  
BACS2003 ARTIFICIAL INTELLIGENCE

**202101 Session, the Year 2020/21**

**Assignment Documentation**

|  |  |
| --- | --- |
| **Full Name:** TEO WEN ZHI | |
| **Student ID:** 20WMD09522 | |
| **Programme:** Bachelor of Computer Science (Honours) in Software Engineering Year 2 | |
| **Tutorial Class:** 7 | |
| **Project Title:** Personality Prediction System | |
| **Module In-Charged:** Neural Network Model | |
| **Other team members’ data**   |  |  |  | | --- | --- | --- | | **No** | **Student Name** | **Module In Charge** | | **1** | Teo Xiao Hui | Logistic Regression Model | | **2** | Lim Shu Ting | Stochastic Gradient Descent | | |
| **Lecturer: Dr Lim Khai Yin** | **Tutor : Dr. Goh Ching Pang** |
| **Deadline: 11th April 2021 (Week 12, Sunday, turn in to Google Classroom before 11.59 pm)** | |

# **Table of Content**

[**Table of Content**](#_heading=h.30j0zll) **1**

[**Introduction**](#_heading=h.s1vjekocekp7) **2**

[Problem Background](#_heading=h.1fob9te) 2

[Objectives/Aims](#_heading=h.uf3kszhfui7t) 3

[Motivation](#_heading=h.2et92p0) 3

[Timeline/Milestone](#_heading=h.910audvzkvzd) 4

[**Research Background**](#_heading=h.co8xeqbpwe50) **5**

[Background of the applications](#_heading=h.4d34og8) 5

[Analysis of selected tool with any other relevant tools](#_heading=h.20jet9eovibm) 8

[Justify why the selected tool is suitable](#_heading=h.17dp8vu) 9

[**Methodology**](#_heading=h.y80kmi8tt2z7) **10**

[Description of dataset](#_heading=h.26in1rg) 10

[Applications of the algorithm(s)](#_heading=h.lnxbz9) 10

[System flowchart/activity diagram](#_heading=h.j9bhaii50wbp) 12

[Proposed test plan/hypothesis](#_heading=h.1ksv4uv) 13

[**Result**](#_heading=h.2mhvw319jej8) **14**

[Results](#_heading=h.2jxsxqh) 14

[Discussion/Interpretation](#_heading=h.gb2nhlng50tm) 21

[First Discussion: Suitable Model](#_heading=h.22sxs7eaquen) 21

[Second Discussion: The Factors Affect the Model Accuracy](#_heading=h.gwnssdeqd6m) 25

[**Discussion and Conclusion**](#_heading=h.406j3sb8kdr1) **26**

[Achievements](#_heading=h.1y810tw) 26

[Limitations and Future Works](#_heading=h.o9cmmjpvcflz) 27

[**Reference & Source**](#_heading=h.w7id95nxqmp8) **29**

# 

# **Introduction**

## Problem Background

The Myers Briggs Type Indicator (MBTI) is a personality type system that divides everyone into 16 distinct personality types across 4 axes, each axis will have 2 kinds of tendency as the table below:

|  |  |  |
| --- | --- | --- |
| Axis | Type | |
| **Favourite world** | Introversion (I) | Extroversion (E) |
| **Information** | Intuition (N) | Sensing (S) |
| **Decisions** | Thinking (T) | Feeling (F) |
| **Structure** | Judging (J) | Perceiving (P) |

The purpose of MBTI is not to separate everyone into different personalities, its purpose is to analyse the personality trend and preferences, and let the people have a better understanding of themself. It's just like we can write the word with both hands, but the preferred hand will write better.

To analyse the MBTI personality of a person usually will go through a set of questions, but the ability of the question set might not be enough to classify the MBTI personality of everyone accurately due to limited information gain.

In today’s world, most people are active on different social media platforms such as Instagram, Facebook and WeChat and prefer to post their status of daily life on the platform, it will be good information to increase the accuracy of MBTI by analysing the content in the posts. This will be a huge workload for us to identify and analyze the posts one by one. Therefore, this assignment will build and train a machine learning model with a suitable algorithm to make the computer have the ability to analyze the post’s content and classify them based on the MBTI personality type system.

Besides, this assignment will only do the analysis on word content in the post, which means it will not include the video clip, picture or music as consideration.

## Objectives/Aims

Find a suitable and effective machine learning algorithm to identify a user personality based on the MBTI personality type system through analysing their social media post’s content. Moreover, training, adjusting and building the higher accuracy machine learning model is also one of the objectives for this research.

## Motivation

Nowadays, most people are likely to use social media to express their emotion, preferences and life status. Social media posts already become a good resource for knowing the personality of a person.

**Application on personal aspects**

This system can as a tool to analyse their activities in social media for classifying them into different MBTI categories, if combined with the MBTI tests will achieve a higher accuracy MBTI prediction. These MBTI predictions can help them better understand themselves, give some suggestions on future development, and improve relationship management and communication in the organisation.

**Application on Industrial aspects**

Since the system can as the MBTI test assistant to increase the accuracy of the MBTI analysis result, it will help the industrial find suitable employees and help the human resources department organise the workgroup easier.

As able to analyse the MBTI personality of the user, social media will have a higher ability to promote and recommend the content that might be preferred by the user to improve the customer experience, increase the user engagement, let them become more addicted to social media.

## Timeline/Milestone

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Task Details | Start Date | End Date |
| Planing | * Research on a potential topic * Decide Topic * Discuss task distribution | 20-JAN-2021 | 18-FEB-2021 |
| Background studies | * Research on problem background * Search a suitable dataset * Search a suitable algorithm model | 19-FEB-2021 | 26-FEB-2021 |
| Develop | * Preprocessing on dataset * Build the model * Model Training | 27-FEB-2021 | 20-MAR-2021 |
| Testing | * Test model accuracy * Adjustment on model | 21-MAR-2021 | 31-MAR-2021 |
| Reporting | * Summaries the result * Prepare relevant documentation | 1-APR-2021 | 10-APR-2021 |
| Submission | | - | 11-APR-2021 |

## 

# 

# **Research Background**

## Background of the applications

**Machine Learning algorithms - Supervised Neural Network Model**

**Algorithm Used: Multi-layer Perceptron [MLP]**

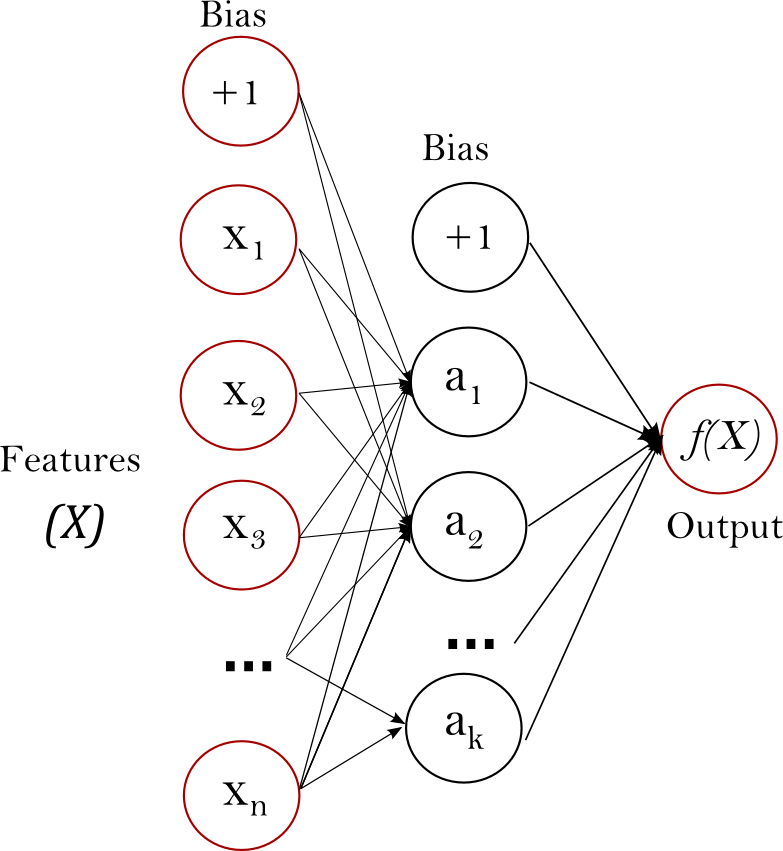


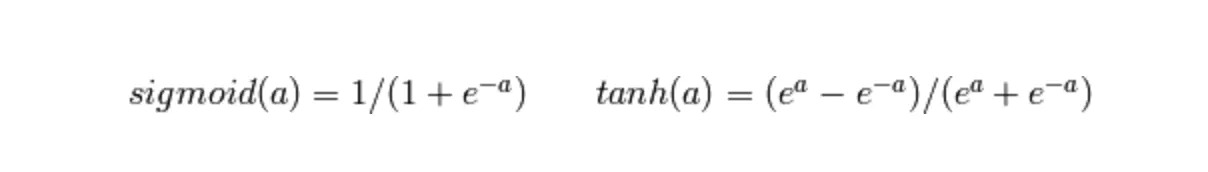
Figure 1: One hidden Layer MLP



Figure 2: Simple View of MLP

As Figure 1 and Figure 2 showed the MLP can divide into 3 parts which are input, output and the hidden layer in between.

The input layer represents the input features of MLP; the number of inputs is the number of neurons in the input layer.



The hidden layer is responsible for transform the value from input or a previously hidden layer with weighted linear summation() followed by a non-linear activation function() as sigmoid or hyperbolic tan function as.

The output layer is responsible for receiving the values from the last hidden layer in the algorithm and transforming them into output values.

The advantage of the MLP classifier is the capability to learn the non-linear models and it is also able to learn models in real-time or online learning by using the partial\_fit property. But the MLP classifier also has some limitations. Firstly, when using the MLP will need the user to tune several hyperparameters such as the number of hidden neurons, layers, and iterations, which will bring some difficulties for a beginner to find out the suitable setting. Moreover, the MLP with hidden layers has a non-convex loss function where there exists more than one local minimum. Therefore different random weight initializations might bring the difference in validation accuracy.

## Analysis of selected tool with any other relevant tools

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tools comparison** | **Remark** | **NLTK** | **Pandas** | **Scikit-learn** | **Joblib** |
| Type of license and open source license | State all types of license | Apache 2.0 | New BSD License | BSD License | BSD-licensed |
| Year founded | When is this tool being introduced? | 2001 | 2008 | 2007 | 2008 |
| Founding company | Owner | Team NLTK | AQR Capital Management | David Cournapeau | Joblib developers. |
| License Pricing | Compare the prices if the license is used for development and business/commercialization | Free | Free | Free | Free |
| Supported features | What features does it offer? | - Tokenize  - Inflexion and derivation  -Stemming and lemmatization  - Stopwords | - Analysis Data  - Series  - Dataframes  - Data mining  - Data clean  - Data visualisation | - Classification  - Regression  - Clustering  - Dimensionality reduction  - Model selection  - Preprocessing | - Transparent and fast disk-caching of output value  -Embarrassingly parallel helper  -Fast compressed Persistence |
| Common applications | In what areas this tool is usually used? | Natural language processing | Powerful tools to analyse the structured data | Machine Learning tools | lightweight pipelining |
| Customer support | How the customer support is given, e.g. proprietary, online community, etc. | <https://www.nltk.org/> | <https://pandas.pydata.org/community/blog/> | <https://scikit-learn.org/stable/support.html> | <https://joblib.readthedocs.io/en/latest/> |
| Limitations | The drawbacks of the software | difficult to learn and use  Slow, only splits text by sentences, without analyzing the semantic structure, no neural network models | low performance and long runtime that ultimately results in insufficient memory usage when dealing with large data analysis | It is friendly to beginners but it is not powerful enough for deep learning.  It also cannot use the GPU to accelerate the speed of processing. | The speed of saving and restore the learning model might slower than the Pickle |

## Justify why the selected tool is suitable

**NLTK**

Open source and provides powerful tools to complete text preprocessing tasks, It is also one of the popular natural language processing tools. I will try to use NLTK stopword to the text preprocessing

**Pandas**

A tool that is open-source and free to use for analysing and visualising the dataset. It can effectively convert the dataset into usable information, instead of writing thousands of lines of code manually.

**Scikit-learn (Sklearn)**

Since Sklearn is based on the BSD license it is an open-source and commercially usable machine learning tool. Moreover, many different simple and efficient tools for predictive data analysis are available, which means that they can support many different types of project. It also has good documentation and community and is a friendly platform for the beginner of machine learning. Since it is based on the python language it also shortens the learning time/curve of the beginner.

**Joblib**

Joblib is a set of tools to provide lightweight pipelining in Python and it is open source. As it provides transparent disk-caching of functions and lazy re-evaluation, it will be used for storing the trained model.

# **Methodology**

## Description of dataset

**(MBTI) Myers-Briggs Personality Type Dataset**

This is a public dataset released by Mitchell J on kaggle.com it contains over 8600 rows of data, on each row is a person’s:

**Type**: This person 4 letters MBTI code/type

**Posts**: Last 50 things they have posted, each entry separated by "|||" (3 pipe characters).

## Applications of the algorithm(s)

**Preprocessing Technique**

Since the **posts** in the dataset are from many different kinds of people, they might have different behaviour or terms used. To get a more accurate result on Natural Language Processing(NLP) it is compulsory to do some preprocessing to standardize the data.

* Remove word with the number
* Replace capital letters with lowercase
* Remove punctuation
* Remove Url

**Natural language processing (NLP)**

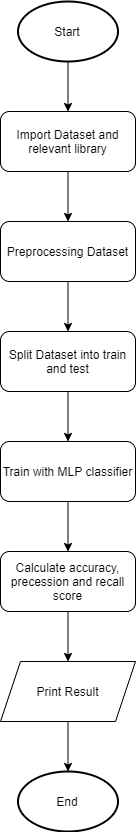
To Numerically encode the input data, as known as extracting the features from the data. This assignment is using the Term Frequency Inverse Document Frequency vectorizer (**TfidfVectorizer)** with English stop words to convert the data into a meaningful representation of numbers which is used to fit machine algorithms for prediction.

**MLP classifier**

* 1 input layer with 1 neuron (represent the posts by a specific user)
* 1-2 hidden layer
* 1 output layer with 16 neurons (represent 16 types of MBTI personality)

The MLP classifier used in this assignment is from the Sklearn supervised neural network model. The parameters settings that will be focused on are types of solver, types of activation, hidden layer design.

## System flowchart/activity diagram



## Proposed test plan/hypothesis

**Test 1: Choose the suitable solver for the MLP classifier model.**

Since the dataset has around 8000 records, official documentation suggested ‘adam’ solver for the large sample. Test on 3 different solvers in the SKlearn’s MLP classifier.

**Test 2: Determine the suitable hidden layer settings.**

Test on the different hidden layer settings to find out a result that is better than the default setting.

**Hypothesis 1: Same parameter settings apply to the ‘adam’ solver will give a better result.**

**Hypothesis 2: ‘Logistic’ activation setting will bring the best result.**

**Hypothesis 3: Built-in stopword in Sklearn and stopword in NLTK will not have much difference**

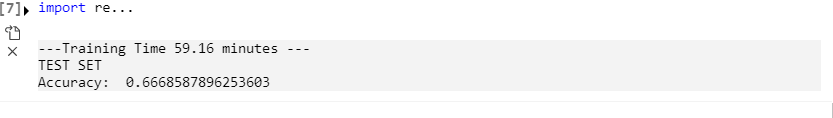
# **Result**

## Results

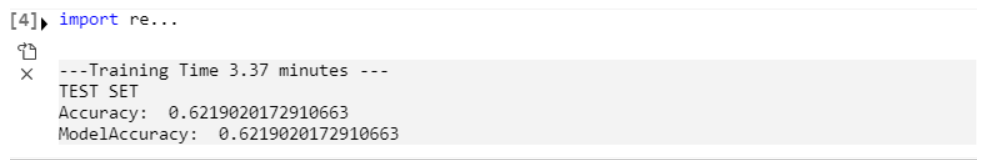
**Test 1 Result**

All the training is using the default setting but the activation set to ‘logistic’. Because the default activation setting will take more time.

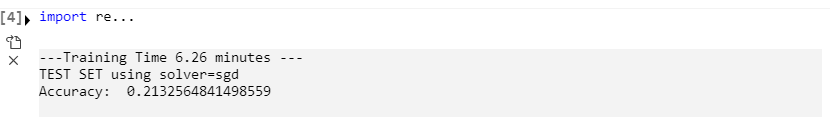
* Training time and accuracy result for ‘adam’ solver

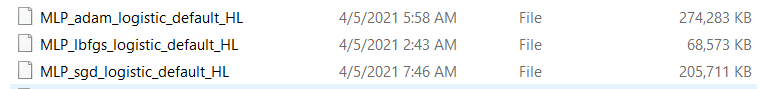
****

* Training time and accuracy result for ‘lbfgs’ solver



* Training time and accuracy results for ‘sgd’ solver

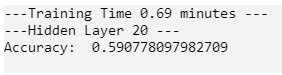


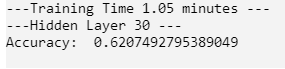


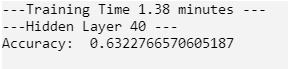
As the above picture shows, the ‘adam’ and ‘sgd’ model is taking a higher storage place.

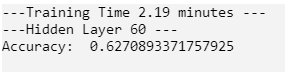
Due to time and space constraints, this assignment will use the ‘lbfgs’ solver as the fundamental to do the testing and adjustment on the parameters settings. In the end will apply the parameter’s setting with the ‘adam’ solver, to get a higher accuracy of the result.

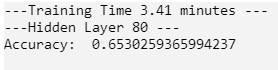
**Test 2 result- Single hidden layer**

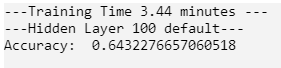
****

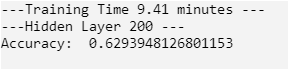
****

****

****

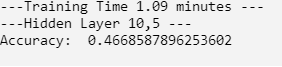
****

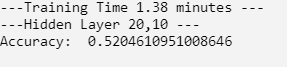
****

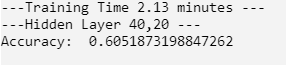
****

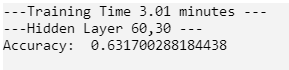
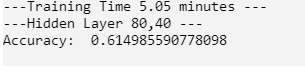
As the figures above, when the hidden layer’s neurons increase the time needed to complete the training will also increase. Although the differences of accuracy for different single hidden layers is small but set to 80 is the best performance in the balance of training time and accuracy requirement.

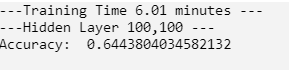
**Test 2 result- Two hidden layer**







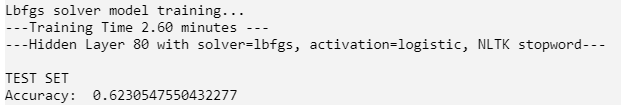


As the figures above, the 2 hidden layers will take more time compared with 1 hidden layer but this does not mean that they generate a better result than 1 hidden layer.

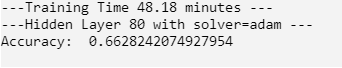
Through these testing, the best result is the single hidden layer with 80 neurons. It shows a better balance between time and accuracy. Since the best result of two hidden layers is still lower than the best result in the single hidden layer, but takes almost double the time to complete the training.

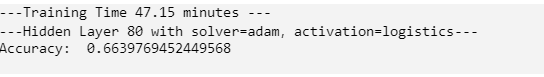
**Hypothesis 1 result**

**Lbfgs**

****

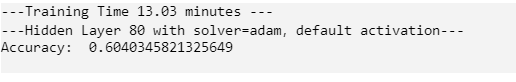
**Adam**

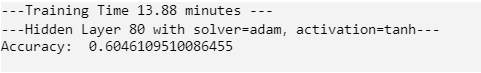
****

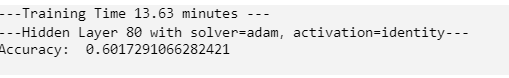


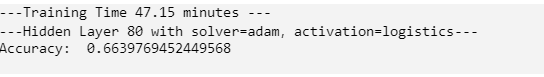
**Hypothesis 1 accepted.** Although the result display is a bit different, the settings are the same. In the ‘lbfgs’ solver, the result is 0.623, when using the ‘adam’ solver the accuracy is increased to 0.663 but the training time also has a lot of increase from 2.6 minutes to 47.15 minutes.

**Hypothesis 2 result**

****

****

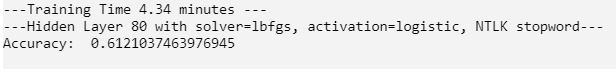
****

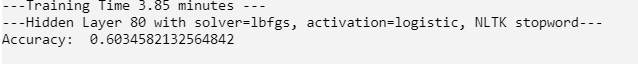


**Hypothesis 2 accepted.** As the results show that, in the same parameters setting the ‘logistics’ activation provides the best accuracy result, but it also takes the longest time to complete the training.

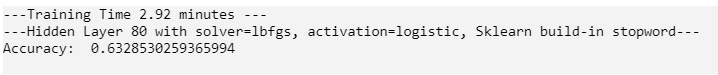
**Hypothesis 3**

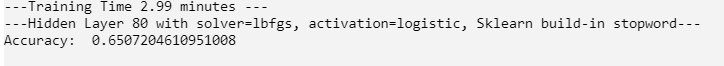
**NLTK stopword**





**TFIDF built-in stopword**





**Hypothesis 3 rejected.** Although the difference between both stopword method’s results is not very big, the TFIDF built-in stopword has better accuracy resulting in the same parameters setting compared with the NLTK stopword.

## Discussion/Interpretation

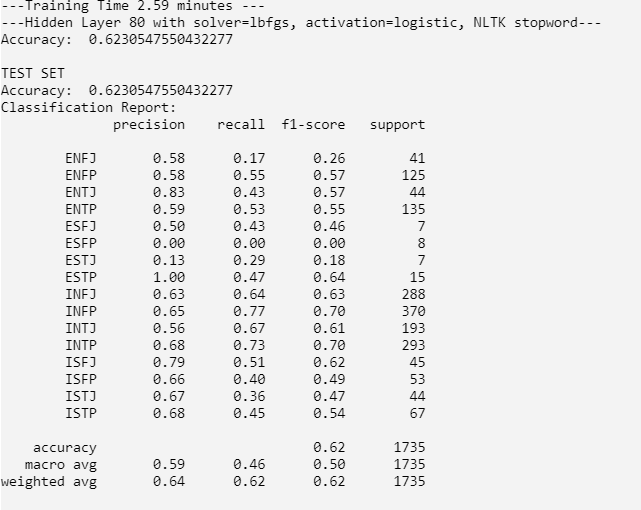
### First Discussion: Suitable Model

After the testing found the suitable MLP classifier parameter settings which are suitable for the MBTI personality prediction system can be divided into 2 types: **Balance and Better accuracy** both using the same hidden layer size, max iteration and the same activation method, only the solver using is the difference.

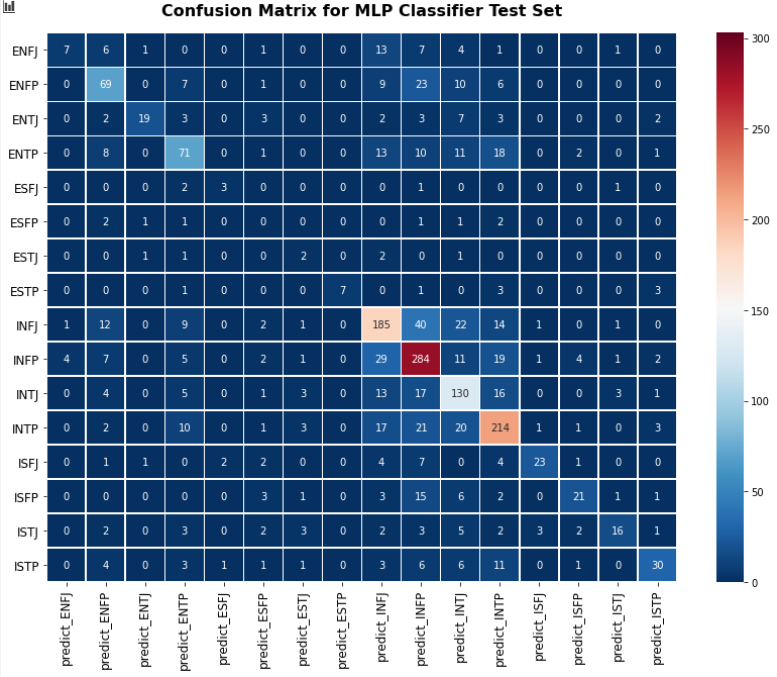
**Type 1: Balance type**

The first type of setting is achieving a better balance between accuracy, time and resource taken. The MLP Classifier setting shown below using.

****

The result for this setting is: 

Confusion Matrix for the Type 1

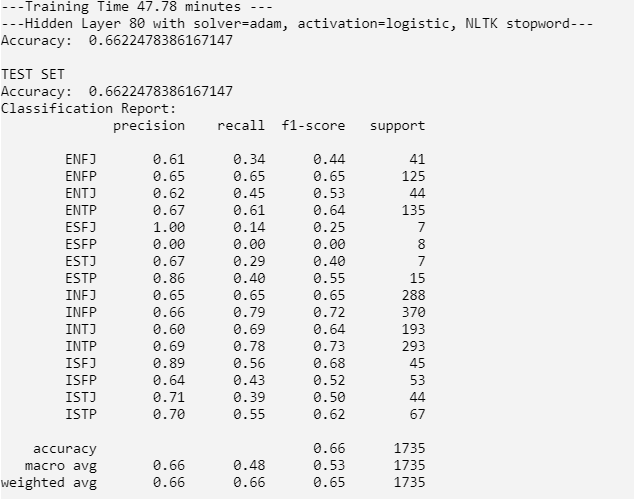


**Type 2: Better accuracy**

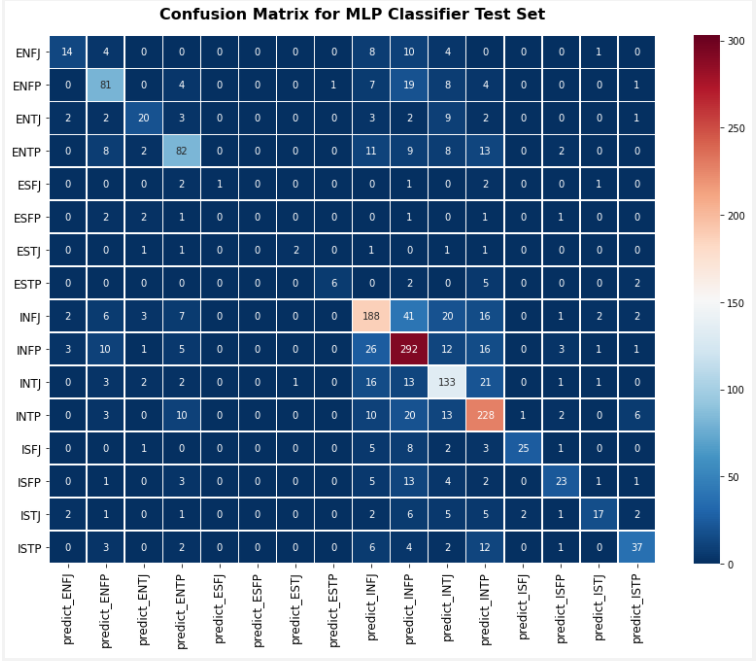
This is the second type of setting for the MLP classifier which is able to get a more accurate result, but the time and resources taken are increased very much compared to type 1.

****

The result for this setting is:



Confusion Matrix for the Type 2



### Second Discussion: The Factors Affect the Model Accuracy

**Hidden layer setting**

There are two types of hidden layer settings applied to this assignment which is **1 layer** and **2 layers**. In both hidden layers setting usually more neurons we set, the result of accuracy will become higher, but it has some limitations, it does not mean that the big number of neurons will bring a relatively good result. As the testing result shows, the increase of neurons does not bring a consistent increase in accuracy result.

**Activation setting**

I tried 4 types of activation settings **relu**(default),**tanh**,**identity** and **logistic**. The highest accuracy result is using **logistic** activation, but it also takes the longest time in training the model. It might be affected by the hidden layer setting and solver selection of the model.

**Stopword setting**

In testing results, the Sklearn stopword brings better performance than the NLTK stopword. So that the different feature extraction method from the posts will affect the result of the accuracy, a better stopword library will bring a more meaningful feature extraction that is able to let the machine identify the feature or term easier.

# **Discussion and Conclusion**

## Achievements

After completing the research, test and build the MLP classifier model to do the MBTI personality prediction. That occurs in two types of settings for the model and both settings are also able to achieve 60% accuracy and above. In the testing, the model that uses the ‘adam’ solver normally will have a higher accuracy result compared with other solvers, and the highest accuracy result can achieve 66%. Although this accuracy result cannot be considered as very well for an accurate MBTI prediction, if let this model is supportive of the MBTI test can provide extra references or different views to analyse the MBTI personality in a better way. Moreover, I also discussed with my teammates, the result we get cannot exceed 70% and above might be caused by the imbalanced data in the dataset. So that the model cannot clearly identify the rare MBTI personality, as we know some people don’t like to share their things with posts on social networks.

## 

## Limitations and Future Works

**Limitations**

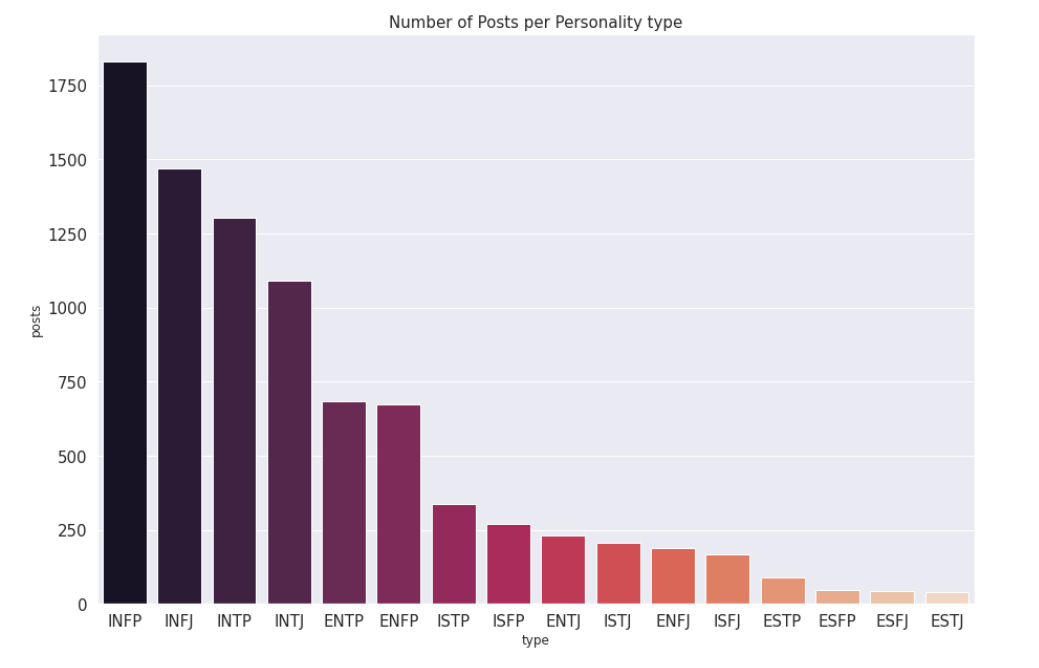
* **Insufficient processing power for deeper machine learning**

As **SKlearn** is not supported the GPU acceleration will highly depend on CPU processing power. The latest intel CPU(10th) already included the Deep Learning Boost that provided a huge improvement on AI calculation performance. But for my PC that will take a lot of time and resources to analyze machine learning, as it will take a lot of processing power that means that I also cannot do other high workload tasks when waiting for the result. Another solution for this limitation is using **TensorFlow** as that provides the GPU acceleration.

* **Insufficient knowledge for machine learning**

This course only provides a very brief overview of artificial intelligence. It is good enough for the student to start to build a simple model, but for the model’s result improvement is compulsory for a deeper understanding of artificial intelligence and deep learning. This limitation will become more obvious when the complexity of the dataset is higher.

* **Imbalance Dataset**

****

As the diagram above shows, some personality types only have very few posts, which might be one of the reasons for the lower accuracy result.

**Future Works**

This assignment let me clearly understand the difference between beginner and advanced in artificial intelligence(AI). As my FYP project is also based on supervised machine learning, I will do the extra study during my semester break, to cover the insufficient knowledge and have a better understanding of machine learning. To build a better quality and higher accuracy machine learning model.

The future study will include but not limited to:

* TensorFlow machine learning library.
* Neural network algorithm.
* Hidden-layer setting.
* The logic of the different algorithms.
* A deeper understanding of Python.

# 

# **Reference & Source**

Kaggle 2021, ‘*Myers-Briggs Personality Type Dataset*’, viewed 20 March 2021

<<https://www.kaggle.com/datasnaek/mbti-type>>.

Scikit-learn 2020,‘*1.17. Neural network models (supervised)*’, viewed 20 March 2021

<<https://scikit-learn.org/stable/modules/neural_networks_supervised.html>>.

Scikit-learn 2020,‘*sklearn.neural\_network.MLPClassifier*’, viewed 20 March 2021

<<https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html>>.

Kaggle 2021, ‘*MBTI\_Prediction*’, viewed 25 March 2021

<<https://www.kaggle.com/sarthak1799/mbti-prediction>>.

NLTK 2020, ‘*Natural Language Toolkit*’, viewed 26 March 2021

<<https://www.nltk.org/>>.

Scikit-learn 2021, ‘*scikit-learn Machine Learning in Python*’, viewed 26 March 2021

<<https://scikit-learn.org/stable/>>.

Kaggle 2021, ‘*Predicting MBTI Personality Types from Forum Posts*’, viewed 20 March 2021

<<https://www.kaggle.com/mosbyjnr/predicting-mbti-personality-types-from-forum-posts>>.

## 