# PROJECT REPORT

## PROJECT : PERSONALITY PREDICTION

DIVYANSHI SINGH BORA (B20EE018)

- ***QUESTION 01***: The Myers Briggs Type Indicator is a personality type system that divides a person into 16 distinct personalities based on introversion, intuition, thinking and perceiving capabilities. You need to identify the personality of a person from the type of posts they put on social media.
- ***About the dataset:***
  - The Myers Briggs Type Indicator (or MBTI for short) is a personality type system that divides everyone into 16 distinct personality types across 4 axis:
    - Introversion (I) – Extroversion (E)
    - Intuition (N) – Sensing (S)
    - Thinking (T) – Feeling (F)
    - Judging (J) – Perceiving (P)
  - This dataset contains over 8600 rows of data, on each row is a person’s:
    - Type (This persons 4 letter MBTI code/type)
    - A section of each of the last 50 things they have posted (Each entry separated by "|||" (3 pipe characters))

```
Index(['type', 'posts'], dtype='object')
Dataframe has 8675 rows and 2 columns
the column has: type and posts
```

  - the number of null values in the dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8675 entries, 0 to 8674
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   type    8675 non-null   object
 1   posts   8675 non-null   object
dtypes: object(2)
memory usage: 135.7+ KB
None

munber of values that are null
type     0
```

```
posts    0
dtype: int64
```

- the different types of output

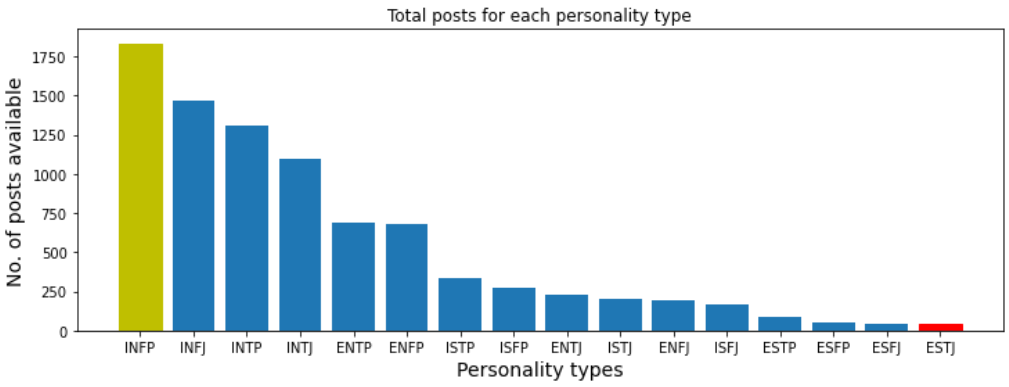
```
'INFJ', 'ENTP', 'INTP', 'INTJ', 'ENTJ', 'ENFJ', 'INFP', 'ENFP',
'ISFP', 'ISTP', 'ISFJ', 'ISTJ', 'ESTP', 'ESFP', 'ESTJ', 'ESFJ'
```

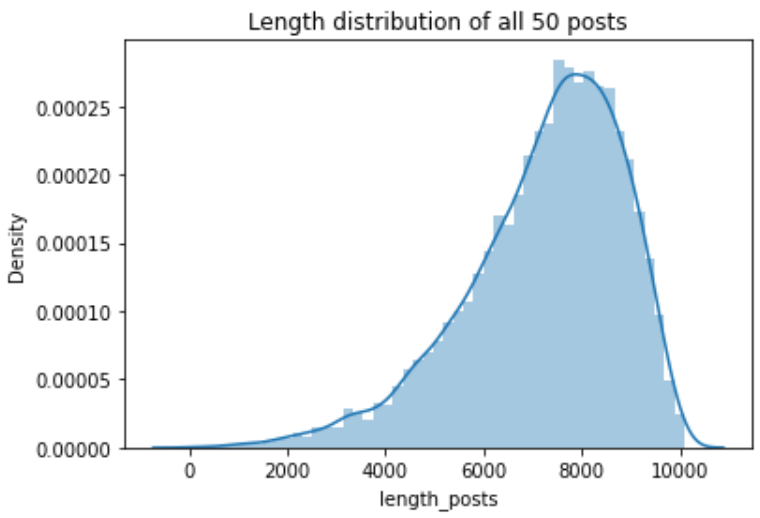
- the dataset provided here looks like this

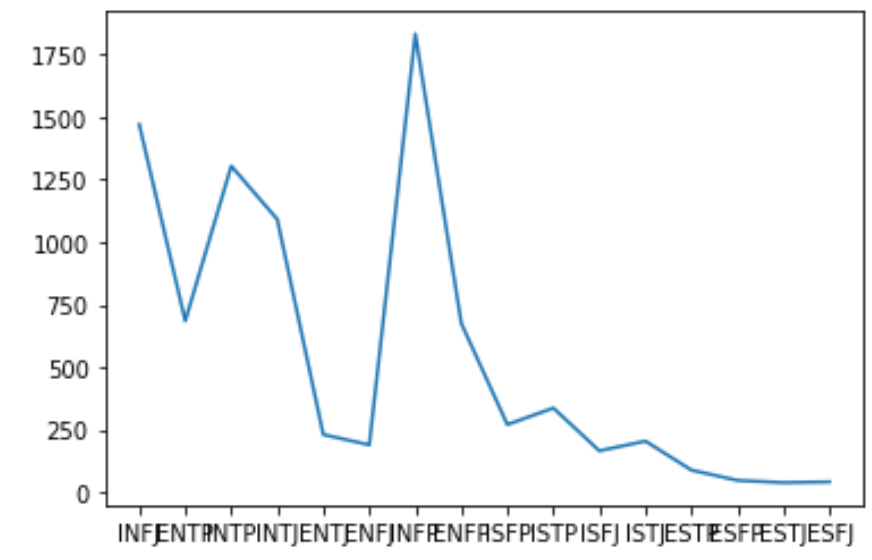
| | type | posts | length_posts | type of encoding |
|---|---|---|---|---|
| **0** | INFJ | 'http://www.youtube.com/watch?v=qsXHcwe3krw\|\|\|... | 4652 | 8 |
| **1** | ENTP | 'I'm finding the lack of me in these posts ver... | 7053 | 3 |
| **2** | INTP | 'Good one _____ https://www.youtube.com/wat... | 5265 | 11 |
| **3** | INTJ | 'Dear INTP, I enjoyed our conversation the o... | 6271 | 10 |
| **4** | ENTJ | 'You're fired.\|\|\|That's another silly misconce... | 6111 | 2 |
| **...** | ... | ... | ... | ... |
| **8670** | ISFP | 'https://www.youtube.com/watch?v=t8edHB_h908\|\|... | 5011 | 13 |
| **8671** | ENFP | 'So...if this thread already exists someplace ... | 7902 | 1 |
| **8672** | INTP | 'So many questions when i do these things. I ... | 5772 | 11 |
| **8673** | INFP | 'I am very conflicted right now when it comes ... | 9479 | 9 |
| **8674** | INFP | 'It has been too long since I have been on per... | 7418 | 9 |

8675 rows × 4 columns

- ***Visualization of the dataset:***
  - the given dataset is visualized with the help of the library matplotlib.
  - the graphs shows that how each personality trait has its distribution in the given dataset.




| | IE | NS | TF | JP |
|---|---|---|---|---|
| **0** | 1 | 1 | 0 | 1 |
| **1** | 0 | 1 | 1 | 0 |
| **2** | 1 | 1 | 1 | 0 |
| **3** | 1 | 1 | 1 | 1 |
| **4** | 0 | 1 | 1 | 1 |

```
Introversion (I) -  Extroversion (E):  6676  /  1999

Intuition (N) – Sensing (S):     7478  /  1197
```

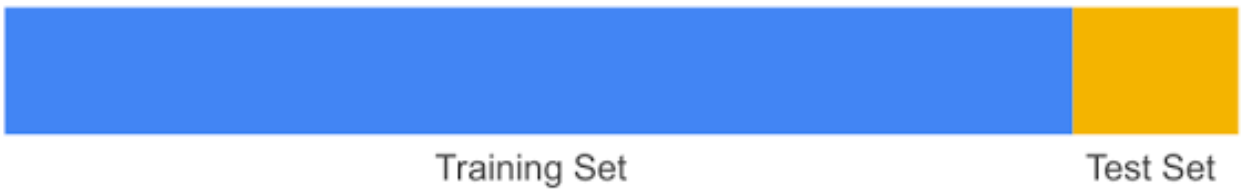
```
Thinking (T) – Feeling (F):    3981  /  4694

Judging (J) – Perceiving (P):    3434  /  5241
```

- ***Data preprocessing:***
  - Tokenize text and return a non-unique list of tokenized words found in the text.
  - Normalize to lowercase, strip punctuation, remove stop words, filter non-ascii characters.
  - Lemmatize the words and lastly drop words of length < 3.
  - In TfidfVectorizer we consider overall document weightage of a word. It helps us in dealing with most frequent words. Using it we can penalize them. TfidfVectorizer weights the word counts by a measure of how often they appear in the documents.
  - after using vectorizer the dataset yielded is given below:

```
  (0, 10184)  0.016234834028634917
  (0, 14391)  0.017904005814618578
  (0, 790)  0.029881381428079272
  (0, 3869) 0.06642775944086109
  (0, 16513)  0.011054077966233629
  (0, 2972) 0.04422895659039223
  (0, 9495) 0.045890756136213844
  (0, 5496) 0.04163794769854996
  (0, 1469) 0.019706876070553533
  (0, 11512)  0.032208854627476156
  (0, 1447) 0.015006292297717744
  (0, 8692) 0.02704911917387719
  (0, 14123)  0.021203016323139957
  (0, 351)  0.01976386629118535
  (0, 16921)  0.013243072200363179
```

- ***Data encoding:***
  - our dataset is preprocessed with the help of the library label encoder.
  - this encodes our ordinal data items into numerical data by encoding them or assigning a unique integer value to each unique data.
  - Label Encoding also refers to converting the labels into a numeric form so as to convert them into the machine-readable form.
- ***Data splitting:***
  - splitting our given dataset into testing and training dataset.
  - the dataset is splitted with 30% of the dataset goes to testing data and 70% of the the dataset to the training part.
  - Training data is the initial dataset you use to teach a machine learning application to recognize patterns or perform to your criteria, while testing or validation data is used to evaluate your model's accuracy.




- ***Training and testing the different model:***
  - model 01: *logistic regression*
    - It is used in statistical software to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation.
    - the accuracy produced after training and test dataset in this model is:

```
Accuracy: 65.27%
```

  - model 02: *support vector machine kernel: linear*

        - Linear Kernel is used when the data is Linearly separable, that is, it can be separated using a single Line. It is one of the most common kernels to be used. It is mostly used when there are a Large number of Features in a particular Data Set.
        - the accuracy produced after training and test dataset in this model is:

```
Accuracy: 65.23%
```

    - model 03: *support vector machine kernel: quadratic*
        - A new quadratic kernel-free non-linear support vector machine (which is called QSVM) is introduced. The SVM optimization problem can be stated as follows: Maximize the geometrical margin subject to all the training data with a functional margin greater than a constant.
        - the accuracy produced after training and test dataset in this model is:

```
Accuracy: 48.33%
```

    - model 04: *support vector machine kernel: rbf*
        - RBF Kernel is popular because of its similarity to K-Nearest Neighborhood Algorithm. It has the advantages of K-NN and overcomes the space complexity problem as RBF Kernel Support Vector Machines just needs to store the support vectors during training and not the entire dataset.
        - the accuracy produced after training and test dataset in this model is:

```
Accuracy: 61.85%
```

    - model 05: *XG boost Classifier*
        - Gradient boosting is a machine learning technique used in regression and classification tasks among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.
        - the accuracy produced after training and test dataset in this model is:

```
Accuracy: 66.69%
```

    - model 06: *Random Forest*
        - Random forest algorithm can be used for both classifications and regression task. It provides higher accuracy through cross validation. Random forest classifier will handle the missing values and maintain the accuracy of a large proportion of data.
        - the accuracy produced after training and test dataset in this model is:

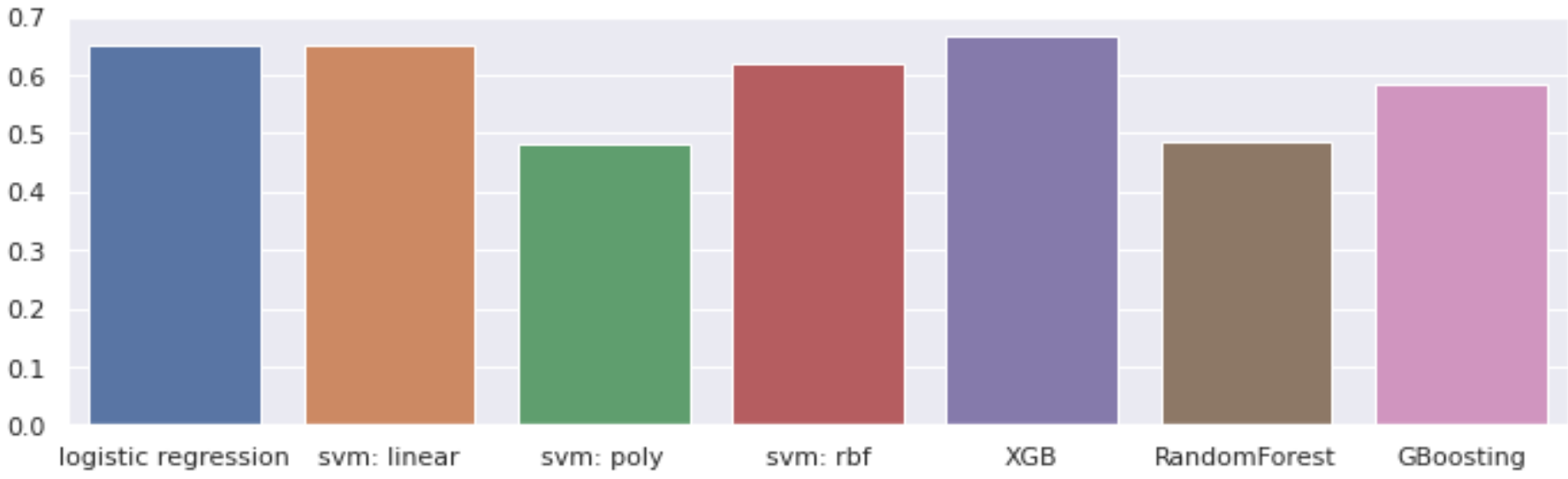
```
Accuracy: 48.67%
```

    - model 07: *GradientBoostingClassifier*
        - Gradient boosting is a machine learning technique used in regression and classification tasks among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.
        - the accuracy produced after training and test dataset in this model is:

```
Accuracy: 58.39%
```

- ***Predicting the best model:***
    - storing all the accuracies in a dictionary and comparing the accuracy of each model until we find the best model which gives us the maximum accuracy.

- to visualize the best accuracy we have used seaborn library to get a better understanding of our dataset and the predictions.




- in the above graph we can see that *XG boost Classifier* has maximum accuracy which is 66.69%.
- we finally conclude that our best model is *XG boost Classifier.*

```
Accuracy: 66.69%
```

- ***Reference:***


1. Sklearn library
2. Data Visualization using Python for Machine Learning and Data science|by sanat|towards data science
3. Cross validation in machine learning | geeks for geeks
4. https://scikit-learn.org/stable/modules/grid_search.html
5. https://analyticsindiamag.com/guide-to-hyperparameters-tuning-using-gridsearchcv-and-randomizedsearchcv/