

PERSONALITY PREDICTION PROJECT

A PROJECT REPORT

SUBMITTED BY:

ROSHIN JOHN (1800911)

SANDEEP KUMAR (1800913)

TARANDEEP SINGH (1800926)

TARANPREET KAUR (1800927)

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

at



BABA BANDA SINGH BAHADUR ENGINEERING COLLEGE

FATEHGARH SAHIB,PUNJAB(INDIA)-140406

AFFILATED TO I.K.G PUNJAB TECHNICAL UNIVERSITY,

KAPURTHALA,PUNJAB(INDIA)

JUNE (2021)

Table of Contents

1)INTRODUCTION	3
1.1) Problem Definition	
1.2) Project Overview/Specification	
1.3) Hardware Specification	
1.4) Software Specification	
2)LITERATURE SURVEY	5
2.1) Existing System	
2.2) Proposed System	
2.3) Feasibility Study	
3)SYSTEM ANALYSIS & DESIGN	6
3.1) Requirement Specification	
3.2) Flowcharts / DFDs /ERDs	
3.3) Design and Test Steps /Criteria	
3.4) Algorithms	
3.5) Testing Process	
4)RESULT / OUTPUTS	7
5)CONCLUSIONS	8
6)REFERENCES	9

INTRODUCTION-

1.1) Problem Definition

- i) Often co-workers face problems due to personality difference.**
- ii) Until now, many jobs needed psychometric tests.**
- iii) Moreover, personality is relevant to several interactions.**
- iv) Personality assessment helps us to get a broader view of predicting job satisfaction and lifestyle.**

1.2) Project Overview / Specifications

The project is based on identifying the personality of an individual using machine learning algorithms and big 5 models. The personality of a human plays a major role in his personal and professional life. Nowadays, many organizations have also started shortlisting the candidates based on their personality as this increases the efficiency of the work because the person is working on what he is good at than what he is forced to do.

- i) We predict the personality on the basis of the data collected from various social media platforms of the user which may be very beneficial in many aspects.**
- ii) Fetch data from the various social media accounts.**
- iii) Each post is treated as raw data and applies a learning model to predict the personality.**

1.3) Hardware Specification

- Operating system that you prefer (it could be Linux, Mac or Windows)
- A PC or laptop that you own
- Your PC or laptop should have enough RAM to run the calculations uninterruptedly.
- Python 3 installed in your system.

1.4) Software Specification

i) MACHINE LEARNING

Machine learning (ML) is a type of artificial intelligence that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: ***The ability to learn***. Machine learning is actively being used today, perhaps in many more places than one would expect.

Research on personality type prediction from textual data is scarce. However, important steps have been taken in this endeavour through machine learning. Classic machine learning techniques and neural networks have been used successfully for predicting MBTI personality types.

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

ii) SUPERVISED LEARNING

Supervised learning as the name indicates the presence of a supervisor as a teacher. Basically supervised learning is a learning in which we teach or train the machine using data which is well labelled that means some data is already tagged with the correct answer.

After that, the machine is provided with a new set of example(data) so that supervised learning algorithm analyses the training data and produces a correct outcome from labelled data.

2) LITERATURE SURVEY-

1. Allport's Trait model:

Gordon Allport set out one of the earliest personality models, which groups personality traits into three categories, cardinal traits, which shape the person, his/her attitudes, and his/her behaviors; central traits, which are the factors that determine most of individual behavior; and secondary traits, which may only be revealed in certain situations.

2. Cattell's 16 personality factor model:

Cattell's model includes 16 essential personality factors, which are listed under five major categories. This model had a profound influence on the development of the later big five model.

3. Eysenck's Giant Three model:

This model was originally known as the PEN model before psychoticism was added to his original two traits of extraversion and neuroticism to form the Giant Three.

4. The Myers-Briggs Type Indicator(MBTI) model:

This model covers four personality dimensions:

- Extroversion(E) vs Introversion (I)
- Sensing(S) vs intuition (N)
- Thinking(T) vs Feeling(F)
- Judging(J) vs Perceiving(P)

5. Maite et. al. [7] focused on Personality prediction from the Author Profiling task. They used PAN-AP-2015 corpus that was collected for social media users from twitter. Four languages were included but this paper focused on English language only. Self- online test was taken, and score was given between -0.5 to 0.5. Big Five model was used for traits. Then Glove representation in vector form was used for word embedding. For short

input data, the padding of many zero was done to as CNN require fixed amount of input. Different filters were used for Convolution layers. All the outputs were merged together, and the pooling layer was applied. Re LU is used as activation function. Fully connected neural network gives output as 5 neurons one for each stage. Deeper CNN can be implemented.

The authors in this paper [8] aim to predict the personality of twitter users for Arabic users in Egypt. They collected the data using Ara Personality. This data set was collected from Arabic dialect twitter user. Questionnaire consisting of several MCQs having 5 choices were translated to Arabic language and then filled by the users. And scores were assigned to each choice chosen by the user on the basis whether the question is Proportional or inversely proportional to the Big Five Personality Traits abbreviated as OCEAN. Apart from questionnaire their feeds were also collected. These Collected users feed then were pre-processed and cleaned by removing noisy data like user names, emails etc. and some non-Arabic words were converted to Arabic. Normalization was done to keep all the words in one form. The data is then divided into Train and test data. TF-IDF was calculated for every user. Three Supervised Machine learning as algorithms namely Decision trees, Support vector Machine and Multinomial Nave Bayes was used.

M. Hassanein et. al. [9] presented an approach to predict the personality on basis of semantics. They used big five model on My Personality Data-set. Vector Space model is used to represent the user text in the vector from that hold counts of every word in the text. Similarity measure is used to measure semantics using WordNet Database.

The Authors of the paper [10] proposed the model for text analysis and predict the personality of brands on Social Media Platform. Big Five model was used to predict the brand personality. This information could help brand to plan its Marketing Strategies as well as Improve relations with the Customers. My Personality data-set was used as well as the one was created for Brands pages and features were extracted from both these data-sets. Feature selection was done by done approaches namely Pearson Correlation and other was Gradient Boosting on 3 different Machine learning approaches as Support Vector Regression (SVR), Gradient Boosting and Feed-Forward Neural Network. XGB models perform best and predict personality.

6. CLSTM that is a bidirectional Long Short-Term Memory network interconnected with CNN to find personality of users.

It focused on structure of text as it can be important feature. Big Five model with 5 traits was used. Two data- sets were used for the experiment. One is long text data- set of essay data-set of 2467 essays tagged with their authors traits and another is Short text of YouTube vloggers. GloVe algorithm was used for word embedding. LSTM is used which has a self-loop and RNN loop as well, it is bidirectional so as to extract more features.

Paper also proposes the concept of Latent sentence Groups (LSG) that means several sentences that are closely related to each other. CNN was used for studying such latent features. Max pooling layer is used after LSTM to get sentence vectors. Soft max classifier was used as the classifier. Various contrast models were used like TF-IDF bayes, 2 and 3 dimensions CNN, one LSTM to compare the results with proposed model, which proved to perform better.

The authors of this paper [12] presented a system that could analyze the personality traits for Facebook users by using their status posts. Big Five personality model was used. They used My Personality data-set that had 250 users and about 10,000 posts updates from these users. These posts after extraction were pre-processed by removing links, symbols etc. All the words were converted in their lowercase. A spelling correction algorithm was used for real time data to correct all the incorrect spellings in the post. Posts also consisted of symbols like Hashtags (#) and emotions, these were removed by keeping the words as it is. TF-IDF was calculated to extract keywords from documents, thus feature vector was formed. This vector was too large so to reduce the size and to get only relevant features, Principal Component Analysis was used.

Machine learning algorithms KNN and SVM were used. KNN was best for Classification of traits.

Previous study on personality prediction has been done by using social media Facebook and some features such as LIWC features, SNA features, time-related features. Their research is very similar with ours especially for the dataset (250

dataset from my Personality) and the features (LIWC and SNA features). Another research in personality prediction based on Facebook status were done by using two approaches such as open vocabulary DLA (Differential Language Analysis) and LIWC features. By using Facebook, a research defining features with bag-of-words and token (unigrams) approaches were conducted as well. Other study was done to make a personality prediction system by using Twitter with LIWC and MRC as featured . All mentioned above researches did personality prediction by using social media in English based on Big Five Personality models. Recent research was conducted to make a personality prediction system using Twitter in Bahasa based on Big Five Personality models. Other research on personality prediction was done using deep learning technique to classify Big Five Personality models from social media .

3) System analysis & design

3.1) Requirement Specification

- i) Dataset
- ii) python 3
- iii) jupyter notebook
- iv) machine learning
- v) supervised learning

3.2) Flowcharts

3.3) Design and Test Steps

- Step 1) Data Gathering
- Step 2) Import Libraries
- Step 3) Import Dataset
- Step 4) Exploratory Data Analysis
- Step 5) Pre-Processing of Dataset
- Step 6) Feature Engineering
- Step 7) Training & Evaluating 60-40 split
- Step 8) Accuracy & comparison of algorithms
- Step 9) Training & Evaluating 70-3- split
- Step 10) Four Classifiers across MBTI Axis
- Step 11) Features correlation analysis
- Step 12) Lemmatization in pre-processing
- Step 13) Feature Engineering include Tf-IDF
- Step 14) Model Testing

3.4) Algorithms

- 1) Random Forrest

The random forest is a classification algorithm consisting of many decision trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

Random forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because of its simplicity and diversity (it can be used for both classification and regression tasks).

2) XG Boost

XG Boost stands for “Extreme Gradient Boosting”. ... It implements Machine Learning algorithms under the Gradient Boosting framework. It provides a parallel tree boosting to solve many data science problems in a fast and accurate way.

XG Boost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

3) Gradient Descent

Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. In machine learning, we use gradient descent to update the parameters of our model.

4) Logistic Regression

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. ... It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

5) KNN

K-Nearest Neighbours (KNN) is one of the simplest algorithms used in Machine Learning for regression and classification problem. KNN algorithms use data and classify new data points based on similarity measures (e.g. distance function). Classification is done by a majority vote to its neighbours.

Breaking it Down – Pseudo Code of KNN

1. Calculate the distance between test data and each row of training data. ...
2. Sort the calculated distances in ascending order based on distance values.
3. Get top k rows from the sorted array.
4. Get the most frequent class of these rows.
5. Return the predicted class

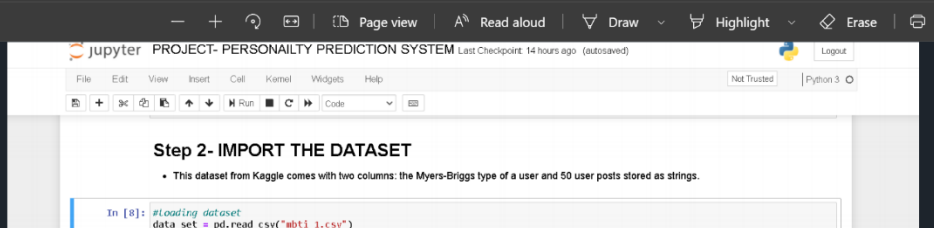
6) SVM

SVM is a supervised **machine learning algorithm** which **can be used** for classification or **regression** problems. It **uses** a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs.

SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and **work** well for many practical problems. The idea of **SVM** is simple: The **algorithm** creates a line or a hyperplane which separates the data into classes.

3.5) Testing Process (test cases to be included)

4) Results / outputs



PERSONALITY PREDICTION SYS

File | C:/Users/sandy/Desktop/report%20of%20project/PERSONALITY%20PREDICTION%20SYSTEM%20(4).pdf

36 of 80

Jupyter PROJECT- PERSONALITY PREDICTION SYSTEM Last Checkpoint: 14 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

Not Trusted | Python 3

Step 2- IMPORT THE DATASET

- This dataset from Kaggle comes with two columns: the Myers-Briggs type of a user and 50 user posts stored as strings.

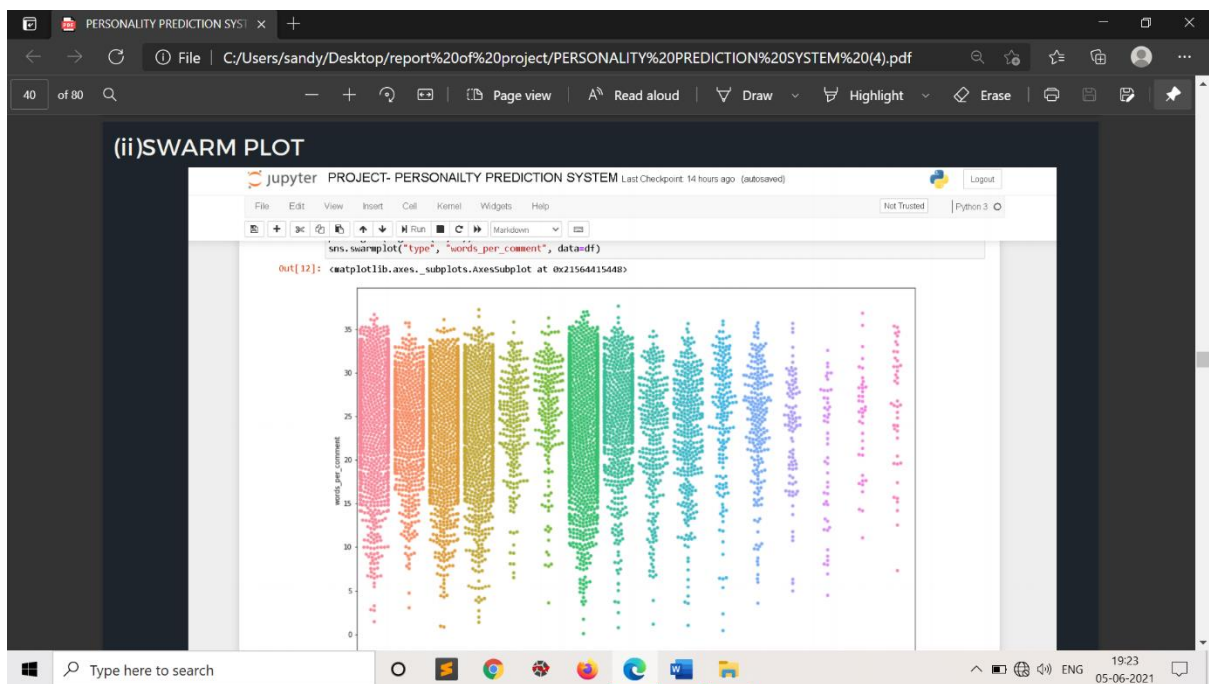
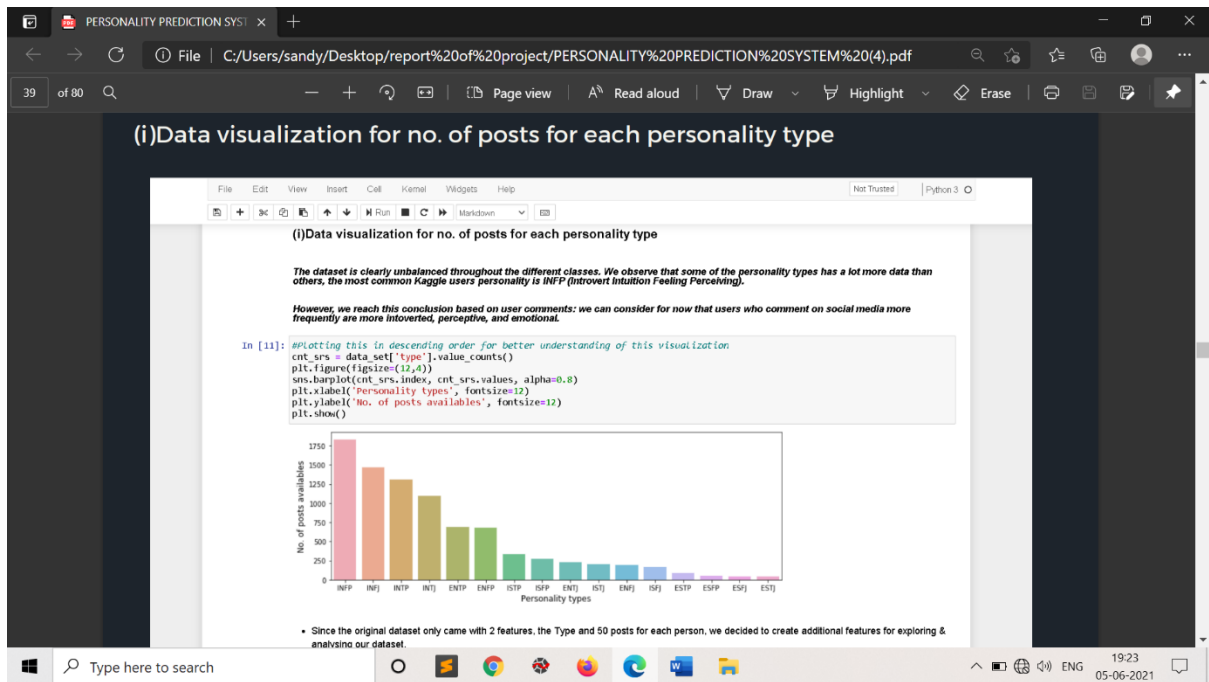
```
In [8]: #loading dataset
data_set = pd.read_csv("mbti_1.csv")

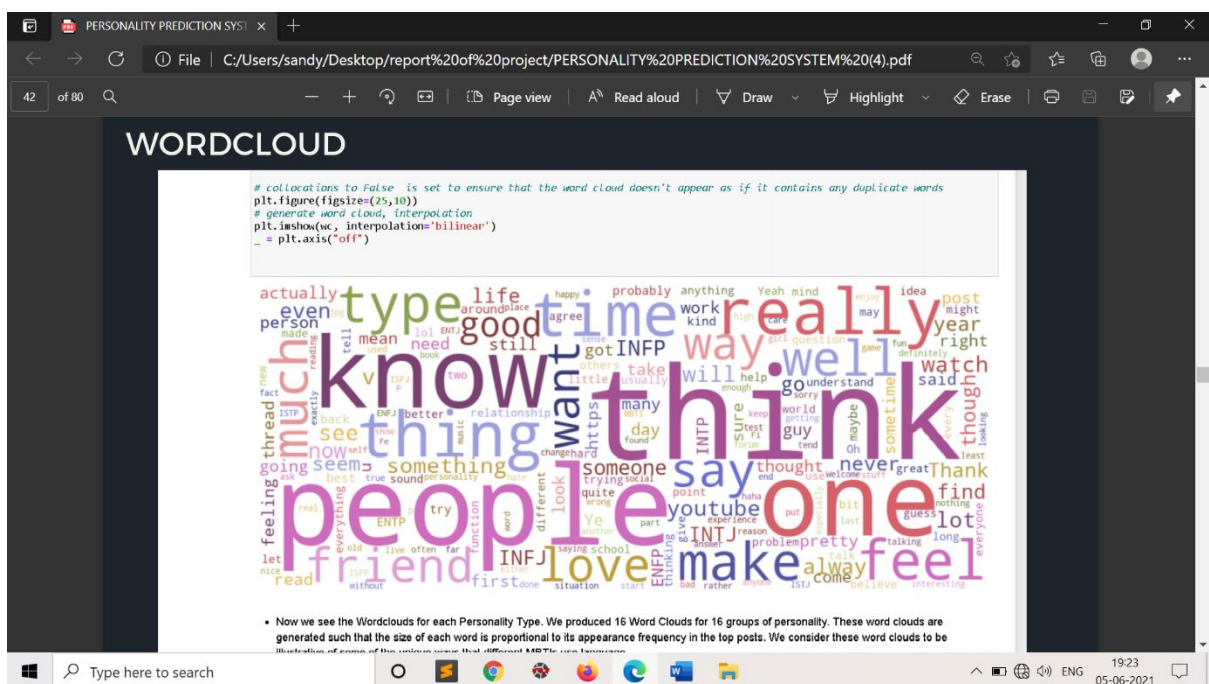
print(data_set.head(10))
print("---*40")
print(data_set.info())
```

```

              type                                posts
0  INTJ      'http://www.youtube.com/watch?v=q50Hcwe3krw|...
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  ENTP      'You're fired.||||that's another silly misconc...
5  INTJ      '18/37 @.g|||science is not perfect. No scien...
6  INTP      'No, I can't draw on my own nails (baha), those...
7  INTJ      'I tend to build up a collection of things on ...
8  INTP      'I'm not sure, that's a good question. The dist...
9  INTP      'https://www.youtube.com/watch?v=8-egjey@q5l|...
.....
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8675 entries, 0 to 8674
Data columns (total 2 columns):
type      8675 non-null object
posts     8675 non-null object
dtypes: object(2)
memory usage: 135.74 KB
None

```





The screenshot shows a presentation slide with a dark background and a red and blue abstract shape on the left. The slide content is as follows:

1. Remove links
2. Keep the End Of Sentence characters
3. Strip Punctuation
4. Remove multiple full stops
5. Remove Non-words
6. Convert posts to lowercase
7. Remove multiple letters repeating words
8. Remove very short or long words
9. Remove MBTI Personality Words

Below the list, there is a Python code snippet:

```

# Remove multiple letter repeating words
def "posts" = df["posts"].apply(lambda x: re.sub("([a-zA-Z])\1+", "\1", x))

# Remove very short or long words
def "posts" = df["posts"].apply(lambda x: re.sub("([a-zA-Z])\1+", "\1", x))

# Remove MBTI Personality words - crucial in order to get valid model accuracy evaluation for
if remove_special:
    pers_types = ["18BP", "18P", "18P", "18P", "18P", "18P", "18P", "18P", "18P", "18P"]
    pers_types = [p.lower() for p in pers_types]
    p = re.compile("(" + "|".join(pers_types) + ")")

    return df
# Preprocessing of entered text
new_df = preprocess_text(df)

In [17]: # Remove posts with less than 4 words
new_words = 25
print("Before: Number of posts", len(new_df))
new_df["no_of_words"] = new_df["posts"].apply(lambda x: len(re.findall("[a-zA-Z]", x)))
new_df = new_df[new_df["no_of_words"] >= new_words]

print("After: Number of posts", len(new_df))

Before: Number of posts 8675
After: Number of posts 856

```

The bottom of the image shows a Windows taskbar with various icons and a system clock displaying 19:25 on 05-06-2021.

PERSONALITY PREDICTION SYS1 x +

File | C:/Users/sandy/Desktop/report%20of%20project/PERSONALITY%20PREDICTION%20SYSTEM%20(4).pdf

49 of 80

(i) LabelEncoder :

Provided by sklearn library that converts the levels of categorical features (labels) into numeric form so as to convert it into the machine-readable form. It encode labels with a value between 0 and n_classes-1 where n is the number of distinct labels. If a label repeats it assigns the same value to it assigned earlier:

```
In [41]: # Converting MBTI personality (or target or X feature) into numerical form using Label Encoding
# encoding personality type
enc = LabelEncoder()
new_dff['type of encoding'] = enc.fit_transform(new_dff['type'])
target = new_dff['type of encoding']

In [42]: new_dff.head(15)
Out[42]:
```

	type	posts	No. of posts	avg. length	avg. sentiment	avg. polarity	avg. subjectivity	avg. sentiment	avg. polarity	avg. subjectivity
0	INFJ	emotional, sensitive, intuitive, idealistic, and very strong	0	0	0.40	0.02	0.08	0.12	0.00	0.00
1	ENTP	great sense of humor, very strong, and very strong	0	0	0.20	0.08	0.08	0.02	0.00	0.00
2	INTP	great sense of humor, very strong, and very strong	0	0	0.10	0.08	0.04	0.08	0.00	0.00
3	INTJ	great sense of humor, very strong, and very strong	0	0	0.04	0.02	0.02	0.02	0.00	0.00
4	ENTJ	great sense of humor, very strong, and very strong	0	0	0.12	0.02	0.02	0.02	0.00	0.00

• We observe that almost all of these were the most occurring words in our wordcloud above

(ii) CountVectorizer

- It is used to convert a collection of text documents to a vector of term/token counts and build a vocabulary of known words, documents using that vocabulary. It also enables the pre-processing of text data prior to generating the vector representation.
- Here, we use stop_words='english' with CountVectorizer since this just counts the occurrences of each word in its vocabulary. Words like 'the', 'and', etc. will become very important features while they add little meaning to the text. This is an important step in our model can often be improved if you don't take those words into account.

```
42: # Vectorizing the posts for the model and filtering stop-words
vect = CountVectorizer(stop_words='english')

# Converting posts (or training or X feature) into numerical form by count vectorization
train = vect.fit_transform(new_dff['posts'])

43: train.shape
Out[43]: (8466, 98555)
```

Type here to search

19:25 05-06-2021

PERSONALITY PREDICTION SYS1 x +

File | C:/Users/sandy/Desktop/report%20of%20project/PERSONALITY%20PREDICTION%20SYSTEM%20(4).pdf

52 of 80

COMPARING ALGORITHMS

Step 6(b)-Comparing Algorithms

```
In [46]: pd.DataFrame.from_dict(accuracies, orient='index', columns=['Accuracies(%)'])
Out[46]:
```

	Accuracies(%)
Random Forest	38.533510
XG Boost	57.888320
Gradient Descent	43.896982
Logistic Regression	58.193091
SVM	35.518158
KNN	18.445232

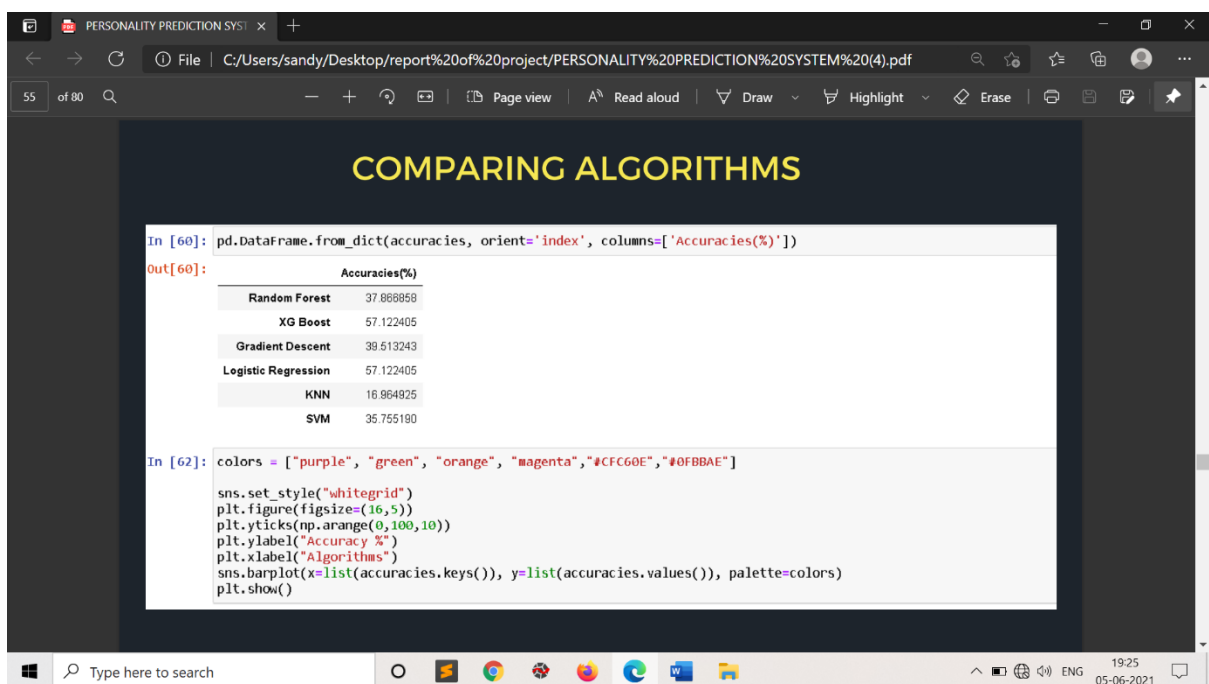
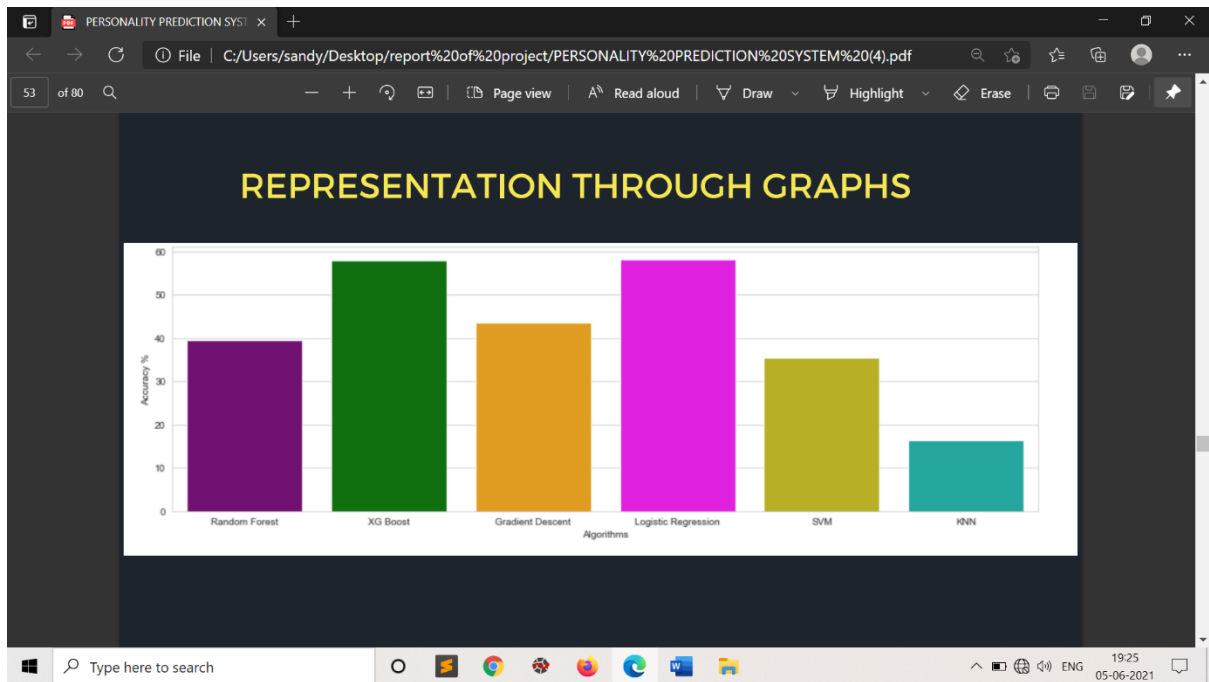
- We can clearly see that this model underfits our dataset when we apply split ratio of 60:40 on our dataset i.e. the model has not learned enough from the training data, resulting in low generalization and unreliable predictions. (almost all the results are near 50%, which is not good)

```
In [45]: colors = ["purple", "green", "orange", "magenta", "#CFC60E", "#0FBBAE"]

sns.set_style("whitegrid")
plt.figure(figsize=(16,5))
plt.xticks(np.arange(0,100,10))
plt.ylabel("Accuracy %")
plt.xlabel("Algorithms")
sns.barplot(x=list(accuracies.keys()), y=list(accuracies.values()), palette=colors)
plt.show()
```

Type here to search

19:25 05-06-2021



PERSONALITY PREDICTION SYS1 x +

File | C:/Users/sandy/Desktop/report%20of%20project/PERSONALITY%20PREDICTION%20SYSTEM%20(4).pdf

56 of 80

REPRESENTATION THROUGH GRAPHS

Algorithms	Accuracy %
Random Forest	~38
XG Boost	~55
Gradient Descent	~40
Logistic Regression	~55
KNN	~18
SVM	~35

- Inference : test_size=0.3 gives marginally better results for all algorithms
- As we can see the above ML classifiers perform at efficiency of nearly 50% only - which is pretty bad. So, instead of selecting all 16 types of personalities as a unique feature, we hence train 4 classifiers individually to classify their personalities based on MBTI type.

Type here to search

19:25 05-06-2021

PERSONALITY PREDICTION SYS1 x +

File | C:/Users/sandy/Desktop/report%20of%20project/PERSONALITY%20PREDICTION%20SYSTEM%20(4).pdf

59 of 80

```
Out[24]:
```

	type	posts	IE	NS	TF	JP
0	INFJ	'http://www.youtube.com/watch?v=qgXhCwe3kqv@...	1	1	0	1
1	ENTP	'I'm finding the lack of me in these posts ver...	0	1	1	0
2	INTP	'Good one _____ https://www.youtube.com/wat...	1	1	1	0
3	INTJ	'Dear INTP, I enjoyed our conversation the o...	1	1	1	1
4	ENTJ	'You're fired !!! That's another silly misconc...	0	1	1	1

- Using the above code, if a person has I, N, T and J, the value across the 4 axis of MBTI i.e. IE, NS, TF and JP respectively, will be 1. Else 0.
- This will help us calculate for e.g. how many Introvert posts are present v/s how many Extrovert posts are present, out of all the given entries in our labelled Kaggle dataset. This is done in order to explore the dataset for all the individual Personality Indices of MBTI

Counting No. of posts in one class / Total no. of posts in the other class

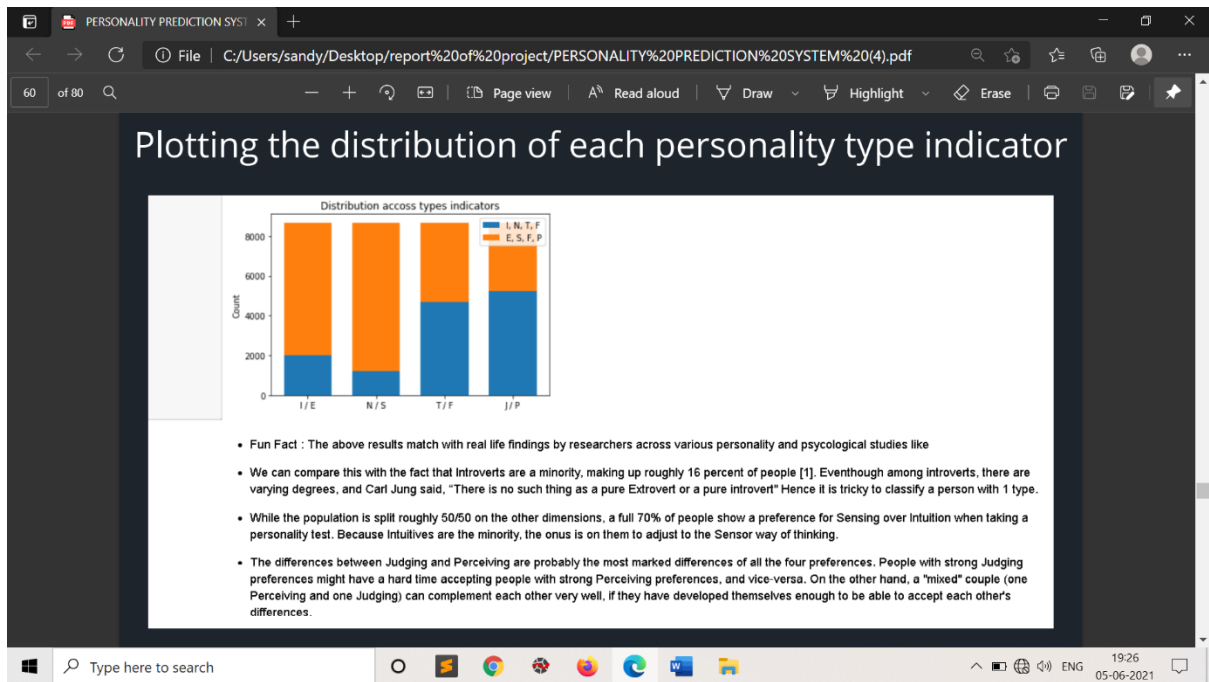
```
In [25]: print ("Introversion (I) / Extroversion (E):\t\t", data['IE'].value_counts()[0], " / ", data['IE'].value_counts()[1])
print ("Intuition (N) / Sensing (S):\t\t\t", data['NS'].value_counts()[0], " / ", data['NS'].value_counts()[1])
print ("Thinking (T) / Feeling (F):\t\t\t", data['TF'].value_counts()[0], " / ", data['TF'].value_counts()[1])
print ("Judging (J) / Perceiving (P):\t\t\t", data['JP'].value_counts()[0], " / ", data['JP'].value_counts()[1])
```

```
Introversion (I) / Extroversion (E): 1999 / 6676
Intuition (N) / Sensing (S): 1197 / 7478
Thinking (T) / Feeling (F): 4694 / 3981
Judging (J) / Perceiving (P): 5241 / 3434
```

- We infer that there is unequal distribution even among each of the 4 axis in the entries of our dataset. i.e. out of IE: E is the majority, in NS: S is the majority. While TF and JP have relatively less difference between them.

Type here to search

19:25 05-06-2021




```
0. 0.08968056 0. 0. 0.0860263 0.
0. 0. 0. 0.06318282 0. 0.
0. 0.04256832 0. 0. 0.
0.06642087 0. 0. 0.09201473 0.
0. 0.0831116 0. 0. 0.
0. 0. 0. 0. 0.
0. 0. 0. 0. 0.06971149
0.09554125 0.04625983 0.08531558 0.
0.06799661 0.07466644 0. 0.09843694
0. 0. 0. 0.06502346 0.
0. 0. 0. 0. 0.
0. 0.06869092 0. 0.
0. 0. 0. 0. 0.
0.08067849 0. 0. 0.
0. 0. 0. 0.
0. 0.0466968 0.0539756 0.08760887
0.1533845 0. 0. 0.09298479
0. 0. 0. 0.
0. 0. 0.01058077 0.
0. 0.11100899 0.13361762 0.
0.06046932 0. 0.08258902 0.2392193
0. 0.09217882 0. 0.04223906 0.
0.08665848 0.04111178 0. 0.16434695 0.04117693
0. 0. 0.07231244 0.
0. 0. 0. 0.
0.11467609 0.09387096 0. 0.
0. 0. 0.04833236 0.
0.
]

Let's see how the posts look in Binarized MBTI personality indicator representation: (we have taken 1st post for demonstration)

In [48]: print("for MBTI personality type : %s" % translate_back(list_personality[0,:]))
print("Y : Binarized MBTI 1st row: %s" % list_personality[0,:])

for MBTI personality type : INFJ
Y : Binarized MBTI 1st row: [0 0 0 0]

Therefore we have successfully converted the textual data into numerical form
```

PERSONALITY PREDICTION SYSTEM

File | C:/Users/sandy/Desktop/report%20of%20project/PERSONALITY%20PREDICTION%20SYSTEM%20(4).pdf

71 of 80

Page view | Read aloud | Draw | Highlight | Erase

Out of all the models, seen above we see that on an average XG Boost gives relatively good performance, hence we choose it to build our Personality prediction model. This will be beneficial as XGBoost model [2] can even be used to evaluate and report on the performance on a test set for the model during training.

```
In [ ]: # setup parameters for xgboost
param = {}

param['n_estimators'] = 200 #100
param['max_depth'] = 2 #3
param['nthread'] = 8 #1
param['learning_rate'] = 0.2 #0.1

# Individually training each mbti personality type
for l in range(len(personality_type)):
    y = list_personality[:,l]

    # split data into train and test sets
    seed = 7
    test_size = 0.33
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=seed)

    # fit model on training data
    model = XGBClassifier(**param)
    model.fit(X_train, y_train)

    # make predictions for test data
    y_pred = model.predict(X_test)
    predictions = [round(value) for value in y_pred]

    # evaluate predictions
    accuracy = accuracy_score(y_test, predictions)
    print("%s Accuracy: %.2f%%" % (personality_type[l], accuracy * 100.0))
```

We find that these accuracies are improved than before. Hence we fine tune the hyperparameters XG boost and then train the Personality detection model.

Type here to search

19:26 05-06-2021

PERSONALITY PREDICTION SYSTEM

File | C:/Users/sandy/Desktop/report%20of%20project/PERSONALITY%20PREDICTION%20SYSTEM%20(4).pdf

72 of 80

Page view | Read aloud | Draw | Highlight | Erase

PERSONALITY PREDICTION #1 - COVER LETTER

Step 11(a)- Personality Prediction 1 - cover letter

```
In [56]: my_posts = """ Hi I am 21 years, currently, I am pursuing my graduate degree in computer science and management (Mba Te

# The type is just a dummy so that the data prep function can be reused
mydata = pd.DataFrame(data={'type': ['INFJ'], 'posts': [my_posts]})

my_posts, dummy = pre_process_text(mydata, remove_stop_words=True, remove_mbti_profiles=True)

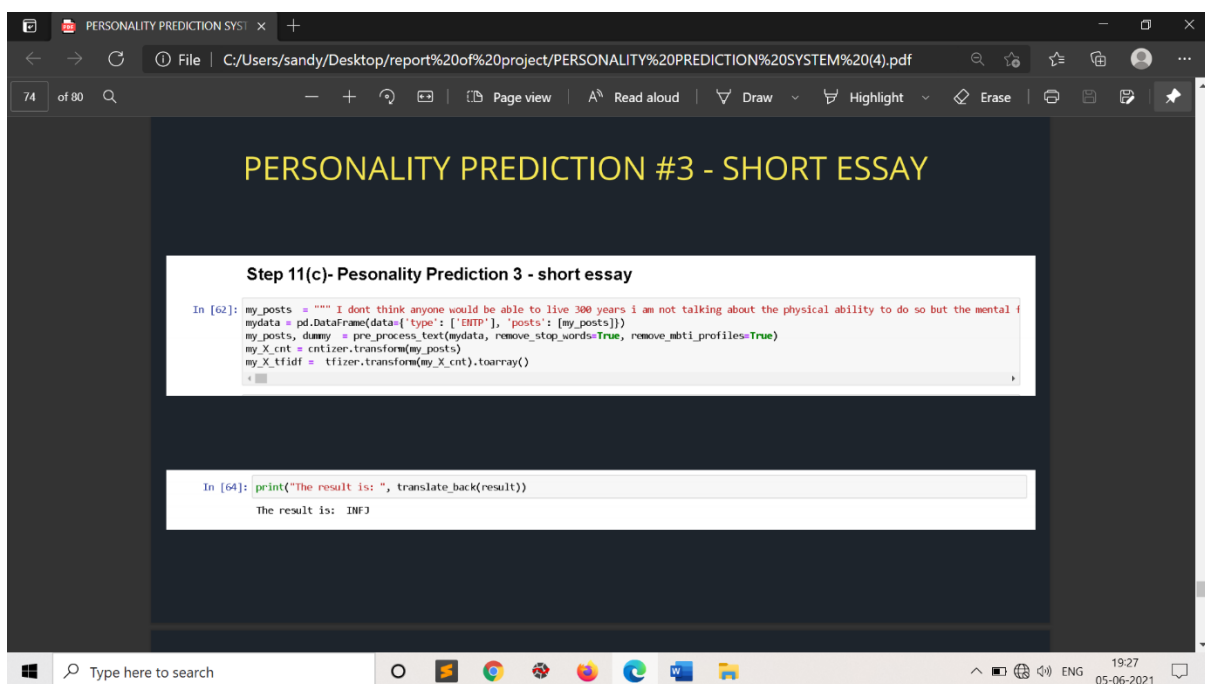
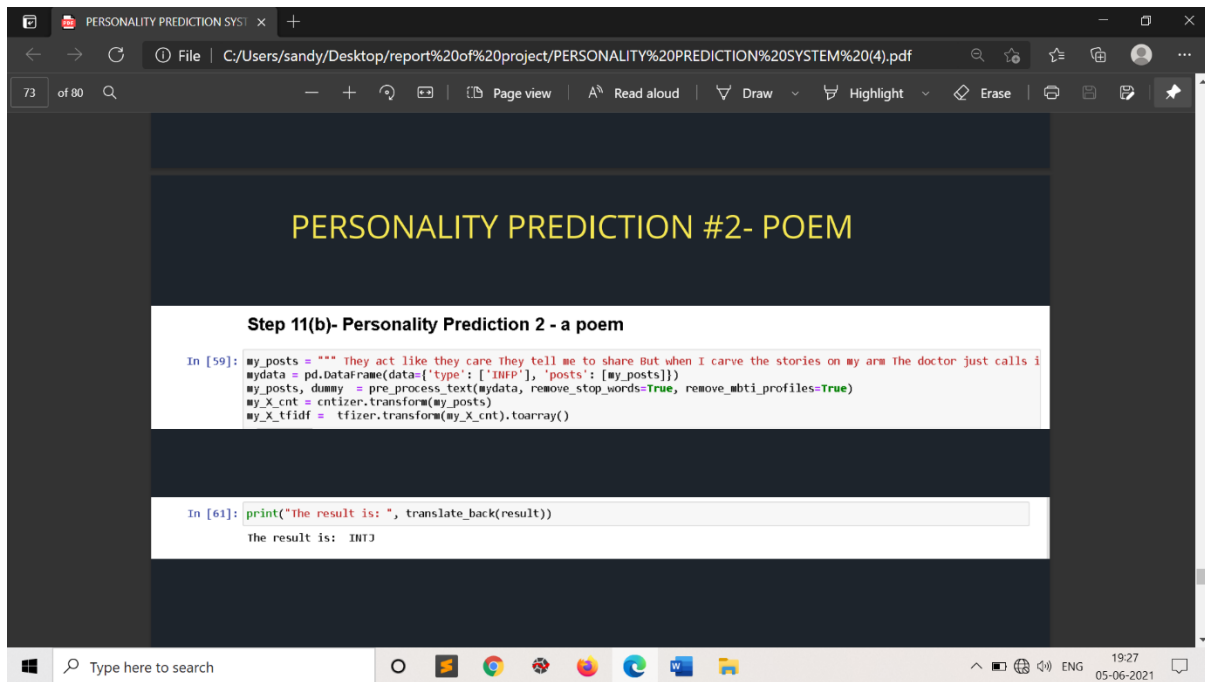
my_X_cnt = cntizer.transform(my_posts)
my_X_tfidf = tfizer.transform(my_X_cnt).toarray()
```

```
In [58]: print("The result is: ", translate_back(result))

The result is: INFJ
```

Type here to search

19:26 05-06-2021



5) CONCLUSION-

This paper provided an insight on existing attempts of the task of personality prediction from text on social media to-date, along with the various kinds of social medias which have been utilized for said task. While most personality prediction studies to-date require a dataset to perform supervised learning, it is costly to obtain a dataset labelled with personality traits of social media users. Recent studies have tried applying semi-supervised and unsupervised learning to tackle this problem. Further improvements to the existing state of personality prediction can be made by expanding the target language, applying more suitable algorithms

or preprocessing methods to achieve higher accuracy, and implementing said task to other personality models that can be used to predict a person's personality with an accuracy of 85.81%. Used to identify the right candidate to the right candidate based on his personality and skill.

Behaviour on Social media sites of users can help in predicting the traits of User based on various personality models. Earlier questionnaire method was used that could be a Costly and time-consuming process. The goal of this paper is to give summary of the work done for Predicting the personality on text from Social media sites and to summarize the future trends. Table I Shows the Overview of the Current research techniques Performed analysis shows the Various techniques and models used. Working on the future directions, accuracy can be increased of prediction as well as can be used to provide some Customized services and other recommendations.

6) REFERENCES-

- Diener, E. and Lucas, R. (2019). Personality Traits. [online] Noba. Avail- able at: <https://nobaproject.com/modules/personality-traits> [Accessed 30 Sep. 2019]
- <https://www.geeksforgeeks.org/overview-of-personality-prediction-project-using-ml/>
- <https://www.kaggle.com/c/twitter-personality-prediction/overview>
- https://www.sas.com/en_in/insights/analytics/machine-learning.html#:~:text=Machine%20learning%20is%20a%20method,decisions%20with%20minimal%20human
- https://www.researchgate.net/publication/316254176_Exploring_personality_prediction_from_text_on_social_media_A_literature_review
- <https://www.javatpoint.com/machine-learning>
- <https://www.ijert.org/personality-prediction-from-social-media-text-an-overview>