IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Analysis of Human Traits Using Machine Learning

Ms. Kavita Agarwal Assistant Professor[1], Ms. Vimala Manohara Ruth Assistant Professor[2], Kedarnath Chaturvedula Student [3], Vishal Chandra Jongoni Student[4]

[1]*Department of Computer Science and Engineering,Chaitanya Bharathi Institute of Technology, Gandipet,Hyderabad, 500075,India*
[2]*Department of Computer Science and Engineering,Chaitanya Bharathi Institute of Technology, Gandipet,Hyderabad, 500075,India*
[3]*Department of Computer Science and Engineering,Chaitanya Bharathi Institute of Technology, Gandipet,Hyderabad, 500075,India*
[4]*Department of Computer Science and Engineering,Chaitanya Bharathi Institute of Technology, Gandipet,Hyderabad, 500075,India*

Abstract: Personality is a parameter that makes one individual different from the other individual. Predicting personality has many applications in real world. The main objective of this paper is to take textual data as input from the user and then run the trained machine learning model on this data to predict his 4 personality traits which are Introversion vs Extroversion, Sensing vs Intuition, Thinking vs Feeling, Judging vs Perceiving. The main objective is to build an application where users can answer the questions which are processed and analyzed to output his personality traits. The output is a string of 4 characters where each character determines a personality trait, total of 16 personality types are possible. The Machine learning model XGboost is used to classify the text and output four personality traits. Processing of large textual data is to be done using Natural Language Processing (NLP) techniques with the help of nltk libraries to process and categorize the data. In order to increase the performance of the model hyper parameter tuning along with cross fold validation is done.

Keywords: *Logistic regression, Naive Bayes classifiers, XGBoost, Decision Tree*

1.INTRODUCTION

Personality is what distinguishes the people from one another so it is considered an important parameter. Personality is a key aspect of human life. The study of personality more specifically comes under the branch of psychological study. Personality is constituted of elements like a person's thoughts, feelings, behavior which continuously keeps on changing over time. The Prediction of personality is treated as a classification problem in computer science as the people are classified into the different classes of the personality types. There are a number of psychological tests that yield different types of personality classes. Popular tests include MBTI, Big Five, DISC. The Myers-Briggs Type Indicator (MBTI) is one of the most famous and widely used personality tests or descriptors. It describes the way people behave and interact with the world around them with four binary categories and 16 total types. They are as follows: Introversion vs Extroversion, Sensing vs Intuition, Thinking vs Feeling, Judging vs Perceiving.Understanding personality traits can be very useful as it helps users to discover why people behave in certain ways, the areas in which they can improve and also helps users in finding other people with similar personality traits.

2.LITERATUTE REVIEW

There is significant amount of work has been done in automated personality prediction among researchers in the Natural Language Processing and Social Science fields all over the world. Most of the studies done on personality prediction focused on the Big Five or MBTI personality models, which are the two widely used personality models in the world. The Big Five personality model can be explained as a set of five broad trait dimensions namely, extroversion, agreeableness, conscientiousness, neuroticism and openness. On the contrary, the Myers–Briggs Type Indicator classifies personality types in 16 ways via four dimensions, namely introversion vs extroversion, sensing vs intuition, thinking vs feeling, judging vs perceiving. Classic machine learning techniques and neural- networks have been widely used for predicting MBTI personality types.

One of the studies on personality prediction using machine learning could accurately predict a user's personality type based on MBTI personality type indicator and by considering the information presented on their Twitter. In another study the Naïve Bayes and Support Vector Machine (SVM) techniques were used to predict an individual's personality type based on their word choice. SVM performed better among all these methods. The same database used in previous research, the Myers–Briggs Personality Type Dataset from Kaggle. classification techniques such as logistic regression, Naïve Bayes, Random Forest, K Nearest Neighbour (KNN), Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM) have all been used for personality type prediction based on the MBTI Gradient Boosting is a machine learning technique that has achieved considerable success in a wide range of practical applications because it is highly customizable to the particular needs of the application..

Stop words were removed using natural language processing, then tokenization and stemming were performed to the recovered posts**[1]**. Personality qualities were predicted using AdaBoost. LDA algorithms were utilized to examine algorithm accuracies alongside AdaBoost Multinomial Nave Bayes. F1 score, Precision, and Recall were the metrics used to evaluate the algorithms. The ratio of accurately predicted positive observations to the total predicted positive observations is known as precision. Recall (sensitivity) is the ratio of accurately predicted positive observations to all observations in the actual class. . The highest accuracy obtained was 73.43% , precision of 0.7, and recall of 0.71 and F1-score of 0.72.

The variation of words and the length of sentences were studied, and it was discovered that the data obtained was primitive, with annotations such as.com and.xml, so it was preprocessed and then cleaned to avoid duplicated findings by deleting stop words [2]. Feature extraction was accomplished by encoding the dataset with Tf-Idf vectorization, which reduced the likelihood of keywords appearing in each document. To check the medium frequent words, the N-gram (unigram and bigram) was calculated. Multinomial Nave Bayes, Logistic regression, SVM, and Random Forest models, as well as 5-fold cross validation, were employed for classification. Highest F-Value and Accuracy of 66.59% was achieved using Logistic Regression.

To integrate word forms, the NLTK lemmatizer was employed**[3]**. To standardise, special text (URLs, numerals, dates, and emojis) were recognised using regex and replaced with special escape tokens. The frequency of words in the training set was used to assign numerical indices to them. The 16-class classifier and four binary classifiers were offered as two techniques. The algorithms Naive Bayes and Regularized SVM were utilised. They chose to develop four separate binary classifiers, one for each category, because dividing all 16 groups based on such brief text passages proved to be too challenging. Instead of having a one-size-fits-all model, four independent classifiers can be trained and optimised separately to best suit their respective purposes. The metric employed was accuracy, which is defined as the percentage of true outcomes among the total number of instances studied. Highest Accuracy of 76.1% was achieved using SVM for E/I classifier.

Tokenizing, stemming, and filtering were used to remove stop words during preprocessing[4]. The dataset was then vectorized using Tf-Idf to fit it into the model. After then, the dataset's collection frequency or total number of instances was reduced. The number of features was limited to 750 words from the dataset that appeared the most frequently. To reduce the load and processing time, as well as improve efficacy and accuracy, the number of words was limited. The Nave Bayes Classifier, K-Nearest Neighbors (KNN), and SVM classifier were used to train the model.Highest Accuracy of 72.29% was achieved using MNB.

Preprocessing was carried out first, followed by selective word removal[5]. The lemmatized text was then tokenized using a keras word tokenizer, yielding 2500 most common words. That is, the most common word became 1, the second most common word became 2, and so on until the total number of words reached 2500. Any remaining words in the lemmatized text have been eliminated, leaving the text in the form of integer lists. Because the lengths of the tokenized postings varied greatly, it was important to make them all the same number of tokens long. This was accomplished by padding each tokenized post with exactly 40 integers. If the tokenized post contained less than 40 integers, zeroes were added until it had 40 tokens, and if it had more than 40 tokens, tokens were removed until it had 40 tokens. Because their dataset is made up of sequential text data, they opted to employ a recurrent neural network to capture some of the information that would otherwise be lost in the text data. Highest accuracy of 71% was achieved using various RNN methods.

Our paper can be explained by explaining the following models or functionalities and why they are the best choices to do the desired work.The first step in the implementation is pre-processing of the data. The main objective of this step is to make the raw and unreadable data readable and less gibberish so it can be used to train the models. In every NLP application pre-processing is necessary and it is usually being done by nltk methods. Lemmatizing and stemming followed by converting the text into the list of tokens being done.

The TFIDF Vectorizer is used to encode the data and create the feature vectors. The text data cannot be directly used so it needs to be encoded into the numerical data for this purpose the TFIDF vectorizer is used. Bag of Words is another encoding method available but the reason for going with the TFIDF vectorizer is using the Bag of Words method to create the feature vectors for a large document will result the vectors with large dimension and will contain far too many null values resulting in sparse vectors. Logistic regression is a statistical model that uses a logistic function to model a binary dependent variable in its most basic form, though there are many more complex extensions. Logistic regression is a type of regression analysis that involves estimating the parameters of a logistic model. When compared to other supervised classification techniques such as kernel SVM or ensemble methods (discussed later in the book), logistic regression is relatively fast, but it suffers from some inaccuracy. It suffers from the same flaws as linear regression in that both techniques are overly simplistic when dealing with complex relationships between variables. Finally, when the decision boundary is nonlinear, logistic regression tends to underperform.

Naive Bayes classifiers are a type of "probabilistic classifier" based on Bayes' theorem and strong independence assumptions between features. They are among the most basic Bayesian network models, but when used in conjunction with kernel density estimation, they can achieve higher levels of accuracy. The number of parameters required for Nave Bayes classifiers is linear in the number of variables in a learning issue, making them extremely scalable. In contrast to many other forms of classifiers, maximum-likelihood training can be done simply evaluating a closed-form expression, which requires linear time, rather than by complex iterative estimation.

The most powerful and widely used tool for categorization and prediction is the decision tree. A decision tree is a flowchart-like tree structure in which each internal node represents an attribute test, each branch reflects the test's decision, and each leaf node stores a class label. Decision trees can provide rules that are easy to understand. They accomplish categorization without requiring a lot of processing power. They're capable of dealing with both continuous and categorical data. The training of a decision tree can be computationally expensive. Growing a decision tree is a high computational process. Each potential splitting field must be sorted at each node before the optimal split can be found. Combinations of fields are employed in several algorithms, and appropriate combining weights must be found. Pruning methods can be costly due to the large number of candidate sub-trees that must be produced and analyzed.

The main model of the paper is XGBoost and it is evaluated against other machine learning models like logistic regression, naïve bayes and Decision tree classifier. Among these the performance of the XGBoost model is significantly better. Although

the XGBoost algorithm performs well for a wide range of challenging problems, it offers a large number of hyperparameters, many of which require tuning in order to get the most out of the algorithm on a given dataset.

3.DESIGN OF THE PROPOSED SYSTEM

Block diagram represents all the important steps involved in predicting personality traits are shown in Fig 1. As depicted the process start with the selecting the training data (i.e,Data set Splitting) and it is followed by removing unnecessary symbols numbers and stop words from the data or text and chopping the words down to the root words and creating these words in to numerical data using the TF-IDF vectorizer and passing that feature vector to train the model after doing the hyper parameter training to find out the best parameters and those values that can affect the training process . The last step of the process being generating the personality traits which is also that is to be generated from the model.

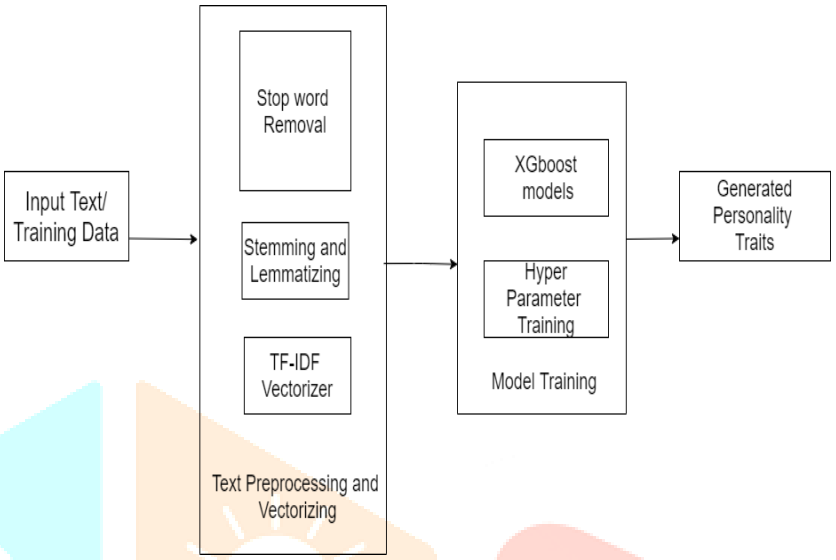

Fig 1. Block Diagram of the proposed system

The proposed system is being developed as three modules as follows:

Front End Module: The Front Module is being developed using HTML, CSS, Bootstrap4. Through the front-end module users access the software. There is a form containing questions followed by the text box. The text entered by the users in all the text boxes is combined to generate a single input text on which the model is run to generate the personality traits. The results are being displayed on the results page where the user personality and few characteristics of that personality type are being displayed.

Integration Module: In this module, the input from the front end module is received and passed to the trained model. The output generated from the trained model is displayed in the front-end module. It is built using flasks. After training the models there are saved as pickle objects and are loaded using the flask framework and being used to predict the model. The results generated are displayed back using flask Jinja templates

Back End Module: The Back End Module is where all the work related to the model training and saving is done. The Data set is loaded and divided into test and training sets. The pre-processing of the data is being done and feature vectors are created. The models are trained and hyper parameter tuning is also done in this module to increase the performance of the model. The trained models are saved here as the pickle objects.

4.IMPLEMENTATION

Pre-Processing:

```
[["'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 p
  '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 re
  'May the PerC Experience immerse you.',
  'The last thing my INFJ friend posted on his facebook before committing suicide the next day. R
  "Hello ENFJ7. Sorry to hear of your distress. It's only natural for a relationship to not be pe
  '84389  84390  http://wallpaperpassion.com/upload/23700/friendship-boy-and-girl-wallpaper.jpg
  'Welcome and stuff.',
  'http://playeressence.com/wp-content/uploads/2013/08/RED-red-the-pokemon-master-32560474-450-33
  "Prozac, wellbrutin, at least thirty minutes of moving your legs (and I don't mean moving them
  "Basically come up with three items you've determined that each type (or whichever types you wa
  'All things in moderation.  Sims is indeed a video game, and a good one at that. Note: a good o
  'Dear ENFP:  What were your favorite video games growing up and what are your now, current favo
  '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
  "Yo entp ladies... if you're into a complimentary personality,well, hey.",
  '... when your main social outlet is xbox live conversations and even then you verbally fatigue
  'http://www.youtube.com/watch?v=gDhy7rdfm14  I really dig the part from 1:46 to 2:50',
```

Fig 2. Raw dataset

The text from the dataset need to be processed before using and it is done in following steps

- Stop words present in the English language are stored in a list.
- Iterate through the entire dataset using the for loop
- From each entry the text is extracted and it is converted into lower case.
- Use the regular expressions to remove numbers, symbols and hyperlinks
- Use the Lemmatizer to create the lexemes from the words
- Check if the sentence contains stop words remove the stop words

The result of the above process is cleaned text.

```
'    moment sportscenter top ten play prank life changing experience life repeat tod
ay may perc experience immerse last thing  friend posted facebook committing suicid
e next day rest peace hello  sorry hear distress natural relationship perfection ti
me every moment existence try figure hard time time growth welcome stuff game set m
atch prozac wellbrutin least thirty minute moving leg mean moving sitting desk chai
r weed moderation maybe try edible healthier alternative basically come three item
determined type whichever type want would likely use given type cognitive function
whatnot left thing moderation sims indeed video game good one note good one somewha
t subjective completely promoting death given sim dear  favorite video game growing
current favorite video game cool appears late sad someone everyone wait thought con
fidence good thing cherish time solitude b c revel within inner world whereas time
workin enjoy time worry people always around yo  lady complimentary personality wel
```

Fig 3. Cleaned Text after Preprocessing

Vectorization/Encoding:

The text data(corpus) consists of so many words and should be summarized to a few keywords only. In the end, we want some method to compute the importance of each word.One way to approach this would be to count the no. of times a word appears in a document. So, the word importance is directly proportional to its frequency. This method is, therefore, called Term Frequency(TF).Few words can appear so many times in a particular document so the important feature cannot be directly selected based on the no.of times it appears in the document, the uniqueness of word should also be taken into consideration and it is done using the Inverse Document Frequency(IDF) method.

461	0.009033	ne
291	0.008526	guy
403	0.007722	lol
267	0.007063	fun
292	0.006774	haha
396	0.006556	listening
685	0.006194	tell
761	0.006189	wink
53	0.006170	awesome
183	0.006107	dream
772	0.005912	world
313	0.005892	hey
477	0.005847	nt
569	0.005841	relationship
551	0.005829	quiet
455	0.005762	music
467	0.005734	ni
139	0.005721	crazy
265	0.005697	fuck
271	0.005654	game
75	0.005644	bored
213	0.005609	everyone
629	0.005543	sometimes
26	0.005496	animal
608	0.005425	shy

Fig 4. Top 25 important Features

Figure 4 shows the top 25 important features from the total list of features.

The text data is converted into the numerical in the following steps:

- Cleaned text to a matrix of token counts using the countvectorizer
- Learn the vocabulary dictionary and return term-document matrix
- Transform the count matrix to a normalized tf or tf-idf representation using TFIDF Transformer
- Learn the idf vector (fit) and transform a count matrix to a tf-idf representation

The result after the above process is the encoded feature matrix .

Training the XGBoost model:

- Early Stopping and Performance Monitoring is done while training XGBoost with each classifier model to avoid overfitting.
- The validation Metric used here is Logarithmic Loss.

- As shown in Figure 5, Early Stopping is done for every 10 rounds if the performance hasn't improved in 10 rounds or else the performance is monitored upto 100 rounds and the best value of the metric is taken and the corresponding accuracies are obtained for each classifier.

```
[96]    validation_0-logloss:0.558403
[97]    validation_0-logloss:0.5582
[98]    validation_0-logloss:0.55801
[99]    validation_0-logloss:0.557773
* FT: Feeling (F) - Thinking (T) Accuracy: 71.78%
JP: Judging (J) - Perceiving (P) ...
[0]     validation_0-logloss:0.684115
Will train until validation_0-logloss hasn't improved in 10 rounds.
[1]     validation_0-logloss:0.676784
[2]     validation_0-logloss:0.670084
[3]     validation_0-logloss:0.665309
[4]     validation_0-logloss:0.661329
[5]     validation_0-logloss:0.65796
[6]     validation_0-logloss:0.655015
[7]     validation_0-logloss:0.651983
[8]     validation_0-logloss:0.649615
[9]     validation_0-logloss:0.647476
[10]    validation_0-logloss:0.645069
```

Fig 5. Early Stopping and Performance Monitoring

- To increase the model's performance further, Hyperparameter Tuning is performed for XGBoost and the best parameter values are obtained.
- These values are then used to train the model in order to achieve the best possible accuracy.
- Hyperparameter tuning is performed by using Grid Search CV method here by using K Fold cross Validation where the dataset is split into k parts, k is 10 for our model training
- The accuracy metrics are used to check the performance of the model along with the roc curve.
- To compare the XGBoost model, Logistics Regression ,Naïve Bayes, Decision Tree Classifier are also trained with the same dataset.

Dataset Description

The data set used in the paper is the mbti dataset which is publicly available in the Kaggle. The dataset contains 8675 rows/entries. There are two columns the first being the personality type which is a word of four letters where each letter represents a personality type and the second column is the combination of 50 social media posts corresponding to that particular user separated with the pipeline character. This information contained in the dataset cannot be directly used as it contains http links, numbers, symbols and stop words which has no impact on the personality of the user. The representation of the personality as said above is not direct; all the traits are combined to generate a single personality type. For better processing of the entries in the binary classification needed one character at a time and it should be as per what model is running and what are the expected traits . In order to make the above process easier four extra columns are added to the dataset where each column is a binary classification result. The dataset obtained is not so balanced either. There is high class imbalance in the data set there are so many entries of the Extrovert compared to the Introvert, number of entries of sensing are significantly more than the number of intuition entries, similarly the number of entries of Judging are more than the number of entries of the Perceiving.

	type	posts
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...
1	ENTP	'I'm finding the lack of me in these posts ver...
2	INTP	'Good one _____ https://www.youtube.com/wat...
3	INTJ	'Dear INTP, I enjoyed our conversation the o...
4	ENTJ	'You're fired.|||That's another silly misconce...
5	INTJ	'18/37 @.@|||Science is not perfect. No scien...
6	INFJ	'No, I can't draw on my own nails (haha). Thos...
7	INTJ	'I tend to build up a collection of things on ...
8	INFJ	I'm not sure, that's a good question. The dist...
9	INTP	'https://www.youtube.com/watch?v=w8-egj0y8Qs||...

Fig 6. Dataset Entries

5 RESULTS

Results of each classifier are displayed and accuracies of each classifier are compared with other models

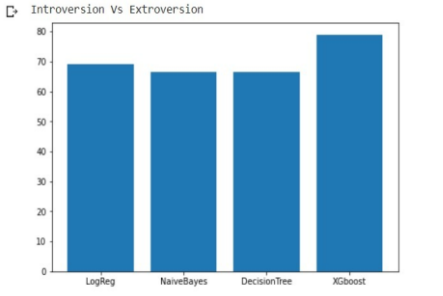

Fig 7. Introversion vs Extroversion Performance Bar Graph

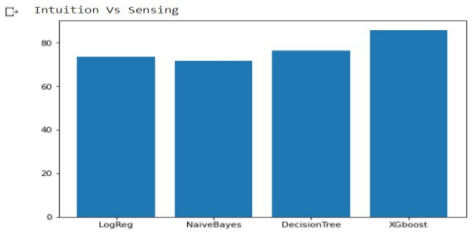

Fig 8. Intuition vs Sensing Performance Bar Graph

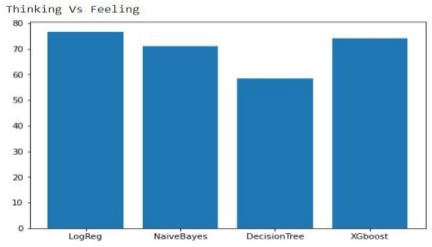

Fig 9. Thinking vs Feeling Performance Bar Graph

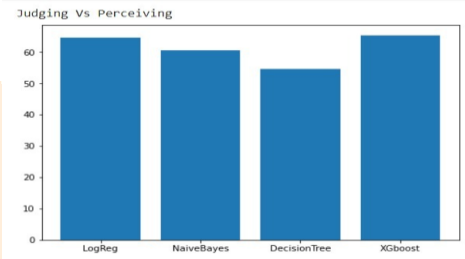

Fig 10. Judging vs Perceiving Performance Bar Graph

Above Figure shows the accuracies received and observed and bar graphs are plotted for the four binary classifiers. For Introversion Vs Extroversion, the highest accuracy was obtained from XGBoost followed by Logistic Regression, Naïve Bayes and Decision Tree. For Intuition vs Sensing, highest accuracy was obtained from XGBoost followed by Decision Tree, Logistic Regression and Naïve bayes. For Thinking vs Feeling, Logistic regression gave the highest accuracy followed by XGBoost, Naïve Bayes and Decision Tree.

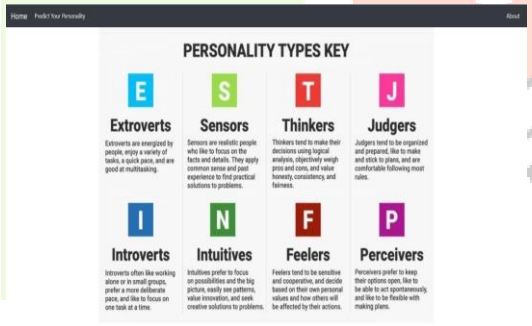

Fig 11: Home page of the website that give brief description about the mbti personality characteristics

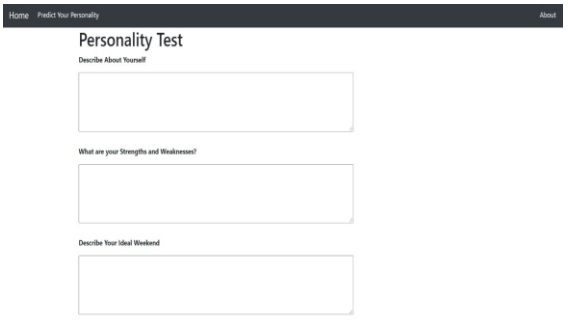

Fig 12: Question Form

The questions form where the user answers the questions. The text entered in all the text area fields are combined to form a single text input against which the model is run.

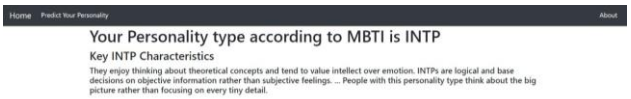

Fig 13 is the result page where the personality type of the user along with what the user might like to do and the characteristics of the user are printed.

6. CONCLUSIONS AND FUTURE SCOPE

The processing of the data is completed using the nltk library which provides the inbuilt list of stop words available in the English language. The regular expressions are used to remove the http links, symbols, numbers. Using the TF-IDF vectorizer feature vectors are created with the low computational overhead. The proposed system for the personality prediction using the XGBoost yielded the best performance. The XGBoost model outperformed the logistic regression model, Naïve bayes Classifier and Decision Tree Classifier. The performance of the XGBoost is increased after the hyper parameter tuning which further increased the accuracy difference between the XGBoost and the other models. The highest of the XGBoost model is 86 for the classifier Intuition vs Sensing.The dataset is highly imbalanced that limits the performance of the system. The system predicts the results purely based on the answers entered by the user at the current time so the user's presence of mind while answering the questions plays a greater impact on the result.

In future the proposed system can be implemented in the real time applications like career advices where one can take advice like what are the best career choices for a person with a particular personality traits for example an extrovert can be good at jobs involving lot of communication whereas the introverted people might be comfortable with jobs involving minimal communication , movie / music recommendations where a particular personality type people can like a certain genre of movies or music, blog websites where the users can meet the similar minds.

REFERENCES

[1] A. V. Kunte and S. Panicker, "Using textual data for Personality Prediction:A Machine Learning Approach," 2019 4th International Conference on Information Systems and Computer Networks (ISCON), 2019, pp. 529-533, doi: 10.1109/ISCON47742.2019.9036220.

[2] Brandon Cui, Calvin Qi, “Survey Analysis of Machine Learning Methods for Natural Language Processing for MBTI Personality Type Prediction”, Stanford,2018

[3] Hernandez, Rayne, Knight, Ian Scott, “Predicting Myers-Briggs Type Indicator with text classification”,31st Conference on Neural Information Processing System, USA,2017

[4] S. Bharadwaj, S. Sridhar, R. Choudhary and R. Srinath, "Persona Traits Identification based on Myers-Briggs Type Indicator(MBTI) - A Text Classification Approach," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 1076-1082, doi: 10.1109/ICACCI.2018.8554828.

[5] J. Golbeck, C. Robles, M. Edmondson and K. Turner, "Predicting Personality from Twitter," 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, 2011, pp. 149-156, doi: 10.1109/PASSAT/SocialCom.2011.33.

[6] B. Y. Pratama and R. Sarno, "Personality classification based on Twitter text using Naive Bayes, KNN and SVM," 2015 International Conference on Data and Software Engineering (ICoDSE), 2015, pp. 170-174, doi: 10.1109/ICODSE.2015.7436992.

[7] Tommy Tandera, Hendro Derwin Suhartono, Rini Wongso and Yen Lina Prasetio, "Personality Prediction System from Facebook Users", 2nd International Conference on Computer Science and Computational Intelligence, 13–14 October 2017.