ENGINEERING
W Booth School of Engineering
Practice and Technology

McMaster University

SEP 788 – Neural Networks and Development Tools

PERSONALITY PREDICTION

PROJECT REPORT- 2

Submitted by
Arshdeep Singh- ID 400328272
Kunal Garg- ID 400387739

Date
March 06th, 2022

Problem Statement:

With the increasing accessibility of the internet across the world, the use of social media is at an all-time high and is growing exponentially. According to Datareportal's article on 'Global Social Media Stats' (2022), "there are more than 4.62 billion social media users around the world in January 2022, equating to 58.4 percent of the total global population."
People respond differently to everyday situations based on their cognitive abilities and accordingly express their opinions on social media. In light of this, the huge amount of data available in the form of tweets, Instagram posts, Facebook posts, etc can be used to predict the personalities of individuals. Such information can be extremely useful, especially for the purposes of talent management. Recruiters and hiring managers can use these personality prediction models to gain insightful information about potential employees and hire the right talent for their companies. Moreover, the data obtained from personality prediction can also be used for the counselling of individuals, online marketing, personal recommendation systems, and bank credit scoring systems (Christian, 2021).
Through this project, we aim to identify the personality type of a person from a given social media post on the internet. We will use Myers–Briggs Type Indicator (MBTI) which is based on C. G. Jung's 'Theory of Psychological Type' and predicts personality types based on a person's preferences and decisions (Myers, 1998). It takes into account four dichotomies: introversion or extraversion, sensing or intuition, thinking or feeling, judging or perceiving, and hence, identifies 16 different personality types.

About the Dataset:

The dataset is taken from Kaggle (MBTI Myers-Briggs Personality type Dataset). It contains 8600 rows of data and each row contains a person's:

- Type (This person's 4 letters MBTI code/type)
- A section of each of the last 50 things they have posted (Each entry separated by "|||" (3 pipe characters))

Data Pre-Processing:

This is the key step in this project. Since this project contains text data, there are lots of modifications that must be made to the dataset. These include:

1. Removing the duplicate values (***using "drop.duplicates" functionality of DataFrames***.)

2. Expanding the contractions. For example, words such as "can't", "didn't", etc must be changed to "cannot", "did not", etc.(***By exploiting Regex functions***.)

3. Converting the data to lower case. This makes it easier during processing as the entire text is converted into an identical lower-case format.(***exploiting strings lower() functionality***)

4. The dataset contains a lot of URLs as well. They are unnecessary as they do not impart any meaning to the text. Hence, they are not useful for us.(***using Regex functions***)

5. Removal of punctuations, numbers, and stop words.(using Regex functions)(***Removing stopwords using "nltk.corpus import stopwords***")

6. Stemming and Lemmatization: Stemming just removes or stems the last few characters of a word, often leading to incorrect meanings and spelling. Lemmatization considers the context and converts the word to its meaningful base form, which is called Lemma. For example, changing the words such as running, playing to run, and play.(***using "nltk.stem import WordNetLemmatizer"***)

The snapshots shown below describe the transition of the first data point of the dataset after preprocessing. The length of the datapoint was reduced from 4660 characters to 2032 characters.

FROM:

```
In [11]: df["posts"][0]

Out[11]: "'http://www.youtube.com/watch?v=qsXHcwe3krw|||http://41.media.tumblr.com/tumblr_lfouy03PMA1qa1rooo1_500.jpg|||enfp and intj moments  https://www.youtube.com/watch?v=iz7lE1g4XM4  sportscenter not top ten plays  https://www.youtube.com/watch?v=uCdfze1etec pranks|||What has been the most life-changing experience in your life?|||http://www.youtube.com/watch?v=vXZeYwwRDw8  http://www.youtube.com/watch?v=u8ejam5DP3E  On repeat for most of today.|||May the PerC Experience immerse you.|||The last thing my INFJ friend posted on his facebook before committing suicide the next day. Rest in peace~  http://vimeo.com/22842206|||Hello ENFJ7. Sorry to hear of your distress. It's only natural for a relationship to not be perfection all the time in every moment of existence. Try to figure the hard times as times of growth, as...|||84389  84390  http://wallpaperpassion.com/upload/23700/friendship-boy-and-girl-wallpaper.jpg  http://assets.dornob.com/wp-content/uploads/2010/04/round-home-design.jpg ...|||Welcome and stuff.|||http://playeressence.com/wp-content/uploads/2013/08/RED-red-the-pokemon-master-32560474-450-338.jpg  Game. Set. Match.|||Prozac, wellbrutin, at least thirty minutes of moving your legs (and I do not mean moving them while sitting in your same desk chair), weed in moderation (maybe try edibles as a healthier alternative...|||Basically come up with three items you have determined that each type (or whichever types you want to do) would more than likely use, given each types' cognitive functions and whatnot, when left by...|||All things in moderation.  Sims is indeed a video game, and a good one at that. Note: a good one at that is somewhat subjective in that I am not completely promoting the death of any given Sim...|||Dear ENFP:  What were your favorite video games growing up and what are your now, current favorite video games? :cool:|||https://www.youtube.com/watch?v=QyPqT8umzmY|||It appears to be too late. :sad:|||There's someone out there for everyone.|||Wait... I thought confidence was a good thing.|||I just cherish the time of solitude b/c i revel within my inner world more whereas most other time i'd be workin... just enjoy the me time while you can. Don't worry, people will always be around to...|||Yo entp ladies... if you are into a complimentary personality,well, hey.|||... when your main social outlet is xbox live conversations and even then you verbally fatigue quickly.|||http://www.youtube.com/watch?v=gDhy7rdfm14  I really dig the part from 1:46 to 2:50|||http://www.youtube.com/watch?v=msqXffgh7b8|||Banned because this thread requires it of me.|||Get high in backyard, roast and eat marshmellows in backyard while conversing over something intellectual, followed by massages and kisses.|||http://www.youtube.com/watch?v=Mw7eoU3BMbE|||http://www.youtube.com/watch?v=4V2uYORhQOk|||http://www.youtube.com/watch?v=SlVmgFQQ0TI|||Banned for too many b's in that sentence. How could you! Think of the B!|||Banned for watching movies in the corner with the dunces.|||Banned because Health class clearly taught you nothing about peer pressure.|||Banned for a whole host of reasons!|||http://www.youtube.com/watch?v=IRcrv41hgz4|||1) Two baby deer on left and right munching on a beetle in the middle.  2) Using their own blood, two cavemen diary today's latest happenings on their designated cave diary wall.  3) I see it as...|||a pokemon world  an infj society  everyone becomes an optimist|||49142|||http://www.youtube.com/watch?v=ZRCEq_JFeFM|||http://discovermagazine.com/2012/jul-aug/20-things-you-didnt-know-about-deserts/desert.jpg|||http://oyster.ignimgs.com/mediawiki/apis.ign.com/pokemon-silver-version/d/dd/Ditto.gif|||http://www.serebii.net/potw-dp/Scizor.jpg|||Not all artists are artists because they draw. It's the idea that counts in forming something of your own... like a signature.|||Welcome to the robot ranks, person who downed my self-esteem cuz I am not an avid signature artist like herself. :proud:|||Banned for taking all the room under my bed. Ya gotta learn to share with the roaches.|||http://www.youtube.com/watch?v=w8IgImn57aQ|||Banned for being too much of a thundering, grumbling kind of storm... yep.|||Ahh... old high school music I have not heard in ages.  http://www.youtube.com/watch?v=dcCRUPCdB1w|||I failed a public speaking class a few years ago and I have sort of learned what I could do better were I to be in that position again. A big part of my failure was just overloading myself with too...|||I like this person's mentality. He's a confirmed INTJ by the way. http://www.youtube.com/watch?v=hGKLI-GEc6M|||Move to the Denver area and start a new life for myself.'"
```

TO:

```
In [23]: print(len(df["posts"][0]))
         df["posts"][0]

         2032

Out[23]: 'intj moment sportscenter top ten play prankswhat lifechanging experience life repeat todaymay perc experience immerse youthe last thing infj friend posted facebook committing suicide next day rest peace enfj sorry hear distress natural relationship perfection time every moment existence try figure hard time time growth welcome stuff game set matchprozac wellbrutin least thirty minute moving leg mean moving sitting desk chair weed moderation maybe try edible healthier alternativebasically come three item determined type whichever type want would likely use given type cognitive function whatnot left byall thing moderation sims indeed video game good one note good one somewhat subjective completely promoting death given simdear enfp favorite video game growing current favorite video game cool appears late sadtheres someone everyonewait thought confidence good thingi cherish time solitude bc revel within inner world whereas time id workin enjoy time dont worry people always around toyo entp lady complimentary personalitywell hey main social outlet xbox live conversation even verbally fatigue quickly really dig part thread requires meget high backyard roast eat marshmellows backyard conversing something intellectual followed massage kiss many b sentence could think bbanned watching movie corner duncesbanned health class clearly taught nothing peer pressurebanned whole host reason two baby deer left right munching beetle middle using blood two caveman diary today latest happening designated cave diary wall see asa pokemon world infj society everyone becomes optimist artist artist draw idea count forming something like signaturewelcome robot rank person downed selfesteem cuz avid signature artist like proudbanned taking room bed ya gotta learn share roach much thundering grumbling kind storm yepahh old high school music heard age failed public speaking class year ago sort learned could better position big part failure overloading tooi like person mentality he confirmed intj way denver area start new life'
```

Text Vectorization and Transformation

For performing Machine Learning on text, we need to transform our documents into Vector Representations such that we can apply Numeric Machine Learning. This process is called Feature Extraction or more simply, VECTORIZATION.

Each property of the Vector Representation is a FEATURE.

We must now make a critical shift in how we think about language—from a sequence of words to points that occupy a high-dimensional semantic space. Points in space can be close together or far apart, tightly clustered or evenly distributed. Semantic space is therefore mapped in such a way that documents with similar meanings are closer together and those that are different are farther apart. By encoding similarity as distance, we can begin to derive the primary components of documents and draw decision boundaries in our semantic space.

The simplest encoding of semantic space is the “bag-of-words” Model.

Four Types of Vector Encoding—

1. Frequency Vectors: This vector encoding model is used to simply fill in the vector with the frequency of each word as it appears in the document. In this encoding scheme, each document is represented as the multiset of the tokens that compose it and the value for each word position in the vector is its count.
2. One-Hot Encoding: This is a Boolean vector encoding method that marks a particular vector index with a value of true (1) if the token exists in the document and false (0) if it does not. In other words, each element of a one-hot encoded vector reflects either the presence or absence of the token
3. Term Frequency-Inverse Document Frequency: Term frequency represents the frequency by which a word appears in a document. Inverse Document Frequency applies to a set of documents and represents the importance of a word in the documents and is decided by how rare the word is present in the document. This type of encoding normalizes the frequency of tokens in a document with respect to the rest of the corpus.
4. Distributed Representations: When document similarity is important in the context of an application, we instead encode text along a continuous scale with a distributed representation, as shown in Figure 4-5. This means that the resulting document vector is not a simple mapping from token position to token score. Instead, the document is represented in a feature space that has been embedded to represent word similarity.

Mushtaq et al., has explained TF-IDF in detail as below:

The term tf−idf is a combination of two distinct terms, term frequency and inverse document frequency. Term frequency is the number that shows how often a particular word appears in a set of documents, and inverse document frequency is the value that tells us how common or rare a particular word is in a set of documents.
Tf−idf approach is a traditional approach mainly used in information retrieval and search engines. The tf−idf score is calculated using the eq:

$$tf\,idf(t,d,D) = tf(t,d).idf(t,D)$$

Where ‘D’ is the set of documents,
and ‘t’ is the term frequency of the document ‘d’.

Whereas term frequency is calculated by the eq.

$$tf(t,d) = \log(1 + freq(t,d)) \; and \; idf(t,D)$$
$$= log(N/count(d \in D : t \in d))$$

Converting our data frame to a tf−idf matrix would make it easier for our machine learning algorithm to grasp the word's real importance, and it would increase the efficiency of our algorithm. In the figure, you can see the number of times a word has appeared in the corpus and that word. After that, we applied principal component analysis so that the tf−idf
matrix can be viewed in a 2-dimensional space. It is applied when we are dealing with a big dataset.

Principal component analysis is a process to reduce dimensionality of a large dataset so that they can be visualized on a 2-dimensional canvas while preserving as much variability as possible. The shape of our tf-idf matrix is (8675,791). This means that our tf-idf matrix has 8675 dimensions and each dimension has 791 elements which makes it impossible to visualize this matrix on a 2-dimensional space.
In the above figure, you can see that data points are converged between 0.1 and 0.2. This again shows the dataset is uneven and mostly consists of similar words and occurred multiple times in the dataset.

Choosing the ML Approach:

Since this is a classification problem, we will choose classification algorithms. We will try out classic supervised machine learning algorithms such as SVM, Random Forest, XGBoost Classifier as well as deep learning method LSTM.

Reason for including a deep learning model for classification:

Currently, we are in the process of exploring different models to perform classification on the given dataset. We have used machine learning models such as XGBoost, SVM and Random Forest. Deep learning is a subset of machine learning and neural network models are way more complex. They are made of neurons in a layered architecture. They have memory and tend to learn features in an incremental manner.

In addition, this dataset consists of sequential data. Therefore, it makes sense to use RNN for capturing the information from the text and exploring the classification potential of neural network models on this dataset.

1. Rayne et al., used different RNN models on the MBTI dataset including a simple RNN, GRU, and LSTM. In their research, they found LSTM to give the best results.
2. Anthony et al., also tried to classify this dataset using RNNs and found LSTM to give the most accuracy of 37%.

Comparison of different ML/DL models:

Name of the model	Accuracy
Random Forest	37.12%
XGBoost	63.86%
SVM Model	56.89%
LSTM	Model could not be run due to memory limitations of the system and GPU

PROS/CONS:

1. **Random Forest:**
   a. **Pros:**
      i. Works well with non-linear data.
      ii. Lower risk of overfitting.
      iii. Runs efficiently on a large dataset
   b. **Cons:**
      i. Random forests are found to be biased while dealing with categorical variables.
      ii. Slow Training.
      iii. Not suitable for linear methods with a lot of sparse features.

2. **XGBoost:**
   a. **Pros:**
      i. It is Highly Flexible.
      ii. It uses the power of parallel processing.
      iii. It supports regularization.
      iv. It is designed to handle missing data with its in-build features.
   b. **Cons:**
      i. XGBoost does not perform so well on sparse and unstructured data.
      ii. The overall method is hardly scalable.

3. **SVM Model:**
   a. **Pros:**
      i. It works well with a clear margin of separation.
      ii. It is effective in high dimensional spaces.
      iii. It is effective in cases where the number of dimensions is greater than the number of samples.
      iv. It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
   b. **Cons:**
      i. It doesn't perform well when we have large data set because the required training time is higher.
      ii. It also doesn't perform very well, when the data set has more noise i.e., target classes are overlapping.

1. The XGBoost classifier gave the best accuracy of 63.86% followed by SVM which gave an accuracy of 56.89%.

1. Applying Random Forest

```
accuracies = {}

#Random Forest
random_forest = RandomForestClassifier(n_estimators=100, random_state = 1)
random_forest.fit(x_train, y_train)

# make predictions for test data
Y_pred = random_forest.predict(x_test)
predictions = [round(value) for value in Y_pred]

# evaluate predictions
accuracy = accuracy_score(y_test, predictions)
accuracies['Random Forest'] = accuracy* 100.0
print("Accuracy: %.2f%%" % (accuracy * 100.0))
```

```
Accuracy: 37.12%
```

2. Applying XGBClassifier

```
#XG boost Classifier
xgb = XGBClassifier()
xgb.fit(x_train,y_train)

Y_pred = xgb.predict(x_test)
predictions = [round(value) for value in Y_pred]

# evaluate predictions
accuracy = accuracy_score(y_test, predictions)
accuracies['XG Boost'] = accuracy* 100.0
print("Accuracy: %.2f%%" % (accuracy * 100.0))
```

```
D:\Arshdeep\Python\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[19:52:41] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
Accuracy: 63.86%
```

3. Applying SVM Model

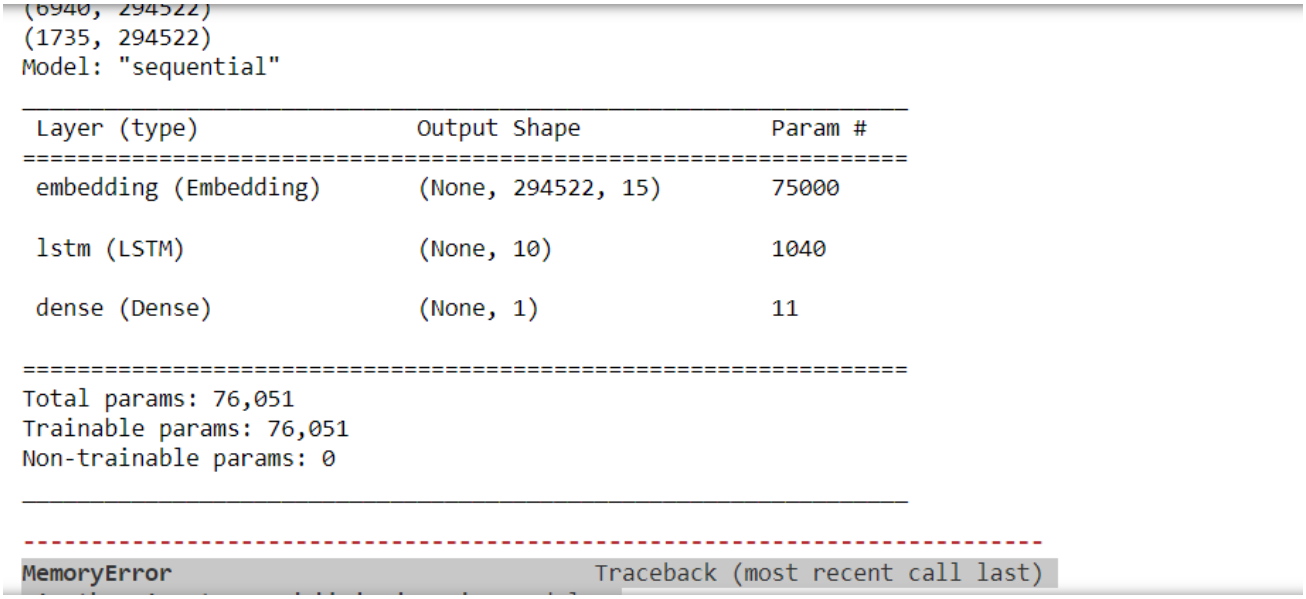
```
from sklearn.svm import SVC
svm = SVC(random_state = 1)
svm.fit(x_train, y_train)

Y_pred = svm.predict(x_test)

predictions = [round(value) for value in Y_pred]
# evaluate predictions
accuracy = accuracy_score(y_test, predictions)
accuracies['SVM'] = accuracy* 100.0
print("Accuracy: %.2f%%" % (accuracy * 100.0))
```

```
Accuracy: 56.89%
```

2. For LSTM, the model execution started but it could not be completed due to heavy memory requirements on the system (both google colab and personal computer)



```
(6940, 294522)
(1735, 294522)
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 294522, 15)        75000

 lstm (LSTM)                 (None, 10)                1040

 dense (Dense)               (None, 1)                 11

=================================================================
Total params: 76,051
Trainable params: 76,051
Non-trainable params: 0
_________________________________________________________________

---------------------------------------------------------------------------
MemoryError                               Traceback (most recent call last)
```

References:

1. https://datareportal.com/social-media-users#:~:text=Kepios%20analysis%20shows%20that%20there,of%20the%20total%20global%20population.
2. Christian, H., Suhartono, D., Chowanda, A. *et al.* Text based personality prediction from multiple social media data sources using pre-trained language model and model averaging. *J Big Data* 8, 68 (2021). https://doi.org/10.1186/s40537-021-00459-1
3. Myers, I. B., McCaulley, M. H., Quenk, N. L., & Hammer, A. L. (1998). The MBTI® Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator. Palo Alto: Consulting Psychologists Press.
   Accessed from:
   https://www.tolarisd.org/cms/lib3/TX01000982/Centricity/Domain/27/Myers%20Briggs%20Personality%20Test%20Manual.pdf
4. Alam F, Stepanov EA, Riccardi G. Personality traits recognition on social network—Facebook. AAAI Workshop—Technical Report, WS-13-01, 2013.
5. Songqiao Han, Hailiang Huang, Yuqing Tang,
   Knowledge of words: An interpretable approach for personality recognition from social media, Knowledge-Based Systems, Volume 194, 2020, 105550, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2020.105550.
   (https://www.sciencedirect.com/science/article/pii/S0950705120300459)
6. Cui, Brandon and Calvin Qi. “Survey Analysis of Machine Learning Methods for Natural Language Processing for MBTI Personality Type Prediction.” (2017).
   https://www.semanticscholar.org/paper/Survey-Analysis-of-Machine-Learning-Methods-for-for-Cui-Qi/08a3043e30ff342f9a92b438646e05d3eeeef6f4
7. https://saejournal.com/wp-content/uploads/2021/07/Personality-Prediction-Using-Machine-Learning.pdf
8. https://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/
9. Hernandez, R.; Knight, I.S. Predicting Myers-Bridge Type Indicator with text classification. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017. Available online:

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6839354.pdf (accessed on 9 September 2018)
https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6839354.pdf
10. Ma, Anthony G. Brown. “Neural Networks in Predicting Myers Brigg Personality Type From Writing Style.” (2017).
https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2736946.pdf
11. Z. Mushtaq, S. Ashraf and N. Sabahat, "Predicting MBTI Personality type with K-means Clustering and Gradient Boosting," 2020 IEEE 23rd International Multitopic Conference (INMIC), 2020, pp. 1-5, doi: 10.1109/INMIC50486.2020.9318078.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9318078&tag=1