MBTI and MBTI500 datasets, when SMOTE is used for re-sampling the datasets. The comparison with baseline models [1] is also depicted in this figure. The results of models trained on MBTI dataset are represented by the blue bar, the results of models trained on MBTI500 dataset are represented by the orange bar and the results of models in [1] are represented by the green bar. As Table 5.1 and Table 5.2 shows that The model in [1] were not trained with MLP and LR classifiers, as shown in Table 5.1 and Table 5.2. Therefore the graphs in Figure 5.2 does not include its representation (green bar) for MLP and LR classifiers.



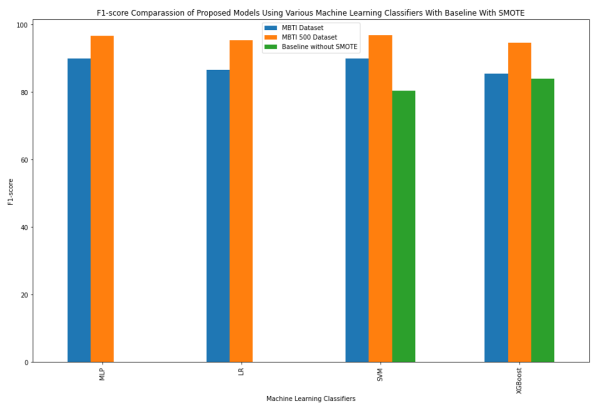
(a) Comparison of Accuracy



(b) Comparison of Precision



(c) Comparison of Recall



(d) Comparison of F1-score

Figure 5.2: Comparison of Test Results of Proposed Models Using Various Machine Learning Classifiers With Baseline With SMOTE

5.4.3 Evaluation of Convolutional Neural Networks

Following the training on classical machine learning classifiers, the deep learning approach was also used for personality prediction using MBTI and MBTI500 dataset. In deep learning, the proposed model used convolutional neural network (CNN) along with glove embeddings for the prediction of personality. In table 5.5, the results of CNN on MBTI and MBTI500 datasets are represented.

Table 5.5: Test Results of Convolutional Neural Networks

Algorithm	Accuracy	Precision	Recall	F1-Score
CNN on MBTI Dataset	99.54%	99.55%	99.55%	99.55%
CNN on MBTI500 Dataset	54.21%	69.07%	39.4%	49.95%
SVM on MBTI Dataset [1]	79.5%	80.5%	80.5%	80.5%
XGB on MBTI Dataset [1]	85%	84.25%	84.25%	84%

As depicted in table 5.5, the proposed deep learning model outperformed the baseline models [1], on MBTI dataset, by a prominent percentage. However, on MBTI500 dataset the proposed model did not produce satisfactory results with 64 batch size. The results on MBTI500 dataset can be improved in the future, by changing the hyper-parameters.

Figure 5.3 depicts the graphical representation of the comparison of results of proposed deep learning model on MBTI and MBTI500 dataset with the baseline models on MBTI dataset. The blue and yellow bar represents the result of the proposed deep learning model on MBTI and MBTI500 dataset respectively. The results of baseline SVM and XGBoost model are represented by green and red bar respectively. As shown in figure 5.3, CNN on MBTI datasets outperformed the existing models by a prominent percentage. However, it did not give good results on MBTI500 dataset.

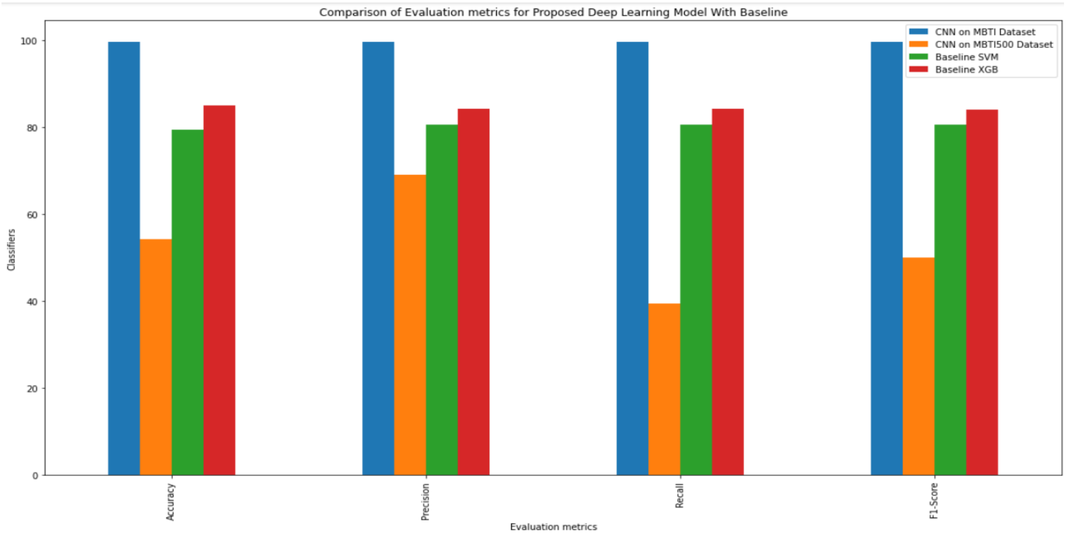


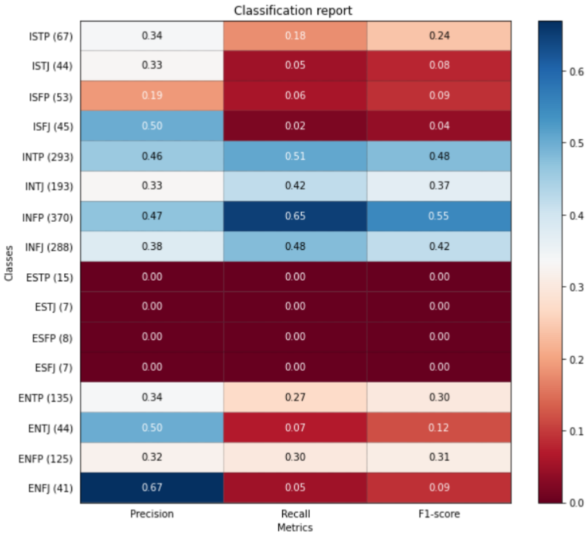
Figure 5.3: Comparison of Evaluation metrics for Proposed Deep Learning Model With Baseline

5.5 Visualization of Classification Report

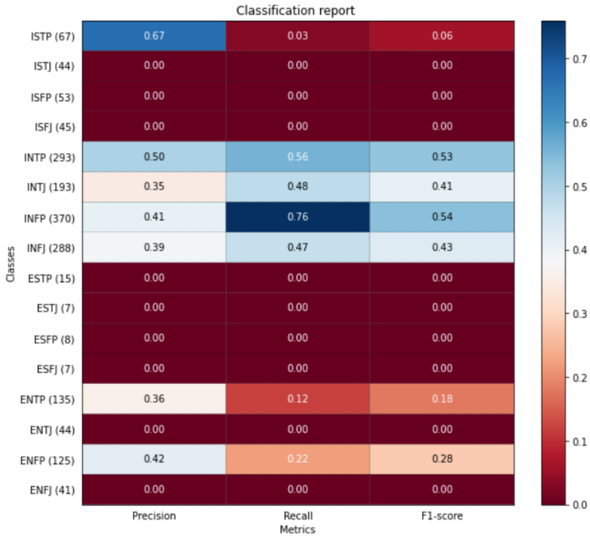
The classification model predicted the goal in a different area, i.e to assign one or greater categories in the context of a reliable variance. The multi-class classification model suggested precision, recall, F1-score, and model support score. To facilitate translation and visualisation, digital scores were combined with color-coded heat maps. All heat maps are among (zero.0, 1. zero) to facilitate comparisons of differential models among extra-ordinary classification reports. The classification score of the visualization tool represents the differences among training and some visual observations.

The classification report shows the representation of the metric classification in each category. It gives a profound sense of separating behavior with worldwide exactness that can stow away operational shortcomings in a single multi-class issue. Visual classifications reports compares the classification models. The darker blue shades represent more grounded or superior evaluated part metrics whereas the darker red shades represent the models which have lower results.

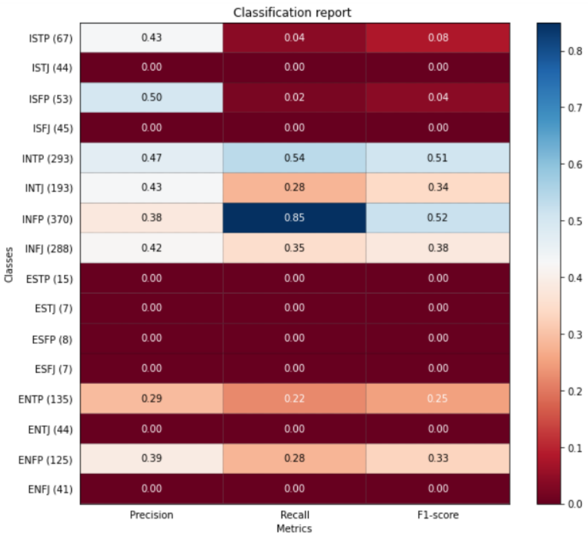
Figure 5.4 shows the heat maps of classification report of classical Machine Learning algorithms on MBTI dataset without the re-sampling of dataset.



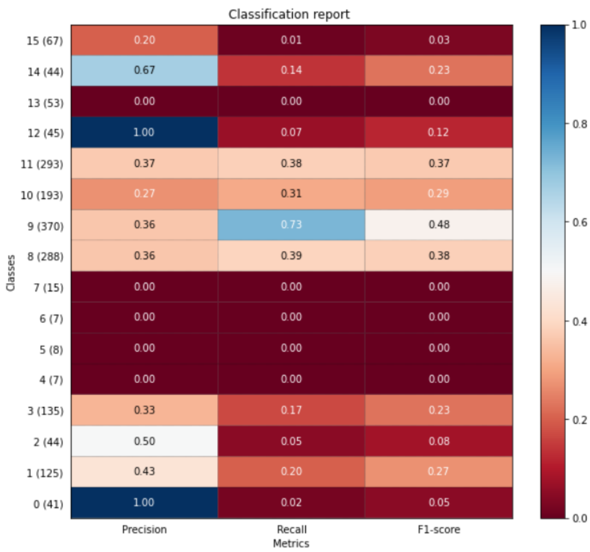
(a) SVM



(b) LR



(c) MLP

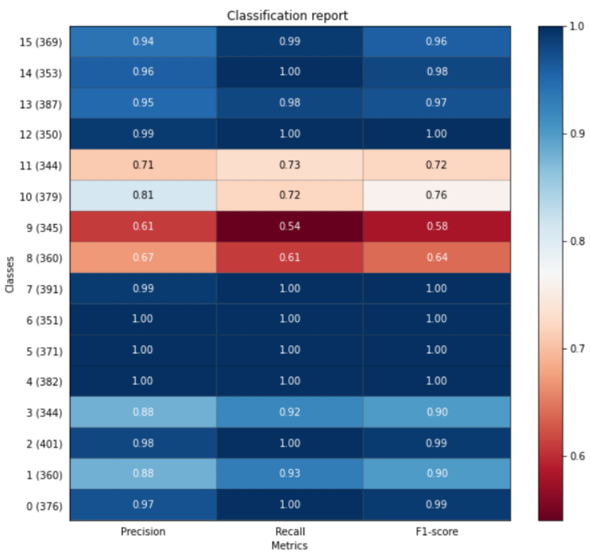


(d) XGB

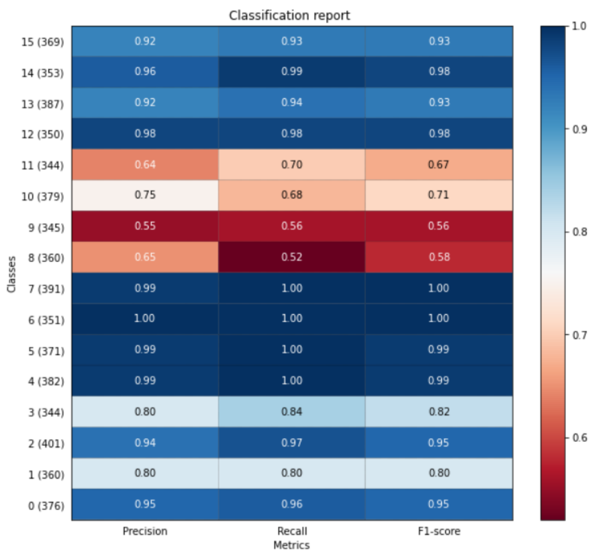
Figure 5.4: Heat maps of classification report of classical Machine Learning algorithms on MBTI dataset without SMOTE

As depicted in figure 5.4, the classes having lower instances in MBTI dataset show extremely lower results. Due to the class imbalance problem and less records available for training, classical machine learning algorithms do not give results of high standard.

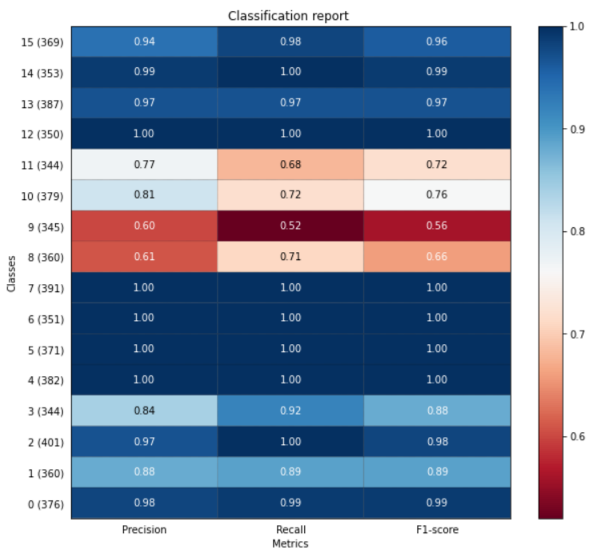
Figure 5.5 shows the heat maps of classification report of classical Machine Learning algorithms on MBTI dataset with the re-sampling of dataset using SMOTE.



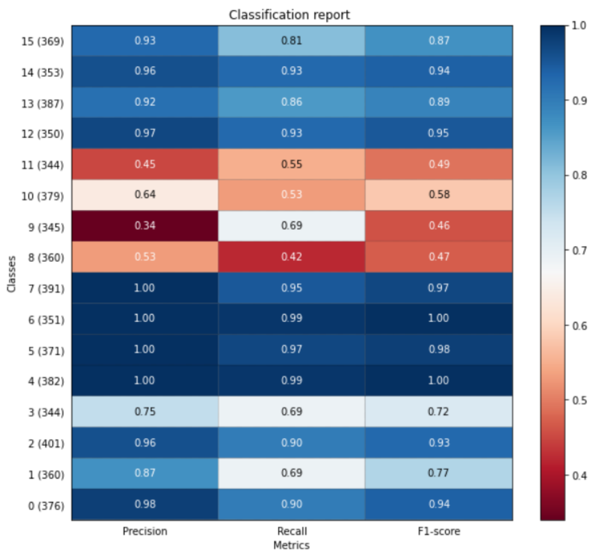
(a) SVM



(b) LR



(c) MLP

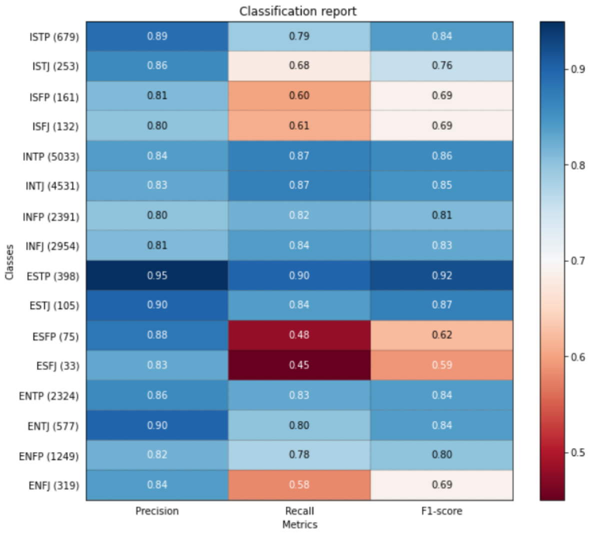


(d) XGB

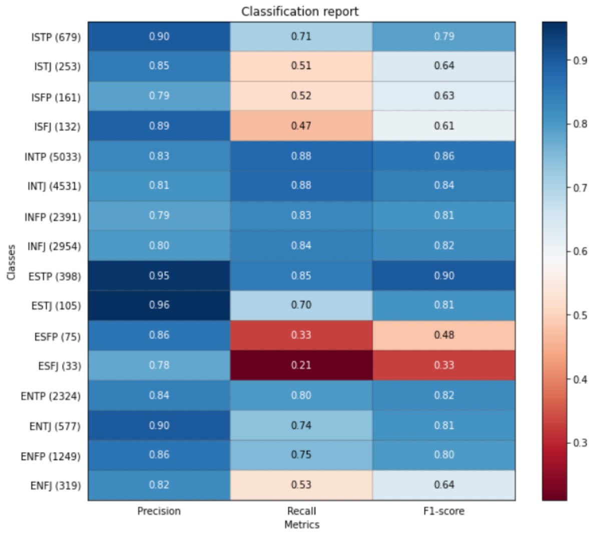
Figure 5.5: Heat maps of classification report of classical Machine Learning algorithms on MBTI dataset with SMOTE

As represented in figure 5.5, after the imbalance class problem was solved, by using SMOTE on MBTI dataset, the results of classical machine learning algorithms, for the proposed model, improved by a huge difference.

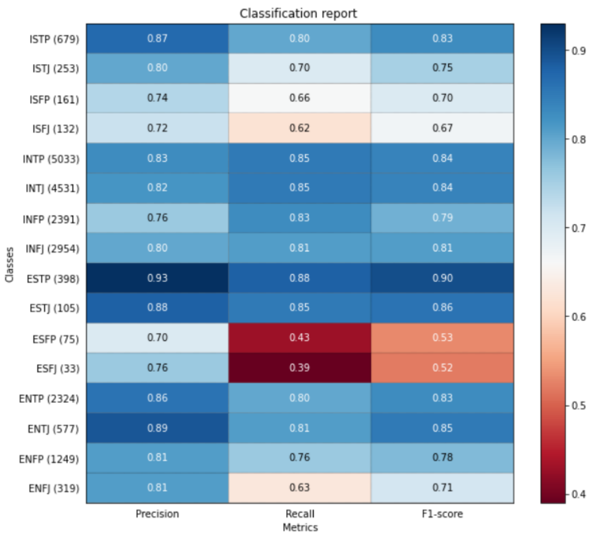
Figure 5.6 shows the heat maps of classification report of classical Machine Learning algorithms on MBTI500 dataset without the re-sampling of dataset.



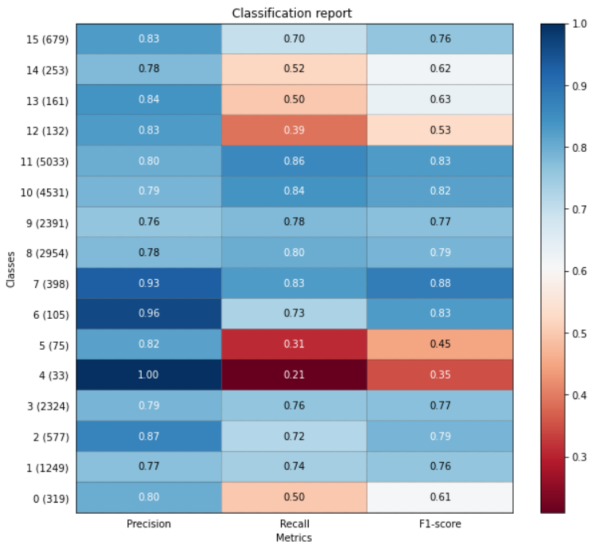
(a) SVM



(b) LR



(c) MLP

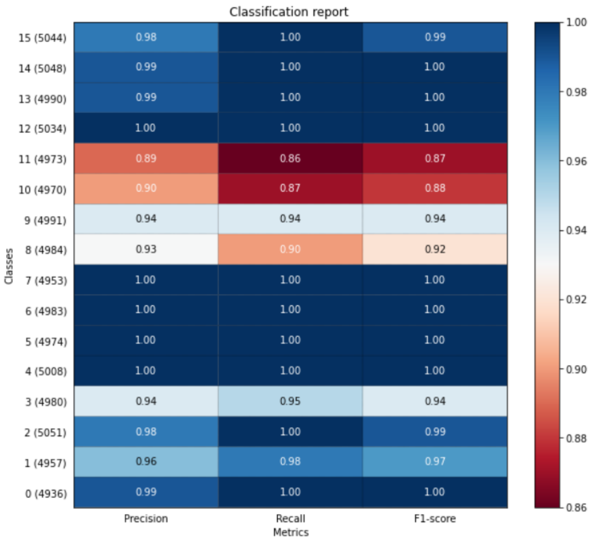


(d) XGB

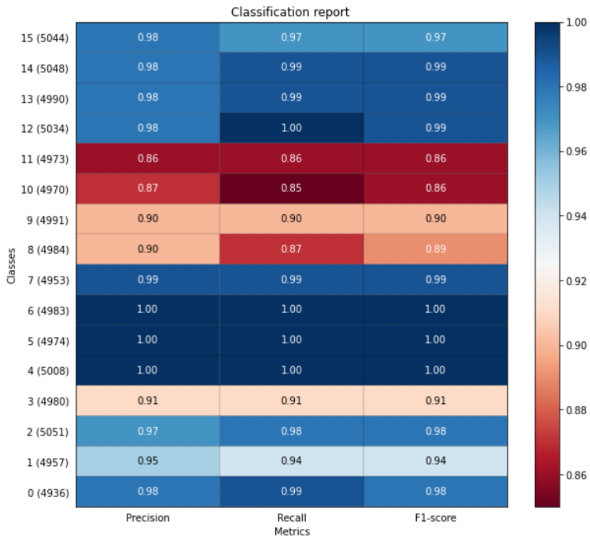
Figure 5.6: Heat maps of classification report of classical Machine Learning algorithms on MBTI500 dataset without SMOTE

As MBTI500 dataset also has the class imbalance problem, the results of classical machine learning algorithms, as shown in figure 5.6, exhibits lower recall and F1-score for classes with less instances.

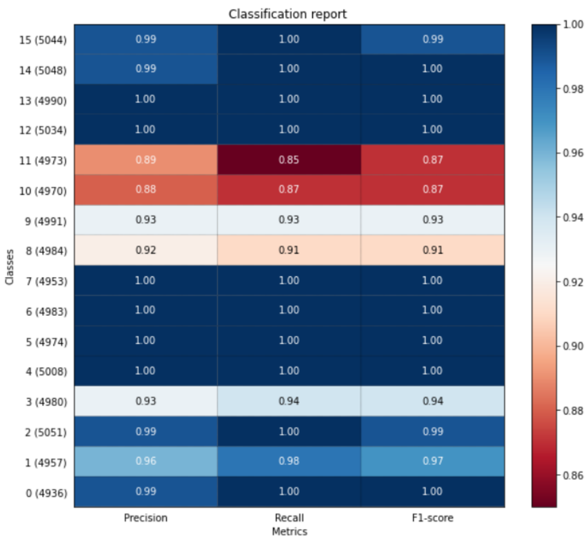
Figure 5.7 shows the heat maps of classification report of classical Machine Learning algorithms on MBTI500 dataset with the re-sampling of dataset using SMOTE.



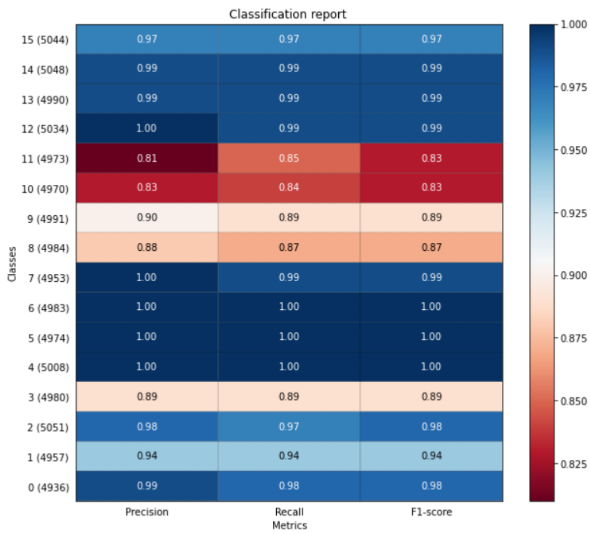
(a) SVM



(b) LR



(c) MLP



(d) XGB

Figure 5.7: Heat maps of classification report of classical Machine Learning algorithms on MBTI500 dataset with SMOTE

When the imbalance class problem in MBTI500 dataset was resolved using SMOTE, the re-

sults, as represented in figure 5.7, improved.
The classification report shows the metrics representation of the main classification in each category. This provides a deep sense of differentiating behavior with global results that can hide operational weaknesses in a single multi-class problem class. Visual classification reports are utilized to compare class models to choose better models, e.g. has stronger or superior evaluated split metrics.

5.6 Summary

In this chapter, the proposed Machine Learning and Deep Learning models were tested using evaluation metrics. The results were presented and compared in both tabular and graphical form. The proposed models were evaluated on MBTI and MBTI500 datasets with imbalanced and balanced instances (using SMOTE). In machine learning, the highest accuracy of 96.81% was achieved by SVM model, on MBTI500 dataset with SMOTE. However, the proposed deep learning model exhibited the highest result of 99.54%, on MBTI dataset.

Chapter 6

Conclusion

6.1 Introduction

In this experiment, the ML and DL trained models were tested and evaluated. The size of the dataset used was small, which did not produce much good results in case of machine learning. However, it was apparent from the testing results that deep learning model supersedes the results of machine learning models. The limitation and restrictions of the presented work are mentioned in section 6.2. Future work and recommendations have been discussed in section 6.3.

6.2 Limitations and Restrictions

i. The size of the dataset used was small for deep learning approach.

ii. Only textual data was analysed in this research for prediction of personality. No other dimensions, like images or videos, were used.

iii. Only English textual data was used in this research. Experimentation with other languages was not incorporated.

iv. Other personality models like "Big Five Model" etc were not experimented with in this research, only "MBTI personality model" was used for personality detection.

v. The proposed deep learning model did not produce satisfactory results for MBTI500 dataset.

6.3 Future Proposal

i. Performing data augmentation to increase the size of dataset.

ii. Incorporating different dimensions of data, like images and videos, for improvement of results in automatic personality detection.

iii. Languages like Urdu, Spanish and Arabic can also be used for the expansion of research scope.

iv. Predictive performance of other personality models like "Big Five Model" can be compared with that of "MBTI personality model" for result improvement.

v. The proposed deep learning model can be improved for MBTI500 dataset by optimizing hyper-parameters.

6.4 Conclusion

The principal focus of this research was to apply different machine learning and deep learning techniques on MBTI and MBTI500 datasets, in order to predict the personality of an individual. Different machine learning techniques, namely, SVM, LR, MLP and XGBoost were experimented, before and after applying class balancing technique (SMOTE), for identification of personality. For deep learning approach, CNN was trained with GloVe word embeddings for personality identification. The results of personality prediction on the benchmark MBTI dataset [32] were compared with an enhanced MBTI500 dataset [33]. The overall efficiency of the predictive model was analyzed and examined using evaluation metrics such as, accuracy, recall, precision and F1-score. The obtained results depicted that in case of machine learning, there were adequate improvements after the class balancing was applied on both datasets. In machine learning, the highest accuracy and F1-score of 96.81% and 96.8% was achieved by SVM model, respectively, on MBTI500 dataset with SMOTE. On the other hand, deep learning produces good enough result without the application of class balancing technique, giving an accuracy and F1-score of 99.54% and 99.55% on MBTI dataset, respectively.

Bibliography

[1] Kulsum Akter Nisha et al. "A Comparative Analysis of Machine Learning Approaches in Personality Prediction Using MBTI". In: *Computational Intelligence in Pattern Recognition*. Springer, 2022, pp. 13–23.

[2] Valerie Priscilla Goby. "Personality and online/offline choices: MBTI profiles and favored communication modes in a Singapore study". In: *Cyberpsychology & behavior* 9.1 (2006), pp. 5–13.

[3] Sakdipat Ontoum and Jonathan H Chan. "Personality Type Based on Myers-Briggs Type Indicator with Text Posting Style by using Traditional and Deep Learning". In: *arXiv preprint arXiv:2201.08717* (2022).

[4] Alam Sher Khan et al. "Personality classification from online text using machine learning approach". In: *International Journal of Advanced Computer Science and Applications* 11.3 (2020).

[5] KN Pavan Kumar and Marina L Gavrilova. "Personality traits classification on twitter". In: *2019 16th ieee international conference on advanced video and signal based surveillance (avss)*. IEEE. 2019, pp. 1–8.

[6] Nur Haziqah Zainal Abidin et al. "Improving Intelligent Personality Prediction using Myers-Briggs Type Indicator and Random Forest Classifier". In: *International Journal of Advanced Computer Science and Applications* 11.11 (2020).

[7] Elma Kerz et al. "Pushing on Personality Detection from Verbal Behavior: A Transformer Meets Text Contours of Psycholinguistic Features". In: *arXiv preprint arXiv:2204.04629* (2022).

[8] James W Pennebaker and Laura A King. "Linguistic styles: language use as an individual difference." In: *Journal of personality and social psychology* 77.6 (1999), p. 1296.

[9] Yash Mehta et al. "Bottom-up and top-down: Predicting personality with psycholinguistic and language model features". In: *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2020, pp. 1184–1189.

[10] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[11] Zhenzhong Lan et al. "Albert: A lite bert for self-supervised learning of language representations". In: *arXiv preprint arXiv:1909.11942* (2019).

[12] Yinhan Liu et al. "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[13] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[14] Mohammad Hossein Amirhosseini and Hassan Kazemian. "Machine learning approach to personality type prediction based on the myers–briggs type indicator®". In: *Multimodal Technologies and Interaction* 4.1 (2020), p. 9.

[15] Zhancheng Ren et al. "A sentiment-aware deep learning approach for personality detection from text". In: *Information Processing & Management* 58.3 (2021), p. 102532.

[16] Hussain Ahmad et al. "A hybrid deep learning technique for personality trait classification from text". In: *IEEE Access* 9 (2021), pp. 146214–146232.

[17] Muskan Kuchhal et al. "Personality Prediction And Group Detection Using Social Media Posts". In: *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*. IEEE. 2022, pp. 1–5.

[18] Hans Christian et al. "Text based personality prediction from multiple social media data sources using pre-trained language model and model averaging". In: *Journal of Big Data* 8.1 (2021), pp. 1–20.

[19] Michael M Tadesse et al. "Personality predictions based on user behavior on the facebook social media platform". In: *IEEE Access* 6 (2018), pp. 61959–61969.

[20] Fatemeh Mohades Deilami, Hossein Sadr, and Mojdeh Nazari. "Using Machine Learning Based Models for Personality Recognition". In: *arXiv preprint arXiv:2201.06248* (2022).

[21] Mirza M Baig, Mian M Awais, and El-Sayed M El-Alfy. "AdaBoost-based artificial neural network learning". In: *Neurocomputing* 248 (2017), pp. 120–126.

[22] Nazila Taghvaei, Behrooz Masoumi, and Mohammad Reza Keyvanpour. "A Hybrid Framework for Personality Prediction based on Fuzzy Neural Networks and Deep Neural Networks". In: *Journal of AI and Data Mining* 9.3 (2021), pp. 283–294.

[23] Kamal El-Demerdash et al. "Deep learning based fusion strategies for personality prediction". In: *Egyptian Informatics Journal* (2021).

[24] Matthew E. Peters et al. "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://aclanthology.org/N18-1202.

[25] Jeremy Howard and Sebastian Ruder. "Universal language model fine-tuning for text classification". In: *arXiv preprint arXiv:1801.06146* (2018).

[26] Amirmohammad Kazameini et al. "Personality trait detection using bagged svm over bert word embedding ensembles". In: *arXiv preprint arXiv:2010.01309* (2020).

[27] Veronica Lynn, Niranjan Balasubramanian, and H Andrew Schwartz. "Hierarchical modeling for user personality prediction: The role of message-level attention". In: *Proceedings of the 58th annual meeting of the association for computational linguistics*. 2020, pp. 5306–5316.

[28] Bruno Fernandes et al. "An adjective selection personality assessment method using gradient boosting machine learning". In: *Processes* 8.5 (2020), p. 618.

[29] Madhura Jayaratne and Buddhi Jayatilleke. "Predicting personality using answers to open-ended interview questions". In: *IEEE Access* 8 (2020), pp. 115345–115355.

[30] In: (2011-2021). URL: https://www.javatpoint.com/machine-learning.

[31] In: (2011-2021). URL: https://www.javatpoint.com/machine-learning.

[32] Mitchell J. *(MBTI) myers-briggs personality type dataset*. Sept. 2017. URL: https://www.kaggle.com/datasets/datasnaek/mbti-type.

[33] Zeyad Khalid. *MBTI personality types 500 dataset*. Dec. 2021. URL: https://www.kaggle.com/datasets/zeyadkhalid/mbti-personality-types-500-dataset.

[34] Dylan Storey. *Myers Briggs Personality Tags on Reddit Data*. Version 0.0.1. Zenodo, July 2018. DOI: 10.5281/zenodo.1323873. URL: https://doi.org/10.5281/zenodo.1323873.

[35] Raj Kishan Cherukuru et al. "Prediction of Personality Trait using Machine Learning on Online Texts". In: *2022 International Conference for Advancement in Technology (ICONAT)*. IEEE. 2022, pp. 1–8.